

확장 제어 채널 프레임워크를 사용한 다중 컨퍼런스 서버 구조

장춘서*

Multiple Conference Servers Architecture using Extended Control Channel Framework

Choonseo Jang*

Department of Computer Engineering, Kumoh Inst. of Tech. Gumi 39177, Korea

요 약

본 논문에서는 SIP(session initiation protocol) 세션에서 확장 제어 채널 프레임워크를 사용하여 대용량 컨퍼런스 시스템에 적용할 수 있는 새로운 다중 컨퍼런스 서버 구조를 제안하였다. 이를 위하여 본 연구에서는 전체 컨퍼런스 시스템의 부하를 복수개의 서버에서 효율적으로 분산 처리하기 위한 새로운 확장된 제어 채널 프레임워크 구조가 제시되었다. 각 서버 사이에 SIP 세션을 설정하여 제어 채널을 연결하고, 이 제어 채널을 통하여 전송될 수 있는 확장 제어 채널 메시지가 설계되었다. 구현된 확장 제어 채널 메시지를 사용하여 다중 컨퍼런스 서버들 사이에 부하가 효율적으로 분산될 수 있도록 하였고, 확장 제어 채널 프레임워크에 맞춘 컨퍼런스 정보 데이터 포맷도 설계되었다. 아울러 확장 제어 채널 메시지의 교환 절차도 함께 제시되었다. 제안된 시스템의 성능은 시뮬레이션 실험을 통하여 분석하였다. 분석 결과 평균 SIP 메시지 지연 시간과 평균 미디어 스트림 지연 시간이 향상됨을 보였다.

ABSTRACT

In this paper, a new architecture of multiple conference servers which use extended control channel framework in SIP(session initiation protocol) session has been presented. For this purpose, in this study, a new extended control channel framework architecture which can distribute total conference system loads to multiple servers effectively has been presented. In the implementation, extended control channels have been connected by using SIP sessions that was established between each conference servers, and extended control channel messages which can be transferred through control channels have been designed in this study. These extended control channel messages can distribute system load effectively between multiple conference servers, and conference information data format that can represent extended control channel framework has also been designed. Furthermore, exchange procedures of extended control channel messages have also been presented. The performance of the proposed system has been analysed by simulation. The analysis results show that average SIP messages delay time and average media stream delay time have improved.

키워드 : SIP, 컨퍼런스 시스템, SDP, 제어 채널 프레임워크

Key word : SIP, Conference System, SDP, Control Channel Framework

Received 20 March 2017, Revised 21 March 2017, Accepted 05 April 2017

* Corresponding Author Choonseo Jang(E-mail:csjang@kumoh.ac.kr, Tel:+82-54-478-7521)

Department of Computer Engineering, Kumoh Inst. of Tech., Gumi 39177, Korea

Open Access <https://doi.org/10.6109/jkiice.2017.21.7.1335>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

다량의 참가자를 처리해야하는 대용량 컨퍼런스 시스템[1-3]에서는 전체 시스템 부하를 복수개의 컨퍼런스 서버로 분산시켜야 한다. 제어 채널 프레임워크는 신뢰성 있는 트랜스포트 세션 연결을 위해 클라이언트와 서버 사이에 SIP[4] 메시지를 서로 교환하며 세션 설정 후 제어 채널 메시지를 사용하여 서버를 제어 할 수 있다. 그러나 다중 서버로 구성된 대용량 컨퍼런스 시스템에 이를 적용하는 효율적 방안은 아직 제시되고 있지 않다. 따라서 본 논문에서는 다중 서버로 구성된 대용량 컨퍼런스 시스템에 사용 할 수 있는 새로운 확장 제어 채널 프레임워크 구조를 제안하였다.

