[UCI]I804:24011-200000232709

*2006*年 *2*月

碩士學位論文

*JXTA*기반 *P2P*시스템에서 검색 성능 향상

朝 鮮 大 學 校 大 學 院

컴 퓨 터 工 學 科

閔 惠 蘭

*JXTA*기반 *P2P*시스템에서 검색 성능 향상

Improving of Search Performance in JXTA-based P2P System

*2006*年 *2*月 日

朝 鮮 大 學 校 大 學 院

컴 퓨 터 工 學 科

閔 惠 蘭

*JXTA*기반 *P2P*시스템에서 검색 성능 향상

指導敎授　李　俊

이 論文을 工學碩士學位申請 論文으로 提出함.

2005 年 *10* 月 日

朝 鮮 大 學 校 大 學 院

컴 퓨 터 工 學 科

閔 惠 蘭

閔惠蘭의 碩士學位 論文을 認准함.

委員長　　朝鮮大學校 敎授 _____ 印

委 員　　朝鮮大學校 敎授 _____ 印

委 員　　朝鮮大學校 敎授 _____ 印

*2005*年 *11*月　　日

朝 鮮 大 學 校 大 學 院

목 차

List of Titles

ABSTRACT ····· V

제 *1* 장 서 론 ····· 1

제 *2* 장 관련 연구 ····· 3

제 1 절 Peer To Peer 프로토콜 ····· 3

제 2 절 JXTA 플랫폼 ····· 8

제 *3* 장 *P2P* 자원 탐색 알고리즘 ····· 24

제 1 절 JXTA의 자원 탐색 알고리즘 ····· 24

제 2 절 랑데부 피어 순회 탐색 메커니즘 ····· 26

제 3 절 랑데부 피어 순회 탐색의 단점 ····· 28

제 4 절 P2P 네트워킹을 위한 분산 해쉬 테이블 방식 ····· 30

제 *4* 장 *JXTA*기반의 *P2P*시스템 설계 ····· 33

제 1 절 릴레이 서버 구성 ····· 34

제 2 절 클라이언트 프로그램 ····· 37

제 3 절 DHT 프락시 피어 탐색 알고리즘 설계 ····· 39

제 5 장 실험 및 고찰 ····· 43

제 1 절 시뮬레이션 환경 ····· 43

제 2 절 실험 결과 ····· 44

제 6 장 결 론 ····· 47

참고문헌 ····· 49

표 목 차

List of Tables

표 1. JXTA에서 정의한 Advertisement ····· 17

표 2. CMS 서비스 객체 ····· 19

그 림 목 차

List of Figures

그림 2-1. 혼합형 P2P 구조(넵스터) ····· 5
그림 2-2. 순수 P2P 구조(누텔라) ····· 7
그림 2-3. JXTA의 계층 구조 ····· 9
그림 2-4. Query를 받는 동작 ····· 13
그림 2-5. Query에 응답하는 동작 ····· 14
그림 2-6. Resolver 서비스의 동작 모델 ····· 15
그림 2-7. ListContentRequest와 GetContentRequest ····· 21
그림 2-8. CMS에 사용되는 ContentAdvertisement ····· 23
그림 3-1. JXTA의 자원 탐색 알고리즘 ····· 25
그림 3-2. 랑데부 피어 순회 탐색 메커니즘 ····· 27
그림 3-3. 랑데부 피어 순회 탐색의 Roll-back 현상 ····· 29
그림 3-4. DHT의 개념도(insert 와 lookup) ····· 32
그림 4-1. 릴레이 서비스 메시지 전송 ····· 35
그림 4-2. 릴레이 서버 구조 ····· 36
그림 4-3. 모바일 클라이언트 피어 구조 ····· 38
그림 4-4. DHT 프락시 피어 탐색 알고리즘 ····· 41
그림 5-1. 광고정보 검색 비용 ····· 45
그림 5-2. 광고정보 검색 총비용 ····· 46

ABSTRACT

Improving of Search Performance in JXTA-based P2P System

Min, Hye-Ran
Advisor: Prof., Lee, Joon Ph.D.
Department of Computer Engineering
Graduate School of Chosun Universiy

Internet in its infancy was just a small-scale network consisting of a small number of computers in which one could access to data by means of a terminal. Gradually, this concept opened an infinitude of possibilities, and Internet has evolved into a global network accommodating millions of computers, which made it possible to connect all around the world, to share and access to information others have, and to communicate with them freely. It has been said that WWW (world wide web) was the second revolution that helped popularize Internet widely. And now P2P is being noted as the third evolution that will fundamentally change the client-server based Internet paradigm resulting from WWW. Personal PCs, which used to play a role as a mere client that accessed to the server and gained

the information, came to possess the server function, and this made the Internet structure change from vertical system to horizontal.

P2P Network Environment indicates the new concept of network with mutual distribution and collaboration among clients. Here, the concept of 'distribution' is to share the hardware resources through the distribution computing of P2P network for the purpose of maximizing efficiency, and that of 'collaboration' is to search the shared resources through collaboration among clients.

In such distributed computing environment as P2P, searching distributed resources for efficient use is of capital importance. Yet, the existing way to search distributed resource information only makes use of the information about the connection between the rendezvous peers, which causes not only much expense in searching resource information, but also thus lots of retardation and traffic. To solve this problem, this study will provide a solution regarding efficient resource information searching expense by designing JXTA based P2P system supporting independent environment of network devices, and by presenting DHT proxy peer algorithm searching distributed resources efficiently in JXTA network environment.

제 1 장 서 론

초기의 인터넷은 단말기를 통해 데이터를 접근할 수 있도록 해 주는 소수의 컴퓨터들로 구성된, 규모가 별로 크지 않은 네트워크였다[1][2]. 점차적으로 이 개념은 무한한 가능성을 열었고, 인터넷은 전 세계를 연결하고 사용자들이 정보를 공유, 접근 그리고 자유로운 통신을 할 수 있는 수백만의 컴퓨터들을 포함하는 글로벌 네트워크로 발전했고, 새로운 도약기를 맞이하면서 움직이는 네트워크의 필요성이 대두되고 있다[3].

최근 수년간 클라이언트 모델에서 발생하는 문제인 서버 병목현상, Dos공격, 그리고 시스템의 확장성에 따른 비용 증가 등의 문제를 해결하는 방법으로 P2P 방식이 활발히 연구되고 있으며, 그 중 사용자간 파일 공유하는 분야에 적용된 P2P가 가장 널리 알려져 있다.

처음 선보인 P2P 솔루션은 P2P 컴퓨팅 기술을 이용해 단순한 작업을 수행하는데 그쳤지만, 최근 선보이고 있는 P2P 솔루션은 여기서 벗어나 하나의 요소기술로 사용되거나 P2P 기술을 활용할 수 있는 총체적인 솔루션으로 개발되는 경향이 늘고 있다[5]. JXTA에서 사용하는 프로토콜은 언어 독립이며, P2P 네트워킹의 여러 가지 특징을 조율하는 XML메시지의 집합을 정의한다[5][6]. P2P 커뮤니케이션에서 일부 개발자들이 XML 사용에 거부감을 나타내지만, XML의 선택은 JXTA 프로토콜의 개발자들에게 XML파싱과 포맷팅 같은 기존의 개발도구를 이용할 수 있도록 한다. 그리고 JXTA 프로토콜의 단순성이 PDA 혹은 휴대전화와 같은 디바이스에서도 P2P솔루션의 구현이 가능하도록 한다.

P2P와 같은 분산 컴퓨팅 환경에서는 효율적인 이용을 위해 분산된 자원의 검색이 무엇보다 중요하다. 그러나 분산된 자원을 검색할 때 각 피어를 순회하며 수

많은 피어들의 자원 정보를 검색해야 하는데, 이로 인해 네트워크 트래픽이 발생하고, 검색 시간이 든다.

본 논문에서는 플랫폼에 독립적이고, 네트워크 디바이스에 독립적인 환경을 지원하는 JXTA 네트워크 환경에서 분산된 자원의 효율적인 검색을 하는 DHT(Distributed Hash Table) 알고리즘을 제안하고자 한다.

DHT는 분산 해쉬 테이블을 이용하는 기법으로써, 모든 피어들은 공유하고 있는 자원들에 대한 메타 데이터를 해쉬 값에 의해 지정된 피어에게 전송하며, 메타 데이터를 수신한 피어들은 다른 피어로부터 검색 요청이 들어 올 경우 자원의 위치를 알려주는 방식으로 동작한다. 기존의 랑데부 피어 탐색 알고리즘의 문제점을 찾고, 이를 해결하기 위한 랑데부 피어의 탐색 알고리즘을 확장하여, 좀 더 효율적인 자원 탐색 비용을 제공할 수 있다.

제 2 장 관련 연구

제 1 절 Peer To Peer 프로토콜

Peer To Peer 프로토콜은 각 컴퓨터가 동등한 능력을 가지고 있어, 어떤 컴퓨터에서라도 통신 세션을 시작할 수 있는 통신 모델을 말한다. 이전의 클라이언트/서버 모델은 서버가 꺼지면 클라이언트들이 모두 활동을 중지해야 한다. 이에 반해, P2P 방식은 사용자가 자기 시스템을 이용한 네트워크에 주인이 되는 서비스이다[1]. 각 Peer는 자신만의 컨텐츠를 주고 받으며 특정 슈퍼서버에 의해 다운되거나 시스템 장애를 일으키지 않는다. P2P 네트워크 모델은 네트워크의 구조에 따라 혼합형 P2P 네트워크 구조와 순수 P2P 네트워크 구조로 분류된다.

혼합형 P2P 구조인 그림 2-1은 서버를 통해 각 피어들에게 자원의 공유 기능을 지원하는 모델로서, 서비스를 제공할 수 있는 중앙서버와 실질적인 공유자원을 가지고 있는 피어들로 이루어진다. 이러한 구조를 갖는 P2P 시스템으로는 대표적으로 냅스터(Napster)가 있다[2][3].

냅스터는 Peer 발견과 컨텐츠 목록 발견이 냅스터 슈퍼서버에 의해 대신 이루어지고 있다. 예를 들어 한 Peer가 냅스터에 로그인을 하면, 다음과 같은 과정으로 Peer 발견과 컨텐츠 목록 발견이 처리된다.

① 로그인한 Peer가 냅스터 서버의 목록에 등록된다.

② 냅스터 검색엔진이 Peer의 공유폴더에서 파일을 검색한다.

③ 그 목록을 유지하여 검색 가능한 DB로 만든다.

④ 새로 로그인하는 Peer는 냅스터가 서비스한 Peer들의 목록과 컨텐츠를 검색할 수 있다.

⑤ Peer간의 1:1 직접 전송으로 통신이 이루어진다.

이런 과정을 통하여 로그인만으로 냅스터에 연결된 Peer들을 한 번에 검색할 수 있는 것이다. 냅스터는 P2P 방식처럼 보이지만 실제로는 종래의 서버/클라이언트의 모델이기 때문에 냅스터라는 슈퍼서버가 24시간 쉬지 않고 가동해야 하며 냅스터가 '유지' 하고 있는 피어들의 컨텐츠 목록이 자료 공유를 불법적인 것으로 만들고 있다. 이 구조는 기존 클라이언트-서버 네트워크 구조에서 사용자들이 특정 파일을 중앙 서버로부터 받아감으로써 발생되는 네트워크 트래픽을 일반 피어들에게 분산시키기 때문에 각 피어들의 자원을 빠른 속도로 공유할 수 있다는 장점을 가지고 있지만, 자원의 공유을 위해서는 모든 피어들이 일단 중앙 서버에 연결해야 하기 때문에 클라이언트-서버 모델의 중앙 서버로 인한 단점을 계속해서 일부 갖게 된다.

