
SIP 기반의 효율적인 혼성형 멀티미디어 컨퍼런스 구조

Architecture of SIP-based Effective Hybrid-type Multimedia Conference

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

중앙에 컨퍼런스 서버가 위치하여 컨퍼런스 전체를 제어하고 관리하는 SIP(Session Initiation Protocol) 기반의 밀결합 형태의 컨퍼런스 방식은 포커스(focus)와 믹서의 위치에 따라 중앙 집중형, 단말 서버형, 미디어 서버 분리형, 믹서 분산형 등 여러 모델로 분류할 수 있는데 모두 각각의 장단점을 가지고 있다. 본 논문에서는 서버와 참가자 사이에 오가는 SIP 시그널링 메시지 양을 줄일 수 있고 서버의 미디어 믹서 부하를 경감시킬 수 있으며 또한, 대규모 컨퍼런스로의 확장이 용이한 SIP 기반의 효율적인 혼성 형태의 컨퍼런스 모델을 제안하고 이를 설계 및 구현하였다. 이 모델에서는 참가자 수가 미리 정해진 일정한 한도를 넘으면 서버는 특정 기능을 가진 참가자들을 선택하여 이들로 하여금 컨퍼런스 상태 이벤트 패키지의 통지 기능 및 미디어 믹서 기능을 서버와 분담하도록 한다. 각 참가자는 서버에 컨퍼런스 상태 이벤트 패키지를 등록할 때 특정 헤더 메시지로 이와 같은 기능의 보유 여부를 알릴 수 있고 서버는 컨퍼런스 정보 데이터베이스에 이를 표시하여 나중에 기능을 분담할 참가자들을 선정할 수 있게 된다. 제안된 방식은 실험을 통하여 성능을 측정하였다.

■ 중심어 : | SIP | 컨퍼런스 | 혼성형 | 컨퍼런스 상태 이벤트 패키지 |

Abstract

SIP-based tightly coupled conference, which has a centrally located conference server for controlling and management, can be classified several models according to location of focus and mixer. These are centralized server model, endpoint server model, media server component model and distributed mixing model. However each model has its strength and weakness. In this paper, we propose and implement a SIP-based effective hybrid-type conference model which decreases amount of SIP signaling messages, lowers load of server media mixer, and can be easily expandable to large scale conference. In this model, when the number of participants exceeds a pre-defined limit, the conference server selects some participants which possess specific functions and let them share functions of notifications of conference state event package and media mixing. When each participant subscribes conference state event package to the server, it can indicate its possession of such functions by a specific header message. The server stores the indication to the conference information database, and later uses it to select participants for load sharing. The performance of our proposed model is evaluated by experiments.

■ Keyword : | SIP | Conference | Hybrid | Conference State Event Package |

* 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

접수번호 : #061113-008

심사완료일 : 2007년 01월 15일

접수일자 : 2006년 11월 13일

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

I. 서론

컨퍼런스 서비스에 사용할 수 있는 대표적인 시그널링 프로토콜로는 H.323과 SIP(Session Initiation Protocol)[1]를 들 수 있다. H.323이 호 처리 및 메시지 형태의 복잡성으로 기능 확장에 어려움이 있는 반면 텍스트 기반의 메시지를 사용하는 SIP는 단순함 및 확장성 면에서 유리하여 컨퍼런스 서비스의 시그널링 프로토콜로 많이 사용되고 있다[2][3]. SIP 기반의 밀결합 형태의 컨퍼런스 방식은 중앙에 컨퍼런스 서버가 위치하여 컨퍼런스 전체를 제어하고 관리하는데, 포커스(focus)와 믹서의 위치에 따라 여러 모델로 분류할 수 있다. 이들은 중앙 집중형, 단말 서버형, 미디어 서버 분리형, 믹서 분산형 모델 등이며 모두 각각의 장단점을 가지고 있다[4][5].

