SIP 환경에서의 효율적인 분산형 컨퍼런스 구조

Efficient Distributed Conference Architecture 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) 환경에서의 컨퍼런스 모델 가운데 중앙 집중형 구조는 컨퍼런스 관리 및 제어가 용이한 장점이 있어 널리 사용되나 컨퍼런스 참가자 수가 늘어남에 따른 확장성의 제약이 있다. 따라서 본 논문에서는 중앙 집중형 컨퍼런스 모델의 확장성을 개선할 수 있는 효율적인 분산형 컨퍼런스 구조를 제안하였다. 여기서는 컨퍼런스 참가자가 정해진 최대값을 넘을 경우 동적으로 새로운 컨퍼런스 서버가 추가된다. 이때 기존의 컨퍼런스 서버의 포커스는 주 포커스가 되고 새로 참여한 컨퍼런스 서버의 포커스는 부 포커스가 되며, 각 컨퍼런스 서버들 사이에 컨퍼런스 참가자들에 대한 동적인 재 할당이 이루어져 부하를 균등하게 분담할 수 있다. 이 과정은 컨퍼런스 참가자 수가 증가함에 따라 반복된다. 본 논문에서는 이러한 동작에 필요한 새로운 컨퍼런스 서버 추가 과정에서의 처리 절차, 교환되는 SIP 호 신호, 컨퍼런스 서버 사이의 RTP(Real Time Transport Protocol) 세션 연결을 위한 신호 처리 절차, 그리고 컨퍼런스 서버 사이의 컨퍼런스 이벤트 패키지 신호 처리 절차를 제시하였다. 제안한 방식은 실험을 통하여 성능을 측정하였다.

■ 중심어 : | SIP | 컨퍼런스 | 컨퍼런스 이벤트 패키지 |

Abstract

The centralized conference architecture, one of the conference architectures in SIP(Session Initiation Protocol) environment, is widely used as it has the advantage of conference management and control. However it has been limited in scalability. Therefore we have proposed an efficient distributed conference architecture to improve scalability of centralized conference model. In our architecture, if the number of conference participants exceeds the predefined maximum number, a new conference server is added to the conference dynamically. In this case, the focus of existing server acts as primary focus and the focus of added server acts as secondary focus, and dynamic reallocation of participants between servers is done to equally divide the loads. This process is repeated as the number of conference participants increases. For this behavior, we have proposed procedure of adding the conference server, SIP call signal exchange, signaling procedure for RTP(Real Time Transport Protocol) sessions between conference servers, and procedure of conference event package between conference servers. The performance of our proposed model is evaluated by experiments.

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

* 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.
접수번호 : #080114-002
접수일자 : 2008년 01월 14일
심사완료일 : 2008년 02월 14일
교신저자 : 조현규, e-mail : blackjo@kumoh.ac.kr

I. 서 론

IETF(Internet Engineering Task Force)에서 제안한 SIP(Session Initiation Protocol)[1] 기반의 컨퍼런스 방식은 포커스(focus)와 믹서의 위치에 따라 중앙 집중형, 단말 서버형, 미디어 서버 분리형, 믹서 분산형 모델 등으로 분류할 수 있다[2]. 포커스는 전체 컨퍼런스 참가자들과의 SIP 신호 연결을 유지하고 관리하는 기능을 하고 믹서는 컨퍼런스 참가자들 사이에 오디오/비디오 패킷을 분배하고 전달한다.

이들 모델 중 중앙에 컨퍼런스 서버가 위치하여 컨퍼런스 전체를 제어하고 관리하는 중앙 집중형 모델은 컨퍼런스의 관리와 서비스가 용이한 장점이 있어 가장 많이 사용되지만 컨퍼런스의 수와 참여자의 수가 증가함에 따라서 컨퍼런스 서버의 부하가 커지고 확장성에 제약을 받는다[3]. 기존에 이를 해결하기 위해서 믹서 기능을 분산시키거나 시그널링 처리 관련 부하를 감소시키는 방법 등 여러 방안이 제시되고 있으나 아직 컨퍼런스 서버를 분산시켜 처리하는 방법에 대한 단일 표준안은 정해져 있지 않다[4-6].

