

## SIP 환경에서의 새로운 프레즌스 리스트 서비스

장춘서\*

### A New Presence Lists Service in SIP Environment

Choon-Seo Jang\*

#### 요약

SIP(Session Initiation Protocol) 환경에서 프레즌스(presence) 이벤트 통지(notification) 동작은 프레즌스 서비스 사용자가 상대방의 프레즌스 정보의 상태 변화를 알 수 있도록 한다. 이를 위하여 기존의 방식에서 사용자는 등록하고자하는 상대방의 주소를 담은 SIP SUBSCRIBE 요청 메시지를 프레즌스 서버에게 각각 보내야 했다. 또 각 프레즌스 등록에 대해서는 주기적으로 리프레시 메시지가 필요하고 등록된 상대방에서 발생하는 통지 메시지의 양도 모두 합할 경우 그 양이 많아지므로 이 경우 네트워크 트래픽을 크게 발생시키고 프레즌스 서버의 부하를 증가시킨다. 본 논문에서는 이와 같은 문제점을 해결하기위하여 프레즌스 정보를 얻기 원하는 상대방의 URI로 구성된 프레즌스 리스트를 사용하여 한번의 SUBSCRIBE 메시지로 프레즌스 등록을 할 수 있도록 함으로써 SUBSCRIBE 메시지 및 리프레시 메시지 양을 크게 줄일 수 있고 등록된 상대방으로부터 발생하는 이벤트 통지 메시지의 처리양도 크게 줄일 수 있는 새로운 프레즌스 리스트 서비스 방식을 제안하였다. 또 동작 효율을 더욱 높이기 위하여 필요한 프레즌스 정보만을 선택하여 통지 메시지에 포함 시킬 수 있는 프레즌스 필터링 방식도 새롭게 제안되었다. 이와 같이 하여 구현된 시스템은 SIP 메시지 전송에 관련된 네트워크 트래픽을 크게 줄일 수 있고 프레즌스 서버에서의 메시지 처리 시간을 감소시킨다. 제안된 시스템의 성능은 실험을 통하여 분석하였고 서버에서의 처리시간이 감소함을 보였다.

#### Abstract

In SIP(Session Initiation Protocol) environment, the presence event notification operation makes presence service user to recognize presence information status changes of the other party. In conventional method, the user should send each SIP SUBSCRIBE message containing the address of the other party to the presence server. Furthermore each presence subscription requires refresh messages, and the amount of notification messages from the other party becomes large. These facts increases network traffic and the load of presence server. In this paper, to solve these

• 제1저자 : 장춘서

• 투고일 : 2010. 08. 20, 심사일 : 2010. 09. 17, 게재확정일 : 2010. 10. 04.

\* 금오공과대학교 컴퓨터공학부 교수

※ 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

problems a new presence list service has been suggested. It uses a presence list consists of URI of the other party and make it possible to subscribe by using one SUBSCRIBE message. So this method decreases the amount of SUBSCRIBE messages and refresh messages, and it also decreases the amount processing event notification messages from the subscribed the other party. And to increase the efficiency, a presence filtering method which enables to include some parts of selected presence information into the notification message body has been also suggested. The implemented system can reduce SIP message related network traffic and message processing time of the presence server. The performance of this proposed system has been evaluated by experiments ,and the results showed decreasing of server processing time.

▶ Keyword : SIP(Session Initiation Protocol), 프레즌스 서비스(Presence Service), 프레즌스 필터링(Presence Filtering)

## I. 서론

프레즌스 서비스[1][2][3]를 사용하는 SIP[4] 환경에서 사용자들은 보통 많은 수의 상대방들에 대한 프레즌스 정보를 얻기 원하며, 이를 위하여 기존의 방식에서는 사용자가 상대방 주소를 담은 SIP SUBSCRIBE 요청 메시지를 서버에게 각각 보내야했다. 이 과정에서 많은 수의 SUBSCRIBE 메시지가 필요하고 각각의 등록 요청에 대해서는 주기적으로 리프레시 메시지도 아울러 필요하므로 발생하는 메시지의 양은 더욱 커지게 된다.

