부분 Publication 및 확장 호처리언어를 사용한 새로운 SIP 프레즌스 서비스에 관한 연구

A Study on a New SIP Presence Service using Partial Publication and Extended Call Processing Language

조현규, 이기수, 장춘서
금오공과대학교 컴퓨터공학과

Hyun-Gyu Jo(blackjo@kumoh.ac.kr), Ki-Soo Lee(kslee@knut.kumoh.ac.kr),
Choon-Seo Jang(csjang@kumoh.ac.kr)

요약

프레즌스(presence) 서비스는 사용자들에 관한 각종 프레즌스 정보를 등록(subscription)과 통지(notification)를 통해 제공하는 SIP(session initiation protocol) 확장 서비스 중 하나로서 VoIP(Voice over IP)나 인스턴트 메시징 서비스 등에서 중요하게 사용되고 있다. 본 논문에서는 프레즌스 서비스를 구현함에 있어 사용자가 다양한 방식으로 프레즌스 서비스와 호 처리 서비스를 조합하여 제어할 수 있도록 호 처리 언어를 확장하고, publication 시 프레즌스 정보에 변화가 있을 경우에는 전체 내용 대신 변화된 부분만 전송하는 새로운 방식을 제안하였다. 본 방식에서는 사용자는 최초 publication 시 전체 프레즌스 정보 및 자신이 원하는 내용을 기술한 호 처리 언어 스크립트를 프레즌스 서버에 등록하며, 서버는 와처(watcher)로 부터의 등록과 통지 시 이를 실행하여 프레즌스 서비스와 호 처리를 조합한 다양한 서비스를 제공할 수 있게 된다. 또한, 사용자는 이후 publication 시 프레즌스 정보의 변화된 부분만 전송하고, 프레즌스 서버도 프레즌스 정보의 통지 시 변화된 부분만 와처에게 전송하도록 하여 전체적인 시스템 효율을 높일 수 있다. 제안된 방식은 실험을 통하여 성능을 측정하였다.

■ 중심어 : | SIP | 프레즌스 서비스 | 호 처리 언어 | publication |

Abstract

The presence service which provides user's presence information by subscription and notification is one of SIP(session initiation protocol) extension services, and it is used importantly in VoIP(Voice over IP) and Instant Messaging service. In this paper, we propose a new method in which users can combine and control presence service and call processing services in various ways by extending call processing language, and only changed parts of the presence information are published instead of full presence information document. Each user registers full presence information document with his own call processing script during the first publication to a presence server. The presence server executes these call processing scripts, so it can provide various services with combination of presence service and call processing services during the presence subscriptions and notifications. Afterwards, users publish only changed parts of the presence information and the presence server notify only these changed parts to watchers. Therefore the efficiency of the overall system can be improved. The performance of our proposed model is evaluated by experiments.

■ Keyword : | SIP | Presence Service | CPL(Call Processing Language) | Publication |

* 본 연구는 금오공과대학교 학술연구비에 의하여 연구된 논문입니다.

접수번호 : #061129-001
접수일자 : 2006년 11월 29일
심사완료일 : 2007년 01월 15일
교신저자 : 조현규, e-mail : blackjo@kumoh.ac.kr

I. 서 론

SIP[1] 확장 서비스 중 하나인 프레즌스 서비스는 네트워크상에서 변화하는 사용자의 통신 상태, 위치, 선호하는 통신 수단 등 각종 정보를 제공하는 기능을 하며 이와 같은 사용자의 상태 정보를 필요로 하는 VoIP나 인스턴트 메시징 서비스 등에 폭넓게 사용할 수 있는 유용한 서비스이다[2][3].

호 처리 언어(CPL:Call Processing Language)는 VoIP에서 사용자가 호를 처리하는 다양한 방식을 기술하고 제어할 수 있는 XML 기반의 언어로서 텍스트 기반의 용이성, 통신 프로토콜에의 독립성 및 확장성 등 여러 장점을 가진다[4]. 본 논문에서는 호 처리에만 사용되던 이 CPL을 프레즌스 서비스에 사용할 수 있도록 확장하여 다양한 사용자의 서비스 요구 사항을 프레즌스 서비스와 호 처리와의 조합으로 가능하도록 하였다.

