

효율적 메시지 제어방식을 사용한 XMPP/SIP 프레즌스 서비스 시스템

장춘서*

XMPP/SIP Presence Service System using Efficient Message Control Method

Choonseo Jang*

Department of Computer Engineering, Kumoh Inst. of Tech. Gumi 730-701, Korea

요 약

본 논문에서는 XMPP(Extensible Messaging and Presence Protocol) 기반 프레즌스(Presence) 정보와 SIP(Session Initiation Protocol) 기반 프레즌스 정보를 모두 처리할 수 있는 XMPP/SIP 프레즌스 서비스 시스템에서 서버의 부하를 줄이고 네트워크 트래픽을 감소시킬 수 있는 새로운 메시지 제어 방식을 제안하였다. 이를 위하여 XMPP 프레즌스 스탠자(stanza)와 IQ 스탠자에 필요한 요소들이 추가된 새로운 XMPP 스탠자 구조가 설계되었고, SIP 통지 메시지와 PUBLISH 메시지 크기를 모두 줄일 수 있는 새로운 프레즌스 정보 데이터 처리 방식이 제안되었다. 아울러 XMPP 도메인과 SIP 도메인 사이에서 서로 프레즌스 정보를 전송할 수 있는 메시지 교환 절차가 제시되었다. 제안된 시스템의 성능은 시뮬레이션 실험을 통하여 분석하였다.

ABSTRACT

In this paper a new message control method that can reduce server load and network traffic in XMPP/SIP presence service system has been proposed. This presence service system can process both XMPP(Extensible Messaging and Presence Protocol) based presence information and SIP(Session Initiation Protocol) based presence information. A new XMPP stanza architecture with added elements for presence stanza and IQ stanza has been designed, and a new presence information data processing method which can reduce size of SIP notification message and SIP PUBLISH message has been suggested. Furthermore a messages exchanging procedure that can transfer presence information between XMPP domain and SIP domain has been also suggested. The performance of the proposed system has been analysed by simulation.

키워드 : 프레즌스 서비스, XMPP, SIP, 프레즌스 정보 데이터 포맷

Key word : Presence Service, XMPP, SIP, Presence Information Data Format

접수일자 : 2014. 07. 29 심사완료일자 : 2014. 08. 24 게재확정일자 : 2014. 09. 15

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

Department of Computer Engineering, Kumoh Inst. of Tech., Gumi 730-701, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2014.18.11.2678>

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

프레즌스 서비스는 사용자의 온라인 상태, 현재 위치, 네트워크 접속 방식 및 접속 주소 등 사용자에게 관한 다양한 관련 정보를 얻을 수 있어 네트워크 응용 분야에 널리 사용된다. 현재 SIP 기반의 프레즌스 서비스[1, 2]와 XMPP 기반의 프레즌스 서비스[3, 4]가 모두 사용되고 있으며 사용자들은 프레즌스 정보를 얻기 원하는 다수의 프레즌스 리소스에 대한 등록 요청 메시지를 프레즌스 서버에게 보내고 이후 등록된 프레즌스 리소스에서 프레즌스 정보의 상태 변화가 발생할 때마다 프레즌스 서버는 각 사용자들에게 프레즌스 통지 메시지를 보내며 따라서 사용자 수가 증가할수록 프레즌스 서버의 부하와 네트워크 트래픽이 크게 증가하게 된다.

본 논문에서는 프레즌스 서비스 시스템에서 사용자 수 증가에 따른 이와 같은 문제점을 해결하기 위하여 프레즌스 서버와 사용자 사이에 교환되는 XMPP 메시지 양을 줄이고 서버의 부하를 감소시키는데 필요한 스탠자 요소들이 추가된 새로운 XMPP 프레즌스와 IQ 스탠자 구조를 제안하였다. 또 본 연구에서는 서버가 사용자에게 보내는 SIP 프레즌스 통지 메시지의 크기와 사용자가 자신의 프레즌스 정보의 변화 시에 전송하는 SIP PUBLISH 메시지의 크기를 모두 줄일 수 있는 확장된 프레즌스 정보 데이터 포맷을 사용하는 데이터 처리 방식이 제안되었다. 아울러 XMPP 도메인과 SIP 도메인 사이에 프레즌스 정보를 전송할 수 있는 메시지 교환 절차를 새롭게 설계하여 SIP 기반의 프레즌스 서비스와 XMPP 기반의 프레즌스 서비스를 하나의 서버에서 모두 처리할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 XMPP 구성과 SIP 프레즌스 정보 데이터 포맷 및 기존 연구에 대해 설명하고 III장에서는 본 논문에서 새롭게 제안하는 XMPP/SIP 프레즌스 서버의 설계 및 구조를 설명하고 이어서 필요한 스탠자 요소들이 추가된 XMPP 프레즌스와 IQ 스탠자 구조 및 SIP 통지 메시지 크기를 줄일 수 있는 프레즌스 정보 데이터 처리 방식에 대해 설명한다. 다음 IV장에서는 구현된 시스템의 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

