

# 객체지향 미들웨어상에서 스트림 서비스 향상을 위한 QoS 통합 모델

\*조동훈<sup>1)</sup> \*정창원<sup>2)</sup> \*\*허용도<sup>3)</sup> \*주수종<sup>4)</sup>  
<sup>1)</sup>원광대학교 컴퓨터공학과  
<sup>2)</sup>건양대학교 정보·전자·통신공학부  
 dhcho@gaebyok.wonkwang.ac.kr

## The Integrated Model of QoS for Improving Stream Services on Object-Oriented Middleware

<sup>1)</sup>Dong-Hoon Cho<sup>1)</sup> <sup>2)</sup>Chang-Won Jeong<sup>2)</sup> \*\*Young-Do Her<sup>3)</sup> \*Su-Chong Joo<sup>4)</sup>  
<sup>1)</sup>Dept. of Computer Engineering, WonKwang University

<sup>2)</sup>Dept. of Information · Electronics · Communication Engineering, Konyang University

### 요 약

최근 인터넷 상에서 동영상을 비롯하여 다양한 멀티미디어 서비스가 제공되고 있는 추세이다. 이러한 서비스를 제공하기 위해 스트림 서비스 기술과 분산 객체 기술의 접목이 시도되고 있고, 다양한 스트림 서비스 관리 모델과 이를 지원하기 위한 프로토콜에 대한 많은 연구가 진행되고 있다. 그러나, 기존에 제시된 관리 모델들은 스트림 전송의 서비스 질(QoS)에 대한 지원이 미흡한 실정이다. 또한, QoS에 관련된 기능들이 특정 응용 서비스의 부속 모듈로 개발됨에 따라, 확장이나 재사용하지 못하는 문제점을 갖고 있다. 따라서, 본 논문에서는 분산 객체 기술의 표준인 CORBA를 기반으로 확장 및 재사용이 용이하고 스트림을 지원하기 위한 QoS 지원 통합 모델을 제시하고자 한다. 제안된 모델은 사용자 모듈, QoS 관리 모듈 및 스트림 객체의 세가지 컴포넌트로 구성되며, QoS 관리 모듈은 자원 모니터링, 협약과 자원 적응 기법을 제공하여 QoS를 향상시키고, 사용자 모듈과 스트림 객체간의 상호연동(interaction)을 지원하도록 하였다.

### 1. 서론

최근 멀티미디어 서비스에 대한 관심이 높아지면서, 비디오 텔레컨퍼런스, VoD(Video-on-Demand), 가상현실, 원격교육, 원격진료와 같은 시스템 개발이 활발해지고 있는 추세이다. 특히, 분산 객체 기술의 표준인 CORBA에서도 인터넷기반의 분산 컴퓨팅 환경에서 실시간 멀티미디어 서비스를 지원할 수 있는 표준안을 제시하였다[1]. 또한, 주요 소프트웨어 업체들 간에는 멀티미디어 데이터 처리를 지원하는 네트워크 프로그래밍 환경을 제공하며, MS사의 ActiveMovie나 NetShow[2] SUN사의 JMF(Java Media Framework)[3] 등은 객체 지향 기술을 이용하여 멀티미디어 데이터를 처리하는 응용프로그램을 체계적으로 개발할 수 있는 기능을 포함하고 있다.

그러나, 이러한 서비스를 제공하는데 사용자의 수가 증가함에 따라 네트워크의 혼잡(congestion) 문제, 서비스 품질(QoS)에 대한 요구사항이 변화되고 있는 시점에서 현재의 인터넷은 최선형(best-effort) 타입의 서비스로서 실시간 멀티미디어 스트림 서비스의 질을 보장할 수 없는 한계점을 가지고 있다. 또한 사용자의 수가 증가함에 따르는 트래픽의 급증은 지연(delay), 지연변이(delay variation, jitter), 패킷 손실(packet loss) 등과 같은 문제점을 낳고 있다.

