
SIP 환경에서의 중앙 집중형 컨퍼런스 모델 기반의 새로운 서비스 모델에 관한 연구

A Study on New Service Model Based on Centralized Conference Model in SIP Environment

조현규, 이기수, 장춘서
금오공과대학교 컴퓨터공학과

Hyun-Gyu Jo(blackjo@kumoh.ac.kr), Ki-Soo Lee(kslee@knut.kumoh.ac.kr),
Choon-Seo Jang(csjang@kumoh.ac.kr)

요약

SIP(Session Initiation Protocol) 기반의 중앙 집중형 방식의 컨퍼런스 서비스 모델은 다른 모델들에 비해 컨퍼런스의 관리와 서비스가 용이한 장점을 가지고 있다. 그러나 구성 요소인 미디어 서버가 컨퍼런스 서버에 포함되어 구현되는 경우 컨퍼런스의 수 및 참여자의 수가 많아지면 서버의 부하가 상대적으로 커지는 단점을 가지고 있다. 본 논문에서는 이를 개선하는 방법으로써 컨퍼런스 서버에 컨퍼런스 이벤트 패키지(Conference Event Package)를 포함시켜 컨퍼런스와 참여자들의 효율적인 관리를 담당하고 미디어 서버 기능을 사용자 시스템에 포함시켜 컨퍼런스를 처음 개설한 참여자의 시스템에서 처리하는 방식의 모델을 제안하고 구현하였다. 따라서 컨퍼런스 서버의 부하가 줄어들어 기존의 중앙 집중형 방식에 비해 보다 많은 컨퍼런스를 개설 및 관리가 가능할 뿐만 아니라 컨퍼런스가 개설된 이후의 참여자들은 미디어 세션의 성립을 위해서 단지 컨퍼런스를 개설한 사용자와만 호 처리를 수행하여 컨퍼런스를 진행할 수 있는 장점을 가진다.

■ **중심어** : | SIP | 중앙 집중형 컨퍼런스 서비스 | 미디어 서버 | 컨퍼런스 이벤트 패키지 |

Abstract

SIP(Session Initiation Protocol)-based centralized conference service model has advantage of easiness in conference management and service as compared to other models. However when media server which is one of the components of conference server is included in the conference server, this model shows disadvantage of high server work load with increasing numbers of conferences and participants. In this paper, to improve this problem, we have suggested and implemented a new conference service model in which an UA(User Agent) who first make the conference acts as a media server instead of conventional conference server and, the conference server with conference event package takes only part of management of the conference and its participants. Therefore, many more conferences can be held and managed compared to the conventional centralized conference service model because load of the conference server decreases in our suggested model and, furthermore the participants needs only make SIP call connections to the UA who first make the conference for establishing media session.

■ **Keyword** : | SIP | Centralized Conference Service | Media Server | Conference Event Package |

* 본 연구는 금오공과대학교 학술연구비에 의하여 수행되었습니다.

접수번호 : #051129-001

접수일자 : 2005년 11월 29일

심사완료일 : 2006년 01월 19일

교신저자 : 조현규, e-mail : blackjo@kumoh.ac.kr

1. 서론

현재 인터넷과 컴퓨터 기술의 발전으로 실시간으로 음성이나 영상과 같은 멀티미디어 데이터를 제공하고 대화 형식의 회의를 진행할 수 있는 다자간 컨퍼런스 서비스에 대한 관심과 연구가 활발히 진행되고 있다. 그 가운데 IETF(Internet Engineering Task Force)의 SIP(Session Initiation Protocol)[1]를 기반으로 적용이 가능한 컨퍼런스 모델로서 중앙 집중형(centralized) 모델, 애드 후(ad-hoc) 모델, 풀 메시(full mesh) 모델 등이 있다[2][3]. SIP는 하나 이상의 참가자간에 세션을 생성, 변경, 종료할 수 있는 응용 계층의 시그널링 프로토콜로서 텍스트 기반의 메시지를 사용함으로 확장성이 뛰어난 장점을 가지고 있는 프로토콜이다[4][5].

기존에 제안된 서비스 모델 가운데 중앙 집중형 방식은 다른 모델에 비해 컨퍼런스의 관리와 서비스가 용이한 장점을 가지고 있지만 컨퍼런스의 수와 참여자의 수가 증가함에 따라 서버의 부하가 커지는 단점을 안고 있다[6][7]. 서버의 부하를 줄이기 위한 기존의 방법에는 컨퍼런스 관리는 서버에서 수행하고 미디어 서버를 독립된 시스템으로 구성하는 방법과 미디어 처리를 서버를 거치지 않고 참여자들 간에 직접 수행하는 방법들이 있다. 그러나 기존의 이러한 방법들에는 미디어 서버를 별도로 구축하는 비용 증가와 호 처리의 복잡성과 같은 단점이 있다.

본 논문에서는 SIP를 기반으로 컨퍼런스 시스템을 구현함에 있어 컨퍼런스와 참여자의 관리는 컨퍼런스 서버에서 수행하고 미디어 서버 기능을 사용자 시스템에 포함시켜 처음 컨퍼런스를 개설한 참여자에서 모든 컨퍼런스 참여자들의 미디어를 수신하고 미디어 서버 기능을 수행하여 전체 참여자들에게 전송하는 방식으로 중앙 집중형 방식의 단점을 개선하였다.