2.1. XMPP

XMPP[5-7]는 XML 형태의 메시지나 프레즌스 정보를 실시간에 근접한 속도로 교환하기 위한 표준 프로토콜이다. XML 스트림은 네트워크를 통해 XML 요소를 서로 교환하기 위한 컨테이너이며 <stream> 태그로 시작한다. 그림 1에 XML 스트림 내에 위치하는 XML 스탠자들을 보였다.

```

<stream>
<presence>
<show/>
</presence>
<message to='foo'>
<body/>
</message>
<iq to='bar'>
<query/>
</iq>
...
</stream>
    
```

그림 1. XML 스트림 예
Fig. 1 Example of XML Stream

XMPP 스탠자는 XML 스트림 내에서 전송되는 XML 요소들이며 메시지 스탠자(<message>), 프레즌스 스탠자(<presence>) 및 IQ 스탠자(<iq>)의 세 가지 종류가 있다.

현재 XMPP를 이용한 기능 구현에 초점을 맞춘 여러 종류의 프레즌스 및 인스턴트 메시징 시스템 관련 연구 [3, 4]가 있으나 이들 연구들은 사용자 수 증가에 따른 XMPP 시스템 부하의 최적화에 관련된 내용은 다루고 있지 않다. 본 연구에서는 이와는 달리 XML 스트림 내에서 교환되는 XMPP 메시지 양을 효율적으로 줄여 서버의 부하와 네트워크 트래픽을 감소시킬 수 있는 새로운 XMPP 스탠자 구조를 연구하였다.

2.2. SIP 프레즌스 정보 데이터 포맷

SIP 기반 프레즌스 서비스에서 프레즌스 서버와 사용자 사이의 프레즌스 정보를 교환하기위한 프레즌스 정보 데이터의 기본 포맷은 PIDF(Presence Information

Data Format)이고 XML 형태의 문서이다. 이 포맷은 최상위 요소(root element)로 <presence>를 가지며 각 프레즌스 정보를 구분하기 위한 하위 요소(child element)인 하나 이상의 <tuple>을 가진다. 이 <tuple>은 다시 하위 요소로 <status>를 가지며 이는 각 프레즌스 리소스의 현재 상태 정보를 나타낸다. 이들 외에도 프레즌스 정보를 나타내기 위한 여러 개의 하위 요소를 가질 수 있으며 이들 요소들은 필요한 경우 속성으로 추가적인 정보를 나타낼 수 있다. 그림 2에 이와 같은 프레즌스 정보 데이터의 예를 보였다.

```

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  entity="pres:student99@irt.kumoh.ac.kr">
  <tuple id="fjweu432">
    <status>
      <basic>open</basic>
    </status>
    <contact>sip:gildong@irt.kumoh.ac.kr</contact>
    <timestamp>2014-06-03T13:21:33Z
    </timestamp>
  </tuple id="kgjrit324j">
    <status>
      <basic>closed</basic>
    </status>
    <contact>sip:sonnim@irt.kumoh.ac.kr</contact>
  </presence>
  
```

그림 2. 프레즌스 정보 데이터 예
Fig. 2 Example of Presence Information Data

이와 같은 프레즌스 정보 데이터는 프레즌스 서버와 사용자 사이에서 통지 메시지와 SIP PUBLISH 메시지를 통해 서로 교환된다. 이때 프레즌스 서비스 사용자 수가 증가 할수록 등록되는 프레즌스 리소스 수 및 각 프레즌스 리소스에서 발생하는 프레즌스 정보의 양도 크게 증가하여 프레즌스 서버의 부하를 높이게 된다.

