

모바일 Ad hoc 네트워크에서의 P2P 네트워크 구축

서강대학교 이세연* · 장주욱**
SK 텔레콤 조태경

1. 서 론

본 고에서는 모바일 Ad-hoc 네트워크 위에 P2P 네트워크 시스템을 구축하는 기법들을 비교, 분석하고 검색의 효율성과 신뢰성 사이 상호 보완에서 Trade-off가 가능한 알고리즘을 제안한다.

모바일 Ad hoc 네트워크(MANET)는 셀룰러 기지국망과 같은 기반 네트워크가 존재하지 않거나 이의 설치가 용이하지 않은 지역에서 이동 단말들이 자율적이고 임시적으로 구성하는 네트워킹 기술이다. 또한 필요에 따라서는 고정된 기반 네트워크와는 독립적으로 이동 단말들만으로 네트워크를 구성할 수도 있다.

P2P 네트워크는 인터넷 연결된 Join/Leave가 빈번한 노드들로 이루어져 있다. 모바일 Ad hoc 네트워크는 multi-hop 무선 연결을 사용하여(Ad hoc routing) 기동성 있는 노드를 네트워크로 구성한 것이다. P2P 네트워크와 모바일 Ad hoc 네트워크는 두 가지 공통점을 가지게 되는데 이는 자기 조직화된(Self-organization) 노드들로 구성되는 점과 분산 네트워크(Decentralized-Network)로 형성되는 점에 기인한다. 첫 번째 공통점은 두 네트워크 모두 수평적이고 매우 유동적인 네트워크 형태(Network Topology)를 가지고 있는 점이다. P2P 네트워크의 경우 검색 그룹에 가입 탈퇴가 빈번하게 이루어진다. 모바일 Ad hoc 네트워크의 경우 각 노드의 전파 영역이 다르기 때문에 전파 영역에서 사라지는 경우가 발생하기도 하고 Power off 현상이 이루어지기도 한다. 두 번째 공통점은 hop by hop connection을 맺는 점이다. P2P 네트워크의 경우 각 노드에게 검색 요청 메시지(Query Message)를 각 노드에게 보내어서 검색 테이블을 구성한다. 모바일 Ad hoc 네트워크의 경우 모든 노드가 라우팅 역할을 하기 때문에 네트워크 통신을 하기 위해서는 노드 간의 hop by hop 라우팅을 하게 된다.

* 학생회원
** 종신회원

이러한 성격을 가지게 되는 P2P 네트워크와 모바일 Ad hoc 네트워크는 운동성이 존재하고 수평적인 네트워크를 보다 원활하게 지원하는 것이 공통 목표이다. 따라서 P2P 네트워크와 모바일 Ad hoc 네트워크는 상호 보완적 관계를 가지게 되고 둘을 혼합한 네트워크는 높은 시너지 효과를 가지게 된다. 이러한 경향을 반영하듯 최근 모바일 Ad hoc 네트워크를 응용한 여러 가지 응용 서비스들 개발에 P2P 네트워크 알고리즘을 사용하는 추세이다. 특히 센서 네트워크의 경우 각 기기들의 성격과 위치 그리고 정보에 대한 것들을 하나의 서비스로 보아서 빠른 시간에 원하는 서비스를 검색하고 정보를 P2P 통신으로 전송하는 방식에 P2P 네트워크 알고리즘인 DHT, PDI, ORION, GNUTELLA, CHORD 등을 모바일 Ad hoc 네트워크에 적용하려는 시도를 하고 있다.

표 1 P2P 네트워크와 모바일 Ad hoc 네트워크

	P2P Network	Mobile Ad hoc Network
계층	Application	Routing
응용	노드 간의 데이터 검색 / 응용	모바일 hop-by-hop 라우팅
응용 서비스	파일 검색/공유 Ubiquitous 네트워크	센서 네트워크 모바일 통신 Ubiquitous 네트워크
공통점	유동적 네트워크 수평적 네트워크	
상호 보완점	모바일 검색에서 Ad-hoc 라우팅 정보 이용	안정적 네트워크 구조 형성에 이용

본 고에서는 이러한 최근의 연구 방향에 발맞추어, P2P 네트워크를 모바일 Ad hoc 네트워크에 적용하기 위해 개발된 PDI, ORION, DHT, DPSR을 소개하고 모바일 Ad hoc 위에서의 효율적인 P2P 검색을 위한 알고리즘인 Backtracking Chord와 Redundant Chord를 제안하고자 한다.

2.1 PDI 검색 알고리즘[1]

PDI(Passive Distributed Indexing)에서는 중간

노드에서 쿼리의 결과를 캐쉬하여 캐쉬에 저장된 값을 이용한다. 기존의 Gnutella에서는 해당 노드까지 도달하여 쿼리의 결과를 발생할 수 있으므로 검색 시간이 길어진다는 단점이 있다. 하지만 PDI에서는 쿼리의 결과를 중간 노드가 캐쉬하여 쿼리 메시지를 생성한 노드로 전송할 수 있다. 따라서 기존의 Gnutella보다 검색 시간을 단축시킬 수 있다.

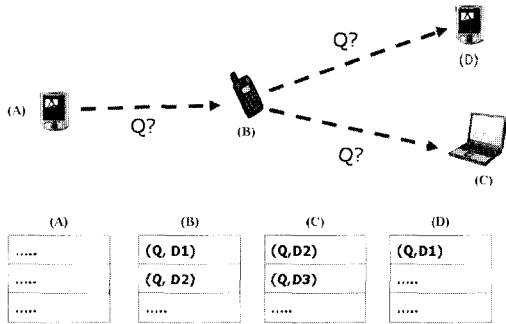


그림 1 모든 노드로 Query를 전달

쿼리 메시지는 생성된 노드로부터 모든 노드로 flooding된다. 그림 1에서 노드 (A)가 Q라는 쿼리 메시지를 생성하면, 노드 (A)는 생성한 쿼리 메시지 Q를 인접한 노드 (B)로 전달한다. 쿼리를 전달 받은 노드 (B)는 인접한 (C)와 (D)로 쿼리를 전달한다. 이러한 방식으로 모든 노드가 쿼리 메시지를 받을 수 있다.