제안하는 방법은 컨퍼런스에 참여하는 사용자 수가 많아지면 미디어 서버 기능을 처리하는 처음 컨퍼런스를 개설한 사용자 시스템의 부하가 증가하는 부분이 존재하지만 컨퍼런스 서버는 개설되는 컨퍼런스 및 참여자의 관리만 담당하면 되므로 기존의 방식에 비해 보다 많은 컨퍼런스의 수를 관리할 수 있으며 또한 컨퍼런스가 개설된 이후의 참여자들은 단지 컨퍼런스를 개설한 사용자

와 미디어 세션을 성립하면 되므로 호 처리 절차가 단순한 기존의 중앙 집중형 방식의 장점을 그대로 가진다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 연구 배경과 중앙 집중형 컨퍼런스 모델 및 컨퍼런스 이벤트 패키지에 대해 설명하고 III장에서 제안하는 모델로 구현된 시스템의 구조, 프로토콜 및 기존의 방식과의 비교를 기술하며 IV장에서 결론과 향후 과제를 제시한다.

II. 관련 연구

1. 연구 배경

IETF에서 정의하고 있는 중앙 집중형 모델은 [그림 1]과 같다[8]. 이 모델은 컨퍼런스의 관리와 미디어 관리를 중앙에서 모두 처리함으로써 서비스가 용이하며 참여자들과 컨퍼런스 서버 사이에만 미디어 세션을 성립하면 되므로 호 처리 과정이 간단한 장점도 있다. 그러나 중앙에서 컨퍼런스 서버가 모든 기능을 수행해야 하므로 컨퍼런스의 수와 참여자의 수가 증가함에 따라 서버의 부하가 커지는 단점을 안고 있다.

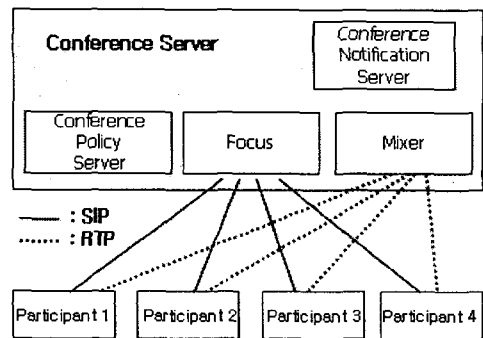


그림 1. 중앙 집중형 컨퍼런스 모델

이러한 문제점을 해결하기 위한 기존 방법의 하나로써 [그림 2]와 같이 컨퍼런스 서버에 포함되어 있는 미디어 믹싱 기능을 가진 믹서(Mixer)를 컨퍼런스 서버와는 별도의 시스템으로 구현하는 방식이 있다.

이 경우, 기본적인 중앙 집중형 방식과 같은 호 처리 절차를 가지며 컨퍼런스 서버의 부하를 줄일 수 있는 장

점이 있지만 별도의 미디어 서버를 구축함으로써 드는 추가적인 비용의 증가 문제가 발생한다. 또한 미디어 서버가 개설되는 모든 컨퍼런스의 참여자들의 미디어를 관리하고 처리해야 하므로 많은 컨퍼런스가 개설되어 전체 참여자의 수가 증가하는 경우 미디어 서버의 부하가 커지게 된다. 이때 독립된 미디어 서버를 여러 개 두는 방법을 사용하여 미디어 서버의 부하를 분산시킬 수는 있지만 상대적으로 시스템의 비용은 증가하게 된다.

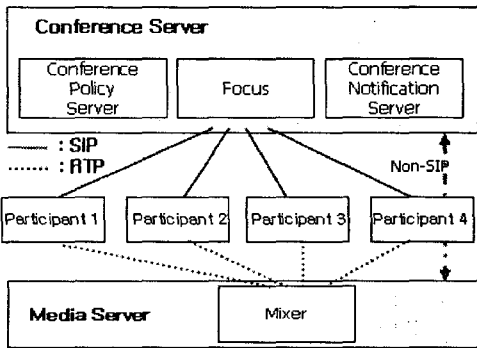


그림 2. 독립된 미디어 서버를 둔 중앙 집중형 모델

서버의 부하를 줄일 수 있는 또 다른 방법으로써 [그림 3]과 같이 컨퍼런스 관리는 컨퍼런스 서버에서 처리하고 미디어 교환은 풀 메시 방식을 사용하는 혼합형이 있다.

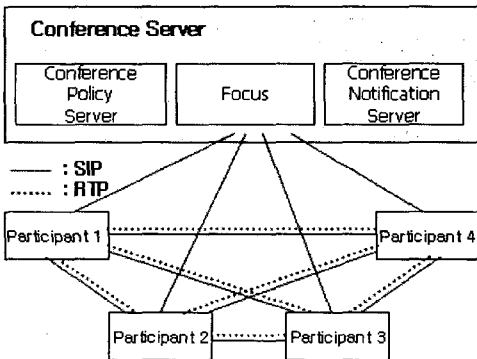


그림 3. 중앙 집중 및 풀 메시 혼합형 컨퍼런스 모델

이 경우는 컨퍼런스 서버의 부하가 줄어드는 장점을 가지고 있지만 새로 컨퍼런스에 참여하는 사용자는 기존의 모든 참여자들과 미디어 세션을 성립하기 위해 SIP

호 처리를 거쳐야 하는 번거로움이 존재할 뿐만 아니라 사용자 시스템에서 모든 참여자들과 RTP(Real-time Transport Protocol)[9]를 통한 미디어 송수신을 처리해야하므로 컨퍼런스에 참여하는 모든 UA(User Agent)의 부하가 증가하는 단점을 갖고 있다.