따라서 본 논문에서는 중앙 집중형 컨퍼런스 모델의 확장성을 개선할 수 있는 중소규모의 효율적인 분산형 컨퍼런스 구조를 제안하였으며 이를 통해 컨퍼런스 진행 중 참가자가 어느 한도를 넘을 경우 동적으로 새로운 컨퍼런스 서버의 추가 및 각 컨퍼런스 서버가 담당하는 참가자 수의 균등 분배 작업이 이루어지도록 하였다.

제안한 구조에서는 포커스와 믹서를 갖춘 최초의 컨퍼런스 서버가 컨퍼런스를 운영하다가 참가자 수가 정해진 최대값을 넘으면 현재 컨퍼런스에 참가하고 있는 부 컨퍼런스 서버들이 있는지 조사한 후 없으면 새로운 컨퍼런스 서버를 추가하는 과정으로 들어간다. 새로운 컨퍼런스 서버가 참여하면 기존의 컨퍼런스 서버의 포커스는 주 포커스가 되고 새로 참여한 컨퍼런스 서버의 포커스는 부 포커스가 되며, 각 컨퍼런스 서버들 사이에 컨퍼런스 참가자들에 대한 동적인 재 할당이 이루어진다.

컨퍼런스 서버들 사이의 정보 교환을 위해서는 확장된 컨퍼런스 이벤트 패키지를 사용하였고 컨퍼런스 참가자 수의 변동에 따라 주 포커스는 동적으로 새로운 컨퍼런스 서버를 더 추가할 수도 있으며 기존의 컨퍼런스 서버를 뺄 수도 있도록 하여 중소규모에서 최적의 컨퍼런스 운영이 가능하도록 하였다.

본 논문에서는 이러한 동작에 필요한 새로운 컨퍼런스 서버 추가 과정에서의 처리 절차 및 이때 교환 되는 SIP 호 신호들과 컨퍼런스 서버 사이의 RTP(Real Time Transport Protocol)[7] 세션 연결을 위한 신호 처리 절차, 그리고 컨퍼런스 서버 사이의 컨퍼런스 이벤트 패키지[8] 신호 처리 절차를 제시하였다. 또한 구현된 시스템에 대한 성능 분석을 위하여 참가자 수를 증가시켜가면서 평균 지연시간을 측정하였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 관련 연구로서 기존의 중앙 집중형 컨퍼런스 모델 및 믹서 분산형 모델에 대해 설명하고 Ⅲ장에서는 본 논문에서 제안하는 시스템의 설계 및 구현과 신호 처리 절차를 설명하고, Ⅳ장에서는 실험을 통하여 본 논문에서 제안한 시스템에 대한 성능 분석을 한 후 Ⅴ장에서 결론을 맺는다.

II. 관련연구

1. 중앙 집중형 컨퍼런스 모델

중앙 집중형 컨퍼런스 모델은 하나의 컨퍼런스 서버에 포커스와 믹서를 두고 전체 컨퍼런스 참가자를 제어하는 방식이며 명확한 컨퍼런스 관리 및 제어가 가능하다는 장점이 있다[2]. 그러나 컨퍼런스의 수와 참가자의 수가 증가함에 따라서 서버의 부하가 커지므로 확장성의 제약을 가지고 있다.

[그림 1]은 이러한 중앙 집중형 컨퍼런스 모델이며 여기서 컨퍼런스 포커스는 참가자들과 서버 사이에 컨퍼런스 세션에 대한 설정, 변경 및 종료를 관리하고 컨퍼런스에서 발생하는 각종 정보들을 컨퍼런스 이벤트 패키지로 전달한다. 믹서는 오디오/비디오 패킷 스트림을 수신해 이를 조합하고 전체 참가자들에게 분배하며 각 참가자들은 서버의 믹서와 일대일 RTP 세션 연결을 맺는다.