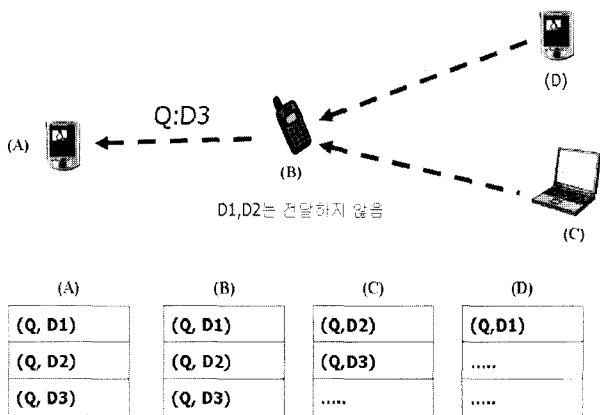


그림 2 경우 노드는 검색 요청 노드에게 전달하지 않은 검색 결과만을 전달

쿼리를 받은 노드는 현재 자기의 인덱스에서 쿼리가 있는지 검사한다. 그림 2에서 (B) 노드는 쿼리의 결과인 D1과 D2를 가지고 있으므로 이를 (A) 노드로 전달한다. 마찬가지로 (C) 노드와 (D)노드에서는 쿼리의 결과인 D2,D3와 D1을 각각 노드 (A)로 전달한다. 이때, 쿼리의 결과를 전달하는 노드는 자기가 전달하지 않은 쿼리 메시지만을 선택하여 전달하고 그 쿼리의 결과를 자신의 인덱스에 저장한다. 그림 3에서 쿼리의 결과

를 전달하는 노드 (B)는 자신의 인덱스를 검색하여 이미 전달한 결과 D1, D2를 제외한 D3 만을 노드 (A)에게 전달하고 D3를 자기 자신의 인덱스에 저장한다.

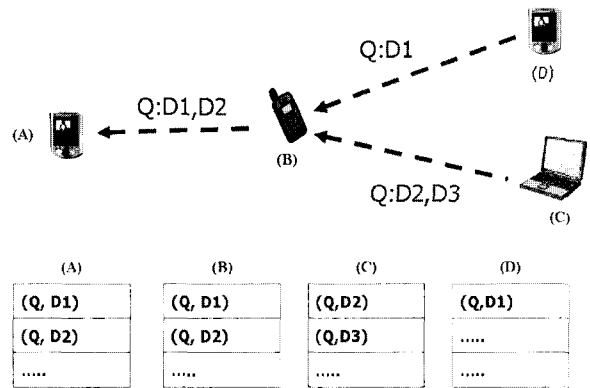


그림 3 검색 요청 노드로 검색 결과 전달

PDI는 중간 노드가 쿼리의 결과를 캐쉬하여 그것을 이용할 수 있으므로 Gnutella에서 보다 검색 시간을 줄일 수 있고, 인덱스만을 캐쉬하므로 캐쉬에 저장되는 인덱스의 크기를 줄일 수 있다. 또한 노드의 join, leave 시에 테이블 생성의 비용이 거의 들어가지 않으므로 확장성(Scalability)이 높다. 하지만 캐쉬를 사용하므로 캐쉬에 대한 validation 문제가 발생한다.

2.2 ORION 검색 알고리즘[1]

ORION 검색 알고리즘은 모바일 Ad hoc 네트워크에서 Query flooding 방식을 적용한 P2P 검색 방식이다. 하지만 무선 broadcasting에 의한 단순한 Query flooding의 경우 자료를 찾아가는 경로의 신뢰성이 떨어지게 된다. 이러한 문제점을 해결한 검색 알고리즘이 ORION 알고리즘이다. ORION 알고리즘은 기존의 PDI (Passive Distributed Indexing)의 방식에 경로의 신뢰성을 높이고자 Ad hoc 라우팅 정보를 이용하였다. ORION은 다음과 같다. 일단 검색 요청 모바일 노드는 Query 메시지를 모든 노드에게 flooding 한다.

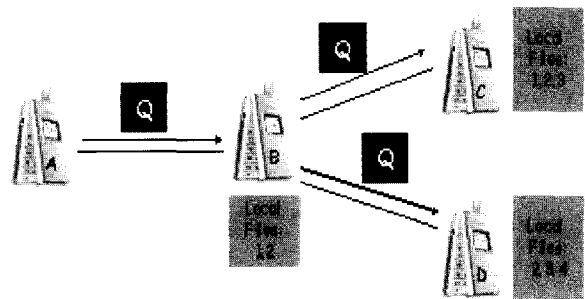


그림 4 Query flooding

Query 메시지를 받은 모든 노드는 각자가 가지고 있

는 데이터 목록을 다른 노드들에게 알려야 한다. 이 때 기존의 방식처럼 Query 메시지에 반응하는 파일리스트를 전송하지 않고 Ad hoc 라우팅 경로에 의해 중간에 경유하는 모든 노드들이 라우팅 정보를 이용한 File Routing table이라는 것을 만들어서 모든 노드에 파일리스트를 Caching하게 된다. 예를 들어 설명하면 그림 5에서 노드 B는 자료 1번 2번 자료를 가지고 있는 것을 노드 A에게 알릴 때 RESPONSE(1,2)라는 메시지를 보낸다. RESPONSE(1,2)를 받은 노드 A는 File Routing table에 1,2번 자료를 찾아가는 경로를 B로 설정하는 정보를 갱신한다.

