

# 분산 멀티미디어 서비스를 위한 CORBA 기반의 QoS 관리자 설계

명진희<sup>U</sup>                      윤용익

숙명여자대학교              컴퓨터학과

jinhhee@cs.sokmyung.ac.kr , yiyoon@sokmyung.ac.kr

## Design of QoS Manager based on CORBA for Distributed Multimedia Services

JinHei Myung<sup>U</sup>                      Yong-Ik Yoon

Dept. of Computer Science, Sookmyung Women's University

### 요 약

최근 네트워크 대역폭과 CPU의 처리 능력이 향상되고 인터넷 사용의 보편화됨에 따라 멀티미디어 데이터를 실시간으로 전송하고 재생하는 멀티미디어 스트리밍 프레임워크가 출현하게 되었다. 이에 서로 다른 하드웨어 플랫폼, 운영체제, 네트워크 환경에서의 상호 운용성(interoperability)을 제공하는 CORBA(Common Object Request Broker Architecture) 기반의 멀티미디어 스트리밍 프레임워크가 출현하고 있다. 현재 멀티미디어 서비스에서 QoS(Quality of Service)를 지원하기 위하여 사용자의 요구 사항 반영 구조 연구, 서비스 지원 미들웨어 구조 연구, 통신 프로토콜 및 전송 시스템 등 부분적으로 QoS(Quality of Service)를 지원하는 연구가 진행되고 있다. 본 논문에서는 상호 운용성을 포함한 CORBA 환경에서 스트리밍 서비스 표준에 의한 시스템에 QoS를 관리 할 수 있도록 해 주는 것이 목적으로, QoS 관리를 위한 QoSM(QoS Manager)을 정의하고, 구성요소의 역할 및 맵핑 방법과 이를 위한 인터페이스를 제시한다.

### 1. 서론

최근 네트워크 대역폭과 CPU의 처리 능력이 향상되고 인터넷 사용이 보편화됨에 따라 많은 멀티미디어 어플리케이션이 개발 및 사용되어지고 있다. 이들 어플리케이션들은 멀티미디어 데이터를 실시간으로 전송하고 재생하는 스트리밍 기술을 사용하여 인터넷 방송, VOD/AOD(Video On Demand/ Audio On Demand) 서비스, 원격 교육, 화상회의 등을 가능하게 해 주고 있다.

사용자들이 멀티미디어 서비스가 점점 더 친숙해 짐에 따라, 서비스 품질(QoS)은 사용자 관점에서뿐만 아니라 지금까지 나타난 네트워크와 시스템 관점에서도 접근해야만 한다.

현재 개발되고 있는 많은 스트리밍 프레임워크들은 멀티미디어 어플리케이션을 위해 각 개발 회사 자체에서 개발되었기 때문에, 어플리케이션들은 서로 다른 하드웨어 플랫폼, 운영체제, 네트워크 환경에서 상호 운용성(interoperability)이 결여되어진, 자체의 스트림 설정/제어 메커니즘을 사용하고 있다. 이에 OMG(Object Management Group)에서는 상호 운용성을 제공하는 CORBA(Common Object Request Broker Architecture)[1]환경에서의 Audio/Video 스트리밍을 제어하고 관리하는데 필요한 주요 인터페이스와 시맨틱을 규정하는 Audio/Video 스트리밍 서비스 표준을 제정하였다.[2] 이를 바탕으로 대표적으로 워싱턴 대학의 TAO Audio/Video 스트리밍 서비스[3]외에 여러 곳[4][5]에서 연구가 되어지고 있는 실정이다. 그러나 현재 Audio/Video 스트리밍 서비스 표준을 기본으로 하여 사용자의 QoS 뿐만 아니라 서비스 지원 미들웨어 구조관점에서의 QoS관리의 보장을 이루어지지 않고 있다.

이에 본 논문에서는 상호 운용성을 포함한 CORBA 환경에서의 스트리밍 서비스 표준에 의한 시스템의 QoS를 관리 할 수 있도록 해 주는 것이 목적이다.

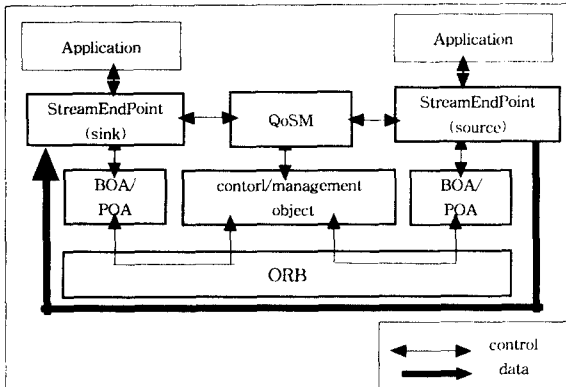
이 논문의 내용 구성은 다음과 같다. 먼저 2장은 멀티미디어 서비스를 위한 QoS 관리자에 대해서 소개할 것이다. 그리고 3장에서는 서비스 품질 관리 시스템을 위한 인터페이스를 IDL(Interface Definition Language)로 정의를 해 주고, 마지막 4장에서는 이 논문의 내용을 요약하고 앞으로의 계획에 대해 언급할 것이다.