따라서 본 논문에서는 중앙 집중형 방식의 단점인 서버의 부하를 줄일 수 있으며 미디어 세션을 성립하기 위한 호 처리 절차를 중앙 집중형 모델과 같이 단순화시킬 수 있는 컨퍼런스 모델을 제안 및 구현하였다.

2. 중앙 집중형 컨퍼런스 모델의 구성

컨퍼런스 서비스 모델은 참가자 시스템간의 호를 제어하는 방식과 미디어를 처리하는 방식의 토폴로지에 따라 구분된다. 그 가운데 중앙 집중형 서비스 모델은 현재 IETF의 워킹 그룹을 중심으로 모델의 표준 및 정책들에 관한 연구가 진행 중에 있다[8][10][11]. 중앙 집중형 컨퍼런스 모델의 구조는 크게 컨퍼런스 참여자에 해당하는 컨퍼런스 UA와 컨퍼런스 서버로 나눌 수 있다.

컨퍼런스 참여자는 컨퍼런스 통보 서버(Conference Notification Server)의 이용이 가능하고 SUBSCRIBE, REFER, NOTIFY와 같은 확장된 SIP 메소드를 사용할 수 있는 능력의 여부에 따라 컨퍼런스 어웨어 UA(Conference Aware UA)와 컨퍼런스 언어웨어(Conference Unaware UA)로 구분된다.

컨퍼런스 서버는 최소한 포커스(Focus)를 포함하여 컨퍼런스 정책 서버(Conference Policy Server) 및 믹서로 구성될 수 있다. 포커스는 SIP UA로서 컨퍼런스를 제어하고 컨퍼런스 다이얼로그들의 전반적인 시그널링을 담당한다. 컨퍼런스 정책 서버는 전반적인 컨퍼런스의 정책을 저장 및 관리하는 역할을 담당한다. 믹서는 비디오, 오디오, 텍스트와 같은 미디어 스트림을 수신하여 각 참여자들에게 배포하는 역할을 수행한다.

3. 컨퍼런스 이벤트 패키지

컨퍼런스 이벤트 패키지(Conference Event Package)는 사용자가 컨퍼런스에 등록(Subscription)하고 동적으로 변화하는 컨퍼런스 정보를 알림(Notification)을 통해 사용자들에게 제공하는 기능을 갖고 있다[12][13]. 등

록과 알림에는 SIP 확장 메소드인 SUBSCRIBE와 NOTIFY 요청 메시지를 사용한다. 이벤트 패키지의 이름은 "conference"이며 요청 메시지의 Event 헤더나 Allow-Events 헤더에 포함되어 사용된다.

알림은 이벤트 패키지에 등록하는 시점과, 이후 새로운 참여자가 들어오거나 기존의 참여자가 컨퍼런스에서 나가는 경우와 같이 컨퍼런스 상태 정보가 변화될 때 발생한다. 만약 알림이 발생하면 변화되는 컨퍼런스 정보는 application/conference-info+xml 데이터 포맷의 XML 문서 형태로 NOTIFY 메시지의 바디에 담아서 통보된다. [그림 4]는 IETF에서 정의하고 있는 컨퍼런스 정보를 표현하기 위한 계층 구조이다.

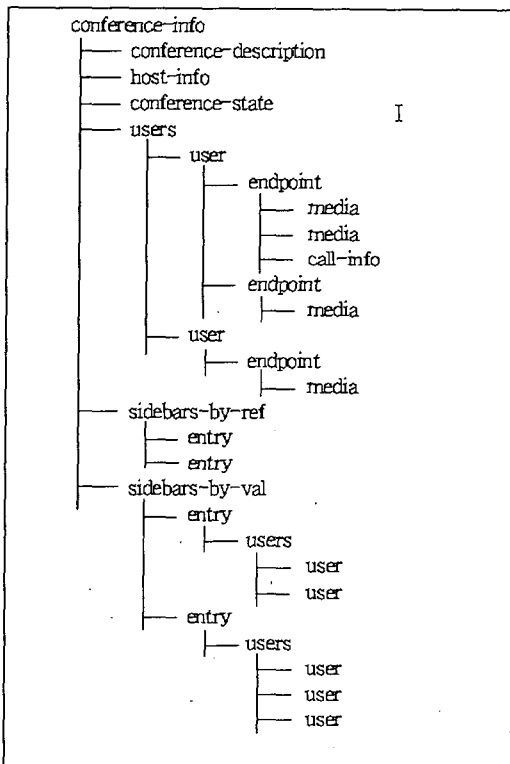


그림 4. 컨퍼런스 정보의 계층 구조

컨퍼런스 정보의 구조는 <conference-info>를 최상위 요소로 갖는다. <conference-info> 요소는 통보되는 문서의 순서를 위한 version, 전체 컨퍼런스 정보를 포함하는지의 여부와 컨퍼런스의 중단에 따라 'full', 'partial',

'deleted'의 값을 가지는 state, 컨퍼런스 URI를 나타내는 entity 등의 속성을 포함한다.

