

유비쿼터스 컴퓨팅에서 분산 멀티미디어 서비스를 위한 하이브리드 QoS 관리 모델

정 창 원[†] · 이 건 엽^{**} · 주 수 중^{***}

요 약

유비쿼터스 컴퓨팅은 컴퓨터 시스템의 물리적인 공간을 확장시켰으며, 보다 동적인 분산 시스템으로 이끌었다. 이러한 환경에서는 다양한 디바이스와 자원 그리고 사용자 요구사항에 대한 새로운 QoS 파라미터가 요구되고 있다. 본 논문에서는 유비쿼터스 컴퓨팅 환경에 적합한 정적 QoS 파라미터와 동적 QoS 파라미터를 정의한 새로운 하이브리드 QoS 관리 모델을 제안하였다. 제안한 시스템 구성은 분산된 형태로 종단간 QoS 제어 및 관리할 수 있도록 클라이언트 측에는 QoS 제어 관리를 위한 모듈(QoS CMM), 서버 측에 QoS 관리 모듈(QoS MM), 그리고 중앙형 QoS 제어 및 관리를 위한 자원 QoS 관리 모듈(RQoS MM)을 구성하였다. 정적 QoS 파라미터는 자원 QoS 관리 모듈에서 처리하며, 동적 QoS 파라미터에 대한 전체적인 QoS 제어는 QoS CMM에 의해 다자간 통신의 경우에 서버 측의 부담을 최소화 시킬 수 있도록 하였다. 끝으로, 그래픽 사용자 인터페이스를 통해 위치 기반 애플리케이션으로 클라이언트 단말형태인 데스크탑 PC, 노트북, PDA에 따른 멀티미디어 서비스 수행 결과 화면을 보였다.

키워드 : 서비스 품질, 하이브리드 관리 모델, 멀티미디어 서비스, 유비쿼터스 컴퓨팅

A Hybrid QoS Management Model for Distributed Multimedia Services in Ubiquitous Computing

Jeong Chang Won[†] · Lee Geon Yeob^{**} · Joo Su Chong^{***}

ABSTRACT

Ubiquitous computing has extended the computer system into the whole physical space and has ushered the emergence of more dynamic distributed systems. This environment require unique QoS parameters for various devices, resources and user requirements. In this paper, we propose a new hybrid QoS management model which defines a static-dynamic QoS parameter that is more appropriate to the ubiquitous computing environment. This model consists of the QoS Control Management Module(QoS CMM) in the client side and the Resource QoS Management Module (RQoS MM) in the server side. The RQoS MM deals with the static QoS parameters and the whole QoS control of the distributed control(QoS CMM) in order to minimize server load in cases of multiple communication. Finally, we present the experimental result of our location based application using a graphical user interface that shows the multimedia service execution of selected client device types such as desktop PC, notebook and PDA.

Keyword : QoS, Hybrid Management Model, Multimedia Services, Ubiquitous Computing

1. 서 론

컴퓨터 환경 변화에 따라 분산 멀티미디어 서비스에 대한 개발과 관심이 활발해 지고 있다. 초기에는 웹 하부 구조가 갖는 단순성과 개방성을 가지고 기본적으로 텍스트 또는 그래픽을 다루도록 설계되었다. 그러나 많은 일반 사용자나 응용 서비스 개발자 및 콘텐츠 제공자들이 텍스트뿐만 아

니라 오디오 및 비디오 데이터를 전송하거나 받을 수 있는 서비스를 요구하고 있다. 이에 따라 다양한 멀티미디어 데이터와 스트림 서비스를 전송하는데 필요한 통신프로토콜 개발을 촉진시켰다. 이러한 통신 프로토콜의 표준으로 인터넷 표준 기관인 IETF(Internet Engineering Task Force)에서는 RTP(Real-Time Transport Protocol), RTCP(Real-Time Control Protocol)[1, 2], RTSP(Real-Time Streaming Protocol)[3], RSVP(Resource ReReservation Protocol)[4]를 제정하였다. 또한 인터넷에서 사용자의 요구사항에 따른 서비스 품질을 제공하기 위해 많은 연구그룹을 통해 통합서비스(Integrated Services)[5] 모델, 차등서비스(Differential

※ 이 논문은 2007년도 원광대학교의 교비 지원에 의해서 수행됨.
† 정 회 원 : 원광대학교 전기전자 및 정보공학부 박사후 연구원
** 정 회 원 : 군장대학 자동차디지털공학부 부교수
*** 정 회 원 : 원광대학교 전기전자 및 정보공학부 교수
논문접수 : 2007년 7월 9일, 심사완료 : 2008년 2월 5일

Service) 모델[6], 다중 프로토콜 레이블 스위칭(Multi Protocol Label Switching)[7] 등 여러 가지 서비스 모델과 메커니즘을 제시하였다. 이와 함께 객체 지향 개념의 등장으로 발전된 분산 객체 기술(CORBA, DCOM, Java 등)을 이용하여 분산 멀티미디어 서비스를 지원할 수 있는 하이텔 베르그 QoS 모델[8], OMEGA[9], CME[10], QuO[11] 모델 등이 제안되었다. 이러한 연구를 통해 제안된 프로토콜이나 QoS 아키텍처는 네트워크 환경에 적용될 수 있는 기술이 대부분으로 클라이언트에서는 QoS의 요청을 위한 임계값(threshold)을 서버에서는 QoS 제어에 중점을 두어 구현하였다[12].

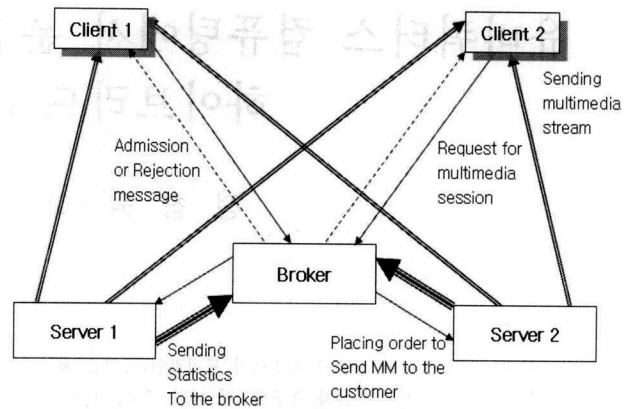
그러나 유비쿼터스 컴퓨팅 환경은 사용자의 단말형태와 단말기의 처리 성능이 향상되고 사용자의 서비스 요구가 다양화됨에 따라 서로 다른 서비스 품질을 갖는 멀티미디어 서비스 제공이 필요하게 되었다[13]. 기존의 QoS 파라미터인 장치 가용성(availability), 응용 요구사항, 정책, 가용한 네트워크 자원과 더불어 개인적인 상황에 대한 기준이 필요하다[14]. QoS의 개념이 초기에는 주로 네트워크 서비스 레벨에서의 측정을 기초로 하였으나, 최근에는 종단 시스템(End-to-End)간의 서비스 개념으로 확대되고 있다[15].