이들 중 중앙 집중형 모델은 컨퍼런스의 수와 참여자의 수가 증가함에 따라서 서버의 부하가 커지는 단점이 있고 단말 서버형은 서버 역할을 하는 단말 쪽에서 포커스와 믹서를 모두 갖추어야 하며 소규모 컨퍼런스만 가능하다는 단점이 있다. 미디어 서버 분리형은 추가로 미디어 서버가 필요하며 믹서 분산형 모델은 모든 참가자가 믹서 기능을 보유해야 하고 전체 참가자 사이에 모두 미디어 연결이 필요해 참가자가 증가할수록 미디어 세션 연결 횟수가 크게 증가하는 문제점 등을 가지고 있다[6].

본 논문에서는 각 모델에서의 이와 같은 문제점을 개선하기 위하여 컨퍼런스 서버와 참가자 사이에 오가는 SIP 시그널링 메시지 양을 줄일 수 있고 서버의 미디어 믹서 부하를 경감시킬 수 있으며 대규모 컨퍼런스로의 확장이 용이한 효율적인 혼성 형태의 컨퍼런스 모델을 제안하고 이를 설계 및 구현하였다. 이 모델에서는 참가자 수가 일정 한도를 넘으면 서버는 특정 기능을 가진 참가자들을 선택하여 이들로 하여금 미디어 믹서 기능 및 컨퍼런스 상태 이벤트 패키지의 통지 기능을 분담하도록 한다. 각 참가자는 서버에 컨퍼런스 상태 이벤트 패키지를 등록할 때 특정 헤더 메시지로 이와 같은 기능의 보유 여부를 알릴 수 있다. 컨퍼런스 참가자 수에 따라 서버의 기능을 분담할 기능을 가진 참가자는

하나에서 여러 개로 늘어 날 수 있다. 따라서 이러한 동작에 기반하여 대규모 컨퍼런스로의 확장이 용이해지며 중앙 집중형 모델에서의 서버 부하를 감소시킬 수 있고 분산형 모델에서 요구하는 각 참가자가 미디어 믹서 기능을 가져야 하는 등의 복잡성을 감소시킬 수 있다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 각 모델의 장단점을 분석하고 III장에서는 본 논문에서 제안하는 SIP 기반 혼성 멀티미디어 컨퍼런스 구조를 설계 및 구현하며 IV장에서는 구현된 시스템의 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

1. 중앙 집중형 모델

SIP 기반의 여러 컨퍼런스 모델 중 중앙 집중형 모델은 중앙의 서버에 컨퍼런스를 위한 전체 참가자들의 SIP 신호를 관장하는 포커스, 미디어 스트림을 수신해 이를 조합하고 전체 참가자들에게 분배하는 믹서, 컨퍼런스 정책 모듈, 그리고 컨퍼런스 등록에 대한 통지 서비스를 해주는 모듈을 모두 포함시켜 동작하는 방식이다[7]. 각 참가자들은 서버의 믹서와 일대일 RTP (Real-time Transport Protocol)[8] 세션 연결을 맺는다. 이는 컨퍼런스의 관리와 서비스가 용이한 장점이 있으나 컨퍼런스의 수와 참여자의 수가 증가함에 따라서 서버의 부하가 커지는 단점을 가지고 있다. [그림 1]은 컨퍼런스 정책 모듈을 생략한 중앙 집중형 모델의 구성도이다.

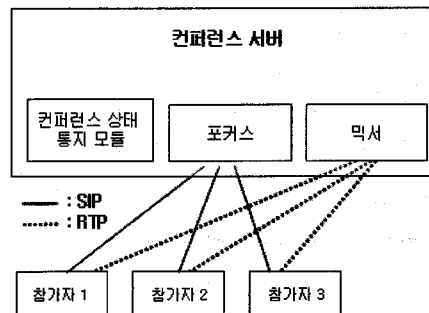


그림 1. 중앙 집중형 모델

2. 단말 서버형 모델

이 모델은 참가자 시스템의 UA(User Agent) 자체에 포커스, 믹서, 컨퍼런스 정책 모듈, 그리고 필요하면 컨퍼런스 상태 통지 모듈까지 모두 포함하는 형태이다. 보통, 두 명의 사용자가 통상의 일대일 SIP 연결로 서로 통신하다가 제3의 참가자를 참여시키고자 할 때 처음 두 명 중 하나가 단말서버로 동작하는 방식이다. [그림 2]는 이러한 단말 서버형 모델의 구성도이다. 이는 매우 소규모 컨퍼런스 시스템에만 적합한 방식이다.

