

컨퍼런스 포커스를 위한 효율적인 SIP 메시지 처리 방법

조현규*, 이기수**, 장춘서**

A Method of Efficient SIP Messages Processing for Conference Focus

Hyun-Gyu Jo *, Ky-Soo Lee **, Choon-Seo Jang **

요 약

컨퍼런스 포커스는 전체 컨퍼런스 참여자들과의 SIP(Session Initiation Protocol) 신호 연결을 유지하고 관리하며 컨퍼런스 서버의 핵심 기능을 한다. 컨퍼런스 포커스는 컨퍼런스 참여자 수가 늘어남에 따라 처리해야 하는 메시지 양의 증가로 인해 컨퍼런스 확장성에 제약을 주게 된다. 본 논문에서는 이와 같은 문제점을 해결하기 위하여 컨퍼런스 포커스로부터 전체 참여자들의 컨퍼런스 정보를 수신 후 이를 자신이 관리하는 참여자들에게 전송하는 기능을 갖는 사용자 시스템(컨퍼런스 정보 에이전트)을 제안하고 컨퍼런스 포커스를 포함한 전체 시스템을 설계 및 구현하였다. 컨퍼런스 정보 에이전트는 컨퍼런스 포커스에 자신을 등록 시 SIP SUBSCRIBE 메시지에 자신이 에이전트 기능을 가지고 있음과 처리 가능 용량을 알린다. 컨퍼런스 포커스는 SIP 응답 메시지를 통하여 컨퍼런스 정보 에이전트로 선택되었음을 상대방에게 알리고 이후 발생하는 모든 컨퍼런스 통지 메시지를 이 에이전트에게 보냄으로써 컨퍼런스 포커스의 부하를 줄일 수 있다. 제안한 시스템의 성능은 실험을 통하여 분석하였다.

Abstract

Conference focus maintains and manages SIP(Session Initiation Protocol) signal connections with total conference participants, and it plays core functions of conference server. When the number of conference participants increases, conference focus restricts scalability of conference as increasing quantity of messages for processing. In this paper, to solve this problem, we propose a user system(conference information agent), which receives conference information from the conference focus and transfers them to the participants it manages. And we design and implement the total system including conference focus. When the conference information agent subscribe to the conference focus, it shows its possession of agent capability and capacity by using SIP SUBSCRIBE message. The conference focus informs to the other party by SIP response message that it was chosen as a conference information agent, and after that the conference focus sends all conference notification messages to it. So load of the conference focus can be reduced by this method. Finally, we have analyzed the performance of our system by experiments.

• 제1저자 : 조현규

• 접수일 : 2007.9.13, 심사일 : 2007.10.29, 심사완료일 : 2007.11.10.

* 금오공과대학교 컴퓨터공학과 계약교수 ** 금오공과대학교 컴퓨터공학과 교수

※ 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

▶ Keyword : SIP(Session Initiation Protocol), 컨퍼런스(Conference), 컨퍼런스이벤트패키지(Conference Event Package)

I. 서론

컨퍼런스 시스템은 참여자들을 서로 연결하여 음성 및 영상 데이터가 교환될 수 있도록 하며, SIP 기반의 컨퍼런스 시스템은 그 구성에 따라 중앙 집중형 모델, Ad-hoc 모델, 풀 메시 모델 등으로 나눌 수 있다[1][2]. 이중 중앙 집중형 모델은 컨퍼런스 서버가 컨퍼런스의 개설과 관련된 기능만 수행하고 미디어는 사용자간에 직접적인 미디어 데이터를 송수신하는 풀 메시 방식의 모델과 비교하였을 때, 참여자 수의 증가에 따른 서버의 부하는 증가하지만 참여자들의 관리가 체계적이며 용이하고, 참여자의 사용자 시스템을 단순화시킬 수 있는 장점을 가진다[3][4].

