

# 다중 스패닝 트리를 이용한 효율적인 피어 투 피어 검색 기법

박홍기, 장주욱  
서강대학교 전자공학과  
e-mail : {hongki77, jjang}@sogang.ac.kr

## An Efficient Peer to Peer Search Based on Multiple Spanning Trees

Hong-Ki Park, Ju-Wook Jang  
Dept of Electronic Engineering, So-Gang University

### 요 약

안정적이고 확장성이 좋은 Peer to Peer 시스템에서의 콘텐츠 검색을 위하여 여러 알고리즘이 제안되었다. 그 예로 Chord[1]는 네트워크에 참여하는 각각의 노드에 콘텐츠의 위치 정보를 분배하는 방식을 사용하기 때문에 노드가 콘텐츠의 위치 정보를 다른 노드에 이양하지 않고 네트워크에서 벗어나는 경우 그 노드가 가진 정보가 손실되어 네트워크 확장성에 제한을 받았다. 그리고 Gnutella[2]는 네트워크에 전파되는 query 수가 노드 당 연결 수에 대하여 지수적으로 증가하기 때문에 대역폭을 낭비하는 경향이 있었다. 본 논문에서 제안한 방식은 원하는 콘텐츠를 찾고자 할 때 Broadcast를 사용하여 query를 전파하는 방법을 수정하여 query가 네트워크 토폴로지를 기반으로 만들어진 여러 개의 Spanning Tree를 따라서 Source Routing 방식으로 전파되도록 하였다. 제안한 방식을 적용한 경우, 네트워크 토폴로지에 따라 콘텐츠를 찾기 위한 query의 수를 Broadcast 방식의 16% ~ 3% 까지 줄일 수 있었다.

### 1. 서론

Peer to Peer 시스템에서 콘텐츠를 검색하기 위하여 제안된 알고리즘들은 실제 네트워크에 적용할 때 여러 가지 한계점을 가지고 있다. Chord[1] 알고리즘은 Peer to Peer 시스템에 참여하는 노드들에 콘텐츠의 위치 정보를 분배하기 때문에, 노드가 위치 정보를 다른 노드에게 이양하지 않고 네트워크에서 사라지는 경우, 이 노드가 가진 정보가 네트워크에서 손실되는 결과를 가져오게 된다. Gnutella[2] 알고리즘은 콘텐츠 검색을 위해서 query를 Broadcast 방식으로 전파하기 때문에 Chord보다 안정적인 운영이 가능하나, query의 수가 노드 당 연결 수에 대하여 지수적으로 증가하기 때문에 중복되는 query가 많아져서 Bandwidth를 필요량의 13배~60배 이상 낭비하는 문제점을 가지고 있다. 이 때문에, 인터넷이 IP를 매개로 전 세계를 하나의 네트워크로 통합하고 있음에도 불구하고, 현재 구현된 Peer to Peer 시스템은 콘텐츠 검색과 정보교환이 매우 협소한 지역에

국한되어 있거나, 콘텐츠를 가진 노드와 연결하기 위해서 중앙 서버에 자신의 정보를 등록시키고, 다른 서버를 하나씩 검색해야 하는 추가적인 네트워크 작용을 필요로 하며, Broadcast방식을 사용하지 못하고 Unicast 방식을 사용하고 있어서 검색하는 시스템의 부담이 크다.[6][7] 본 논문에서는 콘텐츠를 찾기 위해 query를 Broadcast 방식으로 보내는 것을 보완하여 Spanning Tree를 따라서 query를 Source Routing 함으로서 네트워크에 전파되는 query의 수를 줄여서 Peer to Peer 시스템의 범위를 확장하고 Multiple Spanning Tree를 사용함으로서 노드의 에러가 발생하더라도 안정적으로 콘텐츠를 검색하는 것이 가능하다. 2장에서는 기존의 Peer to Peer 시스템에서의 query 전파 알고리즘에 대해서 소개하며, 3장에서는 본 논문에서 제안하는 알고리즘에 대해서 설명하고 4장에서 이를 바탕으로 본 제안 방식의 효율성을 실험한 결과를 보이며 5장에서 결론과 추후 과제에 대해서 논한다.

## 2. 기존의 query search 알고리즘

### 2.1 코드 (Chord)

코드(Chord)[1] 알고리즘의 장점은 DHT을 사용하여 콘텐츠의 킷값을 각 노드들에 분배하기 때문에 노드의 수가 많아도  $O(\log N)$ 번의 검색만으로 빠르게 원하는 콘텐츠를 검색할 수 있다.

