

# SIP 프로토콜을 이용한 하이브리드형 화상회의 시스템의 설계 및 구현

## (Design and Implementation of a Hybrid-type Video Conference System using SIP Protocol)

정 태 운 <sup>†</sup>    김 영 한 <sup>\*\*</sup>  
(TaeUn Jung)    (YoungHan Kim)

**요 약** 본 논문에서는 IETF(Internet Engineering Task Force)에서 제안된 SIP(session initiation protocol)를 사용한 하이브리드(hybrid)형 화상회의 시스템 모델을 설계하고 구현한다. 하이브리드형 화상회의 시스템은 모든 신호처리는 서버를 통해 이루어지고, 컨퍼런스 구성은 서버를 중심으로 구성될 수 있을 뿐만 아니라, UA(user agent)를 중심으로 구성되어질 수 있도록 제안 되었다. 그 결과 집중화 모델의 장점으로써 사용자 상태 관리와 라우팅이 용이할 뿐만 아니라, 애드혹(ad-hoc)모델과 단말 혼합모델의 장점인 이동성 지원이 가능하게 되었다.

제안된 모델을 바탕으로 구현된 서버는 신호처리만을 담당하여 서버의 부하가 적어 소규모 컨퍼런스 뿐만 아닌 중규모 혹은 대규모 컨퍼런스에서도 사용가능함을 확인하였고, UA는 풀메쉬(full mesh) 형태의 유니캐스트(unicast)를 사용하여 10명이하의 소규모 컨퍼런스에 적합함을 실험을 통하여 보였다.

**키워드** : SIP, 하이브리드형 컨퍼런스, 애드혹 네트워크, 화상회의

**Abstract** In this paper, we propose and implement a hybrid-type video conference system using SIP protocol proposed in IETF. This conference system, all signaling is controled by the server, while the media communication can be managed either by UAs or by the server. As a result, the complexity of its routing and user state management is reduced and it is applicable to the mobile environment of the ad-hoc models. The implemented server is scale enough to be used in a large conference because it only takes charge of signaling part. And also we see that UAs are only suitable in the small conference where less than ten people can join.

**Key words** : SIP, Hybrid-Conference, Ad-Hoc Network, Video Conference

### 1. 서론

SIPPING WG에서 연구되고 있는 SIP를 사용한 컨퍼런스 모델은 사용자 관리가 용이한 집중화모델(centralized model), 이동성 지원이 용이한 애드혹 모델(ad-hoc model), 서버의 부하가 적은 풀 메쉬 모델(full mesh model), 간단한 컨퍼런스 구성이 가능한 단말혼합모델 등이 있다[1-3]. 특히, 최근에는 컨퍼런스 서버를 둔 서버중심의 모델에 초점을 맞추어 연구되고 있다[4,5].

본 논문에서는 컨퍼런스 서버 중심의 집중화 모델에 기존 모델의 장단점을 분석하여 소규모 컨퍼런스에 적합하도록 각 모델의 장점을 접목시킨 하이브리드 형태의 화상회의 시스템을 제안하고 구현 하였다.

구현된 시스템은 기능별로 컨퍼런스 서버와 컨퍼런스 UA로 구분하여 구현했다. 컨퍼런스 서버는 웹서버, 데이터 베이스 서버, 신호처리 서버로 구성되었고, 사용자가 웹을 통하여 쉽게 화상회의 컨퍼런스에 참여가능하게 설계되었다. 컨퍼런스 UA는 신호처리 부분과 미디어 전송 부분으로 구분된 구조를 제안하여 차후 확장성을 고려하도록 설계하였다[6].

본 논문의 구성은 다음과 같다. 2장에서는 기존의 SIP를 사용한 회의 모델의 장단점을 분석하고, 장점을 접목시킨 하이브리드 모델을 제안한다. 그리고 제안된 화상회의 시스템이 적용 가능한 일반화된 컨퍼런스 구

· 본 논문은 숭실대학교 교내연구비지원에 의해 이루어졌음

<sup>†</sup> 비 회 원 : 숭실대학교 정보통신공학과  
taeun@dcn.ssu.ac.kr

<sup>\*\*</sup> 종 신 화 원 : 숭실대학교 정보통신전자공학부 교수  
yhhkim@dcn.ssu.ac.kr

논문접수 : 2003년 7월 3일

심사완료 : 2004년 11월 22일

조를 살펴본다. 3장에서는 분석 및 제안을 바탕으로 SIP 프로토콜을 적용한 화상회의 시스템을 설계하고 구현하며, 마지막으로 4장에서는 구현된 화상회의 시스템의 성능 분석을 통해 소규모 컨퍼런스에 적합함을 살펴보고 5장에서 결론을 맺는다.

### 2. 하이브리드 컨퍼런스 모델

기존에 제안된 컨퍼런스 모델들에는 단말혼합 모델, 집중화 모델, 풀 메쉬 모델, 에드 혹 모델 등이 있는데 이들은 표 1과 같은 장단점을 가지고 있다[7-13].

