韓國 컴퓨터情報學會 論文誌
第13卷 第7號, 2008. 12.

2008-13-7-3-8

분산형 컨퍼런스 환경에서의 효율적인 컨퍼런스 이벤트 패키지 처리 방식

장 춘 서**, 조 현 규*, 이 기 수**

A Method of Efficient Conference Event Package Processing in Distributed Conference Environment

Choon-Seo Jang**, Hyun-Gyu Jo*, Ky-Soo Lee**

요 약

중앙 집중형 컨퍼런스 모델은 컨퍼런스 관리 및 제어가 용이한 장점이 있으나 컨퍼런스 사용자 수가 증가할수록 성능의 저하가 커지므로 확장성에 제약이 있다. 따라서 중앙 집중형 컨퍼런스 모델의 확장성을 개선할 수 있는 새로운 분산형 컨퍼런스 모델들이 최근 제안되고 있다. 이러한 분산형 컨퍼런스 구조에서는 컨퍼런스 사용자가 정해진 최대값을 넘을 경우 동적으로 새로운 컨퍼런스 서버가 추가되며, 본 논문에서는 이러한 분산형 컨퍼런스 환경에서 주 컨퍼런스 서버가 담당해야 할 컨퍼런스 이벤트 패키지의 처리 방식을 효율화 시킬 수 있는 새로운 방식을 제안하였다. 주 컨퍼런스 서버는 부 컨퍼런스 서버들 및 사용자들과 컨퍼런스 이벤트 패키지를 사용해 정보를 교환하며, 자신이 관리하는 컨퍼런스 정보 데이터베이스에서 컨퍼런스 사용자들에 대한 통지 기능을 분담할 SIP(Session Initiation Protocol) UA(User Agent)들을 선정하여 이들의 목록을 각 컨퍼런스 서버들에게 전송한다. 이를 수신한 각 컨퍼런스 서버들은 선정된 UA들이 컨퍼런스 이벤트 패키지 처리를 분담하도록 하여 컨퍼런스 서버에서의 SIP 신호 처리 부하를 줄일 수 있으며 분산형 컨퍼런스 모델에서의 확장성을 개선시킨다. 제안된 시스템의 성능은 실험을 통하여 분석하였다.

Abstract

The centralized conference model has advantage of conference management and control, however it's scalability has been limited as performance degrades largely with increasing number of conference users. So new distributed conference models which improve scalability of centralized conference model have been suggested recently. In the distributed conference model, as conference users exceed a predefined maximum number, a new conference server is added to the conference dynamically. In this paper, We have proposed a new method which increases efficiency of conference event package processing that primary conference server should charge in the distributed conference environment. The primary conference server exchanges informations with each secondary conference servers and conference users by using conference event package. And from the conference information database it selects SIP(Session Initiation Protocol) UA(User Agent) which will share notification to the conference users, and transfers lists to each conference servers. The conference servers make the selected UAs share processing of conference event package, so loads of SIP signal processing decrease, and improve scalability of distributed conference model. The performance of our proposed model is evaluated by experiments.

▸ Keyword : SIP(Session Initiation Protocol), 컨퍼런스이벤트패키지(Conference Event Package), 분신형 컨퍼런스모델(Distributed Conference Model)

• 제1저자 : 장춘서
• 접수일 : 2008. 10. 28, 심사일 : 2008. 12. 16, 심사완료일 : 2008. 12. 24.
* 금오공과대학교 컴퓨터공학과 계약교수 ** 금오공과대학교 컴퓨터공학과 교수
※ 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

Ⅰ. 서 론

SIP((Session Initiation Protocol)[1] 기반의 컨퍼런스 모델 중에서 하나의 컨퍼런스 서버가 컨퍼런스 전체를 제어하고 관리하는 중앙 집중형 모델[2][3][4]은 컨퍼런스 기능의 제어가 쉬운 장점을 가지므로 많이 사용되지만 컨퍼런스의 규모가 커질수록 하나의 컨퍼런스 서버에서 처리해야할 부하가 커져 확장성에 제약을 받는다. 이를 해결하기 위한 여러 방식 중 서버 분산형 컨퍼런스 모델[5][6]은 컨퍼런스 진행 중 참가자가 어느 한도를 넘을 경우 동적으로 새로운 컨퍼런스 서버의 추가 및 각 컨퍼런스 서버가 담당하는 참가자 수의 균등 분배 작업이 이루어지게 되어 확장성을 높일 수 있다.