컨퍼런스 포커스는 컨퍼런스 시스템의 주체로써 전체 컨퍼런스 참여자들과의 SIP 신호 연결을 주도하는 SIP UA(User Agent)의 일종이며 컨퍼런스 서버의 핵심 기능을 한다[3]. 컨퍼런스 포커스는 동적으로 변화하는 컨퍼런스 정보 및 전체 참여자의 정보를 제공하여야 하는데 참여자 수가 늘어나게 되면 각각에게 제공해야 하는 정보량이 증가해 컨퍼런스 서버에 작용되는 부하도 아울러 커지게 된다. 본 논문에서는 이와 같은 문제점을 해결하기 위하여 컨퍼런스 포커스로부터 전체 참여자들의 컨퍼런스 정보를 수신 후 이를 자신이 관리하는 참여자들에게 전송하는 기능을 갖는 사용자 시스템(컨퍼런스 정보 에이전트)을 제안하였다.

이와 같은 기능을 가진 SIP UA를 사용 시 컨퍼런스 포커스는 전체 참여자에게 모두 컨퍼런스 정보를 전송 할 필요가 없게 되어 참여자 수가 증가할수록 부하 감소 효과는 커지게 되고 따라서 컨퍼런스 확장성을 높일 수 있게 된다. 이 컨퍼런스 정보 에이전트는 컨퍼런스 포커스에 자신을 등록 시 SIP 메시지의 From: 헤더에 자신이 에이전트 기능을 가지고 있음과 자신의 처리 가능 용량을 알리고 이후 컨퍼런스 포커스로부터 자신이 담당 할 참여자 목록을 전달 받아 컨퍼런스 정보의 전달을 담당하게 된다. 컨퍼런스 포커스는 데이터베이스에 저장된 에이전트 목록과 용량을 참고하여 사용자가 증가하게 되면 적절한 에이전트를 선택해 동작시키게 할 수 있다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 컨퍼런스 포커스의 동작 및 컨퍼런스 이벤트 패키지에 대해 설명하고 III장에서는 본 논문에서 제안하는 시스템을 설계 및 구현하고, 시스템의 성능 분석을 한 후 IV장에서 결론을 맺는다.

II. 관련연구

2.1 컨퍼런스 포커스

컨퍼런스 포커스는 컨퍼런스 참여자들과 서버 사이에 컨퍼런스 세션에 대한 설정, 변경 및 종료를 관리하고 컨퍼런스에서 발생하는 각종 정보들의 전달을 제어한다[5]. 컨퍼런스 참여자는 SIP 기본 메소드 이외의 확장 메소드인 REFER[6], SUBSCRIBE, NOTIFY 등을 이해하고 처리할 수 있으며, isfocus 파라미터를 이해할 수 있는 UA를 가지는 컨퍼런스 어웨어(Aware) UA와 이와 같은 기능을 갖지 못하는 컨퍼런스 언어웨어(Unaware) UA로 나누어진다. 컨퍼런스 포커스는 컨퍼런스 어웨어 UA의 기능 및 컨퍼런스 패키지 기능과 SIP 프락시(proxy) 서버 기능을 모두 포함하며, 컨퍼런스 다이얼로그에 대한 전반적인 시그널링을 담당한다. 그림 1은 이러한 컨퍼런스 포커스의 구성이다.

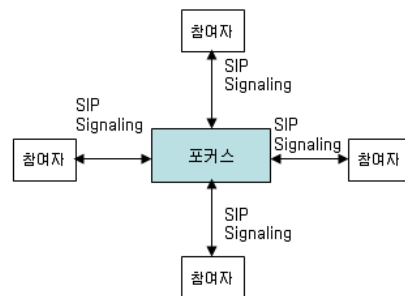


그림 1. 컨퍼런스 포커스
Fig 1. Conference Focus

컨퍼런스 URI는 sip:conf-id@sp.kumoh.ac.kr와 같이 SIP 주소와 동일한 형식으로 이루어진 컨퍼런스 식별자이며, 컨퍼런스를 식별할 수 있는 목적지 주소를 생성한다[3]. 참여자는 INVITE 메시지에 컨퍼런스 URI를 넣어 보냄으로써 컨퍼런스 포커스와 컨퍼런스 세션을 맺는데, 이러한 연결 형태를 다이얼인(dial-in) 방식이라 한다. 그림 2는 다이얼인 방식을 통한 연결 흐름도이다.