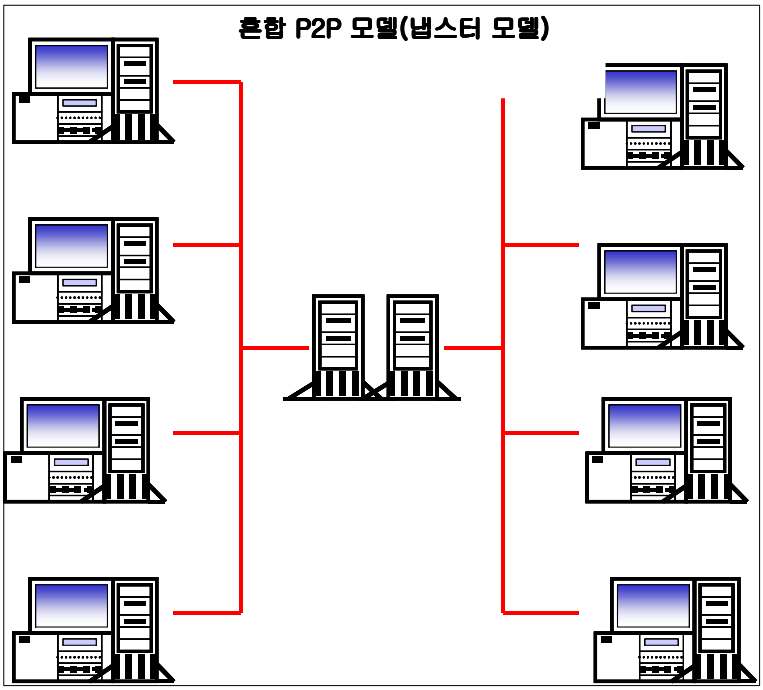

그림 2-1. 혼합형 P2P구조(냅스터)

Figure 2-1. Hybrid P2P structure(Napster)

순수 P2P 네트워크 구조인 그림2-2는 피어들에게 서비스를 제공하는 특정 서버를 두지 않고, 모든 피어들로 하여금 서버와 클라이언트의 기능을 동시에 가지게 하는 방식인데 대표적인 모델이 누텔라(Gnutella) 방식이다.

이 방식을 사용하는 시스템의 각 피어들은 최초 실행 시 네트워크 그룹에 이미 참여하고 있는 피어에게 연결하게 되며, 이러한 방식으로 모든 피어들이 피라미드식으로 연결되어 무제한의 자료를 공유하게 된다. 따라서 이 구조는 특정 서버를 두지 않고도 서비스를 제공하거나 제공받을 수 있다는 장점을 가지게 된다.

누텔라의 서비스 및 개체 발견 방식은 각 Peer가 서로에게 연결된 로컬 상의 Peer들에게 자신이 원하는 것을 얻기 위해 서로에게 질문을 보냄으로써 원격지에 있는 정보를 찾을 수 있다. 하지만 하나의 자료를 찾기 위해 각 Peer가 서로에게 질문을 보내는 상황이 중복되기 때문에 상대적으로 사양이 낮은 Peer들의 컴퓨터는 오버헤드를 일으키기 쉽고, 자신이 보낸 패킷의 수명을 TTL(Time To Live) 값으로 관리하기 때문에, 맡은 임무를 다하고도 네트워크 상에 떠돌아다니는 패킷이 많아져 결과적으로 네트워크 전체가 오버헤드를 일으키게 된다. 누텔라 방식은 서로를 찾고 그 안에서 자료를 발견하는 데에 네트워크 소비가 심했다.

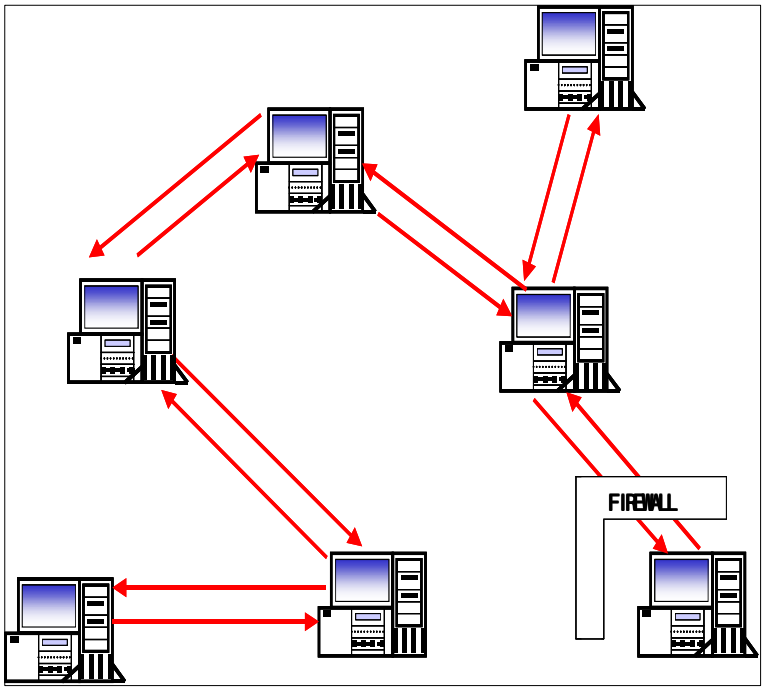

그림 2-2. 순수 P2P구조(누텔라)

Figure 2-2. Pure P2P structure (Gnutella)

제 2 절 JXTA 플랫폼

P2P 개발 플랫폼은 SunMicrosystems에서 개발한 JXTA외에 Oregon 대학교에서 만든 모바일 Ad hoc network을 위한 Proem, 이탈리아 Bologna 대학교의 Anthill 등이 이것의 형성 움직임을 주도하고 있다. 본 논문에서 구현할 기술은 JXTA 애플리케이션 개발 플랫폼을 이용하여 적용된다.

JXTA는 JuXTApose의 약자로 전통적인 Client/Server 방식과도 어울리면서 PC와 PC, PC와 가전제품, PC와 PDA 등이 서로 교신할 수 있는 애플리케이션 개발환경을 제공하는 것이 목표이며, 오픈소스 개발 프로젝트 형식으로 진행되고 있다.

1. JXTA 구조

JXTA의 구성요소와 서비스 기능을 나타내는 그림 2-3은 P2P 모델을 위해 기본적인 프로토콜만 지원하고 있는데, 이는 P2P를 이용하는 모든 서비스나 응용 프로그램들이 기존 형식을 바꾸지 않고 JXTA를 쉽게 활용 할 수 있도록 하기 위해서이다[5].

JXTA 플랫폼은 컴포넌트 관점에서와 서비스 관점으로 나누어 살펴볼 수 있다. 우선 컴포넌트 관점에서는 Peer와 Peer Group, Pipe, 그리고 Service 개념이 있다. 컴포넌트 개념에서 Peer는 JXTA에서 가장 기초가 되는 요소로써 JXTA 네크워크에서 서비스를 제공하거나 서비스를 이용하는 하나의 클라이언트라고 할 수 있다. 각 Peer는 JXTA 네트워크에서 유일한 Peer ID로 구분이 되기 때문에 하나의 Peer가 Peer ID를 가지고 동적으로 물리적 앤드 포인트를 바꿀 수 있다는 장점이 있다.

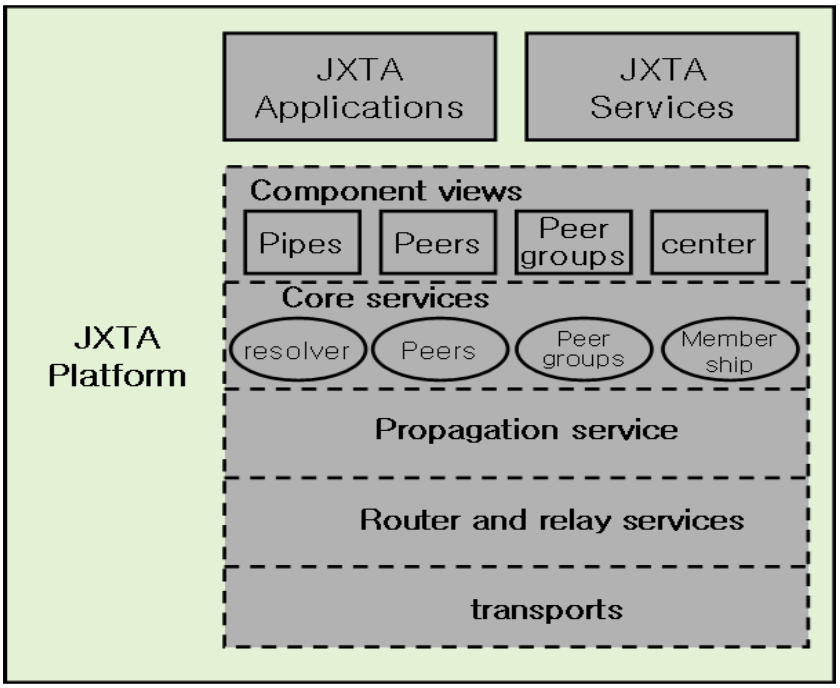

그림 2-3. JXTA의 계층 구조

Figure 2-3. JXTA Hierarchy

Pipe는 Peer가 서로 간에 통신을 할 때 쓰여 지는 통신 채널이다. Peer간의 통신에 기존의 중앙화된 서버의 네이밍 서비스를 제공하는 네트워크 모델은 부적합하기 때문에 JXTA에서는 각 Peer의 앤드 포인트에 동적으로 Binding되는 Pipe로 Peer간의 Virtual 통신 채널을 제공하고 있다.

Peer Group이란 Peer들의 집합을 말한다. Peer들의 집합으로 공통된 Security와 공통된 서비스를 제공하는 Peer들의 집합이다. Peer Group에서는 Query가 그 Peer Group의 범위 안에서 방송되게 하여 네트워크의 과도한 소비를 줄인다. Service는 Peer가 서로 통신 할 수 있는 상태에서 한 Peer가 다른 Peer들을 위해 유용한 일을 할 수 있도록 허락하는 것이다. 이것은 Peer들 사이에 일시적인 클라이언트/서버의 관계가 성립되는 것처럼 보이지만, JXTA에서는 Peer Group 단위로 Peer Group에 참여한 Peer들이 병치되어 서비스를 제공할 수 있는 기능을 제공하기 때문에 기존의 서버/클라이언트와 같은 문제점을 일으키지 않는다.

JXTA의 서비스는 P2P 통신을 위해 Peer Group 단위로 여러 가지 서비스를 제공하고, Peer Group 마다 요구되는 사항에 따라 핵심적인 서비스만 제공하거나 추가적인 서비스를 첨가하는 등 제공될 서비스의 범위를 정할 수 있다. Peer Group 마다 제공되어야 하는 핵심적인 서비스로 Discovery 서비스, Member ship 서비스, Peer Resolver 서비스, Peer Information 서비스, Pipe Binding 서비스, Peer Endpoint 서비스 등이 있다.

Discovery 서비스는 JXTA 컴포넌트들인 Peer나 Pipe, Peer Group을 찾을 때 쓰여 지는 서비스이다. JXTA에서 Peer의 발견은 초기 P2P 방식인 누텔라의 심한 네트워크 소비의 문제점을 피하기 위해 Rendezvous 서비스를 제공해 주는 Rendezvous Peer를 제공한다.

Rendezvous 서비스가 질문 Query를 전파하는 방식인 그림 2-4와 같이 Rendezvous Peer는 Peer들이 Rendezvous Peer를 찾기 쉽도록 하기 위해 특별한 Advertisement를 네트워크 상에 방송하여 다른 Peer들이 쉽게 Rendezvous Peer를 찾을 수 있게 해 준다. 그런 후 Rendezvous Peer는 그림 2-5와 같이 네트워크 상에서 찾아진 컴포넌트들에 대한 정보를 나타내는 Advertisement 객체를 저장하고 있다가 Peer에서 이에 대한 Query를 해오면 곧바로 답을 주며, 다른 Rendezvous Peer와 서로 연결이 되어 있어서, 질문된 Query에 대한 답을 찾지 못했을 경우에는 연결된 Rendezvous Peer 들로 Query를 전파 해 준다. Member ship 서비스는 한 Peer Group에 가입하거나 떠나는 것 그리고 회원 구성에 대한 서비스를 책임진다. 그림 2-6과 같이 Resolver 서비스는 Peer Group으로 서비스를 제공할 수 있게 해주는 바탕이 된다. 즉 어떤 Peer가 Peer Group 서비스에 관심이 있을 때, 그 서비스에 대해 handler라는 것을 등록해 둔다. 이것은 Peer Group에게 Resolver Query를 보내는 것을 말한다.

