公開 (Public Release)

TR-ACR-0003

006

Study on Communication Coordination

Tatsuya Yamazaki

2002.3.19

(株)国際電気通信基礎技術研究所
適応コミュニケーション研究所
〒619-0288 京都府相楽郡精華町光台二丁目 2 番地 2
Tel: 0774-95-1501　Fax: 0774-95-1508

Advanced Telecommunications Research Institute International
Adaptive Communications Research Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
Telephone: +81-774-95-1501　Fax: +81-774-95-1508

Contents

Summary

Motivated by coordinating multimedia communication service from the end-user's viewpoint in heterogeneous communication environments and with various individual requirements, this report proposes a networking architecture and a QoS (Quality-of-Service) management framework. Upon the whole, one main characteristic of the proposed methods is to control QoS at the application level in connection with the upper and lower levels, assuming best-effort type infrastructures.

Chapter 1 describes the backgrounds, motivation, objectives and abstract of this report. In Chapter 2, the notion of QoS considered through this report is defined, and based on it, a layered QoS model is shown. In order to link different QoS levels, QoS mapping plays an important role by translating QoS parameters between different QoS levels. A QoS mapping method using Spline functions is proposed. Relevance of QoS between application and user levels is also discussed. Chapter 3 proposes a networking architecture for heterogeneous communication environments, where a proxy server located between sender and receiver sites transforms media QoS according to available computational and network resources and user's requirements. A prototype of the proxy server is developed for video image transmission applications. In Chapter 4, an adaptive QoS man-

agement framework for distributed media is proposed based on the multi-agent system. One feature of this framework is its 2-tier QoS management. Namely the long-term QoS adaptation is executed in one tier, while the short-term QoS adjustment is executed in the other tier. A one-way video system is developed based on the proposed framework. Chapter 5 presents applications of the proxy server architecture and the multi-agent-based framework to realistic environments. An error resiliency scheme using both channel and source coding techniques is proposed in consideration of QoS management. Then, a QoS management architecture combining the proxy server architecture and the multi-agent-based framework is discussed. Also, an application of the proxy server to home networks is described. Chapter 6 presents an idea of multimedia communication coordination that meets a QoS policy agreement based on a layered QoS model. The multimedia communication coordination consists of system-oriented and user-oriented coordinations. As a typical application, a chat system with video transmission is introduced. Chapter 7 concludes this report and describes the future problems.

Acknowledgements

I am very grateful to Dr. Bokuji Komiyama, the director of ATR Adaptive Communications Research Laboratories (ATR ACR), for his support, guidance and encouragement through my stay at ATR ACR. I greatly indebted to Dr. Kazuo Hasuike, the head of ATR ACR Department 1, for his valuable suggestions and guidance.

I would like to thank all of the current and past members of ATR ACR Department 1. Their great aid and true friendship have propelled my work at ATR ACR.

I give my sincerest appreciation to Professor Jun Matsuda of Osaka Gakuin University, the former head of ATR ACR Department 1. He proposed the basic concept of Communication Coordination and suggested development of the Communication Coordination Server to me.

I owe thanks to Dr. Nagao Ogino of KDDI R&D Laboratories for his valuable discussions and guidance.

I am also thankful to Mr. Masakatsu Kosuga of ATR ACR. He originally proposed and developed the MARM framework.

I wish to thank all members of ATR ACR. It was a great pleasure that I could share a warm, interesting, and fruitful time with them. Especially, I would like to

thank the Planning Section members at ATR ACR. My work at ATR ACR would not have succeeded without their assistance and support.

Finally, but not least, I wish to thank my wife, two daughters, and a future new life for their support, understanding, and patience.

Tatsuya Yamazaki

March 2002

Chapter 1

Introduction

1.1 Backgrounds and Motivation

Along with growth of the Internet and development of digital technologies, digital media have come to surround us. Eventually rapid and facile delivery of digital information are changing our life style. Text and still image data occupy the digital information on networks, however, audio data including voice and music are increasing and followed by video data.

Considering from the viewpoint of service, service requirements differ among various media or applications that deal with the media. The notion of QoS (Quality of Service) was introduced to satisfy the service requirements, to differentiate

CoS (classes of service), or to manage the service quality. Although QoS is one of hot topics for delivery of various digital information through the Internet, QoS has different meanings depending on the people who use it [1].

First and foremost, the history of QoS is looked back briefly. The concept of QoS originally came from the specification of the networking service levels. For example, the Open Systems Interconnection (OSI) Reference Model has a number of QoS parameters describing the speed and reliability of transmission, such as throughput, transit delay, error rate, and connection establishment failure probability [2]. Also, according to ISO (International Standard Organization) standards, QoS was provided by the network level of the communication system [3]. Indeed three major QoS mechanisms are proposed to differentiate QoS at the network level: IntServ (Integrated Services) [4], DiffServ (Differentiated Services) [5], and MPLS (Multiprotocol Label Switching) [6], [7]. Although QoS is regarded as equivalent to differentiating traffic CoS at times, it has a broad and ambiguous connotation [1]. One extension of QoS is to include both the network and end-system domains.

To date, several QoS architectures covering both network and end-system domains have been proposed [8]-[11]. Typical two of them are introduced briefly: one is the OMEGA architecture [9] , the other is the QoS-A (Quality-of-Service

Architecture) [11]. The OMEGA, developed at the University of Pennsylvania, provided a combinative QoS architecture of a transport subsystem and an application subsystem. In the transport subsystem, bounds on delay were provided and bandwidth demands were met. In the application subsystem, application QoS requirements were guaranteed by a real-time mechanism. These subsystems were combined by the QoS broker model. The QoS-A also provided a layered architecture of services and mechanisms for QoS management and control of continuous media flows in multiservice networks.

None the less, since quality should be ultimately judged by the end-users [12], it is important to take personalization of service into consideration, that is service provision according to each user's requirements. To realize the personalization of service, extension of QoS is indispensable because the digital information aims to be perceived and used by end-users, and most of the applications are designed to attain this purpose. Indeed several layered QoS models with extension of the end-user level were proposed [3], [13]-[15]. In particular, Fukuda et al. [16] proposed a method to decide required bandwidths, which is one of network QoS parameters, in consideration of the relationship of application-level QoS parameters and user's preference on video quality, where the user's preference was evaluated by subjective tests.

In addition, heterogeneity of communication environments is a considerable point to realize the personalization of service. Hereupon the heterogeneity of communication environments means that, for example, wireless networks generally have less bandwidth and higher error rates than wired networks. Moreover, end-system performance differs from each other; for example, handheld computers have some limitations with CPU power, memory, window size, video/audio hardware equipments, and battery capacity compared to desktop computers. Furthermore, the available system resources are changeable because of various causes, e.g. the throughput of the best-effort network decreases as network traffic increases, the error rate of wireless links fluctuate according to the electromagnetic wave propagation environment, and available CPU performance is reduced when other applications are in operation. However, the preceding QoS architectures covering both network and end-system domains can support very limited or no adaptive mechanisms to the changing communication environments. If any, their adaptation to the dynamical changes is based on the end-to-end (re)negotiation protocols or specific filtering schemes. For example, in the OMEGA architecture only one QoS parameter was permitted to change in renegotiation during the transmission phase for real-time implementation.

To realize the provision and management of personalized QoS, QoS must be related with the user requirements and judgements, and adaptive mechanisms for variation and dynamic changes in the system performance are indispensable for QoS management; these are motivations of this research.

1.2 Objectives and Abstract

The main purpose of this report is to propose a networking architecture and a QoS management framework for QoS-aware transmission of video images. This report will focus on QoS management for video media since video media would be a critical component in future distributed multimedia applications. The QoS-aware video transmission means to adjust QoS of video streams according to the end-user's requirements and the changing communication environments. A best-effort network without any QoS mechanism like IntServ, DiffServ, or MPLS, is assumed as the infrastructure for this research, because such QoS mechanisms have not been spread out on the current Internet enough widely yet. Therefore, the proposed mechanisms and methods are deployed at the application level in connection with the lower level. In contrast, the preceding studies assumed some QoS-guaranteed network like ATM (Asynchronous Transfer Mode) networks. Also any real-time scheduling mechanism such as a real-time OS used in [17] is not assumed for the end-systems in this report.

In short, one feature of this research is to manage QoS at the application level in connection with the upper and lower levels.

In Chapter 2, a layered QoS model is described and a QoS mapping method using spline functions is proposed. This spline-based QoS mapping method is used for QoS management mechanisms described in Chapters 3 and 4.

In Chapter 3, a QoS adjustment scheme is proposed for real-time video transmission applications for a group of heterogeneous receivers. The proposed scheme manages multiple users who have different communication environments and different requirements for multimedia services, considers the user's respective communication environments and calculates a feasible QoS for each user to utilize the system resources like CPU powers or network bandwidths efficiently. It works at the application level for the best-effort type system so that no special network protocol such as the RTP (Real-time Transport Protocol) nor special coding method such as the scalable coding is needed. The proposed scheme is deployed in a proxy server, which intermediates between a video server and a group of receivers. A prototype of the proxy server is implemented, and it is assumed to become a QoS server in a Local Area Network, a home network [18], or a community network.

In Chapter 4, an adaptive QoS management framework for distributed media called MARM (Multi-Agent Resource Management) is proposed on the basis of the multi-agent system. In the MARM framework, the agents directly or indirectly collaborate to adaptively manage the media QoS according to the available network and terminal resources as well as the user requirements. The particular point of the framework is that it provides 2-tier QoS management. Namely the global and long-term QoS adaptation is executed in one tier, while the local and short-term QoS adjustment is executed in the other tier. A one-way video system is developed on the basis of the proposed framework as an example of communication-intensive applications.

Chapter 5 presents applications of the CCS (Communication Coordination Server), a proxy server, and the MARM framework to realistic environments. For a wireless environment, an error resiliency scheme is proposed by utilizing both channel and source coding techniques. The error resiliency scheme is supposed to function on the CCS. A QoS management architecture combining the CCS approach and the MARM framework is discussed, where the CCS is useful to mitigate the complex QoS negotiation in the MARM framework. Also an application of the CCS to home networks is considered.

Chapter 6 presents a QoS mechanism of multimedia communication coordination that meets a QoS policy agreement based on a layered QoS model. As a typical application, a chat system with video transmission is being developed. The basic design and implementation of the chat system are described.

Chapter 7 concludes this report.

Bibliography

[1] P. Ferguson and G. Huston, Quality of Service: Delivering QoS on the Internet and in Corporate Networks, John Wiley & Sons, New York, 1998.

[2] A. Vogel, B. Kerhervé, G.v. Bochmann, and J. Gecsei, "Distributed multimedia and QOS: a survey," IEEE Multimedia, pp.10-19, Summer 1995.

[3] R. Steinmets and K. Nahrstedt, Multimedia: COMPUTING, COMMUNICATIONS AND APPLICATIONS, Prentice Hall, Upper Saddle River, NJ, 1995.

[4] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," RFC 1633, Internet Engineering Task Force, June 1994.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, Internet Engineering

Task Force, Dec. 1998.

[6] Y. Katsube, K. Nagami, and H. Esaki, "Toshiba's router architecture extensions for ATM: overview," RFC 2098, Internet Engineering Task Force, Feb. 1997.

[7] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow, "Cisco system's tag switching overview," RFC 2105, Internet Engineering Task Force, Feb. 1997.

[8] C. Aurrecoechea, A.T. Campbell and L. Hauw, "A survey of QoS architectures," ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, vol.6, no.3, pp.138-151, May 1998.

[9] K. Nahrstedt and J.M. Smith, "Design, implementation, and experiences of the OMEGA end-point architecture," IEEE J. Sel. Areas Commun., vol.14, no.7, pp.1263-1279, Sept. 1996.

[10] L. Delgrossi, C. Halstrick, D. Hehmann, R.G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media scaling for audiovisual communication with the Heidelberg transport system," Proc. ACM MultiMedia '93, pp.99-104, Anaheim, June 1993.

[11] A. Campbell, G. Coulson, F. Garcia, D. Hutchison, and H. Leopold, "Integrated quality of service for multimedia communications," Proc. IEEE Infocom'93, pp.732-739, San Francisco, March 1993.

[12] M. Murata, "Quality of service guarantees in multimedia networks: approaches and open issues," IEICE Trans. B-I, vol.J80-B-I, no.6, pp.296-304, June 1997 (in Japanese).

[13] K. Mase ed., Multimedia Networks and Communications Quality, IEICE, Tokyo, 1998 (in Japanese).

[14] K. Nahrstedt and R. Steinmets, "Resource management in networked multimedia systems," IEEE Computer, vol.28, no.5, pp.52-63, May 1995.

[15] A. Puliafito, O. Tomarchio and H.de Meer, "An agent-based framework for QoS management," Proc. of WCSS'97, pp.392-396, Singapore, Sept. 1997.

[16] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "QoS mapping between user's preference and bandwidth control for video transport," Proc. Fifth IFIP International Workshop on Quality of Service (IWQoS 97), pp.291-302, May 1997.

[17] K. Kawachiya and H. Tokuda, "QOS-ticket: a new resource-management mechanism for dynamic QOS control of multimedia," Proc. Multimedia Japan '96, pp.14-21, 1996.

[18] A. Dutta-roy, "Networks for homes," IEEE Spectrum, vol.36, no.12, pp.26-33, Dec. 1999.

Chapter 2

Layered QoS Model and QoS Mapping

2.1 Introduction

As Chapter 1 mentioned, there are different perspectives on QoS and the notion of QoS is sometimes elusive, confounding, and confusing [1], [2]. Ultimately the degree of QoS (or quality) should be judged by the end-user as the degree of agreement with "what it is to be" [3]. This is the notion of QoS considered through this report, so that the QoS viewed by the end-user is located at the top-level. On the other hand, the expression of QoS is different from level to level.

For example, it is defined by application-level parameters at application level and by network-level parameters at network level. In addition, end-users do not want to specify QoS for the system resources explicitly [4]. Accordingly a layered QoS model should be introduced.

In this chapter, a generic model for multi-level QoS in distributed multimedia applications is described. Then studies on QoS mapping methods, which translate the QoS expressions from level to level, are reviewed. A novel QoS mapping mechanism is proposed for the multi-level QoS model. The proposed QoS mapping mechanism includes a QoS mapping method using user-specific profile data and a mapping method using spline functions. In addition, relevance of QoS between application and user levels is discussed based on subjective test for video QoS evaluation.

2.2 Multi-level QoS Model

Figure 2.1 shows a multi-level QoS model for distributed multimedia applications. At the user level, QoS is defined as the User QoS, which is sometimes expressed abstractly. At the application level, the Application QoS is specified for each media stream by the application-level parameters, e.g. the frame rate, frame size, quantization scale (if quantization is executed as a video compression coding), for video media. The Terminal QoS is defined by the parameters that operating systems deal with, such as a program thread scheduling period and a task processing time. The Terminal QoS is sometimes omitted and identified with the Terminal Resource QoS when the operating system is equipped with no processing mechanism to deal with these Terminal QoS parameters. The Network QoS is also defined on the terminal. The Network QoS is the QoS required by the network for each media stream and is defined by network-level parameters such as throughput, delay, jitter, and loss rate. The Resource QoS is defined as the resources to be allocated for the media stream and is separated into the Terminal Resource QoS and the Network Resource QoS. The Terminal Resource QoS includes CPU utilization and memory size, and the Network Resource QoS includes bandwidth and node buffer size. We assume that only the highest User QoS can be expressed abstractly and all of the lower QoS than the User QoS are specified by one or more QoS parameter(s).

2.3 QoS Mapping Mechanism

Previous studies on QoS mapping are reviewed firstly. Then a QoS mapping mechanism is proposed for the generic multi-level QoS model. The mechanism includes two mapping methods, a user profile QoS mapping and a spline QoS mapping.

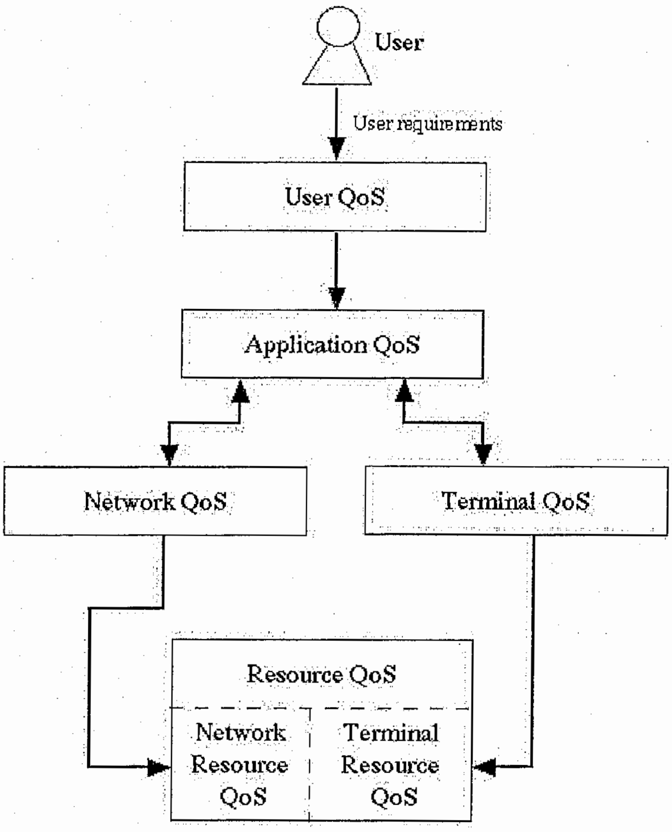


Figure 2.1: A multi-level QoS model.

2.3.1 Related Work

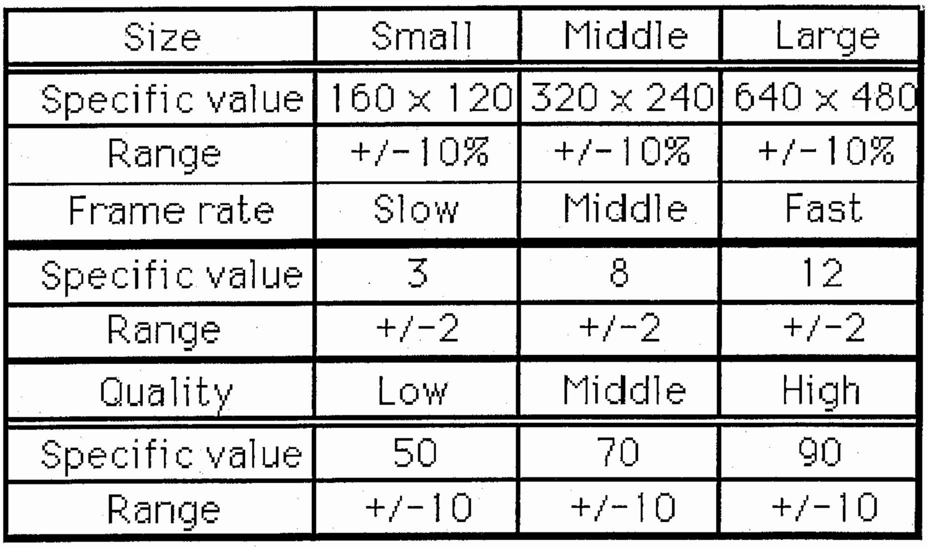
Conventional QoS mapping methods are categorized into two classes. One is a table-based mapping class [5] and another is a function-based mapping class [6]-[8]. The table-based QoS mapping method prepares QoS mapping tables of sample data in advance, and if there is a set of input QoS parameters, it returns the output QoS parameters by looking up the tables. This type of QoS mapping method is not adaptive to various user's requirements. Because it cannot give any answer when it has no corresponding entry to the input QoS parameters in the tables. On the other hand, the function-based QoS mapping method uses mathematical functions for mapping and can give an answer for any QoS requirement from the user by computing the QoS mapping functions. However, the specification of functions is entrusted to the system designer and it is questionable how to reasonably select the functions according to a dynamically changing environment.

2.3.2 User Profile QoS Mapping

When a user has little knowledge on how to set the application-level QoS, the user should provide abstract QoS requests for media streams. For example, the user may utter “I want to view a video with middle-size, fast rate, and fair quality.” Moreover such an abstract expression is based on the user's subjective, and mapping the abstract expression into specific QoS parameters depends on the user's preference and/or habitual behavior. We propose a QoS mapping method using user-specific profile data reflecting the user's preference and habitual behavior.

Table 2.1 shows an example of the user-specific profile data for video media. The mapped Application QoS has three parameters, the size, frame rate, and quantization scale. For the User QoS, they are specified abstractly by the user, for example, small, middle, or large for the size parameter. The abstract QoS expression is mapped into a specific value and a range, where the specific value presents an average and the range presents a granted width. These two values are necessary, because abstract expressions inevitably involve ambiguity.

Table 2.1: An example of user-specific profile data for video media.



Size	Small	Middle	Large
Specific value	160 × 120	320 × 240	640 × 480
Range	+/-10%	+/-10%	+/-10%
Frame rate	Slow	Middle	Fast
Specific value	3	8	12
Range	+/-2	+/-2	+/-2
Quality	Low	Middle	High
Specific value	50	70	90
Range	+/-10	+/-10	+/-10

2.3.3 Spline QoS Mapping

Once the abstract User QoS is mapped into specific QoS parameters, the next task is QoS mapping among different level QoS parameters. To this end, there are two conventional mapping methods, the table-based and function-based methods, as described in Sect. 2.3.1. However, the former is not adaptive to various user requirements and the latter is not adaptive to a dynamically changeable environment. Therefore we propose a QoS mapping method using spline functions, which is considered as a hybrid method of two conventional methods and is adaptive to both of user requirements and changeable environment.

$\boldsymbol{qos}_l$ denotes a QoS parameter vector at the l-th level. We deal with a QoS mapping from $\boldsymbol{qos}_l$ to $\boldsymbol{qos}_{(l+1)}$. $\boldsymbol{qos}_l$ has m QoS parameters and $\boldsymbol{qos}_{(l+1)}$ has n QoS parameters, that is $\boldsymbol{qos}_l = \{q_{l1}, q_{l2}, \ldots, q_{lm}\}$ and $\boldsymbol{qos}_{(l+1)} = \{q_{(l+1)1}, q_{(l+1)2}, \ldots, q_{(l+1)n}\}$. It is assumed that the application has k samples $(\boldsymbol{qos}_l^i), \boldsymbol{qos}_{(l+1)}^i)$ $(i = 1, \ldots, k)$, where

$$(\boldsymbol{qos}_l^i, \boldsymbol{qos}_{(l+1)}^i) = (\{q_{l1}^i, q_{l2}^i, \ldots, q_{lm}^i\}, \{q_{(l+1)1}^i, q_{(l+1)2}^i, \ldots, q_{(l+1)n}^i\}). \quad (2.1)$$

S_j is the spline mapping function that translates $\boldsymbol{qos}_l$ to $q_{(l+1)j}$ $(j = 1, \ldots, n)$ and the sample data points are identified as the knots in spline functions. Given the sample data, interpolation conditions, and end conditions, simultaneous equations for the unknown parameters of S_j are led. By solving the simultaneous equations, S_j is determinately specified and any user requirement at the l-th level can be calculated by S_j. Figure 2.2 depicts relationship among the sample data and S_j for the simplest case, $m = 1$. Details on the spline functions can be found, for example, in [9].

When an environmental change occurs, the expected mapped QoS parameter value may differ from a monitored value. In such a case, the sample datum is

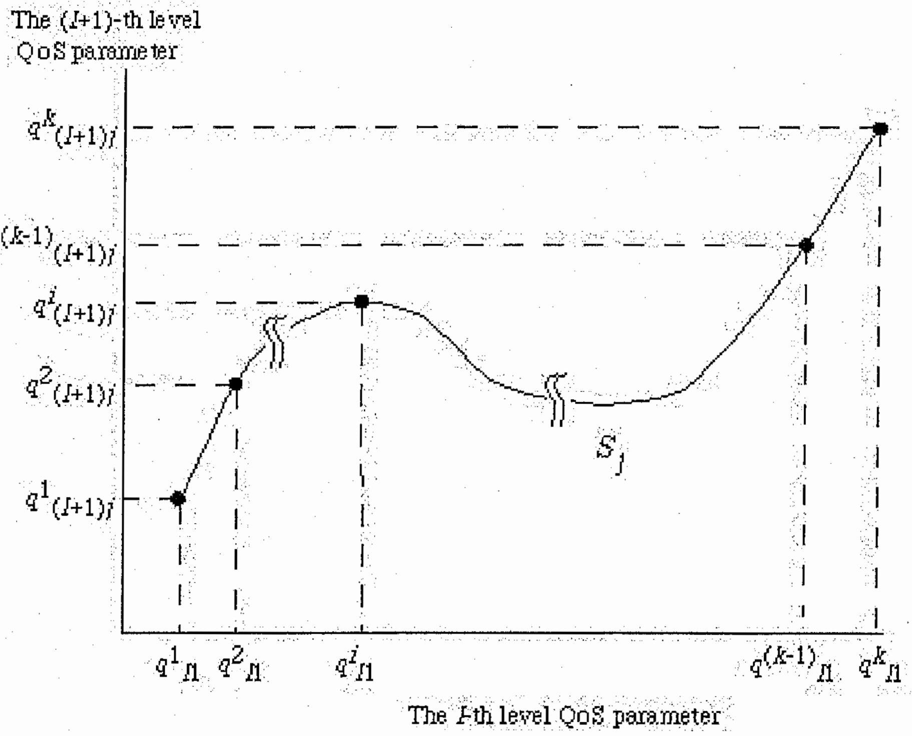


Figure 2.2: A spline function S_j and its sample data points for the simplest case.

replaced by the monitored value and the related spline functions are recomputed to adapt to the new environment.

