

이동 에이전트 기반의 분산 혼합형 Peer-to-Peer 네트워킹[†]

덕성여자대학교 정원호**

ICANTEK 강미연*

1. 서 론

유무선 인터넷 기술의 발달로, 인터넷을 통한 개인 정보의 효율적인 공유와 교환을 위한 시스템들이 속속 등장하고 있으며, 이러한 시스템들은 분산 정보의 공유를 위한 네트워킹 기술로 대부분 P2P를 표방하고 있다. P2P(Peer-to-Peer)의 개념은 네트워크로 연결된 양측의 클라이언트가 중앙의 서버를 경유하여 연결되거나 또는 직접적으로 연결되어 하드웨어, 소프트웨어, 콘텐츠 등 양측의 자원을 공유하는 것을 일컫는다. 이러한 P2P 등장 배경에는 기술적, 환경적 요인으로 광대역 인터넷의 보급과 이로 인한 고속 인터넷 서비스의 대중화 등이 있으며, 특히 PC 성능의 급속한 향상은 기존의 클라이언트-서버 방식의 서버 집중식 모델의 한계와 고비용을 타파시켜 주고 P2P를 성장시킨 주요한 원인이 될 수 있다. 이러한 P2P 서비스는 크게 2가지 방식이 존재하고 있다. 하나는 상대자의 접속 상태 및 요청 자료의 검색을 중앙 서버를 이용하여 제공해 주는 혼합형(hybrid) P2P 방식이고, 다른 하나는 최초의 상대자 정보를 가지는 피어들이 상호간에 IP 주소 리스트 등 개인 정보를 공유하여 서버 없이 직접 연결하는 순수(pure) P2P 방식이다. 혼합형의 경우에도 접속 및 검색 단계 이후는 순수 P2P와 동일하게 피어들 간에 직접 정보를 교환한다. 혼합형의 대표적인 예로, 냅스터(Napster)와 소리바다 등을 들 수 있고, 순수 방식으로는 누텔라(Gnutella) [1]와 Freenet [2]이 대표적이다. 혼합형은 중앙 서버에 공유 정보가 집중되어 있어, 서버가 정보의 중개자 역할을 하고 있기 때문에 저작권 문제에서 자유로울 수 없고, 피어의 등록 및 자료 검색에 대한 경로 정보를 하나의 중앙 서버에 저장하고 있어, 자료 검색이 중앙 서버에서 수행됨으로 인해서, 많은 수의 피어에 의한 자료

검색 시, 서버로의 부하가 편중되는 현상이 발생할 수 있다. 또한 서버 결함 발생 시, 전체 P2P 네트워크가 정지할 수 있다는 단점도 가지고 있다 [3-6]. 반면, 특정 중앙 서버가 존재하지 않는 순수 P2P의 경우에는 검색이 각 피어에 의해 자체적으로 일어나므로, 노드 결함에 의한 영향 범위는 최소화 되지만, 자료 검색 시, TTL (Time-To-Live)의 값에 따라, 기하급수적인 TCP 접속 수의 증가와 동시에 탐색 및 정보의 전달을 위한 TCP 접속을 해당 트랜잭션이 완료될 때까지 유지하고 있어야 하므로, 커다란 전달 지연을 야기시킬 수 있으며 이는 네트워크 scalability에 대한 난제로 남아 있다. 또한, 메시지 전달이 트리 형태로 전파됨에 따라, 스팸 메시지가 빈번하게 발생하며 네트워크 대역폭의 낭비가 심하여, 검색이 비효율적이라는 단점을 나타내게 되어 실제로 사용하기는 어렵다 [3]. 그리하여 혼합형 시스템이 중앙집중식으로 중앙 서버 결함에 의한 전체 네트워크 정지 및 중앙 서버에 대한 부하 편중의 문제점을 가지고 있으나, 검색의 효율성이 좋아 대부분의 상용 P2P 시스템은 혼합형을 사용하고 있는 실정이며, 순수 P2P의 경우, 검색의 비효율성에 관한 문제점 보완을 위해 이동 에이전트 등을 사용하는 연구들이 진행 중이다 [7-10]. 최초의 P2P 시스템은 특정 파일 공유를 목적으로 탄생하였으며 최근에는 서로 다른 응용을 가진 다양한 P2P 시스템들이 개발되고 상용화되고 있으며, 또한 PC 그리드 상에서의 효율적인 자원 검색 및 수집을 위해서도 이동 에이전트를 사용하여 연구하고 있다 [11].

본 논문에서는, 효율적인 혼합형 P2P를 위하여, **다중 서버를 사용함으로써 결함 허용성(fault-tolerance)을 지니도록 하고 동시에 서버의 부하도 줄일 수 있는 새로운 P2P 프레임워크가 제안된다.** 각 서버는 최소의 정보인 IP 주소만 보관하고 있어, 저작권 문제를 해결하고, 자원의 검색 및 전달을 위하여 이동 에이전트를 이용함으로써 전달 지연을 줄이고 있다. 제안된 프레임워크는 혼합형과 순수 P2P 모두를 지원할 수 있다. 기존의 시스템이 작업의 완료시까지 TCP 연결을 지속해야 하는

* 정희원

** 중신회원

[†] 본 연구는 한국과학재단 2003년도 목적기초연구(R06-2002-003-01003-0)지원으로 수행되었음.

단점을 가지고 있는 반면, 설계된 시스템은 일단 에이전트 이동이 완료되면 작업이 완료될 때까지 TCP 접속을 유지할 필요가 없어, 네트워크 대역폭 절감 및 네트워크의 부분적 분리 현상에 정상 동작을 유지할 수 있게 하는 장점을 가지고 있다.

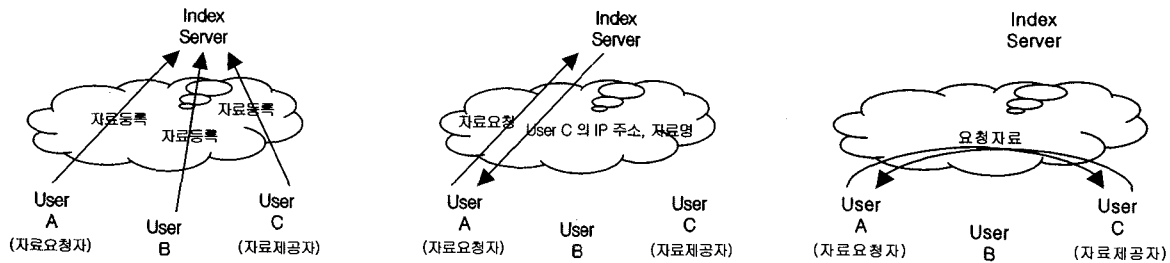
본 논문에서는, 서버 역할을 하는 피어를 슈퍼 피어(super-peer)라 하고, 단지 클라이언트로서만 동작하는 피어를 일반 피어(general peer)라 한다. 단순히 피어라고 하면 슈퍼 피어와 일반 피어를 총칭하는 것으로 한다. 제안된 P2P 프레임워크는 기존의 혼합형 P2P가 가지는 단점인, 슈퍼 피어 결합에 의한 시스템 동작 중지 및 많은 피어들이 존재하는 경우 슈퍼 피어에 걸리는 검색 부하의 편중 문제를 해결할 수 있다는 장점을 가지고 있다. 이는, 2개 이상의 다중 슈퍼 피어를 가질 수 있어 결합 허용 특성을 가지고 있으며, 자료 검색이 서버에 의해서가 아닌 각 피어에 의해 생성된 이동 에이전트에 의해 이루어지므로, 슈퍼 피어의 검색 부하를 최소화시키고 있기 때문이다. 또한, 많은 수의 일반 피어들이 존재하는 경우, 각 슈퍼 피어 당 관리할 수 있는 일반 피어의 수를 제약할 수 있도록 하여, 슈퍼 피어의 관리 부하를 더욱 줄여줄 수 있도록 하고 있다. 특히, 정보 탐색의 경우 기존의 시스템에서는 사용하지 않는 이동 에이전트를 사용하고 있어, 네트워크 대역폭을 효율적으로 사용하고 있다는 장점도 가지고 있다. 본 논문에서는 슈퍼 피어의 부하 감소와 결합 허용 특성에 초점을 맞추어

시스템을 기술하기로 한다. 기타 유용한 기능들은 간략하게 언급하며, 제시된 참고문헌을 참고 바란다.