그러나 분산형 컨퍼런스 구조에서 컨퍼런스 정보를 각 컨퍼런스 서버들과 컨퍼런스 참여자들 사이에 주고받기위해 사용되는 컨퍼런스 이벤트 패키지[7]의 처리는 주 컨퍼런스 서버에 집중되는 문제점이 있고 이는 결국 컨퍼런스 확장성에 제한요소로 작용하게 된다.

본 논문에서는 이러한 문제점을 해결하기 위하여 분산형 컨퍼런스 환경에서 주 컨퍼런스 서버가 담당해야 할 컨퍼런스 이벤트 패키지의 처리를 효율적으로 할 수 있는 새로운 방식을 제안하였다. 여기에서 각 부 컨퍼런스 서버들 및 사용자들은 컨퍼런스 이벤트 패키지를 사용해 주 컨퍼런스 서버와 정보를 교환하며, 주 컨퍼런스 서버는 자신이 관리하는 컨퍼런스 정보 데이터베이스에서 전체 컨퍼런스 사용자들에 대한 통지 기능을 분담할 SIP UA들을 선정하여 이들의 목록을 각 컨퍼런스 서버들에게 전송한다. 이를 수신한 각 컨퍼런스 서버들은 선정된 UA들이 컨퍼런스 이벤트 패키지 처리를 위한 부하를 분담하도록 한다. 따라서 컨퍼런스 서버에서의 SIP 신호 처리 부하를 줄일 수 있으며, 사용자 수가 증가할수록 부하 감소 효과가 커지게 되어 컨퍼런스 확장성을 높일 수 있게 된다.

이와 같은 기능을 제공하기 위하여 본 논문에서 새롭게 확장된 컨퍼런스 이벤트 패키지 포맷과 이를 이용한 신호 처리 절차를 설계하고 제시하였다. 또한 컨퍼런스 사용자 수를 증가시켜 가면서 처리 시간을 측정하여 구현한 시스템에 대한 성능 분석을 하였다.

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

Ⅱ. 관련연구

2.1 중앙 집중형 컨퍼런스 모델

그림 1의 중앙 집중형 컨퍼런스 모델은 하나의 컨퍼런스 서버가 컨퍼런스 전체를 제어하고 관리하는 구조이다. 이는 컨퍼런스 기능의 제어가 쉬운 점 때문에 많이 사용되지만 컨퍼런스의 규모가 커질수록 하나의 컨퍼런스 서버에서 처리해야할 작업량이 많아져 확장성에 제약을 받게 된다. 컨퍼런스 포커스는 컨퍼런스 세션의 설정 등 컨퍼런스 관련 각종 제어 기능을 제공하며 이들 동작은 SIP 호 신호 기반으로 이루어진다. 믹서는 오디오/비디오 패킷 스트림을 조합하고 전체 참가자들에게 분배하는 기능을 제공한다.

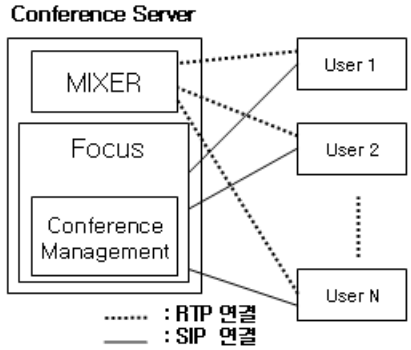

그림 1. 중앙 집중형 컨퍼런스 모델
Fig.1. Centralized Conference Model

2.2 분산형 컨퍼런스 모델

그림 2의 분산형 컨퍼런스 모델은 컨퍼런스 참가자가 증가할 경우 동적으로 새로운 컨퍼런스 서버가 추가되는 구조이다. 각 컨퍼런스 서버는 포커스와 믹서를 갖추고 있으며 복수개의 컨퍼런스 서버들이 참가자들을 서로 나누어 처리함으로써 컨퍼런스의 확장성을 높일 수 있다.

