

# P2P 컴퓨팅 환경 기반의 디스커버리 기법

성균관대학교 김영진 · 엄영익\*

## 1. 서 론

P2P(peer-to-peer) 네트워크 환경이란 "각 피어(호스트)들이 가지고 있는 자원 또는 제공하는 서비스를 피어들 상호간의 직접적인 연결을 통하여 교환할 수 있도록 지원하는 네트워크 환경"이라 정의할 수 있다. P2P 네트워크 모델과 비교해 볼 때 기존의 클라이언트-서버(client-server) 네트워크 모델은 중앙집중형 모델이며, 이러한 환경에서 동작하는 각 피어들은 서버에 접속한 후 연결 세션을 유지하면서 서비스를 제공받는 방식으로 동작하게 된다. 따라서 이러한 클라이언트-서버 환경의 각 피어들은 서버가 운영되는 동안 언제든지 일관성 있는 서비스를 제공받을 수 있다. 그러나 이 모델은 각 피어들로부터 발생하는 검색 메시지들이 서버로 집중되기 때문에, 높은 네트워크 대역폭, 처리 능력, 그리고 방대한 저장 공간을 보유한 서버를 필요로 하게 된다. 또한 각 피어들에게 서비스를 제공하는 서버가 더 이상 운영되지 않는 경우 사용자들이 어떠한 서비스도 제공받지 못하게 된다는 단점을 갖는다[1].

이러한 문제점들을 해결하기 위한 한 가지 방법으로 P2P 네트워크 개념이 사용되기 시작하였다. 모든 자원을 서버가 관리하는 클라이언트-서버 모델과는 달리, P2P 네트워크 모델에서는 각 클라이언트, 즉, 피어들에게 자원이 분산되어 있기 때문에 사용자들은 원하는 자원을 제공하는 피어에게 직접 연결하여 서비스를 제공받게 된다. 이러한 특징을 가진 P2P 컴퓨팅 환경에서는 많은 피어들의 스토리지나 프로세서들을 공동으로 활용하여 대용량의 스토리지 또는 고도의 프로세싱 능력을 제공하도록 지원할 수도 있으며, 각각의 피어에 저장되어 있는 각종 미디어 컨텐츠들이나 공동 작업에 필요한 파일 등의 공유 기능도 지원할 수 있게 된다. 그러나 **P2P 네트워크 모델은 클라이언트-서버 네트워크 모델처럼 특정 피어가 자원을 관리하여 주고 있지 않기**

때문에, 많은 피어들에게 분산되어 있는 여러 자원들을 관리하는 기법이 필요하게 되며, 더불어 P2P 네트워크 구조에 맞는 피어 관리 기법 및 자원 검색 또는 디스커버리 기법 등을 필요로 하게 된다. 본고에서는 P2P 네트워크 환경에서 필요로 하는 여러 가지 디스커버리 기법들에 대해서 알아보도록 한다.

## 2. P2P 컴퓨팅 환경 기반 네트워크 구조

P2P 네트워크 모델은 네트워크의 구조에 따라 혼합형 P2P 네트워크 구조(hybrid P2P network structure)와 순수 P2P 네트워크 구조(pure P2P network structure)로 분류된다[2].

### 2.1 혼합형 P2P 네트워크 구조

이 구조는 서버를 통해 각 피어들에게 자원의 공유 기능을 지원하는 모델로써, 서비스를 제공할 수 있는 중앙 서버(centralized server)와 실질적인 공유 자원을 가지고 있는 피어들로 이루어진다. 중앙 서버는 각 피어들로부터의 네트워크 연결 세션을 유지함으로써 각 피어들 간의 통신에 대한 중재 역할을 한다. 또한, 연결되어 있는 피어들의 GUID(Global Unique Identifier), 공유 자원들에 대한 메타 데이터(metadata) 등을 유지할 수 있으며, 자원의 검색 기능과 기타 통신 기능 등을 제공할 수 있다. 피어들은 서비스를 제공하는 중앙 서버에 접속하여 원하는 자원을 보유한 피어를 검색하게 되지만, 일단 다른 피어와의 네트워크 세션이 성립되면 중앙 서버의 참여 없이도 상호간 정보 교환을 진행할 수 있다. 이러한 구조를 갖는 P2P 네트워크 시스템으로는 대표적으로 Napster가 있다[5]. 이 구조는 기존 클라이언트-서버 네트워크 구조에서 사용자들이 특정 파일을 중앙 서버로부터 받아감으로써 발생하는 네트워크 트래픽을 일반 피어들에게 분산시키기 때문에 각 피어들의 자원을 빠른 속도로 공유할 수 있다는 장점을 가지고 있지만, 자원의 공유를 위해서는 모든 피어들이 일단 중앙

\* 중신회원

서버에 연결해야 하기 때문에 클라이언트-서버 모델의 중앙 서버로 인한 단점을 계속해서 일부 갖게 된다.

## 2.2 순수 P2P 네트워크 구조

이 구조는 피어들에게 서비스를 제공하는 특정 서버를 두지 않고, 모든 피어들로 하여금 서버와 클라이언트의 기능을 동시에 가지게 하는 방식으로, Gnutella와 Freenet 등이 대표적으로 이러한 구조를 갖는다(5). 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. 그러나 수많은 네트워크 연결이 이루어지기 때문에 자원 또는 서비스에 대한 디스커버리를 위해서는 많은 네트워크 트래픽이 발생하게 된다는 단점을 가지기도 한다.

순수 P2P 네트워크는 각 피어들이 형성하고 있는 네트워크 그룹의 특성에 따라 비구조적(unstructured) 네트워크와 구조적(structured) 네트워크로 재분류할 수 있다. 비구조적 네트워크는 다른 피어들과의 연결이나 파일의 위치 등에 관한 어떠한 정보도 관리도 하지 않으며, 초창기의 순수 P2P 네트워크 형태를 그대로 따르는 구조이다. 그러나 구조적 네트워크는 자원 검색을 용이하게 하기 위해 각 피어들로 하여금 특정 규칙을 적용하여 다른 피어들에게 연결하게 함으로써 각 피어들의 네트워크 연결을 보다 구조화된 형태로 구성되게 한다(2,3).

## 3. 디스커버리 기법

P2P 네트워크에서 자원 또는 서비스에 대한 효율적인 검색을 위해 제안되고 있는 검색 기법들은 일반적으로 중앙집중식 디렉토리 모델(centralized directory model), 브로드캐스트 요청 모델(broadcast(flooded) requests model), 그리고 도큐먼트 라우팅 모델(document routing model) 등으로 분류될 수 있다.

### 3.1 중앙집중식 디렉토리 모델

중앙집중식 디렉토리 모델(centralized directory model)은 혼합형 P2P 네트워크 구조에서 사용되는 기법으로, 각 피어들은 지정된 서버에 접속하여 서비스를 제공받게 된다(3,4). 이러한 모델을 이용하는 대표적 시스템으로는 Napster가 있다(5). 이 모델은 서버 기능의 차이에 따라 centralized DB 모델과 decentralized DB 모델로 다시 분류된다.

#### 3.1.1 Centralized DB 모델

각 피어들은 지정된 중앙 서버에 네트워크 연결을 한 후 자신이 가지고 있는 공유 자원에 대한 메타 데이터를 서버에게 전송하게 되며, 중앙 서버는 각 피어들로부터 수신한 메타 데이터들을 저장/관리하게 된다. 차후 중앙 서버가 임의의 피어로부터 검색 메시지를 받게 되면 저장된 메타 데이터에서 검색을 하게 되며, 이러한 작업에 의해 검색된 메타 데이터와 이를 소유하고 있는 피어의 주소 등을 요청한 피어에게 전송하게 된다(1). Centralized DB 모델의 시스템 구조를 그림 1에서 보인다.

