JXTA 기반 서비스 푸시 시스템

정회원 조 윤 식*, 정 인 환*, 황 기 태*

A Service Push System Based on JXTA

Yoon Sik Cho*, In Hwan Jung*, Kitae Hwang* ***Regular Members***

요 약

전통적인 서버/클라이언트 구조에서 클라이언트는 이미 알고 있는 서버에 접속하여 정보를 얻거나 웹 검색 등과 같은 방식으로 정보나 서비스를 이용하였다. 그러나 최근에 다양한 장치들이 인터넷에 접속 가능하게 되었으며, 이들 장치들이 제공하는 정보와 서비스의 갱신 주기가 짧고, 서비스 제공자이면서 동시에 서비스 소비자가 되는 특징을 보인다. 이런 상황에서는 네트워크 상에 존재하는 새로운 정보나 서비스를 찾기에는 전통적인 방식이 부적합하다. 본 논문에서는 장치가 네트워크에 접속되면 서비스의 프로파일을 주변의 모든 장치들에게 방송 형태로 알리는 서비스 푸시 모델을 제안한다. 임의의 네트워크 장치가 새로 연결되면 네트워크 상에 이미 존재하는 서비스 프로파일 목록을 한 번에 얻거나 서비스 제공자로부터 새로 푸시된 서비스 프로파일을 즉각적으로 얻게 되어 언제든지 서비스를 이용할 수 있다. 본 논문에서는 P2P 네트워크를 기반으로 하는 JXTA 플랫폼을 이용하여 서비스 푸시 시스템을 구현하고 실험을 통해 서비스 푸시 시 서비스 프로파일 전파 시간, 네트워크 진입시 서비스 목록을 받는데 걸리는 시간, 서비스 호출에 따른 실행 응답 시간 등을 측정하였다.

Key Words : P2P, service, service push, JXTA, service oriented

ABSTRACT

In the traditional Server/Client architecture, the client connects to the known servers and gets information through web searching. However, nowadays, diverse internet devices providing services (get) working on internet easily. The update interval of the services or information provided by the devices is relatively short. And also they tend to be both a server and a client. In this situation, the way of the traditional information searching such as web searching is not appropriate. In this paper we propose a service push model, where it pushes its service profiles to all the other network devices by broadcasting them on the network. When a device connects to the network newly, it can get all service profiles in the network and it gets service profiles newly pushed by service providers in prompt. Also it can call any service among the service profiles within itself. We implemented a prototype system for the service push model on the JXTA platform based on P2P network. Also we measured the service profile pushing delay, the service discovery time, and the response time of service execution by running test applications written on the prototype system.

Ⅰ. 서 론

네트워크 및 컴퓨터 기술의 발전으로 인해 인터넷에는 대용량의 데이터베이스나 자원을 가진 서버뿐만 아니라 디지털 카메라, 소형 가전에 이르기까지 소형의 장치들도 인터넷 상에 연결되게 되었으며 사용자

※ 본 연구는 2008학년도 한성대학교 교내 연구비 지원 과제임.
* 한성대학교 컴퓨터공학과(calafk@hansung.ac.kr)
논문번호 : KICS2008-08-351, 접수일자 : 2007년 8월 18일, 최종논문접수일자 : 2008년 11월 24일

는 원격에서 이들을 제어하기도 하고 이들이 제공하는 정보 서비스도 받을 수 있게 되었다[1][2]. 또한 과거 전통적인 서버/클라이언트 모델에서는 서비스를 제공하는 서버와 이를 이용하는 클라이언트로 대체로 구분되어 있었지만 현재는 네트워크 장치들이 그들의 사용자에 의해 서비스를 제공하는 동시에 다른 네트워크 장치들로부터 서비스를 이용하는 고객이 되는 특징을 가지고 있다. 뿐만 아니라, 이들 네트워크 장치들은 순식간에 네트워크에 존재하였다가 사라지기도 하는 등 과거와 달리 동적인 행태를 보인다[3]. 그리고 네트워크에는 정보나 서비스를 제공하는 많은 장치들이 존재하기 때문에 어떤 사용자가 어떤 로칼 네트워크 상에 연결하였을 때 이 네트워크로부터 이용 가능한 서비스를 찾는 방법 또한 막연하다. 과거와 같이 홈 페이지를 방문하여 열심히 그 내용을 읽고 네트워크 내의 상황을 파악하는 단순한 방법으로는 부족하다.

본 논문에서는 이러한 상황 아래에서 보다 편리하고 쉽게 네트워크 상에 존재하는 정보나 서비스를 활용할 수 있는 방법에 초점을 맞추었다. P2P 네트워크는 네트워크 상에서 피어들끼리 정보를 주고받는 등 독립적인 통신의 단위를 형성하므로 서비스나 정보를 제공하는 다양한 서버들이 존재하는 곳에 매우 적합한 구조이다[4][5].

본 논문은 서비스의 발견과 서비스의 활용이 빠르고 쉽게 이루어질 수 있는 서비스 푸시 모델을 제안한다. 사용자가 로칼 네트워크에 접속하면 서비스를 가진 네트워크 장치들이 자신의 서비스에 대한 프로파일을 사용자의 컴퓨터로 자동 전송하고 사용자는 자신이 접속한 로칼 네트워크 내에서 지원되는 모든 서비스를 순식간에 인지할 수 있으며 이를 바로 활용할 수 있다.

본 논문은 서비스 푸시 모델은 구성하기 위해 P2P 네트워크를 기반으로 하며 서비스의 발견을 위해서 서비스 지향 구조를 이용한다. 많은 경우 P2P의 응용은 호환성이 없는 고유의 프로토콜을 사용하는 경향이 있었으며, 이러한 비호환성은 각종 장치를 P2P 네트워크로 결합할 때 발생되는 장점을 감소시킨다. 결국 상호 운영성과 소프트웨어 재사용에 대한 중요성이 간과되었다. 선마이크로시스템 사에서 오픈 소스로 개발된 JXTA 플랫폼[6]은 P2P 네트워크를 구성할 수 있게 하는 플랫폼으로서 표준화된 자원 탐색 기능과 표준화된 통신 프로토콜을 제공하는 것을 목적으로 한다. JXTA는 언어와 하드웨어 및 운영체제에 대해서도 독립적이며 XML 메시지를 사용함으로써 기존 개발 도구를 활용할 수 있게 한다. 또한 프로토콜이 단순하여 PDA 혹은 휴대 전화와 같은 장치에서도 P2P 솔루션의 구현이 가능하도록 하는 장점이 있다[7]. JXTA 플랫폼에 메시지의 추가 확장이 용이하여 응용 시스템을 쉽게 구현할 수 있게 하는 장점을 가지고 있으며 많은 응용 시스템이 JXTA 플랫폼 상에서 구현되었다. 이러한 이유로 본 연구에서는 JXTA 플랫폼을 선택하고 이 위에 서비스 푸시 시스템을 설계하고 구현하였다. 또한 간단한 테스트 시스템을 구현하고 기초적인 성능을 측정하였다.