프레즌스 서비스에서 publication은 사용자가 자신의 각종 프레즌스 정보를 프레즌스 서버에게 SIP 확장 메시지 중 하나인 PUBLISH 메시지를 사용하여 알리는 것이다[5][6]. 이때 기존의 방식에서는 publication 시 매번 전체 프레즌스 정보를 서버에 전송하고 서버는 해당 사용자를 등록한 와처에게 전체 내용을 통지하였다. 이와 같은 비효율성을 개선하기 위하여 본 논문에서는 publication 시 변화된 프레즌스 정보만 서버에 전송하고 서버는 등록한 와처(watcher)에게도 이 변화된 내용만 통지하도록 하여 전체적인 시스템 효율을 높일 수 있도록 하였다.

사용자는 최초의 publication 시 자신의 전체 프레즌스 정보 및 자신의 요구대로 작성한 CPL 스크립트를 서버에 등록시킨다. 서버는 이후 와처로부터 해당 사용자에의 등록 요청이 있을 경우 이 CPL 스크립트를 실행하여 상황에 따른 처리를 함으로써 사용자에게 다양한 서비스를 제공할 수 있게 된다. 본 논문에서는 사용자가 쉽게 CPL 스크립트를 생성할 수 있도록 사용자 인터페이스 부분 및 CPL 스크립트 생성 모듈도 함께 개발하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 CPL의 기본 기능 및 구조를 설명하고 이어서 프레즌스 서비스 및 부분 publication에 대해 설명한다. III장에서는 본 논문에서 제안하는 확장 CPL 및 부분 publication을 사용하여 사용자에게 다양한 서비스 기능을 제공할 수 있고 시스템 효율도 높일 수 있는 새로운 프레즌스 서비스 시스템을 설계 및 구현하며, IV장에서는 구현된 시스템의 성능 분석을 한 후 V장에서 결론을 맺는다.

II. 관련연구

1. 호 처리 언어(CPL)

CPL은 VoIP에서 사용자가 호를 처리하는 다양한 방식을 기술하고 제어할 수 있는 XML 기반의 언어이며 이를 이용하여 사용자는 통신 환경에서 실행될 다양한 부가 서비스를 기술하여 서버에 등록할 수 있다. CPL의 호 처리 동작(Call Processing Action)은 <incoming>과 <outgoing> 태그를 사용하는 최고위 단계 동작(Top-Level Action)과 하위 단계 동작(Subaction)으로 구분되며 하위 단계 동작에서 사용되는 <subaction> 태그는 스크립트의 모듈화와 재사용을 위한 부분을 정의하는데 사용되고 <incoming>과 <outgoing> 태그는 각각 수신된 호에 대한 처리와 송신될 호에 대한 처리를 지시하는 스크립트에 사용된다. 또한 이들의 하위에는 Switches, Location Modifiers, Signaling Operations 및 Non-Signaling Operations 동작들이 있다. 이들을 사용하여 호 처리를 위한 조건과 취해야 할 행동들에 대한 세부적인 기능을 기술하고 이들의 적절한 조합을 통해 다양한 호 처리 서비스의 표현이 가능하다.

그러나 표준 문서에서 정의하고 있는 기본적인 CPL 기능들의 조합만으로는 표현할 수 없는 사용자 서비스도 있을 수 있으며 이와 같은 서비스의 제공을 위해서는 CPL의 확장이 필요하다. 기존에는 사용자 호 처리 기능 만 을 위하여 CPL을 확장하였던데 비해 본 연구에서는 프레즌스 서비스에 CPL를 적용시키기 위해 필요한 기능에 따라 CPL을 확장하였으며 이를 위하여 기존의 CPL 최고위 단계 동작에 프레즌스 등록 요청과

통지 시 서비스 처리를 위한 4개의 동작이 추가되었다.

이들은 <pres-subscr-incoming>, <pres-subscr-outgoing>, <pres-noti-incoming>, <pres-noti-outgoing> 이며 각각 수신 및 송신의 경우에 대한 프레즌스 등록과 수신 및 송신의 경우에 대한 프레즌스 통지 시 처리할 내용을 기술하는데 사용된다.