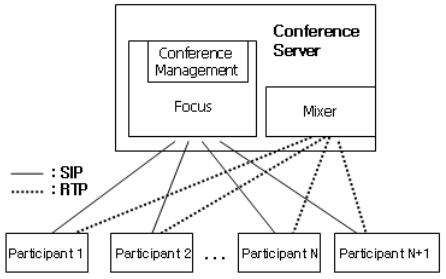

그림 1. 중앙 집중형 컨퍼런스 모델

2. 믹서 분산형 컨퍼런스 모델

중앙 집중형 컨퍼런스 모델의 제약인 확장성을 높이기 위한 여러 연구가 기존에 행해져왔다[4-6]. 이들 연구들은 대부분 믹서 기능을 가진 미디어 서버를 분산시키는 방법, 미디어 믹서 기능을 참가자 시스템에 분산시키는 방식, 또는 시그널링 처리와 관련된 서버 부하를 감소시키는 방안을 제시하고 있다.

그러나 이들 모델들은 전체 컨퍼런스 참가자들과의 SIP 신호 연결을 유지 및 제어하고 각종 컨퍼런스 정보를 관리하는 포커스가 단일 포커스 구조이므로 컨퍼런스 참가자 수가 늘어남에 따라 포커스에서 처리해야 하는 메시지 양의 증가로 인해 컨퍼런스 확장성에 제약을 주게 된다.

본 논문에서는 이와 같은 기존 컨퍼런스 모델의 제약을 개선하기 위하여 중소규모 컨퍼런스에 적합하도록 컨퍼런스 참가자 수의 변동에 따라 동적으로 컨퍼런스 서버를 구성하여 믹서 기능의 분산만이 아니라 컨퍼런스 포커스의 기능도 분산시킬 수 있는 방안을 제시하였다.

III. 시스템 설계 및 구현

1. 제안된 분산형 컨퍼런스 구조

본 논문에서 제안한 분산형 컨퍼런스 구조는 [그림 2]와 같다. 각 컨퍼런스 서버는 포커스와 믹서를 갖추고 있으며 최초의 컨퍼런스 서버의 포커스는 컨퍼런스를 식별할 수 있는 목적지 주소인 컨퍼런스 URI를 가진다. 이 포커스가 주 포커스로 동작하며 주 포커스를 가진 컨퍼런스 서버가 주 컨퍼런스 서버가 된다.

참가자는 INVITE 메시지에 이 컨퍼런스 URI를 넣어 보냄으로써 컨퍼런스 포커스와 컨퍼런스 세션을 맺게 된다. 주 포커스는 컨퍼런스 참가자 수를 지속적으로 조사하다가 정해진 최대값을 넘으면 새로운 컨퍼런스 서버를 추가하는 과정으로 들어간다.

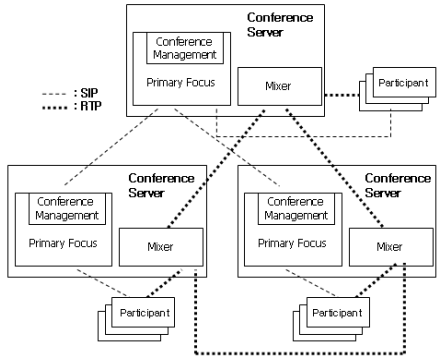

그림 2. 제안된 분산형 컨퍼런스 구조

[그림 3]은 이러한 과정의 순서도이며 동작 과정은 다음과 같다. 먼저 새로운 참가자가 INVITE 메시지를 주 포커스에게 보내면 내부 데이터베이스를 통한 인증 과정을 거친 후 이 참가자를 포함한 전체 참가자 수가 주 컨퍼런스 서버가 담당할 수 있는 최대값보다 큰지 여부를 조사한다. 만약 크지 않으면 200 OK 응답메시지를 보내고 새로운 참가자는 ACK 메시지를 다시 보내어 서로 RTP 세션이 맺어지고 컨퍼런스에 참가하게 된다.

전체 참가자 수가 최대값보다 크면 주 포커스는 현재 컨퍼런스에 참가하고 있는 부 컨퍼런스 서버들이 있는지 조사한다. 참가중인 부 컨퍼런스 서버가 없으면 새로운 컨퍼런스 서버에게 INVITE 메시지를 보내어 주 포커스와 부 포커스 사이에 SIP 다이얼로그를 생성하고 또한, 양 서버의 믹서들 사이에도 RTP 세션을 생성한다. 다음, 새로운 참가자에게 Contact 헤더에 부 포커스의 주소를 담은 302 Redirection 메시지를 보내어 이 부 포커스에 대해 다시 INVITE 메시지를 보내도록 한다. 이때 부 포커스가 응답하여 새로운 참가자는 부 포

커스를 통하여 컨퍼런스에 참가하게 된다.

