tems. She serves very often as a member of committee program of many workshops and conferences (IWPTS, IWTCS, FORTE, CFIP, MMNS, MMM, FIW, NOTERE). She organized several international workshops and conferences (IWPT'93, CFIP'93, FORTE'95, CFIP'96).

[William 92] C. William, Multipoint Multimedia Conferencing, IEEE Communications Magazine, May 1992

[Wolfinger 91] B. Wolfinger and M. Moran, A continuous Media Data Transport Service and Protocol for Real-Time Communication in High Speed Networks, The Second International Workshop on Network and Operating System Support for Digital Audio and Video Heidelberg, Germany, November 1991

[Yin 91] N. Yin and M. Hluchyi, A Dynamic Rate Control Mechanism for Integrated Networks, In Proceedings of INFOCOM91

[Zhang 92] H. Zhang, D. Verma and D. Ferrari, Design and Implementation of the Real-time Internet Protocol, IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems, Tucson, Arizona, February 1992

[Zhang 93] H. Zhang, and D. Ferrari, Rate-Controlled Static -Priority Queuing, in Proceedings of IEEE INFOCOM 1993

[Zhang 95] L. Zhang, et al., RSVP Functional Specification, Working Draft, draft-ietf-rsvp-spec-07.ps, 1995

∞

Biographies

Abdelhakim Hafid is Assistant professor at the Electrical&Computer engineering Dept. and the Computer science Dept., University of western Ontario, and a Research Director of the Advanced Communication Engineering Centre (venture established by UWO, Bay Networks, Bell Canada); he is also an Adjunct Professor at University of Montreal, Department of computer Science. He received his Masters and Ph.D. degrees in computer science from University of Montreal on quality of service management for distributed multimedia applications in 1993 and 1996, respectively. From 1996 to 1997 he was a Researcher Staff Member at the Computer Research Institute of Montreal (CRIM), Telecommunications and Distributed Systems Division, working in the area of distributed multimedia applications. From 1993 to 1994 he was visiting scientist at GMD-FOKUS, Systems Engineering and Methods group, Berlin, Germany working in the area of high speed protocols testing. His current research interests are in QoS, Internet and multimedia networking.

Gregor v. Bochmann is a professor at the University of Montreal since 1972 and holds the Hewlett-Packard-NSERC-CITI chair of industrial research on communication protocols. He is also one of the scientific directors of the Centre de Recherche Informatique de Montreal (CRIM). He is a Fellow of the IEEE and ACM. He has worked in the areas of programming languages, compiler design, communication protocols, and software engineering and has published many papers and some books in these areas. He has also been actively involved in the standardization of formal description techniques for OSI communication protocols and services. From 1977 to 1978 he was a visiting professor at the Ecole Polytechnique Fédérale, Lausanne, Switzerland. From 1979 to 1980 he was a visiting professor in the Computer Systems Laboratory, Stanford University, California. From 1986 to 1987 he was a visiting researcher at Siemens, Munich. His present work is aimed at methodologies for the design, implementation and testing of communication protocols and distributed systems. Ongoing projects include applications to high-speed protocols, distributed systems management and quality of service negotiation for distributed multimedia applications.

Rachida Dssouli is professor in the Département d'Informatique et de Recherche Opérationnelle (DIRO), Université de Montréal. She received the Doctorat d'université degree in computer science from the Université Paul-Sabatier of Toulouse, France, in 1981, and the Ph.D. degree in computer science in 1987, from the University of Montréal, Canada. She has been a professor at the université Mohamed 1er, Oujda, Morroco, from 1981 to 1989, and assistant professor at the Université de Sherbrooke, from 1989 to 1991. She spent a Sabbatical year at NORTEL, Ile des Soeurs. Her research area is in communication protocol engineering and requirements engineering. Ongoing projects include incremental specification and analysis of reactive systems based on scenario language, multimedia applications, conformance testing, design for testability and test of timed communicating sys-

[Ramanathan 93] S. Ramanathan et al., Performance Issues in Designing Network Interfaces: A Case Study, High Performance Networking IV, Liege, Belgium, 1993

[Rangan 93] V. Rangan, Architecture and Algorithms for Digital Multimedia On-Demand Servers, Performance 93 and Sigmetrics 93, L.Donatiello and R.Nelson eds., 1993

[Ravindram 93] K. Ravindram and R. Steinmetz, Transport-Level Abstractions for Multimedia Communications, IEEE INFOCOM, 1993

[Rowe 93] L. Rowe and R. Larson, A Video-On-Demand System, University of California at Berkeley 1993

[Salehi 94] K. Salehi, J. Kurose and D. Towsley, Providing VCR capabilities in Large-Scale Video Servers, ACM Multimedia, October 1994

[Schmidt 96] D. Schmidt and A. Gokhale, The Performance of the CORBA Dynamic Invocation Interface and Dynamic Skeleton Interface over High-Speed ATM Networks, IEEE Globecom'96 Conference, London, England, November 1996.

[Schulzrinne 90] H. Schulzrinne et al., Congestion control by selective packet discarding for real-time traffic in high speed networks, In Proceedings of INFOCOM90, 1990

[Smith 93] J. Smith, C. Brendan and S. Traw, Giving Applications Access to Gb/s Networking, IEEE Network Magazine, July 1993

[Sreenan 92] C.J.Sreenan, Synchronisation Services for Digital Continuous Media, PhD dissertation 1992, University of Cambridge, UK

[Steinmetz 90] R. Steinmetz, Synchronisation properties on Multimedia Systems, IEEE JSAC, September 1990

[Szabo 91] B. Szabo and G. Wallace, Design Consideration for JPEG Video and Synchronized Audio, Unix Summer 91 Conference, Nashville, June 1991

[Tawbi 93] W. Tawbi, F. Horn, E. Horlait and J. Stefani, Video Compression Standards and Quality of Service, The Computer Journal, Vol. 36, No. 1, 1993

[Tawbi 94] W. Tawbi and E. Horlait, Expression and Management of QoS in Multimedia Communication Systems, Annals of Telecommunications, Juin 1994

[Tobe 92] Y. Tobe, H. Tokuda, S. Chou and J. Moura, QoS control in ARTS/FDDI continuous Media Communications, in Proceedings of ACM SIGCOMM 92, 1992

[Tokuda 90] H. Tokuda, T. Nakajima and P. Rao, Real-Time Mach: Towards a Predictable Real-time Systems, USENIX Association Mach Workshop, October 1990

[Topolcic 90] C. Topolcic, Experimental Internet Stream Protocol: Version 2 (ST-II), Internet RFC 1190, 90

[Turletti 94] T. Turletti, The INRIA Videoconferencing System (IVS), Connexions-The Interoperability Report, Vol. 8, No 10, 1994

[Verma 91] D. Verma, H. Zhang and D. Ferrari, Delay Jitter Control for Real-Time Communication in a packet switching network, in Proceedings of IEEE TRICOMM 91, 1991

[Vittal 95] C. Vittal, M. Ozsu, D. Szafron, G. Medani, The Logical Design of a Multimedia Database for a News-On-Demand Application, Technical Report #94-16, The University of Alberta, Canada, 1994

[Vogel 95] A. Vogel, B. Kerherve, G. Bochmann and J. Gecsei, Distributed Multimedia Applications and Quality of Service: - A Survey-, IEEE Multimedia Journal, August 1995

[Volgt 95] C. Volgt, L. Wolf, R. Herrtwich and H. Wittig, HeiRat -Quality of Service Management for Distributed Multimedia Systems, Multimedia Systems Journal, November 1995

[Wall 92] G. Wall, J. Hanko and D. Northcut, Bus Bandwidth Management in a High Resolution Video Workstation, Third International Workshop La Jolla, California, USA, November 1992

[Lazar 96] A. Lazar, K. Lim, and F. Marconcini, Realizing a Foundation for Programmability of ATM Networks with the Binding Architecture, IEEE Journal on Selected Areas in Communications, No. 7, pp.1214-1227, September 1996.

[Liu 73] C. Liu and J. Layland, Scheduling Algorithms of Multiprogramming in a Hard Real-Time Environment, Journal ACM, Volume 20, Number 1, 1973

[Lougher 93] P.Lougher, The Design of a Storage Server for Continuous Media, PhD Thesis, Lancaster University, 1993

[Luther 89] A. Luther, Digital Video in the PC Environment, McGraw Hill, New york, 1989

[Meghini 91] C. Meghini, F. Rabitti and C. Thanos, Conceptual Modeling of Multimedia Documents, IEEE Computer, 1991

[Metzler 92] B. Metzler, I. Miloucheva and K. Rebensburg, Multimedia Communication Platform: Specification of the Broadband Transport Protocol XTPX, CIO, RACE Project 2060, 60/TUB/CIO/DS/A/002/b2, 92

[Miloucheva 92] I. Miloucheva, XTP and ST-II Protocol Facilities for Providing the QoS Parameters of Connection-Mode Transport Services, Research Note TUB-PRZ-W-1029, Berlin, 92

[Miloucheva 94] I. Miloucheva, QoS Management for High Speed Transport Architecture, Workshop on Distributed Multimedia Applications and QoS Verification, Montreal, Canada, June 94

[Miller 93] G. Miller, G. Baber and M. Gilliland, News On-Demand for Multimedia Networks, Proceeding of the ACM Multimedia 93

[Moran 92] M. Moran and B. Wolfinger, Design of Continuous Media Data Transport Service and Protocol, Technical Report, TR-92-019, University of California, Berkeley, 1992

[Nagarajan 93] R. Nagarajan, Quality of Service Issues in High Speed Networks, PhD Thesis, University of Massachusetts, 1993

[Nahrstedt 93] K. Nahrstedt and J. Smith, Application-Driven Approach to Networked Multimedia Systems, In Proceedings of the 18th Conference on Local Computer Networks, CS Press, Los Alamitos, California, 1993

[Nahrstedt 95] K. Nahrstedt, An Architecture for End-to-End Quality of Service Provision and its Experimental Validation, PhD Thesis, University of Pennsylvania, 1995

[Neufeld 93] G. Neufeld, M. Ito, M. Goldberg, M. McCutcheon and S. Ritchie, Parallel Host Interface For an ATM Network, IEEE Network Magazine, July 1993

[Northcut 92] D. Northcut, G. Wall, J. Hanko and E. Kuerner, A High Resolution Video Workstation, Signal Processing: Image Communication, Elsevier, August 1992

[OMG 96] Object Management Group, CORBA 2.0 Specification, OMG Technical Document PTC/96-03-04, USA, March 1996

[OMG 97] Object Management Group RFP Submission, Control and Management of Audio/Video Streams, May 1997

[Parekh 93] A. Parekh and R. Gallager, A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks- the multiple node case, in Proceedings of INFOCOM 93, 1993

[Parris 93a] C. Parris, S. Keshav and D. Ferrari, A framework for the study of pricing in integrated network, International Computer Science Institute, Berkeley, Internal report, 1993

[Parris 93b] C. Parris, G. Ventre and H. Zhang, Graceful Adaptation of Guaranteed Performance Service Connections, Technical Report TR-93-011, International Computer Science Institute, Berkeley, USA

[Pnueli 86] A. Pnueli, Specification and Development of reactive systems, IFIP 86, H.Kugler eds.

[Pryker 91] D. Pryker, P. Peschi and T. Landegem, B-ISDN and the OSI Protocol Reference Model, Third IFIP Conference on High Speed Networks, Berlin, Germany, March 1991

[Race 91] RACE IBC CFS D510, General Aspects of Quality of Service and System Performance in IBC, RIC, Brussels, 91

[Halsall 92] F. Halsall, Data Communications Computer Networks and Open Systems, Addison-Wesley, 1992

[Hanko 91] J. Hanko, E. Kuerner, D. Northcut and G. Wall, Workstation Support for Time-Critical Applications, Second International Workshop Heidelberg, Germany, November 1991

[Hayter 91] M. Hayter and D. Mcauley, The Desk Area Network, ACM Operating System Review, October 1991

[Hehmann 91] D. Hehmann, R. Herrtwich, W. Schulz, T. Schett and R. Steinmetz, Implementing HeiTS: Architecture and implementation Strategy, of the Heidelberg High Speed Transport System, Second International Workshop on Network and Operating System Support for Digital Audio and Video, IBM ENC, Heidelberg, Germany 1991

[Hennessy 90] L. Hennessy and D. Patterson, Computer Architecture: A quantitative Approach, Palo Alto, Morgan Kaufmann Publishers, 1990

[Hodges 89] E. Hodges, R. Sasnett, M. Ackerman, A Construction Set for Multimedia Applications, IEEE Software, January 1989

[Hutchison 94] D. Hutchison, G. Coulson, A. Campbell and G. Blair, Quality of Service Management in Distributed Systems, Technical report MPG-94-02, Lancaster University, UK, 1994

[ISO 10746] ISO/IEC CD 10746-2.3, Basic Reference Model of Open Distributed Processing-Part 2: Descriptive Model, July 1993

[ISO 7498] ISO 7498, Basic Reference Model for Open Systems Interconnection, 1984

[ISO 86] ISO 8072, OSI Transport Service Definition, ISO, Geneva, 86

[ISO 91] ISO/IEC JTC1/SC29/WG12, MHEG Working Group S.5, Information Technology, Coded Representation of Multimedia and Hypermedia information objects

[ISO 93] ISO/IEC JTC1/SC6/WG4 N831, Multimedia Communication Platform: Specification of the Enhanced Broadband Transport Service, 1993

[ISO 94] ISO-QoS, Quality of Service Basic Framework-Outline, ISO/IEC JTC1/SC21/WG1 N1145, International Standards Organization, UK, 1994

[ISO 95] QoS in distributed systems, SC21/WG7/N1034, 1995

[Jung 93] J. Jung and D. Seret, Translation of QoS Parameters into ATM Performance Parameters in B-ISDN, In Proceedings of IEEE INFOCOM 93, San Francisco, 1993

[Kalmanek 90] C. Kalmanek, H. Kanakia and S. Keshav, Rate Controlled Servers for very high speed Networks, In Proceedings of GLOBCOM, 1990

[Kalkbrenner 94] G. Kalkbrenner, T. Pirkmayer, V. Dornik and P. Hofmann, Quality of Service in Distributed Hypermedia-Systems, The Second International Workshop on Principles of Document Processing, Darmstadt, April 1994

[Karmouch 93] A. Karmouch, Multimedia Distributed Cooperative System, Computer Communications, September 1993

[Kerherve 94] B. Kerherve, A. Vogel, G. Bochmann, R. Dssouli, J. Gecsei and A. Hafid**,** On Distributed Multimedia Presentational Applications: Functional and Computational Architecture and QoS negotiation, in Proceedings of High Speed Networks Conference, pp. 1-19,1994

[Keshav 95] S. Keshav and H. Saran, Semantics and Implementation of a Native-Mode ATM protocol Stack, Internal Technical Memo, AT&T Bell Lab, Murray Hill, NJ, January, 1995

[Khanna 92] S. Khanna, M. Sebree and J. Zolonowsky, Real-Time Scheduling in SUN OS 5.0, USENIX Winter Conference, SAN Francisco, January 1992

[Krishnakumar 93] A. Krishnakumar, J. Knener and A .Shaw, HIPOD : An Architecture for High Speed Protocol Implementation, High Performance Networking IV, 1993

[Danthine 92] A. Danthine, OSI95: High Performance Protocol with Multimedia Support on HSLANs and B-ISDN, The Third Joint European Networking Conference, Insbruck, 1992

[Deering 89] S. Deering, Host Extensions for IP multicasting, Internet RFC 1112, August 1989

[Dermler 93] G. Dermler, T. Gutekunst, B. Plattner, E. Ostrowski, F. Ruge and M. Weber, Constructing a Distributed Multimedia Joint Viewing and Tele-Operation Service for Heterogeneous Workstation Environments, IEEE Workshop on Future Trends of Distributed Computing Systems, Lisboa, 93

[Doeringer 90] W. Doeringer, D. Dykeman, M. Kaiserswerth, B. Meister, H. Rudin, R. Williamson, A Survey of Light-Weight Transport Protocols for High-Speed Networks, IEEE Transactions on Communications, November 1990

[Druschel 93] P. Druschel, M. Abbot, M. Pagels, and L. Peterson, Network Subsystem Design, IEEE Network Magazine, July 1993

[Dubois 95] E. Dubois, N. Baaziz and M. Matta, Impact of scan conversion methods on the performance of scalable video coding, SPIE 95

[Feldmeir 93] D. Feldmeir, A framework of Architectural Concepts for High Speed Communications Systems, IEEE JSAC, May 1993

[Ferrari 90] D. Ferrari and D. Verma, A scheme for Real-Time Channel Establishment in Wide-Area Networks, IEEE JSAC, Volume 8, No3, 1990

[Ferrari 92] D. Ferrari, A. Banerjea and H. Zhang, Network Support for Multimedia, Technical report 92-072, International Computer Science Institute, Berkeley, November 1992

[Ferrari 95] D. Ferrari, The Tenet Experience and the Design of Protocols for Integrated Services Internetworks, Multimedia Systems Journal, November 1995

[Friesen 95] J. Friesen, C. Yang, and R. Cline, DAVE: A Plug-and-Play Model for Distributed Multimedia Application Development, IEEE Parallel and Distributed Technology, Summer 1995.