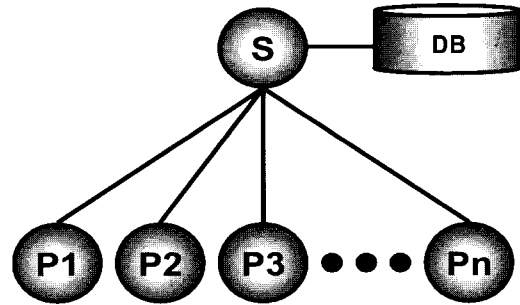


그림 1 Centralized DB 모델

이 방법에서는 각 피어들이 찾고자 하는 자원을 서버에게 문의하여 서버의 DB에서 자원에 대한 검색이 수행되기 때문에, 검색 메시지를 발생시킨 피어 이외의 다른 피어들에게는 네트워크 트래픽이 발생되지 않는다. 그러나 서비스를 받고자 하는 많은 피어들이 중앙 서버에 네트워크 연결을 한 후 자신이 소유한 공유 자원에 대한 메타 데이터를 중앙 서버에게 전송하기 때문에, 중앙 서버의 DB 구축을 위한 많은 네트워크 트래픽이 발생하게 된다. 또한 각 피어들의 검색 메시지가 중앙 서버에 집중되기 때문에, 중앙 서버는 높은 처리 능력과 네트워크 대역폭을 필요로 하게 된다(1).

#### 3.1.2 Decentralized DB 모델

각 피어들은 서비스를 제공하는 중앙 서버에 접속한 후 어떠한 정보도 전송하지 않으며, 중앙 서버는 각 피어들과의 네트워크 연결만을 관리하게 된다. 이러한 모델의 시스템 구조를 그림 2에서 보인다.

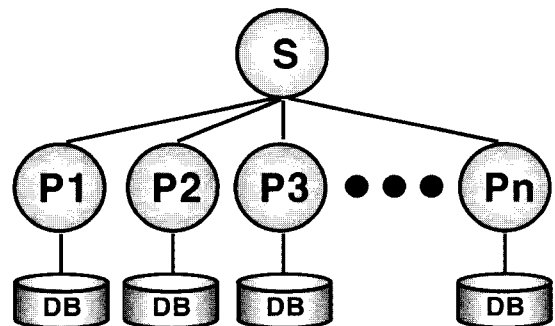


그림 2 Decentralized DB 모델

이 모델에서는 특정 자원을 찾고자 하는 피어가 중앙 서버에게 검색 메시지를 전송하는 경우, 중앙 서버는 그 검색 메시지를 연결되어 있는 모든 피어들에게 전파(relay)하게 된다. 서버로부터 검색 메시지를 받은 각 피어들은 자신이 보유한 자원 중 검색 메시지에서 지정한 자원을 가지고 있을 경우 해당 자원의 메타 데이터를 중앙 서버에게 전송하게 되고, 이 메타 데이터를 받은 서버는 자원 검색을 요청하였던 피어에게 메타 데이터를 전송하게 된다. 따라서 중앙 서버는 각 피어들의 메시지를 중재하는 역할만을 하기 때문에 피어들이 공유한 자원의 메타 데이터를 보관할 필요가 없다. 그러나 각 피어로부터 검색 메시지를 받게 되는 중앙 서버는 연결되어 있는 모든 피어들에게 검색 메시지를 전파해야 하기 때문에 중앙 서버는 어느 정도 높은 네트워크 대역폭을 필요로 하게 된다.

### 3.2 브로드캐스트 요청 모델

브로드캐스트 요청 모델(broadcast(flooded) requests model)은 순수 P2P 네트워크 내에 공유되는 모든 자원들의 인덱스 정보를 유지하는 중앙 서버가 존재하지 않는 환경에서 사용되는 모델이다. 이 모델은 검색 메시지의 전송에 대한 피어들의 참여 여부에 따라 blind search 기법과 informed search 기법으로 분류된다(6,7).

#### 3.2.1 Blind Search 기법

이는 각 피어들이 검색 메시지를 전송할 경우 정해져 있는 단순 규칙에 의해 메시지를 전달하는 방법으로, Gnutella, Modified-BFS, Iterative Deepening, Random Walks, GUESS 등의 기법들이 제시되어 있다.

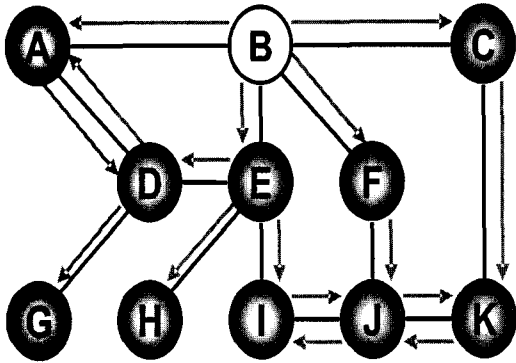


그림 3 Gnutella 알고리즘을 이용한 검색 메시지의 전송  
가. Gnutella 기법

Gnutella 알고리즘은 자원을 찾기 위해 TTL 홉 내에 있는 접근 가능한 모든 피어들에 대해서 BFS(Breath First Search) 기반의 브로드캐스트 기법을 사용한다(3,4,5,6). Gnutella 알고리즘을 이용하여 검색 메시지를 전송하는 모습을 그림 3에서 보인다.

이 방법은 여러 검색 기법 중 가장 간단하여 구현이

용이하다. 비록 자원을 검색할 경우 메시지 내에 TTL을 포함시키기 때문에 네트워크 그룹 내에 참여하고 있는 모든 피어들에게 메시지를 전송하지는 않지만, TTL 값이 허락하는 한 메시지를 받은 피어들은 연결되어 있는 모든 피어들에게 메시지를 전송함으로써 많은 네트워크 트래픽이 발생하게 된다.

#### 나. Modified-BFS 기법

Modified-BFS 알고리즘은 브로드캐스트 기법의 변형된 형태로서, 인접 피어 중 일정 비율에 해당하는 피어들만을 선택하여 전송하는 방법이다(4,6). 이 알고리즘을 이용한 메시지의 흐름을 그림 4에서 보인다.

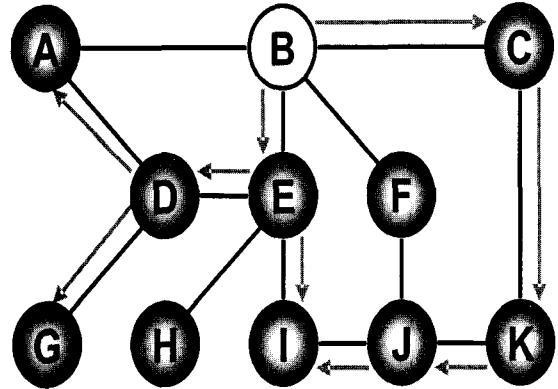


그림 4 Modified-BFS를 이용한 검색 메시지의 전송

이 알고리즘은 인접한 피어들보다 근거리에서 있는 피어들의 검색에 중점을 둔 알고리즘으로써, Gnutella 알고리즘과 비교하였을 때 메시지의 평균 발생률이 확실히 줄어들게 되지만, 메시지를 수신하는 피어의 수와 검색 결과의 수는 Gnutella 알고리즘에서와 유사하다.

#### 다. Iterative deepening 기법

Expanding Ring 기법이라고 불리는 이 알고리즘은 TTL의 값을 점점 증가시키면서 BFS 알고리즘을 이용하여 검색하는 방법이다(3,6,8). TTL 값을 늘리면서 적용시키는 모습을 그림 5에서 보인다.

