A CORBA-BASED QUALITY-OF-SERVICE MANAGEMENT FRAMEWORK FOR DISTRIBUTED MULTIMEDIA SERVICES AND APPLICATIONS

James Won-Ki Hong Jong-Seo Kim
Dept. of Computer Science and Engineering
POSTECH
jwkhong@postech.ac.kr

Jong-Tae Park
School of Electronic and Electrical Engineering
Kyungpook National University
park@ee.kyungpook.ac.kr

Abstract

This paper presents a CORBA-based Quality-of-Service (QoS) management framework for distributed multimedia services and applications. The QoS MIB has been defined for the QoS management of various multimedia services, and consists of information objects that represent a set of layered QoS parameters. These managed objects are organized into four logical groups: service, application, system, and network. We have also defined a set of QoS management services for monitoring and controlling QoS-related resources. Using the designed QoS MIB and management services, we have developed a QoS management system for managing and controlling QoS in a distributed multimedia system called MAESTRO. The prototype management system is Web-based and uses OrbixWeb to interface with the QoS management server which is implemented as a CORBA object, and monitors QoS parameters of distributed multimedia services, which are also implemented as CORBA objects.

Keywords: Quality of Service, QoS MIB, QoS Management, CORBA, Multimedia Services

1. Introduction

The widespread use of distributed multimedia applications has created new challenges in networking, including managing network resources for guaranteeing Quality-of-Service (QoS) [Cam93, Laz91, Woo94, Yam95]. As users become more familiar with multimedia services, QoS must also be approached from the user's point of view, rather than only from the network-oriented view [Alp95, Cro95, Gad95, Hal95]. Users must be given the opportunity to express their requirements for the receiving service in terms of familiar QoS parameters. These parameters can be, in turn, translated into parameters provided by the underlying distributed systems and networks [Jun93].

Distributed multimedia services and applications are time-critical and need management support for ensuring agreed QoS [Geo94, Som95]. An important aspect of distributed multimedia services is that they require QoS guarantees for the transfer and processing of continuous media data (such as video and audio). Emerging networks such as ATM [Min89] and the proposed integrated services with reservations [She94] can provide guarantees on bandwidth and delay for data transfer. Modern computer systems now have sufficient computing power and I/O bandwidth for handling continuous media. Management in computing environments that support multimedia services must promote QoS guarantees for each level of the system [Put95] because the overall QoS depends on the combined QoS of the underlying distributed systems and networks [Hut94]. End-to-end management must include management capabilities for each layer participating in the service.

Researchers have recently proposed new communication architectures which are broader in scope and cover both network and end-system domains. These architectures differ in several ways, probably because different communities developed them. Among differences, QoS specification and QoS parameters are considered fundamental to capture user-level QoS requirements. Currently, no international standard or dominant specification for QoS parameters exists from the service layer to the network layer. This need has motivated us to design and provide a set of standard QoS parameters, as well as QoS management services, which cover the layers from the service layer to the network layer.

In this paper, we propose a CORBA-based QoS management framework for managing QoS of distributed multimedia services and applications. We have developed a set of QoS management services to monitor and control QoS parameters in distributed multimedia applications and their supporting services. These QoS management services have been defined using CORBA IDL and can be used for quick and easy development of management applications. A generic QoS MIB has been defined for the QoS management of various multimedia services. This generic QoS MIB can be extended easily to develop MIBs for specific multimedia services, and used to integrate QoS management into standard network management frameworks (such as SNMP and CMIP [Bla94]). The QoS MIB also provides a set of layered QoS parameters and can be used as standard QoS parameters for managing QoS in multimedia services.

Earlier, we developed a CORBA-based distributed multimedia system called MAESTRO [Yun97] which supports the development and operation of distributed multimedia applications. For validation of our QoS management framework, we have attempted to manage QoS in the multimedia services, which are part of MAESTRO, as well as applications running on MAESTRO. In this paper, we also describe our effort on the prototype implementation of a Web-based QoS management system for MAESTRO. The prototype QoS management system uses OrbixWeb [Ion96] to interface with the management server, which is implemented as a CORBA object and monitors QoS parameters in distributed multimedia services which are also implemented as CORBA objects. Results show that the administrator can manage QoS of the MAESTRO system with a popular Web browser such as Netscape or Internet Explorer.

