

미디어 제어 채널/분산 컨퍼런스 매니플레이션 구조에 의한 다중 서버 컨퍼런스 서비스 시스템

장춘서*

A Multiple Servers Conference Service System by Media Control Channel/Distributed Conference Manipulation Architecture

Choonseo Jang*

요약 본 논문에서는 미디어 제어 채널/분산 컨퍼런스 매니플레이션 구조를 사용한 새로운 다중 서버 컨퍼런스 서비스 시스템을 제안하였다. 이 컨퍼런스 서비스 시스템은 다수의 컨퍼런스 참가자로부터 발생하는 부하를 효율적으로 처리할 수 있다. 이 제안된 구조에서는 각 컨퍼런스 서버 사이에 미디어 제어 채널을 설정하고, 이 제어 채널을 통해 시스템 부하를 제어하기 위한 분산 컨퍼런스 매니플레이션 메시지들이 서로 교환된다. 이 메시지는 안정된 트랜스포트 세션 연결 후 생성되는 미디어 제어 채널을 사용하여 컨퍼런스 서버 사이에서 전송되며, 컨퍼런스 참가자 수에 따른 컨퍼런스 서버의 부하를 효율적으로 분산 처리할 수 있다. 따라서 이 방식으로 대용량 컨퍼런스 서비스 시스템의 구현을 용이하게 할 수 있다. 이를 위하여 미디어 제어 채널을 통해 전송되는 분산 컨퍼런스 매니플레이션 메시지 구조가 설계되었고, 각 컨퍼런스 서버 사이의 메시지 교환 절차도 아울러 제시되었다. 제안된 다중 서버 컨퍼런스 서비스 시스템의 성능은 실험으로 분석하였고, 사용자 수에 따라 성능 개선 효과가 있음을 확인하였다.

Abstract In this paper, a new multiple servers conference service system using media control channel /distributed conference manipulation architecture has been presented. This conference service system can handle loads effectively from many conference participants. In this suggested architecture, media control channels are established between conference server, and distributed conference manipulation messages for distributing system loads are exchanged through this channels. These messages are transported between servers using media control channel created after stable transport sessions, and can be used to process server loads according to participants effectively. So this method can be used to implement large scale conference service system. For these purposes, formats of distributed conference manipulation messages which transmitted through media control channels are designed. and messages exchange procedures between conference servers are also presented. The performance of the proposed conference service system has been analysed by experiments, and the results show that the performances are improved according to participants.

Key Words : Conference System, Session Initiation Protocol, Conference Manipulation, Media Control Channel, Session Description Protocol

1. 서론

다수의 컨퍼런스 참가자로부터 발생하는 부하를 처

리해야 하는 컨퍼런스 서비스 시스템[1][2]에서는 다중 서버를 사용하여 서비스 시스템의 부하를 분산 처리해

This research was supported by Kumoh National Institute of Technology(2018-104-138).

*Corresponding Author : Department of Computer Engineering, Kumoh National Institute of Technology
 Received May 06, 2019 Revised June 11, 2019 Accepted June 11, 2019

야 한다. 이때 컨퍼런스 서버 사이에 부하를 균형 있게 제어 할 수 있는 방식이 중요하며 본 논문에서는 각 컨퍼런스 서버 사이에 미디어 제어 채널[3]을 설정하고 이 제어 채널을 통해 시스템 부하를 제어하기 위한 분산 컨퍼런스 매니플레이션 메시지를 서로 교환하는 방안을 제시하였다.

이 방안에서는 각 컨퍼런스 서버 사이에 SIP(session initiation protocol)[4] INVITE 메시지를 사용하여 신뢰성 있는 트랜스포트 세션 연결을 맺고 이 세션을 통해 먼저 서버 사이에 미디어 제어 채널 동기 메시지를 교환한다. 다음 서버 제어 요청을 위한 기능이 포함된 제어 메시지가 보내지고 이에 대해 상태 코드가 담긴 응답 메시지가 전송된다. 이때 미디어 제어 채널을 통해 전송되는 분산 컨퍼런스 매니플레이션 메시지의 포맷은 컨퍼런스 오브젝트의 생성과 서버 사이에 컨퍼런스 정보 데이터의 교환 등 다중 서버 구조에 적용 할 수 있도록 확장 설계되었다.

