

모바일 환경을 위한 P2P 검색 방법

이세연 장주욱
서강대학교

seiyonl@eeca1.sogang.ac.kr jjang@mail.sogang.ac.kr

Mobile Peer-to-Peer Network: Query Search in Mobile Network

Seiyon Lee Juwook Jang
Sogang University Electronic Engineering Dept.

요 약

Chord는 P2P 네트워크에 n 개의 클라이언트가 있을 때 $O(\log N)$ 의 상태 저장 메모리 용량과 $O(\log N)$ 의 검색 단계가 걸리는 분산형 검색 방법이다. 하지만 모바일 네트워크와 같이 P2P 네트워크를 구성하는 클라이언트의 신뢰성을 보장하지 못하는 네트워크에서는 검색 성능이 현저히 떨어진다. 따라서 본 논문에서는 이러한 Chord의 단점을 보완하여 모바일 네트워크에서도 P2P 검색이 원활하게 이루어질 수 있는 새로운 알고리즘을 제안하고 제안한 검색 방식의 실행 및 결과를 보여주고자 한다.

1. 서 론

P2P 네트워크에서 자료 검색 방식으로는 Napster[12], Gnutella[8], Chord[1]등과 같은 방식을 들 수 있다. Napster는 중앙 집중형 검색 방식으로 $O(1)$ 의 검색 메시지를 보내면 검색을 할 수 있는 방식이다. 즉 중앙 검색 서버에 접속하는 동시에 자료가 있는 위치를 알 수 있는 방식인 것이다. 하지만 이 경우에는 중앙 서버의 신뢰도가 중요하게 된다. 따라서 P2P 네트워크의 본래의 형태인 분산형, 수평적 네트워크의 구성이 어려울 뿐만 아니라 중앙 검색 서버에 $O(N)$ 의 대량의 검색 목록을 저장하는 메모리가 소요되게 된다. 이에 반해 Gnutella[8]는 Napster[12]와는 달리 중앙 검색 서버를 전혀 두지 않고 Client들만으로 구성되었을 때 검색을 가능하게 하는 수평적 네트워크에서의 검색 방식이다. 이 경우에는 $O(N)$ 의 다수의 요청 메시지를 보낸다. 즉 모든 구성 클라이언트들을 검색해봄으로써 원하는 자료의 위치를 파악하는 방식인 것이다. 이 경우에는 중앙 검색 서버가 필요하지 않고 각 구성 클라이언트들의 상태를 저장하는 메모리 용량이 $O(1)$ 으로 적게 들기 때문에 P2P 네트워크 확장성을 보장할 수 있다. 하지만 검색 시간이 지나치게 오래 걸린다. 따라서 이러한 점을 보완하면서 분산형 네트워크에

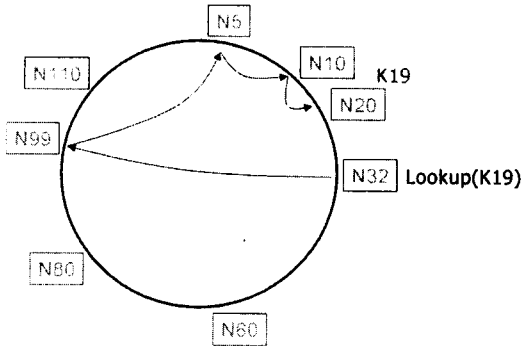
서 P2P 네트워크를 구성하기 위해 나온 검색 방식이 Chord[1]이다. Chord에서는 $O(\log N)$ 의 검색 메시지를 통해서 분산형 P2P 네트워크에서도 빠른 검색 시간을 보장해준다.[1,11]

2.1 Chord [1]

Chord는 Napster와 같은 중앙 집중식 P2P 방식이 아니라 분산형 P2P Network 방식이다. 분산형 P2P 검색 방식의 대표적인 예로는 Gnutella방식을 들 수 있다. Gnutella방식의 경우는 Napster방식처럼 데이터의 인덱스 값을 중앙 검색 서버에 두는 것과는 달리 각각의 클라이언트들이 일정한 해쉬 테이블을 가지고 테이블을 꾸준히 업데이트 하고 거기에 있는 내용을 직접 검색하는 방식을 취하고 있다. 따라서 Gnutella의 경우에는 P2P 네트워크를 이루고 있는 클라이언트의 수와 관계없이 P2P 네트워크에 접속해 있는 모든 클라이언트의 데이터를 검색할 수 있다.[8,11] 하지만 이러한 방식을 사용할 경우에는 검색에 많은 시간이 걸리게 된다. 이러한 문제점을 해결하기 위해서 고안된 검색 방식이 Chord 방식이다. Chord방식의 경우에는 모든 클라이언트를 다 검색하는 Gnutella[8,11]방식과는 달리 Successor라는 것을 Successor의 Hash Table(이 해쉬 테이블에는 자

