효율적인 call-control을 위한 Extension SIP 시스템 구현*

이정훈°, 양형규, 이병호
한양대학교 정보통신대학원
{blur96°, sheep, bhrhee}@scann.hanyang.ac.kr

The Implementation of Extension SIP system for efficient call-control

Jung Hun Lee°, Hyoung Kyu Yang, Byung Ho Rhee
Graduate School of Information and Communications, Hanyang University

요 약

본 논문에서는 SIP(Session Initiation Protocol) 기반의 VoIP(Voice over Internet Protocol) 시스템에서 효율적인 call-control을 위해 필요한 헤더와 파라미터를 추가한 Extension SIP를 제안하였다.

또한 이 제시된 Extension SIP에 따르는 SIP Proxy Server와 User Agent(User Agent Client, User Agent Server)를 리눅스 시스템에서 C언어를 통해 구현하였고, 이 구현된 Extension SIP 시스템을 통해 기존의 SIP 시스템과 call-control을 위한 packet traffic을 비교·분석 함으로써 제안한 Extension SIP의 효율성을 증명하였다.

1. 서 론

인터넷 텔레포니와 관련된 기술들의 개발이 빠르게 진행되면서 인터넷 전화를 비롯한 다양한 인터넷 서비스들이 제공되기 시작하고 있다. 또 이러한 VoIP의 발전에 부합하여 멀티미디어 전송이나 packet 전송을 통한 음성 통화의 call-control을 가능하게 해줄 프로토콜로 SIP가 대두되고 있으며 많은 연구가 수행되고 있다.[1]

본 논문에서는 기존의 SIP를 통해 사용된 call-control을 Extension SIP를 이용해 수신자가 등록한 수신자의 수신 상태에 따라 session 설정의 여부를 결정함으로 packet의 traffic을 줄일 수 있는 효율적인 call-control 시나리오를 제안하였다. 또한 제안된 Extension SIP를 기반으로 SIP proxy server와 User Agent를 구현하고 기존의 SIP 시스템과 call-control을 위한 packet의 traffic을 비교하였다. 2장에서는 SIP에 대해 간단하게 설명하고, 3장에서는 packet traffic을 줄이기 위한 Extension SIP와 효율적인 call-control 시나리오를 제안하고, 4장에서는 구현된 Extension SIP 시스템에 대하여 설명하고 테스트를 통해 기존의 시스템과 구현된 시스템간의 call-control을 위한 packet traffic을 비교하고 분석한다. 마지막으로 5장에서는 결론과 향후과제와 함께 글을 맺는다.

2. Session Initiation Protocol

SIP는 IETF(Internet Engineering Task Force)에서 제안한 프로토콜로서 session 설립, call participant 관리 등의 기능을 수행하고 session negotiation을 위해서 별도로 SDP(Session Description Protocol)를 사용한다.

또한 SIP는 HTTP(Hypertext Transfer Protocol)와 유사한 client-server 구조로 되어 있는데 User Agent와 네트워크 서버로 구성된다. User Agent는 다시 UA Client와 UA Server로 나뉘어 지고 서버는 register server와 proxy server 그리고 redirect server로 나뉘어 지게 된다.

*This work was supported by the research fund of Hanyang University.(HY-2003-T)

SIP의 메시지를 살펴보면 제어 절차를 위해 크게 request와 response, 두 종류의 메시지가 있다. INVITE, ACK, REGISTER, BYE와 같은 request 메시지는 session을 위한 정보를 서버에게 전달하고 response 메시지는 이러한 request 메시지에 대한 서버의 상태나 수신확인 여부를 답하는 메시지이다. 이러한 메시지들을 통해 session 설정을 위한 과정은 다음과 같다.

① 등록 단계: 모든 사용자들은 자신의 위치 정보인 SIP URL을 REGISTER 메시지를 통해 SIP register server에 등록한다.
② 호 설정 단계: 통화를 하고자 하는 상대방의 주소 정보를 입력한 INVITE 메시지를 proxy server에 전달하고, proxy server는 이 메시지는 상대방 UAS에 전달해 두 사용자간 호가 설정된다.
③ 음성 통화: 호가 설정이 되면 사용자간의 음성통화는 RTP session이 열려 음성 데이터가 전송됨으로써 이루어진다. 음성 통화를 위해 지원되는 코덱은 G.711, G.723 등이 있다.
④ 호 해지 단계: 음성 통화 중 호 해지는 수신자가 BYE 메시지를 전송하고 이에 대한 200OK response 메시지를 송신자가 보냄으로써 이루어진다.[2], [3], [4]

3. Extension SIP를 이용한 효율적인 call-control 제안

제시된 call-control은 각각의 도메인에 있는 proxy server에 사용자를 대신하는 권한을 주어 수신자 단말기에서가 아닌 proxy server에서 수신자의 상태에 따라 SIP INVITE 메시지를 돌려보내거나 받아들이는 시나리오이다. 이는 기존의 call-control 보다 적은 packet traffic과 발신자에게 보다 빠른 응답을 보내준다.