2.4 Experimental Results

As a typical example of the multimedia application, we consider a video-conferencing system and a QoS mapping for video streams, because resources are usually consumed much more for them than for audio streams and dealing with video QoS is considered more carefully than audio QoS. An object in these experiments is QoS mapping from the Application QoS into the Network or Terminal Resource QoS, where the Terminal QoS is identified with the Terminal Resource QoS because the experimental system was equipped with no QoS control mechanism for system resources.

2.4.1 QoS Measurement

To clarify relationship between different level QoS, we conducted QoS measurement for video media using the adaptive multimedia application systems developed in our laboratories [10]. The QoS measurement system is shown in Fig. 2.3. A sender and a receiver are connected through an ATM connection. A user specifies the Application QoS for video on the receiver terminal and the Application QoS are transmitted to the sender terminal. On the sender terminal, a video signal captured by CCD camera is encoded by the Motion JPEG (Joint Photographic Experts Group) (M-JPEG) with the specified Application QoS and sent to the receiver. The receiver decodes the encoded data and the Network and Terminal Resource QoS are measured.

The Application QoS are defined by the M-JPEG coding parameters: the frame size, frame rate, and quantization scale. The frame size corresponds to the number of pixels in one frame. The frame rate corresponds to the number of frames to be presented per second, and it takes integer values. The quantization scale is related with the quantization step width used in JPEG, and it takes integer values between 1 and 100. The smaller the quantization scale, the smaller the en-

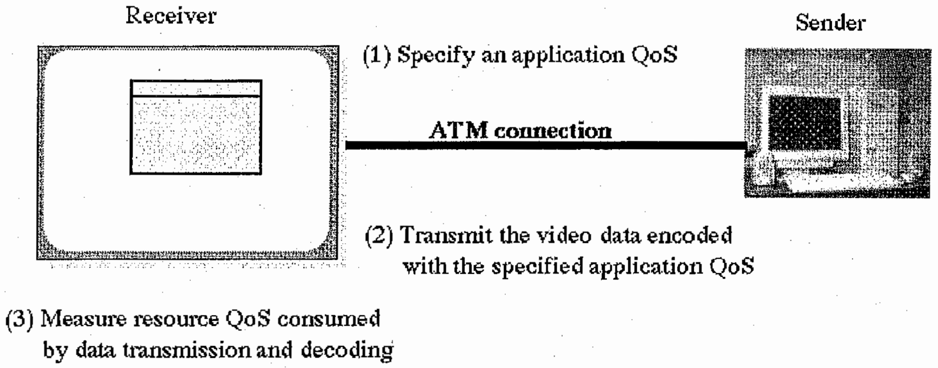


Figure 2.3: A QoS measurement system.

coded video data size, but too small quantization scale values may make the user's evaluation low because of blurs or color defects of images. The Network QoS is defined as the bandwidth needed for transmission of an encoded video stream, and the Terminal Resource QoS is defined as the CPU utilization needed for decoding the video data.

2.4.2 Natural Spline QoS Mapping Results

Figure 2.4 shows a spline QoS mapping result from the frame rate and quantization scale into the Network QoS for the frame size 320 × 240. 25 measured points, all possible combinations of the frame rate $\{1, 2, 3, 4, 5\}$ and the quantization scale $\{5, 25, 50, 75, 95\}$. The mapping is very smooth and gives a result for any user requirement, or any combination of the frame rate and quantization scale. We evaluated the accuracy of the spline QoS mapping by the inconsistent rate ICR defined by

$$ICR = \frac{1}{N} \sum_{i} \frac{| m_i - s_i |}{m_i} \times 100, \tag{2.2}$$

where N is the number of evaluation points, m_i is a measured (monitored) QoS, and s_i is the corresponding computed value by the spline function. For Fig. 2.4, 70 evaluation points, all possible combinations of the frame rate $\{1, 2, 3, 4, 5\}$ and the quantization scale $\{10, 15, 20, 30, 35, 40, 45, 55, 60, 65, 70, 80, 85, 90\}$, are selected and ICR was 12.05%. Although this value seems to be quite large, all of the inconsistencies of larger than 10% came from the quantization scale of larger than 55, where the Network QoS changes largely. It is expected that ICR lessens by an adequate selection of the sample data points, for example, a dense selection in the part of the large quantization scale and a sparse selection in the other part.

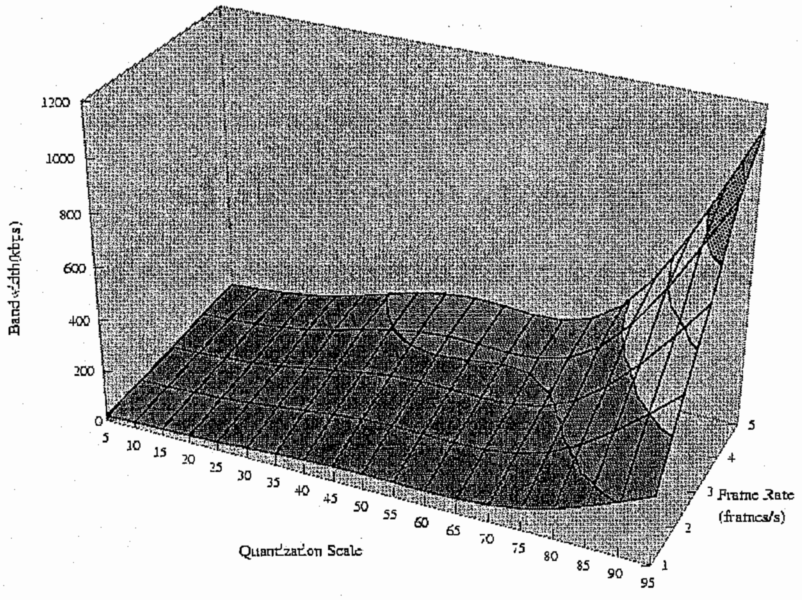


Figure 2.4: A spline QoS mapping result for the frame size 320 × 240.

2.4.3 B-spline QoS Mapping Results

We constructed the spline QoS mapping functions using the B-splines based on the above-mentioned measured data, and then compared mapping accuracy between the natural spline and the B-spline QoS mapping results. Figs. 2.5 and 2.6 present comparative results of the natural spline and B-spline QoS mapping functions. The results show relationship between change of the quantization scale and the consumed bandwidth, where the frame size and frame rate are fixed to be 160×120 and 10. In Fig. 2.5, measured data and the mapping result by the natural spline function are shown, while measured data and the mapping result by the B-spline function are shown in Fig. 2.6. In both cases, only four points of $\{5, 50, 75, 95\}$ of the quantization scale are selected as sample data, and the rest points were estimated by the QoS mapping functions. The knots for the B-spline function were selected as $\{5, 5, 5, 5, 70, 94, 95, 95, 95\}$ of the quantization scale. From the comparative result, it is found that the QoS mapping result by the B-spline is better than that by the natural spline, which slightly vibrated. The error rate was 13.9% for Fig. 2.5 and 1.49% for Fig. 2.6, where the error rate is defined as the difference between the realistically measured data and the values estimated by the QoS mapping function.

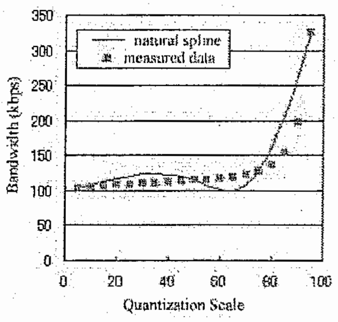


Figure 2.5: A result of the QoS mapping from the quantization scale to the bandwidth by the natural spline-based mapping function.

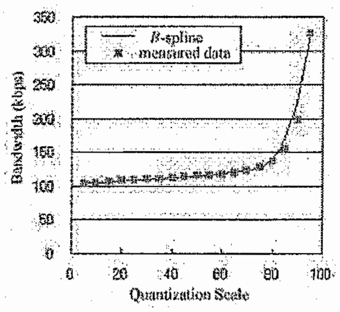


Figure 2.6: A result of the QoS mapping from the quantization scale to the bandwidth by the B-spline-based mapping function.

2.5 Application QoS and User QoS

There are several studies (e.g. [14]) regarding the user perception of QoS of motion pictures where QoS degradation caused by network loss or transmission loss are evaluated mostly. The loss effect is especially important for the motion pictures in standardized format, such as a television system. On the other hand, since the digital videos that the Internet applications deal with are easily transformed by changing the coding parameters (QoS factors) such as the frame rate, relationship between the user perception and the QoS factors is also needed to be clarified. The relationship would be useful for designing a QoS control scenario according to the user's perception or user's preference.

This section examines the effects of different QoS factors on the user's perception. To this end, a subjective test was conducted. In the test we selected the frame size, frame rate, and quantization scale as the QoS factors, and the MPEG-4 codec is used for video streaming, because it is one of the most promising coding schemes for video. The test results are discussed to support the QoS control according to the user's perception or preference for video streaming applications.

2.5.1 Overview of Subjective Test (System and Method)

The experimental system for the subjective test of video QoS is shown in Fig. 2.7. In the system, Ethernet connects a video server and a video receiver, and they operate real-time video encoding and decoding with a set of specific QoS factors using MPEG-4. The MPEG-4 codec is implemented in software. Two monitors are connected to the receiver, and a video splitter enables the monitors to display the same received video stream for two subjects simultaneously.

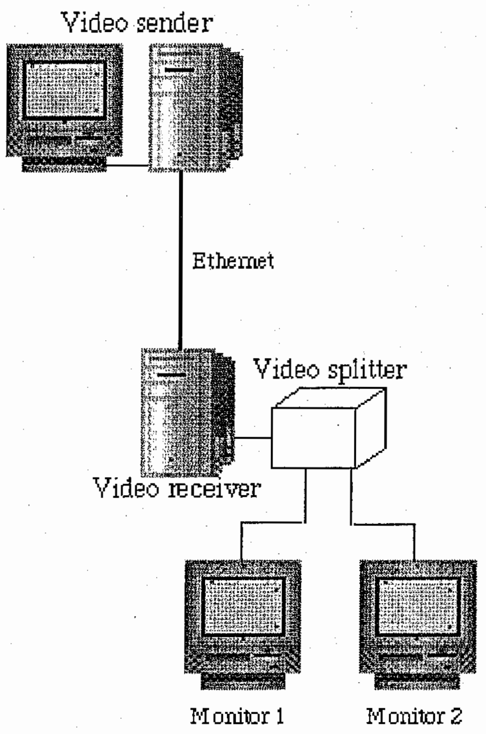


Figure 2.7: The experimental system for the video QoS subjective test.

The double stimulus continuous quality scale (DSCQS) method recommended by ITU was used in the video quality subjective test. The video presentation sequence in a trial in the DSCQS method is shown in Fig. 2.8. Each trial consists of a pair of stimuli, one is the reference, Video A in Fig. 2.8, and one is the test, Video B in Fig. 2.8. The two stimuli are each presented twice in a trial, with the order randomly chosen. The subjects rate each stimulus on a continuous quality scale shown in Fig. 2.9 by drawing a mark "X" on the scale. Thus, two ratings are made for each trial in the DSCQS method, one for the reference and the other for the test. The rating is measured as the distance between the mark and the bottom of the scale. Table 2.2 summarizes the conditions of experiments.

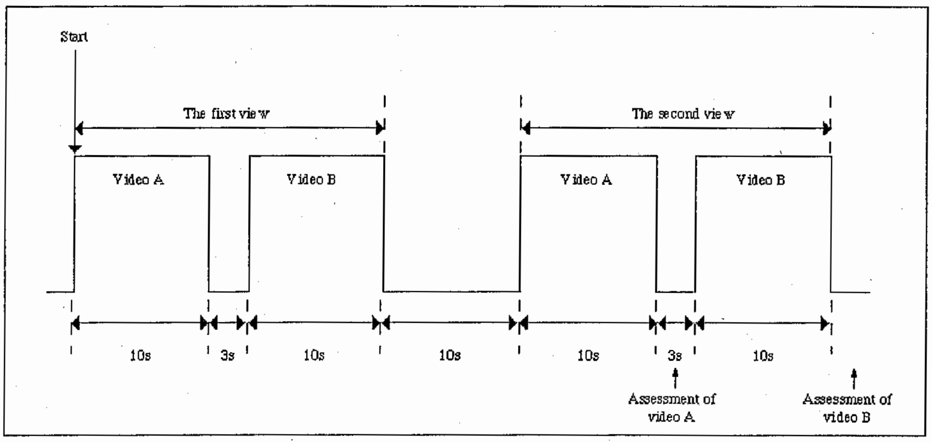


Figure 2.8: The video presentation sequence in the DSCQS method.

Table 2.2: Experimental conditions.

Evaluation method	The DSCQS method
Subject	16 non-professionals
Test video stream	Three kinds of MPEG-2 Test Sequence
Video codec	MPEG-4 simple profile
Monitor for subjects	21 inch CRTs

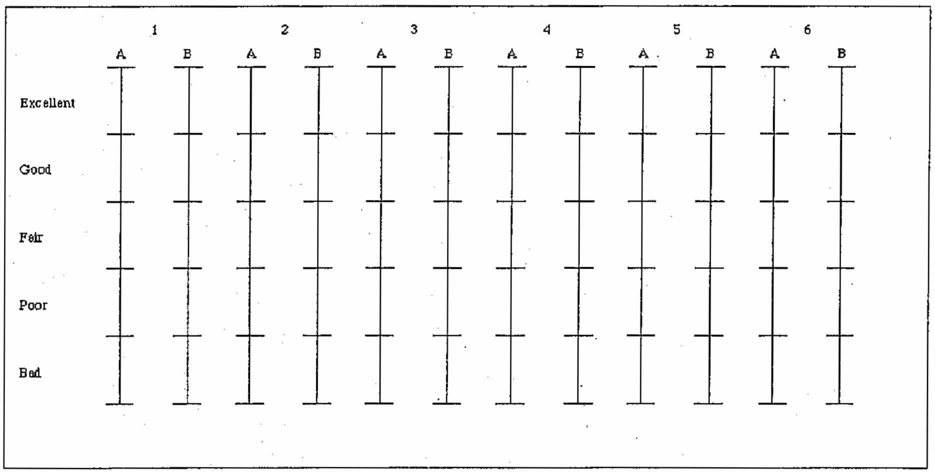


Figure 2.9: The evaluation sheet.

Table 2.3 presents the feature of video streams used in the subjective test. Each original video stream is five minutes long, its frame size is 720 × 480 pixelsframe, and its frame rate is 5 framess.

Table 2.3: Experimental conditions.

Video stream name	Feature of video stream	
	Camera movement	Object or contents
Ball	There is one scene change (SC). The camera moves rightward before the SC, and leftward after the SC in order to track the object with a relatively slow and unchanging speed.	A scene of ballet dance.
Foot	The camera pans rightward to capture a player holding a ball with a slightly changing speed.	A scene of American football game.
Bus	The camera pans leftward to capture a bus with an almost unchanging speed and zooms out at the final stage of the scene.	A bus running through downtown.

2.5.2 QoS Factors

In the test we selected the frame size (S), frame rate (F), and quantization scale (Q) as the QoS factors. A set of specific QoS factors is set for the MPEG-4 encoder in the video server that encodes and transmits the video data in real-time. The MPEG-4 decoder in the video receiver receives the video data and plays back for subjects in real-time. Three levels of the frame size can be set: the Large (360×240 pixels/frame), the Middle (240×160 pixels/frame), and the Small (180×120 pixels/frame). The frame rate can be set in integer between 1 and 30. The quantization scale can be set in integer between 1 and 31. In general, the smaller the quantization scale is, the better the quality of video becomes.

In each trial, the different sets of QoS factors were provided for the reference video and the test video. The absolute value of the rating differs from each subject; therefore we evaluated the difference of the ratings between the reference and test videos. In the following subsections, we tabulate the test results, where $\bar{x}$ represents the sample average of (the rating of the reference - the rating of the test), and 95% means the 95% confidence interval of the expected value by the t-test. If the value of $\bar{x}$ is positive, it means that the reference video was preferable for the subjects.

2.5.3 Experimental Results (Evaluation with Different S)

In trials #1 to #3, we changed the frame size only between the reference and the test, where the random choice of the two stimuli was not conducted in trials #1 and

Table 2.4: The results with changing S.

			ball		foot		bus	
	Reference	Test	$\bar{x}$	95%	$\bar{x}$	95%	$\bar{x}$	95%
#1	S=Large, F=5, Q=3	S=Small, F=5, Q=3	15.00	$\bar{x} \pm 7.64$	11.19	$\bar{x} \pm 6.95$	*	*
#2	S=Middle, F=5, Q=3	S=Small, F=5, Q=3	-9.94	$\bar{x} \pm 7.11$	-13.13	$\bar{x} \pm 6.72$	6.56	$\bar{x} \pm 4.93$
#3	S=Small, F=5, Q=3	S=Large, F=5, Q=3	-11.44	$\bar{x} \pm 7.90$	-5.13	$\bar{x} \pm 4.97$	*	*

#3, that is the reference shown in Table 2.4 was always presented to the subject first. The results are presented in Table 2.4, and "*" means no test was conducted.

2.5.4 Experimental Results (Evaluation with Different F)

In trials #4 to #10, we changed the frame rate only between the reference and the test. The results are presented in Table 2.5. Fig. 2.10 shows the results of trials #6 to #8. In Fig. 2.10, the horizontal axis (DF) is the difference of the frame rate between the reference and the test, and the vertical axis is $\bar{x}$, that is difference of the DSCQS rating in percentage.

Table 2.5: The results with changing F.

			ball		foot		bus	
	Reference	Test	$\bar{x}$	95%	$\bar{x}$	95%	$\bar{x}$	95%
#4	S=Small, F=30, Q=3	S=Small, F=5, Q=3	21.94	$\bar{x} \pm 10.71$	22.06	$\bar{x} \pm 9.10$	12.38	$\bar{x} \pm 11.91$
#5	S=Small, F=30, Q=3	S=Small, F=15, Q=3	4.69	$\bar{x} \pm 3.86$	-3.88	$\bar{x} \pm 6.74$	-4.13	$\bar{x} \pm 3.62$
#6	S=Middle, F=15, Q=3	S=Middle, F=10, Q=3	7.00	$\bar{x} \pm 3.05$	1.19	$\bar{x} \pm 3.00$	-1.13	$\bar{x} \pm 4.56$
#7	S=Middle, F=15, Q=3	S=Middle, F=6, Q=3	-4.50	$\bar{x} \pm 10.45$	10.44	$\bar{x} \pm 3.61$	13.69	$\bar{x} \pm 7.98$
#8	S=Middle, F=15, Q=3	S=Middle, F=3, Q=3	7.44	$\bar{x} \pm 12.56$	19.69	$\bar{x} \pm 7.02$	27.81	$\bar{x} \pm 10.39$
#9	S=Middle, F=10, Q=3	S=Middle, F=6, Q=3	-18.75	$\bar{x} \pm 7.93$	7.63	$\bar{x} \pm 6.39$	10.06	$\bar{x} \pm 8.94$
#10	S=Middle, F=10, Q=3	S=Middle, F=3, Q=3	-0.63	$\bar{x} \pm 11.56$	14.25	$\bar{x} \pm 4.66$	24.50	$\bar{x} \pm 8.41$

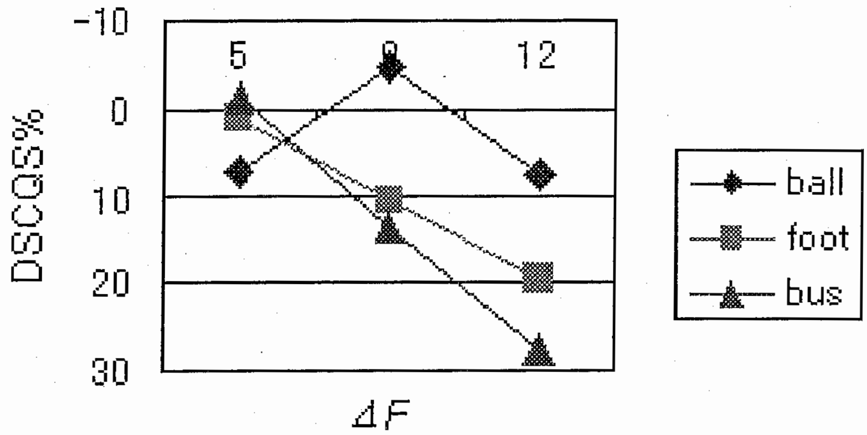


Figure 2.10: Difference of the frame rate between the reference and the test vs. DSCQS rate in percentage.

2.5.5 Experimental Results (Evaluation with Different Q)

In trials #11 to #15, we changed the frame rate only between the reference and the test. The results are presented in Table 2.6 and Fig. 2.11. In Fig. 2.11, the horizontal axis (DQ) is the difference of the quantization scale between the reference and the test, and the vertical axis is $\bar{x}$.

Table 2.6: The results with changing Q.

			ball		foot		bus	
	Reference	Test	$\bar{x}$	95%	$\bar{x}$	95%	$\bar{x}$	95%
#11	S=Middle, F=15, Q=3	S=Middle, F=15, Q=25	31.75	$\bar{x} \pm$ 11.46	21.06	$\bar{x} \pm$ 13.41	21.69	$\bar{x} \pm$ 10.35
#12	S=Middle, F=15, Q=3	S=Middle, F=15, Q=15	29.75	$\bar{x} \pm$ 10.80	15.88	$\bar{x} \pm$ 9.17	12.50	$\bar{x} \pm$ 7.79
#13	S=Middle, F=15, Q=3	S=Middle, F=15, Q=10	21.94	$\bar{x} \pm$ 8.91	7.31	$\bar{x} \pm$ 4.64	6.13	$\bar{x} \pm$ 3.97
#14	S=Middle, F=15, Q=3	S=Middle, F=15, Q=30	39.56	$\bar{x} \pm$ 12.12	23.94	$\bar{x} \pm$ 9.76	28.65	$\bar{x} \pm$ 10.82
#15	S=Middle, F=15, Q=3	S=Middle, F=15, Q=20	34.25	$\bar{x} \pm$ 10.15	17.81	$\bar{x} \pm$ 9.58	21.13	$\bar{x} \pm$ 7.35

2.5.6 Discussion

Changing the frame size effected the evaluation of video quality. The subjects felt about 10.7% DSCQS degradation for the Small and about 5.5% DSCQS degradation for the Middle compared to the Large. As these results were obtained for the specific frame rate and quantization scale (F=5 and Q=3), more evaluation tests

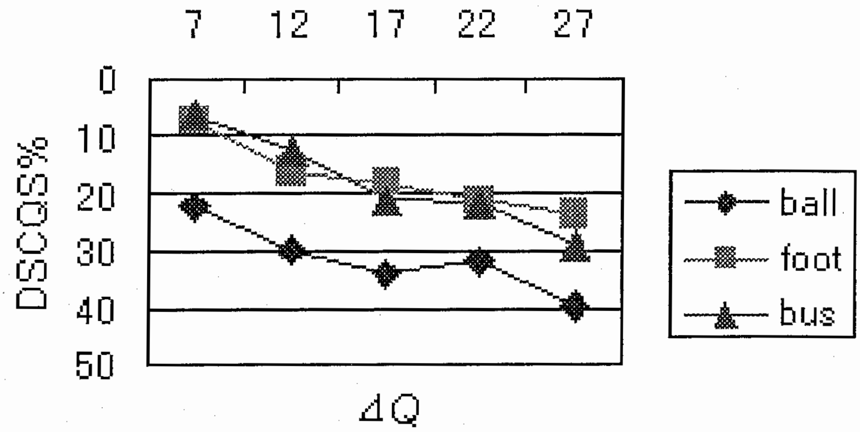


Figure 2.11: Difference of the quantization scale between the reference and the test vs. DSCQS rate in percentage.

with various sets of the frame rate and quantization need be conducted.