SIP 프레즌스 서버의 부하를 줄이기 위한 연구로는 기존에 보낸 프레즌스 정보 문서에서 변경된 부분만 전송하는 부분 통지(partial notification) 방식이나[8, 9] 리소스 리스트 방식을 사용하여 다량의 프레즌스 리소스를 등록하는데 걸리는 시간을 줄이는 방식[10] 등이 제안되고 있다. 본 연구에서는 이와는 달리 프레즌스 정보 데이터 포맷을 확장하여 자주 사용되는 문서 태그 부분을 압축하여 전송할 수 있도록 함으로써 SIP 프레즌스 통지 메시지의 크기 및 SIP PUBLISH 메시지의 크기를 모두 줄일 수 있는 SIP 프레즌스 정보 데이터 처리 방식을 제안하였다.

III. 시스템 설계 및 구현

3.1. XMPP/SIP 프레즌스 서버 구조 설계

본 논문에서 설계한 새로운 XMPP/SIP 프레즌스 서버의 구조를 그림 3에 보였다. 여기서 서버는 XMPP 프레즌스 제어모듈, XMPP 스트림 제어모듈, SIP 메시지 제어모듈, SIP 등록 제어모듈, SIP 통지 제어모듈, XMPP/SIP 변환 제어모듈 및 XMPP/SIP 프레즌스 정보 데이터베이스로 구성된다.

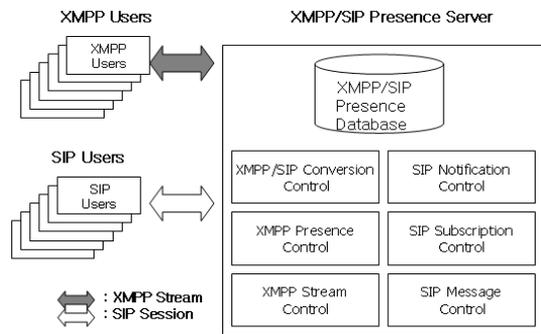


그림 3. XMPP/SIP 프레즌스 서버 구조
Fig. 3 Architecture of XMPP/SIP Presence Server

XMPP 스트림 제어모듈은 스트림 헤더를 처리하고 스트림 인증 및 리소스 바인딩을 한다. 이와 함께 XMPP 사용자와 프레즌스 서버사이의 스트림 생성 및 유지 기능도 처리한다.

XMPP 프레즌스 제어모듈은 각 사용자들의 프레즌스 리소스 목록을 XMPP/SIP 프레즌스 정보 데이터베이스에 생성, 수정 및 관리하고 XMPP 스트림 내의 프레즌스 및 IQ 스탠자를 처리하여 프레즌스 정보의 등록과 통지기능을 제어한다. 본 논문에서는 사용자가 자신의 프레즌스 리소스 목록에 있는 상대방의 현재 프레즌스 정보 데이터를 얻으려 할 때와 상대방의 프레즌스 정보 데이터가 변경되어 서버가 이를 등록된 사용자에게 알리는 프레즌스 정보 통지 동작 시 사용자와 프레즌스 서버 사이에 전송되는 메시지 양을 크게 줄일 수 있도록 이 모듈을 구현하였다.

상대방의 각 리소스 별로 프레즌스 스탠자가 발생하도록 동작하는 기존 방식과 달리 본 연구에서는 프레즌스 스탠자를 확장하여 새로운 프레즌스 스탠자 타입인 'new_probe'와 'new_notify'를 설계하여 <presence

type='new_probe' from='jabber_id_1' to='jabber_id_2'> 와 <presence type='new_notify' from='jabber_id_2' to='jabber_id_1'>처럼 사용하고 이러한 프레즌스 스탠자 타입을 수신 한 경우 각 리소스 대신 해당 사용자 주소로 1회만 전송하도록 설계하였다.

따라서 본 연구에서 제안한 구조를 사용하여 서버의 부하와 네트워크 트래픽을 모두 감소시킬 수 있게 된다. 아울러 사용자가 XMPP/SIP 프레즌스 정보 데이터베이스에 접근하여 프레즌스 리소스 목록을 추가, 삭제 또는 변경하는 동작에서도 메시지 양을 줄일 수 있도록 설계 하였다.