Peer Group에서는 handler 이름으로 등록 되어진 Peer들 중 임의의 Peer에게 Peer Group 서비스를 할 책임을 나누어 주면서 Resolver Response를 보내게 되는 것이다. Peer Resolver 서비스는 Discovery 서비스나 Peer Endpoint 서비스 등의 하위 부분에서 동작 한다. Peer Information 서비스는 다른 Peer들의 연결 가능성과 상태를 알 수 있게 해주는 서비스이다. Pipe Binding 서비스는 Peer의 Pipe와 물리적인 앤드 포인트를 동적으로 연결시켜주는 서비스이다. Peer Endpoint 서비스는 Peer가 다른 Peer에게 메시지를 보내 라우팅 정보를 물어볼 수 있게 해 주는 서비스이다.

JXTA는 이와 같은 핵심적인 서비스를 제공하며 그 외에 추가적인 서비스로 Content Management Service(CMS)와 Rendezvous 서비스, Relay 서비스,

Router 서비스를 제공한다. CMS는 컨텐츠를 네트워크 상에 공유하고, 공유된 컨텐츠의 목록을 받아올 수 있게 해 주며, 받아온 정보를 이용해 CMS를 통해 컨텐츠를 전송 받을 수 있게 해 준다.

Peer Endpoint 서비스는 Peer가 다른 Peer에게 메시지를 보내 라우팅 정보를 물어볼 수 있게 해 주는 서비스이다.

JXTA는 이와 같은 핵심적인 서비스를 제공하며 그 외에 추가적인 서비스로 Content Management Service(CMS)와 Rendezvous 서비스, Relay 서비스, Router 서비스를 제공한다. CMS는 컨텐츠를 네트워크 상에 공유하고, 공유된 컨텐츠의 목록을 받아올 수 있게 하며, 받아온 정보는 CMS를 통해 컨텐츠를 전송 받을 수 있게 해 준다.

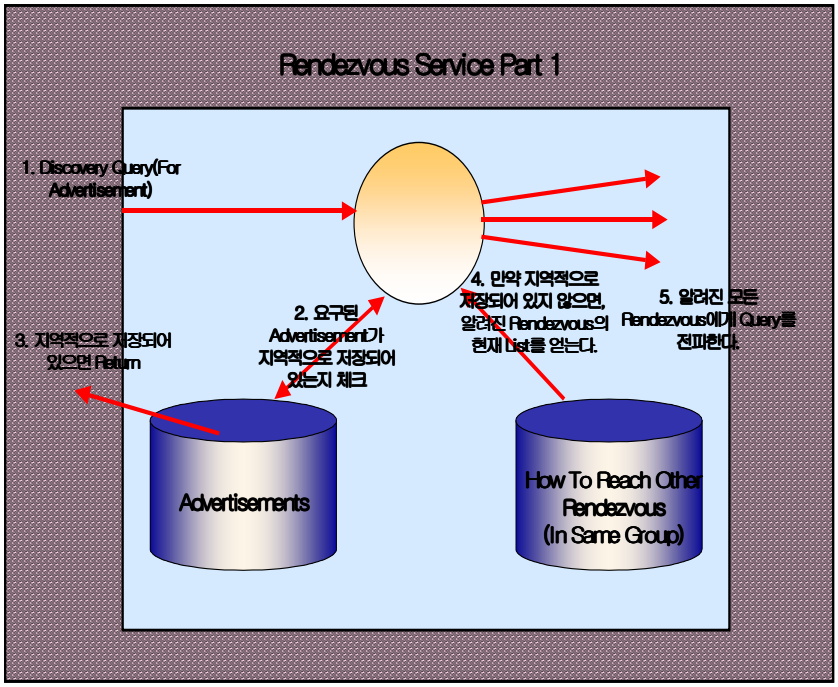

그림 2-4. Query를 받는 동작

Figure 2-4. Query receiving actions

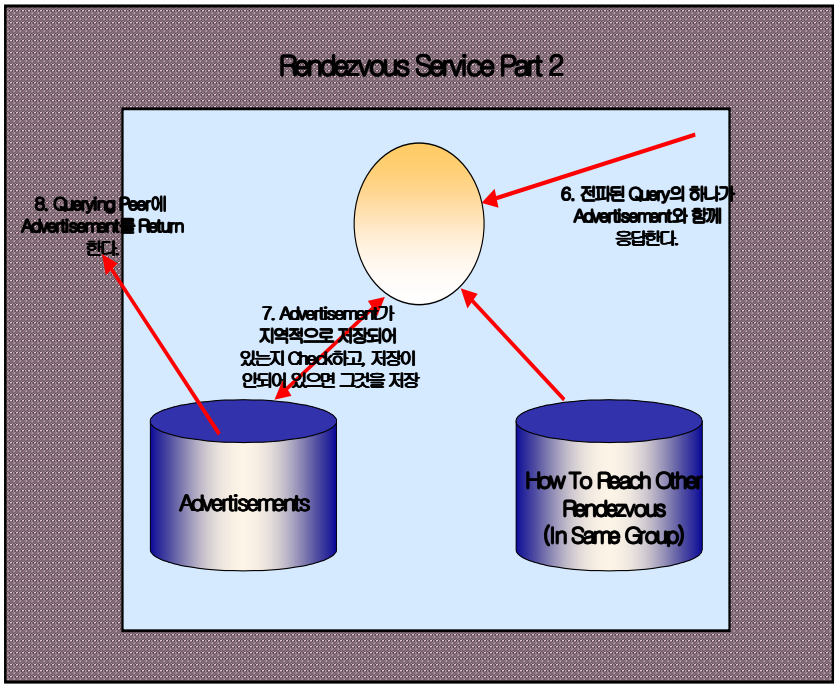

그림 2-5. Query에 응답하는 동작

Figure 2-5. Query responding actions

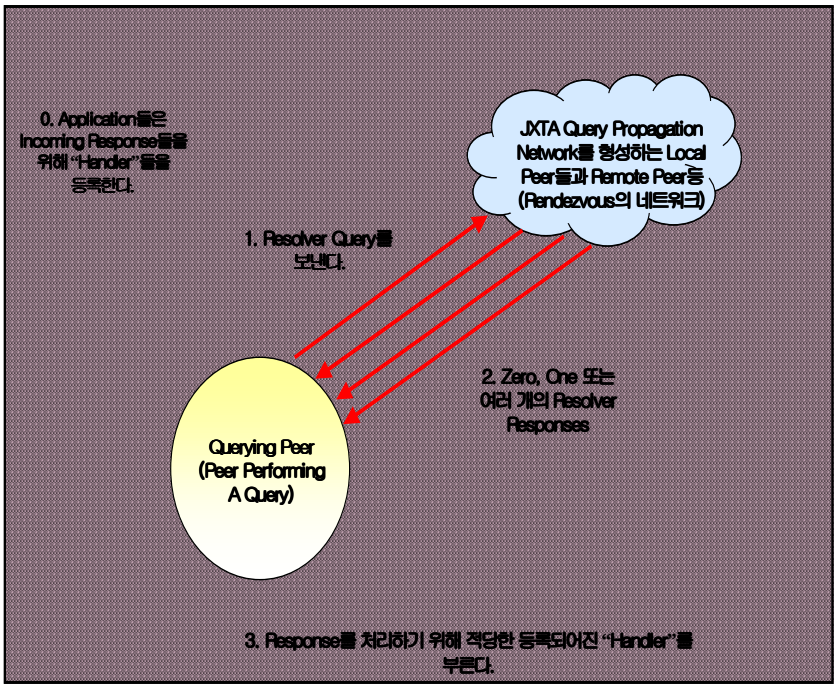

그림 2-6. Resolver 서비스의 동작 모델

Figure 2-6. Action model of resolver service

(3) JXTA의 메시지와 Advertisement

JXTA 메시지는 Peer들이 서로 통신할 때 사용된다. 이 메시지는 Text 문서일 수도 있고 Binary 데이터일 수도 있다. 보통은 HTTP 전송을 위해서는 XML이 TCP 전송을 위해서는 Binary 데이터가 사용된다. Text 문서를 위해 XML을 쓸 때의 장점은 플랫폼과 언어에 상관없이 쓰이고,문서를 확장할 때 단순히 태그요소만 첨가하면 된다.

Advertisement는 JXTA 메시지의 특별한 타입이다. Advertisement 이름처럼 Advertisement는 컴포넌트의 존재를 방송 하므로 표 2-1은 JXTA에서 이미 정의 된 Advertisement를 나타낸다.

Advertisement는 컴포넌트들에 관한 정보를 제공하며, XML 형식을 취하고 있다. XML 형식을 이용하기 때문에 Java는 물론 기타 언어와 WAP이나 PDA와도 연동이 용이하다. JXTA의 기본이 되는 JXTA.jar 파일의 경우 400K 정도의 경량이며, 브라우저 기본 내장 Virtual Machine(VM)을 통해서 실행이 가능하기 때문에 같은 애플리케이션을 리눅스와 윈도우, 솔라리스에서 각각 실행 시킬 수 있는 장점이 있다.

표 2-1. JXTA에서 정의한 Advertisement

Table 2-1. Advertisement defines from the JXTA

Advertisement	Advertisement에 포함된 태그 요소
Peer Advertisement	· 유일한 ID · Peer name · Peer가 속한 Peer Group과 Peer와 관련된 모든 Transport Endpoint
Peer Group Advertisement	· 유일한 Peer Group ID · Peer Group name · Peer Group과 관련된 모든 Peerg Group Service를 위한 Service Advertisement
Pipe Advertisement	· 유일한 Pipe ID · 제공할 수도 있고 그렇지 않을 수도 있는 Name · Pipe와 관련된 Peer ID를 제공한다.
Service Advertisement	· Service와 Service를 접속하는 방법들을 설명

(4) 파일 공유에 이용되는 Content Management Service (CMS)

Content Management Service(CMS)는 JXTA에서 제공하며 파일 공유를 관리해 주는 서비스이다. 표 2-2와 같이 CMS는 Peer Group 단위로 제공되는 서비스로 컨텐츠를 공유하고자 하는 모든 Peer들에게 컨텐츠를 공유하고 공유된 컨텐츠를 전송받을 수 있게 해주는 서비스이다. CMS는 Peer Group에서 데이터를 요청하는 Peer를 위해 두 가지의 서비스 객체를 제공한다.

ListContentRequest 서비스 객체는 Peer Group의 CMS에게 공유된 컨텐츠의 목록을 요청한다. ListContentRequest 서비스 객체 생성을 위해서는 2개의 인자를 필요로 한다. 첫 번째로 ListContentRequest가 동작할 Peer Group이 필요하고, 두 번째는 컨텐츠 요청시 컨텐츠의 데이터 중 특정 String을 포함한 것을 요청하는 String이다. 제한을 두지 않으려면 단순히 값을 가진 String을 인자로 넣어 주면 된다. GetContentRequest 서비스 객체는 CMS에게 특정 컨텐츠에 대한 전송을 요청하게 해 준다. GetContentRequest 서비스 객체 생성을 위해서는 3개의 인자를 필요로 한다. GetContentRequest가 동작할 Peer Group과 요청할 컨텐츠에 대한 정보를 가지고 있는 ContentAdvertisement, 요청할 컨텐츠가 저장될 Local File이다.