Section 2 of this paper describes QoS concepts and compares QoS parameters among several previously proposed QoS architectures. Section 3 presents the QoS MIB design. Section 4 defines a set of QoS management services required for the QoS management of distributed multimedia services and applications. Section 5 describes our prototype implementation of QoS management. Section 6 summarizes our work and discusses possible future work.

2. QoS Concepts and Comparison

Work on QoS models and concepts has been done in various scenarios and within a number of projects. Most of them differ according to their assumptions, targets, and solution approaches.

QoS modeling includes: specifying the structures and types of QoS parameters, negotiation of QoS, mapping of QoS, resource reservation methods, methods for updating QoS parameters, and methods for monitoring and maintaining QoS.

QoS is defined as, "A set of qualities related to the collective behavior of one or more objects" in [Iso95]. The International Organization for Standardization (ISO) definition assumes an abstract model of objects, while the International Telecommunications Union – Telecommunications Standardization Sector (ITU-T), includes user's personal degree of satisfaction. Detailed QoS parameters specified in different documents of protocols vary as well.

Table 1 compares approaches according to their detailed QoS parameters specified and used. Although numerous research work focuses on QoS, we have compared only three works (QoS-A [Cam93], Int-serv [Bra94, She95, Int95], and OSI95 [Iso95]). Only important QoS parameters have been listed to avoid a potentially large number of empty entries. Entries with "O" denote a defined parameter and entries with "X" denote an undefined parameter. "(X)" denotes that although an exactly matching parameter is not defined, the requested semantic can be satisfied by another parameter. In this paper, the QoS parameters required for managing QoS in multimedia services are defined in the form of MIB. This can be used to integrate QoS management into standard network management frameworks, and used as the basic QoS parameters in other QoS management architectures.

QoS Parameters	QoS-A	Int-serv	OSI95
Throughput	O	O	O
Burstiness	O	O	X
Packet Size	O	O	O
Transmission Rate	O	O	X
Delay	O	O	O
Jitter	O	O	O
Response Time	X	(X)	X
Error Control	(X)	X	O
Data Corruption	X	X	O
Data Loss	O	X	O
Data Replication	X	X	X
Ordered Delivery	X	O	X
Protection	X	X	O
Priority	X	O	O
Cost	O	X	X
Synchronization	X	X	X
Compression	O	X	X

Table 1. Comparison of QoS Parameters

Four main functions for QoS are: negotiation, mapping, resource reservation, and delivery. QoS negotiation involves settling down the differences between what has been requested and what can be provided between the parties involved in a unicast or multicast connection. In general, an increase or reduction of a single QoS parameter value is possible. QoS mapping has to be done in various separate steps. The main goal is for the end-system and the network to provide the QoS requested by applications. Therefore, application QoS (A-QoS) has to be mapped onto a system QoS (S-QoS) or network QoS (N-QoS), and onto protocol-relevant parameters. S-QoS themselves have to be mapped onto resources and parameters of the applied operating system and the network resources as well. Resource reservation mainly applies to two areas: the local end-system (normally the operating system), and the intermediate system (normally the network). In general, tasks encompass the reservation of resources, such as memory (buffer), bandwidth, processing time (CPU), and scheduling mechanisms for certain application requirements. Finally, every mentioned method above is irrelevant if a communication protocol is not available. The primary tasks of protocol processing delivers the requested QoS to the application.

3. QoS MIB

In this section, we define a generic Quality-of-Service (QoS) management information base (MIB) for multimedia services. We call it a generic QoS MIB because it contains QoS related management information common to most (if not all) multimedia services. Thus, the QoS MIB can be used to monitor and control QoS values in multimedia services. However, if specific QoS management information must be obtained in order to manage a particular multimedia service, the QoS MIB can easily be extended by adding the service-specific QoS management information.

In designing our QoS MIB, we have used much work that was previously done by several IETF working groups. Recent efforts by IETF on defining MIBs for integrated services are similar to ours. This includes “intSrv MIB” [Bak97a] and “intSrvGuaranteed MIB” [Bak97b]. Additionally, a number of RFCs (from RFC 2210 to RFC 2216 inclusively) are defined for integrated services.