이 방식에서는 대용량 컨퍼런스 시스템의 전체 부하를 복수개의 서버에 분산시키기 위하여 본 논문에서 구현된 확장 제어 채널을 사용하여 각 서버들의 부하 레벨을 모니터링 하고 부하 레벨이 작은 서버에게 참가자들을 분산시키며, 동작 중인 서버들이 컨퍼런스 정책으로 설정된 최대 부하 레벨을 모두 초과 할 경우 새로운 서버를 추가한다. 이때 컨퍼런스 전체 시스템의 정보를 담고 있는 컨퍼런스 정보 데이터[5, 6] 구조도 컨퍼런스 서버 사이에 교환되는 확장 제어 채널 메시지에 맞추어 변경되어야 하므로 이를 위한 확장 컨퍼런스 정보 데이터 포맷도 함께 설계되었다. 아울러 컨퍼런스 서버들 사이의 부하 분산을 위한 확장 제어 채널 메시지의 교환 절차도 함께 제시되었다. 본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 제어 채널 프레임워크와 컨퍼런스 시스템 구성 및 기존 연구에 대해 설명하고 III장에서는 본 논문에서 새롭게 제안하는 확장 제어 채널 프레임워크를 사용한 다중 컨퍼런스 서버 구조를 설명한다. 또 확장 제어 채널 메시지들과 연동하여 동작하는 확장 컨퍼런스 정보 데이터 포맷의 설계와 확장 제어 채널 메시지의 교환 절차에 대해 설명한다. IV장에서는 제안된 컨퍼런스 시스템의 성능 실험 및 분석을 하고 V장에서 결론을 맺는다.

II. 관련연구

2.1. 제어 채널 프레임워크

그림 1에 제어 채널 프레임워크의 기본적인 동작을

보였다.

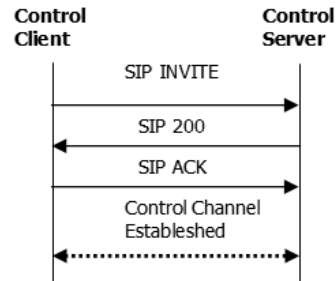


Fig. 1 Basic Operation of Control Channel Framework

여기서 제어 클라이언트는 서버에게 SIP INVITE 메시지를 보낸다. 이때 SIP 메시지 몸체 부분의 Content-Type 헤더는 'application/sdp'이고 SDP(Session Description Protocol)[7] 미디어 디스크립션 엔트리에는 TCP 포트번호와 포맷값으로 'cfw'가 들어간다. TCP 포트번호는 서버와 제어 채널 연결에 사용된다. 속성 엔트리에는 'setup:active', 'connection:new' 및 'cfw-id' 값이 들어가고 'cfw-id'는 서버와 제어 채널 프레임워크 메시지 교환에 사용할 ID 값이다. 서버는 클라이언트의 SIP INVITE 메시지에 대한 응답으로 Content-Type 헤더를 'application/sdp'로 하고 SDP 미디어 디스크립션 엔트리에 서버가 앞으로 사용할 TCP 포트번호와 속성 엔트리에 'setup:active' 및 자신의 제어 채널 ID 값을 'cfw-id'에 넣어 보내온다. 다음 클라이언트에서 SIP ACK 메시지를 보내면 서버와 제어 채널이 설정된다. 이후 제어 채널을 통하여 클라이언트와 서버 사이에 메시지들이 교환된다.

2.2. 컨퍼런스 시스템

기본적인 컨퍼런스 시스템은 컨퍼런스 미디어 믹서와 포커스로 구성된다. 컨퍼런스 참가자에서 발생된 오디오/비디오 패킷 스트림은 미디어 믹서에서 각각의 소스를 구분하여 혼합되고 RTP(real-time transport protocol)를 사용하여 각 참가자들에게 전송된다. 참가자는 컨퍼런스 포커스에게 SIP 메시지를 보내어 SIP 컨퍼런스 세션을 설정하고 이후 컨퍼런스 포커스는 이 컨퍼런스 세션을 유지하는 기능을 한다.