표 2-2. CMS 서비스 객체

Table 2-2. CMS Service object

서비스 객체	내 용
ListContentRequest	"어떤 컨텐츠들이 공유되고 있는지 목록을 보여 달라는 요청이다. 요청 시 사용자에 의해 컨텐츠의 내용 중 특정 문자열을 포함한 컨텐츠를 찾도록 컨텐츠의 범위를 제한할 수 있다.
GetContentRequest	"특정 컨텐츠를 나에게 전송해 달라는 요청이다. 컨텐츠를 공유하고 있는 Peer에서는 요청한 Peer로 컨텐츠를 전송 해준다.

가. CMS에서 ListContentRequest와 GetContentRequest 서비스 객체의 처리 동작

그림 2-7은 CMS에서 ListContentRequest와 GetContentRequest를 이용한 컨텐츠목록 공유와 컨텐츠 전송에 대해 나타낸다.

각각의 Peer는 CMS를 제공하는 서비스 역할을 제공한다. 로컬 Peer에 공유한 파일이 있으면 CMS에서는 공유된 파일에 대한 Index를 유지한다. 리모트 Peer로부터 파일 리스트 요청이 있을 때 CMS는 유지하고 있던 Index 정보를 응답하고, 파일 전송 요청이 있을 때에는 Index와 연결되어 있는 파일을 전송하도록 해 준다. 즉, 각각의 Peer들은 서비스와 클라이언트 기능을 모두 제공한다. Peer의 클라이언트 기능은 ListContentRequest 서비스 객체를 통해 CMS에서 공유된 컨텐츠 목록을 요청하는 기능과 GetContentRequest 서비스 객체를 통해 CMS에서 공유된 특정 켄텐츠에 대해 전송을 요청하는 것이다.

서비스 기능은 리모트 Peer로부터 요청받은 컨텐츠 목록에 대해 로컬 Peer에서 공유하고 있는 컨텐츠에 대한 ContentAdvertisement들을 응답 해 주고, 요청받은 컨텐츠 전송에 대해 로컬 Peer에서 공유하고 있는 컨텐츠를 전송 해 주는 것이다.

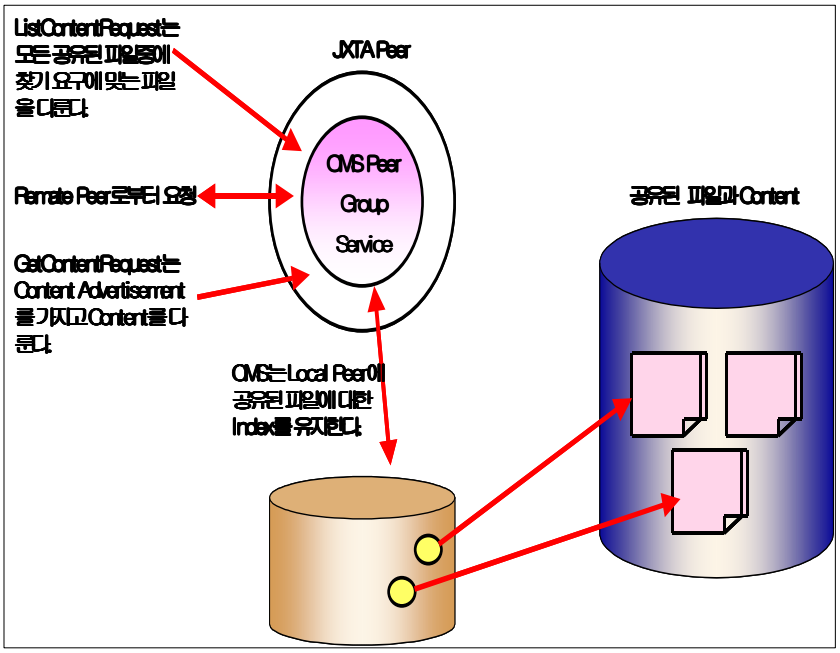

그림 2-7. ListContentRequest와 GetContentRequest

Figure 2-7. ListContentRequest and GetContentRequest

나. CMS에 사용되는 ContentAdvertisement

그림 2-8과 같이 ContentAdvertisement는 CMS에서 공유된 각각의 컨텐츠와 관련된 정보이다. 정보에는 컨텐츠 이름, 컨텐츠의 크기, mime 타입, ID, 설명 등이 있다. 만약 명확히 컨텐츠 이름과 mime 타입을 설정해 주지 않으면, 공유된 파일 이름에 기초한 디폴트 이름과 media 타입이 설정되며 ContentAdvertisement의 타입은 XML이고, ContentAdvertisement는 name 필드와 cid 필드를 반드시 필요로 한다. cid는 128bit의 MD5 checksum으로 컨텐츠를 표현함으로써 컨텐츠를 파일 이름으로 구분하지 않고 Peer Group 내에서 유일한 컨텐츠 ID로 구분한다.

해시 알고리즘 MD5방식은 메시지 압축 기능을 이용, 임의의 길이를 가진 메시지를 일정한 길이로 출력하는 역할을 수행한다. 이것은 보안상 주어진 출력에 대해 그 입력 값을 찾아내는 것과 주어진 입력에 대해 같은 출력 값을 갖는 또 다른 입력 값을 찾기가 계산상 불가능하며 같은 출력을 내는 임의의 다른 두 입력을 찾기 불가능하게 한다.

자료 식별자로써 MD5를 사용하는 이점은 문서 공유시스템에서 네트워크에 있는 많은 Peer들에 의해 쉽게 자료가 다운로드 되고 재 공유될 수 있다는 것이다. 다른 Peer들에게서 공유된 두 파일이 동일할 때 자료 이름이나 설명에 의존하는 것보다는 MD5를 이용하는 것이 자료를 선택하는데 용이하다. 따라서 MD5를 이용하면 어떠한 자료검색도 가능하게 되어 자료를 요청하는 Peer는 가장 가까운 거리에서 자료를 가지고 있는 Peer를 발견하게 된다.

```
<?xml version="1.0">
<!doctype jxta : ContentAdvertisement>
<jxta:contentAdvertisement>
    <name> index.html </Name>
    <cid> md5:abcdefghijklmnopqrstuvw </cid>
    <type> text/html </type>
    <length> 23987 </length>
    <description> Web site index </description>
</jxta:ContentAdvertisement>
```

그림 2-8. CMS에 사용되는 ContentAdvertisement

Figure 2-8. ContentAdvertisement used in CMS

제 3 장 P2P 자원 탐색 알고리즘

제 1 절 JXTA의 자원 탐색 메커니즘

모든 피어들은 자신의 자원정보를 가지고 있으며, 이를 자신이 속한 랑데부 피어에게 광고를 하게 한다. 이렇게 랑데부 피어들은 자신에게 소속되어 있는 모든 피어들의 자원정보를 가지고 있다. 이와 같이 구성된 JXTA 네트워크는 랑데부 피어의 자원정보를 이용하여 다른 피어로부터 피어의 자원정보를 사용하고자 할 경우 해당 자원으로의 접근이 가능하도록 포워딩을 하고 그림 3-1과 같은 과정을 통하여 해당 자원으로 접근 한다.

피어 A가 피어 B가 가지고 있는 광고정보를 탐색할 경우 피어 A는 자신이 속해 있는 랑데부 피어 1에게 쿼리를 전송한다. 피어 A로부터 쿼리를 전송받은 랑데부 피어 1은 자신이 소유한 피어들의 자원정보 중에서 검색을 한다. 만약 랑데부 피어 1이 피어 A가 찾고자 하는 자원정보가 없을 경우 인접한 랑데부 피어 2에게 쿼리를 전송한다. 랑데부 피어 1에게 쿼리를 전송받은 랑데부 피어2는 자신이 소유한 피어들의 자원정보 중에서 피어 A가 요구한 자원정보와 일치하는 자원정보를 검색하고, 일치하는 자원정보가 있을 경우 피어 A가 발생시킨 쿼리를 피어 B에게 포워딩을 시킨다. 만약 랑데부 피어 2에서도 해당 자원정보가 없을 경우 다른 랑데부 피어로 쿼리를 전송하도록 하여 계속적인 자원정보 탐색을 하도록 한다. 한편 쿼리를 받은 피어 B는 피어 A에게 직접 연결하며 쿼리에 해당하는 응답을 한다.

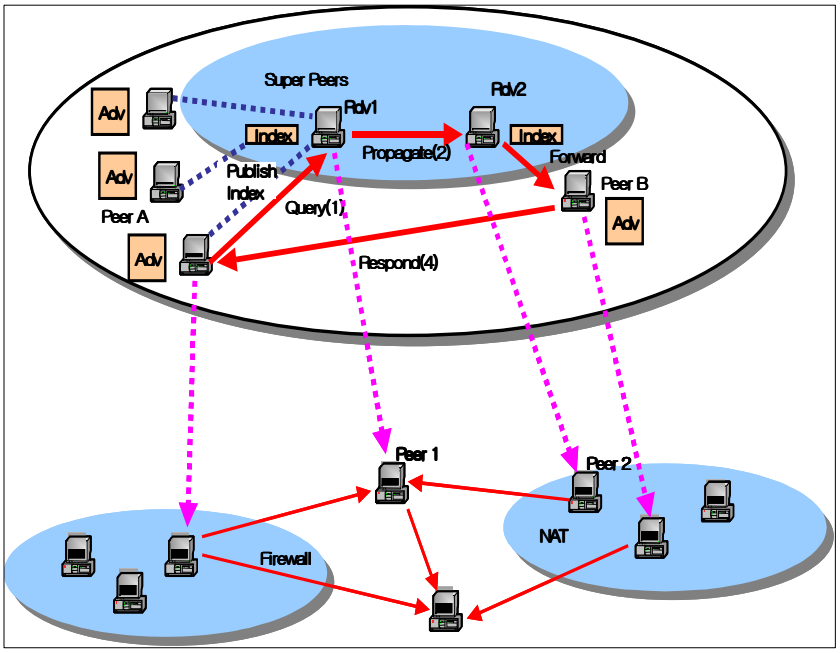

그림 3-1. JXTA의 자원 탐색 알고리즘

Figure 3-1. JXTA resource searching algorithm

제 2 절 랑데부 피어 순회 탐색 메커니즘

각 피어들은 자신이 찾고자 하는 피어의 자원정보를 찾을 때 까지 랑데부 피어들을 순회하면서 검색하는데, 이때 자원정보 검색 시 소요되는 검색 비용은 랑데부 피어의 순회 탐색 횟수에 따라 비례한다.

JXTA는 대규모의 네트워크 상에서 각 랑데부 피어들은 상호 연결 상태를 이용함으로써 자신이 소유하지 않은 피어의 자원정보를 검색하고 네트워크에 산재되어 있는 자원정보를 찾을 수 있도록 한다.

현재 JXTA는 그림 3-2와 같이 Limited Range Walker란 알고리즘을 이용하여 자원 정보를 검색하는 구조를 하고 있다. 우선 피어 P1은 자신이 소속된 랑데부 피어 R7에게 자신의 자원정보를 광고하고 피어 P1의 자원정보를 소유한 랑데부 피어 R7은 피어 P1의 자원정보를 사용하길 희망하는 모든 피어의 쿼리를 받아 포워딩 시킨다. 이때 피어 P2가 피어 P1의 자원정보를 이용하고자 할 경우 피어 P2는 자신이 소속되어 있는 랑데부 피어 R3에게 피어 P1의 자원정보 검색 쿼리를 전송한다. 쿼리를 전송 받은 랑데부 피어 R3는 자신이 소유한 피어들의 자원정보 중에 피어 P1의 자원정보가 없음을 알고 인접해 있는 랑데부 피어 R5에게 쿼리를 전송한다. 쿼리를 받은 R5는 다시 자신이 소유한 자원정보 테이블에 없음을 확인하고, 다른 랑데부 피어에게 자원정보 검색 쿼리를 전송하도록 한다. 이때, 랑데부 피어 R5는 자신의 위치에서 상하에 위치한 랑데부 피어 R6와 랑데부 피어4에게 쿼리를 전송하도록 함으로써 동시에 2개의 랑데부 피어에게 쿼리를 하게 되고 좀 더 빠른 쿼리 결과 값을 얻을 수 있다.

