

새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조

장준서*

A Distributed Conference Architecture with a New Load Control Method

Choonseo Jang*

요약

본 논문에서는 컨퍼런스 서버의 과부하를 방지하기위한 새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조를 제안하였다. 본 논문에서 제안한 새로운 이벤트 패키지는 컨퍼런스 부하 제어를 가능하게하고 새로운 컨퍼런스 정보 데이터 포맷은 컨퍼런스 서버들과 컨퍼런스 참가자들 사이의 SIP 메시지 교환 절차 및 부하 분산을 처리하는데 필요한 요소들이 추가되었다. 또 전체 컨퍼런스 참가자들에게 주기적으로 전송해야 할 컨퍼런스 정보 데이터의 처리도 모든 서버들이 분담하도록 하여 부하를 더욱 줄이도록 하였다. 제안한 부하 제어 이벤트 패키지는 각 컨퍼런스 서버들이 전체 컨퍼런스 서버들의 현재 부하 상태를 실시간으로 파악할 수 있도록 하며 특정 서버의 부하가 커지면 해당 서버는 전체 컨퍼런스 서버 중 가장 부하가 적은 서버를 선택하여 SIP 클라이언트 요청을 분산시키거나 새로운 컨퍼런스 서버를 생성하여 부하를 분산하도록 한다. 제안된 시스템의 성능은 실험을 통하여 분석하였고 평균 지연시간에서 21.6%, 평균 SIP 메시지 처리 시간에서 29.2% 향상됨을 보여주고 있다.

▶ Keyword : 세션시작프로토콜, 분산 컨퍼런스, 컨퍼런스 정보 데이터

Abstract

A distributed conference architecture with a new load control method has been suggested in this paper. A new event package in this paper enables to control conference load. Some additional elements for exchanging SIP messages between server and participants, and for distributing the

• 제1저자 : 장준서

• 투고일 : 2012. 02. 04, 심사일 : 2012. 03. 09, 게재확정일 : 2012. 04. 07.

* 금오공과대학교 컴퓨터공학과 교수(Dept. of Computer Engineering, Kumoh Institute of Technology)

* 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

load, have been added to new conference information data format. Furthermore to lessen the load, all conference servers share the processing of conference information data which should be transferred periodically to all participants. The suggested load control event package makes each server can get current load status of the overall servers. When load increases in one server SIP client requests are distributed by selecting a server which has the lowest load value, or new server is created to share the load. The performance of the proposed system has been evaluated by experiments. They shows 21.6% increase in average delay time, and 29.2% increase in average SIP message processing time.

▶ Keyword : Session Initiation Protocol, Distributed Conference, Conference Information Data

I. 서론

SIP(Session Initiation Protocol) 기반의 컨퍼런스 환경[1][2][3]에서 참가자 수가 어느 한도 이상 많아지면 컨퍼런스 서버에 과부하가 발생하게 되므로 복수개의 컨퍼런스 서버를 가지는 분산형 컨퍼런스 구조가 필요하다[4][5]. 본 논문에서는 참가자 수가 많아질 경우 발생하는 컨퍼런스 서버의 과부하를 방지하기위한 새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조를 연구하였다.

이를 위하여 분산형 컨퍼런스 환경에서 각 컨퍼런스 서버들은 본 연구에서 새롭게 제안한 부하 제어 이벤트 패키지를 사용하여 주기적으로 자신의 현재 부하 상태를 전체 컨퍼런스 서버들에게 알린다. 이와 같은 동작을 통하여 각 컨퍼런스 서버들은 서로 간의 현재 부하 상태를 실시간으로 파악할 수 있으며 특정 서버의 참가자 수가 증가하여 해당 서버의 부하가 커지면 이 서버는 전체 컨퍼런스 서버 중 가장 부하가 적은 서버를 선택하여 이 서버로 SIP 클라이언트 요청을 전달한다.