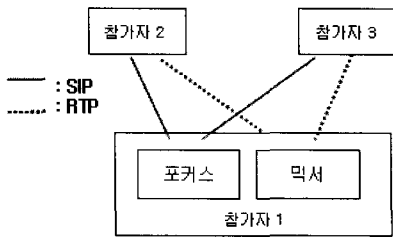


그림 2. 단말 서버형 모델

3. 미디어 서버 분리형 모델

이 모델은 [그림 3]과 같이 믹서 기능을 가진 미디어 서버를 별도로 두는 방식이다. 이와 같이 하여 중앙 서버의 부하를 줄일 수 있으나 별도의 서버를 추가해야 하는 부담이 따르게 된다.

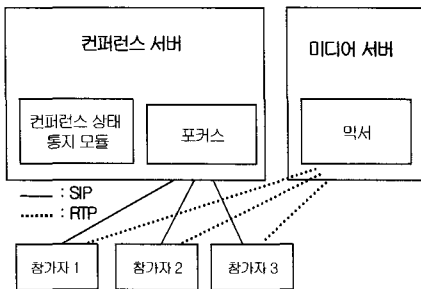


그림 3. 미디어 서버 분리형 모델

4. 믹서 분산형 모델

이 모델은 [그림 4]와 같이 중앙 서버에서는 포커스와 컨퍼런스 정책 모듈 및 컨퍼런스 상태 통지 모듈만 두고 미디어 믹서 기능은 각 단말에 분산시켜 동작하는

방식이다. 이 방식에서는 컨퍼런스 서버의 부하 중 큰 비중을 차지하는 미디어 믹서 기능이 분산되어 서버의 부하를 줄이는 장점이 있지만 모든 사용자들 사이에 미디어 스트림 전송을 위한 RTP 세션 연결과 이에 대한 처리가 필요하다는 단점이 있다.

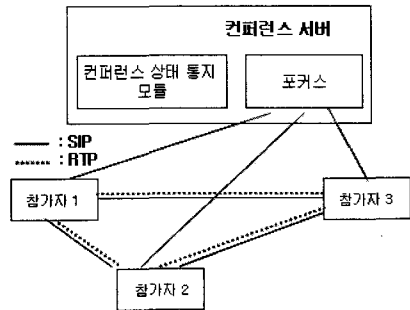


그림 4. 믹서 분산형 모델

III. 제안된 혼성형 모델의 구현

본 논문에서 제안하는 새로운 혼성형 모델은 컨퍼런스 서버와 참가자 사이에 오가는 SIP 시그널링 메시지 양을 줄이고 서버의 미디어 믹서 부하를 경감시켜 중앙 집중형 방식의 단점을 개선하였으며 대규모 컨퍼런스로의 확장이 용이하도록 하였다.

1. 구조

본 논문에서 제안하는 혼성형 모델의 구조는 [그림 5]와 같다. 여기서는 컨퍼런스 참가자 수가 일정 한도를 넘으면 서버는 특정 기능을 가진 참가자를 선택하여 이 참가자로 하여금 미디어 믹서 기능 및 컨퍼런스 상태 이벤트 패키지의 통지 기능을 분담하도록 한다. 참여자 중 일부는 서버에 컨퍼런스 상태 이벤트 패키지를 등록할 때 SUBSCRIBE 메시지의 Contact 헤더에 'is coordinator' 항목을 추가함으로써 이와 같은 기능의 보유를 알리게 된다. 서버는 컨퍼런스 정보 데이터베이스에 이를 표시하며 나중에 컨퍼런스 정보 데이터베이스에서 이 항목을 기초로 해서 기능을 분담할 참가자를 선정할 수 있게 된다. 만일 컨퍼런스 참가자가 더 증가

할 경우 서버는 이와 같은 서버의 기능을 분담할 수 있는 기능을 가진 참가자를 추가로 선정할 수 있어 확장성을 높일 수 있게 된다.