료 인덱스 정보가 들어있다.)에 근접 노드의 자료 정보들을 가지고 있음으로써 Successor에서 검색을 함으로써 보다 빠른 시간에 검색을 할 수 있는 방식이다.[2]

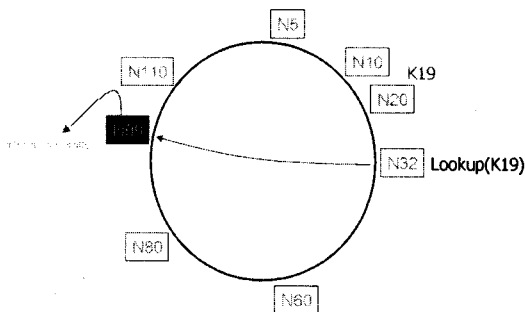
(그림 1 참조)



[그림 1] Chord 검색 방식

Successor들은 Hash Function에 의해서 선출된다. 또한 Successor들은 모든 노드의 정보를 가지고 있는 것이 아니라 근접한 노드의 정보만을 가지고 있다. 따라서 중앙 집중형 검색 방식처럼 많은 정보를 저장하고 있는 것이 아니어서 클라이언트에 많은 부담을 주지 않는다. 따라서 자료를 찾으려하는 클라이언트는 몇 개의 Successor만을 검색하면 원하는 자료를 찾을 수 있게 된다. 따라서 Chord[1]에서는 Napster[12]보다 작은 $O(\log N)$ 의 상태 저장 메모리 용량이 필요하게 됨으로써 분산형 P2P 네트워크를 구성하는데 많은 도움을 줄 수 있고 검색 메시지는 $O(\log N)$ 으로 Gnutella보다는 줄어들게 됨으로써 검색 시간이 Gnutella보다 단축되게 된다.

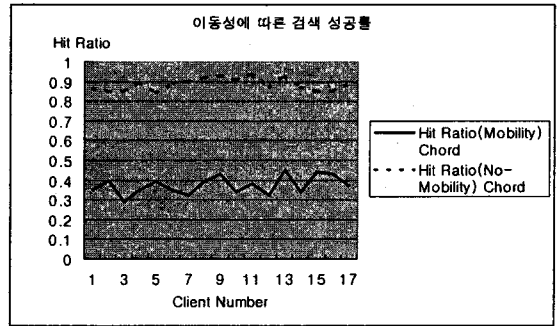
2.2. Chord의 문제점



[그림 2] Chord 쿼리 검색 실패

Chord 쿼리 검색 방식의 문제점은 모바일 노드에서의 운동성이 보장되지 않는다는 점이다. 이는 모바일 노드의 특성상 갑자기 노드가 통신 가능 영역에서 벗어날 수 있다. 또한 전파의 특성상 기지국 간의 Hand-off현상의 발생으로 일시적으로 Node가 없어지는 현상이 생길 수 있다. 이 경우에 다음 노드로 연결해줄 수 있는 연결 고리를 잃어버리기 때문에 더 이상의 검색이 이루어지지 않는다.(그림 2 참조) 그래프 1-1의 경우에서 볼 수 있듯이 검색 성공률이 현저하게 떨어지는 것을 통해서 알 수 있다.

그래프 1-1 결과에서 볼 수 있듯이 Chord 검색 방식은 검색 방법 시간 단축에는 효과를 줄 수 있으나 이동성을 띤 무선 네트워크에서는 검색 자체가 불가능하게 되어서 검색 성공률이 현저하게 떨어지는 것을 볼 수 있다.

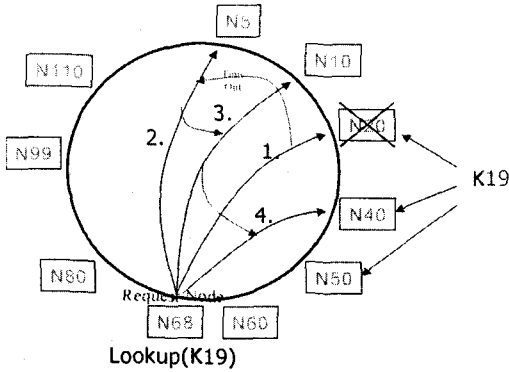


[그래프 1-1] 이동성에 따른 쿼리 검색 성공률 비교: 이동성에 따른 랜덤 Seed를 1로 준다.

