

분산형 컨퍼런스 모델에서의 효율적 SIP 메시지 제어

장춘서*, 이기수*

An Efficient SIP Messages Control in Distributed Conference Model

Choonseo Jang*, Ky-Soo Lee*

요 약

본 논문에서는 SIP(Session Initiation Protocol) 환경에서의 분산형 컨퍼런스 모델에서 각 컨퍼런스 서버들의 부하를 줄여 확장성을 높일 수 있는 효율적인 SIP 메시지 제어 방식을 제안하였다. 제안된 시스템에서 컨퍼런스 참가자는 자신의 처리 능력 및 네트워크 상황에 맞추어 컨퍼런스 서버로부터 이벤트 통지 메시지 발생량을 실시간으로 제어 할 수 있고, 컨퍼런스 서버와의 세션 유지를 위해 필요한 각종 SIP 메시지의 발생량도 능동적으로 제어 할 수 있도록 하였다. 따라서 본 제어 방식을 사용하여 컨퍼런스 서버들이 처리해야 할 SIP 메시지를 크게 줄일 수 있어 대용량 컨퍼런스 시스템으로의 확장이 용이해지며 SIP 환경에서의 전체 네트워크 트래픽도 크게 감소시킬 수 있게 된다. 이를 위하여 본 연구에서는 효율적인 SIP 메시지 제어에 필요한 구성요소들이 추가된 새로운 컨퍼런스 이벤트 패키지 및 컨퍼런스 정보 데이터 모델이 제시되었다. 제안된 시스템의 성능은 실험을 통하여 분석하였다.

▶ Keywords : 세션시작프로토콜, 분산형 컨퍼런스 모델, 컨퍼런스 정보 데이터 모델

Abstract

In this paper, We have proposed an efficient SIP(Session Initiation Protocol) messages control which can be used to reduce conference servers load in distributed conference model. Each conference participant can control the amount of event notification messages from the conference server in real time according to its processing capacity and network circumstances. And the amount of SIP messages needed to maintain session with the conference servers can also be controlled actively in our system. The amount of SIP messages which should be processed by the

· 제1저자 : 장춘서, 교신저자 : 이기수

· 투고일 : 2012. 07. 03, 심사일 : 2012. 08. 09, 게재확정일 : 2012. 08. 28.

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

* 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

conference servers can be reduced largely by using our suggested method. Therefore the conference system can be more easily expandable and total network traffic can be reduced largely in SIP environment. We have suggested new conference event packages and conference information data model with some added elements that are necessary for efficient SIP messages control. The performance of the proposed system has been evaluated by experiments.

▶ Keywords : Session Initiation Protocol, Distributed Conference Model, Conference Information Data Model

I. 서론

복수개의 서버가 참가자를 나누어 처리하는 SIP(Session Initiation Protocol)[1] 환경에서의 분산형 컨퍼런스 시스템[2][3][4]에서 참가자 수가 증가 할수록 각 참가자와 컨퍼런스 서버들 사이에 전송되는 SIP 메시지 양은 빠른 속도로 증가하므로 전체 컨퍼런스 시스템의 성능을 저하 시키고 확장성을 가로 막는 주요 원인이 된다. 본 논문에서는 이와 같은 문제점을 해결하기 위하여 컨퍼런스 서버와 참가자 사이의 이벤트 통지 메시지 발생량을 감소 시킬 수 있고 컨퍼런스 서버와의 세션 유지를 위해 필요한 각종 SIP 메시지의 발생량도 크게 줄일 수 있는 효율적인 SIP 메시지 제어 방식을 제안하였다.

본 연구에서 컨퍼런스 참가자는 자신의 처리 능력 및 네트워크 상황에 맞추어 본 논문에서 새롭게 제안된 구성요소가 추가된 컨퍼런스 이벤트 패키지를 사용하여 컨퍼런스 서버로부터의 이벤트 통지 메시지 발생량을 실시간으로 제어 할 수 있다. 또 컨퍼런스 서버와의 세션 유지를 위해 필요한 리프레시 동작 및 이에 따르는 각종 SIP 메시지의 발생량도 본 논문에서 새롭게 제안된 구성요소가 추가된 컨퍼런스 정보 데이터 모델을 사용하여 능동적으로 제어 할 수 있도록 하였다.

