

# 효과적인 파일 공유를 위한 다중 링 기반 DHT 프로토콜

이유진 · 황영철 · 이동은

청운대학교 인터넷학과

## Multi ring-based DHT Protocol for Efficient File Sharing

Yu-jin Lee · Young-chul Hwang · Dong-eun Lee\*

Dept. of Internet, Chungwoon University

E-mail : {20011741 · 0chul2 · delee}@chungwoon.ac.kr

### 요 약

Peer-to-Peer(P2P) 시스템의 주된 목적은 공유 파일을 효과적으로 관리하며 공유 파일을 저장한 노드의 위치를 보다 효율적으로 검색하는 것이다. 본 논문에서는 Chord 기법을 기반으로 울트라피어 노드들로만 구성된 DHT 링과 그 하부에 또 하나의 서브 DHT 링을 가진 다중 링 구조를 사용하여 기존 DHT 프로토콜에 비해 보다 확장이 용이하며, 보다 효율적인 파일 공유 기법을 제시하였다.

### 키워드

Peer-to-Peer(P2P), Chord, DHT, Multi ring

## 1. 서 론

P2P 검색 기법에는 중앙 집중형 검색 기법과 분산형 검색 기법이 있다.

Napster[1]와 같은 중앙 집중형 검색 기법은 하나의 서버가 연결된 모든 노드들의 파일 정보를 관리하는 구조이다. 서버의 동작으로 각 노드들의 부담은 줄어드는 대신 서버로 집중되는 트래픽의 과부하와 서버에 문제 발생 시 시스템 전체가 마비되는 문제점을 가지고 있다. 이런 단일 서버의 문제점을 해결하기 위해서 분산형 검색 기법이 제시되었다.

분산형 P2P 시스템은 Gnutella[2]와 같은 비구조적 기법과 분산 해쉬 테이블 (distributed hash table:DHT)과 같은 구조적 기법으로 구분할 수 있다.

Gnutella는 파일 검색을 위한 검색 요청 메시지를 이웃하고 있는 다른 피어에게 플러딩 함으로써 전체 네트워크로 브로드캐스팅 한다. TTL을 사용하여 브로드캐스팅 되는 범위를 제한할 수도 있지만 기본적으로 파일 검색 방법이 브로드캐스팅이기 때문에 네트워크 트래픽 증가로 인한 확장성 문제를 야기 시킨다. 또한 검색에 대한 응답의 불규칙성 및 검색 결과의 완전성에도 문제를 갖게 된다.

구조적 P2P 시스템들은 Chord[3], CAN[4], Pastry[5], Tapestry[6]등이 있다.

본 논문에서는 기존의 Chord를 기반으로 울트라피어만으로 이루어진 DHT 링을 구성하고, 일반 노드로 구성된 또 하나의 서브 DHT 링을 구성하여 울트라피어에 포함시키는 다중 링 기반 DHT 프로토콜을 제안한다. 제안한 프로토콜은 기존 DHT 프로토콜에 비해 검색 시 노드 수에 따른 hop 수를 줄이고, 각 노드가 유지해야 할 테이블에 있어 보다 확장적이고 효율적인 파일 공유가 가능하다.

## 2. 관련 연구

DHT는 구조적인 분산 P2P 알고리즘으로 해싱을 통하여 생성된 데이터의 위치 정보들을 시스템에 포함되어 있는 모든 노드들에 균일하게 분산하기 위하여 고안된 lookup 기법이다.

- Chord는 전체 피어(N)들을 IP 주소를 SHA-1과 같은 해쉬 함수를 사용하여 원형 식별자 공간에 할당한다. 데이터의 위치 정보는 (key, value) 쌍으로 표현된다. 여기서 key는 공유 파일 이름이며, value는 공유 파일을 가지고 있는 노드의 URL이다. 공유 파일에 대한 위치를 나타내는 인덱스 정보가 저장될 노드의 위치는 key를 해쉬하여 생성된 해쉬 값과 동일한 노드 식별자에 저장

한다. 공유 파일을 식별자 공간에 대응시킬 때 해쉬 된 키 값과 같은 노드 식별자를 가지는 노드에 저장하거나, 같은 노드 식별자를 가진 노드가 없을 때는 바로 뒤의 노드에 저장한다. 이 노드를 해당 key의 successor 노드라 부른다. 원형 식별자 공간에서 각 노드는 successor의 정보를 유지한다. 또한 각 노드는 전체 네트워크 상의 노드에 대한 정보를 분산된 동적 환경에서 효과적으로 분배시키기 위해 finger table을 유지한다. Finger table은 데이터를 삽입하거나 데이터를 관리하는 노드를 찾기 위해 lookup 메시지를 해당 노드에 전달하기 위해 이용된다.