2. 프레즌스 서비스 및 Publication

프레즌스 서비스는 SIP PUBLISH 메시지를 생성하고 이를 전송하는 PUA(Presence User Agent)와 전송된 프레즌스 정보를 저장 및 처리하는 PA(Presence Agent), 그리고 프레즌스 정보 통지 대상인 와처로 구성된다. 와처는 상대방의 상태 정보를 얻기 위해 SIP SUBSCRIBE 요청 메시지를 PA에게 보내어 등록을 하며 PA는 SIP NOTIFY 메시지를 통해 등록 대상의 프레즌스 정보에 변화가 있을 때마다 알려주게 된다. PUA는 자신의 프레즌스 정보의 변화가 있을 때마다 PUBLISH 메시지를 이용하여 PA에게 통보한다. [그림 1]은 이러한 프레즌스 서비스의 메시지 처리 흐름도이다.

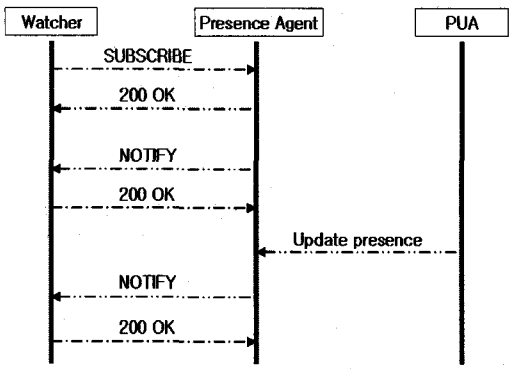


그림 1. 프레즌스 서비스의 메시지 처리 흐름

Publication의 4개 동작(초기화, 리프레시, 변경, 삭제)은 다음 [표 1]과 같이 PUBLISH 메시지에 포함된 몸체 부분의 크기, "SIP-If-Match" 헤더의 포함 여부, 유효 시간의 길이 및 프레즌스 정보의 포함 여부에 의해 이루어진다.

PUBLISH 메시지와 NOTIFY 메시지에 사용되는 프레즌스 정보의 기본 포맷은 XML 형태인 PIDF (Presence Information Data Format)[7]를 사용한다. 이때 기존 방식에서는 전체 프레즌스 정보의 일부만 바뀐 경우에도 매번 전체 프레즌스 정보 문서를 전송한다. 이는 전송량의 증가로 이어지므로 특히 사용자 수가 많을 때나 이동 단말기와 같은 통신 환경의 대역폭이 낮을 경우 전체 시스템의 효율성을 감소시킨다. 이러한 문제점을 해결하기 위하여 본 논문에서는 PUA가 publication 할 때와 PA가 등록한 와처에게 통지할 때 기존의 전체 프레즌스 정보를 담은 포맷에 비해 변화된 프레즌스 정보만 담은 부분 프레즌스 정보 포맷[8]을 사용하도록 구현하였다. 부분 프레즌스 정보 포맷은 각 문서의 변화된 부분을 구분하기 위한 버전(version) 속성을 반드시 가지며 기존 프레즌스 정보에 대해 변화된 부분만을 추가, 삭제, 대체 시킬 수 있는 <add>, <remove>, <replace> 태그를 가진다.

표 1. Publication 동작

동작	Content-Length	SIP-If-Match	Expires 값	프레즌스 문서
초기화	>0	no	>0	있음 (전체)
리프레시	0	yes	>0	없음
변경	>0	yes	>0	있음 (부분)
삭제	0	yes	0	없음

III. 시스템 설계 및 구현

본 논문에서 제안하는 새로운 프레즌스 서비스 시스템에서는 확장 CPL를 이용하여 사용자가 다양한 방식으로 프레즌스 서비스와 호 처리를 조합하여 제어할 수 있도록 하였고, 이를 위하여 최초 publication 시 자신이 원하는 내용을 기술한 CPL 스크립트를 프레즌스 정보와 함께 보내어 프레즌스 서버에 저장하도록 하였으며 이 후의 publication 동작에서는 프레즌스 전체 내용 대신 변화된 부분만 전송하도록 설계 및 구현하였다.