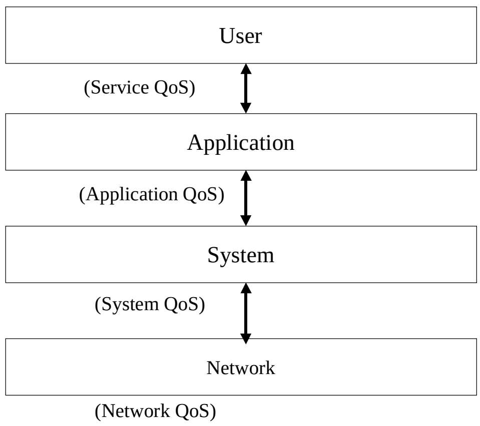

Figure 1. Layered QoS Model

Traditional QoS (as in the ISO standards) mostly referred to measures at the network layer of a communications system. QoS enhancement was achieved by introducing QoS into transport services. For multimedia services, however, the notion of QoS must be extended, as many other services contribute to the end-to-end service quality. To discuss QoS then, we need a layered model of the multimedia service with respect to QoS [Nah95]. Figure 1 provides one model.

The internal architecture of a multimedia service consists of three layers: application, system (including operating system services), and network. Above the application at the client side resides a human user. This layered structure implies the introduction of service QoS, application QoS, system QoS, and network QoS. The QoS MIB mainly includes objects which represent values of each layered QoS parameters. These managed objects are organized into four logical groups (as illustrated in Figure 2): service, application, system, and network. In each group, we define objects maintaining QoS related information. Following is a brief description of the design of each group. Full QoS MIB definitions can be found in [Jsk98].

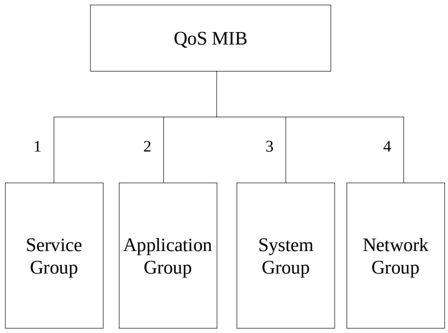

Figure 2. QoS MIB Tree

3.1 Service Group

The service layer QoS parameters specify the service quality which the user wishes to see or hear (e.g., TV quality of video, telephone quality of audio). The service quality, however, is hard to quantify and its evaluation is subjective and user-dependent. Service QoS parameters include the speed which the user listens or views the playback of audio/visual media.

We used a five-level scale to define the quality of a multimedia service. Thus, users can potentially specify one of these levels to express their requirements. The service layer QoS subsumes the widely accepted user requirements, but also allows the specification of a wide range of options. The service group of QoS MIB is defined according to service layer QoS parameters.

3.2 Application Group

The application layer QoS parameters describe requirements for application services specified in terms of media quality, which includes media and transmission characteristics, and media relations, which specify the relations among media. The media quality consists of a stream and component specifications. The stream specification gives the media characteristics of a homogeneous media stream such as sample size, sample rate and priority/importance. If the individual samples in the stream differ in quality, component specification must occur. Each subsample must be specified by the user/application in the stream structure using component specification. The parameterization also includes an application-oriented specification of the required transmission characteristics for end-to-end delivery (e.g., end-to-end delay bounds).

Media relations specify relations among the media streams. Synchronization skew represents an upper bound on time offset between two streams in a single direction. This information can be used for a finer granularity scheduling decision of multimedia streams than a sample rate information of periodic streams provides. If no skew is specified, the system uses the sample rate of each stream for scheduling decisions. Communication relation defines the communication topology, such as unicast (peer-to-peer), multicast (peer-to-group), or broadcast (peer-to-all). The application layer QoS parameters can be mapped from the service layer or directly characterized by users. They can be parameterized differently according to media or service type.

3.3 System Group and Network Group

System QoS parameters describe requirements on the communication services and operating system (OS) services resulting from the application QoS. The OS services require following resources: processing times required for task, secondary storage, and memory buffer. OS resources are needed by the application-layer and network-layer tasks to handle input/output of media and sending/receiving connections. The definition of QoS parameters at the system layer requires mainly CPU scheduling, memory management for buffering, and efficient storage on mass media.

Network layer QoS parameters describe network services requirements. They may be specified in three domains. The first domain includes basic parameters (called throughput specification) such as packet size, packet rate, and burstiness. The second domain includes environment-sensitive parameters (called flow specification) such as inter-arrival delay, round-trip delay and packet loss rate. The last domain includes the specification of the overall communication requirements (called performance specification) such as packet ordering and priorities. We have defined the network QoS parameters on the basis of intSrv MIB [Bak97a] defined by IETF. Additionally, we have used a number of other RFCs and referred to RSVP [Wro97], QoS-A [Cam93] and QoS Broker [Kla95] for our definition.