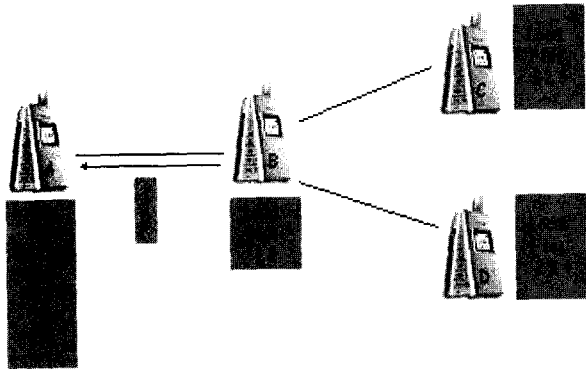


그림 5 노드 B의 반응 메시지

그림 6에서 노드 C와 노드 D의 경우는 노드 A에게 알리기 위해서 노드 B를 경유해서 알리게 된다. 이 때 노드 C는 노드 B에게 RESPONSE(1,2,3)를 노드 D는 RESPONSE(2,3,4)를 전송하게 된다. 노드 B는 자신의 File Routing table에 자신에게 없는 3번, 4번 자료를 갱신하게 되고 RESPONSE(3,4)를 노드 A에게 전송하게 된다. RESPONSE(3,4)를 받은 노드 A는 File Routing table에 3,4번 자료를 찾아가는 경로를 B로 설정하는 정보를 갱신한다. 이 때의 주목할 점은 원자료가 있던 노드로 갱신하는게 아니라 RESPONSE(3,4)를 보낸 Neighbor 노드인 노드 B로 3번 4번 자료를 Indexing 하게 된다는 것이다. 이렇게 File Routing table을 구성한 노드 A는 각각의 자료들을 찾아갈 때 File Routing table에 의해서 경로를 따라가게 된다. 즉 3번 자료를 찾아갈 때는 노드 B를 경유, 노드 B의 File Routing table을 이용하여 노드 C로 이동하게 된다. 즉 자료를 찾아가는 방식이 Ad hoc 라우팅 테이블을 이용한 hop by hop 라우팅처럼 ORION의 경우는 File Indexing 정보를 라우팅 정보에 적용한 File Routing table을 이용한 hop by hop indexing을 하게 된다. 이와 같은 방식을 이용할 경우 기존의 무선 Ad hoc 네트워크의 P2P 검색 방식보다 높은 경로의 신뢰

성을 보여주게 되며 모든 경유 노드들이 Query table(File Routing Table)을 Caching 해주기 때문에 검색 시간을 단축할 수 있다. 또한 모든 자료들의 경로 설정을 전부 다 하는 것이 아니라 Next Hop에 대한 경로만을 설정함으로써 각 노드들의 자료 검색 리스트의 크기를 줄어듬으로써 노드들의 검색 리스트의 크기에 따른 Overhead를 줄여줄 수 있다.

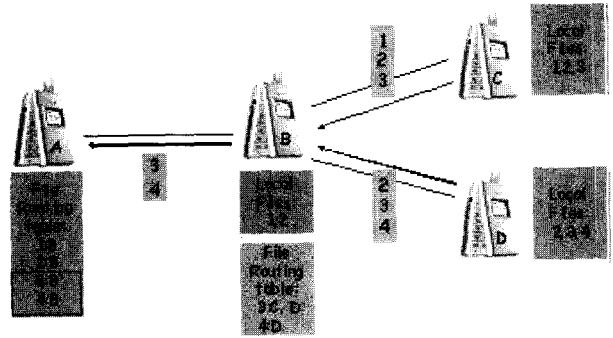


그림 6 노드 C, D의 반응 메시지

2.3 DHT 검색 알고리즘[3]

DHT(Distributed Hashing Table) 검색 알고리즘은 파일과 노드 정보를 노드에 분산 저장하여 빠르고 정확한 검색을 하기 위한 알고리즘이다. Gnutella 기반의 query flooding 방식 경우 라우팅을 위해 주변 노드 모두에게 query를 보내므로 불필요한 메시지를 발생하게 되는데, DHT 기반의 알고리즘은 라우팅 테이블에 의해서 필요한 query 메시지만을 보낸다. 이때 전체 노드에 대한 라우팅 정보를 갖는 것이 아니라 주변 노드에 대한 정보만을 갖는다. 또한 파일 정보를 노드 정보와 함께 이용하여 파일을 검색하는 과정이 곧 라우팅 과정이다. 즉 라우팅을 통하여 파일 검색이 이루어진다.

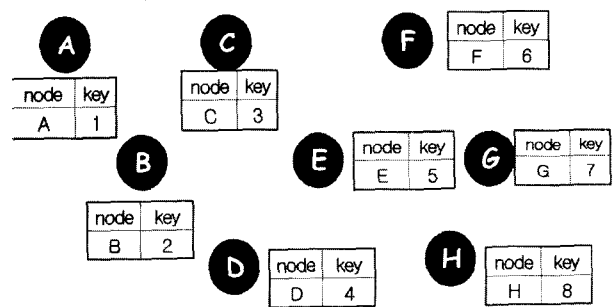


그림 7 모바일 노드들과 파일 정보

그림 7에서 노드 A 부터 노드 H까지 각 노드는 순서대로 파일을 1번부터 8번까지 가지고 있다. 이 파일에 대한 key 값은 해싱 함수를 사용한 고유한 값으로 노드의 IP 주소를 해싱한 값과 서로 묶여 있고 정보를 노드

간 통신을 통하여 공유한다.

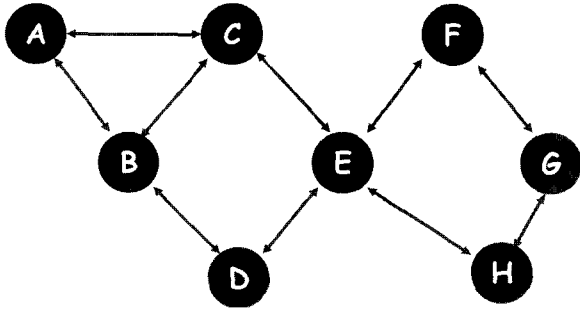


그림 8 노드 간의 정보 교환

노드 간 공유란 Ad hoc 환경에서 노드 범위가 한정되어 있기 때문에 주변 노드만 정보를 교환한다. 그림 8에서 노드 A는 노드 B, 노드 C와 정보를 주고 받는다. 따라서 노드 A는 자신의 파일 key 값인 1과 주변 노드 B, C 정보인 2와 3을 라우팅 테이블에 저장한다.

