

# 이동 애드혹 망에서의 서비스 검색을 위한 근접성 기반 오버레이 네트워크 라우팅

## (Proximity-based Overlay Network Routing for Service Discovery in Mobile Ad-Hoc Network)

윤현주<sup>†</sup> 이은주<sup>\*\*</sup> 정현구<sup>\*\*\*</sup> 김진수<sup>†</sup>  
(Hyeon-Ju Yoon) (Eunju Lee) (Hyunku Jeong) (Jin-Soo Kim)

**요약** 최근 산업 및 연구 단체에서 이동 애드혹 망에 대한 관심이 높아지고 있다. 지금까지의 연구들은 하위 계층에 초점을 맞추고 있었던 것에 반해, 본 논문에서는 이들 연구의 결과로 형성된 이동 애드혹 망을 실제로 사용하기 위한 응용의 하나로 효율적인 서비스 검색 방법을 제시한다. 통신 오버헤드 측면에서 서비스 검색을 효율적으로 하기 위하여 피어 투 피어(P2P) 오버레이 네트워크에서 사용되는 분산 해쉬 테이블 시스템을 이용한다. 그러나 오버레이 네트워크는 물리적 네트워크 토폴로지와 독립적이기 때문에 기존의 유선망에서 사용되던 토폴로지 기반 메커니즘들은 노드들의 이동이 잦아 물리적 토폴로지가 수시로 바뀌는 이동 애드혹 망에 부적합하다.

본 논문에서는 오버레이 네트워크에서 라우팅의 비효율성을 극복하기 위한 근접성 기반 오버레이 네트워크 라우팅 기법을 제안한다. 근접성 기반 오버레이 네트워크 라우팅 기법에서, 각 노드는 1홉 브로드캐스팅을 통해 물리적으로 가까운 노드의 정보를 모으고, 이를 이용하여 논리적으로 목적지에 가까운 노드를 선택하여 메시지를 전송한다. 이러한 방식으로 메시지를 전송할 경우, 분산 해쉬 테이블 시스템과 같이 낮은 오버헤드를 유지하면서 플러딩 기반 기법과 비슷한 정도로 물리적인 홉 수를 줄일 수 있고, 노드들의 이동성이 있는 환경에서도 좋은 성능을 나타낸다는 것을 ns-2 시뮬레이션을 통해 확인할 수 있다.

**키워드** : 이동 애드혹 망, 서비스 검색, 오버레이 네트워크, 피어-투-피어, 분산 해쉬 테이블

**Abstract** Mobile ad hoc networks(MANET) have recently attracted a lot of attention in the research community as well as in industry. Although the previous research mainly focused on the various problems of MANET in data link and network layers, we consider, in this paper, how to efficiently support applications such as service discovery on top of MANET. Peer-to-Peer(P2P) overlay network can be adopted to service discovery mechanism because P2P and MANET share certain similarities, primarily the fact that both are instances of self-organizing decentralized systems. Especially, distributed hash table(DHT) systems used for P2P overlay network can be effective in reducing the communication overhead in service discovery. However, since overlay network is independent of physical network topology and existing topology-aware mechanisms are based on the wired network, they are inefficient in MANET.

We propose a proximity-based overlay network routing to overcome the inefficiency of routing in overlay network. In the proximity-based overlay network routing, each node collects information of physically close nodes by using one hop broadcast and routes messages to the logically closest node to destination. In a detailed ns-2 simulation study, we show that the proximity-based overlay network routing reduces the number of physical hops comparable to the flooding-based mechanism with low communication overhead. We also find that the proposed scheme works well in the mobile environment.

**Key words** : mobile ad-hoc network, service discovery, overlay network, peer-to-peer, distributed hash table

<sup>†</sup> 종신회원 : 한국과학기술원 전자전산학과 교수  
juyoon@kaist.ac.kr  
jinsoo@cs.kaist.ac.kr  
<sup>\*\*</sup> 비회원 : 삼성전자 정보통신총괄 무선사업부  
osis.lee@samsung.com

<sup>\*\*\*</sup> 비회원 : 한국과학기술원 전자전산학과  
hkjeong@camars.kaist.ac.kr  
논문접수 : 2004년 2월 15일  
심사완료 : 2004년 8월 30일

## 1. 서론

1970년대 DARPA 패킷 라디오 네트워크로 처음 모습을 드러낸 이동 애드혹 망(MANET: Mobile Ad-hoc Network)은 인프라스트럭처(infrastructure) 없이 노드들이 필요에 따라 임시적으로 네트워크를 형성하여 패킷 데이터를 주고 받는 망의 한 형태이다. 최근 무선 통신 장비의 가격 저하와 무선 이동 통신 기술의 성장으로 언제 어디서나 컴퓨팅이 가능하도록 하는 기술(pervasive computing)[1]에 대한 요구가 높아지면서, 통신 비용이 적은 MANET에 대한 관심이 더욱 커졌다. 이에 따라 효율적으로 MANET을 구성하기 위한 연구들이 활발히 진행되어 왔다. 지금까지 진행된 연구의 대부분은 노드 사이에 데이터를 전송하기 위한 라우팅 프로토콜, 효율적 전력 소비를 위한 메커니즘, 통신의 품질 보장을 위한 대역폭 확보 등에 초점을 맞추고 있었고, 연구들이 진행됨에 따라서 효과적인 기술들이 소개되었다. 이제는 이들 연구들을 통해 형성된 MANET을 실생활에서 이용할 수 있도록 응용 계층에 대한 연구에 눈을 돌려야 할 때이다.

그러한 연구 중에서 우선적으로 진행되어야 하는 것 중에 하나는 유선망에서처럼 MANET에서도 각 노드들이 가지고 있는 서비스를 효율적으로 검색하는 방법이다. 잘 알려진 서비스 검색 메커니즘으로는 안정적인 중앙 집중형 디렉터리 서버를 가정하거나(SLP[2], UDDI[3], Jini[4]), 플러딩을 기반으로 하는 기법(UPnP[5]의 SSDP, JXTA[6]) 등이 있다. 그러나 이들은 일반적으로 유선망을 기준으로 설계되었기 때문에, 노드가 자주 이동하고 전력 등 사용 자원이 제한되어 있는 MANET에 적용하기에는 적합하지 않고, 특히 중앙 서버 방식은 사용하기 힘들다. 플러딩 기반의 방식은 분산 구조에 적합하지만 무선망과 이동 노드의 제한된 자원을 고려할 때 오버헤드가 커져, 서비스 명세서(description)를 일부 다른 노드에 저장하는 캐싱 기법[7]이나, 노드들을 일정 기준에 따른 그룹으로 묶어 각 그룹 내에서만 서비스 등록/검색 메시지가 전달되도록 하는[8] 등의 개선 방법들이 제안되었다.

한편, 노드들이 스스로 네트워크를 구성(self-organizing)하고 분산적으로 동작한다는 점에서 MANET과 피어-투-피어(P2P: Peer-to-Peer) 오버레이 네트워크가 유사하다는 점에 착안하여, MANET에서의 자원 검색 및 관리를 위하여 P2P 오버레이 네트워크를 이용하는 개념을 제시한 연구들이 있다[9-14]. P2P 오버레이 네트워크란 물리적인 네트워크의 변화에도 상관없이 동작하기 위해서 응용 계층에 구성하는 네트워크를 말한다. 이러한 오버레이 네트워크는 구조적인(structured) 것과

비구조적인(unstructured) 것으로 나눌 수 있다. 오버레이 네트워크에 정해진 논리적인 토폴로지 없이 플러딩(flooding)을 기반으로 동작하는 것이 비구조적인 P2P 오버레이 네트워크로 누텔라(Gnutella)[15]가 그 대표적인 예이다. 반대로, 구조적인 오버레이 네트워크는 링(ring) 또는 다면체(hypercube) 등의 특정한 구조로 네트워크를 형성하여 메시지가 확률적으로 정해진 홉 수 안에 목적지 노드로 라우팅되는 네트워크로 CAN[16], Chord[17], Pastry[18], Tapestry[19] 등이 있다. 이들은 확장성이 있고 분산적이며, 고장 감내성(fault-tolerance)이 좋고, 노드에게 부하를 균등하게 배분한다는 특징을 갖는다.

비구조적인 오버레이 네트워크에서의 서비스 검색은 서비스를 필요로 하는 노드가 요청 메시지를 전체 네트워크에 브로드캐스팅하고, 요청 메시지를 받은 노드가 해당 서비스를 가진 경우 이에 응답하는 식으로 이루어진다. 이러한 방식은 브로드캐스팅으로 인해 네트워크에 돌아다니는 메시지 수가 많기 때문에 네트워크의 크기가 커질 경우, 성능이 현저히 떨어지게 되는 문제가 있다. 반면, 구조적인 오버레이 네트워크에서는 분산 해쉬 테이블(DHT: Distributed Hash Table) 기법을 사용하는데, 이는 일반 해쉬 테이블을 오버레이 네트워크에 적용한 것이다. 즉, 각 데이터는 키(key)와 값(value) 쌍으로, 전체 네트워크는 커다란 해쉬 테이블로, 네트워크 상의 노드들은 해쉬 테이블의 일부로 표현되며, 각 노드들은 자신이 가지고 있는 데이터를 해쉬하여 키를 구하고, 오버레이 네트워크에서 이 키를 담당하는 노드에 데이터를 저장한다. 데이터를 찾을 때 역시, 찾고자 하는 데이터의 해쉬 키를 이용하여 특정 노드에 요청 메시지를 유니캐스팅하는 방식을 취한다. 따라서 플러딩 방식에 비해 메시지 수를 줄일 수 있다. MANET은 대역폭이 매우 제한적인 환경이라는 것을 고려해 볼 때, 서비스 검색에 구조적인 오버레이 네트워크를 사용하는 것이 적합하다.

그러나 오버레이 네트워크의 논리적 토폴로지는 물리적인 토폴로지를 반영하고 있지 않기 때문에, 오버레이 네트워크 상의 두 노드 사이의 라우팅은 실제 물리적인 최단 거리로의 라우팅을 의미하지 않는다. 즉, 찾고자 하는 서비스가 물리적으로 가까운 곳에 있어도, 오버레이 네트워크 상에서 먼 곳에 있다면, 실제 서비스 검색 요청 메시지는 먼 곳을 돌아 목적지 노드에 도착하게 된다. [20], [21], [22], [23]은 이와 같은 비효율적인 라우팅을 막기 위해서 물리적인 토폴로지를 반영하여 오버레이 네트워크를 구성하거나 패킷을 라우팅할 때 토폴로지 정보를 이용하는 것에 초점을 맞춘 연구들이다.