본 논문에서 제안하는 하이브리드 컨퍼런스 모델은 이동성을 고려한 소규모 컨퍼런스에 적합하게 적용하기 위하여 다음과 같이 여러 가지 모델을 혼합하였다. 첫째, 집중화 모델의 관리 및 서비스가 용이한 장점을 살리고, 미디어부분을 UA에서 처리하도록 하여 서버부하가 큰 단점을 해결하였다. 둘째, 에드 혹 모델의 장점인 이동성 지원을 위해 서버를 찾기 위한 메커니즘을 웹상에서 간단히 참여 가능하도록 하여 서버를 찾기 위한 메커니즘의 부재를 해결하였다[14]. 셋째, 풀메쉬 모델의 단점인 중복 알림을 하나의 서버에서 모든 신호처리를 하여 해결하였다. 그리고 마지막으로 단말 혼합 모델의 단점을 UA에서 직접 RTP를 전송 수신 가능하도록 하여 서버 탈퇴시 문제를 해결하였다.

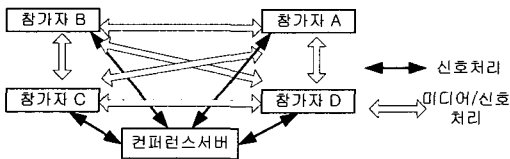


그림 1 하이브리드 모델

각각의 컨퍼런스 시스템은 그림 1과 같이 컨퍼런스 서버를 두어 모든 신호처리는 서버를 통하여 행하여진다. 이때 신호처리는 집중화 모델에서의 다이얼 아웃 신호처리 절차와 에드혹 모델에서의 다이얼 인 신호처리 절차가 혼용된 모델이다. 다이얼 인 모델은 에드혹 모델과 단말 혼합 모델에서 사용되어 지는 모델로서 참가자가 주체가 되어 회의에 참여하는 방법이고, 다이얼 아웃 모델은 집중화 모델에서 사용되어지는 방법으로 서버가

주체가 되어 회의를 개설하고 참가자를 회의에 참여시키는 방법이다. 그리고, 미디어는 풀메쉬 형태로 연결된 미디어 전송 채널을 이용하도록 하여 미디어 서버 필요 없이 단말에서 직접 혼합(mixing)하여 현재의 인터넷 망에서 사용하기 용이하다.

#### 2.1 다이얼 인 신호처리 절차

하이브리드 모델에서 다이얼 인 신호처리는 그림 2와 같이 UA와 UA간 통신중 다른 UA를 참여시키기 위해 하나의 UA가 중심이 되어 컨퍼런스를 개설한다.

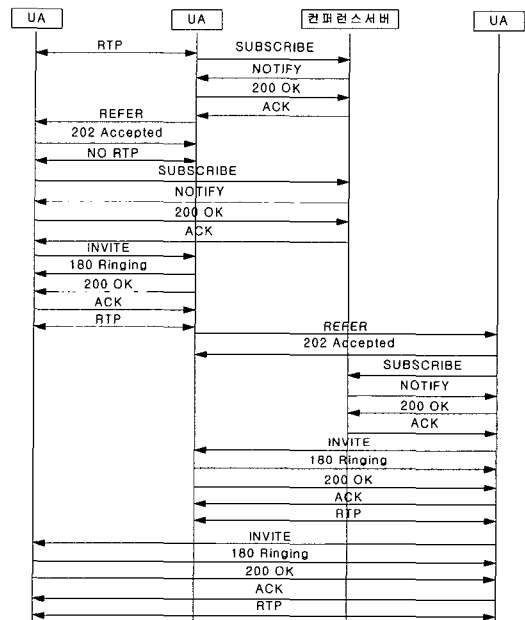


그림 2 다이얼 인 모델 신호처리 절차

먼저 SUBSCRIBE 메시지를 통해 회의를 개설하고 NOTIFY 메시지를 통해 회의의 개설을 인식한다[15]. 그리고 REFER 메시지를 통해 최초 UA를 초대한다. REFER 메시지를 받은 UA는 202 Accepted 응답메시지를 보내고 RTP를 중단한다. 그리고, 처음 UA와 같이 SUBSCRIBE 메시지를 통해 서버의 해당 컨퍼런스 URL에 등록한다. 서버는 컨퍼런스 참가자 리스트를 NOTIFY메시지를 통해 해당 UA에 알린다. 해당 UA는

표 1 기존 모델 장단점 분석

모델	집중화 모델	에드 혹 모델	풀메쉬 모델	단말 혼합 모델
장점	관리, 서비스 용이	이동성 지원 용이	분산	간단
단점	서버 부하 大	서버를 찾기위한 메커니즘 필요	UA부하 大 중복 notification 발생 가능	3자통화시 적당 임시 서버 탈퇴시 문제 발생
신호처리	스타(star)	스타	풀 메쉬	트리(Tree)
미디어	스타	풀 메쉬	풀 메쉬	트리

NOTIFY 메시지를 받고 응답 메시지를 보낸 후 컨퍼런스 참가자 리스트에 있는 UA와 세션을 맺는다. 이 절차는 기본 호 처리 절차와 같다. 그리고 새로운 UA를 참석시키기 위해 전과 같은 절차로 REFER 메시지를 새로운 UA에 보내고 새로운 UA는 컨퍼런스 서버에 SUBSCRIBE를 보낸다. 컨퍼런스 서버는 NOTIFY 메시지를 통해 2개의 컨퍼런스 참가자 리스트를 알린다. 새로운 UA는 참가자 리스트에 있는 컨퍼런스 리스트와 각각의 새로운 세션을 맺고 통화하게 된다. 또 다른 새로운 참가자도 위와 동일한 절차를 거쳐 컨퍼런스에 참여 가능하다.