그림 9는 각 노드의 라우팅 정보 교환 후의 결과이다. 각 노드는 주변 노드들의 파일 정보를 테이블 형태로 저장하고 이 테이블을 이용하여 파일 검색을 한다.

그림 10에서 노드 A가 key 7을 찾고자 할때 먼저 자신의 라우팅 테이블을 보고 key값이 있는지 찾는다. 없다면 key 7과 가장 가까운 key 값인 key 3을 가지고 있는 노드 C로 찾아간다. 만일 노드 C의 라우팅 테이블에서도 없으면 key 7과 가장 가까운 key 5를 가지고 있는 노드 E로 찾아간다. 최종 노드 경로는 노드 A->노드 C->노드 E->노드 H->노드 G이다.

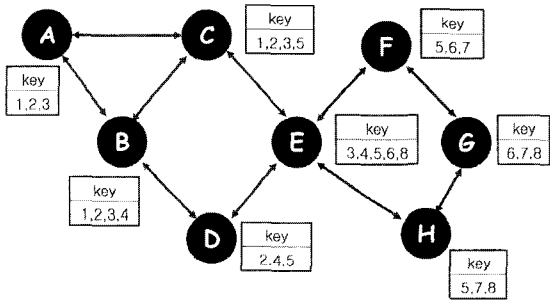


그림 9 노드 간의 정보 교환 후의 라우팅 정보

DHT 알고리즘을 사용하면 라우팅 경로를 찾기 위한 불필요한 메시지를 줄일 수 있다. 이는 모바일 특성상 제약된 파워 소비와 대역폭 사용에 대한 중요한 장점이다. 많은 메시지를 발생할수록 파워 소비도 증가하고 대역폭도 많이 사용하게 된다.

또한 broadcast 방식보다 scalability 측면에서 우수한 성능을 나타낸다. broadcast 방식 경우 노드 전체에 query를 보내야 하므로 노드 수가 많아질수록 메시지의 양도 증가하고 검색을 위해 기다리는 시간 또한 증

가한다. 하지만 DHT 알고리즘은 주변 노드에 대한 정보만을 갖고므로 추가되는 노드에 대한 overhead를 크게 줄일 수 있다.

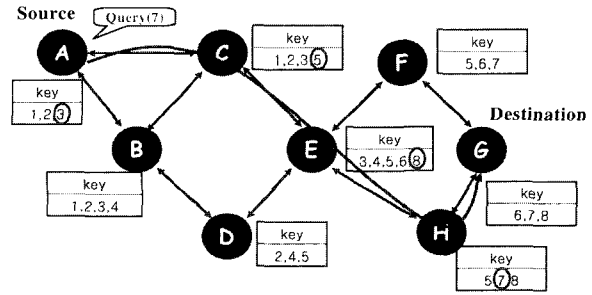


그림 10 노드A가 노드G를 찾는 과정

2.4 모바일 Ad hoc 네트워크의 Chord 검색 알고리즘

Chord[5] 방식은 Query flooding(예, Gnutella) 방식의 단점인 검색 시간의 지연 및 대역폭 낭비와 검색 메시지의 증가를 해결하고자 고안된 P2P 검색 알고리즘이다. Chord는 Query flooding 방식처럼 모든 노드에게 Query를 전송하는 방식이 아닌 몇몇 경유 노드(Successor)를 설정 그림 11에서와 같이 검색 범위를 좁혀가면서 릴레이 검색을 한다.

Chord 방식은 Query Flooding 방식보다 메시지 전송 양과 Query 전송 대역폭이 작다.[표 2]

표 2 P2P 구조에 따른 메시지와 대역폭

검색 구조	Query Message	Bandwidth
Gnutella	$O(N)$	$O(N)$
Chord	$O(\log N)$	$O(\log N)$

하지만 Chord 방식을 모바일 Ad hoc 네트워크에 그대로 적용할 경우 문제점이 있다. 그 이유는 그림 12의 경우와 같이 Successor가 전파 영역을 벗어나거나 갑자기 Power-off되었을 경우에 더 이상 릴레이 검색이 이루어지지 않는다. 이러한 문제점을 해결하고자 제안한 검색 알고리즘이 Backtracking Chord와 Redundant Chord이다.

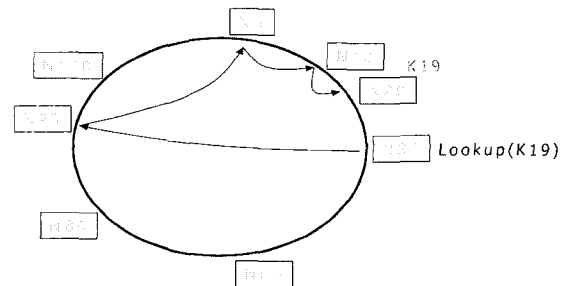


그림 11 Chord 검색 방식

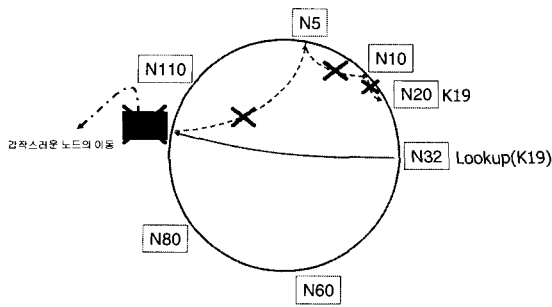


그림 12 Chord 쿼리 검색 실패

2.4.1 Backtracking Chord 검색 알고리즘

Backtracking Chord는 일정한 시간(Time-out)을 설정해 두어서 Successor가 갑자기 사라지더라도 검색을 멈추지 않고 새로운 Successor로 검색 요청을 하도록 한다.