[Gilge 91] M. Gilge and R. Gussella, Motion Video Coding for Packet Switching Networks- An Integrated Approach, In SPIE Visual Communications and Image Processing, 1991

[Golestani 90] S. Golestani, Congestion-free Transmission of Real-Time Traffic in Packet Networks, IEEE INFOCOM, 1990

[Golshani 92] F. Golshani and N. Dimitrova, Design and Specfication of EVA: a Language for Multimedia Database Systems, In Proceedings of the Third International Conference on Database and Expert Systems, Valencia, Spain, 1992

[Gopalakrishna 94]G. Gopalakrishna and G. Parulkar, Efficient Quality of Service in Multimedia Computer Operating Systems, Department of Computer Science, Washington University, Report WUCS-TM-94-04, 1994

[Gutekunst 93] T. Gutekunst, T. Schmidt, G. Schulze, J. Schweitzer and M. Weber, A Distributed Multimedia Joint Viewing and Tele-Operation Service for Heterogeneous Workstation Environments, Workshop on Distributed Multimedia Systems, Stuttgart, 93

[Haas 91] Z. Haas, A Communication Architecture for High Speed Networking, IEEE Network Magazine, January 1991.

[Hafid 95] A. Hafid, J. De Meer, A. Rennoch and R. Roth, QoS Verification Experiments, in Proceedings of International Conference on Multimedia Networking (IEEE Computer Society), Aizu-Wakamatsu, Fukushima, Japan, September 1995

[Hafid 97a] A. Hafid and G. Bochmann, An Approach to QoS Management in Distributed MM Applications: Design and an Implementation, Multimedia Tools and Applications Journal, 1997 (to appear)

[Hafid 97b] A. Hafid, G. Bochmann and R. Dssouli, Quality of Service Negotiation with Present and Future Reservations (NAFUR), Computer Networks and ISDN Systems Journal, 1997 (to appear)

[Hafid 97c] A. Hafid and G. Bochmann, Quality of Service Adaptation in Distributed Multimedia Applications, ACM Multimedia Systems Journal, 1997, (to appear)

[Anderson 90] D. Anderson, Support for Continuous Media in the DASH Systems, Proceedings of the Tenth International Conference on Distributed Computing Systems, 1990

[Anderson 92] D. Anderson, Y. Osawa, R. Govindan, A File System for Continuous Media, ACM Transactions on Computer Systems, November 1992

[Applebaum 90] D. Applebaum, The Galatea Network Video Device Control System, MIT Technical Report 1990

[Armstrong 92] S. Armstrong, A. Freier and K. Marzullo, Multicast Transport protocol, Internet RFC 1301, February 1992

[Banerjea 91] A. Banerjea and B. Mah, The Real-Time Channel Administration Protocol, Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video Heidelberg, Germany, November 1991

[Boecking 94] S. Boecking et al., The Berkom Multimedia Transport System, in Proceedings of SPIE 94, Bellingham, Wash, 1994

[Bochmann 96] G. Bochmann, B. Kerherve, A. Hafid, P. Dini and A. Pons, Architectural Design of Adaptive Distributed Multimedia Systems, Proceedings of the IEEE International Workshop in Distributed Multimedia Systems Design, Berlin, Germany, 1996

[Bochmann 97] G. Bochmann and A. Hafid, Some Principles for QoS Management, Distributed System Engineering Journal, volume 4, No 1, 16-27, 1997

[Bohms 94] K. Bohms and T. Rakow, Metadata for multimedia documents, ACM Sigmod Record, 23, no 4, 1994

[Brauds 93] R. Brauds and S. Zabele, A reliable, Adaptive Multicast Service for High-Bandwidth Image Dissemination, ACM SIGCOMM, 1993

[Campbell 92] A. Campbell, G. Coulson, F. Garcia and D. Hutchison, A Continuous Media Transport and Oerchestration Service, SIGCOMM92, Baltimore, USA, August 1992

[Campbell 93] A. Campbell, G. Coulson and D. Hutchison, A Multimedia Enhance Transport Service in a Quality of Service Architecture, Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, UK, 1993

[Campbell 94] A. Campbell, G. Coulson and D. Hutchison, A Quality of Service Architecture, ACM Computer Communication Review, April 1994

[CCITT I.121] CCITT I.121, Broadband Aspects of ISDN, Blue Book, Volume III.7, Geneva 1989

[CCITT I.350] CCITT Rec. I.350, General Aspects of Quality of Service and Network Performance in Digital Networks, International Telecommunication Union, Geneva, 89

[CCITT R34] CCITT Study Group XVIII, Report R 34, Geneva June 1990

[Chou 94] C. Chou and K. Shin, Statistical Real-time Video Channels over a Multicaccess Network, Proc. High SPIE, 1994

[Clark 87] D. Clark, M. Lambert and L. Zhang, NETBLT: A High Throughput Transport protocol, ACM Computer Communication Review, Vol 17, No. 5, August 1987

[Clark 92] D. Clark, S. Shenker and L. Zhang, Supporting real-time applications in an integrated services packet network: architecture and mechanism, ACM SIGCOMM 92, 1992

[Coulson 93] G. Coulson, G. Blair, B. Stefani, F. Horn and L. Hazard, Supporting the Real-Time Requirements of Continuous Media in Open Distributed Processing, Technical Report 1993, Lancaster University, UK

[Crowcroft 88] J. Crowcroft, K. Paliwoda, A Multicast Transport Protocol, ACM SIGCOMM, 1988

[Cruz 91] R. Cruz, A Calculus for Network Delay, part II: Network Analysis, IEEE Transactions on Information Theory, Vol37, 1991

original configuration to the new one. This adaptation function will increase the user confidence in the service provider and also improve resource utilization.

(4) Future reservation of resources: In most existing systems, the user request for a new instance of an application is made at the time when the application should start. A more flexible resource application is possible if the starting time is decoupled from the time when the request is made; the starting time may be in the future. If, in addition, the duration of the applications is known in advance, more efficient resource allocation may be performed by considering different future starting times; in fact, the starting time may become one of the parameters to be negotiated with the user, at the same level as presentation parameters and the cost. This scheme may also be used to encourage the sharing of resources in the case of video on demand with multicast communications; for instance, upon receipt of a new service request, the QoS available at present and in the future will be determined, where the future times may correspond to the scheduled starting times of the same video requested by other users.

(5) A suitable user interface: An appropriate user interface should provide facilities allowing the user to specify the desired QoS in terms of a set of user-perceived characteristics. It should be expressed in user-understandable language and manifests itself as a number of parameters such as video color quality with values black&white and color, and audio quality with values phone quality and CD (compact disc) quality, or steadiness (absence of flickering/noise) with values perfect and average. This is in opposition with user interfaces, provided by exiting QoS architectures, which allow users to specify their requirements in terms of transport service qualities, such as jitter and loss rate, which determine the steadiness parameter. A suitable interface should also allow the user to set cost requirements and preferences and to perform re-negotiation in the case that the desired QoS cannot be provided or when the user wants to change the presentation qualities during the running application. These functions should be integrated with the functions to control the applications, such as to stop, play and pause a multimedia presentation or to search for a new document to be presented.

A *General framework for QoS management* which includes the above functions has been developed at University of Montreal. A detailed description of this framework can be found in [Hafid 97a, Hafid 97b, Hafid 97c].

7. CONCLUSION

This paper surveyed the QoS issues behind distributed MM applications. MM applications characteristics and requirements are identified; QoS notions are defined. The role of all system components in the provisioning of QoS is described. The need for QoS management to support distributed MM applications is motivated. The different QoS management functions are defined and examples of realizations are presented. Some representative QoS architectures are presented and their limitations are identified. Finally, some references for General Framework for QoS management, which does not have these limitations, are provided.

QoS/multimedia is still an very active research area. Examples of issues that are currently addressed include: Mobile computing/networking and multimedia; Multimedia over Internet (e.g. RTP, RSVP, IP multicasting, etc.); Network and operating system support for multimedia; Distributed multimedia systems: modelling, validation and performances; QoS control and adaptation; QoS routing; etc.

REFERENCES

[Adie 96] C. Adie (editor), A Survey of Distributed Multimedia Research, Standards and Products, http://cuiwww.unige.ch/OSG/MultimediaInfo/mmsurvey/top.html

The second class of architectures concentrate on providing service guarantees with respect to network and end-system resources. They integrate the support of resource management in the network and in the end-systems. The main activity of end-system resource management concerns the scheduling of multimedia streams at the end-system based on application QoS requirements (specified using transport QoS parameters, such as delay and throughput). To meet this goal, real-time scheduling policies and memory (buffer) reservation schemes are used to perform the functions required to support the processing of streams. Examples of those functions are transport protocol stack, e.g. fragmentation and error control, stream synchronization, and data movement.

The third class of architectures concentrate on providing QoS guarantees on an application-to-application basis. That is, the resource management at both end-systems and at the network are based on the input of the application/user in terms of QoS requirements (specified using high level QoS parameters, such as frame rate and video color), and not on the QoS requirements of a single connection (stream or flow). For example, OMEGA provides mechanisms to negotiate, during the call establishment phase, QoS guarantees to all incoming and outcoming connections of the application.

In summary, all the architectures presented above focus on communication systems and operating systems, that is, they concentrate on resource reservation and real-time scheduling issues. The mechanisms and schemes proposed are used in a rather *static manner*, since entities involved in QoS support, e.g. network, sender and receiver, are known before the connection (call) set-up phase. The negotiation mechanisms provides the user with the QoS that can be supported at *the time the service request is made*, and assumes that the service is requested for *indefinite duration*. At last and not least, the adaptation schemes provided react to QoS violations by renegotiating a degraded QoS or simply by aborting the session. For these reasons a General QoS Management.

We can conclude that the basic elements of QoS management have been considered in the approaches mentioned above. However, a number of important functions are not supported by these approaches; in particular, we think that the following functions are important:

(1) The consideration of the cost: QoS management should take into account the cost of the application, i.e., the price the user has to pay. None of the existing QoS architectures considers the cost nor provides mechanisms to handle it. The cost is a key concept that will limit the greediness of the users. Without cost constraints, the users will ask for the best available QoS, thus increasing the blocking probability of the system, since the amount of system resources is finite.Consequently, only a few users will be accepted to use the service (unless the QoS is artificially limited for all users) and the global cost of the system infrastructure will have to be paid by a smaller number of users, thus increasing the cost per application. As a result, only wealthy users will be able to afford the provided services, unless they are subsidized.

(2) Dynamic choice of configurations: In opposition to the static choice provided by current QoS architectures where a single, given configuration of system components is considered for supporting a new instance of an application requested by the user, a flexible QoS management process should consider various suitable configurations to support the QoS requirements of the user. The management process should consider several configurations and select the one which, on the one hand, optimizes the resource utilization and decreases the blocking probability for future user requests, and on the other hand, provides an appropriate level of QoS to the user.

(3) Automatic graceful QoS adaptation: The system should recover automatically from QoS violations without requiring user/application intervention (which is the case for existing approaches). Based on the different system configurations that were originally considered (see point(2) above), the QoS management process could identify a new configuration which might support the initially agreed QoS and perform a user-transparent transition from the

buyer performs the following actions: (1) checks whether there are enough local resources to support the new request, (2) gets information about available network resources (e.g. by communicating with network manager), (3) gets information about resources from the broker-seller at the remote site. After the three operations, the broker-buyer takes the decision to accept or reject the new request. The broker-seller manages local resources, and provides the broker-buyer with information about the resources it can offer for the new request (4).

OMEGA differs from other architectures mainly in the following:

(a) OMEGA includes the application layer: OMEGA provides QoS guarantees for the input of QoS application requirements, where the other architectures provide guarantees for single connection requirements, e.g. flows in QoS-A.

(b) OMEGA provides QoS guarantees to incoming and outcoming connections because an application may need incoming and outcoming calls and not only single connections, for instance, in a tele-robotics application, the operator should be able to send commands to the robot(s), and receive video data from the robot(s).

(c) OMEGA supports sender-initiated reservation and received-initiated reservation, while the other architectures support only sender-initiated or receiver-initiated reservations.

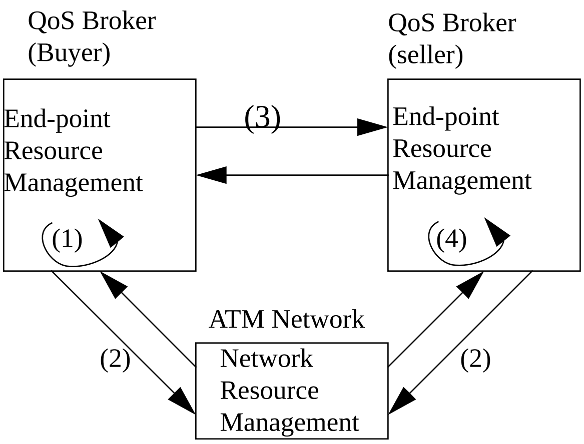

Figure 20. *QoS Brokerage Design*

6.4. Discussion of QoS architectures

The first class of architectures presented above consider network QoS requirements from the application and concentrate on providing guarantees with respect to the network resources. The main activity consists of evaluating the capacity of network resources to accommodate a new connection without affecting the guarantees provided to the existing connections. A connection is characterized by its traffic model, and end-to-end performance requirements. Those architectures provide a range of services to provide deterministic, statistical or best-effort guarantees. These services are provided either using a separate protocol for each service, (e.g. Tenet approach: a protocol for message real-time transport service and a protocol for continuous media transport service), or a single protocol which can be parametrized to allow for a flexible adaptation to the user’s QoS requirements, e.g. XTP allows the transport user to select a retransmission strategy (see Section 3.4).

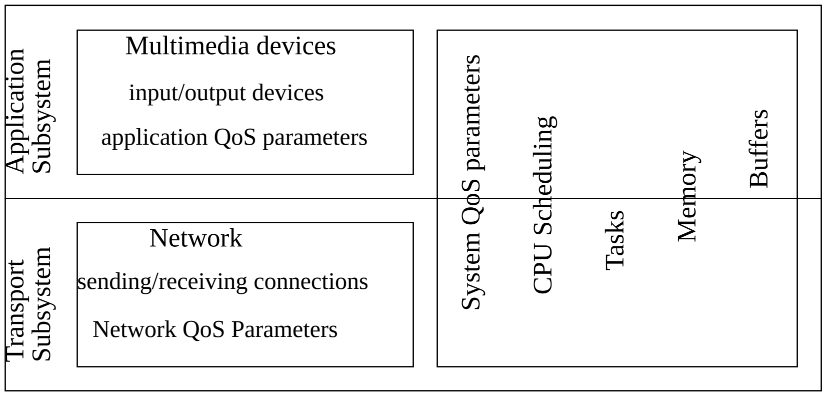

Figure 19. *OMEGA Resource Model*

The communications model is a two-layer system. The transport subsystem includes the functionalities of the network and transport layers; this means that there is no boundary between transport and network functions. Examples of services provided by this layer are connection management,e.g. establishment and termination, and data movement from/to application ring buffers to/from the network host interface.

The application subsystem layer contains the functions of the OSI application and session layers. This layer provides services to support the application requirements. Examples of those services are call (a call may consist of one or more connections, e.g. audio and video connections) management, and input/output device rate control. The service provided at the application subsystem make use of the services provided by the transport subsystem, e.g. call management service uses connection management service.

Both sub-systems provides service guarantees for call/connections for applications. Such guarantees are negotiated by an entity, called the QoS broker, during the call establishment phase. The broker is responsible for the management of local and global end-points resources.

To make guarantees, precise statements of the amount of resources necessary to support a given application are required; this activity is supported by the resource model shown in . More specifically, OMEGA integrates three main components: the application, operating system and the transport sub-system. The application specifies its resource requirements using high-level (application) QoS parameters, such as frame rate and video color; the operating system provides resources, such as CPU slots and buffers, processing and communication; while the transport subsystem provides network resources such as network bandwidth. In summary three logical groups of resources should be managed: multimedia devices, CPU scheduling and memory allocation, and network resources. Since the operating system functions are used by both layers of the communication model, there is no layer boundary in the operating system.