컨퍼런스 시스템의 모든 정보는 컨퍼런스 정보 데이터 포맷을 따르며 최상위 엘리먼트로 <conference-info>

를 갖는 XML 문서 application/conference-info+xml 형태이다. <conference-info>의 속성은 컨퍼런스 URI를 나타내는 'entity', 컨퍼런스 정보의 버전인 'version' 및 전체정보 또는 부분정보 여부를 나타내는 'state'를 가진다. 이 엘리먼트의 하위 엘리먼트로 컨퍼런스 전체 정보를 나타내는 <conference-description>, 현재 컨퍼런스 상태를 표시하는 <conference-state> 및 컨퍼런스 참가자들에 대한 개별 정보를 나타내는 <users> 등이 있다. <users>는 복수개의 <user> 엘리먼트를 가지며 이는 각 참가자들의 컨퍼런스 정보를 나타낸다. 이 엘리먼트는 속성으로 컨퍼런스 참가자에 대한 URI를 표시하는 'entity'를 가진다. <user>의 하위 엘리먼트로는 <endpoint>가 있으며 이는 각 참가자들의 SIP 신호 세션에 대한 정보와 미디어 장치를 나타낸다. 이의 하위 엘리먼트로 참가자에서 발생한 미디어 소스 위치 등 미디어 스트림에 대한 정보를 가지고는 <media>가 있다.

현재까지 연구되고 있는 대용량 컨퍼런스 시스템의 경우 일반 SIP 메시지를 사용하여 서버의 부하를 분산시키는 방식[8]이나 CCMP를 사용하는 방식[9] 등이 있으나 제어 채널 프레임워크를 적용한 경우는 없다. 따라서 본 연구에서는 서버와 서버 사이에 신뢰성 있는 세션 연결을 통한 확장 제어 채널 프레임워크를 사용하여 전체 컨퍼런스 시스템의 부하를 복수개의 서버들에게 실시간으로 분산하며, 동작 중인 서버들이 컨퍼런스 정책으로 설정된 최대 부하 레벨을 모두 초과 할 경우 새로운 서버를 실시간으로 추가하는 컨퍼런스 서버 구조를 설계하였다. 이를 위하여 확장 제어 채널 프레임워크에 맞추어 확장된 컨퍼런스 정보 데이터 구조와 컨퍼런스 서버들 사이의 부하 분산을 위한 제어 채널 메시지의 교환 절차도 함께 제시되었다.

III. 시스템 설계 및 구현

3.1. 확장 제어 채널 프레임워크를 사용한 컨퍼런스 서버 구조 설계

그림 2는 본 논문에서 설계한 확장 제어 채널 프레임워크를 사용한 컨퍼런스 서버의 구조이며 확장 제어 채널 모듈, 부하 제어 모듈, 확장 컨퍼런스 정보 데이터베이스, SIP 스택 및 컨퍼런스 포커스와 미디어 믹서로 구성되어 있다.

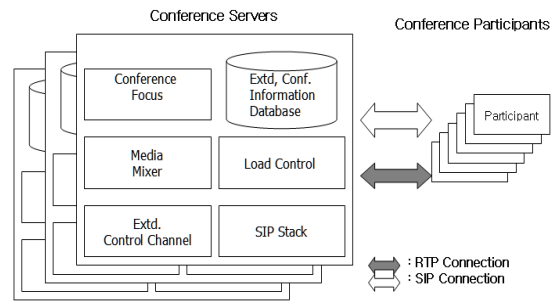


Fig. 2 Architecture of Conference Server using Extended Control Channel Framework

