

모바일 P2P 환경에서 이동 노드 주소 변경 전파를 통한 효율적 Overlay 망 유지 기법

김동욱, 이어형, 홍중표, 김신덕
연세대학교 컴퓨터과학과

e-mail : dwkim@parallel.yonsei.ac.kr, liquid@parallel.yonsei.ac.kr, hulkboy@yonsei.ac.kr,
sdkim@yonsei.ac.kr

An Effective Scheme for Managing Overlay Networks based on Propagation of Network Address Change of Node in a Mobile P2P Environment

Dong-Wook Kim, Eo-Hyung Lee, Chung-Pyo Hong, and Shin-Dug Kim
Dept. of Computer Science, Yonsei University

요 약

최근 유비쿼터스 및 모바일 환경에서 Peer-to-Peer(P2P)시스템에 대한 연구들이 활발히 진행되고 있다. 모바일 환경에서는 모바일 환경의 제약사항과 함께 이동성이라는 특징이 중요하게 고려된다. 노드의 이동에 의해서 네트워크의 주소가 변경되어 P2P 네트워크 망에서 유효하지 않은 접속 정보들이 생기고 이러한 정보들이 검색 및 P2P 시스템의 전체적인 성능을 떨어뜨리게 된다. 이 논문은 이러한 접속 정보들의 불일치를 해결하여 접속 정보를 유지하기 위한 효과적인 방법론인 이동전파 알고리즘을 제안한다. 이 알고리즘은 접속 정보의 불일치를 해결하기 위해 정보를 빠르게 처리해서 P2P 시스템의 성능을 향상하는 것을 목표로 한다. DHT 기반 P2P 시스템을 기반으로 하여 제안하는 알고리즘을 적용하여 실험을 하였으며, 그 결과로 처음 신규 발견된 정보를 재 발견하는 경우에 성능이 기존의 DHT 기반의 P2P 시스템보다 80% 향상되었음을 알 수 있다.

1. 서론

최근 다양한 Peer-to-Peer(P2P) 관련 연구들이 진행되고 있으며, 유비쿼터스(ubiquitous) 및 모바일 환경에서 P2P 시스템을 활용하는 방법에 관한 연구들도 활발히 진행되고 있다.

일반적으로 계층적 P2P 시스템은 상위 네트워크를 구성하는 슈퍼노드(super-node)와 하위 네트워크를 구성하는 일반노드(normal-node)로 구성되어 있으며, 일반노드들은 슈퍼노드와의 통신을 통해서 전체 P2P 네트워크에 참여하게 된다.

대표적인 DHT(Distributed Hash Table)기반의 계층적 P2P 시스템은 Chord[1], CAN[2] 그리고 JXTA[3][4]를 들 수 있다. 이러한 P2P 시스템들은 DHT의 특징을 활용하여 자원 및 전체적인 부하를 노드들 사이에 균등하게 배분한다.

하지만 모바일 환경의 P2P 시스템에서는 모바일 환경의 제약사항들(제한된 컴퓨팅 파워, 작은 유저 인터페이스, 제한된 파워 서플라이 그리고 제한된 대역폭)과 함께 이동성이라는 중요한 특징을 고려해야 한다. 노드의 이동에 의해서 전체 P2P 네트워크에는 유효하지 않은 접속 정보들이 존재하게 된다. 즉, 노드의 네트워크 주소가 변경되어 불일치된 접속 정보들이 생겨나게 된다. 이러한 정보들은 검색 및 전체 P2P 시

스템의 성능을 떨어뜨린다.

본 논문에서는 노드의 이동에 의해서 발생할 수 있는 접속정보의 불일치성을 해결하기 위해 이러한 정보를 빠르게 처리해서 전체적인 P2P 시스템의 성능향상을 목표로 한다. 이를 위해서 DHT 기반의 계층적 P2P 특징을 활용하고 본 논문의 핵심인 이동전파(move-propagation) 알고리즘을 제안한다. 앞에서 간략히 설명한 DHT 기반의 P2P 시스템은 본 논문에서 제안하는 P2P 시스템의 기본 골격으로 활용된다. 본 논문의 구성은 다음과 같다. 2 장에서는 관련연구를 조사하고, 3 장에서는 제안하는 P2P 시스템을 기술하며, 4 장에서는 실험결과를 제시하고, 마지막으로 5 장에서 결론을 맺는다.