본 논문에서는 입-출력 장치와 서버 응용 간의 애플리케이션 레벨에서의 QoS와 사용자가 인지하는 주관적인 QoS를 위해 QoS 파라미터를 정적 QoS 파라미터와 동적 QoS 파라미터로 분류하여 재정의하고, 기존의 멀티미디어 서비스를 제공하기 위한 QoS 관리 모델인 중앙 집중형과 분산형 모델을 분석하여 새로운 모델을 제시한다. 제안한 시스템 구성은 분산된 형태로 종단간 QoS 제어 및 관리할 수 있도록 클라이언트 측에는 QoS 제어 관리를 위한 모듈(QoS CMM), 서버 측에 QoS 관리 모듈(QoS MM), 그리고 중앙형 QoS 제어 및 관리를 위한 자원 QoS 관리 모듈(RQoS MM)을 구성하였다. 정적 QoS 파라미터는 자원 QoS 관리 모듈에서 처리하며, 동적 QoS 파라미터에 대한 전체적인 QoS 제어는 클라이언트 측에서 전담하여 다자간 통신의 경우에 서버 측의 부담을 최소화시킬 수 있도록 하였다. 특히 QoS 제어 기법 중 본 논문에서는 실시간 협약(Real-time negotiation) 및 동적 적응(Dynamic Adaptation)에 중점을 두었다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 기존 QoS 모델의 구조와 비교 분석한 내용에 대해 기술하고 3장에서는 제안한 QoS 모델 구조와 구성요소간의 상호작용을 그리고 4장에서는 제안한 모델을 기반으로 사용자 인터페이스를 통해 검증하고 수행 평가 내용을 보인다. 그리고 5장에서 결론과 향후 연구내용을 기술한다.

2. 관련 연구

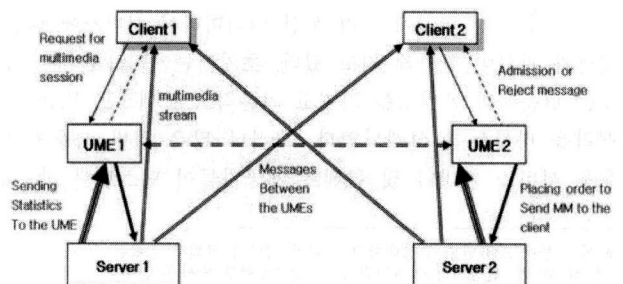
기존 QoS 관리 모델의 구조는 크게 중앙 집중형과 분산형 QoS 관리 모델로 구분되어 연구해 왔다. 특히 중앙 집중형의 경우 브로커 구조를 채택하여 클라이언트와 서버간의



(그림 1) 중앙 집중형 멀티미디어 서비스를 위한 브로커

중재자 역할을 하도록 하였다. (그림 1)은 두 개의 클라이언트와 서버 그리고 이들을 중재하는 브로커를 보인다[17].

클라이언트와 서버간의 QoS 적응과 제어를 담당하는 브로커의 기능은 최초 멀티미디어 서비스를 받기 위한 클라이언트로부터 멀티미디어 서비스 요청을 받아들인다. 브로커는 멀티미디어 자원, 서비스에 대한 상태 정보와 클라이언트가 제시한 품질 정보를 관리한다. 멀티미디어 서비스를 제공하는 각 서버는 자신의 상태 정보를 브로커에게 전달한다. 브로커는 이를 기초로 QoS 제약 조건을 위반하지 않는 시스템의 최대 이용률을 갖는 멀티미디어 스트림의 각 구성요소를 위한 서버와 선택된 QoS 레벨의 집합을 결정한다. 그리고 클라이언트가 요청한 서비스 품질에 대한 협약 및 요청에 대한 승인 또는 거절에 대해 클라이언트에게 통지한다. 또한 여러 개의 서버일 경우 클라이언트에게 데이터를 보내기 위한 서버들의 순서를 정한다. 브로커 기반의 QoS 관리 구조를 채택한 모델은 OMEGA 구조와 OMG CORBA 기반의 오디오/비디오 스트림 구조가 있다[11, 17]. 브로커에 의한 QoS 적응과 승인 제어의 문제점은 중앙집중식의 브로커에 의해 행해짐으로 클라이언트와 서버간의 QoS 파라미터 값의 변화에 따른 빈번한 상호작용과 실시간 적응에 미흡하다. 이와 반면에 분산형 QoS 관리 모델은 각 클라이언트와 서버 사이에 종단간의 QoS 제어를 위한 모듈인 UME(Utility Model Environment)에 의해 QoS 협약과 승인 제어를 지원한다. 이는 중앙집중식 브로커 제어 구조보다 효과적이다. 이에 대한 구조는 다음 (그림 2)와 같다[16, 17].



(그림 2) 분산형 멀티미디어 서비스를 위한 브로커

분산 구조를 채택한 모델로는 QuO, DMSS가 있다. 이러한 모델에서 채택한 분산 구조는 중앙집중식 브로커 제어 구조보다 관리적인 측면이 강하며, 각 분산 멀티미디어 서비스 서버의 각 서버들은 클라이언트들의 요청한 멀티미디어 세션의 QoS 적응성과 승인제어를 UME에 의해 이루어진다. 이는 승인 제어는 UME들에 의해 이루어지기 때문에 서버들을 필요에 의해 추가할 수 있기 때문에 확장성을 제공한다. 또한 서버의 문제점인 다운이나 실패 등을 UME에 의해 감지하여 처리하기 때문에 고장허용(fault-tolerance) 메커니즘을 제공한다. 이러한 분산 구조에 의한 QoS 적응과 승인 제어의 문제점은 클라이언트와 서버간의 UME간 반드시 존재해야 하며, 이들 간의 정보 일관성 문제를 야기시킨다. 또한 중앙 집중형 보다는 복잡한 알고리즘이 요구된다.

기존에 제시된 QoS 구조 중에서 몇 가지를 선택하여 이들 구조의 특징 및 구조에 대해서 기술하였다. 여기에서 제시한 구조들에 대한 비교로 <표 1>을 통해 나타낸다.

기존 멀티미디어 서비스를 제공하기 위한 클라이언트와 서버 시스템이 워크스테이션 기반이거나 데스크탑 PC이기 때문에 대부분 네트워크 위주의 통신에 관련된 지연, 지터, 에러율과 같은 QoS 정의를 통한 서비스 품질에 중점을 두었다. 그러나 컴퓨터 시스템 개발 기술의 발전으로 인하여 다양해진 클라이언트 시스템 환경은 네트워크 위주의 QoS 보다는 중단간 시스템의 환경을 지원하며, 이에 적합한 자원에 대한 QoS가 필요하다. 이와 더불어 사용자의 주관적인 QoS가 더욱 요구되고 있다. 즉, 이러한 컴퓨팅 환경에서 중단 시스템간의 적합한 QoS 승인 및 제어 기술에 기초가 되는 QoS 파라미터에 대한 재정의가 요구된다.