본 논문에서 설계된 미디어 제어 채널을 통한 분산 컨퍼런스 매니플레이션 메시지를 사용하여 각 컨퍼런스 서버들의 부하 상태를 알 수 있고 컨퍼런스 참가자들을 부하가 작은 서버들에게 할당 시킬 수 있으며 동작 중인 서버들의 부하 상태에 따라 필요시 새로운 컨퍼런스 서버를 추가 할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 미디어 제어 채널과 컨퍼런스 서비스 시스템 및 기존 연구에 대해 설명하고 III장에서는 본 논문에서 새롭게 제안하는 미디어 제어 채널/분산 컨퍼런스 매니플레이션을 사용한 다중 서버 컨퍼런스 서비스 시스템 구조를 설명한다. 또 미디어 제어 채널/분산 컨퍼런스 매니플레이션 방식에 사용되는 제어 메시지 교환 절차에 대해서도 다룬다. IV장에서는 제안된 컨퍼런스 서비스 시스템의 성능 실험 및 분석을 하고 V장에서 결론을 맺는다.

2. 관련연구

2.1 미디어 제어 채널

그림 1에 컨퍼런스 서버 사이에 연결되는 미디어 제어 채널의 기본 구성을 보였다.

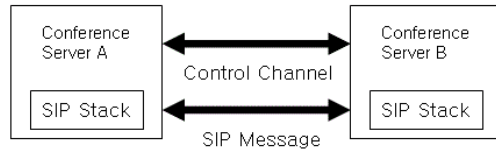


그림 1. 컨퍼런스 서버간 미디어 제어 채널의 기본 구성
Fig. 1. Basic Configuration of Media Control Channel between Conference Servers

그림 1에서 컨퍼런스 서버 A는 컨퍼런스 서버 B에게 Content-Type 헤더가 'application/sdp' 인 SIP INVITE 요청을 한다. 이때 SDP[5] 미디어 디스크립션 항목값으로 미디어 포맷 'cfw'와 양단간의 통신에 사용할 TCP 포트번호가 들어간다. 바디 부분의 속성(attribute) 항목의 값은 'a=setup:active', 'a=connection:new' 및 미디어 제어 채널 메시지 교환에 사용할 ID 값이 들어간 'a=cfw-id:gfhjhnre72u' 식으로 된다. 컨퍼런스 서버 B는 이에 대해 'application/sdp'를 Content-Type 헤더 값으로 하고 미디어 디스크립션 항목에 자신이 사용할 TCP 포트 번호가 들어있는 응답 메시지를 보내온다. 이때 이 메시지의 속성 항목에는 'a=setup:passive', 'a=connection:new' 가 들어있고 'cfw-id' 항목에는 자신의 사용할 제어 채널 ID 값이 들어있다. 다음 컨퍼런스 서버 A에서 SIP 확인 메시지를 보내오면 양단간에 미디어 제어 채널이 설정된다.

2.2 컨퍼런스 서비스 시스템

그림 2에 컨퍼런스 서비스 시스템의 기본 구성을 보였다.

