

# 모바일 환경에서 슈퍼 피어를 이용한 피어-투-피어 시스템 (Peer-to-Peer System using Super Peers for Mobile Environments)

한 정 석<sup>†</sup>      송 진 우<sup>†</sup>  
(JungSuk Han)    (JinWoo Song)

이 광 조<sup>†</sup>      양 성 봉<sup>\*\*</sup>  
(KwangJo Lee)    (SungBong Yang)

**요 약** 모바일 기기 사용이 급증함에 따라 모바일환경에서 이루어지는 Peer-to-Peer(P2P)방식에 관한 연구가 활발히 진행되고 있다. 본 논문에서는 기존의 모바일환경에서의 P2P방식이 지닌 peer들 사이의 broadcasting 방식인 'flooding'의 문제점을 보완하고, 새로운 routing table을 구축하기 위해 peer들을 2개의 계층으로 구분하였다. 즉, peer들을 super peer들과 각 super peer에 의해 관리되는 sub-peer들로 구분하였다. 파일의 탐색과 전송은 super peer들이 관리하므로, 기존의 불필요한 multi-broadcasting 방식을 피할 수 있다. 본 논문에서는 super peer의 개념을 이용한 두 가지의 모바일 P2P 시스템을 제안한다. 첫 번째 시스템은 영역을 일정한 크기로 나누어 각각의 구역마다 super peer를 가지도록 한다. 두 번째 시스템은 자신에게 연결된 이웃 peer의 수를 계산하여 가장 많은 이웃 peer를 가진 peer부터 순서대로 super peer가 되도록 한다. 본 논

- 이 논문은 서울시 산학 협력사업(10854)의 지원을 받았음
- 이 논문은 2007 한국컴퓨터종합학술대회에서 '모바일 환경에서 슈퍼 피어를 이용한 피어-투-피어 시스템'의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 학생회원 : 연세대학교 컴퓨터과학과  
leohan@yonsei.ac.kr  
fantaros@cs.yonsei.ac.kr  
kjlee5435@gmail.com

<sup>\*\*</sup> 종신회원 : 연세대학교 컴퓨터과학과 교수  
yang@cs.yonsei.ac.kr

논문접수 : 2008년 1월 3일

심사완료 : 2008년 2월 20일

Copyright©2008 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제3호(2008.5)

문에서 제안된 P2P 시스템들은 이중 계층구조로 peer들을 구분하여, 될 수 있는 대로 상층의 super peer들 사이의 메시지 교환이 이루어지게 함으로써 broadcasting을 피한다. 본 논문에서 제안한 시스템들의 성능 향상을 확인하기 위하여, 메시지의 수를 측정하는 실험을 하였으며, 그 결과 본 논문에서 제안한 시스템들이 기존의 시스템보다 평균 1.2~1.6배 향상되었음을 보였다.

키워드 : Super Peer, Peer-to-Peer, 모바일 시스템

**Abstract** As the number of mobile device users increases, many researches on peer-to-peer (P2P) systems in mobile environments have been carried out. In this paper, we propose a couple of double-layered P2P file sharing systems to overcome the 'flooding' problem in previous mobile P2P systems. We classify peers into two groups, super peers and sub-peers to establish new routing tables. A super peer manages its sub-peers in the systems. The first proposed system partitions the service area into small cells, each of which is a square. Each super peer is located near the center of the square. The second system selects super peers which have the largest number of adjacent peers. As file transmission and file searches are managed mainly by super peers, unnecessary multi-broadcasting could be avoided. The experimental results show that the proposed systems outperform a typical file sharing system in terms of the amount of message traffic with about 1.2~1.6 times improvement on the average.

Key words : Super Peer, Peer-to-Peer, Mobile System

## 1. 서론

최근 많은 사람들이 모바일 기기인 핸드폰, PDA를 사용하게 되면서, 모바일환경에서의 Peer-to-Peer(P2P)가 많은 각광을 받고 있다. 모바일 기기는 이동성을 가지는 객체이고, 각각의 모바일 기기는 통신범위라는 제약조건이 있기 때문에 기존의 유선환경의 P2P방식을 적용할 수 없다. 따라서 모바일환경의 P2P 시스템은 기존의 유선 P2P 시스템과는 다른 방식으로 개발이 되어왔다. 대표적인 모바일환경의 P2P 시스템으로는 ORION (Optimized Routing Independent Overlay Network) [1]이 있다.