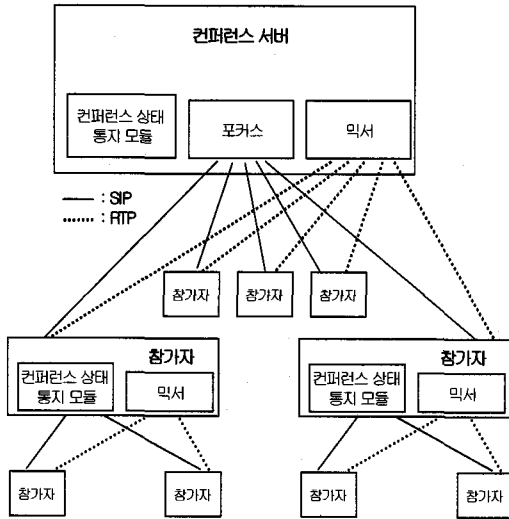


그림 5. 제안하는 혼성형 모델

기능을 분담할 참가자가 선정되면 새로운 참가자의 등장이나 기존 참가자의 탈퇴 등 컨퍼런스 상태 변화가 생겼을 때 기존의 방식에서는 서버의 컨퍼런스 상태 통지 모듈에서 전체 참가자들에게 모두 컨퍼런스 통지 메시지를 전송하였으나 본 구조에서는 컨퍼런스 통지 메시지를 자신과 직접 연결된 참가자 및 기능을 분담할 참가자에게만 전송하며 이를 받은 쪽에서 나머지 참가자들에게 컨퍼런스 통지 메시지를 전송하게 된다. 이와 같은 방식으로 기존 모델에 비해 서버의 부하를 줄일 수 있다.

또한, 이후 새롭게 추가되는 컨퍼런스 참가자들에 대해서는 서버에서 이 기능을 분담할 참가자의 컨퍼런스 URI로 접속하도록 알려줌으로써 미디어 믹서 기능의 자연스러운 분산이 이루어지게 되어 역시 서버의 부하를 줄일 수 있게 된다. 참가자가 더 증가 할 경우에는 서버의 기능을 분담할 참가자가 추가로 선정되어 위와 같은 서버의 부담을 줄이는 과정이 반복된다.

이와 같은 기능을 가지는 컨퍼런스 서버의 내부 구성은 [그림 6]과 같다.

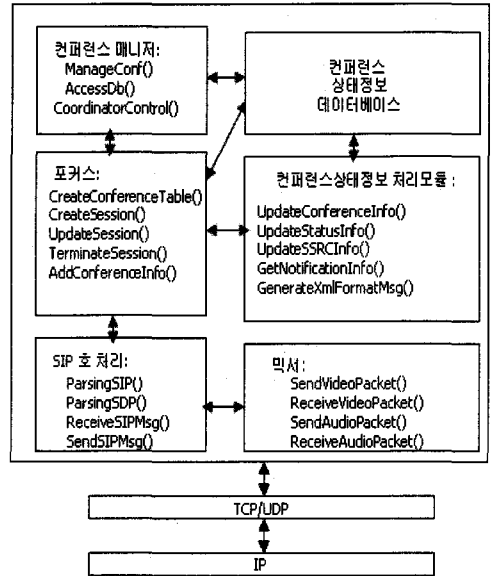


그림 6. 컨퍼런스 서버의 내부 구성

컨퍼런스 서버의 동작은 참가자가 INVITE 메시지를 통하여 컨퍼런스에 참여하는 다이얼인(dial-in) 방식을 기준으로 하였다. 여기서 컨퍼런스 매니저 부분에서는 현재 참여자 수를 계속 보고 있다가 미리 정한 최대치를 넘으면 application/conference-info+xml 데이터 포맷[9]에 맞추어 설계된 컨퍼런스 상태 정보 데이터베이스에서 참가자 각각에 대한 정보를 담고 있는 컨퍼런스 상태 정보 요소인 <user>의 하위 요소 <roles>의 <status> 항목에 해당하는 데이터베이스 테이블 필드 값이 'coordinator' 인 참가자들을 찾아 이들 중 하나를 서버 기능을 분담할 참가자로 선정한다.

이후 포커스 부분에서는 새로운 참가자의 INVITE 요청이 들어오면 Contact 헤더에 이와 같이 하여 선정된 참가자의 주소를 담은 SIP 응답코드 302 (Redirection) 메시지를 전송함으로써 새로운 참가자가 서버 역할을 분담할 선정된 참가자에 INVITE 요청 메시지를 다시 보내도록 한다. 이 과정을 통해 RTP 세션 연결이 성립되어 양자 간에 미디어 전송이 시작된다.