컨퍼런스 포커스는 INVITE 요청 메시지에 대한 200 OK 응답 메시지를 보낼 때 Contact: 헤더에 'isfocus'를 넣어 자신이 포커스임을 참여자에게 알린다.

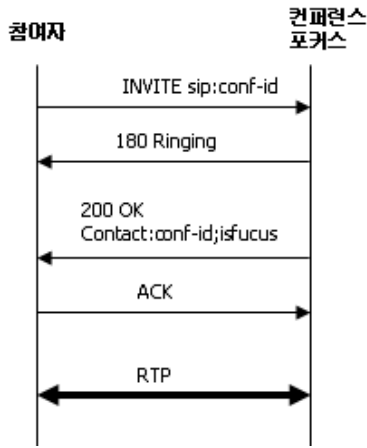


그림 2. 다이얼인 컨퍼런스 세션 연결
Fig 2. Conference Session Connection by Dial-in

2.2 컨퍼런스 이벤트 패키지

컨퍼런스 이벤트 패키지는 사용자들이 컨퍼런스 관련 정보를 얻기 위해서 컨퍼런스 포커스를 가리키는 SIP UA에게 등록 메시지를 보낼 수 있도록 한다(7). 이와 같은 동작으로 사용자들은 변화하는 컨퍼런스 정보를 포커스로부터 통지 메시지를 통해 받을 수 있게 된다(8)(9).

등록에는 SIP SUBSCRIBE 메시지가 사용되고 통지에는 NOTIFY 메시지가 사용된다. 등록 시 SUBSCRIBE 메시지 내의 Event: 헤더에는 컨퍼런스 이벤트 패키지를 나타내는 이름인 'conference'가 들어가며 Accept: 헤더에는 컨퍼런스 정보 포맷인 application/conference-info+xml이 들어간다. 이 포맷은 XML(Extensible Markup Language) 문서 형태로 포커스가 보내오는 NOTIFY 메시지의 몸체 부분에 담겨 사용자에게 통지된다. 그림 3은 SUBSCRIBE 메시지의 한 예이다.

```

SUBSCRIBE sip:6529831@sip.kumoh.ac.kr SIP/2.0
Via: SIP/2.0/UDP 202.31.201.82;branch=z9hG4bK3721182
Max-Forwards: 70
To: sip:6529831@sip2.kumoh.ac.kr;tag=432871
From: sip:johg@sip2.kumoh.ac.kr;tag=983287
Call-Id: 1742983@sip2.kumoh.ac.kr
CSeq: 6 SUBSCRIBE
Contact: johg@202.31.201.82
Event: conference
Accept: application/conference-info+xml
Content-Length: 0
    
```

그림 3. SUBSCRIBE 메시지 예
Fig 3. Example of SUBSCRIBE Message

컨퍼런스 정보 문서 포맷인 application/conference-

info+xml의 구조는 최상위 요소로 conference-info를 가지며, 속성으로는 컨퍼런스 URI 값을 가지는 entity, 부분 컨퍼런스 정보 여부를 나타내는 state, 통지 문서의 수신된 순서를 정하기 위해 송신 시 매 문서마다 값이 계속 증가하는 version이 포함되어야 한다. 다른 주요 요소로는 users가 있고 이 요소의 하위 요소로 컨퍼런스의 각 참여자에 대한 정보를 나타내는 복수개의 user 요소가 있다. user 요소의 하위 요소로 각 디바이스의 상태 정보를 보여 줄 수 있는 복수개의 endpoint 요소가 온다.

각 endpoint 요소는 사용자의 각 디바이스의 연결 상태를 보여주기 위한 status, 컨퍼런스에 가입한 방식을 나타내는 joining-method, 미디어 스트림 정보를 나타내는 media 등의 하위 요소를 다시 가진다. 요소 media는 미디어 종류를 나타내는 type, 미디어 스트림을 구별하기 위한 label, 동기화 소스를 식별하기 위한 src-id를 다시 하위 요소로 가진다.

또, 전체 컨퍼런스 상태 정보를 나타내기 위해 conference-state 요소가 있고 이 요소의 하위 요소로 현 시점에서의 참여자 수를 나타내는 user-count와 컨퍼런스가 현재 동작 중인지 여부를 나타내는 active가 있다. 그리고 컨퍼런스 전반적인 내용을 텍스트로 표시하는 conference-description 요소가 있으며 하위 요소로 display-text와 컨퍼런스 주제를 나타내는 subject가 있다.