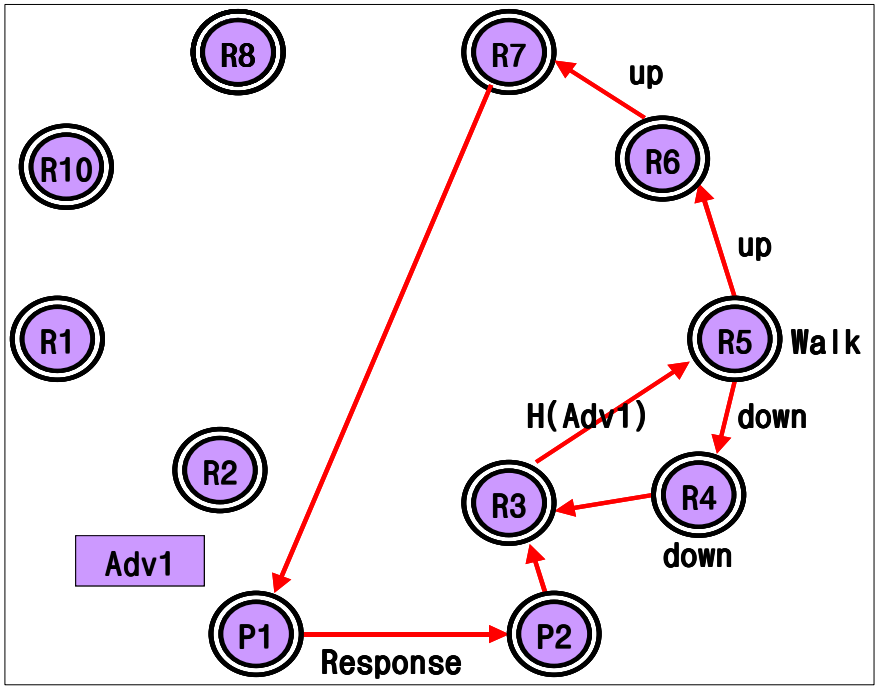

그림 3-2. 랑데부 피어 순회 탐색 메커니즘

Figure 3-2. Rendezvous peer circuit searching mechanism

제 3 절 랑데부 피어 순회 탐색의 단점

해당 피어의 자원정보를 검색 시 동시에 2개의 랑데부 피어의 순회 탐색이 이루어져 빠른 결과는 얻을 수 있어도 이로 인해 엄청난 양의 자원정보 검색 쿼리가 발생한다.

그림 3-3과 같이 찾고자 하는 피어의 자원정보가 가장 마지막에 검색될 랑데부 피어에 있을 경우 기존의 랑데부 피어 순회탐색 메커니즘 Limited Ranged Walker 방식을 이용할 경우 많은 쿼리로 인한 네트워크 트래픽이 발생하며, 이로 인한 검색 비용 상승과 검색완료에 대한 시간이 증가한다. 이는 각 랑데부 피어들 간의 쿼리가 발생할 경우 상호 연결 상태에 대한 정보만을 가지고 자원정보 검색 쿼리를 전송하므로, 검색결과에 신뢰성이 떨어지게 된다. 또한 모든 쿼리는 랑데부 피어를 통해 다음 랑데부 피어로 전송되기 때문에 쿼리 전송 시 많은 시간이 걸리며, 쿼리 전송도중 랑데부 피어가 장애로 인해 다음 랑데부 피어로 쿼리를 전송하지 못할 경우 검색 과정이 중단되는 현상이 일어나게 된다.

여기서 가장 중요한 문제점은 랑데부 피어를 확장하여 검색하기 위해 발생된 자원정보 검색 쿼리가 자신으로 돌아오는 롤백현상이 일어날 경우 자신과 인접한 랑데부 피어에 자신이 찾고자 하는 자원정보가 없는 것으로 간주하여 검색 영역을 더 확장해야 한다. 이때 피어의 자원정보 검색을 요청받은 랑데부 피어는 피어에게 검색 영역을 얼마나 어디까지 검색할 것인지 질의를 하게 된다. 이러한 과정으로 인해 비용검색은 더 증가하며, 검색 시간이 더 많이 걸리게 된다.

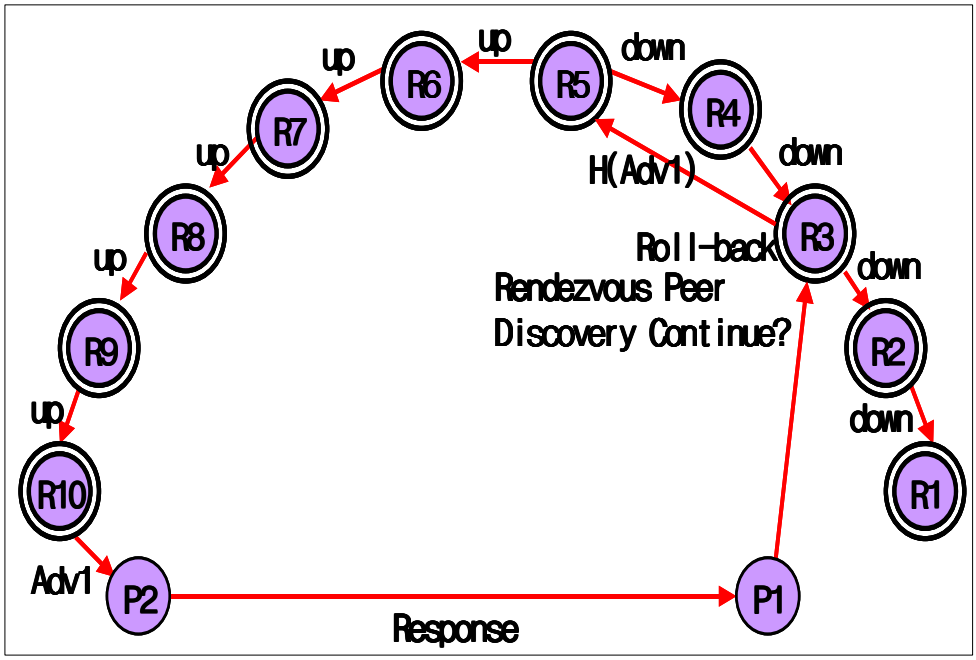

그림 3-3. 랑데부 피어 순회 탐색의 Roll-back 현상

Figure 3-3. Roll-back phenomenon in rendezvous peer circuit searching

제 4 절 P2P 네트워킹을 위한 분산 해쉬 테이블 방식

분산 모델을 이용한 P2P 파일 공유 네트워크인 경우, 중앙 서버를 통한 공유의 경우와 달리 파일들에 대한 디렉토리 서비스를 일괄적으로 제공하지 못하기 때문에 전체 노드들에 검색 요청 메시지를 보내 응답을 기다려야 한다[6][8]. 이때 중복 트래픽이 과도하게 발생하여 네트워크 트래픽 부하가 급속히 증가하게 된다. 또한 검색에 대한 응답의 불규칙성 및 검색 노드 제한에 따라 검색 결과의 완전성에도 문제를 갖게 된다. 이와 같이 비구조적인 P2P 네트워크를 형성하는 Gnutella와 같은 공유 방식의 문제를 해결하기 위해서 P2P 네트워크 형성에 구조적인 특성을 부여하여 검색 효율성을 제고하는 방법이 필요하게 되었다.

최근에는 분산 해쉬 테이블(DHT)를 이용한 P2P에 대한 연구들이 활발하게 진행되고 있는데 해쉬 테이블은 실제 값과 키를 위해 두 개 배열을 사용하여 검색을 빠르게 하기 위한 자료 구조의 하나이며 이를 분산 환경에 적용한 것이 분산 해쉬 테이블이다. 분산 해쉬 테이블 개념 이해를 돕기 위해 파일 공유 P2P를 예로 들면, 분산 해쉬 테이블을 사용하는 가장 큰 목적은 파일 검색을 빠르게 하기 위한 것으로 빠른 검색을 위해 파일 공유 네트워크를 구성하는 노드들에 해쉬 테이블 버킷을 각 노드에 분산 시키는 것이다. 검색은 해쉬 테이블 버킷을 찾는 것으로 파일 공유 네트워크 측면에서는 버킷을 가진 노드를 찾아서 분산 해쉬 테이블 버킷과 노드를 연결시키기 위한 방법이 필요하다. 이를 위해 노드들은 노드 식별자를 사용하여 버킷을 나타내도록 하고 각 파일들을 버킷과 연결시키기 위해 파일 이름을 해쉬한 식별자를 사용한다. 노드 식별자와 해쉬된 파일 이름은 동일한 크기의 이름 공간(비트 수)를 가진다. 파일 공유 네트워크는 노드 식별자를 사용하여 한 노드가 연결하는 노드(이웃 노드)들이 결정되는데 이런 이유로 분산 해쉬 테이블 기반 P2P 네트워크를 구조적이라 한다.

분산 해쉬 테이블 기반 P2P 파일 공유에 필요한 동작은 크게 파일 등록, 파일 검색 그리고 메시지 라우팅으로 구분된다.

첫째, 파일 등록은 가지고 있는 파일을 공유하기 위해 분산 해쉬 테이블에 등록하는 것인데, 이것은 해쉬 테이블의 ***Insert()***와 유사한 동작을 한다. 차이점은 해쉬 테이블 버킷이 분산 되어 있으므로 이를 찾기 위한 과정이 필요하며,. 먼저 등록하고자 하는 파일의 이름을 사용하여 어떤 노드(버킷)에 등록할 것인지를 결정해야 한다. 이때 해쉬 함수가 사용된다. 파일 이름을 해쉬한 값은 목적지 노드(버킷)에 대한 정보를 제공한다. 등록 메시지와 함께 파일 이름을 해쉬한 값을 보내면 라우팅에 따라 목적지 노드에 전달되어 저장된다.

둘째, 파일 검색은 대상 파일 이름을 사용하여 목적지 노드에 대한 정보를 얻은 후 이를 검색 메시지와 함께 보내어 파일을 실제 소유하고 있는 노드의 정보를 얻는 것으로 ***Lookup()***과 유사하며, 검색 메시지가 목적지 노드에 전달되기 위해서는 메시지 라우팅이 필요하다.

마지막으로 메시지 라우팅은 메시지에 포함된 목적지 노드 정보를 사용하여 메시지를 노드에 전달하는 것인데 노드는 이웃 노드들에 대한 라우팅 테이블을 가지며 메시지를 목적지 정보와 이웃 노드들의 노드 식별자를 비교하여 근접한 이웃 노드로 메시지를 전달한다. 그림 3-4의 예는 Insert와 Lookup 함수가 Hash 함수와 함께 사용되며 Key로 파일 이름 “C”인 파일을 DHT상에 Insert 하고 Lookup 한다. Data는 파일 내용이나 파일을 가진 현재 노드에 대한 정보이다.