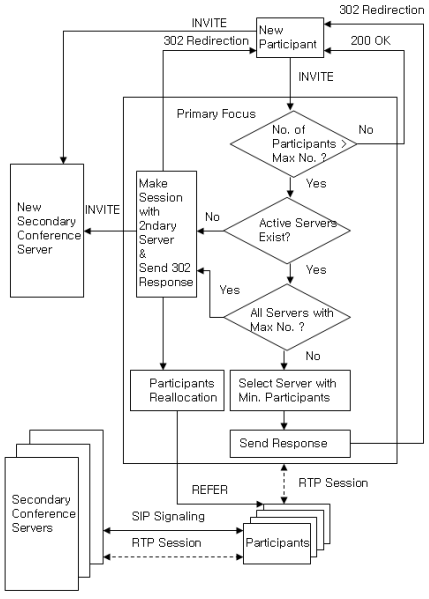

그림 3. 새로운 컨퍼런스 서버 추가 과정 순서도

참가중인 부 컨퍼런스 서버가 있으면 이들 서버들 에게 연결된 참가자 수가 모두 최대치인지 여부를 조사한다. 현재 모두 최대치의 참가자가 연결되어 있으면 위의 새로운 컨퍼런스 서버에게 INVITE 메시지를 보내어 컨퍼런스에 참가시키는 동작으로 들어간다. 그렇지 않은 경우에는 이들 중 연결된 참가자 수가 가장 적은 서버를 선택하여 새로운 참가자에게 Contact 헤더에 선택된 부 포커스의 주소를 담은 302 Redirection 메시지를 보내어 이 부 포커스에 대해 다시 INVITE 메시지를 보내도록 한다. 이때 해당 부 포커스가 응답하여 새로운 참가자는 부 포커스를 통하여 컨퍼런스에 참가하게 된다.

이후 각 포커스들 사이에 컨퍼런스 참가자들에 대한 동적인 재 할당이 이루어진다. 이 과정에서 새로운 컨퍼런스 서버가 담당하는 참가자 수가 정해진 최소값보다 작을 경우 최소값에 도달할 때 까지 기존의 컨퍼런스 서버가 담당하던 참가자를 재 할당시킨다. 재 할당 과정에서는 담당하는 참가자 수가 많은 서버부터 새로운 컨퍼런스 서버로 할당시키고 새로운 컨퍼런스 서버가 담당하는 정해진 최소값에 도달하면 재 할당 과정은 종료된다.

기존의 컨퍼런스 서버가 담당하던 참가자를 다른 서버에게 할당시킬 때는 REFER[9] 메시지를 사용한다. 주 포커스는 재 할당 대상으로 선정된 참가자들에게 이동할 서버의 주소를 Refer-To 헤더에 담은 REFER 메시지를 전송하며 이를 받은 참가자들은 새로운 서버와 연결을 맺고 기존의 컨퍼런스 서버에 BYE 메시지를 보내어 연결을 종료하게 된다. 이후 컨퍼런스 참가자 수의 변동에 따라 주 포커스는 동적으로 새로운 컨퍼런스 서버를 더 추가할 수도 있고 기존의 컨퍼런스 서버를 뺄 수도 있다.

2. 분산형 컨퍼런스 구조에서의 SIP 신호 절차

[그림 4]는 새로운 컨퍼런스 서버 추가 과정에서의 SIP 신호 처리 절차이며 이미 참가자A는 주 포커스를 가진 컨퍼런스 서버와 연결되어 있고 이 상태에서 참가자B를 REFER 메시지를 사용해 새로 컨퍼런스에 참여시키는 과정이다.