이를 위하여 이벤트 통지 메시지 발생 주기를 제어 할 수 있는 새로운 속성 req-conf-event-min 및 req-conf-event-max가 사용되었고 아울러 새로운 속성 conf-refresh-entity와 conf-refresh-entity-match를 사용하여 컨퍼런스 서버와 참가자 사이에 엔티티 값을 서로 교환할 수 있도록 하였다.

이와 같이 제안된 분산형 컨퍼런스 시스템에서 개별 컨퍼런스 서버들의 부하가 감소하므로 컨퍼런스 참가자 수의 증가에 따른 대용량 컨퍼런스 시스템으로의 확장이 용이해지며 전체 네트워크 트래픽도 크게 감소시킬 수 있으므로 통신 대역폭의 제한을 받는 모바일 디바이스 경우나 무선 네트워크 환경에서 유용하게 사용될 수 있다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 여러 컨퍼런스 모델 중 복수개의 서버를 가지는 분산형 컨퍼런스 모델에 대해 설명한다. 다음 컨퍼런스 서버 및 참가자에 관한 각종 정보를 담고 있는 컨퍼런스 정보 데이터 구조에 대해 설명한다. III장에서는 본 논문에서 제안하는 새로운 SIP 메시지 제어 방식을 사용한 분산형 컨퍼런스 시스템을 설계하고 새롭게 제안된 구성요소가 추가된 컨퍼런스 이벤트 패키지 및 컨퍼런스 정보 데이터 구조에 대해 설명한다. 다음 이와같은 컨퍼런스 정보 데이터를 사용한 컨퍼런스 SIP 메시지 교환 절차에 대해 설명한다. IV장에서는 실험을 통하여 본 논문에서 제안한 시스템에 대한 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

1. 분산형 컨퍼런스 모델

컨퍼런스 모델에는 크게 단일 서버를 사용하는 모델 [5][6][7]과 복수개의 서버가 부하를 분담하는 분산형 컨퍼런스 모델이 있다. 분산형 컨퍼런스 모델에는 컨퍼런스 세션의 설정 및 유지 기능을 제공하는 포커스를 하나의 서버가 전담하고 복수개의 미디어 서버를 사용하여 비디오/오디오 패킷 스트림을 분산 시키는 모델[8]과 각 서버가 포커스와 미디어 믹서를 모두 가지고 있어 컨퍼런스 세션 관리와 미디어 처리를 분산시키는 모델이 있다.[4]

하나의 포커스를 사용하는 모델은 전체 컨퍼런스 참가자 수가 증가 할수록 서버와의 SIP 세션 연결 및 관리에 들어가는 처리 양의 증가로 인해 컨퍼런스 확장성에 제약을 주게 된다. 반면 각 서버가 포커스를 가지는 경우 각 서버가 담당하는 부하의 제어는 복잡해지나 컨퍼런스 확장성은 좋아진다. 그림 1은 각 서버가 포커스와 미디어 믹서를 모두 가지고 있는 분산형 컨퍼런스 모델이다.

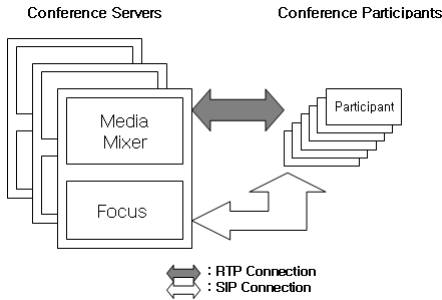


그림 1. 분산형 컨퍼런스 모델
Fig.1 Distributed Conference Model

그림 1에서 각 서버의 포커스는 참가자와 연결되는 컨퍼런스 세션을 설정하고 관리하며 미디어 믹서는 RTP(Real-Time Transport Protocol)[9] 방식을 사용하여 컨퍼런스 서버와 참가자사이의 비디오/오디오 패킷 스트림을 서로 교환하고 분배한다. 이때 참가자 수가 증가할수록 각 서버가 담당하는 부하를 균등하게 제어해야 하며 필요한 경우 새로운 컨퍼런스 서버가 추가되어야 한다. 현재 복수개 컨퍼런스 서버가 균등하게 부하를 분담하는 구조가 발표되고 있다.[4]

본 논문에서는 이와 같은 분산형 컨퍼런스 모델에서 컨퍼런스 참가자가 자신의 처리 능력 및 네트워크 상황에 맞추어 이벤트 통지 메시지 발생량을 실시간으로 제어 할 수 있고, 세션 유지를 위해 필요한 각종 SIP 메시지의 발생량도 능동적으로 제어 할 수 있는 새로운 방안을 제안하였다.

