

효율적인 자원 탐색을 위한 울트라 피어 기반 DHT 기법

이지선* · 노선식** · 이동은*

* 청운대학교 인터넷학과 · ** 광주대학교 정보통신학과

Ultrapeer-based DHT Scheme for Efficient Resources Search

Ji-sun Lee* · Sun-Sik Roh** · Dong-eun Lee*

* Dept. of Internet, Chungwoon University · ** Dept. of info & com engineering, Gwangju Univ

E-mail : zs3710@nate.com, ssroh@gwangju.ac.kr, delee@chungwoon.ac.kr

요 약

본 논문에서는 p2p 환경에서 울트라피어 기반의 DHT 링을 구성하고 울트라피어에서는 리프피어들이 검색했던 문서들의 메타데이터를 관리함으로써 효율적인 자원검색기법을 제안한다. 울트라피어 기반의 DHT링은 피어들의 참가와 탈퇴가 빈번하게 이루어져서 발생하는 확장성의 문제를 극복하고 울트라 피어가 관리하는 이 메타데이터를 이용하여 자원을 보다 효과적으로 검색할 수 있다.

Keywords

분산 해시 테이블(DHT : Distributed, Hash Table), chord, Ultrapeer

1. 서 론

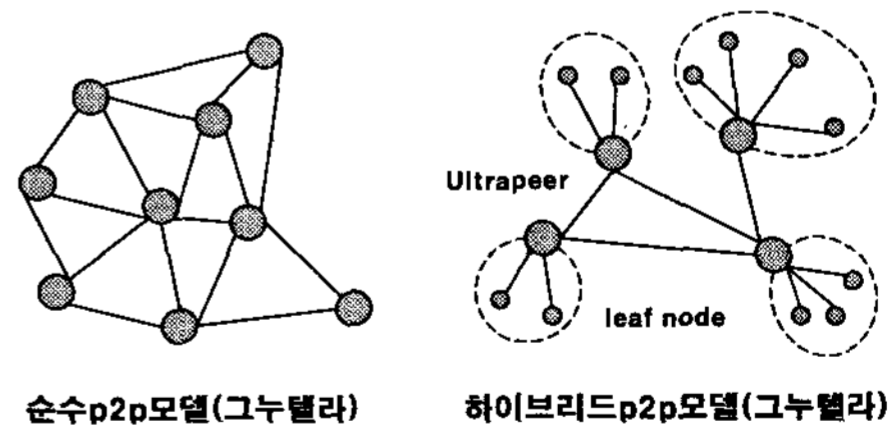
p2p 시스템은 둘 이상의 컴퓨터 사이에서 네트워크를 통해 다양한 자원을 공유하는 분산시스템이다. 분산된 네트워크 환경에서의 p2p 시스템은 서버기반 p2p 순수 p2p 그리고 하이브리드 p2p 시스템으로 구분할 수 있다. 가장 대표적인 p2p 시스템으로는 냅스터[1]와 그누텔라[2]가 있다. 두 시스템은 서버기반 p2p 시스템과 순수 p2p 시스템을 각각 대표하는 시스템이다.

냅스터는 중앙에 서버를 두고 모든 피어들의 공유파일 인덱스를 관리함으로써 피어들의 요청에 빠르게 응답할 수 있는 장점이 있다. 피어의 수가 많아질수록 중앙서버에 저장된 인덱스 수가 매우 커지게 되어 과부하 및 유지비용이 증가하게 된다. 또한 단일서버에 문제가 발생하게 되면 전체피어들에게 파일 검색 및 공유가 불가능해지게 된다.

서버기반 p2p 시스템의 문제점을 해결하기 위해 그누텔라가 제시되었다. 그누텔라에서 파일검색은 TTL의 범위 내에 있는 인접한 다른 피어에게 브로드캐스트되는 요청메시지를 전달함으로써 이루어진다. 하지만 브로드캐스트 기법을 사용하므로 네트워크 트래픽의 양이 많이 발생하게 되고 과도한 네트워크 트래픽으로 인한 확장이 용이하지 않게 된다.