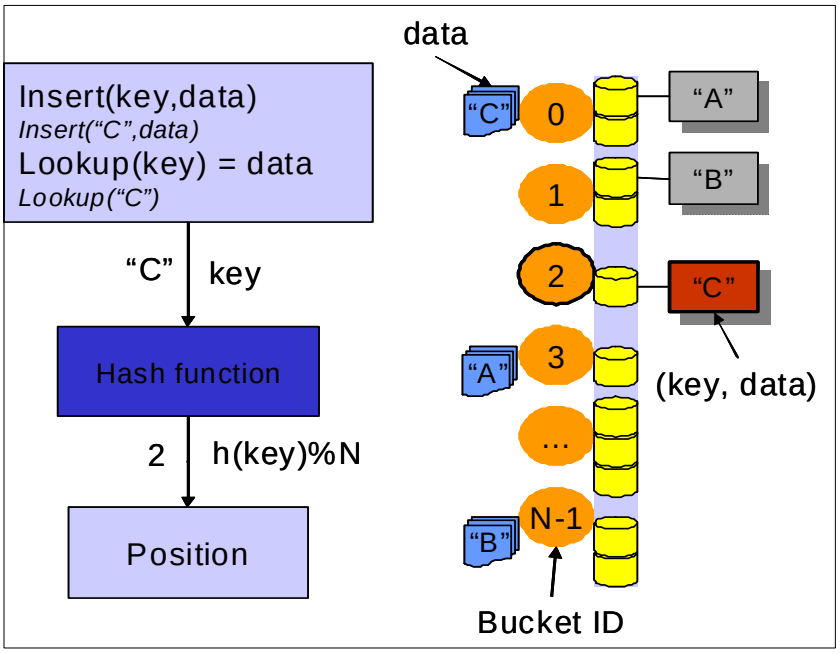

그림 3-4. DHT의 개념도(Insert와 Lookup)

Figure 3-4. DHT concept chart

제 4 장 JXTA기반의 P2P시스템 설계

JXTA의 가장 중요한 설계 선택중의 하나는 피어에 의해 사용되는 운영체제나 개발언어를 가정하지 않는 것이다. JXTA 프로토콜 규격은 명시적으로 네트워크 피어가 소규모 임베디드 장치에서 대규모 슈퍼컴퓨터 클러스터에 이르는 모든 종류의 장치가 될 수 있다. 이러한 기술에 J2ME의 모바일 기능을 추가한 JXME를 사용하여 기존의 제한된 메시지 전달방법을 탈피하여 JXTA기반의 P2P시스템을 만드는데 있다. 커뮤니케이션 시스템은 사용자 등록이나 주소 지정 스키마 관리가 중앙 서버에 결코 수행되지 않으며 JXTA파이프는 피어상에서 가상화 하고, 이 피어는 이종의 엔드 포인트 상에서 가상화함으로써 피어사이에 물리적인 직접 연결이 없어도 피어간 통신이 전달된다. JXTA 프로토콜은 모든 구현 기술에 독립적으로 설계된 반면 JXTA 레퍼런스 구현은 자바 플랫폼에 구현되고. 이 레퍼런스 구현은 자바 API를 사용하여 JXTA프로토콜 메시지를 래핑하고 프로그램 방법으로 자바 애플리케이션에서 JXTA 네트워크에 엑세스한다.

모든 메시지는 XML포맷 아래 SOAP, UDDI 등의 표준 프로토콜에 의해 애플리케이션간의 데이터를 교환하여 언제 어디서나 다양한 플랫폼에서 제약 없이 각종 서비스를 제공하고 받을 수 있게 한다.

제 1 절 릴레이 서버 구성

릴레이 서버는 모바일 피어와 같이 지속적으로 액세스할 수 없는 피어에게 보내지는 메시지를 저장하고 포워딩 해주는 서버로서 모바일 기기에서 할 수 없는 작업을 대행하여 피어에게 제공한다.

그림 4-1는 릴레이 서비스에서의 메시지 전송과 유지에 관한 그림으로 릴레이 서비스가 메시지를 전송하거나 수신할 때 릴레이 서비스의 역할을 보여준다. 릴레이 서비스가 모바일 클라이언트를 대신해서 메시지큐를 유지, 관리하며 이것은 모바일 클라이언트가 지속적으로 네트워크에서 액세스 할 수 없고, P2P 네트워크 상에서 사라져 다시 돌아오지 않을 수도 있기 때문이다.

그림 4-2와 같이 릴레이 서버는 직접적으로 네트워크에 참여할 수 없는 모바일 피어와 같은 네트워크 피어를 위하여 사용하는데 릴레이 서버는 HTTP전송을 사용하여 메시지를 처리한다. 릴레이 서버는 PDA의 제한된 환경 하에서 처리가 어려운 일을 맡아서 처리하는데 피어의 요청에 따라 리스를 생성하고 관리하며, 피어에게 메시지가 오면 대신 받아 처리한 다음 해당 피어에게 전달한다. 리스 기반의 메커니즘은 클라이언트 피어가 네트워크에 사라지거나 나타나지 않을 때 자원을 반환하고, 릴레이 서비스상의 존재하는 모든 클라이언트는 규칙적으로 리스기한을 연장하여 중계기 서비스 상에 있는 자신의 큐가 종료되지 않도록 한다.

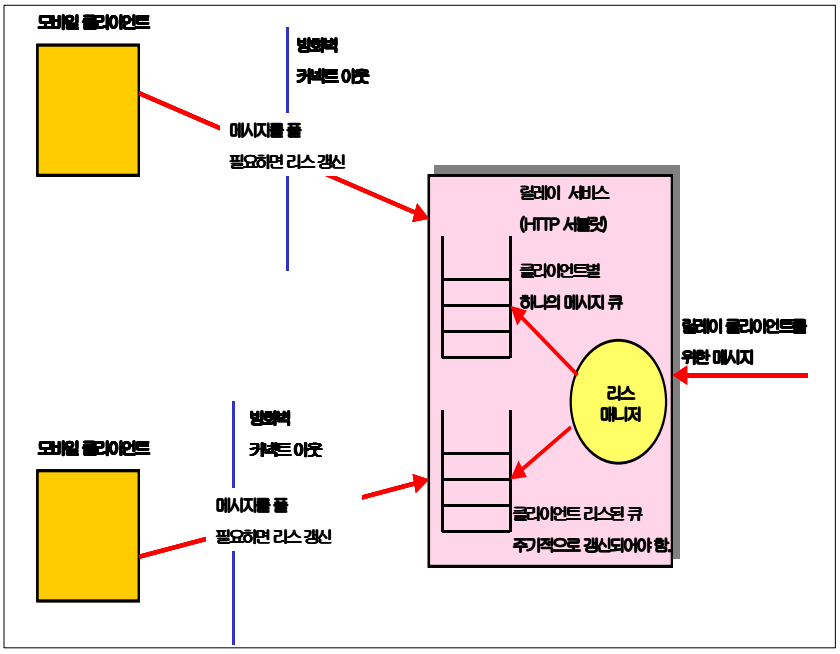

그림 4-1. 릴레이 서비스 메시지 전송

Figure 4-1. Relay service message forwarding

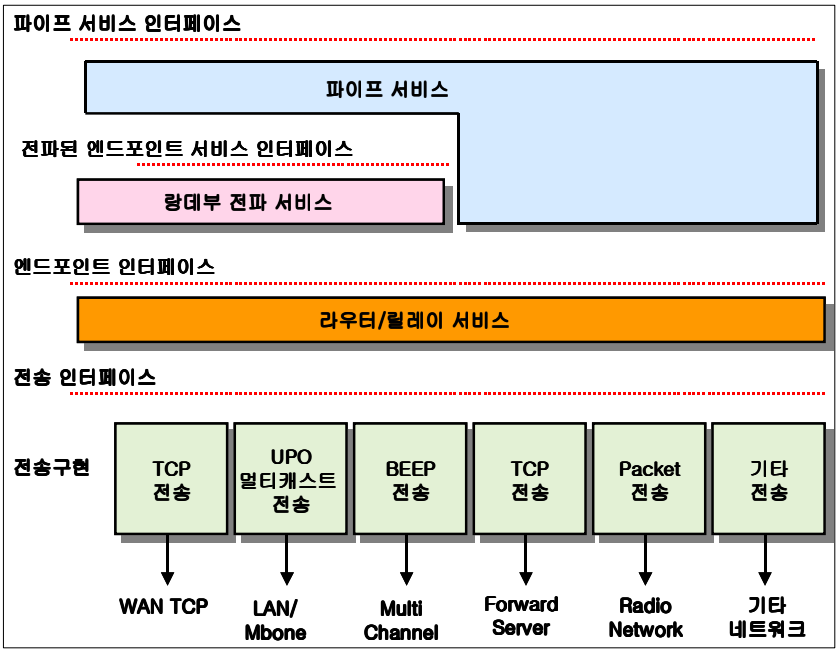

그림 4-2. 릴레이 서버 구조

Figure 4-2. Relay server structure

제 2 절 클라이언트 프로그램

커뮤니케이션 시스템의 테스트는 로컬에서 진행하며 모바일 기기는 모바일 시뮬레이터를 사용하며, 클라이언트의 구성은 그림 4-3과 같이 파이프, 그룹, 디스커버리, 메시징 서비스 구조를 갖는다.

JXTA네트워크 환경에서는 피어들이 파이프를 사용하여 메시지를 주고받는데 모든 피어들은 파이프를 두 개씩 가지고 있다. 하나는 Output pipe이고, 다른 하나는 Input pipe이다. Peer Group은 동일한 관심사와 조건을 가진 피어들의 모임이며 각각의 Peer Group은 특유의 ID로 구별되고, peer group에 참여하는 peer들을 제한할 수 있는 보안 정책도 마련되어 있다. 또한 JXTA의 모든 통신은 메시지를 주고받음으로 이루어지는데 이렇게 오고가는 메시들을 JXTA Message라 하며, 메시지들은 xml과 같은 표준 format으로 정의되고, Pipe서비스나 Endpoint 서비스 등에 사용된다.

클라이언트는 search 메서드를 사용하여 사용자를 발견하고 탐색하는 기능과 그룹을 발견하고 피어를 탐색하여 버디리스트에 추가하는 기능, 파이프를 생성하여 피어 간에 메시지 전달기능과, 그룹을 생성하고 그룹에 참여할 수 있는 기능을 구현하였다.

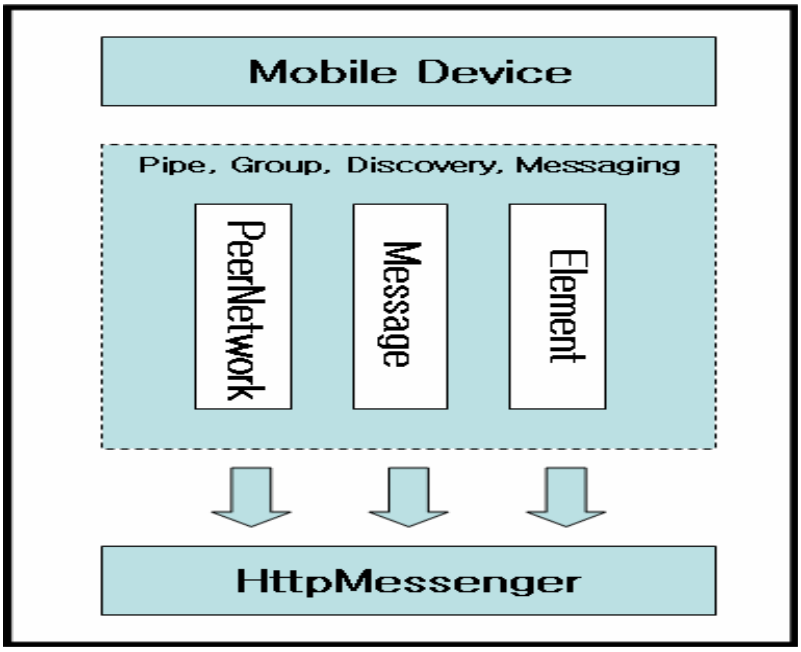

그림 4-3. 모바일 클라이언트피어 구조

Figure 4-3. Mobile client peer structure

제 3 절 DHT 프락시 피어 탐색 알고리즘 설계

JXTA P2P 네트워크 환경에서 프락시 피어를 두어 각 랑데부 피어 간에 발생하는 탐색 비용을 효율적으로 줄이며, 자원정보 검색 쿼리 회귀에 따른 쿼리 손실 해결책을 제안한다.