전체 컨퍼런스 서버들이 부하를 추가로 분담할 상황이 아닌 경우라면 새로운 컨퍼런스 서버를 생성하여 이 서버가 부하를 분산하도록 한다. 또 전체 컨퍼런스 참가자들에게 주기적으로 전송해야 할 컨퍼런스 정보 데이터의 처리도 모든 서버들이 분담하도록 한다. 이를 위하여 본 논문에서는 컨퍼런스 부하 제어를 위한 새로운 이벤트 패키지가 설계되었고 컨퍼런스 서버들과 컨퍼런스 참가자들 사이의 SIP 메시지 교환 절차 및 부하 분산을 처리하는데 필요한 요소들이 추가된 새로운 컨퍼런스 정보 데이터 포맷이 제시되었다. 본 논문에서 제안한 이와 같은 구조를 사용하면 분산 컨퍼런스 환경에서 시스템의 전체 부하를 적절히 제어 할 수 있으며 확장성도 아

울러 높일 수 있게 된다.

본 논문은 II장에서 관련 연구를 기술하고 III장에서는 본 논문에서 제안하는 새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조를 설계하며 이와 같은 분산형 컨퍼런스 환경에서 동작하는 부하 제어 이벤트 패키지 및 부하 분산을 처리하는데 필요한 요소들이 새롭게 추가된 컨퍼런스 정보 데이터 포맷에 대해 설명한다. 또 컨퍼런스 서버와 참가자 사이의 SIP 메시지 교환 절차도 아울러 제시되었다. IV장에서는 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

1. 분산형 컨퍼런스 환경

분산형 컨퍼런스 환경에서는 믹서 기능을 가진 복수개의 미디어 서버를 사용하여 비디오/오디오 패킷 스트림을 분산시키는 방식[6]이 있는데 이 경우 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 포커스와 믹서를 가진 복수개의 컨퍼런스 서버들을 사용하여 컨퍼런스 참가자가 증가할 경우 새로운 컨퍼런스 서버가 추가되는 분산형 컨퍼런스 구조가 필요하다.

그림 1의 분산형 컨퍼런스 구조에서는 각 컨퍼런스 서버가 포커스와 믹서를 갖추고 있으므로 믹서도 복수개가 되어 오디오/비디오 패킷 스트림도 분산 처리할 수 있어 결국 컨퍼런스의 확장성을 높일 수 있다.

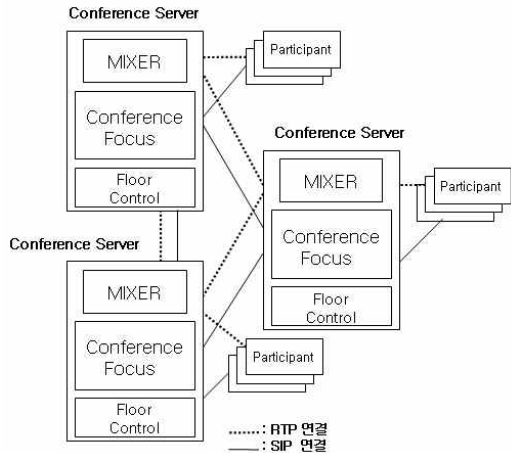


그림 1 분산형 컨퍼런스 구조
Fig. 1 Architecture of Distributed Conference

여기서 컨퍼런스 포커스는 컨퍼런스 세션의 설정 및 유지 기능을 제공하며 믹서는 RTP(Real-Time Transport Protocol)[7] 프로토콜을 사용하여 컨퍼런스 서버와 참가자 사이에서 미디어 패킷 스트림을 전송하는 기능을 한다. 플로어 제어 모듈은 컨퍼런스 환경에서 공유 자원에 대한 접근을 제어하기 위한 기능을 제공한다.