2. 기본적 컨퍼런스 정보 데이터 구조

기본적인 컨퍼런스 정보 데이터(Conference Information Data) 구조[10][11]는 루트 엘리먼트 태그로 <conference-info>를 가진다. 이 태그의 속성으로는 컨퍼런스 URI 값을 나타내는 'entity'와 컨퍼런스 정보 통지 메시지의 버전을 나타내는 'version' 및 이전 컨퍼런스 정보에 비해 변화된 부분만을 담은 정보인지 여부를 나타내는 'state'가 있다. 루트 엘리먼트의 하위 태그로는 컨퍼런스 전체 정보를 담고있는 <conference-description>과 현재 컨퍼런스 상태를 표시하는 <conference-state>가 있고 컨퍼런스 참가자들에 대한 개별 정보를 나타내기 위해서 <users> 태그가 위치한다.

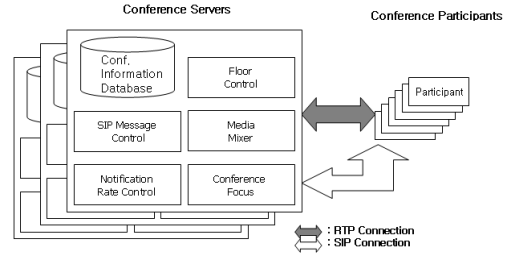


그림 2. 설계된 SIP 메시지 제어방식을 가진 분산형 컨퍼런스 구조
Fig.2 A Designed Distributed Conference Architecture with SIP Messages Control

태그 <users>는 각 참가자들의 정보를 나타내기 위해 하위 태그로 복수개의 <user>를 가진다. 이 태그의 속성 'entity'는 컨퍼런스 참가자에 대한 URI를 표시하고 태그 <endpoint>는 <user>의 하위 태그로써 각 참가자들이 사용하는 미디어 디바이스와 SIP 신호 세션에 대한 정보를 나타낸다. 태그 <endpoint>의 하위에 위치하는 <media> 태그는 미디어 소스 ID 등 컨퍼런스 서버와의 미디어 스트림 정보를 나타내고 <joining-method> 태그는 'dialed-in' 이나 'dialed-out' 등 컨퍼런스에 참가한 방식을 나타낸다. 또 <call-info> 태그는 참가자와 포커스 사이의 세부적인 호출 신호 정보를 나타낸다. 이 태그의 하위 태그인 <sip>는 컨퍼런스 서버와 참가자 사이의 SIP 다이얼로그 식별자를 나타내기 위한 <call-id>, <to-tag> 및 <from-tag> 태그를 가지고 있다.

본 논문에서는 이와 같은 기본적인 컨퍼런스 정보 데이터 구조를 개선하여 복수개의 서버가 부하를 분담하는 분산형 컨퍼런스 모델에서 컨퍼런스 서버와 참가자 사이의 이벤트 통지 메시지 발생량을 크게 감소시킬 수 있고, 리프레시 등 SIP 세션 유지를 위해 필요한 메시지의 발생량도 능동적으로 제어 할 수 있는 필요한 요소들이 추가된 새로운 컨퍼런스 정보 데이터 구조를 제안하였다.

III. 시스템 설계 및 구현

1. SIP 메시지 제어 방식의 분산형 컨퍼런스 구조 설계

그림 2에 본 논문에서 설계한 효율적인 SIP 메시지 제어 방식을 사용한 분산형 컨퍼런스 구조를 보였다. 여기에서 각 컨퍼런스 서버들은 주컨퍼런스 서버 및 부컨퍼런스 서버의 기능 구분이 없이 모두 동등한 기능을 가지며 참가자 수의 증가에 따른 컨퍼런스 부하를 균등하게 분담한다. 컨퍼런스 서버의 구성은 본 논문에서 새롭게 제안한 확장된 컨퍼런스 정보 데이터

구조로 구성된 데이터베이스와 컨퍼런스 포커스, 미디어 믹서, 통지메시지 제어 모듈 및 SIP 메시지 제어 모듈로 구성된다.