It has been found that a drastic change of the frame rate heavily degrades the quality evaluation. For example, the subjects felt about 18.8% DSCQS degradation for a 25 difference of frame rates in trial #4. In addition, it was found from Fig. 2.10 that the video content affected the quality evaluation. While DSCQS% is almost proportional to DF in case of the foot and the bus, it did not change proportionally in case of the ball. On the contrary, DSCQS% lessened for 9 of DF compared to 5 of DF. It seems that different movements of objects caused this. In the foot and the bus, the subjects might watch relatively large objects such as a football player or a bus move. On the other hand, as the ball included delicate movements of the dancer's hands and feet, the subjects might feel some degradation even for 15 of the frame rate.

Changing the quantization scale also affected the quality evaluation as shown in Fig. 2.11. Although the relationship between DSCQS% and DQ was almost proportional, the ball showed more degradation than the foot or the bus. This is also because of the difference of object movements as mentioned above.

2.6 Conclusion

In this chapter, a generic multi-level QoS model was presented for distributed multimedia applications, and studies of QoS mapping from one level to other level were reviewed. Then a realistic QoS mapping mechanism composed of QoS mapping methods was presented. One method maps the highest user level QoS into lower level QoS by user-specific profile data, and the other method performs mapping among lower level QoS parameters than the user level by spline functions. The mapping results by the natural spline and the B-spline QoS mapping functions were compared using the actual measured data of video QoS. It was found that the B-spline QoS mapping functions showed better results once the knots for spline function were selected appropriately [11]-[13].

Moreover, the effects of the QoS factors in application layer on subjective evaluation or preference of quality were examined. In the general, the results agreed our intuition, namely the subjects rated the video quality higher when the better QoS factors were provided, although the QoS factors consume more system resources including network bandwidth and CPU utilization. One important point is that the subjective evaluation depends on the content of video stream or the object movement. It might be useful to categorize the video streams into several genres like sports or arts, and to control the QoS factors in consideration of the genre. User's individual preference for the QoS factors must be considered. To obtain the user's preference, a reinforcement learning method [15] would be applicable.

The QoS elements used in the multi-level QoS model are basically categorized from the viewpoint of the place where the service is provided. Therefore, the multi-level QoS model does not always correspond with the OSI (Open Systems Interconnection) reference model. Although QoS mapping mechanisms connect different levels, further study is needed to establish a generic service flow for the multi-level QoS model. Relevance between the OSI reference model and QoS is referred to in detail in [3].

Bibliography

[1] P. Ferguson and G. Huston, Quality of Service: Delivering QoS on the Internet and in Corporate Networks, John Wiley & Sons, New York, 1998.

[2] G. Armitage, Quality of Service in IP Networks: Foundations for a Multi-Service Internet, MTP, Indianapolis, 2000.

[3] K. Mase ed.: Multimedia Networks and Communications Quality, IEICE, Tokyo, 1998 (in Japanese).

[4] N. Nishio and H. Tokuda, "Simplified method for session coordination using multi-level QOS specification and translation," in Building QoS into Distributed Systems, A. Campbell and K. Nahrstedt eds., Chapman Hall, Tokyo, 1997.

[5] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "Effective algorithms for multicast video transport to meet various QoS requirements,"

IEICE Trans. Commun., vol.E81-B, no.8, pp.1599-1607, Aug. 1998.

[6] A. Richards, G. Rogers, M. Antoniades, and V. Witana, "Mapping user level QoS from a single parameter," Proc. Second IFIP/IEEE International Conference on Management of Multimedia Networks and Services '98 (MMNS'98), Versailles, Nov. 1998.

[7] J.-F. Huard and A.A. Lazar, "On QOS mapping in multimedia networks," Proc. 21th IEEE Annual International Computer Software and Application Conference (COMPSAC '97), Washington, D.C., Aug. 1997.

[8] J.-F. Huard and A.A. Lazar, "On end-to-end QOS mapping," Proc. IFIP Fifth International Workshop on Quality of Service (IWQoS '97), New York, May 1997.

[9] H. Späth, One Dimensional Spline Interpolation Algorithms, A K Peters, Welleslet, MA, 1995.

[10] M. Kosuga, T. Yamazaki, N. Ogino, and J. Matsuda, "A multi-agent based adaptive QoS management scheme," IEICE Trans., vol.J82-B, no.5, May 1999 (in Japanese).

[11] T. Yamazaki and J. Matsuda, "On QoS mapping in adaptive QoS management for distributed multimedia applications," Proc. ITC-CSCC'99, vol.2, pp.1342-1345, July 1999.

[12] T. Yamazaki and M. Sakamoto, "QoS mapping data acquisition mechanisms for distributed multimedia applications," Proc. ITC-CSCC'01, vol.2, pp.1168-1171, July 2001.

[13] T. Yamazaki, "Spline-based QoS mapping mechanisms for hierarchical multilevel QoS models," IEICE Trans. Fundamentals, vol.E85-A, no.6, June 2002 (accepted).

[14] D. Hands and M. Wilkins, "A study of the impact of network loss and burst size on video streaming quality and acceptability," in Interractive Distributed Multimedia Systems and Telecommunication Services, 6th International Workshop, IDMS'99, M. Diaz, P. Owezarski, P. Senac editors. LNCS 1718, pp.45-57, Springer Verlag, Oct. 1999.

[15] K. Nakaoka and J. Matsuda, "A method for acquisition of QoS mapping rules for distributed multimedia applications," Proc. PDPTA'99, pp.496-502, April, 1999.

Chapter 3

QoS Control with a Proxy Server

3.1 Introduction

This chapter proposes a network architecture with a proxy server for heterogeneous communication environments. An adaptive QoS management mechanism performed on the proxy server is also proposed based on the layered QoS model and the spline QoS mapping method as discussed in Chapter 2.

A simple heterogeneous communication environment model is shown in Fig. 3.1. A video sender SND and two receivers, RCV 1 and 3, exist on a wired network and another receiver, RCV 2, is connected to the wired network via a wireless link. The video sender multicasts a video stream to the receivers, but the receivers have different requirements. The high-performance desktop-type receiver RCV 1 on the wired network usually requires higher QoS than the handheld-type receiver RCV 2 or the low-performance desktop-type receiver RCV 3.

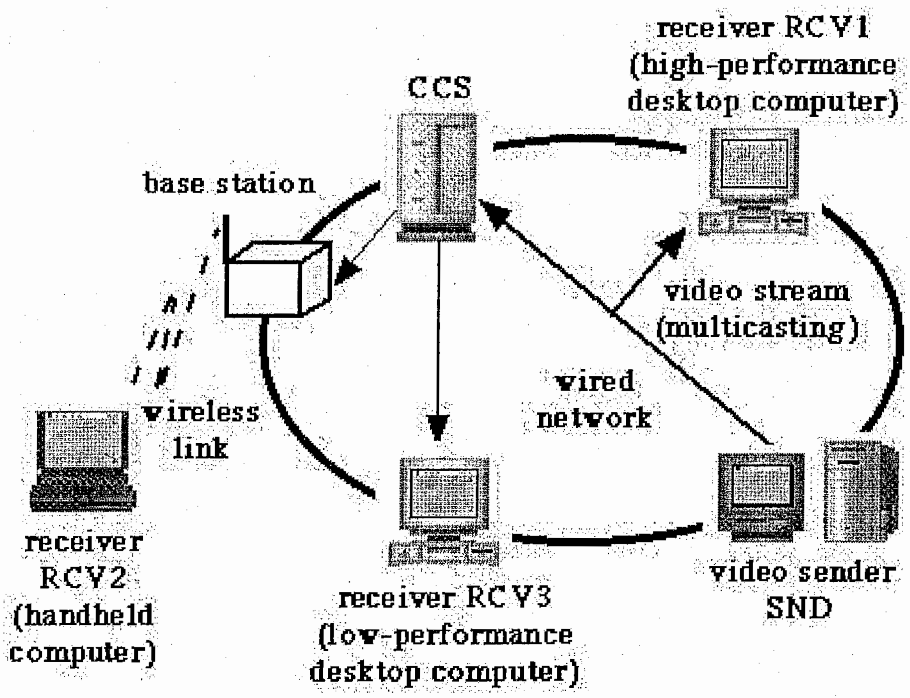


Figure 3.1: A example of heterogeneous communication environments and CCS approach.

Heterogeneities of the communication environments and various user preferences restrain video transmission to multiple receivers. Suppose that a video sender transmits video at the same quality to all receivers. When the sender tries to satisfy the lowest quality requirement by a low-performance computer such as a handheld-type on the wireless network, a high-performance receiver on the wired network has to sacrifice its high quality requirement. On the other hand, when the sender tries to satisfy the highest quality requirement by the high-performance receiver, the low-performance computer cannot deal with the video stream contentedly because processing and/or transmission performance is lacking. As well as adaptability to the static heterogeneous environments, adaptability to dynamical change in network and terminal performance is another important issue.

In this chapter, we present a video proxy server, called CCS (Communication Coordination Server), located between the sender and the receiver to compensate the performance gap. The CCS transforms QoS of the video stream according to the QoS requirement from the receiver, and mediates the QoS according to the receiver's user policy and the current network and terminal performance. The CCS is equipped with the QoS spline mapping mechanism described in Chapter 2 to translate application-level QoS into resource-level QoS.

3.2 Background and Related Work

To date, a lot of efforts have been spent on constructing QoS architectures to support end-to-end QoS management, and Aurrecoechea et al. [1] summarized the QoS architectures. These architectures basically targeted peer-to-peer multimedia communications. Aurrecoechea et al. defined QoS filtering as a mechanism to bridge a heterogeneous QoS capability gap, however, only one architecture, the QoS-A [2], supports the QoS filtering mechanism in the end-system.

The framework of media scaling [3] gives a possible solution to the mentioned multicast problems. In this framework, the video sender prepares a scalable video stream and an intermediator (e.g. a router) filters the video stream according to the receiver's requirements. While this approach is efficient, the video sender must be equipped with an encoder that supports the scalable coding and the level of QoS is limited by the sender, not the receiver.

Ohta et al. [4] proposed SMAP (Selective Multimedia Access Protocol) for multimedia communications in mobile computing environment. The SMAP is a priority-based multimedia communication protocol, with which multimedia data are assigned with priority for each media unit, and selective transport service can be realized by using the priority. However, this is also a sender-initiated service, because the setting of priority has to be done by the author or the provider of media.

The proposed CCS approach is not sender-initiated in the sense that no special coding method such as the scalable coding nor special protocol is needed. In the CCS approach, a proxy server is located between the video sender and the receiver, and it can change the video QoS by transcoding.

As for researches about the transform of video coding methods (transcoding), Amir et al. [5] reported an implementation of the Video Gateway which transcodes between the JPEG [15] and the H.261 [12]. Also Warabino et al. [6] developed a transcoding proxy server between the MPEG-1 (Motion Picture Experts Group 1) [10] and the Quality Motion. Both of them, however, targeted video transmission to a single receiver and did not deal with the resource management in multiple receiver situations.

Moreover, the notions of "translator" and "mixer" have been defined in the proposal of the RTP (Real-time Transport Protocol) [7]. Both translators and mixers are considered as intermediate systems, but the distinction between translators and mixers is that a translator passes through the data streams from different sources separately, whereas a mixer combines them to form one new stream [7]. A translator or a mixer can intermediate a group of receivers homogeneously.

3.3 CCS Approach Architecture

In the CCS approach, the CCS located between the video sender and the receiver receives the QoS requirement from the receiver. If the QoS requirement is admissible for the available network and terminal resources, the CCS transcodes the video stream according to the QoS requirement and sends the transcoded stream to the receiver. Thus, the CCS can provide a QoS management service based on the transcoding, and QoS requirements from all receivers are satisfied in the heterogeneous communication environment.

The CCS approach architecture is shown in Fig. 3.2, and consists of the sender site, the CCS site, and the receiver site. The video sender, CCS, and receiver are connected through wired or wireless networks. The Sender application, CCS application, and Receiver application are the transcoding application staying on each site.

3.3.1 Sender Application

The Sender application manages a video source, which might be archives of encoded videos or a real-time video encoder. With archives, the Sender application lists the archived video files in response to a receiver's inquiry and sends the encoded video selected by the receiver to the CCS. With real-time video transmission, the Sender application captures a video signal from the equipped video camera and encodes it to send to the CCS.

3.3.2 CCS Application

The CCS application mediates QoS between the sender and the receiver to satisfy the QoS requirement from the receiver, using QoS admission, adjustment, allocation, and mapping mechanisms. Then it conducts a transcoding task to realize the allocated QoS. Namely, it receives an encoded video stream from the sender, transcodes it according to the coding format required from the receiver, and sends the transcoded video stream to the receiver. The transcoding is performed by a

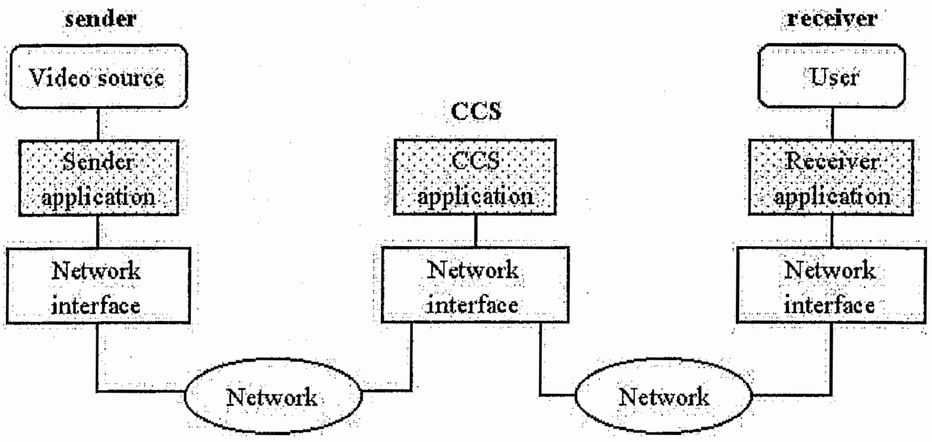


Figure 3.2: The CCS approach architecture.

combination of a video decoder for the input stream format and a video encoder for the output stream format. Both the decoder and the encoder are implemented in the CCS application at the hardware- or software-level.

$$(qos_l^i, qos_{(l+1)}^i) = (\{q_{l1}^i, q_{l2}^i, \ldots, q_{lm}^i\}, \{q_{(l+1)1}^i, q_{(l+1)2}^i, \ldots, q_{(l+1)n}^i\}). \quad (3.1)$$

3.3.3 Receiver Application

The Receiver application mainly has three tasks. The first task acquires QoS requirements from the user and transmits them to the CCS. The second task decodes and playouts the transcoded video stream from the CCS. These tasks are for a user interface. The third task monitors resources such as CPU utilization or network bandwidth.

3.4 Intra-frame and Inter-frame Compression Techniques

Up to now, video data compression techniques have been standardized. The ISO (International Organization for Standardization) has standardized the MPEG-1 [10] and MPEG-2 [11] and the ITU (International Telecommunication Union) has standardized H.261 [12], and H.263 [13], [14]. The main features of these video compression techniques are based on inter-frame and DCT (Discrete Cosine Transform). Namely, in these techniques, the basic operation predicts motion from frame to frame in the temporal direction, and then uses DCTs to organize the redundancy in the spatial directions. We call these video compression techniques inter-frame compression.

On the other hand, ISO standardized the JPEG [15] for still image compression. Video data can be encoded by JPEG as a sequence of JPEG frames, and this technique is often used as M-JPEG. Since the data is compressed frame by frame without prediction from other frames, M-JPEG is referred as intra-frame compression in contrast to the inter-frame compression.

While the compressed video data size using inter-frame compression is smaller than that using intra-frame compression due to the removal of temporal redundancy, there are two advantages in intra-frame compression over inter-frame compression. One advantage is its robustness to transmission error and the other is its lightweight processing algorithm. These advantages come from the frame independence of intra-frame compression processing.

Accordingly, inter-frame compression is generally advantageous when the receiver uses only low-bandwidth links, while intra-frame compression is advantageous when the hardware performance of the receiver, such as CPU or memory, is deficient. If both the network bandwidth and the receiver performance are deficient, the CCS simultaneously carries out intra-frame compression and the QoS admission functions to compensate for network and terminal deficiencies.

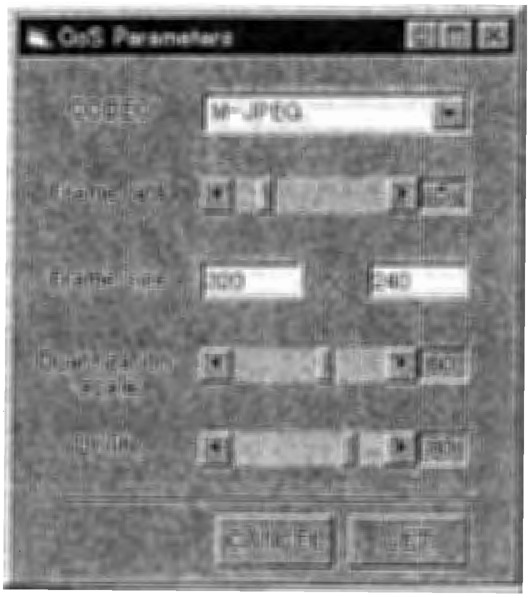
3.4.1 QoS Mediation by CCS

In this section, first of all, we define the QoS for video streams, and then introduce a QoS mapping mechanism that translates the application-level QoS into the resource-level QoS. Finally, QoS admission, adjustment, and allocation mechanisms by the CCS are described.

3.4.2 Definition of QoS for Video Streams

In our architecture, the application-level QoS for video streams is defined by compression parameters (spatial resolution, temporal resolution, and quantization scale). The user of the receiver can specify these parameters using the application-level QoS setting window (Fig. 3.3).

The spatial resolution corresponds to the number of pixels in one frame and we assume that it takes one of three values, 640×480, 320×240, or 160×120. The temporal resolution corresponds to the number of frames presented per second



CODEC M-JPEG

Frame size 320 240

Figure 3.3: The application-level QoS setting window.

and takes integer values. The larger the temporal resolution, the more smooth the observation of objects in the video sequence. The quantization scale is related to the quantization step width used in JPEG or MPEG and takes integer values between 1 and 100. The smaller the quantization scale, the smaller the encoded video data size. However, if the quantization scale values are too small, they may lower the user's evaluation because of image blurring or color defects. In Fig. 3.3, the spatial resolution and the temporal resolution correspond to the frame size and frame rate, respectively. The utility in Fig. 3.3, which is a parameter representing the user's preference for the specified application-level QoS, ranges from 0 to 100.

The larger the value of the utility is, the higher the user's satisfaction. The utility can be used to determine the user's QoS policy. The QoS policy includes a priority order of the QoS parameters, namely it means that the user lays stress on which QoS parameter.

The resource-level QoS consists of the network QoS and the terminal QoS. The network QoS is defined as the bandwidth needed for the transmission of the encoded video stream, while the terminal QoS is defined as the CPU power needed for decoding the encoded video data.

3.4.3 QoS Admission, Adjustment, and Allocation Mechanisms

Fig. 3.4 shows the flow of video stream transmission with QoS management.

(1) The receiver sends a request of the video file list, after accepting the user's QoS requirement.

(2) The sender returns the file list to the receiver.

(3) The receiver sends the selected file name and the QoS requirement.

(4) The CCS verifies that the resources are available for the QoS requirement by QoS admission. If the resources are not enough, the CCS performs QoS adjustment according to the user's QoS policy until the QoS becomes admissible. If the QoS is admitted, go to (5).

(5) The CCS sends a request of the video data selected by the user.

(6) The sender returns the video data to the CCS.

(7) The CCS transcodes the video data according to the QoS requirement.

(8) The transcoded video data are sent to the receiver and presented to the user.

The CCS receives the QoS requirements specified by the QoS parameters and QoS policy from the receiver user. The CCS translates the QoS parameters required by the receiver user into the resource-level QoS by the QoS mapping mechanism, and the QoS admission is carried out by comparing available resources. The available resource can be estimated by knowing the maximum capacity of the resource and the current used resource. The maximum capacity of the bandwidth can be determined or estimated for a guaranteed network such as an ATM network, but it is difficult to specify it for a best-effort network. The current used resource can be specified by using some monitoring mechanism. We have developed an original network monitor to get the current network throughput information. The network monitor works for both ATM and IP networks.

If the required QoS is admitted, the QoS allocation is carried out to transmit a video stream. Otherwise, the CCS adjusts the required QoS by degrading the QoS parameters according to the QoS policy until they become admissible.

An example of QoS adjustment is shown in Fig. 3.5. Supposing that point

A (5 frames/s and a quantization scale of 90) is an initial QoS requirement and it requires a bandwidth of 511.5 kb/s. In case of resource shortage, for example 250 kb/s of the available bandwidth, points B (5 frames/s , a quantization scale of 60, and the required bandwidth of 245.6 kb/s) and C (2 frames/s, a quantization scale of 90, and the required bandwidth of 206.5 kb/s) are examples of new QoS candidates, and one of them is selected according to the user's QoS policy. When the policy is that the frame rate has the lowest priority order than the others, point B is selected so that the frame rate is decreased. Meanwhile, when the policy was that the quantization scale has the lowest priority order than the others, point C is

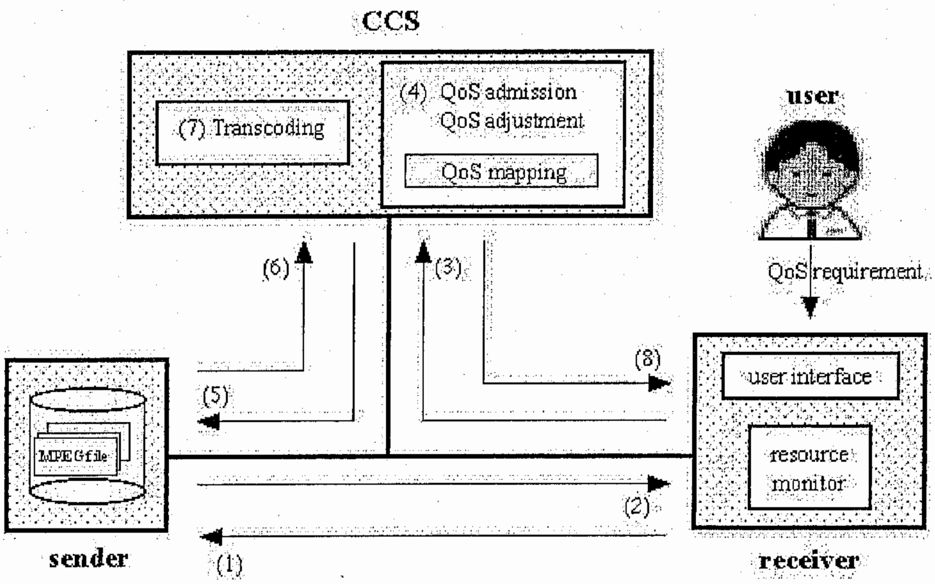


Figure 3.4: The flow of video stream transmission with QoS management.

selected so that the quantization scale is decreased.

These QoS mechanisms are executed periodically in order to adapt to dynamical resource changes. However too much adaptation may degrade user's evaluation. How adaptation frequency and degree relate to the user's evaluation is a

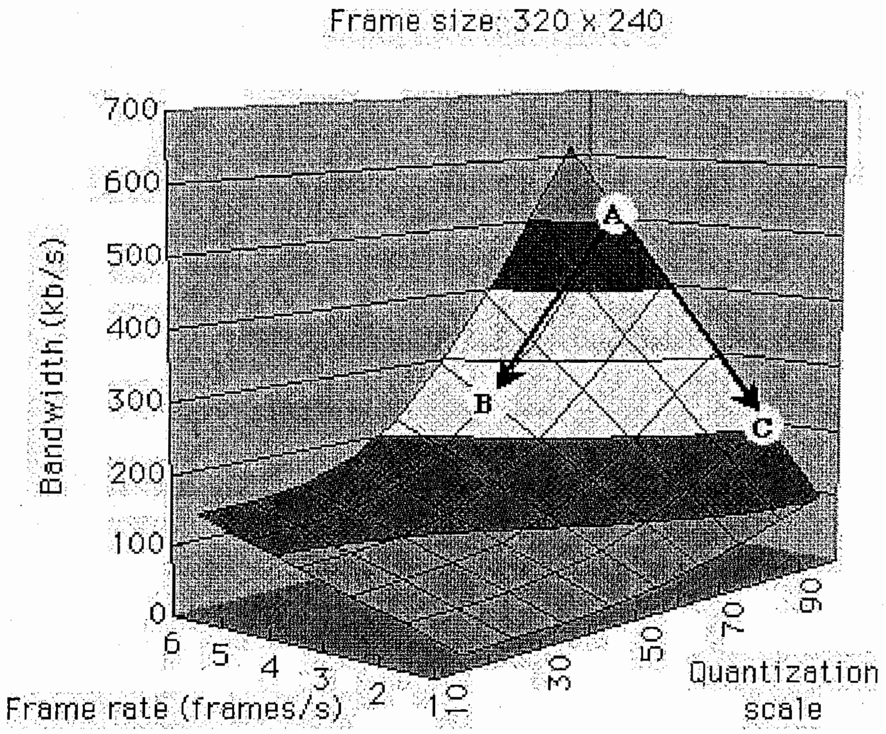


Figure 3.5: A QoS mapping result using spline functions from the application-level QoS into the resource-level QoS. Point A is an initial QoS requirements, and points B and C are new QoS candidates.

further study.