본 논문은 Ⅱ장에서 연구배경을, Ⅲ장에서는 서비스 푸시 시스템의 설계를 소개한다. Ⅳ장에서는 프로토타입 구현 및 성능 평가에 대해 소개하고 Ⅴ장에서 결론을 맺는다.

Ⅱ. 연구 배경

2.1 서비스 지향 구조

서비스 지향 구조(Service Oriented Architecture)란 문제 해결에 필요한 작업을 관련된 조각으로 분할한 후 각 조각들을 조합하여 사용하는 모델이다. 분할된 조각들을 서비스라고 부른다[8]. 작은 단위로 분할된 서비스들을 통해 관리의 편의성과 서비스들의 조합으로 또 다른 작업을 생성할 수 있는 장점을 지니고 있다.

그림 1은 전형적인 서비스 지향 구조의 구성을 나타내고 있다. 서비스 제공자는 서비스 명세서를 서비스 저장소에 배포한다. 서비스 저장소는 서비스 명세서를 저장하며 서비스 소유자 정보 등과 같은 메타 정보를 함께 기록한다. 서비스 제공자가 서비스 저장소에 명세서를 배포함과 동시에 서비스 요청자들에게도 명세서를 배포하는 푸시 방법과 서비스 제공자가 서비스 저장소에만 서비스 명세서를 배포하는 풀 방식으로 분류된다. 풀 방식에서는 서비스 요청자가 서비스 저장소로부터 서비스 명세서를 검색하여 원하는 서비스를 사용한다. 푸시 방법에서는 서비스 요청자

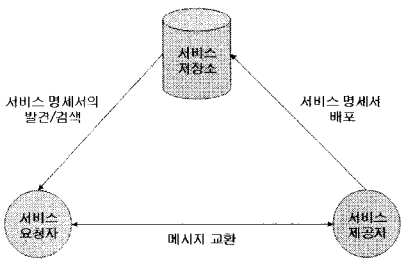


그림 1. 전형적인 서비스 지향 구조

에게 자동으로 명세서가 배포되기 때문에 명세서 검색 과정이 생략되는 장점이 있으며 풀 방법에서는 서비스 제공자는 서비스 저장소에만 서비스 명세서를 배포하면 되는 장점이 있다.

2.2 JXTA

2.2.1 JXTA 플랫폼 개요

JXTA는 그림 2와 같이 3계층 구조로 되어 있으며 JXTA 네트워크에서 피어들은 논리적 링크들로 구성된 가상 네트워크로 연결 된다. 이러한 가상 네트워크를 통해서 각 피어들은 물리적 연결에 대한 고려가 필요 없기 때문에 네트워크의 복잡성을 줄일 수 있게 된다. 그림 3은 JXTA의 가상 네트워크를 보여준다.

물리적인 네트워크에는 방화벽이나 NAT(Network Address Translator) 등이 존재하여 외부에서의 접근이 제한된다. 그러나 가상 네트워크에서는 이러한 제한이 없다. 릴레이 피어(Relay Peer)와 랑데부 피어(Rendezvous Peer)는 방화벽 또는 NAT 내에 존재하는 피어들과 그 외 피어들 사이의 통신을 가능하게 함으로써, 물리적으로 분리된 네트워크 영역 간의 피

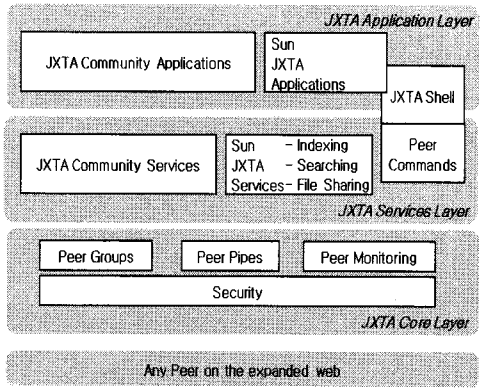


그림 2. JXTA의 계층 구조

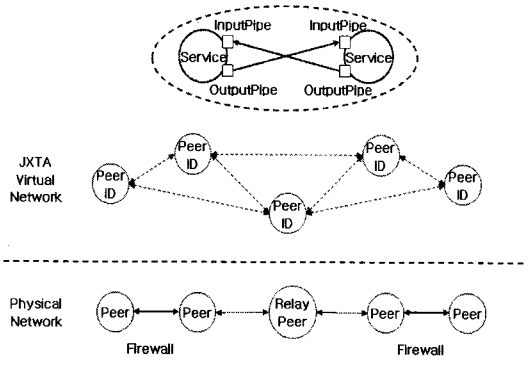


그림 3. JXTA의 가상 네트워크

어들을 연결시켜 준다.

피어 간의 통신은 파이프(pipe)라고 불리는 가상 채널을 이용한다. 메시지를 받는 피어는 입력 파이프(Input Pipe)를 생성하고 메시지를 전송하는 피어는 입력 파이프 정보를 참조하여 출력 파이프(Output Pipe)를 생성하여 피어간 가상 채널이 만들어 진다. 상대방의 피어의 IP, 포트 번호 등의 정보를 사용하지 않는다.

2.2.2 프로토콜

JXTA는 모든 유형의 P2P에 대한 기반을 제공하기 위한 프로토콜을 가지며, 운영체제, 개발 언어에 제한이 없고, 임베디드 장치에서부터 대규모 슈퍼컴퓨터에 이르는 모든 종류의 장치가 피어가 될 수 있음을 명시하고 있다. 표 1은 JXTA의 프로토콜 스택과 그에 대한 요약된 설명이다.

표 1. JXTA 프로토콜 설명

계층	프로토콜	설명
6	Peer Information Protocol	피어의 상태 정보 제공(최상위 계층)
5	Peer Rendezvous Protocol	물리적으로 다른 네트워크 상의 피어에게 메시지 전달
4	Pipe Binding Protocol	피어간의 데이터 송수신을 위한 가상 채널인 파이프 연동
3	Peer Discovery Protocol	다른 피어의 광고를 발견하거나 자신의 광고 전파 시 사용
2	Peer Resolver Protocol	피어의 광고에 대한 탐색 메시지 송수신
1	Endpoint Routing Protocol	피어 사이의 라우팅 프로토콜 (최하위 계층)

2.3 관련 연구

서비스 푸시의 개념은 매우 새로운 개념이 아니다. 홈 네트워크 대표적인 기술인 UPnP(Universal Plug and Play)[9]나 Jini[10] 프로토콜은 서비스 지향 아키텍처의 푸시와 풀 모델을 각각 구현한 사례이며 P2P의 개념으로 작동된다. 그러나 이들은 비교적 작은 공간 내에서 소수의 장치들로 구성되는 네트워크라는 한계를 가지고 있으므로 이들을 그대로 규모가 큰 네트워크에 적용할 경우 서비스 발견 등에서 심각한 성능 문제가 발생한다. JXTA는 보다 큰 네트워크 상에서도 서비스 지향 아키텍처와 P2P의 개념을 지원하도록 설계된 플랫폼의 대표적인 사례이다. 그러나 본 논문에서 추구하는 서비스 푸시의 응용 모델이 설계 개발된 사례는 거의 없으며 이에 대한 성능 평가 역시 이루어진 적이 거의 없다. 현재 진행 중인 서비스

지향 아키텍처의 응용 사례들은 거의 웹 응용에 국한되고 있는 현실이므로 웹 서비스와 웹 클라이언트라는 일종의 전통적인 서버/클라이언트 구조를 답습하고 있다.