주 포커스는 컨퍼런스 참가자 수가 정해진 값보다 커지면 새로운 컨퍼런스 서버를 추가하고 추가된 컨퍼런스 서버의 포커스는 부 포커스가 된다. 주 포커스는 동적으로 변화하는 컨퍼런스 정보 및 전체 참여자의 정보를 컨퍼런스 이벤트 패키지 형태로 처리하며 컨퍼런스 사용자자 수의 증가에 따라 이

부분에 대한 부하도 아울러 증가하게 되어 컨퍼런스 확장성을 제한하는 요소로 작용하게 된다.

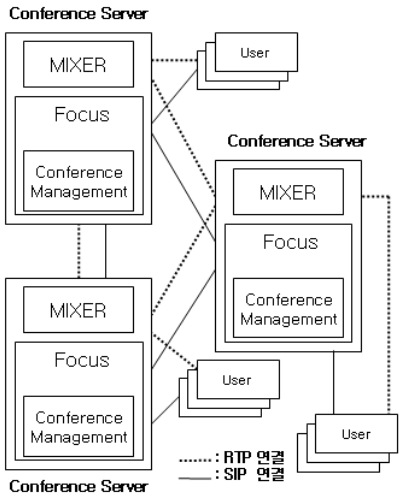

그림 2. 분산형 컨퍼런스 모델
Fig. 2. Distributed Conference Model

2.3 컨퍼런스 이벤트 패키지

컨퍼런스 이벤트 패키지는 동적으로 변화하는 컨퍼런스 정보를 통지(Notification) 메시지를 통해 사용자들에게 제공한다. 이를 위해서는 먼저 컨퍼런스 포커스에게 자신을 등록(Subscription)하는 과정이 필요하다.

컨퍼런스 정보는 application/conference-info+ xml[8] 포맷인 XML 문서로 나타내어진다. 이 문서는 〈conference-info〉를 최상위 요소로 태그로 가지며 이 요소의 속성으로는 entity, version과 state가 있다. 속성 entity는 컨퍼런스 URI 값을 나타내고, 속성 version은 수신된 통지 메시지를 순서대로 정렬하기 위해 사용된다. 속성 state는 컨퍼런스 정보가 전체 내용을 포함하는 상태인지 또는 변화된 부분만 나타내는 상태인지와 현 시점에서의 컨퍼런스의 존재 여부를 나타낸다. 컨퍼런스 사용자들에 대한 개별 정보는 〈users〉 요소의 하위 요소인 〈user〉에 포함되어 있다. 〈user〉 요소의 속성으로는 컨퍼런스 사용자에 대한 URI를 나타내는 entity가 있고, 각 사용자들이 컨퍼런스에 참여하면서 사용하는 디바이스 및 SIP 신호 세션에 대한 정보를 나타내는 〈endpoint〉가 하위 요소로 있다. 이 〈endpoint〉 요소의 하위 요소로는 컨퍼런스 서버와의 미디어 스트림 정보를 나타내는 〈media〉와 디바이스의 연결 상태를 나타내는 〈status〉 등이 온다.

III. 시스템 설계 및 구현

3.1 분산형 컨퍼런스 모델 설계

본 논문에서 설계한 분산형 컨퍼런스 모델에서 컨퍼런스 서버는 포커스와 믹서로 구성되며 주 컨퍼런스 서버의 포커스가 주 포커스로 동작하여 새로운 컨퍼런스 서버들을 컨퍼런스에 추가할 수 있는 기능을 가진다.

주 포커스는 컨퍼런스 참가자 수가 일정 한도 이상 증가하면 새로운 컨퍼런스 서버를 추가한다. 그림 3은 이 과정에서의 SIP 신호 처리 구현 절차이다. 사용자 A는 컨퍼런스의 참여를 위하여 주 포커스를 가진 주 컨퍼런스 서버에게 INVITE 요청 메시지를 보낸다. 이후 사용자 B가 주 컨퍼런스 서버에게 INVITE 요청 메시지를 보내는 경우 주 포커스는 주 컨퍼런스 서버가 감당할 수 있는 사용자 수를 넘었다고 판단하여 이에 대한 응답을 보류하고 대신 새로운 컨퍼런스 서버에게 INVITE 요청 메시지를 보낸다. 이 요청 메시지에 대해 상대방 서버가 응답을 하면 서로 간에 RTP(Real-time Transport Protocol)[9] 연결을 맺게 되고 상대방 컨퍼런스 서버의 포커스는 이때부터 부 포커스로 동작하게 된다.