본 논문에서는 기존의 모바일환경에서의 P2P방식이 지닌 peer들 사이의 broadcasting 방식인 'flooding'의 문제점을 보완하고, 새로운 routing table을 구축하기 위해 peer들을 2개의 계층으로 구분하였다. 즉, peer들을 super peer들과 각 super peer에 의해 관리되는 sub-peer들로 구분하였다. 파일의 탐색과 전송은 super peer들이 관리하므로, 기존의 불필요한 multi-broadcasting 방식을 피할 수 있다. 본 논문에서는 super

node의 개념을 이용한 두 가지의 모바일 P2P 시스템을 제안한다. 첫 번째 시스템은 영역을 일정한 크기로 나누어 각각의 구역마다 super peer를 가지도록 한다. 두 번째 시스템은 자신에게 연결된 이웃 peer의 수를 계산하여 가장 많은 이웃 peer를 가진 peer부터 순서대로 super peer가 되도록 한다. 본 논문에서 제안된 P2P 시스템들은 이중 계층구조로 peer들을 구분하여, 될 수 있는 대로 상층의 super peer들 사이의 메시지 교환이 이루어지게 함으로써 broadcasting을 피한다. 본 논문의 구성다음과 같다. 2장에서 관련연구를 조사하고, 3장에서 제안하는 P2P시스템을 제시하며, 4장에서는 실험결과를 보여주고, 5장에서 결론을 맺는다.

## 2. 관련 연구

### 2.1 모바일 P2P 시스템

기존의 유선 환경 P2P 시스템에는 냅스터(Napster)[2], 오픈냅(OpenNAP)[3], 아이알씨(IRC@find)와 같은 클라이언트-서버 방식과, 뉴텔라(Gnutella)[4]나 프리넷(Freenet)과 같이 모든 peer들이 네트워크에서 동등하게 참여하는 방식들이 있다. 그러나 이러한 유선환경 P2P 시스템은 무선 환경에 바로 적용을 시킬 수가 없다. 왜냐하면 모바일 객체는 이동성과 제한된 통신거리를 가지기 때문이다. ORION은 이러한 어려움을 극복한 대표적인 모바일 P2P 시스템이다.

### 2.2 ORION P2P 시스템

모바일환경에서의 P2P로는 ORION이 대표적이다. ORION은 라우팅을 위한 테이블과, 파일관리를 위한 파일 라우팅 테이블을 따로 관리하여 효율적인 파일 공유를 꾀하였다. 라우팅 방법은 AODV(Ad-hoc On-demand Distance Vector)[5] 라우팅 테이블과 유사하다.

ORION은 네트워크 링크가 구축이 되면, flooding 방식으로 사용자가 원하는 파일을 찾는다. ORION이 그림 1처럼 라우팅 테이블이 구성했다고 가정하자. 사용자 A가 파일 1,2,3,4를 찾기를 원한다면, 자신한테 연결된 모든 peer들에게 multi-broadcasting을 한다. 이 경우, 사용자 B는 사용자 A가 보낸 query를 받고, B는 같은 query를 자신에게 연결된 사용자 C, D한테 보낸다. 사용자 B는 A가 보낸 query를 보고 자신의 local file table을 살펴본다. 그리고 B는 A가 원하는 파일 1,2,3,4 중 1,2를 가지고 있다고 A에게 Response 메시지를 보낸다. C와 D의 경우도 자신의 local file table을 각각 참조하며, B로부터 온 A의 메시지를 본다. 자신의 file table에 있는 파일 1,2,3(C의 경우), 파일 2,3,4(D의 경우)를 Response 메시지로 각각 B로 보낸다. 이러한 flooding방식은 자신에게 연결된 모든 peer들에게 같은 query를 보내주며, peer들의 연결이 많이 있을수록 peer

