

## SIP 기반 분산형 컨퍼런스 구조에서의 새로운 컨퍼런스 정보 데이터 모델

장춘서\*, 이기수\*

### A New Conference Information Data Model in SIP based Distributed Conference Architecture

Choon-Seo Jang\*, Ky-Soo Lee\*

#### 요 약

컨퍼런스 참가자 수가 늘어남에 따라 성능이 저하되어 확장성의 제약이 있는 중앙 집중형 컨퍼런스 구조의 문제점을 해결하기 위하여 복수개의 컨퍼런스 서버에 부하를 분담시키는 분산형 컨퍼런스 구조들이 최근 연구되고 있다. 이들 구조에서는 컨퍼런스 참가자 수가 증가함에 따라 동적으로 새로운 컨퍼런스 서버들이 컨퍼런스 환경에 추가된다. 본 논문에서는 이러한 분산형 컨퍼런스 구조에 사용 할 수 있는 새로운 컨퍼런스 정보 데이터 모델을 제안하였다. 제안된 새로운 컨퍼런스 정보 데이터 모델에서는 주 컨퍼런스 서버와 복수개의 부 컨퍼런스 서버들 사이에 컨퍼런스 운영에 필요한 컨퍼런스 정보를 교환할 수 있도록 하기위한 여러 요소들이 추가되었고, 이를 이용한 컨퍼런스 정보 교환 절차가 제시되었다. 또 이 방식을 사용해 컨퍼런스 정보의 관리와 전체 컨퍼런스 참가자들에 대한 SIP(Session Initiation Protocol) 통지 기능도 컨퍼런스 서버들이 서로 분산 처리 할 수 있게 됨으로써 주 컨퍼런스 서버에 가해지는 부하를 더욱 줄일 수 있게 된다. 제안된 시스템의 성능은 실험을 통하여 분석하였다.

#### Abstract

The centralized conference architecture has a restriction in scalability due to the performance reduction as the number of conference participants increases. To solve this problem several distributed conference architectures have been studied recently. In these architectures new conference servers are added dynamically to the conference environment. In this paper, We have proposed a new conference information data model which can be used in

---

• 제1저자 : 장춘서  
• 투고일 : 2009. 04. 14, 심사일 : 2009. 05. 04, 게재확정일 : 2009. 05. 25.

\* 금오공과대학교 컴퓨터공학부 교수

※ 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

these distributed conference architectures. In our newly proposed conference information data model, several components has been added for exchanging conference information between primary conference server and multiple secondary conference servers. We also proposed a procedure of conference information exchange between these conference servers. And the management of conference informations and SIP(Session Initiation Protocol) notifications to the total conference participants can be processed distributedly with these conference servers, therefore the load to the primary conference server can be decreased by using this method. The performance of our proposed model has been evaluated by experiments.

▶ Keyword : SIP(Session Initiation Protocol), 컨퍼런스 데이터 모델(Conference Data Model), 분산 컨퍼런스 구조(Distributed Conference Architecture)

## I. 서론

컨퍼런스 구조에서 중앙에 컨퍼런스 서버가 위치하여 컨퍼런스 전체를 제어하고 관리하는 중앙 집중형 구조[1][2]는 컨퍼런스 관리 및 제어가 쉬운 장점을 가진 반면 참가자 수가 증가 할수록 성능에 제약을 받는다. 이는 하나의 컨퍼런스 서버에서 처리해야할 부하가 커지지 때문이며, 이와 같은 중앙 집중형 구조의 문제점을 해결하기 위하여 복수개의 컨퍼런스 서버에 부하를 분담시키는 분산형 컨퍼런스 구조들이 최근 연구되고 있다[3][4].

분산형 컨퍼런스 구조에서는 컨퍼런스 참가자 수가 증가함에 따라 동적으로 새로운 컨퍼런스 서버가 컨퍼런스 환경에 추가되는 방식을 취하게 되는데 이때 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에 컨퍼런스 동작에 필요한 컨퍼런스 정보를 효율적으로 교환할 수 있도록 할 필요가 있다.