<conference-info> 요소는 <conference-description>, <host-info>, <conference-state>, <users>, <sidebar-by-ref> 그리고 <sidebars-by-val> 요소를 하위 요소로 가지며 만약 state 속성이 'full' 값을 가지면 최소한 <conference-description>과 <users> 요소는 포함되어야 한다. <conference-description> 요소는 <display-text>, <subject>, <free-text>, 그리고 <keywords> 요소를 하위 요소로 가지고 컨퍼런스 전체적인 관련 정보를 담는다. <host-info> 요소는 호스트의 정보를 나타내며 <conference-state>는 참여자의 수나 컨퍼런스의 활성 상태와 같은 컨퍼런스 정보를 포함한다.

하나 이상의 <users> 요소는 하위 요소들을 가지고 참여자의 SIP URI, 참여자의 이름, 참여자의 컨퍼런스 상태, 참여 방식, 미디어 타입과 같은 참여자 각각의 정보를 담는다. <sidebar-by-ref>와 <sidebars-by-val> 요소는 메인 컨퍼런스내의 참여자가 사이드바(sidebar)에 포함되어지는 경우 나타내기 위한 정보를 담는다.

III. 시스템 구현

1. 제안하는 컨퍼런스 서비스 모델

본 논문에서 제안하는 서비스 모델의 구조는 [그림 5]와 같으며 컨퍼런스 서버에서 미디어 서버를 분리하여 컨퍼런스 UA의 사용자 시스템에 포함시켰다. 이 모델에서는 만약 [그림 5]의 '참여자 1'이 컨퍼런스 개설자인 경우 이 사용자 시스템에서 모든 참여자들의 미디어를 수신하여 미디어 서버의 기능을 수행하고 전체 참여자에게 미디어를 송신한다. 따라서 컨퍼런스 참여자 수가 증가하는 경우 컨퍼런스 개설자의 사용자 시스템의 부하가 증가하는 부분이 존재하지만 컨퍼런스가 개설된 이후에 참여하는 사용자들은 컨퍼런스를 개설한 사용자와만 SIP 다이얼로그를 생성한 후 미디어 송수신이 가능하다. 또한 컨퍼런스 서버는 개설되는 컨퍼런스 및 참여자의 관리만 담당하면 되므로 기존의 중앙 집중형 방식에 비해 보다 많은 수의 컨퍼런스를 개설 및 관리할 수 있는

장점을 가진다.

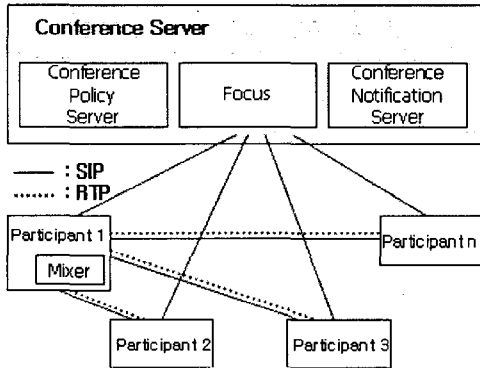


그림 5. 제안하는 방식의 컨퍼런스 모델

시스템을 구현함에 있어 구성요소인 컨퍼런스 서버는 linux OS 상에서 운영하였고 사용자 시스템은 SIP UA 와 미디어 서버 기능을 포함하여 컨퍼런스 UA로서 동작 할 수 있도록 자바를 사용하여 PC 윈도우즈 상에서 GUI 환경으로 작성하였다.

2. 사용자 시스템의 구조 및 구성 요소의 역할

구현한 사용자 시스템의 계층 구조는 [그림 6]과 같고 구성 요소들의 내부동작은 [그림 7]과 같다.

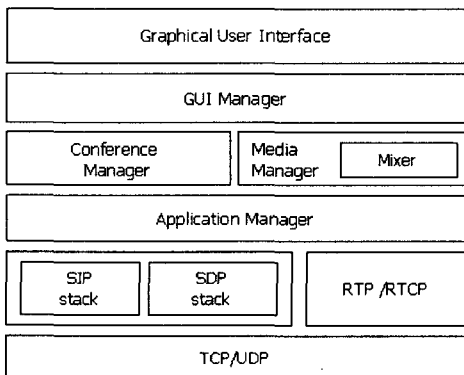


그림 6. 사용자 시스템의 계층 구조

UA는 SIP 스택과 SDP 스택을 포함하여 SIP 메시지를 송수신하고 트랜잭션을 개시 및 종료한다. 컨퍼런스 매니저는 컨퍼런스에 참여하게 되면 회의 설정, 참여, 종

료를 관리하고 컨퍼런스 서버로부터 수신한 컨퍼런스 정보를 관리하는 역할을 담당한다.

미디어 매니저는 컨퍼런스에 참여하여 미디어를 송수신하기 위한 RTP 세션을 관리하며 컨퍼런스 개설자의 시스템에서는 차후 컨퍼런스에 참여하는 다른 모든 사용자들의 미디어를 수신하여 참여자의 미디어 상태 정보를 참조하여 자원을 구별하고 이를 모든 참여자들에게 전송하는 역할을 한다. 따라서 컨퍼런스가 개설된 이후의 참여자들은 컨퍼런스를 개설한 사용자에게 설정된 RTP 세션을 통하여 다른 참여자들의 전체 미디어를 전송받을 수 있으며 사용자 시스템에서는 멀티쓰레드로 미디어를 처리한다.