4. QoS Management Services for Distributed Multimedia Services

The QoS management services for distributed multimedia services are classified into four sets of management-layer services: service, application, system and network. These services are used by the multimedia applications for supporting QoS guarantees. Figure 3 illustrates the services included as part of the QoS Management Service Object (QMSO), which is composed of a number of layers and planes.

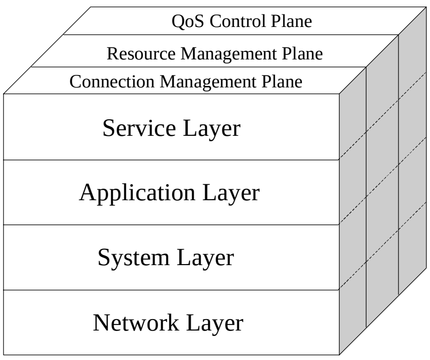

Figure 3. The Structure of QoS Management Services

The service layer offers the user the capability to specify quality requirements and priorities for the employed media services. This service layer also provides the necessary communication primitives for establishing and disconnecting multimedia sessions, translating the underlying QoS parameters to user-perceptible parameters.   The application layer translates user requirements for multimedia services to a set of QoS, application, system, and network requirements. It executes various control mechanisms to compare and control the user-specified quality with the currently available quality. The controlling mechanism operates on different layers (service, application, system and network layer). Examples of control mechanisms are media prioritization or automatic actions to prevent resource saturation in case of QoS violation. The system layer functions as a translator from the application layer QoS parameters to underlying layer QoS parameters and reverse translation. It also has a QoS control mechanism and is responsible for end-system resource management: CPU scheduling, memory and I/O management. The network layer translates upper layer QoS parameters to network QoS parameters and QoS control.

The connection management plane manages multimedia sessions from a user supplied profile. This plane encompasses layer-specific connection managers that bind multimedia processing units (MPUs) at each layer in order to meet end-to-end connectivity. The resource management plane manages and monitors resources. It performs both admission testing and and resource reservation at every level in the end-system. It actively measures the CPU usage, and periodically informs the CPU scheduler of its use. The QoS control plane is the central arbitrator of end-to-end QoS. It is comprised of layer-specific QoS managers that negotiate resources with peer QoS managers, and maintains an internal state associated with application specific QoS. This translates user requests for multimedia flows to a set of QoS parameters. The QoS mapping is based on the definition of QoS MIB, described in the previous section. An important function of the QoS control plane is to monitor layer-specific QoS and report any QoS violations of the contracted profile directly to the multimedia applications. Other QoS violations fielded by QoS control plane include indications from the resource management plane during the negotiation phase.