따라서 본 논문에서는 이러한 분산형 컨퍼런스 구조에서 사용할 수 있도록 새로운 컨퍼런스 정보 데이터 모델을 제안하였다. 본 연구에서 설계한 시스템에서는 컨퍼런스 참가자 수가 현재 서버가 담당 할 수 있는 한도보다 커지면 시스템 내에서 새로운 컨퍼런스 서버가 동적으로 추가되고 또 각 컨퍼런스 서버의 부하에 대한 균등 분배가 이루어지게 된다.

이 과정에서 제안된 컨퍼런스 정보 데이터 모델을 사용해 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에 컨퍼런스 동작 및 관리에 필요한 컨퍼런스 정보가 교환되며 이를 위한 새로운 원소들이 데이터 모델에 추가되었다. 또 이를 이용한 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에서의 컨퍼런스 정보 교환 절차가 제시되었다.

또 본 논문에서는 분산형 컨퍼런스 구조에서 주 컨퍼런스 서버가 전체 컨퍼런스 참가자들에 대한 SIP[5] 통지 기능을

전담함으로써 발생하는 부하 집중 문제를 개선하기위해 이의 처리를 복수의 컨퍼런스 서버들이 분담하는 방안도 아울러 제시되었다. 이와 같이 함으로써 주 컨퍼런스 서버에 가해지는 부하를 더욱 줄일 수 있게 되고 결과적으로 컨퍼런스 시스템의 확장성을 높일 수 있게 된다.

제안된 시스템의 성능 분석을 위하여 복수의 컨퍼런스 서버를 갖춘 환경에서 컨퍼런스 참여자 수를 증가시켜 가면서 처리 시간을 측정하였다. 이때 전체 컨퍼런스 참여자는 복수개의 컨퍼런스 서버에 균등 분배된다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 기존의 중앙 집중형 컨퍼런스 구조와 분산형 컨퍼런스 구조의 비교 및 컨퍼런스 정보 데이터 모델에 대해 설명한다. III장에서는 본 논문에서 제안하는 시스템의 설계 및 구현과 정보 교환 절차가 제시되었고 IV장에서는 실험을 통하여 본 논문에서 제안한 시스템에 대한 성능 분석을 한 후 V장에서 결론을 맺는다.

## II. 관련연구

### 1. 중앙 집중형 컨퍼런스 구조와 분산형 컨퍼런스 구조 비교

하나의 컨퍼런스 서버에 포커스와 믹서 및 플로어제어 모듈을 두고 전체 컨퍼런스 참가자를 제어하는 형태인 중앙 집중형 컨퍼런스 구조는 컨퍼런스 관리가 용이하고 컨퍼런스 전체 제어에 유리하다. 그림 1에 이의 구성을 보였다.

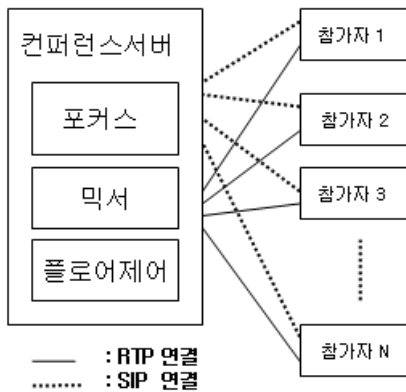


그림 1. 중앙 집중형 컨퍼런스 구조  
Fig.1 A Architecture of Centralized Conference

이 구조에서 포커스는 SIP 신호를 사용하여 컨퍼런스 참가자와 컨퍼런스 서버 사이의 세션 설정 등의 기능을 제공하며 믹서는 참가자와 서버사이에서 맺어진 RTP(Real-Time Transport Protocol) [6] 세션을 사용하여 미디어 패킷 스트림을 서로 교환하고 분배하는 기능을 한다. 플로어 제어 부분은 컨퍼런스 환경에서 공유 자원에 대한 접근을 제어하는 기능을 한다. [7]

이와 같은 중앙 집중형 구조에서는 전체 참가자 수가 증가하고 서버가 관리하는 전체 컨퍼런스가 많아질수록 성능의 저하가 심해져 컨퍼런스의 규모를 크게 할 수 없는 제약이 생긴다. [8] 따라서 이와 같은 문제점을 개선하기 위한 분산형 컨퍼런스 구조가 필요하게 된다.