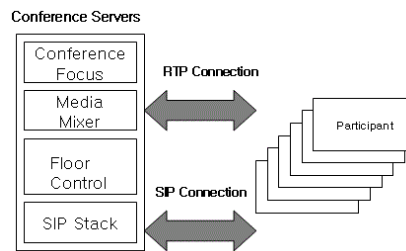


그림 2. 컨퍼런스 서비스 시스템의 기본 구성
Fig. 2. Basic Operation of Conference Service System

그림 2에서 기본적인 컨퍼런스 서비스 시스템은 컨퍼런스 포커스, 미디어 믹서, 플로어 제어[6] 및 SIP 스택으로 구성된다. 컨퍼런스 포커스는 컨퍼런스 서버와 참가자 간의 SIP 컨퍼런스 세션을 설정하고 유지하는 기능을 하며 미디어 믹서는 참가자가 생성한 미디어 패킷 스트림을 각각의 소스를 구분하여 혼합하고 각 참가자들에게 전송한다. 이때 미디어 패킷은

RTP(real-time transport protocol)를 사용하여 전송된다.[7] 플로어 제어는 컨퍼런스 서버와 참가자 사이에서 공유 자원에 대한 접근을 제어하기 위한 기능을 제공한다. 컨퍼런스 정보 데이터는 <conference-info>가 최상위 엘리먼트인 XML 문서 application/conference-info+xml 형태이다.[8][9] <conference-info>는 컨퍼런스 참가자를 포함한 컨퍼런스 서비스 시스템의 모든 정보를 나타내기 위한 여러 하위 엘리먼트를 가진다.

기존 관련 연구로는 인스턴트 메시지 시스템에서 멀티미디어 메시지를 포함한 다량의 인스턴트 메시지 데이터를 다중 서버에 분산 처리 하기 위하여 서버 사이에 미디어 제어 채널을 연결하여 서버의 부하를 제어하는 방식[10] 등이 있으나 본 논문에서 제안한 미디어 제어 채널/분산 컨퍼런스 매니플레이션 구조를 사용하여 다중 서버 구조의 대용량 컨퍼런스 서비스 시스템을 구현한 연구는 아직 발표되고 있지 않다. 본 연구에서는 각 컨퍼런스 서버 사이에 미디어 제어 채널을 설정하고 이 제어 채널을 통해 컨퍼런스 매니플레이션 메시지를 서로 전송하여 시스템 부하를 제어하는 새로운 방안을 제시하였다. 아울러 서버 사이의 제어 메시지 교환 절차도 함께 제시되었다.

3. 시스템 설계 및 구현

3.1 미디어 제어 채널/분산 컨퍼런스 매니플레이션 서버 구조 설계

본 연구에서 설계한 미디어 제어 채널/분산 컨퍼런스 매니플레이션 서버의 구조를 그림 3에 보였다. 여기에서 각각의 컨퍼런스 서버는 컨퍼런스 포커스, 미디어 믹서, 부하 제어 모듈, 컨퍼런스 매니플레이션 모듈, 미디어 제어 채널 모듈 및 SIP 스택으로 구성된다.

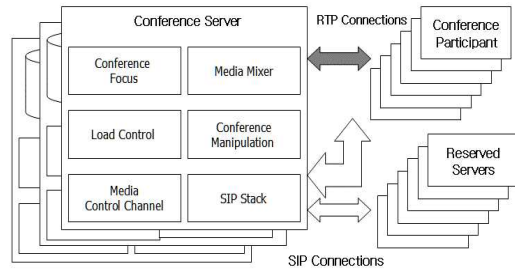


그림 3. 미디어 제어 채널/분산 컨퍼런스 매니플레이션 컨퍼런스 서버 구조

Fig. 3. Server Architecture of Media Control Channel/ Distributed Conference Manipulation

SIP 스택은 컨퍼런스 서버 및 컨퍼런스 참가자와 SIP INVITE, SIP ACK 메시지 및 각종 SIP 응답 메시지를 송수신하며, 미디어 제어 채널 모듈은 각 서버 사이에 트랜스포트 세션을 통한 미디어 제어 채널을 설정하는 기능을 한다, 컨퍼런스 포커스는 SIP 스택을 사용하여 컨퍼런스 참가자와 서버 사이에 SIP 메시지를 교환하여 컨퍼런스 세션을 설정하고 관리한다. 컨퍼런스 매니플레이션 모듈은 컨퍼런스 서버의 부하 제어에 사용되는 메시지들을 생성하고 해독하는 기능을 한다. 부하 제어 모듈은 컨퍼런스 매니플레이션 메시지들을 사용하여 각 컨퍼런스 서버의 부하 레벨을 실시간 모니터링하고 부하를 제어한다. 미디어 믹서는 각 컨퍼런스 참가자들이 발생한 오디오 패킷 및 비디오 패킷 스트림을 혼합하고 분배한다.