3. 하이브리드 QoS 관리 모델

3.1 제안한 QoS 관리 모델

기존 QoS 관리 모델의 중앙 집중식 형태와 분산형태를 혼합한 하이브리드 QoS 관리 모델은 서비스 품질을 정적 QoS와 동적 QoS로 구분하고 이를 관리하는 각각의 객체 모듈에 의해 2단계 QoS 보장을 위한 제어 구조를 갖는다. 먼저 정적 QoS의 경우에는 설정 후 변동이 없는 파라미터로 정의하고, 동적 QoS인 경우에는 클라이언트와 서버간의 변화되는 파라미터로 정의하여 2단계에 걸쳐 멀티미디어 서비스를 제공하도록 한다. 이를 위해 이질적인 플랫폼 및 상호운영성 문제를 해결하기 위해 OMG의 CORBA 기반의 QoS 관리 모델을 채택하고, 정적 QoS는 중앙 집중형의 브로커 형태를 따라 클라이언트/서버 시스템의 사양과 클라이언트 단말 시스템 사양에 대한 제약 조건을 포함한다. 또한 동적 QoS는 CME가 제시한 매핑 기준을 채택한 클라이언트/서버 QoS 제어 기법에 필요한 해상도, 프레임율, 비트율로 정의하여 협약 및 재협약을 통해 제어한다. 다음 (그림 3)은 QoS 관리 모델에 대한 개념을 나타낸다.

본 논문에서는 분산 멀티미디어 서비스를 위한 QoS 관리 및 제어를 위하여 QoS 파라미터를 각 계층별 QoS를 파라미터 특성에 따라 설정 후 변동이 없는 QoS 파라미터를 정적 QoS 파라미터로, 클라이언트와 서버간의 환경 변화에 따라 동적으로 변화하는 QoS 파라미터를 동적 QoS 파라미터로 재정의 하였다. 본 논문에서 정의한 QoS 파라미터는 다음과 같다.

▶ 정적 QoS 파라미터

클라이언트/서버 장치 타입, CPU, 메모리, 버퍼 크기,

<표 1> QoS 관리 모델 비교

QoS Model	QoS Provision					QoS Control				QoS Management	
	QoS Mapping	Adm Control/ Resource allocation	E2Ea Coordination	Flow Scheduling	Flow Shaping	Flow Control	QoS Filtering	Flow Synchronization	Monitoring/ Alerts	QoS Maintenance	
XRM	EN	EN	(E)N	(E)N	-	N	-	-	N	-	
QoS-A	EN	E(N)	EN	E(N)	E	(E)	(E)N	E	EAD	ENRS	
Heidelberg	(E)(N)	EN	EN	E(N)	(E)	(N)	N	-	ED	ERS	
TINA	(E)	(N)	N	-	-	-	-	(N)	(N)	-	
IETF	EN	-	E	-	-	-	-	-	EN	ENR	
Telnet	E(N)	E(N)	E	E	-	-	-	E	E	E	
MASI	E(N)	E(N)	E	E	-	-	-	E	E	E	
OMEGA	E(N)	E(N)	E(N)	E(N)	E	E	-	-	E	ER	
OMG	EN	EN	EN	E	-	E	E	E	E	ER	
QuO	EN	EN	EN	E	-	E	-	E	E	ER	
CME	EN	EN	EN	E	-	E	-	E	E	ER	
DMSS	EN	EN	EN	E	-	E	-	E	E	ER	

- 다루지 않음

E/N 종점 시스템/네트워크에서 세부적으로 다룸

(E)/(N) 종점 시스템/네트워크에서만 적용

R 세부적인 QoS 재협약 처리

(R) QoS 재협약만 처리

S 세부적인 QoS 스켈링 처리

D 세부적인 QoS Degradation 처리

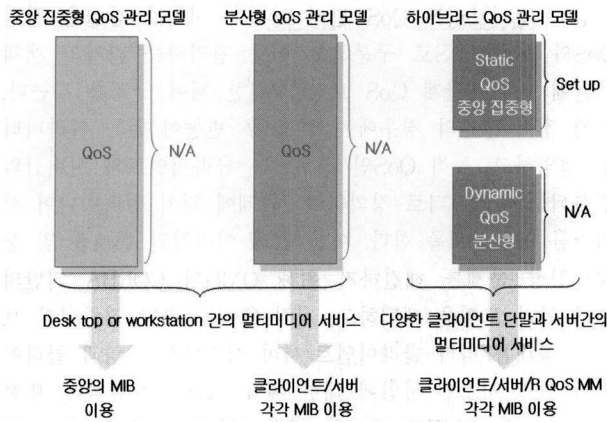
(D) QoS Degradation만 적용

A 세부적인 QoS 이용가능

스크린 크기 등

▶ 동적 QoS 파라미터

QoS 레벨, 프레임 율, 해상도, 비트 율, 패킷 크기 등

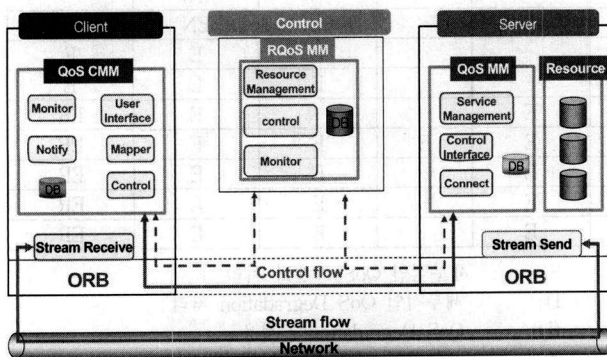


(그림 3) 하이브리드 개념

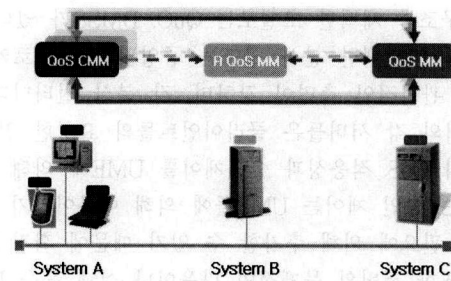
3.2 시스템 구조

본 논문에서 제안하고 있는 QoS 관리 모델은 중앙 집중형 QoS 관리 모델과 분산형 QoS 관리 모델을 혼합한 하이브리드 QoS 관리 모델을 지원하기 위해 세 개의 객체 모듈로 구성하였다. 다음 (그림 4)는 제안하고 있는 하이브리드 QoS 관리 모델을 적용한 전체 시스템을 보이고 있다.

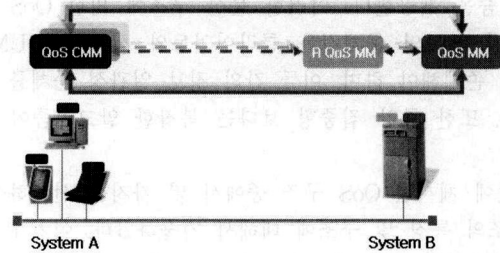
각 객체 모듈은 클라이언트 측의 QoS 제어 관리 모듈(QoS CMM)과 서버 측의 QoS 관리 모듈(QoS MM), 자원 QoS 관리 모듈(RQoS MM)로 구성된다. 각각의 모듈 내부는 중단간 스트림 서비스의 QoS를 저장하기 위한 객체들로 구성되어 있다. 이들 간의 스트림 흐름은 IP 기반에서 RTP 패킷을 송수신할 수 있는 스트림 송수신 객체에서 처리하고, 흐름제어는 CORBA 기반의 ORB를 통하여 이루어진다. 각 구성요소들은 논리적으로 세 개의 객체모듈로 구성되며, 이들은 물리적으로 두 시스템 또는 세 시스템에 배치가 가능하다. 또한 클라이언트와 서버 측 시스템 상에 QoS 제어 관리 모듈, QoS 관리 모듈이 각각 위치하며, 자원 QoS 관리 모듈은 서버 측에 위치한 형태 두 가지 형태의 물리적인 배치가 가능하다. 각 모듈들의 배치 구성은 다음 (그림 5)와 같다.