컨퍼런스 포커스는 SIP 스택 모듈을 통하여 참가자와 SIP 메시지를 교환하여 컨퍼런스 세션을 설정하고 관리하며, 미디어 믹서는 각 참가자로부터 생성된 오디오/비디오 패킷 스트림을 혼합하고 분배한다. 확장 제어 채널 모듈은 다중 서버 부하 제어를 위하여 본 논문에서 새롭게 설계된 확장 제어 채널 메시지들을 처리하며, 부하 제어 모듈은 각 컨퍼런스 서버의 부하를 모니터링하고 각 서버의 부하를 균등하게 제어하며 필요시 새로운 서버를 추가하는 기능을 제공한다. 확장 컨퍼런스 정보 데이터베이스는 기존 컨퍼런스 정보 데이터 포맷을 확장하여 다중 서버 부하 제어에 필요한 엘리먼트들이 포함되도록 설계되었다.

주컨퍼런스 서버는 동작 중인 각 서버들에게 콘텐츠 타입이 'application/extd-sdp'인 SIP INVITE 메시지를 보내며 이때 SDP 미디어 디스크립션 엔트리의 포맷은 'extd-cfw' 이다. 각 서버는 이 SIP INVITE 메시지에 대한 응답으로 SDP 속성(attribute) 엔트리에 자신의 제어 채널 ID 값을 'extd-cfw-id'에 넣어 보내온다. 다음 SIP ACK 메시지에 의해 서버 사이에 제어 채널이 설정되고 이후 이 제어 채널을 통하여 각 서버 사이에 제어 메시지가 교환된다. 그림 3에 확장 제어 채널 설정을 위한 SDP 부분의 예를 보였다.

참가자가 주 컨퍼런스 서버에게 참가 요청을 하면 주 컨퍼런스 서버의 부하 제어 모듈(load control module)은 동작 중인 각 서버의 현재 부하 레벨을 조사하여 가장 작은 값을 가지는 서버의 URI를 참가자에게 알려준다. 참가자는 이 URI에 의해 지정된 서버에게 SIP INVITE 메시지를 보내고 이후 서버와 SIP 세션을 연결하고 이어서 RTP 세션을 연결해 오디오/비디오 스트림을 서버와 교환하게 된다. 동작 중인 각 서버의 현재 부

하 레벨은 <current-load-level> 엘리먼트로 나타내며 각 서버가 초 당 처리하는 SIP 메시지 양과 미디어 패킷 수를 기준으로 산출되고 컨퍼런스 정보 데이터 포맷을 확장하여 이 엘리먼트가 포함되도록 하였다.

```

INVITE sip:channel-con@sip-server.test SIP/2.0
To: <sip:channel-con@sip-server.test>
From: <sip:test9@sip-server.test>;tag=466541
CSeq: 1 INVITE
Max-Forwards: 10
Call-ID: 4375gser@sip-server.test
Contact: <sip:test9@ss4.sip-server.test>
Content-Type: application/extd-sdp
Content-Length: 74

o=originator 183472345 183479732 IN IP4
channel-con.sip-server.test
c=IN IP4 test9.sip-server.test
m=application 8623 TCP extd-cfw
a=extd-cfw-id:vkerufav23fjl
a=connection:new
a=setup:active
    
```

Fig. 3 Example of SDP Messages for Extended Control Channel

동작 중인 모든 컨퍼런스 서버의 부하 레벨이 컨퍼런스 관리자에 의해 설정된 최대값을 보이는 경우라면 주 컨퍼런스 서버는 새로운 서버를 추가하는 과정을 시작한다. 먼저 추가 가능한 서버의 URI를 확장 컨퍼런스 정보 데이터베이스에서 검색하여 선택된 서버에게 콘텐츠 타입이 'application/extd-sdp', SDP 미디어 디스크립션 엔트리의 포맷은 'extd-cfw' 이며 속성 엔트리 값이 'connection:new' 인 SIP INVITE 메시지를 보낸다. 해당 서버로부터 콘텐츠 타입 'application/extd-sdp', 속성 엔트리 'extd-cfw-id'에 자신의 제어 채널 ID 값이 들어있고 'setup:passive' 상태인 SIP 응답이 오면 해당 서버와 제어 채널 세션이 설정된다. 다음 주 컨퍼런스 서버는 확장 제어 채널 프레임워크 메시지를 사용하여 새롭게 추가된 서버에게 전체 컨퍼런스 정보 데이터를 전송하고 참가자에게 이 서버의 URI를 전송하여 SIP 연결을 설정 하도록 한다.

