

SIP 환경에서의 새로운 다중 프레즌스 서버 구조

장 춘 서*

A New Multiple Presence Servers Architecture in SIP Environment

Choonseo Jang*

요 약

SIP(Session Initiation Protocol) 환경에서의 프레즌스 서버는 사용자로부터 다수의 프레즌스 자원의 주소를 담은 SIP SUBSCRIBE 요청 메시지를 처리해야하고 등록된 프레즌스 자원에서 발생하는 다량의 통지 메시지도 처리해야 하므로 사용자 수가 증가 할수록 프레즌스 서버의 부하가 크게 증가하여 확장성의 제한을 받게 된다, 따라서 본 연구에서는 이와 같은 문제점을 해결하기 위하여 사용자 수의 증가에 따라 동적으로 프레즌스 서버가 추가되고 각 프레즌스 서버의 부하를 효율적으로 제어 할 수 있는 다중 프레즌스 서버 구조를 제안하였다. 여기서 각 프레즌스 서버들은 본 논문에서 새롭게 제시한 프레즌스 부하 제어 이벤트 패키지를 사용하여 전체 프레즌스 시스템의 현재 부하 상태를 실시간으로 파악할 수 있다. 특정 프레즌스 서버의 부하가 지정된 한도 이상으로 증가하는 경우 부하 분산을 위하여 전체 프레즌스 시스템에서 가장 부하가 적은 프레즌스 서버를 선택하여 프레즌스 서비스 처리를 분산시키거나 새로운 프레즌스 서버를 동적으로 생성한다. 따라서 이와 같은 구조를 사용하여 프레즌스 시스템의 전체 부하를 적절히 제어 할 수 있으며 확장성도 아울러 높일 수 있게 된다. 이를 위하여 본 논문에서는 프레즌스 부하 제어를 위한 새로운 프레즌스 부하 제어 이벤트 패키지 및 프레즌스 정보 데이터 포맷이 제시되었다. 제안된 시스템의 성능은 실험을 통하여 분석하였고 SIP SUBSCRIBE 메시지 처리시간에서 44.3%, SIP 통지 메시지 처리 시간에서 43.1% 향상됨을 보여주고 있다.

▶ Keywords : 세션시작프로토콜, 프레즌스 서비스, 프레즌스 정보 데이터

Abstract

In SIP(Session Initiation Protocol) environment, the presence server should process SIP SUBSCRIBE request messages including multiple presence resources addresses from users, and

•제1저자 : 장춘서 •교신저자 : 장춘서

•투고일 : 2013. 1. 29, 심사일 : 2013. 3. 19, 게재확정일 : 2013. 4. 9

* 금오공과대학교 컴퓨터공학과(Dept. of Computer Engineering, Kumoh Inst. of Technology)

* 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

also preprocess massive notification messages from the subscribed presence resources. The load of the presence server increases as number of users increase, and it limits system extendability. Therefore a new multiple presence servers architecture has been suggested in this research. In this architecture presence servers can be added dynamically and each server's load can be controlled effectively as number of users increase. Each presence server can monitor current load status of entire presence system by using presence event notification package which newly has been suggested in this paper. When a particular presence server's load increases over predefined limit, the presence service processing is distributed by selecting a server which has the smallest load, or by generating a new server dynamically. In this system the overall load of the entire system can be controlled optimally and extendability of the system can be increased. For this purpose a new presence event notification package and presence information data format have been suggested. The performance of the proposed system has been evaluated by experiments. They shows 44.3% increase in SUBSCRIBE message processing time, and 43.1% increase in Notification message processing time.

▶ Keywords : Session Initiation Protocol, Presence Service, Presence Information Data

I. 서론

SIP[1] 환경에서의 프레즌스(presence) 서비스 [2][3][4]에서 각 사용자들은 프레즌스 정보를 얻기 원하는 다수의 프레즌스 자원에 대한 SIP SUBSCRIBE 요청 메시지를 프레즌스 서버에게 보내야한다. 이후 등록된 프레즌스 자원에서 프레즌스 정보의 상태 변화가 발생할 때마다 서버는 각 사용자들에게 프레즌스 이벤트 통지(notification) 메시지를 보내야 하므로 사용자 수의 증가에 따른 프레즌스 서버의 처리량은 아울러 커지게 된다.

