

# CORBA 환경의 분산 QoS 관리 구조 기반에서 실시간 협약 및 적응 제어의 연구

<sup>U</sup>조동훈\*, 전병택\*, 주수중\*

\*원광대학교 컴퓨터공학과

e-mail:dhcho@gaebyok.wonkwang.ac.kr

## A Study of Real-time Negotiation and Adaptation Control based on Distributed QoS Management Architecture in CORBA Environments

Dong-Hoon Cho\*, Byung-Taek Jun\*, Su-Chong Joo\*

\*Dept of Computer Engineering, Won-Kwang University

### 요 약

최근 다양한 멀티미디어 서비스와 스트림 서비스의 급증과 사용자의 서비스 기대 증가로 인해 기존  
의 컴퓨팅 환경도 인터넷 기반의 분산 멀티미디어 환경으로 변화되고 있다. 이러한 다양한 요구를 만  
족시키기 위해서는 인터넷 환경과 실시간 특성들이 혼합된 폭넓은 범위의 분산 애플리케이션을 지원  
해야 하고, 이들 서비스간의 QoS(Quality of Service)를 보장해야 한다. 이를 위해 본 논문에서는 분산  
환경인 CORBA기반에서 자원들의 효과적인 관리와 서비스들간의 QoS를 보장하는 분산 QoS 관리 구  
조를 제시한다. 즉, 클라이언트 측에 QoS 제어 관리 모듈(QoS Control Management Module)을, 서버  
측에는 QoS 관리 모듈(QoS Management Module)을 각각 분산시켜 QoS 관리 및 제어를 분산 구조화  
함으로써 서버 시스템의 부하를 줄이고, 분산 서비스 응용에 지원을 확장할 수 있도록 하며 모듈들의  
재사용이 가능하도록 하였다. 또한 분산 QoS 제어를 위해 실시간 협약(Negotiation) 및 동적 적응  
(Dynamic Adaptation)과정이 분산환경에서 수행되도록 기능 설계를 하였다. 이들 기능수행의 전반과  
정을 UML(Unified Modeling Language)을 이용하여 순차적 다이어그램으로 기술하였다.

### 1. 서론

요즘과 같이 인터넷 사용자의 멀티미디어 서비스에 대한 기대가 매우 커지고 다양해짐으로 인해 인터넷 기반의 컴퓨팅 환경 자체도 텍스트, 그래픽, 오디오, 비디오 등의 멀티미디어 데이터를 다루는 분산 멀티미디어 컴퓨팅 환경으로 변화되었다. 이러한 환경에서 중요한 특징 중의 하나는 시간 종속적(time-dependent)인 성질을 가지는 미디어를 다룬다는 것이다. 특히, 인터넷 생방송(Live Broadcasting) 서비스와 VoD(Video-on-Demand) 서비스, 실시간 회의(real-time conferencing)등을 통한 오디오/비디오 서비스가 급증하면서 멀티미디어 스트리밍 기술에 대한 관심이 더욱 커지고 있다. 이러한 멀티미디어 스트리밍 기술은 네트워크 상에서 멀티미디어 스

트림 데이터를 실시간으로 주고받는 기술로써, 사용자의 요구를 최대한 수용하기 위한 서비스 기술이다. 최근 이러한 점들을 고려해 분산 멀티미디어 환경에서 다양한 멀티미디어 서비스와 스트림 서비스를 제공하고 이들 서비스들의 QoS를 보장하기 위해 많은 연구들이 진행되고 있다. 특히, 콜롬비아 대학에서는 ATM기반의 다양한 QoS 구조와 명세에 대하여 제안하였고[1], 일리노이스 대학에서는 QoS 자원관리에 대한 연구와 QoS 제어 메커니즘에 대해서 많은 연구가 진행되어 왔다[2][3]. 그리고, OMG CORBA A/V 스트리밍 서비스 명세[4]에 의한 사례 연구로 워싱턴 대학에서는 CORBA를 확장하여 실시간 애플리케이션 개발에 적합한 TAO(The ACE ORB)를 개발하였다[5][6]. 그밖에도 여러 연구 기관에서 서로 특색이 있는 다양한 QoS 프로젝트를 진행하면서 QoS에 관련된 새로운 기술들이 많이 나오게 되었다. 그러나, 이와 같이 많은 연구에도 불구하고

\* 본 연구는 2000년 정통부 대학기초연구지원사업(2000-067-01)의 지원으로 수행 되었음.