1. 확장 CPL 설계

본 프레즌스 서비스 시스템에서는 프레즌스 서비스

에 확장 CPL을 적용시키기 위하여 CPL 최고위 단계 동작에 프레즌스 등록 요청과 통지 시 서비스 처리를 위한 4개의 동작이 추가되었다. 이들은 각각 자신에 대한 SIP 등록 요청이 들어왔을 때의 처리를 위한 <pres-subscr-incoming>, 자신이 서버에 SIP 등록 요청을 보냈을 때의 처리를 위한 <pres-subscr-outgoing>, 자신에 대한 SIP 통지가 발생했을 때의 처리를 위한 <pres-notify-incoming>, 자신이 SIP 통지를 발생했을 때의 처리를 위한<pres-notify-outgoing>이다.

이외에도 프레즌스 정보 대상의 주소를 조건으로 하여 조건에 따른 처리를 하기 위한 CPL 스위치인 <pres-switch>가 추가되었고 또한, 자신에 대한 SIP 등록 요청을 허락하거나 거부하기 위한 <accept>, <deny>와 프레즌스 정보를 조건으로 하여 주어진 주소에 대한 호를 발생할 수 있는 <call> 동작이 추가되었다. [표 2]는 이와 같은 CPL의 확장된 부분들이다.

표 2. CPL의 확장된 부분들

Top-level Actions	Operations	Decision
⟨pres-subscr-incoming⟩ ⟨pres-subscr-outgoing⟩ ⟨pres-notify-incoming⟩ ⟨pres-notify-outgoing⟩	⟨accept⟩ ⟨deny⟩ ⟨call⟩	⟨pres-switch⟩

이러한 확장된 CPL이 사용되는 예들은 다음과 같다. 먼저 [그림 2]는 자신에게 들어오는 외부로부터의 호를 자신의 프레즌스 정보에 기반하여 현재 이를 직접 받기 곤란한 상황인 경우 음성사서함으로 전달시키는 호 전달 제어의 예이다. 여기서 프레즌스 서비스 처리를 위해 확장된 기능인 스위치 <pres-switch>가 사용되었고 PIDF 포맷 프레즌스 정보의 원소 중 하나인 activities 값이 스위치 문의 조건으로 사용되었다.

[그림 3]은 자신에게 들어오는 프레즌스 등록 신청에 대해 자신이 원하는 시간대에 걸쳐, 허락 또는 거절하는 경우이다. 여기서 자신에 대한 SIP 등록 요청이 들어왔을 때의 처리를 위한 <pres-subscr-incoming> 태그와 등록 요청을 허락하는 <accept> 태그 및 등록 요청을 거부하는 <deny> 태그가 각각 사용되었다.

```
<?xml version="1.0" encoding="UTF-8"?>
<cpl xmlns="urn:ietf:params:xml:ns:cpl"
  xmlns:pr-sys="http://www.irt.kumoh.ac.kr/presence-sys"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:ns:cpl cpl.xsd
 http://www.irt.kumoh.ac.kr/presence-sys presence-sys.xsd">

<incoming>
   <pr-sys:pres-switch presentity="sip:jkpark@kumoh.ac.kr">
      <pr-sys:presence activities="away">
         <location url="sip:jkpark-voicemail@kumoh.ac.kr>
            </proxy>
         </location>
      <pr-sys:/presence>
   <pr-sys:/pres-switch>
</ incoming>
</cpl>
```

그림 2. 프레즌스 정보에 기반한 호 전달 제어

[그림 3]의 경우에는 <address> 태그로 지정된 사용자로부터 등록 요청이 들어 왔을 때 <time-switch>를 사용하여 매주 일요일 오전 9시부터 12시간 동안은 이를 거부하고 그 이외의 시간 및 나머지 요일에는 허락하는 동작을 위한 스크립트이다.

```
<?xml version="1.0" encoding="UTF-8"?>
<cpl xmlns="urn:ietf:params:xml:ns:cpl"
  xmlns:pr-sys="http://www.irt.kumoh.ac.kr/presence-sys"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:ns:cpl cpl.xsd
 http://www.irt.kumoh.ac.kr/presence-sys presence-sys.xsd">

<pr-sys:pres-subscr-incoming>
   <address-switch field="origin">
      <address is="sip:dykim@kumoh.ac.kr">
         <time-switch>
         <time dtstart="20061001T090000"
                  duration="PT12H" freq="weekly" byday="SU">
            <pr-sys:deny/>
         </time>
         <otherwise>
            <pr-sys:accept/>
         </otherwise>
      <addrdess/>
   <address-switch/>
</pr-sys:pres-subscr-incoming>
</cpl>
```

