SIP를 이용한 Conference Server 구조에 관한 설계

안세영° 우시남 노원종 안순신
고려대학교 전자컴퓨터 공학과
{asy1604°, niceguy, nwj, sunshin}@dsys.korea.ac.kr

Design of Conferecne Server Framework Using SIP

Seayoung Ahn° Sinam Woo Wonjong Noh Sunshin An
Dept. of Electronic Engineering, Korea University

요 약

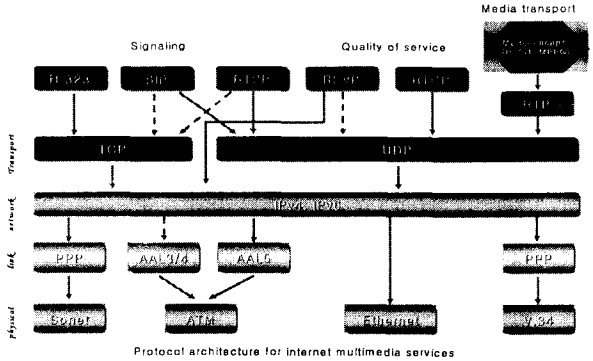
인터넷의 사용자가 증가함에 따라 다양한 서비스가 요구된다. 다자간 회의 시스템 역시 다양한 서비스 제공의 일부분으로 많은 연구가 진행되고 있다. VoIP 프로토콜인 SIP(Session Initiation Protocol)는 기본적으로 다자간 멀티미디어 컨퍼런싱을 지원한다. 그러나 현재는 인터넷 전화등 주로 1:1 호설정에 이용되고 있다. 이를 위해 SIP에 다자간 회의 시스템을 적용하기위하여 새로운 메소드 및 헤더의 확장, 나아가 다자간 회의 시스템 모델 정의등이 요구되어진다. 본 논문은 다자간 회의 시스템 모델 표준을 소개하고 이를 기초로 효율적인 Conference Server를 제시한다. 설계된 Conference Server는 유효한 리소스의 관리 및 처리를 위해 접속 풀링(Connection Pooling)개념을 적용하여 컨퍼런싱에 참여하는 사용자의 요구를 신속하게 처리할수 있도록 하였으며 QoS selector를 두어 사용자의 자원에 맞는 서비스를 보장할수 있게 하여 효율성을 높였다.

1. 서 론

최근 인터넷은 저렴한 사용료와 풍부한 공개 자료를 바탕으로 그 규모가 더욱 확대되어 가고 있다. 규모가 커진 인터넷을 이용하여 기존의 통신망을 대체하는 연구가 활발히 이루어지고 있으며 그 중 화상 전화 서비스와 같은 VoIP는 인터넷의 대표적 서비스가 될 전망이다. VoIP 서비스는 IETF와 ITU가 중심이 되어 이루어지고 있으며 IETF에서 제안한 SIP와 ITU에서 제안한 H.323이 표준화 및 연구되고 있다. 본 논문은 VoIP 서비스를 위한 여러 가지 프로토콜중 IETF가 주도하고 있는 SIP스팩에 따른 SIP 컨퍼런스 서버 모델을 소개하고 효율적인 Conference Server를 설계 한다. 리소스의 관리 및 처리를 위해 풀링개념을 적용하여 컨퍼런싱에 참여하는 사용자의 요구를 신속하게 처리할 수 있도록 하였으며 QoS selector를 두어 사용자의 자원에 맞는 서비스 즉 차별화된 서비스를 보장할수 있게 하여 효율성을 높였다.