(그림 4) 제안한 QoS 관리 모델을 적용한 전체 시스템 구조



(a) 3-tier 구성요소 배치 환경



(b) 2-tier 구성요소 배치 환경

(그림 5) 각 구성요소의 물리적인 배치 환경

3.3 전체 처리 과정

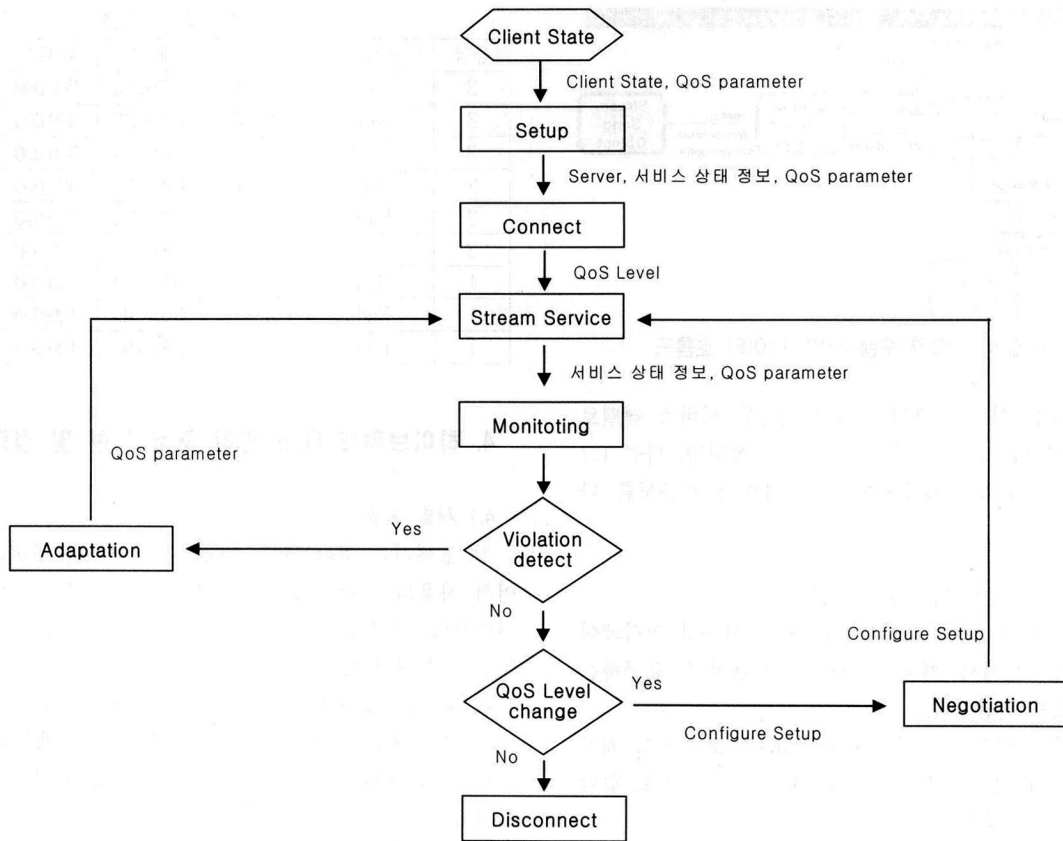
본 논문에서 제안한 하이브리드 QoS 모델의 각 객체 모듈에 대한 물리적인 배치 환경은 (그림 5)에서 (b)의 환경으로 구현하고자 한다. 이러한 환경에서 멀티미디어 스트림 서비스를 제공하기 위한 제안한 모델의 개괄적인 처리과정은 다음 (그림 6)과 같다.

QoS 관리 모듈의 통보 객체(Notify Object)가 자원 QoS 관리 모듈의 제어 객체(Control Object)에 클라이언트 상태 정보를 전송하여 클라이언트의 상태에 따른 서비스 제공과 클라이언트가 요청한 멀티미디어 데이터를 갖는 해당 서버를 선정하고 설정한다. 클라이언트의 서비스 요청에 따라 중단간 서비스 시작을 위한 접속 절차를 수행한 후 멀티미디어 스트림 서비스가 수행하며, 주기적으로 사용자가 요구하는 서비스 레벨과 협의를 위한 QoS 파라미터 정보를 모니터링한다. 모니터링을 통해 협약된 QoS 파라미터 위반시 동적 적응을 수행하며, QoS 레벨 변동시 실시간 협약을 수행한다.

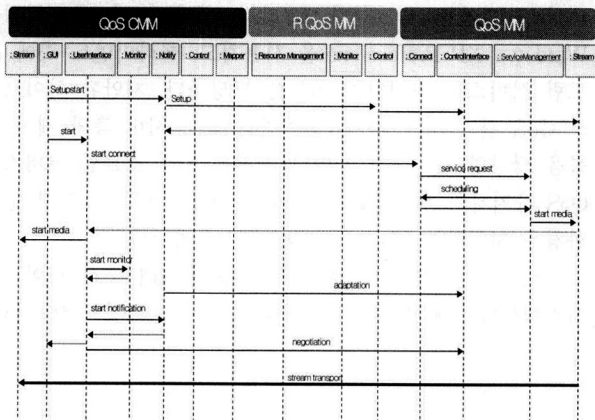
3.4 동적 적응 및 실시간 협약 과정

하이브리드 QoS 관리 모듈의 수행절차는 서비스 접속, 접속 해제, 동적 적응, 실시간 협약, 자원 모니터링 수행절차로 나뉜다. 이는 클라이언트와 서버간의 상호작용에 중점을 두어 이를 위한 표기법으로 UML의 시퀀스 다이어그램 (Sequence Diagram)을 이용해서 나타낸다. 전반적인 수행절차를 ETD(Event Tracing Diagram)으로 표현하면 다음 (그림 7)과 같다.