분산형 컨퍼런스 구조의 한 종류로써 믹서 기능을 가진 미디어 서버를 분산시키거나 미디어 믹서 기능을 참가자 시스템에 분산시키는 방식이 있다. 이와 같은 방식은 컨퍼런스 시스템에서 처리해야 할 커다란 부분인 오디오/비디오 패킷 스트림을 분산 시킨다는 면에서 일정 부분 효과를 보이고 있다. [9]

그러나 믹서만을 분산시키는 방식은 하나의 포커스를 사용하게 됨으로써 전체 컨퍼런스 참가자 수가 증가 할수록 서버와의 SIP 세션 연결 및 관리에 들어가는 처리 양의 증가로 인해 컨퍼런스 확장성에 제약을 주게 된다. 따라서 컨퍼런스 참가자가 증가할 경우 동적으로 새로운 컨퍼런스 서버가 추가되어 전체적으로 복수개의 컨퍼런스 포커스가 작동하는 분산형 컨퍼런스 구조가 필요하게 된다.

이와 같은 분산형 컨퍼런스 구조에서 각 컨퍼런스 서버는 포커스와 믹서를 갖추고 있으므로 믹서도 복수개가 되어 오디오/비디오 패킷 스트림도 분산 처리할 수 있어 결국 컨퍼런스

의 확장성을 높일 수 있다.

## 2. 컨퍼런스 정보 데이터 모델

컨퍼런스 정보 데이터 모델은 컨퍼런스의 각 단계에서 컨퍼런스 개체를 표시하기 위한 정보들의 모음이다. [10] 각각의 컨퍼런스 정보 요소를 나타내기 위해 여러 개의 요소들과 하위요소들로 이루어지며 주요 요소들은 컨퍼런스 전체에 대한 정보를 기술하기 위한 <conference-description> 요소, 컨퍼런스를 주관하는 호스트에 대한 정보를 나타내는 <host-info> 요소, 현재 컨퍼런스 참가자 숫자 등의 컨퍼런스 상태를 표시하기 위한 <conference-state> 요소, 컨퍼런스 환경에서 참가자들의 공유 자원에 대한 접근을 제어하기 위한 <floor-information> 요소 및 컨퍼런스 참가자들에 대한 개별 정보를 나타내기 위한 <users> 요소 등이 있다.

이들 중 <conference-description> 요소의 하위요소인 <available-media> 요소는 각 참가자들의 미디어 스트림을 구분하기 위한 label 속성을 가지며, 코덱 종류를 나타내는 <codecs> 요소와 오디오/비디오 제어를 위한 <controls> 요소를 하위요소로 가진다. 플로어제어를 위한 <floor-information> 요소는 이진 플로어 제어 프로토콜(BFCP) [7]에서 사용하기 위한 32비트 값인 <conference-ID>를 하위요소로 가진다. <conference-floor-policy> 요소는 하위요소로 가지는 <floor>에서 플로어 제어를 위한 컨퍼런스 정책을 나타낸다.

그러나 아직 분산형 컨퍼런스 환경에서 컨퍼런스 동작 및 관리에 필요한 컨퍼런스 정보 데이터 모델이 정의되고 있지 않으며 분산형 컨퍼런스 환경에서 컨퍼런스 정보 데이터 모델에 기반을 둔 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이의 컨퍼런스 정보 교환 절차도 제안된 바가 없다. 따라서 본 논문에서는 분산형 컨퍼런스 구조에서 사용 할 수 있는 새로운 컨퍼런스 정보 데이터 모델을 제안하였다.