여기서 통지메시지 제어 모듈은 컨퍼런스 참가자가 자신의 처리 능력 및 네트워크 상황에 맞추어 컨퍼런스 서버로부터 이벤트 통지 메시지 발생량을 실시간으로 제어할 수 있도록 한다. 이를 위하여 컨퍼런스 참가자는 컨퍼런스 이벤트 패키지 등록 시 SIP SUBSCRIBE 메시지의 Event 헤더에 통지 메시지의 최소 간격 및 최대 간격을 나타내는 파라미터로 각각 req-conf-event-min 및 req-conf-event-max를 사용한다.

각 파라미터의 수치는 초를 나타내며 컨퍼런스 서버는 req-conf-event-min 값에 의해 이벤트 통지 메시지의 주기를 제어하여 이 값보다 크도록 유지하며 이 값 이내에서 발생하는 통지 메시지는 내부에서 버퍼링 함으로써 통지 메시지 발생량을 감소시킨다. 또 서버는 이벤트 통지 메시지의 주기가 req-conf-event-max 값 보다는 작게 유지되도록 하여 사용자에게 주기적으로 업데이트된 컨퍼런스 정보를 제공할 수 있게 한다.

SIP 메시지 제어 모듈은 컨퍼런스 서버와 참가자 사이에 세션이 계속 유지되도록 하기 위하여 필요한 주기적인 리프레시 동작 시 발생하는 통지 메시지의 발생량을 감소시킬 수 있도록 한다. 이를 위하여 컨퍼런스 서버는 참가자에게 보내는 첫 번째 NOTIFY 메시지의 Subscription-State 헤더에 유일한 엔티티 값을 넣은 속성 conf-refresh-entity를 포함시킨다. 컨퍼런스 참가자는 컨퍼런스 이벤트 패키지 등록 시간 연장을 위한 리프레시 과정이나 세션 종료 시 Event 헤더의 속성 conf-refresh-entity-match에 서버가 보낸 엔티티 값을 넣어 보냄으로써 통지 메시지가 필요 없음을 알린다. 서버는 이 값을 비교하여 일치하면 해당 참가자에 대한 통지 메시지를 생성하지 않음으로써 SIP 메시지 발생량을 감소시킬 수 있다.

또 SIP 메시지 제어 모듈에서는 컨퍼런스 정보 데이터베이스에서 컨퍼런스 주소 목록을 추출하여 해당 참가자들에게 직접 SIP INVITE 메시지를 보내어 컨퍼런스에 참가시킬 수 있는 기능을 제공한다. 따라서 컨퍼런스 서버와 참가자 사이에 필요한 SIP REFER 메시지 교환 절차가 필요 없게 되고 SIP 메시지 수를 더욱 줄일 수 있게 된다.

2. 컨퍼런스 이벤트 패키지 및 컨퍼런스 정보 데이터 구조 설계

본 논문에서 설계된 컨퍼런스 이벤트 패키지 및 컨퍼런스 정보 데이터 구조는 효율적인 SIP 메시지 제어를 위하여 다음과 같은 필요 요소들이 추가되었다. 먼저 컨퍼런스 이벤트 패키지에서 새로운 속성 req-conf-event-min 및 req-conf-

event-max가 추가되었다. 속성 req-conf-event-min는 컨퍼런스 정보의 변화가 있을 때 컨퍼런스 서버가 전송하는 SIP NOTIFY 메시지의 최소 간격을 표시하고 req-conf-event-max는 최대 간격을 표시한다. 그림 3에 이 파라미터의 사용 예를 보였다.

```
NOTIFY sip:sip-test.kumoh.ac.kr SIP/2.0
Via: SIP/2.0/TCP sip.kumoh.sc.kr;branch=qr38jar4f6t7
Max-Forwards: 15
From: <sip:gil3521@sip.kumoh.sc.kr>;tag=t8gj3pq4f
To: <sip:pk437@sip.kumoh.sc.kr>;tag=h93nsa4f8t
Call-ID: w4tm8qpa@sip-test.kumoh.ac.kr
CSeq: 54838031 NOTIFY
Contact: <sip:sip-test.kumoh.ac.kr>
Event: conference;req-conf-event-min=10;req-conf-event-max=50
Subscription-State: active;expires=3600
```