그림 3. 프레즌스 등록 요청의 제어

2. 프레즌스 서버

프레즌스 서버는 앞절에서 설명한 확장 CPL 처리 부분과 사용자(PUA)로부터 변화하는 프레즌스 상태 정보를 가지고 있는 PUBLISH 메시지를 수신하여 이 사용자를 등록한 다른 사용자들에게 통보하는 PA의 기능

을 모두 가지고 있다. 이와 같은 기능을 가지는 프레즌스 서버의 내부 구성은 [그림 4]와 같다.

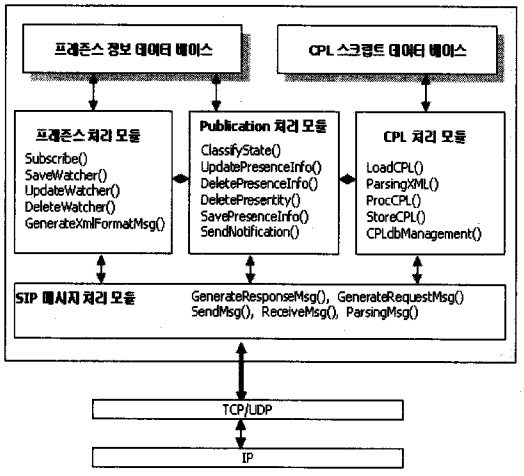


그림 4. 프레즌스 서버의 내부 구성

여기서 CPL 처리 모듈은 사용자가 보내온 CPL 스크립트를 데이터베이스에 등록하고 프레즌스 등록 요청, 통지 및 호 신호에 따라 해당되는 CPL 스크립트를 실행하여 프레즌스 서비스와 호 처리를 조합한 다양한 서비스를 제공할 수 있게 한다. Publication 처리 모듈은 사용자로 부터의 publication 메시지를 처리하여 여기에 포함된 CPL 스크립트를 CPL 처리 모듈로 보내고, 프레즌스 정보는 프레즌스 정보 데이터베이스의 해당 테이블에 기록한다. publication 메시지의 포맷은 CPL 스크립트와 프레즌스 정보가 함께 포함되어 있음을 나타내기 위하여 본 논문에서 제안하는 application/pidf-diff-cpl+xml을 사용한다. 이 과정에서의 publication 처리는 메시지 중 SIP-If-Match 헤더의 포함 여부를 확인하여 이 헤더를 포함하지 않으면, 임의의 값으로 SIP-ETag 값을 생성하고 이를 프레즌스 정보 테이블에 값을 저장한다. 이 경우는 publication의 초기상태에 해당하므로 publication 메시지의 프레즌스 정보는 부분 포맷이 아닌 전체 문서 포맷이며 이 정보를 역시 프레즌스 정보 테이블에 저장한다. 다음 이 테이블에서 publication 메시지를 보낸 사용자에 대해 프레즌스 서비스 등록 요청을 한 다른 사용자(와처)들을 찾아 이들에게 통지 메시지를 보낸다.

만일 publication 메시지에 SIP-If-Match 헤더가 포함되어 있으면 이는 publication의 초기상태 이후에 해당하므로 테이블에 저장되어 있는 SIP-ETag 값과 SIP-If-Match 헤더 값을 비교한다. 값이 일치하지 않으면 412(Precondition Failed) 응답코드를 보내고 세션을 종료하며, 헤더 값이 일치하면 응답 메시지의 SIP-ETag 헤더를 위한 새로운 값을 생성하여 테이블에서 해당 사용자의 SIP-ETag 값을 갱신하고 이 값을 담은 응답 메시지를 상대방에게 전송한다.