## III. 시스템 설계 및 구현

### 1. 분산형 컨퍼런스 구조 설계

그림 2 에 본 논문에서 설계한 분산형 컨퍼런스 구조를 보았다. 여기에서 주 컨퍼런스 서버는 포커스, 믹서 및 컨퍼런스 환경에서 공유 자원에 대한 접근을 제어하는 플로어제어 모듈로 구성되며 부컨퍼런스 서버도 동일한 구성을 가진다. 주 컨퍼런스 서버는 주 포커스를 가지며 컨퍼런스 시스템의 전체 동작 및 관리를 담당한다.

컨퍼런스 참가자 수가 컨퍼런스 정책에서 미리 정해진 값보다 커질 경우 새로운 컨퍼런스 서버가 동적으로 추가되고 또 각 컨퍼런스 서버의 부하에 대한 균등 분배가 이루어지게 된다.

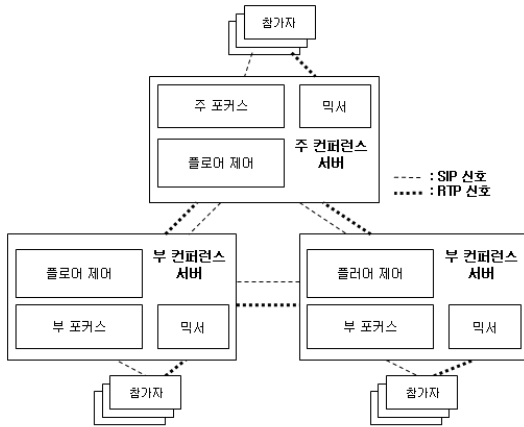


그림 2. 설계된 분산형 컨퍼런스 구조  
Fig.2 A Designed Architecture of Distributed Conference

이 과정에서 본 논문에서 제안된 컨퍼런스 정보 데이터 모델을 사용해 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에 컨퍼런스 동작 및 관리에 필요한 컨퍼런스 정보가 교환되게 된다. 그림 3은 이 과정의 순서도이다.

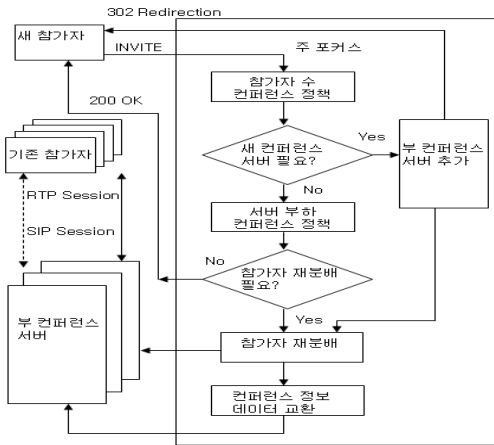


그림 3. 컨퍼런스 동작 순서도  
Fig. 3 Flowchart of Conference Operation

그림 3의 컨퍼런스 동작 순서도에서는 새로운 참가자가 컨퍼런스에 참여를 시도하면 주 포커스는 새로운 컨퍼런스 서버가 필요한지를 결정하기 위하여 참가자 수에 관한 컨퍼런스

정책을 조사한다. 만일 기존 컨퍼런스 서버들로 처리가 가능하다고 판단 될 경우 서버 부하에 관한 정보를 가지고 있는 컨퍼런스 정보 데이터베이스를 검색하여 참가자 재분배가 필요 한지 여부를 판단한다. 필요없는 경우 새로운 참가자는 SIP 응답 신호를 받은 후 RTP 세션을 맺어 즉시 컨퍼런스에 참여하게 된다.