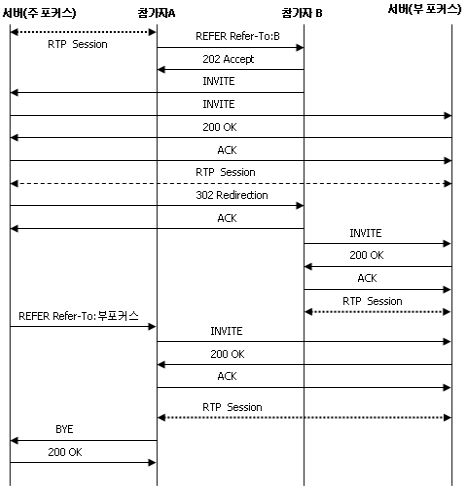

그림 4. 새로운 컨퍼런스 서버 추가 과정에서의 신호 처리 절차

동작 과정은 다음과 같다. 참가자B는 REFER 메시지를 받은 후 주 포커스를 가진 컨퍼런스 서버에 INVITE 메시지를 보낸다. 이때 주 포커스는 자신이 처리 가능

한 최대 참가자 수를 넘었는지 확인한 후 만약 넘게 되면 데이터베이스에 저장된 주위의 컨퍼런스 서버 목록 및 그 서버가 현재 담당하고 있는 참가자 수를 조사한다. 만약 새로운 컨퍼런스 서버를 추가할 필요가 있으면 이 서버에게 SIP INVITE 메시지를 보내어 연결을 맺는다. 이때 상대방 컨퍼런스 서버의 포커스는 부 포커스로 동작하게 된다.

주 포커스를 가진 컨퍼런스 서버는 이후 참가자B에게 302 Redirection 메시지를 보내고 참가자B는 이에 따라 새로 추가된 컨퍼런스 서버에게 INVITE 메시지를 보내어 컨퍼런스에 참가하게 된다.

다음, 주 포커스를 가진 컨퍼런스 서버는 참가자들에 대한 동적인 재 할당 과정으로 들어간다. 이 경우 추가된 컨퍼런스 서버가 담당하는 참가자 수가 정해진 최소값보다 작으면 주 포커스를 가진 컨퍼런스 서버는 참가자A에게 추가된 컨퍼런스 서버 주소를 Refer-To 헤더에 담은 REFER 메시지를 보내어 추가된 컨퍼런스 서버와 연결을 맺도록 한다.

참가자A는 기존 컨퍼런스 서버와의 RTP 연결을 끊고 추가된 컨퍼런스 서버와 미디어 스트림 교환을 위한 RTP 연결을 새롭게 맺으며 기존 컨퍼런스 서버에 BYE 메시지를 보내어 SIP 다이얼로그를 종료한다. 이 과정은 필요하면 다른 참가자들에 대해서도 반복되며 이를 통해 컨퍼런스 서버들 간에 담당하는 참가자 수의 균등 분배를 실현하여 분산 처리의 효과를 높일 수 있다.

[그림 4]에서 보면 새로운 컨퍼런스 서버가 추가되는 과정에서 주 포커스를 가진 컨퍼런스 서버와 미디어 스트림 전송을 위한 RTP 연결이 맺어지는데 이때 이미 동작중인 부 컨퍼런스 서버들이 있으면 이들과도 RTP 연결이 맺어져야 전체 컨퍼런스 참가자들 사이의 미디어 스트림 분배가 가능하다. [그림 5]는 이 과정에 필요한 신호 처리 절차이다.

[그림 5]에서 주 포커스를 가진 컨퍼런스 서버와 부 포커스B와 부 포커스C를 가진 컨퍼런스 서버의 믹서사이에는 RTP 연결이 이미 되어있다. 이때 참가자 수의 증가로 인해 부 포커스D를 가진 컨퍼런스 서버가 새롭게 추가된다.