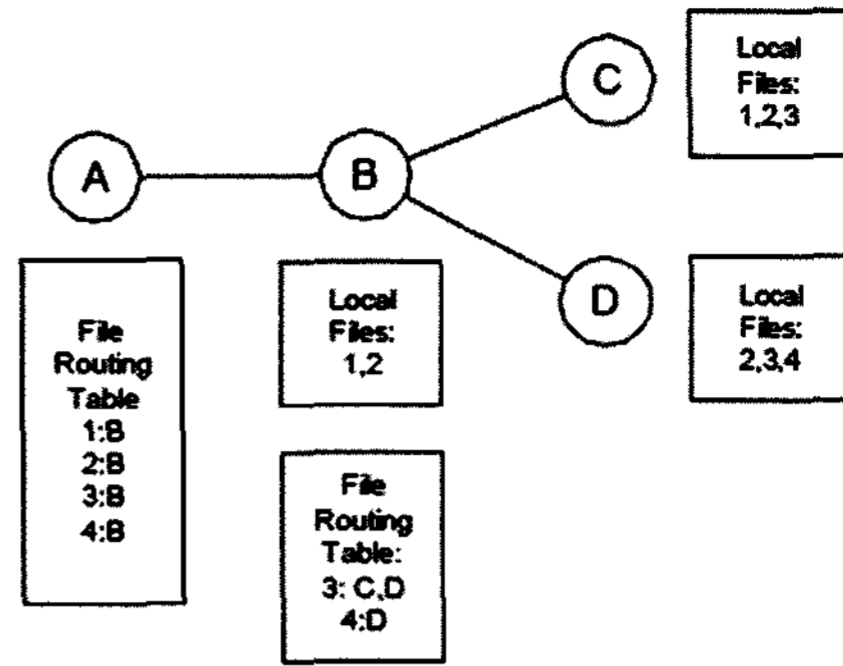


그림 1 ORION의 flooding 방식

들 사이의 query 메시지의 전송 횟수가 기하급수적으로 늘어나게 된다.

본 논문에서 제안하는 P2P 시스템들은 flooding 방식이 아닌 주로 특정 peer들 사이의 메시지 교환을 하게 함으로써 파일 탐색 시 메시지의 수를 줄이도록 설계되었다.

## 3. 제안하는 P2P 시스템

기존의 모바일 P2P 시스템에서는 모든 peer들이 동일한 계층에서 자신 주위에 연결된 모든 이웃 peer들에게 multi-broadcasting 방식을 사용한다. 본 논문에서는 peer들을 두 개의 계층으로 나눈다. 즉, 상위계층에는 하위계층의 peer들을 관리하는 super peer들이 있고, 하위계층에는 super peer에 의해 관리되는 sub-peer들이 있다. super peer는 자신이 관리하는 sub-peer들의 정보(id, 주소, 파일목록)를 가지고 있다. 한 sub-peer가 파일을 찾고자 한다면, 먼저 자신의 super peer에게 원하는 파일을 요청한다. 만약, 해당 super peer가 그 파일에 대한 정보(entry)를 가지고 있다면, super peer는 sub-peer에게 그 파일을 가지고 있는 peer의 주소를 알려준다. 반면에 그 파일에 관한 entry가 없으면, super peer는 다른 super peer들에게 해당 파일을 찾아달라고 요청을 한다. 요청을 받은 다른 super peer들은 각각 자신의 entry에서 찾아보고, 있으면 Response 메시지를 보낸다.

이러한 방법을 이용하면 ORION처럼 자신에게 연결된 모든 peer들에게 multi-broadcasting하는 방식을 피할 수 있어 탐색하는데 필요한 메시지의 수를 줄일 수 있다.

### 3.1 지역 균등분할 시스템

주어진 서비스 지역을 일정 크기로 균등 분할하여 각 지역마다 한 개의 super peer를 할당한다. 분할된 지역의 크기는 모바일기기의 통신범위보다 같거나 작게 잡는다. 그 이유는 각 지역 안의 모든 peer들이 한 hop에