이와 같은 분산형 컨퍼런스 구조에서 컨퍼런스 참가자가 증가하게 되면 새로운 컨퍼런스 서버가 추가되어야 하고 이때 추가된 컨퍼런스 서버와 기존 서버들 사이에서는 컨퍼런스 참가자의 분배, SIP 세션 연결을 위한 메시지 교환 및 컨퍼런스 정보의 교환 등에 관한 제어가 필요하다.

2. 컨퍼런스 정보 데이터 포맷

기본적인 컨퍼런스 정보 데이터 포맷[8][9]은 XML 문서인 application/conference-info+xml 형태를 가지며 'conference-info'가 최상위 요소(element)이다. 이 요소의 속성으로는 컨퍼런스 URI 값을 나타내는 'entity'와 통지 메시지의 버전을 나타내는 'version' 및 이전 컨퍼런스 정보에 비해 변화된 부분만을 담은 정보인지 여부를 나타내는 'state'가 있다. 하위 요소(child element)로는 컨퍼런스 전체에 대한 정보를 기술하는 'conference-description' 과 현재 컨퍼런스 상태를 표시하기 위한 'conference-state' 및 컨퍼런스 참가자들에 대한 개별 정보를 나타내기 위한 'users' 등이 있다.

요소 'users'는 하위 요소로 각 참가자들의 정보를 나타내는 'user'를 가지며 이 요소의 속성으로는 컨퍼런스 사용자에게 대한 URI를 나타내는 'entity'가 있고 하위 요소로는 각 참가

자들이 컨퍼런스에 참여하면서 사용하는 디바이스 및 SIP 신호 세션에 대한 정보를 나타내는 'endpoint'가 있다. 이 요소는 다시 컨퍼런스에 참가한 방식을 나타내는 'joining-method'와 컨퍼런스 서버와의 미디어 스트림 정보를 나타내는 'media' 및 디바이스의 연결 상태를 나타내는 'status' 등의 여러 하위 요소를 가진다. 요소 'media'는 컨퍼런스 내의 여러 미디어 스트림 중 해당 스트림을 구분하기 위해서 포커스에 의해 부여되는 'label'과 실제 소스 미디어를 나타내는 'src-id' 등의 여러 하위 요소를 가진다.

최근에 분산형 컨퍼런스 환경에서 컨퍼런스 서버들을 추가하여 확장성을 높이기 위한 컨퍼런스 정보 데이터 포맷이 연구되고 있다[10]. 본 논문에서는 분산형 컨퍼런스 환경에서 부하 제어를 위해 필요한 요소들이 추가된 새로운 컨퍼런스 정보 데이터 포맷을 제안하였다.

III. 시스템 설계 및 구현

1. 새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조 설계

본 논문에서 설계한 새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조를 그림 2에 보였다. 그림 2에서 각 컨퍼런스 서버들은 포커스, 믹서, 부하 제어 모듈 및 확장된 컨퍼런스 정보 데이터베이스로 구성된다. 컨퍼런스 정보 데이터베이스는 기존의 방식과 달리 부하 제어 기능이 추가된 새로운 컨퍼런스 정보 데이터 포맷을 사용하여 구현되었다. 각 컨퍼런스 서버들은 서로 간에 SIP SUBSCRIBE 메시지를 사용하여 본 논문에서 제안한 부하 제어 컨퍼런스 이벤트 패키지를 등록하고 전체 컨퍼런스 참가자들에 대한 컨퍼런스 정보 데이터의 통지 기능도 분담하여 부하를 줄인다.

컨퍼런스 참가자가 컨퍼런스 URI를 담은 SIP INVITE 메시지를 사용하여 컨퍼런스 참가 요청을 하면 이를 받은 컨퍼런스 서버는 본 논문에서 제안한 부하 제어 컨퍼런스 정보 데이터 포맷을 사용해 구현된 데이터베이스를 액세스한다. 자신의 현재 부하 상태를 나타내는 'media-load-index'와 'sip-message-load-index' 값이 이 참가자 요청을 처리할 수 있는 경우 상대방에게 SIP 200 OK 응답을 보내고 이어서 RTP 세션을 맺어 컨퍼런스에 참가하도록 한다.