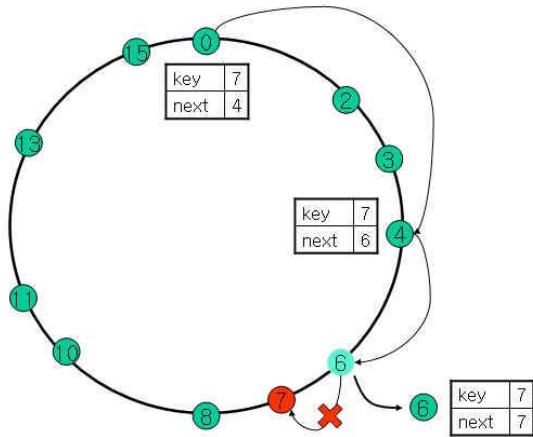


그림 1 노드가 네트워크를 벗어나는 경우

그러나 Chord는 각각의 노드에 콘텐츠의 위치 정보를 분배하여 저장하기 때문에, 노드가 위치 정보를 다른 노드에게 이양하지 않고 네트워크에서 벗어나는 경우, 이 노드가 가지고 있던 정보는 네트워크에서 손실되게 된다. 이러한 문제점은 Chord를 이용한 Peer to Peer 시스템의 확장성을 방해한다.

### 2.2 그누텔라 (Gnutella)

그누텔라(Gnutella)[2]는 네트워크상의 모든 노드에 콘텐츠 검색 query를 전달하는 방식이다. 그렇기 때문에 Chord와 같이 노드에 발생할 수 있는 여러 가지 문제에 크게 영향을 받지 않으며 안정적이다.

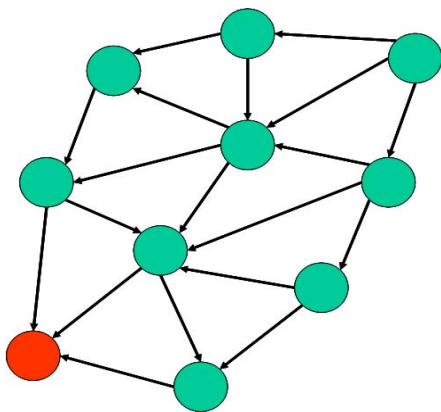


그림 2 그누텔라 query의 flooding

그러나 Gnutella 방식은 콘텐츠를 찾기 위해 query를 전파할 때, 연결된 노드에 모두 query를 전송하며, 각 노드는 query의 결과에 관계없이 TTL의 한계까지 계속해서 query를 전파하기 때문에, 필요한 query수의 12배~60배 이상의 query를 발생시켜 대역폭의 사용하게 된다.

## 3. 개선된 query search 방법

본 논문에서 제안한 방식은 규모가 큰 시스템에서 안정적으로 콘텐츠 검색을 하기 위해서 콘텐츠 검색 시 발생하는 query 수를 최소화하는 데 초점을 맞추었다. 이를 위해서 콘텐츠 검색 query를 미리 구성된 다수의 Spanning Tree를 이용하여 query를 전송하는 방식을 제안하였다.

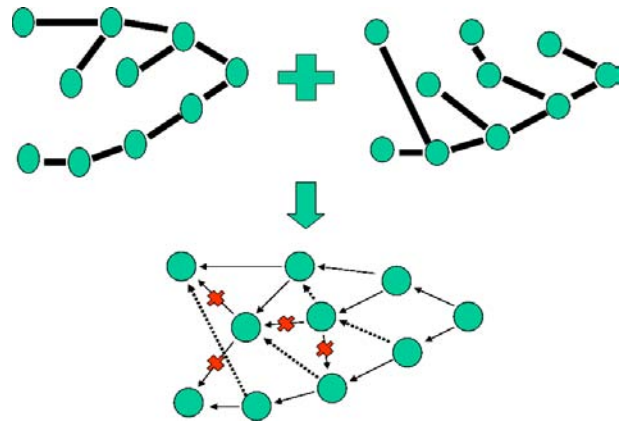


그림 3 제안한 알고리즘의 개념도

Peer to Peer 네트워크에 노드가 가입하게 되면, 각 노드는 전체 네트워크 토폴로지를 기반으로 자신을 Root로 하는 Spanning Tree를 다수 만들게 된다. 그리고 원하는 콘텐츠를 검색할 때, Spanning Tree를 따라서 query를 Source Routing 하게 된다. 위 그림에서 Broadcast 방식과 제안한 방식의 query 수를 비교하면, 콘텐츠를 검색할 때 각 노드에서 만들어내는 query의 수가 줄어들게 됨을 알 수 있다.