이 알고리즘은 사용자가 정의한 특정 검색 결과 수에 만족할 때까지 TTL의 수를 늘리면서 검색 메시지를 전송하게 된다. 사용자가 정의한 특정 검색 결과 수를 만족하지 않을 경우 점점 더 큰 TTL 값을 설정한 검색 메시지를 전송하기 때문에 표준 브로드캐스트 기법보다 더 많은 트래픽을 발생시킬 수 있다.

#### 라. Random Walks 기법

Random Walks 알고리즘은 검색 메시지를 전송할 때 일정한 개수의 검색 메시지만을 생성하여 전달하는 기법이다(3,6). 그림 6은 검색 메시지를 TTL=3으로 하여 2개만 전송하는 모습을 보인다.

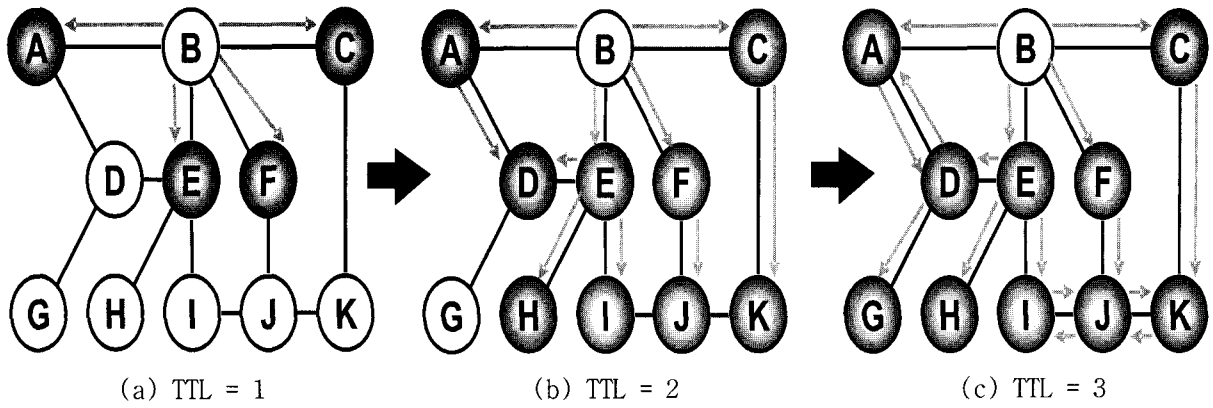


그림 5 TTL에 따른 메시지 전송 범위

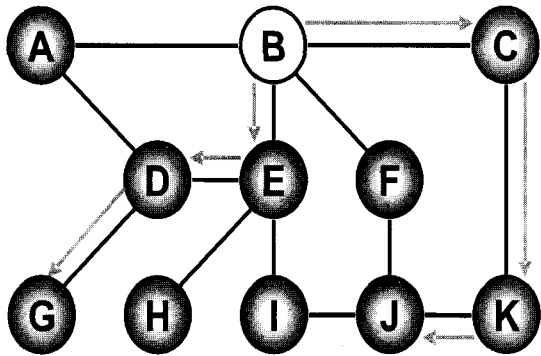


그림 6 Random Walks 알고리즘을 이용한 메시지 전송

검색을 하고자 하는 피어는  $k$ 개의 검색 메시지를 인접 피어들에게 전송하게 되고, 이 메시지를 받은 피어는 검색 메시지에 대한 작업을 처리하게 되며, 검색 메시지 내의 TTL 값이 1 이상이라면 1만큼 감소시킨 후 인접한 피어 중 임의의 피어에게 검색 메시지를 전송하게 된다. 또한 전달되는 검색 메시지들에 의해 검색을 수행한 피어들은 메시지에 대한 작업을 처리한 후 검색 메시지를 최초 발생시킨 피어에게 현재 피어에서의 검색 결과 상태가 만족스러운지를 물음으로써 메시지의 전송 중단을 결정할 수도 있다. 결국 메시지 내에 지정된 TTL 흡수만큼 전달될 수 있기 때문에 최악의 경우 검색 메시지는  $k * TTL$  개수만큼 발생할 수 있다.

#### 가. GUESS 기법

이 알고리즘은 ultrapeer의 개념을 기반으로 한 방법으로, 각 피어들은 해당하는 ultrapeer에 접속하게 되며, ultrapeer는 자식 피어(child peer: leaf-node)들의 프락시 역할을 수행하게 된다[6,9]. Ultrapeer들의 자원 검색 과정을 그림 7에서 보인다.

최초 자식 피어로부터 검색 메시지를 수신한 ultrapeer는 자신의 자식 피어들에게 검색 메시지를 전송하게 되며, 자원 검색 메시지에 해당하는 자원이 검색되지 않을 경우 ultrapeer는 인접한 다른 ultrapeer에게 접속

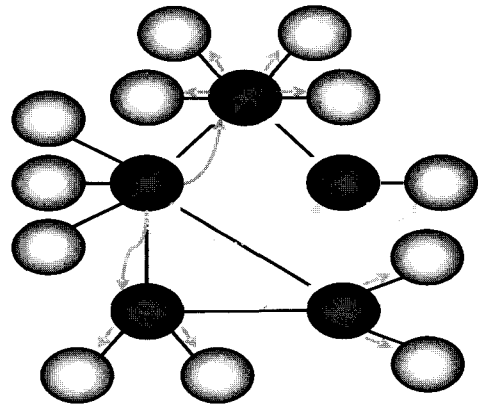


그림 7 GUESS 알고리즘을 이용한 메시지 흐름

하여 검색 메시지를 전송하게 된다. 다른 ultrapeer로부터 검색 메시지를 받은 ultrapeer는 자신의 자식 피어들에게 검색 메시지를 전송하게 되고, 검색된 자원들의 메타 데이터들은 자원 검색을 요청한 피어에게 전송되게 된다. 충분한 검색 데이터 결과가 나올 때까지 ultrapeer는 다른 ultrapeer들에게 검색을 요청하게 되며, 이러한 검색 요청에 있어 ultrapeer들의 순서는 정해져 있지 않다.

#### 3.2.2 Informed Search 기법

이는 각 피어들이 검색 메시지를 전송할 경우 저장된 여러 정보를 이용함으로써 메시지가 전송되어질 피어를 결정하여 전송하는 방법으로, 구체적으로 Intelligent-BFS, APS(Adaptive Probabilistic Search), LI(Local Indices), RI(Routing Indices), DRLP(Distributed Resource Location Protocol), GS(Gnutella with Shortcuts) 등의 기법이 있다.

##### 가. Intelligent-BFS 기법

Intelligent-BFS 알고리즘은 modified-BFS 알고리즘을 확장하여 지식 기반으로 동작하도록 변형시킨 기법이다. 자원을 검색하고자 하는 피어는 modified-BFS 알고리즘을 이용하여 인접한 피어들에게 검색 메시지를

전파하게 된다. 이 후 검색 메시지를 받은 피어들 중에서 해당 자원을 가지고 있는 피어는 역경로(reverse path)로 결과 값을 전송하게 된다. 이때 검색을 요청한 피어와 결과 값을 반환하는 피어 사이에서 결과 메시지를 중재하여 주는 피어들은 결과 메시지로부터 얻게 되는 자원이 검색된 피어에 대한 정보를 자신의 DB에 저장하게 된다. 따라서 이후 어떤 피어로부터 검색 메시지를 수신할 경우 해당 피어는 저장된 결과 값들을 이용하여 검색 메시지가 전달되어야 할 피어를 결정할 수 있게 된다[4,6]. 검색 메시지를 받은 피어들이 저장되어 있는 데이터를 기반으로 전달할 피어를 선택하는 동작

과정을 그림 8에서 보인다.

