

RIOP를 이용한 실시간 CORBA 상에서의 단순화된 자원예약 메커니즘을 이용한 실시간 화상회의 시스템의 설계 및 구현

현 호 재[†] · 홍 성 준^{††} · 한 선 영^{†††}

요 약

인터넷상에서 화상회의, VOD 등과 같은 멀티미디어 서비스가 개발되어왔다. 이런 멀티미디어 서비스의 비디오나 오디오 멀티미디어 데이터는 QoS 보장을 통한 실시간성을 요구한다. QoS 지원을 위해서 인터넷 상에서는 자원 예약을 하기 위해서 RSVP(Resource reSerVation Protocol)가 제안되었다. 그러나 RSVP는 복잡성과 확장성의 문제를 야기하고 있다. 본 논문은 이런 RSVP의 복잡성과 확장성의 문제를 해결하기 위하여 단순화된 자원예약 프로토콜을 개발하고 단순화된 자원예약 프로토콜 상에서 QoS 보장을 지원하는 화상회의 시스템의 설계 및 구현에 대해서 언급한다.

Design and Implementation of Real-Time Teleconferencing System using the Simplified Resource Reservation on Real-Time CORBA Supporting RIOP

Ho-Jae Hyun[†] · Sung-Joon Hong^{††} · Sun-Young Han^{†††}

ABSTRACT

Multimedia services(i.e. teleconferencing and Video on demand) have been developed on Mbone. The video and audio data of them require Real-Time service using QoS(Quality of Service) guarantees. RSVP(Resource reSerVation Protocol) on the Internet has been suggested to support QoS guarantees. But currently, it has two problems: complexity and scalability. To solve these problems, this paper describes the design and implement of teleconferencing system with QoS guarantees by simplifying the resource reservation processing to solve the RSVP's complexity and scalability.

1. 서 론

기존의 컴퓨터 통신은 시간적인 제약을 필요로 하지 않는 일반적인 데이터를 취급하고 주로 데이터 전송의

정확성과 효율성에 중점을 두고 발전해 왔다. 그러나 비디오나 사운드 같은 연속적인 미디어가 컴퓨터 네트워크에 적용되기 시작하면서 미디어에 따른 시간적 특성의 보장을 요구하게 되었다. 현재 대부분의 컴퓨팅 환경은 그 통신수단으로 인터넷을 사용하고 있는데 컴퓨터 시스템의 급격한 성장에 힘입어 90년대를 지나면서 인터넷은 멀티미디어 정보를 수용할 수 있도록 변화를 요구받고 있다. 특히, 원격화상회의, 디지털 TV

* 본 논문은 1999년도 한국과학재단 연구비지원(과제번호 : 981-0017-088-2)에 의한 결과임.

† 정 회 원 : 건국대학교 일반대학원 컴퓨터 정보통신공학과

†† 준 회 원 : 여주대학 정보통신과 교수

††† 정 회 원 : 건국대학교 컴퓨터 정보통신공학과 교수

논문접수 : 1998년 12월 17일, 심사완료 : 1999년 5월 31일

방송, 인터넷 방송, 대화형 멀티미디어 정보시스템 같은 실시간 멀티미디어 데이터는 기존의 네트워크 시스템에서 프로토콜의 한계에 직면하게 되었다. 이러한 실시간 멀티 미디어 데이터를 지원하기 위해서 응용 소프트웨어가 가져야 할 것으로 가장 중요한 것은 QoS(Quality of Service)[1]인데 단말(end-system)간의 QoS 매개(parameter) 정의, 통신망에서의 QoS 보장 및 관리, 오퍼레이팅 시스템과 미들웨어로 부터의 QoS 보장 등의 기술적인 문제가 큰 이슈가 되고 있다. 또한 이러한 문제점들을 해결 하기 위한 방안들로 인터넷 상에서는 자원 예약 프로토콜인 RSVP(resource ReSerVation Protocol)[2,3]가 제안되었고, 분산 환경의 미들웨어 상에서는 실시간성을 지원하기 위하여 실시간 ORB 프로토콜인 RIOP(Real-Time Inter-ORB Protocol)[4]가 제안되었다. 최근 RSVP는 인터넷에 실제 적용하기에는 확장성(scalability)과 복잡성(complexity)의 문제가 있음이 밝혀지고 있다[5]. RSVP는 전화망에서 사용하고 있는 시그널링 프로토콜과 같이 outband 시그널링 방식을 사용하고 있다. 이러한 시그널링 방식이 인터넷과 같은 개방된 망에서는 단말과 라우터(router) 사이의 큰 메시지를 처리할 때 많은 오버헤드가 발생하는 복잡성과 백본 환경하에서 soft state를 유지 하기 위한 refresh 메시지에 의해 소모되는 대역폭과 라우터에서 많은 수의 흐름을 처리하기 위해 요구되는 저장 공간의 부족으로 확장성 문제를 야기하고 있다. 이러한 복잡성과 확장성 문제를 해결하기 위한 자원예약 메커니즘의 개발이 필수적이다. 따라서, 이러한 RSVP의 확장성과 복잡성의 문제를 해결하기 위한 연구가 활발히 진행되고 있다. RSVP의 확장성 문제를 주로 다룬 SRP(Scalable Resource Reservation Protocol)[5]는 시그널링 방식을 쓰지 않는 대체 방안을 제안함으로써 RSVP의 복잡성과 확장성의 문제를 해결하고자 하고 있다. 또한 YESSIR(YEt another Sender Session Internet Reservation)[6]는 inband 시그널링 방식을 사용한 단순화된 구조를 가지고 있고, RSVP의 자원예약 프로세싱의 오퍼레이션인 수신자중심이 아닌 송신자중심의 자원예약 방식을 택하고 있다. 현재 YESSIR는 working draft 상태에 있고, IBM 등에서 구현 중에 있다. 그러나 SRP는 기존 자원 예약 시그널링 방식을 부정한 새로운 방식을 채택하였기 때문에 기존 RSVP 등과의 호환성 여부가 불분명할 뿐아니라 아직 구현이 되지 않았기 때문에 QoS 보장성 여부 또

한 불분명하다. 그리고 YESSIR는 자원 예약의 단순성 문제를 해결하는 구조는 가지고 있으나 확장성 문제는 아직 해결하지 못하고 있는 실정이다.