따라서 Chord가 분산형 검색 시스템의 여러 가지 장점을 가지고 있음에도 불구하고 모바일 환경에서는 검색 자체가 잘 이루어지지 않는 치명적인 약점을 가지고 있다.

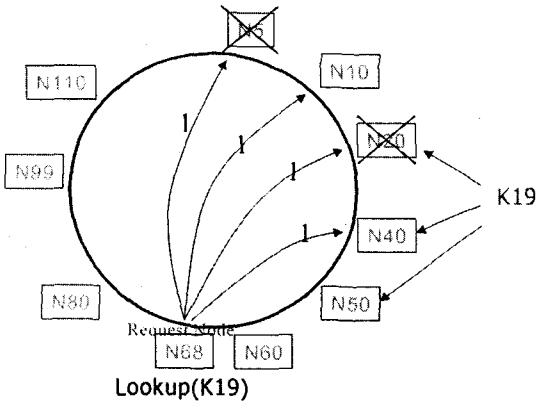
3. 제안된 방식

이와 같은 Chord의 약점을 보완하고 모바일 환경에서도 Chord를 적용시키기 위해서 수정한 것이 제안된 방식이다. 제안된 방식의 경우에는 일정한 시간(Time-out)을 설정해두어서 Successor가 갑자기 사라지더라도 검색을 멈추지 않고 새로운 Successor로 검색 요청을 하도록 한다.(그림 3 참조)



[그림 3] 제안된 방식에서 Time-out에 대한 동작

또한 과도한 Time-out값을 설정해두면 많은 검색 시간이 걸리기 때문에 보다 검색 시간을 단축하기 위해서 Time-out값 설정과 동시에 다중의 Successor들에게 검색을 요청함으로써 검색 시간을 단축시킨다.
(그림 4 참조)

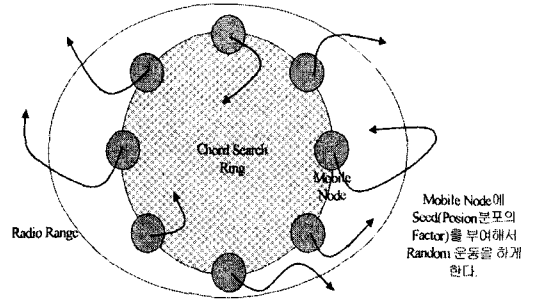


[그림 4] 제안된 방식에서 다중 검색 요청에 따른 동작

4.1 실험 환경

NS-2 Simulator 환경에서 1000개의 임의의 모바일 클라이언트들을 설정하고 그중에서 무작위로 일부 클라이언트들을 On-Off 시킨다. 이러한 동작은 일종의 전파 영역에서 벗어나를 때를 Off시키는 것으로 다시 전파 영역에 있을 때를 On시키는 것으로 동작 시킴으로써 모바일 네트워크에서의 동작과 비슷하게 움직이도록 환경 설정을 했다. 또한 1000개의 노드 중 검색을 빈번하게 하는 자료를 가지고 있는 모바

일 클라이언트 15개를 선정해 두었다. (그림 5 참조)



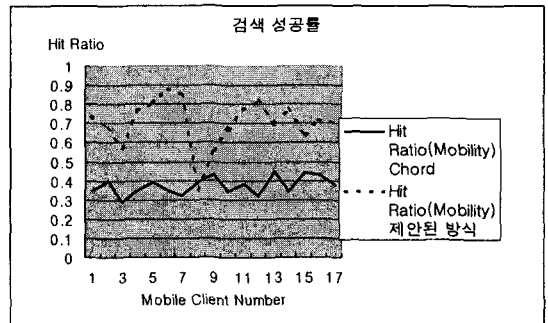
[그림 5] 실험 방법

4.2 실험 방법

설정된 모바일 네트워크 환경에서의 Chord 동작 결과와 안정한 네트워크 환경에서의 Chord 동작 결과를 비교 분석하였다. 또한 모바일 네트워크에서의 제안된 방식의 알고리즘을 적용한 실험 결과를 분석하였다. 더불어서 모바일 환경에서 Gnutella 동작 결과를 분석하였다.

4.3 실험 결과

그래프 2-1과 같이 검색 성공률이 보통의 Chord 방식보다 많은 향상을 보이고 있다.



