

새로운 동적 통지 제어 방식을 사용한 리소스 리스트 구조

장춘서*, 이기수*,

A New Resource Lists Architecture using Dynamic Notification Control Method

Choonseo Jang*, Ky-Soo Lee*

요 약

SIP(Session Initiation Protocol) 서비스에서 사용자는 상대방들에 대한 등록 정보를 얻기 위하여 각각의 SIP 등록 요청 메시지를 SIP 서버에게 보내야 하고 이때 각 사용자 등록에 대해서 리프레시 메시지 및 통지 메시지가 발생하므로 네트워크상에서의 전체적인 SIP 메시지 양이 크게 발생하는 문제가 있다. 본 논문에서는 이를 해결하기 위하여 새로운 동적 통지 제어 방식을 사용한 리소스 리스트 구조를 제안하였다. 여기서는 사용자가 시스템 동작 상황에 맞추어 SIP 서버로부터 이벤트 통지 메시지 발생률을 동적으로 직접 제어 할 수 있으며, SIP 세션 유지를 위한 리프레시 동작 과정에서 발생하는 SIP 메시지 및 세션 연결 상황에 따른 각종 통지 메시지 발생도 동적으로 제어 할 수 있도록 하였다. 이를 위하여 새로운 리소스 리스트 MIME 타입을 설계하였고 SIP 메시지 헤더가 확장되어 동적 통지 메시지 제어에 필요한 여러 파라미터가 새롭게 추가되었으며 이벤트 패키지 처리에 새로운 리소스 리스트 MIME 타입이 사용될 수 있도록 하였다. 이와 같이하여 네트워크상에서의 전체적인 SIP 메시지 양을 크게 줄일 수 있고 리소스 리스트 서버에서의 메시지 처리를 위한 부하도 아울러 감소시킬 수 있다. 제안된 방식의 성능은 실험을 통하여 분석하였다.

▶ Keyword : 세션시작프로토콜, 이벤트 패키지, 리소스 리스트

Abstract

In SIP(Session Initiation Protocol) service, users should send SIP SUBSCRIBE messages to the SIP server to get subscription informations of the other parties, and total amount of SIP messages

• 제1저자: 장춘서, 교신저자: 이기수

• 투고일 : 2011. 09. 27, 심사일 : 2011. 10. 13, 게재확정일 : 2011. 10. 27

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

* 본 연구는 금오공과대학교 교수연구년제에 의하여 연구된 논문입니다.

becomes large in network as refresh messages and notification messages are generated to each SIP subscription. In this paper, a new resource lists architecture using dynamic notification control method has been proposed to solve these problems. In this architecture, service users can control event notification generation rate dynamically from the server according to the operation circumstances, and users can control generation of notification messages during refresh for maintaining SIP session. Furthermore users can also control generation of notification messages according to session connection conditions. A new resource lists MIME type has been designed, and several parameters for dynamic notification message control have been added to the extended SIP message headers. In this system new resource lists MIME type can be used for processing of the event package. Therefore total amount of SIP messages in network can be largely reduced, and load of the resource lists server for processing messages can be reduced also. The performance of the proposed method has been evaluated by experiments.

▶ Keyword : Session Initiation Protocol, Event Package, Resource Lists

I. 서론

SIP(Session Initiation Protocol) 서비스[1][2][3]에서 사용자는 상대방의 SIP 정보의 상태 변화를 알 수 있도록 하기위하여 이벤트 패키지를 SIP 서버에게 등록하며 각 SIP 등록에 대해서는 주기적으로 리프레시 메시지가 필요하고 등록된 상대방에 대해서는 지속적으로 SIP 통지(presence notification) 메시지가 발생한다.[4][5] 이때 등록된 상대방이 많을수록 시스템 전체의 SIP 메시지 발생량은 매우 많아지며 이는 네트워크 트래픽을 크게 하여 SIP 서버 및 사용자 시스템의 부하를 증가시킨다. 특히 이와 같은 상황은 통신 대역폭의 제한을 받는 모바일 디바이스 경우나 무선 네트워크 환경에서 SIP 서비스의 원활한 사용을 어렵게 만든다.

본 논문에서는 이와 같은 문제점을 해결하기위하여 새로운 동적 통지 제어 방식에 의한 리소스 리스트 구조를 제안하였다. 동적 통지 제어란 시스템 동작 상황에 맞추어 리소스 리스트 서버로 부터의 이벤트 통지 메시지 발생률을 동적으로 직접 제어 할 수 있는 기능이며 SIP 세션 유지를 위한 리프레시 과정에서 발생하는 SIP 메시지나 세션 연결 상황에 따른 각종 통지 메시지 발생 여부도 역시 능동적으로 제어 할 수 있도록 하였다.