이런 문제점을 해결하고자 울트라 피어기반에 그누텔라 시스템이 제시되었다.[3] 이 시스템은

울트라 피어와 리프피어로 구분하여 계층적으로 구성하며 울트라피어는 빠른 파일검색을 위해 리프피어들의 공유파일 목록을 관리한다. 따라서 기존 그누텔라보다 빠른 검색이 이루어질 수 있으며 응답메시지가 전달 받지 못하는 경우도 줄어들게 된다. 하지만 피어들의 삽입 또는 삭제가 빈번하게 일어나므로 그누텔라의 확장성문제를 해결하지 못했다.



[그림 2] 그누텔라의 종류

그누텔라의 확장성 문제에 대한으로서 DHT기반의 Chord기법이 제안되었다.[4] 이는 ip주소를 해쉬함수를 사용하여 원형 식별자 공간에 배치하고 핑거테이블을 이용하여 검색에 필요한 자료를 찾게 되는 방식을 이용한다. 또한 울트라피어기반 그누텔라 시스템에서 울트라피어만을 대상으로 DHT링을 구성하는 기법이 제시되었다.[5] 이들 DHT기법은 피어들의 공유파일에 대한 인덱스 정보를 DHT링 내에 피어들에게 분산하여 저장한

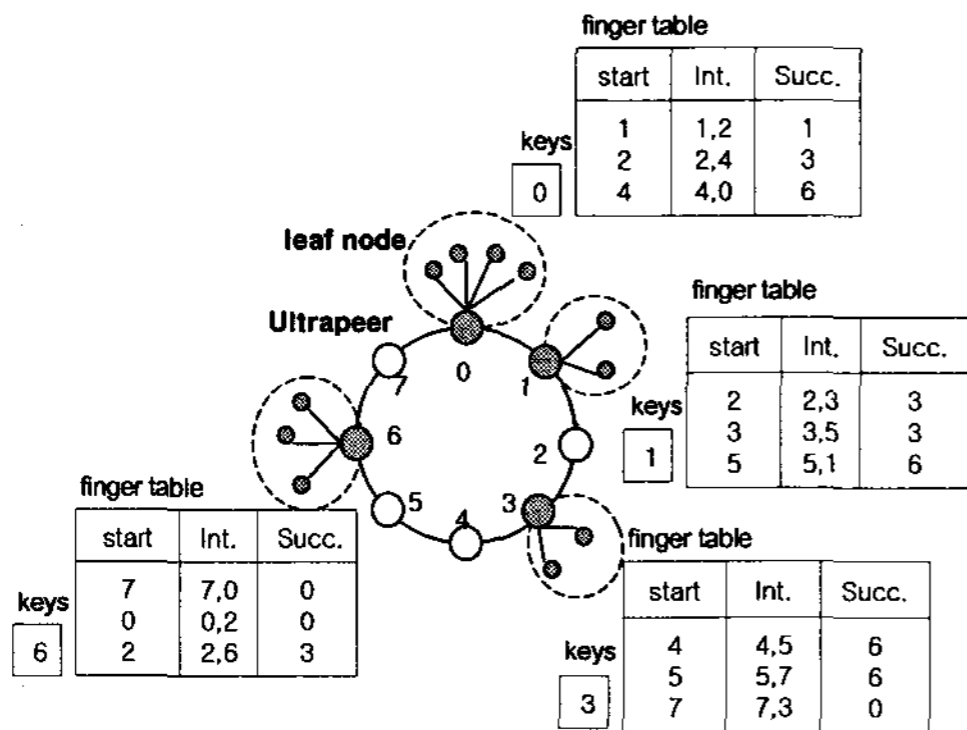
다. 그러나 Chord기법은 울트라피어기반 DHT기법에 비해 네트워크 유지비용이 크며 울트라피어기반 DHT기법은 울트라 피어가 관리하는 리프피어에 수가 커짐에 따라 울트라 피어에서 관리해야하는 인덱스 크기가 증가함에 따라 검색비용이 증가한다.