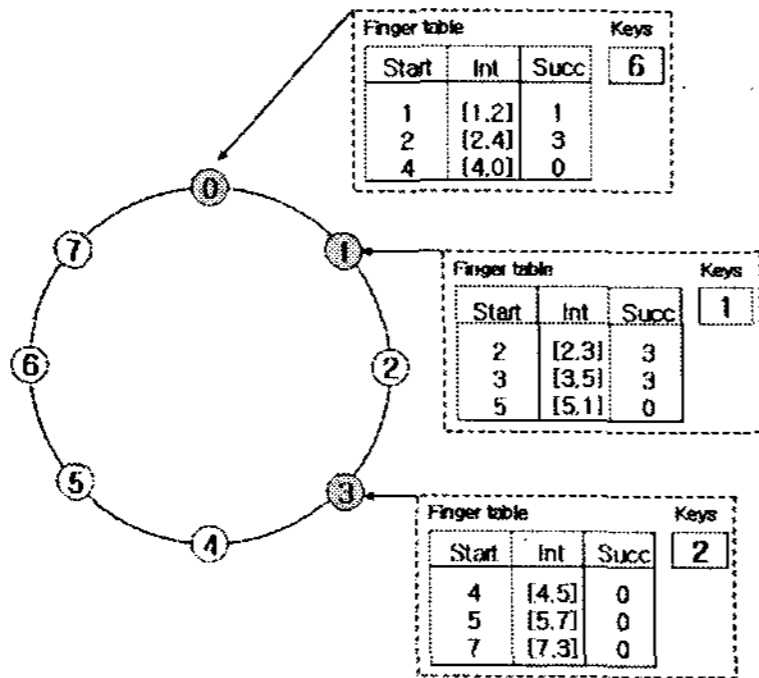


그림 1. DHT을 이용한 Chord

- CAN은 d 차원 존으로 분할되는 가상공간을 갖는 기법이다. 각 노드들은 하나의 존에 대응되며, 해쉬 함수를 통해 가상공간을 분할한 존의 범위에 해당하는 데이터를 저장한다. 또한 노드는 이웃 노드들의 존 정보와 IP 주소를 가진 라우팅 테이블을 갖는다. 검색은 좌표 공간상에서 목적지와 의 거리를 최대한 감소시키도록 이웃 노드를 선택하여 메시지를 포워딩 한다.
- Pastry는 노드 참여 시 IP 주소나 공개 키 암호화 해쉬를 통해 중복되지 않는 랜덤한 128비트 노드 식별자를 만들고, 노드 식별자들이 이름 공간상에 분산되게 하는 기법이다. 노드들은 동일한 prefix를 가진 노드들을 그룹화하고 이를 계층화한다. 검색은 메시지가 가진 목적지와 가장 긴 동일 prefix를 가진 노드로 메시지를 전달하여 목적지 노드에 도착하도록 한다.
- Tapestry는 P2P 링크만을 사용하여 객체에 가장 가까운 위치 사본으로 메시지를 라우팅하는 기법이다. 검색하고자 하는 파일을 가진 노드의 위치는 객체 식별자와 동일한 노드 식별자를 가진 노드로 라우팅 해서 알 수 있다.
- 울트라피어 기반 DHT 링[7]은 파일 검색 및 노드 조인과 리브 오퍼레이션의 비용과 각 노드가 유지해야하는 라우팅 테이블 엔트리의 수 측면에서 기존 DHT 프로토콜에 비해 보다 확장적이고 효율적인 방법을 제공하는 기법이다. 반면 전체 피어와 울트라 피어가 관리하는 노드의 수가 많을수록 검색의 시간과 비용이 많이 발생하는 문

제점들이 있다. 노드의 수가 많아지면 울트라피어가 관리해야 할 정보도 많아지고 관리하는 정보가 많으면 검색 요청이 들어왔을 때 검색하는 시간이 늘어난다.

### 3. 시스템 구조

본 논문에서는 모든 피어들로 구성되는 Chord 링과는 달리 울트라피어만으로 구성된 Chord 링과 일반 노드만으로 구성된 서브 Chord 링의 다중 링 구조로 각 울트라피어가 하나의 서브 링을 관리하도록 구성한다.