본 연구에서는 이를 위하여 새로운 리소스 리스트 MIME 타입이 제안되었고 SIP 메시지 헤더를 확장하여 이벤트 통지 메시지 발생률 제어를 위한 새로운 파라미터들이

추가되었다. 리소스 리스트 서버는 이들 파라미터에 의해 통지 메시지 발생률을 조절한다. 새롭게 설계된 리소스 리스트 MIME 타입을 사용하여 프레즌스 서비스를 구현하였으며 이는 컨퍼런스 서비스나 보이스메일 등에도 폭 넓게 적용 할 수 있다.

본 논문에서는 이와 같은 방식을 사용하여 전체적인 SIP 메시지 양을 크게 줄일 수 있도록 하였으며 네트워크 트래픽의 감소 및 리소스 리스트 서버에서의 메시지 처리를 위한 부하도 아울러 감소시킬 수 있도록 하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 프레즌스 정보 데이터 포맷 및 SIP 통지 메시지에 대해 설명한다. III장에서는 본 논문에서 새롭게 제안하는 리소스 리스트 구조를 위하여 리소스 리스트 MIME 타입 및 이를 실제로 적용한 프레즌스 서버를 설계하고 이벤트 통지 메시지 제어기능을 구현한다. IV장에서는 구현된 본 시스템의 성능 분석을 다양한 실험 조건에서 행하며 마지막으로 V장에서 결론을 맺는다.

II. 관련연구

1. 프레즌스 정보 데이터 포맷

프레즌스 정보 데이터 포맷은 PIDF(Presence Information Data Format) 문서[6][7]에 규정되어 있고 이는 MIME 형태가 application/pdf+xml인 XML 문서이다. 여기서는 최상위 요소(root element)로 <presence>가

있다. 이 요소는 속성으로 프레즌스 정보 소스를 나타내는 URI인 'entity'를 가져야 하고 이름공간(name space) 선언을 포함해야 한다. 이 요소의 하위 요소로는 <tuple>이 있으며 이는 각각의 프레즌스 정보를 구분한다. 각 <tuple>은 속성 'id'를 가지며 하위요소로 <status>를 반드시 가져야 하고 이는 프레즌스 정보 데이터가 가리키는 상대방의 현재 상태 정보를 나타낸다. 요소 <tuple>은 그 외에 <contact>나 <timestamp> 등의 여러 추가적인 하위요소를 가질 수 있다. 이들 추가적인 요소들 중에는 다른 이름공간에서 정의된 요소들도 가능하다. 요소 <status>는 하위요소로 <basic> 및 추가적인 요소들을 가질 수 있다.