한편, 분산 환경에서는 CORBA(Common Object Request Broker Architecture)[7]가 적용될 수 있는 컴퓨팅 표준이며, 시스템을 통합하는 하나의 미들웨어이다. 즉, 분산환경에서 시스템에 대해 독립적이라는 강점과 여러 환경의 시스템에서 일관된 하나의 표준으로서, 분산시스템들을 하나로 통합해 왔다. 하지만 기존 CORBA는 QoS 보장을 제공하기 위한 정책이나 메커니즘을 가지고 있지 않으며, 실시간 프로그래밍을 지원하는데 필요한 중요한 특징을 정의하고 있지 않기 때문에 멀티미디어 응용 서비스와 같은 실시간 처리에는 한계점을 나타내고 있다. 이러한 문제점의 해결 방안으로 워싱턴 대학의 분산 객체 컴퓨팅 그룹에서 Real-Time CORBA(TAO)[4]라 불리는 고성능, 실시간 ORB를 개발하였다

그러므로 본 논문에서는 첫째, RSVP의 복잡성 문제와 확장성 문제를 해결하기 위해서 본 논문에서 SRRP(Simple Resource Reservation Protocol)라고 명명한 inband 자원 예약 방식의 단순화된 자원예약 메커니즘의 구현과 분산 환경에서의 실시간 서비스를 고려한 RIOP와의 연동에 대한 설계와 SRRP/RSVP 연동 게이트웨이를 구현하고, 둘째, Mbone상에서 SRRP 기반의 QoS 보장성을 지원하는 실시간 화상회의 시스템을 구현하였다. 본 논문의 구성은 다음과 같다. 2장에서는 RSVP, SRP, YESSIR, Real-Time CORBA의 관련연구 사항을 소개한다. 3장에서는 단순화된 예약 메커니즘의 설계와 Mbone상의 QoS 보장을 고려한 화상회의 시스템의 설계를 언급하고, 4장에서는 구현사항을 언급한다. 5장에서는 개발 결과 및 평가를 언급하며, 마지막으로 6장에서 결론을 맺는다.

2. 관련연구

2.1 RSVP[2,3]

현재의 인터넷 프로토콜은 데이터그램 개념에 기반을 두고 있어서 패킷은 서로 독립적으로 라우팅되며 QoS의 정의도 존재하지 않는다. 성능은 보장되지 않으며, 단지 최선의 노력을 하는 것이다. 멀티 미디어 서비스에 있어 Best Effort Service는 적합하지 않으며 실시간으로 음성과 화상을 전송하는 데 있어 일정한

지연의 데이터 흐름을 보장받을 수 있어야 한다. 이런 새로운 전송 능력이 요구되면서 인터넷 연구 그룹 사이에서는 실시간 전송이 필요한 응용에서 일정 수준의 서비스를 보장하기 위해서 라우터에서 자원을 미리 예약하는 방법을 고안하게 되었는데 이것이 바로 자원 예약 프로토콜인 RSVP이다.

통합된 서비스의 인터넷을 위하여 설계된 자원 예약 프로토콜인 RSVP는 멀티 캐스트 혹은 유니 캐스트 데이터 흐름에 대하여 수신자가 자원 예약의 설정을 행하는데 이는 쉽게 확장될 수 있으며, 견고하다. RSVP 프로토콜은 네트워크로부터 일정한 QoS(Quality of Service)를 요청하기 위해서 호스트들에 의하여 사용된다. 또한 라우터(router)에서는 데이터 열의 경로에 있는 모든 노드들에게 제어 메시지를 전달하는 데에도, 요청된 서비스를 제공하기 위한 상태정보를 설정하고 유지하는 데에도 사용된다. RSVP는 일반적으로 데이터 경로를 따라 각 노드에 자원이 예약 되도록 한다. RSVP는 단일(simplex) 데이터에 대하여 자원(resource)을 요청한다. 즉, 자원 요청은 한 방향으로만 이루어진다. 그러므로 같은 응용처리가 동시에 송신자와 수신자에서 동작하더라도 RSVP는 송신자와 수신자를 구별해서 처리한다. RSVP 그 자체는 라우팅 프로토콜이 아니라 현재 혹은 미래의 유니 캐스트, 멀티 캐스트 라우팅 프로토콜에서 동작하도록 설계되었다.

2.2 SRP(Scalable Resource Reservation Protocol for Internet)[4]

현재 멀티미디어 네트워크 자원 예약 구조는 대량의 데이터 흐름(flow)을 처리하는데 한계가 있다. 이에 SRP는 각 링크의 집합적인 데이터 흐름들에 의한 대량의 병렬적인 흐름들을 처리하는 것에 확장성(scalability)을 가진다. 이러한 집합은 종단 시스템(end system)에 트래픽 통제(traffic control)를 위탁함으로써 가능해진다. 네트워크에서 각 링크의 흐름(flow)을 자동으로 모아주는 새로운 구조로 네트워크는 개인적 흐름(Individual flow)에 대한 아무 정보도 가지지 않는다. 특정 신호 프로토콜(signaling protocol)은 없고, 프로토콜 오버 헤드는 주로 2비트(bit)로 이루어진 세 가지 값(예약(reserved), 요청(request), 최선(best-effort))의 패킷 형태에 대한 채택으로 이루어진다.

2.3 YESSIR(Yet another Sender Session Internet Reservation)[6]

RSVP보다 간단한 코드와 실행 시 낮은 복잡도를 가

지는 RTP(Real-time Transport Protocol)[8]에 기반한 in-band 시그널링(signaling)으로 송신자 중심의 자원 예약 프로토콜인 YESSIR가 RSVP의 단점을 보완할 수 있다. YESSIR는 새로운 예약 메커니즘으로 RSVP에서 소개된 많은 독특한 특징들을 유지하는 동안 예약된 플로우들을 개설하는 처리를 간소화한다. 단순성(simplicity)은 control 메시지 처리, 데이터 패킷 처리, 사용자 수준의 융통성(user-level flexibility)를 바탕으로 평가된다. RTCP의 상에 있으면서, 자원예약 상태를 유지하기 위해 soft state를 사용한다. 공유 자원예약(shared reservation)을 지원하고, 관계된 플로우를 통합하고, IS model[9]과 호환된다. YESSIR의 경우 적절한 예약을 하기 위해 지역 자원 매니저를 호출할 필요가 있는 것을 제외하면 라우터에서 YESSIR 메시지 처리는 RSVP PATH와 유사하다. YESSIR는 RSVP와 마찬가지로 확장성 문제의 해결책을 필요로 하고 있다[9].

YESSIR는 자원예약을 위하여 RTP를 사용하게 되며, RTP의 특징을 보면 실시간 특징을 가진 데이터에 대한 종점간 배달 서비스를 제공하기 위해 디자인되었다. 응용 프로그램들은 프로토콜에 독립적이지만 멀티플렉싱과 체크섬 서비스를 이용하기 위해 UDP위에 RTP를 보낸다. RTP는 자원 예약 프로토콜은 아니지만, 자원 예약을 inband signaling과 주기적인 송신자와 수신자의 확인으로 할 수 있다.

inband signaling을 하기 위해서 RTP는 두 종류의 패킷을 사용한다. 데이터 전송을 위한 RTP와 제어를 위한 RTCP이다. 각 RTP세션은 하나의 RTP 데이터와 하나의 연관된 RTCP 스트림으로 구성된다. UDP위에 보내지면 데이터와 제어 패킷은 근접한 포트 번호를 사용한다. 그래서 방화벽이 쉽게 제어 스트림으로부터 연관된 데이터 스트림을 맞출 수 있다.