다음 Expire 헤더 값이 0이면 테이블에서 해당 사용자의 프레즌스 정보를 삭제 후 이를 통지 메시지로 와처들에게 알린다. Expire 헤더 값이 0이 아니나 Content-Length가 0인 경우이면 사용자의 유효 시간을 갱신하는 리프레시 동작을 행하고 이 경우 와처들에게 통지 메시지는 보내지 않는다. Content-Length가 0보다 크고 포함된 내용이 부분 PIDF 포맷의 프레즌스 문서이면 프레즌스 정보 전체를 기록하는 대신 변경된 부분에 대한 갱신, 삭제, 추가 기능만 이루어진다. 프레즌스 정보 테이블을 갱신한 후에는 통지 메시지에 역시 부분 PIDF 포맷의 프레즌스 문서를 포함시켜 와처들에게 알린다.

위의 과정에서 수신된 프레즌스 문서의 버전(version) 값이 테이블에 저장되어있는 publication 메시지를 보낸 사용자의 LVC(Local Version Count) + 1 이어야 정상이며 그렇지 않은 경우 전송과정에서 부분 프레즌스 문서의 일관성이 손상된 것이므로 상대방에게 초기상태의 전체 프레즌스 문서를 요청하게 된다. 프레즌스 정보를 기록하는 테이블 PRES의 각 필드 구성은 [표 3]과 같다.

표 3. 프레즌스 정보 관리를 위한 테이블 구조

테이블 명	필드 구성
PRES	activities(String), version(Integer), status(String), basic(String), status-icon(String), mood(String), contact(String), tuple(String), priority(String), note(String), timestamp(String), class(String), device-id(String), place-is(String), place-type(String), relationship(String), expires(Integer), SIP-ETag(String), time-offset(String), service-class(String)

프레즌스 처리 모듈은 프레즌스 정보를 얻고자 하는 와처들로 부터의 등록요청이 들어오면 이를 CPL 처리 모듈로 보내어 저장된 CPL 스크립트에서 처리할 부분이 있는지 조사하게 하고 다음 이를 프레즌스 정보 데이터베이스에 저장하며 유효시간의 조사 등 등록의 유지 관리 기능을 한다.

3. 사용자 시스템

사용자 시스템의 구성은 [그림 5]와 같다.

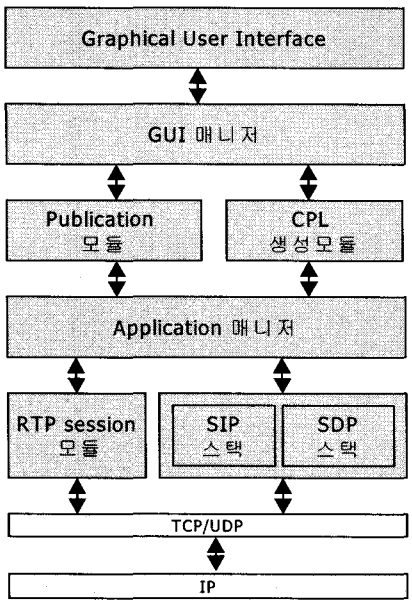


그림 5. 사용자 시스템 구성

CPL 생성 모듈은 GUI 매니저와 Application 매니저 사이에서 사용자로 하여금 프레즌스 정보와 호 처리의 조합을 위한 원하는 CPL 스크립트를 생성 하는 부분이며 GUI 매니저를 통해서 CPL에 익숙하지 않은 일반 사용자들도 쉽게 사용할 수 있는 인터페이스 화면이 제공되도록 구현하였고 본 논문에서 제안한 확장 CPL을 모두 지원할 수 있도록 하였다. [그림 6]은 CPL 스크립트를 생성하기 위한 인터페이스 화면이다.