정보 서비스를 보다 신속하게 받기 위해 최근에 RSS, ATOM 등 웹 신디케이션(syndication) 기술에 대한 연구가 진행 중이다[11][12]. 웹 신디케이션에서, 사이트들은 자신이 현재 제공하고 있는 정보 아이템을 채널(channel)이라는 리스트 형식으로 구성하여 제공하며 클라이언트 컴퓨터 상에 RSS Reader 등의 구독기가 사용자의 사이트의 정기 구독 요청에 따라 사이트로부터 주기적으로 이 채널 정보를 가져와서 사용자에게 보여 주고, 사용자는 얻고자 하는 정보 아이템이 있으면 이 때 사이트에 접속하여 정보를 얻게 된다. 신디케이션은 마치 사용자가 가만히 있어도 원하는 정보들이 날아오는 것 같지만, 사실 내부적으로 사용자의 컴퓨터 상에서 정기 구독기가 주기적으로 계속 정보를 가져오고 있다. 또한 새로운 사이트가 생겨나고 좋은 정보가 있다고 하더라도 사용자가 정기 구독기에게 이 사이트에서 채널 정보를 가져오도록 설정하지 않으면 전혀 정보를 얻을 수 없다. 신디케이션 기술은 진정한 의미에서 서비스 푸시와는 다르다.

Ⅲ. 서비스 푸시 시스템 설계

3.1 시스템 모델

그림 4는 본 논문에서 제안하는 서비스 푸시 시스템 모델을 표현하였다. 본 논문에서는 서비스를 정보 즉 콘텐츠를 제공하는 서비스인 정보 서비스(information service)와 원격으로 장치를 제어할 수 있는 제어 서비스(control service)로 분류한다.

피어들은 서비스 제공자 피어와 서비스 소비자 피어, 릴레이 피어, 랑데부 피어로 구성된다. 서비스 제공자 피어는 서비스 명세 즉 프로파일을 가지고 네트워크 상에 제공하여 서비스 소비자 피어가 서비스를 발견하고 사용할 수 있도록 한다. 서비스 소비자 피어는 서비스 프로파일을 획득한 후 서비스 프로파일에 정의된 서비스를 직접 호출하여 서비스 제공자 피어로부터 서비스를 받을 수 있다. 한 피어는 서비스 제공자와 서비스 소비자가 동시에 될 수 있다. 서비스 프로파일은 서비스에 대한 설명과 서비스 호출시 필요한 파라미터 등을 포함하는 서비스 명세 정보로서, JXTA의 광고(advertisement)라는 방법을 통해 각 피어들에게 전파하고 인식되도록 한다.

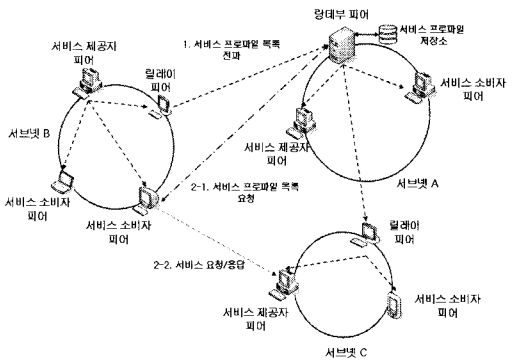


그림 4. JXTA 기반 서비스 푸시 시스템 모델

하나의 서브넷(subnet)은 물리적으로 구분되는 네트워크로서 특별한 릴레이 장치 없이 브로드캐스팅할 수 있는 단위이다. 본 논문에서는 한 서브넷에서 발생된 JXTA 메시지를 다른 서브넷으로 전송하는 중계 장치인 릴레이 피어와 로칼 네트워크 내의 서비스 피어들이 광고한 서비스 프로파일 목록을 저장 관리하며, 새로 도착하는 피어들에게 서비스 프로파일을 보내주거나 사라지는 피어로부터 서비스 프로파일을 삭제하는 등의 기능을 수행하는 랑데부 피어를 사용한다. 랑데부 피어는 릴레이 피어의 역할을 동시에 할 수 있다.

그림 4에는 본 논문의 모델이 되는 P2P 네트워크의 예로서 총 3개의 서브넷이 있고 서브넷마다 릴레이 피어가 있으며 전체 로칼 네트워크에 랑데부 피어가 한 개 존재한다.

3.2 서비스 광고 및 갱신

서비스 제공자 피어는 P2P 네트워크에 참여할 때 자신이 가진 서비스를 다른 피어들에게 알린다. 이것을 서비스 광고라 한다. 그림 5는 이 과정을 보여 주며 다음과 같은 과정으로 이루어진다.

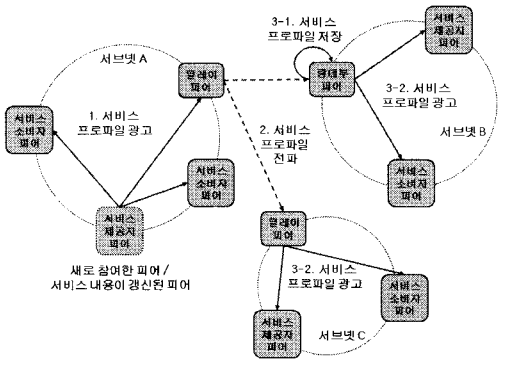


그림 5. 서비스 광고 및 갱신

① 서비스 제공자는 서비스 프로파일 목록을 브로드캐스팅한다. 이때 동일한 서브넷에 존재하는 모든 서비스 소비자 피어는 새로운 서비스 프로파일을 얻게 된다.

② 릴레이 피어는 서비스 프로파일 목록을 다른 릴레이 피어와 랑데부 피어에게 전파한다.

③ 다른 릴레이 피어는 새로 도착한 서비스 프로파일 목록을 자신의 서브넷에 브로드캐스팅한다. 랑데부 피어 역시 서비스 프로파일 목록을 저장하고 자신의 서브넷에 서비스 프로파일 목록을 브로드캐스팅한다.

3.3 서비스 목록 요청/응답

랑데부 피어는 네트워크 상에 존재하는 모든 서비스 목록을 가지고 있는 피어이므로 P2P 네트워크에 참여하는 피어들은 랑데부 피어에 접속하여 서비스 프로파일 목록을 획득한다. 랑데부 피어의 주소는 각 피어들을 셋업할 초기에 설정하여야 한다.