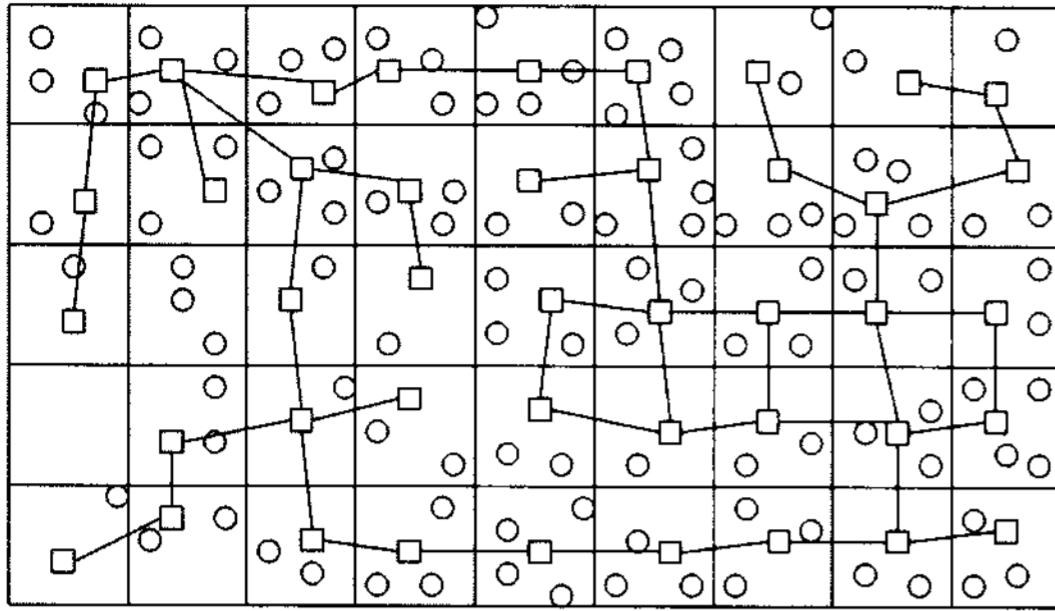


그림 2 지역 균등분할 시스템에서 super peer와 sub-peer의 구성 예

서로 연결되게 하기 위함이다. 각 지역에 m개의 peer들이 있다면, 각 지역의 super peer는 m개의 peer정보(자신의 것도 포함)를 가진다. 반면, sub-peer는 자신의 super peer의 정보만을 기록해 둔다.

그림 2는 지역 균등분할 시스템에서 어떠한 peer들이 super peer가 되고, sub-peer가 되는지를 보여주고 있다. 그림 2안에 작은 사각형은 해당 지역의 super peer를 나타내며, 동그라미가 sub-peer들이다. 그림에서 확인할 수 있듯이 각 지역의 중심에 가장 근접한 peer가 super peer로 선정된다. 이렇게 super peer를 선정하는 이유는 인접 지역의 super peer들끼리 통신을 할 수 있도록 하기 위해서이다. 한 지역의 중심에서 그 인접지역의 중심까지의 거리는 분할된 지역의 크기와 같고, 이는 모바일 기기의 통신거리이내로 되어있다. 각 지역의 sub-peer들은 주변의 sub-peer들과 연결하지 않고, 그 지역의 super peer로만 연결된다. 그림 2에서처럼 일부의 super peer들 사이에는 통신범위 밖에 위치하여 서로 연결이 없다.

한 사용자 A가 어떤 파일을 찾고자 한다면, A는 먼저 자신이 속한 지역의 super peer에게 query를 보낸다. 그 super peer는 자신의 entry안에서 해당 파일을 탐색한다. 만일 탐색을 성공하면 동일 지역 내의 peer가 그 파일을 가지고 있으므로, super peer는 A에게 파일을 가지고 있는 peer의 주소를 알려줘 A가 직접 연결할 수 있게 한다. 만일 실패하면, super peer는 연결된 인접 지역의 super peer들에게 파일 탐색을 요청한다. 인접 지역에 있는 super peer들은 각각 자신들의 entry에서 파일을 탐색해보고 만약에 찾았다면 *Response* 메시지를 시작지역의 super peer한테 보낸다. 인접지역 super peer들도 탐색에 실패하면 새 인접지역들의 super peer들에게 요청한다.

그림 3은 파일 탐색을 위해 주변지역을 탐색하는 과정의 한 예를 보여준다. 별 모양을 가진 peer가 찾고자 하는 파일을 가지고 있는 peer가 위치한 지역의 super