2. 관련 연구

2.1 혼합형(Hybrid) P2P

그림 1은 혼합형 P2P의 서비스 단계를 보여주고 있다. 먼저 사용자가 P2P 클라이언트 프로그램을 다운받아 실행하면 자신이 서비스 할 디렉토리의 위치를 입력하게 한다. 중앙 서버는 사용자가 오픈한 디렉토리 내의 파일을 분석해 리스트(index)를 저장한다. 그리하여 UserA 사용자로부터 자료에 대한 요청이 들어올 때 중앙 서버는 저장한 리스트를 검색한 결과와 UserA의 요청 자료를 보관하고 있는 UserC의 IP 주소를 제공한다. 이때 중앙 서버는 UserC의 온라인 접속 여부 및 회선 속도 등 다양한 부가정보를 제공할 수 있다. 이 과정까지는 중앙 서버가 개입하게 된다. 이후의 과정은 UserA가 직접 UserC의 PC에 접속하여 P2P 방식으로 원하는 자료를 다운로드 하게 된다. 이와 같이 중앙서버의 개입이 필요한 혼합형 P2P 방식은 검색을 위한 인덱스 서비스를 지원하고 기타 사용자의 온라인 접속 여부 등 부가정보를 제공한다. 이러한 혼합형 P2P는 기존의 웹 방식의 서비스에 비해 중앙 서버의 역할이 최소화되었기 때문에 서버의 부담이 없으면서도 많은 사용자를 수용할 수 있다.



(a) 중앙 서버에 주소 및 공유자료 등록

(b) 필요한 자료 요청 및 검색

(c) 자료 전송

그림 1 냅스터 유형의 혼합형 P2P의 서비스 단계

2.2 순수(Pure) P2P

순수 P2P의 대표적인 모델은 누텔라이다. 냅스터의 경우 사용자의 음악 파일에 대한 정보를 중앙 서버에서 관리하고 있으나 누텔라의 경우에는 중앙에서 관리하는 정보가 없다. 누텔라의 경우 사용자 노드에 설치하는 프로그램 자체가 검색 기능을 내장하고 있어 보유하고 있는 IP 주소를 가진 해당 컴퓨터에 직접 연결하여 자료를 찾아낸다. 예를 들면(그림 2 참조) UserA가 알고 있는

UserB에게 자료를 요청하면 UserB의 P2P 프로그램은 자료를 보유하고 있는 경우 자신의 자료를 보내주고, 자료를 보유하고 있지 않은 경우, UserB에 설정된 다른 UserC에게 요청 자료에 대한 메시지를 전달하게 된다. 누텔라의 경우 메시지에 포함된 TTL(Time To Live) 값에 의해 메시지 전달의 깊이가 결정된다. 예를 들면 TTL이 3인 경우 UserB(TTL=1)가 알고 있는 UserC(TTL=2)에게 메시지를 전달하고 UserC는 UserD

(TTL=3)에게 메시지를 전달한다. UserD는 TTL이 3이므로 자신이 보유한 다른 이용자에게 메시지를 전달하지는 않는다. 만약 TTL을 무한대로 적용했다면 이러한 P2P 프로그램이 설치된 전세계의 사용자에게 메시지가 전달되게 되는 것이다. 그림 2에서는 UserB가 요청 자료를 보유하고 있지 않아 UserC에게 메시지를 보내고, UserC가 요청 자료를 보유하고 있다는 메시지를 UserA에게 전달한다. UserC로부터 요청 자료에 대한 보유를 확인한 UserA는 UserC의 노드에 접속하여 원하는 정

보를 다운로드 한다. 누텔라와 같은 순수 P2P 방식은 몇 가지 문제점을 가지고 있다. 첫째는 스팸의 남발이다. 각 P2P 프로그램이 10개의 P2P 프로그램이 설치된 IP 주소를 가지고 있고 처음 P2P 프로그램이 TTL이 6인 메시지를 보낸다면 10의 6승인 1백만 개의 메시지가 인터넷을 돌아다니게 된다. 둘째는 중앙 서버의 부재로 인하여 가입자에 대한 서비스 및 사용자간 정보 공유 시 발생할 수 있는 다양한 문제를 해결하기 어렵고, 정확한 검색 결과와 검색의 성능이 보장되지 않는다.

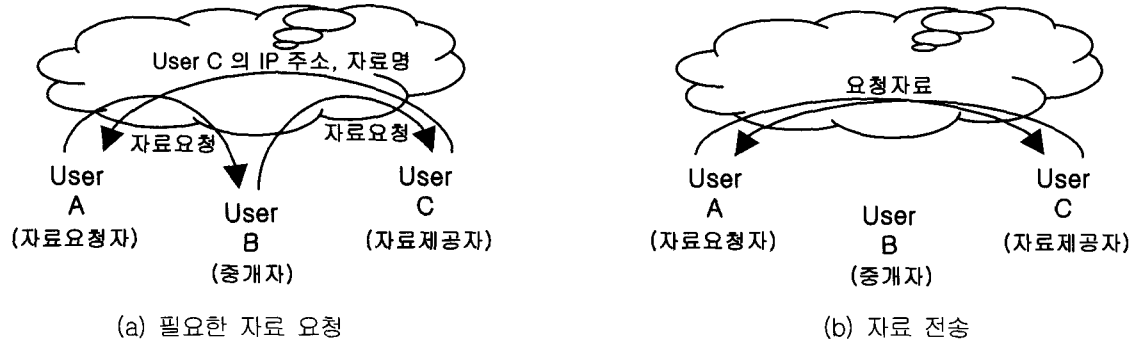


그림 2 누텔라 유형의 순수 P2P의 서비스 단계

2.3 이동 에이전트

이동 에이전트는 네트워크 상을 자율적으로 이동하면서 사용자 작업을 수행할 수 있는 프로그램 객체로, 네트워크 상에서의 다양한 분산 응용을 위해 비교적 쉽게 대처할 수 있을 만큼, 통합성과 유연성이 좋아, 향후 기대되는 분산 응용 기술 중의 하나로, 높은 비동기성, 비접속 동작성, 지능성 및 고장 감내성 등의 특성을 가지고 있다[12]. 이동 에이전트의 장점은 네트워크 부하와 지연의 감소와, 네트워크 오류 시 다른 시스템에 비해 신뢰도가 높다는 것을 들 수 있다. 왜냐하면, 이동 에이전트는 일단 목적하는 노드로의 이동이 이루어지고 나면, 결과물 응답 전까지, 대부분의 작업이 해당 노드에서 로컬하게 수행되므로, 네트워크의 비접속 상태에서 작업 수행이 가능하며, 작업 수행이 완료된 후 접속을 시도하여 결과물을 응답하기 때문이다. 그러므로, 이동 에이전트는 접속이 불안정하거나 부하가 높은 네트워크 환경에서 다른 분산 패러다임에 비해 트래픽을 줄이고 채널을 효율적으로 사용할 수 있는 기대되는 해결책 중의 하나라고 할 수 있다. 또한, 기존의 클라이언트/서버 그리고 애플릿/서블릿 개념을 통합하는 새로운 소프트웨어 패러다임이라 할 수 있어, 구현하기에 따라 이동 에이전트가 가지는 특성은 매우 다양하다고 할 수 있다. 이러한 기대로 인해 여러 형태의 이동 에이전트 프레임워크들이 Java, Tcl 등을 기반으로 개발되고 있다 [13-16]. 특히, Java 기반의 이동 에이전트는 Java

언어의 폭넓은 사용과 더불어, 그 응용의 폭이 넓어, 다양한 형태의 분산 정보 관리, 병렬 및 분산처리, 정보의 전송 및 공유 등, 그 응용 범위를 넓혀가고 있으며, 최근에는 SCM(Supply Chain Management)과 같은 전자상거래 분야에까지 이동 에이전트 적용을 위한 연구와 개발이 활발히 이루어지고 있다[17-21].