3.4 서비스 요청/실행 모델

서비스 소비자 피어는 서비스 제공자 피어에게 서비스의 실행을 요청할 수 있다. 서비스 제공자 피어는 서비스 호출과 매개 변수들을 포함하는 서비스 요청 메시지를 생성한 후 전송한다. 서비스 호출에 필요한 매개변수 등은 서비스 프로파일에 기록되어 있다. 정보 서비스 경우, 요청 메시지를 생성한 후 서비스 제공자에게 전송하고 정보를 얻는 것으로 요청이 완료되지만 제어 서비스의 경우에는 추가적인 과정이 필요하다. 서비스 프로파일에 명시된 매개 변수 형식에 맞도록 메시지를 생성하고 제어 서비스를 요청한다. 예를 들어 프린터 제어 서비스 실행 시 첨부 파일을 요구하는 경우 등과 같이 서비스 실행에 추가적인 데이터가 필요한 경우, 서비스 제공자 피어 측에 첨부 파일의 전송이 완료되어야 한다. 서비스 제공자 피어는 서비스를 실행하고 실행 결과를 포함한 응답 메시지를 서비스 소비자 피어에게 전달한다. 그림 6은 서비스 요청 및 응답에 대한 모델을 표현한 그림이다.

3.5 서비스 프로파일 및 서비스 호출 메시지

서비스 프로파일은 서비스 제공자의 고유 식별자(ID), 서비스 이름, 매개 변수, 메시지를 통신을 위한 파이프 정보를 포함한다. 서비스 프로파일은 그림 7과 같이 3개의 영역으로 구분된다.

서비스 정보 영역은 서비스 이름, 서비스 제공자의 고유 식별자, 서비스 유형(정보 서비스 혹은 제어 서비스) 정보를 포함한다. 서비스 요청 정보 영역에는 매개 변수에 대한 정의를 포함하도록 설계하였다. 매개 변수 정보를 통해 서비스 소비자 피어의 사용자로부터 매개 변수를 입력받기 위한 사용자 인터페이스 정보를 제공한다. 통신 정보 영역에는 서비스 소비자와 제공자 사이의 데이터 통신 시 이용할 파이프 정보를 포함한다. 파이프 정보는 서비스 요청 정보의 전송 시 이용된다.

그림 8은 XML로 표현된 서비스 프로파일의 예이다. 서비스의 이름이 "베스트 셀러 정보 서비스"이고, 서비스 제공자 ID는 JXTA에서 임의로 생성된 고유한 ID로서 urn:jxta:uuid-59616261646162614E50472050325033FFFC4E64814C4DCCA9DCF73D433530FF04이다.

<ServiceRequestForm> 태그는 서비스의 호출 시 필요한 매개 변수들과 그에 대한 설명들을 포함하며, 사용자 화면에 출력하고 매개 변수를 입력받기 위해 제목, 레이블, 버튼, 텍스트 창 등의 요소를 <Title>, <Label>, <Button>, <TextBox> 태그로 각각 정의하였다. 소비자 측에서 <ServiceRequestForm> 태그의 속성들을 분석하여 화면에 UI를 출력하고 사용자로부터 입력을 받아 서비스를 호출한다. 현재 그림 8의 서비스는 톱 순위 번호를 받아 그에 해당하는 책 목록을 전달해주는 서비스이다. 그림 9는 그림 8의

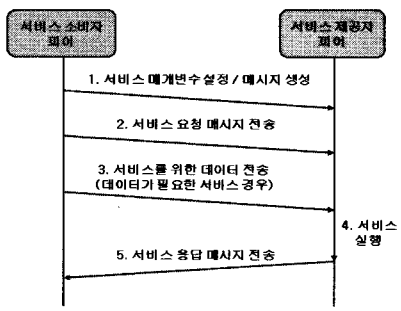


그림 6. 서비스 요청/응답 모델

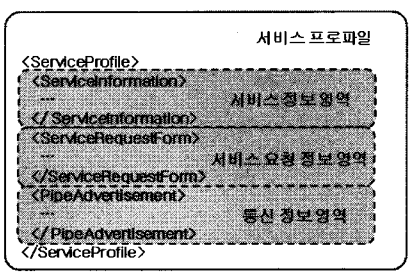


그림 7. 서비스 프로파일의 구조

```
<?xml version="1.0" encoding="euc-kr"?>
<ServiceProfile>
<ServiceInformation>
 <Type>2</Type>
 <Name>베스트 셀러 정보 서비스</Name>
 <SpID>urn:jxta:uuid-A2574815717240FCBBEDCEC34F6475A205</SpID>
</ServiceInformation>
<ServiceRequestForm>
 <Title>Default Service Request From</Title>
 <Label>
  <Name>Title</Name>
  <PositionX>0</PositionX>
  <PositionY>0</PositionY>
  <FontSize>20</FontSize>
  <Width>300</Width>
  <Height>30</Height>
  <Color>BLACK</Color>
  <Value>베스트 셀러 정보 서비스</Value>
 </Label>
 <Label>
  <Name>Name</Name>
  <PositionX>0</PositionX>
  <PositionY>40</PositionY>
  <FontSize>12</FontSize>
  <Width>300</Width>
  <Height>30</Height>
  <Color>BLACK</Color>
  <Value>순위 개수 :</Value>
 </Label>
 <TextBox>
  <Name>TopNumber</Name>
  <PositionX>75</PositionX>
  <PositionY>40</PositionY>
  <Width>30</Width>
  <Height>20</Height>
  <Color>WHITE</Color>
 </TextBox>
 <Button>
  <Name>서비스 요청</Name>
  <PositionX>0</PositionX>
  <PositionY>90</PositionY>
  <Width>150</Width>
  <Height>20</Height>
  <Color>Black</Color>
  <ActionType>4</ActionType>
 </Button>
</ServiceRequestForm>
<jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
 <Id>urn:jxta:uuid-59616261646162614E50472050325033ED4CD2FD91B45218CDCEB2D6332C65B04</Id>
 <Type>JxtaUnicast</Type>
 <Name>InputPipeInfo</Name>
</jxta:PipeAdvertisement>
<Peer>
 <Name>SP</Name>
 <ID>urn:jxta:uuid-59616261646162614A787461503250333CDF199100F14B9A9F46E0196947F11B03</ID>
</Peer>
</ServiceProfile>
```

그림 8. 서비스 프로파일의 예

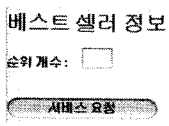


그림 9. 서비스 소비자의 서비스 요청 화면

<ServiceRequestForm> 태그에 포함된 UI 요소들이 출력된 서비스 소비자 컴퓨터의 화면이다. 여기서 사용자가 톱 순위 번호를 입력하고 "서비스 요청" 버튼을 클릭하면 서비스의 호출이 이루어진다.

3.6 메시지

JXTA는 메시지 기반 통신을 수행하며 메시지는 XML 형식으로 피어간 통신의 기본 단위이다. 메시지를 표 2와 같이 5 가지로 설계하였다. 이 중 DRM, DQM 메시지는 JXTA 플랫폼에서 제공되는 기본 메시지를 활용하였으며, 나머지 메시지는 본 논문에서 추가적으로 정의하였다.