기존의 자원정보 탐색 방법은 랑데부 피어간의 연결 상태에 대한 정보만을 이용하여 검색하는데, 이로 인해 자원정보의 검색 비용이 높게 나와 자원 정보를 탐색과 사용함에 있어 많은 시간적 지연과 트래픽이 다수 발생하였다. 이러한 문제점을 해결하기 위해서 각 랑데부 피어 검색 시 우선순위를 두어 자원정보 탐색 시 효율적인 검색 비용을 제공하고자 한다.

어떤 피어가 광고 정보를 찾고자 할 경우, 자신이 속한 랑데부 피어에게 검색을 요청하고, 자신의 상위에 위치한 랑데부 피어에 찾고자 하는 광고가 없으면 다른 랑데부 피어로 자원정보 검색 쿼리를 전송하여 확장하도록 한다. 이때 랑데부 피어가 전파한 자원 정보 검색 쿼리가 자신에게 되돌아 올 경우 모든 랑데부 피어의 정보를 가지고 있는 프락시 피어로 쿼리를 전파한다.

프락시 피어에서는 자신이 소유한 랑데부 피어의 정보를 이용하여 다른 랑데부 피어로 포워딩을 유도하는데, 두 가지 우선순위를 두어 전송한다.

첫째, 프락시 피어는 이미 검색한 랑데부 피어들을 제외하고, 그 중에 가장 많은 피어들을 소유한 랑데부 피어로 연결을 한다. 이는 많은 피어를 소유한 랑데부 피어에게 찾고자 하는 광고정보가 있을 확률이 높기 때문이다.

둘째, 프락시 피어는 이미 검색한 랑데부 피어들을 제외한 다른 랑데부 피어 중 최근에 등록, 수정한 랑데부 피어로 연결을 시킨다. 이는 최근에 등록된 랑데부 피어일수록 피어의 정보가 유동적인 JXTA 네트워크에서신뢰성이 높기 때문이다.

프락시 피어는 위에서 설명한 바와 같이 두 가지 우선순위를 이용하여 가장 많은 피어 정보를 가지고 있는 랑데부 피어와 가장 최근에 랑데부 피어 정보가 등록, 수정된 랑데부 피어로 두개의 쿼리를 전송한다.

그림 4-4에서는 기존의 방식과 동일하게 두 번의 쿼리가 발생하고, 찾고자 하는 광고 정보를 찾을 때 까지 이와 같은 과정을 연속적으로 한다. 만약 찾고자 하는 광고 정보를 찾으면 랑데부 피어의 해쉬 테이블을 이용하여 피어에게 H(Adv)쿼리를 전파하게 되고, 쿼리를 받은 피어는 쿼리를 전송한 피어에게 응답을 한다.

피어 P1은 자신의 자원정보를 랑데부 피어 R10에게 광고를 하면 피어 P1의 자원정보를 관리하는 랑데부 피어 R10은 자신의 아이디와 자신이 관리하는 피어 수, 이 정보가 등록 또는 수정된 시간을 프락시 피어에게 저장하도록 한다. 이때 피어 P2는 피어 P1의 자원정보 사용을 원할 경우 자신이 소속된 랑데부 피어 R3에게 자원정보 검색 쿼리를 전송하고, 랑데부 피어 R3은 자신이 소유한 해쉬 테이블을 이용하여 자원정보를 검색한 후 자신에게 해당 자원정보가 없음을 확인한 후, 자신과 피어 자원정보를 주고받은 랑데부 피어 R5에 쿼리를 확장시킨다. 이후 랑데부 피어 R5에서 자원정보를 검색하고 역시 해당 자원정보를 검색한 후 자원정보가 없음을 알고 자신의 위치에서 상하위에 위치한 랑데부 피어 R4, R6에게 쿼리를 전송한다. 다시 랑데부 피어 R4, R6은 자신의 해쉬 테이블에서 검색을 한 후 자원정보 검색 쿼리를 더 확장시켜 랑데부 피어 R7, R3에게로 쿼리를 전송시킨다. 이때 R7에게 해당 정보가 없고, 랑데부 피어 R3에게 다시 롤백 되었음을 감지하고, 더 이상의 랑데부 피어로 확장을 하지 않고, 모든 랑데부 피어의 정보를 가지고 있는 프락시 피어에게 쿼리를 전송시킨다.

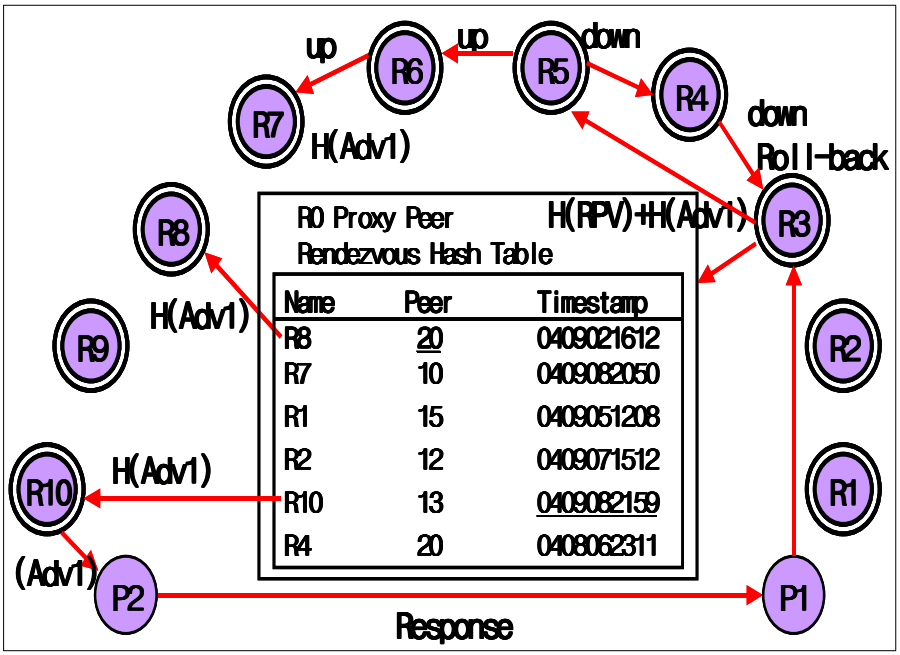

그림 4-4. DHT 프락시피어 탐색 알고리즘

Figure 4-4. DHT proxy peer search algorithm

이후 프락시 피어는 자신이 소유한 랑데부 피어의 해쉬 테이블에 저장되어 있는 랑데부 피어의 이름과 각 랑데부 피어가 관리하는 피어의 수, 그리고 랑데부 피어의 정보를 이용하여 자원정보 검색 쿼리를 전송한다. 이때, 이미 검색한 랑데부 피어를 제외한 랑데부 피어를 대상으로 피어의 수가 가장 많은 피어와 최근에 등록되거나 갱신된 랑데부 피어에게 자원정보 검색 쿼리를 전송하는 과정을 통해 가장 많은 피어를 가지고 있는 랑데부 피어 R8과 가장 최근에 등록된 랑데부 피어 R10에게 쿼리를 전송하게 되고, 피어 P1의 자원정보를 가지고 있는 랑데부 피어 R10은 피어 P1에게 쿼리를 전송한다. 쿼리를 전송받은 피어 P1은 다른 랑데부 피어를 이용하지 않고 피어 P2로 직접 응답을 보내고, 이후 피어 P1과 P2가 직접 서비스를 요청하고 제공하도록 한다. 이와 같이 프락시 피어는 랑데부 피어의 자원정보 저장과 랑데부 피어 검색에 따른 프락시 피어의 발생하는 시스템 부하를 최소한으로 줄이기 위해 접근에 필요한 랑데부 피어의 이름과 랑데부 피어가 소유한 피어 수, 랑데부 피어 자원정보 등록, 수정 시간만을 저장하여 프락시피어 내부의 시스템 부하를 줄이고, 또한 검색 비용에 가장 큰 부분을 차지하는 랑데부피어 순회 탐색 횟수를 줄일 수 있다.

제 5 장 실험 및 고찰

또한, 본 논문에서 제안하는 프락시 피어(Proxy Peer) 기반의 DHT 알고리즘 실험을 위해 다음과 같은 가정을 한다.

① 두개 이상의 랑데부 피어로 구성한다.

② 각 피어가 가지고 있는 광고 정보는 다른 피어에 없이 유일하다.

③ 쿼리로 인해 발생되는 네트워크 트래픽 외에 다른 트래픽은 고려하지 않는다.

본 논문에서는 검색 하고자 하는 광고 정보가 있는 피어를 찾아낼 때 까지 전파되는 쿼리와 이로 인해 발생되는 네트워크상의 모든 쿼리를 이용하여 검색 비용을 산출한다. 또한 광고 정보 검색에 따른 검색 비용에 영향을 주는 사항은 첫째 랑데부 피어간의 연결 형태, 둘째 랑데부 피어 검색 시 발생하는 쿼리수, 세 번째 랑데부 피어 검색 시 네트워크에 발생하는 총 쿼리수 이다.

제 1 절 시뮬레이션 환경

w CPU : Intel Pentium-Ⅳ 1.8 GHz

w Memory : 512 MB 50대

w OS : Microsoft windows XP for program running

본 논문에서는 현재 JXTA 네트워크와 동일한 환경을 제공하도록 모든 피어들의 광고 정보와 각 피어들의 위치정보가 수시로 변경 가능하도록 하였으며, 또한 랑데부 피어의 해쉬 테이블에 저장된 피어의 수를 유동적 변경 하였다. 광고 정보를 검색하기 위해 가장 처음에 생성된 랑데부 피어를 프락시 피어로 지정하고 이 프락시 피어는 동일한 피어그룹에 참여하는 모든 랑데부 피어의 정보를 가진다. 프락시 피어는 각 랑데부 피어의 이름 정보와 랑데부 피어에 소속된 피어의 수, 프락시 피어에 자신의 정보를 등록, 수정한 시간 정보를 갖는다.

제 2 절 실험 결과

실험은 DHT 프락시 피어 검색 알고리즘을 이용하여 모든 피어의 정보를 찾도록 하였으며, 랑데부 피어를 10, 30, 50개씩 가진 JXTA 환경에서 실험하였다. 이때 광고 정보를 찾는데 발생한 쿼리수와 JXTA 네트워크 상에서 광고 정보를 검색할 때 발생한 모든 쿼리수를 조사하였다.

첫 번째 결과는 피어의 광고 정보를 찾고자 하는데 소요된 시간 향상이며, 두 번째 결과는 광고 정보를 검색할 때 발생하는 네트워크 상의 모든 쿼리 정보를 수집한 결과로서, 네트워크 상의 발생하는 트래픽을 나타낸다.

그림 5-1과 같이 검색비용은 전체 랑데부 피어 중에 검색 시 쿼리수를 나타낸 것으로 1에 근접할수록 검색 비용이 많아지고, 0에 근접할수록 검색 비용은 적어진다. 또한 JXTA 네트워크의 규모가 커질수록 랑데부 피어간의 발생하는 쿼리의 수가 많아지는데, 프락시 피어기반의 DHT 알고리즘을 이용함으로써 쿼리수를 줄여 빠른 검색 시간을 제공한다.

그림 5-2의 검색 비용은 광고 정보를 검색 시 네트워크 상에 발생하는 모든 쿼리수를 나타낸 것으로 1에 근접 할수록 검색 비용이 많아지고, 0에 근접 할수 록 검색비용은 적어져 전체 네트워크 상에 트래픽이 줄어들며. 그림 5-2과 같이 JXTA 네트워크의 규모가 커질수록 네트워크 상에 발생하는 트래픽이 기존의 방식에 비해 적어지는 것을 알 수 있다. 또한 작은 규모의 JXTA 네트워크 규모에서는 성능의 차이가 작으나 , 규모가 커질수록 검색 시 발생하는 쿼리수의 차이가 크게 되어 DHT Proxy Walker는 대규모의 JXTA 네트워크에서 검색 비용을 줄일 수 있다는 것을 알 수 있다.