3.5 Communication Environment with Multiple Receivers

3.5.1 System Overview and Architecture

The CCS can be applied to a communication environment with multiple receivers shown in Fig. 3.6. In Fig. 3.6, Server means a video server, and Receiver means a video receiver (client). The CCS manages multiple receivers as its children, and each receiver requests the server of a video stream via its parental CCS. The server sends the video stream to the receiver via the CCS, and the CCS transforms the QoS parameters of the video stream according to the QoS requirements from the receiver and the available system resource amounts. The transformation that the CCS performs includes modulation of coding, and the QoS parameters used in the application level are image size S, frame rate F, and quantization scale Q as mentioned before. Each application QoS has its own priority value. These priority values are different for respective end-system (receiver), and reflect the user's individual preference.

The resource-level QoS also consists of the network QoS and the terminal QoS. Hereupon it is assumed that each resource QoS has its upper and lower thresholds. If the resource utilization is between two thresholds, the utilization situation is considered good (stable).

Compared to the case of one sender and one receiver, the CCS system for the multiple receiver communication environment is more complicated. Figure 3.7

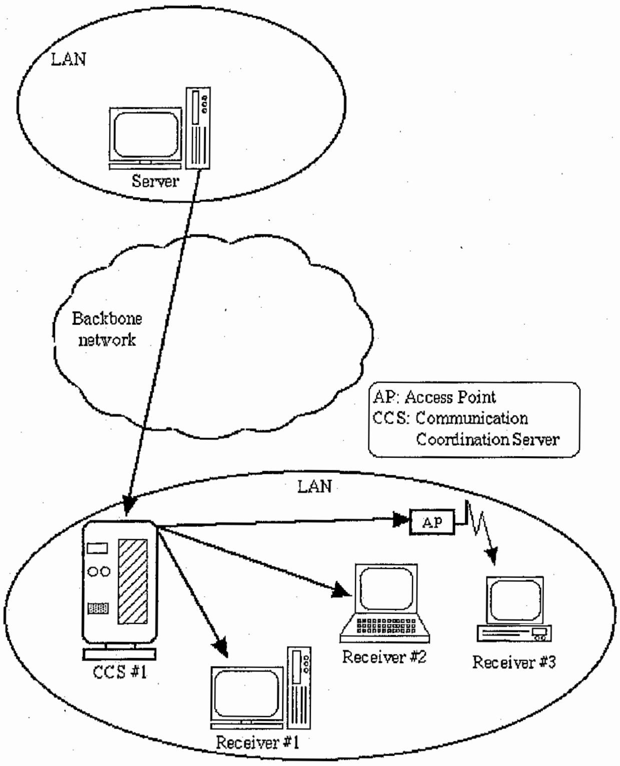


Figure 3.6: An example of communication environments by the proxy server.

shows QoS adjustment modules in the system. Since the server does not concern the QoS adjustment directly, only the video CODEC module is shown at the server site. The receiver has the user interface module and the resource monitoring module that monitors the resource utilization related to the receiver. The CCS has the QoS admission module that determines the QoS for every receiver under control. The QoS admission module consists of the QoS mapping module that relates the application QoS with the resource QoS and the QoS calculation module that computes a feasible QoS within the current resource availability. The CCS also has the resource monitoring module whose difference from the receiver's module is that the CCS's module has a function that measures the available bandwidth of the network domain to which the CCS belongs.

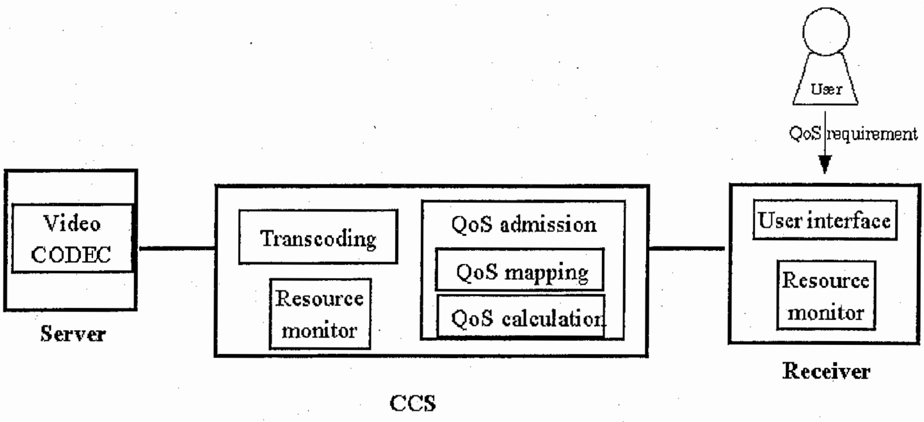


Figure 3.7: QoS adjustment modules.

3.5.2 QoS Adjustment Mechanism by CCS

The CCS and the receivers regularly monitor the system resource utilization, and a QoS adjustment signal is issued when the monitored value exceeds the upper threshold (overuse case) or becomes smaller than the lower threshold (underuse case). The QoS adjustment mechanism differs depending upon which terminal issues the signal and which threshold is concerned. In the following three subsections, we develop three QoS adjustment mechanisms for the situations to be assumed. Then the QoS adjustment method performed by the CCS is explained in the last subsection.

*A. **Over/underuse case at the receiver site***

The first case of the QoS adjustment mechanism is the overuse or the underuse at the receiver site. Figure 3.8 shows a messaging sequence between the CCS and the receiver. When Receiver #1 judges the resource utilization out of the thresholds, it issues a QoS adjustment request to the CCS. Simultaneously Receiver #1 notifies the CCS of its own terminal priority, application QoS priorities, QoS upgrade widths, and the resource information. The QoS upgrade widths are used for the QoS adjustment by the CCS, and it is explained later. The CCS calculates the updated and feasible QoS parameters based on the information of the QoS and the resources provided by Receiver #1, and then notifies Receiver #1 of the admitted QoS parameters.

B. Overuse case at the CCS site

The second case of the QoS adjustment mechanism is the overuse at the CCS site. Figure 3.9 shows a messaging sequence between the CCS and the receivers. When the CCS judges the overuse of resource utilization, it issues a QoS degradation claim to the receiver who has the lowest terminal priority, Receiver #1 in the case of Fig. 3.9. Receiver #1, who received the claim, issues a QoS degradation request to the CCS, as well as notifies the information of the QoS and the resources. The

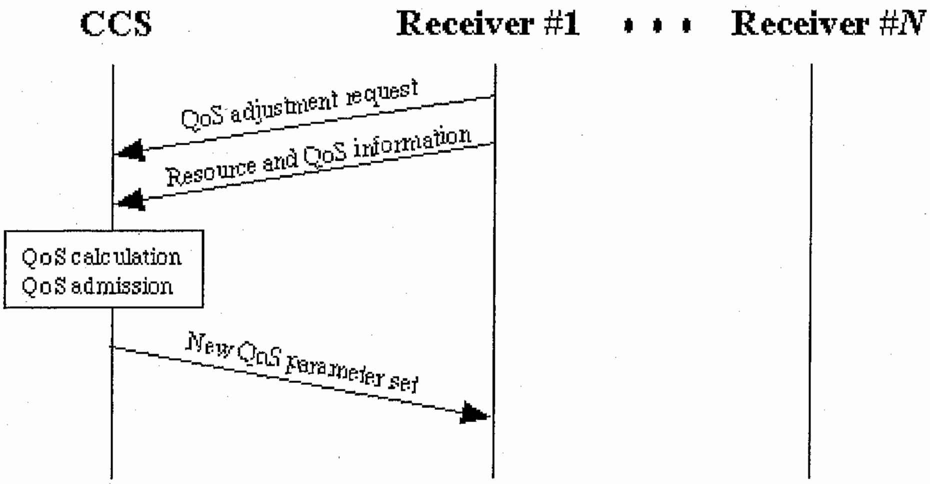


Figure 3.8: The message sequence (the over/underuse case at the receiver site).

CCS calculates the updated and feasible QoS parameters as described before, and notifies Receiver #1 of the admitted QoS parameters. Also the terminal priority of Receiver #1 is increased with a predefined constant. If the resource utilization becomes stable, that is between the two thresholds, the QoS adjustment is halted. Otherwise, the CCS again issues a QoS degradation claim to the receiver with the lowest priority at this moment to continue the QoS adjustment.

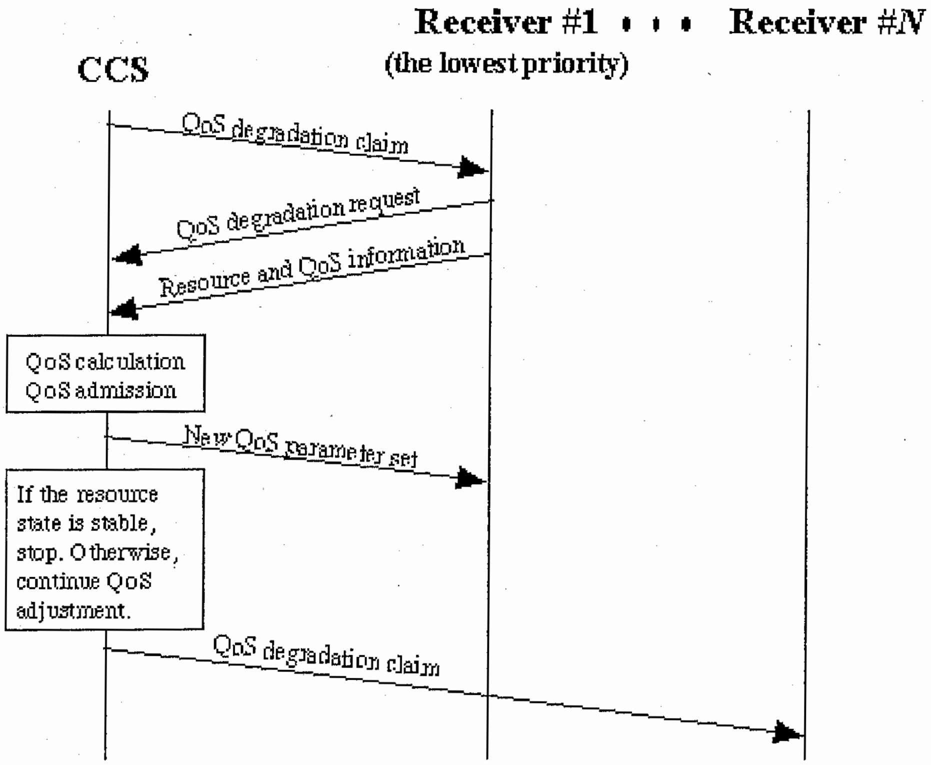


Figure 3.9: The message sequence (the overuse case at the CCS site).

C. Underuse case at the CCS site

The third case of the QoS adjustment mechanism is the underuse at the CCS site. Figure 3.10 shows a messaging sequence between the CCS and the receivers. When the CCS judges the underuse of resource utilization, it issues a QoS improvement permission to the receiver who has the highest terminal priority, Receiver #1 in the case of Fig. 3.10. Simultaneously the terminal priority of Receiver #1 is decreased with a predefined constant. Receiver #1, who received the permission, issues a QoS improvement request to the CCS, and notifies the information of the QoS and the resources, if its resource is in the underuse status. When the CCS receives the QoS improvement request, it calculates the feasible QoS parameters, and notifies Receiver #1 of the admitted QoS parameters. If the resource utilization becomes stable at the CCS site, the QoS adjustment is halted. Otherwise, the CCS again issues a QoS improvement permission to the receiver with the highest terminal priority at this moment to continue the QoS adjustment.

D. QoS adjustment by the CCS

Hereupon, the QoS adjustment mechanism performed by the CCS is described. When the CCS performs the QoS adjustment, it receives the information of the QoS and the resource $(R_n^{UT}, R_n^{LT}, qos_l^0, \Delta qos_l)$ from a receiver. R_n^{UT} or R_n^{LT}

means the upper or lower threshold for the n-th resource, qos_l^0 is the l-th application QoS parameter value, and Δqos_l is the QoS upgrade width for the l-th application QoS parameter, where $n = 1, \dots, N$ and $l = 1, \dots, L$. It is supposed that the QoS parameters are sorted in the decreasing order of the priority values. Namely qos_1^0 is the QoS parameter that has the highest priority, while qos_L^0 is the QoS parameter that can be mostly compromised for its quality.

The CCS has the spline QoS mapping function $M_n()$, which estimates the n-th

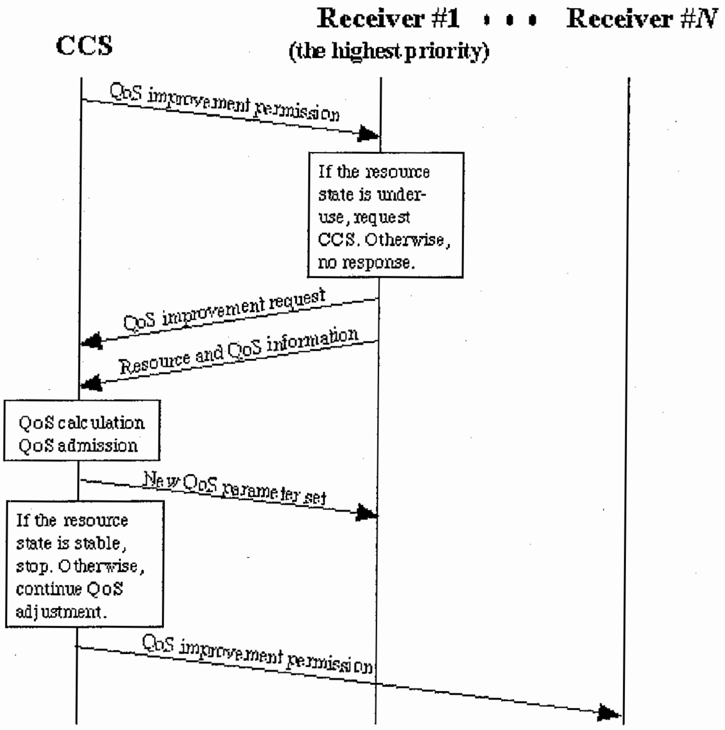


Figure 3.10: The message sequence (the underuse case at the CCS site).

resource amount to be needed based on a specified application QoS parameters [18]. The estimated n-th resource amount $R_n^{E(0)}$ is given as

$$R_n^{E(0)} = M_n(qos_1^0, \ldots, qos_L^0). \tag{3.2}$$

The CCS calculates the updated QoS parameter set so that the following equations are met

$$R_n^{LT} < R_n^E < R_n^{UT}, \tag{3.3}$$

$$R_n^{E(N)} = M_n(qos_1^N, \ldots, qos_L^N). \tag{3.4}$$

There are, however, multiple QoS parameter sets that meet these equations. A QoS adjustment algorithm based on the priority values of the application QoS parameters. The algorithm is as follows. Initially, the QoS parameter set is initialized as the present QoS set $(qos_1^0, \ldots, qos_L^0)$. If the QoS improvement is needed, qos_1^0 is improved with Δqos_1 so that the new QoS set $(qos_1^1, \ldots, qos_L^1) = (qos_1^0 + \Delta qos_1, \ldots, qos_L^0)$. If the QoS degradation is needed, qos_L^0 is degraded with Δqos_L so that the new QoS set $(qos_1^1, \ldots, qos_L^1) = (qos_1^0, \ldots, qos_L^0 - \Delta qos_L)$. After QoS improvement or degradation has been carried out, the judgment of resource status expressed in (2) is executed. If the judgment is approved, that is the resources are within two thresholds, QoS adjustment finishes. Otherwise, QoS improvement or degradation continues. The flow of the QoS adjustment algorithm is depicted in Fig 3.11.

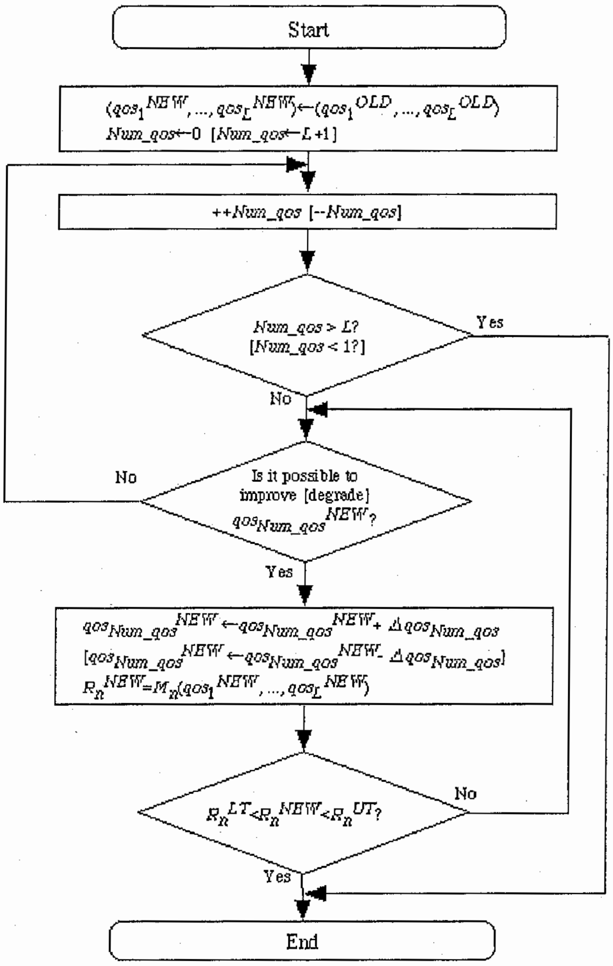


Figure 3.11: Flowchart of QoS adjustment policy algorithm.

3.6 Prototype Implementation

3.6.1 Single Receiver Case

We have implemented a prototype system of the proposed transcoding architecture of the single receiver case in a laboratory testbed. The testbed comprises a video sender, a receiver, and a CCS. Each terminal is a PC-AT machine (Windows NT) with 266 MHz Intel Pentium processor II and a 64 MB memory. All PC-AT machines are connected over an ATM network by an ATM-switch (CISCO LightStream 1010) that gives the maximum network throughput of 155 Mb/s. In this implementation, the MPEG-1 is used as an example of the inter-frame compression, and the M-JPEG is used as an example of the intra-frame compression. The receiver is assumed to be equipped with only an M-JPEG decoder because of its lightweight processing. The CCS is equipped with an MPEG decoding board and an M-JPEG encoding board. Decoded video data are directly transferred into the M-JPEG encoding board. We deal with the case of video archives in this implementation, that is the sender files MPEG-1 video data in its local disk.

QoS management experiments were conducted according to the flow shown in Fig. 3.4 on the implementation. In the experiments, the CCS received a MPEG-1 video stream (352 × 240 pixels and 30 frames/s) from the sender, and

Table 3.1: Experimental results of relationship between the application-level QoS and the resource-level QoS on the testbed implementation.

		Frame Size: 160 x 120			Frame Size: 320 x 240				Frame Size: 640 x 480	
		Frame Rate (frames/s)			Frame Rate (frames/s)				Frame Rate (frames/s)	
		1	5	11	1	2	3	5	1	2
Quantization Scale	10	12.0 5.5	58.9 24.0	130.0 49.0	23.3 10.3	45.5 19.2	67.7 28.2	112.6 46.3	67.3 31.7	136.0 64.5
	50	17.7 5.5	78.6 24.5	173.5 50.0	38.4 10.7	75.6 20.2	112.5 29.9	184.8 50.4	106.3 31.9	188.0 65.3
	90	35.4 6.0	142.5 26.5	312.3 56.0	105.6 11.8	206.5 21.7	307.7 32.4	511.5 57.7	345.4 36.3	613.0 73.5

transcoded it into an M-JPEG stream by the application-level QoS requirement from the receiver. Table 3.1 summarizes typical results of relationship between the application-level QoS (the frame size, frame rate, and quantization scale) and the resource-level QoS (the bandwidth and CPU utilization). In Table 3.1, the cells that include two figures show the resource QoS; the upper figure is the bandwidth (kb/s), and the lower figure is the CPU utilization (%). The rightmost frame rate column for each frame size is the maximum number of the frame rate realized on the implementation.

To verify the effectiveness of the QoS adaptation mechanisms to dynamical resource changes, the available resource decrease situations were simulated. One typical QoS adjustment result for bandwidth shortage is shown in Fig. 3.5 as described previously. We show another typical QoS adjustment experiment, where CPU resource shortage was simulated. At the beginning of the experiment, the available CPU utilization was 60%, and the CCS transcoded the MPEG-1 stream into an M-JPEG stream (320 × 240 pixels, 5 frames/s, and a quantization scale of 90) that consumes the CPU utilization of 57.7%. Then the available CPU utilization decreased to 30%. Immediately the CCS started QoS adjustment by degrading the QoS parameters according to the user's QoS policy. For example, if the policy permits only the frame rate to be decreased, the CCS decreases the frame rate to 2 frames/s. The other QoS parameters remained unchanged and the new QoS allocation consumed the CPU utilization of 21.7%. Similarly, if the policy permits only the frame size to be decreased, the CCS decreases only the frame size to 160 × 120 pixels and the new QoS allocation consumed the CPU utilization of 26.5%. Meanwhile, if the policy permits both the frame rate and the quantization scale to be decreased, the CCS decreases both QoS parameters and a set of 320 × 240 pixels, 3 frames/s, and a quantization scale of 50 can be selected as a new QoS allocation to realize the CPU utilization of 29.9%.

The QoS adjustment mechanism can be applied to the case that the available bandwidth and CPU utilization change simultaneously, by considering the most constrained resource as the limited resource.

3.6.2 Multiple Receivers Case

Next a prototype of the CCS architecture of the multiple receivers case has been implemented in the laboratory testbed. In this case, the testbed system consists of a video sender, two receivers, and a CCS shown in Fig. 3.12. Each terminal is a PC-AT machine (Windows NT), and all terminals are connected by Ethernet.

The QoS adjustment mechanisms of the CCS have been verified by several experimental results. One example is as follows. Initially two receivers had been receiving the same video stream from the video server via the CCS. The realized QoS were different according to the respective QoS requirements. $(S, F, Q) =$

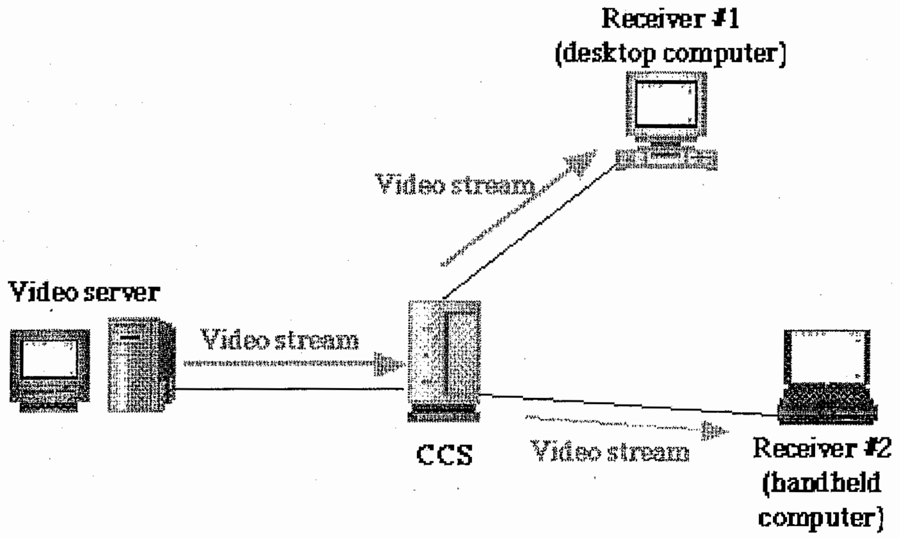


Figure 3.12: Configuration of the experimental system.