2. 관련 연구

SIP 프로토콜은 간단한 텍스트기반의 응용계층 시그널



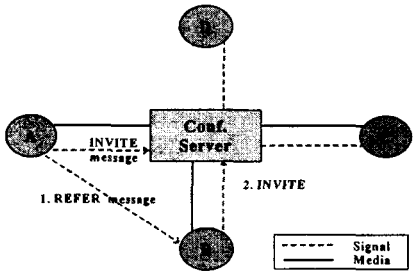
<그림 1> SIP 프로토콜 스택

프로토콜로서 하나 이상의 참가자들과 함께 세션을 생성, 수정, 종료한다. 그리고 SIP는 클라이언트/서버 구조로 멀티미디어 회의 같은 미디어 서비스 및 사용자의 이동성을 지원하며 텍스트 기반이기 때문에 확장성이 용이하다. 그림 1은 SIP 프로 토콜 스택을 보여주고 있다. SIP의 구성요소는 다음과 같다. User Agent(UA)는 UA Client(UAC)와 UA Server(UAS)가 있으면 UA는 호를 요청하는 쪽이며 UAS는 호를 받는 쪽을 의미한다. Network Server는 SIP 네트워크 망을 제어하는 서버로 전체적인 제어를 포괄적으로 관리하는 Proxy Server와 사용자의 이동성을 보장하는 Redirect Server로 되어있다. Registrar는 사용자의 등록을 관리하고 Location Server는 사용자의 위치 정보를 유지한다[1]. 그리고 SIP는 세션 설정을 위해서 INVITE, ACK, BYE, REFER등 다양한 메소드를 가지고 있다.

SIP 프로토콜은 다자간 컨퍼런스 모델로써 6가지를 제안하고 있다[2]. End System Mixing, Large-Scale Multicast Conferences, Dial-In Conference Servers, Ad-Hoc Centralized Conferences, Dial-Out Conferences 및 Centralized Signaling, Distributed Media 모델이다 .

End System Mixing 모델은 사용자가 컨퍼런스에 관련된 시그널 및 미디어를 직접 전송 하는 모델이다. 다시 말해서 사용자가 미디어를 혼합할 수 있는 기능을 가지고 있어야 한다[3]. Large-Scale Multicast Conference 모델은 멀티캐스트가 지원되는 망에서 사용되는 형태로 시그널 및 미디어를 멀티캐스트를 통해서 보내진다. 그러므로 한 사용자는 컨퍼런스에 참여한 사용자의 모든 미디어를 개별적으로 받고 처리하게 된다. 그리고 멀티캐스트 그룹을 형성하기 위한 과정이 요구되어 지며 큰 규모의 컨퍼런스가 가능하다. Dial-In Conference Servers 모델은 서버를 통해서 시그널과 미디어 혼합을 처리하는 방식이다. 서버와 사용자간에 peer to peer 관계를 유지한다. 회의에 참가하는 사용자의 선택은 사용자가 처리를 한다. Ad-Hoc Centralized Conferences 모델은 Dial-In Conference

Servers 모델과 유사하며 두 사용자가 세션을 열고 있는 상태에서 회의 세션으로 변경시 사용되는 모델이다. Dial-Out Conferences 모델은 Dial-In Conference Servers 모델과 유사하며 회의 멤버를 서버가 관리한다. 다시 말해서 서버가 그 멤버들에게 INVITE메시지를 보낸다. Centralized Signaling, Distributed Media모델은 Centralized Controller는 dial-in과 dial-out경우를 의미한다. 서버의 오버헤드를 줄이기 위해서 Centralized Server는 단지 시그널링만을 다루고, 미디어는 사용자들이 직접 전송하게 된다[4]. 컨퍼런스 서버가 새로운 회의 참여자가 있을 때 각각의 참여자들에게 re-INVITEs메시지를 이용하여 알린다.

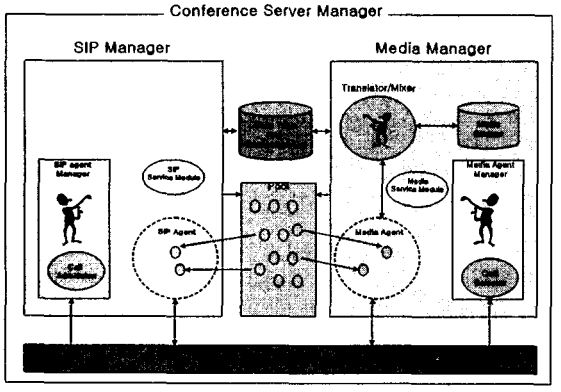


<그림 2> Dial-In Conference Model

그림2은 Dial-In Conference Model을 나타내고 있고 본 논문에서는 이 모델을 기본으로 한 Conference Servers 모델을 제시한다.[5]