MANET 환경에서 오버레이 네트워크를 구성하여 서

비스 검색을 할 경우에도 이와 같은 문제를 해결해야 한다. 그러나 MANET은 유선망과는 달리 노드들이 이동성을 갖기 때문에 오버레이 네트워크를 구성하는 과정에서 물리적인 토폴로지를 반영하는 것보다는 패킷을 라우팅하는 시점에서 토폴로지 정보를 이용하는 것이 더 효율적이다. 본 논문에서는 MANET 환경에서 1홉 브로드캐스팅을 통해 물리적으로 가까운 이웃의 정보를 쉽게 얻을 수 있다는 특징을 이용하여 새로운 해결책을 제안한다. 즉, 1홉 브로드캐스팅으로 물리적으로 가까운 노드의 정보와 함께 해당 노드들의 오버레이 네트워크 상의 논리적 이웃 노드의 정보를 얻어서, 메시지를 라우팅 할 때에 이용하는 라우팅 기법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 DHT 기법 및 물리적 토폴로지 정보를 이용한 오버레이 네트워크 라우팅에 대한 연구 및 MANET에서 DHT를 활용한 연구들을 살펴본다. 3장에서는 본 논문에서 제안하고 있는 근접성 기반 오버레이 네트워크 라우팅 기법을 설명하고, 4장에서는 모의실험 결과를 다양한 측면에서 분석한다. 마지막으로 5장에서 본 논문의 결론과 향후 연구 과제에 대해 기술한다.

## 2. 관련 연구

### 2.1 분산 해쉬 테이블을 이용한 P2P 오버레이 네트워크

DHT 기법은 일반 해쉬 테이블을 네트워크에 적용한 개념이다. 즉, 모든 데이터는 키(key)와 값(value) 쌍으로 표현되고, 네트워크 상의 노드들은 전체 해쉬 테이블 중의 일부를 담당하여 전체 네트워크가 커다란 해쉬 테이블로 표현된다. 네트워크 상의 모든 노드들은 자신이 가지고 있는 데이터를 해쉬하여 키(일련의 숫자)를 구하고, 오버레이 네트워크에서 그 키가 속한 구역을 담당하는 노드에게 전달한다. 데이터를 찾을 때 역시 키 값을 이용하여 같은 방식으로 검색 메시지를 전달한다. 이러한 데이터 등록/검색 메시지는 네트워크 상의 특정 노드에 전달되어야 하는데, 이를 위하여 노드들은 물리적인 네트워크의 라우팅 테이블과 별도로 오버레이 네트워크 상에서의 라우팅을 위한 논리적 라우팅 테이블을 두어 논리적 이웃 노드들의 정보를 관리한다. 어떤 노드가 자신이 담당하고 있지 않은 키에 대한 요청을 받으면, 이 라우팅 테이블에 있는 여러 이웃 노드들 중에서 요청 받은 키와 가까이 있는 노드를 선택하여 메시지를 전달한다. 메시지를 전달하는 방법은 DHT 기법마다 다르지만, 일반적으로 현재 노드의 아이디와 요청 받은 키를 이용한다. 대표적인 DHT 기반 P2P 기법은 다음과 같은 것들이 있다.

Plaxton et al. Plaxton et al.[24]은 P2P 시스템에서

사용할 목적으로 개발된 것은 아니었으나, 데이터 검색을 위한 효율적인 라우팅 방법을 제공한다. 알고리즘은 한번에 하나의 숫자를 맞추으로써 동작한다. 만약 아이디가 36278인 노드가 키가 36912인 데이터를 찾아달라는 요청을 받으면, 앞의 두 자리 숫자는 이미 일치하므로, 라우팅 테이블에서 앞의 세 자리가 일치하는 노드(예를 들어 36955번 노드)에게 요청을 전달한다. 이를 위해서 각 노드는 라우팅 테이블에 자신의 아이디와 앞부분의 숫자는 같고 그 다음 번 숫자가 다른 노드들의 정보를 담고 있어야 한다. 예를 들어, 노드 36278번은 접두사가 31, 32, 33, ..., 361, 362, 363, ..., 3691, 3692, ... 등인 노드들의 정보를 라우팅 테이블에 저장하고 있어야 한다. 따라서  $n$ 개의 노드로 이루어진 시스템에서, 각각의 노드는  $O(\log n)$ 의 이웃 정보를 알고 있다. 또한 한번에 하나의 숫자씩 맞추면서 요청이 전달되므로, 라우팅 경로는 최대  $O(\log n)$ 의 응용 계층의 홉을 거치게 된다.

**Tapestry** Tapestry[19]는 정적인 환경을 고려한 Plaxton et al. 알고리즘을 동적인 환경에 맞도록 변형한 것으로,  $O(\log n)$ 의 이웃 정보 및  $O(\log n)$  홉의 라우팅 경로를 갖는다는 성질을 유지한다.

**Pastry** Pastry[18]에서, 노드들은 자신의 아이디와 가까운 키를 갖는 데이터들을 담당한다. 노드는  $L$ 개의 가까운 노드들의 집합으로 이루어진 leaf set  $L$ 을 유지하며, 이를 이용하여 요청 받은 키와 가장 길게 접두사(prefix)가 일치하는 아이디를 갖는 노드에게 메시지를 전달함으로써 라우팅을 수행한다. Pastry 역시  $O(\log n)$ 의 이웃 정보와  $O(\log n)$  홉의 라우팅 경로를 갖는다.

**Chord** Chord[17]는 1차원 링 형의 키 스페이스를 이용한다. 네트워크에 처음으로 참여하는 노드는 모든 키 스페이스를 담당하고, 후에 참여하는 노드는 자신의 주소를 해쉬한 값을 이용하여 키 스페이스의 한 점을 선택하고, 현재 그 점을 담당하는 노드와 그 점을 기준으로 담당할 영역을 나누어 갖는다. 네트워크에 참여한 각각의 노드들은 키 스페이스 상에서 자신의 키를 기준으로 지수 승에 해당하는 키들을 담당하는  $O(\log n)$ 개의 노드들을 관리하는데 이를 핑거 테이블(finger table)이라 한다. 메시지를 받은 노드들은 핑거 테이블을 이용하여 적절한 이웃 노드에게 전달하는데, Chord 역시 메시지가 목적지까지 도착하는데  $O(\log n)$  홉의 라우팅이 필요하다.

**CAN** CAN[16]은 분산 해쉬 테이블을 위하여  $d$ -차원의 다면체 공간을 사용한다.  $d$ 가 2인 경우, 하나의 데이터는 두 개의 해쉬 함수를 이용하여 키( $x, y$ )로 표현되고, 각각의 노드는 임의의 한 점( $x, y$ )를 정하여

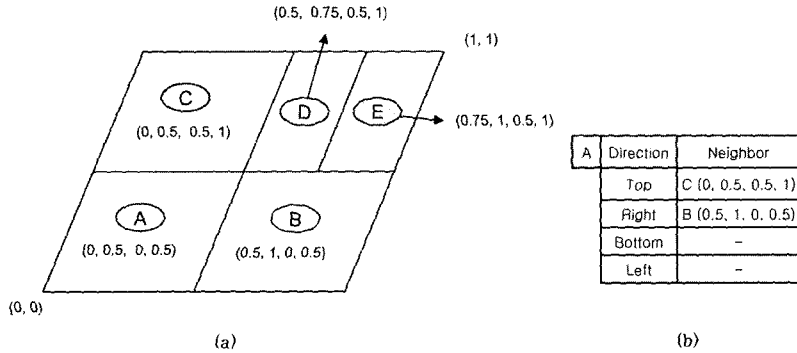


그림 1 5개의 노드로 구성된 2차원 CAN

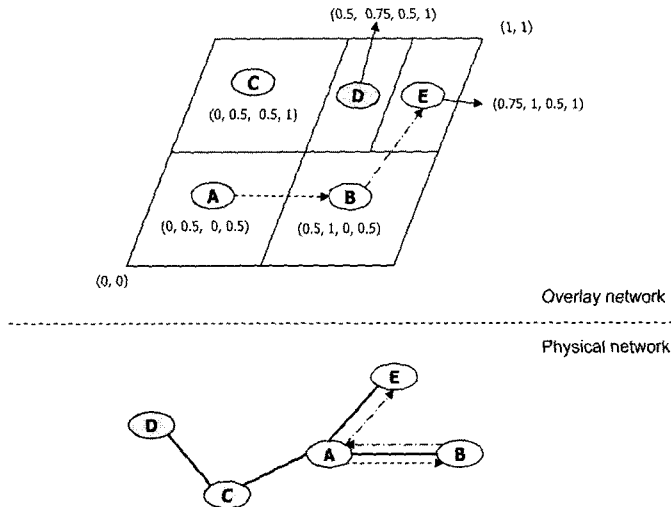


그림 2 CAN 오버레이 네트워크에서의 라우팅

그 점을 담당하고 있는 노드의 담당 구역을 반으로 나누어 담당하게 된다. 그림 1(a)는 CAN에서 2차원 공간을 사용하여 5개의 노드가 오버레이 네트워크를 구성한 것을 보여주고 있다. 만약 새로운 노드가 임의의 점을  $(0.2, 0.4)$ 로 정하고 오버레이 네트워크에 참여하려 하면, 노드 A가 담당하는 구역  $(0.0-0.5, 0.0-0.5)$ 를 반으로 나누어 새로운 노드는  $(0.0-0.25, 0.0-0.5)$ 를 담당하고, 노드 A는  $(0.25-0.5, 0-0.5)$ 를 담당하게 된다. 또한 각 노드는 라우팅 테이블에 자신의 주변 구역을 담당하는 노드들의 정보를 유지한다. 예를 들면, 그림 2(b)는 노드 A의 라우팅 테이블로 자신의 구역에 인접한 노드 C와 노드 B의 정보를 유지한다. 이와 같은 방식으로 각 노드는  $O(d)$ 의 이웃 정보를 갖게 되며, 라우팅 테이블에 있는 정보를 기반으로 메시지를 전송한다. 그림 2에서 노드 A가 키 값이  $(0.8, 0.8)$ 인 데이터를 검색하고자 한다면, 노드 A는 노드 B에게 검색

메시지를 전송하고, 노드 B는 메시지를 받고, 자신의 라우팅 테이블에서  $(0.8, 0.8)$ 에 가까운 노드 E에 다시 전송할 것이다. 노드 E는 자신이 담당하는 구역이 키를 포함하고 있으므로, 노드 A에게 응답 메시지를 보낸다. 이러한 방식으로 자신의 라우팅 테이블의 정보를 이용하여 메시지를 전송하면,  $d$ -차원 공간의 경우 목적지에 도착할 때까지  $O(dn^{1/d})$ 의 중간 노드를 거치게 된다. 일반적으로  $d$ 가 클수록 라우팅 테이블이 커지고 라우팅 경로의 길이는 짧아진다.  $d = \log n$ 인 경우, 다른 알고리즘과 마찬가지로  $O(\log n)$ 의 이웃 정보와  $O(\log n)$ 의 라우팅 경로를 유지하는 셈이다.