새로운 컨퍼런스 서버의 추가 후 주 포커스는 사용자 B의 INVITE 요청 메시지에 대한 응답으로 상태 코드 번호가 302이고 새로운 컨퍼런스 서버를 가리키는 리디렉션 SIP 메시지를 보낸다. 이후 사용자 B는 새로 추가된 컨퍼런스 서버에게 다시 INVITE 요청 메시지를 보내고 서버는 이에 대해 응답함으로써 컨퍼런스에 참가한다.

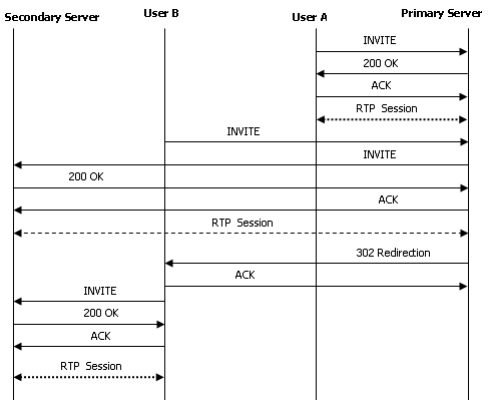

그림 3. 새로운 컨퍼런스 서버 추가 과정에서의 SIP 신호 처리 절차
Fig. 3. SIP Signaling Procedure while New Conference Server added

컨퍼런스 서버들 사이의 사용자들에 대한 동적인 할당은 다음과 같은 과정으로 이루어진다. 주 컨퍼런스 서버는 사용자 수가 자신이 처리할 수 있는 최대값보다 커질 경우 부하 분담을 위하여 동작중인 다른 컨퍼런스 서버들을 컨퍼런스 정보 데이터베이스에서 찾는다. 컨퍼런스 정보 데이터베이스는 서버들 사이의 컨퍼런스 이벤트 패키지 메시지의 교환에 의해서 주 컨퍼런스 서버가 생성하고 관리한다. 동작중인 서버들이 있으면 이들 서버들 중 여유 용량이 가장 큰 서버를 선택하여 컨퍼런스 참가 요청을 한 사용자에게 이 서버의 주소를 넣은 SIP 리디렉션 응답 메시지(SIP 상태코드 302)를 보낸다. 이 메시지를 받은 사용자는 해당 서버에 대해 다시 INVITE 메시지를 보내고 이때 해당 서버가 이에 응답함으로써 컨퍼런스에 참가하게 된다.

동작중인 다른 컨퍼런스 서버들이 없거나 서버들의 여유 용량이 없을 경우에는 새로운 컨퍼런스 서버의 추가과정으로 들어간다. 이를 위하여 새로운 컨퍼런스 서버에게 INVITE 메시지를 보내어 주 컨퍼런스 서버와 부 컨퍼런스 서버 사이에 SIP 세션을 생성하고 미디어 스트림 전달을 위한 RTP 세션도 양단간에 생성한다. 주 컨퍼런스 서버는 이 새로운 서버의 주소를 넣은 SIP 리디렉션 응답 메시지를 새로운 사용자에게 보내고 이 사용자는 새로운 서버에 대해 다시 INVITE 메시지를 보내어 컨퍼런스에 참가하게 된다.

다음으로 컨퍼런스 서버들이 담당하는 사용자 수를 균등 분배하기 위한 동적인 재 할당과정이 이루어진다. 이에 대한 동작으로써 새로운 컨퍼런스 서버가 담당하는 사용자 수가 정해진 값에 도달할 때 까지 기존의 컨퍼런스 서버가 담당하던 사용자를 이 서버에게 재 할당시킨다. 이 과정에서, 담당하는 사용자 수가 많은 서버의 사용자부터 새로운 컨퍼런스 서버에게 할당시키게 된다. 재 할당을 위하여 주 컨퍼런스 서버는 이동할 서버의 주소를 넣은 SIP REFER 메시지를 선택된 사용자에게 보내고 이에 대해 해당 사용자는 이동할 서버에게 INVITE 메시지를 보내어 세션을 맺고 기존의 서버에게는 BYE 메시지를 보내어 연결을 종료하게 된다. 이후 새로운 컨퍼런스 서버가 담당하는 사용자 수가 정해진 값에 도달하면 재 할당 과정은 종료되고 주 컨퍼런스 서버는 이 내용을 컨퍼런스 정보 데이터베이스에 갱신한다.