3. 컨퍼런스 서버의 구조 모델

다자간 컨퍼런스를 위해서는 사용자 및 자원의 효율적인 관리가 요구되어 진다. 이를 위해서 본 논문에서는 컨퍼런스 구조에 대한 모델을 제시한다. 그림 3는 컨퍼런스 서버의 개념적 구조를 보여주고 있다.

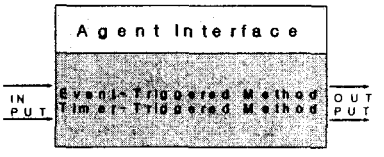


<그림 3> Conference Server 시스템 구조

컨버런스 모델의 구조는 Conference Server Manager, SIP Manager, 그리고 Media Manager로 구성되어 진다. Conference Server Manager는 SIP Manager 와 Media Manager를 로딩 및 관리하고, 컨퍼런스와 관련된 쓰레드 풀 및 쓰레드에 관련된 시스템 리소스를 관리한다. 그리고 Conference의 정보와 사용자의 정보를 갖는 State Table and Conference List가지고 있다. SIP Manager는 SIP Agent Manager와 서비스 module로 나누어진다. 컨퍼런스 서버의 쓰레드 풀링은 프로그램의 실행시 시작 단계에서, 별도의 쓰레드 스타트 과정없이 슬리핑(서스펜드) 상태의 쓰레드들을 이용해 새로운 작업을 도와 줄 수 있도록, 슬리핑 쓰레드를 만든다. 쓰레드 초기 생성 과정은 많은 시스템 리소스를 필요로 하므로, 비록 쓰레드 풀링 작업이 개별적인 쓰레드 생성 과정보다 좀 느릴지라도, 새로운 작업이 필요할 때마다 쓰레드를 사용할 수 있는 쓰레드 풀링은 여기서는 시그널쪽의 쓰레드와 미디어쪽 쓰레드를 쓰레드 풀링으로 만들어 놓고 컨퍼런싱 참여자들의 요청을 기다리므로 미디어 믹싱 및 많은 참여자들의 요청시 시스템 리소스를 절약할수 있다는 장점이 있다. SIP agent Manager는 서비스 요청이 왔을 경우에 SIP agent 를 풀에 요청을 해서 객체를 생성하고 서비스 요청에 대한 제어기능을 위한 Call admission을 포함하고 있다. 서비스 module은 시그널을 처리를 위한 서비스 모듈이며 동적으로 서비스 모듈을 로딩할 수 있다. Media Manager는 Media Agent Manager와 Media Service Module과 Translator/Mixer 그리고 Media Storage로 구성된다. Media Agent Manager는 전송받은 미디어가 있을 경우 Media Agent를 pool에서 요청해서 객체를 생성하고 시그널 과정에서 저장된 Conference List 및 상태정보에서 사용자의 자원을 파악하고 QoS selector두어 사용자 정보를 바탕으로 QoS 를 보장할 수 있게 한다. 여기서는 비디오 resolution에 대한 처리를 우선적으로 처리한다. 또한 RTCP 의 정보를 사용해서 처리 할 수 있는 모듈을 추가할 수 있도록 한다. 생성된 Media agent는 전송받은 미디어 데이터를 Media storage 에 저장하고 미디어 전송에 대한 처리를 담당한다. 또 사용자의 요청에 맞는 conference list/State table 을 참조하여 translator/mixer 에 처리를 요청한다. Service module는 미디어 에이젼트의 서비스 모듈을 가지고 있고 동적으로 로딩할 수 있다. Translator and mixer는 사용자 프로파일에 따라 미디어를 변환/혼합할 수 있다. Media storage는 미디어 데이터를 저장을 담당한다.