본 논문에서는 p2p 환경에서 울트라피어 기반의 DHT 링을 구성하고 울트라피어에서는 리프피어들이 검색했던 문서들의 메타데이터를 관리함으로써 효율적인 자원검색기법을 제안한다. 대부분의 피어는 쿼리 검색에만 관심을 보이고 66% 정도의 피어는 공유파일을 가지고 있지 않고 단지 검색과 다운로드에만 관심이 있다는 연구 결과가 있다. 제안한 기법은 메타데이터 저장을 위한 추가적인 오버헤드 비용으로 검색의 효율성을 개선한다. 울트라피어기반의 DHT링은 피어들의 참가와 탈퇴가 빈번하게 이루어져서 발생하는 확장성의 문제를 극복하고, 검색속도 향상을 위해 울트라 피어는 메타데이터를 관리하여 보다 효과적으로 자원 탐색을 수행한다. 피어들의 검색요청에 대해 엔트리 데이터 테이블이라고 불리는 메타데이터를 먼저 검색하여 검색의 효율성을 높이는 기법을 제시한다.

II. 시스템 구조

2.1 울트라피어기반 DHT 시스템 구조

본 논문에서 제안하는 시스템은 그림2와 같이 그누텔라 시스템의 울트라 피어를 DHT링으로 구성하고 울트라피어는 리프피어의 파일의 목록을 관리한다.



[그림 2] 확장 DHT 시스템 구조

검색의 효율성을 높이기 위해 울트라피어는 핑거 테이블, 엔트리 데이터 테이블, 인덱스 테이블을 관리한다. 핑거 테이블은 울트라피어로 구성된 DHT링에서 울트라피어의 successor 노드를 가리키는 테이블이다. 엔트리 데이터 테이블은 피어들이 검색했던 문서들의 메타데이터이다. 그리고 인

덱스 테이블은 파일을 해쉬하여 얻어진 해쉬값과 일치하는 울트라피어에서 공유자원과 일치하는 (파일명,url)을 관리한다. 리프피어는 울트라피어에게 검색요청을 한다. 후보피어는 울트라피어의 역할을 대신하는 피어이며 울트라피어가 탈퇴 시 울트라피어가 될 리프피어들 중 가장 우수한 피어이다.

2.2 울트라 피어의 DHT링 구성

울트라 피어는 식(1)과 같이 노드의 ip주소를 해쉬한 값을 이용하여 노드 식별자 id를 구성하고 각 울트라 피어들로만 구성된 DHT링에 배치한다.

$$\text{노드 식별자 id} = \text{HASH} [\text{IP}] \quad \text{식(1)}$$

리프피어는 지역별로 가장 근접한 울트라 피어 하위에 위치시키고 울트라 피어를 이용하여 파일을 검색하게 된다.

◎ 울트라 피어의 선정

울트라피어는 지역별로 선정되며, 10~100개의 리프피어를 수용할 수 있어야 한다. 리프피어의 파일검색요청을 수용해야하기 때문에 방화벽이 없어야하고 충분한 대역폭을 가지고 있어야 한다. 또한 평균체류시간이 길어야 한다.

이를 위하여 본 논문에서는 이전에 해당 노드에 연결시간인 Con_old 와 최근에 노드가 접속한 연결시간인 Con_new 그리고 조절 값인 γ 을 두고 식 (2)에서 보는 바와 같이 Con_time을 계산한다.

$$\text{Con_time} = (1-\gamma)\text{Con_old} + (\gamma)\text{Con_new} \quad \text{식 (2)}$$

$$\text{Ultrapeer} = \text{Max}(\text{Con_time}) \quad \text{식 (3)}$$

$$0 \leq \gamma \leq 1, \text{ Correction factor}$$

Con_time을 설정할 때 γ 는 0에 가까울수록 가장 최근의 연결시간의 기여도가 낮아지고 과거의 연결시간의 기여도가 높아진다. 그 γ 의 기준이 되는 값은 휴리스틱하게 최적의 γ 의 값을 설정할 수도 있다.

◎ 후보피어의 선정

울트라 피어의 탈퇴 시 울트라피어의 역할을 대신할 후보피어의 선정은 울트라피어 다음으로 접속시간이 긴 피어를 식(4)에 의해 선정하며 후보피어는 울트라피어의 정보를 같이 공유한다.

$$\text{후보 Ultrapeer} = \text{Next}(\text{Max}(\text{Con_time})) \quad \text{식 (4)}$$

리프피어 중 울트라 피어와 유사한 기능을 가지고 울트라피어 다음으로 한번 접속에 오래 머무는 피어가 후보피어가 된다.