기존의 P2P 네트워크는 메시지 기반의 RPC(Remote Procedure Call)를 사용하는 고정 에이전트를 사용하고 있으며, 채널의 대역폭을 비효율적으로 사용함으로써, 작업 지연을 증가시키는 결과를 초래하고 있으나, 이동 에이전트는 작업 소프트웨어가 직접 작업을 수행하는 호스트로 이동하여 작업을 수행함으로써, 채널의 대역폭을 효율적으로 사용할 수 있도록 하여, 작업 지연을 감소시키는 효과를 내고 있다. 그러므로 부분적인 네트워크 분리(network partition)와 같은 문제 등이 발생하였을 경우 특히 유효하다. 즉, 그림 3에서 보여준 바와 같이, 고정 에이전트를 사용하는 경우, 정보 전달을 위한 채널이 작업이 시작하여 완료될 때까지 접속된 채로 유지되고 있어야 하는 반면((b), (c), (d) 동안 계속 접속이 유지되고 있어야 한다), 이동 에이전트를 사용하는 경우, 에이전트가 H2로 이동되는 동안만 접속을 유지하고 있어야 하며, 이동이 완료된 후에는 접속이 단절되어도 무방하며, 작업을 완료하고 그 결과를 전송할 필요가 있을 경우 접속을 요구하여 결과 정보를 전달하는 방식이기 때문이다((b), (d)에서만 접속이 유지되면 된다).

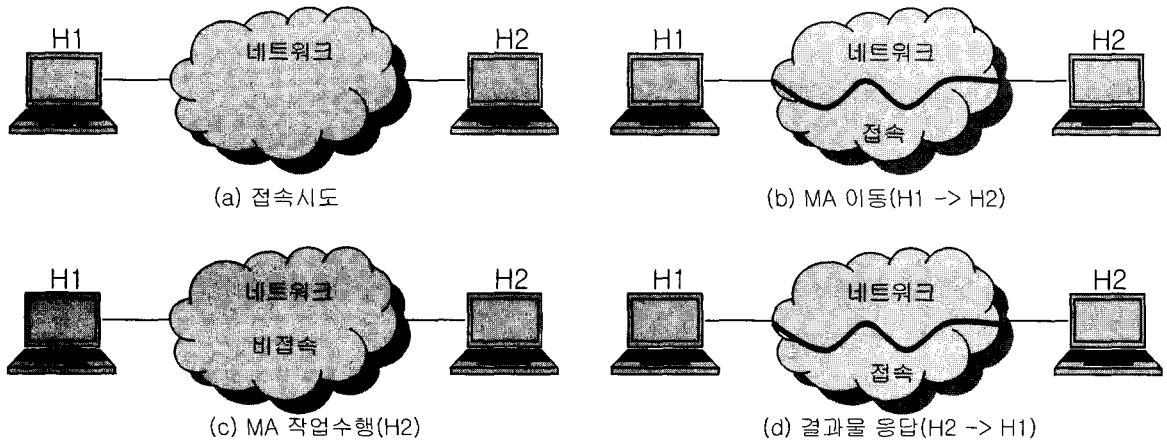


그림 3 이동 에이전트의 작업 수행에서의 채널 접속

표 1 상용 P2P 시스템의 예

사용 목적	서비스	서비스 구조	개발자/업체명	국내/국외
음악 파일공유	Napster	혼합형 P2P	Napster	국외
음악 파일공유	소리바다	혼합형 P2P	soribada	국내
파일공유	P2P웹	혼합형 P2P	SeeFriend	국내
파일공유	Gnutella	순수 P2P	NullSoft	국외
메신저	ICQ	혼합형 P2P	AOL	국외
전자상거래	오프너	혼합형 P2P	Open4u	국내
경매	Lightshare	혼합형 P2P	Lightshare	국외
소프트웨어 배포	Mycio	혼합형 P2P	Mycio	국외
공동작업	Groove	혼합형 P2P	Groove	국외
분산컴퓨팅	SETI@home	혼합형 P2P	UC Berkely	국외

3. 상용 P2P 시스템

최근에는 표 1에 보여준 바와 같이 다양한 응용을 표방하는 P2P 시스템들이 상용화되고 있는데, 본 장에서는 현재 상용화되어 각종 응용에 사용되고 있는 몇 가지 P2P 시스템을 소개한다. Open4u는 사용자가 Open4u 서버에 접속하여 P2P 프로그램인 오프너를 다운로드 받아 설치하고, 구매자의 경우 이 프로그램을 이용해 자신이 구매하고 싶은 물품의 사양을 적어 메시지를 보내면 중앙 서버인 Open4u 서버가 이 사양의 물품을 제공할 수 있는 판매자의 IP 주소를 구매자에게 전달한다. 이 정보를 가지고 구매자는 P2P 방식으로 각 판매자에게 견적 요구서를 보낸다. 견적 요구서를 받은 판매자는 견적서를 구매자에게 보내며 구매자를 여러 견적서 중 적합한 하나를 고르게 된다. Open4u의 서버는 최종 거래가 어느 조건으로 성립되었는지에 대한 정보만 확보하고 두 당사자간의 거래에는 개입하지 않는다.

버클리 주립대에서 주관하고 있는 SETI@home 프로젝트는 우주상에 떠다니는 아주 미세한 신호들을 분석

해서, 이들 중 생명체가 발산하는 신호가 있는지를 찾아, 외계인의 존재를 규명하는 프로젝트이다. 프로그램을 다운로드 받아 설치해 놓으면, 중앙 서버는 사용자가 컴퓨터를 사용하지 않을 때 일정한 데이터를 보내어 관련 계산을 한다. 계산된 데이터는 중앙 서버로 다시 모여서 통합, 처리되게 된다.

ICQ는 전세계 6200만 명의 사용자를 가지고 있는 대표적인 인스턴스 메신저이다. 사용자가 ICQ를 자신의 PC에 설치하고, 이메일 주소나 ID, ICQ 번호 등을 이용하여 통신할 상대자를 자신의 PC에 있는 ICQ 프로그램에 저장하면, 상대자가 허용할 경우 상대자 IP 주소를 이용하여 직접 메시지 통신할 수 있게 된다.

누텔라는 윈앰프로 유명한 NullSoft에서 만들어졌다. 누텔라 소프트웨어 패키지 중 하나를 웹을 통해 다운로드 받아 설치하여 컴퓨터를 servent(server+client)로 만든 후, 누텔라 네트워크 상의 또 다른 servent를 찾아서 서로 접촉한다. 이들 servent의 주소는 여러 웹 사이트, 채팅룸에서 찾을 수 있다. 누텔라는 이미지, 텍스트, 소프트웨어, MP3 등 여러 형태의 파일을 공유할 수 있다.

냅스터는 MP3 파일 교환 서비스를 제공한다. 냅스터는 회원들이 보유한 MP3 파일 목록을 만들어 제공하고 서로의 컴퓨터를 연결해 주는 서버를 운영한다. 냅스터의 소프트웨어를 다운로드 받아 인스톨하게 되면 자신의 컴퓨터에 공유 디렉토리가 생성된다. 냅스터는 공유 디렉토리에서 MP3로 끝나는 모든 확장자를 찾아낸 후 이 용자 ID와 MP3 음악 리스트를 냅스터 중앙 서버로 가져간 후 MP3 파일 목록을 보관한다. 사용자가 가수이름, 음악곡명, 키워드 등으로 검색하여 일치한 음악을 찾아 요청하면 그 음악을 보유하고 있는 컴퓨터와 자동으로 연결시켜 다운로드 받게 해 준다.