III. 시스템 설계 및 구현

3.1 제안된 컨퍼런스 포커스 및 컨퍼런스 정보 에이전트

본 논문에서 구현된 컨퍼런스 포커스의 구조는 그림 4와 같다. 전체적인 구성은 기본 SIP 메시지를 처리하는 부분, 컨퍼런스 이벤트 패키지의 등록 요청을 담당하는 부분, 포커스로부터 수신한 컨퍼런스 정보를 참여자들에게 대신 전송해 주는 컨퍼런스 정보 에이전트 관련 제어 부분 및 컨퍼런스 정보에 변화가 생겼을 때 이를 참여자에게 통지해주는 부분으로 나누어진다.

컨퍼런스 이벤트 패키지의 등록 부분은 컨퍼런스 참여자로부터 받은 SUBSCRIBE 메시지의 From: 헤더에 추가된 conf-pkg-agent 항목이 있는지 조사하여 만약 있으면 이 참여자를 컨퍼런스 정보 에이전트 후보로 데이터베이스에 등록한다. 이때 capacity 항목이 있으면 이 값도 함께 저장한다. 그림 5는 에이전트 기능을 가진 참여자가 보낸 SUBSCRIBE

메시지의 예이다. 여기서 자신이 처리 가능한 용량을 나타내는 capacity 항목 값이 20임을 알 수 있다.

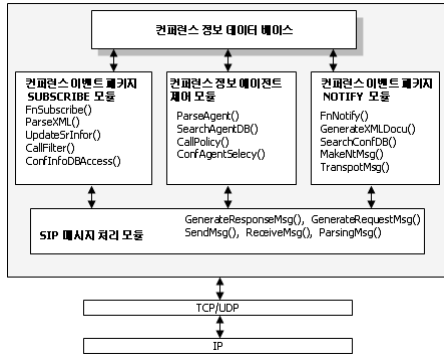


그림 4. 컨퍼런스 포커스의 구조
Fig 4. Architecture of Conference Focus

```

SUBSCRIBE sip:6529831@sip.kumoh.ac.kr SIP/2.0
Via: SIP/2.0/UDP 202.31.201.84;branch=z9hG4bK1874326
Max-Forwards: 70
To: sip:6529831@sip.kumoh.ac.kr;tag=8327612
From: sip:johg2@sip2.kumoh.ac.kr;tag=3786541; conf-pkg-agent:capacity=20
Call-Id: 4568161@sip2.kumoh.ac.kr
CSeq: 9 SUBSCRIBE
Contact: johg@202.31.201.84
Event: conference
Accept: application/conference-info+xml
Content-Length: 0
    
```

그림 5. 에이전트에서의 SUBSCRIBE 메시지
Fig 5. SUBSCRIBE Message from Agent

포커스는 현재 동작 중인 컨퍼런스 정보 에이전트가 있는지 조사하여 만약 없으면 이 참여자에 대한 200 OK 응답 메시지의 From: 헤더에 conf-pkg-agent 항목을 추가하여 컨퍼런스 정보 에이전트로 선택되었음을 알리고 데이터베이스의 해당 참여자 정보 테이블에 에이전트 활성화 표시를 한다. 이 메시지를 받은 참여자는 이후 컨퍼런스 정보 에이전트로 동작하게 되어 포커스로부터 받은 application/conference-info+xml 포맷의 통지 메시지에서 users 요소 아래의 각 user 하위요소들에 대해 포커스를 대신해 통지 메시지를 전달한다. 포커스는 컨퍼런스 정보 에이전트를 선정하였으므로 이후 발생하는 모든 통지 메시지는 전체 참여자들에게 보내는 대신 이 에이전트에게만 보내면 된다. 이와 같은 방식을 통해 포커스의 부하를 줄일 수 있게 된다.