이런 문제를 해결하기 위하여 본 논문에서는 사용자 수의 증가에 따라 동적으로 프레즌스 서버가 추가되고 프레즌스 부하 제어 이벤트 패키지를 사용하여 각 프레즌스 서버의 부하를 효율적으로 제어함으로써 시스템의 확장성을 높일 수 있는 다중 프레즌스 서버 구조를 새롭게 제안하였다.

이와 같은 구조에서 각 프레즌스 서버들은 프레즌스 부하 제어 이벤트 패키지를 사용하여 현재 부하 상태를 주기적으로 교환함으로써 서로 간의 부하 상태를 파악할 수 있다.

동작 중 특정 서버의 사용자 수가 증가하여 해당 서버의 부하가 지정된 한도 이상으로 증가하는 경우 주 서버는 전체 프레즌스 서버 중 가장 부하가 적은 서버를 선택하여 이 서버가 프레즌스 서비스를 분산 처리하도록 한다. 만일 전체 프레

즌스 서버들이 부하를 추가로 분담할 상황이 아닌 경우라면 새로운 프레즌스 서버를 동적으로 생성하여 이 서버가 부하를 분산하도록 한다.

따라서 이와 같은 구조를 사용하여 프레즌스 서비스 시스템의 전체 부하를 적절히 제어 할 수 있으며 확장성도 아울러 높일 수 있게 된다. 이를 위하여 프레즌스 부하 제어를 위한 새로운 프레즌스 부하 제어 이벤트 패키지가 설계되었고 복수의 프레즌스 서버와 사용자 사이의 SIP 메시지 교환 절차 및 프레즌스 부하 분산을 처리하는데 필요한 요소들이 추가된 새로운 프레즌스 정보 데이터 포맷이 제시되었다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 프레즌스 정보 포맷과 프레즌스 서비스 구성 및 기존 연구에 대해 설명하고, III장에서는 본 논문에서 새롭게 제안하는 다중 프레즌스 서버 구조를 구현하기 위한 프레즌스 서버의 설계 및 프레즌스 부하 제어 이벤트 패키지의 구현에 대해 설명한다. 다음 IV장에서는 구현된 시스템의 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

1. 프레즌스 정보 데이터 포맷

프레즌스 정보를 프레즌스 서버와 사용자 사이에 교환하기위

한 데이터의 기본 포맷은 XML 문서 기반의 PIDF(Presence Information Data Format)[5]이고 프레즌스 문서의 MIME 형태는 'application/pidf+xml'로 나타낸다. 이 프레즌스 정보 데이터 포맷의 최상위 요소는 'presence'이고 하위 요소로 하나 이상의 'tuple'이 있으며 각 프레즌스 정보를 구분한다. 여기에 추가적인 정보를 나타내기 위하여 필요한 경우 속성을 가질 수 있다. 각 'tuple'에는 하위 요소로 'status'를 가지며 이는 각 사용자의 현재 상태 정보를 포함하고 있다. 이외에도 여러 개의 추가적인 정보를 나타내는 하위 요소를 가질 수 있다. 그림 1에 이와 같은 프레즌스 정보 데이터 포맷의 예를 보였다.

```
<?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:testsub="http://id.example.com/presence/"
  entity="pres:someone@example.com">
  <tuple id="qre3zhrj9e">
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
      <testsub:location>home</testsub:location>
    </status>
    <contact>sip:student32@irt.kumoh.ac.kr</contact>
    <timestamp>2013-01-03T13:21:33Z
    </timestamp>
  </tuple>
</presence>
```

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

이 그림에서는 다른 'tuple'과 구분 하기위하여 속성으로 'id'가 사용되었고 요소 'status'는 이 사용자의 현재 상태 정보가 'open'임을 나타낸다. 또 추가적인 프레즌스 정보를 나타내기 위하여 문서에 대한 시간 정보 요소인 'timestamp'와 사용자의 위치 정보를 가지고 있는 'contact' 등이 사용되었다. 이와 같은 프레즌스 정보 데이터는 프레즌스 서버와 사용자 사이에 통지 메시지나 SIP PUBLISH 메시지를 통해 서로 교환된다.