그림 3. 이벤트 통지 메시지 제어 파라미터 사용 예
Fig. 3. An Example of Using Parameter of Event Notification Messages Control

여기서는 SIP NOTIFY 메시지 간격을 최소 10초에서 최대 50초로 설정한 경우이다. 이와 같은 속성을 사용하면 참가자가 통신 대역폭의 제한을 받는 모바일 디바이스를 사용하는 경우 이벤트 통지 메시지 발생량을 적절히 제어할 수 있다.

속성 conf-refresh-entity와 conf-refresh-entity-match는 컨퍼런스 서버와 참가자 사이에 엔티티 값을 서로 교환할 수 있도록 한다. 이 기능을 사용하여 컨퍼런스 이벤트 패키지 등록 유지를 위해 반드시 필요한 리프레시 동작에서 SIP 메시지 발생량을 줄일 수 있다.

다음 루트 엘리먼트 태그 <conference-info>의 하위 요소로 새롭게 <participant-address-group>이 추가되었다. 이 태그는 컨퍼런스 참가자 그룹의 주소를 나타내며 하위 요소로 각 컨퍼런스 참가자 그룹을 구분하기 위한 태그 <participant-group-id>가 온다. 이 요소는 다시 하위 요소로 해당 그룹에 속한 각 컨퍼런스 참가자의 컨퍼런스 주소를 나타내는 <participant-conf-address>를 가진다.

컨퍼런스 참가자가 보내는 INVITE 메시지의 Contact 헤더에는 파라미터 'participant-group-id'가 추가되어 자신이 속한 컨퍼런스 그룹을 나타낸다. 이와 같은 기능을 사용하여 컨퍼런스 서버는 해당 참가자들에게 직접 SIP INVITE 메시지를 보내어 컨퍼런스에 참가시킬 수 있으므로 SIP 메시지 수를 더욱 감소시킬 수 있게 된다.

현재 컨퍼런스의 상태를 표시하기 위한 태그인 <conference-state>의 하위 요소로 현재 각 컨퍼런스 서버에 연결된 참가자를 나타내는 태그 <current-participants>가 추가되었다. 이 태그의 하위 요소로는 각 컨퍼런스 참가자를 구분하기 위한 태그 <participants-entity>와 각 참가자가 연결된 컨퍼런

스 서버를 구분하기 위한 태그 <conf- server-identity>가 추가되었다.

이와 같은 SIP 메시지 제어를 위한 필요 요소들이 새롭게 추가된 컨퍼런스 이벤트 패키지 및 컨퍼런스 정보 데이터 구조를 사용한 컨퍼런스 서버들과 참가자들 사이의 SIP 메시지 교환 절차를 그림 4에 보였다.

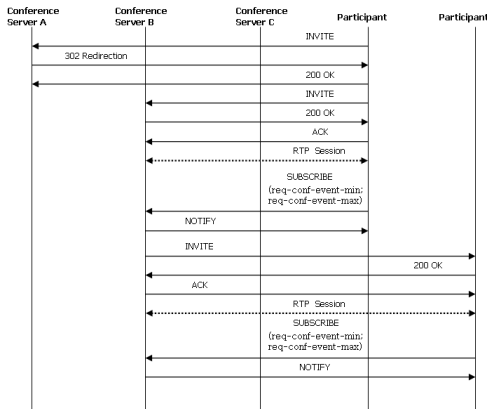


그림 4. 설계된 시스템에서의 SIP 메시지 교환 절차
Fig. 4 Exchange Procedure of SIP Messages in Designed System