The QMSO is composed of two management service objects: the application QMSO (appQMSO) and the network QMSO (netQMSO). The appQMSO object provides functions for the service, application, and system-layer QoS management services. The netQMSO object provides functions for the network-layer QoS management service. A subset of appQMSO and netQMSO defined in CORBA IDL are given below, showing the QoS translator procedures involved in the QoS control plane, and the admission control procedures involved in the resource management plane:

```
interface appQMSO {

  short queryQMSO (  // QMSO's entry procedure, called by multimedia application
         in QosParam srvParam, // service QoS parameter
         in QosParam addParam, // additional QoS parameter
         out NoticeType notice); // accept, reject, and modify

  short Service2App ( // QoS translation from service layer
                     // parameter to application layer parameter
         in QosParam srvParam, // service QoS parameter
         out QosParam appParam);  // application QoS parameter

  short App2System ( // QoS translation from application layer parameter
                    // to system layer parameter
         in QoaParam appParam, // application QoS parameter
         out QosParam sysParam); // system QoS parameter
   short GetAppQoS (// Get QoS parameter from application QoS profile (QoS MIB)
         out QosParam appParam); // application QoS parameter
   short SetAppQoS ( // Set QoS parameter in application QoS profile (QoS MIB)
         in QosParam appParam); // application QoS parameter
   short GetSysQoS (// Get QoS parameter from system QoS profile (QoS MIB)
         out QosParam sysParam); // system QoS parameter
   short SetSysQoS ( // Set QoS parameter in system QoS profile (QoS MIB)
         in QosParam sysParam); // system QoS parameter
   short AdmitSysQoS ( //Admission test procedure
         in QosParam sysParam,
         out NoticeType notice); // accept, reject, and modify
   short NegotiateAppQoS ( // Negotiates system QoS between service sender and receiver
         in QosParam appParam,
         out NoticeType notice);
   TrapType MonitorSysQoS ( // Monitor local system resource
         in ServiceType service, // Best or Guaranteed
         in TrapType trap, // signal or no_signal
         in AdaptType adapt); // Adapt, signal, or no_action
};

interface netQMSO {
  short Sys2Net ( // QoS translation from system layer parameter
                 // to network layer parameter
```

```
         in QosParam sysParam, // system QoS parameter
         out QosParam netParam);  // network QoS parameter
  short App2Net ( // QoS translation from application layer parameter
                  // to network layer parameter
         in QoaParam appParam, // application QoS parameter
         out QosParam netParam); // network QoS parameter

   short GetNetQoS (// Get QoS parameter from network QoS profile (QoS MIB)
         out QosParam netParam); // network QoS parameter
   short SetNetQoS ( // Set QoS parameter in network QoS profile (QoS MIB)
         in QosParam netParam); // network QoS parameter
   short AdmitNetQoS ( //Admission test procedure
         in QosParam netParam,
         out NoticeType notice); // accept, reject, and modify
   short NegotiateNetQoS ( // Negotiates network QoS between service sender and receiver
         in QosParam netParam,
         out NoticeType notice);
   TrapType MonitorNetQoS ( // Monitor network resource
         in ServiceType service, // Best or Guaranteed
         in TrapType trap, // signal or no_signal
         in AdaptType adapt); // Adapt, signal, or no_action
};
```

The CORBA IDL definition for QMIO (QoS Management Interface Object) is given below. QMIO is an instrumentation object created for each object that needs to be managed related to QoS. It provides QoS-related information to QMSO from the service object it is instrumented into. It also allows QMSO to modify various values as part of QoS control actions.

```
interface QMIO {
   readonly attribute short attr_count;
   boolena get ( // Get value from a service object in MAESTRO
         in OID oid, // Object ID
         out Attr attr); // Attribute value
   boolena set ( // Set value into a service object in MAESTRO
         in OID oid, // Object ID
         in Attr attr); // Attribute value
};
```

5. Prototype Implementation

In Section 3, we presented a MIB definition for the QoS management of distributed multimedia services. Using this and the QoS management services defined in Section 4, we have developed a prototype QoS management system to validate our concepts. The target system we wish to manage is MAESTRO [Yun97], which is a CORBA-based distributed multimedia system. MAESTRO is an object-oriented, distributed multimedia system, whose goal is to provide multimedia services needed to easily develop and operate a variety of multimedia applications. MAESTRO is composed of four essential distributed service objects: Name Service Object (NSO), Communication Service Object (CSO), Session Service Object (SSO), and Storage and Retrieval Service Object (SRSO).

The Management Service Object (MSO) manages and controls service objects in MAESTRO [Jyk97a, Jyk97b]. MSO interacts with managed service objects via Management Interface Object (MIO) for management purposes. We have extended MSO and MIO and developed QoS Management Service Object (QMSO) and QoS Management Interface Object (QMIO) as shown in Figure 4. QMIO is a specialized version of MIO used for accessing management information related to QoS from multimedia service objects. That is, QoS parameters defined in the QoS MIB are instrumented into the managed objects in the form of QMIO.