3.2 제안된 컨퍼런스 이벤트 패키지 처리

주 컨퍼런스 서버 및 부 서버들과 사용자들 사이에 컨퍼런스 정보를 주고받기위해서 컨퍼런스 이벤트 패키지가 사용된다. 각 컨퍼런스 서버들은 자신들의 컨퍼런스 정보 및 자신들이 담당하는 사용자들에 대한 컨퍼런스 정보를 컨퍼런스 이벤트 패키지 형태로 주 컨퍼런스 서버에게 보내며 주 컨퍼런스 서버는 컨퍼런스 서버들이 보내온 정보와 각 컨퍼런스 사용자들이 보내온 정보를 모두 받아 컨퍼런스 정보 데이터베이스를 구축한다.

컨퍼런스 서버의 동작을 위하여 컨퍼런스 이벤트 패키지의 정보 포맷에서 다음 사항들을 확장하였다. 먼저 최상위 요소인 〈conference-info〉의 하위 요소인 〈conference-description〉에서 〈maximum-servers〉 요소를 추가하여 컨퍼런스 서버의 최대 개수를 나타내었다. 이 정보는 주 포커스가 컨퍼런스 서버를 추가 할 때 사용한다. 또 〈conference-state〉 요소의 하위 요소로 〈maximum-users〉를 추가하였다. 이 정보는 각 컨퍼런스 서버가 담당할 수 있는 최대 사용자 수를 나타낸다. 또한 〈current-user-count〉 요소를 추가하여 각 컨퍼런스 서버가 현재 담당하는 사용자 수를 나타내도록 하였다.

컨퍼런스 사용자를 위해서는 컨퍼런스 이벤트 패키지의 정보 포맷에서 다음 사항들을 확장하였다.

먼저 〈conference-info〉의 하위 요소인 〈user〉 요소의 속성으로 〈notify_agent〉를 추가하였다. 이 속성은 컨퍼런스 서버가 보내온 컨퍼런스 이벤트 패키지의 SIP NOTIFY 메시지를 다른 컨퍼런스 사용자들에게 대신 전달해 줄 수 있는 기능을 나타낸다. 그리고〈ua_focus〉 속성은 SIP UA가 포커스임을 나타내기 위한 속성이며, 〈agent_capacity〉 속성은 NOTIFY 메시지를 다른 참가자들에게 전달해줄 때의 처리 능력을 나타낸다.

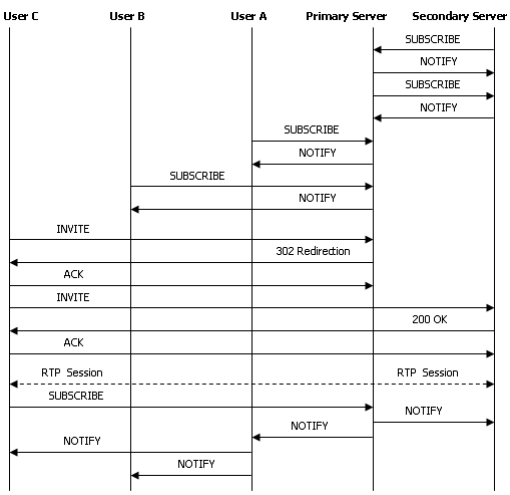

그림 4. 제안된 컨퍼런스 이벤트 패키지 동작
Fig. 4. Operation of Suggested Conference Event Package

그림 4는 제안된 컨퍼런스 이벤트 패키지의 동작이다. 주 컨퍼런스 서버는 SIP SUBSCRIBE 메시지와 NOTIFY 메시지를 사용해 다른 부 컨퍼런스 서버들과 컨퍼런스 정보를 교환한다. 사용자들도 주 컨퍼런스 서버에게 컨퍼런스 이벤트 패키지를 등록한다. 여기서는 그림 표현의 복잡성을 줄이기 위해 SUBSCRIBE 메시지와 NOTIFY 메시지에 대한 응답 메시지는 모두 생략하였다. 주 컨퍼런스 서버는 컨퍼런스 정보 데이터베이스를 생성하고 관리한다.