고 기존의 연구에서는 사용자의 대다수가 사용하고 있는 인터넷 기반이 아닌 특정 프로토콜이나 특정 환경에 적용되고 있어 QoS 기능이 매우 제한적이며, 확장성과 재사용성의 문제점을 가지고 있다. 또한 QoS 제어와 관리 메커니즘에 있어서도 실시간 특성을 고려한 QoS 제어와 관리 기법이 매우 부족한 실정이다.

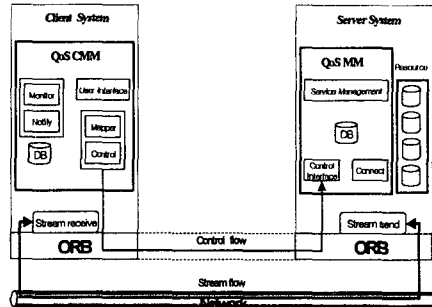
따라서 본 논문에서는 일반적인 QoS 프레임워크에 포함되어야 하는 사항들인 QoS 원리, QoS 명세, QoS 메커니즘에 따라 CORBA 환경에서 멀티미디어 서비스와 스트림 서비스 사용자의 기대 요구를 보다 만족시키고 이러한 서비스간의 QoS를 보장할 수 있는 분산 QoS 관리 시스템을 설계하였다. 제안된 시스템의 구성은 분산된 형태로 종단간(End-to-End) QoS의 제어와 관리를 할 수 있도록 클라이언트 측의 QoS 제어 관리를 위한 모듈과 서버 측의 QoS 관리 모듈로 구성되어 설계하였다. 전체적인 QoS의 제어는 클라이언트 측에서 담당하여 다자간 통신의 경우에 서버 측의 부담을 최소화시킬 수 있도록 하였다. 그리고 각각의 모듈 안에는 이들간의 스트림 서비스와 QoS의 제어 및 관리를 위한 객체들로 구성하였다. 특히, 본 논문에서 제안하고 있는 QoS 제어 기법은 실시간 협약 및 동적 적용에 중점을 두었다.

본 논문의 구성은 2장에서 제안하고 있는 분산 QoS 관리 구조와 서비스 수행절차 대해서 전반적으로 기술한다. 3장에서는 분산 QoS 관리를 위해 객체 모듈로 구성된 QoS 제어 관리 모듈과 QoS 관리 모듈, 그리고 각 모듈을 구성하고 있는 객체에 대해 기술하고, 4장에서는 분산 QoS 관리를 위해 제어 구조에 적용된 QoS 제어 기법인 실시간 협약 및 동적 적용과정의 수행 전반을 UML을 이용하여 순차적 다이어그램으로 기술한다. 그리고, 마지막으로 5장에서는 결론 및 향후 연구방향에 대해서 기술한다.

2. 분산 QoS 관리 구조

제안된 분산 QoS 관리 구조는 분산 제어방식의 형태이며, 두 개의 객체 모듈로 구성된다. 객체 모듈은 클라이언트 측의 QoS 제어 관리 모듈과 서버 측의 QoS 관리 모듈로 구성되며, 모듈 내부는 종단간 스트림 서비스의 QoS를 보장하기 위한 객체들로 구성되어 있다. 이들간의 스트림 흐름(Stream Flow)은 IP기반에서 RTP 패킷을 송수신할 수 있는 스트림 송수신 시스템에서 처리하고, 제어 흐름(Control Flow)은 ORB를 통해서 이루어진다. 다음 [그림 1]은 제안하고 있는 분산 QoS 관리 구조를 나타낸다.

본 구조 내에 QoS 제어 관리 모듈과 QoS 관리 모듈에 대해서는 다음 3장에서 자세히 설명하고, 모듈들 내와 그들간의 절차는 4장에서 기술한다.