소리바다는 냅스터와 유사한 MP3 파일 교환 서비스

로, 사용자끼리 서로의 PC에 들어 있는 MP3를 검색하고 편리하게 다운로드도 할 수 있다. 다운로드가 끝나지 않은 상태에서도 실시간으로 음악을 즐길 수 있으며, 가수과 제목별 검색, 채팅도 할 수 있다.

표 2에 기존의 혼합형 P2P 방식이 가지는 문제점과 본 논문에서 제안되는 P2P 방식의 장점을 비교 기술하였다.

이외에도 JXTA P2P 네트워크를 사용하여 유비쿼터스 환경을 위한 이동 에이전트 시스템인 UbiMAS[22]가 개발되었으며, 분산해쉬테이블을 이용한 구조적 P2P 시스템이 제안되었다[23]. 그리고 P2P 시스템에 관한 연구 방향 및 활용 등에 관한 항목들이 [24-26]에 일목요연하게 정리가 되어 있으니 참고하기 바란다.

표 2 기존 혼합형 P2P와 제안된 시스템의 비교

	기존 P2P 방식의 문제점	제안된 방식에서의 개선
혼합형	<ul style="list-style-type: none"> · 슈퍼 피어가 공유 정보의 중개자 역할을 하므로 <ul style="list-style-type: none"> - 콘텐츠 저작권법에 위반될 수 있다. - 슈퍼 피어가 동작하지 않으면 전체 시스템의 동작이 멈춘다. · 슈퍼 피어에 의해 정보 검색이 수행되므로 <ul style="list-style-type: none"> - 온라인 상의 사용자가 많을 경우 슈퍼 피어에 과중한 부하가 걸릴 수 있으며, 따라서 슈퍼 피어의 증설이 필요하여 경비의 증가를 필요로 한다 	<ul style="list-style-type: none"> · 루트(root) 슈퍼 피어와 자생(voluntary) 슈퍼 피어 2가지 유형의 슈퍼 피어를 가지고 있으며 자생 슈퍼 피어는 온라인 상의 일반 피어들 중에서 선정되므로, <ul style="list-style-type: none"> - 루트 슈퍼 피어는 온라인 된 자생 슈퍼 피어의 주소만 보유하고 있으므로 저작권법에 위반되지 않는다. - 온라인 상의 일반 피어들은 자생 슈퍼 피어에 등록되어 온라인 상의 사용자가 많아도 슈퍼 피어에 과중한 부하가 걸리지 않는다. - 2개 이상의 자생 슈퍼 피어가 존재하므로, 일부의 자생 슈퍼 피어에 결함이 발생하여도 동작에는 지장을 주지 않는다 - 2개 이상의 이동 에이전트에 의해 병렬로 검색이 일어나므로 효율적이다

4. YAMUS: 분산 혼합형 P2P 시스템

4.1 구성

본 논문에서 제안된 시스템은 기존의 혼합형 P2P 및 순수 P2P 2가지 유형의 P2P 시스템이 가지는 장점들을 모아 구성된 절충형 P2P 프레임워크이며, YAMUS (You-And-Me-US)라고 한다. YAMUS 시스템을 구성하는 각 피어 노드는 그림 4에서 보여준 것과 같은 구조를 가지고 있다. 하나의 피어 노드는 슈퍼 피어 모듈 및 일반 피어 모듈, 2개의 컴포넌트 모듈로 구성되어 있으며, 슈퍼 피어로서 동작할 경우에는 2개의 모듈이 모두 동작하여, 일반 피어로서의 기능도 수행하면서 슈퍼 피어 기능을 수행한다. 일반 피어로서만 동작할 경우에는 일반 피어 모듈만 동작하게 된다. 또한, YAMUS 시스템에는 그림 5에서 보여준 바와 같이, 서로 다른 역할을 하는 3가지 유형의 노드들이 존재하고 있는데, 루트 슈퍼 피어(root super-peer, RSP), 자생 슈퍼 피어(voluntary super-peer, VSP), 그리고 일반 피어

(general peer, GP)들이 그것들이다. RSP의 핵심 기능은, VSP들의 등록 및 해지이며, 그 외 다른 기능으로는 통계를 위한 다양한 정보의 유지관리를 들 수 있는데, 이는 응용에 따라 달라질 수 있다. 예로서 등록 횟수, 검색 횟수, 다운로드 횟수 등을 들 수 있다. VSP의 핵심 기능은 일반 피어들의 등록 및 해지이며, 각 VSP가 관리할 수 있는 일반 피어들의 최대 수를 지정할 수 있도록 하고 있다. 각 VSP에 등록되는 일반 피어들의 집합을 ZONE이라고 할 때, YAMUS 시스템은 ZONE들의 집합이라고 할 수 있다. Dom를 YAMUS 시스템 도메인이라 하고, Z(i)를 VSP-i에 의해 관리되는 ZONE이라 하면,

$$\text{Dom} = \{Z(i) \mid i=1, \dots, N\},$$

$$Z(i) = \{GP(i, k) \mid k=1, \dots, \text{Max}\}$$

가 된다. 여기서 N은 도메인에 속한 ZONE의 총 개수이며, Max는 Z(i)에 포함될 수 있는 최대 허용 일반 피어의 수이다.