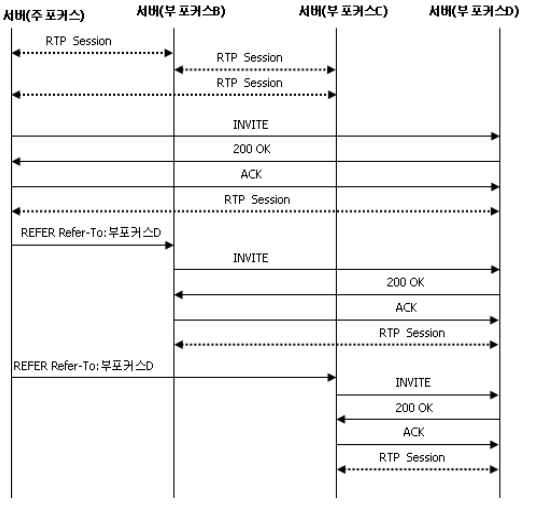

그림 5. 컨퍼런스 서버 사이의 RTP 세션 연결을 위한 신호 처리 절차

추가되는 과정에서 주 포커스를 가진 컨퍼런스 서버와는 RTP 연결이 이루어지나 나머지 서버와는 아직 RTP 연결이 없으므로 주 포커스는 나머지 두 개 서버에 각각 부 포커스 D를 가리키는 REFER 메시지를 보내어 부 포커스D를 가진 컨퍼런스 서버의 믹서와 RTP 연결을 생성하도록 한다.

주 포커스는 컨퍼런스 이벤트 패키지를 통하여 부 포커스들과 컨퍼런스 관련 정보를 서로 교환한다. 또한 주 포커스는 컨퍼런스 이벤트 패키지를 등록한 컨퍼런스 참가자들에게도 컨퍼런스 관련 정보를 NOTIFY[10] 메시지를 이용하여 전송한다.

본 논문에서는 포커스와 포커스 사이의 컨퍼런스 정보 교환을 위하여 컨퍼런스 정보 문서 포맷인 application/conference-info+xml[8]의 구조를 확장하여 user 요소의 속성으로 포커스임을 나타내기 위한 isfocus 속성이 추가되었고, conference-state 요소의 하위 요소로 각 포커스가 현재 담당하는 사용자 수를 나타내는 current-user-count 요소가 추가되었다.

컨퍼런스 생성 초기에 부 포커스들은 주 포커스에게 SUBSCRIBE 메시지를 이용하여 컨퍼런스 이벤트 패키지를 등록하고 주 포커스도 부 포커스 각각에 대해 자신을 등록한다. 이후 주 포커스는 새로운 참가자가

들어오면 모든 부 포커스들에게 NOTIFY 메시지를 사용하여 통보한다. 또, 주 포커스는 컨퍼런스 이벤트 패키지를 등록한 전체 컨퍼런스 참가자들에게도 이를 알린다. 부 포커스가 관리하는 참가자가 컨퍼런스를 종료하면 부 포커스는 이를 NOTIFY 메시지로 주 포커스에게 알리며 이때 주 포커스는 나머지 부 포커스들과 컨퍼런스 참가자들에게 이를 알린다. [그림 6]은 이러한 과정에 필요한 신호 처리 절차이다.

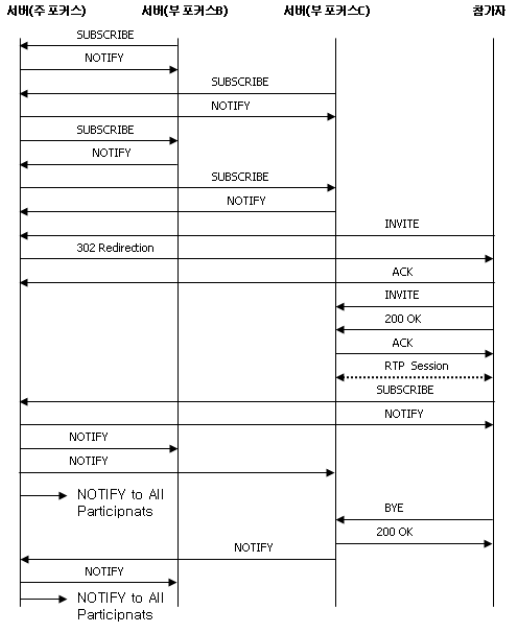

그림 6. 컨퍼런스 서버 사이의 컨퍼런스 이벤트 패키지 신호 처리 절차

[그림 6]에서 3개의 컨퍼런스 서버가 서로 간에 SUBSCRIBE 메시지를 사용하여 컨퍼런스 이벤트 패키지를 등록하고 있으며 여기서는 신호 처리 절차를 간략히 표시하기 위해 SUBSCRIBE와 NOTIFY 메시지에 대한 200 OK 응답 메시지는 모두 생략하였다.