사용자가 IQ 스탠자를 사용하여 프레즌스 정보 데이터베이스에 접근하는 경우 기존의 방식에서는 프레즌스 서버가 변경된 프레즌스 정보 데이터를 사용자의 등록된 모든 프레즌스 리소스에게 보내고 각 리소스는 수신 결과를 다시 프레즌스 서버에게 보내야 했다.

이와는 달리 본 연구에서는 IQ 스탠자를 확장하여 새로운 타입인 'new_set'와 'new_result'를 설계하여 <iq from='jabber_id' type='new_set' id='roster_id'>와 <iq to='jabber_id' type='new_result' id='roster_id'>처럼 사용하도록 하였다. 이러한 IQ 스탠자 타입을 수신 한 경우 각 리소스 대신 해당 사용자 주소로 1회만 전송하도록 함으로써 사용자와 프레즌스 서버 사이에 전송되는 메시지 양을 줄일 수 있다.

SIP 메시지 제어모듈은 SIP 프로토콜 스택을 가지며 SIP 기반의 프레즌스 서비스 사용자와 프레즌스 서버 사이의 메시지 교환 및 연결 제어를 담당한다. SIP 등록 제어모듈은 각 SIP 사용자들의 프레즌스 리소스 목록을 XMPP/SIP 프레즌스 정보 데이터베이스에 생성, 수정 및 관리하고 SIP 통지 제어모듈은 등록된 사용자의 SIP 프레즌스 정보의 변화가 발생했을 경우 서버와 SIP 사용자 간의 통지기능을 제어한다.

마지막으로 본 연구에서는 자주 사용되는 문서 태그 부분을 압축하여 전송 할 수 있도록 SIP 프레즌스 정보 데이터 포맷을 확장하였다. 이를 위하여 SIP SUBSCRIBE 메시지의 Accept 헤더에 'application/pidf-compress+xml' 속성이 추가되었고 이와 같은 SIP SUBSCRIBE 메시지를 수신한 서버에서는 응답 메시지에 앞으로 양단간에 사용 할 압축 목록을 보내도록 하였다. 이후 해당 사용자에 대한 프레즌스 정보 통지 시 압축된 SIP 프레즌스 정보 데이터 포맷을 사용함으로써

써 통지 메시지 크기를 크게 줄일 수 있다. 이를 위하여 최상위 요소인 <presence>의 하위 요소로 <compress>가 설계되어 이의 하위요소로 압축된 프레즌스 정보 데이터의 항목들이 들어 갈 수 있도록 하였다.

이와 같은 방식으로 서버로부터 프레즌스 정보 데이터 압축 목록을 받은 사용자는 이후 자신의 프레즌스 정보의 변화 시에 생성해야하는 SIP PUBLISH 메시지에도 동일한 압축 방식을 적용 할 수 있어 SIP 프레즌스 통지 메시지의 크기 및 SIP PUBLISH 메시지의 크기를 모두 줄일 수 있게 된다.

3.2. 프레즌스 서버 동작 및 XMPP/SIP 메시지 교환절차

이상에서 설명한 프레즌스 서버의 동작 과정에 대한 순서도를 그림 4에 보였다. 여기서 XMPP 스트림 제어 모듈을 통하여 사용자와 프레즌스 서버사이의 스트림 연결이 되면 먼저 목적지 도메인이 XMPP인지 여부를 확인 후 XMPP인 경우 XMPP 프레즌스 스탠자와 IQ 스탠자 처리를 한다. 이때 XMPP 프레즌스 정보는 XMPP/SIP 프레즌스 정보 데이터베이스에 기록되며 프레즌스 정보의 처리는 3.1절에서 설명한 본 연구에서 제안된 새로운 방식을 사용하여 사용자와 프레즌스 서버 사이에 전송되는 메시지 양을 크게 줄일 수 있게 된다.

