

# SIP 메시지 제어 세션에 의한 대용량 분산 프레즌스 서비스 시스템

장춘서\*

## A Large Scale Distributed Presence Service System by SIP Message Control Session

Choonseo Jang\*

**요 약** 프레즌스 서비스는 사용자의 위치, 온라인/오프라인 여부, 네트워크 접속 방식 등 네트워크 사용자 관련 다양한 정보를 제공하며 모바일 환경에서 각 사용자들이 요구하는 프레즌스 리소스의 수는 크게 증가하고 있다. 따라서 이를 처리하기 위하여 프레즌스 서버의 부하를 효율적으로 감소시킬 수 있는 방안이 필요하다. 본 논문에서는 다수의 사용자에 대한 다양한 프레즌스 정보를 제공하는 대용량 분산 프레즌스 서비스 시스템 구현에 메시지 제어 세션을 사용하여 프레즌스 서버의 부하를 효율적으로 분산 처리 할 수 있는 방식을 제안하였다. 본 연구에서는 이를 위하여 사용자 수의 증가에 따른 프레즌스 서버의 부하를 동적으로 복수 개의 서버에게 효율적으로 분산시키기 위한 메시지 제어 세션 구조가 제안되었고 프레즌스 서버 부하 제어를 위한 새로운 프레즌스 정보 데이터 구조가 설계되었다. 이 구조에서 각 프레즌스 서버들은 현재 부하 레벨을 실시간으로 교환하며 사용자 수 변화에 따른 전체 시스템 부하의 변화를 파악하여 각 서버의 부하 레벨이 고르게 유지되도록 분산한다. 제안된 프레즌스 서비스 시스템의 성능은 실험으로 분석하였다. 실험 결과 본 연구에서 제안한 구조의 경우 평균 프레즌스 등록 처리 시간이 기존 방식에 비해 42.6%에서 73.6%까지 감소함을 보여주었고 평균 프레즌스 통지 처리 시간은 37.6%에서 64.8%까지 감소함을 보여주었다.

**Abstract** Presence service provides various information about users such as locations, status of on/offline and network access methods, and number of presence resources required by each users increases largely in mobile environment. Therefore an effective method which can reduce load of presence servers is needed. In this paper, a large scale distributed presence service system which can distribute effectively total presence system load of presence servers using message control session has been presented. This large scale distributed presence service system provides various presence information for massive volumes of users. In this study, a new message control session architecture which can dynamically distribute loads of the presence servers to multiple servers has been presented, and a new presence information data architecture for controlling load of the presence servers has been designed. In this architecture, each presence server can exchange current load level in real time to get variance of the total system load change according to user numbers, and can distribute system load to maintain load level of each server evenly. The performance of the proposed large scale distributed presence service system has been analysed by experiments. The results has been showed that average presence resource subscription processing time reduced from 42.6% to 73.6%, and average presence notification processing time reduced from 37.6% to 64.8%.

**Key Words** : Load Control, Message Control Session, Notification Message, Presence Service, Presence Information Data Format, Session Initiation Protocol

### 1. 서론

프레즌스 서비스는 사용자의 위치, 온라인/오프라인 여부, 네트워크 접속 방식 등 네트워크 사용자 관련

This Paper was supported by research Fund, Kumoh National Institute of Technology.

\* Department of Computer Engineering, Kumoh National Institute of Technology

Received September 19, 2018

Revised October 01, 2018

Accepted October 10, 2018

다양한 정보를 제공한다[1-3]. 현재 모바일 환경에서 각 사용자들이 요구하는 프레즌스 리소스의 수는 크게 증가하고 있으며 이를 위해 프레즌스 서버는 각종 프레즌스 리소스[4]에 대한 등록을 위해 사용자들이 보내온 다량의 SIP SUBSCRIBE 메시지를 처리해야 하고, 해당 프레즌스 리소스에서 프레즌스 정보의 변화가 있을 때 발생하는 다량의 프레즌스 NOTIFY 메시지도 역시 처리해야 한다. 따라서 사용자 수 증가에 의한 프레즌스 서버의 부하도 급격히 증가하므로 다중 서버 구조를 사용하여 프레즌스 서버의 부하를 효율적으로 분산 처리 할 수 있는 방안이 필요하다.