2.2 다이얼 아웃 신호처리 절차

하이브리드 모델에서 다이얼 아웃 신호처리는 예약 시스템과 같은 컨퍼런스 서비스에서 이용가능한 모델로서 컨퍼런스 서버가 주체가 되어 각각의 UA를 컨퍼런스에 참여 시키는 방법이다. 컨퍼런스 서버는 예약된 시간에 각각의 UA에 REFER 메시지를 보낸다. 이때 REFER TO 헤더에는 컨퍼런스 서버의 URL을 입력하여 보낸다. REFER 메시지를 받은 UA는 순서대로 컨퍼런스 서버와 SUBSCRIBE 메시지를 보내고 NOTIFY를 받아 상태정보를 알아와 다이얼 인 신호처리에서와 같은 절차를 통해 각각의 UA와 세션을 맺고 컨퍼런스에 참여한다. 이러한 절차를 그림 3에 나타내었다.

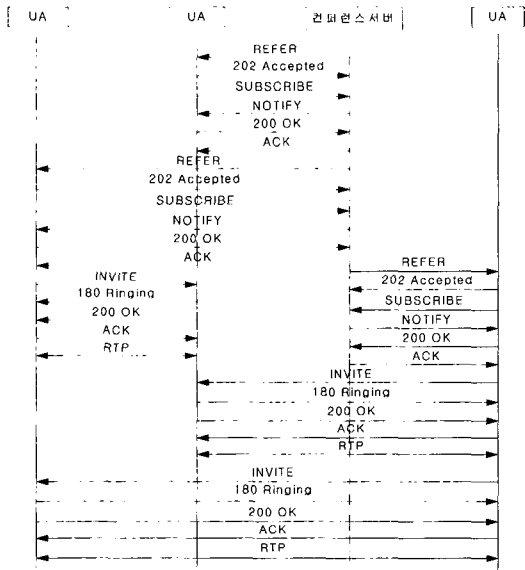


그림 3 다이얼 아웃 모델 신호처리 절차

2.3 화상회의 시스템 시나리오

하이브리드 모델 화상회의 시스템 시나리오는 그림 4와 같다. 컨퍼런스에 참여하기 위한 방법은 세 가지가

있다. 첫째 웹서버를 통한 가입, 둘째, REFER 메시지를 통한 가입, 셋째, SUBSCRIBE를 통한 가입 방법이 있다. 이러한 세가지 방법을 통해 컨퍼런스 참여절차를 살펴본다.

참가자 A가 Web 서버를 통해 컨퍼런스를 개설하고 참가자 B를 REFER 메시지를 통해 컨퍼런스에 참여하도록 권유한다. 참가자 B는 참가자 A에 REFER 메시지를 수신하였음을 알리고, 컨퍼런스에 참여를 원할 경우 SUBSCRIBE 메시지를 통해 컨퍼런스 서버에 참가를 신청한다. 컨퍼런스 서버는 데이터 베이스 서버를 수정하고 NOTIFY 메시지를 통해 참가자 리스트를 참가자 B에 알린다. 참가자 B는 NOTIFY 메시지를 수신하였음을 컨퍼런스 서버에 알리고, 기존 참가자 A와 세션을 맺고 미디어를 전송한다. 참가자 C는 웹서버를 통해 직접 컨퍼런스에 참여를 원한다. 참가자 C는 웹서버에서 해당 컨퍼런스 URL에 참여한다. 웹서버는 참가자 C에 REFER 메시지를 보내고, 참가자 C는 응답 메시지를 보낸 후 SUBSCRIBE 메시지를 통해 컨퍼런스에 참여한다. 웹서버는 컨퍼런스 서버에 참가자 C가 참여했다는 사실을 알리고, 데이터 베이스를 변경, NOTIFY 메시지를 통해 참가자 리스트를 C에 알린다. 참가자 C는 참가자 리스트의 모든 참가자 A, B와 새로운 세션을 맺고 미디어를 전송한다. 참가자 D는 참가하고자 하는 컨퍼런스 서버의 URL을 알고 컨퍼런스 서버에 SUBSCRIBE 메시지를 통해 직접 참가신청을 한다. 컨퍼런스 서버는 위와 같이 데이터 베이스를 변경하고 NOTIFY 메시지를 통해 참가자 리스트를 알린다. 참가자 D는 모든 참가자 A, B, C와 세션을 맺고 미디어를 전송한다.

3. 화상회의 시스템 구현

3.1 화상회의 시스템 구성

화상회의 컨퍼런스 서버는 컨퍼런스를 만들고 유지하며 참가자의 리스트를 관리하여 참가자에게 상태 정보를 제공하는 기능을 수행한다.