그림 1에 프레즌스 정보 데이터 문서의 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"
  entity="pres:kim@pres.kumoh.ac.kr">
  <tuple id="t6r8hj">
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
    </status>
    <contact>sip:pc26@sip.kumoh.ac.kr</contact>
    <timestamp>2011-05-11T19:21:31Z
    </timestamp>
  </presence>
  
```

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

이 문서에서는 <tuple>의 하위요소인 <status>에서 요소 <basic>을 사용하여 현재 상태가 'open'임을 나타내고 있다. 또 <status>의 하위요소로 다른 이름 공간에서 정의된 요소인 <im:im>이 사용되었다. 그밖에 추가적인 정보를 나타내기 위한 하위요소로 해당 사용자의 위치 정보를 가지고 있는 <contact>와 문서에 대한 시간 정보 요소인 <timestamp> 등이 사용되었다. 최근에는 이와 같은 프레즌

스 정보 데이터포맷을 응용 분야에 따라 다양하게 확장하는 연구가 발표되고 있다. 이 중에는 인스턴트 메시징 시스템에서 프레즌스 정보 데이터 포맷을 확장하는 연구[8]와 부분 프레즌스 정보 포맷을 사용하여 네트워크 트래픽을 줄이기 위한 연구[9]가 있다.

2. SIP 통지 메시지

프레즌스 서버에서의 SIP 통지 메시지는 그림 2와 같이 사용자가 SUBSCRIBE 메시지를 보냈을 때와 PUA(Presence User Agents)에서 PUBLISH 메시지[10]를 보내올 때 생성된다. SUBSCRIBE 메시지는 사용자가 프레즌스 정보를 얻기 원하는 상대방의 SIP URI를 등록 할 때 및 프레즌스 이벤트 등록 시간의 연장을 위한 리프레시 시와 세션 종료 시에 발생한다. 프레즌스 이벤트 등록을 유지하기 위해서는 유효기간 내에 지속적으로 프레즌스 서버에게 리프레시 메시지를 보내야한다. PUBLISH 메시지는 PUA에서 프레즌스 정보의 변동이 발생했을 때 해당 PUA가 이를 서버에게 알리기 위해 보내온다.

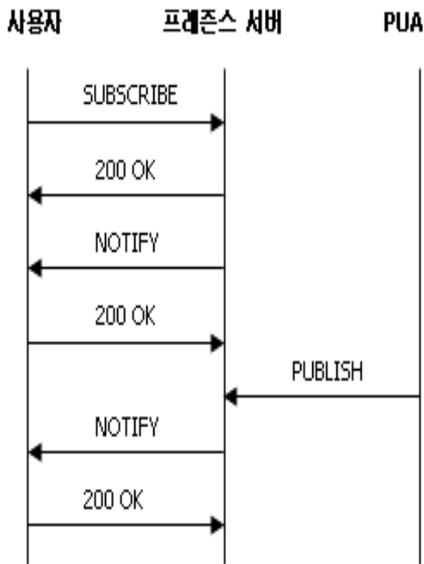


그림 2 SIP 통지 메시지
Fig. 2 SIP Notification Message

프레즌스 서버는 프레즌스 정보의 변화가 발생 할 때 마다 프레즌스 이벤트를 등록한 해당 사용자에게 이 데이터를 담은 SIP NOTIFY 메시지를 보내며 PUA에서 프레즌스 서버에게 SIP PUBLISH 메시지를 보낼 때에도 프레즌스 정보 데이터가 사용된다.

최근에는 통지 메시지에 포함되는 프레즌스 정보의 크기를 줄여 네트워크 트래픽 및 시스템 부하를 감소시키기 위한 연구가 시도되고 있으며 이중에는 프레즌스 정보 데이터 문서에서 이전에 보낸 내용과 비교하여 변경된 부분만 전송하는 부분 통지(partial notification) 방식[11]에 대한 연구도 시도되고 있다. 그러나 프레즌스 서비스에서는 프레즌스 이벤트 등록을 위한 리소스의 수가 많아질수록 각 리소스들에 대한 등록 메시지 및 리프레시 메시지와 각 리소스에서의 통지 메시지들이 생성되므로 전체적인 SIP 메시지 발생량은 상당히 커지게 된다. 따라서 이런 방식들은 대량의 리소스에서 발생하는 네트워크 트래픽을 줄이기에는 한계가 있으며 본 논문에서는 이런 문제점들을 해결하기 위하여 새로운 동적 통지 제어 방식을 사용한 리소스 리스트 구조를 제안하였다.

III. 시스템 설계 및 구현

1. 리소스 리스트 MIME 타입 설계

본 연구에서는 등록하고자하는 개별 리소스를 나타내는 SIP URI들로 구성된 XML 문서 포맷을 가지는 application/rsc-pres-list+xml 리소스 리스트 MIME 타입을 새롭게 설계하였다. 이 문서 포맷은 리소스 리스트에 대한 메타 정보(meta information)를 가지는 첫 번째 부분과 실제 프레즌스 데이터 정보를 가지는 나머지 부분들로 구성된다. 리소스 리스트 메타 정보 부분의 최상위 원소는 <rsc-pres-list>이고 속성 'rsclist-uri'가 리소스 리스트의 URI를 나타낸다. 원소 <entity-pr>은 <rsc-pres-list>의 하위원소로써 복수 개가 올 수 있으며 각각이 각 리소스의 프레즌스 데이터 정보를 나타낸다. 이 원소의 속성 'uri'는 실제 프레즌스 데이터 정보를 가지는 부분에서의 <presence> 원소의 속성인 'entity'를 가리킨다. 원소 <entity-pr>의 하위원소인 <entity-pr-case>는 실제 프레즌스 리소스의 상태를 나타낸다. 이 원소의 속성 'status'는 해당 프레즌스 리소스에 대한 등록 상태를 나타내며 유효한 상태일 때 'valid' 값을 가진다. 또 속성 'rsc-cont-id'는 실제 프레즌스 데이터 정보를 가지는 부분에서 각 리소스의 위치를 나타낸다. 원소 <uid>는 리소스 이벤트에 등록된 상대방을 나타낸다.

이와 같이 설계된 리소스 리스트 메타 정보의 예를 그림 3에 보였고 여기서는 세 개의 <entity-pr> 원소를 사용하여 세 개의 리소스를 나타내었다. 이 중 두개는 <entity-pr-ca

se> 원소를 사용하여 현 서버에서 관리하는 프레즌스 리소스의 상태를 나타내었고 나머지 하나는 다른 도메인에서 관리하는 리소스이다.