The essence of OMEGA is resource reservation and management of end-to-end resources: OMEGA takes the application requirements in terms of application QoS parameters, e.g. frame rate, for a multimedia call, and reserves required resources in terms of network and end-system resources. This activity is supported by a QoS broker which consists, in a distributed system, of a broker-buyer and a broker-seller (). Upon receipt of a user request, the broker-enough resources to create a new real-time thread and the network layer will determine whether or not it can accommodate a new connection.

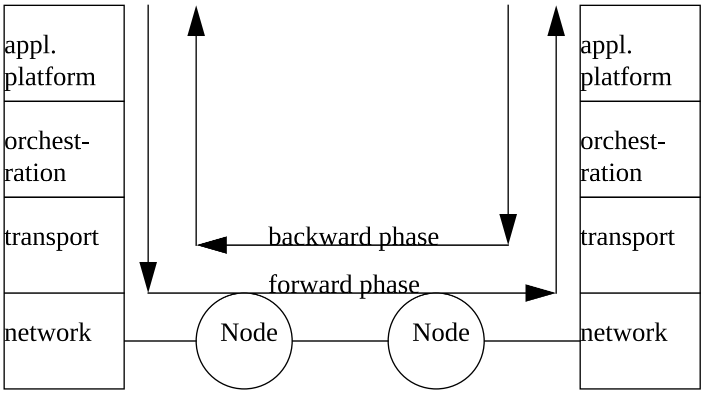

Figure 17. *QoS Negotiation in QoS-A*

6.3. Inclusion of the application layer

The third category of architectures takes into account the application requirements in terms of high-level QoS parameters, such as image size and audio quality, and makes use of resource management to provide end-to-end guarantees. Negotiation between the source and the sink end-systems is provided as part of the session establishment procedure. A significant example of this category is the OMEGA architecture [Nahrstedt 95] which is the result of a research effort which examines the ability of local and global resource management to satisfy QoS requirements of applications. In the following we present a description of the OMEGA architecture.

6.3.1. OMEGA Architecture

The OMEGA architecture is built upon a network which provides bounds on delay and errors, can meet bandwidth demands, and concentrates on the propagation of QoS provision in the end-points [Nahrstedt 95]. The concept of OMEGA is based on the integration of two models, the communication model (Figure 18) and the resource model (Figure 19).

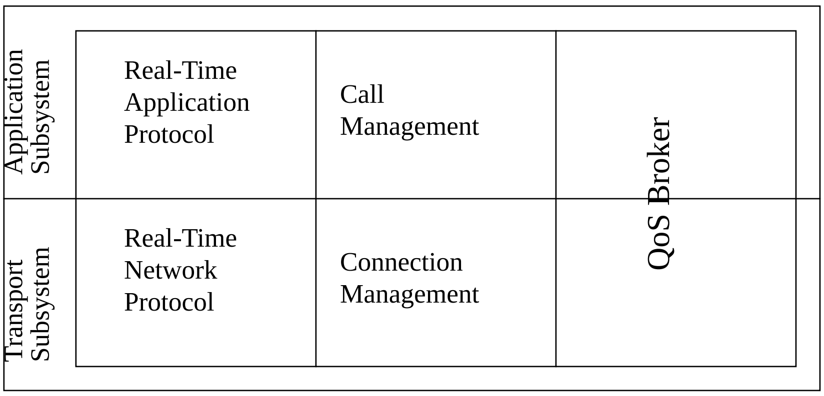

Figure 18. *OMEGA Communication Model*

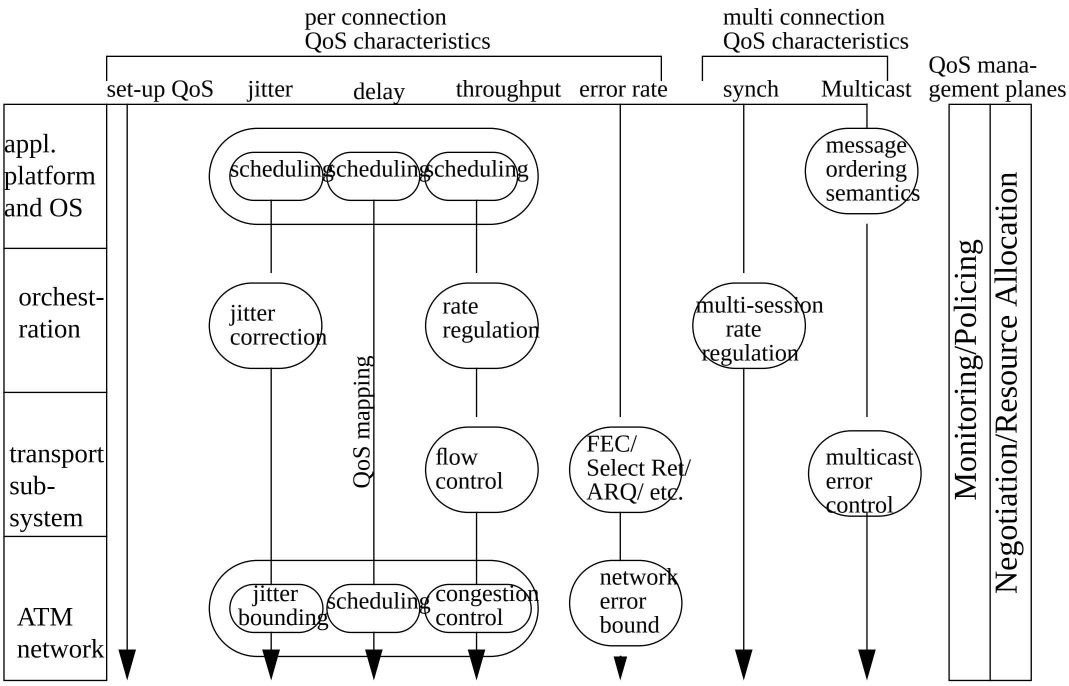

Figure 16. *QoS-A*

The main contribution of (QoS)-A is the multimedia enhanced transport system (METS). METS is designed specifically for continuous media communications. It incorporates buffer sharing, rate regulation, scheduling, and basic flow monitoring. To allocate and adapt the protocol resources, a resource management component is used. The important QoS parameters are: throughput, end-to-end delay, jitter, priority, error rate, and error management profiles.Three classes of service guarantees are provided: deterministic, statistical and best effort.

The communication infrastructure is provided by a multiservice network adapted to continuous media requirements. The principal role of this layer is to provide suitable resource management to provide QoS guarantees. Three classes of guarantees are supported: deterministic QoS, statistical QoS and best effort. an ATM network is selected to support continuous media transmission.

The most fundamental architectural concept used is the notion of flow. A flow characterizes the handling, e.g. transmission, of a stream as an integrated activity. The QoS requirements of the user and the degree of service commitment (guarantee classes) of the provider are formalized in a service contract agreed to by both parties. QoS-A supports flow establishment with a QoS as specified in the service contract, QoS re-negotiation, QoS mapping and QoS adaptation. When a QoS degradation occurs three adaptation actions are possible (1) abort the connection, (2) send a notification of QoS degradation to the user, (3) initiate a re-negotiation.

The process of flow establishment (negotiation protocol) consists of two phases (called forward phase and backward phase, respectively) similar to those of the Tenet protocol suite (). However further network resources, end-system resources are also considered. For example the operating system (upper layer) will check whether it has back towards the source node, and (2) each node frees the reserved resources; otherwise an acceptance message with the remaining excess QoS “the provided QoS minus the requested QoS” is sent back towards to the source node for effective resource reservation. At each node that committed to support the new connection, the level of performance (QoS) is relaxed based on some established policies [Nagarajan 93].

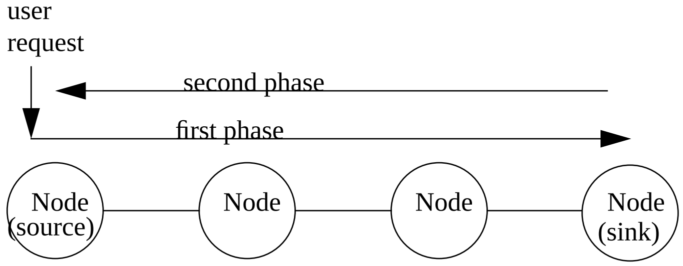

Figure 15. *QoS negotiation in Tenet approach*

Then the network responds to the user with an acceptance or rejection message. To enhance the utilization of the network resources, to increase the number of channel successfully established in the network, and to reduce the time required for the establishment, Tenet approach was augmented with mechanisms to improve the information given to users by the network when a channel request is rejected. Instead of sending a simple rejection to the user, the network will include in the rejection message the best performance offer that can be supported immediately.

6.2. Inclusion of the end-systems

The QoS architectures of this category take into account, in addition to network resource reservation, the resource reservations in the end-systems, such as memory, CPU processing, and network interfaces. The services considered concerns mainly data flows, that is, the goal is to establish connections with QoS guarantees between two or more fixed application entities. Examples of QoS architectures which provide QoS guarantees at the network&transport&end-system level are QoS-A, End-System QoS Framework and Heidelberg QoS Model. In the following we present a description of QoS-A.

6.2.1. (QoS)-A

(QoS)-A [Campbell 94] proposes an integrated view of QoS, including the end-systems and networks. More specifically (QoS)-A is constituted of four layers: application platform and operating system, orchestration, transport subsystem and network. The QoS is described at each layer and mechanisms and functions for QoS support within the layers are defined ().

The highest layer consists of a distributed application platform compatible with ODP enriched with MM communication services, QoS configuration and synchronization functions. Real-time scheduling mechanisms are required to ensure that data which arrives correctly is delivered correctly to its final destination.

The orchestration layer provides services to control jitter and rate regulation for continuous media streams. These services support also multimedia synchronization services across related streams (inter-stream synchronization). Specifically, the orchestration service provides a set of primitives to ensure the synchronization functionalities [Campbell 92].

6.1. QoS guarantees provided by the network and transport layers

The QoS architectures of this category consider network QoS requirements of the application and concentrate on providing guarantees by making use of reservation protocols to reserve resources within the network along the path between the communication entities. These protocols are sender-oriented, or receiver-oriented. Examples of architectures which provide QoS guarantees at the network&transport level are Tenet protocol suite, the Internet protocol stack and the Native-Mode ATM Protocol Stack. In the following we present a description of the tenet approach.

6.1.1. Tenet Approach

The Tenet protocol suite [Ferrari 92] has been designed to establish and run real-time channels in a network, using packet switching (or cell switching) transfer mode, with arbitrary topology. The user specifies the desired QoS in terms of delay, jitter and loss rate (due to buffer overflow) and also the traffic characterization parameters values. The traffic model considered is the simplest traffic model based on the minimum packet inter-arrival time and the maximum packet size. The Tenet approach defines five protocols:

(1) At the network level the Real-Time Internet Protocol (RTIP) [Zhang 92] has been defined. It is concerned with the data transfer.

(2) At the transport level two protocols have been defined:

(2.1) Real-time message transport protocol (RMTP) [Zhang 92] which is concerned with message based real-time transport between endpoints.

(2.2) Continuous media transport protocol (CMTP) [Wolfinger 91] which is concerned with continuous media transport

(3) Two control protocols:

(3.1) Real-time channel administration protocol (RCAP) [Banerjea 91] takes clients requests containing traffic description and performance requirements and sets up real-time channels in the internetwork

(3.2) Real-time message control protocol (RTCMP) is concerned with data transfers control (management), e.g. detection of error conditions.

The Tenet protocol suite has two phases to establish (negotiate) a real-time channel ():

(1) *First phase:* The user issues a request message to establish a real-time channel with values of the parameters characterizing the connection to be established. The request message is exchanged between the network nodes of the route selected between the source and the sink. Each node attempts to reserve resources to accommodate the new connection without affecting the performance of the already existing channels. If it succeeds the request message is forwarded to the next node; otherwise a reject message is sent back towards the source node. Thus the previous nodes must free the resources that they already reserved to support the new connection.

(2) *Second phase:* The sink node verifies that the end-to-end requirements are satisfied by evaluating the levels of performance supported by the different nodes of the route. If the answer is no then (1) a rejection message is sent

- Optimistic approach: The system has to reserve resources based on the average characteristics, such as average bit rate. While the optimistic approach leads in principle to high resource utilization, overload situations may occur, e.g. when all the sources are generating traffic at the peak rate.

Both approaches require mechanisms to reserve a certain amount of resources in terms of CPU slots, buffer space, memory, bus bandwidth, and network bandwidth. The mechanisms proposed in the literature depend on the traffic models and scheduling mechanisms supported, and the characteristics of the system, e.g. single processor or LAN.

The most simple traffic model is one which specifies the minimum interarrival time and the maximum packet size [Ferrari 90]. Another notable traffic model is the so-called linear bounded arrival process (LBAP) [Cruz 91]. It is a two parameter (a,b) deterministic traffic model which requires that the number of arrivals in any time interval t, N(t), be such that $N(t) \leq a+bt$. Examples of real-time scheduling mechanisms are stop-and-go queuing [Golestani 90], Hierarchical Round Robin [Kalmanek 90], Generalized processor sharing [Parekh 93], rate-controlled static priority queuing [Zhang 93], and earliest due date for jitter [Verma 91].

5.11. QoS Termination

When the service is terminated, the reserved resources must be freed. This means that (1) all the resources reserved to support the requested service should be freed, and (2) all the software processes created to support the requested services tasks, such as coding/decoding and sending/receiving data, should be killed.

The termination activity should send a notification to system components involved in the service provision to free the resources and to update the amount of available resources. Such an activity is not trivial, especially if the communication protocol between system components is not reliable. Finally QoS termination facility should be “clean” in order not to affect the existing MM sessions, e.g. a process that is used by more than one session should be killed only when all those sessions are terminated.

6. QOS ARCHITECTURES

Most work within the field of QoS has occurred in the context of individual system components, such as the operating system or the network. However, to support distributed MM applications, the entire distributed system must participate in providing the guaranteed performance levels. In recognition of this, a number of QoS architectures have been proposed to provide QoS guarantees. Examples of those architectures are: the "Quality of Service Architecture" (QoS)-A (Lancaster University) [Campbell 94], the OSI QoS Framework [ISO 94], the Tenet Architecture (Berkeley University) [Ferrari 95], the "Heidelberg QoS Model" (IBM's European Networking Center) [Volgt 95], the "MASI End-to-End Architecture" (Université Pierre et Marie Curie) [Tawbi 94], the "End-System QoS Framework" (Washington University) [Gopalakrishna 94], the Internet Protocol Stack [Zhang 95], the Native-Mode ATM Protocol Stack [Keshav 95], and the OMEGA Architecture (University of Pennsylvania) [Nahrstedt 95]. All these architectures fall into one of the three categories: (1) QoS guarantees at the network&transport level, (2) QoS guarantees at the network&transport&end-system level, and (3) QoS guarantees at the network&transport&end-system level including the application/user layer.

ity in order to verify that QoS requirements can be met for this new request but also that no QoS guarantees are being violated.

Typically admission control activity consists of performing a set of *tests* to determine whether the new service request can be supported without jeopardizing the guarantees given to the existing sessions. The mathematical form of these tests depend on the scheduling mechanisms and the component's architecture, e.g. monoprocessor or multiprocessor, as well as on the traffic model. The tests, if successful, are accompanied by some results which include the QoS the component can commit to, such as the minimum delay and the maximum loss probability due to buffer overflow. If one of the tests is unsuccessful, the component has not enough resources to accommodate the new service request.

An example of admission control tests is described by Liu and Layland [Liu 73]: a set of m periodic tasks with processing time p_i and periods of t_i, for $1 \le i \le m$, is schedulable for deadline driven scheduling if $\sum_{i=1}^{m} \frac{p_i}{t_i} \le 1$ ($d_i = t_i + p_i$ where d_i indicates the deadline for task i). In the following we make use of Liu's and Layland's test to verify whether a component C which consists of a single hardware device, e.g. single CPU, is able to accommodate a new service request. Let us assume that C is carrying a load of $m - 1$ streams such that $throughput_j$ represents the maximum throughput (data units per time units) and st_j represents the service time required to process a data unit of stream j. $\sum_{j=1}^{m} throughput_j \times st_j \le 1$ indicates that C has enough processing power to support stream m, while already supporting the streams $1 \ldots m-1$.

Ferrari et al. [Ferrari 92] defined a set of admission control tests using (1) a traffic model based on the minimum packet inter-arrival time and the maximum packet size, and (2) singular network node running the earliest-deadline first scheduling mechanisms. Such tests allow (1) to verify that a network node has enough processing power to accommodate a new connection (deterministic test), and (2) to evaluate the minimum delay to be assigned to such a connection (delay bound test). Chou et al. [Chou 93] proposed a similar set of admission control tests for multiaccess networks, such as Ethernet and FDDI, using a more general traffic model.