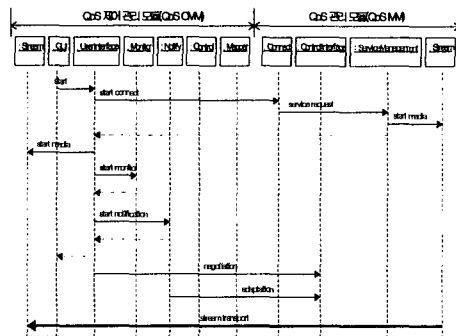


[그림 1] 분산 QoS 관리 구조

2.1 분산 QoS 관리 구조의 수행절차

분산 QoS 관리 구조에서 수행절차는 초기에 서비스 사용자가 QoS 제어 관리 모듈의 사용자 인터페이스 객체를 통해 QoS 관리 모듈의 접속 객체에 연결을 요청한다. 연결이 확정되면 QoS 관리 모듈의 접속 객체는 서비스 관리 객체에게 사용자가 요구한 서비스를 요청한다. 서비스 관리 객체는 자원(Resource)중에서 사용자가 요구한 서비스를 선택한다. 선택된 서비스는 스트림 전송 시스템을 통해 스트림 데이터의 전송을 시작하고, QoS 제어 관리 모듈의 사용자 인터페이스 객체의 오퍼레이션을 통해서 스트림 수신 시스템을 동작시킨다. 그리고, 스트림 수신 시스템을 모니터링하고 수신 상태를 감시할 수 있도록 모니터 객체와 통보 객체를 동작시킴으로써 초기 수행절차가 이루어진다.

다음의 [그림 2]는 분산 QoS 관리 구조에서의 수행절차를 ETD(Event Tracing Diagram)로 표현하였다.



[그림 2] 분산 QoS 관리 구조의 수행절차

### 3. 분산 QoS 관리 구조를 위한 객체 모듈

본 논문에서 제안된 분산 QoS 관리 구조를 위한 객체 모듈은 클라이언트 측의 QoS 제어 관리 모듈과 서버 측의 QoS 관리 모듈 사이의 상호 작용을 통해 스트림 서비스가 이루어지며, 이들간의 QoS를 보장하기 위해 클라이언트 측의 QoS 제어 관리 모듈에서 협의된 QoS 레벨을 유지하고, 협의된 QoS 레벨에 위배될 경우에 현 상황에 맞는 QoS 레벨 값을 재구성한다. 서버 측의 QoS 관리 모듈은 클라이언트 측에서 넘어온 QoS 레벨 값을 가지고 서비스가 가능한 QoS 레벨로 보장해 주는 모듈이다.

본 장에서는 QoS 제어 관리 모듈과 QoS 관리 모듈 내부에 있는 이들 객체들의 구성요소와 이 객체들이 담당하고 있는 기능에 대해서 기술한다.

#### 3.1 QoS 제어 관리 모듈(QoS CMM)

QoS 제어 관리 모듈은 사용자 인터페이스 객체, 모니터 객체, 통보 객체, 매퍼 객체, 제어 객체로 구성되며, 전반적인 QoS의 제어가 이루어진다.

- ① 사용자 인터페이스 객체(User Interface Object)
  - ▶ QoS 제어 관리 모듈과 QoS 관리 모듈을 구성하는 객체들과 통신할 수 있는 오퍼레이션을 제공하는 기능을 담당
- ② 모니터 객체(Monitor Object)
  - ▶ 스트림 데이터의 전송과 수신상태(대역폭, 지연시간, 전송율, 지연율) 및 시스템 자원(CPU, Memory) 상태를 기록, 모니터링 과정을 통해 얻어진 내용을 DB에 저장하고 관리하는 기능을 담당
- ③ 통보 객체(Notify Object)
  - ▶ 모니터 객체에 의해 기록된 정보 분석을 통해 QoS 위반 상황을 검출하여 협약이 위반되었음을 알리는 기능을 담당
- ④ 매퍼 객체(Mapper Object)
  - ▶ 스트림 서비스 형태에 따라 응용에 맞는 MIB의 등록, 삭제, 변경 등을 관리하며, 여기에서는 시스템 카탈로그 정보를 관리하는 기능을 갖는다. 그리고 시스템 카탈로그는 스트림 서비스 환경에서 접속 가능한 시스템 목록을 관리 함
- ⑤ 제어 객체(Control Object)
  - ▶ QoS의 전반적인 제어기능을 담당하고, 매퍼 객체가 제공하는 MIB를 이용하여 실시간 협약 및 동적 적응기법을 통해 두 시스템간에 협약된 서비스가 보장 되도록 함