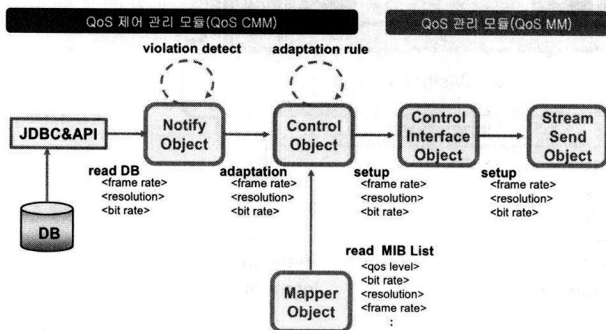
(그림 7)을 기초로 세부적인 동적 적응은 요청된 서비스 레벨의 스트림 서비스 과정에서 발생하는 예외 상황이나 협약된 레벨의 유지가 어려울 때 QoS 레벨 등급을 낮추거나



(그림 6) 제안한 하이브리드 QoS 관리 모델의 개괄적인 처리과정



(그림 7) 하이브리드 QoS 관리 모델의 전반적인 수행절차



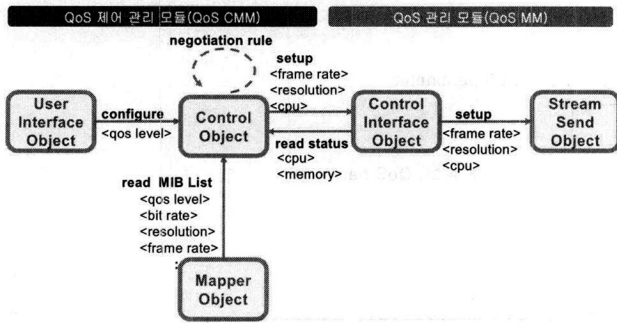
(그림 8) 동적 적응 수행절차의 데이터 흐름도

높여서 서비스가 중단 없이 지속되기 위한 목적으로 이용된다. 다음 (그림 8)은 동적 적응 수행절차에 대한 데이터 흐름도를 보인다.

동적 적응 수행절차는 다음과 같다.

- 클라이언트 측 QoS 제어 관리 모듈의 통보 객체가 QoS 협약 위반사항을 모니터 객체를 통해 관리되는 정보인 QoS 정보를 감시한다.
- 통보 객체가 QoS 정보를 감시하다가 협약위반 사항이 검출되면 제어 객체에게 적응 알고리즘을 수행하도록 요청한다.
- 요청을 받은 제어 객체는 현재의 서비스 상태 정보를 참조하여 현재 서비스 수신이 가능한 QoS 파라미터 정보를 제어 인터페이스 객체로 넘긴다.
- 제어 인터페이스 객체는 전달받은 QoS 파라미터 정보를 기준으로 QoS 레벨을 재조정된 후, 서비스 관리 객체에게 조정된 QoS 레벨로 스트림 서비스를 송신하도록 요청한다.
- 요청받은 서비스 관리 객체가 스트림 송신 객체를 재구성하여 적응 절차가 이루어진다.

실시간 협약은 사용자의 요구 메시지를 전달하는 사용자 인터페이스 객체와 협약알고리즘을 수행하는 제어 객체, 협약에 사용되는 관리 정보를 제공하는 매퍼 객체, 제공된 QoS 파라미터 정보에 의해 서비스 레벨을 결정하는 제어 인터페이스 객체, 재구성된 서비스 레벨에 의해서 서비스



(그림 9) 실시간 협약 수행절차의 데이터 흐름도

제공을 요청하는 서비스 관리 객체와 협약된 서비스 레벨로 서비스를 송신하는 스트림 송신 객체로 구성된다. 다음 (그림 9)는 실시간 협약 수행절차에 대한 데이터 흐름도를 나타내고 있다.

실시간 협약 수행절차는 다음과 같다.

- 초기 클라이언트 측의 서비스 사용자가 사용자 인터페이스 객체를 통해서 제어 객체에 사용자 요구하는 QoS 정보를 전달한다.
- 제어 객체는 매퍼 객체에 QoS MIB를 요청하고 제공받는 MIB와 사용자가 요구한 QoS 정보를 가지고 협약 알고리즘을 수행한다.
- 협약이 확정되면 제어 인터페이스 객체에 협약된 QoS 파라미터 정보를 전달한다.
- 제어 인터페이스 객체는 협약된 QoS 파라미터 정보를 바탕으로 원하는 QoS 레벨을 재구성한다.
- 서비스 관리 객체에 재구성된 QoS 레벨에 맞게 스트림 서비스를 송신 요청을 한다.
- 서비스 관리 객체는 재구성된 QoS 레벨에 맞게 스트림 전송 포맷을 재설정함으로써 두 모듈간 협약이 이루어진다.

다음 <표 2>는 사용자 레벨에서 정의한 등급에서부터 시스템 레벨의 QoS에 대한 매핑 테이블을 나타낸다.

<표 2> QoS MIB 테이블

등급	서비스 품질	프레임율	해상도	비트율	CPU
3	Good	25-30	620×320	700,000	1.0
3	Good	25-30	320×240	400,000	1.0
3	Good	25-30	200×150	300,000	1.0
2	Fair	6-14	620×320	400,000	0.6
2	Fair	6-14	320×240	350,000	0.6
2	Fair	6-14	200×150	200,000	0.6
1	Bad	1-2	620×320	100,000	0.2
1	Bad	1-2	320×240	100,000	0.2
1	Bad	1-2	200×150	100,000	0.2

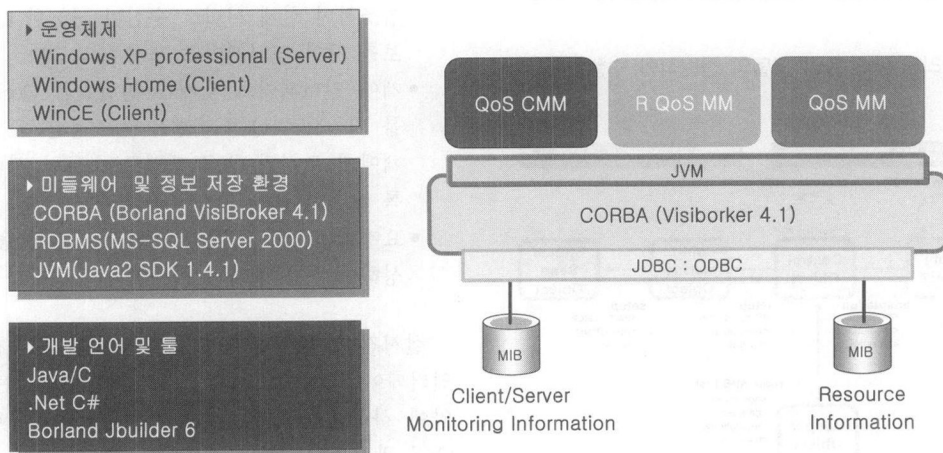
4. 하이브리드 QoS 관리 구조 구현 및 검증

4.1 개발 환경

본 절에서는 하이브리드 QoS 관리 구조를 구현하는데 있어서 사용되는 각각의 요소들에 대한 사항들을 세부적으로 설명한다. 구현된 시스템은 클라이언트 측에 QoS 제어 모듈을 등록하고, 서버 측에는 QoS 관리 모듈, 자원 QoS 관리 모듈을 등록하여 클라이언트와 서버 간의 스트림 송신이 가능하도록 구현하였다. 논 논문에서 제안하고 있는 하이브리드 QoS 관리 모듈의 프로토타입 환경은 다음 (그림 10)과 같다.