이 때, 하나의 Spanning Tree만 사용하여, query를 전달할 경우, 임의의 링크에 문제가 생겼을 때, 검색 query가 모든 노드에 전달되지 않게 된다. 각 링크에서의 Packet Error Rate를 R이라고 하면, 하나의 Spanning Tree를 사용할 경우 검색이 불완전할 확률을 근사한 식은 다음과 같다.

$$\text{식(1)}$$

$$1 - (1 - R)^M$$

여기서 M은 콘텐츠를 가지고 있는 노드에 도달하기까지의 평균 중간 노드 수이다. 그러나 다수의 Spanning Tree를 통하여 query가 전달되는 경우 검

색이 불안전할 확률은 작아지며 근사한 식은 다음과 같다.

식(2)

$$(1 - (1 - R)^M)^N$$

여기서 N은 사용된 Spanning Tree의 수이다. 사용하는 Spanning Tree의 수가 늘어날수록 검색에 실패할 확률은 지수적으로 감소하게 된다. 그러나 Spanning Tree를 유지하기 위한 Overhead는 사용하는 Spanning Tree의 수에 따라서 증가할 것이므로 네트워크의 Packet Drop Rate 특성과 사용 될 Spanning Tree의 수 사이에 적절한 Trade off가 필요할 것이다.

제안한 알고리즘은 Spanning Tree를 사용하여 query를 전송하므로 Broadcast 방식과 비교하여 예상되는 query 수의 절감 근사 비율은 다음과 같다.

식(3)

$$\frac{NS}{(K-1)^M}$$

S는 네트워크를 구성하는 전체 노드의 수이며, K는 노드의 평균 링크 수이다. Spanning Tree방식으로 전달할 경우의 query 수는 전체 네트워크를 구성하는 노드의 수에 좌우되지만, Broadcast의 경우에는, 콘텐츠를 보유한 노드까지의 거리와 노드의 링크수와 연관되어 query 수가 지수적으로 증가하게 된다.

#### 4. 시뮬레이션 결과

시뮬레이션은 네트워크가 유선망으로 구성 되었으며 Packet Error가 발생하지 않는 환경이라는 가정 하에 실시되었다. 그리고 각 노드는 정해진 공간 내에서 임의로 위치가 정해지며 일정 범위 내의 모든 노드와 직접 연결을 맺는다.

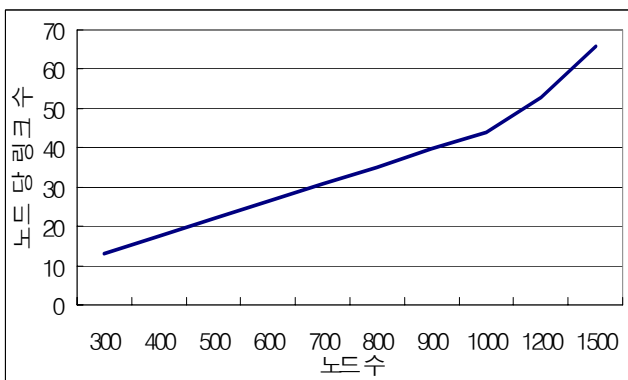


그림 4 노드 간의 연결 수 증가 곡선

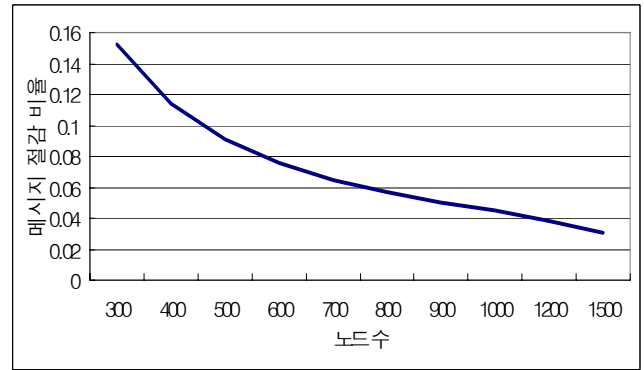


그림 5 그누텔라 대비 query 절감 비율

그림 4, 그림 5는 제안한 알고리즘과 네트워크 토폴로지의 상관관계를 검증하는 실험이다. 고정된 범위 내에서 지속적으로 노드 수를 증가시켜서 노드 사이의 평균 연결 수를 증가시켰을 때, 제안한 방식과 그누텔라 방식의 query 수 증감 비율을 검사하였다. 시뮬레이션 결과(그림5) 그누텔라 방식의 query 증가 수 대비 제안한 알고리즘의 query 증가 수의 비율은 네트워크의 노드 사이의 평균 연결 수가 선형적으로 증가 할수록 지수적으로 낮아짐을 확인할 수 있었다. 이는 앞에서 설명한 바와 같이 노드의 평균 연결 크기가 증가하는 것에 대하여 제안한 알고리즘이 크게 영향 받지 않는 것에 반하여, 그누텔라 방식은 query의 수가 노드의 평균 연결 크기의 증가에 따라 지수적으로 증가하기 때문이다.