### 2.2 토폴로지 정보를 이용한 오버레이 네트워크

P2P 오버레이 네트워크는 응용 계층에서 구성되어, 네트워크 계층의 물리적 토폴로지와는 독립적으로 동작한다. 이 때문에 응용 계층에서 유지하고 있는 정보만으로 패킷을 라우팅하므로 비효율적일 수 있다. 그림 2의

예를 다시 보자. MANET에 참여하고 있는 5개 노드의 물리적인 연결 관계와 응용 계층에 형성된 CAN 오버레이 네트워크를 보여주고 있다. 노드 A가 해쉬 키 (0.8, 0.8)의 데이터를 찾으려고 하는 예에서, 노드 A와 노드 E는 물리적으로 1홉 안에 있음에도 불구하고, 논리적 라우팅 테이블만으로는 목적지가 노드 E임을 모르기 때문에, 물리적 경로는  $A \rightarrow B \rightarrow A \rightarrow E$ 가 된다. 또한 오버레이 네트워크에서는 인접한 노드도 물리적으로는 1홉 이상 떨어져 있으므로, 메시지가 오버레이 네트워크 상에서 여러 노드를 거쳐 전달되는 경우, 물리적인 전달 횟수는 상수 배만큼 커지게 된다. 이러한 비효율성을 극복하기 위하여 오버레이 네트워크를 구성하거나 오버레이 네트워크에서 메시지를 라우팅할 때, 네트워크 계층의 토폴로지 정보를 이용하는 방법에 대한 연구가 구조적 P2P 오버레이 네트워크 시스템의 중요한 주제가 되고 있다. 유선망에서 연구된, 토폴로지를 고려한 오버레이 네트워킹 기법들은 물리적인 토폴로지 정보를 이용하는 방법에 따라 다음과 같이 나눌 수 있다[20].

- 1) **토폴로지 기반 노드 아이디 할당** 이 방법은 오버레이 네트워크에서 노드의 논리적인 아이디 공간을 물리적인 네트워크에 사상하여, 물리적으로 가까운 노드를 논리적으로도 가깝게 만드는 방식으로, CAN과 같이 임의의 노드 아이디를 결정할 수 있는 알고리즘에 사용될 수 있다. 그러나 아이디를 균일하게 사용하지 않게 되어 몇몇 노드에 부하가 치중될 가능성이 있다. 또한, Chord, Pastry, Tapestry 같이 1차원의 아이디 공간을 사용하는 경우에는 잘 동작하지 않는다.
- 2) **근접성을 이용한 이웃 노드 선택** 앞의 방법과 유사하게 토폴로지 정보를 기반으로 하여 오버레이 네트워크를 구성하는 방법이다. 이 방법은 라우팅 테이블에 이웃 노드 정보를 추가할 때, 각 엔트리에 넣을 수 있는 여러 노드 중에서 물리적으로 가까운 노드를 선택함으로써 네트워크 계층의 토폴로지를 반영하여 전송 지연을 낮춘다. Tapestry나 Pastry와 같이 접두사 기반(prefix-based)의 알고리즘에 사용할 수 있는 방법이다. 그러나 이 방법은 CAN이나 Chord와 같이 라우팅 테이블의 각 엔트리가 아이디 공간에서 한 점으로 고정되는 경우 사용할 수 없다는 단점이 있다.
- 3) **계층적 오버레이 네트워크** [21], [23]에서 각 노드들은 물리적으로 가까운 노드끼리 그룹을 형성하여 각 그룹 내에 오버레이 네트워크를 구성하며 그룹을 대표하는 노드 집합을 두고, 그룹 대표들 사이에 또 다시 오버레이 네트워크를 형성한다. 데이터 등록이나 검색 요청은 대표 노드에게 먼저 보내고, 대표 노드는 그룹 간의 오버레이 네트워크에서 요청 받은 키를 담당하는 대표 노드에게 메시지를 전달한다. 목적지 그룹의 대

표 노드가 메시지를 받으면, 자신의 그룹 내부의 오버레이에서 다시 목적지 노드에게 라우팅을 통해 메시지를 전달한다. 이 방법은 그룹 내의 노드들이 물리적으로 근접해 있기 때문에 메시지 라우팅은 효율적이지만, 그룹 멤버 노드들이 모두 대표 노드 정보를 알고 있어야 하고, 모든 메시지가 대표 노드를 거치게 되므로 부하가 치중된다는 문제가 있다.

- 4) **근접성을 이용한 라우팅** 임의의 노드가 메시지를 받으면, 자신이 목적지가 아닌 경우, 라우팅 테이블에서 목적지에 가까운 노드를 찾아 메시지를 전달하게 되는데, 이 때, 다음 홉이 될 수 있는 노드들 중에서 물리적으로 가까운 노드를 선택하여 메시지를 전달하는 방법이다. 각 노드에서  $k$ 개의 가능한 다음 홉의 이웃이 있을 경우, 메시지 전송 지연을  $k$ 개 중 가장 짧은 지연으로 줄일 수 있다. 그러나 실제 프로토콜에서  $k$ 는 상대적으로 작은 수( $O(\log n)$ )이기 때문에 이로 인해 줄일 수 있는 지연이 크지 않으며, 지역적인 정보만을 이용하므로 실제 라우팅 홉 수를 증가시킬 수도 있다. 그럼에도 불구하고, 일반적으로 토폴로지 정보를 이용하지 않는 방법과 비교할 경우, 성능 향상을 볼 수 있다.

이 중, 1), 2), 3)은 오버레이 네트워크를 구성할 때 토폴로지를 이용하는 것이고, 4)는 구성과 관계없이 메시지를 라우팅하는 시점에서 물리적 노드 정보를 참조하는 것이다. MANET은 시간이 지남에 따라 물리적 토폴로지가 변화하므로 오버레이 구성 시에 토폴로지를 고려하는 것은 적합하지 않고, 라우팅하는 시점에서 근접성을 이용하는 개선 방안이 연구되어야 한다.

### 2.3 MANET에서의 DHT 기법 활용

MANET과 P2P 오버레이 네트워크의 유사성에 주목하여 MANET에서 DHT 기법을 활용하는 연구가 최근 활발해지고 있다. 이들은 대부분 인터넷에서의 P2P 응용과 같이 파일과 같은 자원 공유와 관리를 위해 응용 계층에 P2P 오버레이를 구성하고 활용하는 것에 초점을 맞추고 있으며[9-11], 본 논문에서와 같은 서비스 검색을 주 응용으로 하는 연구도 있다[12]. 또한, 응용과 관계 없이 네트워크 성격의 동일성에 기초하여, MANET의 IP 계층 라우팅과 P2P 오버레이 라우팅을 결합시키는 방향의 연구가 있다[13,14]. 이들의 특징을 표 1에 정리하였다.

[9]는 유무선이 혼합된 다양한 네트워크 환경에서 이동 사용자들을 위한 컨텍스트(context) 기반 서비스를 제공하는데 필요한 데이터 관리 기법을 지원하기 위해, P2P와 MANET 기법을 통합한 시스템 아키텍처를 제안하고 그 가능성과 문제점에 대해 고찰하였다. 특히 논리적 오버레이 네트워크에서의 라우팅과 물리적 네트워

표 1 MANET에서 DHT를 활용하는 연구 사례

	주요 쟁점	기반 구조	응용
[9]	MANET에 P2P를 결합한 아키텍처 제안 및 해결해야 할 문제점 제시	P Grid	자원 공유
[10]	P2P 오버레이에 IP 라우팅 활용	플러딩	파일 공유
[11]	Fault-tolerant routing	Chord	파일 공유
[12]	오버레이 구조 유지 오버헤드 감소	Lanes	서비스 검색
[13]	IP 계층 라우팅에 P2P 활용	Pastry	IP 라우팅
[14]	파일 공유 프로토콜 모델 제안/분석	추상적	파일 공유
Ours	Topology-aware routing	CAN	서비스 검색

크 환경에서의 라우팅 간의 격차를 해결해야 할 큰 문제로 지적하고 있다. MANET 환경에서 고려해야 할 또다른 문제는 노드들의 움직임으로 인해 네트워크 구조가 자주 변하고 노드들이 갑자기 사라져 엄격한 구조를 유지하고 그에 따라 라우팅해야 하는 DHT 기반 오버레이에서 라우팅이 실패할 수 있다는 점이다. [11]에서는 Chord 구조를 MANET에 적용할 때, MANET의 특성으로 인해 노드가 전파 영역을 벗어나거나 갑자기 전원이 소진되었을 때 등의 네트워크 불량 상황에서도 라우팅이 이루어질 수 있도록 Backtracking Chord와 Redundant Chord 알고리즘을 제안하였다.

MANET에서 DHT를 사용하고자 할 때 또다른 큰 문제는 오버레이 네트워크 구조를 구성하고 유지하기 위한 오버헤드임을 지적한 연구가 있다[12]. [12]에서는 오버레이 구조를 유지하기 위한 엄격한 조건을 완화시킨 Lanes라는 구조를 제안하였다. Lanes는 여러 개의 노드를 하나의 그룹으로 묶고 각 그룹에 해쉬 테이블의 키 영역을 배정하였다. 서비스 명세서들은 같은 그룹 내의 노드들에 복사되어 캐시에 저장되고, 검색은 IPv6의 anycast 기능을 사용하는 것으로 가정하였다. 1차원 DHT와 그룹 기반의 캐싱으로 메시지 오버헤드를 줄이고, 검색 성능 향상을 꾀하였으며, 일반적으로 MANET에서 그룹 또는 계층적 구조를 사용할 때의 단점인 대표 노드에게 부하가 집중되는 것 또한 anycast에 의해 방지될 수 있다.