그림 10은 서비스 요청 메시지의 한 예로서 사용자가 그림 9에서 순위개수에 3을 입력하였을 때 톱 순위 3위까지의 베스트 셀러 정보를 요청하는 서비스

표 2. 메시지 종류

메시지	설명
서비스 프로파일 갱신 메시지(SPUM)	서비스 제공자 피어에서의 서비스 변동 사항을 포함
서비스 프로파일 발견 메시지(DRM)	랑데부 피어에 접속하여 서비스 프로파일 목록을 요청하는 메시지
서비스 프로파일 발견 응답 메시지 (DQM)	서비스 프로파일 발견 메시지에 대한 응답으로 서비스 프로파일 목록을 포함한 메시지
서비스 요청 메시지 (SRQM)	서비스 제공자 피어에게 서비스 실행을 요청하는 메시지
서비스 응답 메시지 (SRSM)	서비스 실행에 대한 응답 메시지

```
<?xml version="1.0" encoding="euc-kr"?>
<ServiceRequest>
 <Type>2</Type>
 <ServiceName>베스트 셀러 정보 서비스</ServiceName>
 <ServiceID>urn:jxta:uuid-A2574815717240FCBBEDCEC34F6475A205</ServiceID>
 <Peer>
  <Name>서비스소비자</Name>
  <ID>urn:jxta:uuid-59616261646162614A787461503250333A530B63C6869486182338856FBACA964803</ID>
 </Peer>
 <ParamMsg>
  <Name>TopNumber</Name>
  <Value>3</Value>
 </ParamMsg>
 <jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
  <Id>urn:jxta:uuid-59616261646162614E50472050325033396F1EC99D7C746698C988E70B970F5CD04</Id>
  <Type>JxtaUnicast</Type>
  <Name>InputPipeInfo</Name>
  <Desc>JxtaUnicast</Desc>
 </jxta:PipeAdvertisement>
 <SocketAdv>
  <jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
   <Id>urn:jxta:uuid-59616261646162614E50472050325033606873E3FAAB444A4BC588F8CACDA2A9104</Id>
   <Type>JxtaUnicast</Type>
   <Name>Socket</Name>
   <Desc>JxtaUnicast</Desc>
  </jxta:PipeAdvertisement>
 </SocketAdv>
</ServiceRequest>
```

그림 10. 서비스 요청 메시지 예

```
<?xml version="1.0" encoding="euc-kr"?>
<ServiceResponse>
 <ServiceName>베스트 셀러 정보 서비스</ServiceName>
 <ServiceID>urn:jxta:uuid-A2574815717240FCBBEDCEC34F6475A205</ServiceID>
 <FileName>1689800950.gif</FileName>
 <Contents>1 위. 하악하악 (이외수의생존법) | 국내수필 | 이외수 | 해냄 </Contents>
 <FileName>1282500884.gif</FileName>
 <Contents>2 위. 시크릿 | 처세 | 론다번 | 살림출판사(도)</Contents>
 <FileName>1282501005.gif</FileName>
 <Contents>3 위. 마지막강의 | 처세 | 제프리재슬로 | 살림출판사(도) </Contents>
</ServiceResponse>
```

그림 11. 서비스 응답 메시지 예

실행 요청 메시지이다. 그림 10에서 <Peer> 태그는 서비스 요청자 정보를 포함하며 <PipeAdvertisement> 태그는 서비스의 실행 결과 메시지를 받기 위한 파이프 정보이다. <SockAdv> 태그는 이진 데이터를 주고 받기 위한 소켓 파이프 정보를 포함한다.

그림 11은 그림 10의 서비스 요청에 대한 응답 메시지이다. <FileName>은 소켓으로 들어온 정보에 대한 파일 이름을 포함하며 서비스에서 제공하는 파일이 없는 경우에는 none으로 표시된다. <Contents>는 서비스 실행의 결과를 포함한다.

3.7 서비스 제공자 피어 구조

그림 12는 서비스 제공자 피어의 소프트웨어의 구조를 보여준다. 총 4 개의 계층으로 구성되며 이 중

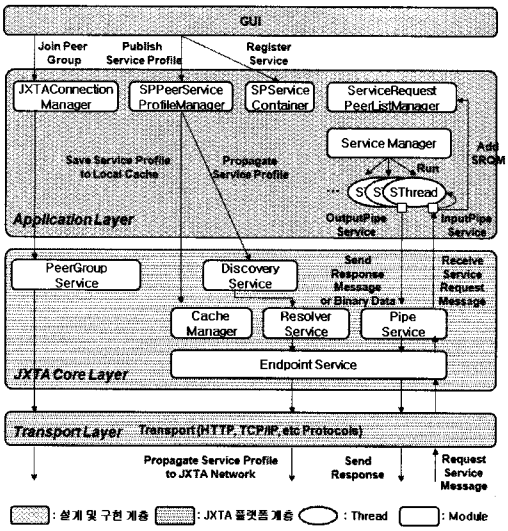


그림 12. 서비스 제공자 피어의 소프트웨어 구조

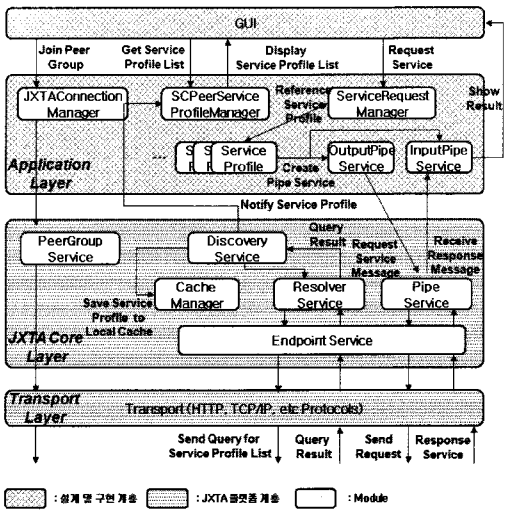


그림 13. 서비스 소비자 피어의 소프트웨어 구조

Applicaton Layer와 GUI를 설계 구현하였다. JXTA ConnectionManager 모듈은 최초 JXTA 플랫폼을 실행시키고, 본 논문에서 정의한 서비스 프로파일 메시지를 인식할 수 있도록 구현한 리졸버 서비스 핸들러를 JXTA Core Layer에 등록한다.

SPPeerServiceProfileManager는 제공하는 서비스 프로파일들을 저장하고 하위 계층의 Discovery Service를 이용하여 서비스를 외부에 알리는 등의 기능을 수행하며, SPServiceContainer는 서비스 제공자 피어의 사용자로부터 개발된 서비스들을 동적으로 추가, 수정, 제거하는 등의 서비스 관리 기능을 수행한다.