또 등록된 상대방에 대해서는 지속적으로 프레즌스 통지(notification) 메시지[5]가 발생하므로 사용자가 등록된 상대방의 수가 많아질수록 전체적인 메시지 발생량은 더욱 커지게 되며, 이는 시스템 전체의 메시지 트래픽을 크게 발생시키고 서버의 부하를 증가시키며 특히 통신 대역폭의 제한을 받는 모바일 환경이나 무선 환경에서의 프레즌스 서비스의 원활한 사용을 어렵게 만든다.

이와 같은 문제점을 해결하기위하여 본 연구에서는 프레즌스 정보를 얻기 원하는 상대방의 URI로 구성된 프레즌스 리스트를 사용하여 한번의 SUBSCRIBE 메시지로 프레즌스 등록을 할 수 있도록 하여 SUBSCRIBE 메시지 및 등록 상태 유지를 위한 리프레시 메시지 양을 크게 줄일 수 있고 서버에서의 SIP NOTIFY 메시지 양도 줄일 수 있는 새로운 방식을 제안하였다. 제안된 시스템에서는 동작 효율을 더욱 높이기 위하여 필요한 프레즌스 정보만을 선택하여 통지 메시지에 포함 시킬 수 있는 프레즌스 필터링 방식도 아울러 제안되었다.

본 연구에서 구현된 프레즌스 서비스 시스템에서 사용되는 프레즌스 필터링을 위해서 XML 문서 포맷을 가지는 application

/pres-filter+xml MIME 타입이 새롭게 제안되었다. 이 문서 포맷을 사용하여 사용자는 전체 프레즌스 정보 중 통지 메시지에 포함시키거나 제거시킬 부분만 선택 할 수 있으며, 프레즌스 정보의 특정한 상태 변화에 따라 통지 메시지를 발생시킬 수 있다. 이와 같이 하여 구현된 프레즌스 서비스 시스템은 메시지 전송에 관련된 네트워크 트래픽을 크게 줄일 수 있고 프레즌스 서버의 부하를 감소시킨다. 제안된 시스템은 다양한 조건에서의 실험을 통하여 성능을 분석하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 프레즌스 서비스 시스템 및 프레즌스 정보 포맷에 대해 설명하고, III장에서는 본 논문에서 새롭게 제안하는 프레즌스 리스트 서비스를 구현하기 위한 프레즌스 리스트 서버 및 프레즌스 필터링 기능을 설계 및 구현하며, IV장에서는 구현된 시스템의 성능 분석을 한 후 V장에서 결론을 맺는다.

## II. 관련연구

### 1. 프레즌스 서비스

SIP 환경에서의 프레즌스 서비스는 사용자의 온라인 상태나 사용 가능한 통신 서비스의 종류, 접속 주소 등에 관한 각종 정보를 제공하며 특히 인스턴트 메시징 시스템[6]에서 반드시 필요한 요소이다. 프레즌스 서비스의 기본 구성 요소는 프레즌스 정보를 담은 SIP PUBLISH 메시지[7][8]를 생성하고 이를 전송하는 PUA(Presence User Agents)와 각각의 PUA로부터 전송된 프레즌스 정보를 모아서 처리하는 PA(Presence Agent), 그리고 처리된 내용의 알림 대상이며 프레즌스 정보를 실제로 수신하는 사용자(watcher)이다. 그림 1에 이의 구성을 보였다.