본 알고리즘을 이용함으로써 네트워크에서 발생하는 검색 메시지의 양을 줄일 수는 없지만, 자원이 있을 법한 피어에게 검색 메시지를 전달함으로써 검색의 적중률을 높일 수 있다. 이후 많은 피어들이 검색 메시지를 전송하게 되면 다른 피어가 보유하고 있는 자원에 대한 정보가 많아지게 되기 때문에 검색 적중률을 향상시킬 수 있다. 그러나 이 알고리즘은 피어 삭제 및 데이터의 삭제 등에 관련된 메시지는 전송하지 않기 때문에, 객체 삭제나 피어 삭제 등이 자주 발생하는 네트워크 환경에는 부정확한 결과를 만들어낼 수 있다.

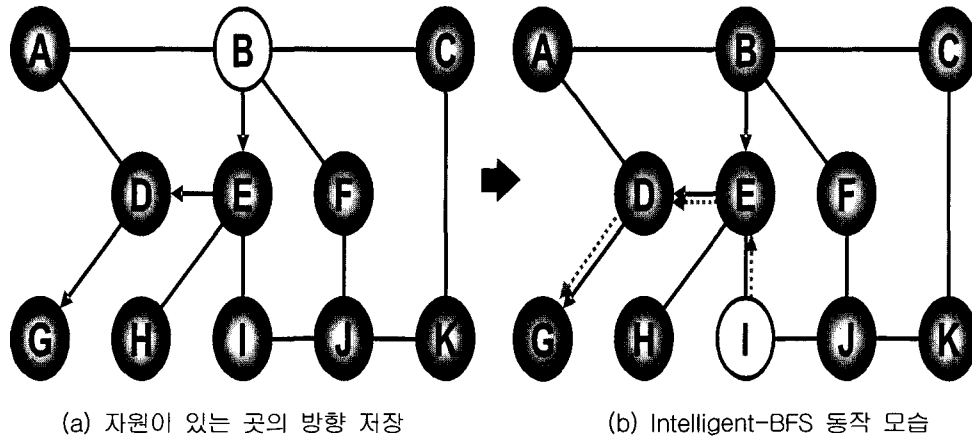
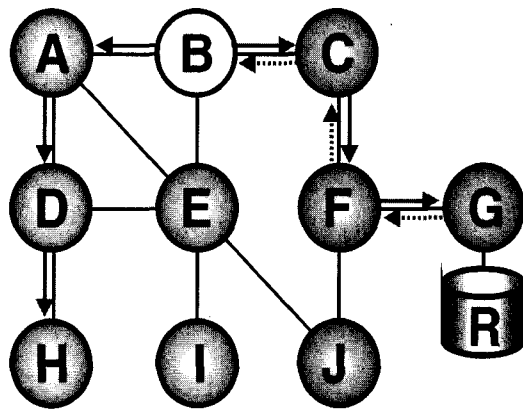


그림 8 Intelligent-BFS 알고리즘을 이용한 메시지 전송



해당 피어	이웃 피어	현재 값	결과 메시지 전송 후	피어 결과 값 도달 후
D	A	20	20	20
	E	20	20	20
	H	20	10	10
A	B	20	20	20
	D	20	10	10
	E	20	20	20
B	A	20	10	10
	C	20	10	
	E	20	20	20
C	B	20	20	20
	F	20	10	
	J	20	20	20
F	C	20	20	20
	G	20	10	
	J	20	20	20

그림 9 APS 알고리즘을 이용한 엔트리 업데이트 과정

나. APS(Adaptive Probabilistic Search) 기법

APS 알고리즘은 각 피어로 하여금 검색 메시지의 송신과 검색에 대한 결과 메시지들의 수신에 있을 때마다 해당 자원에 대한 엔트리의 수치를 변경함으로써 이후의 검색에 대한 적중률을 높이게 하는 방법이다[2,6]. 본 알고리즘을 이용하여 검색하는 과정을 그림 9에서 보인다.

검색 메시지를 수신한 피어는 다른 피어에게 검색 메

시지를 전송할 때 전송하려는 피어에 해당하는 엔트리의 값을 감소시키고, 결과 값이 피드백하는 경우에는 피드백이 발생한 피어에 해당하는 엔트리의 값을 증가시키게 된다. 이후 동일한 검색 메시지를 받는 피어는 인덱스에 저장되어 있는 해당 자원의 확률을 이용하여 검색 메시지가 전송되어질 이웃 피어를 결정하게 된다. 이 방법은 Random walks와 매우 유사한 레벨로 동작하지만, 자

원이 있을 법한 피어에게 전송하기 때문에 보다 많은 결과물을 얻을 수 있게 된다.

다. LI(Local Indices) 기법

LI 기법에서, 각 피어들은  $r$  홉 내에 있는 모든 피어들이 보유한 자원들의 목록을 유지한다[6,8]. 그림 10은 피어 A가 가지고 있는 메타 데이터가 가리키는 실제 자원의 위치를 보인다.

하나의 피어가 검색 메시지를 수신할 경우 해당 피어는  $r$  홉 내의 모든 피어들이 보유한 자원에 대한 메타 데이터를 가지고 있기 때문에 인접한 피어들에게 검색 메시지를 전송한 것 같은 결과를 얻을 수 있다. 이 방법을 이용함으로써 몇 개의 피어에게만 검색 메시지를 전송하여도 많은 피어들에게서 검색한 것과 동일한 결과 값을 얻을 수 있기 때문에 낮은 비용으로도 충분한 결과물을 획득할 수 있게 된다.

라. RI(Routing Indices) 기법

RI 기법에서 각 피어들은 자원을 보유한 실제의 피어를 저장하지 않고 그 자원이 있는 곳의 "방향"을 저장하는 기법이다[2,6]. 표 1은 그림 11의 네트워크 구조에 대해

RI 기법을 이용하여 구축한 RI 테이블을 보인다.

그림 11에서, 피어 A, B, C, D, I, J 등은 몇 개의 검은 선으로 연결되어 있다. 또한 표 1은 각 피어들이 보유한 RI 테이블을 나타내며, 음영으로 표시된 부분은 해당 피어가 보유하고 있는 자원의 종류를 나타낸 것이다. 여기서 #은 해당 피어가 보유하고 있는 전체 자원의 수를 나타내며, DB, N, T, L의 값은 각각 Database, Networks, Theory, Language 자원의 수를 나타낸다. 각 피어들이 보유한 RI들은 검색 메시지를 전송할 때 이용된다. 만약, 피어 A에서 Database와 Language를 찾고자 하는 검색 메시지를 받게 되었을 경우 자신이 보유한 자원부터 검색하게 된다. 만약, 검색 후 충분한 자원이 검색되지 않을 경우 피어 A는 인접한 피어들 중 Database와 Language가 가장 많이 있을 법한 피어를 결정하여 검색 메시지를 전송하게 되며, 표 1의 정보를 기반으로 피어 A는 인접한 피어 중에서 Database와 Language의 비율이 가장 많은 피어 D( $\frac{100}{200} \cdot \frac{150}{200} \cdot 200 = 75$ )를 선택하게 된다(구체적 메커니즘은 참고문헌 참조). 이러한 과정을 통해 검색 결과를 만족할 때까지 검색 메시지를 각 피어들에게 전송하게 된다[10].