현재 부하 상태가 커서 다른 컨퍼런스 서버에게 컨퍼런스 참가자 요청을 전달 할 경우 응답으로 상태 코드 번호가 302이고 다른 컨퍼런스 서버를 가리키는 Redirection SIP 메시

지를 보내어 참가자가 이 서버와 연결을 맺도록 한다. 새로운 컨퍼런스 서버를 추가해야 할 경우에는 컨퍼런스 정보 데이터베이스에서 선택 가능한 새로운 컨퍼런스 서버를 선택하여 이 서버에게 INVITE 메시지를 보낸다. 해당 컨퍼런스 서버가 응답하면 부하 제어 컨퍼런스 이벤트 패키지를 사용하여 현 컨퍼런스 정보 데이터를 보내고 이어서 기존에 동작 중인 컨퍼런스 서버들과 SIP 세션 및 미디어 스트림 전달을 위한 RTP 세션을 생성하도록 한다. 이와 같은 컨퍼런스 동작 과정에 대한 순서도를 그림 3에 보았다.

그림 3에서 새로운 컨퍼런스 참가자의 요청을 받은 컨퍼런스 서버는 부하 제어 기능을 가진 컨퍼런스 정보 데이터베이스를 액세스하여 현재 가장 부하 상태가 낮은 컨퍼런스 서버를 검색하고 이 서버에게 요청을 전달한다. 모든 컨퍼런스 서버의 부하 상태가 높아 새로운 컨퍼런스 서버의 추가가 필요한 경우 컨퍼런스 정보 데이터베이스에서 새롭게 추가 될 수 있는 컨퍼런스 서버들의 URI를 가지고 있는 'conf-server-list-uri' 항목에서 컨퍼런스 서버를 선택한다. 새로운 컨퍼런스 서버 추가 후 기존 컨퍼런스 서버들 중 부하 상태가 큰 서버가 담당하는 참가자들에게 새로 추가된 서버의 주소를 넣은 SIP REFER 메시지를 보낸다. 이에 대해 해당 참가자는 새로 추가된 서버에게 INVITE 메시지를 보내어 세션을 맺고 기존의 서버에게는 BYE 메시지를 보내어 연결을 종료함으로써 각 컨퍼런스 서버의 부하를 조절 할 수 있게 된다.

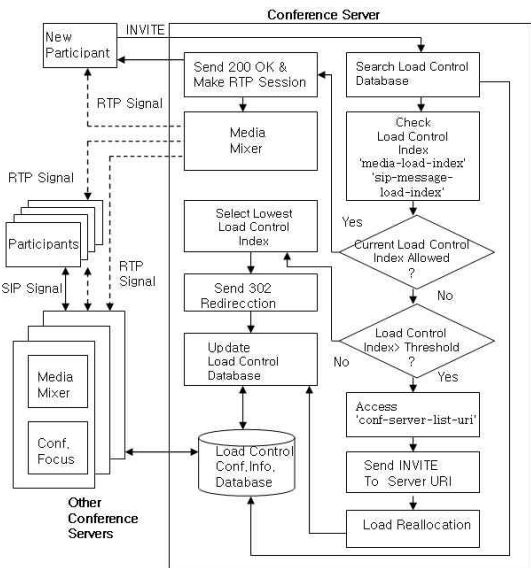


그림 2 부하 제어 방식을 사용한 분산형 컨퍼런스 동작 과정 순서도
Fig. 3 Flowchart of Distributed Conference Operation using Load Control