[그림 6]에서 컨퍼런스 상태 정보 처리 모듈 부분은 등록 및 통지 기능을 가진다. 등록 기능에서는 참가자로 부터의 SUBSCRIBE 메시지의 Contact 헤더에 'iscoordinator' 항목이 있으면 컨퍼런스 상태 정보를

데이터베이스에 저장할 때 컨퍼런스 상태 정보의 요소인 <user>의 하위요소 <roles>의 <status> 항목에 해당하는 데이터베이스 테이블 필드에 'coordinator'를 추가한다.

통지 기능에서는 평소에는 컨퍼런스 상태 정보에 변동이 생겼을 경우 서버에 연결된 모든 참가자에게 컨퍼런스 상태 정보 통지 메시지를 전송한다. 그러나 이를 통한 메시지 전송량은 참가자 수가 늘어날수록 계속 증가하므로 서버의 부하 증가 요인이 된다. 컨퍼런스 참가자 수가 미리 정한 최대치를 넘게 되면 부하를 분담할 특정 참가자 시스템이 선정되므로 이후 발생하는 컨퍼런스 상태 정보 통지 메시지는 현 서버에 연결된 참가자 및 부하 분담을 위해 선정된 특정 참가자에게만 전송된다. 이 메시지를 받은 선정된 참가자 시스템에서는 자신과 연결된 모든 다른 참가자들에게 이 메시지를 대신 전송하게 된다.

컨퍼런스 서버의 부하를 분담할 수 있는 기능을 갖추도록 구현한 참가자 시스템의 UA 내부 구성은 [그림 7]과 같다. 여기서 컨퍼런스 상태 정보 처리 모듈은 등록 및 통지 기능을 가지는데 등록 기능에서는 컨퍼런스 서버에 SUBSCRIBE 메시지를 보낼 때 Contact 헤더에 'iscoordinator' 항목을 추가함으로써 서버에게 자신이 기능을 분담할 수 있는 참가자임을 알린다. 이 부분은 사용자 인터페이스에서 이 기능을 끄고 켤 수 있도록 하였다. 그냥 단순한 참가자로만 참여하고 싶은 경우는 이 기능을 꺼서 SUBSCRIBE 메시지 생성 시 일반적인 Contact 헤더를 사용하게 하면 된다. 이 모듈의 통지 기능에서는 서버로부터 수신한 NOTIFY 메시지의 바디 부분인 application/conference-info+xml 데이터 포맷의 컨퍼런스 상태 정보의 요소인 <user>의 속성인 entity 값이 자신을 가리키고 <user>의 하위요소인 <roles>에 <status>coordinator</status> 항목이 있는지 여부를 조사하여 만약 있으면 이후 서버의 기능을 분담하는 동작 모드로 들어간다.

이 모드에서는 서버로부터 컨퍼런스 상태 정보 통지 메시지를 수신하면 이를 자신과 연결된 참가자들에게 서버를 대신하여 전송한다. 믹서 부분은 서버의 기능을 분담하는 동작 모드로 들어간 이후부터 자신에게

INVITE 메시지를 보낸 참가자들과 RTP 세션 연결을 맺어 이들과의 미디어 데이터 처리를 담당한다.