ServiceRequestPeerListManager는 서비스를 요청한 서비스 제공자 피어들의 목록을 관리하며 ServiceManager는 서비스 초기화 및 실행을 관리하고 실제 서비스를 실행시키는 (스)레드들을 관리하게 된다. 서비스 당 하나의 독립된 스레드가 실행을 담당하며 그 결과를 하위 계층의 PipeService를 통해 서비스 소비자 피어에게 전달한다. 서비스 소비자 피어로부터 서비스 실행 요청 메시지가 전송되어 오면 PipeService를 통해 ServiceRequestPeerListManager에게 전달되며 즉각 서비스 실행이 시작된다.

3.8 서비스 소비자 피어 구조

그림 13은 서비스 소비자 피어의 소프트웨어 구조를 보여준다.

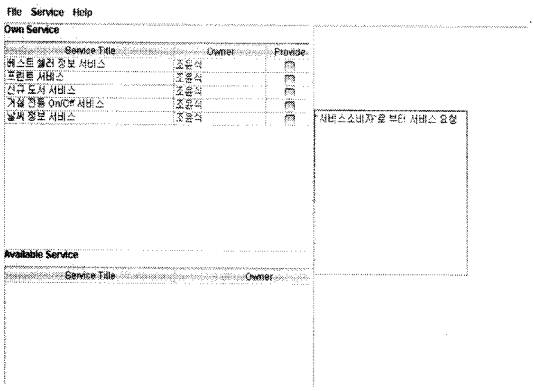
JXTAConnectionManager는 서비스 제공자 피어와 동일하다. SCPeerServiceProfileManager는 랑데부 피어 또는 동일한 서브넷 내의 서비스 제공자 피어로부터 전송받은 서비스 프로파일들을 관리한다. GUI 모듈을 통해 사용자에게 서비스 프로파일 목록을 출력한다. ServiceRequestManager는 사용자로부터 받은 매개변수 값과 함께 서비스를 요청한다. Output PipeService 모듈을 통해 서비스 실행 요청 메시지를 전송하고 실행의 결과는 InputPipeService 모듈을 통해 전달받고 GUI를 통해 화면에 출력한다.

Ⅳ. 프로토타입 구현 및 성능 평가

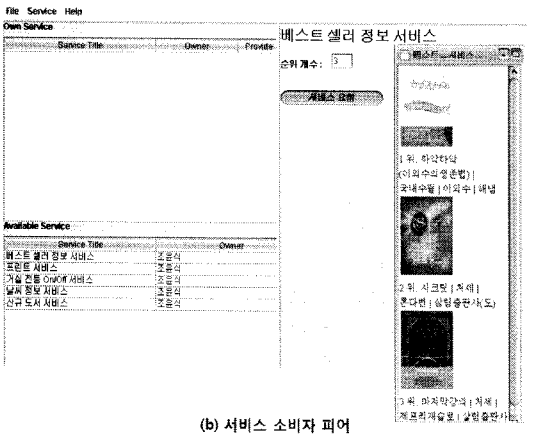
4.1 프로토타입 구현 및 실행

구현된 프로토타입의 실행 화면은 그림 14와 같다. JAVA JDK 6.0과 JXTA 2.4.1 라이브러리를 사용하여 윈도우즈 XP를 기반으로 하는 PC 환경에서 구현되었다. 릴레이 피어와 랑데부 피어가 작동하도록 설치하였다. 서비스 제공자 피어가 JXTA 네트워크에 참여할 처음에 자신이 가진 서비스를 광고하여 서비스 소비자 피어들이 이를 알게 된다. 그림 14(a)에는 서비스 제공자가 제공하는 서비스의 목록이 출력된 예이다. 그림 9, 10의 경우처럼"베스트 셀러 정보 서비스"를 요청하는 서비스 요청이 있을 때 오른쪽 박스에 요청이 있음을 표시하는 메시지를 출력하였다.

그림 14(b)의 실행 화면에서 서비스 소비자가 발견(service discovery)한 이용 가능한 서비스 목록을 출력하였다. 여기서 서비스를 선택하게 되면 그림 14(b)와 같이 "베스트 셀러 정보 서비스"호출을 위한 UI가 출력된다. 서비스 실행을 위해 순위 개수를 기입하고 서비스 요청 버튼을 클릭하면 서비스 제공자로부터 베스트 셀러 목록을 받아 그림 14(b)처럼 화면에 출력한다.



(a) 서비스 제공자 피어



(b) 서비스 소비자 피어

그림 14. 프로토타입의 실행 화면

4.2 성능 평가

4.2.1 실험 구성

본 성능 평가의 목적은 서비스 푸시 시스템의 가장 기본적인 지연 시간에 대한 추이를 일차적으로 분석하고 이해하는데 있다. 그러므로 본 논문에서는 그림 15와 같이 기본적인 테스트 시스템을 구성하였다. 실험에 이용된 PC는 모두 동일한 것으로 Intel Duo Core 6300의 1.86GHz CPU와 1 GB RAM, 100Mbps LAN의 사양을 가진다.

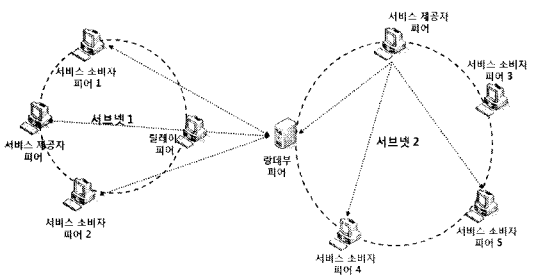


그림 15. 실험 환경

4.2.2 성능 지수

본 논문에서 제시된 시스템의 성능은 서비스를 가진 제공자가 서비스를 배포할 때 얼마나 빨리 전파되는지 그리고 서비스 소비자가 서비스 실행을 요청하였을 때 얼마나 빨리 실행되는지에 달려 있다. 다음과 같은 성능 지수를 정의한다.

· T_{sp}

서비스 제공자 피어가 서비스 프로파일을 전파할 때, 서비스 소비자 피어들에게 서비스 프로파일이 도착할 때까지의 전파 지연 시간(Service Profile Pushing Delay).

· T_{isd}

서비스 소비자 피어가 초기에 랑데부 피어에 접속하여 서비스 프로파일 목록을 획득까지의 경과 시간(Initial Service Profile Discovery Delay).

· T_{rse}

서비스 실행 성능으로서 서비스 소비자 피어의 사용자가 서비스 실행을 요구한 시점에서 서비스 제공자에서 서비스 실행 바로 직전까지의 경과 시간(Remote Service Execution).