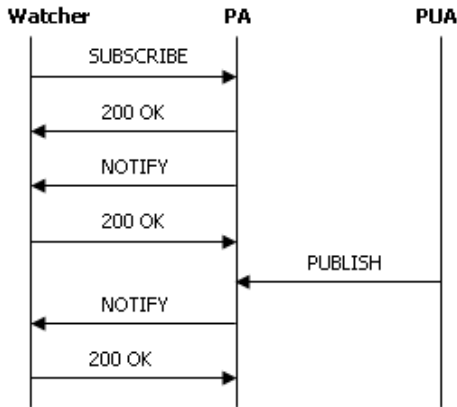


그림 1. 프레즌스 서비스 구성요소  
Fig.1 Elements of Presence Service

그림 1에서 PA는 SIP 프록시(proxy) 서버 기능을 함께 가진 프레즌스 서버이며 사용자는 프레즌스 서버에게 필요한 프레즌스 서비스 등록을 하며 이때 SIP SUBSCRIBE 메시지가 사용된다. SUBSCRIBE 메시지는 서버에서 사용자 인증을 거쳐 프레즌스 이벤트 등록 작업이 수행되고 등록된 자원에서 프레즌스 정보의 변화가 있는 경우 프레즌스 서버는 SIP NOTIFY 메시지를 사용하여 해당 사용자에게 통지한다. 프레즌스 이벤트 등록은 유효기간이 있으며 등록을 유지하기 위해서는 지속적으로 프레즌스 서버에게 리프레시 메시지를 보내야한다.

이와 같이 프레즌스 서비스를 사용하기 위해서는 각 프레즌스 자원들에 대한 등록 메시지, 각 자원에서의 통지 메시지 그리고 각 자원에 대한 등록을 유지하기 위한 리프레시 메시지가 필요하므로 등록된 자원의 수가 많이 질수록 전체적인 메시지 발생량은 상당히 커지게 된다. 본 논문에서는 이를 해결하기 위하여 새로운 프레즌스 자원 리스트 서비스 시스템을 연구하였다.

## 2. 프레즌스 정보 포맷

프레즌스 정보의 기본 포맷은 XML 문서의 문법을 따른 PIDF(Presence Information Data Format)[9]를 사용한다. 프레즌스 정보 포맷에서는 최상위 요소로 <presence>가 있고 하위 요소로 각각의 프레즌스 정보를 구분하기 위한 하나 이상의 <tuple>을 갖는다. 각 <tuple>은 해당 사용자의 현재 상태 정보를 가지고 있는 요소인 <status>를 반드시 하위요소로 가져야하고 그 외에 여러 추가적인 하위요소를 가질 수 있다. 이들 추가적인 요소들 중에는 다른 이름공간에서 정

의된 요소들도 가능하다. 다음에 프레즌스 정보 문서의 예를 그림 2에 보았다.

```

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:im="urn:ietf:params:xml:ns:pidf:im"
  xmlns:myex="http://id.example.com/presence/"
  entity="pres:someone@example.com">
  <tuple id="bs35r9">
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
      <myex:location>home</myex:location>
    </status>
    <contact>sip:gildong@202.31.200.86</contact>
    <timestamp>2010-07-27T16:49:29Z
    </timestamp>
  </tuple>
</presence>
    
```

그림 2. 프레즌스 정보 문서의 예  
Fig.2. Example of Presence Information Document

이 문서에서는 <tuple>의 속성으로 'id'가 사용되어 다른 <tuple>과 구분 하고 있고 추가적인 정보를 나타내기 위한 하위요소로 해당 사용자의 위치 정보를 가지고 있는 <contact>, 다른 이름공간에서 정의된 요소들인 <im:im>과 <myex:location>과 문서에 대한 시간 정보 요소인 <timestamp> 등이 사용되었다. 프레즌스 문서의 MIME 형태는 application/pidf+xml로 나타낸다.

이와 같은 프레즌스 정보 문서는 SIP NOTIFY 메시지의 몸체 부분에 넣어 사용자에게 보내지거나 SIP PUBLISH 메시지를 통해 PUA에서 프레즌스 서버에게 보내진다. 이때 문서의 양을 줄이기 위하여 기존에 보낸 프레즌스 정보 문서에서 변경된 부분만 전송하는 부분 통지(partial notification) 방식[10][11]도 함께 사용된다.