컨퍼런스 서버의 구조는 그림 5와 같다. 화상회의 서버는 웹서버, 데이터 베이스 서버, 컨퍼런스 신호처리 서버로 구성되어진다. 웹서버는 사용자가 웹을 통하여 컨퍼런스 URL을 쉽게 확인하고 컨퍼런스에 참여하는 기능과 예약 기능을 두어 예약된 시간에 참여를 원하는 참가자에게 참여 알림기능을 갖는다. 데이터 베이스 서버는 컨퍼런스 참가자 리스트와 개설된 회의방 리스트를 관리하는 기능을 갖으며, 컨퍼런스 신호처리 서버는 SUBSCRIBE와 같은 SIP 확장 메서드를 수신하였을 때 데이터 베이스 서버와 연결하여 해당되는 컨퍼런스에 참여하게하기 위한 신호처리를 담당하는 서버이다. 참가자는 화상회의 서버를 통하여 알아온 정보를 가지고 각

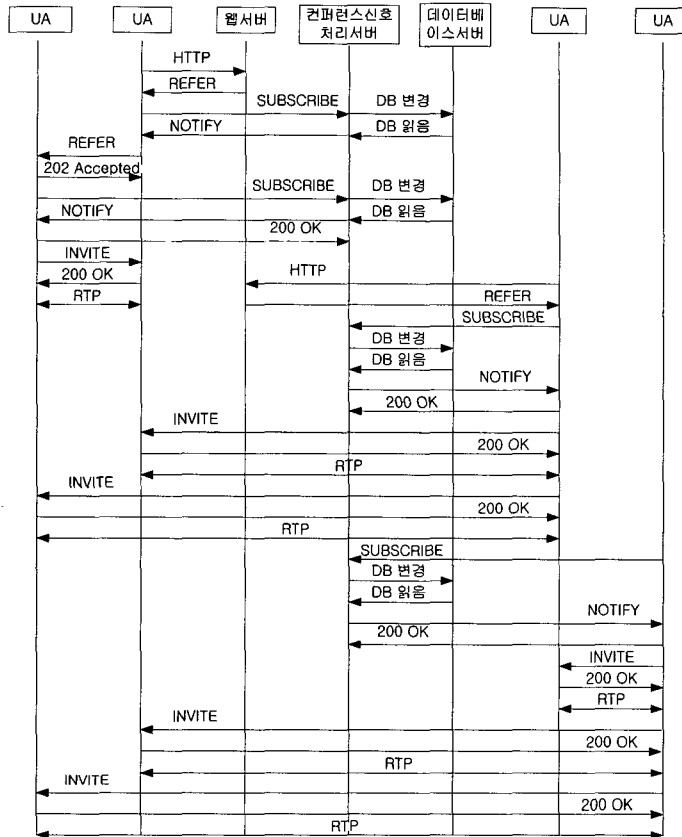


그림 4 하이브리드 컨퍼런스 모델 시나리오

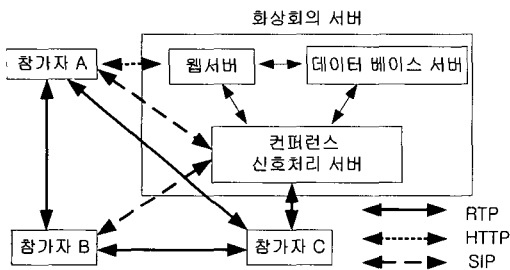


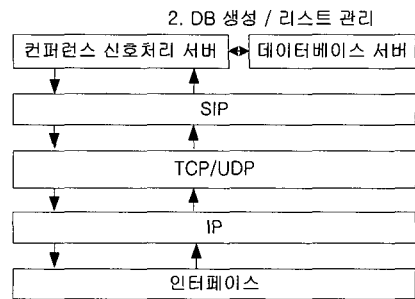
그림 5 화상회의 시스템의 구성

참가자간 미디어를 전송하여 화상회의 시스템을 구성한다.

3.2 컨퍼런스 신호처리 서버

3.2.1 컨퍼런스 신호처리 서버 구조

컨퍼런스 신호처리 서버는 컨퍼런스 참가자의 신호처리(참여/탈퇴)를 담당하는 서버로서 그림 6과 같은 프로토콜 스택을 가지고 있다. 인터페이스를 통해 들어온 패킷을 SIP 스택을 통해 파싱해서 컨퍼런스 신호처리 서버에 보낸다. 컨퍼런스 신호처리 서버는 데이터 베이스



3. SIP 신호처리(응답)      1. SIP 신호처리(요청)

그림 6 컨퍼런스 신호처리 서버의 구조

서버와 통신을 통해 컨퍼런스가 개설되어 있지 않을 경우 컨퍼런스 URL을 개설하고, 컨퍼런스가 개설되어 있을 경우 컨퍼런스 참가자 리스트를 알려준다. 즉 컨퍼런스 신호처리 서버는 컨퍼런스를 생성, 수정, 삭제하고, 사용자 리스트를 관리한다. 그리고, 필요시 발언권자를 제어하는 플로어 컨트롤의 기능을 갖는다[16].

3.2.2 컨퍼런스 신호처리 서버 상태 천이도

그림 7에 컨퍼런스 신호처리 서버의 상태 천이도를 나타내었다. 컨퍼런스 신호처리 서버는 처음 INITIAL 상태에서 참가자로부터 SUBSCRIBE 메시지를 받으면 SUBSCRIBE\_RCV상태가 되고, 데이터 베이스를 수정한다. 그리고, 데이터 베이스에서 자료를 읽어와 NOTIFY를 보낸 후 NOTIFY\_SENT상태가 된다. NOTIFY상태에서 200 OK 메시지를 받으면 200\_RCV상태가 되고 ACK를 보내고 INITIAL상태로 돌아온다. 컨퍼런스가 시작된 후 참가자로부터 BYE를 받으면 BYE\_RCV상태가 되고, 데이터 베이스에서 참가자의 정보를 삭제한 후 200 OK 메시지를 보내고 200\_SENT 상태가 된다. 200\_SENT상태에서 ACK를 받으면 INITIAL상태가 된다. 컨퍼런스 서버가 예약 컨퍼런스를 시작하기 위해 INITIAL상태에서 REFER를 전송하고 REFER\_SENT상태가 된다. REFER\_SENT상태에서 202 Accept를 받으면 INITIAL상태로 돌아가 메시지를 기다린다.