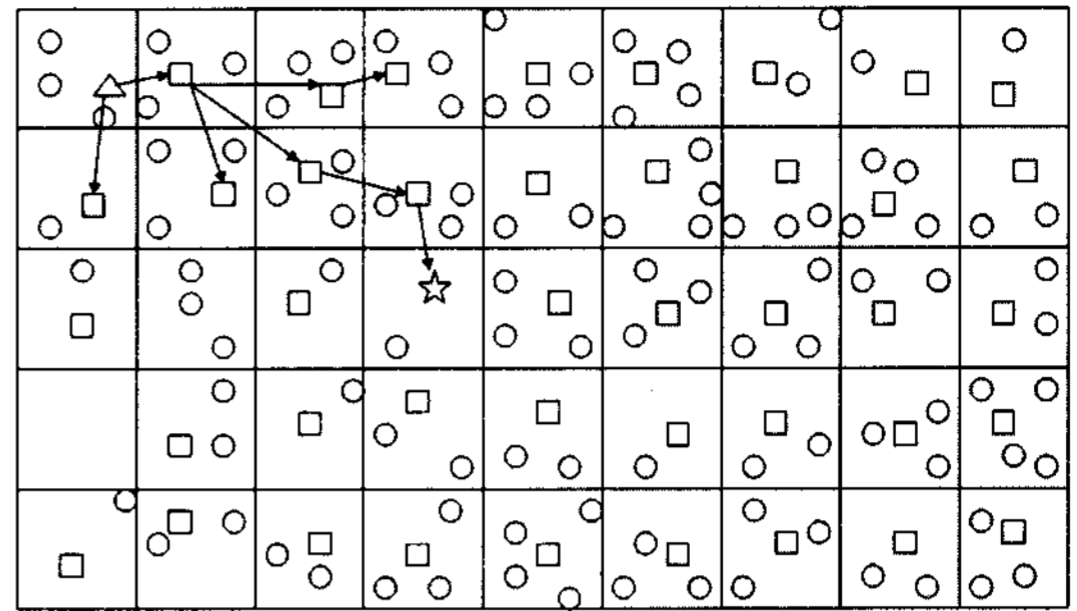


그림 3 지역 균등분할 시스템에서 파일을 찾는 과정

peer이고, 사용자가 있는 지역의 super peer가 삼각형이라고 가정한다면, 시작지역부터 목적지까지 메시지가 퍼져 나가는 과정을 보여준다.

지역 균등분할 시스템의 경우, 각각의 peer는 현재 자신이 어디 있는지 알아야한다. 또한 모든 peer들의 위치 정보를 분석하여 peer의 계층을 나눠주는 중앙 서버가 필요하다. 자신의 위치를 알기 위해서 모든 peer들은 GPS 기능이 있어야 하고, 모든 peer들은 자신의 위치정보를 획득한 후, 그 정보를 중앙 서버로 주기적으로 보내야한다. 중앙 서버는 모든 peer들로부터 받은 정보를 일정 시간마다 분석하여 peer들을 계층 구조로 연결시키고, 그 정보를 다시 각각의 peer에게 알려준다.

•Peer들의 삼입과 삭제에 의한 super peer의 재구성

모바일 사용자는 이동성으로 인해 peer들의 지역 간의 이동이 고려되어야 한다. 지역 X에 새로운 peer가 등장하면, 새 peer의 위치정보와 지역 X의 중심위치와의 거리를 계산한 후, 현재의 super peer와 중심과의 거리를 비교하여 보다 중심 쪽에 더 가깝게 위치한 peer가 지역 X의 super peer가 된다.

지역 X에 새로운 peer가 추가 된 경우, 새로운 peer는 지역 X의 인접 지역으로부터 왔을 경우가 확률이 크기 때문에, 그 인접 지역의 super peer에게 새로운 peer의 id를 알려준다. 이웃 super peer들은 자신의 entry를 검사해서 일치하는 id가 있는 경우, 이와 관련된 정보(id, 주소, 파일 목록)를 삭제한다.

3.2 Greedy super peer 시스템

일정한 지역 안에 여러 peer들이 존재한다면, 각 peer의 통신범위 안에는 다수의 peer들이 존재할 수 있다. 본 논문에서 제안하는 두 번째 시스템의 super peer 선정방법은 다음과 같다. 각 peer마다 자신에게 이웃한 peer의 수를 세어본다. 이웃 peer수가 가장 많은 peer가 첫 super peer가 된다. 이 super peer와 연결된 peer들은 첫 super peer의 sub-peer가 되며, 이 sub-peer들은 super peer선정에서 제외된다. 첫 super peer에 연결이 안 된 peer들만으로 이웃 peer수를 재계산하여 가