여기서 컨퍼런스 참가자가 컨퍼런스 서버들 중 하나에게 INVITE 메시지를 보내면 해당 컨퍼런스 서버는 전체 부하 상황에 의해 자신보다 부하가 낮은 컨퍼런스 서버가 있는 경우 이 서버의 주소를 담은 SIP 302 Redirection 메시지를 응답 신호로 보낸다. 컨퍼런스 참가자는 이 메시지에 의해 새로운 컨퍼런스 서버(컨퍼런스 서버 B)에게 SIP INVITE 메시지를 보내고 이어서 미디어 전송을 위한 RTP 세션 연결을 맺은 후 SUBSCRIBE 메시지를 보낸다. 이때 Event 헤더에 NOTIFY 메시지의 최소 간격 및 최대 간격을 나타내는 파라미터로 각각 req-conf-event-min 및 req-conf-event-max를 사용한다.

다음 컨퍼런스 서버 B는 컨퍼런스 정보 데이터베이스를 액세스하여 해당 그룹에 속한 각 컨퍼런스 참가자의 컨퍼런스 주소를 얻어 해당 SIP 주소로 INVITE 메시지를 보내어 새로운 참가자를 컨퍼런스에 참가시킨다. 이 과정에서 SIP REFER 메시지 교환 절차가 필요 없어 필요한 SIP 메시지 수를 더욱 줄일 수 있다. 새로운 참가자는 컨퍼런스 서버와 RTP 세션을 맺은 후 SUBSCRIBE 메시지를 보내는데 이때에도 컨퍼런스 서버로부터 NOTIFY 메시지의 최소 간격 및 최대 간격을 나타내는 파라미터인 req-conf-event-min

및 req-conf-event-max이 사용된다.

그림 5는 속성 conf-refresh-entity와 conf-refresh-entity-match를 사용한 컨퍼런스 서버들과 참가자들 사이의 SIP 메시지 교환 절차이다.

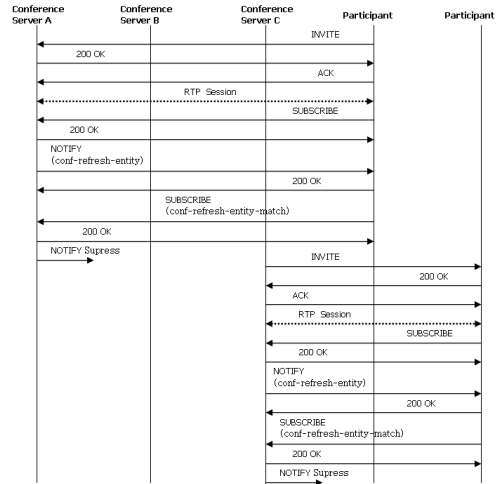


그림 5. 속성 conf-refresh-entity를 사용한 SIP 메시지 교환 절차
Fig. 5 Exchange Procedure of SIP Messages using Attribute conf-refresh-entity

여기서는 컨퍼런스 참가자가 컨퍼런스 서버 A에게 INVITE 요청을 하고 RTP 세션을 맺은 후 컨퍼런스 참가자의 SUBSCRIBE 메시지에 대한 응답으로 컨퍼런스 서버는 속성 conf-refresh-entity를 넣은 NOTIFY 메시지를 보낸다. 컨퍼런스 참가자가 리프레시를 위한 SUBSCRIBE 메시지 전송에 속성 conf-refresh-entity-match에 동일한 엔티티 값을 넣어 보내면 서버는 NOTIFY 메시지를 억제한다. 컨퍼런스 서버 C와 참가자 경우도 동일 한 동작을 한다. 이와 같이 컨퍼런스 서버와 참가자 사이에 엔티티 값이 서로 교환되도록 함으로써 컨퍼런스 이벤트 패키지 등록 유지를 위해 반드시 필요한 리프레시 동작에서 SIP 메시지 발생량을 더욱 줄일 수 있게 된다.

IV. 성능 분석

본 논문에서 제안한 분산형 컨퍼런스 모델에서의 효율적 SIP 메시지 제어 방식의 성능 분석을 위하여 두 개의 컨퍼런스 서버와 18개의 컨퍼런스 참가자 PC가 준비되었다. 컨퍼런스 참가자 PC의 운영체제는 MS 윈도우즈 XP이고 컨퍼런스 서버의 운영체제는 리눅스(커널 버전 2.6)이다. 모든 PC 및 서버는 100Mbps 속도의 동일한 LAN 세그먼트로 구성하였다.