이러한 문제점을 해결하기 위해, 현재 사용하고 있는 인터넷 환경에서 멀티미디어 스트림 서비스를 제공하기 위해 네트워크 트래픽 제어(Traffic Control), 혼잡제어(Congestion Control), 자원예약(Resource Reservation), 대역폭 적응(Bandwidth Adaptation) 방법 등을 이용하여 이를 해결하려는 연구가 진행되어 왔다[4,5,6].

따라서, 본 논문에서는 분산 객체 기술인 CORBA에서 제안하고 있는 오디오/비디오 스트리밍 표준을 참조하여 플랫폼에 독립적이며 확장이나 재사용이 용이한 객체 서비스 형태의 QoS 통합 모델을 제안하고자 한다. 여기에 QoS 제어 기법은 사용자 제어가 가능한 애플리케이션 레벨의 협약 기법과 자원 모니터링을 통한 동적 적응 기법이 가능한 QoS 관리 모듈과 클라이언트/서버의 사용자 모듈간의 상호연동을 지원하도록 하였다.

본 논문의 구성은 2장에서 제안하고자 하는 QoS 통합 모델에 대해서 설명한다. 3장에서는 이에 대한 설계 부분으로 QoS 관리 모듈과

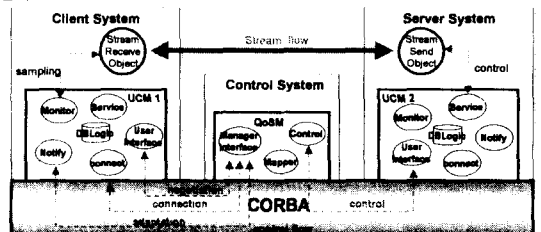
사용자 제어 모듈을 UML의 클래스 다이어그램과 IDL설계 부분으로 나타내고, 4장에서는 본 논문에서 제안하고 있는 QoS 관리 모듈에서 사용된 QoS 제어 과정에 대해서 설명하고, 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해서 기술한다.

### 2. QoS 통합 모델

본 장에서 제안하고 있는 CORBA 기반의 분산 객체 시스템에서 QoS 통합 모델에 대한 전반적인 내용에 대해 기술한다.

#### 2.1 전체시스템 구성

본 논문에서 제안된 QoS 관리 모듈은 중앙 제어방식으로 종단간에 스트림 흐름을 관리하며, 관련 있는 객체를 묶기 위한 객체 모듈(Object Modul)과 스트림 송수신 객체 또는 프로그램으로 구성된다. 객체 모듈은 관련된 객체를 패키징화하는 단위로 정의하며, 하나이상의 객체와 이들 객체를 관리하는 인터페이스 객체로 구성된다. 제안된 통합 모델에서는 스트림 송수신 시스템간에 QoS를 보장할 수 있도록 상호연동하는 두 개의 사용자 제어 모듈(UCM: User Control Module)과 하나의 QoS 관리 모듈(QoSM: QoS Module)을 객체 모듈로 정의한다. 다음 (그림 1)은 본 논문에서 제안한 QoS 통합 모델의 전체 구조를 나타낸다.



UCM : User Control Module QoSM : QoS Management Module

(그림 1) 시스템 구성도

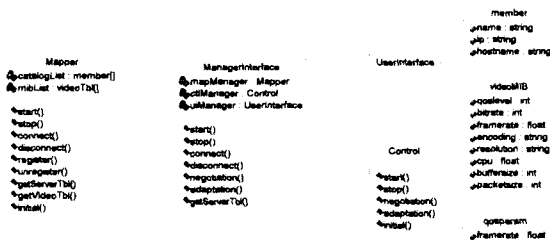
### 3. QoS 통합 모델 설계

본 장은 설계 부분으로 먼저 UML(Unified Modeling Language)을 이용하여 QoS 통합 모델의 구성요소 중 QoS 관리 모듈과 사용자 제어 모듈에 대한 클래스 다이어그램을 보이고, 관리자 인터페이스 객체에 대한 IDL(Interface Definition Language)을 기술한다.