$(416 \times 336, 5, 11)$ for receiver #1, while $(S, F, Q) = (272 \times 216, 5, 20)$ for receiver #2. Then an urgent FTP stream occurred between the CCS and Receiver #2, and consequently the video QoS for receiver #2 was automatically degraded into $(S, F, Q) = (176 \times 144, 1, 20)$ by the QoS adjustment mechanisms to give priority to the FTP stream. When the FTP stream finished, the QoS for receiver #2 was improved to $(S, F, Q) = (416 \times 320, 1, 20)$ by the QoS adjustment mechanisms. The final QoS had been improved for its frame size, because the priority of the frame size was set to be higher than two other parameters.

3.7 Conclusion

In this chapter we have presented a QoS management architecture for distributed multimedia applications in heterogeneous communication environments. In the proposed architecture, a proxy server called CCS intermediates between a video sender and a receiver or a group of receivers and manages the QoS adjustment. The CCS monitors the currently available resources and receives the QoS and resource information from the receiver(s). Based on these information, the CCS calculates a feasible QoS for each receiver to utilize the system resources efficiently. Then the CCS carries out the transcoding to transform the video QoS to satisfy the receivers' requirements.

Prototype systems of the CCS have been implemented in a laboratory testbed. In the prototype systems, the transcoding mechanisms between MPEG and M-JPEG codings were implemented in hardware or software. With the prototype system, it is verified that the CCS can resolve the network and terminal heterogeneities between the sender and receiver sides by the transcoding and the QoS adjustment mechanism.

Digital television broadcasting service will start in the near future. While an interlace format is used for television video signal, a progressive format is used for computer video signal. Although MPEG-2-TSs (transport streams) used for video transmission in the digital television broadcasting service can deal both the interlace and progressive formats, quality control and application QoS adaptation are needed to merge the broadcasting service with multimedia communication services, where the CCS architecture would be applied to.

Which application QoS parameter should be adjusted primarily is an open issue and depends on video contents. Subjective video quality assessment tests were conducted, and it is found that subjective evaluation depends on the content of video stream or the object movement [19]. Still further study is needed to clarify the relationship between the application QoS and the user QoS which is the user's evaluation of media quality.

Bibliography

[1] C. Aurrecoechea, A.T. Campbell and L. Hauw, "A survey of QoS architectures," ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, vol.6, no.3, pp.138-151, May 1998.

[2] A. Campbell, G. Coulson, F. Garcia, D. Hutchison, and H. Leopold, "Integrated quality of service for multimedia communications," Proc. IEEE Infocom'93, pp.732-739, San Francisco, March 1993.

[3] L. Delgrossi, C. Halstrick, D. Hehmann, R.G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media scaling for audiovisual communication with the Heidelberg transport system," Proc. ACM MultiMedia 93, pp.99-104, Anaheim, June 1993.

[4] K. Ohta, T. Watanabe, and T. Mizuno, "Selective multimedia access protocol for wireless multimedia communications," Proc. 1997 IEEE Pacific

Rim Conference on Communications, Computers and Signal Processing (PACRIM'97), pp.81-84, Aug. 1997.

[5] E. Amir, S. McCanne, and H. Zhang, "An application level video gateway," Proc. ACM Multimedia '95, pp.255-265, San Francisco, CA, Nov. 1995.

[6] T. Warabino, S. Ohta, D. Morikawa, M. Ohashi, H. Nakamura, H. Iwashita, and F. Watanabe, "Video transcoding proxy for 3Gwireless mobile Internet access," IEEE Commun. Mag., vol.38, no.10, pp.66-71, Oct. 2000.

[7] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," RFC 1889, Internet Engineering Task Force, Feb. 1996.

[8] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer, "Adapting to network and client variation using infrastructural proxies: lessons and perspectives," IEEE Personal Communications, vol.5, no.4, pp.10-19, 1998.

[9] T. Nakajima and A. Hokimoto, "Adaptive continuous media applications in mobile computing environments," Proc. IEEE International Conference on Multimedia Computing and Systems (ICMCS'97), pp.152-160, June 1997.

[10] "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s," ISO/IEC IS 11172-2, 1993.

[11] ISO/IEC JTC1/SC29/WG11, "Generic coding of moving pictures and associated audio," ISO/IEC CD 13818-2, 1993.

[12] "Video codec for audiovisual services at p × 64kbit/s," ITU-T Recommendation H.261, 1993.

[13] "Video coding for low bit rate communication," ITU-T Recommendation H.263, 1996.

[14] K. Rijkse, "H.263: Video coding for low-bit-rate communication," IEEE Commun. Mag., pp.42-45, Dec. 1996.

[15] G. K. Wallace, "The JPEG still picture compression standard," Commun. ACM, vol.34, no.4, 1991.

[16] G. Nurnberger, J. W. Schmidt, and G. Walz (eds.), Multivariate Approximation and Splines, Birkhauser Verlag, Basel, 1997.

[17] A. Dutta-roy, "Networks for homes," IEEE Spectrum, vol.36, no.12, pp.26-33, Dec. 1999.

[18] T. Yamazaki and J. Matsuda, "On QoS mapping in adaptive QoS management for distributed multimedia applications," Proc. ITC-CSCC'99, vol.2, pp.1342-1345, July 1999.

[19] T. Yamazaki, "Subjective video quality assessment for adaptive quality-of-service control," Proc. ICME2001 (2001 IEEE International Conference on Multimedia and Expo), pp.517-520, Aug. 2001.

Chapter 4

QoS Control with Multi-agent System

4.1 Introduction

Recently, it has been demonstrated that the multi-agent system can be applied for QoS management to distributed multimedia applications, to which centralized systems cannot be applied, because of its useful features such as parallelism, robustness, and scalability [5], [6]. Suganuma et al. [7] proposed the ADIPS (Agent-based Distributed Information Processing System) and developed a video-conferencing system based on it. Puliafito et al. [8] presented an agent-based

QoS management framework, but concrete QoS negotiation and adaptation mechanisms were missing in their discussion.

Meanwhile, Aurrecoechea et al. [1] proposed a generalized QoS framework based on a set of principles that govern the behavior of QoS architectures. The generalized QoS framework is composed of three QoS mechanisms: QoS provision mechanisms, QoS management mechanisms, and QoS control mechanism. QoS provision mechanisms perform static resource management in the flow establishment and QoS renegotiation phases. On the other hand, QoS management and control mechanisms deal with dynamic resource management in the media-transfer phase. QoS control is distinguished from QoS management by an operational time-scale. QoS control operates on a faster time-scale than QoS management.

In this chapter, we propose an adaptive QoS management framework for distributed multimedia based on the multi-agent system and the generalized QoS framework. In the proposed framework, the agents directly or indirectly collaborate to adaptively manage the media QoS according to the available network and terminal resources as well as the user requirements. The particular point of the framework is that it provides 2-tier QoS management. Namely the global and long-term QoS adaptation is executed in one tier, while the local and short-term

QoS adjustment is executed in the other tier. A one-way video system is developed on the basis of the proposed framework as an example of communication-intensive applications.

4.2 QoS Management Based on the Generalized QoS Framework

Aurrecoechea et al. [1] proposed a generalized QoS framework based on a set of principles that govern the behavior of QoS architectures. The generalized QoS framework is composed of three QoS mechanisms: QoS provision mechanisms, QoS management mechanisms, and QoS control mechanism. QoS provision mechanisms perform static resource management in the flow establishment and QoS renegotiation phases. On the other hand, QoS management and control mechanisms deal with dynamic resource management in the media-transfer phase. QoS control is distinguished from QoS management by an operational time-scale. QoS control operates on a faster time-scale than QoS management.

Figure 4.1 presents a QoS management flow on the basis of the generalized QoS framework. In the flow establishment and renegotiation phases, the QoS mapping module translates user QoS requests into QoS candidates that are understandable for the system (terminals and networks). The QoS negotiation module selects the QoS for each media stream from the QoS candidates via intra-terminal and inter-terminal negotiations. The QoS admission module tests whether the selected QoS will be guaranteed or not for the system, and reserves the system resources by resource reservation protocols if possible. Otherwise, the QoS admission module issues a renegotiation message for the QoS negotiation module. Then the modules in the media-transfer phase succeed to the QoS management. The selected QoS is transferred to QoS control and management mechanisms. A real-time flow control module in the QoS control mechanism tries to maintain the QoS through flow filtering, flow shaping, flow scheduling, and so forth. In the QoS management mechanism, the QoS monitoring module perceives fluctuations in system resources, and notifies the QoS management module. The QoS management module deals with the QoS adjustment within the admissible range, which is specified by the user, using the resource information. The real-time flow control module receives the adjusted QoS, and continues QoS maintenance. When the QoS management module can no longer perform the QoS adjustment because, for example, the resource fluctuation is too severe to recover, it issues a renegotiation request message to the QoS negotiation module.

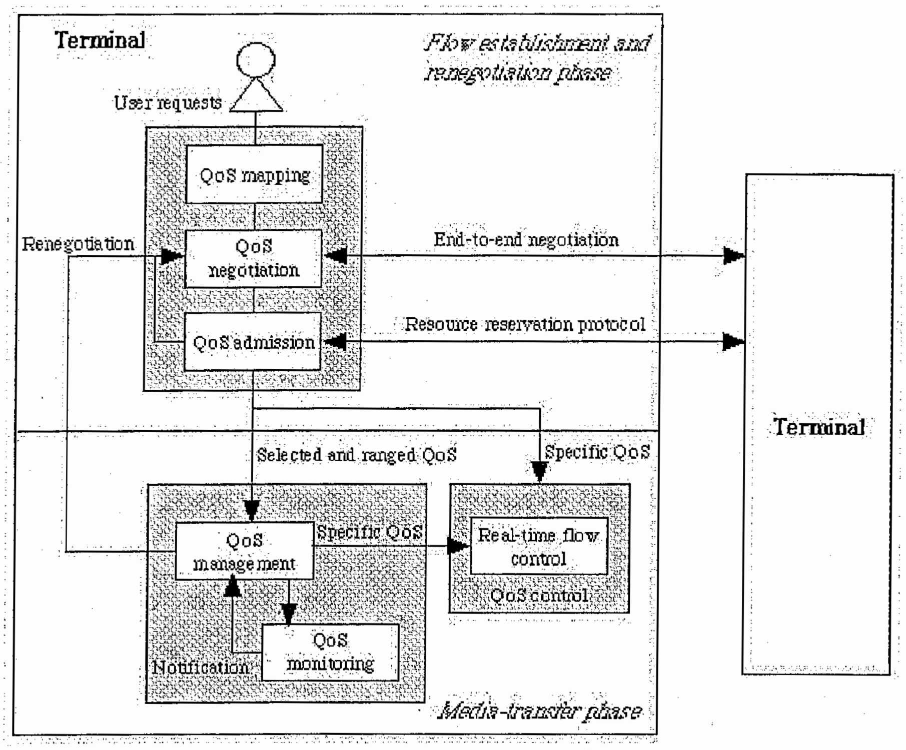


Figure 4.1: A QoS management flow on the basis of the generalized QoS framework.

4.3 An Agent-based Adaptive QoS Management Framework

We propose an agent-based adaptive QoS management framework called MARM (Multi-Agent Resource Management) framework as a common platform for various communication-intensive applications (Fig. 4.2). The MARM framework,

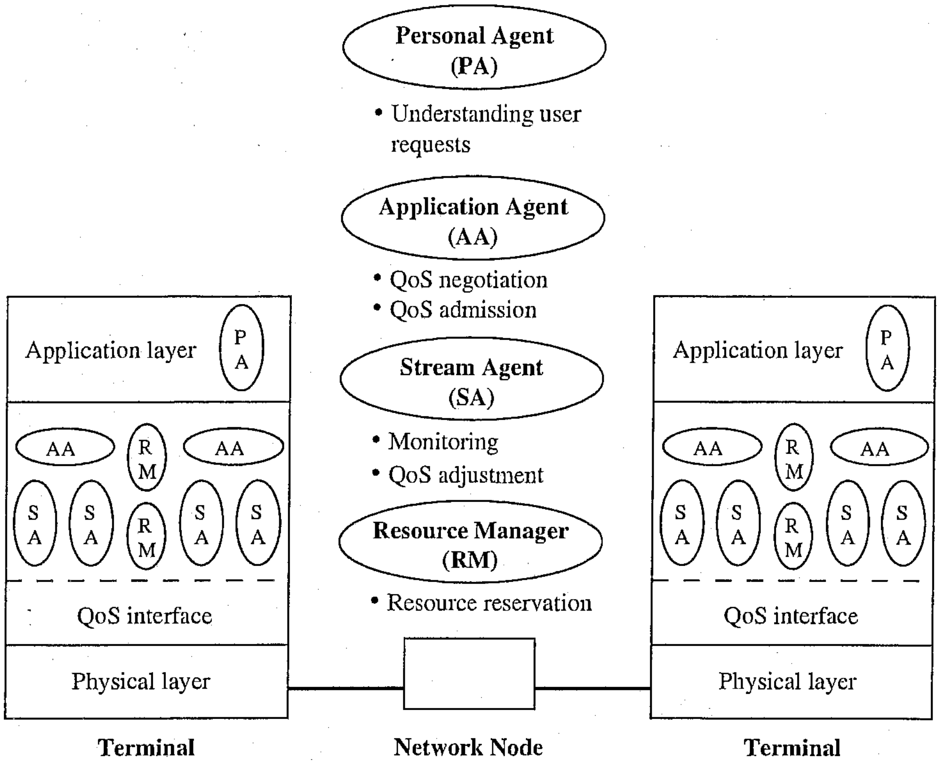


Figure 4.2: An agent-based adaptive QoS management framework.

whose core is the QoS management flow presented in Fig. 4.1, consists of three kinds of agents, a resource manager module, and a QoS interface.

4.3.1 Personal Agent

When a user has little knowledge on how to set the application-level QoS, the user might provide abstract QoS requests for media streams. For example, the user may utter “I want to view a video with a middle-size, fast rate, and fair quality.” A Personal Agent (PA) then interprets the abstract user-level QoS using the user profile database, which reflects the user’s preference, and transfers it to the QoS provision mechanisms as the application-level QoS. PA also updates the user profile database using a learning mechanism.

4.3.2 Application Agent

An Application Agent (AA) selects the best viable QoS for each media stream from the application-level QoS candidates by intra-terminal and inter-terminal negotiations. By the best viable, we mean the QoS that maximizes the total user utility under resource constraints. Since QoS negotiations do not need to be executed in real-time, it is desirable that AA is deliberative to take optimality into consideration from a global and long-term viewpoint. We define deliberative agents as the agents that can directly communicate with each other and utilize knowledge to make a decision. In [9], research on various planning mechanisms used in deliberative agent architecture is introduced.

4.3.3 Stream Agent

A Stream Agent (SA) adjusts the selected QoS within the admissible range specified by the user. Since QoS adjustment by the SA is carried out while the multimedia applications are in operation, it must be done in real-time. Hence SA must be reactive. We define reactive agents as agents that have simpler construction than deliberative agents and behave according to their environment using distributed and decentralized interactions between agents without any explicit knowledge or inference mechanisms. One of the most famous reactive agents is the subsumption architecture by Brooks [10]. By using a kind of the blackboard architecture [13], we have implemented the reactive SA behavior.

4.3.4 Resource Manager and QoS Interface

A Resource Manager (RM) performs scheduling and reservation for the terminal resources, such as CPU and memory. The QoS interface mediates between terminals and networks, and enables the terminal to reserve the network resources. Instead of developing a new QoS interface, it is possible to utilize the existent QoS interface architecture deployed in the OMEGA [2] or the QoS-A [4].

4.4 A One-way Video System with Adaptive QoS Management

We have designed a one-way video system called MARM-Video1 (Multi-Agent Resource Management Video1) on the basis of the MARM framework. Figure 4.3 depicts the interrelationship between the MARM-Video1 modules. In this section, we will describe the agents' behaviors according to the QoS flow.

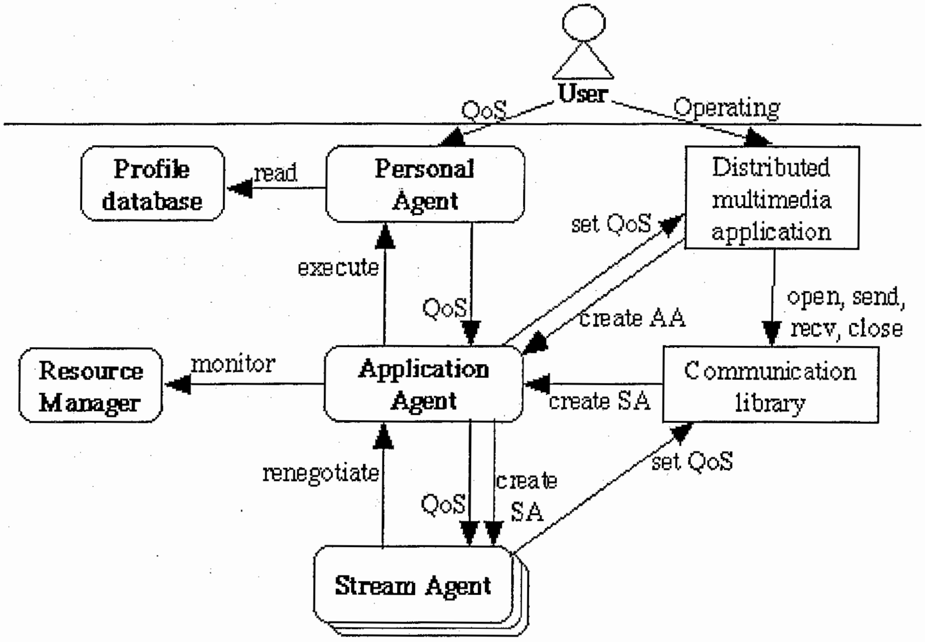


Figure 4.3: The module interrelationship in the MARM-Video1 (Multi-Agent Resource Management Video1).

4.4.1 QoS Specification

In the MARM-Video1, a user specifies multiple ranged QoS candidates. Figure 4.4 shows an example of multiple QoS candidates for a video stream, and each

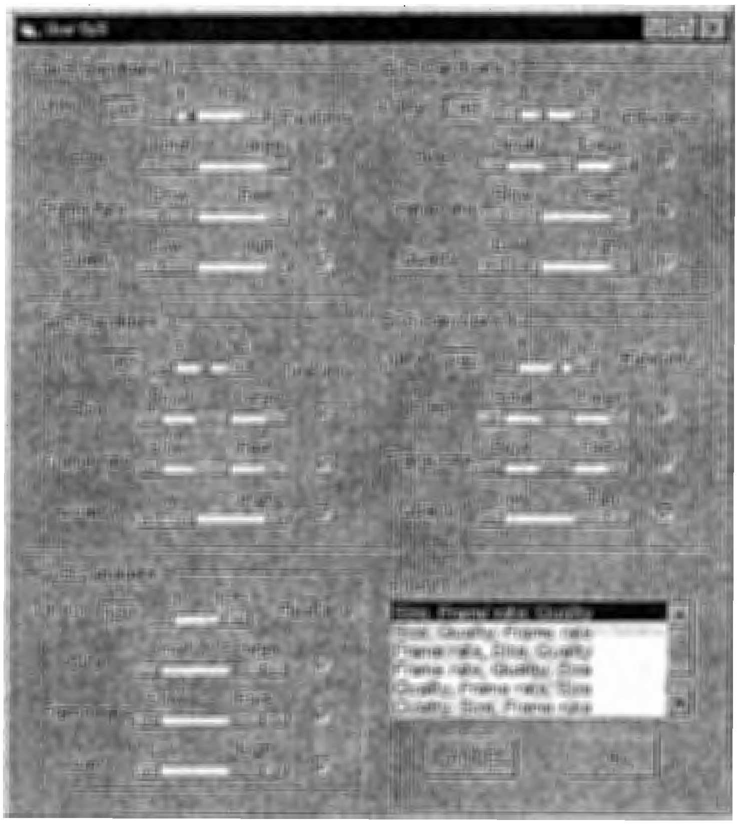
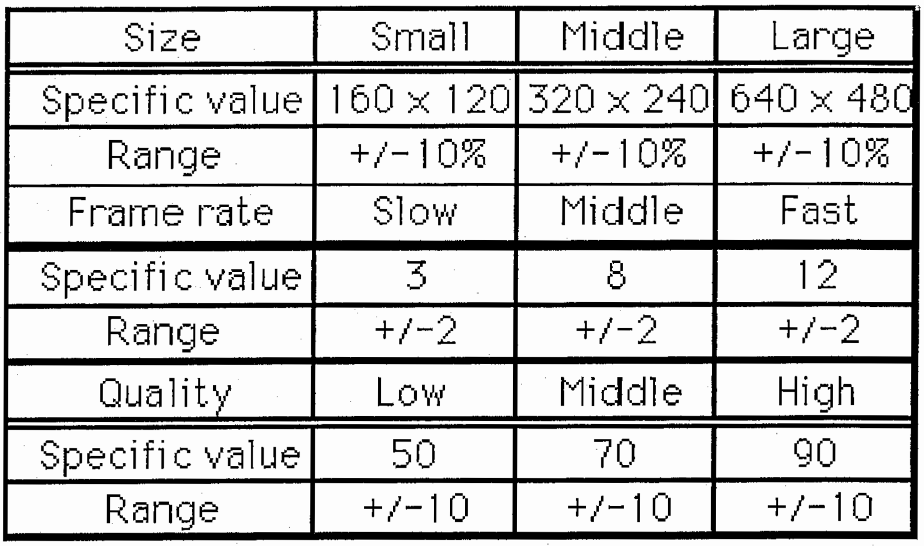


Figure 4.4: An example of setting of multiple QoS candidates for a video stream. A user can set five QoS candidates from this font-end, and each QoS candidate has three QoS parameters (the size, frame rate, and quantization scale). The user can specify the priority order of the QoS parameters by choosing one from the priority policies (the lower right window).

Table 4.1: An example of specific profile database, which presents the relationship between user's abstract QoS expression, such as *small* or *slow*, and the application-level QoS defined by a specific value and a range.



Size	Small	Middle	Large
Specific value	160 × 120	320 × 240	640 × 480
Range	+/-10%	+/-10%	+/-10%
Frame rate	Slow	Middle	Fast
Specific value	3	8	12
Range	+/-2	+/-2	+/-2
Quality	Low	Middle	High
Specific value	50	70	90
Range	+/-10	+/-10	+/-10

QoS candidate has three media-specific QoS parameters (the size, frame rate, and quantization scale) that are expressed abstractly. PA translates the QoS candidates into the application-level QoS using a user-specific profile database such as that in Table 4.1. The user profile database stores the mapping data between the user's abstract expression and specific values within a range, and the data can be updated by learning.

The user specifies the priority order of QoS parameters by choosing one from the prepared priority policies. Examples of the priority policies include a definitive order, such that the size is the first, the frame rate is the second, and the quantization scale is the third, and the order in which the lowest priority is given for the highest quality of QoS parameter. The priority order of QoS parameters is used when SAs perform the QoS adjustment that will be described later. The utility parameter is given for each QoS candidate, and represents the user's satisfaction when the QoS candidate is selected. The larger the value of the utility parameter is, the higher the user's satisfaction. In Fig.4.4, if Flexibility option button is on, the range value in Table 4.1 is used, otherwise the range is set to 0 by compulsion and the application parameter has no range.

Two priority parameters, ranging from 1 to 100, are offered in the MARM-Video1. One is the application priority, which represents the rank of application among all applications in the terminal. The other is the stream priority, which represents the rank of stream among all streams managed in the application.

4.4.2 QoS Selection

AAs select the best viable QoS from the multiple QoS candidates by intra- and inter-terminal negotiations. In the intra-terminal negotiation, the AAs negotiate the allocation of resources that maximizes the total user utility. The procedure of the QoS negotiation is as follows. The AA who requests an intra-terminal negotiation sends a QoS negotiation request message to all AAs concerned on the terminal. The AA who sends the request message is called the master agent. If an AA receiving the request message can participate in the negotiation, it returns its multiple QoS candidates and utility parameters to the master agent. The master agent selects a QoS set for the streams so that the total utility U defined in (4.1) is maximized under the resource constraint conditions in (4.2).

$$U = \sum_{S} w(S) \log u(S, q), \tag{4.1}$$

$$\sum_{S} r_m(S, q) \leq R_m, \tag{4.2}$$

where $u(S, q) \in (1, 100)$ is a utility parameter when a stream S has a QoS of q, $w(S)$ is the priority of stream S taking the priority of the concerned application into consideration, and $r_m(S, q)$ is the amount of the m-th resource required by processing of stream S with QoS q, and R_m indicates the maximum availability of the m-th resource. In (4.1) and (4.2), the summation is operated for all streams involved in the negotiation.