이를 위하여 본 논문에서는 메시지 제어 세션 구조를 프레즌스 시스템에 적용 할 수 있도록 설계하여 각 프레즌스 서버의 부하를 동적으로 제어하고 서버의 부하 레벨이 증가하면 동적으로 서버가 추가될 수 있는 분산 프레즌스 서비스 시스템 구조를 제시하였다.

이 구조에서 각 프레즌스 서버들은 현재 부하 레벨을 실시간으로 교환하며 각 서버의 부하 레벨이 고르게 유지되도록 분산한다. 사용자 수가 일정 범위를 초과하여 동작 중인 모든 프레즌스 서버들의 부하 레벨이 지정된 값을 초과할 경우에는 대기중인 서버 풀에서 동적으로 새로운 프레즌스 서버가 시스템에 추가되도록 하여 부하를 분산 처리한다. 이와 아울러 프레즌스 시스템의 전체 정보를 담고 있는 프레즌스 정보 데이터[5-7] 구조도 프레즌스 서버 사이에 적용한 메시지 제어 세션 구조에 맞추어 설계한 확장 프레즌스 정보 데이터 포맷도 함께 제시되었다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 프레즌스 서비스 시스템 및 기존 연구에 대해 설명하고 3장에서는 메시지 제어 세션 구조를 적용한 분산 프레즌스 서버의 구성 및 동작을 설명하고 이어서 프레즌스 서버의 부하를 제어하는데 필요한 기능들이 추가된 메시지 제어 세션 구조의 설계에 대해 설명한다. 다음 4장에서는 구현된 프레즌스 서비스 시스템의 성능 분석을 한 후 5장에서 결론을 맺는다.