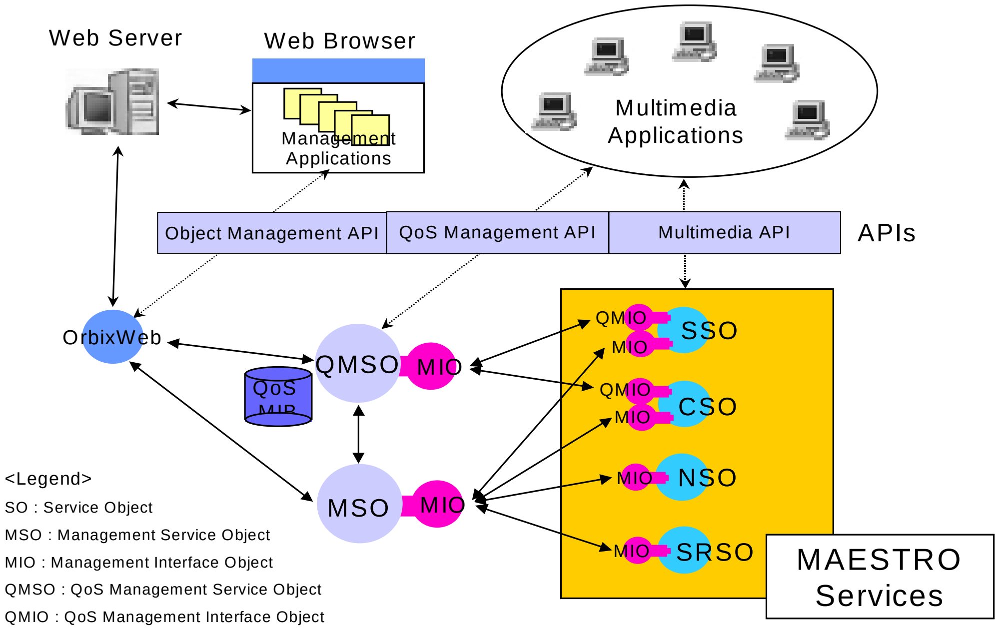

Figure 4. A Web-based QoS Management System Prototype for MAESTRO

In the QoS management system, QMSO performs QoS functions (including QoS control, QoS mapping, resource monitor and control). When users start a multimedia application (such as video conferencing), they are asked to provide the quality of service desired. In our prototype, this is done by the user-level QoS manager tool, specifying the quality of video, quality of audio, window size, etc. QMSO then translates these user requirements into application-specific parameters, and into QoS requirements for the underlying resources. Additionally, QMSO monitors and controls the available host and network resources through the interaction with CSO and SSO. If QoS violation occurs, QMSO sends an event message to MSO and performs an appropriate QoS control procedure.

Because MAESTRO was developed on a CORBA platform, IONA Orbix 2.2 [Ion97], our QoS management service object has also been implemented on the same platform. In order to provide a uniform interface and multi-platform management, our management system has been implemented using Web technology (i.e., using Web server and browser). Administrators can easily manage QoS of MAESTRO services from any platform where a Java-enabled Web browser such as Netscape or Internet Explorer is run. OrbixWeb [Ion96] has been used to allow Java-based management applications to invoke methods on management service objects that have been implemented using CORBA.

Figure 4 is the prototype for our work. The QMSO is initiated by the input of user requirements. The input of user's QoS requirements are performed by the graphical user interface (GUI). This GUI interactively aids the user in selecting QoS characteristics by viewing a sample video with image rate, image height, image width, color, etc. specified through manipulation of a slider (audio might use a test sound passage with specified amplitude, etc.). The user selections (e.g., image height, image color) are translated into application QoS parameters such as frame size, frame rate and others.

The application QoS requirements are mapped into resource requirements for the local system. The QMSO negotiates with the system, using an admission procedure implemented at the application QMSO (appQMSO) level. The admission procedure at the appQMSO level performs two tests against temporal resources: (1) a local schedulability test to see if the tasks can manage I/O from media devices within the required time bounds; (2) an end-to-end delay test to see if tasks can meet the specified end-to-end delay upper bound.

Once local system resources are reserved, then negotiation at the appQMSO level with the remote appQMSO occurs. The negotiation at the appQMSO concerns the sender's ability to accommodate requested multimedia characteristics. This depends on the sender having appropriate I/O devices, processing capacity and storage space available. Until these are determined, appropriate bandwidth allocation cannot be made. Furthermore, this negotiation can also be used to exchange additional application information. If the answer is "accept", the appQMSO initiates the request for network QoS and their resource reservation/allocation, which correspond to the multimedia application QoS.

In network QMSO (netQMSO), the application QoS requirements are translated into network QoS requirements using the QoS translator. The QoS translator translates the provided parameters into network QoS parameters using the QoS MIB definition. After translation, admission to the network level is invoked. The admission service at the network level tests both the network resources such as end-to-end delay and bandwidth, and availability of MAESTRO services (CSO, SSO) through interaction with QMIO. If the admission at the netQMSO is successful, negotiation with remote netQMSO is initiated by the local netQMSO. Finally, the netQMSO waits for the reply from the remote netQMSO. These responses are translated back to the appQMSO, so that the user understands which media at what quality will be transmitted. The multimedia application is initiated with QoS guarantees and provides multimedia services using MAESTRO's services.