동작 중인 컨퍼런스 서버들은 서로 SDP 콘텐츠 타임을 가진 SIP INVITE 메시지와 이에 대한 응답 메시지를 교환하며 여기에는 SDP 포맷형태로 양단간의 통신에 사용할 TCP 포트번호, 속성(attribute) 엔트리 및 미디어 제어 채널 메시지 교환에 사용할 ID 값이 들어간다. 이어서 SIP ACK 메시지가 교환되고 각 컨퍼런스 서버들 사이에 컨퍼런스 매니플레이션 메시지들을 교환 할 수 있는 미디어 제어 채널이 설정된다.

동작 중인 컨퍼런스 서버에게 컨퍼런스 URI를 담은 SIP INVITE 메시지를 컨퍼런스 참가자가 송신하면 컨퍼런스 서버는 본 논문에서 설계된 분산 컨퍼런스 매니플레이션 메시지의 현재 부하 레벨을 나타내는 엘리먼트인 <ctrl-svr-load>의 값을 조사한다. 이 값은 각 컨

퍼런스 서버가 초 당 처리하는 SIP 메시지 양과 초 당 처리하는 오디오/비디오 패킷 수를 합하여 계산된다, 컨퍼런스 서버의 현재 부하 레벨이 미리 설정된 상한값보다 작으면 컨퍼런스 서버는 컨퍼런스 참가자에게 SIP '200 OK' 응답 메시지를 보내고 컨퍼런스 참가자는 SIP ACK 메시지를 보내와 컨퍼런스 서버와 RTP 세션을 설정한다. 컨퍼런스 서버의 현재 부하 레벨 값이 설정된 상한값에 도달하는 경우 해당 서버는 SIP redirection 메시지에 현재 부하 레벨이 가장 낮은 컨퍼런스 서버의 URI를 넣어 상대방에게 알려주며 이를 수신한 컨퍼런스 참가자는 이 서버와 RTP 세션을 맺게 된다.

현재 동작 중인 모든 컨퍼런스 서버의 부하 레벨이 설정된 상한값에 도달하면 미리 예약된 컨퍼런스 서버들 중에서 새로운 서버를 추가하는 동작을 시작한다. 이를 위하여 먼저 시스템에 추가될 새로운 서버에게 콘텐츠 타입이 SDP인 SIP INVITE 메시지를 보내며 이 메시지에는 TCP 포트번호와 컨퍼런스 서버간의 미디어 제어 채널 메시지 교환에 사용할 ID 값이 포함된다. 새로운 서버는 이에 대한 응답으로 SDP 속성 (attribute) 엔트리에 자신이 사용할 미디어 제어 채널 ID 값을 넣어 보내오며 이어서 미디어 제어 채널이 생성된다, 새로운 서버를 추가시킨 컨퍼런스 서버에서는 이 미디어 제어 채널을 통해 현재 컨퍼런스 서비스 시스템의 전체 컨퍼런스 정보를 추가된 서버에게 전송한다. 이때 본 논문에서 설계된 분산 컨퍼런스 매니플레이션 메시지의 컨퍼런스 정보 데이터를 나타내는 < crt-conf-info > 엘리먼트가 사용된다. 다음 컨퍼런스 참가자에게 이 추가된 서버의 URI를 전송하여 이 서버와 컨퍼런스 세션을 설정 하도록 한다.