본 논문에서 제안하고 있는 하이브리드 QoS 관리 모듈의 구현에 사용된 시스템은 클라이언트 측에 데스크탑 PC, 노트북, PDA를 사용하고, 스트림 수신과 그래픽 사용자 인터페이스 제어를 담당한다. 그리고 서버 측에 운영체제는 Windows XP Professional을 사용하고, 클라이언트 측에 스트림 서비스를 송신하는 기능을 담당한다. 제안된 하이브리드 QoS 관리 모듈에서 클라이언트 측과 서버 측의 제어 흐름을 담당하고 종단간 제어 메시지를 통해 스트림 서비스의 QoS 유지하기 위해 사용된 ORB는 자바 애플리케이션 개발 환경을 지원하는 Visiborker 4.1을 사용하였다.

스트림 객체의 개발을 위해 사용된 API는 선 마이크로 시스템사의 스트림 서비스 개발을 위한 프레임워크인 JMF



(그림 10) 개발 환경

API 2.0을 사용해 구현하였다. JMF는 자바의 장점을 그대로 유지하면서 다양한 미디어 데이터 타입을 자바 애플리케이션이나 애플릿에서 일관되게 지원하기 위한 자바 클래스로써 인터넷 멀티미디어 스트림을 위한 스트리밍을 위한 표준 패킷 포맷인 RTP를 이용한 스트리밍 기능을 기본적으로 제공하고 있다. 따라서 네트워크를 통해서 멀티미디어를 재생하는 웹 기반 애플리케이션을 보다 폭 넓게 지원하고, JMF로 개발되지 않은 다양한 기기 또는 응용 프로그램과 멀티미디어 정보를 교환할 수 있도록 하였다. 그래픽 사용자 인터페이스 구현에 사용된 언어는 자바의 경량 컴포넌트인 JFC/Swing을 이용하였고, 개발의 주된 구현언어는 자바 애플리케이션 개발도구인 JDK 1.4.1을 사용하여 구현하였다.

4.2 사용자 인터페이스

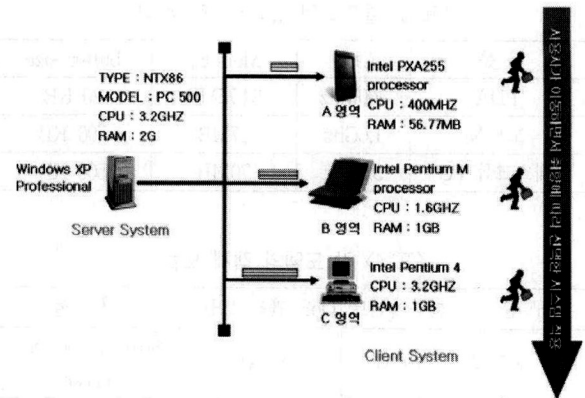
논문에서 제안한 하이브리드 QoS 관리 모델을 검증하기 위하여 사용자의 위치와 가용한 클라이언트 단말 시스템에 따라 적합한 서비스 품질을 제공하는 지 여부를 확인하기 위해 구현한 애플리케이션의 사용자 인터페이스에 대해 기술하였다. 먼저 사용자가 사용자의 위치에 따라 다양한 시스템을 이용하여 멀티미디어 서비스를 끊김 없이 제공하는지와 해당 시스템에 적합한 멀티미디어 데이터를 제공하는지 여부를 확인하고자 한다. 이를 위해 사용자 인터페이스의 구성은 홈 환경에서 멀티미디어 서비스를 제공하기 위한 환경으로 실내 영역을 거실(Living Room), 침실(Bed Room), 부엌(Kitchen)으로 구분하였다. 그리고 각 영역 내에는 서로 다른 클라이언트 단말을 데스크톱 PC, 노트북 그리고 PDA를 배치하였다. 그리고 사용자의 취향에 따라 단말을 선택하여 멀티미디어 스트림 서비스 제공함을 각각의 클라이언트 GUI를 통해 보였다.

(그림 11)은 침실 영역에 위치한 클라이언트 단말인 PDA를 사용자가 선택하여 서비스를 제공받는 결과 화면과 이와 관련된 멀티미디어 데이터의 정보 그리고 현재 제공되는 멀티미디어 스트림 서비스의 품질에 대한 정보를 나타낸다.

그 외에 데스크톱 PC나 노트북에서 멀티미디어 서비스를 받는 각각의 클라이언트 측 GUI 결과를 보이고 있다.

4.3 수행 평가 환경

수행 평가 시스템 환경은 서버는 데스크톱 PC CPU 3.2Ghz인 시스템과 클라이언트 단말은 3개의 영역별 PDA, 노트북, 데스크톱 PC로 구분하였다. 그리고 동일한 콘텐츠를 사용자의 취향에 따라 위치 이동하면서 서비스 받는 수행 과정을 중앙 집중형 모델과 분산형 모델 그리고 본 논문에서 제안한 하이브리드 모델을 비교하고자 한다. 다음 (그림 12)는 수행 평가 시스템 환경을 나타내고 있다.



(그림 12) 수행 평가 시스템 환경

클라이언트는 멀티미디어 서비스를 받기 위해 특정 영역 내의 단말의 선택을 통해 서비스를 제공 받는다.

- ▶ 자원 타입 선정 시간(St)
- ▶ QoS 적응 처리 시간(Qt)
- ▶ 서비스 처리 시간(Rt)
- ▶ 전체 서비스 시간(Tst) = $ST + Qt + Rt$

각 수행시간을 중앙 집중형과 분산형 그리고 하이브리드



클라이언트 단말의 가용성에 맞춘 멀티미디어 서비스 제공

현재 QoS 정보

(그림 11) 클라이언트 단말 형태에 따른 서비스 제공화면

관리 모델에 적용하여 실험하였다. 실험에 사용한 스트림 데이터는 동일한 콘텐츠를 해상도 620×320, 200×150의 두 가지 크기를 지원하는 1.687KB, 780KB의 사이즈를 갖는 2개의 동영상 파일로 테스트하였으며, 스트림 전송 속도는 최대 30 프레임/초의 속도로 송수신하여 수행 평가하였다. <표 3>은 클라이언트가 멀티미디어 서비스를 제공받기 위해 활용 가능한 장치로 PDA, 노트북, 데스크톱 PC로 선정하였으며 이에 대한 사양을 나타낸다.

멀티미디어 서비스를 위한 QoS 관리 모델인 중앙 집중형과 분산형 그리고 하이브리드 모델의 클라이언트와 서버 상의 객체 모듈을 구분하면 다음 <표 4>와 같다.