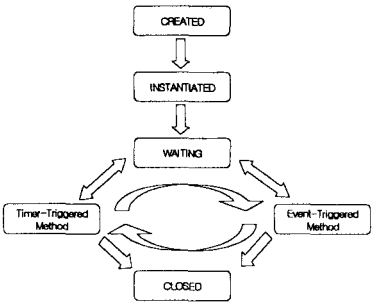
4. Agent Interface

본 논문의 Conference 시스템에서 사용자 서비스를 처리하는 Agent의 인터페이스는 그림 4과 같다.



<그림 4> Agent Interface

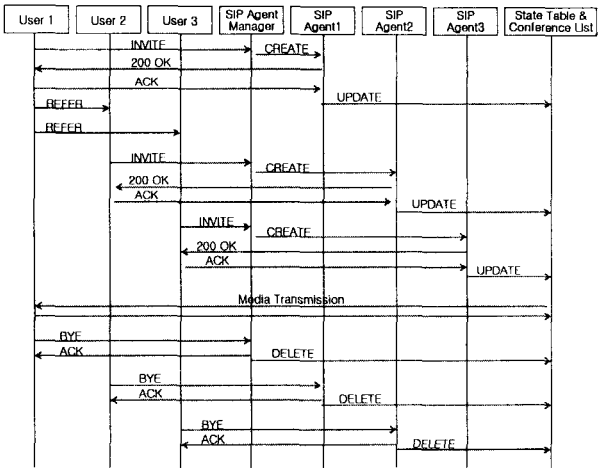
Agent 인터페이스는 두 가지 메소드를 가지고 있다. 첫째 Event를 받아 처리하는 방법으로 모든 메시지를 이벤트로 처리하는 Event-Triggered Method방식이 있다. 둘째 Timer를 두어 처리하는 방법으로 전송에서 발생되는 time-out에 대한 처리하는 Timer- Triggered Method이다. 실제 Agent 객체가 생성되어 위와 같은 메소드로 동작을 하게 되며 이에 따른 상태 다이어 그램을 그림 5에서 보여준다. 객체의 생성을 알리는 CREATED상태, 생성된 객체를 인스턴스시키는 INSTANTIATED상태, 인스턴스가 이루워진 상태에선 메소드 즉 Timer-Triggered Method와 Event-Triggered Method의 상태변이를 기다리는 WAITING상태, 각각의 메소드 상태는 서로의 상태변이를 관찰하고 마지막으로 컨퍼런스를 종료하고자 할때 생성된 객체들은 소멸시키는 역할의 CLOSED상태가 있다.



<그림 5> Agent State Diagram

5. 서비스 시나리오

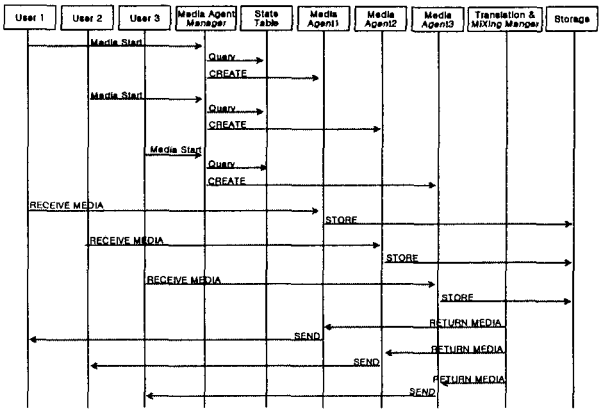
본 논문에서 컨퍼런스 구조에서 서비스를 제공하기 위한 각 구성요소간 제어관계를 본절에서는 소개를 한다.