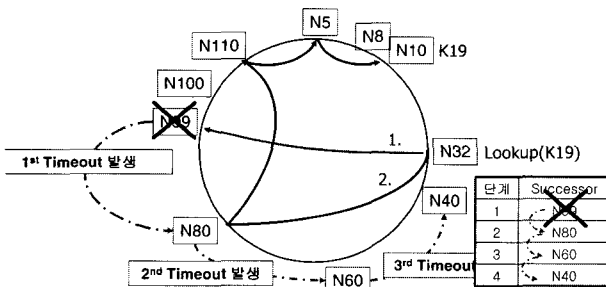


그림 13 Backtracking Chord

이때 새로운 Successor로 지정되는 노드는 현재 Chord를 구성하고 있는 노드들 중에서 사라진 Successor¹⁾의 Successor table의 entry 중 바로 앞에 해당하는 노드(Predecessor²⁾)가 설정된다. 이 Predecessor는 요청했던 노드에게 새로운 성공자로서 Successor Table에 등록되게 된다. 이러한 방식은 $O(\log N)$ 검색 요청 메시지의 사용으로 t 배 만큼의 검색 성공률을 향상시킬 수 있다. 하지만 기존의 Chord 방식보다 검색시간이 $t \times Timeout$ 만큼 더 지연되게 된다. 하지만 검색 요청에 드는 대역폭은 한 번 보내는데 $O(\log N)$ 메시지를 사용하기 때문에 절약된다.

(t : Timeout의 횟수($0 < t < \log N$))

2.4.2 Redundant Chord 검색 알고리즘

검색 요청 메시지를 순차적으로 보내는 Backtracking 방식과는 달리 동시에 Successor table에 있는 복수의

- 1) Successor : Chord 검색 그룹에서 파일 검색을 위해 선정하는 노드[1](N개의 노드가 존재할 경우 각 노드는 최대 $\log N$ 개의 Successor를 가지게 된다.)
- 2) Predecessor : Successor table에서 검색 요청할 Successor와 함께 노드가 인지하는 entry 내의 다른 Successor[1]

Successor들에게 전송한다. 이 경우에 검색 요청에 걸리는 메시지는 $O(r \log N)$ 으로 증가하게 되지만 동시에 다중의 Successor들에게 검색 요청을 하기 때문에 빠른 검색 속도를 보일 수 있다.

(r : 복수 검색 요청 메시지(Query)의 전송 횟수($0 < r < \log N$))

중복된 테이블에 의한 릴레이 검색으로 인해서 중복된 검색 요청 중 검색이 보다 빠르게 이루어지는 검색 노드를 이용함으로써 그림 6과 같이 노드가 없어져도 검색 요청이 성공적으로 이루어지는 쪽을 동시에 선택하기 때문에 최소 $O(\log N)$ 의 검색 시간을 유지하면서 모바일 Ad-hoc 환경에서 안정적인 검색을 유지할 수 있다. [JCCI 2004 게재 예정]

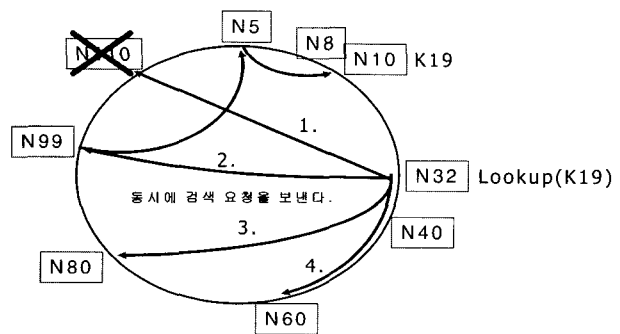


그림 14 Redundant Chord

2.5 DPSR(Dynamic P2P Source Routing) 알고리즘[6]

앞에서 설명한 방식이 P2P 검색에 중점을 둔 것이었다면 DPSR은 P2P에서의 라우팅(Routing) 기법을 모바일 Ad hoc 네트워크에서의 라우팅 경로 구성 방법에 적용한 알고리즘이다. 즉, 앞에서 서술한 ORION, PDI, DHT, Backtracking Chord, Redundant Chord 등이 모바일 Ad hoc 네트워크상에서 좀 더 효율적인 "P2P 파일 검색방법"을 제안한 알고리즘이라면, DPSR의 경우 좀 더 "광범위한 모바일 Ad hoc 네트워크의 구성"을 위해 Ad hoc 라우팅에 P2P라우팅 기법을 도입한 알고리즘인 것이다.

기존의 DSR(Dynamic Source Routing)에서 소스(Source) 노드는 hop by hop 방식으로 라우팅 경로를 탐색하고, 그 결과를 목적지(Destination) 별로 라우팅 경로 전체를 캐싱(Caching) 한다. 따라서 모바일 기기의 특성으로 인한 캐싱 메모리의 제약과 one hop 단위의 라우팅 때문에 광범위한 네트워크를 구성하기에는 한계가 있었다.

반면, DPSR은 기존의 DSR에 Pastry P2P 라우팅

기법을 도입함으로써 각 노드가 저장하는 캐싱 크기 (Caching size)는 줄어들면서도 모바일 Ad hoc 네트워크의 확장성은 높인다.

DPSR의 라우팅 방법을 소개하면 다음과 같다. DPSR에서는 Pastry P2P 라우팅에서처럼 모든 노드에 각각의 IP 주소에 따라 SHA-1(secure hashing)을 이용하여 유일하게 계층화된 노드 ID를 부여한다. 예를 들어 설명하자면, 그림 15에서 소스 노드인 0112는 목적지 노드 2001까지 갈 수 있는 라우팅 경로를 탐색하고자 한다.