#### 3.1 다중 링 구성

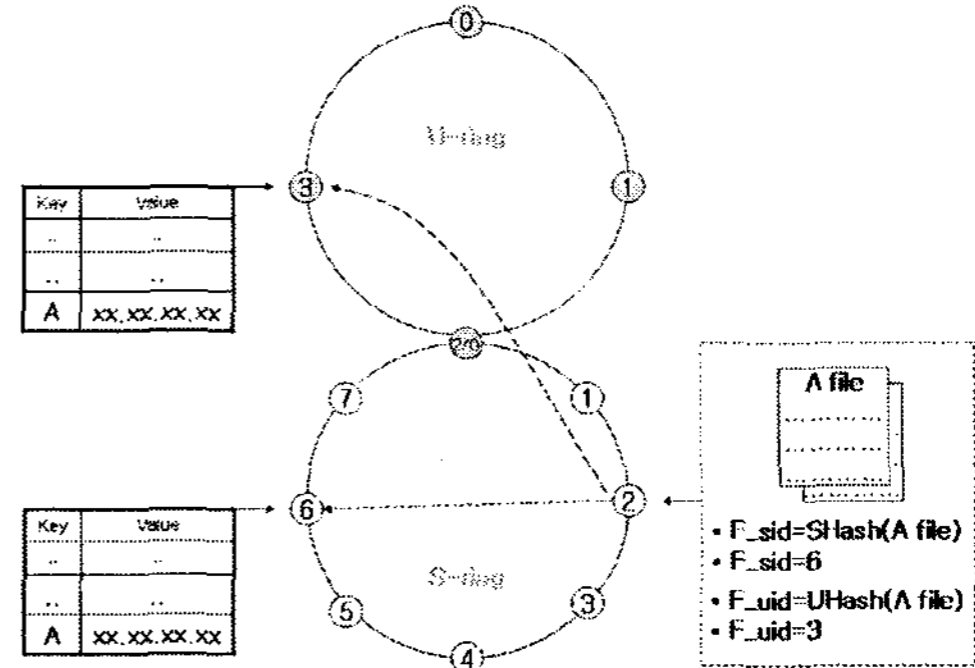


그림 2. DHT 기반 다중 링 구조

제안한 다중 링 구조는 서브 링 내에서의 질의 요청이 실패하였을 시, 울트라피어 링으로 질의 요청이 넘어가기 때문에 하나의 파일에 대하여 두개의 해쉬 값(U-Hash, S-Hash)을 사용한다.

$$F\_sid = S-Hash(\text{file name}) \quad (1)$$

$$F\_uid = U-Hash(\text{file name}) \quad (2)$$

그림2와 같은 경우 서브 링의 2번 노드가 가진 A의 파일 정보를 분산 관리하기 위하여, 식(1)과 식(2)를 사용한다. 식을 사용하여 계산된 값으로 서브 링의 6번 노드와 울트라피어 링의 3번 울트라피어에게 파일 정보를 보낸다. 정보를 받은 울트라피어와 노드는 그 정보를 테이블의 형태로 관리한다. 따라서 하나의 울트라피어는 서브 링에서 사용하는 S-Hash 값으로 해싱 한 데이터 위치 정보를 관리하는 테이블과 울트라피어 링에서 사용하는 U-Hash 값으로 해싱 한 데이터 위치 정보를 관리하는 테이블을 관리한다. 이때 전체 노드(N) 중에서 울트라피어(Nu)가 관리해야 할 서브 링의 노드 수(Nc)는,

$$\frac{(N - Nu)}{Nu} = Nc \quad (3)$$

와 같은 식으로 구할 수 있다.

3.2 노드의 참여, 이탈

1) 참여

새로운 노드가 참여 할 때는 먼저 노드가 어떤 울트라피어가 관리하는 서브 링에 참여하여야 하는가를 판단해야 한다. 참여할 노드의 IP 주소를 U-Hash 값으로 해싱하여 얻어진 노드의 식별자 id값과 매칭되는 울트라피어를 선택한다. 다음으로 IP 주소를 S-Hash 값으로 다시 해싱하여 서브 링의 원형 식별자 공간에서 어떤 위치를 할당받게 될지를 결정한다. 위치를 할당받은 노드는 자신이 공유할 파일을 U-Hash와 S-Hash 값으로 해싱한다. 해싱하여 얻어진 노드의 식별자 id값을 사용하여 각 값과 매칭되는 노드들에게 파일을 정보를 전송하여 공유할 파일 목록을 분산 관리한다.