3.1 Extension SIP

제시된 call-control을 위해서는 SIP를 확장할 필요가 있다. 이는 proxy server에게 자신의 수신 상태를 알려주기 위해 헤더 필드가 필요하기 때문이다. 따라서 본 논문에서는 기존의

SIP REGISTER 메시지에 대한 필수적인 헤더 필드와 이에 대한 파라미터를 정의한다. 다음의 <표-1>에 정의한 헤더 필드와 파라미터를 보여주고 있다.

<표-1> 확장된 헤더 필드와 파라미터

Header Field Name	Parameter	Value
Reception-state	state	Can Receive
		Can't Receive
	reservation	ON
		OFF
	res-time	ex) Sat, 13 Nov 2010 23:29:00 GMT
	reason	ex) I'm sleeping

Reception-state 헤더 필드는 자신의 수신 상태를 나타내는 헤더 필드이다. state 파라미터는 자신의 수신 상태를 나타내는 파라미터로 수신이 가능하면 Can Receive를 보내고 수신이 불가능하면 Can't Receive를 보내게 된다. 또한 나머지 파라미터인 reservation, res-time, reason 파라미터는 state 값이 Can't Receive일 때만 사용한다. reservation 파라미터의 값이 OFF일 경우 등록된 후부터 바로 Can't Receive 상태가 되고 ON일 경우 res-Time부터 Can't Receive의 상태가 된다. 또한 reason 파라미터를 둠으로써 발신자에게 수신하지 못하는 이유를 알려주게 된다. 또한 제시한 Reception-state 필드와 함께 이미 SIP에 정의된 Retry-After도 함께 사용된다. Retry-After 필드는 서비스를 얼마동안 이용할 수 없는 지를 알려준다. 따라서 Reception-state의 state가 Can't Receive가 될 경우, 선택적으로 사용하게 됨으로 나중에 등록을 다시 할 필요가 없는 장점을 이용할 수 있다.

3.2 등록 과정

제시된 효율적인 call-control 시나리오를 사용하기 위해서는 수신자가 통화할 수 없는 경우 proxy server와의 등록을 시행하여야 한다. <표-2>는 제시한 헤더 필드를 이용한 등록과정의 예를 보여준다.

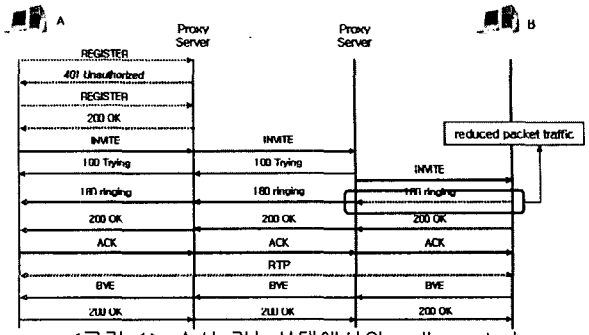
<표-2> REGISTER 메시지에 헤더가 사용되는 예

```
Reception-state: state="Can't receive", reservation="ON",
                 res-time="Sat, 21 Feb 2004 22:30:00 GMT",
                 reason="I'm sleeping"
Retry-After: 18000
```

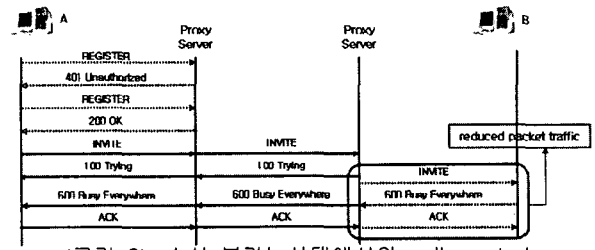
위의 예는 취침을 위해 2004년 2월 21일 토요일 22시 30분부터 18000초 동안 수신할 수 없다는 내용을 REGISTER 메시지를 통해 proxy server에게 보내주고 있다.

3.3 제안된 call-control 시나리오

<그림 1>과 <그림 2>는 제안된 시나리오를 사용해서 기존의 call-control 보다 얼마만큼의 traffic을 줄여주는 지를 보여주고 있다. <그림 1>은 수신자가 수신 가능한 상태에서의 call-control 시나리오이고, <그림 2>는 수신자가 수신할 수 없는 상태에서의 call-control 시나리오이다.



<그림 1> 수신 가능상태에서의 call-control



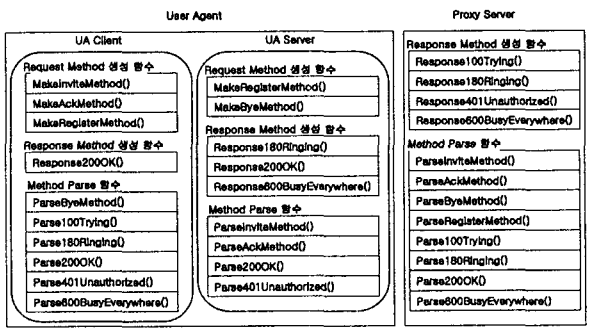
<그림 2> 수신 불가능상태에서의 call-control

4. Extension SIP 시스템 구현 및 테스트

본 논문에서는 리눅스 7.3 기반에서 C언어를 이용하여 Reception-state 헤더 및 그에 사용되는 state, reservation, res-time, reason 파라미터들이 포함된 User Agent와 Proxy server를 구현하였고 제안한 헤더를 포함할 때 기존의 SIP 시스템에서의 call-control보다 얼마나 call-control packet을 줄여줄 수 있는지 실험해 보았다.