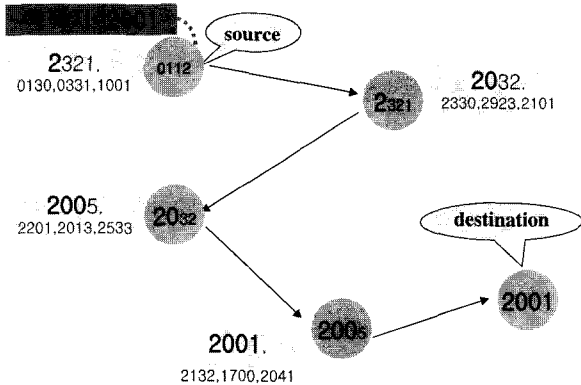


그림 15 소스(Source) 노드에서 적지(Destination) 노드까지 경로 탐색 과정

일단 소스 노드(0112)는 자신의 캐싱 테이블에 2001까지 직접 갈 수 있는 경로가 있는지 확인한다. 자신의 테이블에는 (2321, 0130, 0331, 1001)으로의 경로만 있을 뿐 2001까지 직접 갈 수 있는 경로가 없으므로, 캐싱 테이블에 목적지 노드 ID(2001)의 첫째 자리수인 2와 일치하는 노드가 있는지 만을 확인한다. 그 결과 2321이 있으므로 목적지 2001에 대한 소스의 탐색 (Discovery) 메시지를 2321에게 전송한다. (이때, ID의 첫째 자리수가 2인 노드가 없다면 2001과 가장 가까운 노드를 선택한다.) 메시지를 수신한 2321 노드는 자신의 캐싱 테이블에서 목적지 노드 ID의 자릿수(2001) 중 두 개 자리인 20과 일치하는 노드가 있는지 확인한다. 그 결과 2032가 있으므로 메시지를 2032에게 전송한다. 같은 방법으로 2032도 목적지 노드 ID의 자릿수 (2001) 중 세 개 자리인 200과의 일치 여부만 확인한 후 메시지를 2005에게 전송하고, 결국 2005는 최종 목적지인 2001에게 소스의 탐색 메시지를 전달한다. 이러한 방식을 통해 소스 노드는 "0112-> 2321-> 2032-> 2005-> 2001" 이라는 특정 라우팅 경로 전체를 캐싱하지 않고도 단 하나의 테이블 만으로 원하는 목적지까지의 경로를 확보한다.

그림 16은 소스(Source) 노드 0112의 목적지(Destina-

tion) 노드 2001에 대한 탐색(Discovery) 메시지가 소스 (Source) 노드를 출발하여 목적지(Destination) 노드 까지 수렴하는 과정을 보다 간명하게 나타내고 있다. 결국 DPSR에서는, 기존 알고리즘(DSR)에서와 같은 라우트 요청(Route Request) / 라우트 응답(Route Reply)의 과정 없이도 소스 노드 자신의 캐싱 테이블 정보만으로 목적지 노드까지의 경로를 탐색함을 알 수 있다.

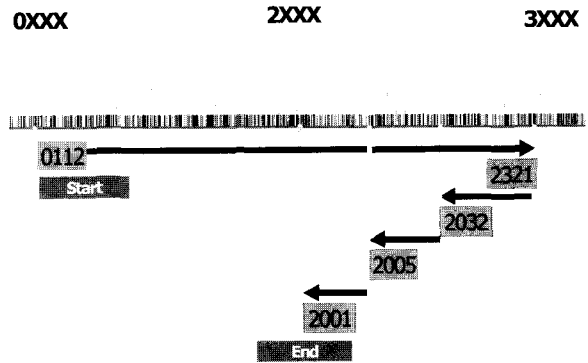


그림 16 탐색(Discovery) 메시지의 수렴 과정

기존 알고리즘(DSR)에 비해 DPSR의 진보성은 그림 17과 그림 18에서와 같이 캐싱 테이블을 단순화하여 살펴볼 때 더욱 명확히 드러난다. [그림 17]은 기존 DSR 소스 라우팅(Source Routing) 방법으로서, S는 자신의 캐싱 테이블에 『S-> F-> G-> J-> K-> D』의 형식으로 S로부터 목적지 D까지의 전체 경로를 저장한다. 뿐만 아니라 S와 D 사이의 중간 노드 (intermediate node)들도 각자가 자신들로부터 D까지의 전체 경로를 캐싱 테이블로 저장하므로 캐싱을 위한 각 노드의 Overhead가 크다.

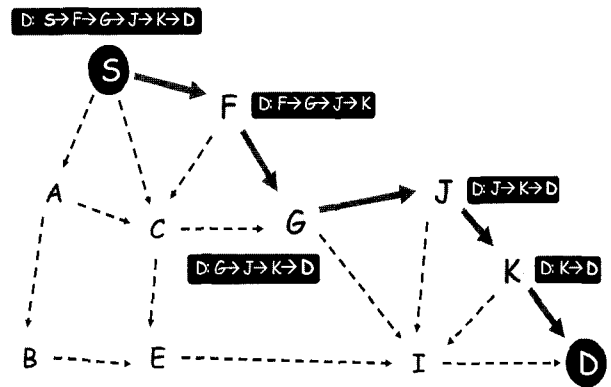


그림 17 DSR의 소스(Source) 라우팅과 캐싱(Caching)

반면, 그림 18의 DPSR 소스 라우팅(Source Routing) 방법의 경우, S는 자신의 캐싱 테이블에 자신보다 D에 근접한 노드인 G 만을 저장하고 있으며 중간노드들도 다음 홉(Next hop) 만을 저장하기 때문에 캐싱 크기가

매우 작다. 특히, DPSR에서 주목할 만한 점은 Multi-hop 소스 라우팅을 구현한다는 점이다. [그림 17]의 기존 라우팅 알고리즘의 경우, S는 단순히 one-hop 방식으로 라우팅을 구현함에 비해 [그림 18]의 DPSR의 경우 [그림 15]에서와 같은 계층화를 통해 『S→G→K→D』와 같이 Multi-hop 방식으로 라우팅을 구현한다.