새로운 컨퍼런스 서버의 추가가 필요한 경우 부 컨퍼런스 서버 추가 과정으로 들어가며 부 컨퍼런스 서버 생성 후 이 참가자에게는 이 서버로의 연결을 위한 SIP Redirection 신호가 보내진다. 다음 각 컨퍼런스 서버의 부하에 대한 균등 분배를 위해 추가된 서버를 포함한 전체 컨퍼런스 서버들에 대한 참가자 재분배 과정이 이어지고 이 결과를 토대로 각 서버들에 대한 컨퍼런스 정보 데이터 교환이 행해진다.

## 2. 새로운 컨퍼런스 정보 데이터 모델 및 컨퍼런스 정보 교환 절차

본 논문에서는 분산형 컨퍼런스 구조에 사용 할 수 있는 새로운 컨퍼런스 정보 데이터 모델을 제안하였다. 여기서는 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에 컨퍼런스 정책을 설정하고 컨퍼런스 운영에 필요한 컨퍼런스 정보를 교환할 수 있도록 하기위한 다음과 같은 요소들이 추가되었다.

컨퍼런스 전체에 대한 정보를 기술하기위한 요소인 <conference-description>에 하위 요소로써 현재 운용 중인 전체 컨퍼런스 서버의 개수를 나타내는 <tot-num-servers>, 각 컨퍼런스 서버의 주요 항목을 표시하기위한 <servers-desc>가 추가되었다. <servers-desc>는 하위요소로 각각의 컨퍼런스 서버를 구분하기위한 <server-identity>, 각 컨퍼런스 서버가 처리 할 수 있는 최대 참가자 수를 나타내는 <max-users>를 가진다.

현재 컨퍼런스의 상태를 표시하기위한 요소인 <conference-state>에 하위 요소로써 현재 전체 컨퍼런스 참가자의 수를 나타내기 위한 <current-tot-num-users>, 각 컨퍼런스 서버가 현재 처리하고 있는 참가자수를 나타내기 위한 <current-num-server-users>, 각 컨퍼런스 서버가 현재 담당하고 있는 참가자 목록인 <current-users>가 추가되었다. <current-num-server-users>는 하위요소로써 각각의 컨퍼런스 서버를 구분하기위한 <server-identity>, 현재 처리중인 참가자 수를 나타내는 <current-num-users>를 가진다. <current-users>는 하위요소로 각각의 컨퍼런스 참가자를 구분하기위한 <user-identity>를 가진다.

컨퍼런스 참가자들에 대한 개별 정보를 나타내기 위한 요

소인 (users)에는 하위 요소로 컨퍼런스 정보 통지 여부를 나타내기 위한 (allow-notify), 컨퍼런스 정보 통지할 경우 사용하기 위한 (notify-id), 부분 컨퍼런스 정보 데이터 포맷 가능 여부를 나타내기 위한 (allow-differential)가 추가되었다.

그림 4에 이와 같은 내용을 사용한 주 컨퍼런스서와 부 컨퍼런스 서버들 사이의 컨퍼런스 정보 교환 절차가 제시되었다. 여기서는 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에 SIP 신호에 기반을 둔 등록 절차를 거쳐 서로 간에 컨퍼런스 정보를 SIP 통지 메시지로 교환한다. 이때 컨퍼런스 정보 데이터의 포맷은 본 논문에서 제안된 바와 같이 분산형 컨퍼런스 구조에 맞추어 구성된 XML 문서 형태이다.