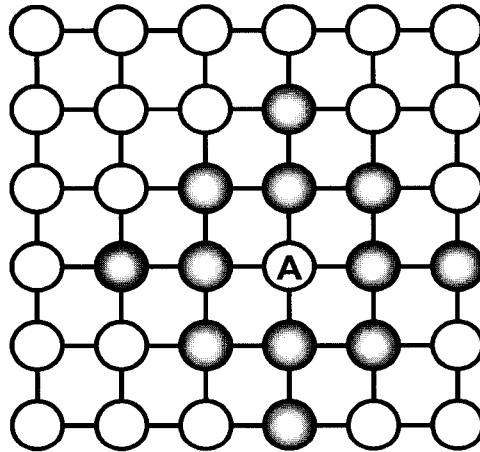


그림 10 피어 A가 검색을 대신 처리할 수 있는 피어들을 표시한 모습

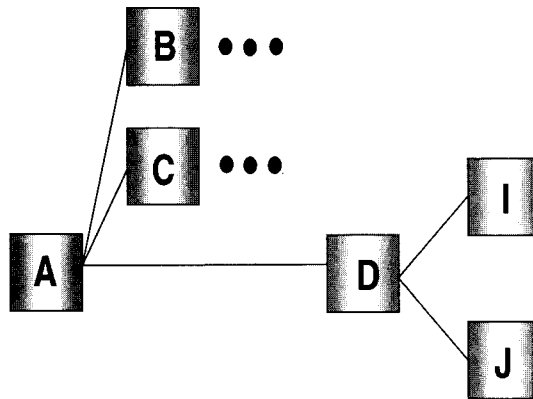


그림 11 RI 기법을 이용하여 연결되어 있는 모습

표 1 각 피어들의 RI 테이블

	#	DB	N	T	L
A	300	30	80	0	10
B	100	20	0	10	30
C	1000	0	300	0	50
D	200	100	0	100	150

(a) Node A의 Routing Indices

	#	DB	N	T	L
D	100	60	0	60	75
A	1400	50	380	10	90
I	50	25	0	15	50
J	50	15	0	25	15

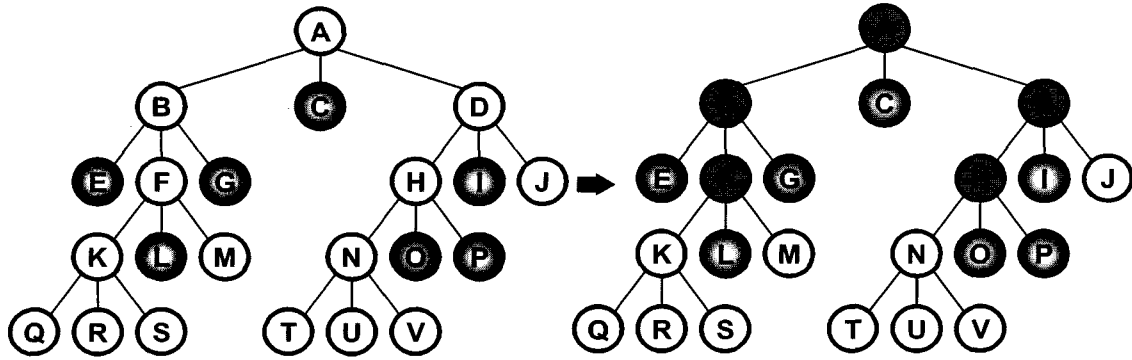
(b) Node D의 Routing Indices

	#	DB	N	T	L
I	50	25	0	15	50
D	1550	125	380	95	190

(c) Node I의 Routing Indices

	#	DB	N	T	L
J	50	15	0	25	25
D	1550	135	380	85	215

(d) Node J의 Routing Indices



(a) 검색 메시지 전송 전 상태

(b) 검색 메시지의 동작 이후 상태

그림 12 DRLP 알고리즘을 기반으로 한 각 피어들의 메타 데이터 유지 모습

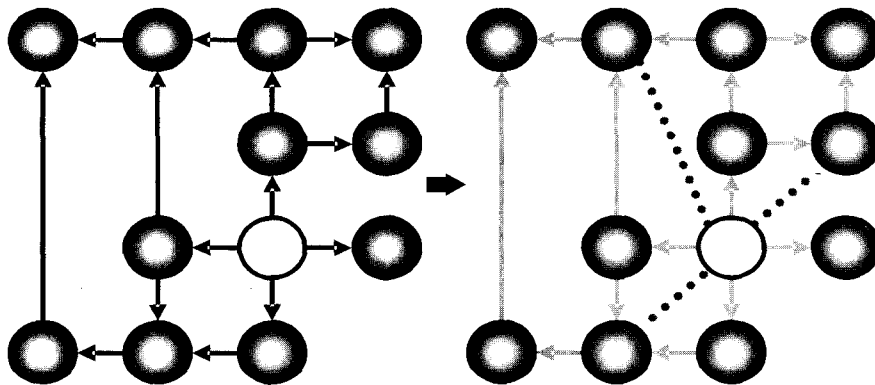


그림 13 GS 알고리즘에 의한 shortcuts

마. DRLP(Distributed Resource Location Protocol) 기법 LI 알고리즘과 유사하며, 이 기법을 이용하는 피어들은 처음부터 다른 피어들의 메타 데이터를 저장하지 않는다. 다른 피어로부터 검색이 이루어질 경우 각 피어들은 결과 뿐만 아니라 각 피어들의 메타 데이터도 함께 전송하여 LI 테이블을 업데이트하게 된다. 이 방법을 이용할 경우 각 피어들은 자신이 보유한 자원들을 가리키는 LD(Local Directory)와 다른 피어의 LD를 가리키는

DC(Directory Cache)를 가지게 된다[6,11]. 그림 12는 검색 메시지의 전송으로 인해 피어의 DC가 업데이트 되는 과정을 보인다.

그림 12에서, 흰색 피어는 DC를 가지고 있지 않은 피어이며, 음영이 있는 피어는 DC를 가지고 있는 피어를 뜻한다. 검색을 하고자 하는 피어 A는 자원의 위치에 대한 정보를 가지고 있지 않기 때문에 인접한 이웃 피어들에게 검색 메시지를 전송하게 된다. 만약 검색 메시지

를 받은 피어에서 자원이 발견된다면 자원을 발견한 피어는 자원을 요청한 피어에게 역경로와 자원의 메타 데이터가 저장된 메시지를 전송하게 된다. 또한 검색 결과를 전송하는 피어는 보유하고 있는 DC 정보도 역경로로 전송하기 때문에, DC 관련 메시지를 수신한 피어들은 특정 피어의 DC 정보를 유지할 수 있게 된다. 그러나 자원이 발견되지 않는다면 같은 방법으로 이웃한 피어들에게 메시지를 전송하게 된다. 피어 A의 검색 메시지를 수신한 피어 C는 이웃한 다른 피어들의 DC를 가지고 있기 때문에 자신이 가지고 있는 DC에서 자원을 검색하게 되며, 인접한 피어들에게는 검색 메시지를 전송하지 않는다. 그러나 피어 A의 검색 메시지를 수신한 피어 B와 D는 이웃한 다른 피어들의 DC를 가지고 있지 않기 때문에 인접한 피어들에게 검색 메시지를 전송하게 된다. 그림 12-a에서 피어 A가 검색을 시도함으로써 검색 결과 메시지가 거치게 되는 피어 A, B, D, F 그리고 H는 검색 단계에서 다른 피어들의 DC가 추가되기 때문에, 그림 12-b에서 보이는 바와 같이, 피어 A가 피어 B, C 그리고 D에게만 메시지를 송신하여도 검색 메시지를 수신한 피어들은 이웃한 다른 피어들에게 검색 메시지를 전달하지 않아도 된다.

이 알고리즘을 이용하게 되면 처음에는 자원의 위치를 찾기 위해서 BFS 알고리즘을 이용하여 많은 검색 메시지들을 전송하게 되지만 이후 시도하는 검색 과정에서는 DC를 보유하고 있는 피어들에 의하여 메시지 전송 과정이 감소하게 된다. 따라서 검색에 의해 발생하는 많은 비용과 큰 작업 부하를 줄일 수 있게 된다. 그러나 네트워

크 구조가 변경될 경우 BFS 방법을 사용하여 다시 구축하기 때문에, 이러한 네트워크 구조의 변화가 자주 일어나게 된다면 검색의 실패율이 증가하게 되고 BFS 검색에 의해 많은 네트워크 트래픽이 발생할 수 있게 된다.