III. 울트라피어의 동작

3.1 울트라 피어 테이블 구조

울트라 피어는 리프피어들의 효율적인 검색을 하기 위해 핑거테이블과 엔트리데이터 그리고 인덱스 테이블을 가지고 검색을 수행한다.

◎핑거테이블

시작값(start) - 핑거테이블이 자신의 울트라피어에서 검색을 못했을 경우 처음 검사를 시작하는 값이다.

범위(interval) - DHT링에서는 검색 속도를 빠르게 하기 위해 핑거테이블을 구성하고 그의 지수승 크기의 범위를 설정하여 검색한다. 예를 들어 울트라 피어수가 8개이면 검색할 울트라피어의 범위를 2,4,8개로 증가시켜가며 범위를 정한다. 즉, 핑거테이블의 i번째 항목은 2ⁱ승의 범위 값을 가진다.

successor - 울트라피어링에서 공유파일을 검색할 때 처음 검색해야하는 울트라피어이다.

◎엔트리 데이터 테이블

리프노드와 가장 근접한 지역의 울트라 피어는 리프피어들이 검색했던 공유파일을 엔트리 데이터 테이블 형태로 유지한다. 리프피어들이 울트라 피어에게 검색요청을 했을 때 먼저 엔트리 데이터 테이블에서 검색을 수행한다. 엔트리 데이터 테이블은 지역 내 피어들이 요청했던 공유파일 위치정보에 대한 캐쉬역할을 수행한다. 지역 내 리프노드에 검색 질의에 대해 해당 파일에 대한 인덱스 테이블을 가지고 있는 울트라 피어가 지역 내 울트라 피어에게 응답할 때 엔트리 데이터 테이블에 파일의 위치정보를 파일명을 기준으로 정렬하여 기록한다.

엔트리 데이터 테이블은 파일명을 기준으로 바이너리 검색을 수행하여 리프피어가 요청한 파일을 검색한다. 검색 시 해당 공유파일에 대한 위치 정보는 가장 최근에 들어온 것을 먼저 검색할 수 있도록 LIFO(Last In First Out) 방식의 리스트를 이용한다.

[그림 3] 엔트리 데이터 테이블

파일명	IP address
비틀즈	210.95.165.62 210.95.165.63
다 위	210.95.165.97 210.95.165.61 210.95.165.96 210.95.165.60
:	:
비	172.31.53.100

◎인덱스 테이블

피어들은 자신이 가지고 있는 공유파일을 어떤 울트라피어에 저장할지 정해야 한다. 울트라피어의 인덱스 테이블에 저장되는 엔트리 데이터정보

는 (파일명, url)쌍 형태로 저장된다.

· 파일 식별자 id = HASH [파일] 식(5)

파일 식별자 id값은 식(5)와 같이 파일을 해쉬하여 얻어진 값을 이용하게 되는데 표 1에서 나오는 예로 리프피어가 가지고 있는 파일명인 배토벤을 공유할 때 파일 식별자 id를 구하기 위해 해쉬를 하고 그 식별자 id의 값이 1이 나왔다면 식별자 id값인 1을 1의 킷값을 가지고 있는 울트라피어는 이 메타데이터를 관리하게 된다.

[표 1] 인덱스 테이블

파일명	URL
배토벤	http://210.111.100.1/share
사 계	http://172.100.10.1/share

한편 리프피어가 조인하는 경우 식(2)에 의해 Con_time을 계산하고 지역 내 Ultrapeer에게 전송함으로써 지역 Ultrapeer 탈퇴 시 울트라 피어 선정에 활용하도록 한다.

탈퇴 할 경우 해당 리프피어의 파일 정보를 삭제하게 된다. 업데이트 시 존재하지 않는 리프노드의 인덱스정보를 삭제하기 때문에 울트라피어에게 알리지 않고 탈퇴한다.

3.2 울트라 피어와 후보피어의 탈퇴

◎울트라 피어의 탈퇴

울트라 피어는 Max(Con_time)값을 가진 피어이며 울트라피어가 탈퇴 할 때 울트라 피어의 정보를 공유하고 있는 후보피어가 울트라 피어가 된다. 리프피어 중 최근의 연결시간이 다음으로 가장 큰 피어가 후보피어가 되며 울트라 피어가 탈퇴 했을 시 이 후보피어(울트라 피어와 동일한 정보를 관리한다.)가 울트라피어가 된다.