```
<?xml version="1.0"?>
<rsc-pres-list
  xmlns="urn:ietf:params:xml:ns:rsclistmeta"
  rsclist-uri="sip:pr_list@sip.kumoh.ac.kr">
  <entity-pr
    uri="sip:jang@sip.kumoh.ac.kr">
    <uid>Choonseo</uid>
    <entity-pr-case
      status="valid"
      rsc-cont-id="dtRuaw@sip.kumoh.ac.kr"/>
  </entity-pr>
  <entity-pr
    uri="sip:hyun@sip.kumoh.ac.kr">
    <uid>JHgue</uid>
    <entity-pr-case
      status="valid"
      rsc-cont-id="JdtUtha@sip.kumoh.ac.kr"/>
  </entity-pr>
  <entity-pr
    uri="other_domain_list@other.domain.net">
    <uid>other domain list</uid>
  </entity-pr>
</rsc-pres-list>
```

그림 3 설계된 리소스 리스트 메타 정보 예
Fig. 3 Example of Designed Resource Lists Meta Information

그림 4에 이와 같은 리소스 리스트 메타 정보에 의해 생성된 실제 리소스 리스트의 예를 보았다. 여기서는 리소스 리스트 메타 정보에서 사용된 <entity-pr-case> 원소에 의해 실제 프레즌스 정보를 나타내었고 이 정보 포맷은 PDPF를 따른다. 여기서 <status> 원소의 하위원소인 <basic>은 각 리소스의 현재 프레즌스 정보 상태를 나타낸다. 이상에서 구현한 리소스 리스트를 사용하여 얻을 수 있는 네트워크 트래픽의 감소는 각 리소스에 개별 등록할 때 발생하는 SIP SUBSCRIBE 메시지 및 이에 대한 SIP 200 OK 응답 메시지, 서버에서 발생하는 개별 NOTIFY 메시지 및 이에 대한 SIP 200 OK 응답 메시지, 리프레시 요청 메시지 및 이에 대한 SIP 200 OK 응답 메시지와 이때 발생하는 NOTIFY 메시지의 총 합이 되며 사용자 수 및 각 리소스 개수가 증가 할수록 네트워크 트래픽의 감소 효과가 커짐을 알 수 있다.

```
Content-Transfer-Encoding: binary
Content-ID: <dtFuaw@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:jang@sip.kumoh.ac.kr">
  <tuple id="f6HryU">
    <status>
      <basic>open</basic>
    </status>
    <contact priority="1.0">
      sip:jang@mail.kumoh.ac.kr</contact>
    </tuple>
  </presence>