#### 3.2 QoS 관리 모듈(QoS MM)

QoS 관리 모듈은 서비스 관리 객체, 접속 객체,

제어 인터페이스 객체로 구성되며, 클라이언트 측에서 요청한 서비스와 QoS 레벨에 따라서 서비스를 수행하고 서버 측의 자원을 관리한다.

- ① 서비스 관리 객체(Service Management Object)
  - ▶ 협의된 QoS 레벨에 따라 자원(Resource) 및 스트림을 송신하고 제어할 수 있는 기능을 담당
- ② 접속 객체(Connect Object)
  - ▶ QoS 제어 관리 모듈과 QoS 관리 모듈사이에 스트림 서비스가 개시되도록 접속 설정 및 해제 기능을 담당하며, Mapper Object에서 관리하고 있는 시스템 카탈로그에 등록되어 있는 정보를 통해 상대방 시스템의 정보 획득
- ③ 제어 인터페이스 객체(Control Interface Object)
  - ▶ 제어 객체를 통해서 제공되는 QoS 파라미터 값을 가지고 제공 가능한 QoS 레벨 값을 선택 및 조절하는 기능을 담당

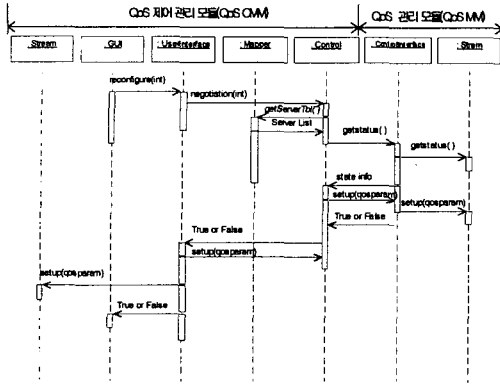
### 4. 실시간 협약 및 동적 적응 절차

제안된 분산 QoS 관리 구조는 QoS 제어기법에서 실시간 협약 및 동적 적응 제어가 가능토록 설계하였다. 이는 QoS 제어 관리 모듈과 QoS 관리 모듈의 상호작용으로써 이루어진다. 본 장에서는 제안된 QoS 관리 구조에서 실시간 협약 및 동적 적응 절차가 이루어지는 과정 및 절차에 대해서 UML을 이용한 순차적 다이어그램으로 기술한다.

#### 4.1 QoS 실시간 협약 절차

QoS 실시간 협약 절차는 초기에 서비스 사용자가 QoS 제어 관리 모듈의 사용자 인터페이스 객체를 통해 QoS 관리 모듈과 QoS 레벨을 협의하기 위해서 제어 객체에 사용자가 요구하는 QoS 정보를 넘긴다. 제어 객체는 매퍼 객체에게 MIB를 요청하고 제공받은 MIB와 사용자가 요구한 QoS 정보를 가지고 협약 알고리즘을 수행한다. 수행 후 협약이 확정되면 QoS 관리 모듈에 있는 제어 인터페이스 객체에 협의된 QoS 파라미터 값을 넘긴다. 제어 인터페이스 객체는 협의된 QoS 파라미터 값을 바탕으로 QoS 관리 모듈의 QoS 레벨을 재구성한다. 그리고 서비스 관리 객체에게 재구성된 QoS 레벨에 맞게 스트림 서비스를 송신할 수 있도록 요청한다. 서비스 관리 객체는 재구성된 QoS 레벨에 맞게 스트림 전송 포맷을 재 설정함으로써 두 모듈간의 협약이 이루어진다.

다음의 [그림 3]은 실시간 협약 절차를 순차적 다이어그램으로 나타내었다.