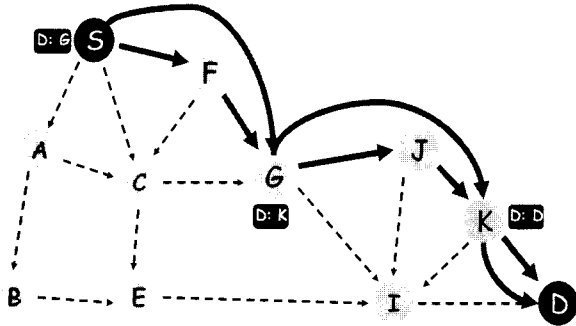


그림 18 DPSR의 소스(Source) 라우팅과 캐싱(Caching)

DPSR은 모바일 Ad hoc 네트워크의 라우팅 경로를 구성함에 있어, On-demand 방식의 소스 라우팅의 성격을 그대로 유지하면서도 Pastry P2P 라우팅 기법을 접목시켜 각 노드에 계층화된 ID를 부여한다. 그리고 이에 기초하여 라우팅 경로를 결정하므로 다량의 라우트 요청(Route Request) 메시지나 라우트 응답(Route

Reply) 메시지로 인한 대역폭(Bandwidth) 낭비 및 Collision 발생을 억제한다. 뿐만 아니라 캐싱 크기를 획기적으로 줄이면서도 Multi-hop 방식의 소스 라우팅을 실현하여, 모바일 기기의 캐싱 크기 Overhead 문제와 모바일 Ad hoc 네트워크의 확장성 제한문제를 극복하였다.

3. 비교 분석

모바일 Ad hoc 네트워크는 매우 유동적이고 변화가 심한 네트워크이다. 이러한 환경에서 P2P 네트워크 구축을 위해서는 네트워크 변화에 민감한 P2P 검색 알고리즘과 함께 Ad hoc의 라우팅에 적합한 P2P 알고리즘 필요하다. 본 고에서는 모바일 Ad hoc 네트워크의 특성에 적합한 P2P 알고리즘으로 빠른 검색 시간이 장점인 PDI, 높은 신뢰성이 장점인 ORION, 검색 메시지의 양을 축소시킨 DHT를 소개하였다. 하지만 소개한 알고리즘들은 제한적인 무선 대역폭 측면과 각 단말기들의 제한적인 메모리 및 파워문제에 의해 실제 네트워크에서의 실효성에는 의문이 제기된다.

반면, 본문 2.4절에서 제안한 Backtracking Chord 검색 알고리즘과 Redundant Chord 검색 알고리즘은 기존의 알고리즘에 비해 모바일 Ad hoc 네트워크 환경에 최적화된 P2P 검색 알고리즘이라고 할 수 있다.

표 3 모바일 Ad hoc 네트워크에서 P2P 알고리즘 비교 분석

	PDI 검색 알고리즘	ORION 검색 알고리즘	DHT 검색 알고리즘
장점	<ul style="list-style-type: none"> • 중간 노드가 쿼리 결과를 저장하므로 검색시간을 줄임 • 초기 및 노드의 leave, join 시 테이블 생성 비용이 없어 scalability가 높음 	<ul style="list-style-type: none"> • 자료를 찾아가는 경로의 신뢰성이 높음. • 쿼리 테이블의 캐싱으로 검색시간의 단축 • Next hop에 대한 경로만을 저장하여 검색 리스트의 크기가 줄어듦 	<ul style="list-style-type: none"> • 불필요한 query 메시지를 줄임으로써 메시지 발생과 대역폭 낭비를 줄임 • 주변 노드의 라우팅 정보만 저장함으로써 메모리 사용을 줄이고 scalability를 높임
단점	<ul style="list-style-type: none"> • 캐쉬에 대한 Invalidity 문제 발생 	<ul style="list-style-type: none"> • 라우팅 정보를 저장하므로 노드의 이동시 경로의 신뢰성이 떨어짐 	<ul style="list-style-type: none"> • 모바일 기기의 특성인 mobility가 자주 일어날 경우 라우팅 테이블을 유지하기 위한 overhead가 큼
출처	Proc. 2nd IEEE Conf. on Peer-to-Peer Computing (P2P 2002), Linköping, Sweden, 71-83, September 2002	IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), Orlando, FL, October 2003	International Workshop on Mobile Peer to Peer Computing 2004 (IWMP2P '04)
저자	C. Lindemann and O. Waldhorst, Dept. of Computer Science, University of Dortmund	C. Lindemann and O. Waldhorst, Dept. of Computer Science, University of Dortmund	Bharat K Bhargava, Dept. Computer Science, Purdue university

표 4 모바일 Ad hoc 네트워크에서 P2P 알고리즘 비교 분석

	Backtracking Chord 검색 알고리즘	Redundant Chord 검색 알고리즘	DPSR 검색 알고리즘
장점	<ul style="list-style-type: none"> 작은 검색 요청 메시지로 높은 검색 성공률을 보장(무선 대역폭 절약) 	<ul style="list-style-type: none"> 빠른 시간에 일정 수준의 검색 성공률을 보장(검색 시간 절약) 	<ul style="list-style-type: none"> 라우트 요청/응답 메시지를 사용하지 않음으로서 대역폭 절약 및 다량의 메시로 인한 메시지 지연 억제 지역성을 반영한 계층화를 통한 멀티홉 방식의 소스라우팅 구현 라우팅 과정에서 캐싱 크기를 줄임
단점	<ul style="list-style-type: none"> 검색 시간의 지연 	<ul style="list-style-type: none"> 많은 무선 대역폭 사용 동시에 많은 검색 요청 메시지 전송으로 인한 프로세싱 오버헤드 발생 	<ul style="list-style-type: none"> 모든 노드에게 계층화된 유일한 ID를 부여하는 과정의 부담 각 노드에 부여된 ID가 지역성을 완벽히 반영 못할 수 있음
출처	Peer to Peer Query Search Over Mobile Ad hoc Network (JCCI 2004 게재 예정)		Ninth Workshop on Hot Topics in Operating Systems, Lihue, Kauai, Hawaii, May 18 21, 2003. (19 long/144)
저자	이세연, 장주욱(서강대학교), 이경근(세종대학교), 조태경(SKT)		Y. Charlie Hu, Distributed Systems and Networking Lab, Purdue University

특히, Redundant Chord 검색 알고리즘은 중복된 테이블에 의한 릴레이 검색 방법을 이용하여 중복된 검색 요청 중 검색이 보다 빠르게 이루어지는 검색 노드를 이용함으로써 노드가 없어져도 검색 요청이 성공적으로 이루어지는 쪽을 동시에 선택하기 때문에 최소 $O(\log N)$ 의 검색 시간을 유지하면서 모바일 Ad-hoc 환경에서 안정적인 검색을 유지할 수 있다. 그러나 제안된 알고리즘 또한 제한된 무선의 대역폭을 고려해서 P2P 검색 메시지의 크기 또한 작아야 한다는 문제점을 해결해야 한다.