## 2. 관련연구

### 2.1 SIP 프레즌스 정보 데이터

SIP 프레즌스 정보 데이터 포맷은 최상위 엘리먼트(root element)로 <presence>를 가지며 그림1에 프레즌스 정보 데이터 포맷의 예를 보였다.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  entity="pres:pretest1804@sip.kumoh.ac.kr">
  <tuple id="fjsurw326he">
    <status>
      <basic>open</basic>
    </status>
    <contact>sip:stud76@sip.kumoh.ac.kr</contact>
    <timestamp>2018-02-03T08:02:07
    </timestamp>
  </tuple id="fjsurw326he">
  <tuple id="jhbvyreq742hj8">
    <status>
      <basic>closed</basic>
    </status>
    <contact>sip:test321@sip.kumoh.ac.kr</contact>
  </tuple id="jhbvyreq742hj8">
</presence>
```

그림 1. 프레즌스 정보 데이터 포맷 예  
Fig. 1. Example of Presence Information Data Format

이 프레즌스 정보 데이터는 프레즌스 서비스 시스템에서 프레즌스 서버들 사이 및 서버와 사용자 사이에 프레즌스 정보를 서로 교환하기위하여 사용된다. 최상위 엘리먼트 밑으로 프레즌스 정보를 구분하기 위한 복수개의 자식 엘리먼트(child element)인 <tuple>이 위치한다. 엘리먼트 <tuple>은 각 프레즌스 리소스의 정보를 구분하기 위하여 속성(attribute) 'id'를 가지며 자식 엘리먼트로 각 프레즌스 리소스의 현재 상태 정보를 나타내는 <status>를 가진다. 엘리먼트 <tuple>은 프레즌스 정보 데이터에 반드시 포함되어야 하는 필수 엘리먼트이다.

엘리먼트 <tuple>은 <status> 외에도 프레즌스 리소스에 액세스하기위한 주소정보인 <contact> 나 프레즌스 리소스의 시간정보인 <timestamp> 등 추가적인 자식 엘리먼트를 가질 수 있으며 이들도 추가적인 정보를 나타내는 속성값을 가질 수 있다.

### 2.2 프레즌스 서비스 시스템

그림 2에 프레즌스 서비스 시스템의 기본 동작을 보였다.

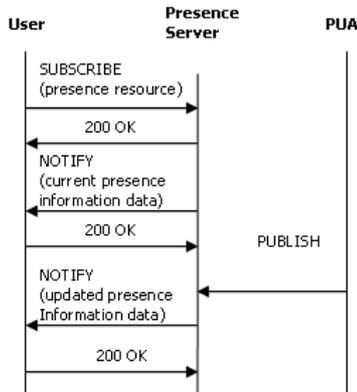


그림 2. 프레즌스 서비스 시스템의 기본 동작  
Fig. 2. Basic Operation of Presence Service System

그림 2에서 프레즌스 서비스 사용자는 프레즌스 리소스의 주소를 담은 SIP SUBSCRIBE 메시지를 프레즌스 서버에게 보내어 등록한다. 프레즌스 서버는 이에 대한 응답으로 해당 프레즌스 리소스에 대한 현재 상황의 프레즌스 정보 데이터를 SIP NOTIFY 메시지에 담아 알려준다. 등록된 프레즌스 리소스에 대해서 프레즌스 정보의 변화가 발생한 경우 PUA(Presence User Agents)는 SIP PUBLISH 메시지[8]를 사용하여 프레즌스 정보의 변화를 프레즌스 서버에게 알려주고 프레즌스 서버는 해당 프레즌스 리소스를 등록된 프레즌스 서비스 사용자에게 SIP NOTIFY 메시지를 사용해 이를 통지한다.

현재까지 제안되고 있는 프레즌스 서버의 부하를 줄이기 위한 방식으로는 프레즌스 통지 메시지의 양을 줄이거나 문서 태그 부분을 압축하여 전송하는 방법 및 XMPP[9]를 사용하는 방식 등이 있다[10]. 그러나 메시지 제어 세션을 사용하여 프레즌스 서비스 시스템에서 다중 서버의 부하를 분산시키는 예는 아직 발표되고 있지 않다. 본 연구에서는 메시지 제어 세션 구조를 사용하여 사용자 수의 변화에 따른 전체 시스템 부하의 변동을 각 프레즌스 서버에게 효율적으로 분산하고 필요시 동적으로 프레즌스 서버가 추가될 수 있도록 하는 대용량 분산 프레즌스 서비스 시스템 구조를 새롭게 제안하였다.