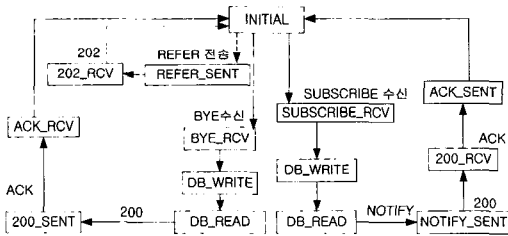


그림 7 컨퍼런스 신호처리 서버 상태천이도

3.3 웹 서버

웹 서버는 그림 8과 같은 사용자 인터페이스를 갖고, 기능은 컨퍼런스 실시간 참가 기능과 예약 참가 기능이 있다. 실시간 참가 기능은 컨퍼런스에 참여하기를 원할 때 해당 컨퍼런스 참가 버튼을 클릭하여 참가한다. 컨퍼런스 참가 버튼을 클릭할 경우, REFER 메세드를 참가

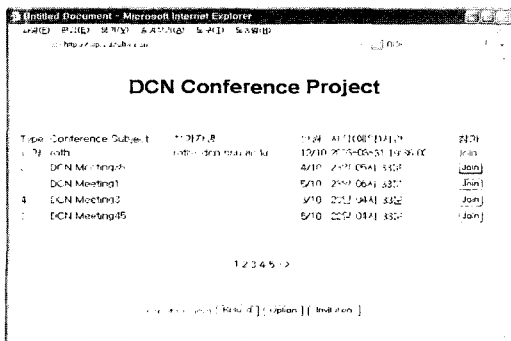


그림 8 웹서버 사용자 인터페이스

자 자신에게로 보내어 회의 서버의 위치를 알린다. 그러면 참가자는 회의 서버에게 회의 참여 메시지를 보내어 회의가 구성되어진다. 예약 기능은 참가자가 30분단위로 원하는 회의 예약시간을 설정하여 참가를 원하는 모든 사람에게 지정된 시간에 REFER 메시지를 보내어 회의가 구성되어진다.

3.4 데이터 베이스 서버

데이터 베이스 서버는 공개형 프로그램인 MySQL을 사용하였다. MySQL은 원격 처리가 가능하여 독립적인 서버로 분리할 수 있다. 데이터 베이스 서버는 컨퍼런스 신호처리 서버와의 연결을 통해 컨퍼런스 및 참가자 정보등을 관리한다. 컨퍼런스 신호처리 서버로부터 회의 개설 이벤트를 받으면 회의 방을 개설하고 회의방 정보를 알리고, 새로운 참가자가 참여하면 데이터 베이스에 정보를 추가한 후 기존 참가자의 정보를 알려준다.

3.4.1 데이터 베이스 구조

사이트에서 회의에 참가하기 위해서는 회원으로 등록해야 회의 참가 가능하다. 사이트에 회원 등록하는 정보는 이름, 사용자ID, 패스워드, 이메일이 있다. 사이트를 통해 등록된 회원은 회의에 참석하기 위한 회원목록에서 하나의 방을 선택한다. 회의 방 리스트에 대한 상세정보로는 방 주체, 참가자명, 회의시작시간이 있다. 그림 9는 이와같은 시나리오를 표현하였다.

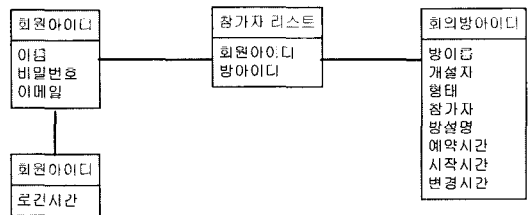


그림 9 데이터 베이스 구조

3.5 컨퍼런스 UA

컨퍼런스 UA는 컨퍼런스 서버와의 신호처리를 통해 알아온 참가자 리스트로 미디어를 전송하고, 화면에 보이는 역할을 수행한다.

3.5.1 컨퍼런스 UA의 구조

컨퍼런스 UA의 구조는 그림 10과 같다. SIP 스택을 통해 신호처리 메시지를 전송하고 수신하여, 참가자 리스트를 알아온다. 알아온 참가자 리스트를 컨퍼런스 어플리케이션에 전달하고, 컨퍼런스 어플리케이션에서는 참가자 리스트별로 신호처리를 하고 참가자 리스트를 미디어 어플리케이션에 전달한다. 제안된 구조의 미디어 어플리케이션의 전송방법에 따라 여러 가지방법으로 미디어를 전송 가능하다[17]. 대규모의 컨퍼런스는 멀티캐

스트를 사용하여, 중규모의 컨퍼런스는 명시적 멀티캐스트를 사용한 미디어 전송방법이 가능하다[18].