한편, MANET과 P2P 오버레이 네트워크의 성격이 비슷하며, 모든 노드가 메시지를 중개하는 라우터로서 역할을 하는 특성에 기초하여 두 계층의 라우팅을 하나로 결합하거나 오버레이 라우팅에 MANET 라우팅을 이용하는 것도 중요한 주제로 제시되고 있다. [14]에서는 P2P 파일 공유를 위해 MANET에서 가능한 응용 프로토콜을 브로드캐스트 방식과 DHT 방식으로 구분하고, 그를 다시 IP 라우팅과 결합했을 때와 별도로 작동했을 때로 나누어, 각 경우에 예상되는 복잡도와 라우팅 거리를 매우 간단한 모델을 이용해 분석하였다. DHT 기반 프로토콜은 브로드캐스트에 비해 구조의 유

지와 구현이 좀 더 복잡하지만 확장성이 좋고 에너지 효율이 더 좋은 것으로 나타났으며, 두 방식 모두 IP 계층 프로토콜과 결합 시에 보다 좋은 성능을 보일 수 있다고 한다. 그러나, 분석에 사용된 모델과 기법이 지나치게 단순화되어 있고 구체적인 알고리즘이 제시되어 있지 않아 보다 실제적인 구현과 분석이 필요하다. MANET에서의 파일 공유를 위해 [10]에서 제안한 라우팅 알고리즘은 DHT 기반은 아니지만 P2P 오버레이와 IP 계층 라우팅을 결합시킨 실제적인 연구 결과이다. P2P 오버레이를 미리 구성하는 것이 아니라 요구형(on-demand) 방식으로 필요할 때마다 구성하고 라우팅하는 방법을 제시하였다. PDI와 ORION 두 가지의 검색 알고리즘을 제안하였는데, 쿼리를 전체에 플러딩하지 않고 캐싱하는 방법과 MANET의 IP 계층 라우팅 정보를 이용하는 방법으로 성능 향상을 꾀하였다. [13]에서는 잘 알려진 MANET 라우팅 알고리즘인 DSR과 응용 계층의 P2P 프로토콜인 Pastry를 결합한 IP 계층 라우팅 알고리즘을 제안하였다.

### 3. 근접성 기반 오버레이 네트워크 라우팅

#### 3.1 MANET 특성을 고려한 오버레이 라우팅의 개선 방안

2장에서 설명된 관련 연구로부터 MANET에 DHT 기반의 오버레이 네트워크를 활용하는 데는 1) 논리적 라우팅과 물리적 라우팅의 격차 해소, 2) 오버레이 네트워크 구조를 유지하는 오버헤드의 최소화, 3) 구조 결합이 발생했을 때의 검색 기능 유지, 4) IP 계층 라우팅과의 효과적인 결합 등이 주요 쟁점이 되고 있음을 알 수 있다. 본 논문에서는 논리적 라우팅과 물리적 라우팅의 격차를 줄여 메시지 수와 라우팅 경로의 길이를 줄이는 것을 주요 목표로 한다. 또한, 오버레이 네트워크 구조를 유지하는 오버헤드를 줄이기 위해 기존 DHT 기법 중, 이웃 노드와의 통신량이 상대적으로 적은 구조를 채택하여 최적화하기로 한다.

2.2절에 소개된 토폴로지 정보를 이용하는 개선 방안들은 대부분 무선망 또는 이동 노드들에 대해 사용하기

어렵다. 오버레이 구성 시에 토폴로지를 기반으로 아이디를 할당하거나 근접성을 이용하여 이웃 노드를 선택하는 것은 MANET의 물리적 토폴로지가 계속해서 변화하므로 시간이 갈수록 효율이 떨어지고, 계층적 오버레이 역시 특정 노드들에 부하가 집중되므로 전력 등 자원이 제한적인 이동 노드들에게는 적합하지 않다. 따라서, MANET에서 오버레이 네트워크를 이용할 때 토폴로지 정보를 이용할 수 있는 방법은 실제 패킷을 라우팅할 때, 이웃 노드들 중에서 근접성을 이용하여 가까운 노드를 선택하는 방법이다. 그러나, 2.2절에 설명된 방식은 제한된 수의 논리적 이웃 중에서 물리적으로 가까운 노드를 선택하는 것으로 성능 향상에 한계가 있다. 본 논문에서는 MANET의 특징적인 1홉 브로드캐스트를 활용하여, 물리적 이웃을 먼저 참조하고 그 중 논리적으로 가까운 노드를 선택하는 방식의 근접성 기반 라우팅을 제안한다.

유선망에서의 1홉 브로드캐스트는 서브넷(subnet)의 모든 노드를 대상으로 하고, 라우팅은 물리적 이웃 노드와 상관없이 정해진 라우터를 거치게 된다. 그러나 MANET에서 각 노드는 메시지의 송신/수신자로서의 역할 뿐만 아니라 라우터 역할을 수행하여 멀티홉 통신이 이루어지며, MAC 프로토콜 및 IP 계층 라우팅에서 1홉 브로드캐스트를 이용해 자신의 통신 영역(transmission range) 내에 위치한 물리적 이웃 노드들을 검색하고 서로 주기적으로 메시지를 주고 받는 것이 필수적이다. 즉, MANET에서의 1홉 브로드캐스트는 여러 개의 라우터에 동시에 응용 계층의 메시지를 보내 정보를 주고 받을 수 있음을 의미한다. 본 논문에서 제안하는 근접성 기반 오버레이 네트워크 라우팅은 이러한 MANET 고유의 1홉 브로드캐스트를 활용한다. 더불어, 서비스 검색 프로토콜의 특성상 각 노드가 자신의 서비스 명세서를 네트워크 상의 특정 노드에 등록함으로써 하나의 서비스에 대해서 두 개의 노드가 알고 있다는 점을 이용하는 개선 방안도 제시한다.

### 3.2 DHT 기법을 활용하는 서비스 검색

MANET 환경에서 노드들은 유선망에서와 마찬가지로 다른 노드와 서비스를 공유할 수 있어야 한다. 즉 자신이 가지고 있는 서비스를 제공하고, 다른 노드들이 가지고 있는 서비스를 사용할 수 있어야 한다. 2.1절에서 설명한 DHT를 MANET의 서비스 검색에 이용하는 방식은 다음과 같다.

각 노드는 자신이 제공하는 서비스 및 서비스 명세서 목록과 DHT에서 자신이 맡은 구역에 해당하는 외부 서비스의 명세서 목록을 관리한다. 이를 위해 네트워크에 참여한 노드들은 자신이 가지고 있는 서비스의 이름을 해쉬함으로써 키 값을 구하고, DHT에서 그 키 값을 담

당하는 노드에 서비스 명세서를 등록한다. 서비스 명세서에는 서비스에 대한 정보와 더불어 서비스를 제공하는 노드에 대한 정보를 포함하고 있다. 이렇게 크기가 큰 실제 서비스 대신 서비스 명세서를 등록함으로써 네트워크에 돌아다니는 메시지 수를 줄일 수 있다. 노드들은 필요한 서비스가 있을 때 역시 해쉬 키를 이용하여 키를 담당하고 있는 특정 노드에 서비스 명세서를 요청하는 메시지를 보낸다. 요청에 대한 응답으로 하나 이상의 서비스 명세서를 받으면, 그 중에서 가장 적합한 서비스를 선택하여 그 서비스 명세서에 기술되어 있는 서비스 제공 노드 정보를 이용하여 서비스 제공자에게 서비스를 요청함으로써 필요한 서비스를 받을 수 있게 된다. 그림 3(b)에서 노드 *B*와 *D*는 각각 MP3 서비스와 날씨(weather) 서비스를 제공한다. 노드 *B*와 *D*는 그림 3(a)의 해쉬 키 생성기를 통해 키를 구하여 키가 속한 구역을 담당하고 있는 노드 *A*와 *E*에 서비스 명세서를 등록한다. 노드들이 서비스를 등록하고 검색하는 과정에서 주고 받는 메시지들은 오버레이 네트워크에서 사용하는 DHT 알고리즘에서의 라우팅 기법에 의해 전송된다.

본 논문이 제시하는 근접성 기반 오버레이 라우팅 기법은 CAN의 라우팅 방법을 토대로 한다. MANET은 무선망이기 때문에 유선망과 달리 각 노드가 사용할 수 있는 평균 대역폭이 제한적이다. 따라서 전체 네트워크의 성능을 향상시키기 위해서는 전송 메시지 수를 최소화하는 것이 우선적으로 요구되는데, [12]의 연구에서 지적된 바와 같이 DHT 구조를 유지하며 라우팅 테이블 관리를 위한 주기적인 통신의 수를 줄이는 것이 가장 중요하다. 2.1절에 소개된 주요 DHT 기법 중, CAN은  $O(d)$ 의 이웃 노드 정보를 유지하는 반면, 다른 알고리즘들은  $O(\log n)$ 의 라우팅 테이블 엔트리를 유지해야 한다. 즉  $d$ 를 작은 수로 유지한다면 CAN은 적은 양의 이웃 노드 정보만을 관리하며 주기적인 메시지 전송 횟수를 고정시키는 것에 비해, 다른 알고리즘들은 네트워크에 참여한 노드 수가 많아질수록 주기적인 메시지 전송이 많아진다. 그러한 이유로, 본 논문에서는 MANET 상에서 오버레이 네트워크를 구성하기 위하여 CAN 구조를 이용한다. CAN 이외에도, 라우팅 테이블을 이용하여 메시지를 전송하며 목적지까지의 거리를 좁혀가는 모든 DHT 기법에 대해 이 기법을 적용하여 라우팅 성능을 향상시킬 수 있다.

### 3.3 근접성 기반 오버레이 네트워크 라우팅 알고리즘

근접성 기반 오버레이 네트워크 라우팅 기법은 MANET에서 서비스 검색을 위해 오버레이 네트워크를 형성하고 DHT 구조를 이용할 때, 물리적인 토폴로지를 반영하여 효율적으로 메시지를 라우팅하는 방법이다. DHT 시스템의 기본 라우팅 기법을 다음 두 가지 방법

으로 수정하여 전체 라우팅 경로를 줄인다. 그 첫 번째는 물리적인 토폴로지 정보를 이용하여 논리적 홉 수를 줄이는 것이고, 두 번째는 라우팅 경로 상에 있는 실제 서비스 제공자로부터 직접 서비스 명세서를 얻는 것이다.