### 3. 시스템 설계 및 구현

#### 3.1 분산 프레즌스 서버 구조 설계

본 연구에서 설계한 분산 프레즌스 시스템에서의 서버 구조를 그림 3에 보였다. 그림 3에서 각각의 프레즌스 서버는 프레즌스 정보 데이터베이스, 부하 제어 모듈, 메시지 제어 세션 모듈, 프레즌스 정보 데이터 처리 모듈 및 SIP 스택으로 구성된다.

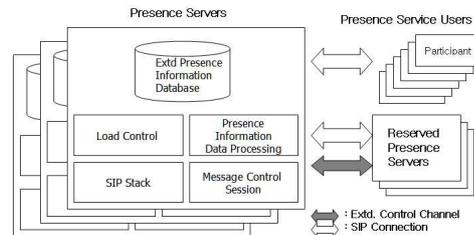


그림 3. 분산 프레즌스 시스템 서버 구조  
Fig. 3. Server Architecture of Distributed Presence Service System

프레즌스 정보 데이터베이스는 메시지 제어 세션 구조에 맞추어 필요한 엘리먼트들이 추가되도록 설계된 확장 프레즌스 정보 데이터 포맷을 가진다. 여기에는 프레즌스 서비스 시스템 전체 사용자에게 대한 프레즌스 정보 외에도 전체 시스템 부하를 분산하는데 필요한 각 프레즌스 서버들의 부하 레벨 등의 정보를 저장하고 처리한다. 부하 제어 모듈은 메시지 제어 세션을 통하여 각 프레즌스 서버의 부하 레벨을 실시간 모니터링하고 부하 제어에 필요한 제어 메시지를 서로 교환한다.

메시지 제어 세션 모듈은 프레즌스 서버들 간에 TCP 기반의 신뢰성 있는 SIP 제어 세션을 설정하고 전체 시스템 부하의 분산 제어를 위한 제어 메시지를 처리한다. SIP 스택은 프레즌스 서버들 사이 및 서버와 사용자간의 SIP 세션 생성 및 유지를 담당한다.

프레즌스 서버들 중 프레즌스 서비스 시스템의 URL을 가지고 있는 주 프레즌스 서버는 SIP INVITE 메시지를 동작 중인 각 서버들에게 전송하며 이때 메시지의 'Content-Type'은 'application/extdpres-sdp'이고 메시지 내의 SDP[11] 'm(media description

entry) 값은 'extdpres'으로 설정된다. 각 프레즌스 서버는 이에 대한 응답으로 SDP 'a(attribute)' 항목에 자신의 세션 ID 값을 'extdpres-id' 필드 이후에 넣어 전송한다. 주 프레즌스 서버는 이에 대해 확인(acknowledge) 메시지를 보내어 프레즌스 서버 간에 제어 세션을 설정하고 이후 프레즌스 서버 부하 제어를 위한 메시지들이 이 제어 세션을 통하여 서로 전송된다.

프레즌스 서비스 사용자가 프레즌스 서비스 URL이 포함된 SIP SUBSCRIBE 메시지를 사용하여 프레즌스 리소스 등록 요청을 하면 주 프레즌스 서버의 부하 제어 모듈은 프레즌스 정보 데이터베이스를 액세스하여 얻은 현재 동작 중인 각 서버의 부하 레벨을 나타내는 <curr-pressvr-ldlvl> 엘리먼트의 값을 사용해 현재 부하 레벨이 가장 작은 프레즌스 서버를 찾는다. 다음 이 서버의 URL을 담은 SIP Redirection 메시지를 사용해 해당 사용자와 SIP 세션을 설정하고 프레즌스 리소스 등록 처리를 하도록 한다. 이 서버는 해당 사용자에게 SIP NOTIFY 메시지를 보내어 등록된 프레즌스 리소스의 현재 프레즌스 정보 데이터를 알려준다. 본 논문에서는 기본적인 프레즌스 정보 데이터 포맷을 확장하여 <curr-pressvr-ldlvl> 엘리먼트가 포함되도록 설계하였다.

현재 동작 중인 모든 프레즌스 서버의 부하 레벨이 미리 설정된 최대값에 도달한 경우에는 새로운 프레즌스 서버 추가 단계가 시작된다. 주 프레즌스 서버는 프레즌스 정보 데이터베이스에서 예비 프레즌스 서버의 목록을 얻고 이들 중 선택된 서버에게 SIP INVITE 메시지를 보낸다. 이 메시지의 구성은 'Content-Type' 값으로 'application/extdpres-sdp', 메시지 내의 SDP 'm(media description entry)' 값은 'extdpres' 이고 'a(attribute)' 값은 'connection:new'이다. 상대방 프레즌스 서버로부터 SIP 메시지에 'Content-Type' 값으로 'application/extdpres-sdp', 'a(attribute)' 값으로 'setup:passive' 가 들어있고 'extdpres-id' 필드에 자신의 세션 ID 값이 들어있는 응답이 오면 서로 간에 동작 가능한 메시지 제어 세션이 설정된다.