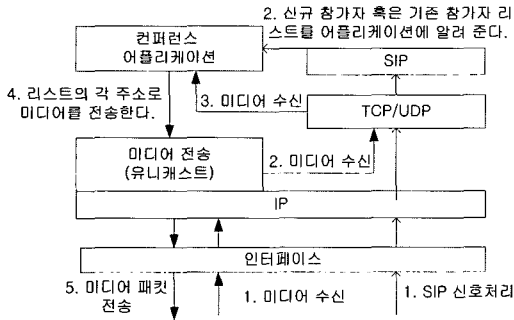


그림 10 컨퍼런스 UA의 구조

본 논문에서는 현재 인터넷 망에서 전개 가능하고 소규모의 컨퍼런스에 적합하도록 유니캐스트를 사용하여 모든 참가자간 폴 메쉬로 미디어를 전송한다.

3.5.2 UA 사용자 인터페이스

사용자 인터페이스는 그림 11과 같이 신호처리 부분과 미디어 부분을 통합하여 구현하였다. 신호처리 부분은 1:1 통화를 위한 전화 버튼과 컨퍼런스 서버에 직접 가입하기위해 SUBSCRIBE메시지를 보내기위한 가입 버튼, 새로운 참가자를 초대하기위한 REFER 버튼, 그리고 세션을 종료하기위해 BYE를 보내기 위한 종료 버튼을 구현 하였다.

미디어 전송 부분은 다음 그림과 같이 참가자 자신을 나타내는 창과 그 외 9명까지 가능하도록 구성하였으며 참가자 간 채팅이 가능하도록 구현하였다.

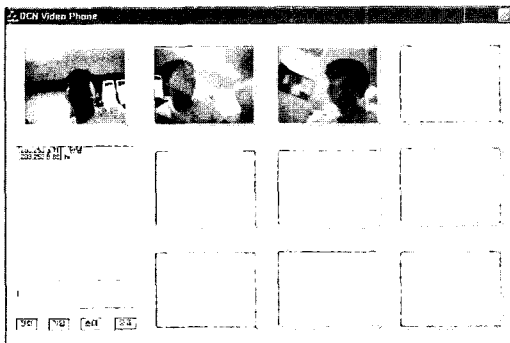


그림 11 UA 사용자 인터페이스

3.5.3 UA 상태 천이도

그림 12에 UA의 상태 천이도를 나타내었다. INITIAL 상태에서 UA는 SUBSCRIBE 메시지를 컨퍼런스 서버에 보내고 SUBSCRIBE\_SENT 상태가 된다. 이 상

태에서 NOTIFY를 받고 NOTIFY\_RCV 상태가 되고 200OK를 보낸다. NOTIFY 메시지를 확인해 자신이 아닌 다른 참가자가 없을 경우는 INITIAL 상태가 되고, 있을 경우 INVITE를 보낸다. INVITE를 보낸후 UA는 100 Trying을 수신하고 180 Ringing을 수신하면 상태 편의 선택에 따라 통화를 원할 경우 200 OK를 전송하고 CONN 상태가 되며, 통화를 원하지 않을 경우는 INITIAL 상태가 된다.

INITIAL 상태에서 REFER를 수신할 경우는 REFER\_RCV 상태가 되고 202 SENT를 전송하고, REFER의 REFER-TO 헤더에 따라 SUBSCRIBE를 보내고 전과 같은 처리를 한다.

INITIAL 혹은 CONN 상태에서 REFER를 전송할 경우는 REFER\_SENT가 되고 202 Accept를 수신하면 202\_RCV 상태가 되고 다시 CONN 혹은 INITIAL 상태로 돌아간다.

INITIAL 상태 혹은 CONN 상태에서 INVITE를 수신 하면 INVITE\_RCV 상태로 설정되고 180 Ringing을 보낸다. 180 Ringing 상태에서 수신을 허용할 경우는 200을 전송하고 200\_RCV 상태가 되고, ACK를 받으면 CONN 상태로 연결되게 된다. 하지만 수신을 허용하지 않을 경우는 183을 보내고 200을 수신하여 INITIAL 상태 혹은 CONN 상태로 돌아가게 된다.

4. 성능 분석

본 장에서는 구현된 컨퍼런스 서버와 UA의 성능을 실험을 통하여 분석한다. 서버는 CPU 펜티엄 III 1.8GHz, RAM 256 MB으로 구성하였다.

4.1 컨퍼런스 서버 성능 분석

컨퍼런스 서버의 성능 측정은 초당 접속자 수를 1000 명단위로 증가시키면서 SUBSCRIBE 메시지를 서버에 보내어 NOTIFY를 받기까지의 처리 속도를 측정하였다.

측정 결과는 그림 13과 같이 선형적인 증가 속도를 보이고 있으나 처리 속도가 지연되는 속도가 300ms이하의 작은 지연으로 서버로써 적합한 수치가 나왔으며, 2만호부터는 윈도우에서 커넥션 연결을 지원하지 않음을 확인 하였다. 따라서 실험을 통해 본 시스템은 10명 정도의 참가자를 갖는 컨퍼런스 2000개정도를 동시에 개설할 수 있다. 본 컨퍼런스 시스템은 실험에서 알 수 있듯이 소규모 컨퍼런스 뿐만 아니라, 중규모 혹은 대규모 컨퍼런스에서도 사용 가능하다.