6. Conclusions and Future Work

In this paper, we have proposed a CORBA-based QoS management framework for managing QoS of distributed multimedia services. A generic QoS MIB has been defined for the QoS management of various multimedia services. The definition and adoption of the QoS MIB allows for a uniform and standardized QoS management information in multimedia services that may be easily extended to develop MIBs for specific multimedia services.

The QoS MIB can be used easily to integrate QoS management into standard network management frameworks (such as SNMP and CMIP [Bla94]). The QoS MIB provides a set of layered QoS parameters for managing QoS in multimedia services. We have described the QoS management services for distributed multimedia services. They are classified into four layers of QoS management services: service, application, system and network. Multimedia applications use these services to support QoS guarantees. These services are included as part of the QoS Management Service Object (QMSO), which is composed of a number of layers and planes. We also have described our efforts on the prototype implementation of a Web-based QoS management system for MAESTRO. The prototype QoS management system uses OrbixWeb [Ion96] to interface with the management server, which is implemented as a CORBA object and monitors QoS parameters in distributed multimedia services, which are also implemented as CORBA objects. Obviously, the complete standardization of the QoS MIB is a complex and time-consuming process. Therefore, our work is not completed but progressing.

Many open issues still require further investigation. A better understanding of user requirements is necessary and system layer QoS parameters need to be refined. More work is also needed in mapping user requirements into application layer parameters and resource (including system resource and network resource) parameters. Additional parameters for each layer should be taken into account. More work is also needed to define accurate QoS control mechanisms.

Our work, moving towards the vision of a real multimedia environment, makes the best use of network protocols and operating systems that offer QoS guarantees. Unfortunately, today's dominant network, the Internet, ensures only a best-effort approach considering data delivery. The deployment of network protocols that offers QoS guarantees [Ban94] has been rather disappointing due to required transition to new network technologies like ATM.

From the given scenario emerges the need for a set of interrelated protocols that offer QoS guarantees and that easily integrate with the Internet protocol suite. The IETF has developed its Resource Reservation Protocol (RSVP) [Bra97] that permits the reservation of network bandwidth and assignment of priorities to various traffic types. The Real Time Protocol (RTP) [Sch96] works alongside TCP, providing end-to-end delivery of such data as video broadcasting and multi-participant audio and video. With the deployment of these protocols, multimedia services that run on the Internet may offer QoS guarantees. With such guarantees, standardized QoS management information like QoS MIB will be essential for QoS management in distributed multimedia services.

References

[Alp95] B. Alpers, "Applying domain and policy concepts to customer network management", Proc. of ISS'95, Berlin, April 1995, Vol. 2, pp. 356-360.

[Ban94] A. Banerjea et al, "The tenet Real-time Protocol Suite:Design, Implementation, and Experiences", University of California at Berkeley, TR-94-059, November 1994.

[Bak97a] F. Baker, J. Krawczyk, and A. Sastry, "Integrated Services Management Information Base using SMIv2", RFC 1213, September 1997.

[Bak97b] F. Baker, J. Krawczyk, and A. Sastry, "Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2", RFC 1214, September 1997.

[Bla94] U. Black, *Network Management Standards: SNMP, CMIP, TMN, MIBS, and Object Libraries*, McGraw Hill, November 1994.

[Bra94] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, 1994.

[Bra97] R. Braden, D. Clark, and S. Shenker, "Resource ReServation Protocol (RSVP) – Version 1 Functional Specification", RFC 2205, 1997.

[Cam93] A. Campbell, G. Coulson, F. Garc, D. Hutchison, and H. Leopold, "Integrated Quality of Service for Multimedia Communications", Proc. IEEE INFOCOM'93, San Francisco, USA, April 1993, pp. 732-739.

[Cro95] S. Cronjaeger and P. Vindeby, "Translation of multimedia user requirements into ATM switching requirements by using appropriate transport protocols", Proc. of ISS'95. Berlin, April 1995, Vol. 1, pp. 180-184.

[Gad95] B. Gadher et al, "A distributed broadband metropolitan network for residential multimedia applications", Proc. of ISS'95, Berlin, April 1995, Vol. 2, pp. 190-194.