이어서 주 프레즌스 서버는 이 제어 세션을 통하여 추가된 프레즌스 서버에게 프레즌스 서비스 시스템의

전체 프레즌스 정보 데이터를 전송하고 SIP Redirection 메시지를 사용해 해당 사용자와 연결해 프레즌스 리소스 등록 처리를 하도록 한다. 새롭게 추가된 프레즌스 서버는 해당 사용자에게 SIP NOTIFY 메시지를 보내어 등록된 프레즌스 리소스의 현재 프레즌스 정보 데이터를 알려준다. 그림 4에 이와 같은 시스템 동작 과정에 대한 흐름을 보였다.

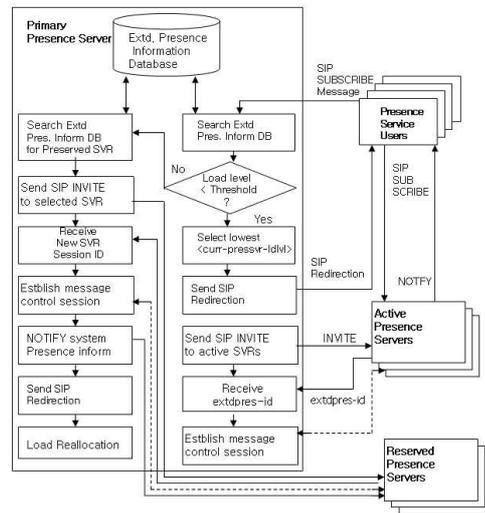


그림 4. 시스템 동작 과정 흐름  
Fig. 4. Operational Procedure Flow of the System

### 3.2 프레즌스 서버를 위한 제어 세션 구성

프레즌스 서버를 위한 메시지 제어 세션에서는 먼저 확장 SDP 포맷으로 'Content-Type'이 'application/extdpres-sdp'인 SIP INVITE 메시지가 프레즌스 서버 사이에 전송되어 제어 세션이 설정된다. 이와 같은 제어 세션을 통하여 전체 프레즌스 시스템의 정보를 담은 프레즌스 정보 데이터가 교환되며 본 연구에서는 분산 프레즌스 서버 구조에 맞추어 프레즌스 정보 데이터 포맷이 확장 설계되었다.

먼저 최상위 엘리먼트(root element)인 <conference-info>의 하위 엘리먼트로 각 프레즌스 서버가 현재 담당하고 있는 사용자 리스트인 <usrlist> 엘리먼트가 추가되었는데 이 엘리먼트는 기본 프레즌스 정보 데이터 포맷과는 달리 각 프레즌스

서버를 구분하기 위한 속성(attribute) 'svrlbl'를 가진다. 다음 동작 중인 각 프레즌스 서버의 현재 부하 레벨을 나타내기 위하여 <curr-pressvr-ldlvl> 이 추가되었다. 엘리먼트 <curr-pressvr-ldlvl> 의 속성으로는 각 서버를 구분하기 위한 'svrlbl'을 두었다. 다음 시스템 최대 부하 허용레벨과 최소 부하 허용레벨을 나타내는 <ldlvlmax>과 <ldlvlmin> 을 설계하였다. 프레즌스 서버를 시스템에 추가하는 기준으로 최대 부하 허용레벨이 사용되고 서버의 부하가 너무 낮은 경우 동작 중인 시스템에서 제외시켜 대기 상태로 두기 위한 용도로 최소 부하 허용레벨이 사용된다. 또 현재 대기 중에 있는 프레즌스 서버들의 URL을 나타내는 <rsrvd-svruri>를 두어 필요시 추가 될 수 있도록 하였다. 다음 현재 동작 중인 프레즌스 서버의 숫자를 나타내기 위하여 <curr-svrs> 가 설계되었다. 그림 5에 이와 같이 설계된 프레즌스 시스템에서의 제어 메시지 교환 절차를 보였다.

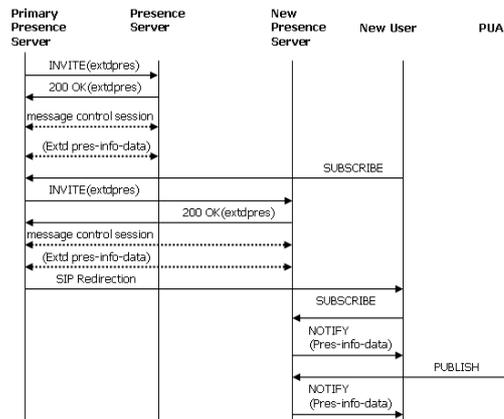


그림 5. 분산 프레즌스 시스템에서의 제어 메시지 교환 절차  
 Fig. 5. Exchange Procedure of Control channel Messages for Presence System

여기에서 먼저 주 프레즌스 서버는 동작중인 다른 프레즌스 서버들에게 컨텐츠 타입이 'extdpres' 인 SIP INVITE 메시지를 전송한다. 상대방 프레즌스 서버는 자신의 제어 세션 ID 값을 SIP 메시지에 넣어 응답하여 서버 사이에 메시지 제어 세션이 설정된다. 다음 시스템 전체 정보를 담은 확장 프레즌스 정보 데이

터가 이 제어 세션을 통하여 전송된다. 새로운 사용자가 프레즌스 리소스를 담은 SIP SUBSCRIBE 메시지를 주 프레즌스 서버에게 보내면 주 프레즌스 서버는 동작 중인 서버들의 부하 레벨을 조사한 후 새로운 프레즌스 서버의 추가가 필요하다고 판단하여 이 서버와 메시지 제어 세션을 설정한다. 다음 이 서버의 URL를 담은 SIP Redirection 메시지를 사용자에게 보내고 해당 사용자는 이 서버에게 프레즌스 리소스 등록을 한다. 다음 등록된 프레즌스 정보의 변화가 발생하면 PUA(presence user agent)는 SIP PUBLISH 메시지를 프레즌스 서버에게 전송하고 서버는 등록된 사용자에게 변화된 프레즌스 정보를 통지한다.