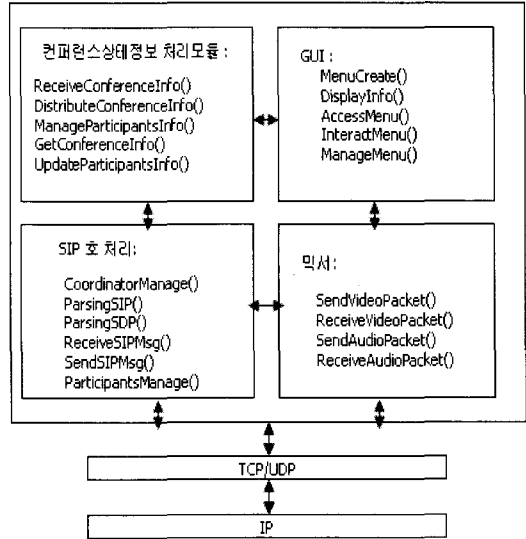


그림 7. 참가자 시스템 UA 내부 구성

2. 호 신호 처리 절차

구현된 모델에서의 호 신호 처리 절차는 [그림 8]과 같다. 여기서는 이미 컨퍼런스 서버에 여러 참가자가 연결되어 있다고 가정한다. 참가자 F가 컨퍼런스 참여를 위해 서버에 INVITE 메시지를 보낸다. 이때 서버가 내부적으로 설정한 처리 가능 참가자 수를 넘게 되면 서버는 데이터베이스에서 서버 기능을 분담할 수 있는 참가자 목록을 조사하여 그 중 하나(참가자 E)에게 NOTIFY 메시지를 보낸다. 이 메시지에 들어있는 컨퍼런스 상태 정보에는 <user>의 하위요소인 <roles>에 <status>coordinator</status> 항목을 넣어 서버의 기능을 분담하는 동작 모드로 들어가도록 알린다. 상대방(참가자 E)으로부터 SIP 응답코드 200 OK 메시지를 받은 서버는 Contact 헤더에 이 참가자 E의 주소를 넣은 SIP 응답코드 302(Redirection) 메시지를 참가자 F에게 보낸다. 참가자 F는 이를 받아 새로운 INVITE 요청을 참가자 E의 주소로 하여 미디어 전송을 위한 RTP 세션 연결이 양자 간에 이루어진다.

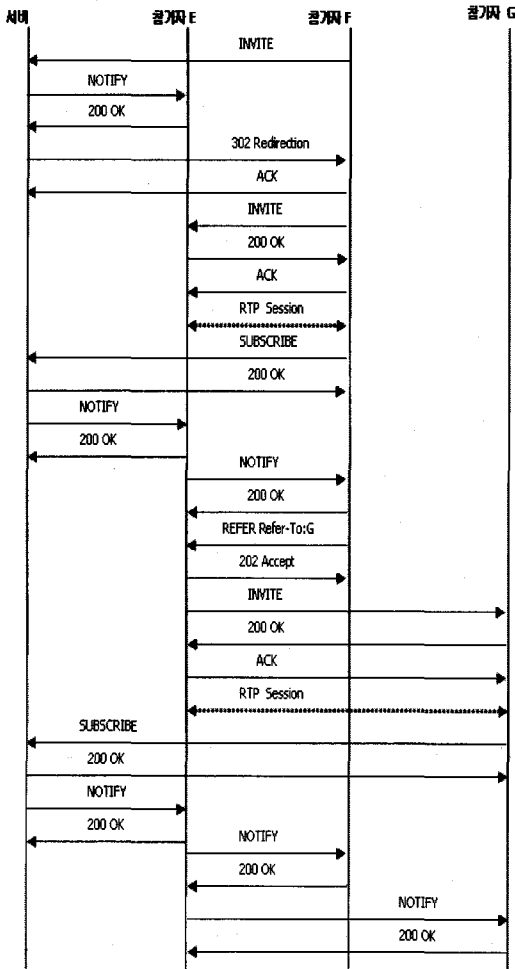


그림 8. 구현된 모델에서의 호 신호 처리 절차

이후 참가자 F는 컨퍼런스 이벤트 패키지 등록을 위하여 SUBSCRIBE 메시지를 서버에 보내고 서버는 변경된 컨퍼런스 상태 정보를 NOTIFY 메시지에 담아 참가자 E에게 보낸다. 서버 역할을 분담하는 참가자 E는 이를 받아 참가자 F에게 다시 전송한다. 참가자 F가 참가자 G를 컨퍼런스에 참여시키기 원할 때에는 Refer-To 헤더에 참가자 G의 주소를 담은 REFER 메시지를[10] 서버 역할을 하는 참가자 E에게 보낼 수 있다. 참가자 E는 참가자 G에게 INVITE 요청 메시지를 보내 컨퍼런스에 초청하고 서로 간에 RTP 세션을 맺을 수 있게 된다.