5.10. Resource reservation

Unlike traditional systems, MM systems are required not only to handle a broad range of media types with different requirements, but also to do so while providing a guaranteed QoS to some of those types. QoS requirements for MM applications are typically end-to-end requirements which impose corresponding performance requirements on both the network and the end-systems. Thus to maintain a given level of QoS, a certain amount of network and end-system resources should be reserved.

Two approaches may be used to reserve system resources: an optimistic approach and an pessimistic approach [Nahrstedt 95].

- Pessimistic approach: The system has to reserve sufficient link bandwidth and switching/processing capacity, based on the worst case traffic, to provide the requested service. It is obvious that the reserved resources are severely under-utilized, e.g. due to the bursty nature of compressed video, but a guaranteed QoS is provided.

acteristics. This cost may correspond to the copyright cost associated with the service. For example, different costs may be associated with video-on-demand and tele-conferencing systems. Even more for a given service different costs may be associated with different instances of the service. For example in a tele-conferencing system the cost associated with conferences with the main topic “urbanity” may be different from the cost associated with conferences with the main topic “world wide web”, and in video-on-demand the cost associated with the movie “Jurassic Park” may be different from the cost associated with the movie “Casablanca”.

- *Cost per QoS provided*: It is computed on the basis of the values of the QoS parameters set by the service user. This is directly related to the resources required to provide the requested QoS. In this case, all system components involved in the support of the requested service, such as network and servers, compute the cost based on the resources used and their administration. Obviously the global cost is calculated as the sum of costs computed by system components. For example the cost to provide a movie with <color = color, frame rate = TV frame rate, size = large> is higher than the cost to provide a movie with <color = black&white, frame rate = TV frame rate, size = small>.

- *Service guarantees type*: Deterministic, statistical, best effort, or predictive. It is obvious that a guaranteed service is more expensive than a service provided on best-effort.

- *Duration of the requested service.*

- *Amount of data exchanged*.

- *Time period when the service is provided:* Weekend, monday evening, or Christmas period. For example, we may consider that during night a reduction of 30% is applicable.

- *Advanced service request*: The time the service is provided and the time the service request is made are decoupled, e.g. the user makes his/her service request at 7 am to get the service at 8 pm. This assumes that the system supports future resources reservations as described in [Hafid 97b].

- *Degree of security*: This indicates the effort spent by the system to provide a certain grade of security, e.g. encrypting complexity.

To the best of our knowledge, no proposal has been made to compute the cost to charge to the user in distributed MM systems in general. A general discussion on the cost computation in high speed networks can be found in [Parris 93a]. However, the schemes presented in [Parris 93a] allow (1) to compute only the cost per QoS provided, and (2) assumes the existence of a small number of QoS classes in terms of throughput.

5.9. Admission control

The role of admission control is to compare the resource requirements arising from the QoS associated with a new service request with the available resources in the system. The decision to accept a new service request depends on the system scheduling mechanisms, the characteristics of data traffic to be generated by the new request, as well as resource availability.

Admission control activity is closely related to QoS negotiation and resource reservations activities. Indeed, before making effective resource reservation, admission control mechanisms are used by the QoS negotiation activinitiated only when automatic adaptation is not possible because of serious problems, e.g. some system components crash.

Similar to QoS negotiation, QoS renegotiation should consider the cost parameter. In particular the cost that corresponds to the new QoS should be sent to the user. The transition from the initially agreed QoS and the new QoS is performed only when the user accepts the new cost.

In the literature it is argued that QoS renegotiation uses similar mechanisms as QoS negotiation. However we think that intelligent QoS renegotiation may take advantage of the information produced during the original QoS negotiation phase. That is, during renegotiation with system components, the resources reserved to support the initially agreed QoS, can be considered as available resources.

illustrates an example network consisting of four nodes: A, B, C, and D, and 6 links: A-B:4 Mbits (maximum bandwidth), A-D: 10 Mbits, D-B: 8 Mbits, C-B: 5 Mbits, D-C: 4 Mbits, A-C: 5 Mbits. Dashed lines: D-B and B-C-D represent current MM connections with 7 Mbits and 4 Mbits throughput requirements, respectively. Let us assume that the user initiates renegotiation to change the requirements of B-C-D from 4 Mbits to 5 Mbits. Using a classical renegotiation procedure, in the form of a new negotiation, the system response is a rejection: links B-A and B-C have 4 Mbits and 1 Mbits available bandwidth respectively. Thus there are not enough resources to support the new requirements of 5 Mbits. However if we consider that the resources allocated for B-C-D are free during renegotiation, we can easily find the required resources to support the new requirements: B-C-A-D.

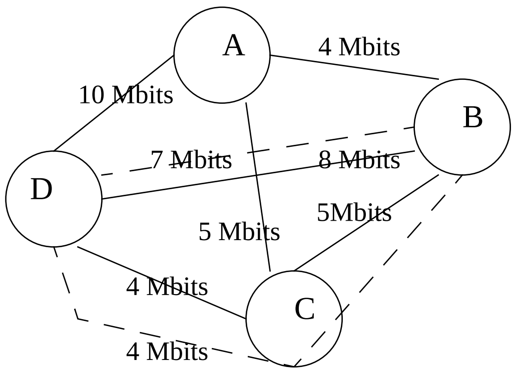

Figure 14. *An Example of Network*

5.8. QoS Accounting

QoS accounting concerns the calculation of the cost relative to the service requested by the user. Since distributed multimedia applications provide a vast range of QoS, QoS accounting is a complex activity.

The cost concept is a key parameter in any QoS management function. Without cost constraints, the users will ask for the best QoS available increasing the blocking probability of the system since the amount of system resources is finite. Consequently, only a few users will be accepted to use the service and thus the service provider will be obliged to increase the cost of its services to get at least the money required to use the resources. In distributed MM applications the cost includes several cost components and depends on several factors, namely:

- *Cost per service type*: It is natural to associate a fixed cost to a given service independently of its dynamic charfollowing characteristics: (1) they are restricted to the communication sub-system, (2) they react only after the occurrence of end-to-end QoS violations, e.g. the problem is passed on to the user or the application, or (3) they react to QoS violations by renegotiating a degraded QoS, and thus they are restricted to applications that can accept a varying QoS. We think that the QoS adaptation activity should be *automatically* performed by the system itself where possible, and only require user/application intervention when built-in strategies fail.

To adapt to variable temporal requirements, such as jitter, buffering mechanisms may be used [Sreenan 92]. Those mechanisms may be inefficient if the variations are important, due to severe resource shortage. Steinmetz suggests [Steinmetz 90] a partial blocking mechanism which allows the specification of actions to be taken in the case of a loss of synchronization (e.g. for a video stream, redisplay the previous frame to deal with lost and late frames). However such a mechanism is useless if the media stream cannot be delivered for a long period of time, such as for voice, since it has no meaning if stopped (e.g. if an audio stream is suspended for more than 10 ms because of congestion, then there is little to do for adaptation).

To adapt to variable data rates, there are mainly three different techniques: interpolation techniques, scalable encoding and dynamic compression ratios.

- Interpolation techniques: Missing data, e.g. video frames and audio samples, can be interpolated from the preceding and following segments. The performance of such techniques depends on the media, video content, and the compression techniques used.

- Dynamic compression technique: Gilge et al. [Gilge 91] describe a scheme which reacts to data rate changes by changing dynamically the coding parameters, e.g. compression ratio. This requires a flexible coding scheme, one that operates at a range of compression ratios, and a protocol between the source and the system, e.g. to compute the available bandwidth and to estimate the quality of reception at the sinks [Turletti 94].

- Scalable encoding techniques: A scalable encoding scheme allows the representation of a media, such as video [Dubois 95], at different levels of resolution (or bit rates) through the use of multiple bitstreams. Such a scheme constructs a base representation (basic quality) of the media, and then produces successive enhancements. The coded video is represented in the form of a set of bitstreams. High quality information, e.g chrominance, is contained in enhancements bitstreams while low resolution information, e.g. luminance, is contained in the base representation bitstream. Scalable encoding techniques react to variable data rates by sending a subset of the available bitstreams, e.g. in the worst case send only the base representation.

5.7. QoS Renegotiation

A renegotiation may be initiated by the user or the underlying system (e.g. communication system). The user-initiated renegotiation allows a user to request a better quality, e.g. a user asks for color quality while the current quality is black&white, or to reduce his/her requirements from the service provider to reduce, for example, the cost of the current session. This type of renegotiation is crucial for applications/users who cannot specify the desired QoS for the whole duration of the service. Examples are medical applications, tele-conferencing systems, and computer supported co-operative work systems where one cannot predict accurately the scenarios to be executed during the active phase of the session.

On the other hand, the system initiated renegotiation is generally due to lack of resources (e.g. network congestion) and aims to reduce the provided quality to avoid a service interruption. Typically such a renegotiation should be

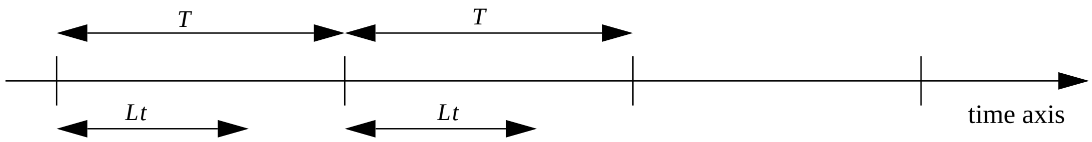

Figure 13. *The Frequency and the Interval of Measurement Parameters*

Many papers raised the importance of QoS monitoring to support MM applications requirement [Hutchison 94, ISO 94, Tawbi 94]. Most existing QoS monitoring protocols are limited to the communication systems and are concerned with transport service monitoring [Miloucheva 94, Danthine 92, ISO 93].

5.5. QoS Policing

We define QoS policing as the set of actions taken by the system to monitor and control user traffic in order to protect the system resources from user misbehavior (malicious or not) which can affect the existing QoS guarantees. In other terms, QoS policing allows to react when users violate the initially agreed QoS, e.g. the agreed throughput is 2 Mbits/s and the user is sending data at 3 Mbits/s. This will help to guarantee the QoS for other sources which keep within the agreed QoS parameters values agreed upon during the establishment phase.

Examples of actions which may be taken by the QoS policing activity are: ignoring the user’s violations, notifying the user, increasing the cost to charge to the user, shaping the traffic to a level that corresponds to the agreed QoS, e.g. if the user is sending at 3 Mbits/s and the agreed throughput is 2 Mbits/s then discard 1 Mbits/s at the source. Ignoring the violations, notifying the user or increasing the cost may be suitable actions when resources are available, while shaping the traffic is required when resource utilization is maximal in order to avoid violating existing guarantees.

More generally, to support QoS policing, monitoring the traffic characteristics of each stream, in terms of bit rate, is required. The traffic generated by the source should be intercepted by the monitoring process to check the agreed QoS before sending it to the recipient via the network.

5.6. QoS Adaptation

The role of QoS adaptation is to maintain, as far as possible, the QoS agreed during the negotiation phase. A QoS adaptation protocol is executed to react adaptively to the changes of the environment, each time some QoS violation is detected.

When the maintenance of the agreed QoS is not possible, the QoS adaptation activity must be able to exhibit graceful degradation, reacting adaptively to changes in the environment. Indeed, it may be more desirable to degrade the quality of the affected service rather than to abort it. More generally the role of QoS adaptation is to keep providing the service, eventually lowering the service quality in case of resources shortage.

Generally, resource overload is solved by application- and user-defined policies. The system must detect (end-to-end) QoS violations by using some monitoring mechanisms, and follow the policies to solve, or react to, overload situations. Most existing approaches [Topolcic 90, Gilge 91, Yin 91, Parris 93b, Tobe 92] have one or several of the

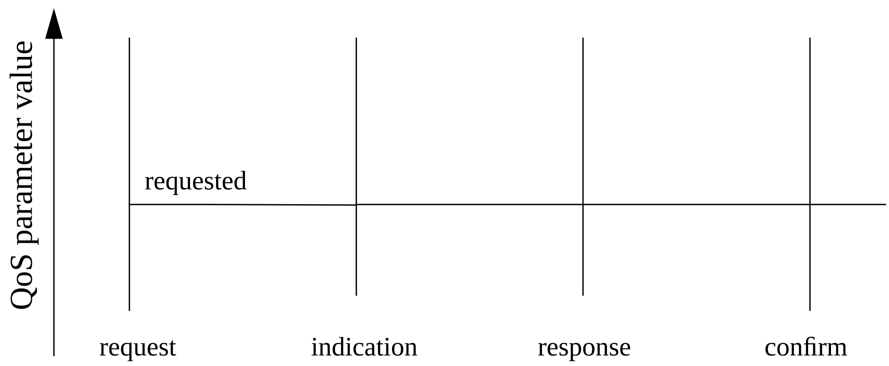

Figure 12. *Unilateral Negotiation*

5.4. QoS Monitoring

Monitoring mechanisms allow to perform a continuous measurement of the QoS which is actually provided. They have mainly two tasks:

(1) To detect and notify any QoS violation (notification task): When the measured value of a QoS parameter does not meet the agreed one, a notification is issued, indicating the violation, and preferably the cause.

(2) To store information (collection task): a description of the information to be selected when monitored and a description of any computations to be performed of the retained information are required.

QoS monitoring is an important activity required in the support of other QoS functions, such as QoS policing, QoS adaptation, QoS renegotiation initiated by the system. Indeed, the system needs to know about the QoS currently provided in order to take the appropriate actions, that is, to activate the appropriate QoS management functions. Monitoring data (collection task) may be computed to produce useful statistics, such as the number of violation of a given type, the time of the last violation of a given type, the mean time between two violations, the time spent to recover from violations, and the most frequent cause of violation. Such statistics may be used for capacity planning, bottleneck analysis, availability component calculation, and system deployment.

QoS monitoring requires QoS measurements. So measurement procedures and methods must be defined; only measurable parameters, e.g. delay and throughput, in opposite to sound quality which is not measurable, must be considered; points where measurements can be performed must be identified [Hafid 95]; the frequency $(1/T)$ of the measurement and the length Lt of the interval over which the QoS parameters are measured must be carefully determined ().

A system claiming to support MM applications should support monitoring facilities; otherwise (1) service guarantees are not possible, (2) QoS violations cannot be detected, and thus, QoS adaptation facility cannot be used, and (3) accurate cost computation is not possible since no information on the components resource utilization is available.

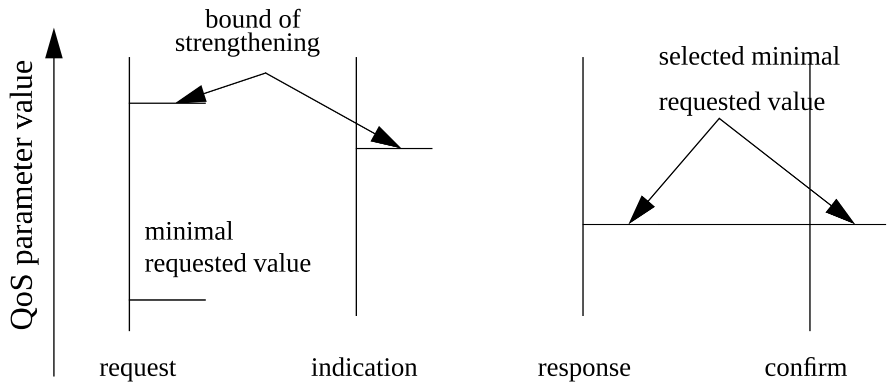

Figure 10. *Triangular Negotiation for a Contractual Value*

5.3.4. Bilateral Negotiation

The negotiation take place only between the service users (). The service provider is not allowed to modify the request value for a QoS parameter.

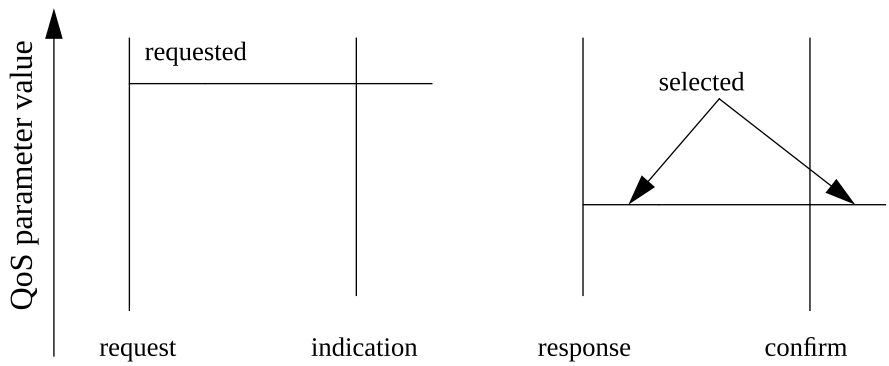

Figure 11. *Bilateral Negotiation*

5.3.5. Unilateral Negotiation

The service provider and the called service user are not allowed to modify the request QoS parameter value requested by the calling service user ().