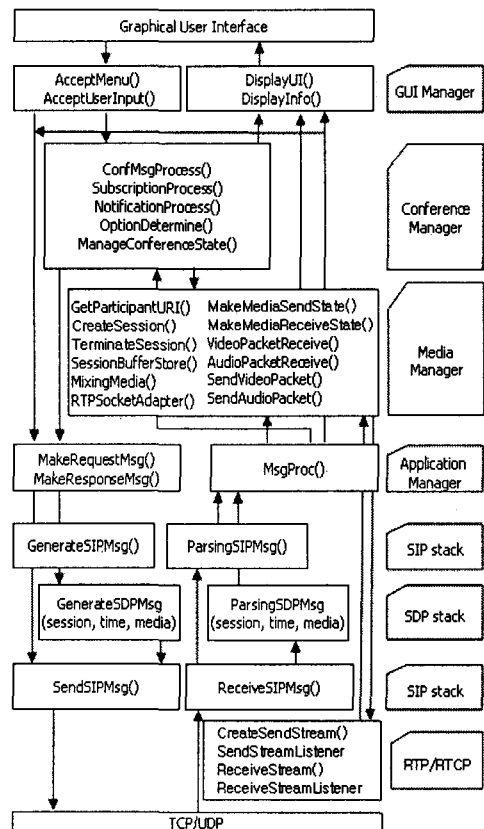


그림 7. 사용자 시스템 구성 요소의 내부 동작

GUI 매니저는 사용자의 메뉴선택 또는 입력에 대한 기능 수행을 어플리케이션 매니저에게 요청하거나 처리

되는 과정 및 결과 등을 받아 사용자에게 보여주는 역할을 담당한다. 어플리케이션 매니저는 GUI 매니저를 통한 사용자의 요청이나 UA가 받은 SIP 메시지 형태에 따라 필요한 기능 호출을 UA, 미디어 매니저, 컨퍼런스 매니저 및 GUI 매니저에게 지시하는 역할을 수행한다.

3. 컨퍼런스 서버의 구현 및 동작

구현된 컨퍼런스 서버의 내부 구조 및 동작은 [그림 8]과 같다.

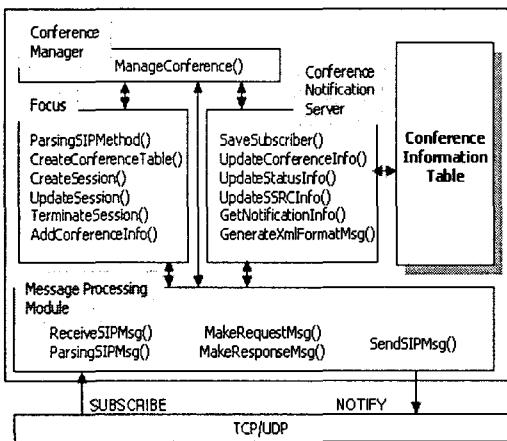


그림 8. 컨퍼런스 서버의 내부 구조 및 동작

컨퍼런스 서버 내에는 컨퍼런스 매니저를 두어 구성 요소들의 관리를 원활히 하도록 하였고 포커스와 컨퍼런스 통보 서버를 포함시켜 컨퍼런스 세션에 대한 설정, 변경, 종료를 담당하고 컨퍼런스 참여자들의 연결 설정 및 관리를 효율적으로 수행하는 컨퍼런스 제어 기능을 가지도록 하였다.

포커스를 구성하는 주요 모듈들은 컨퍼런스의 전반적인 시그널을 담당하는 모듈, 컨퍼런스 세션에 대한 설정, 변경, 종료를 처리하는 모듈, 컨퍼런스에서 발생하는 모든 정보의 전달을 제어하는 모듈, 컨퍼런스 UA의 컨퍼런스 참여 및 종료 요청을 처리하는 모듈 등이다. 컨퍼런스 통보 서버의 주요 모듈들은 컨퍼런스 정보와 참여자의 정보를 등록 및 갱신하는 모듈, 변경된 컨퍼런스 정보를 갱신하는 모듈, 컨퍼런스 상태 정보를 참여자에게 통보하는 모듈 등이다.

특히 컨퍼런스 통보 서버 내에는 컨퍼런스 이벤트 패키지를 포함시켜 동적으로 변화하는 컨퍼런스 정보 및 참여자의 정보를 등록과 알림을 통해 제공한다. 알림은 새로운 사용자가 컨퍼런스에 참여하거나 참여한 사용자의 상태가 변화하는 등 컨퍼런스 정보가 갱신되는 경우에 발생하며 변화된 컨퍼런스 정보를 application/conference-info+xml 데이터 포맷으로 담아 NOTIFY 메시지의 바디에 포함시켜 통보한다.

4. 제안하는 방식의 호 처리 및 컨퍼런스 시나리오

구현한 시스템은 다이얼 인 방식으로 컨퍼런스를 진행한다. 다이얼 인 방식은 컨퍼런스 UA가 컨퍼런스 설정 또는 참여를 서버에 요청하는 방식이다[10][14]. [그림 9]는 제안하는 방식으로 컨퍼런스를 개설하여 다른 사용자를 초청하고 컨퍼런스를 진행함에 있어 미디어 세션을 성립하기 위한 호 처리 절차 과정이다.