3.2 분산 컨퍼런스 매니플레이션 제어 메시지 설계

본 연구에서는 다중 서버 구조의 컨퍼런스 서비스 시스템에서 미디어 제어 채널을 통하여 전송되는 분산 컨퍼런스 매니플레이션 메시지를 설계하였다. 설계된 컨퍼런스 매니플레이션 메시지들은 XML 문서 포맷을 가지며 XML 스키마 인스턴스 형태이고 컨퍼런스 서버 사이에 연결된 미디어 제어 채널 세션을 사용해 전송된다. 각 메시지들은 요청(request)/응답(response) 형태로

동작하며 요청 메시지에 포함된 제어 엘리먼트들을 수신측에서 해독해 그에 대한 동작 처리를 하고 이에 대한 결과코드를 응답 메시지에 넣어 상대방에게 전송한다. 먼저 컨퍼런스 서버의 부하 제어를 위하여 현재 서버 부하 레벨을 나타내는 엘리먼트인 < crt-svr-load > 이 설계되었다. 이 엘리먼트는 속성으로 새로운 서버를 시스템에 추가하는 기준인 서버의 최대 부하 허용레벨을 나타내는 'max-load'와 최소 부하 허용레벨인 'min-load'을 가지며, 또 각 컨퍼런스 서버를 구분하기 위한 'svr-id'을 가진다. 최소 부하 허용레벨은 부하가 너무 낮은 서버는 해당 서버가 관리하는 사용자들을 다른 서버로 이동시키고 대기 상태로 두기위한 것이다.

다음 < crt-conf-info > 엘리먼트가 현재 컨퍼런스 서비스 시스템의 정보를 표시하기위하여 설계되었다. 이 엘리먼트의 하위 엘리먼트로 < crt-system >과 < crt-users >가 있으며 < crt-system >은 컨퍼런스 서버에 관한 컨퍼런스 정보를 가진다. 이 엘리먼트의 속성으로 < crt-svr-load >와 마찬가지로 'svr-id'가 각 컨퍼런스 서버를 구분하기위하여 사용되었고 'active-svrs'가 현재 동작 중인 컨퍼런스 서버의 개수를 나타내기 위해 사용되었다. < crt-users >는 하위 엘리먼트로 현재 컨퍼런스 오브젝트 참가자들의 컨퍼런스 정보를 나타내기 위한 < conf-user >를 가지며 속성으로 컨퍼런스 오브젝트 ID인 'conf-obj-id'를 가진다.

다음 < reserved-svrs > 엘리먼트는 동작 중인 모든 서버의 부하가 기준치를 넘어설 경우 새롭게 추가 될 수 있는 예약된 컨퍼런스 서버들의 목록을 가진다. < conf_op_sys > 엘리먼트는 컨퍼런스 오브젝트를 생성, 수정 및 삭제하기 위한 하위 엘리먼트들인 < conf_sys_create >, < conf_sys_mod >, < conf_sys_del >를 각각 가진다. < conf_op_ptcs > 엘리먼트는 컨퍼런스 참가자의 생성, 수정 및 삭제를 위한 하위 엘리먼트들인 < conf_ptcs_create >, < conf_ptcs_mod >, < conf_ptcs_del >를 각각 가진다.

그림 4에 이와 같이 설계된 컨퍼런스 서비스 시스템에서의 제어 메시지 교환 절차를 보였다.