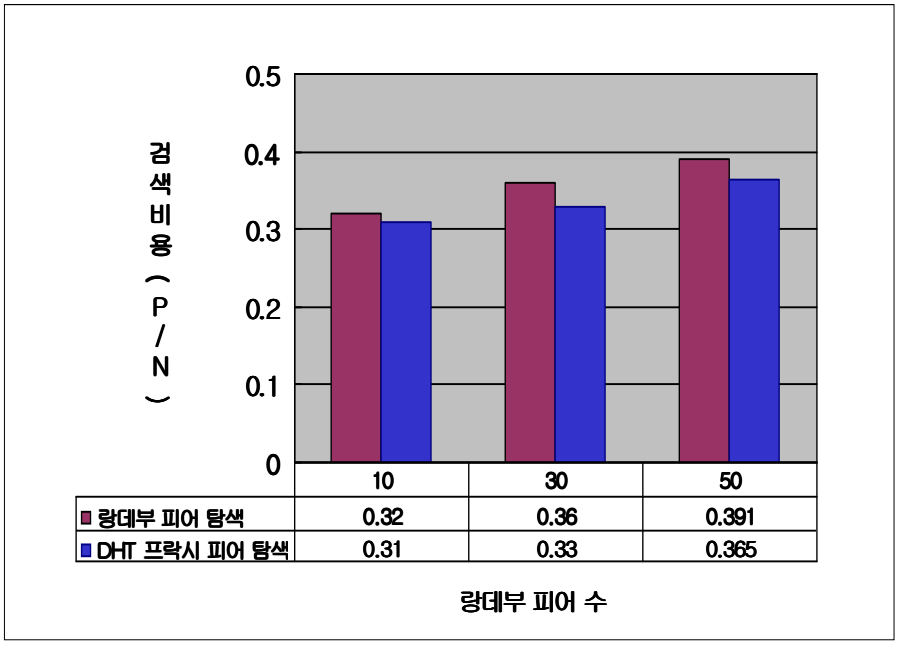

그림 5-1. 광고정보 검색 비용

Figure 5-1. Advertisement information search expense

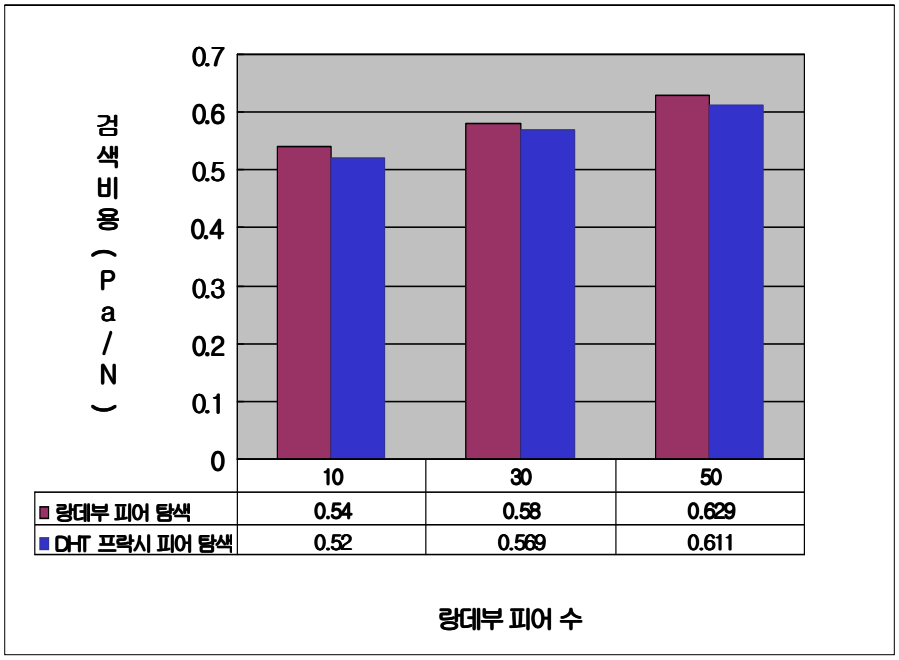

그림 5-2. 광고정보 검색 총비용

Figure 5-2. Advertisement information search total expense

제 6 장 결 론

P2P 네트워크 환경이란 클라이언트 상호간 분산 및 협력이라는 새로운 개념의 네트워크라 할 수 있는데 분산 개념은 효율성의 극대화를 목적으로 P2P 네트워크가 분산 컴퓨팅을 통해 하드웨어 자원을 공유하는 것을 의미하며 협력 개념은 클라이언트 간에 상호 협력해 공유한 자원을 검색하는 것을 의미한다. P2P와 같은 분산 컴퓨팅 환경에서는 효율적인 이용을 위해 분산된 자원의 검색이 무엇보다 중요 하지만 분산된 자원을 검색할 때 기존의 자원정보 탐색 방법은 랑데부 피어간의 연결 상태에 대한 정보만을 이용하여 검색하게 되므로 이로 인해 자원정보의 검색 비용이 높아 많은 시간적인 지연과 트래픽이 다수 발생 하였다. 이런 문제점을 해결하기 위해서 각 랑데부 피어 검색 시 우선 순위를 두어 자원정보 탐색 시 효율적인 검색 비용을 제공하고자 한다.

본 논문에서는 플랫폼에 독립적이고, 네트워크 디바이스에 독립적인 환경을 지원하는 JXTA기반의 P2P 시스템을 설계하고, JXTA 네트워크 환경에서 분산된 자원을 효율적으로 검색하는 DHT 프락시피어 알고리즘을 제안하였다. DHT(Distributed Hash Table)는 분산 해쉬 테이블을 이용하는 기법으로써, 모든 피어들은 공유하고 있는 자원들에 대한 메타 데이터를 해쉬 값에 의해 지정된 피어에게 전송하며, 메타데이터를 수신한 피어들은 다른 피어로부터 검색 요청이 들어올 경우 자원의 위치를 알려주는 방식으로 동작한다.

프락시 피어는 랑데부 피어의 자원정보 저장과 랑데부 피어 검색에 따른 프락시 시스템 부하를 최소한으로 줄이기 위해 접근에 필요한 랑데부 피어의 이름과 랑데부 피어가 소유한 피어 수, 랑데부 피어 자원정보 등록, 수정 시간만을 저장하여 프락시피어 내부의 시스템 부하를 줄이고, 검색 비용에 가장 큰 부분을 차지하는 랑데부 피어 순회 탐색 횟수를 줄일 수 있다.

본 논문에서 제안한 프락시 피어 탐색 알고리즘을 이용하여 랑데부 피어들의 자원정보를 관리하고 다른 랑데부 피어로 포워딩을 유도 하는데 확률적으로 높은 랑데부 피어들을 우선적으로 검색하게 함으로서 검색 비용을 개선하였고, 랑데부 피어 정보의 신뢰성을 기반으로 하여 효율적인 검색 결과를 얻을 수 있었다.

향후 연구방향으로 본 논문에서 제안한 탐색 알고리즘은 안정성을 제공한다는 전제 조건하에서 효율적인 검색 비용을 나타낼 수 있었으므로 P2P 환경에서 안정성을 제공하기 위한 프락시 피어의 장애에 따른 좀 더 능동적이고 정확한 복구 기법에 대한 연구가 요구된다.

[참 고 문 헌]

〔1〕 Bernard Traversat and Mohamed Abdelaziz and Eric Pouyoul, "Project JXTA: A Loosely-Consistent DHT Rendezvous Walker", Sun Microsystems, 2004.

〔2〕 Bernard Traversat, "Project JXTA 2.0 Super-Peer Virtual Network", Project JXTA and Sun Microsystem Inc, 2003.

〔3〕 Changtao Qu, "Interacting the Edutella/JXTA Peer-to-Peer Network with Web Services", SAINT'04, 2004.

〔4〕 IBM developerWorks, "JXTA2: A high-performance, massively scalable P2P network", http://www-903.ibm.com/developerworkers/kr/ , 2003.

〔5〕 IBM developerWorkers, "Mobile P2P messaging Part 2: Develop mobile extensions to generic P2P networks", http://www-903.ibm.com/developerworkers /kr/

〔6〕 Jean-Marc Seigneur and Gregory Biegel and Christian Damsgaard, "P2P with JXTA-Java pipes".

〔7〕 JXTA Korea, http://www.jxtakorea.org/

〔8〕 PeerCom, http://www.peercom.com/

〔9〕 이석희, 그룹관리와 신뢰성을 위한 Hybrid P2P 시스템 설계 및 구현, 청주대학교, 2002.

〔10〕 주형렬, P2P 기반 자료공유시스템의 설계 및 구현, 중앙대학교 대학원, 2002.

〔11〕 전형성, 조용중, 박천구, 세상을 바꾸는 힘의 중심 P2P, 프로그램세계, 2002.

〔12〕 김봉한, 이재광, P2P 어플리케이션 보안을 위한 JXTA 분석, 한국정보학회, 제 13권 제 3호, 2003.

〔13〕 최정희, "대규모 네트워크에서의 P2P 프로토콜 설계", 충북 대학교 석사학위 논문, 2002.

〔14〕 A Peer to Peer(P2P) Architecture for Dynamic Workflow Management, Georgios John Fakasa,1, Bill Karakostas, Feb, 2003.

〔15〕 Oram, A., Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly, 2001.

〔16〕 Andy Oram, "Peer-to-Peer-Harnessing the Power of Disruptive Technologies", O'Reilly, Sep., 2001.

〔17〕 Nelson Minar, "Distributed Systems Topologies: Part 2", http://www.openp2.com/pub/a/p2p/2002/01/p2p_topologies_pt2.html, Jan. 2002.

〔18〕 Li, S., "Design P2P Systems that Extend Beyond Traditional Network Boundaries",
http://www-106.ibm.com/deveoperworks/java/library/j-p2pint3/ , Apr, 2002.

〔19〕 Saroiu, S, and P. Gummadi, "A Measurement Study of Peer-to-Peer File Sharing Systems", Technical Report, Jan., 2002.

〔20〕 Kan. G., "Gnutella", in Andy Oram(ed), Peer-to-Peer: Harnessing the Power of Disruptive Technologies, 2001.

〔21〕 Ganchev. Philip, "Clusters-a Proposed Topology for a Peer-to-Peer Network", University of Pittsburgh, Oct 2000.

〔22〕 M. Handley, C. Perkins and E. Whelan, "SAP: Session Announcement Protocol", IETF RFC 2974, 2000.

〔23〕 M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, "Topology-Aware Routing in Structured Peer-to-Peer Overlay Network", in FuDico 2002 : International Workshop on Future Directions in Distributed Computing Italy, Jun 2002.

〔24〕 N. Blundell and L. Mathy, "An Overview of Gnutella Optimisation Techniques", PGNet, 2002.

〔25〕 S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer File Sharing Technologies", Athens University of Economics and Business, 2002.

〔26〕 Clarke, O. Sandberg, and B. Wiley, "Freenet: A Distributed Anonymous Information Storage and Retrieval System. Designing Privacy Enhancing Technologies", in Proc. International Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009, 2001.

〔27〕 Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", in Proc. ACM SIGCOMM, 2001.

〔28〕 S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable content-Addressable Network", in Proc. ACM SIGCOMM, 2001.

〔29〕 R. Schollmeier. "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications", IEEE Conference P2P 2001, Linkping Sweden, Aug 2001.

〔30〕 Marcel Waldvogel and Roberto Rinaldi, "Efficient Topology-Aware Overlay Network", Hot Topics in Networks I (HotNets-I), Oct 2002.

〔31〕 The HotNets-I Proceedings will also Appear in Computer Communication Review, Volume 33, Number 1, Jan 2003.