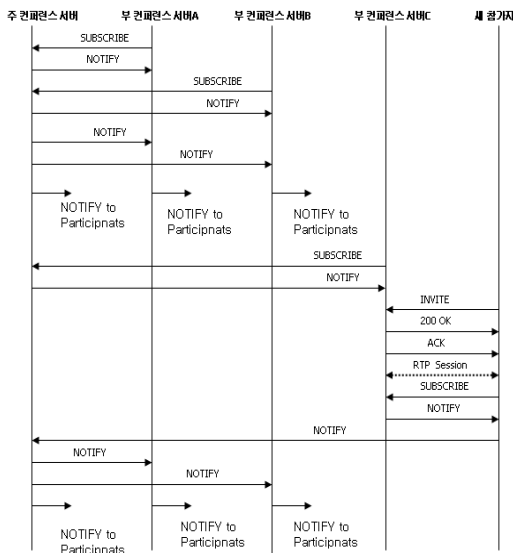


그림 4. 컨퍼런스 정보 교환 절차  
Fig. 4 Procedure of Conference Information

주 컨퍼런스 서버는 컨퍼런스 정보의 변화가 발생하면 이를 부 컨퍼런스 서버들에게 통지 메시지로 보낸다. 부 컨퍼런스 서버들은 이를 자신들의 데이터베이스에 저장 후 이 컨퍼런스 정보를 자신들이 관리하는 컨퍼런스 참여자들에게 통지 메시지로 보낸다.

이와같은 통지 처리 과정에서 전체 컨퍼런스 참가자들에 대한 통지 기능을 주 컨퍼런스 서버가 모두 처리하는 것이 아니라 본 논문에서 제안한 컨퍼런스 정보 데이터 모델을 사용해 각 부 컨퍼런스 서버들이 이를 서로 분산 처리 할 수 있어 주 컨퍼런스 서버에 가해지는 부하를 더욱 줄일 수 있게 된다.

컨퍼런스 참가자의 증가로 인해 부 컨퍼런스 서버 (서버

C)가 새로이 시스템에 추가되는 경우 먼저 주 컨퍼런스 서버에 SIP 등록 메시지를 보낸 후 이에 대한 통지 메시지를 통하여 현재의 전체 컨퍼런스 정보 데이터를 얻게 된다. 다음 새로운 참가자로부터 주 컨퍼런스 서버가 분산 처리를 위해 Redirection 시켰던 INVITE 요청 메시지를 받아 처리하게 된다. 이 후 이 참가자와 RTP 세션을 맺어 이 참가자에 대한 미디어 스트림 처리를 담당하게 된다.

새로운 참가자가 자신을 담당하는 부 컨퍼런스 서버에게 SIP 등록 요청을 하면 해당 부 컨퍼런스 서버는 현 컨퍼런스 정보를 담은 SIP 통지 메시지를 보내고 이어 주 컨퍼런스 서버에게 이 참가자에 대한 정보를 통지한다. 주 컨퍼런스 서버는 나머지 부 컨퍼런스 서버들에게 변화된 컨퍼런스 정보 데이터를 전송하게 된다. 부하 분담을 위하여 컨퍼런스 서버들 사이에 컨퍼런스 참가자들에 대한 동적인 재 할당이 이루어져 컨퍼런스 정보의 변화가 생겼을 경우에도 각 컨퍼런스 서버들에게 이를 전송한다.

#### IV. 성능 분석

본 논문에서 제안한 분산형 컨퍼런스 구조의 성능 분석을 위하여 컨퍼런스 서버로는 운영체제로 Fedora 버전 4 리눅스를 설치한 두 대의 PC가 사용되었고 이중 한 대가 주 컨퍼런스 서버로 동작하고 나머지 한대가 부 컨퍼런스 서버로 동작한다. 컨퍼런스 참가자용으로는 MS 윈도우즈 XP를 설치한 10대의 PC가 사용되었다. PC의 네트워크 인터페이스 속도는 100Mbps 이며 하나의 LAN 세그먼트 상에 모두 배치하였다.

먼저 본 논문에서 제안한 분산형 컨퍼런스 구조와 기존의 중앙 집중형 컨퍼런스 구조와의 평균 지연시간을 비교 측정하여 제안된 방식이 어느 만큼의 성능 개선 효과를 거둘 수 있는지 분석하였다. 분석을 위하여 컨퍼런스 참가자 수가 5명 이하인 경우 주 컨퍼런스 서버만 동작하다가 계속 증가해 5명을 넘어서면 6명부터 새로운 부 컨퍼런스 서버가 추가되도록 컨퍼런스 정책을 설정하였다.