위 실험결과는 임의로 생성되는 다양한 네트워크 토폴로지에서의 노드들에 대한 1000회 이상의 반복 실험 결과의 평균값이며, 이러한 네트워크 환경에서, 그누텔라 방식에 비하여 제안한 알고리즘이 16% ~ 3% 정도로 콘텐츠 검색 query가 절감되는 것을 확인하였다.

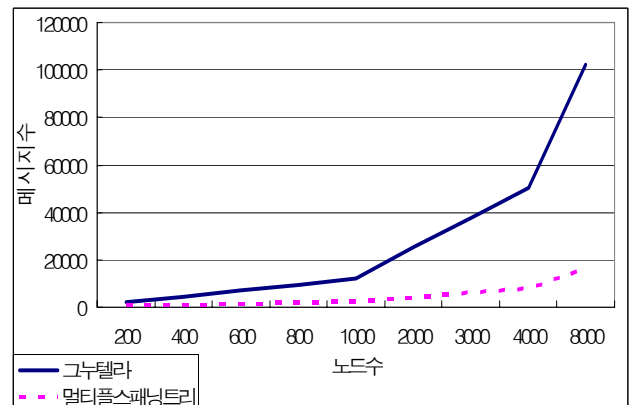


그림 6 노드의 평균 링크 수가 일정한 경우

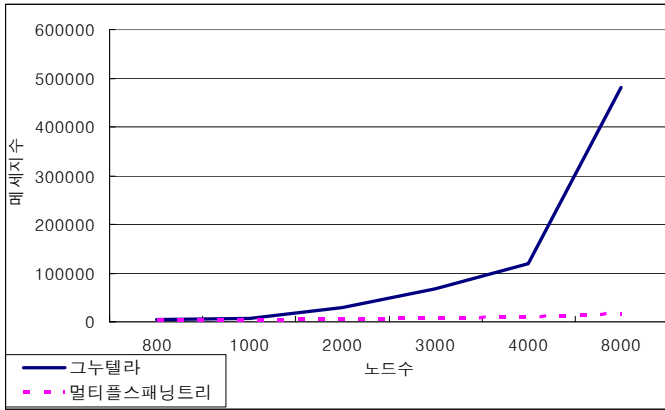


그림 7 일정 범위 내에 노드가 계속 증가하는 경우

그림 6, 그림 7은 노드의 평균 연결수가 고정된 경우와 그렇지 않은 경우에 그누텔라 방식과 제안한 방식의 query의 생성수를 직접 비교한 그래프이다.

앞에서 보인 바와 같이 제안한 방식이 그누텔라 방식으로 query를 전송하는 방식보다 query 수가 절약됨을 알 수 있으며, 노드 간의 연결이 많은 환경일수록 제안한 알고리즘으로 query를 보내는 것이 더 많이 query수를 줄일 수 있음을 알 수 있다.

## 5. 결론 및 추후과제

실험 결과, 제안한 알고리즘은 Broadcast 방식으로 콘텐츠를 검색하는 그누텔라 방식보다 query 수를 네트워크 토폴로지에 따라서 기존의 16% ~ 3%로 줄일 수 있었다. 인터넷과 같이 노드와 노드의 연결이 많은 환경에서, 제안한 알고리즘이 실제 Peer to Peer 콘텐츠의 검색 방식에 적용될 때, 네트워크 대역폭의 절감에 크게 기여할 것으로 예상된다.

추후 과제로, 제안한 방식이 무선 환경에서와 같이 노드의 이동이나 통신 매체의 특성에 의해 링크 에러율이 큰 환경에서도 우수한 성능 결과를 나타내는지 추가 실험할 것이며 MANET에의 적합성에 대한 여부를 추가 검증할 것이다. 이는 Adhoc을 기반으로 하는 Peer to Peer 시스템에서의 응용에도 도움이 될 것으로 생각된다.

## 참고문헌

- [1] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari, Balakrishnant, Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications, SIGCOMM 2001
- [2] CLIP2, Gnutella Protocol Specification v0.4,

Documentaion Revision 1.2

<http://www.clip2.com>

[3] Sylvia Ratnasamy, Paul Francis, Mark handley, Richard Karp, Scott Shenker, A Scalable, Content-Addressable Network, SIGCOMM 2001

[4] Jinyang Li, John Jannotti, Douglas S.J. Decouto David R.Karger, Robert Morris, A Scalable Location Service for Geographic Ad Hoc Routing, SIGCOMM 2001

[5] Stefan Saroiu, P.Krishna,Gummadi, Steven D,Gribble, A Measurement Study fo Peer to Peer File Sharing System, Univ. of washington technical report, 2002

[6] eDonkey2000, network file sharing system

<http://www.edonkey2000.com>

[7] fileguri, network file sharing system

<http://www.fileguri.com>