#### 바. GS(Gnutella with Shortcuts) 기법

GS 기법은 Gnutella 알고리즘을 기반으로 하며, 검색 이후 만족스러운 자원을 가지고 있는 피어를 shortcut 피어로 등록하는 방법을 말한다(6,12). GS 기법을 구축하는 모습을 그림 13에서 보인다.

검색을 원하는 피어는 브로드캐스트 기법을 이용하여 원하는 자원을 찾게 되며, 이후 검색 결과를 받은 피어는 검색된 피어들을 shortcut로 사용 가능하게 된다. 따라서 이후의 검색 과정에서 shortcut에 등록된 피어에게 직접 연결하여 검색 메시지를 전송함으로써 검색을 요청하는 피어는 보다 빠르게 자원을 검색할 수 있게 된다. 이러한 shortcut 경로에 해당하는 피어들은 "shortcut-list"라 하는 곳에 저장이 된다. 이 리스트에 저장되어 있는 shortcut 피어에게 접속하여 원하는 자원을 찾지 못할 경우 인접한 피어들에게 브로드캐스트 기법을 이용하여 재검색을 시도하게 되며, 검색 조건을 만족하는 피어가 검색되었을 때 shortcut list를 업데이트시키게 된다.

### 3.3 도큐먼트 라우팅 모델

도큐먼트 라우팅 모델(document routing model)은 순수 P2P 네트워크에서 사용되는 검색 모델로, 공유되는 콘텐츠의 ID를 기반으로 자원을 검색하는 모델이다.

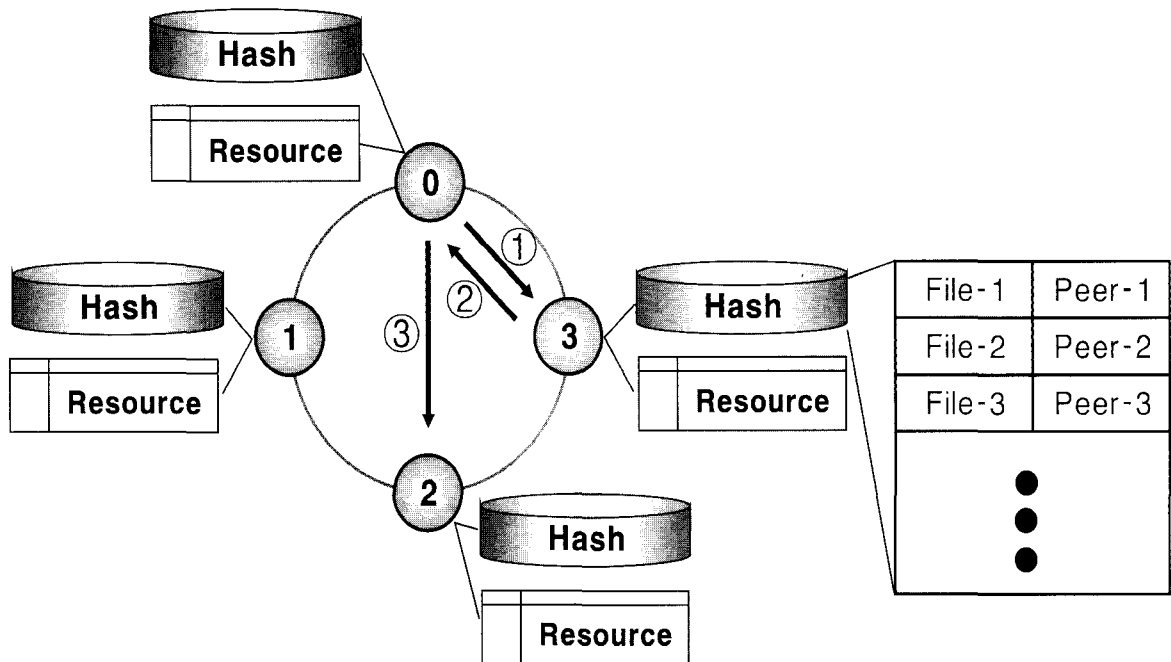


그림 14 DHT를 이용한 자원 검색 과정



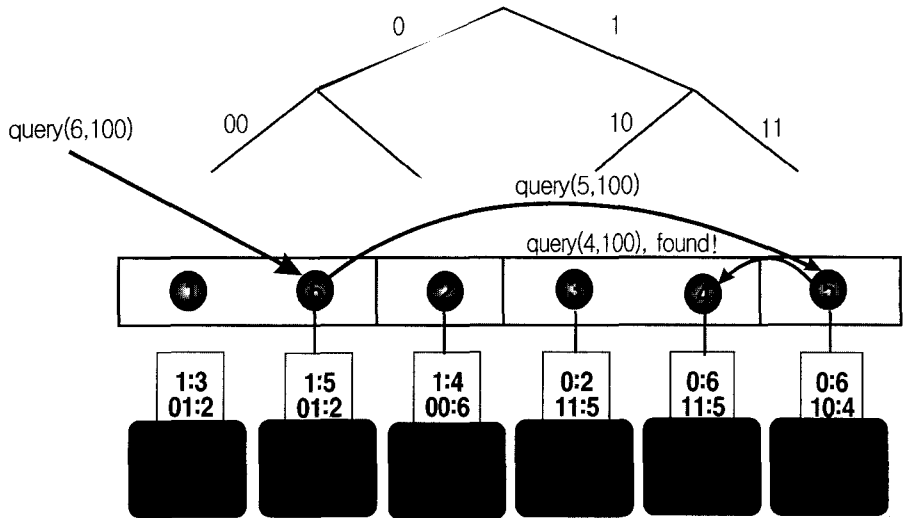


그림 15 가상 이진 검색 트리에서의 자원 검색 과정

**3.3.1 DHT(Distributed Hash Table) 기법**

이는 해쉬(hash) 테이블을 이용한 검색 기법으로써, 모든 피어들은 공유하고 있는 자원들에 대한 메타 데이터를 해쉬 값에 의해 지정된 피어에게 전송하며, 메타 데이터를 수신한 피어들은 다른 피어로부터 검색 요청이 들어올 경우 자원의 위치를 알려주는 방식으로 동작하게 된다[6,13,14,15]. 그림 14는 DHT를 이용한 자원 검색 과정을 보인다.

그림 14에서, 자원 File-2를 검색하고자 하는 피어 0은 File-2에 해당하는 해쉬 값 3을 계산하게 된다. 이후 해쉬 값 3에 해당하는 피어 3에게 File-2에 대한 검색 메시지를 전송(①)하게 되고, 이를 수신한 피어 3은 File-2를 보유하고 있는 피어 위치를 알려주게 된다(②). 이후 검색 결과를 받은 피어 0은 File-2가 존재하는 피어 2에게 접속(③)하여 자원을 얻을 수 있게 된다[1,15,16].

**3.3.2 VBST(Virtual Binary Search Tree) 기법**

이는 Gridella에서 사용된 P-Grid(Peer Grid) 구조로서, 피어가 공유하고 있는 자원의 각 사본들을 네트워크 그룹 내에 있는 여러 피어들에게 분포시키고 각 피어로 하여금 필요한 자원을 효율적으로 검색할 수 있도록 라우팅 테이블을 제공하는 방법이다[5,17,18,19].