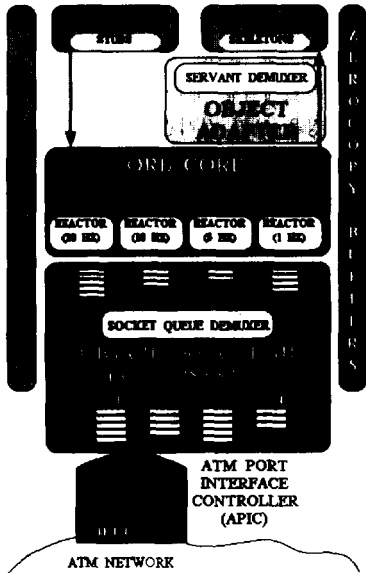
주기적인 송신자/수신자 확인을 위해서 송신자와 수신자들은 정기적으로 멀티캐스팅 그룹 report를 포함한 RTCP를 보낸다. 데이터 송신자들은 최후의 round trip 시간 정보와 report 이후 전송된 바이트와 패킷수를 가리키는 SR(Sender Report)를 보낸다. 데이터 receiver들은 패킷 손실과 지연 통계를 나타내는 RR(Receiver Report)를 보낸다. 이런 report를 통하여 모든 참가 멤버들은 흐름의 특징과 네트워크의 부하를 알 수 있으며, 라우터는 세션의 자원 예약을 알 수 있다.

라우터에서 RTCP같은 패킷을 잡기 위해서 IP Router Alert Option[10,11]을 이용하게 된다.

이 옵션을 이용하면 라우터는 통과되는 패킷의 내용을 세밀하게 검사할 수 있다. 즉, 라우터는 적은 성능의 영향을 주면서 직접 패킷들을 인터럽트 할 수 있다. YESSIR에서는 RTCP의 SR(Sender Report)를 잡기 위해 사용된다.

2.4 Real-Time CORBA(TAO)[4]

워싱턴 대학의 분산 객체 컴퓨팅 그룹은 Real-Time CORBA라 불리는 고성능, 실시간 ORB(The ACE ORB)를 개발하였다. Real-Time CORBA는 CORBA 미들웨어를 운영체제, 통신 프로토콜 그리고 네트워크 인터페이스를 통합함으로써 응용프로그램에 QoS 보장을 제공하도록 설계 되어있다.



(그림 1) Real-Time CORBA ORB 구조적 요소

(그림 1)과 같이 Real-Time CORBA의 ORB 종단 시스템(End System)은 망 인터페이스, I/O subsystem, 통신 프로토콜 그리고 CORBA 미들웨어 요소를 포함한다.

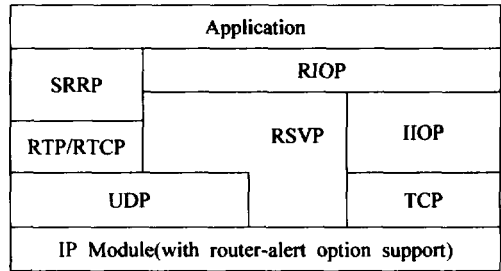
Real-Time CORBA의 프토토콜 엔진은 SunSoft IOP 참조구현(Reference Implementation)의 최적화된 실시간 버전이다. 클라이언트, 서버 상의 ORB 코어는 QoS 클래스 속성에 맞게 요구를 처리하는 사용된다. 이런 설계는 클라이언트 요구의 상대적인 우선순위를 가라키고 클라이언트의 QoS 요구를 수행할 수 있게

한다. 운영체제와 망 플랫폼사이에 호환성을 증가시키기 위해서 Real-Time CORBA의 프로토콜 엔진은 Real-Time CORBA의 ORB 코어에 분리된 계층으로 설계되어 있다.

Real-Time CORBA의 ORB는 Rate-Based Concurrency 구조, Rate-Based Connection 구조, 그리고 RIOP (Real-Time Inter-ORB Protocol)을 지원한다. Real-Time CORBA의 Rate-Based Concurrency 구조에서 Real-Time CORBA ORB 코어는 각 등급(rate) 그룹에 대해서 실시간 쓰레드(Thread)를 할당하도록 구성될 수 있다. 등급 그룹은 특정한 등급으로 발생하는 모든 주기적인 처리 오퍼레이션을 모아서 그것들을 고정된 우선순위 스케줄링을 사용해서 각 쓰레드에 할당한다.

3. 설 계

3.1 단순화된 자원 예약 메카니즘의 설계



(그림 2) 단순화된 자원 예약 프로토콜 스택

(그림 2)는 단순화된 자원예약 프로토콜의 스택을 보이고 있다. 본 논문에서는 단순화된 자원 예약 프로토콜을 SRRP(Simple Resource Reservation Protocol)라고 명칭하기로 한다. 본 논문의 중심이 되는 SRRP는 UDP/IP상의 RTP/RTCP위에서 동작을 한다. 반면에 RSVP는 UDP/IP위에 모두 위치 할 수 있고, raw모드의 RSVP인 경우는 IP위에서 직접 동작할 수 있다. 본 논문에서 설계된 SRRP는 RSVP의 복잡성 문제를 해결하기 위해서, IP 패킷의 내용을 좀더 자세히 라우터에서 볼 수 있도록 하는 IP Router Alert Option의 이용과 기존의 RTCP에서의 확장부분을 추가하였으며, 또한 확장성 문제를 해결하기 위해서 SRRP/ RSVP 연동 게이트 웨이를 설계하였다. 분산망 환경과의 연동을 위하여 연동 게이트 웨이에서는 RSVP 상위에서 데이

터 전송 이외에 자원 예약 정보를 가지는 QoS 정보의 전송 처리를 한다. 그리고 IIOP와 RSVP를 통합하여 공통된 프로토콜인 RIOP가 위치하게 된다.

3.1.1 IP Router-Alert Option

0	8	16	31
Type	Length	Value	

Type:

- Copied flag : 1 bit ; 1
- Option class: 1 bit ; 0 : 제어
- Option number: 6 bit ; 20

Length: 8 bit ; 4

Value: 16 bit

- 0 ; 라우터가 패킷을 검사한다.
- 1 ; RTCP 메시지를 가진 데이터그램이다.
- 2 - 65535 ; Reserved

(그림 3) Router Alert Option의 구조

(그림 3)은 설계된 IP Router Alert Option의 구조를 보이고 있다. 각 필드에는 타입, 길이, 값에 대한 정보가 들어가 있다. 이 옵션을 이용하여 라우터는 더 세밀히 통과되는 패킷의 내용을 검사하게 된다. 즉, 라우터는 적은 성능의 영향으로 직접 패킷들을 인터럽트할 수 있다. SRRP에서는 RTCP의 SR(Sender Report)을 잡기 위해 사용된다.