#### 4. 성능 분석

본 논문에서 제안한 분산 프레즌스 서비스 시스템의 성능을 실험으로 분석하였다. 시뮬레이션 환경에서 사용자 수를 증가시켜가며 프레즌스 리소스 등록 처리 시간을 본 연구에서 제안한 방식과 기존 단일 서버 방식과 비교 측정하였다. 이때 본 연구에서 제안한 방식의 경우 최대 부하 허용값을 나타내는 <ldlvlmax> 엘리먼트 값을 사용자 수 및 사용자가 등록된 프레즌스 리소스에서 발생하는 SIP 메시지 양을 사용하여 500으로 설정하였고 그림 6에 이의 결과를 보였다.

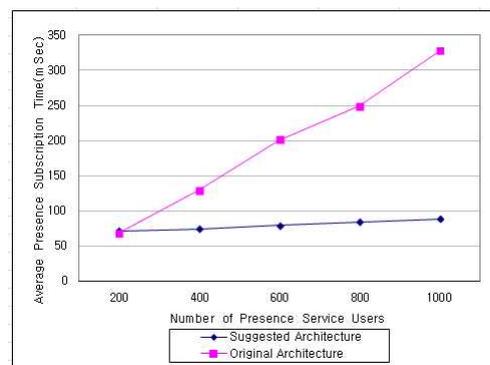


그림 6. 평균 프레즌스 리소스 등록 처리시간(최대부하레벨=500)  
 Fig. 6. Average Presence Resource Subscription Processing Time(Max. Load Level=500)

여기서 각 사용자는 30개의 프레즌스 리소스를 가

지며 1초에 1회의 프레즌스 등록 요청을 하는 것으로 설정하였고 사용자수 증가로 프레즌스 서버의 부하 레벨이 최대 부하 허용값을 초과하면 서버가 새롭게 추가된다. 제안된 방식의 경우 사용자 수 200명 일 때 까지 하나의 서버만을 사용하므로 기존 방식과 동일한 성능을 보이지만 200명을 초과하는 경우 서버가 추가 되어 기존 방식과 비교해 처리 시간이 크게 감소함을 보여준다. 400명인 경우 기존 방식과 비교해 평균 처리 시간이 42.6% 감소하고, 600명 일 때는 66.3% , 800명에서 63.6%, 1000명 일 때는 73.2%가 감소함을 보여준다. 따라서 본 논문에서 제안한 방식의 개선 효과는 사용자가 수가 증가할수록 커짐을 알 수 있다.

다음 그림 7은 다른 조건은 앞의 경우와 동일하게 하고 <math>\langle \text{ldvlmax} \rangle</math> 엘리먼트의 값만 1000으로 변경하여 사용자 수 증가에 따른 평균 등록 처리 시간을 측정 한 결과이다. 여기서는 <math>\langle \text{ldvlmax} \rangle</math> 값의 증가로 그림 6 경우와는 달리 사용자 수가 400명 일 때까지는 1개의 서버만을 사용하다가 이를 초과하면 프레즌스 서버가 새롭게 추가된다. 다음 800명까지는 서버 추가가 없다가 800명을 초과하면 다시 서버가 추가된다. 측정 결과 600명에서 46.8%, 800명에서 46.2%, 1000명에서 65.2% 만큼 감소하여 역시 처리 시간이 기존 방식과 비교해 크게 감소함을 보여준다.

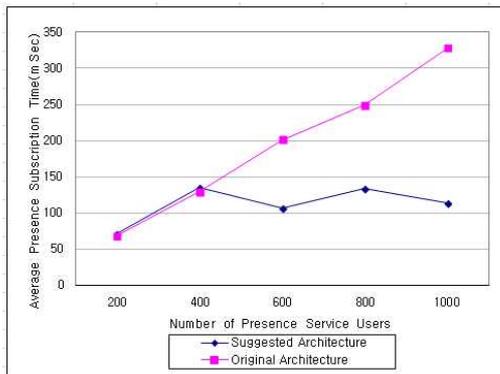


그림 7. 평균 프레즌스 리소스 등록 처리시간(최대부하레벨=1000)  
Fig. 7. Average Presence Resource Subscription Processing Time(Max. Load Level=1000)

다음으로 프레즌스 서버에서 통지 메시지를 처리하

는데 걸리는 시간을 기존 방식과 서로 비교하였다. 이때 프레즌스 통지 메시지의 크기는 30Kbyte로 하였고 각 사용자 당 30개씩의 프레즌스 리소스에서 초당 1회의 프레즌스 정보가 생성되도록 하였다. 그림 8에 측정 결과를 보였다. 본 논문에서 제안된 방식의 경우 사용자 수가 200명을 초과하면 프레즌스 서버가 추가 되어 부하를 분산시키는 효과를 확인 할 수 있고 기존 방식과 비교하여 사용자 수 400명에서 37.6%, 600명에서 47.9%, 800명에서 63.6%, 1000명에서 64.8% 까지 통지 메시지 처리시간이 감소되는 것을 보여준다.

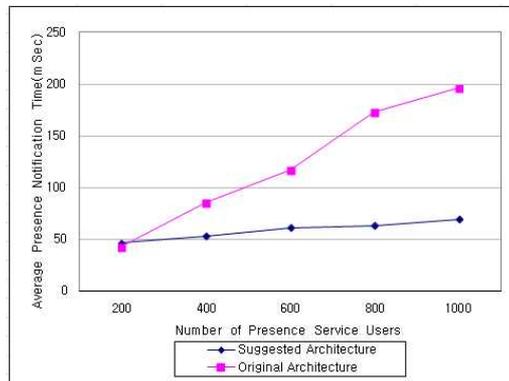


그림 8. 평균 프레즌스 통지 처리시간  
Fig. 8. Average Presence Notification Processing Time

### 5. 결론

본 논문에서는 SIP 메시지 제어 세션을 프레즌스 서버 구조에 적용하여 각 서버의 부하를 동적으로 제어하고 서버의 부하 레벨이 증가하면 서버가 자동으로 추가될 수 있는 분산 프레즌스 서비스 시스템 구조를 연구하였다. 시스템 동작에서 각 서버의 부하 레벨은 고르게 유지되도록 분산 처리되며 사용자 수가 일정 범위를 초과하여 동작 중인 모든 프레즌스 서버들의 부하 레벨이 지정된 값을 초과할 경우 서버 풀에서 동적으로 새로운 프레즌스 서버가 추가된다. 이때 프레즌스 정보 데이터 포맷도 메시지 제어 세션 구조에 맞게 설계되었다.