4.2 UA 성능 분석

4.2.1 대역폭 소모량 분석

대역폭 소모는 표 2와 같이 LAN 환경과 ADSL 환경에서 측정되었다.

UA의 미디어 대역폭 소모량은 음성과 영상으로 구분

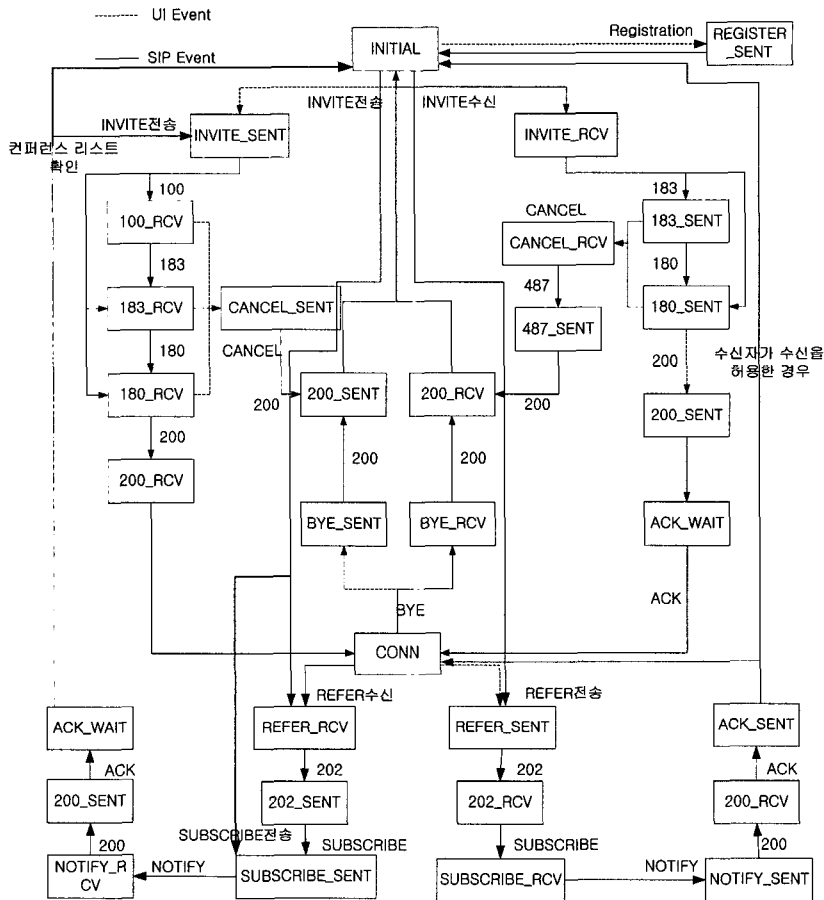


그림 12 UA 상태 천이도

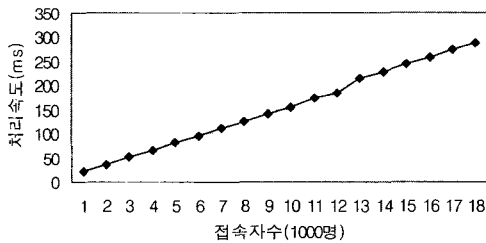


그림 13 접속자 수에 따른 처리 속도

표 2 테스트를 위한 링크

링크	이론적 전송 속도
LAN	10 Mbps
ADSL	5 Mbps

하여 측정하였다. 음성은 20ms당 172byte 데이터를 보내도록 설계되어 초당 8.6Kbyte의 데이터를 송신하고, 영상은 초당 56.4 Kbyte의 데이터를 송신한다.

표 3 미디어 별 대역폭 소모량

구분	음성	영상
대역폭 소모량	8.6 Kbyte/sec	56.4Kbyte/sec

따라서 음성만을 통한 컨퍼런스 구성시 이론상으로 LAN 환경에서는 140명 정도의 참가자가 컨퍼런스에 참여 가능하며, 음성과 영상을 같이 하여 컨퍼런스를 구성할 경우 18명 정도의 참가자가 컨퍼런스에 참여 가능하다.

표 4 네트워크 상황에 따른 컨퍼런스 참여 가능 인원

구분	음성	영상	음성+영상
컨퍼런스 참여 가능 인원(LAN)	140명	21명	18명
컨퍼런스 참여 가능 인원(ADSL)	70명	10명	9명

#### 4.2.2 참가자에 따른 지연시간 분석

4.2.1절 분석에서 UA의 컨퍼런스구성은 이론상으로 LAN상에서 18명 정도가 가능하다. 하지만 실제 통화시

망 지연시간이 300ms 이상을 넘어 갈 경우 사용자가 답답함을 느낀다. 망 지연시간에 따른 실제적인 참가자수를 구하기 위해 참가자가 증가함에 따른 지연시간을 그림 14와 같이 측정하였다.

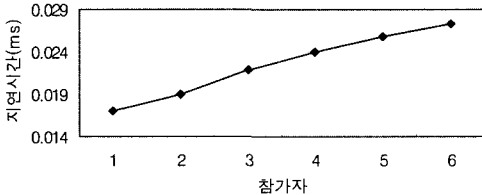


그림 14 참가자 수 증가에 따른 지연시간

수치상으로 참가자가 10명 이상이 되면 300ms 이상의 미디어 전송 지연시간을 갖게 된다. 따라서 폴 메쉬의 미디어 전송 방법을 택하고 있는 컨퍼런스 UA는 10명 이하의 소규모 컨퍼런스에 적합함을 알 수 있다.