3.1.2 RTP 확장의 설계

3.1.2.1 예약 요청 : Sender Report

(그림 4)와 같이 RTCP SR(Sender Report) 메시지를 예약 요청을 위해 IP Router Alert Option과 자원 예약 정보를 넣어 확장한다.

예약 요청 메시지 중 RTCP Header, Sender Information, Report Block들은 기존의 RTP 사양에 정의된 내용과 동일하다.

0	31
IP Standard Header : Header Length = 6	
Router Alert Option	
Transport Protocol Header : RTCP는 port number로 홀수 사용	
RTCP (Sender Report) Header	
Sender Information	
Report blocks	
Resource reservation extension	

(그림 4) 확장된 RTCP Sender Report(SR) 메시지 구조

3.1.2.2 예약 모니터링 : Receiver Report

송신자에게 예약 결과를 넘겨주기 위해 사용된다. 자원예약 확장(Resource reservation extension) 정보는 강제적으로 필요하지 않다.

3.1.3 SRRP 메시지 포맷

(그림 5)와 같이 자원예약 확장 메시지는 다른 메시지와 구별하기 위하여 예약 확장이라는 타입과 다른 응용에서 사용할 수 있는 타입을 정의한 *common header*, 예약 스타일에 대한 정의, 예약 과정을 진행하면서 거처온 링크들에 대한 예약 상태 정보, 예약 실패에 따른 에러 플래그, 예약 명령, SR을 만들어 보내는데 대한 refresh 주기에 관한 정보를 담은 *Generic Fragment*와 예약을 하기 위하여 QoS 파라미터 정보를 담은 *Flowspec Fragment*와 몇 개의 선택적인 *fragment*들로 구성된다.

0	31
Common Header	
Generic Fragment(always present)	
Flowspec Fragment(always present)	
Network Monitoring Fragment(optional)	
Reservation Error Fragment(optional)	

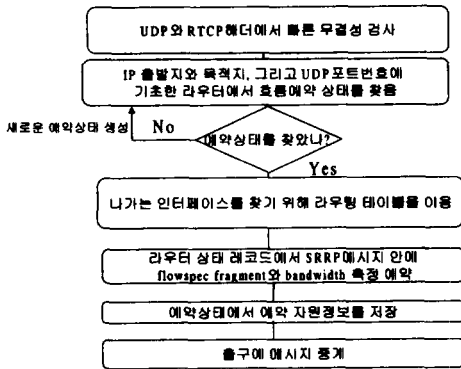
(그림 5) 자원예약 확장메시지 구조

3.2 단순화된 자원 예약 메커니즘을 위한 SRRP/RSVP 연동 게이트웨이 설계

연동 게이트웨이에서는 SRRP의 inband 시그널링과 RSVP의 outband 시그널링을 상호 연동 시킬 수 있는 기능이 필요하며, SRRP의 일반 fragment 구조 중 예약 명령(Reservation Command)이 RSVP 백본망으로 들어가는 게이트웨이에서는 예약을 하지 않고 지나가야 하며, 백본망을 나가는 지점의 게이트웨이에서 다시 예약을 진행할 수 있도록 해야 한다.

라우터에서 SRRP 메시지 처리는 RSVP의 PATH 메시지와 유사하다. (그림 6)은 라우터에서 SRRP 메시지를 처리하는 알고리즘을 보인다.

RSVP는 작은 흐름(flow)들을 모으는 능력으로 인해 확장성 문제가 발생한다. IP 백본상에서 모든 흐름이 RSVP를 통한 연결을 가진다면, 한 라우터는 각 링크마다 수 천개의 흐름들을 처리 해야 한다. RSVP의 확장성 문제를 해결하기 위해서 백본(Backbone)은 RSVP



(그림 6) 라우터에서의 SRRP의 처리알고리즘

를 이용하고, Regional Network는 SRRP를 이용하여 기존 RSVP의 시그널(signal) 때문에 발생하는 확장성의 문제를 해결하는 구조를 가진다. inband 시그널방식을 outband 시그널로 변환하기 위한 게이트웨이를 개발하여 기존 RSVP 지원 라우터에서는 outband 시그널방식으로 대역폭이 큰 흐름을 관리하고 대역폭이 작은 흐름은 inband 시그널방식으로 라우터에 적용되게 하여 기존 RSVP 라우터가 많은 흐름을 관리하지 못하는 확장성 문제를 해결할 수 있다.

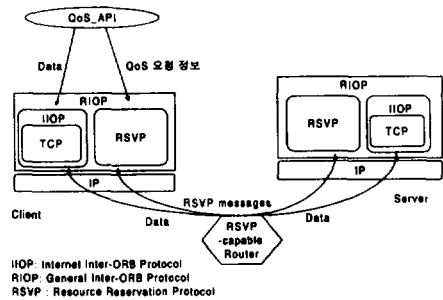
일반적으로 제어 시그널링 방식에는 inband 방식과 outband 방식이 있는데 inband 방식은 RSVP와 같이 별도의 제어 채널을 가지고 처리하는 것이 아니라 데이터와 함께 제어 시그널(QoS 정보)을 보내는 것을 의미하고, outband 방식은 RSVP와 같이 데이터 채널과는 별도로 제어 채널을 가지는 것을 의미한다.

분산망과의 연동을 위해서 백본 RSVP망의 연동 게이트웨이들은 실시간 CORBA를 지원하기 위하여 RIOP라는 통합된 프로토콜의 구조를 가진다.

기존의 Real-Time CORBA ORB에는 RIOP라 불리는 ORB 프로토콜이 있지만 각 종단 시스템에서만 QoS 보장을 지원하려고 하기 때문에 효과적인 메커니즘이 필요하다.

IIOP와 RSVP를 통합한 RIOP는 데이터 전송 이외에 자원 예약 정보를 가지는 QoS 정보를 전송하는 필드 부분을 가지고 있으며, 데이터 전송 부분은 기존의 IIOP로 전송되고 QoS 전송 부분은 RSVP와 연동된다. RIOP 구조는 IIOP 구조를 확장하여 QoS 정보를 전송하기 위한 필드를 추가하여 데이터와 함께 전송이 된다. (그림 7)은 QoS 보장 메커니즘을 고려하여 IIOP와

RSVP 통합구조를 보이고 있다. 사용자의 QoS 보장 요구는 RIOP내의 RSVP로 전송된다. QoS요구를 데이터 전송과는 별도로 자원 예약 시그널을 각 RSVP 지원 라우터에게 전송하게 한다. IIOP와 통합된 RSVP는 QoS 정보를 RSVP 지원 라우터에게 전달한다. RSVP 지원 라우터는 QoS 파라미터를 받아 자원을 예약하게 된다. 이러한 자원 예약 기능을 사용하여 사용자는 IIOP를 통해서 데이터가 전송되는 도중이라도 요구하는 QoS 파라미터 값을 조절할 수 있다.