이후 참가자 G는 역시 SUBSCRIBE 메시지를 서버에 보내고 서버는 변경된 컨퍼런스 상태 정보를 NOTIFY 메시지에 담아 참가자 E에게 보내며, 참가자 E는 이를 받아 참가자 F와 참가자 G에 다시 전송한다. 참가자 수가 계속 증가할 시에는 서버는 데이터베이스에서 서버의 기능을 분담할 수 있는 새로운 참가자를 또 선택하여 위의 과정을 반복하게 됨으로써 대용량 컨퍼런스로의 확장이 유연하게 이루어질 수 있다. 만약 서버기능을 분담하는 참가자가 컨퍼런스에서 탈퇴하고자 할 때는 Refer-To 메시지를 사용해 자신이 관리하던 참가자들의 주소를 서버에 알려주며 서버는 이를 이용해 이들 참가자들에 직접 INVITE 메시지를 보내어 새로운 RTP 연결을 설정하여 이들이 계속 컨퍼런스에 참여할 수 있게 된다.

IV. 성능 분석

성능 분석을 위하여 구현한 컨퍼런스 서버의 각 모듈과 참가자 시스템의 UA는 모두 자바로 작성하였고 운영체제로는 컨퍼런스 서버는 리눅스(커널 2.4)를 사용하였으며 참가자 시스템의 UA는 MS 윈도우 XP를 각각 사용하였다. 서버의 사양은 CPU 펜티엄 IV 1.8GHz, 메인메모리 256MB이다. 실험을 위한 링크로서 컨퍼런스 서버와 각 참가자들은 모두 100Mbps LAN에 연결되었다. 미디어 데이터 전송을 위한 음성 코덱으로는 8KHz 샘플주파수에 샘플 당 8비트로 코딩되는 PCM(Pulse Coded Modulation)을 사용하고 영상 코덱으로는 클럭 주파수 90KHz인 RTP 프로파일 상에서의 페이로드 타입 26번을 사용하였다.

컨퍼런스 참가자 수의 증가에 따른 지연 시간을 비교 측정하기 위해 기존의 방식대로 서버가 전체 참가자를 모두 처리하는 경우와 참가자 수가 4명 이상이 되면 앞에서 제안한대로 컨퍼런스 서버로 하여금 부하를 분담할 참가자를 선정하여 동작하는 경우를 서로 비교하였다. 결과는 [그림 9]와 같으며 여기서 기존의 방식에 비해 제안된 방식이 참가자 수가 4명일 때 6.5%, 참가자 수가 5명일 때는 8.1%, 그리고 참가자 수가 6명일 경우

에는 11.0%의 지연시간이 개선됨을 보여주고 있다.

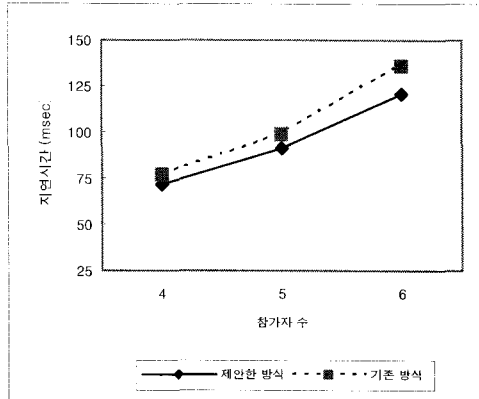


그림 9. 구현된 모델에서의 지연시간 비교

실험 결과에서 참가자 수가 증가할수록 지연시간의 개선 효과가 커짐을 알 수 있으며 이는 참가자 수가 증가할수록 믹서 기능의 분담 및 컨퍼런스 상태 이벤트 패키지 통지 기능 개선에 대한 효과가 크게 나타난다고 볼 수 있다. 제안된 방식에서 컨퍼런스 상태 이벤트 패키지의 등록 시에는 서버 측에서의 이점이 없으나 참가자 수가 많을 경우 통지 과정에서의 메시지 양을 크게 줄일 수 있게 된다. 그러나 참가자의 일부이지만 이와 같이 서버의 기능을 분담할 수 있는 참가자 시스템의 UA 구현이 기존의 UA에 비해 복잡해지는 단점이 있다.