먼저 본 논문에서 제안한 SIP 메시지 제어 방식을 적용한 분산형 컨퍼런스 모델과 기존의 분산형 컨퍼런스 모델과의 평균 지연시간을 비교 측정하였다. 그림 6에 이때의 측정 결과를 보였다.

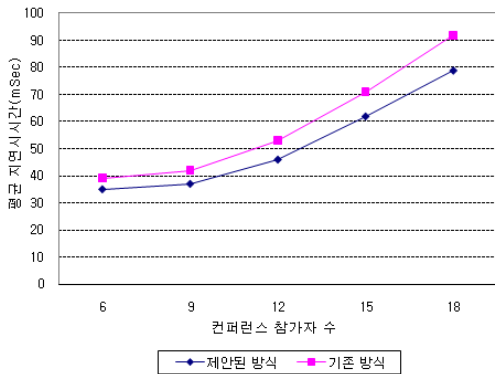


그림 6 분산형 컨퍼런스 모델에서의 평균지연시간 비교
Fig. 6 Comparison of Average Delay Time in Distributed Conference Model

여기서는 두 개의 컨퍼런스 서버를 모두 동작시키기 위하여 컨퍼런스 참가자 수가 6명 이상인 경우부터 측정하였다. SIP 메시지 제어 방식을 사용한 분산형 컨퍼런스 모델의 경우 컨퍼런스 서버가 처리해야 할 전체적인 메시지 양이 감소하여 부하를 감소시키므로 성능의 향상을 보여준다. 컨퍼런스 참가자 수가 9명 일 때는 제안된 분산형 컨퍼런스 모델에서의 평균 지연 시간이 11.9% 감소함을 보여주고 있고 참가자 수가 12명 일 때는 13.2% 감소되고, 참가자 수가 18명 일 때는 16.4%가 감소되어 참가자가 수가 증가할수록 본 논문에서 제안된 방식에 의한 평균 지연시간의 개선 효과가 커짐을 알 수 있다.

그림 7은 req-conf-event-min 및 req-conf-event-max 속성을 사용하여 이벤트 통지 메시지 발생률을 제어한 경우 컨퍼런스 서버에서 분담하는 평균 SIP 메시지 처리 시간을

비교 측정된 결과이다. 측정 결과는 참가자 수가 9명일 때 제안된 방식에서는 12.1%의 SIP 메시지 처리시간 감소를 보여 주고 있으며, 12명일 때 14.5% 감소하고 18명일 때 15.1% 감소함을 보여주고 있어 컨퍼런스 참가자 수가 증가 할수록 SIP 메시지 처리시간 감소 효과가 커짐을 보여주고 있다.

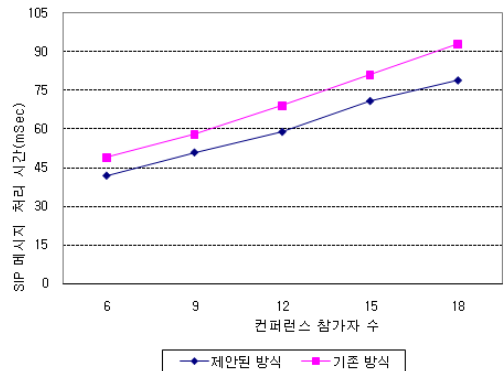


그림 7. 속성 req-conf-event-min를 사용한 경우의 평균 SIP 메시지 처리시간 비교
Fig. 7 Comparison of Average SIP Message Processing Time using Attribute req-conf-event-min

그림 8은 본 논문에서 제안한 SIP 메시지 제어 방식을 모두 사용 했을 경우 컨퍼런스 서버의 평균 SIP 메시지 처리 시간을 비교 측정된 결과이다. 속성 req-conf-event-min 및 req-conf-event-max만 사용한 경우에 비해 처리 시간 감소 효과가 더 커짐을 알 수 있다. 측정 결과 컨퍼런스 참가자 수가 9명일 때 기존 방식에 비해 15.5%의 처리시간이 감소하고, 12명일 때 18.8% 감소하며 18명일 때 19.4% 감소함을 보여주고 있다.