<표 3> 클라이언트별 시스템 환경

구 분	CPU	Memory	buffer size
PDA	400Mhz	34.75MB	20 KB
노트북	1.6Ghz	57MB	500 KB
데스크톱 PC	3.2Ghz	70MB	500 KB

<표 4> 각 모델의 객체 모듈

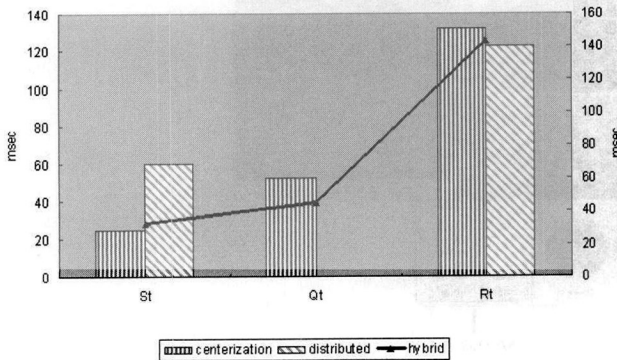
구 분	클라이언트	QoS 관리 모듈	서버 모듈
중앙 집중형	UCM	QoSM	Stream Service Server
분산형	QoS CMM	-	QoS MM
하이브리드형	QoS CMM	RQoS MM	QoS MM

4.4 수행 평가 결과

4.4.1 각 객체 모듈의 처리 시간

멀티미디어 서비스를 제공하기 위해 각 모델의 객체 모듈의 처리 시간인 클라이언트의 상황과 자원의 타입을 선정 시간과 설정된 QoS 레벨에 의해 승인 제어와 QoS 적응 처리 시간 그리고 서버에서 해당되는 스트림 데이터를 전송하여 클라이언트가 이에 해당하는 데이터를 받는 시간은 실험 결과는 (그림 13)과 같다.

QoS 관리 모델별 각 객체 모듈별 처리 시간은 본 논문에서 제시한 방법과 기존의 방법이 최소 34msec에서 최대 65msec로 별 차이가 없음을 확인하였다. 본 논문에서 제안



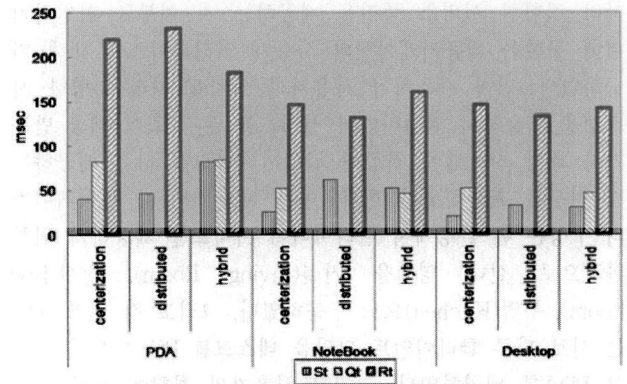
(그림 13) 각 QoS 관리 모델별 객체 모듈 처리 시간

한 하이브리드 모델의 경우 클라이언트와 서버 그리고 자원 QoS 관리 모듈이 참조해야하는 정적 QoS정보가 있어 기존의 모델보다 각 객체 모듈 별 처리 비용이 더 소요된다.

4.4.2 자원별 처리 시간

서로 다른 클라이언트 단말의 형태인 데스크탑 PC, 노트북, PDA에 해상도 620×320이며, 파일 크기가 1.687KB인 동일한 스트림 데이터를 전송하여 각 객체 모듈별 처리 시간인 St, Qt, Rt의 각 처리 결과를 측정하였다. 그 결과는 다음 (그림 14)에서 보인다.

(그림 14)에서 나타난 바와 같이 클라이언트 단말 형태에 따라 각 구성요소의 처리 시간은 PDA의 경우 처리 능력 때문에 노트북이나 데스크톱 PC 보다 패킷 사이즈를 줄이는 처리 과정이 많이 소비됨으로 인하여 제안한 모델은 중앙 집중형이나 분산형에 비해 90msec에서 100msec 더 많은 시간을 소비함을 보이고 있다.

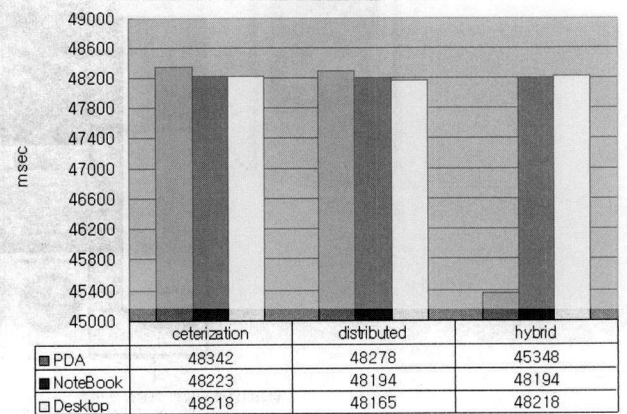


(그림 14) 자원별 각 모델의 처리 시간

4.4.3 전체 서비스 처리 시간

전체 서비스 처리 시간은 앞서 실험한 결과인 각 객체 모듈의 처리 시간과 클라이언트 단말 형태에 따라 적합한 스트림 데이터를 받아 클라이언트 단말에 디스플레이 서비스 종료까지의 시간의 합으로 이에 대한 각 QoS 관리 모델별 처리 결과는 다음 (그림 15)와 같다.

전체 서비스 처리 시간은 스트림 데이터를 선정하고 서버에 바인딩하여 스트림 데이터를 클라이언트 단말 시스템



(그림 15) 전체 서비스 처리 시간

에 디스플레이하여 종료까지의 시간으로 각 모델의 방식에 따라 약간의 차이가 있지만 본 논문에서 제안한 클라이언트 단말의 형태에 따라 선정된 자원에 따라 멀티미디어 서비스를 제공함에 있어서 PDA의 경우 타 단말에 비해 약 2800msec가 적게 소요됨을 알 수 있었다.

본 실험 결과, 동일한 멀티미디어 서비스 제공을 위한 객체 모듈의 처리 시간은 기존 중앙 집중형 QoS 관리 모델과 분산형 QoS 관리 모델과 비교하였을 경우, 제안한 모델이 이들 모델 보다 참조해야할 QoS 정보와 클라이언트 단말 시스템에 가용한 스트림 데이터로 처리해야할 정보가 많으므로 처리 비용이 더 많이 들을 수 있었다. 그리고 동일한 멀티미디어 데이터를 클라이언트의 단말 형태를 고려하지 않고 각 객체 모듈의 처리시간을 비교한 결과 분산형 QoS 관리 모델이 가장 성능이 우수하였으며, 중앙 집중형 그리고 제안한 모델 순으로 평가되었다. 그러나 클라이언트 단말 시스템 형태에 따른 QoS 관리 모델간 전체 서비스 처리 시간의 결과에서 나타난 바와 같이, 클라이언트 단말 시스템 형태와 가용한 콘텐츠를 매핑하고, 서버로부터 멀티미디어 서비스 종료까지의 시간을 비교한 결과 제안한 하이브리드 QoS 관리 모델이 기존 모델보다 우수함을 알 수 있었다.