새로운 컨퍼런스 서버 추가 시 기존 컨퍼런스 서버와의 참가자 균등 분담을 위한 컨퍼런스 정책은 6명인 경우 각 컨퍼런스 서버에 각 3명씩 분담하고, 7명인 경우 주 컨퍼런스 서버에 3명, 부 컨퍼런스 서버에 4명, 8명인 경우 각 4명, 9명인 경우 주 컨퍼런스 서버에 4명, 부 컨퍼런스 서버에 5명, 10명인 경우 각 5명 씩 분담하도록 하여 재조정이 이루어지도록 하였다.

그림 5는 제안된 분산형 컨퍼런스 구조와 기존의 중앙 집중형 컨퍼런스 구조와의 평균 지연시간을 비교 측정한 결과이다.

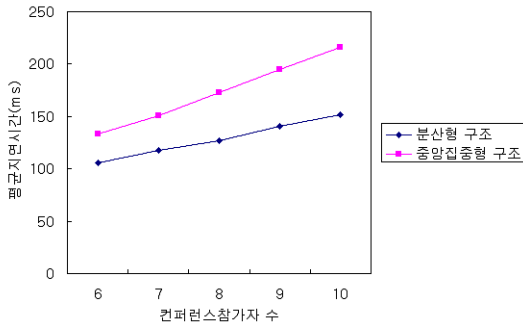


그림 5 제안된 분산형 컨퍼런스 구조와 중앙집중형 컨퍼런스 구조와의 평균 지연시간 비교  
 Fig. 5 Comparison of average delay time of suggested distributed conference architecture with that of centralized conference architecture

참가자 수가 6명 일 때의 측정 결과는 제안된 분산형 컨퍼런스 구조에서의 평균 지연 시간이 20% 감소함을 보여주고 있다. 참가자 수가 7명 일 때는 22% 감소되고, 참가자 수가 8명 일 때는 26.6% 가 감소함을 보여주고 있다. 참가자 수가 9명 일 때는 27.7% 감소되고 참가자 수가 10명일 때는 29.6% 감소하였다. 따라서 측정 결과는 참가자가 수가 증가 할수록 분산형 컨퍼런스 구조에서의 평균 지연시간의 개선 효과가 커짐을 보여주고 있다.

그림 6은 전체 컨퍼런스 참가자에 대한 SIP 통지 메시지 처리를 주 컨퍼런스 서버만 사용하여 처리한 경우와 부 컨퍼런스 서버들과 이를 분담한 경우를 비교 측정한 결과이다.

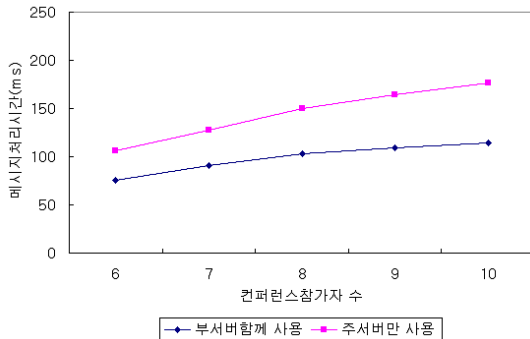


그림 6 제안된 방식에 의한 메시지 처리시간 비교  
 Fig. 6 Comparison of message processing time by the suggested method

여기서 부 컨퍼런스 서버들은 본 논문에서 제안한 컨퍼런스 정보 데이터 모델을 사용하여 컨퍼런스 서버들 사이의 컨퍼런스 정보 교환이 이루어지므로써 통지 처리를 분담할 수 있도록 하였다. 측정 결과는 참가자 수가 6명일 때 주 컨퍼런스 서버만 사용한 경우와 비교하여 28%의 처리시간 감소를 보여주고 있으며, 7명일 때는 28.9% 감소함을 보여주고 있다. 이 후 참가자 수가 증가 할수록 각각 31.3%, 35.5% 및 35.6%의 처리시간 감소를 보여주고 있다.