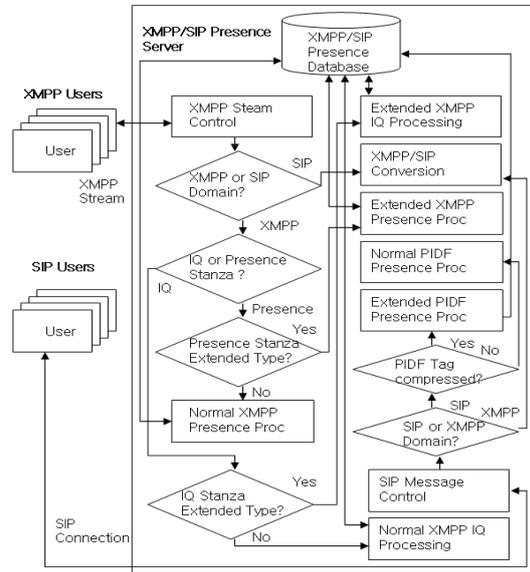


그림 4. 서버 동작 과정에 대한 순서도
Fig. 4 Flowchart of Server Operation Procedure

사용자가 SIP 메시지를 사용하는 경우 먼저 목적지 도메인이 SIP인지 여부를 확인하고 SIP 프레즌스 정보는 XMPP/SIP 프레즌스 정보 데이터베이스에 기록된다. SIP 메시지의 Accept 헤더에 'application/pidf-compress+xml' 속성이 있으면 확장 PIDF 프레즌스 처리 모듈에서 양단간에 사용할 압축 목록을 보내고 이후 해당 사용자와의 프레즌스 정보 교환 시 압축된 SIP 프레즌스 정보 데이터 포맷을 사용함으로써 통지 메시지 크기를 크게 줄일 수 있게 된다.

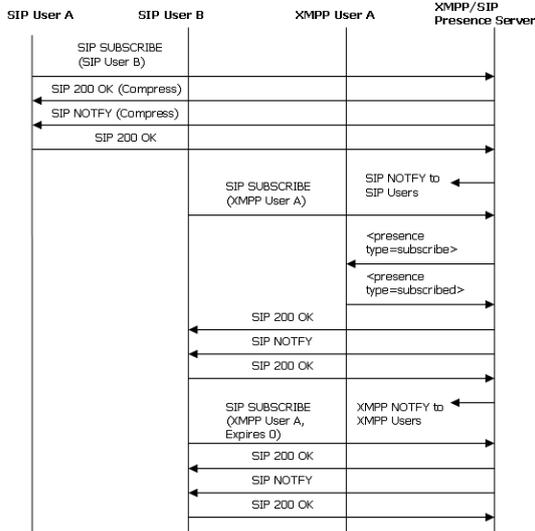


그림 5. 프레즌스 서버 및 사용자 사이의 메시지 교환 절차
 Fig. 5 Message Exchange Procedure between Presence Servers and Users

본 논문에서는 XMPP 도메인과 SIP 도메인 사이에 프레즌스 정보 교환이 가능하도록 하였으며 이 기능은 서버의 XMPP/SIP 변환 제어모듈에서 실행된다. 그림 5에 프레즌스 서버를 통한 SIP 사용자와 XMPP 사용자와 사이의 메시지 교환 절차를 보였다. 여기서는 SIP 사용자 A가 동일한 도메인의 SIP 사용자 B에 대해 SIP SUBSCRIBE 메시지를 프레즌스 서버에게 보내고 이후 본 연구에서 제시한 압축된 프레즌스 정보 데이터 포맷으로 정보를 교환한다.

SIP 사용자 B는 자신의 영역이 아닌 XMPP 사용자 A에 대한 등록 요청을 하면 서버는 XMPP/SIP 변환을 통해 XMPP 사용자 A에 대한 프레즌스 등록을 한다.

SIP 사용자 B가 XMPP 사용자 A에 대해 SIP SUBSCRIBE 메시지의 Expire 헤더 값이 0인 요청을 보내면 프레즌스 서버는 내부적으로 본 논문에서 제시한 <presence type='new_probe'> 스탠자를 생성하여 SIP NOTIFY 메시지를 사용해 해당 XMPP 사용자 A의 프레즌스 정보를 알려준다.