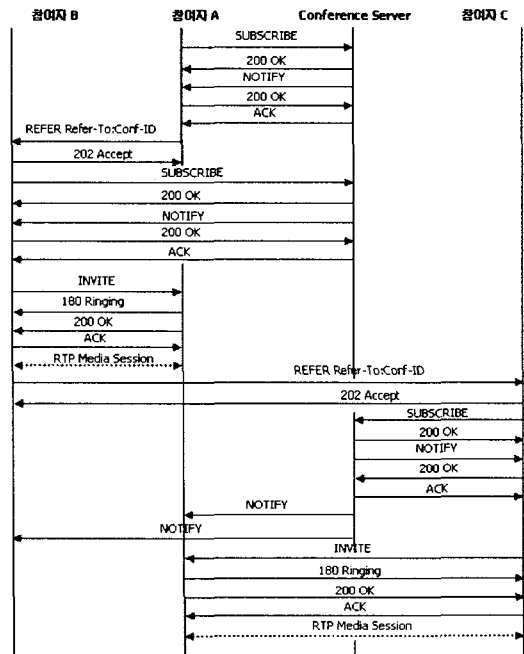


그림 9. 제안하는 방식에서의 호 처리 절차

제안하는 방식으로 컨퍼런스를 진행하는 시나리오는 다음과 같다. 먼저 참가자 A는 컨퍼런스 서버에게

SUBSCRIBE 요청 메시지를 사용하여 컨퍼런스를 개설한다. 이때 컨퍼런스 서버는 새로운 컨퍼런스 생성에 대한 요청이므로 참가자 A의 정보를 등록 시 컨퍼런스를 개설한 사용자임을 함께 저장한다. 이 정보는 차후 다른 사용자가 개설되어 있는 컨퍼런스에 참여할 때 컨퍼런스 서버로부터 통보받는 컨퍼런스 정보에 포함되어지며 미디어 서버 기능을 수행하는 개설자를 확인할 수 있는 용도로 사용된다. 이를 위해 내부적으로는 컨퍼런스 정보의 <user> 요소의 하위 요소로서 <conf-creator> 요소를 추가하여 'yes' 또는 'no'의 값을 가지도록 하였다.

참여자 A는 REFER 메시지[15]를 사용하여 참여자 B를 초대한다. 참여자 B는 정상적으로 REFER 메시지를 수신하였음을 참여자 A에게 알리고 컨퍼런스 서버에게 SUBSCRIBE 메시지를 통해 참여를 신청한다. 이때 서버는 컨퍼런스 이벤트 패키지에 등록되어 있는 컨퍼런스 정보와 컨퍼런스 개설자의 여부를 포함한 참여자 정보를 NOTIFY 메시지를 사용하여 통보한다.

참여자 B는 이를 파싱하여 회의를 개설한 참여자 A와 세션을 맺고 회의를 시작한다. 참여자 A의 미디어 매니저는 컨퍼런스에서 사용될 오디오 및 비디오 채널을 정하고 각 미디어 별로 RTP 세션을 설정한다. 또한 자신과 수신한 참여자 B의 미디어를 믹서를 통해 믹싱하고 자신의 화면에 보여주며 참여자 B에게 전송한다. 이후 참여자 C의 초정은 [그림 9]에서처럼 컨퍼런스 개설자가 아니라도 가능하다. 제안하는 모델의 특성상 새로운 참여자는 컨퍼런스 개설자와 미디어 세션을 맺어야 하는데 새로운 참여자는 서버로부터 NOTIFY 메시지를 수신하여 개설자의 정보를 확인할 수 있으므로 이를 통해 세션을 성립하고 회의에 참여가 가능하게 된다. 이후 다른 사용자들도 동일한 과정을 거쳐 회의에 참여가 가능하다.

5. 실험 및 기존 모델과의 비교

미디어 처리를 위한 실험 환경은 10Mbps LAN에서 음성은 RTP 프로파일의 음성 인코딩 부분에서 샘플 당 8bit로 코딩되는 CCITT/ITU-T G.711 포맷을 사용하여 전송률 64Kbps로 데이터를 송신하고 영상은 클럭 주파수 90KHz인 RTP 페이로드 타입 26을 사용하여 약

450Kbps 속도로 데이터를 송신하였다. 그러므로 이론적으로 음성과 영상을 포함한 하나의 컨퍼런스에서는 약 18명 정도의 사용자가 참여가 가능하다. 그러나 현실적으로 서버와 사용자 시스템의 처리 시간 및 네트워크 지연 시간이 존재하며 참여자가 증가함에 따라 만약 지연 시간이 300ms 이상 발생하면 컨퍼런스 진행은 부자연스러워지므로 실제 참여자 수는 줄어들게 된다. 따라서 제안하는 모델에서 참여자가 증가함에 따른 지연 시간을 측정하여 실제 참여가 가능한 사용자 수를 알아보았다.

실험 결과, [그림 10]에서와 같이 참여자가 1명씩 늘어날수록 약 20ms의 추가적인 지연 시간이 발생하였으며 그래프 상의 추세라면 컨퍼런스 서버를 통해 개설되는 각각의 컨퍼런스 당 약 10명의 사용자 참여가 가능함을 알 수 있다.

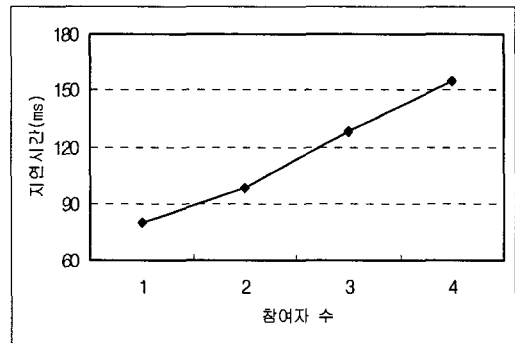


그림 10. 참여자 수의 증가에 따른 지연시간

제안하는 모델과는 달리 기존의 중앙 집중형 모델은 컨퍼런스의 관리와 서비스가 용이한 장점이 있는 반면 컨퍼런스 서버에서 컨퍼런스의 개설 및 관리뿐만 아니라 개설되는 모든 컨퍼런스의 참여자들의 미디어 관리를 담당해야 하므로 참여자의 증가에 따른 지연시간이 제안하는 모델 보다 훨씬 길어질 수밖에 없다. 그러므로 컨퍼런스의 수와 참여자의 수가 제안하는 모델보다 훨씬 제한적이다.