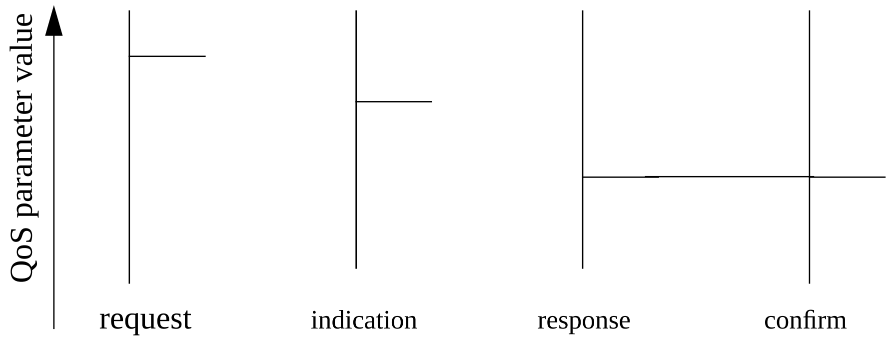

Figure 8. *Triangular Negotiation for Information Exchange*

5.3.2. Triangular Negotiation for a Bounded Target

The calling service user supplies two values for a QoS parameter: target value and lowest quality acceptable value (). The service provider and the called service user are permitted to reduce the proposed target value if it can not be provided. However the reduction must not violate the lowest quality acceptable value.

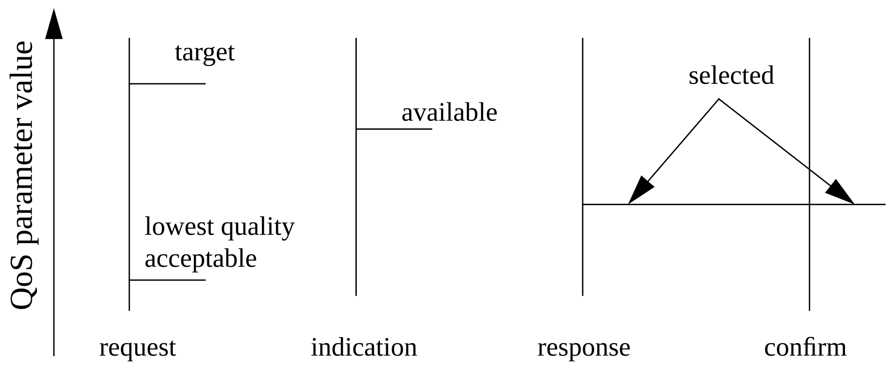

Figure 9. *Triangular Negotiation for a Bounded Target*

5.3.3. Triangular Negotiation for a Contractual Value

The goal is to obtain a contractual value of a QoS parameter which will bind both the service provider and the service users (). The calling service user supplies two values: minimal request value and bound of strengthening value. The latter represents the maximal acceptable value. The service provider and the called service are permitted to reduce the proposed bound of strengthening value if it can not be provided. However the reduction must not violate the minimal requested value.

succeeds, effective reservation of the required amount of resources is performed; otherwise a rejection is sent to the user.

The cost is an important parameter in QoS negotiation. It must be included in any QoS negotiation protocol, otherwise the negotiation may be useless. Indeed without cost constraints, the user will always ask for the best QoS available. We propose that each time the system finds an agreement between the components, in terms of QoS requirements, a notification should be sent to the user before starting the service provision. Such a notification indicates the cost the user should pay for the service provided. The user has three options: (1) accept the cost and then the service provision starts, (2) reject the cost and abandon the service, or (3) initiate a new negotiation to relax his/her QoS requirements ().

Figure 6. *QoS Negotiation with Cost Constraints*

Using the ISO terminology, in the peer-to-peer connection mode service [ISO 7498], the three actors of the negotiation are the service users and the service provider. Four primitives () are exchanged during the negotiation phase.

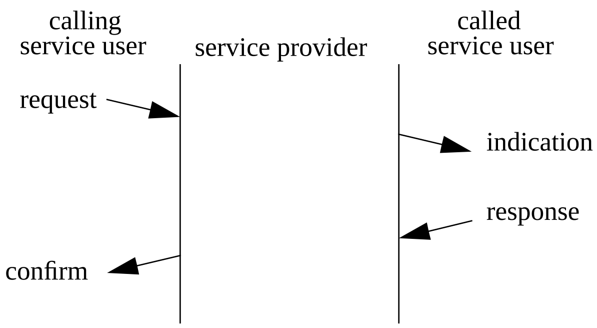

Figure 7. *Primitives Exchanged for QoS Negotiation*

Several variants of this QoS negotiation have been defined [Danthine 92] based on the QoS parameters values exchanged and the behavior of the service provider, as described below.

5.3.1. Triangular Negotiation for Information Exchange

The calling user supplies a QoS value for a QoS parameter using the request primitive (). The service provider can decrease this value before presenting it to the called user using the indication primitive. The called user can also decrease the received QoS value before using the response primitive. At the end, the service provider presents the QoS value, without change, included in the response primitive to the calling user using the confirm primitive.

user to specify the importance of different media (or streams) required for the requested service. In a news-on-demand service, the user should be able to specify that (1) the delivery of audio and video streams is a requirement, that is, the delivery of only one stream is not accepted, (2) the delivery of audio and video streams is desired, however the delivery of only audio stream is accepted, or (3) the delivery of audio and video streams is desired, however the delivery of only video stream is accepted. In [Ravindram 93] it was proposed to use the logical operators (AND, OR, NOT) to specify the atomicity relationships, thus (1) becomes (audio stream AND video stream), (2) becomes ((audio stream AND video stream) OR audio stream), and (3) becomes ((audio stream AND video stream) OR video stream). The atomicity relationships are important at the establishment phase, and may be used by QoS adaptation activities to react to QoS violations.

5.2. QoS mapping

The classical role of QoS mapping activity is the translation between QoS representations at different system levels. Mapping the customer requirements into relevant QoS parameters for service provider is required to support any customer service request. In the OSI model, QoS mapping between the adjacent layers should be performed to process any service request.

More generally QoS mapping is required to allow the service provider to handle and manage meaningful parameter representations. For example the network provider do not know how to handle and manage the frame rate parameter; In opposite it knows how to handle and manage the throughput parameter (packets/s). Thus a mapping of frame rate into throughput is necessary to allow the network provider to support the services requested by the user. Several studies have proposed methods of determining the system requirements, in terms of QoS, from application/user QoS specifications. In [Jung 93] QoS mapping approaches from AAL to ATM layer have been proposed. Moran et al. [Moran 92] describes how the transport layer parameters can be translated into network parameters. The proposals found in the literature focus on deriving QoS parameters between different layers, e.g. OSI layers, in a rather static manner.

We identify three types of mapping:

QoS-QoS mapping: This type of mapping supports the mapping of layer QoS parameters into the next-lower layer QoS parameters. For example it allows to map XTP QoS parameters to ATM QoS parameters.

QoS-resources mapping: This type of mapping supports the mapping of QoS parameters into certain amounts of resources such as buffers, CPU and bandwidth. In others terms it allows to derive the amount of resources required to support the requested QoS. Such an activity is the basis for the resource reservation function.

Service-system mapping: This type of mapping supports the mapping of services onto system components, that is, the system components (hardware and software) required to support the requested service are identified. This mapping is required when we consider services different from flow services, such as services requiring specific processing functions, e.g. synthesizer and image processing. Currently there is no proposal concerning this type of mapping; the existing proposals assume only flow services.

5.3. QoS Negotiation

The role of QoS negotiation is to find an agreement on the required values of QoS parameters between the system and the user(s). The system uses the user QoS specification and performs admission control. If such an activity

3. QoS negotiation to ensure that the system components can meet the requested QoS. To support service guarantees, each component should reserve an amount of its available resources. This requires mapping functions between values of QoS parameters and components resources in terms of bandwidth, buffers and CPU slots.

If the components commit to reserve resources to support the requested QoS, an instance of the application can start. However, during the active phase, the system should monitor the QoS currently provided and take appropriate actions when the agreed QoS cannot any more be supported, e.g. because of network congestion. Depending on the MM application capacity, three actions can be performed: the application session is aborted, a renegotiation is initiated with the user, or the application performs an automatic graceful adaptation.

To ensure that the requirements of the users are satisfied *QoS management* is essential. QoS management functions can be defined as a set of activities that allow the support by the service provider of the desired QoS. QoS management functions include: QoS negotiation, QoS renegotiation, QoS mapping, resource reservation, QoS monitoring, QoS adaptation, QoS accounting, Admission control, QoS policing, QoS related security and QoS termination, as explained in some details below.

5.1. QoS specification

The service users should be able to specify their requirements in terms familiar to them. For example a human user likes to specify video frame rate instead of system related terms, such as megabytes per second, while a transport system user likes to specify reliability constraints in terms of the percentage of lost transport data units instead of the percentage of lost cells. In [Jung 93] QoS parameters were introduced at the ATM and AAL layer, the requirements at the application layer are described in [Campbell 94] in the form of network QoS parameters [Ferrari 90], while QoS requirements at the user are described in [Hafid 97a, Kalkbrenner 94] in terms familiar to human user such as video color, resolution and image size.

More generally the service user specifies values for a list of relevant QoS parameters. Required values may be expressed in terms of upper and lower limits, target values or a variety of other forms, such as a list of discrete acceptable values. In [Danthine 92] a new semantic for the QoS parameters, at the transport level, has been introduced: a parameter is seen as a structure of three values, respectively called *compulsory, threshold* and *maximal* quality, all those values being optional. When a *compulsory* value has been selected for QoS parameter, the service provider will monitor this parameter and aborts the service when it cannot still provide the requested service. When a *threshold* value has been selected for QoS parameter, the service provider will monitor this parameter and notifies the user when it cannot still provide the requested service. The *maximal* value has the same semantic as an upper limit value, that is, the system will avoid to violate this value. The *maximal* value may be used for cost reasons to prevent the service provider from using too many resources (and thus from charging high cost).

Another approach to QoS specification is the use of QoS classes. Schill et al. define four QoS classes that can be automatically mapped onto QoS parameter values. Those classes divide application requirements to reliable, unreliable, time dependent, and time-independent aspects (class 1: reliable and time dependent, e.g. sensor data in a tele-robotics application; class 2: reliable and time-independent, e.g. file transfer; class 3: unreliable and time dependent, e.g. video conference; class 4: unreliable and time-independent). We think that this approach for QoS specification is too coarse and inflexible. Indeed the user can select a QoS, while issuing his/her service request, only from a fixed number of QoS classes which may seriously limit his/her QoS choices.

Since a MM session consists of the handling, e.g. delivery and processing, of one or more media and connections, the specification of atomicity relationships may be recommended. Atomicity relationships allow the service

The enhancement of MM database systems with QoS related meta-information is crucial for efficient QoS management. Thus one can select the appropriate objects (or variants [Bochmann 96]), in terms of location, format and initial QoS, that satisfy the user wishes and the system constraints, e.g. bandwidth and decoders.

4.4. User interface

The success of MM applications can be achieved only if the human end-user is satisfied. The applications must enable users to freely and simply create, retrieve, modify, send and receive MM information.

One of the most important issues is the interaction between the user and the system to specify his/her requirements concerning QoS. The user's preferences must be described in terms of a set of user-perceived characteristics of the performance of the service. It should be expressed in user-understandable language and manifests itself as a number of parameters such as video color quality, e.g. black&white and color, and audio quality, e.g. phone quality and CD (compact disc) quality.

A user-friendly user interface may be used to implement an interface allowing the user to set his/her requirements in an easy and efficient way. Such an interface should also allow (1) the user to change his/her preferences during the lifetime of the session, and (2) the system to present to the user (during the establishment phase or during the lifetime of the session) the QoS which might be actually supported [Hafid 97a].

The user interface for QoS negotiation should provide also facilities to store QoS profiles, for future re-use, to avoid making the user repeat the lengthy QoS parameters setting process. Thus when the user starts a session, he/she may select a desired profile if already stored. The selection facility of a profile is supported by naming different profiles, in a descriptive way such as "good quality video" or "free document only". Thus the user has only to set the name of the profile desired via the QoS user interface. A profile can be used further to store values of user QoS parameters, but also (1) general information on the maximum cost the user is willing to pay and the time period allowed to wait to get the service, and (2) specific information such as the maximum duration of the news-article to play in a news-on-demand application, and system components of preference, e.g. a user prefers to use the Bell Canada network.

The system can also display examples of varying quality such as video with different color and resolution, or related audio and video with different synchronization qualities. Thus before storing a profile or starting a new session, the user can display a monomedia or MM document, which is stored locally, of a quality that corresponds to the profile. This process gives a good and clear idea to the user about the profile he/she is selecting.

5. QOS MANAGEMENT

A typical scenario for QoS processing in a distributed MM application involves the following steps:

1. The user asks for a MM service, e.g. video-on-demand, with a desired QoS. Thus the user should specify the values of a set of user QoS parameters, e.g. TV resolution for video and CD quality for audio.

2. Mapping the user/application QoS onto QoS parameters for various system components, e.g. network and end-systems. For example, CD quality for audio is mapped onto certain throughput, jitter and loss rate.

for optimal placement on disk, e.g. to maximize parallelism in the disk, and to offer some guarantees on retrieval to maintain correct playout. For example it does not make sense to read (access) part of a video frame.

- style and granularity of I/O access: The file access techniques, e.g. file transfer, page transfer and byte transfer, supported by conventional file servers are inefficient for continuous media files. A continuous media file server should be able to access data based on specific scenarios, e.g. access schedule, to deliver data in a way to support continuous media requirements. For example the access granularity of continuous media will be one video frame or one second's worth audio (in opposite to page or byte).

A continuous media file server should be designed in a way to support operations, such as play, stop, fast-forward and pause. Thus the availability of resources at the server, such as CPU slots, RAM, and disk bandwidth, must be guaranteed: real-time resource management is required.

QoS parameters of continuous media file servers are similar to those of end-systems. To support these QoS parameters, appropriate disk layout and disk striping are crucial to allow the storage and a *concurrent* efficient access, in terms of delay, to a huge amount of continuous data under different formats, e.g. MPEG or MJPEG video. Disk layout concerns the organization of data blocks on a single disk, while disk striping concerns the writing of data blocks to multiple disks in parallel. Let us note that a large number of continuous media file servers have been designed and implemented [Lougher 93, Salehi 94, Miller 93].

4.3. Database Systems

Database systems are an important component of distributed multimedia applications. They provide facilities to store and concurrently access two types of information:

- multimedia information represents the content of MM objects, such as video, audio, text, data, graphics and still images;

- meta-information concerns information about the representation, the structure, QoS, the storage and the versions [Bohms 94]. Meta-information is used to locate, access, deliver and present MM information.

MM database systems may benefit from traditional database systems in terms of data abstraction, high level access through query languages, application neutrality, concurrency control, fault tolerance, e.g. transaction mechanisms, and access control [Vittal 95].

Research on MM databases has focused on the modelling of MM information [Meghini 91], data manipulation languages [Golshani 92], and strategies for multimedia object storage, e.g. disk layout in continuous media file servers. We think that meta-information has an essential role for the viability of MM database systems. Meta-information about representation indicates the format, coding and compression techniques as well as characteristics associated with those techniques. Multimedia objects have temporal and spatial relationships, such as synchronization and display location of information between captioned text and video. Those relationships should be modelled and stored as part of the structure information. Information about the content, e.g. key words, is required for efficient support of the search facility. The location of MM objects and the space required for storage are pertinent to access specific objects. Meta-information about the versions describes the evolution of multimedia objects. Much work has focused on the use of meta-information to support the searching facility, while less work has focused on meta-information to support QoS management, such as QoS negotiation.

Dedicated systems assume that all system activities are deterministic, which is not the case for the workstation environment where the user can activate or deactivate several MM applications, e.g. a teleconference participant who activates a video-phone application to communicate with his wife at home, in a non-deterministic way.

It is important to have a system which supports guaranteed QoS but it must be flexible and not based on pre-defined information, e.g. specific devices. Such a system must support the requests of QoS changes initiated by a user who wants to increase the QoS of received information or to get a less expensive service, or initiated by the underlying system because of QoS violation. Furthermore it must be able to use different synchronization protocols, different stream types and different coding and compression schemes since, depending on the chosen types, the combination of different requirements must be supported by the system.