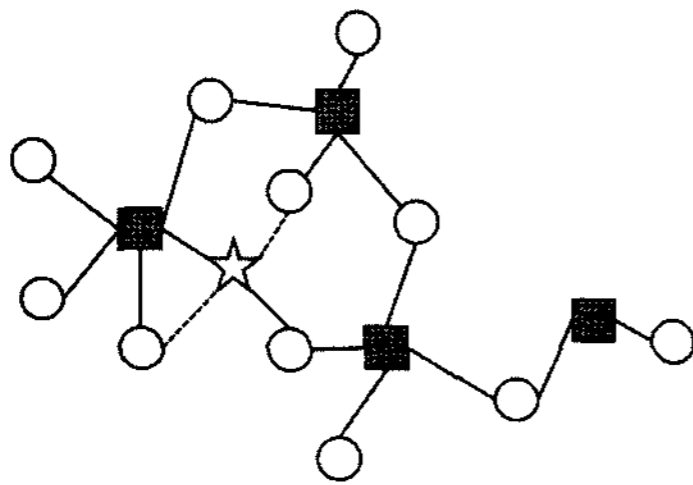


그림 4 Greedy super peer 시스템에서의 Super peer와 sub-peer의 구성 예

장 많은 이웃 peer를 가진 peer가 두 번째 super peer가 된다. 이러한 greedy 방식으로 super peer를 정한다. 이 방식은 [6]에 제시된 NP-완전문제인 Set-covering 문제를 위한 greedy approximation algorithm과 동일하다.

그림 4는 Greedy super peer 시스템에서의 super peer와 sub-peer의 구성 예를 보여주고 있다. 그림 4에서 사각형이 super peer들이고 동그라미가 sub-peer들이다. 별 모양의 peer는 자신에게 인접한 이웃의 수가 다른 super peer들처럼 4개 이상이지만 기존의 super peer와 인접한 peer이기 때문에 super peer가 될 수 없다. Greedy super peer 시스템이 완성된 후, 점선의 경로는 사용되지 않는다. 그 이유는 별 모양의 peer는 sub-peer로 되기 때문에 다른 모든 이웃 sub-peer들과 통신을 할 필요가 없기 때문이다.

Greedy super peer 시스템도 마찬가지로 super peer가 자신과 연결된 sub-peer들을 관리한다. 즉, super peer는 자신 주변의 sub-peer의 id, 주소 그리고 파일 목록을 저장하고 있다. 각 sub-peer는 자신의 super peer의 id와 주소를 저장하여 파일의 공유나 탐색을 원할 때, 자신의 super peer에게 query 메시지를 전달할 수 있다.

Super peer가 생성이 되면 주변의 super peer를 찾기 위해 자신의 sub-peer들을 활용한다. 즉, sub-peer들은 자신의 통신범위 안에 들어온 다른 peer들을 탐색하고, 그들의 super peer의 id를 물어본다. 새로운 super peer id를 발견하게 되면, 그 sub-peer를 통해 두 super peer의 경로가 생성시킨다.

그림 5는 Greedy super peer 시스템에서 파일을 찾는 과정을 보여준다. 그림에서 삼각형은 특정 파일을 원하는 사용자이고, 사각형들은 super peer, 동그라미는 sub-peer 그리고 별 모양의 peer가 해당 파일을 가지고 있다. 먼저 사용자는 파일을 찾고 싶다고 자신의 super peer에게 알린다.(과정 ①) Super peer는 자신의 entry를 살펴봐지만 탐색에 실패하여, 다른 인접한 super peer들에게 찾아 달라고 query를 보낸다.(과정 ②,③)

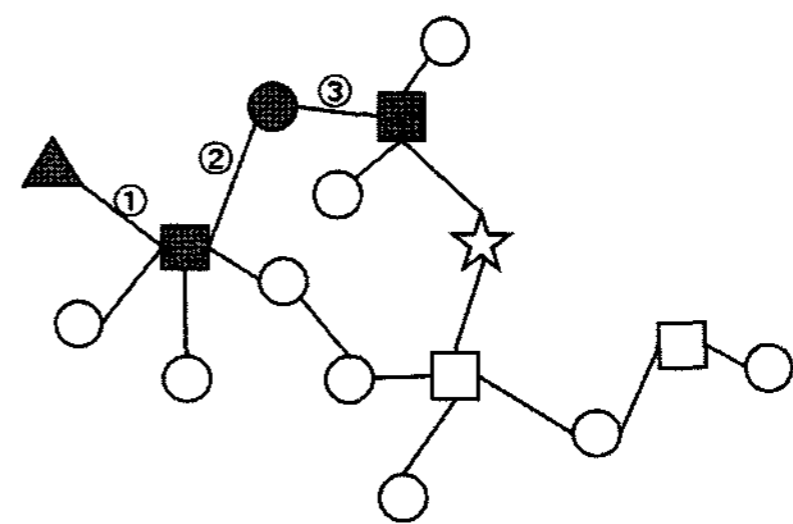


그림 5 Greedy super peer 시스템에서 파일을 찾는 과정

앞서 설명되었듯이 super peer들끼리의 경로는 sub-peer들에 의하여 생성되어 있다. 탐색 query가 새로운 super peer에 도착하면 entry를 살펴보게 되고, 이 경우 entry안에 찾고자 하는 파일이 있기 때문에 탐색에 성공한다. 탐색에 성공한 후, 시작지역의 super peer한테 Response 메시지를 즉시 보내준다.