2. 부하 제어 이벤트 패키지 및 컨퍼런스 정보 데이터 포맷 설계

본 연구에서는 부하 제어 이벤트 패키지 방식을 처리하기 위하여 다음과 같은 필요 요소들이 추가된 새로운 컨퍼런스 정보 데이터 포맷이 설계되었다. 먼저 최상위 요소 'conference-info'의 하위 요소로 부하 제어를 위하여 'media-load-index'와 'sip-message-load-index'가 새롭게 추가되었다. 요소 'media-load-index'는 각 컨퍼런스 서버에서 미디어 스트림 처리를 위한 부하 지수를 나타내며 RTP 세션에서 사용하는 오디오 샘플 및 비디오 프레임의 페이로드 타입에서 결정되는 초당 전송량의 전체 합을 기준으로 산출한다. 속성으로는 최대 부하 허용값을 나타내는 'max-media-load-index'와 각 컨퍼런스 서버를 구분하기 위한 'server-label'을 가진다. 요소 'sip-message-load-index'는 각 컨퍼런스 서버에서 SIP 메시지 처리를 위한 부하 지수를 나타내며 컨퍼런스 포커스에 수신되는 초당 SIP 요청 메시지 개수의 합을 기준으로 산출한다. 속성으로는 최대 부하 허용값을 나타내는 'max-sip-message-load-index'와 각 컨퍼런스 서버를 구분하기 위한 'server-label'을 가진다.

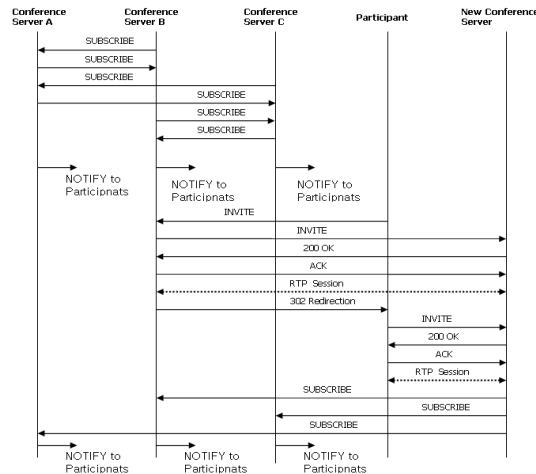


그림 3 SIP 메시지 및 컨퍼런스 정보 교환 절차
Fig. 4 Exchange Procedure of SIP Messages and Conference Information

다음 추가된 요소로는 현재 동작 중인 전체 컨퍼런스 서버의 개수를 나타내는 'servers-number'와 시스템의 부하 상황에 따라 새롭게 추가 될 수 있는 컨퍼런스 서버들의 URI를 가지고 있는 'conf-server-list-uri' 및 각 서버가 현재 담당

하고 있는 참가자 목록인 'current-participants-list'가 있다. 요소 'current-participants-list'는 속성으로 각 컨퍼런스 서버를 구분하기 위한 'server-label'을 가진다. 또 해당 서버가 현재 처리하고 있는 참가자수를 나타내는 'current-participants-number'가 있고 이 요소 역시 각 컨퍼런스 서버를 구분하기 위한 'server-label'을 속성으로 가진다.

그리고 현재 분산형 컨퍼런스의 상태를 표시하기 위한 요소인 'distributed-conference-state'가 새롭게 추가 되었고 이의 하위 요소로써 현재 전체 컨퍼런스 참가자의 수를 나타내는 'current-total-participants-number'와 컨퍼런스 참가자들에 대한 개별 정보를 나타내기 위한 요소인 'participants-info'가 추가되었다. 요소 'participants-info'는 하위 요소로 부하 제어 이벤트 패키지 사용 여부를 나타내기 위한 'load-control-notify'와 부하 제어 이벤트 패키지에 의한 통지 메시지에 메시지 크기를 줄이기 위한 부분 컨퍼런스 정보 데이터 포맷 가능 여부를 나타내기 위한 'load-control-notify-differential'이 추가되었다. 이와 같은 부하 제어를 처리하기 위한 필요 요소들이 새롭게 추가된 컨퍼런스 정보 데이터 포맷을 사용한 SIP 메시지 교환 절차를 그림 4에 보였다. 여기서 SIP 200 OK 응답 메시지는 주요 부분에만 보였다.