At Sun Microsystems Laboratories, the time critical computing project is employing a technique known as time-driven resource management (TDRM) to support critical real-time applications and particularly MM applications [Hanko 911, Wall 92]. According to TDRM resource management (processor allocation, memory allocation, synchronization operations, interprocess communication, input/output operations, etc.) is based upon the requested deadline, importance and expected resources requirements. To study the performance of various scheduling policies (TDRM, first come first served (FCFS), the earliest deadline scheduling (ED) [Liu 73] and static priority scheduling) experimentation has been carried out [Wall 92] upon the HRV (high resolution video) workstation [Northcut 92]. The results suggest that at low workstation load, the four policies are equivalent. As load in the workstation decreased FCFS delivered the poorest results. As overload occurred, ED began to perform worse than FCFS while the TRDM policy provided the best result in terms of total effective bus utilization. Furthermore the TRDM scheme produced superior visual results through the whole experimentation phase.

Other operating systems (OS) have been developed and implemented to support real-time requirements, particularly MM applications requirements. Examples are SUN OS 5.0 [Khanna 92], RT (real-time) MACH [Tokuda 90], DASH [Anderson 90] and extended CHORUS [Coulson 93].

4.2. File systems

File servers provide the user with facilities to create and delete files, modify the content of files, and read and write from files.

Traditional file servers are unsuitable for continuous media because of a number of deficiencies. In both aspects of environment and file attributes, conventional file servers differ in respect to the requirements of continuous media. First, most file servers were designed based on computer systems which are actually obsolete: major design characteristics were constrainted by the “poor” characteristics of these systems, such as RAM, CPU power and disk space. Second, file servers are optimized for traditional file characteristics which are very different from continuous media file characteristics. Examples of those characteristics are the size, information structure, style and granularity of I/O access, file access patterns and volatility [Lougher 93].

- size: the size of conventional files is very small, e.g. the size of tens of Mbits is considered as extremely high, while the size of continuous media files is extremely high, e.g. the size of one minute of uncompressed standard TV is about 1 Gbits.

- Internal information structure: traditional file servers process files as unstructured sequences of characters. Such a processing is not suitable for continuous media files which are complex structures of MM information, e.g. video frames and audio samples. Particularly, a continuous media file server should know about the media structures

4.1.1. Protocol Implementation Architecture

Protocol architectural performance issues have been well studied in [Feldmeir 93]. In [Neufeld 93] it was reported that parallel protocol processing is required in order that the end-systems keep with the improvements in network performance. Some new architecture for protocol development have been proposed, such as the horizontally oriented protocol (HOPS) architecture [Haas 91], and the high speed protocol development (HIPOD) architecture [Krishnakumar 93].

4.1.2. Network Interfacing and Architectural Issues

The host interface attachment is aimed at supporting high throughput (to support the high throughput provided by the new networks) and low delay at low cost. The key implementation decisions have been usually the partitioning of functions between the network interface and the host processor, the signalling between the host and interface, and the level of integration of the network interface to the host subsystem (placement in the host computer architecture) [Smith 93]. Approaches to host interface hardware and supporting software have been largely discussed in [Smith 93]. Several related performance design alternatives to build a network interface were evaluated [Ramanathan 93] based on the performance analysis of a workstation FDDI adapter.

In [Druschel 93] some hardware and software techniques for avoiding CPU/memory bottleneck are discussed. It was argued that the number of times network data traverses the CPU/memory data must be minimized. In the current workstation, the most consuming part of protocol processing is the actual data movement. Hence the reduction of the number of data copies in transit from the user space to the network is required. The problem is with the workstation's memory architecture [Druschel 93]. The memory performance of workstations has not kept up with the improvement in processor performance and network bandwidth. [Hennessy 90] reports that processor performance has improved at a rate of 50 to 100% per year since 1985 while memory performance has improved at a rate of only 7% per year.

The end user will not be able to take advantage of the new high speed networks with the current workstation performance architecture. In [Hayter 91] a novel architecture of the DESK area network (DAN) workstation has been defined. It uses an ATM network as an interconnect for a MM workstation allowing the use of multiple high-bandwidth streams without significant contention on the resources. In conventional workstations, the bus bandwidth problem forces, for example, decompression to be performed on the same physical board as the device [Luther 89, Szabo 91] requiring hardware which is dedicated to each workstation.

4.1.3. Operating System

MM applications must perform activities such as acquisition/presentation, compression/decompression and processing of continuous media within specified time constraints to guarantee the QoS required by the service user. The correctness of real-time data streams depends on timing relationships in the presentation as on the accuracy of the bit stream.

Thus the most important requirements at the operating system level is to provide predictable real-time behavior for the processing of a range of media types. Most real-time conventional systems have been built based on a dedicated system concept [Hanko 911]. The resources are reserved in advance assuming the worst case and a static environment. The interface of such systems provides a set of predefined operations and the corresponding responses which results in an efficient system, but rigid for most applications. Thus when a dedicated system is developed it remains static and the user cannot modify the system characteristic, e.g. by QoS renegotiation.

- scheduling mechanisms used. Examples are stop-and-go queuing [Golestani 90], hierarchical round robin [Kalmanek 90], and generalized processor sharing [Parekh 93].

In the following we describe the operation of ST-II. ST-II [Topolcic 90] provides a connection-oriented guaranteed service for data transport across the Internet network using Heidelberg resource administration technique [Volgt 95] to reserve resources. ST-II consists of two components: the ST Control Message Protocol (SCMP), which is reliable connectionless transport for the protocol messages and the ST protocol which is an unreliable transport for the data. ST-II serves as a framework for the negotiation of QoS parameters for end-to-end connections.

When requesting the establishment of a connection, the user should specify his/her requirements in terms of throughput, delay and reliability (flow specification). Five classes of reliability are defined: (1) class 0: Ignore both bit and packet errors, (2) class1: ignore bit errors, indicate packet errors, (3) class 2: indicate both bit and packet errors, (4) class 3: ignore bit errors, correct packet errors, and (5) class 4: correct both bit and packet errors. The process of the connection establishment (negotiation protocol) consists of two phases similar to those of the tenet protocol suite. A connection establishment message, which includes the flow specification is sent from the origin to the target via the network nodes of the path. Each ST-II agent (by means of its local resource manager) evaluates the establishment message and reserves the corresponding resource capacities. If the reservation fails (due to resource shortage, or the target does not accept the connection) a refuse message is sent back to the origin releasing all reservation made so far; Otherwise the connection message is updated and sent to the next node in the path. The target decides, based on the message received to accept or reject the connection. If it accepts the new connection an accept message is sent to the origin. It is up to the origin to relax the reservations when the nodes made reservations, during the establishment phase, which exceed the origin requirements.

4. QUALITY OF END-SYSTEMS

Quality of service is also important for end-systems. Transport systems allow the transmission of data, while end-systems allow the processing of data, such as encoding/decoding and presenting audio/video to the user. End-systems should process data in real-time to satisfy QoS requirements of MM applications. In the following we present the issues of QoS support in end-systems, file systems and database systems which are specializations of end-systems, as well as the user-system interactions related to QoS issues.

4.1. End-system architecture and operating system

It is obvious that the operating system that run on the workstation is one of the most important factor which can influence end-to-end QoS support of distributed MM applications. In terms of throughput, the new high speed networks are able to support hundreds of Mbits/s which enable the new emerging services. It is the responsibility of the operating system (OS) to turn network throughput into equivalent application-to-application throughput.

End-system-related resources include bus bandwidth, main and virtual memory, CPU, and input/output devices. In terms of QoS parameters, such resources should be reserved (scheduled) in a way to guarantee a certain delay, jitter and throughput.

It was reported [Smith 93, Ramanathan 93] that the main design features to improve the end-system performance are: optimizing the processing functions in the protocol architecture, optimizing the operating system support for data transfer and careful placement of the hardware for network interfacing.

3.5.1. ATM

To integrate all services regardless of their required QoS over a single network the ATM has been recommended by CCITT as the transport vehicle for broadband networks [CCITT I.121, CCITT R34]. ATM is based upon the evolution of two technologies: packet switching and time division multiplexing by modifying them to support the high speed environment [Pryker 91]. It is an efficient multiplexing protocol to allow direct multiplexing of different bitrate services, offering the bandwidth on demand and enabling MM applications. To facilitate the fast switching and minimize the packet processing per switch (node), by using hardware with integrated circuit devices, packets are segmented to small fixed length packets, called cells (a cell has 53 octets of which 5 octets are the cell header).

ATM networks are connection-oriented packet-switching networks. Accordingly, the QoS parameters in ATM networks can be categorized into call control parameters and information transfer parameters. The call control parameters are similar to those defined in the OSI connection mode, e.g. establishment delay (). The information transfer parameters are shown in Table 3.

QoS parameter	Description
cell transfer delay	the elapsed time from which the first bit of a cell leaves the first observation point (OP) to the time the last bit passes the second OP
cell delay variation	the difference between the values of the transfer delay of the cells on a given link
cell loss ratio	the ratio of the number of lost cells over the total number of cells transmitted
cell insertion ratio	the ratio of the number of inserted cells over the total number of cells transmitted
bit error rate	the ratio of the number of errored bits on a link over the total number of bits transmitted on this link

Table 3. *ATM QoS parameters*

It is worth noting that the XTP connection oriented protocol mechanisms have been designed in a such way to facilitate the ‘direct mapping’ of XTP connection QoS parameters to the QoS parameters of ATM virtual channels. Hence ATM networks are suitable subnetworks for the XTP protocol.

3.5.2. Protocols for providing QoS guarantees

To support QoS guarantees, a number of network protocols, e.g. ST-II and SRP, which provide mechanisms for resources reservations have been proposed. They differ mainly on three factors:

- network node characteristics. Examples are stop-and-go network nodes and ATM switches.

- the traffic model supported. Examples are the simple traffic model [Ferrari 90], LBAP [Cruz 91] and the traffic model defined in [Chou 94].

The XTP user QoS parameter for connection oriented mode have been extended with parameters, such as jitter, required to specify the communication requirements of MM applications.

3.4.2. New Characteristics of high speed transport services

The main contributions of the new transport services and protocols can be summarized as follows:

- New QoS semantics: Further best-effort guarantees service [ISO 86], compulsory guarantees service and threshold guarantees service are introduced [Danthine 92]. compulsory guarantees means that the service provider will monitor the QoS and abort the service facility when it notices that it cannot achieve the agreed QoS, while threshold guarantees means that the service provider will monitor the QoS and indicate to the service user(s)) when it notices that it cannot achieve the agreed QoS.

- New QoS parameters: new QoS parameters are considered, namely delay jitter, error selection policy, and type of guarantees. Concerning error selection policy, OSI-95 [Danthine 92] provides the service user with three options: corruptions of content in TSDUs are not accepted, corruption of contents in TSDUs are accepted but have to be indicated, or corruptions of contents in TSDUs are accepted and do not have to be indicated.

- Multipeer communication: multicast and broadcast facilities are supported.

- QoS negotiation: several variants of QoS negotiation have been defined based on the new QoS semantics. OSI-95 defines five QoS negotiation schemes: triangular negotiation for information exchange, triangular negotiation for a bounded target, triangular negotiation for a contractual value, bilateral negotiation, and unilateral negotiation (see Section 5.3).

- Support of QoS monitoring and QoS renegotiation: QoS monitoring is based on some QoS measurement mechanisms to indicate the service user about the “actual” values of QoS parameters depending on the QoS provision classes, e.g. threshold or compulsory.

- Flow control: further window flow control, rate-based flow control is supported [Metzler 92].

- ATM: high speed transport protocols, e.g. XTP, have been designed and implemented to use ATM as subnetwork.

3.5. Resource allocation and ATM

Today’s networks and network protocols are not appropriate to support the new MM applications requirements; the provided services have been designed and optimized for classical applications, e.g. telephony and file transfer.

The link between the networks and the applications are transport protocols; thus, the new generation of networks and network protocols should be able to support the new services and functionalities directly specified by MM applications or via high speed transport protocols. Examples of high speed networks are ATM, FDDI and DQDB; examples of network protocols which provide QoS guarantees are ST-II [Topolcic 90] and SRP [Anderson 90].

- Multipeer data communication, e.g. multicast facility, is not provided.

- The provision of QoS provides only the ‘best effort guarantees’ service which is not appropriate for continuous media that require a predictable QoS. “best effort guarantees” means that there is no guarantee that the agreed QoS will be maintained throughout the connection lifetime, and that changes in QoS are not explicitly signalled by the transport service provider [ISO 86].

- QoS re-negotiation and the handling of QoS violations are not provided.

3.4. High speed transport protocols

A large number of researchers have investigated the provision of QoS at the transport level. Several transport protocols and services have been proposed, some of them have been implemented [Doeringer 90, Campbell 93, Hehmann 91, Miloucheva 92]. The need of new transport protocols to support MM applications comes from the limitations of the traditional transport protocols (see Section 3.3). Examples of high speed services and protocols designed and implemented are the Heidelberg Transport System (HeiTS) [Hehmann 91], the Berkom MM transport service (MMTS) [Boecking 94], and eXpress Transport Protocol (XTP) at Technical University Berlin (TUB) [Miloucheva 92]. In the following we give a short description of XTP, and the new characteristics of high speed transport services. XTP is selected for presentation, because it has been extensively studied in the literature and it is becoming an ISO standard.

3.4.1. XTP

XTP is high performance protocol with the primary goal of hardware realization as a VLSI chipset. It is a protocol that integrates the transport layer functionality and some network layer mechanisms. Hence XTP can handle allocation policies, flow or rate control strategies, resource reservation mechanisms and retransmission strategies. It has been designed to support real-time transactional and multi-media systems. XTP support two kinds of QoS provision: the best effort and the guaranteed services.

The XTP mechanisms developed to support the new emerging services include:

- Rate based flow control. It supports also the window flow control.

- Flexible selection of error control mechanisms. The mechanisms are reliable error control (based on selective retransmission) and no error control.

- Selection of protocol mechanisms and QoS based policing.

- QoS measurement: uses some measurement policies, e.g. measurement based on TSDUs numbers or time intervals, to obtain the ‘actual values’ of the QoS parameters to be monitored.

- QoS monitoring: is based on QoS measurement to indicate to the user the values of QoS parameters depending on the QoS provision classes, e.g. guaranteed or best effort.

- Multicast data transfers

- The use of windows as flow control mechanisms is not efficient because it introduces silence periods and generates bursty traffic.

- The retransmission mechanism used to provide reliable transmission is not adequate for continuous media. Because of real-time constraints of continuous media, the resulting time needed to transfer data correctly from sender to receiver may be too long (thus the data received is likely useless), e.g. a variation of delay of 10 ms between two successive audio samples is too long.

OSI QoS parameter	Description
Throughput	number of TSDUs successfully sent or received during a specified interval
transit delay	elapsed interval between the issuing of T-DATA request and the corresponding T-DATA indication
resilience	the probabilty that a service provider will release the connection within a specified interval of time
transfer failure probability	ratio of the total transfer failures to total transfer samples observed during a performance measurement
residual error rate	ratio of total incorrect, lost and duplicate TSDUs to total TSDUs transferred across the TS boundary during a measurement period
establishment delay	elapsed time between the issuing T-CONNECT request and the corresponding T-CONNECT.confirm
establishment failure probability	probability that a request connection is not established within the specified maximum acceptable establishment
release delay	the maximum acceptable delay between a TS user initiated T-DISCONNECT.request and the successful release of the transport connection at the peer TS-user
release failure probability	the probability that the service provider is unable to release the connection within a specified maximum release delay
cost	a parameter to define the maximum acceptable cost for a network connection
protection	extent to which a service provider attempts to prevent unauthorised monitoring or manipulation of data of connection
priority	the relative importance of the transport connection

Table 2. *OSI Transport QoS Parameter*s

- The difficulty in handling timers which are employed by the error control mechanisms [Clark 87] do not allow to achieve high throughput.

- Transit delay jitter: the variation of the transit delay; several definitions for the delay jitter are used in the literature: (1) jitter is the variance of the transmission delay of a stream, (2) jitter is the difference between the values of the transit delays of the segments of a stream, and (3) jitter is the instant delay variation from the mean.

- Loss rate: the ratio of the number of bits lost to the total of number of bits sent (received at the input). Generally, one talks about packet loss rate or cell loss rate since the stream data is usually sent in segments (blocks of bits).

We note that the throughput is a property of the stream which is transmitted by the network, and not a property of the network [Bochmann 97].

3.3. Traditional transport protocols

Several transport protocols have been designed, implemented and widely used, such as TCP and ISO TP4 [ISO 86]. These protocols have been initially designed to support classical applications, such as FTAM [Halsall 92]. In the following we give a short description of the OSI TP4 and TCP, and the limitations of traditional transport protocols to support MM applications.

3.3.1. TCP and OSI TP4

TCP is a connection oriented, highly reliable and secure transport protocol which is designed to run over an unreliable network service. Reliability, duplication prevention, and ordered delivery of data is accomplished using sequence numbers on bytes and retransmissions in the case of data loss or damage. That is, data (TPDU's) is sent to the receiver, a copy of this data is made to a local buffer, and the sender keeps waiting for an acknowledgment to free its buffers. Positive acknowledgments (from the receiver) serve to indicate (to the sender) the correct delivery of the data, while negative acknowledgments serve to initiate retransmission of data erroneously received. If an acknowledgment is not received (by the sender) after a time out, data will be retransmitted. A checksum allows error detection.