2. 프레즌스 서비스 구성 및 기존 연구

그림 2에 기본적인 프레즌스 서비스 구성을 보였다.

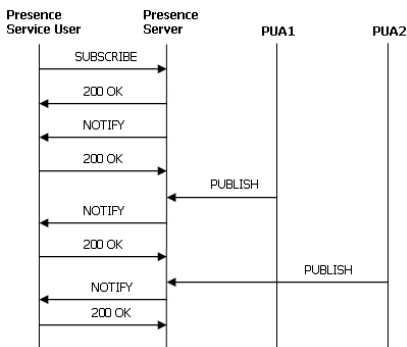


그림 2 프레즌스 서비스 구성
Fig.2 Configuration of Presence Service

그림 2에서 프레즌스 서버는 사용자의 온라인 상태나 사용 가능한 통신 서비스의 종류, 특정 사용자의 접속 주소 등에 관한 각종 정보를 제공하며 사용자는 프레즌스 서버에게 등록할 프레즌스 자원의 SIP 주소를 담은 SUBSCRIBE 메시지를 사용하여 필요한 프레즌스 서비스 등록을 한다. 이때 프레즌스 서버에서는 사용자 인증 과정을 거쳐 해당 프레즌스 자원에 대한 프레즌스 이벤트 등록 작업이 처리된다.

이와 같이 각 사용자에게 대해 등록된 프레즌스 자원을 PUA(Presence User Agents)라 하며 이들은 프레즌스 정보의 변화가 있는 경우 프레즌스 정보를 담은 SIP PUBLISH 메시지(6)[7]를 생성하여 이를 프레즌스 서버에게 전송하고 프레즌스 서버는 SIP NOTIFY 메시지를 사용하여 해당 PUA를 등록한 사용자에게 프레즌스 정보 데이터 포맷에 의해 생성된 정보를 전송한다.

프레즌스 서버는 사용자들이 등록하는 각 프레즌스 자원에 대한 처리, 각 자원의 상태변화에 따른 통지 메시지 처리 그리고 각 자원에 대한 등록을 유지하기위한 리프레시 메시지에 대한 처리가 필요하므로 프레즌스 서비스 시스템의 사용자 수가 많이 질수록 프레즌스 서버의 부하는 크게 증가하게 된다.

프레즌스 서버의 부하를 줄이기 위한 현재까지의 연구는 주로 단일 프레즌스 서버를 사용하고 프레즌스 정보 문서의 크기를 줄이기 위하여 기존에 보낸 프레즌스 정보 문서에서 변경된 부분만 전송하는 부분 통지(partial notification) 방식이나(8)[9] 프레즌스 자원을 등록하는데 걸리는 시간을 줄이기 위한 방식(10)에 집중되고 있다. 그러나 이와 같은 방식으로는 대용량 프레즌스 서비스 시스템으로의 확장이 어렵다는 문제점이 있어 본 연구에서는 사용자 수의 증가에 따라 동적으로 프레즌스 서버가 추가되고 각 프레즌스 서버의 부하를 효율적으로 제어 할 수 있는 다중 프레즌스 서버 구조를 제안하여 대용량 프레즌스 서비스 시스템의 구성이 가능하도록 하였다.

III. 시스템 설계 및 구현

1. 새로운 다중 프레즌스 서버 구조 설계

그림 3에 본 논문에서 설계한 새로운 부하 제어 방식을 사용한 다중 프레즌스 서버 구조를 보였다.

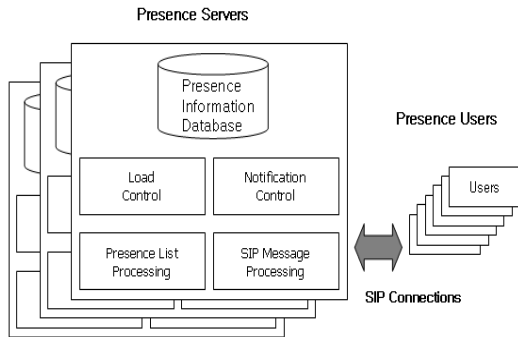


그림 3 설계된 다중 프레즌스 서버 구조
Fig.3 A Designed Architecture of Multiple Presence Server

여기에서 각 프레즌스 서버들은 부하 제어 모듈, 통지 제어 모듈, 프레즌스 리스트 처리 모듈, SIP 메시지 처리 모듈 및 프레즌스 정보 데이터베이스로 구성된다. 여기서 프레즌스 정보 데이터베이스는 기존의 방식과 달리 본 논문에서 새롭게 구현한 부하 제어 기능이 추가된 새로운 프레즌스 정보 데이터 포맷을 사용하여 구현되었다.