기존의 중앙 집중형 및 풀 메시의 혼합 방식은 제안하는 모델과 마찬가지로 컨퍼런스의 관리와 미디어 부분을 분리함으로써 개설 가능한 컨퍼런스 수나 참여자의 수가 비슷하겠지만 미디어 처리를 함에 있어 모든 참여자 시

시스템의 부하가 증가하며 호 처리 과정이 훨씬 복잡하다. 즉, [그림 11]과 같이 컨퍼런스에 참여하고 있는 참여자가 n명일 경우 n+1 번째 참여자는 n명의 사용자 모두와 호 처리 과정을 거쳐 RTP 세션을 성립해야한다. 반면 제안하는 모델은 기존의 중앙 집중형 방식에서 참여자들이 컨퍼런스 서버와 1:1로 미디어 세션을 성립하는 [그림 12]와 같은 방식과 동일하게 [그림 9]와 같은 호 처리 과정을 거치므로 중앙 집중형 방식의 장점을 가진다. 더불어 제안하는 모델은 컨퍼런스 개설자의 시스템에서만 미디어 믹싱 처리에 따른 부하가 발생하므로 개설자 이외의 다른 참여자들은 또 다른 컨퍼런스에 동시에 참여할 수 있는 등의 장점을 가질 수 있다.

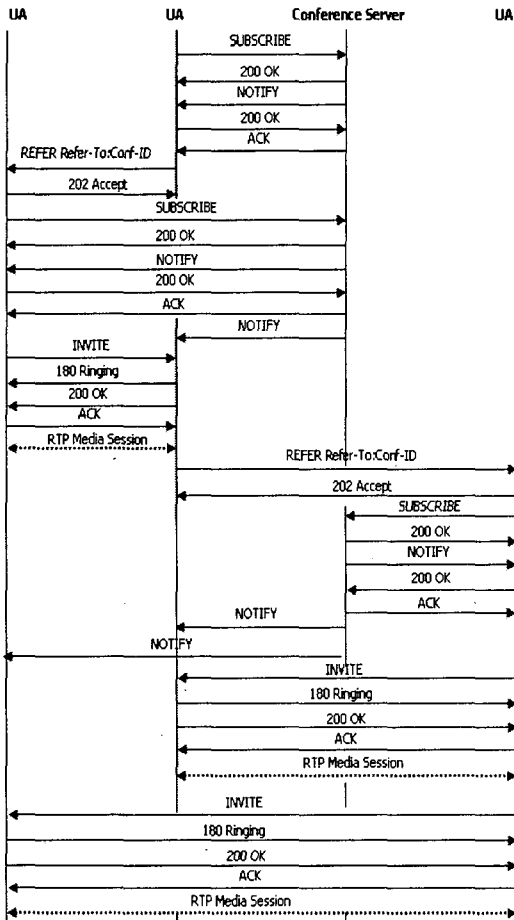


그림 11. 중앙집중 및 플레이스 혼합 방식의 호 처리 절차

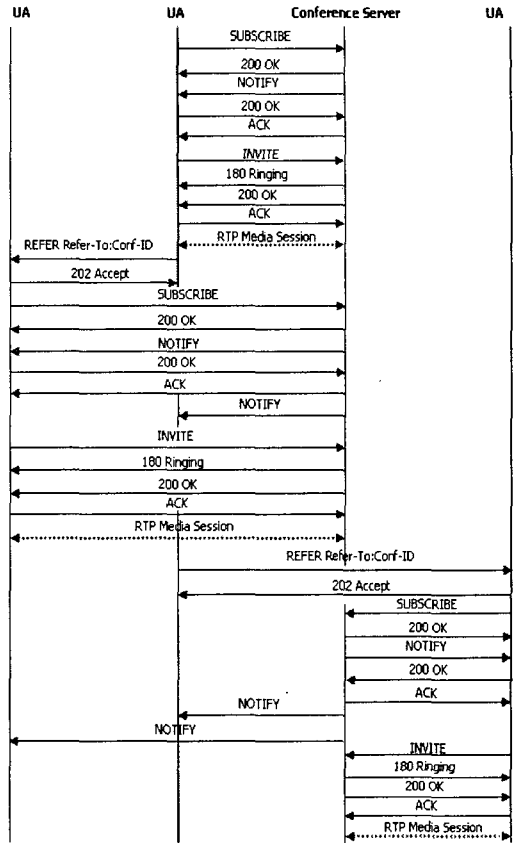


그림 12. 중앙 집중형 컨퍼런스 방식의 호 처리 절차

6. 사용자 인터페이스

[그림 13]은 제안하는 방식으로 구현한 컨퍼런스 서비스를 이용하여 화상 컨퍼런스를 진행하고 있는 화면이다.