만일 현재 동작 중인 에이전트가 존재하면 포커스는 추가된 참여자를 포함한 전체 참여자 수가 현 에이전트의 처리 가능한 최대 참여자 수를 넘는지 조사한다. 이때 최대 수를 넘지 않으면 이 참여자는 일반 참여자와 동일하게 취급된다. 만

일 최대 수를 넘는 경우, 이 참여자를 포함하여 에이전트 기능이 있으나 현재 에이전트로 동작 중이 아닌 참여자 목록을 데이터베이스에서 검색하여 그 중 용량이 가장 큰 참여자를 선택하고 해당 참여자에게 To: 헤더에 conf-pkg-agent 항목을 추가한 NOTIFY 메시지를 보낸다. 이후에는 이 참여자도 에이전트로 동작하게 된다. 그림 6은 이러한 경우 생성되는 메시지의 예이다. 여기서 참여자로 보내지는 메시지의 To: 헤더에 conf-pkg-agent 항목이 있음을 알 수 있다. 그림 7은 구현된 시스템에서 컨퍼런스의 참여자에 대한 정보를 나타내는 user 요소를 보여 주는 실제 동작 화면이다.

```

NOTIFY sip:csj@sip2.kumoh.ac.kr SIP/2.0
Via: SIP/2.0/UDP 202.31.201.87;branch=z9hG4bK653021
Max-Forwards: 70
To: sip:csj@sip2.kumoh.ac.kr;conf-pkg-agent:tag=176212
From: sip:6529831@sip.kumoh.ac.kr;tag=219524; conf-pkg-agent
Call-Id: 328716@202.31.201.87
CSeq: 31 NOTIFY
Contact: 6529831@sip.kumoh.ac.kr
Event: conference
Content-Type: application/conference-info+xml
Content-Length: 937

<conference-info version="0" state="full"
  entity="sip:6529831@sip.kumoh.ac.kr">
  <users>
    <user entity="sip:csj@sip2.kumoh.ac.kr" state="full">
      <display-text>kumoh </display-text>
      <endpoint entity="csj@sip2.kumoh.ac.kr">
        <status>connected </status>
        <joining-method>dialled-in </joining-method>
        <media id="1">
          <display-text>Main Audio </display-text>
          <type>audio </type>
          <src-id>328721 </src-id>
          <status>sendrecv </status>
        </media>
        <media id="2">
          <type>video </type>
          <src-id>137622 </src-id>
          <status>sendrecv </status>
        </media>
      </endpoint>
    </user>
  </users>
</conference-info>
    
```

그림 6. Agent 설정을 위한 NOTIFY 메시지
Fig 6. NOTIFY Message for Agent Selection

```

C:\Program Files\XInox Software\JCreator Pro\WGE2001.exe
NOTIFY Message output --- user element
<user entity="sip:jhk@ip2.kumoh.ac.kr" state="full">
<endpoint entity="jhk@ip2.kumoh.ac.kr">
<status>connected </status>
<joining-method>dialled-in </joining-method>
<media id="1">
<type>audio </type>
<src-id>10421 </src-id>
<status>sendrecv </status>
</media>
<media id="2">
<type>video </type>
<src-id>21918 </src-id>
<status>sendrecv </status>
</media>
</endpoint>
</user>
    
```

그림 7. 컨퍼런스 참여자 정보를 보여 주는 실제 동작 화면
Fig 7. Operation Messages showing Information of Conference Participant

포커스는 이후에 추가로 컨퍼런스에 참여하는 참여자의 컨퍼런스 정보를 이 새로운 에이전트에 보내게 되며 따라서 시스템 전체적으로 복수의 에이전트가 일을 분담하게 되어 시스템의 확장성이 향상된다. 에이전트 기능을 하는 참여자가 컨퍼런스 탈퇴를 하면 포커스는 컨퍼런스 정보 데이터베이스에서 에이전트 기능을 가졌으나 현재 활동을 하지 않는 참여자를 검색하여 그 중 가장 용량이 큰 참여자를 에이전트로 선정하고 탈퇴한 에이전트가 담당하던 내용을 선정된 새로운 에이전트에게 보낸다. 그림 8은 컨퍼런스 내에서 두 개의 에이전트가 동작하는 구성이다.