### 2. 멀티미디어 서비스를 위한 QoS 관리

본 논문에서는 어플리케이션이 CORBA A/V 서비스를 사용하여 스트리밍 서비스를 하되, QoSM(QoS Manager)에 의해 멀티미디어 데이터에 사용자의 요구사항을 반영한 QoS를 관리할 수 있도록 QoSM을 제안하고 있다. <그림 1>에서 도시하고 있는 것과 같이, QoSM은 멀티미디어 데이터에 사용자 요구의 맞는 서비스 품질의 전송과 재생할 수 있도록 QoS를 관리할 책임이 있다. 또한 사용자의 요구사항이 자원 계층의 QoS 인자와 맞물려야 할뿐만 아니라 네트워크 계층의 QoS의 인자와도 최종적으로 변환이 되어야 한다. 그리하여, QoSM은 <그림 1>와 같이 개괄적으로 그릴 수 있다.

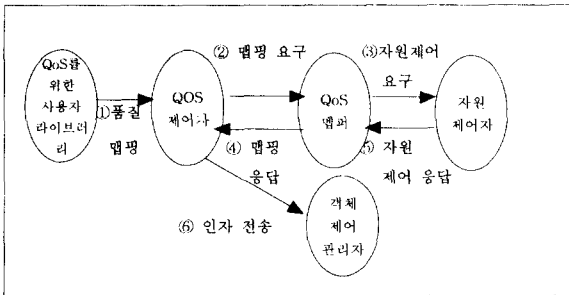
### 2.1 서비스 품질의 제어

서비스 품질의 제어는 일반적으로 미디어 서비스의 품질에 기대되는 사용자의 요구사항을 만족시키기 위해서 적용되는 다양한 종류의 메커니즘을 포함한다. 서로 다른 각 계층에서의 작동이 다르게 일어나며 서비스 품질의 제어는 서비스 품질 제어자에 의해 이루어지며, 조건에 의해 행동을 수행하는 규칙에 의해서 행하여진다. 조건이 참인 경우에 이를 사건(event)이라고 말한다. 서비스 품질 관리자는 사용자, 어플리케이션 계층,



<그림 1> QoS 관리를 포함한 OMG Audio/Video 스트리밍 서비스 구조

자원 계층, 그리고 네트워크 계층에서 사건을 다르게 되며, 이들 사건에 의해 서로 다른 행동(action)을 취하게 된다. <그림 2>에서 도식화한 것 같이, QoSM에서 서비스 품질의 제어가 사용자의 QoS 요구를 받았을 시의 사건 흐름을 보여주고 있다.



<그림 2> QoS 요청시 사건의 흐름

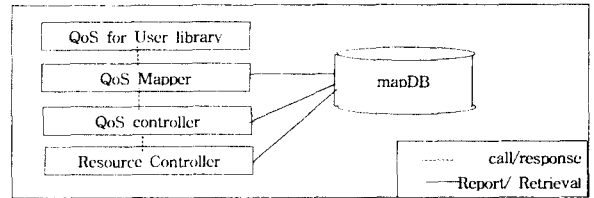
또한, 사용자가 현재 제공받는 미디어 데이터에 새로운 서비스의 품질을 요청 시에 QoS 제어자가 제어해 주게 된다.

### 2.2 QoS의 구성 요소

<그림 3>에 나타난 것과 같이 QoS의 구성요소는 QoS를 위한 사용자 인터페이스, 서비스 품질 맵퍼, 서비스 품질 제어자, 자원 제어의 라이브러리를 제공한다.

- **QoS를 위한 사용자 라이브러리** : 사용자가 멀티미디어 데이터를 위해 QoS를 요청하기 위한 사용자 라이브러리이다.
- **서비스 품질 맵퍼** : 사용자의 요구를 미디어 서비스를 위해 어플리케이션의 특별한 인자로 변환시켜 주고 어플리케이션의 특별한 서비스 품질의 인자는 기반 자원을 위한 서비스 품질 요청으로 변환시켜 준다.
- **서비스 품질 제어자** : QoS 제어자는 현재 제공된 서비스 품질로 제공되는 데이터에 새로운 서비스 품질을 적용시키려 할 때, 기존의 서비스 품질과 새로운 서비스 품질을 비교하고 제어할 수 있게 해준다.
- **자원 제어자** : 미디어의 입력과 출력에서 다루는 미디어 데이터를 보내고 받기 위해 어플리케이션 계층과 네트워크 계층 사이에서 호스트의 자원을 할당받기 위해서 필요하다.
- **mapDB** : QoS 인자가 서로 다른 계층에서 만족하는 QoS 인자로 변환할 수 있도록 도움을 주는 데이터베이스이다. mapDB는 테이블의 집합으로 구성되어 있어, 주로 호출된 미디어 데이터의 정보를 저장하

고, QoS 변환시 정보를 저장하여 필요시 꺼내어 사용할 수 있도록 해 준다.