## III. 시스템 설계 및 구현

### 1. 프레즌스 리스트 서버

본 논문에서 설계한 프레즌스 리스트 서버는 프레즌스 정보를 얻기 원하는 상대방의 URI로 구성된 프레즌스 리스트를 처리하는 부분과 전체 프레즌스 정보 중 통지 메시지에 포함시키거나 제거시킬 부분만 선택하고 프레즌스 정보의 특정한 상태 변화에 따라 통지 메시지를 발생시키는 기능을 하는 메시지 필터링 부분으로 구성된다. 프레즌스 리스트 처리부분에서는 우선 프레즌스 리스트 등록을 위한 SIP SUBSCRIBE 메시지를 처리하는데 이때 서버와 사용자 사이에 프레즌스 리

스트 지원 여부를 확인하기 위해서 SIP Supported 헤더와 SIP Require 헤더를 사용하였다. 프레즌스 리스트를 지원하는 사용자는 SIP SUBSCRIBE 메시지의 Supported 헤더에 'preslist' 태그를 포함시키고 이 메시지를 받은 프레즌스 리스트 서버는 이에 대한 응답 메시지의 Require 헤더에 'preslist' 태그를 넣어 보내도록 설계하였다. 프레즌스 리스트 서버는 이후의 모든 SIP NOTIFY 메시지에 이 'Require:preslist' 식으로 포함시킨다.

본 연구에서는 프레즌스 리스트를 구현하기 위하여 application/preslist+xml MIME 타입을 새롭게 설계하였고 이는 XML 문서 포맷을 가진다. 새롭게 설계된 이 프레즌스 문서는 SIP NOTIFY 메시지에 포함된다. 사용자 시스템은 SIP SUBSCRIBE 메시지의 ACCEPT 헤더에 이와 같은 MIME 타입을 표시하도록 하였고 프레즌스 리스트 서버는 통지 메시지의 Content-Type 헤더에 이를 표시하도록 하였다. 본 논문에서 제안된 application/preslist+xml MIME 문서 모델의 첫 번째 부분에는 프레즌스 리스트에 대한 메타 정보 부분이 오고 나머지 부분에는 실제 프레즌스 정보가 오도록 설계하였다.

이 application/preslist+xml 문서 모델에서 최상위 원소(element)는 <preslist>이다. 이 원소의 속성(attribute)으로는 'version'과 'Partial' 을 가지도록 설계하였고 여기서 'version'은 문서의 버전을 나타내고, 'Partial'은 문서가 부분 프레즌스 정보를 포함하는지 여부를 나타낸다. 부분 프레즌스 정보를 담은 문서일 경우 'Partial=true' 속성값을 가진다. 원소 <preslist>의 하위원소(child element)로 하나 이상의 <presentity> 원소가 온다. <presentity> 원소의 속성으로는 'pruri'가 있고 이는 실제 프레즌스 정보를 나타낸다. 원소 <presentity>의 하위원소로는 등록된 상대방을 표시하기 위한 <descr>과 실제 프레즌스 자원의 상태를 나타내기 위한 <prinstance>를 두었다. 원소 <prinstance>의 속성으로는 해당 프레즌스 자원의 인스턴스를 구별하기 위한 'prid'와 해당 프레즌스 자원에 대한 등록 상태를 나타내기 위한 'state'를 두었다. 속성 'state'는 등록 상태가 유효일 때 값으로 'active'를 가지며 이때 속성 'prcid'는 메타 정보 이후에 나오는 문서 부분에서 실제 프레즌스 정보의 위치를 나타내는데 사용되도록 하였다.

다음 그림 3에 이와 같이 설계된 프레즌스 리스트에 대한 메타 정보를 가지는 문서의 예를 보였다. 여기서는 세 개의 <presentity> 원소가 사용되었고 그 중 두개에는 실제 프레즌스 자원의 상태를 나타내기 위한 <prinstance> 원소가 사용되었고 나머지 하나는 다른 도메인에서 관리하는 프레즌스

리스트가 사용되었다. 이와 같은 메타 정보에 의해 생성된 실제 프레즌스 리스트의 예를 다음 그림4에 보였다.

```
<?xml version="1.0"?>
<preslist xmlns="urn:ietf:params:xml:ns:prli"
  uri="sip:pres_group@lists.kumoh.ac.kr"
  version="3" Partial="false">
  <presentity
    pruri="sip:gildong@sip.kumoh.ac.kr">
    <descr>gildong's presence Information</descr>
    <prinstance
      prid="juwigmtboe" state="active"
      prcid="12345,aaa@sip.kumoh.ac.kr"/>
    </presentity>
  <presentity
    pruri="sip:dave@sip.kumoh.ac.kr">
    <descr>Dave's presence Information</descr>
    <prinstance
      prid="ehfuenfd" state="active"
      prcid="njert@sip.kumoh.ac.kr"/>
    </presentity>
  <presentity
    pruri="gildong_group@pres.ex.net">
    <descr>Gildong Group in ex domain</descr>
  </presentity>
</preslist>
```