3.2. 확장 제어 채널 프레임워크 설계

본 연구에서는 전체 컨퍼런스 시스템의 부하를 다중 서버로 분산 할 수 있도록 확장 제어 채널 프레임워크

를 설계하였다. 콘텐츠 타입이 'application/extd-sdp' 인 확장 SDP 포맷을 담은 SIP INVITE 메시지가 컨퍼런스 서버 사이에 교환되면 제어 채널 세션이 설정되고 이어서 extd-SYNC 메시지가 전송되어 서버 사이에 세션을 유지할 시간간격과 사용할 패키지명을 상대방에게 알려준다. 다음 extd-CONTROL 메시지를 사용하여 상대방에게 컨퍼런스 제어에 필요한 메시지를 보내고 extd-REPORT 메시지를 사용하여 이에 대한 응답을 수신하게 된다. 그림 4에 이 메시지의 예를 보였다.

```

extd-CFW 485vhkv8eur extd-SYNC
Dialog-ID:vkerufav23fjl
Keep-Alive: 50
Packages: extd-distr-conf-control/0.9

extd-CFW 485vhkv8eur 200
Keep-Alive: 50
Packages: extd-distr-conf-control/0.9

extd-CFW 53fhrtu21y extd-CONTROL
Control-Package: extd-distr-conf-control/0.9
Content-Type: extd-distr-conf-fmt/xml
Content-Length: 164

extd-CFW 53fhrtu21y extd-REPORT
Seq: 1
Status: update
Timeout: 50
Content-Type: extd-distr-conf-fmt/xml
Content-Length: 274
    
```

Fig. 4 Example of Extended Control Channel Framework Messages

이와 같은 확장 제어 채널 프레임워크를 통하여 전체 시스템의 정보를 담은 컨퍼런스 정보 데이터가 전송되며 다중 서버 구조에 맞추어 컨퍼런스 정보 데이터 포맷이 확장 설계되었다. 동작 중인 각 서버의 현재 부하 레벨을 나타내기 위하여 <current-load-level> 이 최상위 엘리먼트인 <conference-info>의 하위 엘리먼트로 추가되었다. <current-load-level> 의 속성으로 최대 부하 허용레벨과 최소 부하 허용레벨인 'max-load-level' 과 'min-load-level' 을 두었고 하위 엘리먼트로는 개별 서버의 부하 레벨을 나타내는 <individual-load-level> 를 두었다. 최대 부하 허용레벨은 컨퍼런스 서버를 시스템에 추가하는 기준이 되고 최소 부하 허용레벨은 부하가 너무 낮은 서버는 시스템에서 제외시켜 대기 상

태로 두기 위한 것이다. <individual-load-level>의 속성으로는 각 서버를 구분하기 위한 'svr-label'을 두었고 <reserved-svr-uri>는 현재 대기 중에 있는 컨퍼런스 서버들의 URI를 나타내어 필요시 추가될 수 있도록 설계되었다.

다음 <current-svr-number> 엘리먼트가 현재 동작 중인 컨퍼런스 서버의 개수를 나타내기 위하여 사용되었고 <current-svr-users>는 각 서버가 담당하는 사용자들에 대한 연결 정보 및 오디오/비디오 정보 등 컨퍼런스 정보 데이터를 나타내기 위해 설계되었다. 이는 개별 사용자 정보를 가진 하위 엘리먼트 <user-info>를 가진다. 속성으로는 사용자가 속한 서버를 구분하기 위한 'svr-label'과 각 서버에 연결된 사용자 숫자를 나타내기 위한 'users-count'를 두었다. 그림 5에 설계된 시스템에서의 메시지 교환 절차를 보였다.