신호 처리 절차는 다음과 같다. 새로운 참가자가 주 포커스에게 INVITE 메시지를 보내면 주 포커스는 이 참가자를 부 포커스C에게 담당하게 한다. 이 참가자는 새로운 INVITE 메시지를 부 포커스C에게 보내어 컨퍼런스에 참가한 후 주 포커스에게 SUBSCRIBE 메시지를 보내어 등록한다.

주 포커스는 이 참가자에게 SUBSCRIBE 메시지에 대한 응답으로 현재 컨퍼런스에 참가중인 사용자들에 관한 정보를 담아 NOTIFY 메시지를 보내고 이어 부 포커스B와 부 포커스C에도 새로운 참가자 정보를 담은 NOTIFY 메시지를 보낸다. 또한 등록한 나머지 전체 참가자들에도 NOTIFY 메시지를 보내어 새로운 참가자가 있음을 알린다.

이후 이 참가자가 컨퍼런스에서 탈퇴하면 이 참가자를 담당하던 부 포커스C는 주 포커스에게 NOTIFY 메시지로 이를 알리며 주 포커스는 다시 부 포커스B와 등록한 나머지 전체 참가자들에도 이를 알린다.

Ⅳ. 성능 분석

본 논문에서 제안한 분산형 컨퍼런스 구조의 성능을 측정하기 위하여 PC상에 리눅스(커널 2.6)를 설치한 두 대의 컨퍼런스 서버와 윈도우즈 XP를 설치한 8대의 클라이언트(컨퍼런스 참가자)로 컨퍼런스 환경을 구성하였다.

PC의 사양은 모두 펜티엄 IV, CPU 2.4GHz, 메인메모리 512MB이며 성능 측정을 위한 링크로써 100Mbps LAN에 연결하였고 동일한 IP 서브넷 상에 위치하도록 하였다. 오디오와 비디오 인코딩은 8KHz PCM과 JPEG_RTP를 사용하였다. 두 대의 컨퍼런스 서버중 하나는 주 포커스와 믹서를 가지고 다른 하나는 부 포커스와 믹서를 가진다. 컨퍼런스 서버와 클라이언트의 각 처리 모듈은 자바로 작성하였다.

실험 방법으로는 컨퍼런스 참가자 수를 증가시켜 가면서 기존의 중앙 집중형 컨퍼런스 모델 구조를 사용한 경우의 평균 지연시간과 본 논문에서 제안한 컨퍼런스 모델 구조를 사용한 경우의 평균 지연시간을 비교 측정하였다.

본 논문에서 제안한 컨퍼런스 모델 구조의 경우 [표 1]과 같이 컨퍼런스 참가자 수의 증가에 따라 부 포커스를 가진 컨퍼런스 서버와의 사용자 분담이 이루어지는 것으로 하였다. 즉, 새로운 컨퍼런스 서버를 추가하

는 기준인 참가자 수의 최대값은 4이고 추가되는 컨퍼런스 서버의 참가자 수의 최소값은 2로 하였다.

표 1. 컨퍼런스 서버 사이의 참가자 수의 분배

전체참가자 수	4	5	6	7	8
주 포커스 분담참가자수	4	3	3	4	4
부 포커스 분담참가자수	0	2	3	3	4

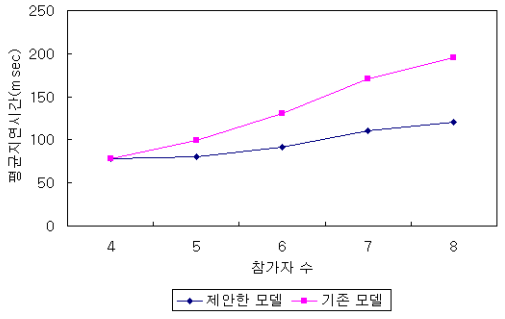

그림 7. 제안한 컨퍼런스 모델과 기존 모델과의 평균 지연시간 비교

[그림 7]은 실험 측정 결과이며 컨퍼런스 참가자가 4명일 때까지는 컨퍼런스 서버가 추가되지 않으므로 기존의 중앙 집중형 컨퍼런스 모델과 동일하게 동작한다. 그러나 참가자가 5명이 되면 주 포커스에 의해 컨퍼런스 서버가 추가되며 이 추가되는 컨퍼런스 서버의 포커스는 부 포커스로 동작하고, 최소값이 2이므로 참가자 재 할당 과정을 거쳐 각각 3명, 2명으로 부하가 분산된다. 이때 측정 결과로써 평균 지연 시간은 19%감소하였다.