--8HykYBn73KmQdjsp
Content-Transfer-Encoding: binary
Content-ID: <JdtUfha@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:hyun@sip.kumoh.sc.kr">
  <tuple id="bsTrkUh3e">
    <status>
      <basic>closed</basic>
    </status>
  </tuple>
</presence>
```

그림 4 구현된 실제 리소스 리스트 정보의 예
Fig. 4 Example of Implemented Real Resource Lists Information

2. 이벤트 통지 메시지 제어

동적 통지 제어란 시스템 동작 상황에 맞추어 리소스 리스트 서버로부터의 이벤트 통지 메시지 발생률을 동적으로 직접 제어 할 수 있는 기능이며 본 논문에서는 다량의 이벤트 통지 메시지에 의한 네트워크 트래픽을 감소시키기 위하여 이 기능을 연구하였고 SIP 세션 연결 상황에 따라 각종 통지 메시지 발생 여부도 능동적으로 제어 할 수 있는 새로운 이벤트 통지 제어 방식을 제안하였다. 현재까지의 이벤트 패키지는 이러한 기능을 제공하고 있지 않다. 이를 위하여 SIP SUBSCRIBE 메시지의 Event 헤더를 확장하여 통지 메시지의 최소 간격을 나타내는 파라미터로 notify-min-duration을 사용하고 평균 간격을 나타내는 파라미터로 notify-avg-duration을 사용하였다.

각 파라미터의 수치는 초를 나타내며 서버는 notify-min-duration 값에 의해 통지 메시지의 주기를 제어하여 이 값보다 크도록 유지하며 이 값 이내에서 발생하는 통지 메시지는 내부에서 버퍼링한다. 또 서버는 notify-avg-duration 값에 의해 통지 메시지의 평균 주기 간격을 제어 하는데 리소스에 대한 통지 메시지를 최근에 자주 보냈으면 다음번 통지 메시지의 간격은 이 값에 의해 길게 한다. 이때의 시간 간격은 다음 식을 사용하였다.

$$D = (\text{notify-avg-duration} \wedge 2) \times C / P \quad (1)$$

여기서 D는 통지 메시지 간격이고 P는 평균 주기이며 C는 평균 주기 내에서 발생한 통지 메시지 회수이다. 사용자는 SIP SUBSCRIBE 메시지에 최소 및 평균 간격을 나타내는 두 개 파라미터를 모두 표시 할 수도 있고 둘 중 하나만 표시할 수도 있다. 서버는 최초의 SIP NOTIFY 메시지의 Event 헤더에 이 파라미터 값을 넣어 전송하여 사용자의 요청을 처리 할 수 있음을 알린다. 그림 5에 이벤트 통지 메시지 발생률 제어 사용 예를 보였다. 여기서는 통지 메시지의 최소 간격을 나타내는 파라미터 notify-min-duration로 20, 평균 간격을 나타내는 파라미터 notify-avg-duration로 30을 사용하고 있으나 시스템 정책에 의해 적절히 변경할 수 있다. 또 이 서버가 리소스 리스트를 처리 할 수 있음을 알리기 위하여 Require 헤더에서 rsc-event-list 태그를 사용하였다.