6. 결론

최근 유·무선 환경 인프라를 기반으로 유비쿼터스 컴퓨팅 환경으로 변화되고 있는 시점에서 보다 다양해진 클라이언트 단말 시스템의 변화와 사용자의 요구사항은 더욱 다양해지고 있다. 즉, 특정 사용자 한사람이 다루는 클라이언트 디바이스가 데스크탑 PC, 노트북, PDA 그리고 모바일 단말기에 이르기까지 사용자 취향에 따라 멀티미디어 서비스를 제공받는 환경으로 변화되었다. 이에 따라 콘텐츠 또한 이에 적합한 형태로 재구축하고 있는 실정이다. 따라서 본 연구에서는 기존 QoS 파라미터와 클라이언트 시스템과 자원을 지원하기 위한 QoS를 재분류하고, 이를 기초로 기존 QoS 관리 모델인 중앙 집중형과 분산형을 혼합한 하이브리드 QoS 관리 모델을 제시하였다. 제안한 시스템 구성은 분산된 형태로 종단간 QoS 제어 및 관리할 수 있도록 클라이언트 측에는 QoS 제어 관리를 위한 모듈(QoS CMM), 서버 측에 QoS 관리 모듈(QoS MM), 그리고 중앙형 QoS 제어 및 관리를 위한 자원 QoS 관리 모듈(RQoS MM)을 구성하였다. 정적 QoS 파라미터는 자원 QoS 관리 모듈에서 처리하며, 전체적인 QoS 제어는 클라이언트 측에서 전담하여 다자간 통신의 경우에 서버 측의 부담을 최소화 시킬 수 있도록 하였다. 이를 검증하기 위해 위치 기반 멀티미디어 서비스에 적용하여 사용자의 취향에 따라 단말을 선택하여 멀티미디어 서비스 제공을 보였다. 또한 이를 기반으로 기존의 중앙 집중형과 분산형 그리고 제안한 모델을 비교 분석하였다. 분석 결과 동일한 멀티미디어 서비스 제공을 위한 객체 모듈의 처리 시간은 타 모델에 비해 참조해야할 정보가 많기 때문에 소요 비용이 크지만 클라이언트 단말 시스템 형태에

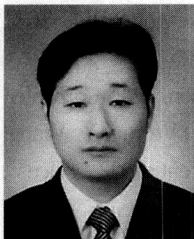
따른 QoS 관리 모델간 전체 서비스 처리 시간으로 비교하였을 경우에는 적합한 콘텐츠를 제공함으로써 적게 소요됨을 수행 평가를 통해 검증하였다.

향후 연구는 본 논문에서 제시한 하이브리드 모델에서 사용한 테이블 기반 QoS 매핑 방법을 개선하여 동적인 QoS를 처리하기 위한 새로운 매핑 메커니즘에 대한 연구와 다양한 응용 서비스에 적용하여 검증하고자 한다.

참고 문헌

- [1] H.Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP : A Transport protocol for realtime application", RFC-1889, Feb., 1996.
- [2] H. Schulzrinne, "RTP Profile for Audio and Video Conference with Minimal Control", RFC-1890, May, 1996.
- [3] H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol(RTSP)", RFC-2326, 1998.
- [4] Braden. R, Zhang. L, Berson. S, Herzog. S, Jamin. S, "Resource Reservation Protocol(RSVP)-Version 1 Functional Specification", RFC-2205, 1997.
- [5] R. Barden, D. Clark, and S. Shenker, "Intergrated Service in the internet Architecture : An Overview" RFC-1633, June, 1994.
- [6] Y. Bernet, J Binder, S. Blake, M. Carson, B, E. Carpenter, S. Keshav, E. Davies, B. Ohlman, D.Verma, Z. Wang, W. Weiss, "A Framework for Differentiated Services", Internet Draft, Feb., 1999.
- [7] Eric C. Rosen, "Multiprotocol Label Switching Architecture", Internet Draft, Feb., 1999.
- [8] Carsten Vogt, Lars C. Wolf, Ralf Guido Herrtwich, Hartmut Witting, "HeIRAT Quality-of-Service management for distributed multimedia systems", IBM European Networking Center, 1998.
- [9] 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.
- [10] Marco Alfano, Rolf Sigle, "Controlling QoS in a Collaborative Multimedia Environment", Proceedings of HPDC '96, pp.340-347, 1996.
- [11] OMG, "Control and Management of A/V streams specification", OMG Document telecom/98-10-05, October, 1998.
- [12] Tarek F. Abdelzاهر, Kang G. Shi, Nina Bhatti, "User-Level QoS-Adaptive Resource Management in Server End-Systems, IEEE Transactions on Computers, Vol.52, No.5, pp.678-685, May, 2003.
- [13] Kentarous Fukuda, Naoki Wakamiya, Masayuki Murata

- and Hideo Miyahara, "QoS guarantees based on end-to-end resource reservation for real-time video communications," Proceedings of 16th International Teletraffic Congress, pp.857-866, June, 1999.
- [14] El-Khatib K, Bochmann G, ZhongY, "Selecting the QoS parameters for multicast applications based on user profile and device capability," Lecture Notes in Computer Science, Vol.2158, pp.221-232, 2001.
- [15] 조성호, 노선식, 이형욱, "ATM-MPLS 기반 초고속국가망 서비스의 서비스품질 강화 방안", 정보화정책 제12권 제 4호, pp.23-41, 2005년 겨울.
- [16] Lei Chen, Shahadat Khan, Kin F. Li, Eric G. Manning, "Building an Adaptive Multimedia System using the Utility Model," Parallel & Distributed Processing, Lecture Notes in Computer Science 1586, pp.289-298, April, 1999.
- [17] Mostofa Akbar, Eric G Manning, Gholamali Shoja, "Distributed Utility Model for Distributed Multimedia Server System," Proceedings of ICCIT 02, Dhaka, Bangladesh, December, 2002.



정창원

e-mail : mediblue@wku.ac.kr

1993년 원광대학교 컴퓨터공학과(학사)
 1998년 원광대학교 컴퓨터공학과(석사)
 2003년 원광대학교 컴퓨터공학과
 (공학박사)

2004년~2006년 전북대학교 학술연구교수

2006~현 재 원광대학교 전기전자및정보공학부 박사후 연구원
 관심분야: 분산객체 컴퓨팅, 멀티미디어 서비스, LBS, 텔레매틱스



이건엽

e-mail : gylee@kunjang.ac.kr

1995년 군산대학교 정보통신공학과(학사)
 1997년 군산대학교 정보통신공학과(석사)
 2007년 원광대학교 컴퓨터공학과(공학박사)
 1997년 9월~현 재 군장대학 자동차디지탈
 공학부 부교수

관심분야: 분산객체 컴퓨팅, 멀티미디어 스트림 서비스, 텔레매틱스



주수중

e-mail : scjoo@wku.ac.kr

1986년 원광대학교 전자계산공학과(학사)
 1988년 중앙대학교 컴퓨터공학과(공학석사)
 1992년 중앙대학교 컴퓨터공학과(공학박사)
 1993년 미국 University of Massachusetts
 at Amherst, Post-Doc.

2003년 미국 University of California at Irvine, Visiting Professor

1990년~현 재 원광대학교 전기전자 및 정보공학부 교수

2007년~현 재 원광대학교 정보전산원 원장

관심분야: 분산 실시간 컴퓨팅, 분산객체모델, 시스템 최적화,
 멀티미디어 데이터베이스