2) 이탈

이탈 할 노드는 자신이 공유하고 있는 파일들을 U-Hash와 S-Hash로 해싱하여 각 파일 정보를 관리하고 있는 피어를 찾는다. 그리고 해당 피어들에게 이탈 사실을 통보하고 파일 정보의 삭제를 요청한다. 요청을 받는 피어는 데이터 위치 정보 테이블에서 관련 파일 정보를 삭제하고 목록을 갱신 한다. 또한 이탈 할 노드가 관리하고 있던 데이터 위치 정보 테이블은 successor 노드에게 전달한다.

먼저 질의 피어로부터 파일 질의 요청이 들어오면 질의를 받은 노드는 자신이 속한 서브 링에서 우선적으로 검색을 진행한다. 질의된 파일을 S-Hash값으로 해싱하고, 원형 식별자 공간 내 존재하는 노드 중 그 값과 매칭되는 노드 식별자를 찾는다. 그 후 노드가 관리하는 데이터 위치 정보 테이블에 질의된 파일과 동일한 정보가 있는지 검색을 한다. 테이블 내에 질의 파일의 정보가 존재한다면 응답과 함께 파일의 인덱스 정보를 요청 노드에게 전송하여 두 노드 간 다운로드를 수행한다. 만약 파일의 정보를 찾을 수 없다면 해당 서브 링을 관리하는 울트라피어에게 새로운 질의 요청을 한다.

질의를 받은 울트라피어는 해당 파일을 울트라피어의 해쉬 값인 U-Hash로 재해싱하여 얻어진 노드 식별자 id 값으로 울트라피어 링 내 매칭되는 노드 식별자를 찾는다. 해당 노드 식별자가 관리하는 데이터 위치 정보 테이블에서 파일 정보의 존재를 검색하고, 파일 정보가 있을 경우 요청한 노드에게 응답과 함께 파일의 인덱스 정보를 보내어 파일을 가진 노드와 요청 피어 간 다운로드가 수행되고, 파일 정보가 없을 경우 더 이상 검색이 진행될 수 없으므로 요청 피어에게 검색 불가 메시지를 전달한다.

4. 검색 기법

4.1 다중 링 검색기법

그림 3은 본 논문이 제시한 기법에서 질의 요청이 들어오고 질의에 대한 검색 과정과 파일에 대한 다운로드가 이루어지기까지의 전체적인 동작을 도식화한 것이다.