[그래프 2-1] Redundancy Chord와 Chord의 비교 분석(무선 네트워크 환경)

그래프 2-1에서 제안된 방식이 불안정한 값을 가지게 되는데 이는 제안된 방식에서 Time-out값과 다중 검색 요청 횟수의 Trade-off를 자동으로 이루어지게 하였기 때문에 불안정한 값을 가지게 되는 것이다. 특히 클라이언트 8번의 경우에 기존의 Chord

보다 낮은 성공률을 보이게 되는데 이 경우는 8번 자료 요청이 지나치게 많이 생성되었기 때문에 이와 같은 결과가 발생하였다.

	Gnutella	Redundancy Chord
Node 0	0.64	0.23
Node 1	0.77	0.5
Node 2	0.67	0.23
Node 3	0.57	0.56
Node 4	0.88	0.34
Node 5	0.46	0.35
Node 6	0.9	0.46
Node 7	0.34	0.33
Node 8	0.67	0.31
Node 9	0.58	0.8
Node 10	0.68	0.23
Node 11	0.71	0.56
Node 12	0.8	0.46
Node 13	0.56	0.37
Node 14	0.64	0.29
Node 15	0.37	0.36
Average	0.64	0.39875

표 1. 자료 검색에 걸리는 시간 비교

Gnutella와 제안된 방식간의 검색 시간 비교에서는 표 1에서와 같이 평균 검색 시간이 모든 클라이언트를 다 검색해보는 Gnutella방식보다 현저하게 빠른 검색 시간을 제안된 방식이 보여주고 있다.

5. 결론

본 논문에서 제안한 방식의 경우 어떠한 네트워크 환경에서도 검색 성공률이 평균 70.4%의 검색 성공을 보임으로 Chord가 불안정한 네트워크에서 보이는 검색 성공률(평균 37.4%)보다 탁월한 검색 성공을 보였다. 또한 모든 클라이언트를 검색하는 방식인 Gnutella 검색 방식(검색 성공률 평균 82.4%)과 비교 했을 때 검색 성공률은 약간 떨어졌지만 검색 시간에서는 많은 단축을 보였기 때문에 본래의 Chord의 장점을 잃지 않고 모바일 환경에 잘 적용될 수 있었다.

6. 앞으로 할 일

Time-out값과 다중 검색 횟수에 대한 최적의 Trade-off값을 정함으로써 모든 클라이언트에서 일정하고 높은 성공률을 보이게 하려고 한다.

[참고 문헌]

[1] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications MIT Laboratory for Computer Science.(ACM Conf.2002)

[2] CHEN, Y., EDLER, J., GOLDBERG, A., GOTTLIEB, A., SOBTI, S., AND YIANILLOS, P. A prototype implementation of archival intermemory. In *Proceedings of the 4th ACM Conference on Digital Libraries* (Berkeley, CA, Aug. 1999), pp. 28-37.

[3] CLARKE, I. A distributed decentralised information storage and retrieval system. Master's thesis, University of Edinburgh, 1999.

[4] CLARKE, I., SANDBERG, O., WILEY, B., AND HONG, T.W. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability* (Berkeley, California, June 2000). <http://freenet.sourceforge.net>.

[5] DABEK, F., BRUNSKILL, E., KAASHOEK, M. F., KARGER, D., MORRIS, R., STOICA, I., AND BALAKRISHNAN, H. Building peer-to-peer systems with Chord, a distributed location service. In *Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems (HotOS-VIII)* (Elmau/Oberbayern, Germany, May 2001), pp. 71-76.

[6] DABEK, F., KAASHOEK, M. F., KARGER, D., MORRIS, R., AND STOICA, I. Wide-area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)* (To appear; Banff, Canada, Oct. 2001).

[7] DRUSCHEL, P., AND ROWSTRON, A. Past: Persistent and anonymous storage in a peer-to-peer networking environment. In *Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems (HotOS 2001)* (Elmau/Oberbayern, Germany, May 2001), pp. 65-70.

[8] Gnutella. <http://gnutella.wego.com/>.

[9] KUBIATOWICZ, J., BINDEL, D., CHEN, Y., CZERWINSKI, S., EATON, P., GEELS, D., GUMMADI, R., RHEA, S., WEATHERSPOON, H., WEIMER, W., WELLS, C., AND ZHAO, B. OceanStore: An architecture for global-scale persistent storage. In *Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)* (Boston, MA, November 2000), pp. 190-201.

[10] LEWIN, D. Consistent hashing and random trees: Algorithms for caching in distributed networks. Master's thesis, Department of EECS, MIT, 1998. Available at the MIT Library, <http://thesis.mit.edu/>.

[11] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. Tech. Rep. TR-819, MIT LCS, March 2001. <http://www.pdos.lcs.mit.edu/chord/papers/>.

[12] Napster. <http://www.napster.com/>.