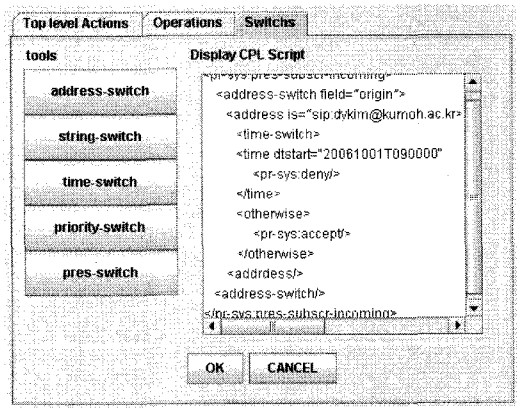


그림 6. CPL 스크립트 생성 화면

[그림 7]은 [그림 6]에서 새롭게 추가된 CPL 스위치인 pres-switch를 누르면 이 스위치에 필요한 속성인 presentity 값을 입력받기 위한 화면이다.

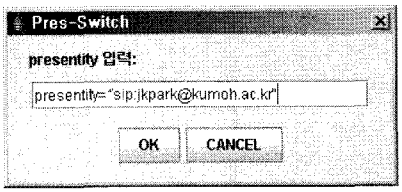


그림 7. pres-switch의 presentity 입력화면

Publication 모듈은 자신의 프레즌스 정보가 변화하였을 때 프레즌스 서버에게 부분 publication 메시지를 생성해 전송하는 기능과 프레즌스 서버가 보내오는 부분 PIDF 포맷의 프레즌스 문서를 처리하는 기능을 한다. RTP[9] 세션 모듈은 오디오와 비디오 송수신 기능을 한다.

Ⅳ. 성능 분석

부분 publication이 프레즌스 서버의 성능에 미치는 효과를 측정하기 위하여 100Mbps LAN 환경에서 프레즌스 서버와 사용자 시스템을 각각 구성하였다. 프레즌스 서버의 사양은 펜티엄 IV CPU 1.8GHz, 메인메모리 256MB이며 운영체제로는 리눅스(커널 2.4)를 사용하

였고 각 모듈은 자바로 구현하였다. 또 사용자 시스템은 펜티엄 IV CPU 2.4GHz, 메인메모리 256MB이며, 운영체제로는 MS 윈도우 XP를 사용하였고 각 모듈은 역시 자바로 작성하였다.

실험방법은 다수의 사용자들이 publication 요청을 서버에 보내고 서버는 이를 처리하여 와처들에게 통지 메시지를 보내는 환경을 시뮬레이션하기 위하여 1개의 사용자 시스템이 1초에 10개, 20개, 40개의 전체(full) publication 요청메시지 및 부분 publication 요청메시지를 각각 발생하게 하고 서버에서 이 요청에 대해 각각 10개, 20개, 40개의 전체 PIDF 및 부분 PIDF 포맷의 통지 메시지를 발생하게 하여 이들 처리가 완료되는 시간 간격을 측정하였다. 측정 결과는 [그림 8]과 같다.

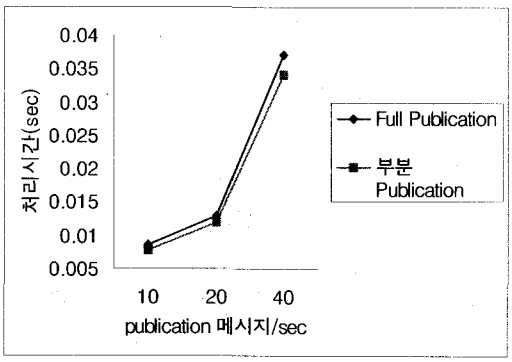


그림 8. full publication 및 부분 publication 방식의 처리 시간 비교

실험에서 전체 publication의 경우 크기 2Kbyte의 전체 PIDF 포맷의 프레즌스 문서를 사용하였고 부분 publication인 경우 최초의 publication에서는 크기 2Kbyte의 전체 프레즌스 문서를 보내고 이후부터는 이 프레즌스 문서의 1개의 튜플(tuple)에서 통신 상태를 표시하는 <basic> 태그의 값이 "open" 과 "closed" 를 번갈아 가며 변화하는 것으로 하였다. 여기서는 본 논문에서 구현한 부분 publication 방식이 기존의 전체 publication 방식에 비해 서버에서의 처리 시간이 약 8.1% ~ 9.4% 단축됨을 알 수 있다. 최초 publication시 포함되는 CPL 스크립트는 처리시간에 거의 영향을 미치지 않았다. 그러나 사용자 시스템에서도 부분 PIDF 포맷의 프레즌스 문서 처리를 위한 추가 로직이 필요하다. 또 부분 publication은 무선 단말기 등과 같은 낮은 전송대역폭을 가진 사용자 시스템의 경우 더욱 효과를 볼 수 있으며 추후 이와 같은 데이터 링크 환경에서 효율 측정을 위한 연구가 추가로 필요할 것으로 보인다.