5. 결론

본 논문에서는 소규모 컨퍼런스에 적합한 다자간 화상회의 시스템을 위해 SIP를 사용한 하이브리드 컨퍼런스 모델을 제안하고 구현하였다. 본 모델은 호처리를 담당하는 컨퍼런스 서버와 미디어전송을 담당하는 UA로 구성된 하이브리드 화상회의 모델이다.

컨퍼런스 서버는 미디어 전송 부분을 제외한 신호처리 부분만을 담당하여 서버의 부하가 작고 사용자 상태 정보 관리가 용이하며, 웹을 통한 쉬운 컨퍼런스 참가 및 이동성을 지원하는 장점을 가지고 있다.

UA는 미디어를 직접 혼합하여 미디어 서버가 필요 없는 구조를 갖는다. 즉 컨퍼런스 서버의 미디어 처리 부하를 제거 하였고 인터넷에의 용이한 적용을 위해 유니캐스트를 사용한 풀메쉬 형태를 취한다.

본 컨퍼런스 시스템은 4장의 실험에서 증명하였듯이 소규모 컨퍼런스에 적합하다. 본 시스템을 중규모 이상의 컨퍼런스로 확대할 경우 UA의 미디어 전송 구조가 수정되어야 한다. 본 논문에서는 컨퍼런스의 확대를 위해 UA의 구조를 신호처리부분과 미디어 전송 부분으로 구분하였다. 따라서 중규모 이상의 컨퍼런스를 위한 확장이 필요할 경우 미디어 전송 부분만을 멀티캐스트나 명시적 멀티캐스트로 쉽게 확장할 수 있다.

참 고 문 헌

[1] J. Rosenberg et al., "SIP:Session Initiation protocol," IETF RFC 3261, June 2002.

[2] Henning Schulzrinne, "A comprehensive multimedia control architecture for the Internet," NOSSDAV, 1997.

[3] Wenyu Jiang, "Integrating internet telephony services," Internet computing, Jun. 2002.

[4] O. Levin, and R. Even, "High Level Requirements for Tightly Coupled SIP Conferencing," IETF draft, draft-ietf-sipping-conferencing-requirements-00, Apr. 2003.

[5] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol," IETF draft, draft-rosenberg-sipping-conferencing-framework-00.txt, October 2002.

[6] H.Schulzrinne, S.Casner, R.Fredrick, and V.Jacobson, "RTP:a transport protocol for real-time applications," IETF RFC 1889, Jan. 1996.

[7] A. Johnston, "Session Initiation Protocol Call Control-Conferencing for User Agents," IETF draft, draft-johnston-sipping-cc-conferencing-00.txt, October 2002.

[8] Hennig Schulzrinne, "Centralized Conferencing using SIP," Internet Telephony Workshop, New York, Apr. 2001.

[9] J. Rosenberg, "Models for Multi Party Conferencing in SIP," IETF draft, draft-ietf-sipping-conferencing-models-01.txt, July 2002.

[10] Andrew Zmolek, Roni Even, and Nermeen Ismail, "SIP Conferencing Scenarios," IETF draft, draft-even-sipping-conference-scenarios-00.txt, August 2002.

[11] Petri Koskelainen, "A SIP-Based conference control framework," NOSSDAV, Apr. 2002.

[12] H. Khartabil, "Conferencing using SIP," IETF draft, draft-khartabil-sip-conferencing-01.txt, September 2001.

[13] Mahy et al., "A Multi-party Application Framework for SIP," IETF draft, draft-ietf-sipping-cc-framework-01.txt, June 2002.

[14] E. Guttman, C. Perkins and Sun Microsystems, "Service Location Protocol version2," IETF RFC 2608, June 1999.

[15] J. Rosenberg, "A Session Initiation Protocol (SIP) Event Package for Conference State," IETF draft, draft-ietf-sipping-conference-package-00.txt, June 2002.

[16] Koskelainen et al., "Use of SIP and SOAP for Conference Floor Control," IETF draft, draft-wu-sipping-floor-control-02.txt, November 2002.

[17] Xue Li, "Video multicast over the Internet," IEEE Network, Vol. 13, 1999.

[18] Thierry Turletti, "video conferencing on the Internet," IEEE/ACM Transactions on Networking, vol. 4, 1996.





정 태 운

1994년 3월~1998년 2월 송실대학교 전기공학과(학사). 2000년 7월~2001년 7월 ㈜동국계강. 2001년 9월~2003년 8월 송실대학교 정보통신공학과(석사). 2003년 8월~현재 경북기술(주). 관심분야는 SIP, Multicast, Autonomous Decentralized

Traffic Management System



김 영 한

1984년 2월 서울대학교 전자공학과 졸업(공학사). 1986년 2월 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 1990년 8월 한국과학기술원 전기 및 전자공학과 졸업(공학박사). 1987년 1월~1994년 8월 디지콤정보통신연구소 데이터통신연구부장. 1994년 9월~현재 송실대학교 정보통신전자공학부 부교수. 2000년~현재 VoIP포럼 차세대기술분과위원장. 관심분야는 차세대 네트워크, 인터넷 네트워킹