그림 3. 프레즌스 리스트에 대한 메타 정보의 예  
Fig.3. Example of Meta Information for Presence Lists

```
Content-Transfer-Encoding: binary
Content-ID: <12345,aaa@sip.kumoh.ac.kr >
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  entity="sip:gildong@sip.kumoh.ac.kr">
  <tuple id="gjijy63">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
</presence>

--Yhrtujg67Ujif943Dfrjk
Content-Transfer-Encoding: binary
Content-ID: < njert@sip.kumoh.ac.kr/>
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  entity="sip:dave@sip.kumoh.ac.kr">
  <tuple id="jfrug7ew">
    <status>
      <basic>closed</basic>
    </status>
    <contact priority="1,0">
      sip:dave@mail.kumoh.ac.kr</contact>
  </tuple>
</presence>
```

그림 4. 실제 프레즌스 리스트 정보의 예  
Fig.4. Example of Real Presence List Information

위의 그림 4에서는 프레즌스 리스트에 대한 메타 정보에서 <prinstance> 원소를 사용하여 실제 프레즌스 정보를 나타내

는 예이며 이 경우 <status> 원소의 하위원소인 <basic>의 값으로 하나는 'open' 값을 갖고 다른 하나는 'closed' 값을 가짐을 알 수 있다.

2. 프레즌스 필터링

프레즌스 필터링은 전체 프레즌스 정보 중 통지 메시지에 포함시키거나 제거시킬 부분을 선택 할 수 있어 서버에서 전송되는 통지 메시지의 양을 줄일 수 있으므로 네트워크 트래픽과 메시지 전송에 관련된 서버의 부하도 줄일 수 있다. 프레즌스 필터링을 위해서 본 연구에서는 XML 문서 포맷을 가지는 application/pres-filter+xml MIME 타입이 새롭게 제안되었다. 이 포맷에서는 최상위 원소로 <presfilter\_list>를 가지며 하위원소로 각 필터링 기능을 표시하기위한 <spec>을 가진다. <spec>의 속성으로는 문서 내에서 프레즌스 필터들을 구분하기위한 'filterid'와 필터가 적용되는 해당 프레즌스 자원을 나타내기 위한 'filteruri'가 있고 하위원소로는 실제 통지 메시지에 포함 시킬 프레즌스 정보와 그 조건을 나타내기 위한 <filterON>과 제거 시킬 프레즌스 정보와 그 조건을 나타내기 위한 <filterOFF>가 있도록 설계하였다.

또 프레즌스 정보의 특정한 상태 변화에 따라 통지 메시지를 발생시키기 위하여 <NotifyStart> 원소를 두었다. 이 원소의 속성으로 'fromstate'와 'tostate'를 두어 통지 메시지를 발생시키기 위한 상태 변화의 조건을 나타낼 수 있도록 하였다. 그림 5에 이와 같이 설계된 프레즌스 필터링 포맷의 예를 보였다.

```
<?xml version="1.0" encoding="UTF-8"?>
<presfilter_list xmlns="urn:ietf:params:xml:ns:pres-filter">
  <spec filterid="eyu34g"
    filteruri="sip:dave@sip.kumoh.ac.kr">
    <filterON>
      ::[class="SMS" or class="MMS"]::status::basic
    </filterON>
    <NotifyStart fromstate="closed" tostate="open">
      ::tuple::status::basic
    </NotifyStart>
  </spec>
</presfilter_list >
```

그림 5. 프레즌스 필터링 포맷의 예  
Fig.5. Example of Presence Filtering Format

여기서는 프레즌스 정보 중 <class> 원소의 값이 'SMS' 또는 'MMS'인 경우 <status>의 하위원소인 <basic>의 값을 포함하라는 의미이며 또 통지 메시지 발생을 위한 조건으로 <basic>의 값이 'closed'에서 'open'으로 변화 할 경우를 명시하고 있다.