그림 15는 피어 6이 자원 100을 찾자 하는 검색 메시지를 수신한 경우의 한 가지 시나리오를 보여준다. 피어 6은 라우팅 테이블에 기록된 자료(1:5)에 의해 1로 시작하는 자원은 피어 5에 있다는 것을 알게 되고 이에 따라 피어 5에게 검색 메시지를 전송하게 된다. 또한, 검색 메시지를 받은 피어 5는 라우팅 테이블에 기록된 자료(10:4)에 의해 10으로 시작하는 자원이 피어 4

에 있다는 것을 알게 되고 피어 4에게 검색 메시지를 전송하게 된다. 결과적으로 검색 요청자는 자원 100이 있는 피어 4를 검색할 수 있게 된다.

**3.4 하이퍼큐브 토폴로지(Hypercube topology) 기반 Broadcast and Search 알고리즘**

이는 하이퍼큐브 모양을 이용한 검색 기법이다. 각 피어들의 연결을 하이퍼큐브 모양으로 구축하여 각 피어들의 네트워크 연결에 weight를 부여함으로써, 메시지를 전달시 항상 높은 weight를 가진 방향으로 메시지를 전송하는 방법이다[20]. 하이퍼큐브를 기반으로 하여 메시지를 전송하는 모습을 그림 16에서 보인다.

그림 16은 피어 0이 자원을 검색하기 위해 검색 메시지를 송신하고, 이를 받은 피어들이 검색 메시지를 재전송 하는 경로를 보여준다. 그림 16-a에서, 최초 자원을 검색하기 원하는 피어 0은 인접한 피어 1, 2 그리고 4에게 검색 메시지를 전송하게 된다. 검색 메시지를 수신한 피어 1은 weight가 2인 경로로 메시지가 수신되었기 때문에 이보다 높은 weight를 가진 경로로 메시지를 전송하게 된다. 그러나 피어 1은 다른 경로의 weight가 각각 0, 1이기 때문에 더 이상 검색 메시지를 전송하지 않는다. 피어 2는 weight가 1인 경로로 메시지를 수신하였기 때문에 이보다 높은 weight인 2를 가진 경로로 검색 메시지를 전송하게 된다. 이와 같은 방법으로 각 피어들로부터 검색 메시지들이 발생되며, 그림 16을 기반으로 한 네트워크 구조에서는 검색 요청에 의해 발생하는 검색 메시지의 총 개수는 7개가 된다. 이 방법은 Gnutella 알고리즘에서 각 피어들이 메시지들을 중복으로 수신할 수 있는 문제를 해결할 수 있기 때문에 많은 네트워크 트래픽을 감소시킬 수 있다.

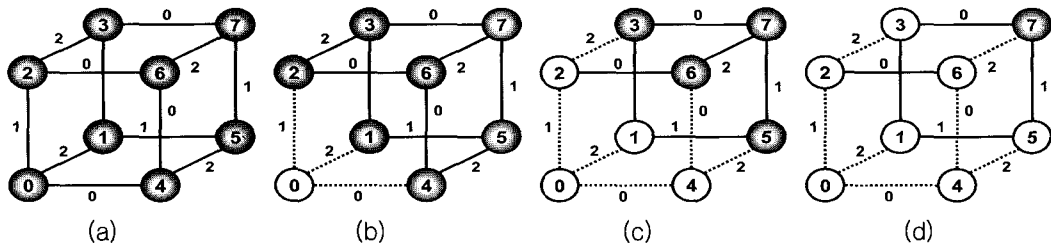


그림 16 Hypercube를 이용한 메시지 전송 과정

검색기법		비	교
Blind Search 기법	Gnutella	<ul style="list-style-type: none"> <li>순수 P2P 컴퓨팅 환경에서 자원을 검색하기 위해 사용하기 가장 쉬운 방법               <ul style="list-style-type: none"> <li>불규칙적으로 연결되어 구성된 네트워크 구조에서 적용하기 용이</li> <li>주위에 연결된 피어들에게 TTL 값에 의해 메시지를 전송</li> <li>디스커버리 기법 중 가장 많은 네트워크 트래픽 발생</li> </ul> </li> </ul>	
	Modified-BFS	<ul style="list-style-type: none"> <li>Gnutella 알고리즘을 사용할 때보다 검색 메시지에 의한 네트워크 트래픽 감소</li> <li>인접한 모든 피어들에게 전송하는 것이 아니기 때문에 Gnutella 알고리즘보다 적중률도 감소</li> </ul>	
	Iterative Deepening	<ul style="list-style-type: none"> <li>가까운 피어들부터 자원 검색 메시지를 전송</li> <li>원하는 결과를 얻지 못하였을 경우 TTL의 값을 증가시키면서 재검색</li> <li>작은 TTL 값으로도 원하는 결과물을 얻을 경우 Gnutella 알고리즘만을 사용하는 것보다 효율적</li> <li>초기 검색에 대한 결과물이 없을 경우 BFS 기법보다 많은 네트워크 트래픽이 발생될 수 있음</li> </ul>	
	Random Walks	<ul style="list-style-type: none"> <li>검색에 의해 발생될 메시지를 한정하여 전송하는 방법</li> <li>Modified-BFS 알고리즘보다 적은 검색 메시지 발생</li> <li>Modified-BFS 알고리즘보다 적은 검색 메시지로 인해 검색 결과물의 양도 줄어들게 됨</li> </ul>	
	GUESS	<ul style="list-style-type: none"> <li>ultraproviders들은 각각 소규모 네트워크를 구성</li> <li>검색 과정의 1단계는 ultraprovider에 접속되어 있는 피어들에게 한정               <ul style="list-style-type: none"> <li>만족스러운 결과 값을 얻지 못하였을 경우</li> <li>→ 다른 ultraprovider에게 검색 메시지를 전송하여 자원을 검색</li> <li>→ 많은 ultraprovider에게 접속하게 되면 네트워크 트래픽 증가</li> </ul> </li> </ul>	
Informed Search 기법	Intelligent-BFS	<ul style="list-style-type: none"> <li>임의로 선택된 피어에게 검색 메시지를 전송</li> <li>→ 낮은 적중률을 보이는 modified-BFS 알고리즘의 단점이 해결됨</li> <li>modified-BFS 알고리즘보다 검색 메시지의 수가 줄어들지 않음</li> <li>modified-BFS 알고리즘보다 많은 결과물을 얻을 수 있음</li> </ul>	
	APS	<ul style="list-style-type: none"> <li>검색 과정에 대한 결과물이 많아질수록 이후의 검색 과정에서 많은 결과물을 얻을 수 있음</li> <li>검색하고자 하는 자원에 대해 여러 피어들을 선택하여 메시지를 전송할 수 있음</li> <li>검색 메시지와 검색 결과 메시지에 따라 자원에 해당하는 피어의 비율이 변경</li> <li>→ 검색되어질 확률이 높은 여러 피어들에게 검색 메시지를 전송할 수 있음</li> </ul>	
	LI	<ul style="list-style-type: none"> <li>각 피어들은 일정 홉 이내에 존재하는 모든 피어들의 메타 데이터를 보관</li> <li>검색 메시지를 전송하지 않아도 검색 결과물을 얻어낼 수 있음</li> <li>네트워크 그룹에 참여할 경우 공유한 자원의 메타 데이터들을 인접한 피어들에게 전송</li> <li>네트워크 그룹에 참여/탈퇴하는 피어들이 많아지는 경우 LI 테이블 구축에 필요한 메시지 증가</li> </ul>	
	RI	<ul style="list-style-type: none"> <li>각 피어들이 보유한 자원들은 분류되어 있음</li> <li>DFS 방법과 유사</li> <li>검색하고자 하는 데이터의 부분이 명확할 경우 많은 결과물을 얻을 수 있음</li> <li>특정 피어로부터 원하는 결과물을 얻지 못하였을 경우 다시 다른 피어에게 검색 시도</li> <li>네트워크 그룹에 참여/탈퇴하는 피어들이 많아질 경우 RI 테이블 내 데이터들의 효용가치 감소</li> </ul>	
	DRLP	<ul style="list-style-type: none"> <li>LI 알고리즘과 유사</li> <li>초기에 모든 피어들의 자원을 관리하지 않음</li> <li>검색 과정이 이루어질 경우에만 DC를 업데이트하게 됨</li> <li>LI 알고리즘보다 네트워크 트래픽을 줄일 수 있음</li> </ul>	
	GS	<ul style="list-style-type: none"> <li>Gnutella 알고리즘과 검색 메시지의 양은 동일</li> <li>Gnutella 알고리즘보다 많은 결과물을 얻을 수 있음</li> </ul>	
Dccurent Routing 기법	DHT	<ul style="list-style-type: none"> <li>해쉬 값에 의한 자원 검색</li> <li>해쉬 테이블 구축을 위해 네트워크 그룹에 참여하고 있는 피어의 수를 알아야 함</li> </ul>	
	VBST	<ul style="list-style-type: none"> <li>완벽한 분산 구조</li> <li>모든 피어들은 검색을 위한 라우팅 테이블을 가지고 있음</li> </ul>	