Greedy 시스템의 경우, 각각의 peer는 자신의 통신범위 안에 있는 이웃 peer들을 계산하고, 그 결과를 중앙 서버에 보내야 된다. 모든 peer들은 같은 시각에 이웃 peer수를 계산하고 그 결과를 중앙서버로 보내야 정확한 결과를 얻을 수 있다. 중앙서버는 peer들로부터 받은 정보를 바탕으로 어느 peer가 super peer가 되고 sub-peer가 되는지 계산해야 한다. 즉, Greedy 시스템은 시간의 동기화가 필요하고, 이는 GPS를 이용하면 된다.

• Peer들의 삽입과 삭제에 의한 super peer의 재구성

모바일환경에서는 사용자의 이동이 자유롭기 때문에 자신에게 이웃한 peer의 수가 수시로 바뀔 수 있다. 중앙서버는 일정 주기마다 peer들이 보내온 정보를 바탕으로 peer들을 다시 구성해야 한다.

4. 실험

본 논문은 기존의 ORION 방식과는 달리 peer들을 두 개의 계층으로 나누었다. 그 결과 peer 간의 라우팅 테이블 구성과 통신방식이 ORION과는 다르다. 표 1의 환경 하에 각각의 시스템 성능 비교하였다.

Text file에는 노드의 id, 주소, 위치, 파일 목록이 기록되어 있어, text file을 한 줄씩 읽어 한 peer에 저장하고 1000\*1000 m<sup>2</sup> 지역에 추가했다. 이러한 text file을 3개 만들어서 본 논문에서 제시한 2개의 시스템과 ORION을 비교해보았다.