4.1 Extension SIP 시스템

다음의 <그림 3>은 User Agent와 Proxy server내에서 사용된 함수를 바탕으로 User Agent와 Proxy Server의 구조를 보여준다. 본 시스템에서 proxy server는 register server의 역할까지도 같이 수행하도록 설계되었다.

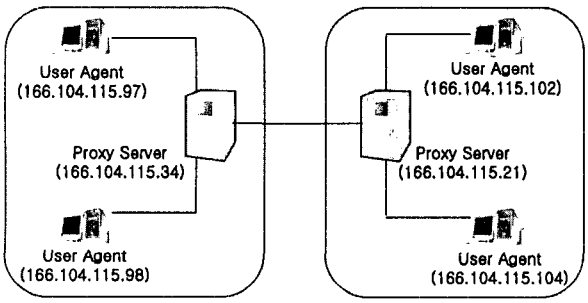


<그림 3> Extension SIP 시스템 구조

SIP는 텍스트 기반의 프로토콜이기 때문에 대부분의 함수들이 메시지를 생성하고 그것을 파싱하는 역할을 하고 있다. 또한 UA server에서의 REGISTER 메시지 생성 함수에서 제안한 Reception-state 헤더를 포함해서 메시지를 생성하고 proxy server에서 이것을 파싱하고 reception_state라는 구조체에 정보를 저장한다.

4.2 결과 분석

테스트는 1000회의 전화시 부재중 통화율(10%, 20%, 30%)에 따른 총 packet 수를 조사하였다. 또한 사용자가 모든 부재중 통화를 예방하기 위해 REGISTER 메시지를 proxy server에게 전달하는 것이 아니기 때문에 부재중 통화의 50%는 기존의 방식으로 call-control을 수행하게 정의하고 실험을 시행하였다. 실험을 위한 망 구성도는 다음의 <그림 4>와 같다.

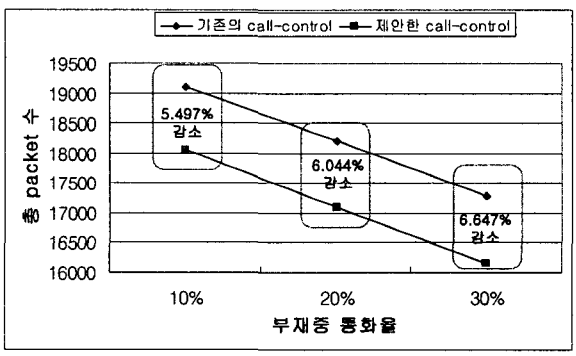


<그림 4> 실험을 위한 망 구성도

<표-3>을 보면 기존의 call-control과 Extension SIP를 이용한 call-control을 위해 사용된 총 packet 수를 부재중 통화율에 따라 보여주고 있다. 부재중 통화율이 10%일 때(즉 5%의 부재중 통화를 예방)는 기존의 call-control보다 5.497%가 감소되었고, 20%일 때(즉 10%의 부재중 통화를 예방 적용)는 6.044%가 감소되었고, 30%일 때(즉 15%의 부재중 통화를 예방)는 6:647%가 감소되었다.

앞에서 살펴본 시나리오에서 부재중 통화일 경우에 절약되는 packet의 수가 더 많기 때문에 부재중 통화율이 증가할수록 더 많은 비율의 packet이 감소하게 된다.

<표-3> 총 packet 수 비교 그래프



5. 결 론

본 논문에서는 Reception-state 헤더와 state, res-time, reservation, reason 파라미터들을 추가한 Extension SIP를 이용해 call-control을 위해 필요한 packet의 수를 감소시키는 효율적인 call-control을 제안하였다. 또한 제안한 Extension SIP를 적용한 proxy server와 user agent를 리눅스에서 C언어를 통해 구현하고, 이 시스템을 바탕으로 실험을 통해 기존의 call-control보다 부재중 통화율에 따라 5.497% ~ 6.647% 정도의 packet 감소를 측정할 수 있었다.

앞으로 인터넷 전화의 보급이 늘어남에 따라 인터넷 망에 packet traffic이 지금보다 훨씬 더 늘어날 전망이다. 이에 본 논문에서 제안한 Extension SIP를 이용해서 인터넷 전화를 위한 call-control의 traffic을 줄인다면 인터넷 전화의 발전에 보탬이 될 수 있으리라 전망한다.

참고 문헌

[1] I. Curcio and M. Lundan, "study of Call Setup in SIP-based Videotelephony", 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001), Vol.IV, pp. 1-6, July. 2001

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

[3] M. Handley and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, Apr. 1998

[4] J. Rosenberg, P. Mataga, H. Schulzrinne, "An application server component architecture for SIP", Internet Draft, IETF, March. 2001