4. 향후 전망

모바일 Ad hoc 네트워크를 이용한 서비스 도입은 여러 분야에서 도입을 시도하고 있다. 특히 작년 Ubiquitous 개발의 원년의 해가 되면서 차세대 네트워크의 형태로 Ubiquitous 네트워크가 각광을 받고 있다. 특히 앞에서 언급했던 센서 네트워크가 이러한 모바일 Ad hoc 네트워크에 P2P 네트워크를 구축한 통합 네트워크를 이용하려 하고 있다. 센서 네트워크는 움직이는 사람의 이동 경로와 위치한 지점을 인식하고 센서 네트워크에 연결된 다른 기기들을 자동으로 제어하려는 네트워크이다. 따라서 각 기기들이 제공하는 서비스와 움직이는 사람이 원하는 서비스를 빠른 시간에 검색하고 그에 맞는 서비스를 제공받는 것에 그 목적이 있다. 따라서 움직이는 사람은 모바일 노드로서 모바일 Ad hoc 네트워크 기반

으로 통신을 하게 되고 그 위에서 서비스 검색 및 제공을 받을 경우 P2P 네트워크의 구축을 통해서 빠르고 정확한 검색을 하고 서비스를 제공받게 되는 것이다.

그림 19은 바로 그러한 센서 네트워크 개념도를 표현한 그림이다. Target이 움직이는 사람이라 할 수 있다. 센서들은 바로 이 Target과 모바일 Ad hoc 네트워크에서 P2P 통신으로 서비스를 제공한다.

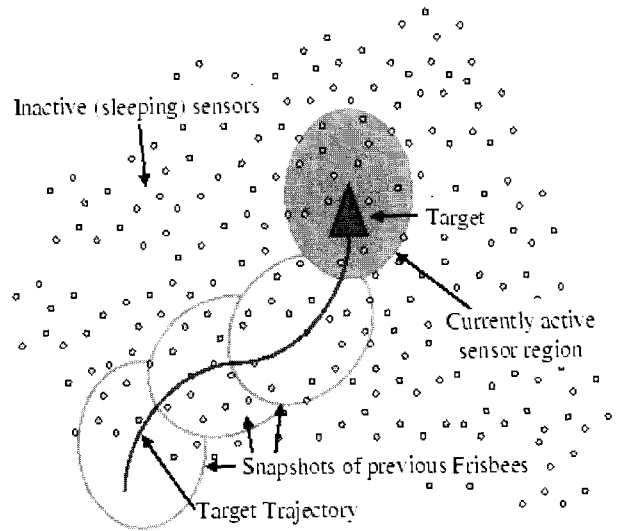


그림 19 센서 네트워크의 개념도

이처럼 Ubiquitous Network의 구축으로 인해 모바일 Ad hoc 네트워크에서 P2P 네트워크를 구축을 하고

자 하는 움직임은 통신 수단의 발달과 통신 수요의 증가 측면에서 무시할 수 없는 부분으로 떠오르고 있다. 따라서 앞으로도 이러한 모바일 Ad hoc 네트워크에 적합한 P2P 네트워크 알고리즘에 대한 연구는 Ubiquitous 네트워크에서 꼭 필요한 분야이다.

참고문헌

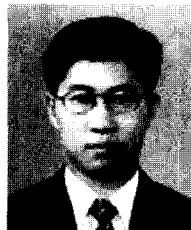
- [1] Alexander Klemm, Christoph Lindemann, and Oliver P. Waldhorst: "A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks" IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), Orlando, FL, October, 2003.
- [2] Elias C. Efstathiou and George C. Polyzos: "Designing a Peer-to-Peer Wireless Network Confederation" 3rd ACM Workshop on Data Engineering for Mobile and Wireless Access, San Diego CA, Sept. 2003.
- [3] Gang Ding and Bharat Bhargava: "Peer-to-peer File-sharing over Mobile Ad hoc Networks" International Workshop on Mobile Peer to Peer Computing 2004 (IWMP2P '04)
- [4] Y. Charlie Hu, Saumitra M. Das, and Himabindu Pucha: "Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks" 9th Workshop Hot Topics in Operating Systems(HotOSIX)'03
- [5] I. Stoica, R. Morris, D.Karger, M.F Kashoek, and H. Balakrishnan: "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications" Proc. ACM SIGCOMM'01, 149-160, 2001
- [6] B.Y.Zhao, J.D.Kubiatowicz, and A.D.Joseph: "Tapestry: An Infrastructure for Fault-Resilient Wid-area Location and Routing." Technical Report UCB//CSD-01-1141,U.C. Berkeley, April, 2001

이 세 연



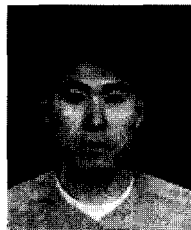
2003 서강대학교 전자공학과(공학사)
2003~현재 서강대학교 전자공학과 인터넷 연구실 석사 과정
관심분야: P2P network, Mobile Internet, Ad-hoc, Home Network
E-mail : seiyon@eecl.sogang.ac.kr

장 주 욱



1983 서울대학교 전자공학과(공학사)
1985 한국과학기술원 전기및전자공학과(공학석사)
1985~1988 삼성전자 연구원
1993 University of Southern California 전자공학과(공학박사)
1995~현재 서강대학교 전자공학과 부교수
관심분야: P2P network, High Speed Networking, Ad hoc network
E-mail : jjang@sogang.ac.kr

조 태 경



1995. 2 숭실대학교 전산과 학사
1997. 2 한국과학기술원 전산과 석사
2000. 4 대우 고등기술 연구원
현재 SK 텔레콤 네트워크 연구원
관심분야: WCDMA PS Core Network, IMS, P2P
E-mail : tkcho@sktelecom.com

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>