◎ 후보피어의 탈퇴

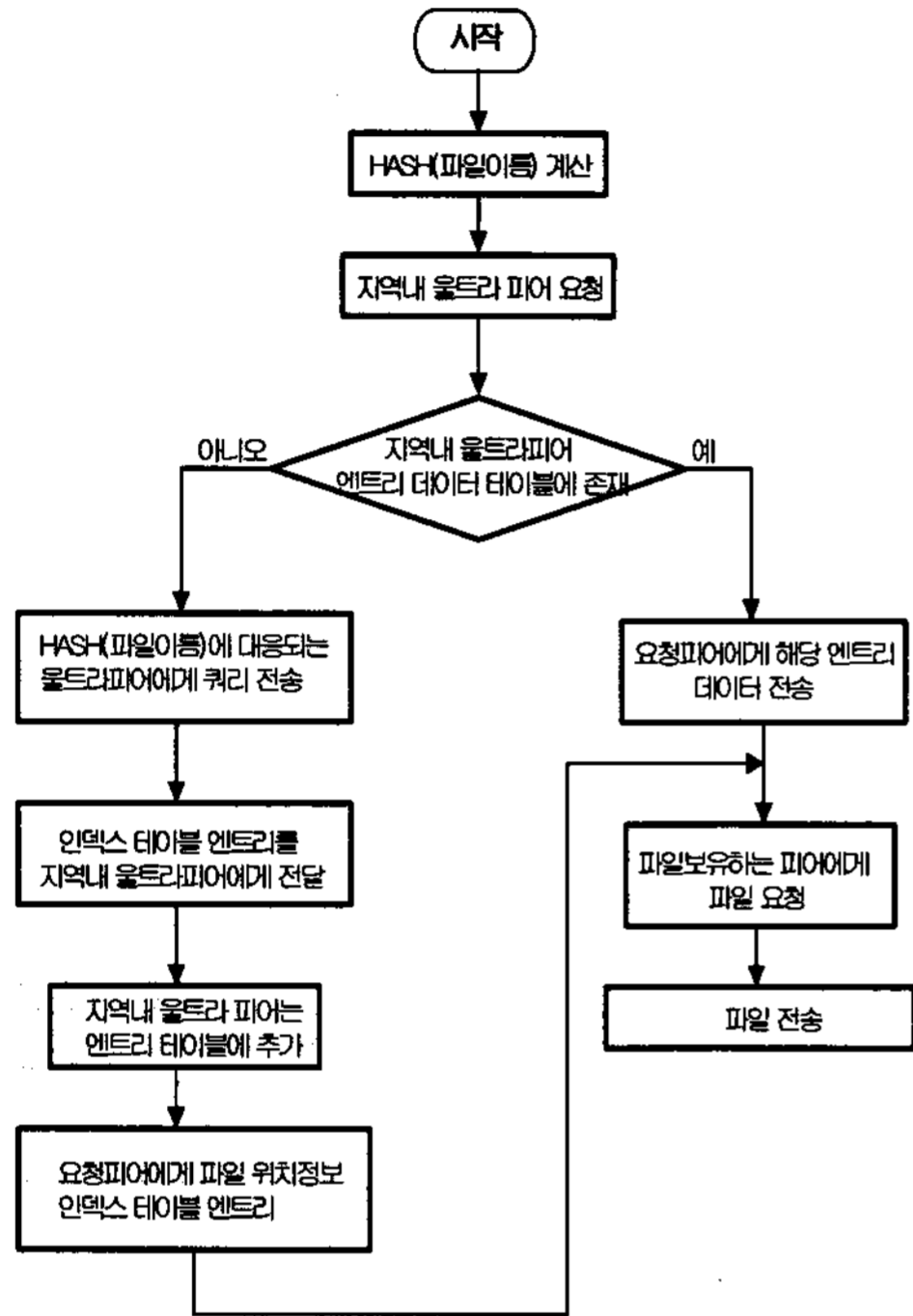
울트라피어를 제외한 지역 내 리프피어 중에서 식(2)에 의해 연결시간이 가장 큰 피어를 찾아 후보피어로 선택한다. 울트라 피어와 후보 피어는 주기적으로 서로의 정보를 주고받으며 일정시간 경과 했음에도 응답이 없다면 후보피어를 선출한다.