각 프레즌스 서버들은 서로 간에 SIP SUBSCRIBE 메시지를 사용하여 본 논문에서 제안한 부하 제어 프레즌스 이벤트 패키지를 등록하고 전체 사용자에게 대한 프레즌스 정보 데이터의 통지 기능도 분담하여 부하를 줄인다.

프레즌스 리스트 처리 모듈은 사용자가 한번의 SUBSCRIBE 메시지로 프레즌스 정보를 얻기 원하는 상대방의 URI로 구성된 프레즌스 리스트를 사용하여 프레즌스 등록을 할 수 있도록 함으로써 SUBSCRIBE 메시지 및 등록 상태 유지를 위한 리프레시 메시지 양을 크게 줄일 수 있도록 한다. 사용자가 주 프레즌스 서버에게 SIP SUBSCRIBE 메시지를 사용하여 프레즌스 정보 등록 요청을 하면 프레즌스 서버는 본 논문에서 제안한 부하 제어 프레즌스 정보 데이터 포맷을 사용해 구현된 데이터베이스를 액세스하여 자신의 현재 부하 상태를 나타내는 'presence-load-index' 값이 사용자 요청을 처리할 수 있는 경우 상대방에게 상태 코드 번호가 200인 SIP 수락 메시지를 보내고 이어서 프레즌스 정보 통지를 위한 SIP NOTIFY 메시지를 보낸다.

만일 현재 부하 상태가 커서 다른 프레즌스 서버에게 사용자 요청을 전달하여야 할 경우에는 상태 코드 번호가 302인 Redirection SIP 메시지를 보낸다. 이 메시지는 다른 프레즌스 서버를 가리키는 URI가 포함되어 있으므로 사용자가 이 프레즌스 서버와 연결을 맺도록 한다.

모든 프레즌스 서버의 부하 상태가 높아 새로운 프레즌스 서버를 추가해야 할 경우에는 프레즌스 정보 데이터베이스에

서 선택 가능한 새로운 프레즌스 서버를 선택하여 이 서버에게 INVITE 메시지를 보낸다. 해당 프레즌스 서버가 응답하면 부하 제어 프레즌스 이벤트 패키지를 사용하여 현 프레즌스 정보 데이터를 보낸다. 이와 같은 프레즌스 서버 동작 과정의 대한 순서도를 그림 4에 보였다.

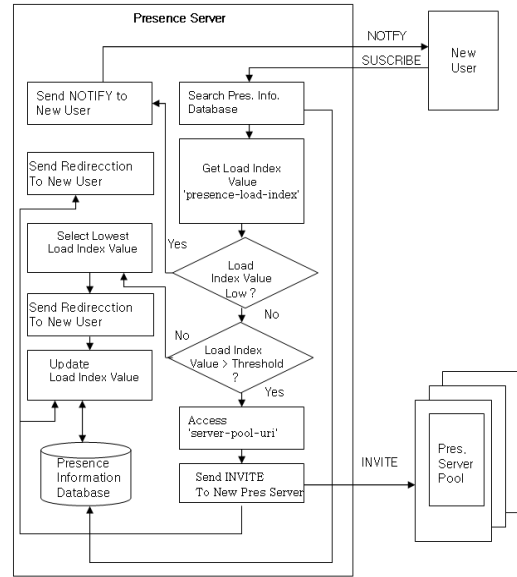


그림 4 설계된 다중 프레즌스 서버 동작 과정 순서도
Fig. 4 Flowchart of Designed Presence Server Operation

그림 4에서 새로운 사용자의 요청을 받은 주 프레즌스 서버는 프레즌스 정보 데이터베이스를 액세스하여 현재 가장 부하 상태가 낮은 프레즌스 서버를 검색한다. 이때 모든 프레즌스 서버의 부하 상태가 높아 새로운 프레즌스 서버의 추가가 필요한 경우 프레즌스 정보 데이터베이스에서 새롭게 추가 될 수 있는 프레즌스 서버들의 URI를 가지고 있는 'added-pres-server-uri' 항목에서 프레즌스 서버를 선택한다.