#### 4. 디스커버리 기법 비교

순수 P2P 네트워크 환경에서의 자원 디스커버리 기법들을 정확히 비교할 수는 없다. 이는 네트워크의 구조가 정해져 있지 않고 각 피어들의 특성에 따라 다른 결과를 보일 수 있기 때문이다. 이러한 각 디스커버리 기법의 특성 및 성능 등은 네트워크의 안정성과 자원의 분포 정도에 좌우되기도 한다.

#### 5. 결 론

P2P 네트워크는 본질적으로 서버가 없이 모든 피어들이 동등한 입장에서 서비스를 제공하거나 받도록 지원하는 네트워크 환경을 말한다. 따라서 각 피어들이 공유하고 있는 자원들의 관리 기법과 다른 피어들이 공유한 자원들 중에서 원하는 자원을 찾을 수 있는 효율적인 검색 기법이 절실하게 요구된다.

본고에서는 혼합형 P2P 네트워크 구조에서의 자원 검색보다 순수 P2P 네트워크 구조에서의 자원 검색 기법을 주로 제시하였다. 이는 각 기관에서 P2P 네트워크 구조를 서버 없는 구조로 방향을 잡고, 순수 P2P 컴퓨팅 환경에서 각 피어들의 네트워크 연결을 구조화하여 효율적인 검색을 지원할 수 있도록 하는 연구가 활발하게 진행되고 있기 때문이다. 본고에서 나열한 여러 가지 자원 검색 기법들은 네트워크 특성에 따라 각기 다른 성능이 나타나게 된다. 따라서 순수 P2P 컴퓨팅 환경에서 검색 기법을 적용할 경우에는 설계하고자 하는 네트워크의 구조에 맞는 검색 기법을 사용하여 전체적인 효율성을 높일 수 있도록 해야 한다.

#### 참고문헌

- [1] D. S. Milojevic, et. al., "Peer-to-Peer Computing," HP Technical Report, HP Laboratories, Mar. 2002.
- [2] D. Tsoumakos and N. Roussopoulos, "Adaptive Probabilistic Search for Peer-to-Peer Networks," Proc. of the 3rd IEEE International Conference on P2P Computing, Sep. 2003.
- [3] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," ICS, 2002.
- [4] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A Local Search Mechanism for Peer-to-Peer Network," Proc. of the 11th International Conference on Information and Knowledge Management, McLean, Virginia USA, Nov. 2002.
- [5] K. Aberer and M. Hauswirth, "An Overview on Peer-to-Peer Information Systems," Workshop on Distributed Data and Structures(WDAS-2002), Paris, France, 2002.
- [6] D. Tsoumakos and N. Roussopoloulos, "Analysis and Comparison of P2P Search Methods," Technical Report, University of Maryland, Dept. of Computer Science, Nov. 2003.
- [7] E. Cohen, A. Fiat, and H. Kaplan, "Associative Search in Peer to Peer Networks: Harnessing Latent Semantics," Proc. of IEEE INFOCOM, San Francisco, California, USA, Mar. 2003.
- [8] B. Yang and H. Garcia-Molina, "Improving Search in Peer-to-Peer Networks," Proc. of the 22nd International Conference on Distributed Computing Systems (ICDCS), 2002.
- [9] B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network," Proc. of the IEEE International Conference on Data Engineering, Mar. 2003.
- [10] A. Crespo and H. Garcia-Molina, "Routing Indices For Peer-to-Peer Systems," Proc. of the International Conference on Distributed Computing Systems, Jul. 2002.
- [11] Daniel A. Menasce and Lavanya Kanchanapalli, "Probabilistic Scalable P2P Resource Location Services," ACM SIGCOMM, Sep. 2002.
- [12] K. Sripanidkulchai and Bruce Maggs, Huio Zhang, "Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems," Proc. of the IEEE INFOCOM, San Francisco, California, USA, Mar. 2003.
- [13] Sean C. Rhea and John Kubiatowics, "Probabilistic Location and Routing," Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), Jun. 2002
- [14] P. Ganesan, Q. Sun and H. Garcia-Molina, "YAPPERS: A Peer-to-Peer Lookup Service over Arbitrary Topology," Proc. of IEEE

INFOCOM, San Francisco, California, USA, Mar. 2003.

- [15] M. Freedman and R. Vingralek, "Efficient Peer-to-Peer Lookup Based on a Distributed Trie," Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS 02), Mar. 2002.
- [16] I. Stoica, et. al., "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," In Proc. of the ACM SIGCOMM '01 Conference, San Diego, California, Aug. 2001.
- [17] Y. Lee, C. Oh, and, E. K. Park, "Intelligent Knowledge Discovery in Peer-to-Peer File Sharing," Proc. of the 11th International Conference on Information and Knowledge Management, 2002.
- [18] K. Aberer, M. Puceva, M. Hauswirth, and R. Schmid, "Improving Data Access in P2P Systems," IEEE Internet Computing, Jan. 2002.
- [19] K. Aberer, et. al., "P-Grid: A Self-organizing Structured P2P System," ACM SIGMOD Record, Sep. 2003.
- [20] M. Schlosser, et. al., "A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services," Peer-to-Peer Computing, 2002.

---

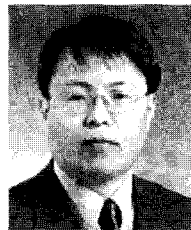
**김 영 진**



2002. 2 대전대학교 컴퓨터공학과 졸업 (학사)  
2004. 2 성균관대학교 대학원 컴퓨터공학과 졸업 (석사)  
2004. 2~현재 (주)삼성전자 연구원  
관심분야: P2P 컴퓨팅, 이동 에이전트, 분산 컴퓨팅 등  
E-mail : door21c@dclab.skku.ac.kr

---

**엄 영 익**



1983. 2 서울대학교 계산통계학과 졸업 (학사)  
1985. 2 서울대학교 대학원 전산과학과 졸업 (석사)  
1991. 8 서울대학교 대학원 전산과학과 졸업 (박사)  
2000. 9~2001. 8 Dept. of Info. and Comm. Science at UCI 방문 교수  
1993~현재 성균관대학교 정보통신공학부 교수  
관심분야: 분산 컴퓨팅, 이동 컴퓨팅, 이동 에이전트, 시스템 보안, 시스템 소프트웨어 등  
E-mail : yieom@ece.skku.ac.kr

---

**• 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>