그림 6은 본 연구에서 설계된 프레즌스 리스트 서버의 내부 구성이다. 프레즌스 리스트 서버는 SIP 메시지의 파싱을 포함하여 SIP 동작 관련 기본적인 통신 기능을 처리하는 SIP 메시지 처리모듈 부분과 앞에서 상세히 설명한 프레즌스 리스트 처리 모듈 및 프레즌스 필터링 처리모듈을 가지고 있다. 또 통지메시지 처리모듈은 기존에 보낸 프레즌스 정보 문서에서 변경된 부분만 전송하는 부분 통지 방식 기능을 포함한 SIP NOTIFY 메시지 발생과 관련 한 작업을 처리한다.

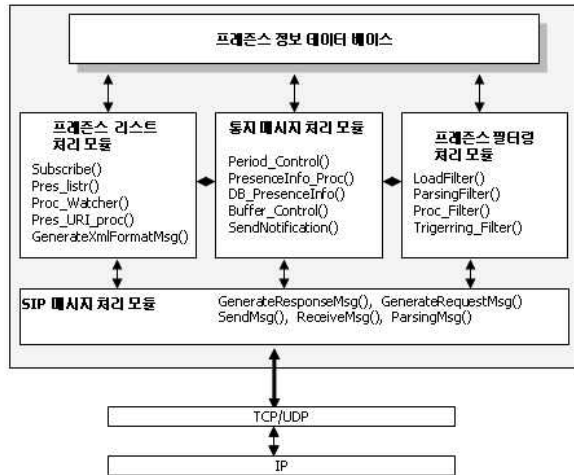


그림 6. 프레즌스 리스트 서버의 내부 구성  
Fig.6 Internal Structure of Presence Lists Server

IV. 성능 분석

본 논문에서 제안한 프레즌스 리스트 서비스 시스템의 성능 분석을 위하여 운영체제로 커널 버전 2.6인 리눅스를 가진 PC가 프레즌스 리스트 서버로 사용되었다. 일반 사용자용으로는 MS 윈도우즈 XP를 설치한 15대의 PC가 사용되었다. 각 PC의 사양은 CPU는 펜티엄 IV 2.4GHz이고 메인메모리 용량은 1GB이다. 각 PC의 네트워크 인터페이스 속도 및 허브를 포함한 LAN 환경의 속도는 100Mbps이고 모든 PC는 하나의 LAN 세그먼트로 구성된 동일한 서브넷(subnet) 상에 위치하도록 배치하였다.

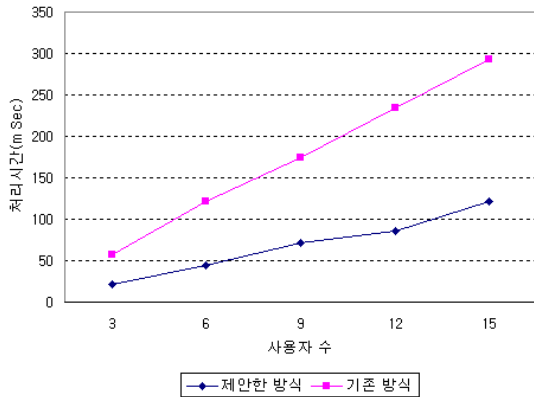


그림 7. SUBSCRIBE 메시지의 서버 처리 시간 비교  
Fig. 7. Comparison of Server Processing Time for SUBSCRIBE Message