2. 관련 연구

앞서 언급한 것처럼 DHT 기반의 P2P 시스템들은 자원이나 서비스를 전체 P2P 네트워크상 노드들 사이에 균등하게 배분하는 장점을 가지고 있다. P2P 네트워크는 네트워크에 참여하는 모든 노드들이 기능을 분담하는 분산방식으로 서비스를 제공하고 있으므로, 네트워크의 일부 노드들이 손실되거나 장애가 발생해도 전체 시스템의 기능이나 데이터는 손실되지 않기 때문이다. 그러므로 모바일 환경에서도 이러한 장점

을 유지하기 위해서는 노드의 이동에 따른 처리가 필수라고 할 수 있다. 본 장에서는 제안하는 P2P 시스템의 기반이 되는 DHT 기반 P2P 시스템을 설명한다.

대표적인 DHT 기반의 P2P 시스템인 JXTA 는 두 단계의 층으로 구분되는 계층적 P2P Overlay 네트워크로 구성되어있다. 상위 층은 랑데부 피어(Rendezvous Peer)로 이루어져 있고, 하위 층은 에지 피어(edge Peer)로 이루어져 있다. 이는 각각, 슈퍼노드와 일반노드에 해당한다. 랑데부 피어는 에지 피어에 비해서 일반적으로 성능이 우수한 노드를 의미하고 에지 피어로부터 오는 쿼리(Query)를 처리한다.

또한, DHT 를 활용해서 자원이나 서비스의 광고(Advertisement)의 인덱스를 유지한다. 검색과정을 간략히 기술하면, 우선 에지 피어가 검색 쿼리를 랑데부 피어에게 전달한다. 이를 수신한 랑데부 피어는 DHT 를 활용해서 인덱스 랑데부 피어에게 전파하고, 해당 랑데부 피어는 자원을 소유한 에지 피어가 연결되어 있는 랑데부 피어에게 다시 전파한다. 이를 수신한 랑데부 피어는 자원 소유자인 에지 피어에게 다시 전파해 줌으로써 검색과정이 끝난다. 그러나, 모바일 환경에서 이러한 검색과정이 제대로 이루어지기 위해서는 노드의 이동으로 인한 접속 정보의 불일치를 해결하여 접속 정보를 유지 해야 한다.

3. 제안하는 P2P 시스템

모바일 환경에서 지리적 혹은 의미적으로 가까운 노드들은 서로간의 목적에 의해서 빈번한 재 접근이 발생한다. 노드들의 이동에 따라서 새로운 네트워크 topology 가 발생하고, 이 새로운 topology 가 발생함에 따라 물리적 한계에 의한 세션 단절이 발생한다.

본 장에서는 P2P Overlay 망이 노드의 이동에 의해서 생긴 접속 정보를 불일치를 해결하기 위한 방법으로 이동전파 알고리즘을 제안한다. 노드의 이동이 정확하게 전해져야 하고 Overlay 망의 노드들의 정보들이 정확해야 하기 때문에 노드의 이동 정보는 정해진 루틴을 따라 저장, 전달 및 관리 되어야 한다. 노드의 이동에 대한 정보를 관리하는 것은 그 정보를 망에 반영하는 목적을 달성하기 위한 중요한 기능이라고 볼 수 있다.

이동전파 알고리즘에서는 노드의 이동 접속정보를 관리하기 위해 검색 히스토리 Data 구조를 이용한다. 검색 히스토리 Data 구조는 표 1, 표 2에 보여지는 구조를 사용한다. 그럼 먼저 검색 히스토리 Data 에 대해서 살펴보면, 검색 히스토리 Data 의 관리는 이동 접속정보를 유지 하고 관리하는 곳에 사용된다. 검색 히스토리는 overlay 망 내의 각 Peer 들 중에 발견을 요청한 Peer 들의 목록을 담은 자료구조이다. 검색 히스토리 Data 의 구조는 Peer 와 Super Peer 의 경우가 다르다. 왜냐하면 Peer 에서는 자신이 검색 한 정보만 기억했다가 변경하면 되고, Super Peer 는 자식 Peer 들이나 다른 Super Peer 들의 요청에 의한 중계자 역할을 수행할 수 있기 때문에 Peer 의 검색 히스토리에 추가된 정보를 사용한다. 따라서 Peer 와 Super Peer 의 두 가지 경우를 나눠서 그 내용을 아래에 기술한다.