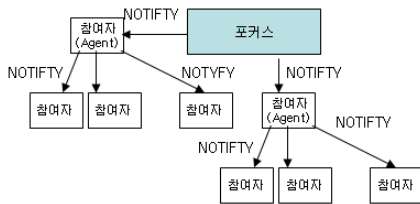


그림 8. 컨퍼런스 정보 에이전트의 동작
Fig 8. Operation of Conference Information Agent

컨퍼런스 정보 에이전트 기능을 가진 참여자 시스템은 포커스에 최초 등록 시 SUBSCRIBE 메시지의 From: 헤더에 conf-pkg-agent 항목을 넣고 추가로 사용자 인터페이스로부터 자신이 처리 가능한 최대 숫자를 입력 받은 값이 있으면 이 값을 capacity 항목에 함께 넣는다.

포커스에서는 동작할 에이전트를 선택 시 capacity 항목이 없는 경우 자체 컨퍼런스 정책으로 일정 값을 할당해 처리하게 된다. 그림 9는 컨퍼런스 포커스와 컨퍼런스 정보 에이전트 기능을 가진 참여자 및 나머지 참여자들 사이의 SIP 메시지 흐름이다.

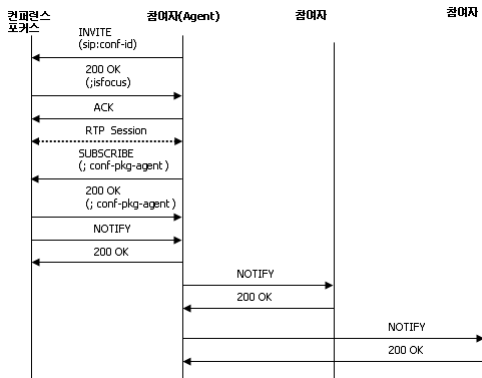


그림 9. SIP 메시지 흐름
Fig 9. Flow of SIP Messages

여기에서는 컨퍼런스 정보 에이전트 기능을 가진 참여자가 INVITE 메시지를 이용한 다이얼인 방식으로 컨퍼런스에 참여한 후 포커스로부터 에이전트로 선택되어 포커스의 통지 기능을 분담함을 보여주고 있다.

3.2 성능 분석

성능분석에서는 컨퍼런스 정보 에이전트의 사용으로 인하여 감소되는 SIP 메시지의 양이 컨퍼런스 포커스에서의 처리 시간에 미치는 영향을 측정하였다. 컨퍼런스 포커스는 리눅스를 탑재한 PC상에 구현하였다. PC의 사양은 펜티엄 IV, CPU 2.4GHz, 메인메모리 512MB이며 리눅스 커널은 2.4 버전을 사용하였다. 각 처리 모듈은 자바로 작성하였다. 컨퍼런스 참여자 시스템은 펜티엄 IV CPU 2.4GHz, 메인메모리 512MB이며, 운영체제로는 마이크로소프트 윈도우 XP를 사용하였고 각 처리 모듈은 역시 자바로 작성하였다. 컨퍼런스 포커스와 컨퍼런스 참여자 시스템은 모두 100Mbps LAN에 연결되었다.

실험 방법은 컨퍼런스 참여자 수를 증가시키면서 컨퍼런스 포커스가 전체 참여자에게 컨퍼런스 정보 통지를 완료하는데 소요되는 시간을 컨퍼런스 정보 에이전트를 사용할 때와 사용하지 않을 때를 비교해 측정하였다. 측정 결과는 그림 10과 같다. 여기서는 참여자수를 증가 시키는데 실험 환경의 제약이 있어 컨퍼런스 정보 에이전트의 개수는 하나를 사용하여 측정하였다.