사용자 C가 컨퍼런스의 참여를 위해 주 컨퍼런스 서버에게 INVITE 메시지를 보내면 주 컨퍼런스 서버는 컨퍼런스 정보 데이터베이스를 참조한 후 이 사용자를 부 컨퍼런스 서버에게 연결시키기 위해 상태 코드 번호가 302인 리디렉션 SIP 응답 메시지를 보낸다. 사용자 C는 부 컨퍼런스 서버에게 다시 INVITE 메시지를 보내어 RTP 세션을 맺고 컨퍼런스에 참여하게 된다. 이후 사용자 C는 주 컨퍼런스 서버에게 SUBSCRIBE 메시지를 보내어 컨퍼런스 이벤트 패키지 등록을 한다.

주 컨퍼런스 서버는 각 부 컨퍼런스 서버들에게 이를 통지하고 모든 등록된 사용자들에게도 이를 통지하는 절차로 들어가며, 이 과정에서 각 컨퍼런스 서버들은 자신이 담당하는 사용자들 중에서 컨퍼런스 참가자들에 대한 통지 기능을 분담할 UA들을 선정한다. 이는 주 컨퍼런스 서버에게 받은 컨퍼런스 정보에서 〈conference-info〉의 하위 요소인 〈user〉 요소의 속성으로 〈notify_agent〉가 부여된 목록을 조사함으로써 가능하다. 각 컨퍼런스 서버들은 선정된 UA에게 NOTIFY 메시지를 보내고 이 메시지를 받은 UA는 컨퍼런스 서버를 대신하여 다른 사용자들에게 NOTIFY 메시지를 보내게 된다. 그림 4에서는 주 컨퍼런스 서버가 담당하는 사용자 A가 이에 해당한다.

Ⅵ. 성능 분석

본 논문에서 제안한 방식의 성능 분석을 위한 컨퍼런스 환경은 컨퍼런스 서버 두 대와 컨퍼런스 사용자 10명으로 구성하였다. 컨퍼런스 서버는 커널 2.6 리눅스를 설치한 PC이며 사양은 펜티엄 IV 2.4GHz CPU와 DDR2 SDRAM 512MB 메인메모리이다. 컨퍼런스 참가자용 PC는 위와 동일한 사양이며 윈도우즈 XP를 사용하였다. 연결된 LAN 속도는 100Mbps 이며 모두 동일한 LAN 세그먼트 상에 위치한다.

두 대의 컨퍼런스 서버중 하나는 주 컨퍼런스 서버로 동작하며 다른 하나는 부 컨퍼런스 서버가 된다. 제안된 방식의 성능 분석을 위하여 사용자 수를 증가시켜가면서 주 컨퍼런스 서버가 전체 사용자에게 컨퍼런스 정보 통지를 완료하는데 소요되는 시간을 측정하여 비교하였다. 이를 위하여 다음 표 1과 같이 컨퍼런스 서버 사이의 사용자 수가 분배되는 것으로 하였다.

여기서는 사용자 수 6명부터 주 컨퍼런스 서버이외에 새로운 컨퍼런스 서버가 추가되는 것으로 하였고, 새로운 컨퍼런스 서버 추가 시 기존 컨퍼런스 서버와의 사용자 균등 분담을 위한 재조정이 이루어지는 것으로 하였다.

표 1. 컨퍼런스 서버에서의 사용자 분배
Table 1. User distribution of Conference Servers

전체 사용수	6	7	8	9	10
주 컨퍼런스 서버 담당 명수	3	4	4	5	5
부 컨퍼런스 서버 담당 명수	3	3	4	4	5

그림 5는 제안된 방식을 사용했을 경우 사용자 수를 증가시켜 가면서 측정한 주 컨퍼런스 서버에서의 컨퍼런스 정보 통지 메시지 처리 시간이다.

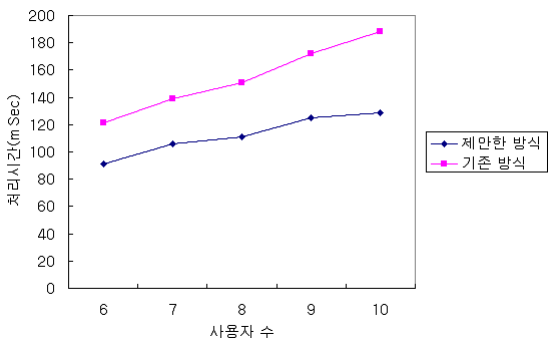

그림 5. 컨퍼런스 정보 통지 메시지 처리 시간
Fig. 5. Conference Notification Message Processing Time

기존 방식에서는 전체 사용자에게의 통지 처리를 주 컨퍼런스 서버가 모두 담당하지만 제안된 방식에서는 컨퍼런스 정보 데이터베이스에서 선정된 UA가 이를 분담한다. 실험 환경의 제약 상 컨퍼런스 사용자 수를 크게 증가시키기가 힘들므로 여기서는 각 컨퍼런스 서버당 하나의 UA가 통지 처리를 분담하는 것으로 하였다.