#### 3.1 QoS 관리 모듈

QoS 관리 모듈은 관리자 인터페이스, 매퍼, 제어 객체로 구성되며, 메소드에 사용되는 자료구조는 member, qosparam 구조 외에 videoMIB 클래스의 자료 구조를 사용한다. QoS MIB를 구성하는 각 파라미터는 videoMIB에 정의한다.

다음 (그림 2)는 QoS 관리 모듈의 클래스 다이어그램을 나타낸다.

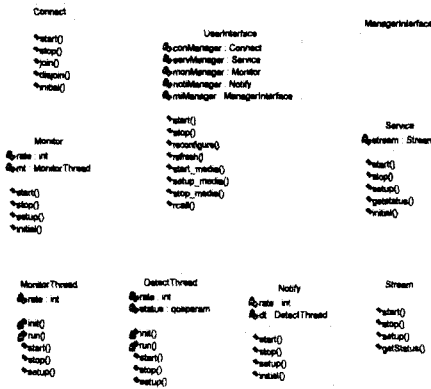


(그림 2) QoS 관리 모듈의 클래스 다이어그램

#### 3.2 사용자 제어 모듈

사용자 제어 모듈은 사용자 인터페이스 객체, 접속 객체, 서비스 객체, 통보 객체, 모미터 객체, DB 로직(DB Logic)으로 구성된다.

다음 (그림 3)은 사용자 제어 모듈의 클래스 다이어그램을 나타낸다.



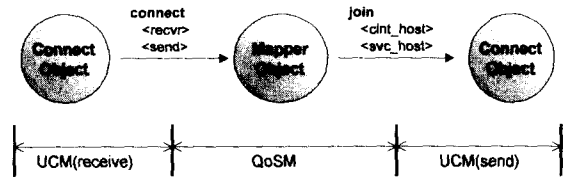
(그림 3) 사용자 제어 모듈의 클래스 다이어그램

사용자 제어 모듈과 스트림 서비스 종단간에 상호작용에 대한 초기 접속에 대한 수행질차는 다음과 같고, 접속 과정은 (그림 4)에서 나타낸다.

▶ 수신측 접속 객체가 QoS 관리 모듈내의 매퍼 객체에게 연결 요청

메시지를 접속할 사용자 정보와 함께 보낸다.

- ▶ 매퍼 객체는 서비스 연결을 위해 송신측 접속 객체에게 join 메시지를 보낸다.
- ▶ 송신측 접속 객체가 스트림 서비스 접속 승인을 함으로써 접속 절차가 이루어진다.



(그림 4) 접속 과정의 데이터 흐름도

#### 3.3 IDL 설계

관리자 인터페이스 객체는 QoS 관리 모듈을 구성하는 매퍼 객체와 제어 객체에 대한 객체 참조자(Object Reference)를 관리하기 위해 mapManager, ctlManager, uiManager의 애트리뷰트를 유지하며, 매퍼 객체와 제어 객체를 초기화하기 위해 start() 오퍼레이션을 제공하고, QoS 관리 모듈의 제어 서비스를 정지시키기 위해 stop() 오퍼레이션을 제공한다. 매퍼 객체에게 서비스 연결/해체를 요청하기 위해 connect() 오퍼레이션과 disconnect() 오퍼레이션을 제공하며, 제어 객체에게 협약과 자원 적응 알고리즘 수행을 요청하는 negotiation() 오퍼레이션과 adaptation() 오퍼레이션을 제공하며, 시스템 카탈로그 데이터를 제공하기 위해 getServerTbl() 오퍼레이션을 제공한다.