다음 새롭게 선택된 프레즌스 서버에게 Event 헤더에 'pres-multi' 값을 넣은 INVITE 메시지를 보내어 프레즌스 서비스에 참가하도록 한다. 이 새로운 프레즌스 서버는 주 프레즌스 서버에게 이벤트 패키지 등록을 위한 SUBSCRIBE 메시지를 보내어 이어서 NOTIFY 메시지를 통해 전체 프레즌스 정보 데이터를 받게 된다. 다음 이 프레즌스 서버는 사용자에게 대한 프레즌스 서비스 처리를 분담한다.

2. 프레즌스 부하 제어 이벤트 패키지 및 프레즌스 정보 데이터 포맷 설계

본 연구에서는 새롭게 제안된 프레즌스 부하 제어 이벤트

패키지 방식을 구현하기 위하여 다음과 같은 필요 요소들이 추가된 프레즌스 정보 데이터 포맷이 설계되었다. 먼저 프레즌스 부하 제어를 위하여 최상위 요소 'presence'의 하위 요소로 'presence-load-index'가 새롭게 추가되었다. 요소 'presence-load-index'는 각 프레즌스 서버에서 프레즌스 서비스 처리를 위한 부하 지수를 나타내며 등록된 프레즌스 자원에서 초당 발생하는 SIP 메시지 전송량의 전체 합을 기준으로 산출한다. 속성으로는 최대 부하 허용값을 나타내는 'max-presence-load-index'와 각 프레즌스 서버를 구분하기 위한 'pres-server-id'가 설계되었다.

또 해당 프레즌스 서버가 현재 처리하고 있는 사용자수를 표시하는 'current-users-number' 요소가 추가되었고 현재 프레즌스 서비스 시스템의 상태를 표시하기 위한 요소인 'multiple-presence-state'가 새롭게 추가 되었다. 이 요소의 하위 요소로 현재 전체 프레즌스 서비스 사용자 수를 나타내는 'current-total-pres-users-number'와 프레즌스 사용자들에 대한 개별 정보를 나타내기 위한 요소인 'pres-users-status'가 추가되었다.

요소 'pres-users-status'는 하위 요소로 프레즌스 부하 제어 이벤트 패키지 사용 여부를 나타내기 위한 'pres-load-control-notify'와 프레즌스 부하 제어 이벤트 패키지에 의한 통지 메시지에 메시지 크기를 줄이기 위한 부분 프레즌스 정보 데이터 포맷 가능 여부를 나타내기 위한 'pres-load-control-notify-differential'이 추가되었다.

다음으로 현재 동작 중인 전체 프레즌스 서버의 개수를 나타내는 'current-server-number'와 시스템의 부하에 따라 새롭게 추가 될 수 있는 프레즌스 서버들의 URI를 가지고 있는 'server-pool-uri'가 추가되었다. 또 각 서버가 현재 담당하고 있는 사용자 리스트인 'current-pres-users-list'가 추가되었고 속성으로 각 프레즌스 서버를 구분하기 위한 'pres-server-entity'을 설계하였다. 그림 5에 이와 같은 프레즌스 부하 제어를 처리하기 위한 필요 요소들이 새롭게 추가된 프레즌스 정보 데이터 포맷을 사용한 SIP 메시지 교환 절차를 보였다.

여기서는 먼저 각 프레즌스 서버들이 SUBSCRIBE 메시지를 사용하여 본 논문에서 제안한 부하 제어 이벤트 패키지를 등록하고 NOTIFY 메시지를 통하여 현 프레즌스 정보 데이터를 동기화 시킨다. 새로운 프레즌스 서비스 사용자가 주 프레즌스 서버에 프레즌스 등록 메시지를 보내면 프레즌스 서버는 부하 상태를 조사하기 위하여 프레즌스 정보 데이터베이스를 액세스하고 만일 새로운 프레즌스 서버를 추가해야 할 경우에는 이 서버에게 Evnet 헤더에 'multi-pres' 값을 넣은

INVITE 메시지를 보낸다. 이 새로운 프레즌스 서버는 SUBSCRIBE 메시지를 주 프레즌스 서버에게 보내어 NOTIFY 메시지를 통해 현 프레즌스 정보 데이터를 얻는다.