측정 결과 사용자 수가 6명일 때는 기존 방식에 비해 24.8%의 성능 향상을 얻을 수 있었으며, 7명일 때는

23.7%, 8명일 때는 26.5% 등, 사용자 수가 증가 할수록 31.4% 까지의 시간 감소를 얻을 수 있었다. 실험 결과 컨퍼런스 사용자 수가 많아질수록 제안된 방식에서의 성능 향상이 커짐을 알 수 있다.

V. 결론 및 향후 과제

본 논문에서는 분산형 컨퍼런스 환경에서 컨퍼런스 이벤트 패키지의 처리를 효율적으로 할 수 있는 새로운 방식을 제안하여 컨퍼런스 모델에서의 확장성을 개선하였다.

제안된 방식에서 주 컨퍼런스 서버는 컨퍼런스 이벤트 패키지를 사용해 부 컨퍼런스 서버들 및 사용자들과 컨퍼런스 정보를 교환하고 컨퍼런스 정보 데이터베이스를 생성하고 관리한다. 컨퍼런스 정보의 변동으로 이를 통지해야 할 경우, 주 컨퍼런스 서버는 전체 컨퍼런스 사용자들에 대한 통지 기능을 분담할 SIP UA들을 선정하여 이들의 목록을 각 컨퍼런스 서버들에게 전송하며, 각 컨퍼런스 서버들은 선정된 UA들이 컨퍼런스 이벤트 패키지 처리를 위한 부하를 분담하도록 하여 SIP 신호 처리 부하를 줄일 수 있도록 하였다. 이를 위해 본 논문에서는 새롭게 확장된 컨퍼런스 이벤트 패키지 포맷에 의한 신호 처리 절차를 설계하여 제시하였고, 실제 구현된 시스템에 대한 성능 분석을 통하여 처리 시간을 줄일 수 있음을 보였다.

향후 과제로는 실험 측정을 위한 컨퍼런스 규모를 더욱 크게 하여 복수개의 부 컨퍼런스 서버들을 둠으로써 각 이들 서버들에 미치는 부하 경감 효과들도 아울러 측정할 필요가 있다.

참고문헌

〔1〕 J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "Session Initiation Protocol," RFC 3261, June 2002.

〔2〕 J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol (SIP)," RFC 4353, Feb. 2006.

〔3〕 K. Singh, G. Nair and H. Schulzrinne. "Centralized conferencing using SIP," in Internet Telephony Workshop 2001, New York, April 2001.

〔4〕 조현규, 이기수, 장춘서, "SIP 환경에서의 중앙 집중형 컨퍼런스 모델 기반의 새로운 서비스 모델에 관한 연구," 한국콘텐츠학회논문지, 제6권, 제 2호, pp.17-26, 2006.

〔5〕 Y. Cho et al., "Distributed management architecture for multimedia conferencing using SIP," RFC 4579, Int. Conf. DFMA, pp.98-105, Feb. 2005.

〔6〕 R. V. Prased, R. Hurni, and H. Jamadagni , "A Scalable Distributed VoIP using SIP," Proc. 8th IEEE ISCC, pp.608-613, June 2003.

〔7〕 J. Rosenberg, H. Schulzrinne, O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State," draft-ietf-sipping-conference-package-09, Feb. 2005.

〔8〕 A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, June 2002.

〔9〕 H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003.

저 자 소 개

장 춘 서

1993년 2월 : 한국과학기술원 공학박사

1981년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

〈관심분야〉 : SIP, 임베디드 시스템, 인터넷텔레포니

조 현 규

2005년 8월 : 금오공과대학교 컴퓨터공학과 공학박사

2006년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 계약교수

〈관심분야〉 : SIP, VoIP, 실시간 인터넷 통신

이 기 수

1982년 2월 : 서울대학교 대학원 공학석사

1982년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

〈관심분야〉 : 디지털시스템, 데이터베이스