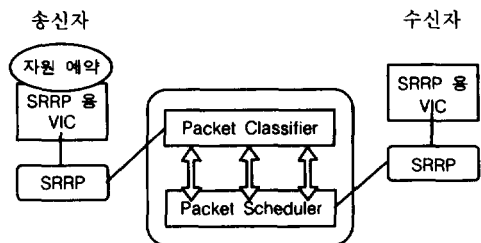


(그림 7) IIOP와 RSVP를 통합한 QoS 보장 메커니즘 구조

IIOP와 통합된 RSVP는 클라이언트로부터 서버의 객체에까지 RSVP를 지원하는 라우터를 경유해서 QoS 파라미터를 전송할 수 있게 한다.

RSVP의 자원 예약 메시지는 QoS 파라미터(대역폭/지연시간)를 표현하게 되며, 실시간성을 가지는 RIOP 프로토콜과 기존 RSVP 백본망과의 연동을 통하여 분산 환경하에서 SRRP/RSVP 연동 게이트를 통하여 각종 단 시스템 뿐만아니라 망에서도 RSVP를 통해 QoS를 지원할 수 있다.

3.3 단순화된 자원 예약 메커니즘을 이용한 화상회의 시스템의 설계



(그림 8) 단순화된 자원 예약 메커니즘을 갖는 화상회의 시스템의 구조

(그림 8)은 단순화된 자원 예약 메커니즘으로 화상회의 시스템을 구축하여 QoS 보장성을 제공하는 구조를 보이고 있다. 기존의 화상회의용 VIC 프로그램을 단순화된 자원 예약을 위해 VIC 내부에 단순화된 예약 메커니즘을 적용시킨다. 그리고 SRRP 기반의 화상회의 프로그램을 개발하여 송신자와 수신자 양단에 위치 시켜 동작을 하게 한다. 중간 라우터에서는 QoS 보장을 위해서 내부적으로 *Packet Classifier*와 *Packet Scheduler*가 있다. *Packet Classifier*는 수신된 패킷의 헤더정보(수신자 주소, 수송 계층의 프로토콜 타입, 수신자 포트 번호)를 식별하고 *Packet Scheduler*에 의해 이 패킷이 서비스 될 클래스가 결정된다.

4. 구현

4.1 단순화된 자원 예약 메커니즘의 구현

(그림9)는 Router Alert Option을 세팅하기 위한 함수의 구현 부분을 보인다. *ra_ipv4*의 값은 (그림 3)의 파라미터들을 바이트로 나누어 표현한 값이다. (그림 9)에서 *ra_ipv4*값은 *options*에 할당된다.

조작의 대상이 되는 소켓은 *fd* 파라미터에 의해 지정된다. 소켓 옵션은 프로토콜 스택의 여러 단계에 대해 정의할 수 있고 모두 *setsockopt()* 함수를 통해 접근할 수 있다.

proto 파라미터는 소켓 옵션이 정의되어 있는 단계를 나타내며, 두 가지의 값 중 하나를 가질 수 있다. SOL_SOCKET이나 IPPROTO_TCP이다. *opt* 파라미터는 인출되고 있는 옵션을 나타내며, *options* 파라미터는 옵션 값을 수신하는 버퍼를 가리키며, *size* 파라미터는 *options* 버퍼의 크기를 나타낸다. *Setsockopt()* 함수는 에러가 발생하지 않으면 0을 반환하고, 에러가 발생하면 SOCKET_ERROR를 반환한다.

```
static u_char ra_ipv4[4] = {148, 4, 0, 0};

int
Smp:: set_router_alert(int fd,int proto,int opt,void *options,int size)
{
    if (FAILED(setsockopt(fd,proto,opt,options,size)))
        net_error(NET_ERROR_SYSTEM,Router alert);
        return(SYS_ERROR);
    }
    return(SYS_NOERROR);
}
```

(그림 9) Router Alert Option을 세팅하기 위한 함수

RTCP의 확장을 위해서 rtp.h에 (그림 10)의 자원예약 확장 정보 구조체를 추가하였다.

```
enum qos_serv_type {
    QOS_CNTR_LOAD = 0, /* Controlled-Load service */
    QOS_GUARANTEED = 1, /* Guaranteed Service */
};

enum style_flag{
    DISTINCT = 0, /* distinct reservation style */
    SHARED = 1, /* shared reservation style */
};

enum verify_flag{
    ALL = 0, /* all link success */
    Some = 1, /* some failed */
};

.....
struct qos_Tspec{
    float32_t spec_Tspec_b; /* Token bucket size(Bytes)*/
    float32_t spec_Tspec_r; /* Token bucket avg rate(Bps)*/
    float32_t spec_Tspec_p; /* Peak data rate(Bps)*/
    u_int32_t spec_Tspec_m; /* Min policed unit(Bytes)*/
    u_int32_t spec_Tspec_M; /* Max pkt size (Bytes) */
};

struct qos_flowspec{
    u_int32_t spec_type; /* qos_service_type */
    qos_Tspec spec_Tspec; /* Tspec */
    float32_t spec_R; /* Rate (Bps) */
    u_int32_t spec_S; /* Slack term (microsecs)*/
};

struct comMon{
    u_int16_t re_flags; /* V:4 F:4 T:8 */
    u_int16_t re_len;
};

struct genRfc /* Generic Fragment */
{
    u_int16_t re_flags; /* S:T:8 S:1 V:1 B:1 x:1 R:4 */
    u_int16_t re_len;
    u_int32_t re_refresh;
};

struct flowSpec /* Flowspec Fragment */
{
    u_int16_t re_flags; /*S:T:8 F:T:8 */
    u_int16_t re_len;
    u_int32_t re_info;
};

struct res_ext /* Resource reservation extension */
{
    comMon com_hdr;
    genRfc gen_frag;
    flowSpec flow_spec;
};
```

(그림 10) 자원예약 확장 구조체

*qos_serv_type*은 자원 예약시 controlled-load 서비스와 guaranteed 서비스에 대한 구별을 위한 부분이다. *style_flag*에서는 SRRP의 두 가지 예약방식을 구별하기 위한 부분이고, *verify_flag*에서는 자원 예약을 위한 각 링크들의 상태를 알기 위해 필요한 부분이다. 이의

에 QoS 파라미터를 위한 *Tspec*과 *flowspec*에 대한 부분이 있고, SRRP 메시지의 자원 예약을 위한 확장된 부분(*res_ext*)으로 *comMon*, *genRic*, *flowSpec*으로 각 *fragment*들을 추가하였다.