4.2.3 실험 1 - T_{sp}

서비스 프로파일의 전파는 서비스의 추가, 수정, 삭제 등의 경우에 발생한다. 이들의 과정은 거의 동일하므로 본 실험은 새로운 서비스를 추가하는 경우에 대해서만 이루어졌으며, 서비스 프로파일을 포함하는 서비스 갱신 메시지를 생성하고 광고하고 이에 따라 각 피어들이 서비스 갱신 메시지를 받을 때까지의 지연 시간을 측정하였다. 표 3은 실험을 위해 설

표 3. 실험 1의 파라미터

파라미터	요 소	
메시지	서비스 추가 메시지	
생성	메시지 생성 20회	
T_{sp}	$T_{sp} = \frac{\sum_{i=0}^{C_{exp}} (T_{recvspum} - T_{sendspum})}{C_{exp}}$	
	C_{exp}	총 실험 횟수 (10회)
	$T_{recvspum}$	메시지 도착 시각
	$T_{sendspum}$	메시지 전송 시각

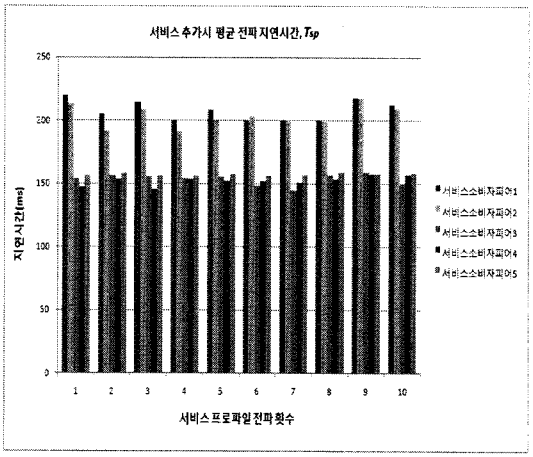


그림 16. 서비스 추가에 따른 T_{sp}

정된 파라미터이며 그림 16은 서비스 프로파일 전파의 평균 지연 시간을 측정한 결과이다.

정확한 측정을 위해서 한국표준과학연구원의 인터넷 시간 서버인 time.kriss.re.kr에 시간을 요청하는 방식으로 각 PC 간 시계를 맞추었다.

실험 결과, 서비스 프로파일 전파 시간은 크지 않은 것으로 평가되었다. 서비스 소비자 피어 1, 서비스 소비자 피어 2는 서비스 제공자 피어와 다른 서브넷에 위치하므로 서비스 소비자 피어 3, 4, 5에 비해 서비스 프로파일 수신이 늦어짐을 보이고 있다. 그 이유는 서비스 소비자 피어 1, 2가 서비스 제공자와 물리적으로 다른 네트워크에 존재하기 때문에 메시지가 전달되려면 JXTA 엔드-포인트 라우팅 프로토콜에 의존하여 그들의 경로를 획득해야 하기 때문이다. 이들 간에는 약 50-60ms 정도의 시간 차이가 측정되었다.

4.2.4 실험 2 - T_{isd}

서비스 소비자 피어가 랑데부 피어에 접속하여 현재 네트워크 상에 존재하고 있는 서비스 목록을 획득하는데 지연되는 시간을 측정한다. 임의의 피어가 P2P 네트워크에 처음으로 참여할 때 얼마나 빠른 시간 내에 서비스 프로파일을 획득하는지 성능을 측정하는 실험이다. 표 4는 실험 2를 위한 파라미터이며 그림 17은 실험의 결과이다.

표 4. 실험 2의 파라미터

파라미터	요 소	
프로파일	목록 내 서비스 프로파일 : 1-50개	
T_{isd}	$T_{isd} = \frac{\sum_{i=0}^{Cexp} (T_{recvspl} - T_{reqspl})}{C_{exp}}$	
	C_{exp}	총 실험 횟수 (20회)
	$T_{recvspl}$	서비스 프로파일 목록이 랑데부 피어로부터 요청한 피어로 전달 완료된 시각
	T_{reqspl}	임의의 피어가 랑데부 피어에게 서비스 프로파일 목록을 요청한 시각

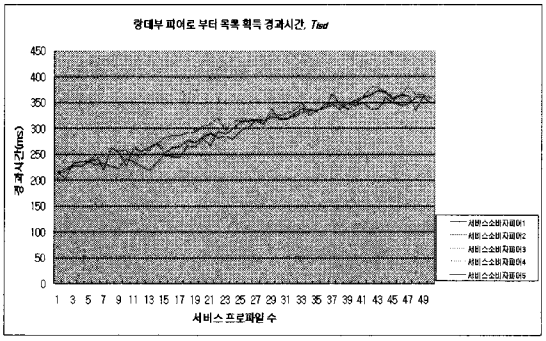


그림 17. T_{isd} 결과

그림 17은 목록에 포함된 서비스 프로파일의 개수를 달리하면서, 서비스 프로파일 목록 획득까지의 경과 시간을 측정한 결과를 보여준다. 서비스 소비자 피어들은 위치에 상관없이 모두 유사한 결과를 보여주고 있다. 이는 모든 피어들은 랑데부 피어의 IP 주소로 직접 접속하기 때문이며, 서비스 갱신 메시지 전파의 경우와 달리 JXTA의 엔드-포인트 라우팅 프로토콜에 의존하지 않기 때문이다. 또한 목록에 포함된 서비스 프로파일의 개수가 증가할수록 경과 시간도 같이 증가한다.

4.2.5 실험 3 - T_{rse}

서비스 요청의 실행 시간을 측정하였다. 표 5와 같이 실험이 진행되었으며 결과는 그림 18과 같다.

실험 결과, 서비스 제공자 피어와 동일한 서브넷에 존재하는 피어 3,4,5의 경우, 피어 1.2에 비해 약

표 5. 실험 3의 파라미터

파라미터	요 소	
메시지	서비스 요청, 서비스 응답	
T_{rse}	$T_{rse} = \frac{\sum_{i=0}^{Cexp} (T_{sendsrqm} - T_{recvsrsm})}{C_{exp}}$	
	C_{exp}	총 실험 횟수 (100회)
	$T_{sendsrqm}$	요청 메시지의 전송 시각
	$T_{rcvdsrsm}$	응답 메시지가 도착한 시각

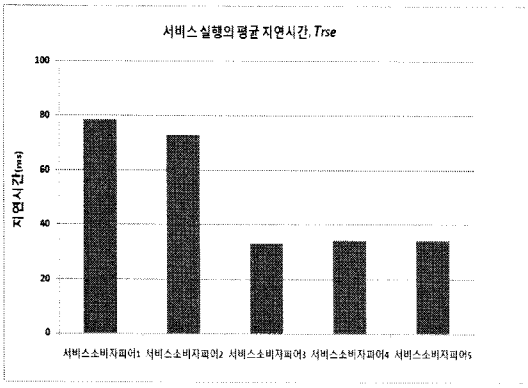


그림 18. T_{rse} 결과

40-50ms 정도 빨리 응답을 받는 것으로 측정되었다. 이는 서비스 프로파일 갱신 메시지를 전파하는 경우와 유사한 이유에서 비롯된다.