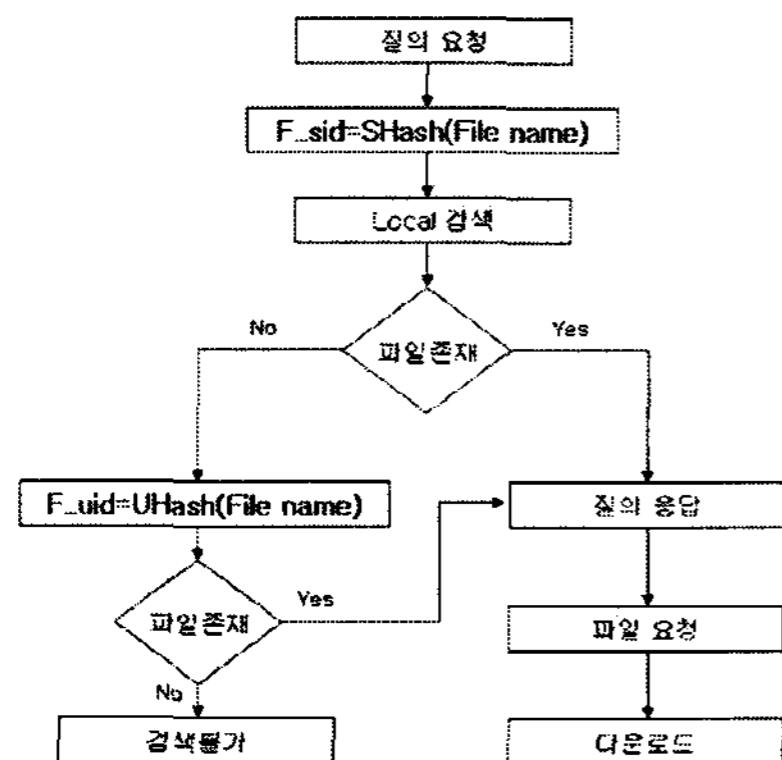


그림 3. 동작 알고리즘

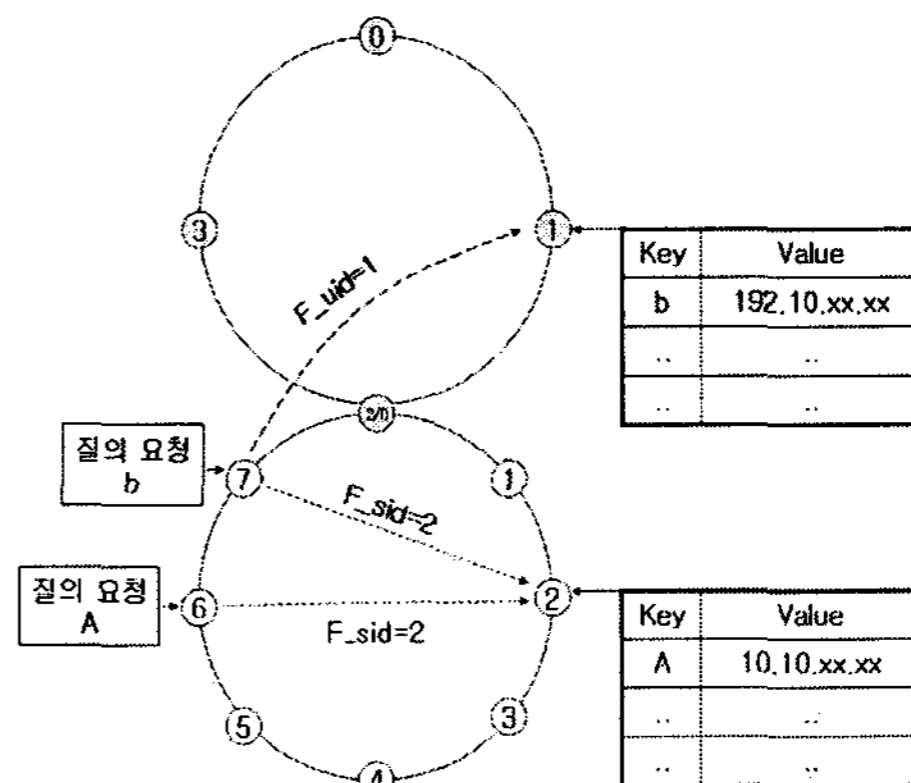


그림 4. 동작 과정

그림 4에서와 같이 A파일과 b파일에 대한 질의 요청이 있었다고 가정한다.

6번 노드에서 요청한 A파일의 경우 S-Hash값과 매칭되는 2번 노드의 데이터 위치 정보 테이블에서 질의 파일에 대한 정보가 존재하므로 2번 노드가 요청 피어인 6번에게 파일의 인덱스 정보를 전송하고 다운로드를 수행한다.

반면, b파일의 경우 S-Hash 값과 매칭되는 2번 노드에서 파일의 정보를 찾을 수 없으므로 자신이 속한 서브 링을 관리하는 울트라피어에게 새로운 질의 요청을 한다. 울트라 피어는 요청을 받으면 파일을 U-Hash로 해싱하고 얻어진 값으로 울트라피어 링 내에서 검색을 시도한다. U-Hash

값과 매칭되는 1번 울트라피어가 관리하는 데이터 위치 정보 테이블에서 파일 정보를 검색하고 해당 파일 정보가 있으므로 요청 피어에게 파일의 인덱스 정보를 전송하고 다운로드를 수행한다.

만일 이 단계에서 1번 울트라피어에도 b파일에 대한 정보가 없다면, 질의를 요청한 7번 노드에게 검색 불가 메시지를 전송하게 된다.

향후 연구 과제로써 네트워크 내 울트라피어의 참여 또는 이탈에 대한 동작에 대하여 보다 정교하고 효과적인 동작을 제시하도록 하겠다. 또한 실제 사용되는 Gnutella와 같은 네트워크 내에 제안한 기법을 포함시켰을 때 실제 시스템에서의 동작 상황과 그에 따른 결과가 어떻게 도출되는지 연구, 측정하고자 한다.