[Geo94] F. Georges, "Performance evaluation & management of interconnected networks", Proc. of SBT/IEEE International Telecommunications Symposium, Rio de Janeiro, August 1994, pp. 322-326.

[Hal95] J. Hall et al, "Customer requirements on teleservice management", Integrated Network

Management IV", Ed. A. S. Sethi et al., Chapman & Hall 1995, pp. 143-155.
[Hut94] D. Hutchison et al, "Quality of service management in distributed systems. Network and Distributed Systems Management", Ed. M. Sloman, Addison-Wesley 1994, pp. 273-302.
[Int95] IETF, Slides from IETF meeting 31, Integrated Service Working Group, ftp://mercury.lcs.mit.edu/pub/intserv, 1995.
[Ion96] IONA, *OrbixWeb*, IONA Technologies Ltd, Release 2.0, December 1996.
[Ion97] IONA, *Orbix 2*. IONA Technologies Ltd, Release 2.2, March 1997.
[Iso95] ISO, "Quality of Service Framework", ISO/IEC JTC1/SC21/WG1 N9680, International Standards Organization, UK, 1995.
[Jsk98] J. S. Kim, J. W. Hong and H. W. Park, "Design and Implementation of Quality of Service (QoS) MIB for Distributed Multimedia Services", Proc. of IEEE GLOBECOM'98, Sydney, Australia, Nov. 8 - 12 1998.
[Jun93] J. Jung and D. Seret, "Translation of QoS parameters into ATM performance parameters in B-ISDN", Proc. of IEEE INFOCOM'93, San Francisco, March 1993, pp. 748-755.
[Jyk97a] J. Y. Kong, J. W. Hong, J. T. Park and D. J. Kim, "A CORBA-Based Management Framework for Distributed Multimedia Services and Applications", Proc. of DSOM, Sydney, Australia, October 1997, pp. 132-144.
[Jyk97b] J. Y. Kong, T. H. Yun, S. W. Park, S. H. Kim, Y. M. Shin and J. W. Hong, "Design and Implementation of a Management System for CORBA-Based Distributed Systems", Proc. of Asia-Pacific Network Operations and Management Symposium, Seoul, Korea, October 1997, pp. 473-485.
[Kla95] K. Nahrstedt and J. M. Smith, "The QoS Broker", University of Pennsylvania, 1995.
[Laz91] A. A. Lazar and G. Pacifici. "Control of resources in broadband networks with Quality of Service guarantees", IEEE Communications Magazine, October 1991, pp. 66-73.
[Min89] Minzer, S.E, "Broadband ISDN and asynchronous transfer mode (ATM)", IEEE Communications Magazine, September 1989, pp. 17-24.
[Nah95] K. Nahrstedt and R. Steinmetz, "Resource management in networked multimedia systems", IEEE COMPUTER, pages 52--63, May 1995.
[Put95] K. J. Putz, "Field trials of multimedia applications in ATM networks: Quality-of-Service, functional and implementation aspects", Proc. of ISS'95, Berlin, April 1995, Vol. 2, pp. 180-184.
[Sch96] H. Schulzrinne et al, “RTP: A Transport Protocol for Real-time Applications”, RFC 1889, January 1996.
[She94] S. Shenker, "An architecture of future Internet", Proc. of Hot Interconnects Symposium, August 1994.
[She95] S Shenker and J. Wroclawski, "Network Element Specification Template", Internet Draft, June 1995.
[Som95] F. Somers and M. Edholm, "Intelligent resource dimensioning in ATM networks", Proc. of ISS'95, Berlin, April 1995, Vol. 2, pp. 62-66.
[Woo94] M. Woo et al, "A synchronization framework for communication of pre-orchestrated multimedia information", IEEE Network, January 1994, pp. 52-61.
[Wro97] J. Wroclawski, "The Use of RSVP with Integrated Services", RFC 2210, September 1997.
[Yam95] N. Yamanaka et al, "Full-Net: A flexible multi-QoS ATM network based on a logically con- figured VC-network", Proc. of ISS'95 Berlin, April 1995, Vol. 2, pp. 195-199.
[Yun97] T. H. Yun, J. Y. Kong and J. W. Hong, "A CORBA-based Distributed Multimedia System", Proc. of the Fourth Workshop on Distributed Multimedia Systems, Vancouver, Canada, July 1997, pp. 1-8.