The utility parameter $u(S, q)$ presents the user's satisfaction when stream S has a QoS of q. In the MARM-Video1, the utility parameters are specified by respective users manually. Assuming that users are more sensitive to quality degradation than to quality improvement, we have introduced nonlinear property by logarithms.

After the intra-terminal QoS negotiation, the AAs execute inter-terminal QoS negotiation to resolve QoS conflicts between the stream sender and the receiver. As a possible solution for the QoS conflicts, we have proposed an inter-terminal negotiation using the cooperative game theory, where each AA exchanges the terminal's utility, which is defined in advance, and finds a compromise between the sender and the receivers [11]. Another simple solution is to select the lowest QoS among the QoS selections from the sender and the receivers. Which solution should be chosen depends on the communication situation, e.g. the number of terminals involved, or the number of resource to be considered. In the following experiments, the latter simple solution was chosen because of its implementation facility.

4.4.3 QoS Adjustment

After a ranged QoS candidate is selected by the AAs, the SAs residing in the same terminal adjust the selected QoS parameters within the range provided by PAs, and determine a specific QoS for each media stream. We adopt a kind of the blackboard architecture [13] for the QoS adjustment mechanism, where the SAs use priority parameters to order the QoS adjustment, and a priority threshold parameter Th is shared among the SAs as a common datum.

Figure 4.5 shows the QoS adjustment procedure according to the asynchronous and autonomous behaviors of the SAs. An SA monitors resources. If the SA recognizes a shortage or surplus (excess) of resources, it refers to Th. By comparing its own priority parameters with Th, it decides whether to execute the QoS adjust-

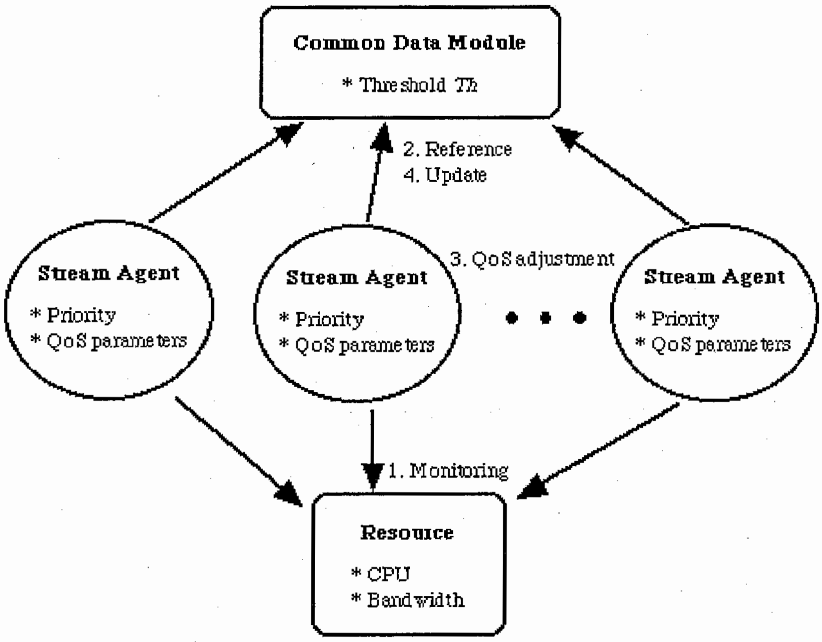


Figure 4.5: QoS adjustment procedure by the SAs. An SA monitors the resources. If QoS adjustment is needed, the SA refers to the threshold *Th*, and decides whether to execute the QoS adjustment by comparing its stream priority with *Th*. If the QoS adjustment is executed, the SA updates *Th*. The SAs behave asynchronously and autonomously.

ment or not. If the SA undertakes the QoS adjustment (adjusts QoS), it updates the value of Th after the adjustment. The behavior of the SA for the QoS adjustment differs according to whether there is a resource shortage or a resource surplus. For example, in a resource shortage, the SA degrades (decreases) the QoS parameter stepwise according to the ascending priority order of the QoS parameters (by the increasing QoS parameter priority order, and then the SA increases the value of Th). After the QoS adjustment, the SA increases the value of Th. In a resource surplus, the SA upgrades (increases) the QoS parameter stepwise according to the descending priority order of the QoS parameters (by the decreasing QoS parameter priority order, and then the SA decreases the value of Th). After the QoS adjustment, the SA decreases the value of Th. To this end, only low priority streams participate in the QoS adjustment when the resource shortage or excess is small, while higher priority streams also participate in the QoS adjustment when the resource insufficiency is quite large. When all streams fail to adjust QoS because the resource insufficiency is too large, the re-negotiation phase is invoked and the application agent negotiation starts to select new QoS for the media streams.

Th's initial value is set to the maximum, minimum, or average value of all priority parameters. Th updating is done by increasing (decreasing) a constant value or by setting to the average value of the priority parameters of the SAs excluded from the QoS adjustment process. How Th is specified and used is described concretely in Section 4.5.4.

The QoS adjustment by stream agents is carried out reactively in media-transfer phase, and it corresponds to the QoS management mechanisms in [1].

4.5 Experiments

To verify the behaviors and linkage of the agents in the MARM-Video1, we designed a computer simulation system, which is called a simulator in the following, on PC-AT machines, and conducted a set of experiments on the simulator.

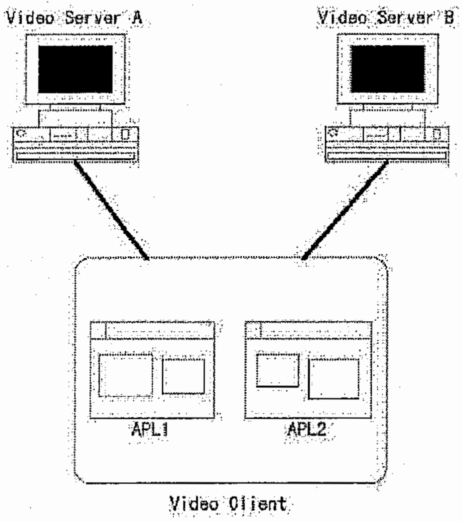
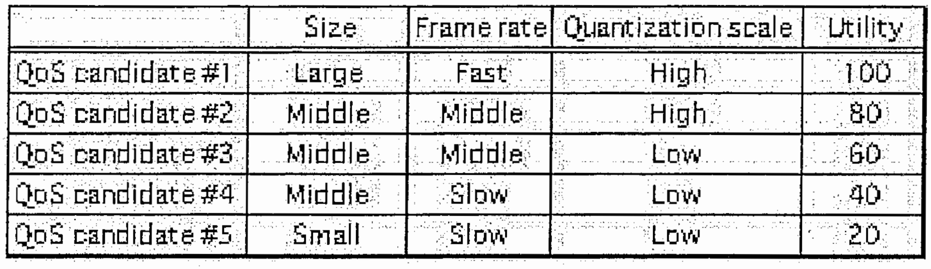


Figure 4.6: The experimental network configuration in computer simulations.

Table 4.2: Abstract user-level QoS for video streams used in the experiments.



	Size	Frame rate	Quantization scale	Utility
QoS candidate #1	Large	Fast	High	100
QoS candidate #2	Middle	Middle	High	80
QoS candidate #3	Middle	Middle	Low	60
QoS candidate #4	Middle	Slow	Low	40
QoS candidate #5	Small	Slow	Low	20

4.5.1 Configuration

Figure 4.6 illustrates the network configuration of the experiment, where a client is connected to two video servers. One MARM-Video1 server application runs on each server and two MARM-Video1 applications run on the client. It is assumed that multiple video streams are coming from each sender to the client, and each video stream is either a real-time video or an accumulated video.

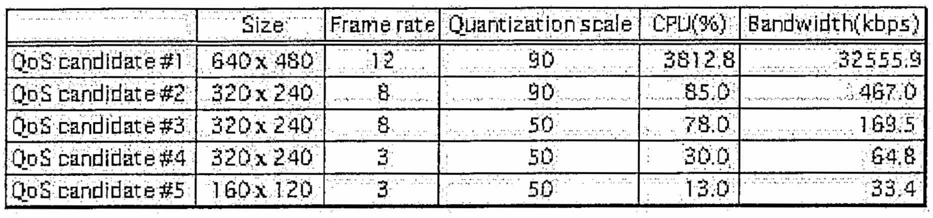
4.5.2 QoS Mapping

First, we describe how the QoS mapping mechanisms are executed in the simulator. The multiple video QoS candidates are provided by the user in an abstract expression as shown in Table 4.2 and the priority order of the QoS parameters is chosen as the frame rate, quantization scale, and size in the experiment. The size corresponds to the number of pixels in one frame, and it is represented as, for example, 320 × 240. The frame rate corresponds to the number of frames to be presented per second. The quantization scale is related to the quantization step width used in video coding methods, and it takes an integer value between 1 and 100. The smaller the quantization scale, the smaller the encoded video data size, but quantization scale values that are too small may make the user's evaluation low because of blurs or color defects in images.

The QoS mapping mechanisms proposed in Chapter 2 are used. In the user level, the multiple video QoS candidates are provided by the user in an abstract expression as shown in Table 4.2. Also the priority order of the QoS parameters is chosen by the user as the frame rate, quantization scale, and size in the experiment. Meanwhile, in the application level, the QoS needs to be expressed by a specific value. PA interprets the abstract user-level QoS (Size, Frame rate, Quantization scale) in Table 4.2 into the corresponding application-level QoS expressed by specific values by the user specific profile database in Table 4.1. Utility in Table 4.2 is the same as the utility parameter representing the user's satisfaction in Section 4.4.2. The parameter values in Table 4.2 were determined subjectively.

Moreover the application-level QoS needs to be translated into the required resource-level QoS. In the experiment, the CPU utilization rate consumed for decoding and presenting the video data and the bandwidth (throughput) needed to

Table 4.3: The QoS mapping results for the user-level QoS in Table 4.2. Size, Frame rate, and Quantization scale represent the application-level QoS, and CPU and Bandwidth represent the resource-level QoS.



	Size	Frame rate	Quantization scale	CPU(%)	Bandwidth(kbps)
QoS candidate #1	640x 480	12	90	3812.8	32555.9
QoS candidate #2	320x 240	8	90	85.0	467.0
QoS candidate #3	320x 240	8	50	78.0	169.5
QoS candidate #4	320x 240	3	50	30.0	64.8
QoS candidate #5	160x 120	3	50	13.0	33.4

send the encoded data between the sender and the receiver are selected as the resource-level QoS. The former is referred to as CPU in percentage, and the latter is referred to as Bandwidth in kilo-bits per second (kbps) in Table 4.3.

The QoS mapping (translation) between the application-level QoS and the resource-level QoS is done by the AA using the spline QoS mapping scheme [12]. The QoS mapping result between the application-level QoS and the resource-level QoS is shown in Table 4.3. Although a non-realistic value can be computed because of extrapolation, QoS candidates with non-realistic values should be excluded as impractical.

4.5.3 Global and Long-term QoS Negotiation

Next, we present a scenario to show the resource allocation during the AAs' QoS negotiation. In the scenario, the receiver operates two applications, APL1 and APL2. APL1 receives video streams from the video server A, APL2 receives video streams from the video server B, and each application can receive at most two streams. In this experiment, it is assumed that the bandwidths between the senders and the receiver and the senders' CPU capability are enough to enable all of four streams to select the best QoS candidate #1. Therefore only the receiver's CPU resource is taken account of as the system resource. The application priority values for APL1 and APL2 are 50 and 100, respectively. APL1 deals with Stream 1 and Stream 2 and APL2 deals Stream 3 and Stream 4. The stream priority values for Stream 1, 2, 3, and 4 are 90, 50, 90, and 50, respectively. Each time a stream increases or decreases, the QoS negotiation is carried out to determine a new resource allocation according to the application and stream priorities.

For the QoS negotiation, (4.1) and (4.2) are used to select a QoS set for the streams. $w(S)$ in (4.1) means the priority of stream S and is expressed as the product of a stream priority and an application priority, which is called the total priority in the following. In this experiment, the total priorities for four streams are computed as $w(Stream1) = 90 \times 50 = 4500$, $w(Stream2) = 50 \times 50 = 2500$,

$w(Stream3) = 90 \times 100 = 9000$, $w(Stream4) = 50 \times 100 = 5000$. $u(S, q)$ in (4.1) is a utility parameter and Utility in Table 4.2 is used in this experiment. (4.2) means resource constraints and the receiver's CPU resource is evaluated in this experiment. Maximization of (4.1) was carried out by searching all combinations of QoS candidates and the number of combinations was at most 54 in this experiment.

Figure 4.7 shows how the CPU resource on the receiver is allocated to the streams as the number of streams changes. Initially no streams are transmitted, and at 3s into the scenario Stream 2 is transmitted with QoS #2 (in Table 4.3), which is the best viable QoS. At 12s into the scenario, Stream 1 transmission begins and the QoS for Stream 1 and 2 are selected as #2 and #5, respectively, because the priority of Stream 1 is larger than that of Stream 2. At 26s into the scenario, Stream 3 transmission from terminal B starts and three streams renegotiate QoS to conclude that all streams share the same QoS #4. Then, at 36s into the scenario, Stream 4 transmission starts, and QoS renegotiation among four streams allocates QoS #5 for Stream 1 and 2 and QoS #4 for Stream 3 and 4 by taking the application and stream priority values into consideration. At 64s into the scenario, Stream 3 transmission is terminated and the same QoS #4 is allocated to the remaining three streams. At 68s into the scenario, Stream 4 transmission is terminated, and QoS #2 and #5 are selected for Stream 1 and 2, respectively, according to the stream priority. At 81s into the scenario, Stream 1 transmission is terminated, and Stream 2 retrieves QoS #2. Finally at 89s into the scenario, all of the stream transmissions terminate.

From this experimental result, it is found that QoS allocation has been fairly done according to the priority values, because we assumed that the senders' CPU capability are enough. However, when the sender's CPU capability is limited, we have to use the inter-terminal QoS negotiation mechanism described in Section 4.4.2. Let us consider one example of the inter-terminal QoS negotiation. Assum-

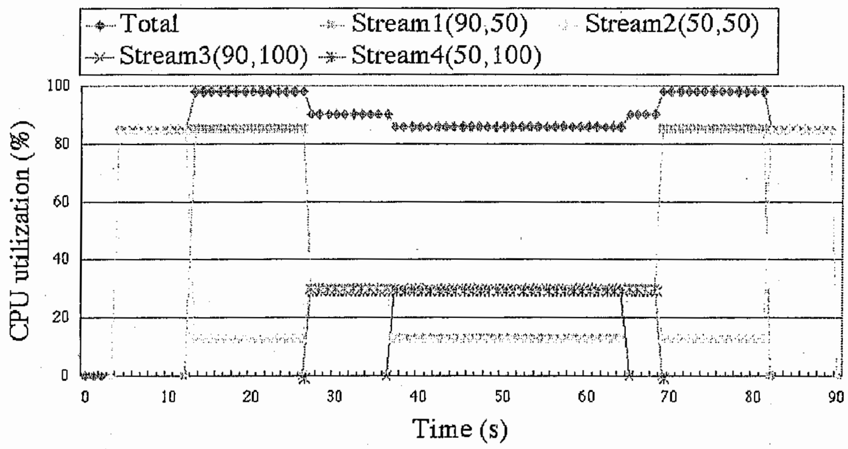


Figure 4.7: An experimental result of CPU resource allocation by QoS negotiation among AAs.

ing that video server A can not deal with two video streams with QoS #4 at 26s, we use the simple inter-terminal QoS negotiation to select the lower QoS between the receiver and the sender. Then, the QoS of Stream 2 will decrease to QoS #5 because its total priority is the lowest, while Streams 1 and 3 will keep QoS #4.

4.5.4 Local and Short-term QoS Adjustment

We present a scenario to show the SAs' QoS adjustment for sudden and transient resource variation. Although resource variation includes resource shortage and surplus, we describe the latter case here.

The scenario succeeds the global and long-term QoS adaptation and its initial condition was set to the same situation as at the point of 27s in Fig. 4.7, where Stream 1, 2, and 3 stably shared the same QoS #4. At 1s, 30% of CPU load was additionally given by the simulator as an unexpected disturbance. Then, every SA performed to QoS adjustment until a stable situation was recovered.

The SAs use the architecture illustrated in Fig. 4.5. The priority threshold parameter Th has the same dimension as the stream's total priority, that is the product of a stream priority and an application priority. In this experiment, Th is set to the minimum total priority of all streams initially, and updating of Th is by increasing a constant value 1000. Monitoring period is set to 100ms for each

SA. The simulator can increase or decrease the resource such as CPU utilization on purpose for experiments. In a resource surplus case, an SA adjusts QoS by degrading a QoS parameter according to the order that is given to the PA as the user's policy.

The result is shown in Fig. 4.8. Each SA can only perceive whether the CPU utilization is exceeded or not, although excessive CPU utilization rates over 100% (such as 120%) are explicitly depicted to visualize the simulator operations in Fig. 4.8. Stream 2 that has the lowest total priority of SAs starts firstly QoS adjustment. Then Stream 1 does QoS adjustment according to the threshold. Stream 2 degrades the size and the quantization scale in succession (at 1.5s), and the frame rate (at 2.5s) until Stream 1 starts to QoS adjustment. Stream 3 keeps its QoS.

From this experimental result, it is found that the SAs successfully deal with the abrupt resource change by autonomously adjusting its own QoS. Namely, while Stream 2 with the lowest total priority degraded its QoS most, Stream 3 with the highest total priority kept its QoS.

4.6 Conclusion

For communication-intensive applications using distributed multimedia, we have proposed an agent-based adaptive QoS management framework. The adaptability is targeted at various users' QoS requirements and resource fluctuations. The adaptive QoS management task is accomplished by direct and indirect agents' collaboration. The most remarkable and characteristic point is the mutually supplemental cooperation of AAs and SAs. Namely, in our approach, the AAs work in the flow establishment phase for the global and long-term QoS adaptation, while

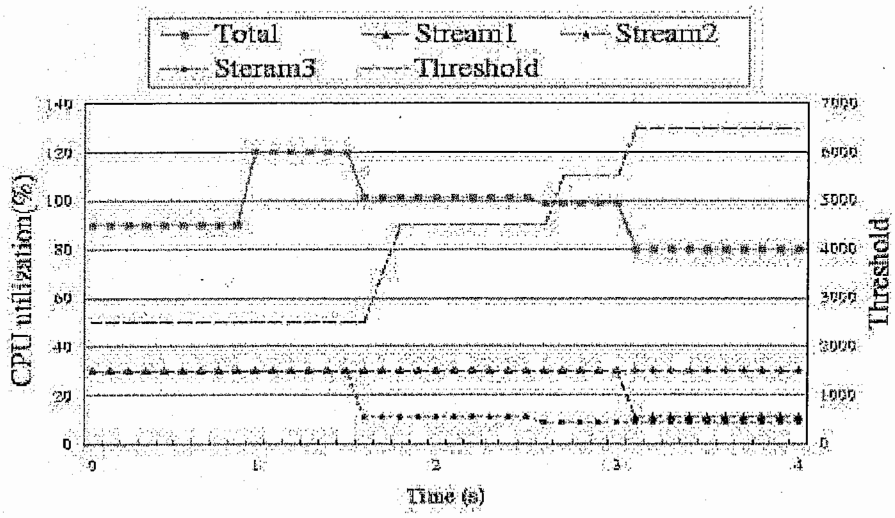


Figure 4.8: An experimental result of CPU resource allocation by QoS adjustment among SAs.

the SAs work in the QoS maintenance phase for the local and short-term QoS adjustment.

As an example of the proposed framework-based multimedia application, we developed a one-way video systems called MARM-Video1. The MARM-Video1 accepts abstract QoS requirements with utility from users, and manages media stream QoS to maximize the total user utility under resource constraints. We have evaluated the performance of MARM-Video1 through experiments of computer simulation, and it is verified that the proposed QoS negotiation and adjustment mechanisms work appropriately. We have implemented MARM-Video1 without SAs in a laboratory testbed. This prototype system works practically on actual computers connected by an ATM or Ethernet network. We are planning to extend the prototype system to implement SAs. The assumption that users are more sensitive to quality degradation than to quality improvement has to be verified by further studies.

Bibliography

[1] C. Aurrecoechea, A.T. Campbell and L. Hauw, “A survey of QoS architectures,” ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, vol.6, no.3, pp.138-151, May 1998.

[2] K. Nahrstedt and J.M. Smith, “Design, implementation, and experiences of the OMEGA end-point architecture,” IEEE J. Sel. Areas Commun., vol.14, no.7, pp.1263-1279, Sept. 1996.

[3] L. Delgrossi, C. Halstrick, D. Hehmann, R.G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, “Media scaling for audiovisual communication with the Heidelberg transport system,” Proc. ACM MultiMedia ’93, pp.99-104, Anaheim, June 1993.

[4] A. Campbell, G. Coulson, F. Garcia, D. Hutchison, and H. Leopold, “Integrated quality of service for multimedia communications,” Proc. IEEE

Infocom'93, pp.732-739, San Francisco, March 1993.

[5] C. Zhang and D. Lukose, eds., Multi-Agent Systems Methodologies and Applications (Lecture Notes in Artificial Intelligence 1286), Springer-Verlag, Berlin, 1997.

[6] G. Weiss, ed., Adaptation and Learning in Multi-Agent Systems (Lecture Notes in Artificial Intelligence 1042), Springer-Verlag, Berlin, 1996.

[7] T. Suganuma, T. Kinoshita, K. Sugawara, and N. Shiratori, "Flexible videoconference system based on ADIPS framework," Proc. 3rd Int. Conf. and Exhibition on the Practical Applications of Intelligent Agents and Multi-agents (PAAAM98), pp.83-98, London, March 1998.

[8] A. Puliafito, O. Tomarchio and H.de Meer, "An agent-based framework for QoS management," Proc. of WCSS'97, pp.392-396, Singapore, Sept. 1997.

[9] J. Allen, J. Hendler, and A. Tate, eds., Readings in Planning, Morgan Kaufmann Publishers, San Mateo, California, 1990.

[10] R.A. Brooks, "A robust layered control system for a mobile robot," IEEE J. Robotics and Automation, vol.2, no.1, pp.14-23, 1986.

[11] N. Ogino, M. Kosuga, T. Yamazaki, and J. Matsuda, "A model of adaptive QoS management platform based on cooperation of layered multi-agents," Proc. Globecom'99, pp.406-413, Rio de Janeiro, Brazil, Dec. 1999.

[12] T. Yamazaki and J. Matsuda, "On QoS mapping in adaptive QoS management for distributed multimedia applications," Proc. ITC-CSCC'99, vol.2, pp.1342-1345, July 1999.

[13] B. Hayes-Roth, "A blackboard architecture for control," Artif. Intell., vol.26, pp.251-321, 1985.

Chapter 5

Application of CCS to Realistic Environments

5.1 Introduction

In this chapter, application of the CCS approach to more realistic environments is discussed. Firstly, the proxy server approach to which the CCS approach belongs is reviewed again. Secondly, the CCS approach is applied to a realistic environment, and an error resiliency scheme is proposed by utilizing both channel and source coding techniques. Thirdly, a QoS management architecture combining the CCS approach and the MARM framework is shown. Finally, a CCS application to home networks is presented.

5.2 Related Work

Difficulties lie in adaptability to various individual user's QoS requirements and heterogeneous terminal and network performance [1]. A new paradigm to approach the problems is the proxy server approach. The approach includes the Video Gateway architecture [2], the cluster-based TACC (Transformation, Aggregation, Caching, and Customization) server architecture [3], and the service proxy approach [4]. In the Video Gateway architecture, the Video Gateway transcodes one video format into another video format, and performs bandwidth adaptation by rate-control. However, the Video Gateway did not refer to the receiver's resource, and it did not also implement any application-level or user-level QoS control mechanism. In the TACC server architecture, a cluster of workstations can serve tens of thousands Internet users datatype-specific distillation services, which is a kind of data filtering. However, handling of video streams was not referred in the TACC server architecture, but only handling of text data or still images was referred. The service proxy approach has been proposed to help mobile applications to be adapted to dynamically changing environments. In this approach, an application is partitioned into two pieces, one for a mobile computer and the other for a stationary computer, and the mobile application is developed by composing objects that contain small functionalities. The object composition can be changed to adapt to the communication environmental changes. But it seems difficult to reconfigure the object composition in real-time.