이들 방법들 각각을 설명하기에 앞서, 본 논문에서 대표적인 DHT 구조로 선택한 CAN이 제시하는 기본 라우팅 기법이 어떻게 동작하는지 간략히 정리해보자. 오버레이 네트워크에서 CAN을 구성하고 있는 모든 노드들은 필요한 서비스를 찾기 위해서, 서비스를 해쉬하여 키를 구하고, 그 키를 담아서 서비스 검색 메시지를 보낸다. 검색 메시지를 보내는 노드와 이 메시지를 받는 노드들은 자신이 담당하는 구역이 해당 키를 포함하지 않는 경우, 라우팅 테이블에 유지하고 있는 이웃 노드들 중에서 그 키에 가까운 구역을 담당하고 있는 노드에 메시지를 전달한다. CAN이 제안하는 기본적인 라우팅 기법은 이와 같은 방식으로 한 홉씩 메시지를 전달할

때마다 요청한 키와의 거리를 좁혀가서 결국 주어진 키가 속한 구역을 담당하는 노드까지 전달한다. 여기서 이용하는 라우팅 테이블은 네트워크 계층에서 사용되는 라우팅 테이블과는 독립적으로, 응용 계층에서 해쉬 값을 근거로 계산되는 논리적인 근접성에 의해 관리된다. 특별한 설명이 없다면, 앞으로 언급되는 라우팅 테이블은 모두 논리적 라우팅 테이블을 의미한다. 각 노드가 라우팅 테이블에 유지하는 정보는 자신이 담당하고 있는 구역에 인접한 구역을 담당하는 노드들(이하 논리 이웃 노드)의 주소와 그 구역의 위치 정보이다(그림 1(b) 참고).

3.3.1 물리적 이웃 노드 참조

기존 CAN의 라우팅 방법에서 각 노드는 라우팅 테이블에 기반하여 다음 홉을 결정하는 반면, 근접성 기반 오버레이 네트워크 라우팅 기법에서는 메시지를 전달해야 하는 노드들이 라우팅 테이블을 참고하기에 앞서, 물

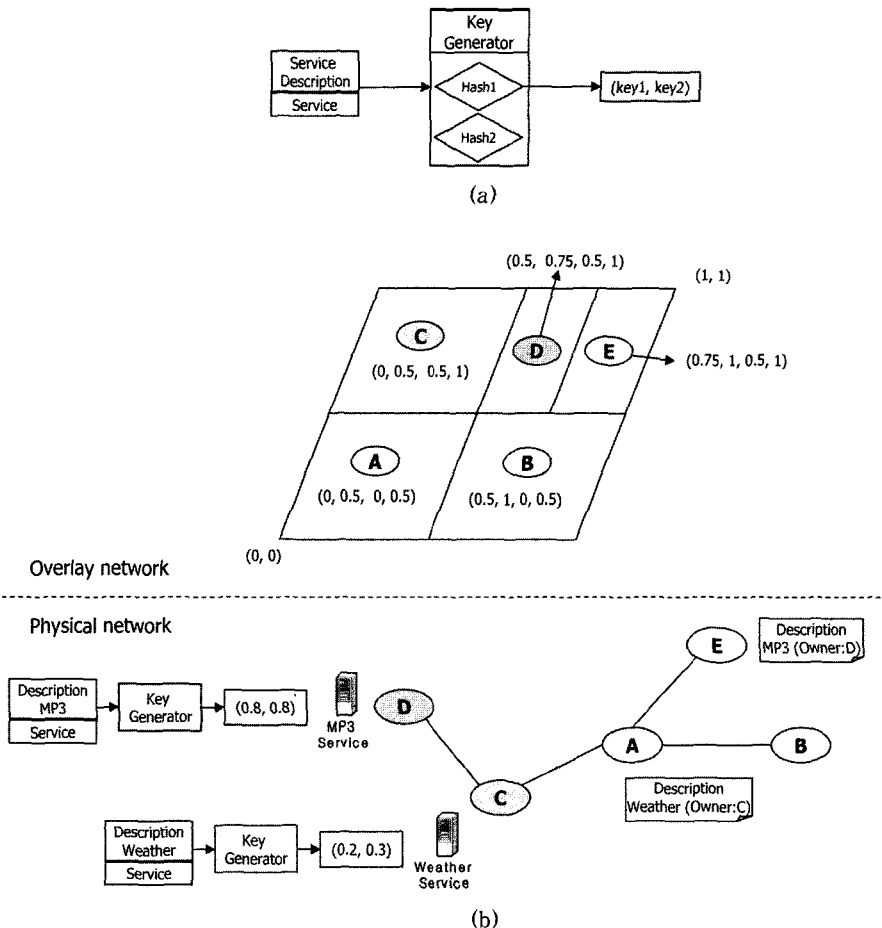


그림 3 서비스 검색 시나리오



리적인 이웃 노드들의 정보를 이용한다. 네트워크 계층에서 물리적으로 가까운 노드들 중, 응용 계층에서 논리적으로 목적지에 가장 가까운 노드에게 메시지를 전달함으로써, 오버레이 네트워크 상에서 논리적인 홉 수를 줄인다는 것이 기본 아이디어이다. 논리적인 홉 수를 줄이는 것은 결과적으로 감소된 논리적 홉 수의 상수 배에 해당하는 물리적인 홉 수를 줄이는 효과가 있다.

근접성 기반 오버레이 네트워크 라우팅 기법에서 노드들이 물리적 이웃 노드의 정보를 모으고 이를 이용하는 방법은 다음과 같다. 네트워크 상의 한 노드가 서비스 검색 메시지를 받으면, 가장 먼저 자신의 구역이 요청된 키를 포함하고 있는지 살핀다. 만약 포함하고 있다면, 요청자에게 유니캐스팅으로 응답 메시지를 보낸다. 그렇지 않다면, 물리적으로 가까운 노드의 정보를 얻기 위해, 서비스 검색 메시지 내의 정보를 이용하여 근접 노드 요청 메시지(close\_node\_request)를 만든다. 근접 노드 요청 메시지에는 요청 받은 서비스의 키와 오버레이 네트워크 상에서 목적지와 자기가 맡은 구역 사이의 논리적인 거리(distance) 및 자신의 주소를 넣고 1홉 브로드캐스트한다. 이 메시지를 받은 노드들은 자기가 담당할 구역과 서비스 키 사이의 논리적 거리를 구한다. 이 거리가 메시지 안의 거리보다 일정 크기(threshold) 이상 작은 경우에 자신의 주소와 거리를 담아서 근접

노드 응답 메시지(close\_node\_response)를 보낸다. 근접 노드 요청 메시지를 보낸 노드는 일정 시간 동안 돌아오는 응답 메시지를 모으고, 그 중에서 거리가 가장 가까운 노드에게 서비스 검색 메시지를 전달한다. 한편, 각 노드들은 라우팅 테이블에 논리적인 이웃 정보를 관리하고 있으므로, 근접 노드 요청 메시지를 받았을 때 목적지에 해당하는 키가 자신의 논리적 이웃 노드의 구역에 속하였다면 이웃 노드의 주소를 담아 응답을 보냄으로써 논리적인 한 홉을 줄일 수 있다. 즉, 메시지를 다음 홉으로 전송하기 위해서, 각 노드는 자신의 1홉 이웃 노드와 이웃 노드의 논리적 이웃 노드까지 살펴서 다음 홉의 노드를 선택한다.

그림 4는 그림 3에서 보인 사례에서 근접성 기반 오버레이 네트워크 라우팅 기법으로 물리적 이웃 노드 정보를 이용하는 방법을 나타낸 것이다. 노드 A는 MP3 서비스를 찾기 위해서 이를 해쉬한 키 값 (0.8, 0.8)을 생성하고, 서비스 검색 메시지를 보내기에 앞서, 서비스 이름(MP3), 키 값((0.8, 0.8)), 키와 자기가 담당하는 구역 사이의 논리적 거리(0.424)를 기록하여 근접 노드 요청 메시지를 1홉 브로드캐스팅 한다(그림 4(a)). 이 메시지를 받은 노드 E는 자신의 구역에 키가 포함되므로 근접 노드 응답 메시지에 자기 주소와 논리적 거리 0을 넣어 노드 A에게 보낸다. 한편 노드 C는 키까지의 논리

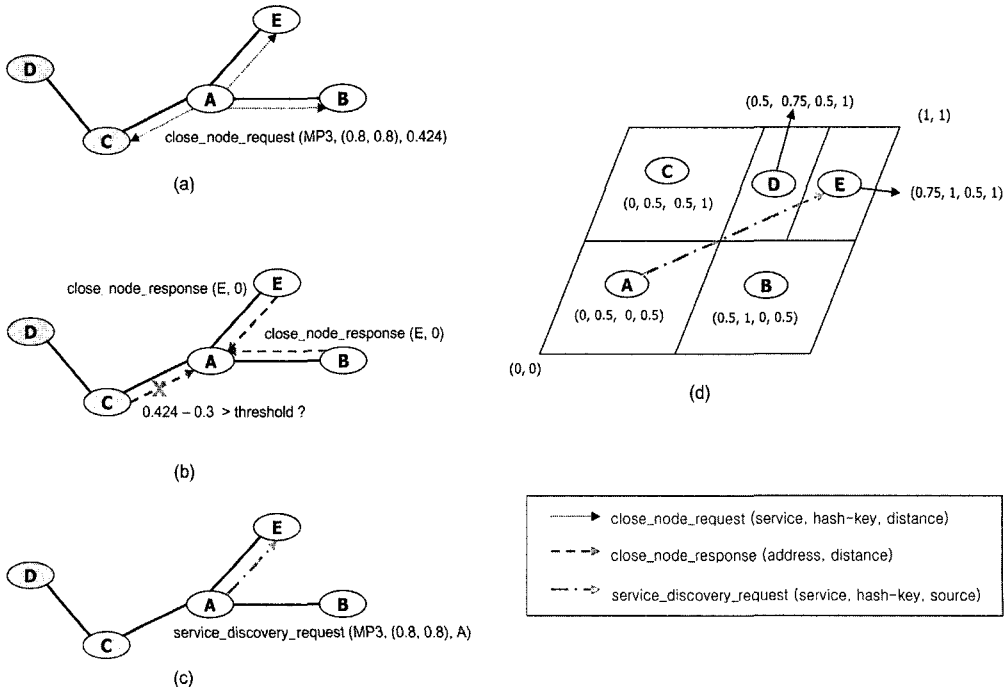


그림 4 근접성 기반 오버레이 네트워크 라우팅 기법

적 거리가 0.3으로 0.424와 비교하여 threshold (여기에서 0.2) 이상만큼 작지 않으므로, 응답을 보내지 않는다. 노드 B 역시 키와의 논리적 거리가 0.3이지만, 자신의 논리적 이웃 노드 중에서 노드 E가 키를 포함하므로 자신의 정보가 아닌 노드 E의 정보를 넣어 근접 노드 응답 메시지를 보낸다(그림 4(b)). 일정 시간 후, 노드 A는 자신이 받은 응답 메시지 중에서 키 값과의 거리가 가장 작은 노드인 E에게 서비스 검색 메시지를 전달한다(그림 4(c)). 노드 E는 자신이 키 값을 포함하고 있으므로, 자신의 서비스 목록에서 MP3 서비스를 찾아서 서비스 명세서를 노드 A에게 보내준다. 그림 4(d)는 근접성 기반 오버레이 라우팅 기법을 사용하였을 때, 오버레이 네트워크에서 실제 경유한 노드를 나타내고 있다. CAN의 라우팅 기법을 따르면, A→B→E를 거치게 되는 반면, 여기서는 A→E로 직접 메시지가 전송되므로 라우팅 효율을 높일 수 있다.