IV. 성능 분석

본 논문에서 제안한 XMPP/SIP 프레즌스 서비스 시스템의 성능을 분석하기 위하여 먼저 본 연구에서 제안한 새로운 프레즌스 스탠자 타입인 'new_probe'와 'new_notify'를 적용 시킨 경우와 그렇지 않은 경우를 사용자 수를 증가시켜가며 프레즌스 서버에서의 프레즌스 정보 처리 시간을 비교 측정하였다. 이를 위하여 시뮬레이션 환경에서 각 사용자는 5개의 리소스를 가지며 사용자 당 1초에 5회의 프레즌스 요청을 하는 것으로 설정하였다. 그림 6에 이의 결과를 보였다.

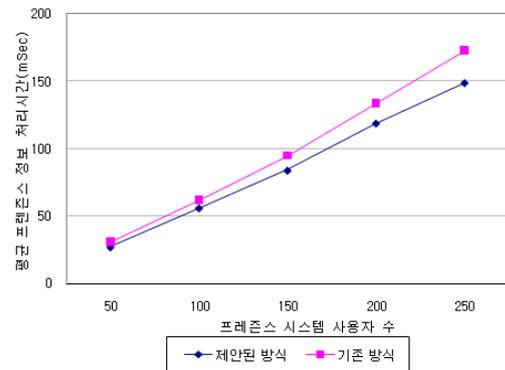


그림 6. 평균 프레즌스 정보 처리시간(프레즌스 스탠자 사용)
 Fig. 6 Average Presence Information Processing Time (using presence stanza)

여기서 제안된 방식을 사용하는 경우 사용자 수가 50명 일 때는 평균 처리 시간이 7.8% 감소하고, 150명 일 때는 11.6% 감소하고, 250명 일 때는 13.9%가 감소하여 사용자가 수가 증가할수록 본 논문에서 제안한 방식의 개선 효과가 커짐을 알 수 있다.

다음 본 연구에서 제안한 IQ 스탠자와 프레즌스 스탠자가 시스템 성능에 미치는 영향을 분석하기 위하여

이들을 모두 적용 시킨 경우와 그렇지 않은 경우에 대해 사용자 수를 증가시켜가며 프레즌스 서버에서의 평균 처리 시간을 비교 측정하였다. 이를 위하여 시뮬레이션 환경에서 각 사용자는 5개의 리소스를 가지며 사용자 당 1초에 5회의 프레즌스 요청과 1회의 프레즌스 리소스 목록 갱신 요청 동작을 하는 것으로 설정하였다. 그림 7에 이의 결과를 보였다.

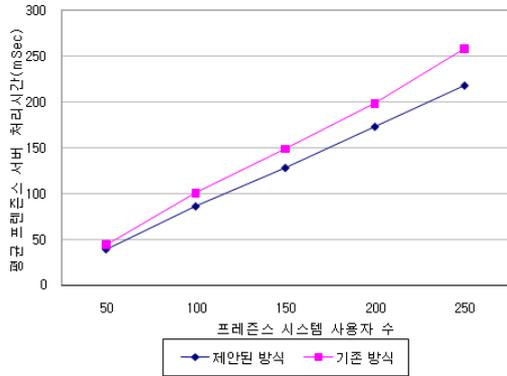


그림 7. 평균 프레즌스 서버 처리 시간
Fig. 7 Average Presence Server Processing Time

여기서 제안된 방식을 사용하는 경우 사용자 수가 50명 일 때는 평균 처리 시간이 11.4% 감소하고, 150명 일 때는 14.0% 감소하고, 250명 일 때는 15.8%가 감소하여 사용자가 수가 증가할수록 본 논문에서 제안한 방식의 개선 효과가 커짐을 알 수 있다.

다음 SIP 프레즌스 정보 데이터 포맷을 확장하여 자주 사용되는 문서 태그 부분을 압축하여 전송 할 수 있도록 제안한 방식의 성능을 기존 방식과 비교하기 위하여 10개의 <tuple>로 구성된 PIDF 포맷의 프레즌스 문서를 사용자당 1초에 10개씩 전송하도록 하였다. 이때 압축되는 문서 태그는 <status>, <contact>, <servcaps> 및 <devcaps>이다. 그림 8에 메시지 전송에 사용되는 프레즌스 서버에서의 평균 처리 시간을 비교 측정한 결과를 보였다. 여기서 제안된 방식을 사용하는 경우 사용자 수가 50명 일 때는 평균 처리 시간이 9.5% 감소하고, 150명 일 때는 11.2% 감소하고, 250명 일 때는 13.8%가 감소하여 사용자가 수가 증가할수록 본 논문에서 제안한 방식의 개선 효과가 커짐을 알 수 있다.