V. 결론

본 논문에서는 SIP 기반의 효율적인 혼성 형태의 컨퍼런스 모델을 제안하고 이를 설계 및 구현하였다. 이 모델에서 서버는 컨퍼런스 참가자 수가 일정 한도를 넘으면 미디어 믹서 기능 및 컨퍼런스 상태 이벤트 패키지의 통지 기능을 분담할 수 있는 참가자를 선택하여 이로 하여금 믹서 기능 및 컨퍼런스 상태 정보의 통지 기능을 분담하도록 하였고 이 과정에 대한 호 신호 처리 절차를 보였다.

제안하는 모델은 참가자 수가 계속 증가하게 되면 서버의 기능을 분담할 기능을 가진 참가자도 여러 개로

늘어날 수 있으므로 대규모 컨퍼런스로의 확장이 용이하다. 참가자 중 일부는 서버에 컨퍼런스 상태 이벤트 패키지를 등록할 때 SUBSCRIBE 메시지의 Contact 헤더에 'iscoordinator' 항목을 추가함으로써 이와 같은 기능의 보유를 알리게 되고 서버는 컨퍼런스 정보 데이터베이스에 이를 표시하여 기능을 분담할 참가자를 선정할 수 있게 된다. 서버의 기능을 분담하는 동작 모드로 들어간 참가자는 서버로부터 컨퍼런스 상태 정보 통지 메시지를 수신하면 이를 자신과 연결된 참가자들에게 서버를 대신하여 전송하며 자신에게 INVITE 메시지를 보낸 참가자들과 RTP 세션 연결을 맺어 이들과의 미디어 데이터 처리를 담당한다.

이와 같이 하여 서버와 참가자 사이에 오가는 SIP 시그널링 메시지 양을 줄일 수 있고 서버의 미디어 믹서 부하를 경감시켜 실제 멀티미디어 데이터가 오가는 컨퍼런스 상황에서 참가자 수가 증가하는데 따르는 전체적인 지연 시간을 감소시킬 수 있음을 실험을 통하여 보였다. 추후 서버 기능을 분담하는 참가자가 컨퍼런스에서 탈퇴하고자 할 때의 효율적인 부하 분담 처리 방법에 대한 연구가 필요하다.

참고 문헌

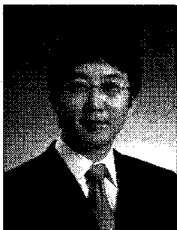
- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [2] I. Dalgic and H. Fang, "Comparison of H.323 and SIP for IP Telephony Signaling", Proc. of Photonics East, Boston, Massachusetts, September 1999.
- [3] F. Fingal and P. Gustavsson, "A SIP of IP-telephony", <http://www.cs.columbia.edu/~hgs/sip/papers.html>, February 1999.
- [4] J. Rosenberg, "A Framework for Conferencing with Session Initiation Protocol(SIP)", RFC 4353, February 2006.

- [5] I. Miladinovic and J. Stadler, "Multiparty Conference Signaling using the Session Initiation Protocol(SIP)", in International Network Conference, July 2002.
- [6] K. Singh, G. Nair and H. Schulzrinne, "Centralized conferencing using SIP", in Internet Telephony Workshop 2001, New York, April 2001.
- [7] P. Koskelainen and H. Khartabil, "Requirements for Conference Policy Control Protocol", draft-ietf-xcon-cpcp-reqs-04, August 2004.
- [8] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.
- [9] J. Rosenberg, H. Schulzrinne and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.
- [10] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

저자 소개

조 현 규(Hyun-Gyu Jo)

정회원



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

• 2006년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 계약교수

<관심분야> : SIP, VoIP, 실시간 인터넷 통신

이 기 수(Ki-Soo Lee)

정회원

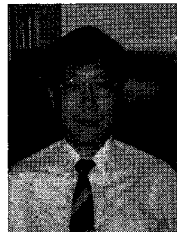


- 1979년 2월 : 경북대학교 전자공학과(공학사)
- 1982년 2월 : 서울대학교 대학원(공학석사)
- 1982년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

<관심분야> : 디지털시스템, 데이터베이스

장 춘 서(Choon-Seo Jang)

정회원



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

• 1981년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

<관심분야> : SIP, 임베디드 시스템, 인터넷텔레포니