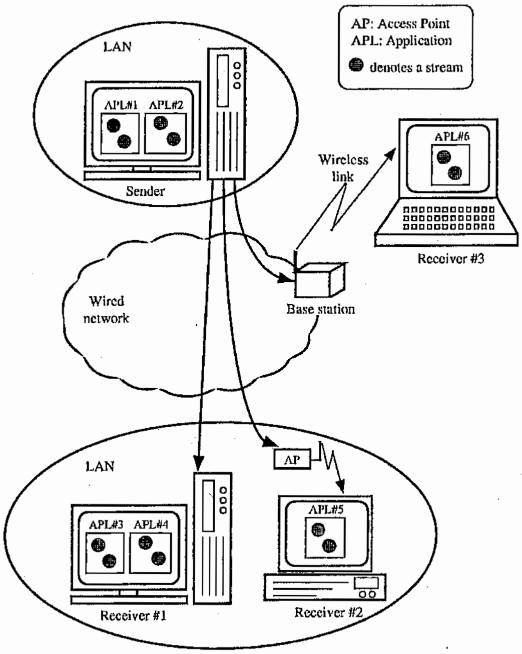


Figure 5.1: A realistic communication environment with heterogeneity.

5.3 CCS Approach in Heterogeneous Environments

Figure 5.1 shows a realistic communication environment with heterogeneity. The heterogeneities in terminal and network performance restrain a video transmission to multiple receivers. For example, in Fig. 5.1, when the sender delivers a stream to the multiple receivers #1-#3 whose QoS requirements are different, the stream QoS has to be set as the lowest QoS requirement. Especially the QoS gap increases when a handheld terminal connected by wireless link such as the receiver #3 exists in the multicast group.

Figure 5.2 shows CCS application into the heterogeneous environment of Fig. 5.1. CCS #1 is located in the intra-network of the receivers #1 and #2, and CCS #2 is located between the wired network and the wireless link to intermediate between the sender and the receiver #3. CCS #1 performs the aggregation of QoS requirements from the receivers #1 and #2, and negotiates with the sender. Also, receiving a media stream from the sender, CCS #1 transforms the media QoS according to individual receiver's QoS requirement, and distributes the transformed media streams to the receivers. CCS #2 also performs QoS transformation according to the receiver's QoS requirement. In addition, CCS #2 enhances error resiliency of the media stream. This error resiliency enhancement works effectively when the receiver is connected by wireless link. Because wireless links can

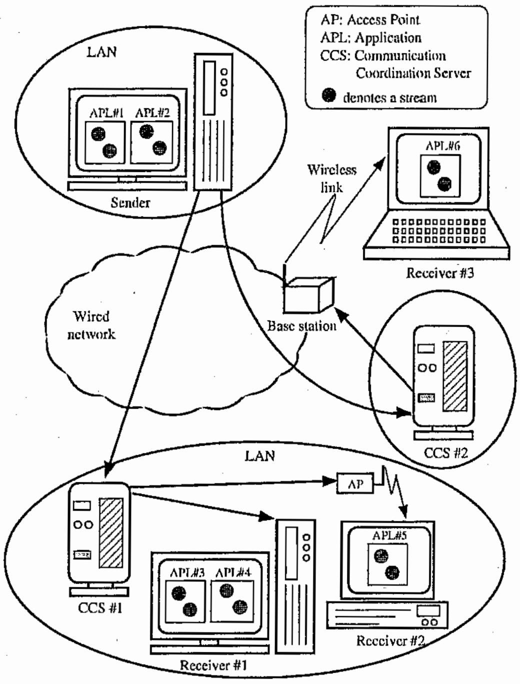


Figure 5.2: CCS application into the heterogeneous environment.

suffer from much higher BER up to 10^{-2} or more than wired networks [5].

5.3.1 Error Resiliency Enhancement Method with QoS Consideration

An error resiliency enhancement method utilizing channel and source coding techniques is proposed with relation to QoS management, and Fig. 5.3 shows its module structure on the CCS.

The error resiliency enhancement method works as follows; the CCS receives the QoS requirement from the client as well as collects the network and client terminal QoS information by resource monitoring or message passing from the client. The CCS enhances error resiliency of the video stream by utilizing channel and source coding techniques selectively. The channel coding techniques include FEC (Forward Error Correction) and ARQ (Automatic Repeat reQuest), while the source coding techniques include the RVLC (reversible variable length codes)

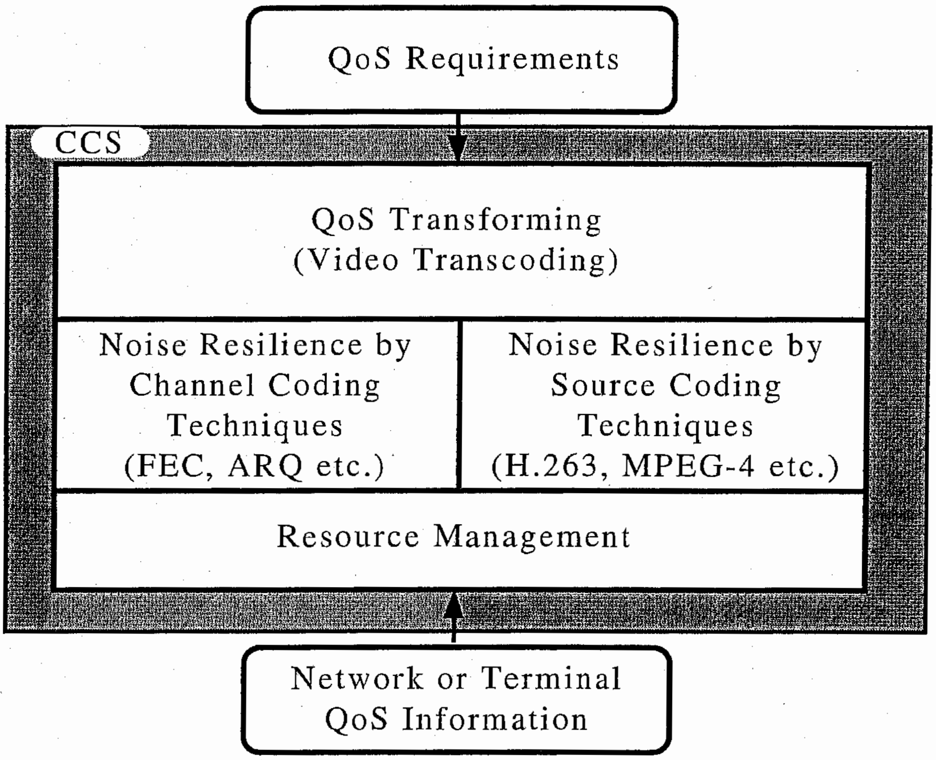


Figure 5.3: Module structure on the CCS.

[6]. Since the error resiliency enhancement increases the redundancies of codes, available resource QoS of the network or the terminal also changes (generally decreases). The CCS then transforms the video QoS according to the resource QoS and the QoS requirement from the client by the QoS mapping and transcoding techniques.

We show an example of the error resiliency enhancement and QoS adjustment. Here, the case of 384.0 kb/s of wireless link between a receiver and a CCS (which is supposed to be realized in IMT-2000) and M-JPEG for the video compression technique is assumed.

When the BER is low, the FEC of coding rate $2/3$ yields a data payload of 256.0 kb/s. Figure 5.4 shows the relationship between the M-JPEG parameters (video QoS), that is frame size, frame rate, and a quantization scale, and the bandwidth value (network QoS) obtained by a QoS mapping technique. Figure 5.4 tells that a frame size of 320×240, a frame rate of 5, and a quantization scale of 90 can be substantiated as transcoding parameters for the payload (the point A in Fig. 5.4).

When the BER becomes higher, the coding rate of FEC is decreased to $1/2$ and the RVLC, which also decreases the coding rate by 10%, is used by the CCS. Accordingly the data payload decreases to 174.5 kb/s. As the result, the transcoding parameters have to be adjusted, and Fig. 5.4 tells that a frame size of 320 × 240, a frame rate of 5, and a quantization scale of 80 (the point B in Fig. 5.4) or a frame size of 320 × 240, a frame rate of 3, and a quantization scale of 90 (the point C in Fig. 5.4) can be substantiated. Which point is chosen between the points B and C depends the requirement from the client.

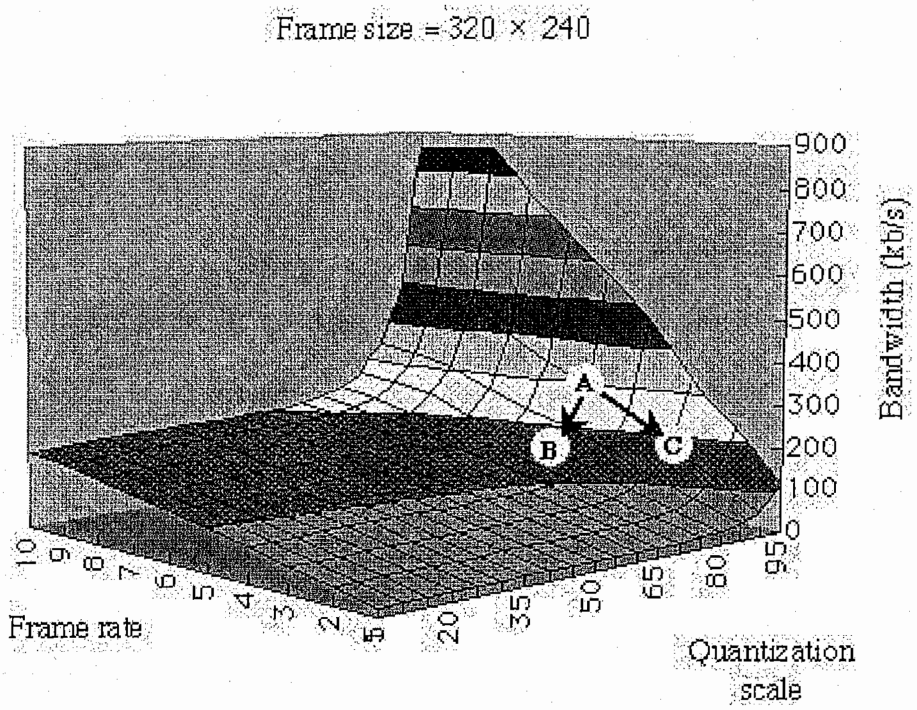


Figure 5.4: The relationship between the M-JPEG parameters and the bandwidth value.

5.4 Combination of CCS and MARM

While the CCS approach is suitable for a configuration of transmission from a server to multiple receivers, the MARM framework is suitable for a configuration of transmission from multiple servers to a receiver, Combining the CCS approach and the MARM framework realizes a QoS management for a configuration of transmission from multiple servers to multiple receivers.

In the following, a scenario shows that the CCS is able to contribute to multicasting in a heterogeneous environment in conjunction with the MARM framework.

5.4.1 Testbed Configuration

To evaluate the MARM framework and the CCS approach performance, we have built a laboratory testbed. The testbed network configuration is illustrated in Fig. 5.5, which comprises two video senders, two receivers, and a CCS. All machines except for the receiver #2 are desktop-type PC-AT machines (Windows NT), and the receiver #2 is a portable PC-AT machine (Windows NT). All machines are connected over an ATM network. In the following, we focus on CPU resource management because limitation for the CPU resource is tighter than the one for the network resource in our testbed.

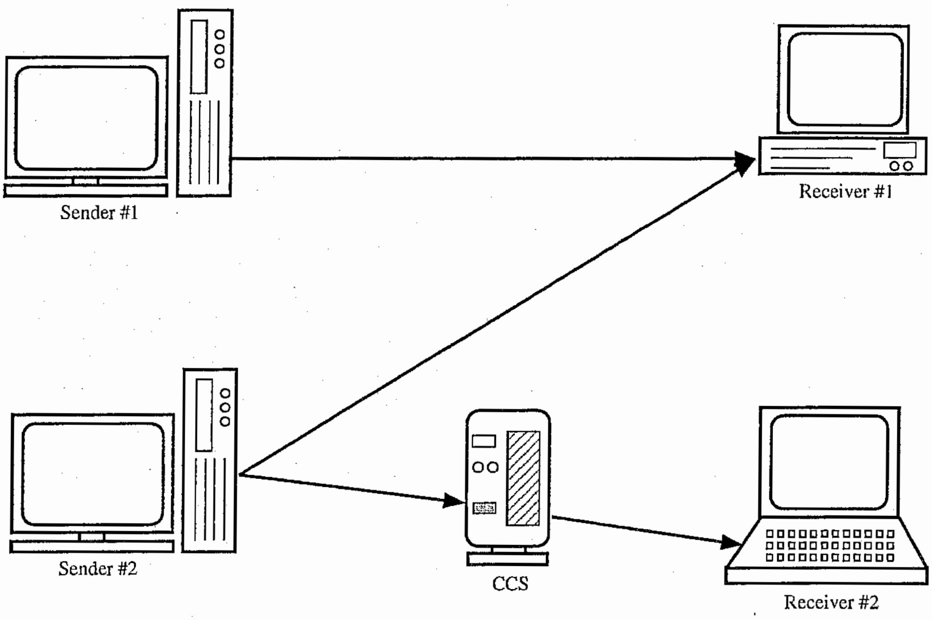


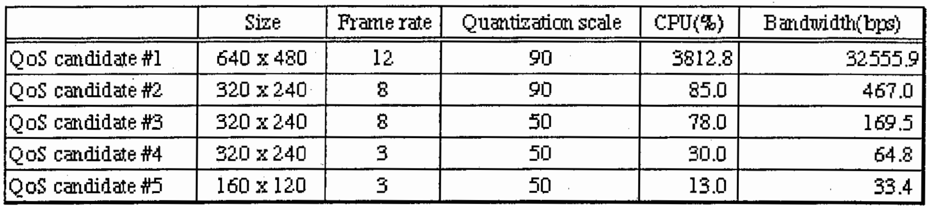
Figure 5.5: The testbed network configuration.

5.4.2 Application Agent Negotiations

The first experiment shows the QoS management for multiple M-JPEG video streams by AA negotiations. In the experiment, the receiver #1 receives multiple video streams from two senders, the sender #1 and #2. Each stream has five QoS candidates in Table 5.1 specified by the user. In Table 5.1, the CPU utilization rate and the bandwidth (throughput) for each specified QoS are for the receiver #1, and these values are calculated by the AA using the spline QoS mapping scheme [7].

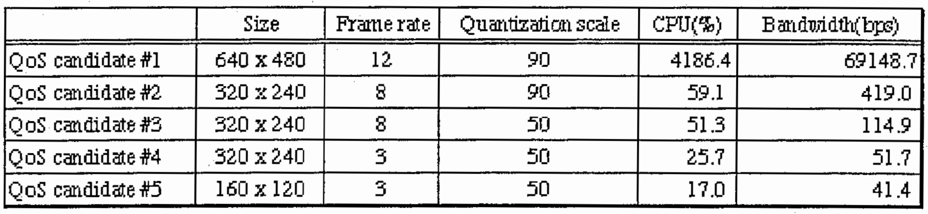
Although a non-realistic value such as 3812.8% of the CPU utilization rate can be calculated because of the spline extrapolation, QoS candidates with non-realistic values should be excluded as impractical. Also Table 5.2 shows the CPU utilization rate and the bandwidth (throughput) for each specified QoS for the receiver #2. All of QoS mapping values in Tables 5.1 and 5.2 are for the M-JPEG coding.

Table 5.1: QoS candidates used in the experiment. CPU and Bandwidth are calculated for the corresponding set of Frame size, Frame Rate, and Quantization scale for the receiver #1.



	Size	Frame rate	Quantization scale	CPU(%)	Bandwidth(bps)
QoS candidate #1	640 x 480	12	90	3812.8	32555.9
QoS candidate #2	320 x 240	8	90	85.0	467.0
QoS candidate #3	320 x 240	8	50	78.0	169.5
QoS candidate #4	320 x 240	3	50	30.0	64.8
QoS candidate #5	160 x 120	3	50	13.0	33.4

Table 5.2: QoS candidates used in the experiment. CPU and Bandwidth are calculated for the corresponding set of Frame size, Frame Rate, and Quantization scale for the receiver #2.



	Size	Frame rate	Quantization scale	CPU(%)	Bandwidth(bps)
QoS candidate #1	640 x 480	12	90	4186.4	69148.7
QoS candidate #2	320 x 240	8	90	59.1	419.0
QoS candidate #3	320 x 240	8	50	51.3	114.9
QoS candidate #4	320 x 240	3	50	25.7	51.7
QoS candidate #5	160 x 120	3	50	17.0	41.4

QoS candidate #1 is very high quality QoS, and Table 5.1 shows that it is impossible to be realized in the present testbed system. Since the number of streams changes according to time, the video QoS has to be controlled by the AA negotiations under the limitation of the receiver's CPU resource.

The receiver #1 operates two applications, APL1 and APL2, and APL1 receives video streams from the server #1, while APL2 receives video streams from the server #2. The application priority values for APL1 and APL2 are 50 and 100, respectively. Figure 5.5 shows how the CPU resource on the receiver is allocated to the streams as the number of streams changes. Each time a stream increases or decreases, the QoS negotiation is carried out to determine a new resource allocation according to the application and stream priorities. In Fig. 5.5, APL1 deals with Stream 1 and Stream 2 and APL2 deals Stream 3 and Stream 4. The stream priority values for Stream 1, 2, 3, and 4 are 90, 50, 90, and 50, respectively. From this experimental result, it is found that QoS allocation has been fairly done according to the priority values. For example, two video streams with different priority values were transmitted from each sender, and AAs negotiated QoS for these four streams. Eventually, QoS #5 was allocated for Stream 1 and 2 and QoS #4 was allocated for Stream 3 and 4 by taking the application and stream priority values into consideration.

5.4.3 QoS Transformation by CCS

The second experiment shows the QoS transformation by the CCS. In the experiment, the sender #2 multicasts an MPEG-1 video stream (352 × 240 pixels and 30 frames/s) to the receiver #1 and the receiver #2. It is assumed that there are enough terminal and network resources for the receiver #1 to process the MPEG-1 video stream. However, since the receiver #2 is equipped with only an M-JPEG decoder because of its lightweight processing, it cannot process directly the MPEG-1 video stream. Hence the CCS intermediates between the sender and receiver, transcodes the video format, and transforms the video QoS. In our testbed, the CCS is equipped with an MPEG decoding board and an M-JPEG encoding board for real-time implementation of the transcoding module. Consequently the CCS transcodes the MPEG-1 stream into an M-JPEG video stream (320 × 240 pixels, 8frames/s, quantization scale 90), and sends the M-JPEG stream to the receiver #2. The QoS values can be determined by the QoS mapping results in Table 5.2. The receiver #2 can receive the M-JPEG stream with desirable QoS without distubing the MPEG-1 stream receiving of the receiver #1.

When the MPEG-1 stream QoS cannot be kept because of unexpected system disturbance such as the network congestion or the user's QoS request changes, the CCS recalculate a viable QoS parameters for the new situation. For example, when 60% of additional CPU utilization is loaded to the receiver #2, the QoS candidate #2 cannot be kept yet. Then the QoS candidate #3 is selected by the CCS and a new M-JPEG stream with the new QoS parameters will be sent to the receiver #2 without changing the QoS parameters for MPEG-1 stream for the receiver #1.

5.5 CCS Application to the Networked Home

As a typical application example of CCS and the proposed QoS transformation mechanism, we now discuss networks for the home. Thanks to the advent of the IEEE 1394, IEEE 802.11, and Bluetooth standards, various digital electronic devices including TV and stereo systems and PCs are expected to be connected in the ordinary home in the future. Also, the existing phone lines or power lines may become the infrastructure of the home network. A home equipped with a home network that connects the electronic devices and PCs is called the networked home [8]. In addition, the networked home will be connected to the Internet, and a so-called home server will provide the access interface for the Internet.

The networked home will be a representative example of a heterogeneous communication environment, since the media processing performances of electronic devices and PCs will vary individually. Moreover, the network performance varies according to the technology used, that is, some devices connected by a rather fast network such as IEEE 1394, some are connected by a wireless link like Bluetooth, and others use a slow power line infrastructure. In such a heterogeneous networked home environment, it is desirable for the home server to provide the QoS transformation mechanism.

Figure 5.6 is an example of CCS application to a networked home, where the CCS plays the role of the home server. The home network is connected to the Internet by a Cable TV network that has the throughput of about 30–40 Mbps for the downlink. In the networked home, a desktop PC is connected by Ethernet

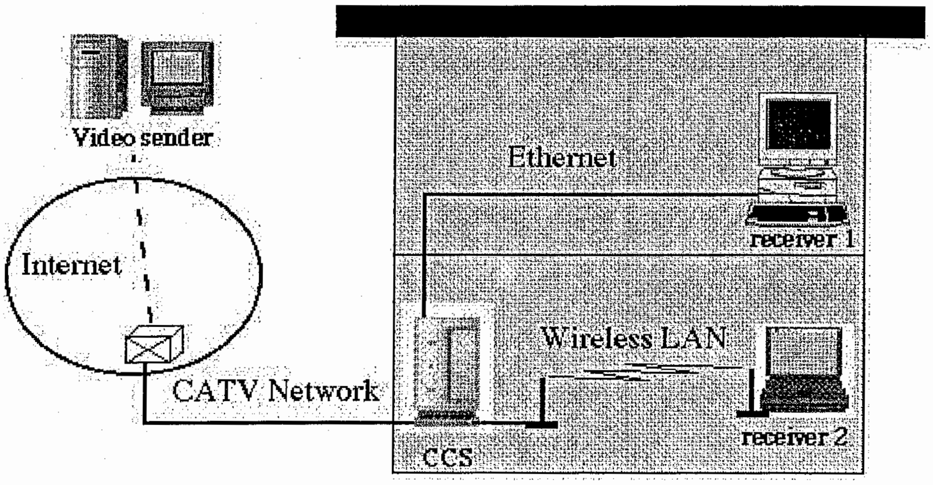


Figure 5.6: An example of CCS application to a networked home.

performing at about 4–5 Mbps, and a portable PC is connected by a wireless LAN (the IEEE 802.11) performing at about 1 – 2 Mbps. The laboratory testbed we developed aims at this example of the networked home.

5.6 Conclusion

In this chapter, some aspects of application of the CCS approach to more realistic environments were described. The discussion included an error resiliency scheme, combination of the CCS approach and the MARM framework, and CCS application to the networked home. Accordingly the CCS is useful to mitigate the complex QoS negotiation in the MARM framework, and facilitates multicasting of media streams in heterogeneous environments. One of the most promising application domains of the CCS is a home serve in the networked home. Also, the CCS can be applied to as the edge server in CDN (Content Destribution or Delivery Network) [9].

Bibliography

[1] R. LaMaire, M. Naghshineh, and S. Palazzo, "Adapting to network and client variability," IEEE Personal Communications, vol.5, no.4, pp.8-9, Aug. 1998.

[2] E. Amir, S. McCanne, and H. Zhang, "An application level video gateway," Proc. ACM Multimedia '95, pp.255-265, San Francisco, CA, Nov. 1995.

[3] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer, "Adapting to network and client variation using infrastructural proxies: lessons and perspectives," IEEE Personal Communications, vol.5, no.4, pp.10-19, 1998.

[4] T. Nakajima and A. Hokimoto, "Adaptive continuous media applications in mobile computing environments," Proc. IEEE International Conference on Multimedia Computing and Systems (ICMCS'97), pp.152-160, June 1997.

[5] T. H. Meng, "Low-power wireless video systems," IEEE Commun. Mag., pp.130-136, June 1998.

[6] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes", IEEE Trans. Commun., vol.43, no.2/3/4, pp.158-162, Feb./Mar./Apr. 1995.

[7] T. Yamazaki and J. Matsuda, "On QoS mapping in adaptive QoS management for distributed multimedia applications," Proc. ITC-CSCC'99, vol.2, pp.1342-1345, July 1999.

[8] A. Dutta-roy, "Networks for homes," IEEE Spectrum, vol.36, no.12, pp.26-33, Dec. 1999.

[9] T. Yamazaki, K. Hasuike, K. Mase, and M. Sengoku, "A study of multimedia communication coordination with content processing," IEICE Technical Report, CQ2001-99, Feb. 2002 (in Japanese).