표 1. Super Peer 의 검색 히스토리 Data 구조

Requestor			Destination			
Name	IP	Port	Name	IP	Port	Count
...

표 2. Peer 의 검색 히스토리 Data 구조

Name	IP	Port	Count
...

● Case 1: Super Peer

Super Peer 의 경우에는 Peer 의 요청한 Query 를 보고 검색 히스토리에 추가하게 된다. 첫 번째로 자신의 자식 Peer 가 Query 를 요청한 경우에는 요청한 Peer 의 정보를 Requester 부분에, 목적으로 하는 Peer 의 정보를 Destination 부분에 채워서 추가 한다. 이때 초기 Count 의 값은 0 으로 설정한다. 두 번째로 자식 Peer 가 아닌 다른 Super Peer 로부터의 Query 가 요청되어서 자식 Peer 를 검색 하거나 다른 Super Peer 로 전해져야 하는 경우에도 위와 같은 방법으로 Requestor 와 Destination 의 대상을 설정해서 검색 히스토리에 추가 하고 이때에도 Count 의 값은 0 으로 한다. Data 구조는 표 1에 나와 있다.

● Case2: Peer

Peer 에서는 자신이 Query 를 요청해서 성공을 하면 그 Query 에 대해서 정보를 표 2 의 검색 히스토리 Data 구조의 각각의 항목에 추가한다. Count 의 초기값은 0 으로 설정한다. Super Peer 에 비해서 간단하게 검색 히스토리를 유지 할 수 있다. 왜냐하면 자신이 Query 를 요청해서 성공한 Query 에 대해서만 그 정보를 유지하고 관리하면 되기 때문이다. 그러므로 표 2 와 같이 Super Peer 의 구조보다는 더 작은 구조를 가지게 된다.

표 1, 표 2의 검색 히스토리 Data 구조 중 Count 항목을 활용하여 검색 히스토리 수를 일정하게 유지 하게 된다. 검색 히스토리에 저장되는 항목들은 중복 될 수 있는 상황이 일어날 수 있는데 다른 Peer 들이 같은 자원에 대해서 요청을 했다는 것을 의미한다. 따라서 Count 의 값을 증가시켜서 자원에 대해 얼마나 많은 요청이 있었는지 즉, 요청에 대한 빈도를 표시할 수 있다. 검색 히스토리는 요청된 시간 순서에 따라 정렬을 통해 유지되기 때문에 가장 최근에 요청된 항목에 대해서 가중치를 부여할 수 있게 된다.

검색 히스토리의 수가 일정한 개수에 도달하게 되면 가장 오래된 정보는 1 점, 가장 최근 정보는 전체 개수만큼의 점수를 부여하고,

$$\text{점수} = \text{Count} * \frac{\text{전체개수} * 3}{\text{가장 큰 Count}} \quad (1)$$

식(1)의 점수를 더해서 최종 점수를 정한다. 그 다음 검색 히스토리 중 작은 쪽 절반을 제거한다. 위와 같은 정리방법을 통하여 Peer 와 Super Peer 의 검색 히스토리는 일정개수를 유지하게 된다.

표 3. 검색 히스토리 예

Name	IP	Port	Count
simulation_peer_0001	165.132.121.xxx	9002	9
simulation_peer_0005	165.132.121.xxx	9002	4
simulation_peer_0006	165.132.121.xxx	9002	1

예를 들어 표 3의 simulation_peer_0001 인 Peer 가 최근 정보라고 하면 전체 개수인 3 점을 부여한다. 그리고, 식(1)을 이용하여 Count 값인 9 에 ((3*3)/9)를 곱하면 점수가 9 가 된다. 이 점수에 부여된 3 점을 더해 최종 점수는 12 점이 된다. 이렇게 점수를 계산한 후 검색 히스토리 중 작은 쪽의 절반을 제거한다.