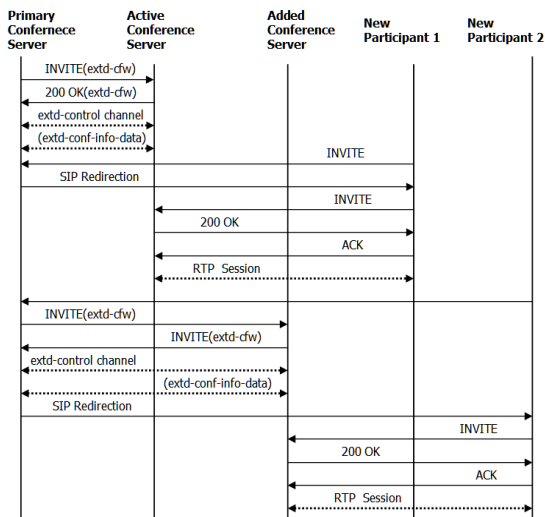


Fig. 5 Messages Exchange Procedure for Designed Conference System

그림 5에서 주 컨퍼런스 서버는 동작 중인 컨퍼런스 서버에게 'extd-cfw' 형식의 SIP INVITE 메시지를 보내면 서버는 이에 대한 응답으로 자신의 제어 채널 ID 값을 넣어 보내오고 이어서 양단간에 확장 제어 채널이 설정된다. 이후 이 제어 채널을 통하여 확장 컨퍼런스 정보 데이터가 전송된다. 새로운 참가자 1이 주 컨퍼런스 서버에게 SIP INVITE 메시지를 보내면 현재 각 서버의 부하 레벨을 조사하여 해당 서버의 URI를 참가자

에게 알려준다. 참가자는 이 서버와 SIP 및 RTP 세션을 설정해 컨퍼런스에 참가하게 된다. 새로운 참가자 2가 주 컨퍼런스 서버에게 SIP INVITE 메시지를 보내면 주 컨퍼런스 서버는 새로 추가할 서버와 확장 제어 채널을 설정하고 이 채널을 통하여 확장 컨퍼런스 정보 데이터를 전송한다. 다음 추가된 서버의 URI를 이 참가자에게 알려주면 해당 참가자는 추가된 서버와 SIP 및 RTP 세션을 설정해 컨퍼런스에 참가하게 한다.

IV. 성능 분석

본 논문에서는 연구된 컨퍼런스 시스템의 성능을 시뮬레이션을 사용하여 분석하였다. 먼저 참가자 수 증가에 따른 평균 SIP 메시지 지연 시간을 측정할 결과를 그림 6에 보였다.

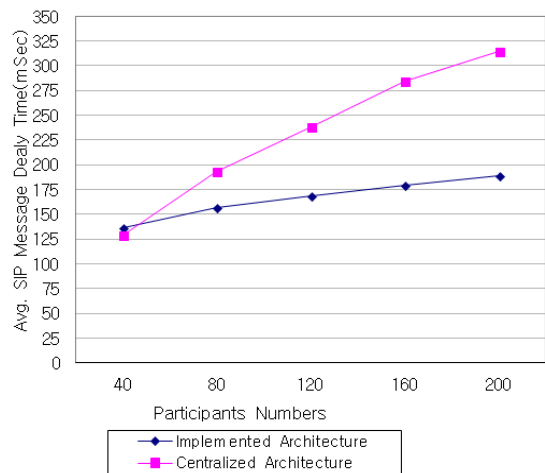


Fig. 6 Average SIP Messages Delay Time

여기서는 본 논문에서 연구한 컨퍼런스 구조와 기존의 중앙 집중형 컨퍼런스 구조를 비교하였고 최대 부하 레벨을 200으로 하여 참가자 수 40명 단위로 서버가 추가 되도록 하였다. 따라서 40명 경우 지연 시간의 차이가 거의 없으나 80명부터는 본 논문에서 연구한 컨퍼런스 구조의 경우 기존 방식에 비해 지연 시간이 감소됨을 보여준다. 측정 결과 참가자 수 80명인 경우 18.6%, 120명인 경우 28.9%, 200명인 경우 39.8%가 각각 감소함을 보여주었다.