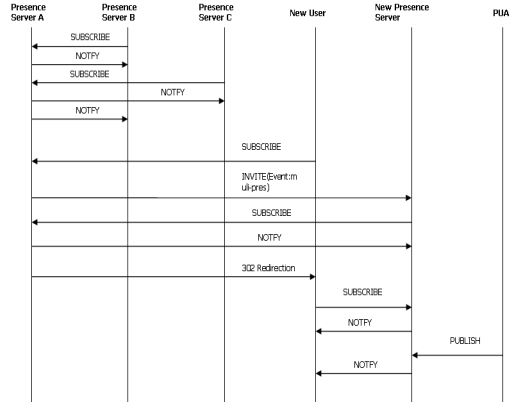


그림 5 SIP 메시지 및 프레즌스 정보 교환 절차
Fig. 5 Exchange Procedure of SIP Messages and Conference Information

다음 새로운 프레즌스 서비스 사용자에게 새로운 프레즌스 서버를 가리키는 응답 메시지인 Redirection SIP 메시지를 보내어 사용자가 이 프레즌스 서버와 서비스 연결을 맺도록 한다. 새로운 사용자가 등록한 PUA에서 PUBLISH 메시지가 발생하면 이 사용자를 담당하는 새로운 프레즌스 서버가 이에 대한 처리를 하여 해당 사용자에게 프레즌스 통지 메시지를 보내게 된다.

IV. 성능 분석

본 논문에서 제안한 다중 프레즌스 서버 구조의 성능 분석을 위하여 운영체제로 커널 버전 2.6인 리눅스를 가진 두 대의 PC를 프레즌스 서버로 사용하였다. 일반 프레즌스 서비스 사용자용으로는 15대의 PC가 사용되었고 여기에 설치된 운영체제는 마이크로소프트 Windows 7이다. 각 PC의 네트워크 인터페이스 및 허브를 포함한 LAN 환경의 속도는 100Mbps이고 모든 PC는 하나의 LAN 세그먼트로 구성된 서브넷 상에 배치되었다.

본 논문에서 제안한 프레즌스 부하 제어 방식을 사용 한 다중 프레즌스 서버 구조와 기존 방식과의 처리시간을 비교 측정하였다. 이를 위하여 각 사용자당 100개씩의 프레즌스 자원을 등록하도록 하고 프레즌스 서버에서 이들 SUBSCRIBE 메시지를 모두 처리하는데 걸리는 시간을 비

교하였다. 본 실험에서는 물리적으로 PC 개수를 늘리는 것은 한계가 있으므로 각 사용자 당 프레즌스 자원의 수를 크게 하여 대용량 프레즌스 시스템의 효과를 내었다. 이때 제안된 방식의 경우 프레즌스 부하 제어 정책으로 최대 부하 허용값을 나타내는 'presence-load-index' 값을 100으로 하여 처음에 하나의 프레즌스 서버로 동작하다가 사용자가 증가하여 동작 중 최대 부하 허용값이 이를 초과하면 새로운 프레즌스 서버가 추가되도록 하였다. 이때의 측정 결과를 그림 6에 보였다.

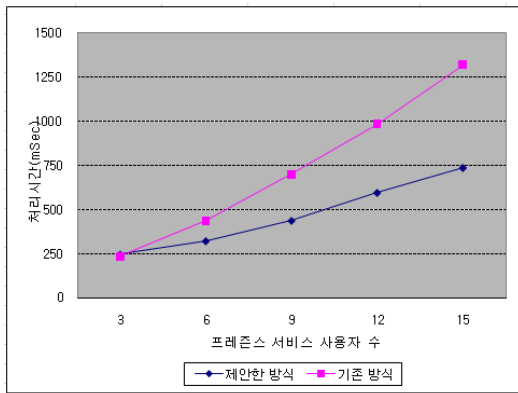


그림 6 SUBSCRIBE 메시지 프레즌스 서버 처리 시간 비교(최대 부하 허용값=100)