위에서 설명한 검색 히스토리 Data 를 이용해서 노드가 이동을 하였을 때 이동 전과 알고리즘을 살펴보면 검색 히스토리와 같이 Peer 와 Super Peer 의 두 가지의 경우로 나누어진다.

Peer 가 이동을 하면 이동 통보 Message 를 수신한다. 먼저 수신 받은 노드가 Peer 인 경우에는 이동 Message 를 수신하면 자신의 히스토리를 검색한다. 검색을 해서 해당하는 항목이 존재하는지 체크를 하여 존재 유무에 따른다. 해당 항목이 존재 할 경우에는 검색 히스토리에서 항목을 찾아 갱신을 한다.

다음은 수신 받은 노드가 Super Peer 일 경우를 알아본다. 이동 통보 Message 를 수신하면 그 Message 를 전파하게 된다. 이동된 Peer 의 이전 Super Peer 는 자식 목록의 이동 Peer 의 해당 항목을 제거하게 된다. 그리고 이동된 Peer 의 Super Peer 는 히스토리를 검색한다. 검색을 해서 해당하는 항목이 존재하는지 체크 하여 존재 유무에 따른다. 해당 항목이 존재 할 경우에는 검색 히스토리에서 항목을 찾아 갱신을 한다.

아래의 그림에 이동전과 알고리즘을 간략하게 도식화하여 보여준다.

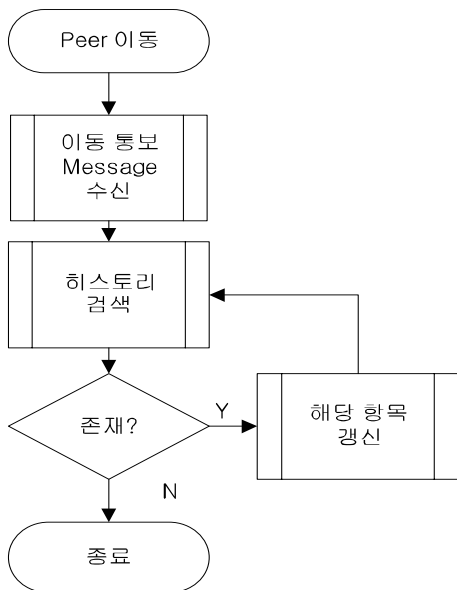


그림 1. 이동전과 알고리즘 도식 (Peer)

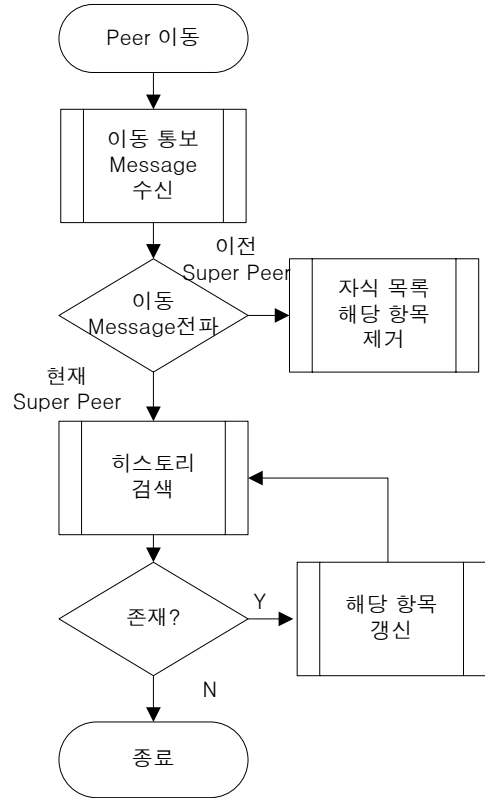


그림 2. 이동 전과 알고리즘 도식 (Super Peer)

4. 실험 결과

이동전과는 성능을 측정하기 어려운 항목이다. 왜냐하면 Overlay 망의 구성과 노드의 이동 환경과 상황에 따라서 성능이 다르게 검출되기 때문이다. 따라서 본 장에서의 실험은 각각의 노드가 이동하는 도중의 한 시점에서의 성능을 보여준다.

실험은 전형적인 JXTA DHT 기반 P2P SYTEM 에 알고리즘을 적용하여 기본 JXTA 와 알고리즘이 적용된 JXTA 를 비교하였다.