먼저 각 컨퍼런스 서버들 사이에 SUBSCRIBE 메시지를 사용하여 본 논문에서 제안한 부하 제어 이벤트 패키지를 등록하고 있다. 새로운 참가자가 컨퍼런스 서버 중 하나에 컨퍼런스 참가 요청을 하면 해당 컨퍼런스 서버는 컨퍼런스 정보 데이터베이스를 액세스하여 부하 상태를 조사하고 새로운 컨퍼런스 서버를 추가해야 할 경우에는 이 서버에게 INVITE 메시지를 보낸다. 이 후 새로운 컨퍼런스 서버와 RTP 세션을 맺고 부하 제어 컨퍼런스 이벤트 패키지를 서로 등록하며 현재의 컨퍼런스 정보 데이터를 전송한다.

이어서 새로운 참가자에게 이 서버의 주소를 넣은 SIP 302 Redirection 메시지를 보내어 새롭게 추가된 서버와 SIP 세션 및 RTP 세션을 생성하도록 한다. SIP 302 Redirection은 다른 주소로 SIP 세션을 연결하는데 사용되는 메시지이다. 전체 참가자들에 대한 컨퍼런스 정보 데이터의 통지는 각 컨퍼런스 서버들이 분담함으로써 부하를 더욱 줄일 수 있게 된다.

IV. 성능 분석

본 논문에서 제안한 새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조의 성능 분석을 위하여 컨퍼런스 서버로는

운영체제로 커널 버전 2.6인 리눅스를 가진 두 대의 PC를 사용하였다. 일반 컨퍼런스 참가자용으로는 운영체제로 MS 윈도우 XP를 설치한 15대의 PC가 사용되었다. 모든 PC는 하나의 LAN 세그먼트로 구성된 서브넷 상에 배치하였고 각 PC의 네트워크 인터페이스 및 허브를 포함한 LAN 환경의 속도는 100Mbps이다. 각 PC의 사양은 1.8GHz 코어2듀오를 CPU로 사용하고 메인메모리 용량은 2GB이다.

본 논문에서 제안한 부하 제어 방식을 사용한 분산형 컨퍼런스 구조와 부하 제어 방식을 사용하지 않은 기존의 컨퍼런스 구조와의 평균 지연시간을 비교 측정하였다. 이를 위하여 부하 제어 방식을 사용하는 경우 컨퍼런스 정책으로 최대 부하 허용값을 나타내는 'max-media-load-index'와 'max-sip-message-load-index'를 500으로 하여 처음에 하나의 컨퍼런스 서버로 동작하다가 동작 중 두 개의 인덱스 값 중 하나라도 이 값을 넘으면 컨퍼런스 서버가 추가되도록 하였다. 그림 5에 이때의 측정된 결과를 보였다.

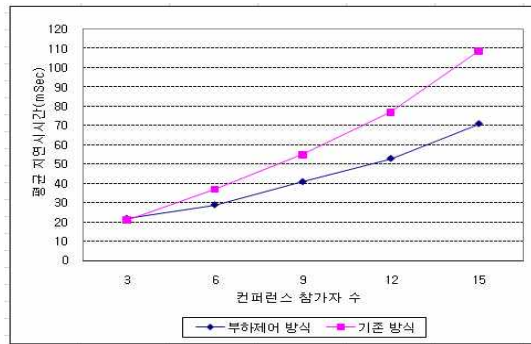


그림 5 평균 지연시간 비교(최대 부하 허용값=500)
Fig. 5 Comparison of Average Delay Time (Max. Load Allowance=500)