그림 5의 (a), (b)는 각각 같은 물리적 토폴로지를 갖는 네트워크에서 노드 2가 해쉬 키가 (0.8, 0.2)인 서비스를 검색하는 과정에서, CAN이 제시하는 기본 라우팅 기법을 사용할 때와 근접성 기반 오버레이 네트워크 라우팅 기법을 사용할 때, 서비스 검색 메시지가 전달되는 경로를 물리적인 네트워크 및 오버레이 네트워크 관점에서 나타낸 것이다. CAN의 라우팅 방법을 이용한 경우(그림 5(a)), 오버레이 네트워크에서 2→3→4→10을, 물리적 네트워크에서 2→11→3→4→8→10을 경유한 반

면, 근접성 기반 오버레이 네트워크 라우팅 기법을 이용한 경우(그림 5(b)), 오버레이 네트워크에서 2→10으로 2홉을 줄이고, 물리적 네트워크에서 2→11→10을 경유하여 3홉을 줄인다. 특히, 네트워크가 커질수록 오버레이 네트워크에서 경유하는 홉 수가 많아지게 되는 문제를 근접성 기반 오버레이 네트워크 라우팅 기법으로 해결할 수 있다.

3.3.2 물리적으로 가까운 서비스 명세서 이용

서비스 검색을 위해 DHT를 이용할 때, 오버레이 네트워크 상의 노드들은 자신이 갖고 있는 서비스의 명세서를 전체 키 공간 중 서비스 키가 속한 구역을 담당하는 특정 노드에 등록하여 검색을 용이하게 한다. 따라서 DHT 시스템에서는 특별한 캐쉬 메커니즘을 사용하지 않고도, 하나의 서비스 명세서가 두 개의 노드에 저장된다. 그러므로 서비스 검색 메시지가 요청자에서부터 목적지에 해당하는 노드까지 전송되면서 경유하게 되는 노드들 중에 실제 서비스 제공자가 포함되어 있을 수 있다. 이러한 특성을 이용하여, 만약 서비스 제공자가 자신이 가지고 있는 서비스를 요청하는 메시지를 받으면 다음 홉으로 전달하는 대신, 요청자에게 바로 응답을 해줌으로써 라우팅 경로를 줄일 수 있다.

CAN의 경우, 각 노드는 서비스 검색 요청 메시지를 받았을 때 자신이 제공하는 서비스 명세서 목록에 요청 받은 서비스가 포함되어 있는지를 확인한다. 반면, 근접성 기반 오버레이 네트워크 라우팅 기법을 사용하는 경

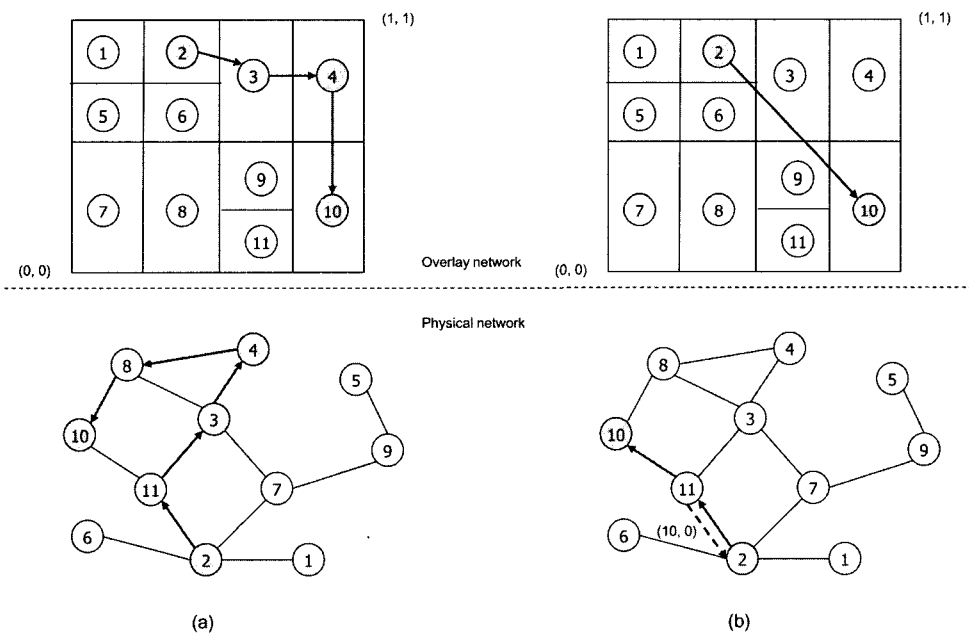


그림 5 물리적 이웃 노드 정보를 이용한 성능 향상

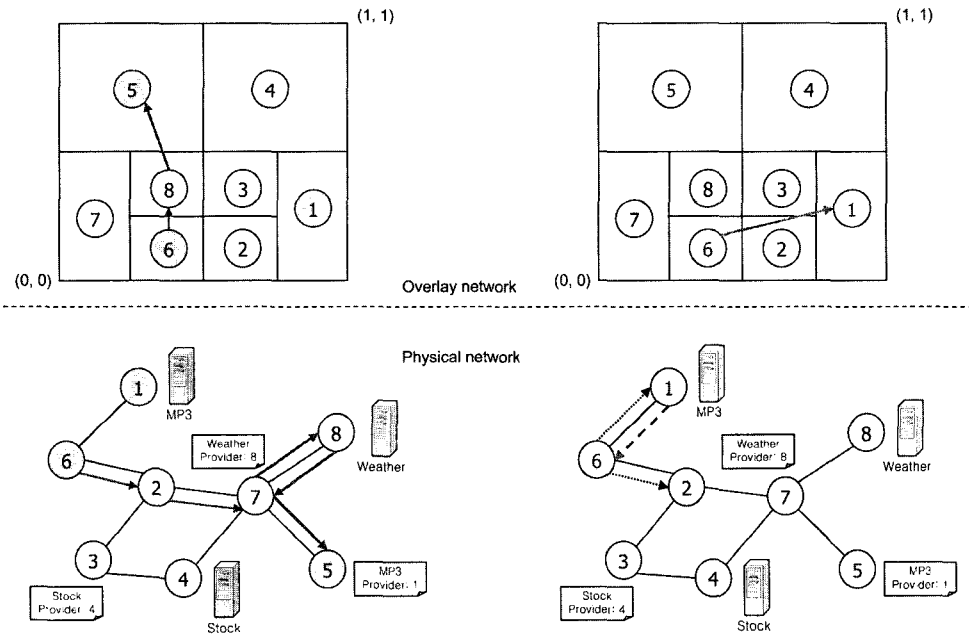


그림 6 물리적으로 가까운 서비스 명세서를 이용한 성능 향상

우, 각 노드들이 서비스 검색 메시지를 라우팅 하는 과정에서 1홉 브로드캐스팅으로 보내게 되는 근접 노드 요청 메시지를 받았을 때, 자신이 제공하는 서비스 명세서 목록에서 요청 받은 서비스를 찾는다. 전체 노드 수가  $N$ 인 오버레이 네트워크에서, CAN을 사용하는 경우 다음 홉에 실제 서비스의 제공자를 만날 확률은  $1/N$ 이다. 반면, 근접성 기반 오버레이 라우팅 기법을 사용하는 경우 노드들의 평균 물리적 이웃 노드 수가  $k$ 개라면, 서비스의 제공자를 만날 확률은  $k/N$ 이다. 따라서, 물리적인 이웃 노드 수( $k$ )가 많아질 수록 서비스 검색 메시지가 이동하는 경로가 줄어들게 된다. 그림 6은 서비스 제공자가 직접 근접 노드 요청 메시지에 응답함으로써 라우팅 경로가 줄어들어 주는 예를 보여준다. 같은 토폴로지를 갖는 네트워크에서 노드 1, 4, 8은 각각 MP3, Weather, Stock 서비스를 제공하고, 자신의 서비스에 대한 설명서를 노드 5, 3, 7에 각각 등록하였다. 이 때, 노드 6이 MP3 서비스를 검색하기 위해 CAN 라우팅 기법을 사용한다면, 검색 메시지는 6→2→7→8→7→5를 차례로 거치게 되는 반면, 근접성 기반 오버레이 네트워크 라우팅 기법을 사용한다면, 노드 6은 근접 노드 요청 메시지를 1홉 브로드캐스트하고, 이를 받은 MP3 서비스 제공 노드 1이 응답을 보내주어 여러 홉을 거치지 않고 성공적으로 서비스를 검색하게 된다.