실험의 환경은 표 4에 기술된 대로 구성을 하였다. 실험의 결과로 그림 3 과 같은 성능의 그래프를 얻을 수 있었다.

그림 3 의 그래프는 새로운 자원을 발견한 경우와 기존에 발견된 자원을 재 발견하고자 하는 상황을 비교하여 성능을 보여준다. 제일 왼쪽의 그래프는 목적 노드가 이동중인 상황에서 각 노드들이 목적 노드를 신규로 발견하려고 할 때 발견 성공률을 보여준다. 가운데 그래프는 신규 발견된 노드를 각 노드들이 재 발견 하고자 할 때 발견 성공률을 보여준다. 마지막으로 제일 오른쪽의 그래프는 앞의 2 개의 성공률을 합하여 평균적인 발견 성공률을 보여준다.

그림 3 의 그래프에서 보면 알 수 있듯이 재발견의 경우에 성능이 기존보다 80% 향상됨을 알 수 있다. 이렇듯 노드의 접속 정보는 항상 최적의 상태로 유지 된다는 것을 알 수 있다. 목적 노드가 이동할 경우 자신을 발견했던 노드들에게 자신이 이동한 정보를 전달하여 주기 때문이다.

표 4. 실험환경

종류	내용
망 구성	RDV16, Edge 32
이동	32 번(20 초당 1 회)
발견	초기 31 번, 재발견 72 번(4 초당 1 회)
이동과 발견 사이의 휴식	5 분

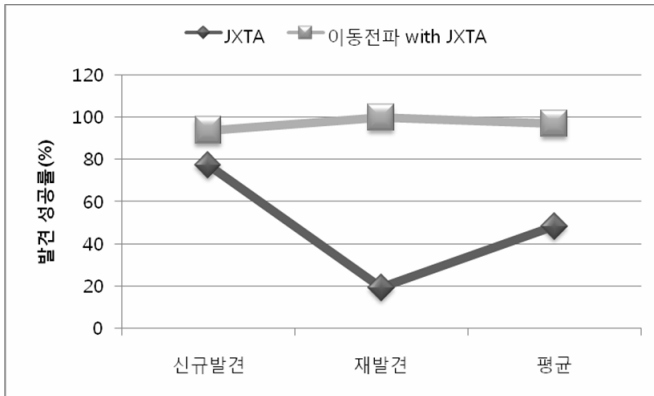


그림 3. 발견 성공률

5. 결론

모바일 환경에서 노드의 이동에 의해서 네트워크 주소가 변경되므로 인해 P2P 네트워크 망에서 유효하지 않은 접속 정보들이 생긴다. 이러한 정보들이 P2P 시스템의 전체적인 성능을 떨어뜨리게 된다. 유효하지 않은 접속 정보들의 불일치를 해결하여 Overlay 망을 유지하기 위해 이동전파라는 알고리즘을 제안했다. 이 알고리즘은 노드 정보의 정확하고 빠른 갱신을 통해서 재 접근성의 향상을 가져왔고 이로 인해서 안정된 망 구성이 이루어진다. 이를 통해 이 논문에서는 제안된 이동 전파 알고리즘을 사용하여 노드 정보를 정확하게 갱신하고 노드의 이동을 인식하지 못해서 발생할 수 있는 불필요한 네트워크 사용을 최소화함으로써, 재 발견 시에 80% 성능 향상을 보여준다. 이러한 향상은 앞으로의 모바일 환경에서의 안정된 P2P 시스템의 성능에 기여할 수 있다.

참고문헌

- [1] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," In Proceedings of SIGCOMM, Aug. 2001, pp. 149-160.
- [2] S.Ratnasamy, P. Francis, M. Handley, R.Karp, and S. Schenker, "A Scalable Content-Addressable Network," In Proceedings of SIGCOMM, Aug. 2001, pp. 161-172.
- [3] Project JXTA: Java™ Programmer's Guide. Sun Microsystems, Inc. https://jxtaguide.dev.java.net/source/browse/*checkout*/jxtaguide/trunk/src/guide_v2.5/JXSE_ProgGuide_v2.5.pdf
- [4] The JXTA-C/C++ Project. <https://jxta-c.dev.java.net/>