Fig. 6 Comparison of Presence Server Processing Time for SUBSCRIBE Message(Max. Load Allowance=100)

여기서 프레즌스 참가자 수가 3명일 때는 제안된 방식에서도 1개의 프레즌스 서버만을 사용하므로 기존 방식과 같은 처리 시간을 보이지만 6명 일 때부터는 프레즌스 부하 제어 방식을 사용한 다중 프레즌스 구조에 의해 프레즌스 서버가 새롭게 추가되어 부하를 분담하므로 큰 성능의 향상을 보여준다. 프레즌스 참가자 수가 6명 일 때는 제안된 분산형 프레즌스 구조에서의 처리 시간이 26.7% 감소하고 참가자 수가 9명 일 때는 37.5% 감소하며 참가자 수가 15명 일 때는 44.3% 감소됨을 보여주고 있다. 이와 같이 본 논문에서 제안된 다중 프레즌스 서버 구조에서의 처리 시간의 개선 효과는 프레즌스 사용자가 수가 증가할수록 높아짐을 알 수 있다.

그림 7은 프레즌스 부하 제어 정책으로 최대 부하 허용값을 200으로 하여 측정한 결과이다. 여기서는 프레즌스 참가자 수가 6명 일 때까지는 1개의 프레즌스 서버만을 사용하다가 9명 일 때부터 부하 제어 방식을 사용한 경우이다. 따라서 사용자 수 9명 이상부터 프레즌스 서버가 새롭게 추가되어 처리 시간이 크게 감소함을 알 수 있다. 이와 같이 본 연구에서는 프레즌스 정책으로 최대 부하 허용값을 상황에 따라 조정 할 수 있

으므로 프레즌스 서비스 시스템의 확장성을 높일 수 있다.

그림 8은 프레즌스 서버에서 통지 메시지를 처리하는데 걸리는 시간을 기존 방식과 서로 비교한 결과이다. 이때 50Kbyte 크기의 Pidf 포맷의 프레즌스 문서를 프레즌스 정보로 사용하였고 각 사용자 당 100개씩의 프레즌스 자원에서 프레즌스 정보가 생성되도록 하였으며 최대 부하 허용값 'presence-load-index'는 100으로 하여 실험하였다.

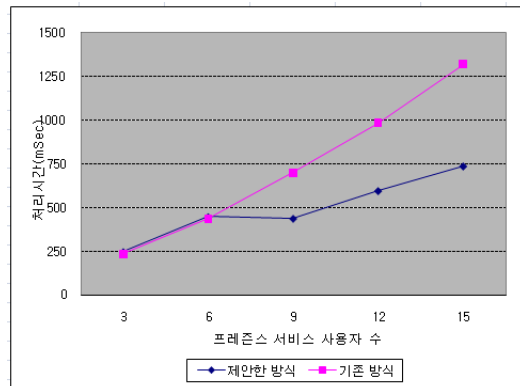


그림 7 SUBSCRIBE 메시지의 프레즌스 서버 처리 시간 비교(최대 부하 허용값=200)

Fig. 7 Comparison of Presence Server Processing Time for SUBSCRIBE Message(Max. Load Allowance=200)

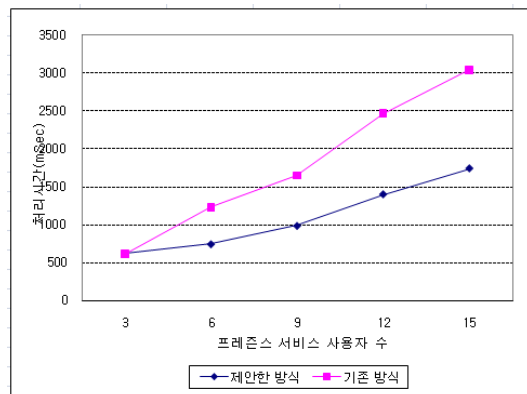


그림 8 프레즌스 서버의 통지 메시지 처리 시간 비교

Fig. 8 Comparison of Notification Messages Processing Time in Presence Server

사용자 수 6명 이상부터 프레즌스 부하 제어 정책에 의한 프레즌스 서버 추가로 처리 시간이 크게 감소하는 것을 보여준다. 여기서는 본 논문에서 제안한 방식이 기존의 방식에 비해 프레즌스 서버에서의 메시지 처리 시간을 사용자 수 6명에서 38.8%, 사용자 수 15명에서 43.1% 까지 단축시킴을 보

여준다. 따라서 본 논문에서 제안한 방식이 프레즌스 서비스 사용자 수가 증가함에 따라 프레즌스 서버에서의 통지 메시지 처리 시간 감소 능력이 향상됨을 보여주고 있다.