<그림 3> QoS의 구성 요소

### 2.3 서비스 품질(QoS)의 맵핑

서비스 품질의 맵핑은 본 논문의 핵심으로 사용자, 미디어 서비스 그리고 자원 계층에서 서비스 인자의 변환을 다루는 것이다. 사용자의 요청은 미디어 서비스를 위해 어플리케이션의 특별한 인자로 맵핑되고, 그 맵핑된 인자는 자원 계층과 네트워크 계층을 위한 서비스 품질의 요청으로 변환한다.

- **사용자와 어플리케이션의 계층사이의 QoS 변환**  
사람마다 주관이 다르기 때문에 접근하기가 어렵다. 이에 맵핑표를 두어 비디오는 5가지 수준으로 오디오는 2가지 수준으로 나누어주어, 맵핑표에 의한 변환을 해준다.
- **어플리케이션과 자원 계층 사이의 QoS 변환**  
멀티미디어 어플리케이션의 중요한 면 중의 하나는 호스트로부터의 서비스 품질의 보장을 위해 자원 계층에서 큰 미디어 버퍼링을 위한 메모리 관리와 서비스를 위한 지연 시간을 고려해야 한다. 버퍼 할당은 실제 메모리 안에서 서버로부터 받은 미디어 데이터를 위해 필요한 충분한 메모리 공간을 준비해 주는 것이다. 버퍼를 할당하기 전에 실제 미디어에서 필요로 하는 데이터의 양을 미리 산출할 수 있어야 한다. 그리하여 다음 <수식 1>에 의해 클라이언트가 자신의 호스트에 버퍼를 요구하게 된다.

$$f(dt, jt) = R(S(1 + DT(dt) + JT(jt)))$$

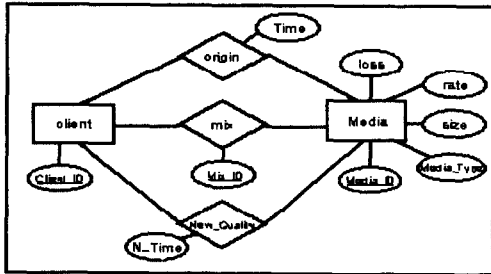
- $f(dt, jt)$ : requested buffer space
- $dt$ : reuquest delay time
- $jt$ : jitter time
- $R$ : Sample Rate
- $S$ : Sample Size
- $DT(dt)$ : number of the Sample size by dt
- $JT(jt)$ : number of the Sample size by jt

<수식 1> 미디어 데이터를 받기 위한 버퍼 요구량

### • 어플리케이션과 네트워크 사이의 QoS 변환

QoS 맵퍼에 의해 수행이 되며, 미디어 데이터의 QoS 인자를 네트워크 QoS인자로 양방향으로 변환시켜 줄 것이다. <그림 4>와 같이, ER 모델(Entity Relation Model)로 표현한 mapDB는 QoS인자가 서로 다른 계층에서 만족하는 QoS로 변환 할 수 있도록 도움을 주고 있다. ER 모델을 바탕으로 mapDB의 스키마를 표현하면 다음과 같다.

Origin (Client\_ID, Media\_ID, Media\_Type, size, rate, loss, Time)  
 Mix(Mix\_ID, Client\_ID, Media\_Type, size, rate, loss, Time)  
 New\_Quality (Client\_ID, Media\_ID, Media\_Type, size, rate, loss, N\_Time)



<그림 4> mapDB의 ER 모델

3. 인터페이스 정의

이 장에서는 2장에서 설명한 QoS에서 QoS를 위해 사용하는 인터페이스를 <표 1>과 같이 제시하였다

인터페이스	설명
RequireQoS	사용자가 멀티미디어를 위한 QoS 요청한다.
UserToApp	사용장의 QoS 요구사항을 어플리케이션 QoS 요구사항으로 변환시킨다.
AppToResource	어플리케이션 QoS를 자원 계층의 QoS 요구사항으로 변환시킨다.
AppToNet	어플리케이션 QoS를 네트워크 계층의 QoS 요구사항으로 변환시킨다.
NetToApp	네트워크 QoS를 어플리케이션 QoS의 요구사항으로 변환시킨다.
ManyToOne	여러 미디어의 QoS인자를 네트워크 QoS 인자로 변환시키기 위해 하나의 미디어 QoS 인자로 혼합시켜 준다.
OneToMany	여러 미디어로 구성된 네트워크 QoS인자를 여러개의 미디어 QoS 인자로 분열시켜준다.