표 1 실험 환경

변수	값
Peer의 개수	40
통신범위 (m)	115
지역의 크기(m <sup>2</sup> )	1000*1000

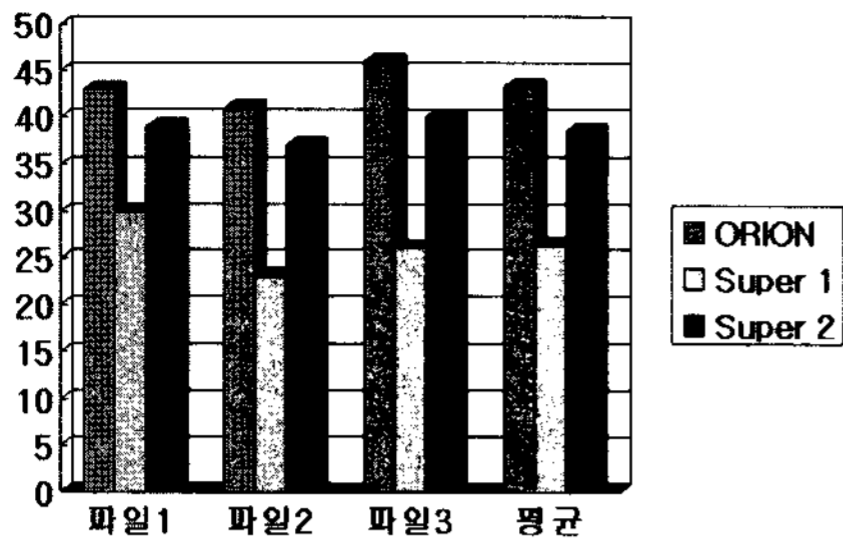


그림 6 필요한 파일 찾기 위해 필요한 메시지의 수

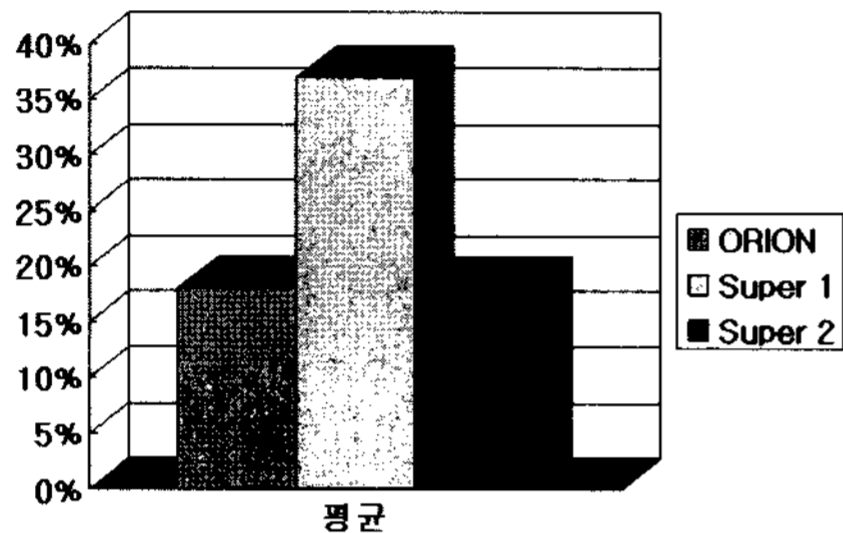


그림 7 필요한 파일을 찾는데 실패 확률

그림 6은 임의의 한 peer가 필요한 파일을 찾기 위해 사용한 메시지의 수이다. 지역 균등분할 시스템(Super 1)은 ORION보다 약 1.6배 정도 성능 향상을 보였고, Greedy super peer 시스템(Super 2)은 ORION보다 약 1.2배정도 향상됨을 보여 주었다. Super 2가 Super 1만큼 좋지 않은 이유는 구축된 peer들 사이의 routing table이 크게 개선되지 않았기 때문이다.

다음으로 시스템 간 비교하여야 할 항목은 파일 탐색 시 실패할 확률이다. 그림 7은 파일을 찾는데 실패하는 정도를 나타낸 것이다. Super1은 다른 시스템에 비해 탐색 실패 확률이 높다. 그 이유는 Super1의 경우 각 지역의 super peer가 서로 연결될 확률이 높지 않기 때문이다. 각 지역의 super peer는 중심에 가까운 peer가 되지만, super peer의 위치가 항상 지역의 중심에 있다고 보장할 수 없다. 따라서 인접지역 super peer들 사이의 연결이 원활하게 이루어지지 못하여 고립되는 peer들이 발생하기 때문에 실패하는 경우가 자주 일어난다. Super2의 경우는 ORION과 동일한 실패 확률을 보이고 있다. 그 이유는 Super2의 peer들도 ORION처럼 같은 도달 가능성을 가지기 때문이다.

5. 결론

본 논문에서 제안한 모바일 P2P 시스템은 peer들을 두 개의 계층으로 나누었고, 이를 위해 기존 routing table의 형식을 바꾸었다. 그 목적은 주로 상층의 peer들 사이에서만 통신에 이루어지게 하여 peer들 간의

multi-broadcasting 방식을 피하기 위함이었다. 지역 균등분할 시스템의 경우, 파일 탐색에 요구되는 메시지의 수는 이전 방식(ORION)에 비해 적게 소요되지만, 파일 찾기에 실패하는 경우가 종종 발생하는 단점을 보여주고 있다. 그러나 Greedy super peer 시스템은 ORION과 동일한 실패 확률을 보였으며, 성능도 ORION보다 우수함을 보여주었다. 지역 균등분할 시스템과 Greedy super peer 시스템은 어느 peer가 super peer이고 sub-peer인지 알기 위해서, 중앙서버와 GPS의 도움을 필요로 한다. 향후 중앙서버의 도움 없이 계층구조를 가진 모바일 P2P시스템에 대한 연구가 요구된다.

참고 문헌

- [1] A. Klemm, C. Lindemann, and O. Waldhorst, "A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad hoc Networks," *Proceedings on the Vehicular Technology Conference (VTC) 2003*, Vol.4, pp. 2758-2763, October 6-9, 2003.
- [2] The Napster home page, <http://www.napster.com/>
- [3] The openNap home page, <http://opennap-ng.sourceforge.net/>
- [4] Gnutella, <http://www.gnutelliums.com/>
- [5] C. Perkins, E. Royer, and S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, <http://www.ietf.org/internet-drafts/draft-ietf-manetaodv-11.txt>, IETF Internet Draft(work in progress), June 2002.
- [6] V. Chvatal, "A greedy heuristic for the set-covering problem," *Math. of Oper. Res.*, Vol.4, No.3, pp.233-235, 1979.