여기서는 컨퍼런스 참가자 수가 3명일 때는 양쪽 모두 1개의 컨퍼런스 서버만을 사용하므로 동일 한 성능을 보이나 6명일 때부터는 부하 제어 방식을 사용한 분산형 컨퍼런스 구조의 경우 컨퍼런스 서버가 추가되어 부하를 분담하게 되므로 성능의 향상을 보여준다. 컨퍼런스 참가자 수가 6명 일 때는 제안된 분산형 컨퍼런스 구조에서의 평균 지연 시간이 21.6% 감소함을 보여주고 있고 참가자 수가 9명 일 때는 25.5% 감소되고, 참가자 수가 15명 일 때는 31.1%가 감소되어 참가자가 수가 증가할수록 본 논문에서 제안된 분산형 컨퍼런스 구조에서의 평균 지연시간의 개선 효과가 커짐을 알 수 있다.

그림 6은 컨퍼런스 정책으로 최대 부하 허용값을 700으로 하여 측정된 결과이다. 여기서는 컨퍼런스 참가자 수가 6명

일 때까지는 1개의 컨퍼런스 서버만을 사용하다가 9명 일 때 부터 부하 제어 방식을 사용하여 컨퍼런스 서버가 추가되어 성능이 향상됨을 보여준다. 이와 같이 컨퍼런스 정책으로 최대 부하 허용값을 상황에 따라 조정 할 수 있어 컨퍼런스의 확장성을 높일 수 있게된다.

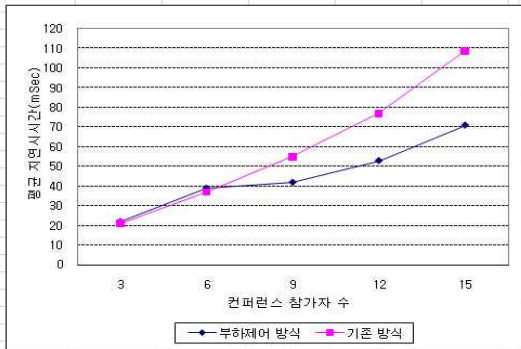


그림 6 평균 지연시간 비교(최대 부하 허용값=700)
Fig. 6 Comparison of Average Delay Time (Max. Load Allowance=700)

그림 7은 전체 컨퍼런스 참가자에 대한 컨퍼런스 정보 데이터의 전달에 소요되는 평균 SIP 메시지 처리 시간을 비교 측정한 결과이다. 최대 부하 허용값은 500으로 하였으며 따라서 컨퍼런스 참가자 수가 6명일 때부터 컨퍼런스 서버가 추가되어 SIP 메시지 처리를 분담하게 된다. 측정 결과는 참가자 수가 6명일 때 제안된 방식에서는 29.2%의 처리시간 감소를 보여주고 있으며, 9명일 때 34.0% 감소하고 15명일 때 39.4% 감소함을 보여주고 있으며 참가자 수가 증가 할수록 처리시간 감소폭이 커짐을 보여주고 있다. 아울러 INVITE 메시지와 SUBSCRIBE 메시지 처리에 리소스 리스트를 사용한 분산형 컨퍼런스 구조[11]를 사용하면 전체적인 SIP 메시지 처리시간을 더욱 단축시킬 수 있다.

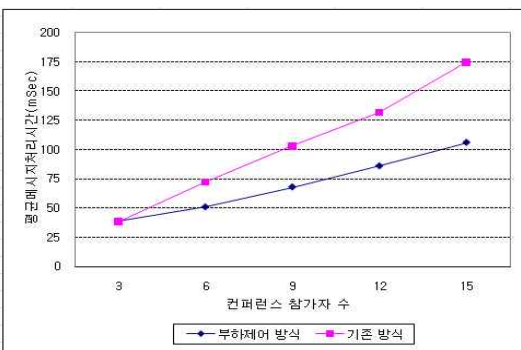


그림 7. 평균 SIP 메시지 처리시간 비교
Fig. 7 Comparison of Average SIP Message Processing Time