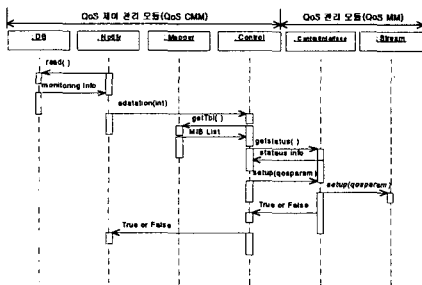


[그림 3] 실시간 협약 절차의 순차적 다이어그램

4.2 QoS 동적 적용 절차

QoS 동적 적용 절차는 QoS 제어 관리 모듈의 통보 객체가 QoS 협약 위반 사항을 모니터 객체를 통해서 관리되는 QoS 정보를 감시하다가 위반 사항이 검출되면 현재의 서비스 상태 정보와 적용 알고리즘을 수행하도록 제어 객체에게 요청한다. 요청을 받은 제어 객체는 현재의 서비스 상태 정보를 참조하여 현재 서비스 수신이 가능한 QoS 파라미터 값을 QoS 관리 모듈의 제어 인터페이스 객체로 넘긴다. 제어 인터페이스 객체에서는 넘어온 QoS 파라미터 값을 기준으로 QoS 레벨을 조정한 후 서비스 관리 객체에게 조정된 QoS 레벨로 스트림 서비스를 송신하도록 요청한다. 요청을 받은 서비스 관리 객체가 스트림 송신 객체를 재구성함으로써 QoS 동적 적용 절차가 이루어진다.

다음의 [그림 4]는 동적 적용절차를 순차적 다이어그램으로 나타내었다.



[그림 4] 동적 적용절차의 순차적 다이어그램

5. 결론 및 향후 연구

본 논문에서는 분산 멀티미디어 환경인 CORBA

기반에서 자원들의 효과적인 관리와 다양한 서비스 간의 QoS를 보장하는 분산 QoS 관리 시스템을 제시하였다. 이를 위해, QoS의 제어와 관리를 분산된 형태의 시스템을 설계하였으며, 클라이언트 측에 QoS 제어 관리 모듈을, 서버 측에는 QoS 관리 모듈을 객체 형태로 구성하였다. 이는 분산 QoS 관리를 구조화함으로써 서버 시스템의 부하를 줄이고, 다양한 서비스 응용에 확장할 수 있도록 하며, 모듈의 재사용이 가능하도록 하였다. 마지막으로 분산 QoS 관리를 위한 제어 구조에서 QoS 제어기법은 실시간 협약과 동적 적용에 중점을 두어 보다 효과적인 QoS 제어기능을 제시하고 있다.

향후 연구로는 좀더 세분화된 QoS 제어와 관리 기능을 위해서 다양한 알고리즘을 본 논문에서 제안하고 있는 분산 QoS 관리 구조에 적용시켜 보고, 적용 가능한 분산 애플리케이션에 이를 응용해서 확장해 보고자 한다.

참고문헌

[1] Campbell, A., Coulson, G. and D. Hutchison, "A Quality of Service Architecture", ACM Computer Communications Review, April 1994.  
 [2] Klara Nahrstedt, Hao-hua Chu, Srinivas Narayan, "QoS-aware Resource Management for Distributed Multimedia Applications", Journal on High-Speed Networking, Special Issue on Multimedia Networking, IOS Press, Vol. 8, No. 3-4, 1998, pp.227-255.  
 [3] Baochun Li, Klara Nahrstedt, "A Control-based Middleware Framework for Quality of Service Adaptations", in IEEE Journal of Selected Areas in Communication, Special Issue on Service Enabling Platforms, 1999, Vol. 17, No. 9, September 1999, pp. 1632-1650.  
 [4] OMG, "Control and Management of A/V streams specification", OMG Document telecom /98-10-05, October 1998.  
 [5] D. C. Schmidt, D. L. Levine, and S. Mungee, "The Design and Performance of Real-Time Object Request Broker", Computer Communications, vol. 21, pp. 294-324, Apr. 1998.  
 [6] S. Mungee, N. Surendran, D. C. Schmidt, "The Design and Performance of a CORBA Audio/Video Streaming Service", HICSS-32 International Conference on System Science, minitrack on Multimedia DBMS and the WWW, Jan. 1999.