#### 4. 실험 및 결과

#### 4.1 실험 환경

본 연구에서는 Carnegie Mellon 대학교에서 개발한 무선 환경이 추가된 ns-2[25] 시뮬레이터를 사용하여 근접성 기반 오버레이 네트워크 라우팅 기법의 성능을 평가하였다. MAC 프로토콜은 IEEE 802.11[26]을 사용하였고, 이 때 각 노드의 전송 범위는 250m로 하였다. 네트워크 계층에서의 라우팅 프로토콜로는 AODV(Ad-hoc On Demand Distance Vector)[27]를, 전송 계층 프로토콜로는 MANET에 적합한 UDP를 사용하였다.

ns-2에는 MANET에서의 하부 통신 프로토콜이 비교적 잘 구현되어 있지만 응용 계층의 프로토콜은 구현되지 않은 부분이 많다. 본 연구의 실험을 위하여 ns-2 시뮬레이터 내에 서비스 검색을 위한 응용과 CAN 오버레이 네트워크 구성 및 검색 프로토콜, 그리고 본 논문에서 제안하는 개선된 라우팅 기법을 각각 구현하였다. 또한, 비교를 위해 플러딩 기반의 서비스 검색 프로토콜 또한 구현하였다. 구현된 응용은 자신의 서비스를 네트워크 상의 특정 노드에 등록하거나 필요한 서비스를 검색하는 등 서비스와 관련된 전반적인 동작과 사용자 인터페이스를 담당한다.

실험 시나리오는 다음과 같다. 노드들은 30초 간격으로 오버레이 네트워크에 참여(join)하여 CAN DHT의 일부가 된다. 그 이후, 노드들은 자신의 논리적 이웃들과 주기적으로 통신하며 오버레이 라우팅 테이블을 구성하거나 수정한다. 모든 노드가 네트워크에 참여한 후

에, 전체 노드들 중 약 50 퍼센트의 노드가 60초 간격으로 임의의 3개의 서비스를 네트워크 상의 특정 노드에 등록하기 시작하고, 주기적으로 재등록을 시행한다. 노드들 사이의 통신이나 서비스 재등록을 위한 주기는 네트워크의 상황에 따라 유동적으로 변하며, 적어도 10분에 한 번씩은 통신하도록 하였다. 서비스를 등록하는 마지막 노드가 등록을 시작하고 200초 후부터, 전체 노드들 중 약 30 퍼센트의 노드가 50초 간격으로, 임의의 10개의 서비스를 검색을 시작한다. 서비스를 검색하는 마지막 노드가 검색을 시작한 후, 300초 후에 실험을 종료한다.

각 노드가 송신 또는 수신 모드일 때의 동작이 그림

7과 그림 8의 알고리즘에 기술되어 있다.

실험을 통해서 측정하고자 한 측정값(metric)은 표 2에 나타냈으며, 플러딩을 기반으로 하는 검색 방법(이하 플러딩 기법)과, CAN 오버레이 라우팅 기법(이하 CAN 기법)을 비교군으로 설정하였다. 실험은 세 가지 방법으로 시행하였다. 첫째는, 1500x1500 크기의 공간에서 노드 수를 25개부터 150개까지 증가시키며 측정하였다. 둘째로, 노드 수를 100개로 고정시키고, 전체 공간의 크기를 1000x1000에서부터 2250x2250까지 증가시켜 1홉 이웃 노드 수를 변화시키며 실험하였으며, 마지막으로 1500x1500 크기의 공간에서 100개의 노드들의 이동 속도를 증가시키며 실험하였다.

```

void retrieve_svc(char *service_name) {
    ...
    service_key = make_service_key(service_name); // 서비스 이름으로 서비스 키를 구한다.
    if (zone.is_contain(key) || lookup_my_service(service_name)) { // 자신의 구역이 키를 포함하거나, 서비스 제공자면,
        retrieval_success(); // 서비스 검색 완료
    } else if (neighbor = zone.is_neighbor_contain(key) >= 0) { // 자신의 논리적 이웃 노드가 key를 담당하면,
        forward_to(neighbor, msg); // 그 노드에게 바로 서비스 검색 요청 메시지를 전송.
    } else {
        logical_distance = zone.distance(service_key); // 키와 자기 구역사이의 논리적 거리를 구한다.
        send_close_node_request(service_name, service_key, logical_distance); // 근접 노드 요청 메시지를 브로드캐스팅한다.
        set_broadcast_timer(); // 브로드캐스트 타이머 설정
    }
    ...
}

void broadcast_timer_expire(Event *) { // 근접 노드 요청 메시지를 보내고 일정 시간이 지나면,
    ...
    closest = select_close_node(); // 받은 응답 메시지 중에서 가장 거리가 작은 노드를 찾아서
    forward_to(closest, msg); // 그 노드에게 서비스 검색 요청 메시지 전송.
    ...
}

```

그림 7 송신측 알고리즘

```

void recv_msg(int size, char *msg){
    ...
    switch(msg_type) {
        case SERVICE_RETRIEVAL_REQUEST: // 서비스 검색 요청 메시지를 받은 경우,
            if (zone.is_contain(key) || lookup_my_service(service_name)) { // 자신의 구역이 키를 포함하거나, 서비스 제공자면,
                send_retrieval_response(service_name, service_description); // 서비스 검색 응답 메시지를 전송.
            } else if (neighbor = zone.is_neighbor_contain(key) >= 0) { // 자신의 논리적 이웃 노드가 key를 담당하면,
                forward_to(neighbor, msg); // 그 노드에게 바로 서비스 검색 요청 메시지를 전송.
            } else {
                // 근접 노드 요청 메시지를 브로드캐스팅한다.
                send_close_node_request(service_name, service_key, logical_distance);
            }
            break;
        case CLOSE_NODE_REQUEST: // 근접 노드 요청 메시지를 받은 경우,
            if (lookup_my_service(service_name)) { // 자신이 서비스 제공자이면,
                send_retrieval_response(service_name, service_description); // 서비스 검색 응답 메시지를 요청자에게 직접 전송
            } else if (neighbor = zone.is_neighbor_contain(key) >= 0) { // 자신의 논리적 이웃 노드가 key를 담당하면,
                send_close_node_response(service_name, neighbor, 0); // 근접 노드 응답 메시지에 노드와 거리 0을 넣어 전송.
            } else {
                my_distance = zone.distance(key); // 자신과 key 사이의 논리적인 거리를 구하여
                if (my_distance < msg->logical_distance) { // 그 거리가 요청자보다 threshold 이상 작으면
                    send_close_node_response(service_name, address, my_distance); // 근접 노드 응답 메시지 전송.
                }
            }
            break;
        case CLOSE_NODE_RESPONSE: // 근접 노드 응답 메시지를 받은 경우,
            add_res(msg); // 일정 시간 동안 오는 응답 메시지를 모은다.
            break;
    }
    ...
}

```

그림 8 수신측 알고리즘

4.2 실험 결과

4.2.1 노드 개수 변화에 따른 결과

그림 9는 노드 개수를 변화시키며 서비스 검색을 수행했을 때, 검색 메시지가 서비스 명세서를 가진 노드에 전달될 때까지 거친 물리적인 홉 수를 나타낸 것이다. 이 때, 임의의 두 노드는 최대 7홉 떨어져 있다. 오버레이 네트워크를 사용하는 경우에는, 노드 수가 증가할수록 DHT가 잘게 쪼개져서 거처야 하는 노드 수가 많아지게 된다. CAN 기법은 물리적인 토폴로지를 고려하지 않으므로, 오버레이 네트워크 상의 홉 수가 증가할수록 물리적인 네트워크 상에서의 홉 수도 상수 배 증가하게 되는 반면, 근접성 기반 오버레이 네트워크 라우팅 기법은 토폴로지 정보를 이용하여 최소의 홉 수를 거치게 되는 플러딩 기법과 유사한 결과를 내는 것을 볼 수 있다.

그림 10은 노드 수의 증가에 따른 메시지 오버헤드를 측정한 것으로, (a)는 각 노드가 받은 메시지의 수를, (b)는 보낸 메시지 수를 나타낸 것이다. 앞에서 설명했던 것처럼, 플러딩 기법은 노드 수가 증가할수록 노드들이 보내거나 받는 메시지 수도 급속히 증가한다. 반면, 오버레이 네트워크를 구성하는 CAN 기법과 근접성 기반 기법은 비교적 메시지 증가율이 낮은 편이다. 특이할 만한 점은, 근접성 기반 기법의 경우 1홉 브로드캐스트를 이용함으로 인하여 노드가 받는 메시지 수가 CAN 기법에 비해 많을 것이라 예상됨에도 불구하고, 결과에서 보여지듯 더 적은 메시지를 보내고 받는데, 이는 물리적 이웃 노드의 정보를 통해 메시지가 이동하는 홉

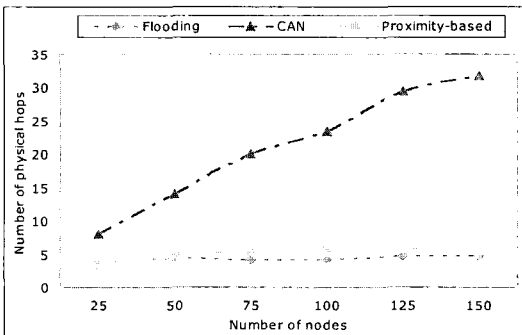
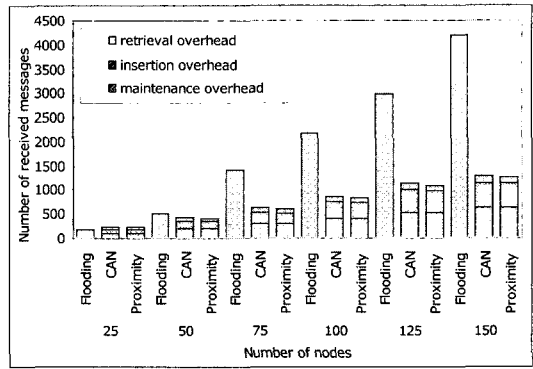
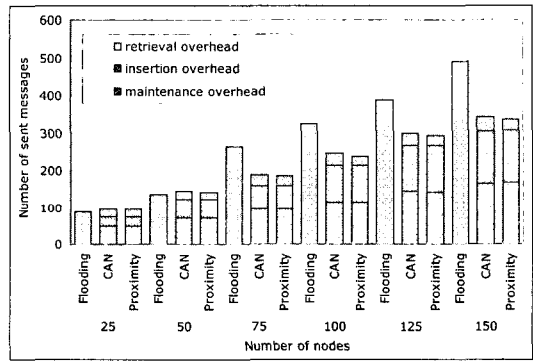


그림 9 노드 개수의 변화에 따른 물리적 홉 수



(a)



(b)

그림 10 노드 개수의 변화에 따른 메시지 오버헤드

수를 현저히 줄였기 때문이다.