4.2 SRRP/RSVP 연동 게이트 웨이의 구현

SRRP/RSVP게이트웨이의 요건은 SRRP와 RSVP 둘 다 있는 환경이므로 SRRP와 RSVP가 활동하는지 검사하여 작동을 하게 된다. 우선 들어온 RTCP SR 메시지중 Generic fragment의 Reservation Command를 2로 세팅하여 passive mode로 전환시키고 나가는 RTCP SR 메시지에 대해서는 0으로 다시 세팅하여 예약 가능 상태로 전환시킨다.

(그림 11)에서 함수는 flag가 0인 경우는 초기 예약 요청 실행시 mode를 0으로 하여 예약 작업을 수행하고 RSVP 백본망 진입시 passive mode로 전환시켜 다른 부가적인 예약이 행해지지 않게 하고 백본망에서 벗어나는 게이트웨이 지점에서 다시 mode를 0으로 하여 남은 예약 작업을 진행한다. flag가 1인 경우는 예약이 파기되는 경우로 mode를 1로 하여 현재 예약되어 있는 상태를 파기하게 한다.

```
int
Srrp::convert_mode(int mode,int flag)
{
    if(flag == 0) mode = 0; /* do reservation */
    elseif(flag == 1) mode = 1; /* tear reservation */
    else mode = 2; /* passive mode */
    return mode;
}
```

(그림 11) mode 변환 함수

4.3 단순화된 자원 예약 메카니즘을 이용한 화상회의 시스템의 구현

4.3.1 화상회의 시스템의 구현

VIC과 SRRP를 연동하기 위해 (그림 12)와 같은 SRRP 객체를 이용한다. 기존의 RSVP용 VIC에서 이용한 방법으로 다음 페이지의 객체와 함수들, 사용자 인터페이스 부분의 코드 그리고 위에서 언급한 각 헤더들이 VIC 소스에 추가적으로 들어간다.

소스 부분의 *srrp* 객체는 *TclObject* class를 상속받아 *command()* 멤버 함수를 재정의하여 *ui-srrp.tcl*, *ui-ctrlmenu.tcl*에서 VIC 프로그램과 연결 동작 시키기 위하여 *srrp session*, *srrp bps*, *srrp reserve*, *srrp*

unreserve, *srrp status* 같은 형태로 각 매개변수의 수와 매개변수에 의해 각 세부함수로 동작시게 하였다.

각 세부함수로선 섹션을 생성하기 위한 함수부터 시작해서 예약 작업을 수행시 필요한 예약 변수 세팅함수와 SR 메시지에 실리게 하는 함수, 예약 진행상태를 나타내는데 필요한 함수까지의 여러 함수들이 쓰이게 되며 함수들은 소켓 주소 구조체 변수와 대역폭 변수 그리고 TTL 변수를 이용한다.

```
class Srrp: public TclObject {
    /* Tcl 객체와 연동하기 위해 클래스 상속 */
public:
    /* class의 외부에서도 사용할 수 있는 member들 */
    Srrp(const char *); /* 생성자 */
    ~Srrp(); /*소멸자*/
    virtual int command(int argc, const char*const* argv);
    /*TclObject의 command()를 재정의한다.*/
protected: /* Srrp class를 상속받는 경우 그 클래스에서도 사용할
    가능한 member들 */
    int do_session(char *, const char *const *argv);
    /* 섹션을 생성하기 위한 함수 */
    int do_sender(char *, const char *const *argv);
    /* 송신자의 정보, 자원 할당 값 이용 자원 예약 처리 */
    /* 각 예약 변수들 값이 SR메시지에 실리게 된다 */
    int do_receiver(char *, const char *const *argv, struct
    sockaddr *, int);
    /* 수신자의 정보 및 TTL값 세팅 */
    int do_status(char *, const char *const *argv,struct
    sockaddr *);
    /* 예약 진행 상태 정보를 나타낸다 */
    int delayed_receiver();
    /* rate control을 자주 바꾸게 될 때 1/4초로 delay증 */
    int update_receiver(); /* 예약 매개변수들을 변경한다 */
    int configure(struct sockaddr *,int);
    /* 파일 디스크립션을 열고 TK와 연결 */
    static void dispatch(ClientData, int);
    /* 할당한 파일 디스크립션을 해제 */
    static void set_status(); /* 예약 진행 상태 세팅 */
    static void alarm_handler(int);
    /* 예약 변수를 변경키 위해 사용 */
    static Srrp *pending;
    /* delay와 예약 변수 변경시 사용 */
private: /* 아래의 변수들은 멤버 함수들에 의해서만 접근가능하
    다.*/
    struct SOCKADDR send_; /* 각 소켓 주소 구조체 변수 */
    struct SOCKADDR recv_;
    struct SOCKADDR host_;
    double bw_; /* 대역폭 값 */
    u_char ttl_; /* TTL 값 */
};
```

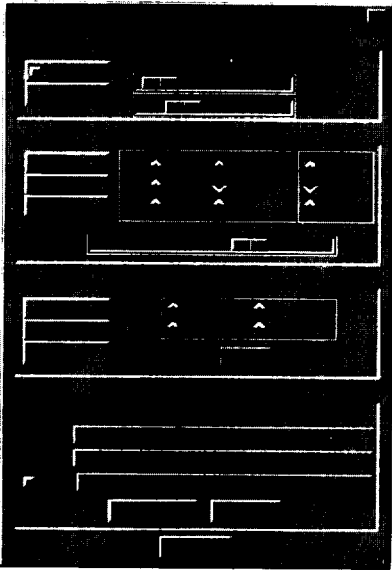
(그림 12) VIC 프로그램 내부에 구현된 SRRP 소스 코드

4.3.2 화상회의 시스템의 구현 결과

VIC 프로그램은 *tcl/tk*와 C++로 구성되어 있다. 따

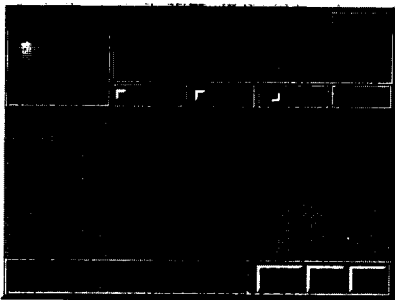
라서 SRRP와 VIC 프로그램의 연동을 위해서 C++ 코드와 tcl로 SRRP의 주요 함수들을 구성하였다. SRRP enabled VIC 2.8.1beta를 실행시키게 되면 SRRP의 동작 여부를 검사하고 검사 결과에 의해서 에러 메시지 창 열거나 reserve 버튼의 생성을 하며 우선적으로 수신자상태를 취하게 된다.

(그림 13)에서 rate control을 마친후 Transmit 버튼을 누르면 srp_sender가 실행되고, ttl값에 따라 송신자의 flowspec을 세팅하게 된다.