```
NOTIFY sip:client39,kumoh.ac.kr SIP/2.0
Via: SIP/2.0/TCP sip.kumoh.sc.kr:branch=h43urz9hg4
Max-Forwards: 70
From: <sip:ja-list@pres.kumoh.sc.kr>;tag=zpNctbZq
To: <sip:jun@pres.kumoh.sc.kr>;tag=ie4hbb8t
Call-ID: h5dT3k5s@client39,kumoh.ac.kr
CSeq: 813663721 NOTIFY
Contact: <sip:client39,kumoh.ac.kr>
Event: presence; notify-min-duration=20;
                               notify-avg-duration=30
Subscription-State: active;expires=3600
SIP-Etag: fea4g5
Require: rsc-event-list
```

그림 5 이벤트 통지 메시지 발생률 제어 예
Fig. 5 Example of Event Notification Rate Control

SIP 시스템에서는 세션이 계속 유지되도록 하기위하여 주기적으로 리프레시 동작이 필요하고 이 과정에서 통지 메시지가 계속 발생한다. 본 연구에서는 필요시 이를 사용자 쪽에서 억제 할 수 있도록 하여 메시지 발생량을 더욱 줄일 수 있게 하였다. 이를 위하여 서버는 사용자에게 대한 첫 번째 NOTIFY 메시지의 Event 헤더의 속성 rsc-entity에 유일한 엔티티 값을 넣어 보낸다. 사용자는 등록 시간 연장을 위한 리프레시 과정이나 세션 종료 시 통지 메시지가 필요 없

는 경우 Event 헤더의 속성 rsc-match-entity에 서버가 보낸 엔티티 값을 넣어 보내고 서버는 이 값을 비교하여 일치하면 해당 SIP 메시지에 대한 NOTIFY 메시지를 생성하지 않는다.

그림 6에 이의 동작을 보였다. 그림 6에서 리소스 리스트 서버는 최초의 SUBSCRIBE 메시지에 대한 NOTIFY 메시지의 Event 헤더의 속성 rsc-entity에 유일한 엔티티 값을 생성해 넣어 보낸다. 이후 사용자가 리프레시 시 통지 메시지를 억제하기 원하면 SUBSCRIBE 메시지의 Event 헤더의 속성 rsc-match-entity에 서버로부터 받은 엔티티 값을 넣어 보내고 서버는 이 값을 비교하여 일치하면 리프레시에 수반되는 통지 메시지를 보내지 않으며 이를 SIP 응답코드 204인 메시지로 알려준다. 서버는 리소스로 부터의 프레즌스 정보 변화가 있으면 이를 NOTIFY 메시지로 사용자에게 알려주고 이때 새로 생성한 엔티티 값도 함께 전송한다. 사용자는 서버가 보내온 엔티티 값을 사용해 세션 종료 시 SUBSCRIBE 메시지에 넣어 보냄으로써 통지 메시지를 억제할 수 있다.

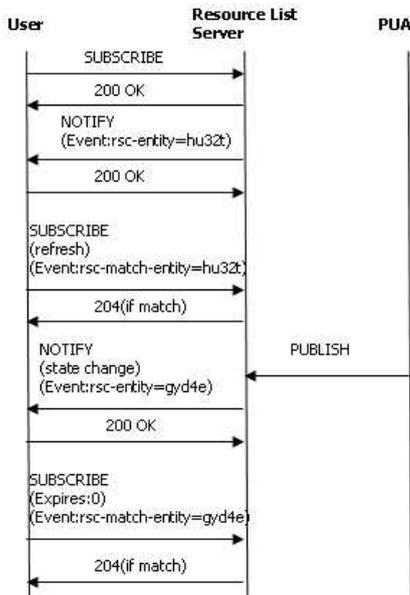


그림 6 통지 메시지 억제 동작
Fig. 6 Suppress operation of Notification Message

3. 리소스 리스트 서버

이상과 같은 기능을 갖도록 구현된 리소스 리스트 서버의 기능 블록을 그림 7에 보였다. 리소스 리스트 서버는 네트워

크와 직접 연결되는 부분에 SIP 프론트엔드 모듈을 가지는데 이는 SIP 메시지의 파싱 및 생성을 포함하여 SIP 동작과 관련된 기본적인 네트워크 기능을 처리한다. 리소스 리스트 처리 모듈은 개별 프레즌스 리소스를 나타내는 SIP URI들로 구성된 리소스 리스트를 사용 할 수 있도록 하여 네트워크 트래픽을 크게 줄일 수 있게 한다. 이벤트 통지 메시지 발생률 제어 모듈은 앞에서 설명한대로 사용자가 서버에서의 이벤트 통지 메시지 발생률을 직접 제어 할 수 있도록 하며, 이벤트 통지 메시지 억제 제어 모듈은 SIP 세션 유지를 위한 리프레시 과정이나 세션 연결 상황에 따른 각종 통지 메시지의 발생 여부를 능동적으로 제어할 수 있도록 한다. 리소스 리스트 데이터베이스는 각 모듈에서 사용되는 리소스 리스트 정보를 관리한다.

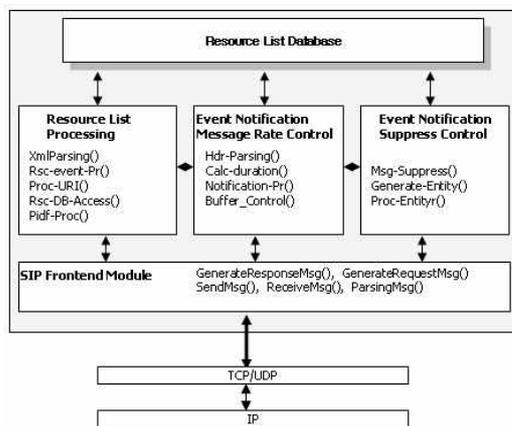


그림 7 구현된 리소스 리스트 서버의 기능 블록
Fig. 7 Function Block of Implemented Resource Lists Server

IV. 성능 분석