#### 4.2 성능 비교

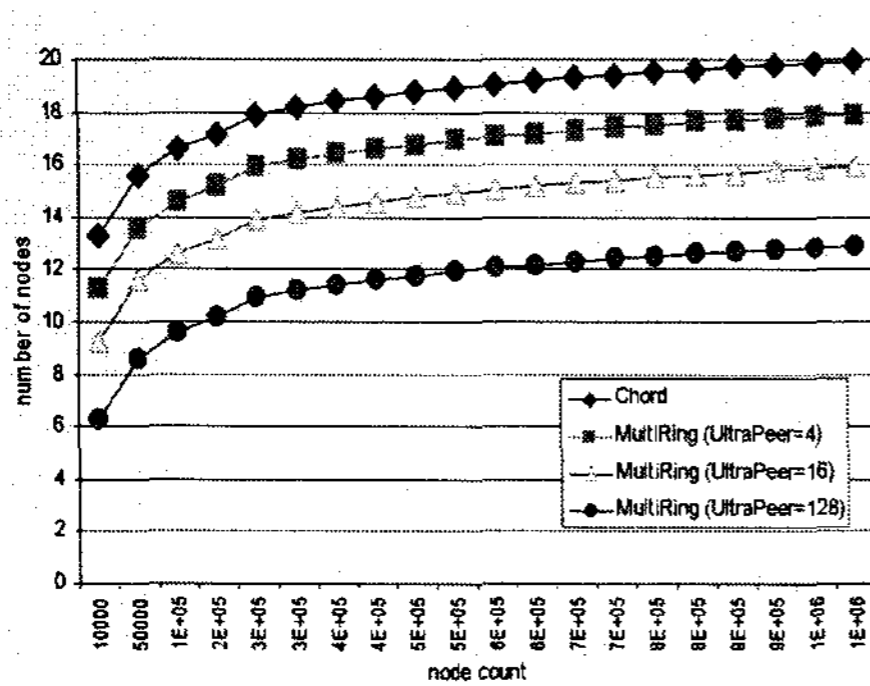


그림 8. 검색 시 노드 수 대비 홉 카운트

그림 5는 기존 Chord와 제안한 기법과의 성능 비교이다. Chord는 10,000의 노드에서 13 hop의 검색 비용을 갖는다. 다중 링에서는 울트라피어의 수를 각각 4, 16, 128개로 가정하면 13, 9, 6의 hop의 비용을 갖는다. 평균적으로 기존 Chord 검색 기법보다 낮은 89.1%, 87.8%, 79.1%의 hop 수로 검색할 수 있는 장점이 있다. 관리해야 할 전체 노드의 수와 울트라 피어의 수가 많을수록 비용 절감의 효과는 더 커진다.

#### 5. 결 론

본 논문에서 제안한 방법은 네트워크에 속한 노드의 수에 상관없이 일정한 검색 시간과 비용을 제공한다. 울트라피어들의 수를 Nu이라하면, 서브 링의 원형 식별자 공간 내에서 검색을 하는데 있어  $O(\log N/Nu)$ 의 hop에서 자원을 탐색할 수 있다. 이는 기존의 Chord보다 hop 수를 줄여 lookup의 효율성을 향상 시킬 수 있는 이점을 제공한다. 또한 일반 노드의 링을 Chord로 구성함으로써 기존의 방법보다 유연하고 확장이 용이한 방식으로 시스템을 구성할 수 있다. 또한 다중 링 형태를 사용함으로써, 기존의 방법보다 좀 더 효율적이며 희귀 파일과 같이 높은 비용을 소요하는 작업의 비용을 줄이고 대중화를 제공할 수 있을 것이라고 기대한다.

#### 참고문헌

- [1] Napster, <http://www.napster.com>
- [2] Gnutella, <http://www.gnutella.com>
- [3] I. Stoica, R.Morris, D.karger, M.F Kaashoek, and H.Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", ACM SIGCOMM'01, Aug.2001
- [4] S. Ratnasamy, P.Francis, M.Handley, R.Karp, "A Scalable Content-Addressable Network", ACM SIGCOMM'01, Aug.2001
- [5] A.Rowstron, P.Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems platforms (Middleware), pp.329-350, November 2001
- [6] B. Y. Zhao, J.Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for faulttolerant wide-area location and routing." Tech. Rpt. UCB/CSD-01-1141, Computer Science Division, University of California, Berkeley, Apr 2001
- [7] 백승재, 안진호, "파일 공유를 위한 확장적인 울트라피어 기반 DHT 프로토콜", 한국정보기술학회논문지, 제5호, pp. 34~42, 2007.3