### V. 결론 및 향후 과제

본 논문에서는 분산형 컨퍼런스 구조에 사용 할 수 있는 새로운 컨퍼런스 정보 데이터 모델을 제안하였다. 제안된 새로운 컨퍼런스 정보 데이터 모델에서는 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에 컨퍼런스 운영에 필요한 컨퍼런스 정보를 교환할 수 있도록 하기위하여 <conference-description> 요소, <conference-state> 요소 및 <users> 요소에 각종 하위 요소들이 추가되었다. 또 이를 이용하여 주 컨퍼런스 서버와 부 컨퍼런스 서버들 사이에서 컨퍼런스 동작 및 관리에 필요한 컨퍼런스 정보가 교환되는 컨퍼런스 정보 교환 절차가 제시되었다.

아울러 제안된 방식을 통하여 컨퍼런스 정보의 관리와 컨퍼런스 참가자들에 대한 통지 기능을 부 컨퍼런스 서버들에게도 분산 처리 할 수 있어 주 컨퍼런스 서버의 부하를 더욱 줄일 수 있음을 보였다. 제안된 시스템의 성능 분석을 위하여 주 컨퍼런스 서버와 부 컨퍼런스 서버를 갖춘 환경에서 컨퍼런스 참여자 수를 증가시켜 가면서 처리 시간을 측정하였다. 측정 결과 제안한 컨퍼런스 정보 데이터 모델에 기반을 둔 분산형 컨퍼런스 구조에서 20%에서 29.6%까지의 평균 지연시간의 개선 효과와 28%에서 35.6%까지의 통지 메시지 처리 시간의 개선 효과를 얻을 수 있음을 보였다. 향후 과제로는 컨퍼런스 참가자 수를 현재보다 늘리고 부 컨퍼런스 서버를 여러 개 구성하여 복수의 부 컨퍼런스 서버들의 추가가 컨퍼런스 시스템 성능에 미치는 영향을 측정하고 광역 대규모 컨퍼런스로 모델을 확장하여 시뮬레이션에 의한 성능 측정을 할 할 필요가 있다.

## 참고문헌

- [1] M. Barnes, C. Boulto and O. Levin, "A Framework for Centralized Conferencing," RFC 5239, June 2008.
- [2] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol (SIP)," RFC 4353, Feb. 2006.
- [3] Y. Cho et al., "Distributed management architecture for multimedia conferencing using SIP," Int. Conf. DFMA, pp.98-105, Feb. 2006.
- [4] R. V. Prasad, R. Hurni, and H. Jamadagni, "A Scalable Distributed VoIP using SIP," Proc. 8th IEEE ISCC, pp.608-613, June 2003.
- [5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "Session Initiation Protocol," RFC 3261, June 2002.
- [6] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003.
- [7] G. Camarillo, J. Ott and K. Drage, "The Binary Floor Control Protocol (BFCP)," RFC 4582, Nov. 2006.
- [8] 조현규, 이기수, 장춘서, "컨퍼런스 포커스를 위한 효율적인 SIP 메시지 처리 방법," 한국컴퓨터정보학회논문지, 제12권, 제 6호, 187-192쪽, 2007년 12월.
- [9] Y. Cho et al., "Policy-based distributed management architecture for large-scale enterprise conferencing service using SIP," Int. Conf. DFMA, pp.98-105, Feb. 2006.
- [10] O. Novo, G. Camarillo, D. Morgan and R. Even, "Conference Information Data Model for Centralized Conferencing," RFC Internet-draft, June 2008.

## 저 자 소 개



### 장춘서

1993년 2월 : 한국과학기술원 공학 박사

1981년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

〈관심분야〉 : SIP, 임베디드 시스템, 인터넷텔레포니



### 이기수

1982년 2월 : 서울대학교 대학원 공학석사

1982년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

〈관심분야〉 : 디지털시스템, 데이터베이스