Chapter 6

Multimedia Communication Coordination

6.1 Introduction

Controlling QoS in heterogeneous communication environments needs a consideration of policies that rule QoS agreement among the end-users who use a distributed multimedia service. Although there are several studies about management of the QoS policies [1], they were targeting QoS management only at the network layer (e.g. [2]). Since concept of QoS need to be extended up to the user layer because quality should be ultimately judged by the end-users, the QoS policy management should be extended into a multi-layer architecture. Fukuda et al. [3] proposed a method to decide required bandwidths, which is one of network QoS parameters, in consideration of the relationship of application-layer QoS parameters and user's preference on video quality, where the user's preference was evaluated by subjective tests. There was a QoS consideration through multi-layers for a video transmission service, but the video transmission was limited between a server and a client.

Assuming a best-effort network with no reservation mechanism as the infrastructure, this chapter deals with a multimedia service based on a layered QoS model among multi-users, where the QoS policy agreement among the multi-layers is considered. In the user layer, some agreement among the users is brought out considering outputs from the lower layers. The agreement goes down to the user and application layers below. In the application layer, the application-layer parameters are translated into the resource requirements. In the system layer, if there is a resource conflict, it is solved according to a QoS policy and the solution is fed back to the upper layers up to the meta-user layer. This multi-layered multimedia service mechanism is called multimedia communication coordination. As an application of the multimedia communication coordination, a chat system with video transmission is developed. In this application, similarity of respective users' interests is a policy in the meta-user layer, and resource constraints are the QoS policies to solve the QoS conflicts among video streams in the system layer.

The concept that supports this line of research is realization of adaptive multimedia communication coordination, that is to provide networked multimedia service adaptively according to variety of communication systems and fluctuation of communication systems to satisfy user's requirements. The multimedia communication coordination consists of system-oriented and user-oriented communication coordinations shown in Fig. 6.1. Namely, collaboration of the system and the user is expected to creat a new paradigm of multimedia service.

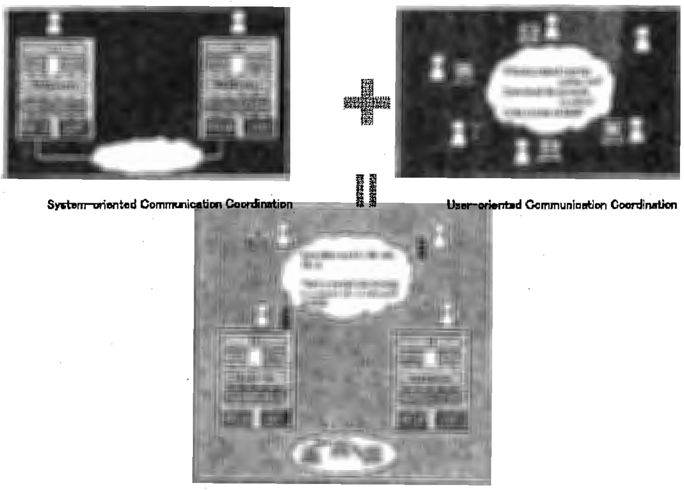


Figure 6.1: Communication coordination by system and user collaboration.

6.2 Model

The video chat system that we are developing is based on the generic Layered Adaptive QoS (LAQoS) model that we proposed. The LAQoS model is composed of the following elements (Fig. 6.2).

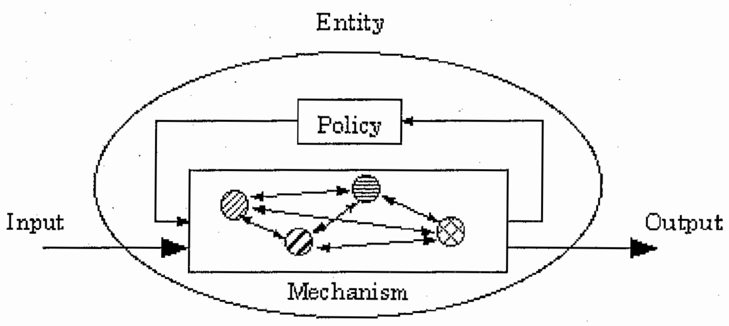


Figure 6.2: Elements in the LAQoS model.

- Entity

  An Entity is a unit which performs a layer-specific function. Entity consists of a Policy and a Mechanism, receives Inputs, and brings out an Output.

- Mechanism

  A Mechanism subsumes Entities of the lower layer and is created by an interaction of them.

- Policy

A Policy is a evaluation function for the interaction of the subsumed Entities.

- Input

  An Input consists of Outputs from other Entities in the same layer and constraints from the upper layer.

- Output

  When a Mechanism is driven by an Input, an Output from the Entity is sent out as a collection of outputs of Entities subsumed in the Mechanism. The output of each Entity subsumed is, for example, a constrained resource value in the system layer.

Figure 6.3 shows the layered model of the video chat system that we are developing. This model is a specification of the generic LAQoS model. In the meta-user layer, the participating users use a chat module to communicate. A chat server observes the conversation and finds out the users' relationship as similarity of interests by a distance measurement of key word vectors for users' utterance [4]. The distance measurement plays a role of policy in this layer. In the user layer, it is assumed that a user uses the specified video chat application solely and no conflict among applications occurs, so that the Entity is identified with that in the application layer. In the application layer, an application subsumes multiple resources to be consumed. Although a policy is needed to regulate the multiple resources when there is any interaction among them, no policy is implemented in the actual video chat system at present because there is little need of controlling between the CPU and network resources. In the system layer, there are two kinds of Entities for the CPU and network resources. The mechanism of CPU Entity is simple. It is regulated to stay within the available upper limit because it is assumed that an application occupies the CPU resource in an end-system. On the other hand, the mechanism of network resource Entity is more complicated because the network resource has to be shared among plural applications in the best-effort network that we assumed. To this end, a priority parameter is introduced to differentiate the users (end-systems) priority. The policy in the network resource Entity performs bandwidth allocation based on the priority parameter.

6.3 Implementation

Based on the LAQoS model, we are developing a chat system with video transmission on a laboratory network. The overall basic design of the chat system is shown in Fig. 6.4. Figure 6.5 also presents a brief logical architecture of the sys-

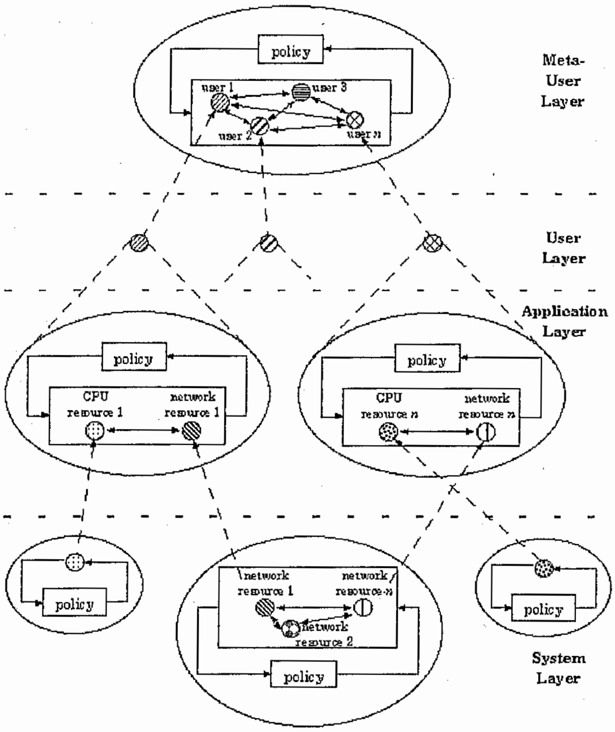


Figure 6.3: Layered model of the video chat system that we developed based on the LAQoS model.

tem including n client end-systems, a video mediation server, and a chat server. Chat text data are distributed via the chat server, which analyzes the data and finds relationships among the users based on the policy in the meta-user layer. Video data of respective user faces are intermediated by the video mediation server to be transmitted to the appropriate companions with adjusted QoS. Before this, network resource allocation is carried out based on the policy in the system layer by the QoS adjusting module in this server.

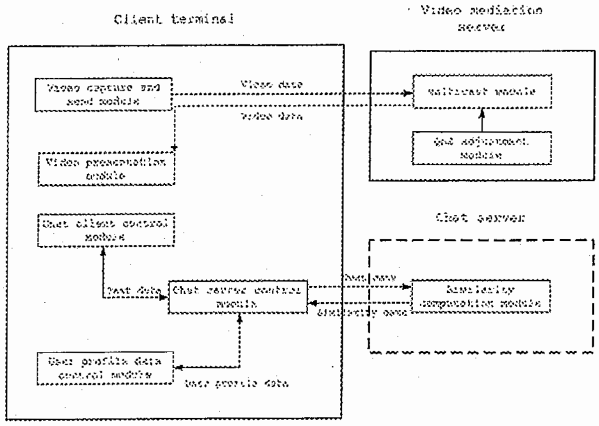


Figure 6.4: Overall basic design of the video chat system.

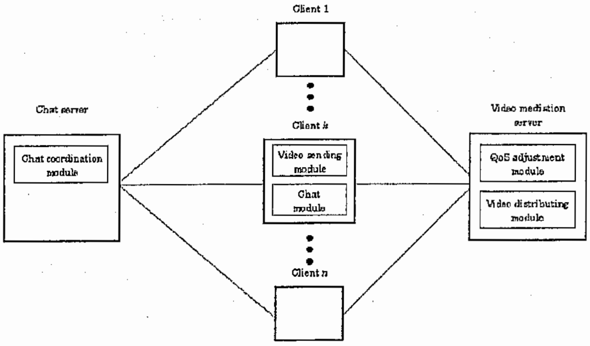


Figure 6.5: Logical architecture of the video chat system.

Figure 6.6 presents an architecture of chat communications. Chat communications consist of the chat client control module and the chat server control module. The chat client control module sends the text data input by a user to the chat server and presents the text data from other users sent by the chat server. The chat server control module manages connection establishment and release with the chat clients and distributes the text data from a user to other users. Also, the chat server control module has an interface with the similarity computation module.

Figure 6.7 presents an architecture of video communications. The video capture and send module captures a video stream from a video capture card equipped at a client terminal and sends it to the video server. The frame size, frame rare, and etc. are controlled by the QoS adjustment module. and the chat server control module.

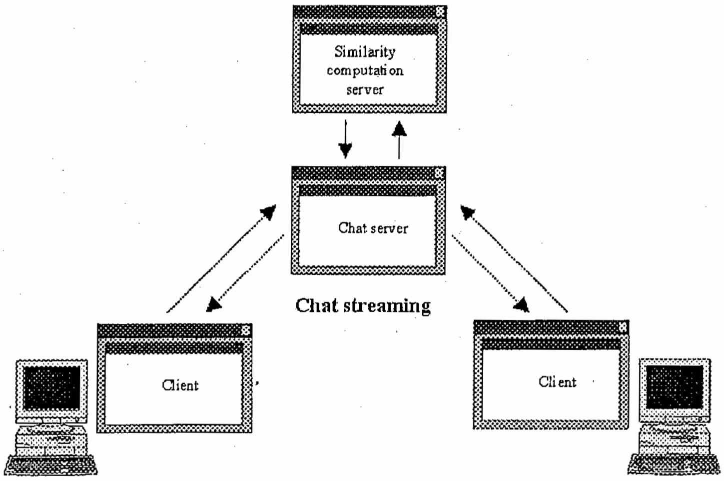


Figure 6.6: Architecture of chat communications.

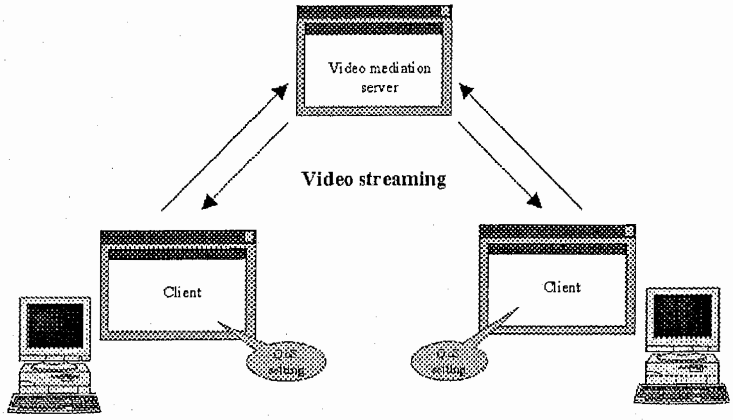


Figure 6.7: Architecture of video communications.

6.4 Conclusion

A QoS mechanism which meets a QoS policy agreement from the most-top meta-user layer to the lowest system resource layer has been proposed. Based on it, we are developing a video chat system that is expected to provide more user-centric multimedia services. This chapter presented the overall design of the video chat system.

Bibliography

[1] L. Lewis, "Implementing policy in enterprise networks," IEEE Commun. Mag., vol.34, no.1, pp.50-55, Jan. 1996.

[2] Z. Dziong and L.G. Mason, "Fair-efficient call admission control policies for broadband networks - a game theoretic framework," IEEE/ACM Trans. Networking, vol.4, no.1, pp.123-136, Feb. 1996.

[3] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "QoS mapping between user's preference and bandwidth control for video transport," Proc. Fifth IFIP International Workshop on Quality of Service (IWQoS 97), pp.291-302, May 1997.

[4] N. Kurabayashi, T. Yamazaki, K. Hasuike, and T. Yuasa, "ZINGER: conversation support based on interest similarity," Proc. ICME2001 (2001

IEEE International Conference on Multimedia and Expo), pp.557-560, Aug. 2001.

Chapter 7

Conclusions

In this report, controlling quality of image media transmitted in heterogeneous communication environments has been studied. Since the degree of quality (or QoS) should be judged by the end-user, deployment of QoS control from the end-user's viewpoint has been discussed, and a layered QoS model from the User QoS level to the Resource QoS level was proposed. On the assumption that the infrastructure networks are best-effort, a networking architecture with a proxy server and a QoS adjustment scheme were proposed for video transmission applications. An adaptive QoS management framework based on the multi-agent system was proposed, too. Applications of the proxy server and multi-agent approaches to realistic communication environments were discussed.

In Chapter2, a generic multi-level QoS model was presented for distributed multimedia applications, and studies of QoS mapping from one level to other level were reviewed. Then a realistic QoS mapping mechanism composed of QoS mapping methods was presented. One method maps the highest user level QoS into lower level QoS by user-specific profile data, and the other method performs mapping among lower level QoS parameters than the user level by spline functions. The second spline QoS mapping method, which is a novel idea, is adaptive to both of user requirements and changeable environment. The mapping results by the natural spline and the B-spline QoS mapping functions were compared using the actual measured data of video QoS. It was found that the B-spline QoS mapping functions showed better results once the knots for spline function were selected appropriately. The further research issues include the way of determination for appropriate positioning of the knots in B-spline functions, and an automatic acquisition of the user-specific profile data. Also, relevance of QoS between the application and user levels was discussed by showing valuable subjective test results for QoS evaluation.

In Chapter3, a QoS management architecture was presented for distributed multimedia applications in heterogeneous communication environments. In the proposed architecture, a proxy server called CCS intermediates between a video sender and a receiver or a group of receivers and manages the QoS adjustment. The CCS monitors the currently available resources and receives the QoS and resource information from the receiver(s). Based on these information, the CCS calculates a feasible QoS for each receiver to utilize the system resources efficiently. Then the CCS carries out the transcoding to transform the video QoS to satisfy the receivers' requirements.

Prototype systems of the CCS were implemented in a laboratory testbed. In the prototype systems, the transcoding mechanisms between MPEG and M-JPEG codings were implemented in hardware or software. With the prototype system, it is verified that the CCS can resolve the network and terminal heterogeneities between the sender and receiver sides by the transcoding and the QoS adjustment mechanism. As further research, scalability of the CCS architecture, that is how many receivers the CCS can deal with in which conditions, must be studied.

In Chapter4, for communication-intensive applications using distributed multimedia, we proposed an agent-based adaptive QoS management framework. The adaptability is targeted at various users' QoS requirements and resource fluctuations. The adaptive QoS management task was accomplished by direct and indirect agents' collaboration. The most remarkable and characteristic point is the mutually supplemental cooperation of AAs and SAs. Namely, in our approach, the AAs work in the flow establishment phase for the global and long-term QoS adaptation, while the SAs work in the QoS maintenance phase for the local and short-term QoS adjustment.

As an example of the proposed framework-based multimedia application, we developed a one-way video systems called MARM-Video1. The MARM-Video1 accepts abstract QoS requirements with utility from users, and manages media stream QoS to maximize the total user utility under resource constraints. We have evaluated the performance of MARM-Video1 through experiments of computer simulation, and it is verified that the proposed QoS negotiation and adjustment mechanisms work appropriately. We have implemented MARM-Video1 without SAs in a laboratory testbed. This prototype system works practically on actual computers connected by an ATM or Ethernet network. We are planning to extend the prototype system to implement SAs. An extension to multiple media processing, for example, video, audio, and their relationship, is one of further research issues.

In Chapter5, some aspects of application of the CCS approach and the MARM framework to more realistic environments were described. The discussion included an error resiliency scheme, combination of the CCS approach and the MARM framework, and CCS application to the networked home. Accordingly the

CCS is useful to mitigate the complex QoS negotiation in the MARM framework, and facilitates multicasting of media streams in heterogeneous environments. One of the most promising application domains of the CCS is a home serve in the networked home. The future research and development should deal with implementations of the proposed approaches in a real heterogeneous communication environment.

In Chapter 6, a conceptual idea of multimedia communication coordination, which consists of system-oriented and user-oriented coordinations, was introduced. A layered QoS model was proposed to meet a QoS policy agreement. Then a chat system with video transmission was planned based on the layered QoS model. The implemetation is not completed and proceeding yet.

In addition to the above-mentioned further studies for respective proposed approaches, there are three generic future research issues. The first issue is how to decide QoS policies to keep QoS fairness, that is to arrange a QoS allocation among the users concerned not to make any partiality. This issue is related with costs of QoS in multimedia communications. The second issue is how to construct or update the user-specific profile database introduced in Chapter 2. This issue is quite difficult because it requires the analysis of human thinking and feeling patterns. Learning mechanisms such as an inforcement learning may be useful to solve the issue. The final issue is how to apply the proposed approaches to actual situations. Although some aspects of application of the proposed approaches to realistic environments were discussed, there are several points to be considered: scalablity of the terminal number that the CCS can take care of, applicability of the CCS to various standards such as the IEEE 1394 and Bluetooth, relaxation of the best-effort assumption to cooperate with other QoS mechanisms, and so on.

Year after year, bandwidths of networks broaden and performance of end-system advances. Therefore, existent media processing technologies will progress, and new multimedia services will appear. Hereupon, it will be important to develop the technologies and services from the viewpoint of the end-user, since the quality of multimedia services should be judged by the end-user. Moreover, personalization of service, that is service provision according to each user's requirements, will open the door of a new "multimedia network society" era. It is greatly desirable that this report will contribute to the evolution of the new era as well as the improvement of quality fo people's life.

Publications by Author

The listed following publications are published by the author as the first author based on the research work at ATR ACR.

Full papers:

(1) T. Yamazaki and J. Matsuda, "Adaptive QoS management for multimedia applications in heterogeneous environments: a case study with video QoS mediation," IEICE Trans. Commun., vol.E82-B, no.11, pp.1801-1807, Nov. 1999.

(2) T. Yamazaki, M. Kosuga, N. Ogino, and J. Matsuda, "MARM: an agent-based adaptive QoS management framework," IEICE Trans. Commun., vol.E84-B, no.1, pp.63-70, Jan. 2001.

(3) T. Yamazaki, "Data-driven multi-dimensional histogram method for image segmentation," Journal of the Institute of Image Electronics Engineers of Japan, vol.30, no.6, pp.716-722, Nov. 2001.

(4) T. Yamazaki, S. Fukunaga, and N. Sato, "Video transmission with QoS adjustment for multiple receivers using transcoding," Trans. IEICE, vol.J85-B, no.1, pp.50-59, Jan. 2002 (in Japanese).

Letters:

(1) T. Yamazaki, "Spline-based QoS mapping mechanisms for hierarchical

multilevel QoS models," IEICE Trans. Fundamentals, vol.E85-A, no.6, June 2002 (accepted).

International conferences:

(1) T. Yamazaki, "Introduction of EM algorithm into color image segmentation," Proc. 2nd IEEE International Conference on Intelligent Processing Systems (ICIPS'98) , Aug. 1998.

(2) T. Yamazaki, "A Robust Clustering Technique for Multi-spectral Satellite images," Proc. 1998 International Symposium on Noise Reduction for Imaging and Communication Systems (ISNIC-98), G2, pp.233-238, Nov. 1998.

(3) T. Yamazaki and J. Matsuda, "On QoS mapping in adaptive QoS management for distributed multimedia applications," Proc. ITC-CSCC'99, vol.2, pp.1342-1345, July, 1999.

(4) T. Yamazaki, M. Kosuga, N. Ogino, and J. Matsuda, "Adaptive QoS management for multimedia applications in heterogeneous environments," Proc. IMMCN2000 (JCIS2000), vol.2, pp.564-567, Atlantic City, NJ, Feb. 2000.

(5) T. Yamazaki and O. Nishida, "Data-Driven color image segmentation based on a multi-dimensional histogram," Proc. ISPACS 2000, A7-3-3, vol.1, pp.399-404, Nov. 2000.

(6) T. Yamazaki and K. Hasuike, "QoS transformation by a proxy server and

its application to the networked home," Proc. ICACT2001 (the 3rd International Conference on Advanced Communication Technology), pp.457-461, Feb. 2001.

(7) T. Yamazaki and M. Sakamoto, "QoS mapping data acquisition mechanisms for distributed multimedia applications," Proc. ITC-CSCC'01, vol.2, pp.1168-1171, July 2001.

(8) T. Yamazaki, "Subjective video quality assessment for adaptive quality-of-service control," Proc. ICME2001 (2001 IEEE International Conference on Multimedia and Expo), pp.517-520, Aug. 2001.

(9) T. Yamazaki, "QoS adjustment scheme for video transmission to a heterogeneous receivers group," Proc. IMMCN2002 (JCIS2002), pp.996-1000, Durham, NC, March 2002.

Domestic conferences and meetings:

(1) T. Yamazaki and J. Matsuda, "QoS mapping using spline functions," Proc. 1998 IEICE General. Conf., SB-11-2, pp.728-729, March 1998 (in Japanese).

(2) T. Yamazaki and J. Matsuda, "A communication coordination server for adaptive multimedia applications," Proc. 1998 Comm. Soc. Conf. of IEICE, B-11-10, p.512, Sept. 1998 (in Japanese).

(3) T. Yamazaki and J. Matsuda, "Dynamic QoS mapping by spline neural networks," Proc. 1999 IEICE General. Conf., B-11-15, p.549, March 1999 (in

Japanese).

(4) T. Yamazaki and J. Matsuda, “A survey of adaptive technologies for mobile computing,” Proc. 1999 Comm. Soc. Conf. of IEICE, SB-11-6, pp.427-428, Sept. 1999 (in Japanese).

(5) T. Yamazaki and J. Matsuda, “Multimedia multicasting with QoS management,” Proc. 2000 IEICE General. Conf., B-11-8, p.579, March 2000 (in Japanese).

(6) T. Yamazaki and K. Hasuike, “Communication service coordination in home networking,” IEICE Technical Report, CQ2000-21, July 2000 (in Japanese).

(7) T. Yamazaki, “A study of image segmentation based on a muti-dimensional histgram,” IEICE Technical Report, IE2000-43, Sept. 2000 (in Japanese).

(8) T. Yamazaki, “An image segmentation method based on a muti-dimensional histgram,” IPSJ Technical Report, 3V-10, pp.2-227-228, Oct. 2000 (in Japanese).

(9) T. Yamazaki, N. Sato, and S. Fukunaga, “Subjective video quality measurement experiments for adaptive QoS control,” IEICE Technical Report, CQ2000-83, Feb. 2001 (in Japanese).

(10) T. Yamazaki, S. Fukunaga, N. Sato, and H. Shirahata, “Implementation of video double-casting with QoS adjustment,” IPSJ Technical Report, demo-4, pp.3-511-512, March 2001 (in Japanese).

(11) T. Yamazaki and M. Sakamoto, "Automatic measurement for QoS mapping function based on B-spline," Proc. 2001 IEICE General. Conf., SB-9-9, pp.741-742, March 2001 (in Japanese).

(12) T. Yamazaki, K. Hasuike, K. Mase, and M. Sengoku, "A study of multimedia communication coordination with content processing," IEICE Technical Report, CQ2001-99, Feb. 2002 (in Japanese).

Others:

(1) T. Yamazaki, "A study on QoS mapping (with a data FD)," ATR Technical Report, TR-AC-0032, June 1999.