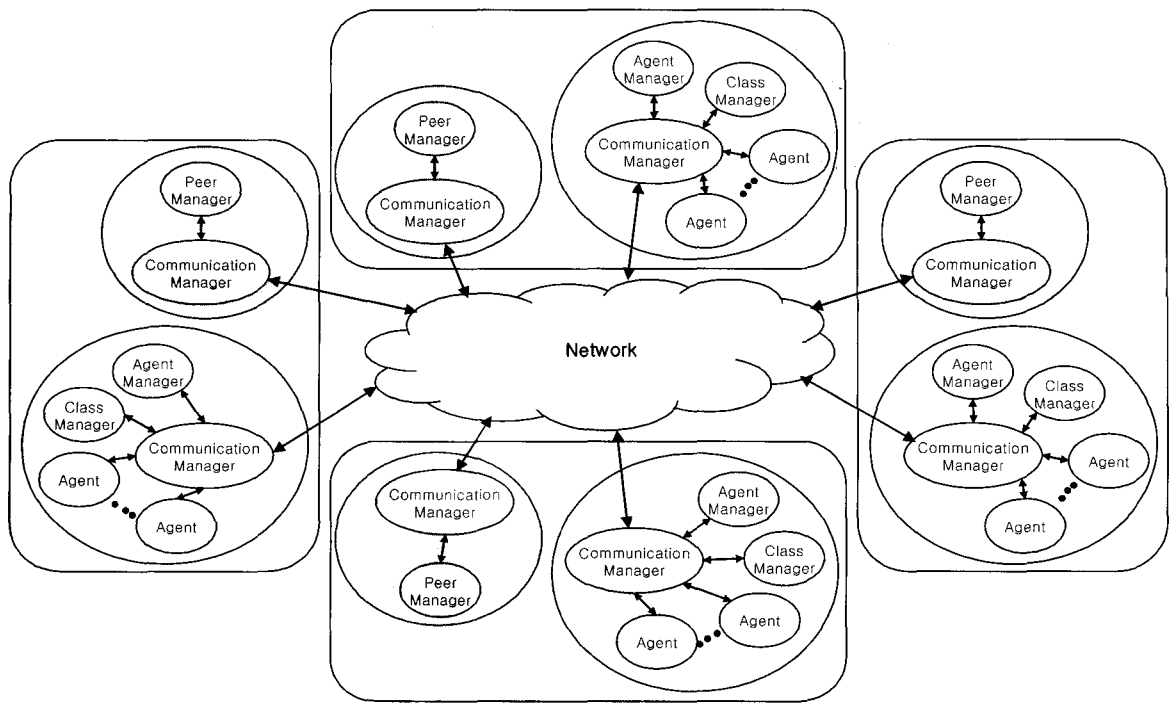


그림 4 YAMUS 네트워크를 구성하는 각 피어의 컴포넌트

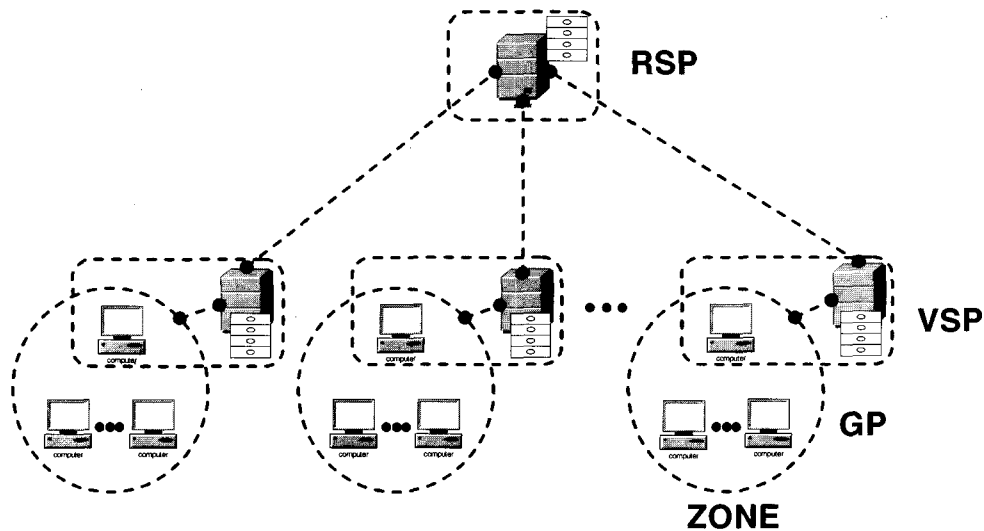


그림 5 YAMUS 구성 요소: RSP, VSP, GP

4.2 VSP 수퍼 피어 설정 알고리즘

온라인 상태에 있는 각 일반 피어는 어느 하나의 VSP에 반드시 등록이 되어야 한다. VSP의 설정은 일반 피어로서 자신을 등록하는 과정에서 등록 행위와 더불어 자생적으로 일어나게 되는데, YAMUS에서 GP의 등록 및 VSP 생성 알고리즘 과정은 다음과 같다. STEP 2-2에서 등록이 받아들여지는 경우는 현재 일반 피어가 등록을 요청한 ZONE에 지정된 일반 피어의 수가 Max보다 작은 경우이며, 등록이 거부되는 경우는 현재 등록된 일

반피어의 수가 Max 이상인 경우이다. 등록이 거부되고 더 이상 등록을 요청할 VSP가 존재하지 않는 경우, 등록 요청을 한 일반 피어는 스스로가 VSP가 되면서 스스로를 자신에게 등록하고 이를 RSP에 통보한다. 그러므로 각 VSP는 항상 자신에게 하나의 GP로써 등록됨을 알 수 있다. 또한 초기에 VSP가 없는 경우, 즉 $Dom = \emptyset$ 인 경우, 처음 등록을 요청하는 일반 피어는 VSP가 된다는 것도 알 수 있다. 그림 6에는 등록을 요청한 일반 피어가 Dom 상의 모든 VSP들로부터 등록을 거부당

해 스스로가 VSP가 되는 경우를 그림으로 보여주고 있는 경우이며, 그림 7에는 등록을 요청한 일반 피어가 Dom 상의 특정 VSP로부터 등록을 허락받아 해당 VSP에 등록이 되어, 그 ZONE의 멤버가 되는 과정을 보여주고 있다. 그림 7의 경우, 등록을 요청한 일반 피어가 VSP-3에 등록되는 것을 보여주고 있다. 알고리즘의 Dom 정보의 구현을 위해 RSP는 온라인 상에 있는 VSP들의 목록인 VSP 리스트(VSPL)를 유지 관리한다. VSPL은 새로운 VSP가 등록될 때마다 갱신되며, 특정 파일로써 저장되어, RSP 구동 시, 기존 VSPL 파일을 사용할 것인가 아니면 새로운 파일을 사용할 것인가

결정하게 된다. 이 기능은 RSP에 결함이 발생하였을 경우, RSP 복구 후 전 상태를 유지하기 위해서도 필요하다. 또한, 각 VSP는 온라인 상태에 있는 일반 피어들의 목록인 GP 리스트(GPL)를 유지 관리한다. 각 Z(i)는 VSP-i로 대표되며 모두 RSP의 VSPL에 등록되어 있으며, 또 온라인 상태에 있는 각 일반 피어는 해당 VSP의 GPL에 등록되어 있다. 그러므로 도메인 Dom은 현재 동작중인 VSP들의 목록인 VSPL이라는 자료 구조로써 구현될 수 있으며, 각 ZONE, Z(i)는 그에 속한 온라인 상태의 일반 피어들의 목록인 GPL이라는 자료 구조로써 구현될 수 있다.

[일반 피어 등록 및 VSP의 생성]

STEP-1: RSP에게 Dom 정보(VSPL) 요청 및 수신

```
/* Dom = {Z(i) | i=1, ..., N} */
```

STEP-2: 수신한 Z(1:N)에 대해,

```
if Dom = ∅, { 새로운 VSP가 되어 자신에게
  등록하고 RSP에 통보; exit() ;
}
```

```
while(i =< N) {
```

```
  2-1) VSP=Z(i) 에게 등록 요청
```

```
  2-2) if (|Z(i)| < Max), VSP에 등록; exit();
```

```
    /* VSP=Z(j) ∃ GP */
```

```
  2-3) else if (i == N), /* |Z(i)| ≥ Max */
```

```
    {자신이 새로운 VSP가 되어 자신에게 등록
```

```
    하고 RSP에 통보; exit() ;
```

```
    /* Dom = Dom ∪ (newVSP=GP) */
```

```
  2-4) else i = i+1;
```

```
    /* 다음 ZONE에 등록 시도 */
```

```
}X
```