다음 (그림 5)는 관리자 인터페이스 객체의 IDL을 나타낸다.

```
interface ManagerInterface {
// Attributes
attribute Mapper mapManager;
attribute Control ctlManager;
attribute UserInterface uiManager;

// Operations
boolean start(in string local, in string remote,
              in string manager);
boolean stop();
boolean connect(in member receive, in member send);
boolean disconnect();
boolean negotiation(in long qoslevel);
boolean adaptation(in long qoslevel, in qosparam param);
catalogList getServerTbl();
};
```

(그림 5) 관리자 인터페이스 객체의 IDL

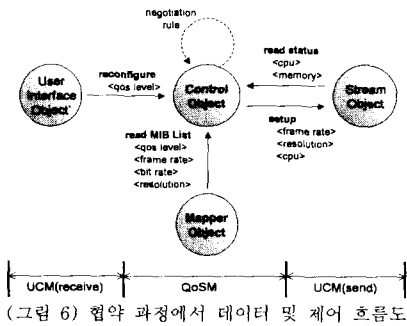
### 4. QoS 제어 기법

본 장에서는 객체간의 관계를 중심으로 QoS 관리 모듈에서 제공되는 QoS 제어기법 중 협약, 자원 적응, 모니터링 과정을 객체간의 데이터 및 제어 흐름도로 나타낸다.

#### 4.1 협약 과정

협약 과정은 (그림 6)에 나타난 데이터의 교환을 통해 이루어지며, 이에 관련된 객체로는 사용자의 재설정 메시지를 전달하는 사용자 인터페이스 객체와 협약 알고리즘을 수행하는 제어 객체, 협약에 사용되

는 관리 정보를 제공하는 매퍼 객체 그리고 협약된 서비스 레벨을 제공하는 스트림 송신 객체로 구성된다.

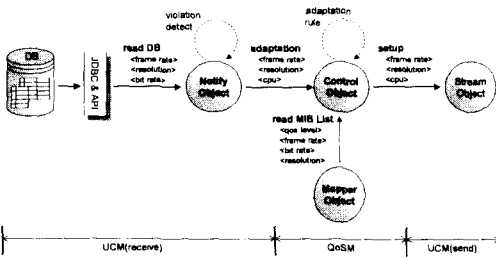


(그림 6) 협약 과정에서 데이터 및 제어 흐름도

- ▶ GUI을 이용해 사용자가 원하는 QoS의 등급과 함께 제어 객체에게 서비스 재 설정을 요구한다.
- ▶ 사용자의 요구를 받은 제어 객체는 매퍼 객체에게 서비스 관리 정보인 QoS MIB을 요청한다.
- ▶ 매퍼 객체는 요청한 QoS MIB를 제어 객체에게 넘긴다.
- ▶ 제어 객체에 기술된 협약 알고리즘에 따라 서비스 제공 여부를 판단한다.
- ▶ 만약 협약이 성공한 경우 제어 객체가 스트림 전송 객체에게 협약된 서비스가 가능하도록 서비스 질(QoS) 파라미터를 포함하여 재 설정 메시지를 보낸다.
- ▶ 재 설정 메시지를 받은 스트림 전송 객체는 전달된 파라미터로 재 설정한다.

4.2. 자원 적응 과정

자원 적응 과정은 스트림 서비스 과정에서 발생하는 예외 상황이나 QoS 유지가 어려운 경우에 적절한 QoS로 낮추어서 종단간에 서비스의 중단을 방지하기 위한 목적에서 주로 이용된다. 적응 과정은 (그림 7) 의 객체간의 메시지와 데이터의 교환으로 수행된다.

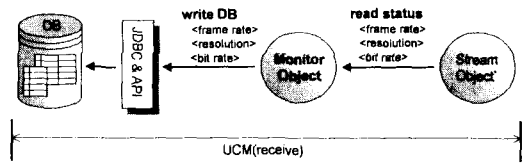


(그림 7) 자원 적응 과정의 데이터 흐름도

- ▶ 모니터 객체에 의해 기록된 서비스 상태 정보를 DB로부터 읽는다.
- ▶ DB에서 읽은 데이터를 분석하여 협약된 서비스 질(QoS) 위반 여부를 검사한다.
- ▶ 만약 위반 사항이 검출되면, QoS 관리 모듈의 제어 객체에게 Adaptation 메시지를 통해 자원 적응 과정을 요청한다.
- ▶ QoS 관리 모듈내의 제어 객체에서 현재 서비스되는 파라미터 정보를 기초로 설정되어 있는 QoS MIB에서 적절한 서비스 레벨을 선정한다.
- ▶ 선정된 레벨에 맞는 파라미터 값으로 스트림 전송 객체를 재 설정한다.