다음 평균 미디어 스트림 지연시간을 비교 측정된 결과를 그림 7에 보였다.

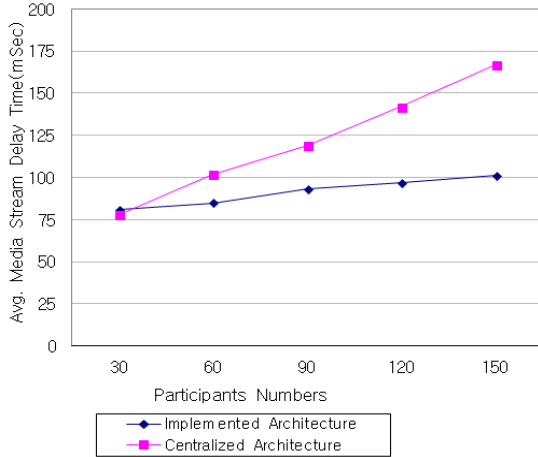


Fig. 7 Average Media Stream Delay Time (Max. Load Level=150)

그림 7에서 처음에는 컨퍼런스 서버 한 개로 동작하다가 점차 참가자 수가 증가하여 최대 부하 레벨을 넘으면 새로운 컨퍼런스 서버가 추가되는 식으로 동작한다. 각 컨퍼런스 서버의 최대 부하 허용값은 150으로 하였고 구현된 방식의 경우 30명 단위로 서버가 추가되어 부하를 분산 처리하게 된다.

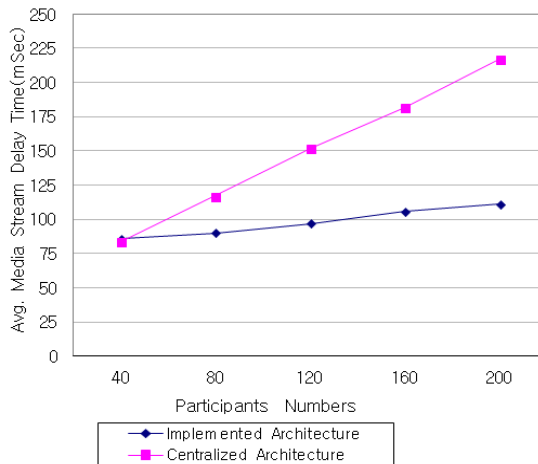


Fig. 8 Average Media Stream Delay Time (Max. Load Level=200)

이 경우 참가자 수 30명까지는 1개의 서버 만 사용하므로 양쪽은 동일 한 성능을 보이거나 구현된 방식의 경우 참가자 수 60명에서는 평균 지연 시간이 16.7% 감소하고, 90명에서는 21.8%, 150명에서는 39.5% 각각 감소한다. 따라서 제안된 구조에서의 평균 미디어 스트림 지연시간의 개선 효과가 커짐을 알 수 있다.

다음 최대 부하 레벨을 200으로 하여 측정된 결과를 그림 8에 보였다. 여기서는 최대 부하 레벨의 상승으로 참가자 수 40명 단위로 컨퍼런스 서버가 추가된다. 측정 결과 참가자 수 80명인 경우 23.0%, 120명인 경우 36.2%, 200명인 경우 48.8%로 평균 미디어 스트림 지연시간이 각각 감소함을 보여준다.