V. 결론

본 논문에서는 프레즌스 서비스를 구현함에 있어 사용자가 다양한 방식으로 프레즌스 서비스와 호 처리를 조합하여 제어할 수 있도록 확장 CPL를 사용하고, publication 시 프레즌스 정보에 변화가 있을 경우 전체 내용 대신 변화된 부분만 전송하여 효율을 높일 수 있는 새로운 프레즌스 서비스 시스템을 설계 및 구현하였다. 사용자는 최초 publication 시 전체 프레즌스 정보 및 자신이 원하는 내용을 기술한 CPL 스크립트를 프레즌스 서버에 등록하며, 서버는 와처로 부터의 등록과 통지 시 이 스크립트를 실행하여 프레즌스 서비스와 호 처리를 조합한 다양한 서비스를 제공할 수 있게 된다. 사용자는 이후 publication 시에는 프레즌스 정보의 변화된 부분만 프레즌스 서버에게 보내고 프레즌스 서버는 와처에게 통지할 때 부분 PIDF 포맷의 프레즌스 문서를 사용하여 변화된 부분만 전송한다.

이와 같은 동작에 의하여 프레즌스 서버에서 publication 요청 메시지의 처리 및 와처에의 통지시간을 감소시킬 수 있음을 실험을 통하여 알 수 있었다. 본 논문에서는 사용자가 쉽게 CPL 스크립트를 생성할 수 있도록 이를 위한 사용자 인터페이스 부분 및 CPL 스크립트 생성 모듈도 함께 개발하였다.

참 고 문 헌

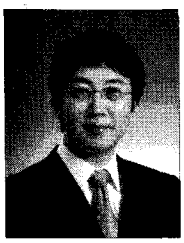
[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[2] J. Peterson, "Common Profile for Presence

(CPP)", RFC 3859, August 2004.
[3] M. Day, J. Rosengerg and H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, February 2000.
[4] J. Lennox, Xiaotao Wu and H. Schulzrinn, "CPL: A Language for User Control of Internet Telephony Service", RFC3880, October 2004.
[5] A. Niemi, Ed., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.
[6] A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.
[7] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr and J. Peterson, "Presence Information Data Format (PIDF)", RFC3863, August 2004.
[8] M. Lonnfors, E. Leppanen, H. Khartabil and J. Urpalainen, "Presence Information Data format (PIDF) Extension for Partial Presence", draft-ietf-simple-partial-pidf-format-05, October 2005.
[9] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

저 자 소 개

조 현 규(Hyun-Gyu Jo) 정회원



- 1991년 2월 : 금오공과대학교 전자공학과 (공학사)
- 1995년 2월 : 금오공과대학교 전자공학과 (공학석사)
- 2005년 8월 : 금오공과대학교 컴퓨터공학과(공학박사)
- 2006년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학과 계약교수

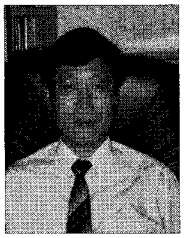
<관심분야> : SIP, VoIP, 실시간 인터넷 통신

이 기 수(Ki-Soo Lee) 정회원

- 1979년 2월 : 경북대학교 전자공학과(공학사)
- 1982년 2월 : 서울대학교 대학원(공학석사)
- 1982년 3월 ~ 현재 : 금오공과대학교 컴퓨터공학부 교수

<관심분야> : 디지털시스템, 데이터베이스

장 춘 서(Choon-seo Jang) 정회원



- 1978년 2월 : 서울대학교 전자공학과(공학사)
- 1981년 2월 : 한국과학기술원(공학석사)
- 1993년 2월 : 한국과학기술원(공학박사)
- 1981년 3월 ~ 현재 :금오공과대학교 컴퓨터공학부 교수

<관심분야> : SIP, 임베디드 시스템