4.3. 자원 모니터링 과정

자원 모니터링 과정은 스트림 서비스의 유지에 이용되는 서비스 질(QoS) 파라미터를 모아 DB에서 저장하는 과정을 말하며, 모니터링 절차는 (그림 8)의 메시지와 데이터의 교환으로 이루어진다.



(그림 8) 자원 모니터링 과정의 데이터 흐름도

- ▶ 모니터 객체가 주기적으로 스트림 수신 상태(프레임 전송률, 해상도) 및 시스템 자원(CPU 사용량)의 소비량을 검출한다.
- ▶ 검출된 내용을 DBMS 모듈이나 전용 API를 통해 DB 테이블에 기록한다.
- ▶ 위 과정을 반복적으로 수행한다.

5. 결론

본 논문에서는 객체지향 미들웨어인 CORBA 기반의 분산 객체 시스템에서 실시간 스트림 서비스를 지원하기 위한 QoS 통합 모델을 제안하였으며, 제안한 모델은 사용자 제어 형태의 협약과 스트림 수신 상황에 동적으로 자원 적응이 가능하도록 사용자 제어 모듈과 QoS 관리 모듈을 설계하였다. 제안한 모델을 설계하기 위해 객체지향 모델링 도구인 UML을 지원하는 Rational Rose이용하여 모델링 하였다. 본 논문에서 CORBA를 이용한 QoS 관리 모듈과 스트림 송수신 시스템과 분리된 형태로 설계하여 다양한 응용 스트림 서비스 개발이 용이하고, 플랫폼 독립성과 네트워크 자원의 통합 관리가 가능하도록 설계하고, 재사용 모듈을 사용해 쉽게 멀티미디어 응용 서비스 개발이 가능하도록 일반적인 분산 객체 서비스 형태의 QoS 관리 모델을 제시하였다.

향후 연구과제로는 서비스 질(QoS) 관리의 정확도를 높이고, 서비스 중에 발생하는 예외 상황에 능동적으로 대처할 수 있도록, 예외 처리 클래스와 루틴을 추가하는 연구가 필요하다. 또한 오디오와 비디오 스트림을 동시에 전달하는 다중 스트림 처리 기법과 다중 스트림 처리에 필요한 동기화 기법을 제안하고자 한다. 그리고 스트림 플레이어의 기능을 확장시켜 다양한 응용이 가능하도록 하며, 다자간 서비스를 위하여 세션 관리(Session Management) 방안을 본 통합 모델에 추가하고자 한다.

참고 문헌

[1] Object Management Group, Control and Management of A/V Streams Spec., OMG Document telecom/98-10-05 ed., October 1998.  
 [2] MS:NetShow, <http://www.microsoft.com/windows/windowsmedia/>  
 [3] SunSoft, "Java Media Framework API Guide", <http://java.sun.com/products/java-media/jmf/2.0/jmf20-08-apidocs/index.html>  
 [4] M.Alfano, R.Sigle, "Controlling QoS in a collaborative multimedia environment." Proc. of the 5th IEEE International Symposium on High-Performance Distributed Computing(HPDC-5), Aug7-9, 1996, Syracuse(NY), USA  
 [5] Witana V. Fry, M. and Antoniadis M. "A software Framework for Application Level QoS Management". To appear in Proc. of the Seventh International Workshop on Quality of Service(IEEE/IFIP IWQoS '99), 1-4th June, 1999.  
 [6] Stardust Forum, Internet Bandwidth Management 2 (iBAND2) Conference, <http://www.stardust.com/iband2>. San Francisco, May 1999.