구현된 시스템의 성능 분석을 위하여 리소스 리스트 서버로는 리눅스(커널 버전 2.6)가 설치된 PC가 사용되었고, 프레즌스 서비스 사용차용으로는 MS 윈도우즈 XP가 설치된 15대의 PC가 사용되었다. 각 PC는 CPU로 1.8GHz 코어2듀오와 2GB 메인메모리를 가지고 있다. 각 PC의 네트워크 인터페이스를 포함한 LAN 환경의 속도는 100Mbps로 하였고 하나의 LAN 세그먼트로 구성하였다.

본 논문에서 제안한 리소스 리스트 처리 방식의 효율을 분석하기 위하여 각 사용자가 프레즌스 리소스 수를 변경시켜가며 SUBSCRIBE 메시지를 서버에게 보내는데 소요되

는 평균 시간을 측정하였다. 그림 8에 이때의 결과를 보였다. 여기서는 리소스 리스트를 사용하여 본 논문에서 제안한 방식과 각각의 SIP 등록 요청 메시지를 SIP 서버에게 보내는 기존 방식을 사용하여 비교한 결과이다. 이를 위하여 프레즌스 리소스 수를 5에서 25까지 증가시키며 사용자 시스템에서의 평균 처리 시간을 측정하였다.

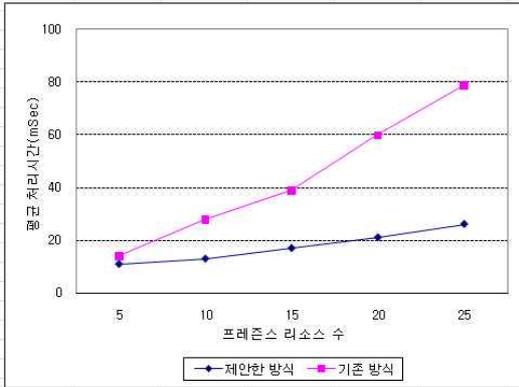


그림 8 SUBSCRIBE 메시지 평균 처리 시간
Fig. 8 SUBSCRIBE Message Average Processing Time

측정 결과 본 논문에서 제안한 리소스 리스트 방식이 기존의 방식에 비해 처리 시간을 리소스 수에 따라 5개일 때 21.4%에서 25개 일 때 67.1% 까지 단축시키며 프레즌스 리소스 개수가 증가 할수록 감소폭이 커짐을 보여주고 있다. 이는 본 논문에서 제안한 방식이 기존 방식에 비해 프레즌스 리소스 수가 증가 할수록 사용자가 서버에 전송해야 하는 SUBSCRIBE 메시지의 발생량을 크게 감소시키기 때문에 결국 서버의 부하를 감소시켜 SIP 메시지 처리 시간을 단축시키는 것으로 분석된다.

사용자가 직접 서버로부터의 이벤트 통지 메시지 발생량을 제어함으로써 발생하는 효과를 측정하기 위하여 사용자 수를 증가시키며 서버에서의 SIP NOTIFY 메시지 처리 시간을 제안된 방식과 기존 방식에 대해 서로 비교하였다. 이때 통지 메시지 최소 간격 파라미터 notify-min-duration=20, 평균 간격을 나타내는 파라미터 notify-avg-duration=30을 적용하였고 사용한 프레즌스 정보는 10Kbyte 크기의 PDF 포맷의 프레즌스 문서이다. 각 사용자 당 10개씩의 프레즌스 리소스에서 프레즌스 정보가 생성되게 하였고 프레즌스 생성 주기는 10초 간격이 되도록 하였다. 그림 9에 이때의 결과를 보였다.

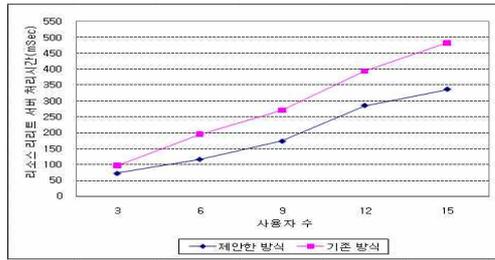


그림 9 SIP NOTIFY 메시지 처리 시간
Fig. 9 SIP NOTIFY Message Processing Time

여기서는 본 논문에서 제안한 방식이 이벤트 통지 메시지 발생률 제어기능이 없는 기존의 방식에 비해 리소스 리스트 서버에서의 처리 시간을 사용자 수에 따라 25%에서 30.6%까지 단축시키며 사용자 수가 증가 할수록 역시 감소폭이 커짐을 보여주고 있다. 이는 본 논문에서 제안한 방식이 기존 방식에 비해 서버에서 처리해야 할 SIP NOTIFY 메시지 수를 사용자 수가 증가 할수록 더욱 많은 양을 줄여주기 때문으로 분석된다.

리소스 리스트를 사용하는 환경에서 SIP 세션 유지를 위한 리프레시 동작 시 Event 헤더의 속성 rsc-match-entity를 사용하여 SIP NOTIFY 메시지를 억제하는 경우의 효과를 그림 10에 보였다. 여기서는 각 사용자 당 10개씩의 프레즌스 리소스를 사용하는 것으로 하였다.

본 논문에서 제안한 방식을 사용하였을 경우 서버에서의 처리 시간이 SIP NOTIFY 메시지 억제기능이 없는 기존 방식의 경우에 비하여 사용자 수에 따라 28.1%에서 37.7%까지 단축시킬 수 있음을 보여주고 있다. 이는 속성 rsc-match-entity에 의하여 사용자에게 전송해야 할 SIP NOTIFY 메시지 발생량이 작아지므로 서버의 부하를 감소시키기 때문이며 이와 같은 결과로 사용자 시스템에서도 필요하지 않은 프레즌스 정보 데이터의 처리 시간을 줄일 수 있고 네트워크 트래픽도 줄일 수 있는 장점이 있다.

V. 결론 및 향후 과제

본 논문에서는 새로운 동적 통지 제어 방식에 의한 리소스 리스트 구조를 설계하고 구현하였다. 이 구조에서는 시스템 동작 상황에 맞추어 리소스 리스트 서버로부터의 이벤트 통지 메시지 발생률이 동적으로 직접 제어 될 수 있도록 하였고 SIP 세션 유지를 위한 리프레시 과정에서 발생하는 SIP 메시지가 세션 연결 상황에 따른 각종 통지 메시지 발생 여부도 역시 능동적으로 제어 될 수 있도록 하였다. 이를