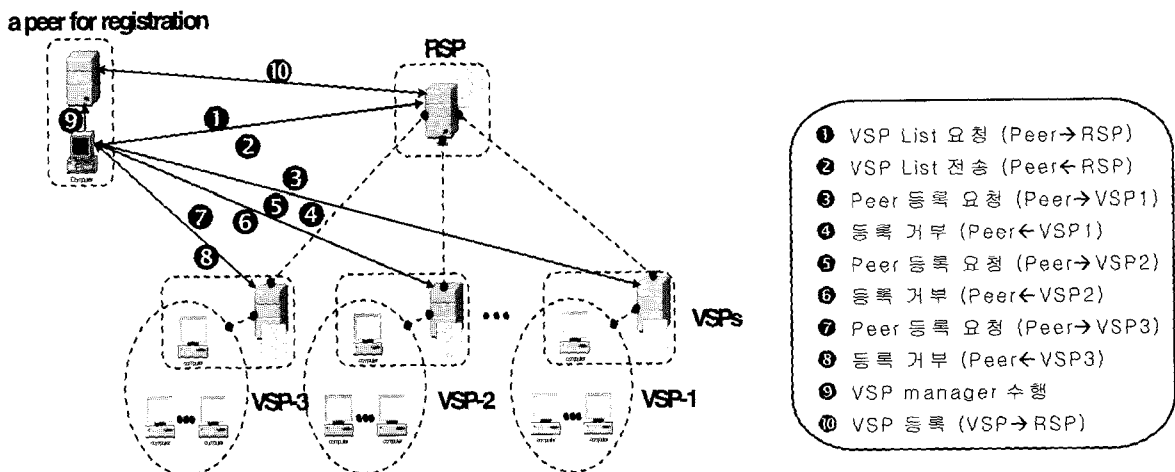


그림 6 등록을 요청한 GP가 등록이 거부되어 스스로 VSP가 되는 과정

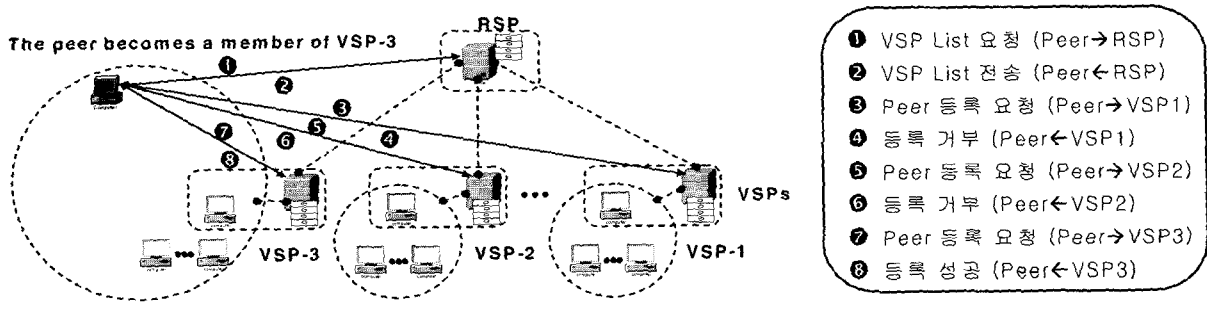


그림 7 등록을 요청한 GP가 특정 VSP에 등록되는 과정

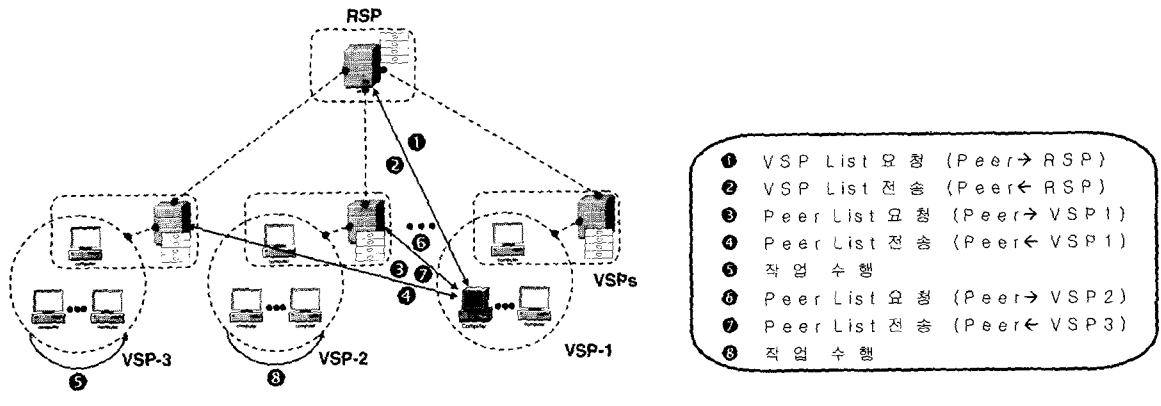


그림 8 YAMUS에서 GP의 자료 탐색 과정

4.3 자료 검색

일반 피어가 자료를 검색하기 위해서는 먼저 전체 ZONE 리스트인 VSPL을 RSP로부터 얻어야 한다. 입수한 VSPL을 기반으로 ZONE 단위로 탐색을 시도한다. 그 과정을 보여주고 있는 것이 그림 8이다.

YAMUS에서 각 GP의 정보 탐색 과정을 알고리즘으로 기술하면 다음과 같다.

RSP의 결함 혹은 네트워크 결함으로 RSP로부터 VSPL을 수신할 수 없는 경우에는 기존에 받아 두었던 VSPL을 사용하여 자료 탐색을 수행한다.

[정보의 탐색]

```

STEP-1: RSP에게 VSPL을 요청하여 VSPL을 수신
STEP-2: 수신한 VSPL(1:N)에 대해
while(j <= N) {
  2.1: VSPL(j)에게 GPL을 요청하여 GPL 수신
  2.2: 수신한 GPL(1:Max)에 대해
    while(i <= Max) { /* resource discovery */
      2.2.1 GPL(i)로 에이전트를 보내 탐색 결과 수신
      2.2.2 i = Max 이면, exit() ;
      2.2.3 i = i+1 ;
    }
    /* 이 부분에 적절한 프로그램을 삽입하여
    검색 제어 가능 */
  2.3 j = N 이면, exit ;
  2.4 j = j+1 ;
}

```


5. 결합 허용 특성 및 부하 감소

본 논문에서 제안된 분산 혼합형 P2P 시스템은 ZONE 단위로 검색을 수행하며, 결합 허용 특성은 RSP 및 VSP 측면 모두에서 가지고 있다. 각 일반 피어는 정보 탐색을 위해 RSP로부터 VSPL을 받게 되는데, RSP 결합 혹은 네트워크 결합으로, VSPL 수신에 이루어지지 못하는 경우, 이전에 수신하여 메모리에 저장된 VSPL을 기반으로 정보 탐색을 수행한다. 그리하여 RSP의 정상적인 액세스가 가능할 때까지는 기존의 ZONE들을 대상으로 검색이 일어나게 되며, 그 동안은 VSPL에 대한 갱신이 이루어지지 않는다는 단점이 있다. RSP가 복구되면 새로운 VSPL을 얻어 갱신된 VSPL 정보를 기반으로 탐색을 할 수 있다. VSPL을 수신한 각 일반 피어는 VSPL에 등록된 각 VSP로부터 GPL을 수신하여 GPL 상의 일반 피어들을 대상으로 검색을 수행하므로 등록된 VSP들 모두가 수신 불가능인 경우를 제외하고는 검색이 수행되게 된다. 즉, VSP 결합 혹은 네트워크 결합으로 특정 VSP로의 접속이 불가능할 경우에는 해당 VSP가 관리하는 ZONE에 속한 일반 피어들에 대한 검색만이 이루어지지 않으며, 그 외 다른 VSP들이 관장하는 ZONE들에 대해서는 정상적인 검색이 일어나게 된다. 결국, YAMUS 시스템은 RSP, VSP 혹은 네트워크 결합으로 VSPL 혹은 GPL의 수신에 불가능해도 정상적으로 동작하는 VSP가 존재하는 한정해진 범위 내에서의 검색은 이루어지게 되는 것이다.

VSP를 두지 않는 구조일 경우, RSP가 모든 피어에 관한 주소 정보를 관리하고 있어야 하므로, 각 피어가 자료 검색을 위하여 RSP에게 피어들의 리스트를 요청하여야 하는데, 이 경우 피어 리스트 전체가 피어에게 전송되므로 전송되는 데이터의 량을 증가시키며, 피어의 수가 많은 경우, RSP의 부하를 증가시키게 된다.

이외에도 YAMUS는 다양한 특징을 지니고 있다. 자세한 사항은 참고문헌 [8],[13]을 참조하기 바란다. 그 중 대표적인 몇 가지를 들면, 1) 응답 시간 감소를 위한 에이전트 병렬 전송 메카니즘, 2) 네트워크 상황에 따른 다양한 여정 방식 및 응답 방식, 3) 프로그램의 용이성을 위한 고급 인터페이스 및 4) 기초적인 보안을 위한 바이트 코드 검증과 프록시 기능을 가지고 있어 다양한 P2P 응용 개발을 위해 매우 효율적인 특성을 가지고 있다.

6. YAMUS 응용

본 논문에서 제안된 분산 혼합형 P2P 프레임워크인 YAMUS를 기반으로 다양한 작업을 가능하게 하는 종합 P2P 응용 시스템인 “올-인(All-In)”이 구현되었다.