V. 결론

본 논문에서는 전체 컨퍼런스 시스템의 부하를 복수 개의 서버에서 효율적으로 분산 처리하기 위하여 확장 제어 채널 프레임워크를 사용한 다중 컨퍼런스 서버 구조를 연구하였다.

이 구조에서는 각 서버 사이에 설정된 SIP 세션에서 제어 채널을 연결하고 이 제어 채널을 통하여 다중 컨퍼런스 서버들 사이에 부하를 분담 할 수 있는 확장 제어 채널 메시지가 교환되도록 구현되었고 이와 함께 확장 제어 채널 프레임워크와 연동되어 동작 할 수 있는 컨퍼런스 정보 데이터 포맷도 설계되었다. 아울러 컨퍼런스 서버 사이의 확장 제어 채널 메시지의 교환 절차도 함께 제시되었다.

구현된 컨퍼런스 구조의 성능 분석을 위하여 평균 SIP 메시지 지연 시간과 평균 미디어 스트림 지연 시간을 측정하였다. 측정 결과 평균 SIP 메시지 지연 시간의 경우 참가자 수에 따라서 18.6%에서 39.8% 까지 개선 효과가 있음을 보여주었다. 평균 미디어 스트림 지연 시간의 경우 참가자 수에 따라서 16.7%에서 48.8% 까지 개선 효과가 있음을 보여주었다.

향후 과제로는 컨퍼런스 참가자 수를 더욱 증가시킨 대용량 컨퍼런스 시스템 환경에서 확장 제어 채널을 통한 제어 절차 및 메시지 교환 과정을 더욱 최적화 할 필요가 있다.

ACKNOWLEDGMENTS

This paper was supported by Research Fund, Kumoh National Institute of Technology.

REFERENCES

- [1] R. Shekh-Yusef and M. Barnes. (2013, December). Indication of Conference Focus Support for the Centralized Conferencing Manipulation Protocol. *IETF* [Online] RFC 7082. Available: <https://www.ietf.org/rfc/rfc7082.txt>.
- [2] J. Rosenberg. (2006, Feb.). A Framework for Conferencing with the Session Initiation Protocol (SIP). *IETF* [Online] RFC 4353. Available: <https://www.ietf.org/rfc/rfc4353.txt>.
- [3] M. Barnes et al, (2008, June), A Framework for Centralized Conferencing. *IETF* [Online] RFC 5239. Available: <https://www.ietf.org/rfc/rfc5239.txt>.
- [4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler. (2002, June). Session Initiation Protocol. *IETF* [Online] RFC 3261 . Available: <https://www.ietf.org/rfc/rfc3261.txt>.
- [5] O. Novo, G. Camarillo and D. Morgan (2012, March). Conference Information Data Model for Centralized Conferencing. *IETF* [Online] RFC 6501. Available: <https://www.ietf.org/rfc/rfc6501.txt>.
- [6] G. Camarillo, S. Srinivasan and J. Urpalainen. (2012, March). Conference Event Package Data Format Extension for Centralized Conferencing. *IETF* [Online] RFC 6502 . Available: <https://www.ietf.org/rfc/rfc6502.txt>.
- [7] F. Andreassen. (2010, Sept.). Session Description Protocol (SDP) Capability Negotiation. *IETF* [Online] RFC 5939. Available: <https://www.ietf.org/rfc/rfc5939.txt>.
- [8] C. Jang, "Distributed Large-Scale Conferencing System Architecture with Dynamic Server Allocation in SIP Environment," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no.5, pp.411-428, May 2015.
- [9] C. Jang, "A New Distributed Conference System Architecture using Extended CCMP in SIP Environment," *Journal of Korea Inst. of Information and Communication Eng.*, vol. 20, no.12, pp.2252-2258, Dec. 2016.



장춘서(Choonsoo Jang)

한국과학기술원 공학박사

현재 : 금오공과대학교 컴퓨터공학과 교수

※관심분야 : SIP, 임베디드 시스템, 인터넷텔레포니