To avoid the receiver congestion, a flow control mechanism using a sliding window is used. With every acknowledgment the receiver sends the number of transport data units it is able to receive. Given this number, the sender regulates data transmission.

The OSI TP4 uses similar mechanisms to those of TCP, however it provides other functions related to QoS. During the establishment phase, the OSI TP4 allows the negotiation of a set of QoS parameters (Table 2) between the sender, the receiver, and the transport service provider. The sender proposes target values, and the receiver may accept any 'lower' value, e.g. lower throughput or higher transit delay. The transport service users (sender or receiver) cannot change, during the lifetime of the connection, the values of QoS parameters negotiated during the establishment phase. It is worth noting that there is no guarantee to support the negotiated QoS, however, the transport service provider will do its best to meet this QoS.

3.3.2. Limitations of Traditional Transport Protocols

The OSI TP4 and TCP, are not suitable to support MM applications requirements for several reasons:

- The QoS parameters considered are not sufficient to specify continuous media transport requirements, e.g. jitter is not considered (), and therefore must be expanded.

- Technical problems may be introduced when using the multiplexing facility particularly if the coding and compression schemes used are different: complex mechanisms to restitute information at destination are required.

- parallel processing of related media can not be performed.

Multiple Channel Approach

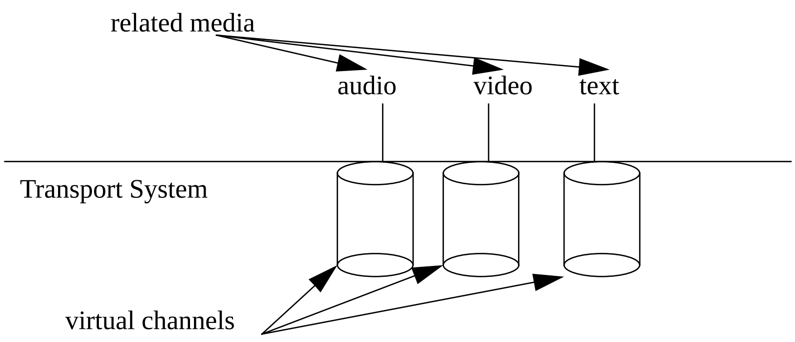

Figure 5. *Multiple Channel Approach*

The multiple channel approach () transmits related media using different channels (media per channel). Each channel supports the QoS required by the transmitted media. All the drawbacks of the single channel approach disappear. However, synchronization mechanisms are required at the destination to restitute the temporal relationships between the different media streams since the channels may take different paths and hence encounter different transit delay and jitter.

It can be seen that the multiple channel approach is preferable to the single channel approach. However, an integration of the two approaches [NAH93] may be also considered.

2.2. QoS characteristics of data communication services

When a stream passes through a communication system, the processing performed by the network consists of reconstituting at the output point the same stream of bits received at the input point. However, this activity is performed with a certain delay and may alter some of the stream data. The delay introduced by the communication system to process a data unit depends on several factors, such as the size of the data unit, propagation delays, retransmission strategies, and queueing delay; The stream data may be altered because of physical deficiencies, e.g. the transmission line is cut or high magnetic interferences, the communication system congestion, etc. For these reasons, the communication services are characterized by the following QoS parameters, to be evaluated over a certain time interval:

- Transit delay: the elapsed time from which the first bit of a segment (data unit) of the stream leaves the input point to the time the last bit of the segment passes the output point. One may distinguish the maximum delay, the average delay over a given time interval, and the minimum delay.

More detailed description of the impact of system components on QoS support, and an overview of mechanisms used by these components to provide QoS guarantees will be presented in the next sections.

3. QUALITY OF COMMUNICATION SYSTEMS

Two main components are identified in communications systems: transport protocols and networks. Before presenting network and transport protocols related QoS parameters, we would like to discuss the issue related to the transmission of more than one stream for a given session, e.g. video and audio streams of a given movie when audio and video streams are stored (and thus encoded) separately.

3.1. MM communication

Two approaches may be used to transmit related streams, e.g. video and audio streams of a given movie: single channel approach and multiple channel approach.

Single Channel Approach

The single channel approach () proposes to transmit related media using a single channel in order to preserve the temporal relationships between media during the transfer phase. This approach necessitates a small number of control packets (signalling packets), that is, control packets required to maintain inter-stream synchronization are not issued.

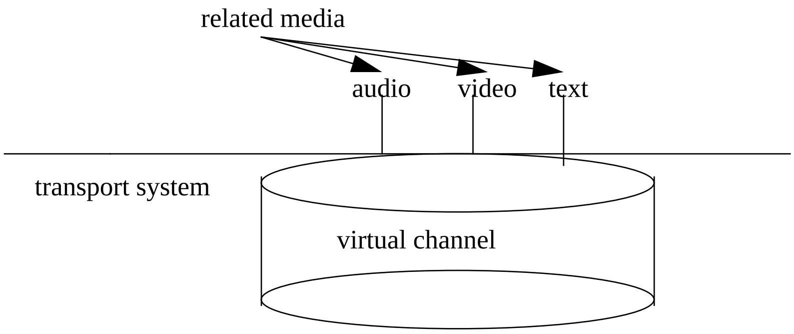

Figure 4. *Single Channel Approach*

However, the single channel approach has several drawbacks:

- Related media, e.g. video and audio, usually have different QoS requirements but only one QoS can be set. To guarantee QoS required by all media, the highest QoS is set. This induces a not optimal usage of the network resources.

- Reliability problem: if the channel is interrupted because of a problem, e.g. hardware failure, no information (all media) is delivered to the destination. This is not suitable for many applications, e.g. video-phone, which prefer to receive a part of the information rather than no information.

- Related media must be available at the same location which is not always the case.

- A new type of objects (reactive objects [Pnueli 86]) has been introduced into the model which has synchronization related semantics.

- The addition of declarative statements of QoS to support the QoS specification.

A lot of efforts and time is still needed to enable the developers of distributed MM applications using ODP concepts. Currently, a great discussion and research effort is conducted by ISO on the development of the application of QoS concepts to general object-oriented specifications and systems, and particularly ODP systems [ISO 97]. It does worth noting that the Object Management Group (OMG) is working toward the extension of CORBA [OMG 96] to support QoS, and thus to enable the deployment of CORBA-based MM applications [OMG 97]; to meet similar goals, several proposals have been also proposed. Examples are: Xbind [Laz 96], the Shiva project [Sch 96], and the Distributed Audio Video Environment [Fri 95].

2.3.3. QoS in Distributed MM Applications

To support MM applications requirements, all the system components involved should support a level of QoS in a way to meet end-to-end requirements. Each component realizes a particular stream processing function, such as video encoding or decoding performed by some hardware or software component of a computer, or data transmission performed by some communication networks. Each stream processing function has usually at least one stream input and stream output, which may have different real-time properties.

In a video-on-demand system, to deliver a movie with a desired QoS, the continuous media file server, network, transport protocols, codec, and end-system should support a certain level of QoS (Figure 14). Each of these components should commit itself, by means of some resource reservation mechanisms, to support a certain level of delay, jitter, and loss rate in a way to deliver the stream according to the user's requirements.

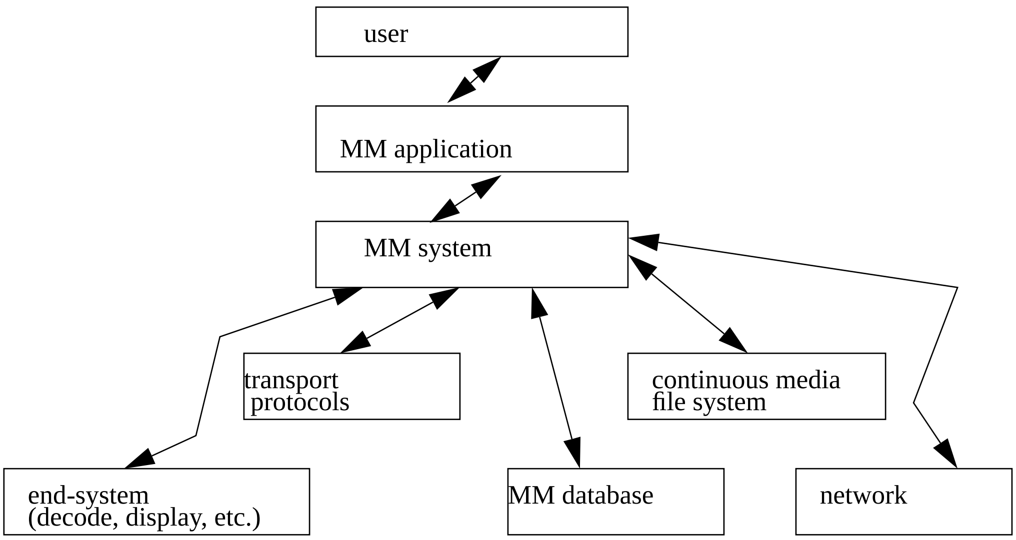

Figure 3. *Multimedia System Components*

new requirements stemming from distributed MM systems. Indeed the transport system is only one component from a set of components involved in the QoS support of MM applications (see Section 2.3.3).

Currently a great discussion and research effort is conducted by ISO to standardize and define useful QoS concepts, such as a basic QoS framework, QoS categories, QoS characteristics and QoS management functions for the support of MM applications. A large number of drafts have been produced, however no official document is published.

ODP's QoS Viewpoint

The ODP reference model [ISO 10746] allows to describe distributed systems that support heterogeneous distributed processing through the use of a common interaction model. In other terms, a system which is defined within ODP RM behaves, when implemented, as a unified entity. Thus the distributed nature of the system is transparent to the user. The ODP framework is aimed at supporting distribution, internetworking between ODP systems, interoperability and portability of applications across heterogeneous platforms. To deal with the complexity of distributed systems, ODP has defined the viewpoint concept. Each viewpoint represents a different level of abstraction of the system under consideration and has its specific concepts and terminology defined as a 'viewpoint language'. There are five viewpoints:

The *Enterprise viewpoint* is concerned with business requirements, structure and organization of an enterprise, e.g. policies.

The *Information viewpoint* is concerned with the information requirements of the system, e.g. information flow and information presentation.

The *Computational viewpoint* is concerned with the specification of the functionality of the system. The computational viewpoint defines the objects within the system and the interactions that occur among those objects as interfaces based on a client/server model.

The *Engineering viewpoint* is concerned with the infrastructure required to support the distribution (communication) according to the third viewpoint.

The *Technology viewpoint* is concerned with the implementation, realized as hardware and/or software components, of the system.

The ODP reference model defines QoS as "A set of quality requirements on the collective behavior of one or more objects". Behavior means the occurrence of events. Thus the quality of the performance of an event can be measured right after it happened. ODP's QoS definition is too general to be meaningful, since it includes all system parameters without distinction [Vogel 95].

To support the style of interactions required by continuous media, enhancements of the model, mainly concerning the computational viewpoint, have been proposed. Particularly, Coulson et al. [Coulson 93] proposed the following extensions:

- A new type of interface has been introduced into the model which has special continuous media related semantics, namely the stream interface.

The service is provided at the MM Service Access Points (MM-SAP). The MM-SAP interface represents a filter which distinguishes the user-visible provider performance parameter (= QoS parameters of the service specification) from the internal provider knowledge on performance features, e.g. multiplexing or checksum.

The provider knowledge on the customer's service requirements is restricted to the aspects stated by the customer (customer opinion). The customer requests individual QoS parameters, e.g. video resolution and color quality, for its application. A mapping between the customer requirements on the QoS and the provider performance parameters from the service provider is needed. Furthermore, other functions such as media coding/decoding, e.g. MPEG and MJPEG, and multiplexing different streams, e.g. video and audio streams, may be required. It is up to the service provider to make effective and efficient use of these functions to meet the customer's requirements.

2.3.2. QoS in Standard Frameworks

OSI's QoS Viewpoint

During the 1970's, companies, e.g. IBM, DEC, have developed communication protocols to interconnect their product, e.g. computers. Because of heterogeneity of such protocols the interconnection of computers issued by different constructors was impossible. To overcome this problem, the ISO and ITU have developed, adopting a layered approach, the OSI reference model [ISO 7498, Halsall 92]. The complete communication subsystem is broken down into seven layers each of which performs a well defined function.

In the OSI Reference Model [ISO 7498, Halsall 92] the notion of QoS is defined as a set of qualities related to the provision of an (N)-service as perceived by an (N)-service-user. Hence the QoS concept is associated with parameters to quantify the characteristics of the transfer of data between service access points of an OSI layer. The (N)-service users and the (N)-service provider exchange QoS related information (Figure 2).

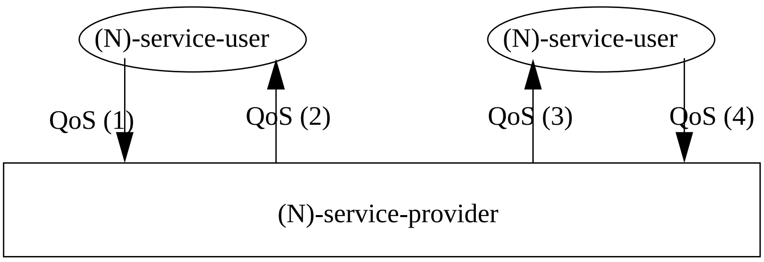

Figure 2. *Example of Flow of QoS Requirements in a Confirmed (N) Service Facility*

When exchanging QoS information at (N)-SAP, the (N)-service-provider performs the following actions:

- It receives QoS information from the (N)-service-user ((N+1)-layer) acting as the requester and possibly (if the service is confirmed) delivers related confirmations to that same (N)-service-user.

- It delivers QoS to the (N)-service-user at the destination, and possibly receives related responses from that same (N)-service-user.

The QoS parameters defined by OSI apply mostly to lower protocol layers (especially transport layer); The application layer has a very limited notion of QoS: QoS parameters are just handed down to the transport layer without any processing, e.g. mapping. Consequently OSI's QoS coverage is incomplete and is not satisfactory with the

Further multipoint communications services, facilities for group management are required to support group activities such as joining and leaving the group. Providing such facilities will simplify considerably multiparty cooperative applications.

2.3. Quality of Service

2.3.1. General Definitions

A starting point for QoS description is the frequently used general definition by ITU[CCITT I.350]: “QoS is the collective effect of service performance which determine the degree of satisfaction of a user of the service”. An improvement of the QoS definition could be found within ETSI and RACE [Race 91]. They have stated that QoS “is described in terms of a set of user-perceived characteristics of the performance of a service. It is expressed in user-understandable language and manifests itself as a number of parameters, all of which have either subjective of objective values”. QoS parameters include parameters which express the system behavior performance w. r. t. time (failure probability, throughput) and parameters which express other service characteristics as protection (security) or priority [ISO 86]. There are two types of QoS parameters:

- Objective parameters that can be directly observed and measured at the points at which the service is accessed by the service user, e.g. delay and throughput.

- Subjective QoS parameters that depend upon user equipment, e.g speaker quality, and opinion and can not be measured directly.

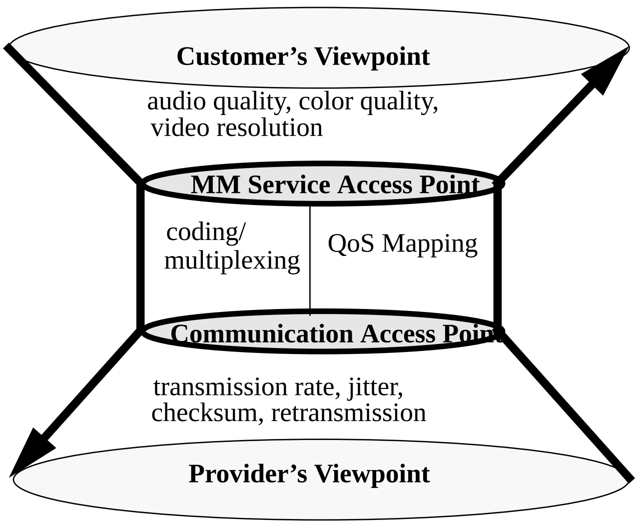

Figure 1. *Different views on QoS*

Judgements on Service Quality are provided from two sides: the provider and the customer of the service (). And in real life, both opinions might be different. Originally every party thinks in its own terms. The provider engineers a service by the implementation of provider specific functions, e.g. transport classes, and procedures, e.g. checksum and error recovery. In many cases the provider makes use of other additional services (or networks) and their individual performance, e.g. transmission rate.

quality of the requested services, since data will be lost or delayed due to resources shortage. MM applications must provide service guarantees, and this for different quality classes which are associated with the services available.