V. 결론 및 향후 과제

본 논문에서는 SIP 환경에서 사용자 수의 증가에 따라 동적으로 프레즌스 서버가 추가되고 각 프레즌스 서버의 부하를 효율적으로 제어 할 수 있는 다중 프레즌스 서버 구조를 제안하였다. 여기서 각 프레즌스 서버들은 본 연구에서 새롭게 제시한 프레즌스 부하 제어 이벤트 패키지를 사용하여 전체 시스템의 현재 부하 상태를 실시간으로 파악할 수 있어 특정 프레즌스 서버의 부하가 한도 이상으로 증가하면 전체 프레즌스 시스템에서 가장 부하가 적은 프레즌스 서버를 선택하여 프레즌스 서비스 처리를 분산시키거나 새로운 프레즌스 서버를 동적으로 생성하도록 한다. 이를 위하여 본 논문에서는 프레즌스 부하 제어를 위한 새로운 프레즌스 부하 제어 이벤트 패키지 및 프레즌스 정보 데이터 포맷이 제시되었다. 본 연구에서 제시된 구조를 사용하여 프레즌스 시스템의 전체 부하를 적절히 제어 할 수 있으며 확장성도 아울러 높일 수 있다.

제안된 시스템의 성능 분석을 위하여 복수개의 프레즌스 서버를 갖춘 환경에서 프레즌스 서비스 사용자 수를 증가시켜 가면서 서버에서의 등록 및 통지 메시지 처리 시간을 측정하였다. 측정 결과 본 연구에서 제안한 다중 프레즌스 서버 구조에서 등록 메시지의 경우 26.7%에서 44.3%까지의 처리시간의 개선 효과가 있으며 통지 메시지 처리 시간의 경우 38.8%에서 43.1%까지 감소시킬 수 있음을 보였다. 향후 과제로는 프레즌스 서비스 사용자 수 및 PC를 더욱 늘려 대규모 프레즌스 서비스 시스템 환경을 구성하고 무선 모바일 환경에서 본 논문이 제안한 구조가 얼마만큼 개선 효과를 얻는가를 측정하고 이에 맞춘 개선된 방식을 연구 할 필요가 있다.

참고문헌

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "Session Initiation Protocol," RFC 3261, June 2002.
- [2] P. Saint-Andre, "Extensible Messaging and Presence Protocol", RFC 6502, March. 2011.
- [3] J. Rosenberg, "A Watcher Information Event

Template-Package for the Session Initiation Protocol", RFC 6120, August 2007

- [4] A. Houri, E. Aoki, "Presence & Instant Messaging Peering Use Cases", RFC 5344, June 2008
- [5] H. Sugano, G. Klyne, "Presence Information Data Format (PIDF)," RFC 3863, August 2004.
- [6] A. Niemi, Ed., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.
- [7] A. Niemi, M. Lonnfors, E. Leppanen, "Publication of Partial Presence Information", RFC 5264, September 2008.
- [8] M. Lonnfors, E. Leppanen, H. Khartabil, J. Urpalainen, "Presence Information Data format (PIDF) Extension for Partial Presence", Internet-Draft, Nov. 2010.
- [9] M. Lonnfors, E. Leppanen, H. Khartabil, J. Urpalainen, "SIP Extension for Partial Notification of Presence Information", Internet-Draft, September 2011.
- [10] C. S. Jang, "A New Presence Lists Service in SIP Environment" Journal of The Korea Society of Computer and Information, Vol 15, No.12, pp 67-73, December 2010.

저 자 소 개



장 춘 서

1993년 2월 : 한국과학기술원 공학박사
 1981년 3월~현재 : 금오공과대학교
 컴퓨터공학과 교수
 <관심분야> : SIP, 임베디드 시스템,
 인터넷텔레포니
 Email: csjang@kumoh.ac.kr