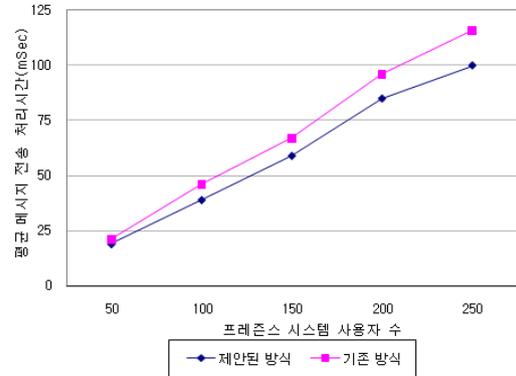


그림 8. 평균 프레즌스 서버 처리 시간(압축 태그 사용)
Fig. 8 Average Presence Server Processing Time(using compressed tag)

V. 결론

본 논문에서는 프레즌스 서버와 사용자 사이에 교환되는 XMPP 메시지 양을 줄이고 서버의 부하를 감소시키는 데 필요한 요소들이 추가된 새로운 XMPP 프레즌스 스탠자와 IQ 스탠자 구조를 제안하였다. 또 프레즌스 정보 데이터 포맷을 확장하여 자주 사용되는 문서 태그 부분을 압축하여 전송 할 수 있도록 함으로써 SIP 프레즌스 통지 메시지와 SIP PUBLISH 메시지의 크기를 모두 줄일 수 있는 방식을 제안하였다. 제안된 시스템의 성능 분석을 위한 측정 결과 본 연구에서 제안한 새로운 프레즌스 스탠자와 IQ 스탠자를 적용한 경우 서버에서의 평균 프레즌스 정보 처리 시간이 11.4%에서 15.8%까지 감소하여 사용자가 수가 증가할수록 제안한 방식의 개선 효과가 커짐을 보여 주었다. 또 PIDF 포맷의 프레즌스 문서 태그 부분을 압축하여 전송하는 경우 측정 결과로 9.5%에서 13.8%까지 감소하여 역시 사용자가 수가 증가할수록 제안한 방식의 개선 효과가 커짐을 보여 주었다. 향후 과제로는 확장성을 높이기 위하여 다중 서버 구조를 가진 XMPP/SIP 프레즌스 서비스 시스템을 연구 할 필요가 있다.

감사의 글

본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

REFERENCES

- [1] Arup Acharya et al, "Presentials: a flexible middleware for presence-enabled applications," *Proceedings of the 5th International Conference on Principles, Systems and Applications of IP Telecommunications*, August 2011.
- [2] Robin Acker et al, "Ubiquitous home control based on SIP and presence service," *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, November 2010.
- [3] Ronny Klauck and Michael Kirsche, "Combining Mobile XMPP Entities and Cloud Services for Collaborative Post-Disaster Management in Hybrid Network Environments," *Mobile Networks and Applications*, Volume 18 Issue 2, pp. 253-270, April 2013.
- [4] Huber Flores and Satish Srirama, "Mobile cloud messaging supported by XMPP primitives," *MCS '13: Proceeding of the fourth ACM workshop on Mobile cloud computing and services*, June 2013.
- [5] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP):Instant Messaging and Presence," RFC 6121, March 2011.
- [6] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," RFC 6120, March 2011.
- [7] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Address Format," RFC 6122, March 2011.
- [8] M. Lonnfors, E. Leppanen, H. Khartabil and J. Urpalainen, "SIP Extension for Partial Notification of Presence Information", RFC 6261, September 2011.
- [9] A. Niemi, M. Lonnfors and E. Leppanen, "Publication of Partial Presence Information", RFC 5264, September 2008.
- [10] G. Camarillo , A.B. Roach and O. Levin, "Subscriptions to Request-Contained Resource Lists in the Session Initiation Protocol (SIP)," RFC 5367, October 2009.



장춘서(Choonsoo Jang)

1993년 8월 : 한국과학기술원 공학박사
현재 : 금오공과대학교 컴퓨터공학과 교수
※관심분야 : SIP, 임베디드 시스템, 인터넷텔레포니