구현된 환경은 9대의 PC 노드들로 구성되어 있으며, 각 노드는 10Mbps 이더넷 스위치를 통해 외부 네트워크와 연결되어 있다. 각 노드에는 YAMUS 시스템이 설치되어 있으며, “올-인” 응용 프로그램이 같이 설치되어 있다. 올-인 시스템은 5가지 P2P 응용을 포함하고 있다. Job 찾기, 물품 거래, 파일공유, 지식 상담, 그리고 메신저 기능들이 그것들이다. 본 장에는 이들 중 물품 거래(그림 9의 메뉴 중 ‘아나다바’ 버튼)에 관한 수행만 기술한다. 다른 응용에 관한 사항들은 YAMUS 사이트를 참조하기 바란다[27]. 본 응용, ‘아나다바’는 물품 정보를 검색하려는 사용자와 물품 정보를 제공하는 사용자로 구성되며, 본 논문에서는 각각을 구매자와 판매자로 정의한다. 즉, 탐색 주체 노드의 사용자는 물품의 구매자 역할을 하며, 탐색 대상 노드의 사용자는 판매자가 되는 것이다. 본 실험에서는 Node-1을 물품의 구매자이며, Node-2부터 Node-9까지는 판매자로 설정하였다. 그리고 구매자인 Node-1에게로 정보 탐색 결과를 응답하도록 하였다.

물품 정보 검색 과정의 실험은 설계된 실험 시나리오에 충실하게 단계별로 정보 탐색 작업이 수행되도록 하였다. YAMUS 시스템이 제공하고 있는 유용한 기능들을 사용하면 쉽게 작업을 수행할 수 있다. YAMUS 메인 인터페이스를 보여주고 있는 것이 그림 9이다.

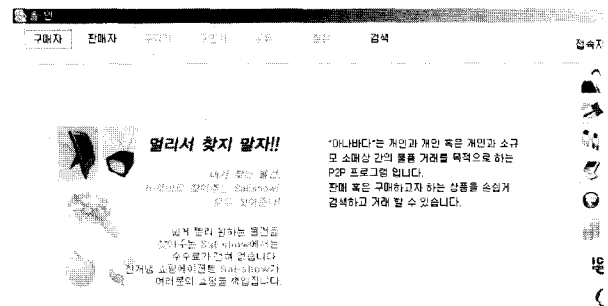
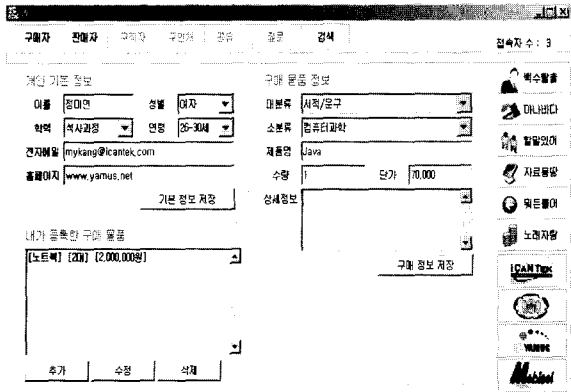


그림 9 YAMUS 기반 올-인 메인 화면

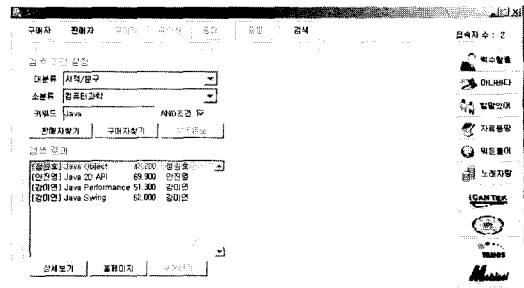
설계된 실험 모델에 기반을 두고 물품 정보 탐색 시나리오에 따라 작업을 수행하는 과정은 다음과 같다.

- 1) 탐색 주체 노드, 즉 구매자 노드인 Node-1에서 이동 에이전트가 제공하는 인터페이스를 통하여 구매하려는 물품의 정보를 등록한다. 인터페이스에서 [구매자] 버튼을 선택하면 그림 10의 (a)와 같은 구매 물품 정보를 등록하는 박스가 나타난다. 구매하려는 물품이 해당하는 분류를 대분류, 소분류 순서로 리스트에서 선택하고, 수량, 단가, 물품명, 모델명, 상세 사양 등을 입력한다.
- 2) 탐색 대상 노드, 즉 판매자 노드인 Node-2부터

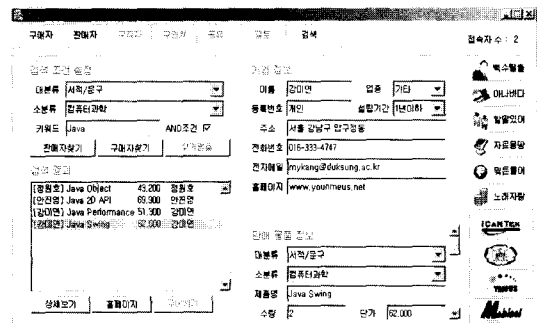
Node-9에서는 판매하려는 물품의 정보를 등록한다. 인터페이스에서 [판매자] 버튼을 선택하면 그림 10의 (b)와 같은 판매 물품 정보를 등록하는 박스가 나타난다. 판매하려는 물품이 해당하는 분류를 대분류, 소분류 순서로 리스트에서 선택하고, 수량, 단가, 물품명, 모델명, 상세 사양 등을 입력한다.



(a) 구매 물품 정보 등록

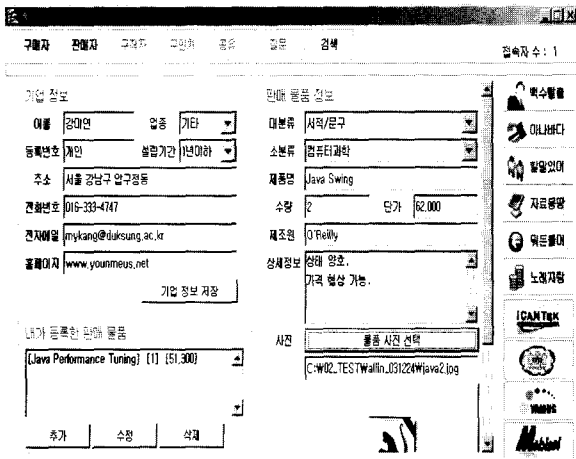


(a) 검색 결과 리스트



(b) 특정 물품에 대한 상세보기

그림 11 구매 물품 정보의 검색 결과



(b) 판매 물품 정보 등록

그림 10 구매 및 판매 물품 정보의 등록

3) 창에서, [검색] 버튼을 선택하고, 검색하고자 하는 물품 정보의 내용을 입력한 후 [구매자 찾기] 혹은 [판매자 찾기]를 하면 탐색 에이전트가 생성되어 여정 리스트에 따라 이동이 일어난다. 탐색 노드에 도착한 에이전트는 판매자(혹은 구매자)가 등록한 물품의 정보와 구매자(혹은 판매자)가 요청한 정보를 비교하는 작업을 수행한다. 그리하여, 구매자(혹은 판매자)가 요청한 물품 정보에 적합한 판매자(혹은 구매자)의 물품 정보를 응답 대상 노드인 Node-1으로 전송한다. 검색된 결과물의 출력의 예시가 그림 11에 보여주고 있다.