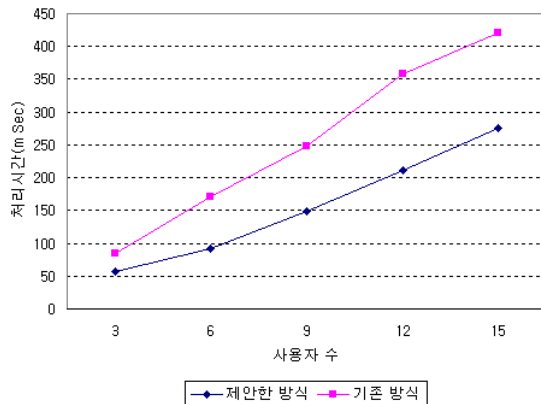


그림 8. 통지 메시지의 서버 처리 시간 비교  
Fig. 8. Comparison of Server Processing Time for Notification Message

본 논문에서 제안한 프레즌스 리스트 서비스의 효율을 측정하기 위하여 각 사용자들이 5개씩의 프레즌스 자원에 SUBSCRIBE 메시지를 보내도록 하고 프레즌스 서버에서 이들 요청을 처리하는데 걸리는 시간을 제한된 프레즌스 리스트를 사용한 방식과 기존 방식에 대해 서로 비교하였다. 측정 결과를 그림 7에 보였다. 이 그림은 사용자 수를 증가시켜 가며 서버에서의 처리 시간을 비교한 결과이다. 여기서는 본 논문에서 제안한 프레즌스 리스트 방식이 기존의 방식에 비해 시스템 서버에서의 처리 시간을 사용자 수에 따라 3명일 때 36msec에서 15명 일 때 172msec 까지 단축시키며 프레즌스 서비스 사용자 수가 증가 할수록 감소폭이 커짐을 보여주고 있다. 이는 본 논문에서 제안한 방식이 기존 방식에 비해 사용자 수가 증가 할수록 서버에 전송하는 SUBSCRIBE 메시지 수를 크게 감소시키기 때문으로 분석된다.

그림 8은 프레즌스 서버에서 통지 메시지를 처리하는데 걸리는 시간을 제한된 프레즌스 리스트를 사용한 방식과 기존 방식에 대해 서로 비교한 결과이다. 이때 사용한 프레즌스 정보는 5Kbyte 크기의 PIFD 포맷의 프레즌스 문서이며 각 사용자 당 5개씩의 프레즌스 자원에서 모두 프레즌스 정보가 생성되는 것으로 처리하였고 프레즌스 필터링 기능은 사용하지 않았다. 여기서는 본 논문에서 제안한 방식이 기존의 방식에 비해 프레즌스 서버에서의 처리 시간을 사용자 수에 따라 28msec에서 146msec 까지 단축시키며 프레즌스 서비스 사용자 수가 증가 할수록 역시 감소폭이 커짐을 보여주고 있다. 이는 본 논문에서 제안한 방식이 기존 방식에 비해 사용자 수가 증가 할수록 서버에서 생성해 사용자에게 전송해야할 통지 메시지 수를 줄여주기 때문으로 분석된다.

그림 9는 통지 메시지에 프레즌스 필터링 기능을 사용하였을 경우와 사용하지 않았을 경우를 비교한 측정 결과이다. 각 프레즌스 자원에서 생성된 프레즌스 정보 중 <basic>과 <contact> 원소만 필터링하여 전송하는 것으로 하였다. 프레즌스 필터링을 사용하는 경우 시스템 서버에서의 처리 시간은 사용자 수가 3명에서 15명까지 증가함에 따라 41msec에서 228msec까지 측정되었다. 반면에 프레즌스 필터링을 사용하지 않는 경우는 57msec에서 275msec까지 측정되었고 따라서 사용자 수에 따라 16msec에서 47msec까지 처리 시간을 단축시킬 수 있음을 보여주고 있다. 이는 필터링을 사용함으로써 사용자들에게 전송하는 통지 메시지의 크기가 줄기 때문으로 분석되며 네트워크 트래픽도 줄일 수 있고 사용자 측에도 필요한 프레즌스 정보만 담은 작은 크기의 통지 메시지를 받을 수 있어 메시지 처리 시간을 줄일 수 있는 장점이 있다.