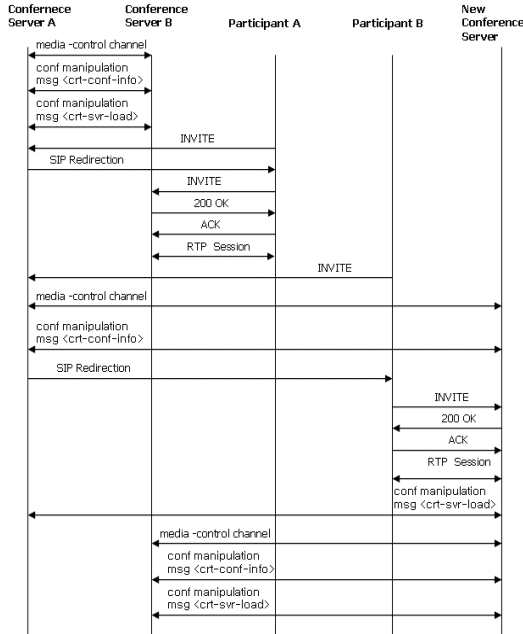


그림 4. 미디어 제어 채널/분산 컨퍼런스 매니플레이션컨퍼런스 서비스 시스템의 제어 메시지 교환 절차

Fig. 4. Exchange Procedure of Control Messages for Media Control Channel/ Distributed Conference Manipulation Conference Service System

여기에서 현재 동작 중인 컨퍼런스 서버 A와 컨퍼런스 서버 B는 서로 미디어 제어 채널을 설정하고 이 채널을 통하여 컨퍼런스 매니플레이션 메시지가 서로 교환된다. 이들 메시지에는 현재 컨퍼런스 서비스 시스템의 정보를 나타내는 < crt-conf-info > 엘리먼트와 현재 서버 부하 레벨을 나타내는 < crt-svr-load > 엘리먼트가 포함된다. 참가자 A가 컨퍼런스 서버 A에게 컨퍼런스 참가를 위하여 SIP INVITE 메시지를 보내면 현재 동작 중인 서버의 부하 레벨을 조사하고 가장 부하 레벨이 낮은 서버의 주소를 담은 SIP redirection 메시지로 응답한다. 참가자 A는 이 서버에게 SIP INVITE 메시지를 보내 컨퍼런스 세션을 설정하게 된다. 참가자 B가 컨퍼런스 참가 요청을 하는 경우는 동작 중인 모든 컨퍼런스 서버의 부하 레벨이 높아 새로운 컨퍼런스 서버를 추가할 때에 해당된다. 컨퍼런스 서버 A는 새로 추가할 서버와 미디어 제어 채널을 설정하고 이 채널을 통하여 현재 컨퍼런스 정보를 나타내는 컨퍼런스 매니

플레이션 메시지를 전송한다. 이어서 새로운 컨퍼런스 서버의 주소를 담은 SIP redirection 메시지를 참가자 B에게 보내어 컨퍼런스에 참가하도록 한다. 새로 추가된 서버는 동작 중인 컨퍼런스 서버 B와도 미디어 제어 채널을 설정하고 이 채널을 통하여 컨퍼런스 매니플레이션 메시지를 서로 교환한다.

4. 성능 분석

본 논문에서는 미디어 제어 채널/분산 컨퍼런스 매니플레이션 구조를 가진 컨퍼런스 서비스 시스템의 성능을 시뮬레이션으로 분석하였다. 컨퍼런스 포커스 및 미디어 믹서를 포함한 시스템 구성 모듈인 부하 제어 모듈, 컨퍼런스 매니플레이션 모듈, 미디어 제어 채널 모듈과 SIP 스택은 Fedora 리눅스 환경에서 자바로 구현되었고 시뮬레이션 파라미터로 샘플링 주파수 8KHz, 초당 미디어 패킷 수 16, 패킷 크기 512바이트, 프레임 길이 10msec를 사용하였다. 여기서는 구현된 시스템 기능 중 가장 중요한 부분인 사용자 수 증가에 따른 컨퍼런스 서버의 부하 레벨 증가에 따라 서버가 실제 동작적으로 추가되어 부하를 분산 할 수 있는지에 관한 부분만 실험하였다. 실험은 본 논문에서 연구한 컨퍼런스 구조와 기존의 중앙 집중형 컨퍼런스 구조에 대해 평균 SIP 메시지 지연시간을 측정하였고 그림 5에 사용자 수를 증가시켜가며 측정한 결과를 보였다. 이때 최대 부하 값을 500으로 하여 사용자 수 50명 단위로 컨퍼런스 서버가 추가 되도록 하였다. 따라서 50명 경우 양쪽에서 지연시간의 차이가 거의 없으나 100명 이상의 경우 본 논문에서 제시한 방식이 기존 방식에 비해 평균 SIP 메시지 지연시간이 크게 개선됨을 보여준다. 측정 결과는 사용자 수 100명인 경우 17.4%, 150명인 경우 28.6%, 250명인 경우 39.5%가 각각 개선되었다. 여기서 평균 SIP 메시지 지연시간은 컨퍼런스 시스템 동작 중에 발생하는 모든 SIP 메시지들을 서버가 처리하는데 소요되는 시간이다. 이때 발생하는 SIP 메시지는 시스템 전체에서 발생하는 SIP INVITE 메시지, SIP 응답 메시지 외 각종 SIP 메시지의 총합이다.