7. 결론 및 향후 연구

이동 에이전트는 여러 가지 분산 응용에서 네트워크 부하와 지연을 극복할 수 있는 기법 중 하나로 최근 활발히 연구되고 있는 새로운 패러다임이다. 본 논문에서는 이동 에이전트 기반의 효율적 분산 응용을 위한 P2P 프레임워크인 YAMUS가 설계, 구현되었다. YAMUS는 간단한 구조, 유용하며 풍부한 메소드, 그리고 유연성 있는 이동 방식, 그리고 응답 방식 등을 제공하고 있어, 이동 에이전트 기반의 P2P 응용에 쉽게 응용될 수 있다. 기존의 혼합형 시스템의 문제점인 부하 편중과 결합 특성 문제를 제거하기 위하여 RSP와 VSP로 구성되는 2단계 다중 수퍼 피어 구조를 취하였다. 그리하여, RSP, VSP 혹은 네트워크 분할 등의 문제가 발생하여도 결합 부분을 제외하고는 정상적인 탐색이 이루어지도록 하고 있다. YAMUS는 Java 이동 에이전트 기반의 P2P 시스템이다. 이동 에이전트를 사용함으로써, 혼합형 P2P의 경우, 중앙 서버의 부하를 줄이고, 대역폭을 효율적으로 사용하는 장점을 가지고 있으며, 순수 P2P의 경우 TCP 접속 수를 줄이고, 전달 지연을 감소시키는 장점들을 지지고 있다. 이동 에이전트 시스템에서의 중요한 문제 중에 하나인 보안의 경우, 현재의 YAMUS는 자바가 제공하는 바이트 코드 검증기에 의한 1차적인 보안 기능을 가지고 있어, 정의되지 않은 다른 에이전트

의 수행은 수행이 불가능하도록 되어 있어, YAMUS를 통한 다른 에이전트의 침투가 어렵게 되어있다. 추후, 시스템 내에서 2중 프록시를 사용하는 2차적인 보안 기능을 연구 중에 있으며, YAMUS 시스템이 지원하는 기능을 바탕으로 다른 응용에도 쉽게 대처할 수 있도록 개선해 나갈 예정이다.

참고문헌

- [1] Gnutella, <http://gnutella.wego.com/>
- [2] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," Proc. of the ICSI Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, 2000
- [3] B. Yang and H. G.-Molina, "Comparing Hybrid Peer-to-Peer Systems," Proc. Of the 27th VLDB Conference, 2001
- [4] S. A.-Theotokis, "White Paper: A Survey of Peer-to-Peer File Sharing Technologies," ELTRUN, Athens Univ. of Economics and Business, Greece, 2002
- [5] Bo Leuf, "Peer-to-Peer: Collaboration and Sharing over Internet," Addison-Wesley, June, 2002
- [6] L. Ciminiera, et al., "Survey on Grid and Peer-to-Peer Network Technologies," European Grid of Solar Observations(EGSO), EGSO-DE01-01-D02-021001, Oct. 2002
- [7] 김 인숙 외, "순수 P2P 환경을 위한 이동 에이전트 기반 자원 검색 기법", 정보처리학회 논문지 D, 제10-D권, 제2호, 2003. 4
- [8] 강미연, 김윤수, 정원호, "이동 에이전트 기반의 Peer-to-Peer 네트워킹", 한국정보과학회 춘계학술대회 논문집, 2003
- [9] Ozalp Babaoglu et al., "Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems," In Proceedings of the 22th International Conference on Distributed Computing Systems (ICDCS '02), Vienna, Austria, July 2002. Also appears as Technical Report UBLCS-2001-09, University of Bologna, Italy
- [10] Alberto Montresor, "Anthill: a Framework for the Design and Analysis of Peer-to-Peer Systems," In Proceedings of the 4th European Research Seminar on Advances in Distributed Systems, Bertinoro, Italy, May, 2001
- [11] Munehiro Fukudu et al., "A Mobile-Agent-Based PC Grid," The 5th Annual International Workshop on Active Middleware Services (AMS2003), Seattle, WA, June 2003
- [12] D. B. Lange and M. Oshima, "Programming And Deploying Java Mobile Agents with Aglets," Addison Wesley, 1998
- [13] 정원호, 강미연, 김윤수, "효율적 분산 응용을 위한 이동에이전트 프로그래밍 시스템", 한국정보처리학회논문지, 10-A권, 5호, 2003년 10월
- [14] D. Kotz, et al., "AGENT TCL: Targeting the Needs of Mobile Computers," IEEE Internet Computing, July/Aug., 1997
- [15] J. Kiniry and D. Zimmerman, "A Hands-On Look at Java Mobile Agents," IEEE Internet Computing, July/Aug., 1997
- [16] J. Baumann et al., 'Mole Concepts of a Mobile Agent System,' Tech. Report, Institute for Parallel and Distributed High-Performance Computers(IPVR), Stuttgart Univ., 1997.
- [17] C. Panayiotou et al., "Parallel Computing Using Java Mobile Agents," A working paper at Dept. of CS, Univ. of Cyprus, 1999.
- [18] N. Minar et al., "Cooperating Mobile Agents for Dynamic Network Routing," MIT Media Lab., 1999.
- [19] N. Ivezic et al., "Agent-based Technologies for Virtual Enterprises and Supply Chain Management," A draft version for submission, IEEE Internet Computing, CTRC, Oak Ridge National Lab., 1999.
- [20] Won-Ho Chung et al., "A Mobile Agent Scheme with Flexible Reply and Routing for Supply Chain Management," Prof. of Asia-Pacific Conf. on Communications(APCC), 2000
- [21] D. D. Roure, et al., "Agents for Distributed Multimedia Information Management," Proc. of 1st Int'l Conf. on the Practical Applications of Intelligent Agents and Multi-Agent Technology, 1996

- [22] F. Bagci et al., "Ubiquitous Mobile Agent System in a P2P-Network," Proc. of the Workshop on System Support for Ubiquitous Computing at the Fifth Annual Conference on Ubiquitous Computing (UbiComp 2003) Seattle, Oct. 2003
- [23] S. El-Ansary, "A Framework for the Understanding, Optimization and Design of Structured Peer-to-Peer Systems," Ph.D Dissertation of Royal Institute of Technology, June, 2003
- [24] K. Kant et al.: "On the Potential of Peer-to-Peer Computing", Working Paper available at: <http://kkant.ccwebhost.com/papers/taxonomy.pdf>
- [25] D. Barkai, "An Introduction to Peer-to-Peer Computing," P2P Architecture Group, Microcomputer Research Lab., Intel Corporation, Oct. 2000
- [26] M. Parameswaran, A. Susarla, and A. Whinston, "P2P Networking: An Information-

Sharing Alternative," IEEE Computer, Vol. 34, No. 7, pp.31-38, 2001

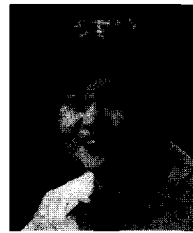
- [27] YAMUS homepage, <http://www.yamus.net>

정 원 호



1979 서울대학교 전자공학과(학사)
 1981 한국과학기술원 전기및전자공학과(석사)
 1989 한국과학기술원 전기및전자공학과(박사)
 1979~1982 대한전선(주)
 1983~1984 대우통신(주)
 1993 IBM T. J. Watson 연구소 방문 연구원
 2002~2003 (주)아이캔텍 고문
 1989~현재 덕성여자대학교 컴퓨터과학부 교수
 E-mail : whchung@duksung.ac.kr

강 미 연



1999 덕성여자대학교 전산학과(학사)
 2001 덕성여자대학교 전산및정보통신학과(석사)
 1999~2001 덕성여자대학교 전산학과 조교
 2001~현재 (주)아이캔텍 기술연구소
 관심분야 : 이동 에이전트 시스템, 임베디드 시스템 등
 E-mail : mykang@icantek.com

• The 14th Joint Conference on Communications & Information(JCCI 2004) •

- 일 자 : 2004년 4월 28~30일
- 장 소 : 금호 충무 마리나리조트(충무)
- 주 최 : 정보통신연구회
- 상세안내 : KAIST 이용훈 교수(Tel. 042-869-4411)
<http://aitrc.kaist.ac.kr/~dasfaa04>