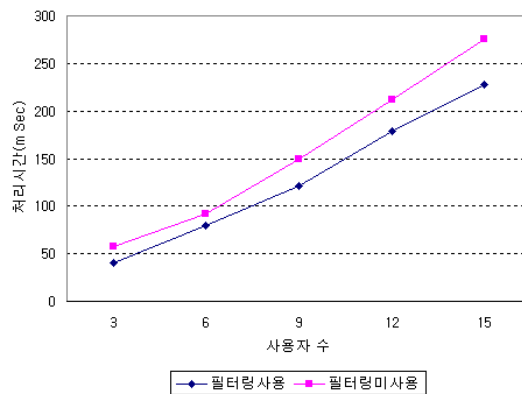


그림 9. 프레즌스 필터링 사용 시 서버 처리 시간 비교  
Fig. 9. Comparison of Server Processing Time with Presence Filtering

## V. 결론 및 향후 과제

본 논문에서는 프레즌스 정보를 얻기 원하는 상대방의 URI로 구성된 프레즌스 리스트를 사용하여 한번의 SUBSCRIBE 메시지로 프레즌스 등록을 할 수 있도록 함으로써 SUBSCRIBE 메시지 및 리프레시 메시지 양을 크게 줄일 수 있고 등록된 상대방으로부터 발생하는 이벤트 통지 메시지의 처리양도 크게 줄일 수 있는 새로운 프레즌스 리스트 서비스 방식을 제안하였다. 이를 위하여 XML 문서 포맷을 가지는 application/preslist+xml MIME 타입을 새롭게 설계하였다. 이와 같은 포맷의 프레즌스 문서는 SIP NOTIFY 메시지에 포함된다. 또 동작 효율을 더욱 높이기 위하여 필요한 프레즌스 정보만을 선택하여 통지 메시지에 포함시킬 수 있는 프레즌스 필터링 방식도 함께 제안되었다. 이를 위하여 application/pres-filter+xml MIME 타입이 새롭게 설계되었고 이를 사용하여 사용자는 전체 프레즌스 정보 중 통지 메시지에 포함시키거나 제거시킬 부분만 선택 할 수 있으며, 프레즌스 정보의 특정한 상태 변화에 따라 통지 메시지를 발생시킬 수도 있다. 이와 같이 하여 구현된 프레즌스 리스트 서비스는 네트워크 트래픽을 줄일 수 있고 서버에서의 메시지 처리 시간을 감소시킨다. 제안된 시스템은 다양한 조건에서의 실험을 통하여 성능을 분석하였고 서버 처리시간이 감소됨을 확인하였다. 향후 과제로는 다수의 단말 장비를 갖춘 무선 모바일 환경을 구축하여 제안된 시스템이 무선 모바일 환경에서 얼마만큼 효율적인가를 관측하고 이에 맞춘 개선된 방식을 연구 할 필요가 있다.

## 참고문헌

- [1] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," RFC 3856, August 2004.
- [2] J. Rosenberg, "A Watcher Information Event Template-Package for the Session Initiation Protocol", RFC 3857, August 2004.
- [3] 장춘서, "SIP 환경에서의 확장 CPL을 사용한 새로운 인스턴트 메시징 시스템", 한국컴퓨터정보학회논문지, 제14권, 제 9호, 67-74쪽, 2009년. 9월.
- [4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "Session Initiation Protocol,"

RFC 3261, June 2002.

- [5] A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.
- [6] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.
- [7] A. Niemi, Ed., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.
- [8] A. Niemi, M. Lonnfors, E. Leppanen, "Publication of Partial Presence Information", RFC 5264, September 2008.
- [9] H. Sugano, G. Klyne, "Presence Information Data Format (PIDF)," RFC 3863, August 2004.
- [10] M. Lonnfors, E. Leppanen, H. Khartabil, J. Uralainen, "Presence Information Data format (PIDF) Extension for Partial Presence", RFC 5262, September 2008.
- [11] M. Lonnfors, et al., "Session Initiation Protocol (SIP) extension for Partial Notification of Presence Information", Internet-Draft, January 2008.

## 저자소개



### 장 춘 서

1993년 8월 : 한국과학기술원 공학 박사

현재 : 금오공과대학교 컴퓨터공학부 교수

관심분야 : SIP, 인터넷텔레포니, 임베디드 시스템