V. 결론 및 향후 과제

본 논문에서는 분산형 컨퍼런스 환경에서 각 서버들의 부하를 제어하여 참가자 수가 증가함에 따르는 컨퍼런스 서버의 과부하를 방지하기위한 새로운 부하 제어 방식을 사용한 분산형 컨퍼런스 구조를 연구하였다. 이를 위하여 본 논문에서는 컨퍼런스 부하 제어를 위한 새로운 이벤트 패키지가 설계되었고 컨퍼런스 서버들과 컨퍼런스 참가자들 사이의 SIP 메시지 교환 절차 및 부하 분산을 처리하는데 필요한 요소들이 추가된 새로운 컨퍼런스 정보 데이터 포맷이 제시되었다. 또 전체 컨퍼런스 참가자들에게 주기적으로 전송해야 할 컨퍼런스 정보 데이터의 처리도 모든 서버들이 분담하도록 하여 부하를 더욱 줄이도록 하였다. 본 논문에서 제안한 이와 같은 구조를 사용하면 분산 컨퍼런스 환경에서 시스템의 전체 부하를 적절히 제어 할 수 있으며 확장성도 아울러 높일 수 있게 된다.

제안된 시스템의 성능 분석을 위하여 복수개의 컨퍼런스 서버를 갖춘 환경에서 컨퍼런스 참여자 수를 증가시켜 가면서 처리 시간을 측정하였다. 측정 결과 본 연구에서 제안한 분산형 컨퍼런스 구조에서 21.6%에서 31.1%까지의 평균 지연시간의 개선 효과가 있으며 컨퍼런스 정보 데이터의 전달에 소요되는 평균 SIP 메시지 처리 시간을 29.2%에서 39.4%까지 감소시킬 수 있음을 보였다. 향후 과제로는 컨퍼런스 참가자 수를 더욱 늘려 대규모 컨퍼런스 환경을 구성하고 SIP 메시지 처리에 리소스 리스트를 사용한 분산형 컨퍼런스 구조를 채택하여 전체적인 SIP 메시지 처리시간을 더욱 단축시키는 방안과 무선 모바일 환경에서 본 논문이 제안한 구조가 얼마만큼 개선 효과를 얻는가를 관측하고 이에 맞춘 개선된 방식을 연구 할 필요가 있다.

참고 문헌

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "Session Initiation Protocol," RFC 3261, June 2002.
- [2] M. Barnes, C. Boulton and O. Levin, "A Framework for Centralized Conferencing," RFC 5239, June. 2008.

[3] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol (SIP)," RFC 4353, Feb. 2006.

[4] Y. H. Cho, M. S. Jeong, "Policy-based distributed management architecture for large-scale enterprise conferencing service using SIP," IEEE Journal on Communications, pp.1934-1949, Oct. 2005.

[5] Y. Cho et al., "Distributed management architecture for multimedia conferencing using SIP," Int. Conf. DFMA, pp.98-105, Feb. 2006.

[6] H. K. Cho, K. S. Lee, C. S. Jang, "A Method of Efficient SIP Messages Processing for Conference Focus," Journal of The Korea Society of Computer and Information, Vol 12, No.6, pp 187-192, December 2007.

[7] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2005.

[8] M. Barnes et al., "A Framework for Centralized Conferencing," RFC 5239, June 2008.

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

[10] C. S. Jang, H. K. Cho, K. S. Lee, "A Method of Efficient Conference Event Package Processing in Distributed Conference Environment" Journal of The Korea Society of Computer and Information, Vol 13, No.7, pp 199-205, December 2008.

[11] C. S. Jang, K. S. Lee, "A New Distributed Conference Architecture using Resource Lists" Journal of The Korea Society of Computer and Information, Vol 16, No.9, pp 123-129, December 2011.

저 자 소 개



장춘서

1993년 2월: 한국과학기술원
공학박사

1981년 3월 ~ 현재: 금오공과
대학교 컴퓨터공학과
교수

관심분야: SIP, 임베디드 시스
템, 인터넷텔레포니

Email: csjang@kumoh.ac.kr