위하여 application/rsc-pres-list+xml 리소스 리스트 MIME 타입을 새롭게 설계하였다. 또 서비스 사용자가 상황에 맞추어 리소스 리스트 서버로부터의 이벤트 통지 메시지 발생률을 동적으로 직접 제어 할 수 있는 새로운 방식을 제안하였고 아울러 SIP 세션 유지를 위한 리프레시나 세션 연결 상황에 따른 통지 메시지 발생 여부도 능동적으로 선택 할 수 있도록 하였다. 이와 같은 방식을 사용하여 네트워크 상에서의 전체적인 SIP 메시지 양을 크게 줄일 수 있고 리소스 리스트 서버에서의 SIP 메시지 처리를 위한 부하도 감소시키며 사용자 시스템에서의 SIP 메시지 처리시간도 줄일 수 있다. 제안된 방식은 다양한 조건에서의 실험을 통하여 성능을 분석하였다. 향후 과제로는 제한된 네트워크 대역폭을 가진 무선 모바일 환경에서 각 단말기들이 다수의 프록시 리소스를 사용하는 경우 본 연구에서 제안된 시스템의 성능을 추가로 관측하고 이에 맞춘 개선된 방식을 연구 할 필요가 있다.

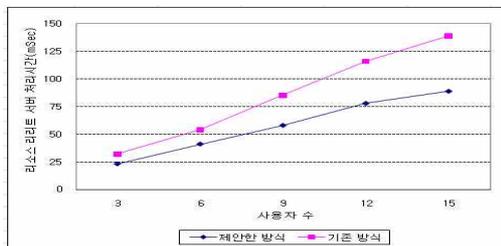


그림 10 rsc-match-entity 효과 비교
Fig. 10 Comparison of rsc-match-entity Effect

참 고 문 헌

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

[2] C. Holmberg, E. Burger, H. Kaplan, "Session Initiation Protocol (SIP) INFO Method and Package Framework" RFC 6086, January 2011.

[3] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," RFC 3856, August 2004.

[4] M. Lonnfors, et al., "Session Initiation Protocol (SIP) extension for Partial Notification of Presence

Information", RFC 5262, September 2008.

[5] B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[6] H. Sugano, G. Klyne, "Presence Information Data Format (PIDF)," RFC 3863, August 2004.

[7] Lonnfors, M. K. Kiss,, "Session Initiation Protocol(SIP) User Agent Capability Extension to Presence Information Data Format (PIDF)", RFC 5196, September 2008.

[8] C. S. Jang, "A New Instant Messaging System using Extended CPL in SIP Environment", Journal of The Korea Society of Computer and Information, Vol 14, No.9, pp 67-74, September 2009.

[9] W. S Kim, et al., "Implementation of SIP Presence Service based on Partial Publication and Notification", Conf. on Korea Information Processing Society, Nov. 2008.

[10] A. Niemi, Ed, "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, Oct. 2004.

[11] A. Niemi, M. Lonnfors, E. Leppanen, "Publication of Partial Presence Information", RFC 5264, September 2008.

저 자 소 개



장 춘 서
1993년 한국과학기술원 공학박사
1981년~현재 : 금오공과대학교 컴퓨터공학과 교수
관심분야 : SIP, 임베디드 시스템, 인터넷텔레포니
Email : csjang@kumoh.ac.kr



이 기 수
1982년 : 서울대학교 대학원 공학 석사
1982년~현재 : 금오공과대학교 컴퓨터공학과 교수
관심분야 : SIP, 디지털시스템, 데이터베이스
Email : kyssoolee@kumoh.ac.kr