<그림 6> Signal 서비스 시나리오

서비스 시나리오는 시그널 서비스 시나리오와 미디어 서비스 시나리오로 구분되어 진다. SIP Manager 입장에서 시그널 시나리오는 그림 6와 같다. User1은 컨퍼런스 서버의 SIP Agent Manager에 컨퍼런싱 서비스를 요청하는 INVITE 메시지를 보내고 이를 받은 SIP Agent Manager는 서비스 제어를 위한 Call admission을 확인하고 SIP Agent1를 쓰레드 풀에 요청하여 생성한다. 생성된 SIP Agent1은 생성이 되었다는 200 OK 메시지를 보내고 User1은 이에 대한 ACK 보낸다. SIP Agent1은 상태 테이블과 컨퍼런스 리스트를 업데이트한다. User1은 User2와 User3를 REFER메세지로 컨퍼런스에 초대한다[6]. REFER메세지를 받은 User2와 User3 역시 같은 형식으로 컨퍼런스 서버와 연결되고 State Table과 Conference List에 업데이트를 한다. Media Transmission 부분은 Media Manager가 처리하며 컨퍼런스를 종료할 때 User1은 SIP Agent1에게 BYE메세지를 보내고 SIP Agent 1은 이에 대한 ACK를 User1에게 보내고 State Table과 Conference List에 정보를 삭제한다.

Media Manager입장에서 미디어 처리 시나리오는 그림 7과 같다. 전송할 미디어가 있는 사용자 User1는 Media Agent Manager에 미디어를 전송하기 시작하면 이를 받은 Media Agent Manager는 State Table에서 상태 정보를 보고 미디어를 받을 수 있는지 여부를 판단하여 받을 수 있다면 Media Agent 1을 생성한다. 미디어를 전송할 사용자들은 모두 이런 과정을 거쳐 각각의 Media Agent를 생성한



<그림 7> Media 서비스 시나리오

다. 실제 미디어 데이터를 보낼 때 사용자 User1은 Media Agent1에게 미디어를 보내고 이를 받은 Media Agent1은 미디어 내용을 Storage에 저장한다. User1와 User2 역시 같은 절차로 동작한다. User1에게 미디어를 보낼 때에는 Storage에서 User2와 User 3의 미디어 데이터를 Translation과 Mixing Manger에서 트랜스 코딩 및 합성한 미디어 데이터를 Media Agent1에게 보내고 이를 User1에게 전송하고 이 과정은 User2와 User3도 같다.

시그널 및 미디어처리를 멀티쓰레드 및 쓰레드 풀 개념을 사용함으로써 관리가 용이하고 서비스를 효율적으로 처리를 할 수 있다.

6. 결론 및 향후 계획

본 논문은 효율적이고 보다 신속한 처리하는 Conference Server 설계를 위해 유효 리소스의 관리를 위해 멀티쓰레드 및 풀링 개념을 이용하였고 컨퍼런싱에 참여하는 사용자의 제어 및 자원에 따라 차별화된 서비스 위해 Call Admission 및 QoS Selector를 사용하였다. 향후 계획은 설계된 Conference Server를 구현하고 효율적인 미디어 처리를 위해서 Queue 관리 메카니즘 및 서버의 오버헤드를 줄이기 위해서 분산화된 모듈과 망 차원의 bandwidth를 고려하는 알고리즘이 추가적으로 연구되어야한다[7].

7. 참고 문헌

[1] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol",RFC2543 , March 1999

[2] J. Rosenberg, H. Schulzrinne, "Models for Multi Party Conferencing in SIP", draft-ietf-sipping-conferencing-models-01, January 2003

[3] Audio-Video Transport Working Group, H.Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC1889, January 1996

[4] J. Rosenberg, J. Peterson, H. Schulzrinne, G.Camarillo, "Best Current Practices for Third Party Call Control in the Session Initiation Protocol",draft-ietf-sipping-3pcc-02, June 5, 2002

[5] 최선완, "SIP 기반 다자간 컨퍼런스",안양대학교, July 25,2002, KRnet 2002 Conference

[6] R. Sparks, "The SIP Refer Method", draft-ietf-sip-refer-07, May 26, 2003

[7] Kundan Sing h, Gautam Nair and Henning Schulzrinne, "Centralized Conferencing using SIP" Columbia University