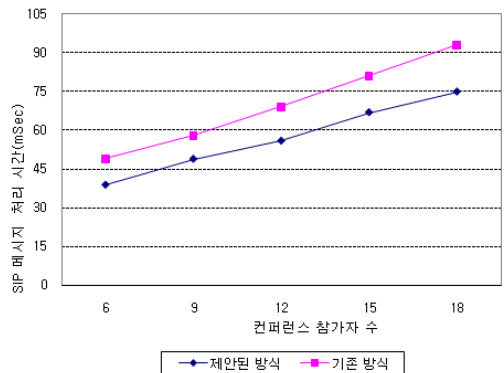


그림 8 평균 SIP 메시지 처리시간 비교
Fig. 8 Comparison of Average SIP Message Processing Time

V. 결론 및 향후 과제

본 논문에서는 SIP 환경에서의 분산형 컨퍼런스 모델에서 각 컨퍼런스 서버들의 부하를 줄여 확장성을 높일 수 있는 효율적인 SIP 메시지 제어 방식을 연구하였다. 이를 위하여 효율적인 SIP 메시지 제어에 필요한 구성요소들이 추가된 새로운 컨퍼런스 이벤트 패키지 및 컨퍼런스 정보 데이터 모델이 제시되었다. 컨퍼런스 참가자는 새로운 속성 req-conf-event-min 및 req-conf-event-max를 사용하여 자신의 처리 능력 및 네트워크 상황에 맞추어 컨퍼런스 서버로 부트의 이벤트 통지 메시지 발생량을 실시간으로 제어 할 수 있다. 아울러 새로운 속성 conf-refresh-entity와 conf-refresh-entity-match를 사용하여 컨퍼런스 서버와의 세션 유지를 위해 필요한 각종 SIP 메시지의 발생량도 능동적으로 제어 할 수 있다. 따라서 본 제어 방식을 사용하여 컨퍼런스 서버들이 처리해야 할 SIP 메시지를 크게 줄일 수 있어 대용량 컨퍼런스 시스템으로의 확장이 용이해지며 SIP 환경에서의 전체 네트워크 트래픽도 크게 감소시킬 수 있게 된다.

본 연구에서 제안한 컨퍼런스 시스템의 성능 분석을 위하여 컨퍼런스 참가자 수를 증가시켜 가면서 복수개의 컨퍼런스 서버에서의 처리 성능을 측정하였다. 평균 지연시간을 비교 측정한 결과 본 연구에서 제안한 방식에서 11.9%에서 16.4% 까지의 성능 개선 효과가 있음을 확인하였다. 또 평균 SIP 메시지 처리 시간을 15.5%에서 19.4% 까지 감소시킬 수 있음을 측정하였다. 향후 과제로는 컨퍼런스 서버와 컨퍼런스 참가자 수를 더욱 늘린 대규모 컨퍼런스 환경 및 무선 모바일 디바이스를 사용한 환경에서 본 논문이 제안한 방식의 성능 개선 효과를 연구 할 필요가 있다.

참고문헌

[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] 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.

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

[4] C. S. Jang, "A Distributed Conference Architecture

with a New Load Control Method" Journal of The Korea Society of Computer and Information, Vol 17, No.6, pp 103-108, June 2012.

[5] M. Barnes, C. Boulto and O. Levin, "A Framework for Centralized Conferencing," RFC 5239, June 2008.

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

[7] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control-Conferencing for User Agents," RFC 4353, Feb. 2006.

[8] 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.

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

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

[11] 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.

저 자 소개



장 춘 서

1993년 2월 : 한국과학기술원 공학박사
1981년 3월 ~ 현재 :
금오공과대학교 컴퓨터공학과 교수
<관심분야> : SIP, 임베디드 시스템,
인터넷텔레포니
Email : csjang@kumoh.ac.kr



이 기 수

1982년 2월 : 서울대학교 대학원
공학석사
1982년 3월 ~ 현재 :
금오공과대학교 컴퓨터공학과 교수
<관심분야> : SIP, 디지털시스템,
데이터베이스
Email : kysoolee@kumoh.ac.kr