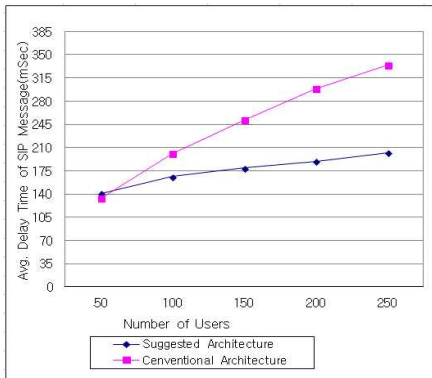


그림 5. 평균 SIP 메시지 지연시간
Fig. 5. Average Delay Time of SIP Messages

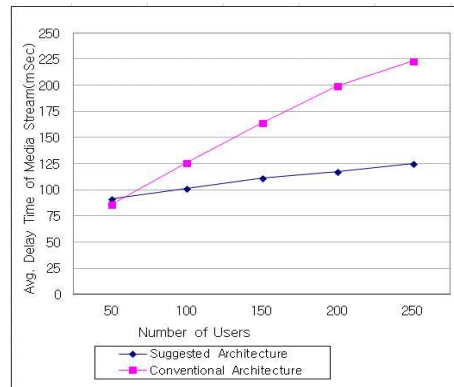


그림 6. 평균미디어스트림 지연시간(최대부하레벨=500)
Fig. 6. Average Delay Time of Media Stream (Max. Load=500)

다음 컨퍼런스 서비스 시스템 내에서 전송되는 미디어 패킷들의 평균 미디어 스트림 지연시간을 측정하였고 그림 6에 이의 결과를 보였다. 여기서도 사용자 수가 작을 때는 단일 서버로 동작하고 사용자 수의 증가에 따라 정해진 최대 부하 값을 초과 시 새로운 서버가 추가된다. 이때 최대 부하 값을 500으로 하여 사용자 수 50명 단위로 서버가 추가되도록 하였다. 측정 결과 사용자 수 50명까지는 양쪽은 동일 한 성능을 보이거나 사용자 수 100명부터는 제안된 방식의 경우 평균 지연시간이 25.8% 감소하고, 150명에서는 32.3%, 250명에서는 43.9% 감소하여 성능이 개선됨을 보여준다.

다음 그림 7에 최대 부하 값을 700으로 하여 평균 미디어 스트림 지연시간을 측정하였고 그림 7에 이의 결과를 보였다. 이 경우 최대 부하 값의 증가로 사용자 수 70명마다 서버가 추가된다. 측정 결과 평균 미디어 스트림 지연시간이 사용자 수 140명인 경우 26.7% 감소하고 210명인 경우 39.3%, 350명인 경우 50.5%로 각각 감소하여 시스템 성능이 크게 개선되는 결과를 나타내었다.