그림 13. 메인 사용자 인터페이스

메인 사용자 인터페이스 화면은 컨퍼런스에 참여중인 사용자 목록, 컨퍼런스 이벤트 패키지를 통해 제공받은 참여자 정보, 텍스트 송수신 출력 부분, 참여자들의 화상 화면 및 컨퍼런스의 개설과 초청, 참여, 종료 등에 필요한 SIP 메시지 생성을 위한 버튼들로 구성하였다.

IV. 결론

본 논문에서는 SIP 환경에서 중앙 집중형 컨퍼런스 모델을 기반으로 서버의 부하를 줄여 보다 많은 컨퍼런스를 관리할 수 있고 기존의 중앙 집중형 모델이 가지는 미디어 세션을 성립하기 위한 호 처리 과정이 간단한 장점을 가지는 컨퍼런스 서비스 모델을 제안하고 구현하였다.

구현한 시스템에서는 컨퍼런스와 참여자의 관리는 서버에서 수행하고 미디어 서버의 기능을 분리하여 컨퍼런스를 개설한 사용자 시스템에서 수행하고 모든 참여자들에게 전체 참여자의 미디어를 송신하도록 하였다. 따라서 중앙 집중형 방식에 비해 컨퍼런스 서버의 부하가 줄어들며 또한 중앙 집중형 및 풀 매시 혼합방식과는 달리 컨퍼런스가 개설된 이후의 새로운 참여자들은 미디어 서버 기능을 수행하는 컨퍼런스를 개설한 사용자와만 미디어 세션 성립을 위한 한번의 SIP 다이얼로그 생성 과정을 거치면 된다.

컨퍼런스 서버는 컨퍼런스의 참여와 동적으로 변화하는 컨퍼런스 정보를 참여자에게 제공하기 위해서 컨퍼런스 통지 서버 내에 컨퍼런스 이벤트 패키지를 포함시켜 구현함으로써 원활한 등록과 알림이 가능하도록 하였다.

그러나 제안하는 방식은 컨퍼런스에 참여하는 사용자 수가 많아지면 미디어 서버 기능을 수행하는 처음 컨퍼런스를 개설한 사용자 시스템의 부하가 상대적으로 증가하게 되어 성능이 저하되는 부분이 존재한다. 따라서 향후 과제으로써 참여자 수가 많아지는 경우 미디어 서버 기능을 다른 참여자들의 시스템에 분산시켜 효율을 높일 수 있는 추가적인 연구가 필요하다.

참고 문헌

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, 2002.
- [2] J. Rosenberg and H. Schulzrinne, "Models for Multi Party Conferencing in SIP," draft-ietf-sipping-conferencing-models-01, 2002.
- [3] I. Miladinovic and J. Stadler, "Multiparty Conference Signaling using the Session Initiation Protocol(SIP)," in International Network Conference, 2002.
- [4] Ismail Dalgic and Hanlin Fang, "Comparision of H.323 and SIP for IP Telephony Signaling," Proc. of Photonics East, Boston, Massachusetts, 1999.
- [5] F. Fingal and P. Gustavsson, "A SIP of IP-telephony," http://www.cs.columbia.edu/sip/drafts/Fing9902_SIP.pdf, 1999.
- [6] K. Singh, G. Nair, and H. Schulzrinne. "Centralized conferencing using SIP," in Internet Telephony Workshop, New York, 2001.
- [7] I. Miladinovic and J. Stadler, "Optimization of signaling traffic in centralized conferences using SIP," in Book *ADVANCES IN COMMUNICATIONS AND SOFTWARE TECHNOLOGIES*, pp.101-106, 2002.
- [8] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol," draft-ietf-sipping-conferencing-framework-05, 2005.
- [9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, 2003.
- [10] A. Johnston and O. Levin, "Session Initiation Protocol Call Control - Conferencing for User Agents," draft-ietf-sipping-cc-conferencing-07,

2005.

- [11] P. Koskelainen and H. Khartabil, "Requirements for Conference Policy Control Protocol," draft-ietf-xcon-cpcp-reqs-04, 2004.
- [12] A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, 2002.
- [13] J. Rosenberg, H. Schulzrinne, and O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State," draft-ietf-sipping-conference-package-12, 2005.
- [14] O. Levin and R. Even, "High Level Requirements for Tightly Coupled SIP Conferencing," draft-ietf-sipping-conferencing-requirements-01, 2004.
- [15] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method," RFC 3515, 2003.

이 기 수(Ki-Soo Lee)

정회원



- 1979년 2월 : 경북대학교 전자공학(공학사)
- 1982년 2월 : 서울대학교 대학원(공학석사)
- 1982년 3월~현재 : 금오공과대학교 컴퓨터공학부 교수

<관심분야> : 교육, IT

장 춘 서(Choon-Seo Jang)

정회원



- 1978년 2월 : 서울대학교 전자공학(공학사)
- 1981년 2월 : 한국과학기술원(공학석사)
- 1993년 2월 : 한국과학기술원(공학박사)

- 1981년 3월~현재 : 금오공과대학교 컴퓨터공학부 교수

<관심분야> : 실시간 인터넷 통신, VoIP, SIP, 임베디드 시스템

저자 소개

조 현 규(Hyun-Gyu Jo)

정회원



- 1991년 2월 : 금오공과대학교 전자공학과(공학사)
- 1995년 2월 : 금오공과대학교 전자공학과(공학석사)
- 2005년 8월 : 금오공과대학교 컴퓨터공학과(공학박사)

<관심분야> : SIP, VoIP, 실시간 인터넷 통신