참가자가 6명이 되면 부 컨퍼런스 서버는 3명의 참가자를 담당하게 되고 이때 평균 지연시간은 30%감소하였고 참가자가 8명일 때는 평균 지연시간이 38%감소하였다. 이는 서버 추가로 인한 서버들 사이의 통신 트래픽에 의한 영향보다는 각 서버가 처리하는 미디어 패킷의 분산 처리로 인한 효과가 더 크기 때문으로 보인다. 측정 결과로 참가자가 수가 증가할수록 평균 지연시간의 개선 효과가 커짐을 알 수 있다.

V. 결론

본 논문에서는 중앙 집중형 컨퍼런스 모델의 확장성을 개선할 수 있는 효율적인 분산형 컨퍼런스 구조를 위하여 컨퍼런스 참가자가 어느 정해진 최대값을 넘을 경우 동적으로 새로운 컨퍼런스 서버의 추가 및 각 컨퍼런스 서버가 담당하는 참가자 수의 균등 분배 작업이 이루어지도록 하였다.

제안한 방식에서는 포커스와 믹서를 갖춘 최초의 컨퍼런스 서버는 참가자 수가 최대값을 넘게 되면 새로운 컨퍼런스 서버를 추가하며 이때 기존의 컨퍼런스 서버의 포커스는 주 포커스가 되고 새로 참여한 컨퍼런스 서버의 포커스는 부 포커스 동작하고, 부하 분담을 위해 참가자들을 컨퍼런스 서버들에게 동적으로 재 할당시킨다.

컨퍼런스 서버들 사이의 정보 교환을 위해서는 확장된 컨퍼런스 이벤트 패키지를 사용하였고 새로운 컨퍼런스 서버 추가 과정에서의 처리 절차 및 교환되는 SIP 호 신호들과 컨퍼런스 서버 사이의 RTP세션 연결을 위한 신호 처리 절차를 제시하였다.

제안한 시스템의 성능 측정은 컨퍼런스 참가자 수를 증가시켜가면서 기존의 중앙 집중형 컨퍼런스 모델 구조와 평균 지연시간을 비교하였고 실험을 통해 제안한 모델에서 성능 개선 효과가 있음을 확인하였다. 향후 과제로는 컨퍼런스 참가자 수를 더욱 확대하여 여러 개의 컨퍼런스 서버가 추가되는 과정에서의 성능 비교와 교환되는 SIP 시그널링 메시지 양을 줄이기 위한 연구가 추가로 필요하다.

참 고 문 헌

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "Session Initiation Protocol," RFC 3261, 2002(6).

[2] J. Rosenberg, "A Framework for Conferencing with Session Initiation Protocol(SIP)," RFC

4353, 2006(2).

[3] K. Singh, G. Nair, and H. Schulzrinne, "Centralized conferencing using SIP," in Internet Telephony Workshop 2001, New York, 2001(4).

[4] R. V. Prased, R. Hurni, and H. Jamadagni, "A Scalable Distributed VoIP using SIP," Proc. 8th IEEE ISCC, pp.608-613, 2003(6).

[5]http://www.ikn.tuwien.ac.at/ftw-a1/WSEAS2002_ConferenceSignalingOptimization.pdf

[6] 조현규, 이기수, 장춘서, "SIP 환경에서의 중앙 집중형 컨퍼런스 모델 기반의 새로운 서비스 모델에 관한 연구", 한국콘텐츠학회논문지, 제6권, 제2호, pp.17-26, 2006.

[7] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, 2003(7).

[8] J. Rosenberg, H. Schulzrinne, and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State," RFC 4575, 2006(8).

[9] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method," RFC 3515, 2003(4).

[10] A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, 2002(6).

저 자 소 개

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, 실시간 인터넷, 임베디드 시스템

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, 실시간 인터넷, 임베디드 시스템