(그림 13) VIC 사용자 인터페이스

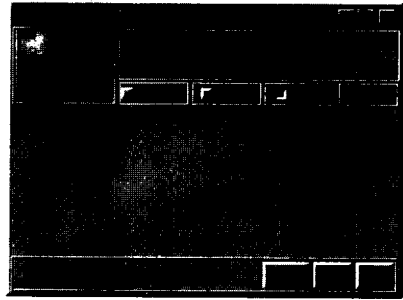
(그림 14)의 reserve 버튼을 누르게 되면 VIC 메뉴에서 rate control로 세팅된 값으로 파라미터를 설정하여 srp_reserve 작업을 하게 된다. rate control에 의



(그림 14) 송신자 등록

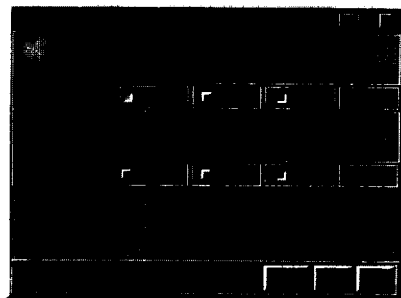
한 대역폭 값이 자주 바뀌는 것을 막기 위해 delayed_receiver로 hold down timer를 1/4로 한다. 예약의 진행 여부는 reserve 버튼의 왼쪽에 있는 박스에 색깔로 표시하는데 녹색이면 예약 성공을 나타내는 것이고 예약 실패일 경우는 파란색을 띄고 예약이 진행 과정일 경우는 노란색을 띤다.

(그림 15)는 수신자쪽에서 예약이 성공되어 나오는 송신자에서 전송된 화상 데이터를 보이고 있다. (그림 15)의 quit 버튼을 누르게 되면 Resource reservation extension 메시지 중 Generic Fragment의 Reservation Command를 1로 바꾸어 현재 설정된 예약을 파기하게 된다.



(그림 15) 수신자쪽에서의 송신자 모습

송신자와 수신자가 모두 전송을 할 경우 (그림 16)과 같이 송신자와 수신자의 2개의 창이 떠서 화상 정보를 볼 수 있게 된다.



(그림 16) 송신자쪽에서의 수신자 모습

VIC 초기 실행시 SRRP가 동작 중이지 않으면 에러 윈도우가 뜬다

이 외에도 송신자, 수신자의 주소가 잘못 됐거나 gethostname()을 사용할 수 없을 때 에러 메시지 창을

띄우게 된다.

4.4 구현환경

본 시스템은 운영체제로 Solaris 2.5 시스템을 사용하는 Sun Sparc 5, Sun Sparc 2 워크스테이션 상에서 개발되었고, 컴파일러는 GCC 2.7.2를 사용하였다. SRRP enabled VIC2.8.1 beta의 사용자 인터페이스는 X-윈도우를 기반으로 하는 tc18.0/tk8.0을 사용하여 SRRP와 연동할 수 있게 수정하였고 멀티미디어 화상 정보를 전송하기 위하여 Sun 비디오 보드를 이용하였다. 본 논문의 RSVP Router의 구성은 FreeBSD 2.2.1 운영체제 상에 네트워크 카드 2개와 RSVP 처리에 사용될 Traffic control kernel로 ALTQ/CBQ로 구성된 환경을 구축하였다. 송신자로는 OMEGA라고 명칭된 워크스테이션과 수신자로는 CCLAB이라고 명칭된 워크스테이션을 사용하였다.

각각의 설치된 RSVP 버전과 부수적인 작업을 살펴보면 CCLAB : ISL_VERSION = 4.2a4, MBone 환경을 위한 mrouted 3.8 설치(tunneling 정보 : tunnel 203.252.134.46 147.47.1.8 metric 1 threshold 32 rate_limit 500), OMEGA : ISL_VERSION = 4.2a4, RSVP router = ISL_VERSION = rel4.2a2, altq-0.4.3로 구성되어 있다.

5. 성능평가

하나의 자원 예약 과정을 행할 때 단순화된 자원 예약 메커니즘과 기존 RSVP의 예약 메커니즘의 시간으로 측정되는 부하시간(overhead time)을 측정하였다.

테스트 환경은 동일하며, 새벽에 네트워크 양이 적을 때 측정하였다. 코딩(Coding) 방법도 VIC과의 연동성과 코딩 방법에 따른 차이를 고려해야 하기 때문에 SRRP도 tc1과 C++로 동일하게 하였다.

<표 1>에서 보인 값을 볼 때 하나의 예약을 하는 과정에서 걸리는 시간이 RSVP가 SRRP에 비해서 1.5배정도 걸리는 것을 볼 수 있다. 결과적으로 RSVP의 예약과정을 보다 단순화된 방법으로 좀 더 빠른 속도로 예약을 할 수 있다는 것을 이론이 아닌 실험값으로 확인 할 수 있었다. RSVP용 Vic과 와 SRRP용 Vic에 대한 패킷손실과 jitter에 대한 성능 비교 값은 결과적으로 큰 차이점을 보이지 않았다. <표 1>은 자원예약을 한 후 초당 110(+,-)25Kbyte의 화상 데이터를 10시간에 걸쳐서 전송하며 얻은 측정값이며, 화상데이터의

특성과 중간 라우터의 용량 문제 등으로 패킷 손실과 jitter값이 발생하는 것을 확인하였다. SRRP용 Vic에서의 예약 방식은 예약 상태를 유지하기 위한 작업으로 흐름 체크만 하는 반면에 기존 RSVP용 VIC에선 송신자와 수신자 모두 refresh 메시지를 전송하여 패킷 손실과 jitter값에서 비교적 적은 값을 가지는 것을 확인할 수 있었다.

<표 1> 패킷 손실, jitter, 자원 예약 시간 비교

예약 방법 \ 비교 값	패킷 손실	Jitter (micro sec)	자원 예약 설정시 소요 시간 Time(micro sec)
RSVP용 VIC	5.23%	4,102.07(+,-) 2.13	3,309.13 (+,-) 3.21
SRRP용 VIC	5.22%	4,098.12(+,-)1.56	2,007.34 (+,-) 2.00

6. 결론 및 향후과제

실시간성을 요구하는 멀티미디어 서비스가 대두되면서, 운영체제, 망, 미들웨어 등 많은 부분에서 실시간성을 지원해야 하는 요구가 급증하고 있다.