The degree of guarantees for providing for the requested service must also be specified. The following degrees of guarantees may be distinguished: deterministic guarantees, statistical guarantees, and best-effort guarantees [Nagarajan 93]. A deterministic guarantee is a service guarantee that holds for every service data unit transmitted between two service access points during the service duration. For example, requiring that every video frame of a video stream be delayed by no more than 200 ms is a deterministic guarantee. Deterministic guarantees are suitable for applications with severe requirements, such as control applications in planes. In contrast to deterministic guarantees, statistical guarantees are not required to hold for every service data unit transmitted. For example requiring that 50% of video frames of a video stream be delayed by no more than 200 ms is a statistical guarantee. Statistical guarantees are appropriate for a vast range of MM applications, since they can tolerate an amount of lost and delayed data units, e.g. video frames, while still providing an acceptable presentation quality. Best-effort services are based on either no guarantees, or on partial guarantees. Most of the current systems, networks and end-system, provide best-effort guarantees. More sophisticated services, based on best-effort guarantees, are proposed to support MM applications at the network level. Examples of those services are predictive service [Clark 92], and the service proposed in [Schulzrinne 90]. To accept or to reject a new connection, a predictive service [Clark 92] makes use of estimation based on the past network performance behavior. Schulzrinne et al. [Schulzrinne 90] propose a selective discarding mechanism at network nodes for delay-sensitive applications, which drops packets with a little chance of meeting the end-to-end deadline, thus freeing network resources to speed up the service of other packets.

2.2.4. Group communication

Group communication is important since most MM applications involves several users sharing the same information. Such applications have constraints on the acceptable time in which the information must be received by all members of the group. To support those requirements high performance multipoint communication services are required. Thus the provision of multicast and broadcast service with specific data rates and latency is of great importance.

Multicast and broadcast (in opposite to unicast) facilities allow to reduce the time and resources required for the delivery of the same data to several recipients:

- reduces the time it takes for all destination to receive information.

- reduces the processing, memory, end-system bus bandwidth, and protocol overhead. In video-on-demand service, using multicast facility only one stream is opened to serve several users who ask to play the same movie at the same time.

- reduces the network bandwidth.

Intensive research on multicasting issues have been published in the literature [Deering 89, Armstrong 92, Brauds 93, Crowcroft 88]. The proposals differ mainly in terms of cost, e.g. total bandwidth consumption, scalability, e.g. scalable to large groups, efficiently, robustness and reliability and ordering. The usage of one protocol or another depends on the application requirements: for delay-sensitive applications, such as teleconferencing systems, it is suitable to use low-delay multicasting protocols, while for reliability-sensitive applications, such as databases, it is recommended to use reliable multicasting protocols.

zation.

Stream synchronization

Stream synchronization is concerned with latency and jitter issues with respect to a single stream (intra-stream synchronization) as well as maintaining existing temporal relationships between more than one stream, e.g. audio and video, during the presentation phase (inter-stream synchronization).

Intra-Stream Synchronization

Latency imposes an upper bound on the maximum acceptable delay and any data exceeding this delay is useless. In other terms, late data is considered as lost data and is simply ignored. Each component of the end-to-end path between generation and presentation of data introduces a fraction of delay and thus contributes to end-to-end delay. Those fractions of delay depend on the media and the (dynamic and static) state of the components used for the media delivery. Examples of components are networks, routers, decoders, presentation devices.

Jitter imposes an upper bound on the maximum variation in delay introduced by the components of the end-to-end path; it is crucial to support jitter requirements to preserve continuity of media presentation.

Inter-Stream Synchronization

Inter-stream synchronization deals with temporal relationships which may exist between different streams, e.g. lip-synchronization. Each stream passes through several components in the path from source to destination accumulating delay at each component. Furthermore they may have different paths. Thus the difference in the delays which may be encountered by related streams may be important and probably not acceptable to MM applications unless appropriate guarantees can be made (network and end-system guarantees).

Event-based synchronization

Event-based synchronization aims to allow MM applications to take appropriate actions in response to a notification event in an ongoing media presentation. Typical actions may affect construction of the data values, e.g. to modify MM document or to alter the JVTOS shared windows, or the temporal behavior of a continuous media presentation, e.g. start, stop, or slow. This induces severe timing constraints for the execution of an action, e.g. stop.

Group synchronization

Group synchronization concerns the principle “what you see is what I see” (WYSIWIS): all participants of the session must have the same view of shared windows at virtually the same time. This becomes important for MM applications that are concerned with interactions between a number of users who may be receiving the same MM information at the “same time”. An important range of applications, e.g. teleconferencing systems and collaboration tools, are concerned with group synchronization, and thus, it is essential to support this type of synchronization.

2.2.3. Service guarantees

In order to support MM applications, service guarantees are required. That is, networks and end-systems should be able, by means of some kind of resource reservations, to support the requested data rates and temporal constraints for the duration of the service to provide. Otherwise MM applications *may not* provide an acceptable

More generally, conferencing systems depend on coordinated presentations of shared MM information. The WYSIWIS (What You See Is What I See) concept must be supported to provide an acceptable conferencing system. When a participant points an object, by using his telepointer, all participants must see the telepointer in the same position. The same position concept is related to the local shared window characteristics, e.g. position on screen and the size of the local shared window.

There are already a number of commercial products available which offer distributed multimedia capabilities, including videoconferencing. Examples of desktop videoconferecing products are QuickTime Conferencing from Apple Computer, ProShare vide system 200 from Intel, DV100R from Mosaic Information Technologies, ShowMe from Sun Microsystems, and VC8000 from British Telecom. A detailed list of the industrial products, along with their description, can be found in [Adie 96].

2.2. MM applications requirements

MM applications are characterized by the handling of continuous media, and the support of a variety of media with their temporal relationships, e.g. video and audio. This imposes new requirements on communication systems and end-systems namely (1) high data rate, (2) temporal constraints, (3) service guarantees, and (4) communication groups.

2.2.1. High data rate

Due to the huge amount of data associated with continuous media, especially video, underlying systems should support a high data rate to support MM applications requirements (). Video demands on bandwidth range from $p \times 64$ Kbit/s, where p ranges from 1 to 30, for ISDN video phones to 1.2 Gbits for High Definition TV (HDTV) [Tawbi 93]. Data rates are usually reduced using compression, but this reduces flexibility, and introduces additional latency. A number of compression techniques and standards are available, However even compressed video, still requires high bandwidth. The high bandwidth of continuous media means that unless high speed networks, powerful workstations with real-time facilities, and suitable host interface attachments are available, concrete support of MM applications is simply not possible.

Service	Average bit rate
uncompressed CD-quality audio	100-200 Kbit/s
MPEG Video	~ 2 Mbit/s
Uncompressed standard-quality video	140 Mbit/s
Uncompressed high definition TV	1.2 Gbit/s
Compressed high definition TV	128 Mbit/s

Table 1. *Average data rates of some services*

2.2.2. Temporal constraints

Continuous media have an inherent temporal dimension resulting in a set of synchronization requirements. Those requirements are divided into three categories: stream synchronization, event synchronization, and group synchroni-

Most MM applications have both presentational and conversational aspects. Besides communicating audio-visually with other users, they may share and manipulate MM information by using the facilities provided by the system, e.g. authoring tools. Examples of such applications include tele-conferencing systems, tele-education systems, tele-commuting systems, and medical systems. For sake of clarity let us present an example of this type of MM applications: Joint viewing and tele-operation service (JVTOS) [Dermler 93, Gutekunst 93].

JVTOS is a tele-cooperation environment which allows to share single-user multimedia (MM) applications. It provides also support for cooperation-aware MM applications. JVTOS offers several services: session management service, floor control service, telepointer service, application sharing service, audio and video support service and synchronization service.

Session management service

The role of this component is to manage and run JVTOS sessions. The operations provided by the session management service (SMS) are: Open a session, Close a session, Invite a user to a session, Drop a user from a session, Join a session, leave a session, and Change the session chairman. The SMS contains a management information base (MIB) and the floor control service (FCS). The MIB stores the session's static and dynamic data (e.g. list of current participants in a given session) and it is also used by the managed services. The FCS is used to assure the input of one application user at a time avoiding problems of single user applications being accessed from multiple users as well as preventing network overhead caused by multiple input that would not be processed. The main operation provided by the FCS are: Request the floor, Release the floor, Assign the floor, and Revoke the floor.

Telepointer service

The telepointer service (TS) permits to monitor the telepointers owned by a user and the mouse pointer of the floor holder and to display those movements locally and at remote user sites. The main operations provided by the TS are: Create a telepointer, Remove a telepointer, and Change the telepointer attributes (color, size, orientation, update interval, etc.).

Multimedia application sharing service

The multimedia application sharing service (MASS) allows to share multimedia (MM) applications. Hence the output of shared MM applications are viewed at virtually the same time by all the session participants. The main operations provided by the MASS are: Open an application, and Close an application.

Audio/video service

The audio/video service (AVS) allows JVTOS session participants to communicate audio-visually (e.g video-phone service). The main operations provided by the AVS are: Open a video or/and audio connection, and Close a video or/and audio connection.

In summary, JVTOS provides a support to the users to:

- Jointly view the output of shared MM applications.
- Add, delete and feed the shared applications.
- Communicate with each other by using audio and/or video services.
- Use globally visible pointing tools to point objects jointly viewed.
- To filter their input of received data by dropping and establishing connections during the lifetime of a session.
- Leave and join a current session.

MM applications are characterized by the manipulation of continuous media information and traditional media information. Examples of multimedia applications include video-on-demand, teleconferencing systems, tele-education, tele-shopping, and health care systems. By this definition, MM applications create important requirements for communication systems and end-systems, since they require much higher rates, especially those handling video, and much severe temporal constraints than classical applications.

2.1. Classification of multimedia applications

Broadly speaking, MM applications can be classified into presentational applications, e.g. video-on-demand [Rowe 93] and news-on-demand [Miller 93], conversational applications, e.g. video-phone systems, or having both aspects, e.g. computer supported cooperative work [Karmouch 93, William 92].

2.1.1. Presentational applications

Presentational applications [Kerherve 94] take the form of MM information digitally stored in one or more high capacity storage devices (server computers). Users can retrieve MM information in real-time from MM servers over a broadband network onto their display devices. In a video-on-demand system, customers have the possibility to select different movie titles and request the desired movie to be presented.

The difficulty in building presentational applications relates to the high data rate and severe temporal delivery constraints of continuous media. Firstly the current disk bandwidth imposes limitations on the support of many simultaneous streams, and thus the number of users to be served at the same period of time is relatively small. Secondly the traditional file systems techniques do not provide sufficient real-time guarantees for continuous data delivery, e.g. slow seek time, that is, they are not optimized to support operations, such as play, fast-forward, or fast-backward.

It is argued in [Lougher 93] that presentational application goals will be achieved once users can create, retrieve, and manipulate stored multimedia files in much the same way that users do so today with traditional files. Current implementations of presentational applications are far from achieving these goals [Lougher 93]. Examples are the applications developed at the MIT MediaLab [Hodges 89], the Galatea audio/video service [Applebaum 90], and the storage server for continuous media at Lancaster University [Lougher 93]. Representative theoretical works on digital storage systems, particularly scheduling issues, are the work at Berkeley [Anderson 92] and San Diego [Rangan 93]. There are already a number of industrial products, such as Apple’s Quicktime and Intel’s Audio Video Kernel. A complete list of commercial products can be found in [Adie 96].

The development of presentational applications depends on the advances in communication systems, e.g. communication protocols and high speed networks, storage, e.g. object oriented database and optic disc, and processing, e.g. real-time operating system.

2.1.2. Conversational (and presentational) applications

Conversational applications, e.g. video-conferencing, involve real-time communication among people from their offices, homes or laboratories. Those applications are more delay sensitive than presentational applications, since the information is created in real-time, e.g. from a camera, and should be consumed in real-time, e.g. by a monitor. The data produced is only valid for a bounded period and once this time is elapsed the data is useless.

Quality of service (QoS) provides a unique concept which allows to express MM application requirements and a good basis for their support. A possible definition of QoS is as follows: " Quality of service represents the set of those quantitative and qualitative characteristics of a distributed MM system necessary to achieve the required functionality of an application" [Vogel 95]. It includes performance-oriented attributes, such as transmission delay, format-related attributes, such as compression scheme, synchronization aspects, such as skew between audio and video sequences, cost issues, such as copyright charges, and user-oriented attributes, such as subjective sound quality.

In order to meet the service user requirements, *QoS management* is essential. It is concerned with finding appropriate QoS characteristics for the different system components in a distributed MM application and reserving the corresponding resources which are necessary to achieve the required functionality of a given application and to optimize the overall system performance. That is, all system components, involved in the QoS provision, should support a level of QoS in a way that the required end-to-end QoS is satisfied.

This paper presents a survey on distributed MM applications and QoS. This provides a framework for evaluating QoS architectures (Section 6) and a basis for the definition of a general QoS management framework which is described in [Hafid 97a].

Section 2 presents new characteristics and a classification of distributed MM applications, and identifies the requirements of this type of applications. That is, any system claiming the support of MM applications should provide guarantees supporting those requirements. QoS definitions are also presented. Section 3 presents QoS in communication systems; more specifically it explains the role of transport protocols and networks, and identifies related QoS parameters. Examples of realizations, such as XTP and ATM, are presented. Section 4 presents QoS in end-systems; more specifically it explains the role of operating systems, file systems, database systems, and users, and identifies related QoS parameters. Examples of realizations, such as real-time scheduling mechanisms, are highlighted. Section 5 argues the need of QoS management for supporting QoS guarantees. The enumeration and definitions of QoS management functions are based on the existing literature and our *personal views*. Examples of realizations of some of these functions are highlighted. Section 6 presents some representative QoS architectures, and a critical review of these architectures which represents a starting point for our thesis research. Section 7 concludes the paper.

2. DISTRIBUTED MULTIMEDIA APPLICATION

Multimedia can be defined [ISO 91] as the property of handling several types of representation media where the representation media is the type of data which defines the nature of the information as described in its coded format, e.g. for audio: CCITT G711, MIDI and MPEG/audio. Some information media consist of a continuous sequence of finite sized samples which have strict temporal dependencies. Such media are categorized as continuous and the term “stream” is used to refer to the sequence of samples constituting a continuous media. For example if an audio stream is suspended for a period longer than 10 ms, then a noticeable degradation in quality will be perceptible by the user; that is, the maximum delay difference between any two successive samples of an audio stream should be smaller than 10 ms.

Continuous media are voluminous in nature causing transmission bandwidth, processing, and storage problems. Examples are digital video, e.g. MPEG or MJPEG video, audio, animation, and continuous sampling of data. Classical media such as text and graphics (excluding animation) are categorized as static, since they do not have any temporal constraints. Temporal relationships may exist not only between the components of a single stream, but also between the components of different streams of the same media or different media, e.g. between audio and video, also called lip-synchronization.

Electronic Journal on Networks and Distributed Processing, No. 6, 2 1998, pp. 1−50.
Manuscript received 20 October 1997 .

DISTRIBUTED MULTIMEDIA APPLICATION AND QUALITY OF SERVICE: A REVIEW

Abdelhakim Hafid[1], Gregor von Bochmann[2] and Rachida Dssouli[2]

[1]*Advanced Communication Engineering Centre, University of Western Ontario, London, Ontario, Canada, hafida@bille.ppc.engga.uwo.ca*
[2]*Université de Montréal, Dept. d'Informatique et de Recherche Operationnelle, Montréal, Canada, {bochmann,dssouli}@iro.umontreal.ca*

ABSTRACT-Distributed multimedia (MM) applications are very sensitive to the quality of service (QoS) provided by their computing and communication environment. This paper surveys the QoS issues behind distributed MM applications. MM applications characteristics and requirements are identified; QoS notions are defined. The role of all system components, namely transport protocols, networks, operating systems, file servers, multimedia databases, and user interfaces, in the provisioning of QoS is described. The need for QoS management to support distributed MM applications is motivated. The different QoS management functions are defined and examples of realizations are presented. Finally some representative QoS architectures are presented, their limitations are identified, and the requirements of an "ideal QoS architecture" are presented.
KEY WORDS: *multimedia, quality of service, quality of service management, quality of service architecture*

1. INTRODUCTION

Currently new emerging services, particularly *distributed multimedia (MM) applications*, e.g. video conferencing and video-on-demand, based on broadband communications (B-ISDN) are of great importance in industry, academic research and standardization. Large scale deployment of distributed MM applications will impose very high constraints on the overall system components, such as network and end-system. MM applications requirements range from high data rates, due to the voluminous nature of MM data, to severe temporal constraints, due to the continuous nature of video and audio.