<표 1> QoS 관리를 위해 사용하는 인터페이스

이들 인터페이스는 어플리케이션 프로그램에서 사용자가 더 나은 서비스 품질을 받기 위해, 어플리케이션 수준에서 QoS를 요청하였을 시, 네트워크 수준의 QoS로 변환시켜주기 위해 필요한 것으로, IDL로 표현하기 위해 다음의 구조를 정의한다.

```

struct MQoS {
    string      Media_Num;
    streamQoS  Media;
};

typedef sequence<MQoS> MediaQoS;
    
```

MQoS 구조체는 IDL로 작성이 되었으며, OMG Audio/Video 스트리밍 서비스에서의 QoS 구조체를 확장한 것이다. 한 클라이언트가 서버에 여러 개의 미디어를 요청했을 시, 그에 대한 자료를 정리하기 위한 것으로 한 미디어의 타입과 속성들을 구조체로 묶어준 것이다. 이를 바탕으로 <표 1>에 있는 각각의 인터페이스를 CORBA의 IDL로 정의하면 다음과 같다.

```

RequireQoS ( in string ClientID, in string Media_Num,
             in string Media_Type, in int level);

UserToApp ( in string ClientID, in string Media_Num,
            in string Media_Type, in int level,
            out MediaStream Media);

AppToResource ( in string ClientID,
    
```

in MediaQoS Media);

```

AppToNet ( in string ClientID, in string Mix_ID,
           in streamQoS MixQoS);
    
```

```

NetToApp ( in string ClientID, in string Mix_ID,
           out streamQoS MixQoS);
    
```

```

ManyToOne ( in string ClientID, in MediaQoS splitQoS,
            out string Mix_ID, out streamQoS MixQoS);
    
```

```

OneToMany ( in string ClientID, in MediaQoS MixQoS,
            out streamQoS splitQoS);
    
```

이들 인터페이스에 의해서 더 나은 서비스 품질로 나타나게 된다.

4. 결론 및 향후 과제

본 논문은 서로 다른 하드웨어 플랫폼, 운영체제, 네트워크 환경에서의 상호 운용성을 지닌 CORBA 기반에서의 멀티미디어 서비스에서 서비스 품질을 보장하기 위해 QoS를 제안하였다. QoS는 사용자의 요구 사항에 맞는 서비스 품질로 멀티미디어 데이터를 전송 및 재생시킬 수 있도록 QoS를 관리할 책임이 있다. 그리하여 멀티미디어 데이터 전송을 위한 관련 연구들을 바탕으로 QoS 인터페이스를 정의하였다.

QoS에는 사용자의 QoS 요구 사항을 받아들여 네트워크 QoS까지의 변환을 중점으로 다루고 있으며, QoS는 사용자 인터페이스, 서비스 품질 맵퍼, 서비스 품질 제어자, 자원 제어자로 구성되어 있으며, 서비스 품질 맵퍼에 의해 4가지 계층, 즉, 사용자 계층, 어플리케이션 계층, 자원 계층, 그리고 네트워크 계층으로 서로 QoS를 변환시킨다. 그러나 QoS의 경우, 네트워크 QoS 인자까지의 변환을 하고 있어, 네트워크 상에 데이터가 지나갈 라우터에 대해 RSVP[6]를 사용할 거라고 가정만 했지 완벽히 하지 않았다. 이에 향후 과제는 네트워크 상의 QoS를 효과적으로 하기 위한 라우팅에서의 QoS까지 집목하여 깊이 연구해야 할 것이다.

5. 참고문헌

- [1]Object Management Group, The Common Object Request Broker : Architecture and Specification, 2.2 ed., Feb. 1998.
- [2]Object Management Group, Control and Management of A/V Streams specification, OMG Document telecom/98-10-05 ed., October 1998
- [3]D. C. Schmidt, D.L. Levine, and S. Mungee, "The Design and Performance of Real-Time Object Request Brokers", Computer Communication, vol.21, pp.294-324, Apr. 1998
- [4]J. W. Hong, J. S. Kim, J. K. Park, " A CORBA\_Based Quality of Service Management Framework for Distributed Multimedia Services and Applications" IEEE Network, pp.70-79, March/April 1999
- [5]Y. R. Hong, H. I. Kim, S. Y. Lee, B. S. H. Yoon, C. G. Jeong, " Design and Implementation of a Content Manager in the Multimedia Streaming Framework", Korea Information Processing Society, Vol. 7. NO 2S, pp.733-743, Feb. 2000
- [6]R. Braden, Ed. L.Zhang et al. "Resource ReReservation Protocol(RSVP) - Version 1 Functional Specification." RFC2205, Sept.1997