실험 결과 본 연구에서 제안한 구조의 경우 평균 프레즌스 등록 처리 시간이 42.6%에서 73.6%까지 감소

함을 보여주었고 평균 프레즌스 통지 처리 시간은 37.6%에서 64.8%까지 감소함을 보여주었다. 향후 과제로는 프레즌스 서비스 시스템에서 보다 효율적인 프레즌스 리소스 데이터 구조와 프레즌스 통지 메시지 데이터 구조에 대하여 연구 할 필요가 있다.

## REFERENCES

- [1] A. Houri and J. Hildebrand. "Interworking between the SIP and the Extensible Messaging and Presence Protocol: Presence". IETF RFC 7248, May, 2014.
- [2] A. Houri, E. Aoki and S. Parameswar "Presence and Instant Messaging Peering Use Cases," IETF RFC 5344, October, 2008.
- [3] R. Acker, S. Brandt, S Buchmann and T. Fugmann "Ubiquitous home control based on SIP and presence service," Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, p p.274-278, September, 2010.
- [4] H. Sugano and G. Klyne, "Subscriptions to Request-Contained Resource Lists in the Session Initiation Protocol (SIP)". IETF RFC 5367, October, 2008.
- [5] H. Sugano and G. Klyne, "Presence Information Data Format". IETF RFC 3863, August, 2004.
- [6] M. Thomson and J. Winterbottom, "Representation of Uncertainty and Confidence in the Presence Information Data Format Location Object". IETF RFC 7459. February 2015.
- [7] M. Lonnfors, E. Leppanen, H. Khartabil and J. Urpalainen, "SIP Extension for Partial Notification of Presence Information" IETF RFC 6261, September, 2011.
- [8] A. Niemi, M. Lonnfors and E. Leppanen. "Publication of Partial Presence Information". IETF RFC 5264, September, 2008.
- [9] P. Saint-Andre, A. Houri and J. Hildebrand. "Interworking between the Session Initiation Protocol (SIP) and XMPP: Presence". IETF RFC 7248, May, 2014.
- [10] C. Jang, "XMPP/SIP Presence Service System using Efficient Message Control Method," Journal of Korea Inst. of. Information and Communication Eng., vol. 18, no.11, pp.2678-2684, November 2014
- [11] C. Holmberg. "SDP Offer/Answer Clarifications for RTP/RTCP Multiplexing", IETF RFC 8035, November, 2016.

---

## 저자약력

---

장 춘 서(Choonseo Jang)

[정회원]



- 한국과학기술원 공학박사
- 현재 : 금오공과대학교 컴퓨터공학과 교수

<관심분야>

SIP, 인터넷텔레포니