4.3 성능 평가 결과

본 실험을 통해 프로토타입 시스템 상에서 서비스 푸시에 따른 서비스 프로파일 전파 시간, 서비스 프로파일 획득 시간, 서비스 요청의 실행 시간을 실제 측정하고 평가하였다. 한 피어가 네트워크에 처음으로 접속하였을 때 서비스 프로파일을 획득하는 시간은 피어의 네트워크 위치에 거의 관계없이 유사하였으나, 서비스 프로파일을 전파하는 시간과 서비스를 요청하고 응답을 얻는 시간은 JXTA의 구조적인 이유로 서비스 소비자 피어와 서비스 제공자 피어가 서로 다른 서브넷에 위치하고 있으면 동일한 서브넷에 있는 경우에 비해 지연 시간이 증가하였다. 만일 많은 서브넷으로 구성되는 네트워크의 경우라면 릴레이 피어를 경유하며 메시지가 전파되는데 릴레이 피어의 개수만큼 지연시간이 커지게 되는 문제점이 존재한다. 그러나 실험 3의 결과를 보면 대체로 한 서브넷을 거쳐가는 경우에 약 40-50ms의 차이를 보인다면, 서브넷이 10개라고 하더라도 약 500ms 내로서 절대 시간에 있어 큰 문제가 존재한다고 보이지 않는다.

그러나 본 성능 평가는 매우 기초적이고 기본적인 테스트 환경에서 실험되었기 때문에 성능 결과에 한계가 있다. 즉 본 논문에서 측정된 시간 값이나 예측 결과가 대규모의 피어를 가진 경우와 다양한 작업 부하가 존재하는 실제 상황에서 발생하는 것과 차이가 있을 수 있다. 그러므로 추후 연구를 통해 서비스 소비자와 서비스 사용자의 네트워크 진입 및 탈퇴, 서비스 소비 비율의 변화, 서비스 푸시나 서비스 소비의 동시적 발생, 이종 네트워크의 연결로 인한 상황 등 실제 상황에 따른 작업 부하를 세밀하게 설계하고 성능을 평가 분석할 필요성이 있다.

V. 결 론

본 논문은 서비스 푸시를 통해 서비스의 발견 시간을 줄이고 서비스를 즉각 이용할 수 있는 방법으로 제안하고 P2P 네트워크 구조를 가지는 JXTA 기반으로 프로토타입 시스템을 설계 및 구현한 내용을 소개하였다. 구현된 프로토타입 시스템 상에서 성능을 측정한 결과 네트워크 상에 처음으로 접속한 피어가 서비스 프로파일을 발견하는 시간은 매우 짧았으며 새로운 서비스가 발생하고 이것이 서비스 소비자들에게 전파되는 시간도 매우 짧은 시간으로 측정되었다. 또한 서비스를 요청하고 응답을 얻는데 걸린 시간도 몇십 ms의 범위로서 사용상에 큰 문제는 없는 것으로 평가되었다. 그러므로 본 논문에서 제안된 서비스 푸싱은 소규모의 로칼 네트워크에서 활용성이 있다고 평가된다. 그러나 본 논문에서 시행된 실험은 작은 범위의 네트워크로 구성되었기 때문에 보다 광범위한 규모의 네트워크 상에서 다양한 실험을 할 필요성이 여전히 남아 있다.

참 고 문 헌

[1] Bernard Traversat, Mohamed Abdelaziz, Dave Doolin, Mike Duigou, Jean-Christophe Hugly, Eric Pouyoul, "Project JXTA-C:Enable a Web Things", in Proceeding of the 36th Hawaii International Conference on System Sciences 2002 IEEE.

[2] KATO Tomoya, YOKOI Shigeki, "Application of P2P Technology to Marketing", Proceedings of the 2003 International Conference on Cyberworlds IEEE.

[3] Haoyi Wan, Norihiro Ishikawa, "Design and Implementation of a Simulator for Peer-to-Peer Networks:Optimal Sim", 2005 PACRIM IEEE Pacific Rim Conference on 24-26 Aug 2005.

[4] Granville, L.Z.; da Rosa, D.M.; Panisson, A.; Melchiors, C.; Almeida, M.J.B.; Tarouco, L.M.R.; "Managing Computer Networks Using Peer-to-Peer Technologies", Communications Magazine, IEEE Volume 43, Issue 10, Oct. 2005 Page(s):62-68.

[5] Wei Zheng, Sheng Zhang, Yi Ouyang, Filla Makedon, James Ford, "Node Clustering Based on Link Delay in P2P Networks", Symposium on Applied Computing archive Proceedings of the 2005 ACM symposium on Applied computing table of contents Santa Fe, New Mexico.

[6] JXTA, http://www.jxta.org

[7] M. Bisignano, G.D. Modica, O. Tomarchio, JMibiPeer:A middleware for mobile peer-to-peer computing in MANETS, Proc. of the 25th IEEE International conference on Distributed Computing Systems Workshops.

[7] Xie Dan, Ying Shi, Zhang Tao, Jia Xiang Yang, Liang Zao-Qing, Yao Jun-Feng, "An Approach for Describing SOA", Wireless Communications, Networking and Mobile Computing, 2006 WiCOM 2006 International Conference on 22-24 Sept.2006 IEEE.

[8] Nickull Duane, 『Service Oriented Architecture』, Adobe Systems 2005.

[9] JINI, http://www.jini.org/

[10] UPnP, http://www.upnp.org/

[11] Sayre, R, "Atom: the standard in syndication", Internet Computing, IEEE Volume 9, Issue 4, July-Aug. 2005 Page(s):71-78.

[12] Scholtes, I., Gorgen, D., Gratz, P., "Web Service Interface Syndication and its Application to a Collaborative Weblog Search", Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006. International Conference on 17-20 Nov. 2006 Page(s):1-5.

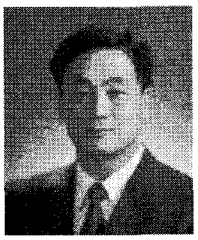
조 윤 식 (Yoon Sik Cho) 정회원

2005년 2월 한성대학교 컴퓨터공학과 졸업

2008년 8월 한성대학교 컴퓨터공학과 석사

<관심분야> 분산시스템, P2P

정 인 환 (In Hwan Jung) 정회원



1984년 2월 한양대학교 원자력공학과 졸업

2000년 2월 한국과학기술원 정보및통신공학과 박사

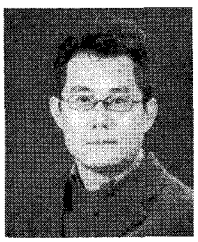
1985년~1998년 삼성전자 시스템사업부 수석 연구원

2007년~2008년 University of Colorado 방문 교수

2001년~현재 한성대학교 컴퓨터공학과 부교수

<관심분야> 멀티미디어 통신, 분산 처리

황 기 태 (Kitae Hwang) 정회원



1986년 2월 서울대학교 컴퓨터공학과 졸업

1988년 2월 서울대학교 컴퓨터공학과 석사

1994년 2월 서울대학교 컴퓨터공학과 박사

2000년~2001년 University of California, Irvine 방문 교수

1994년~현재 한성대학교 컴퓨터공학과 교수

<관심분야> 유비쿼터스 컴퓨팅, 분산 시스템