◎ Update

피어의 추가와 탈퇴가 빈번하게 일어나므로 데이터의 정확성을 위해 울트라 피어의 탈퇴 시, 핑거테이블과 인덱스 테이블 그리고 엔트리 데이터 테이블을 후보 울트라피어에게 업데이트한다.

IV. 파일검색 알고리즘

리프피어가 울트라피어에게 검색을 요청하고 공유파일을 다운로드 하기까지 과정은 그림 4에서 보는 바와 같다.



[그림 4] 파일 검색 과정

먼저 리프피어에서 쿼리가 발생되면 파일 이름을 해쉬하여 얻은 파일 식별자 id와 파일이름을 지역 내 울트라피어에게 검색요청을 하게 된다. 지역 내 울트라피어는 먼저 엔트리 데이터 테이블을 먼저검색하고 되는데 엔트리 데이터 테이블에 해당파일이 존재하는 경우 요청피어에게 해당 엔트리데이터를 전송한다. 파일을 보유하고 있는 해당 피어들에게 파일을 요청하여 전송받는다. 그러나 엔트리 데이터에 자료가 존재하지 않을 경우 대응되는 파일 식별자 id에 대응되는 울트라 피어에게 쿼리를 전송한다. 쿼리를 수신한 울트라 피어는 파일 식별자 id를 수신한 울트라피어는 인덱스 테이블을 검색하여 지역 내 울트라피어에게 엔트리를 전달한다. 지역 내 울트라피어는 엔트리데이터에 파일명에 대응되는 파일의 위치정보를 추가하여 리스트를 유지한다. 그리고 울트라 피어는 요청피어에게 파일의 위치 정보를 알려준다. 위치정보를 수신한 요청피어는 실제 파일을 보유한 피어에게 파일을 요청하고 파일을 다운로드 받는다. 파일을 보유하는 피어에게 파일을 요청하고 파일을 전송받는다. 각 과정을 마친 후에는 다음 검색요청이 발생 할 때까지 대기하고 다

시 쿼리가 발생할 경우 지역 내 울트라피어에게 검색을 요청함으로써 검색과정을 반복하여 진행 하게 된다.

IV. 결 론

본 논문에서는 울트라피어기반 DHT링으로 구성된 환경에서 울트라피어와 리프피어 그리고 후보노드로 나누어 피어들을 관리한다. 대부분의 피어는 쿼리 검색에만 관심을 보이고 66%정도의 피어는 공유파일을 가지고 있지 않고 단지 검색과 다운로드에만 관심이 있다는 연구 결과가 있다. 제안한 기법은 메타데이터 저장을 위한 추가적인 오버헤드 비용으로 검색의 효율성을 개선한다.

검색의 효율성을 높이기 위해 울트라피어는 핑거테이블, 엔트리 데이터 테이블, 인덱스 테이블을 관리한다. 핑거테이블은 울트라피어로 구성된 DHT링에서 울트라피어의 successor 노드를 가리키는 테이블이다. 엔트리 데이터 테이블은 피어들이 검색했던 문서들의 메타데이터이다. 그리고 인덱스 테이블은 파일을 해쉬하여 얻어진 해쉬값과 일치하는 울트라피어에서 공유자원과 일치하는 (파일명,url)을 관리한다. 울트라피어의 핑거테이블, 엔트리 데이터 테이블 그리고 인덱스 테이블을 두어서 효율적인 검색을 수행하고 유지관리 비용이 줄어들게 된다.

참고문헌

- [1] Napster. <http://www.napster.com>
- [2] Gnutella. <http://www.gnutella.com>
- [3] A. Singla, C. Rohrs, "Ultrapereers: Another Step Towards Gnutella Scalability", Gnutella Developer's Forum, 2002.
- [4] I. Stoica, R. Morris, D. Liben-Newell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakishnan."Chord: a scalable peer-to-peer lookup protocol for internet applications" IEEE/ACE Trans. Netw., 11(1): 17-32, 2003.
- [5] 백승재, 안진호, "파일 공유를 위한 확장적인 울트라 피어기반 DHT 프로토콜" 한국정보기술학회, 한국정보기술학회 논문지 韓術文5卷, 2007.