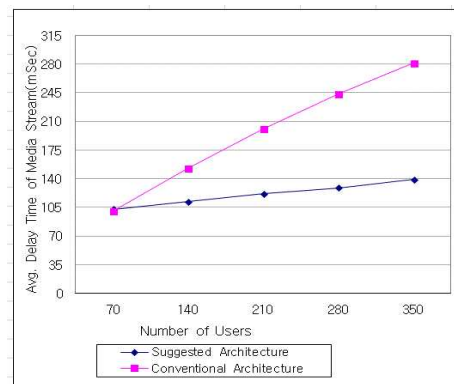


그림 7. 평균미디어스트림 지연시간(최대부하레벨=700)
Fig. 7. Average Delay Time of Media Stream (Max. Load=700)

5. 결론

본 논문에서는 미디어 제어 채널/분산 컨퍼런스 매니플레이션 구조에 의한 새로운 다중 서버 컨퍼런스 서비스 시스템을 연구하였다. 본 연구에서 제안한 방식에서는 각 컨퍼런스 서버 사이에 미디어 제어 채널을 설정하고 이 제어 채널을 통해 시스템 부하를 제어하기 위한 분산 컨퍼런스 매니플레이션 메시지를 교환하게 된다. 이 메시지는 트랜스포트 세션 연결 후 생성되는 미디어 제어 채널을 사용해 컨퍼런스 서버 사이에 전송되며 컨퍼런스 참가자 수의 변화에 따른 컨퍼런스 서버의 부하를 효율적으로 분산 처리 할 수 있다. 제안된 시

스텝의 성능 분석을 위하여 평균 SIP 메시지 지연시간과 평균 미디어 스트림 지연시간을 측정하였고 평균 SIP 메시지 지연시간의 경우 사용자 수 증가에 따라 17.4%에서 39.5%까지 개선 효과가 있음을 보여주었으며 평균 미디어 스트림 지연시간의 경우 25.8%에서 50.5%까지 개선 효과가 있음을 보여주었다. 향후 과제로는 사용자수 증가 이외에 실제 시스템 운용 상황에서 발생 할 수 있는 다양한 상황에서의 성능 측정 실험도 필요하며, 제어 메시지 교환 절차의 최적화 및 멀티미디어 데이터를 포함하는 대용량 인스턴트 메시지 시스템과의 통합 방식을 연구 해볼 필요가 있다.

REFERENCES

[1] J. Rosenberg. "A Framework for Conferencing with the Session Initiation Protocol". IETF RFC 4353, April, 2011.

[2] M. Barnes et al, "A Framework for Centralized Conferencing", IETF RFC 5239, October 2012.

[3] R. C. Boulton, T. Melanchuk and S. McGlashan, "Media Control Channel Framework", IETF RFC 6230, May 2011

[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[5] F. Andreassen, "Session Description Protocol (SDP) Capability Negotiation", IETF RFC 5939, September 2010.

[6] G. Camarillo, "Connection Establishment in the Binary Floor Control Protocol (BFCP)". IETF RFC 5018. December 2015.

[7] C. Perkins et al, "Multiplexing RTP Data and Control Packets on a Single Port". IETF RFC 5761, October 2015.

[9] O. Novo, G. Camarillo and D. Morgan, "Conference Information Data Model for Centralized Conferencing" IETF RFC 6501, March 2012.

[9] G. Camarillo, S. Srinivasan and J. Urpalainen, "Conference Event Package Data Format Extension for Centralized Conferencing" IETF RFC 6502, March 2012.

[10] B. Kim and C. Jang, "A Distributed Instant Message System Architecture using Media

Control Channel", Journal of Korea Inst. of Information and Communication Eng., vol. 20, no.5, pp.979-985, May 2016.

저자약력

장 춘 서 (Choonseo Jang)

[정회원]



- 한국과학기술원 공학박사
- 현재 : 금오공과대학교 컴퓨터공학과 교수

〈관심분야〉 SIP, 인터넷텔레포니