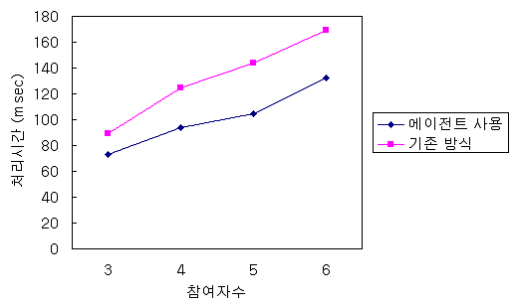


그림 10. 구현된 모델에서의 메시지 처리시간
Fig 10. Message Processing Time of Implemented Model

그림 10에서 참여자 수가 3명일 때 컨퍼런스 정보 에이전트를 사용하는 경우 18%의 처리 시간 감소를 얻을 수 있으며, 참여자 수가 증가 할수록 22% ~ 27%의 시간 감소를 얻을 수 있었다. 여기서 참여자 수가 많아질수록 포커스에서의 처리 시간 감소 효과가 커짐을 알 수 있다. 그러나 이와 같은 효과를 얻기 위해서는 컨퍼런스 정보 에이전트로 동작할 참여

자 시스템에 기능의 추가가 필요하며 제안한 방식은 기본적으로 전체 참여자의 미디어 스트림 처리는 하나의 서버에서 이루어지므로 대규모 컨퍼런스로의 확장성에는 제약이 받게 된다. 따라서 대규모 컨퍼런스의 경우에는 복수의 컨퍼런스 서버에 부하를 분산시키면서 본 논문에서 제안한 방식을 병행해 사용할 수 있을 것으로 보이며 이 부분에 대한 추가 연구가 필요하다.

VI. 결론 및 향후 과제

본 논문에서는 컨퍼런스 포커스로부터 전체 참여자의 컨퍼런스 정보를 수신 후 이를 자신이 관리하는 다른 참여자들에게 전송하는 SIP UA를 사용함으로써 컨퍼런스 포커스의 부하를 분담하여 컨퍼런스 확장성을 높일 수 있는 방안을 제안하고 이를 위한 전체 시스템을 설계 및 구현하였다. 참여자중 하나인 컨퍼런스 정보 에이전트는 컨퍼런스 포커스에 자신을 등록 시 SIP SUBSCRIBE 메시지에 conf-pkg-agent와 capacity 항목을 추가하여 자신이 에이전트 기능을 가지고 있음과 자신의 처리 가능 용량을 알린다. 컨퍼런스 포커스는 컨퍼런스 환경과 내부 정책 및 데이터베이스 내용을 참조하여 해당 사용자 시스템의 컨퍼런스 정보 에이전트 선정 여부를 정하고, 선정이 되면 SIP 응답 메시지를 통해 알려준다. 이후 컨퍼런스 포커스는 발생하는 모든 통지 메시지를 전체 참여자들에게 보내는 대신 이 에이전트에게만 보냄으로써 부하를 줄일 수 있고 따라서 컨퍼런스 확장성을 높일 수 있게 된다. 제안한 시스템의 성능은 실험을 통하여 확인하였다. 향후 과제로는 컨퍼런스 참여자 수를 확대한 실험을 통해 제안된 방식이 실제 컨퍼런스 확장성에 미치는 효과를 측정할 필요가 있다.

참고문헌

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "Session Initiation Protocol," RFC 3261, June 2002.

[2] I. Miladinovic, J. Stadler, "Multiparty Conference Signaling using the Session Initiation Protocol(SIP)," in International Network Conference, 2002.

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

4353, Feb 2006.

[4] K. Singh, G. Nair, and H. Schulzrinne. "Centralized conferencing using SIP," in Internet Telephony Workshop 2001, New York, April 2001.

[5] A. Johnston, O. Levin, "SIP Call Control - Conferencing for User Agents," RFC 4579, August 2006.

[6] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method," RFC 3515, April 2003.

[7] J. Rosenberg, H. Schulzrinne, O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State," draft-ietf-sipping-conference-package-09, february 2005.

[8] A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, June 2002.

[9] P. Koskelainen, H. Khartabil, "Requirements for Conference Policy Control Protocol," draft-ietf-xcon-cpcp-reqs-04, August 2004.

저 자 소개



조 현 규

2005년 8월 : 금오공과대학교 컴퓨터공학과 공학박사
 2006년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 계약교수
 <관심분야> : SIP, VoIP, 실시간 인터넷 통신



이기수

1982년 2월 : 서울대학교 대학원 공학석사
 1982년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수
 <관심분야> : 디지털시스템, 데이터베이스



장준서

1993년 2월 : 한국과학기술원 공학박사
 1981년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수
 <관심분야> : SIP, 임베디드 시스템, 인터넷텔레포니