화상회의와 같은 실시간 멀티미디어 정보의 처리가 요구되는 응용에서는 QoS 보장을 위한 방법으로 RSVP가 갖는 수신자 중심의 자원 예약 방식이나 단순화된 자원 예약 방식을 사용하여 실시간성을 요구하는 정보에 대해 일정한 지연의 데이터 흐름을 보장받게 처리해야 한다. 기존 RSVP 백본망과 분산 망과의 연동을 하는 환경에서는 실시간 서비스를 제공하기 위하여 실시간 CORBA를 이용해야 한다. 실시간 CORBA는 기존 CORBA의 구조적인 단점을 보완하기 위해 나온 새로운 CORBA 구조를 말하며, 이러한 실시간 CORBA는 그 응용 분야가 광범위하며, VOD, 화상 회의 등과 관련해 국내외적으로 활발한 연구가 진행 중에 있다. 또한, 화상회의를 위한 환경인 인터넷 환경을 고려한 본 논문의 단순화된 자원 예약 방식은 멀티 캐스트 전송을 이용하여 데이터의 중복 전송으로 인한 네트워크 자원의 낭비를 최소화 할 수 있다.

본 논문에서는 QoS 보장 차원에서 RSVP의 복잡성과 확장성 문제를 해결하고자 제안한 YESSIR와 SRP 등의 제안을 기반으로 한 SRRP라는 단순화된 자원예약 메커니즘을 개발하였다. RSVP 백본 분산망과의 연동을 고려하여 SRRP/RSVP 연동 게이트웨이에서 실시간성을 제공하는 RIOP의 적용에 관한 설계를 하였고 본 논문에서 개발한 단순화된 자원예약 메커니즘을

기반으로 한 화상회의 시스템을 구현하였다.

QoS 보장을 지원하기 위해서 단순화된 자원 예약 메커니즘기반의 화상 회의 시스템은 RSVP와는 달리 inband 자원예약 방식중심으로 구현하여 기존의 화상회의 프로그램이 갖지 못하는 QoS 보장 문제와 RSVP가 자원예약 시 발생하는 복잡성과 확장성의 문제를 해결하였다. 또한 Mbone상에서 개발된 본 논문의 화상회의 시스템은 Mbone상에서 멀티미디어 서비스가 좀 더 나은 QoS 보장을 지원할 수 있도록 하였으며, 분산망과의 연동을 위한 설계를 통하여 그 응용에 대한 적용을 시도하였다.

향후 연구 방향으로는 예약이 백본망에 진행시 가까이 있는 SRRP/RSVP 연동 게이트웨이를 찾는 참가 메커니즘의 연구와 효과적인 실시간 CORBA 구현을 위해서 IOP와 RSVP의 통합뿐만 아니라 RSVP의 QoS 대상이 되는 자원을 MIB(Management Information Base)로 정의하여 자원을 관리하는 부분에 관한 연구와 Ipv6상에서의 적용에 관한 연구도 심도 있게 다루어 저야 할 것이다.

참 고 문 헌

[1] N. Iguch and F. Uchio, "Dynamic QoS Control Function of Video Communication for remote Technical Consultation System," The 12th International Conference on Information Networking (ICOIN-12),(Tokyo Japan), pp.179-197, Jan. 1998.

[2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP : a new resource Reservation protocol," IEEE Network, Vol.7, pp.8-18, Sept. 1993.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol(RSVP)-version 1 functional specification," RFC 2205, Internet Engineering Task Force, Sept. 1997.

[4] D. C. Schmidt, Aniruddha Gophale, Tim Harrison, and Guru Parulker, "A high-performance End System architecture for Real-Time CORBA," IEEE Communication Magazine, Feb. 1997, pp.72-77, available at <http://www.cs.wustl.edu/~schmidt/TAO.html>.

[5] W. Almesberger, T. Ferrari, J. Boudec, "SRP : a

Scalable Resource Reservation Protocol for the Internet," EPFL, March. 1998.

[6] P. Pan and H. Schulzrinne, "YESSIR : A simple reservation mechanism for the internet," in IBM Research Technical Report TC20967, 1998.

[7] J. Siegel, CORBA Fundamentals and Programming, WILEY, 1996, pp.84-204.

[8] H.Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP : a transport protocol for real-time applications," RFC 1889, Internet Engineering Task Force, Jan. 1996.

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

[10] D. Katz,"IP router alert option," RFC2113, Internet Engineering Task Force, Feb. 1997.

[11] D. Katz, R. Atkinson, C. Patridge, and A. Jackson, "IPv6 router alert option," Internet Draft, Internet Engineering Task Force, June 1997. Work in progress.

현 호 재

e-mail : hjhyun@cclab.konkuk.ac.kr
 1997년 2월 동국대학교 전자계산
 학과 졸업(공학사)
 1999년 2월 건국대학교 컴퓨터공
 학과 졸업(공학석사)
 1999년~현재 건국대학교 컴퓨터
 공학과 박사 재학 중

관심분야 : CORBA, Mobile IP, IPsec, Differential service

홍 성 준

e-mail : sjhong@mail.yeojoo.ac.kr
 1991년 2월 경원대학교 전자계산
 학과 졸업(공학사)
 1993년 2월 건국대학교 전자계산
 학과 졸업(공학석사)
 1998년 8월 건국대학교 컴퓨터공
 학과 졸업(공학박사)

1993년~1999년 2월 한국통신 서울통신 운용연구단 전
 임연구원

1999년 2월~현재 여주대학 정보통신과 전임강사
관심분야 : 차세대 인터넷 프로토콜, CORBA, 인터넷 전자상거래



한 선 영

e-mail : syhan@cclab.konkuk.ac.kr

1977년 서울대학교 계산통계학과 (학사)

1979년 한국과학기술원 전산학 석사

1988년 한국과학기술원 전산학 박사

1981년~현재 건국대학교 컴퓨터 정보통신공학과 교수

1989년 1월~1990년 1월 미국 Maryland대 컴퓨터 과학과 객원부교수

1991년 1월~1997년 12월 금융결제원 자문교수

1990년 9월~1997년 12월 한국과학기술원 인공지능 연구센터 참여교수

1992년 1월~1997년 12월 개방형 컴퓨터 통신 연구회 TG-VT 의장

1991년 7월~1993년 12월 한국정보과학회 정보통신연구회 부위원장

1990년 2월~1997년 12월 ISO/IEC JIC1/SC21 WG8 위원장(국내 위원회)

1995년 1월~1997년 12월 개방형 컴퓨터 통신 연구회 TG-Web의장

1995년 6월~1997년 12월 건국대학교 산업기술연구소 정보통신연구센터 소장

1996년 1월~1998년 1월 개방형 컴퓨터 통신 연구회 총무이사

1997년 1월~현재 한국 인터넷 협회 기술위원회 위원

1997년 3월~1998년 1월 건국대학교 정보통신원 교육지원 센터 소장

1998년 1월~1999년 1월 미국 Maryland 대학교 컴퓨터 과학과 객원교수

1998년 1월~현재 개방형 컴퓨터 통신 연구회 이사
관심분야 : Real-Time CORBA, Internet Caching, 차세대 인터넷 프로토콜, Mobile IP