그림 11은 CAN 기법과 근접성 기반 기법에서 전체 서비스 검색 메시지에 대한 응답 메시지 중에서 서비스 제공자로부터 받은 응답 메시지의 비율을 나타내고 있다. CAN은 노드 수가 많아져도 유니캐스트로 검색 메시지를 전송하므로 서비스 제공자를 만날 확률이 큰 변화가 없는 반면, 근접성 기반 기법에서는 노드 수가 많아질수록 1홉 브로드캐스트 메시지를 받는 이웃 노드들이 많아지게 되고 이에 따라 서비스 제공자를 만날 가능성이 더 높아지는 결과를 낳는다.

4.2.2 노드 밀집도 변화에 따른 결과

근접성 기반 기법을 사용하는 경우, 각 노드는 서비스

표 2 성능 분석을 위한 측정값

Metric	Description
Number of physical hops	물리적인 네트워크에서 메시지가 경유한 노드 수
Number of receiving messages	각 노드가 받은 평균 메시지 수
Number of sending messages	각 노드가 보낸 평균 메시지 수
Proportion of using neighbor info.	물리적 이웃 노드의 정보를 이용한 비율
Proportion of meeting service owner	서비스 제공자가 검색 요청에 응답한 비율

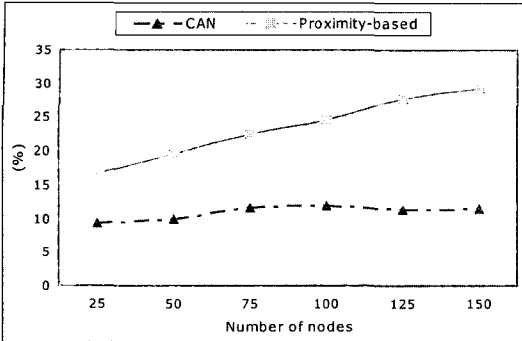


그림 11 노드 개수의 변화에 따른 서비스 제공자를 만날 비율

검색 메시지를 받아서 자신이 목적지가 아닌 경우, 메시지를 목적지까지 전달하기 위해서, 이웃 노드로부터 얻은 정보를 바탕으로 목적지에 가장 가까운 노드 또는 목적지인 노드를 선택하여 메시지를 전달하게 된다. 다음 홉으로 선택될 수 있는 노드는 물리적으로 1홉 내에 있는 이웃 노드들 및 그들의 논리적 이웃 노드 중 목적지를 포함하고 있는 노드, 또 논리적 라우팅 테이블에서 목적지에 가까운 노드이다. 즉 물리적인 이웃 노드의 정보를 이용하거나, 논리적인 라우팅 테이블을 이용하여 다음 홉의 노드를 선택하게 된다.

그림 12는 다음 홉의 노드를 선택할 때 물리적인 이웃 노드의 정보를 이용하는 비율을 노드 밀집도의 변화에 따라 나타낸 것이다. 노드 수를 100개로 고정하고, 정사각형 형태의 실험 영역의 크기를 조절하여 노드 밀집도를 변화시켰다. x축은 실험 영역의 한 변의 길이를 나타내며, 막대 그래프 위에 괄호 안에 쓰여진 숫자는 평균 1홉 이웃 노드 수로 노드 밀집도를 표현하고 있다. 노드 밀집도가 증가할수록 물리적 이웃 노드의 정보를 이용하는 비율이 높아지는 것을 볼 수 있는데, 이는 노드 밀집도가 높아질수록, 1홉 안에 있는 노드 수가 많

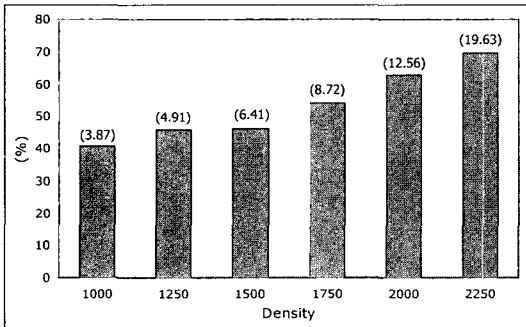


그림 12 노드 밀집도의 변화에 따른 물리적 이웃 정보의 이용률

아져서 그들 중에 목적지에 더 가까운 노드가 있는 확률이 높아질 뿐만 아니라, 그들의 논리적 이웃 노드 수도 큰 폭으로 증가하여 그들 중 목적지 키를 포함하는 노드가 있을 확률도 높아지게 되기 때문이다. 이와 같이 논리적 라우팅 테이블을 이용하는 대신에 물리적인 이웃 노드로부터 받은 정보를 이용함으로써 그림 9와 같이 라우팅 홉 수를 줄일 수 있는 것이다.

4.2.3 노드 이동성 변화에 따른 결과

그림 13은 노드 이동성을 변화시키며 서비스 검색을 수행했을 때, 서비스 검색 메시지가 서비스 명세서를 가진 노드에까지 전달될 때까지 거친 물리적인 홉 수를 나타낸 것이다. 노드의 이동성을 위하여 랜덤 웨이포인트 모델(random waypoint model)을 이용하였는데, 이 모델은 다음과 같이 동작한다. 사용자로부터 이동 속도의 최대값과 이동 후 한 점에서 머무는 시간인 휴지 시간(pause time)의 최대값을 입력으로 받는다. 네트워크 상의 모든 노드는 임의의 한 점을 선택하여 최대 이동 속도보다 작은 임의의 속도로 이동하고, 최대 휴지 시간보다 작은 임의의 시간 이후에 다른 점으로 이동하는 것을 반복한다. 이 실험에서는 최대 이동 속도를 40m/s로 하고, 최대 휴지 시간을 달리하며 실험하였다. 결과에서 볼 수 있듯이, 세가지 기법 모두 노드의 이동 속도와 상관없이 비슷한 수의 노드를 경유하였다. 여기에 나타내진 않았지만 다른 실험 결과에서도 노드의 이동성은 성능에 큰 영향을 끼치지 않았다. 플러딩 기법의 경우, 요청자와 목적지 노드가 이동하여 서로 멀어질 수도 있고 가까워질 수도 있으므로 노드 이동성에 민감하지 않게 동작하게 된다. CAN 기법의 경우에도, 물리적인 네트워크의 변화에도 상관없이 동작하는 것을 기본 원리로 하는 오버레이 네트워크의 특성 상 당연한 결과라고 할 수 있다. 한편 근접성 기반 기법은 오버레이 네트워크를 형성할 때가 아닌 메시지를 라우팅하고자 하는 순간 현재의 물리적 토폴로지 정보를 이용하여 메시지를 전달하게 되므로 노드의 이동성에도 최상에 가까

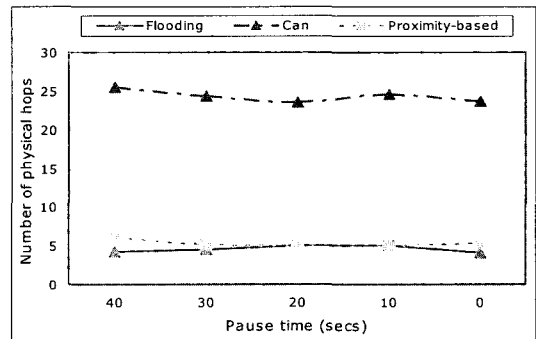


그림 13 노드 이동성의 변화에 따른 물리적 홉 수

운 성능을 내고 있다.

## 5. 결론 및 향후 과제

본 논문에서는 MANET에서 효율적인 서비스 검색을 위한 근접성 기반 오버레이 네트워크 라우팅 기법을 제안하였다. P2P 오버레이 네트워크의 개념을 MANET에 적용하여 응용 계층에 오버레이 네트워크를 구성함으로써 확장성을 증대시키는 한편, 물리적으로 가까운 이웃 노드의 정보를 이용함으로써 메시지 전송의 효율을 높였다.

근접성 기반 오버레이 네트워크 라우팅 기법은 MANET에서 1홉 브로드캐스트를 통해 물리적으로 가까이 있는 노드의 정보를 쉽게 얻을 수 있다는 특징을 이용하여, 물리적인 이웃 노드 중에서 논리적으로 가까운 노드에게 메시지를 전달함으로써 메시지 전송에 필요한 홉 수를 줄이는 방법이다. 이와 더불어 DHT 시스템의 특성 상, 각 노드는 자신의 서비스를 네트워크 상의 특정 노드에 등록하여 한 개의 서비스 명세서가 두 개의 노드에 저장되는 캐쉬 효과를 낼 수 있는데, 근접성 기반 오버레이 네트워크 라우팅 기법에서는 1홉 브로드캐스트로 이를 더 잘 이용할 수 있게 되어 라우팅의 효율을 높였다.

다양한 변수들을 가지고 실험한 결과, 근접성 기반 오버레이 네트워크 라우팅 기법은 DHT의 이점인 적은 메시지 오버헤드를 유지하면서, 메시지가 전송될 때 경유하는 노드 수를 최소화라고 할 수 있는 플러딩 기법과 거의 비슷하게 감소시킨다. 노드 수가 증가하는 경우나 노드 밀집도가 높아지는 경우처럼 물리적으로 1홉 내에 있는 노드의 수가 많아질수록, 라우팅 시에 이용할 수 있는 정보가 많아져서 다른 기법들에 비해 상대적으로 성능이 더욱 좋아지는 것을 알 수 있다.

앞으로의 연구에서는 노드들이 예고 없이 네트워크를 떠날 때의 성능을 평가해야 한다. 또한, MANET이 분할되거나 병합될 때에도 서비스 검색의 성공률이 떨어지지 않도록 하면서 효율적으로 서비스를 검색할 수 있는 방법도 MANET에서 필수적으로 연구해야 할 부분이다. 그리고, [12]에서 지적한 바와 같이 그림 10의 결과를 보면, DHT 구조를 유지하기 위해 주기적으로 주고받는 메시지 수가 전체 메시지의 반 이상을 차지할 만큼 오버헤드가 크다. MANET에 보다 적합한 DHT 구조 또는 IP 계층 라우팅과 결합된 방식의 오버레이 프로토콜에 대한 연구가 더 필요하다.

## 참고 문헌

[1] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Communications*,

Aug, 2001.

[2] E. Guttman, C. Perkins, J. Veizades, M. Day, "Service Location Protocol, Version 2," IETF RFC 2608, 1999.

[3] <http://www.uddi.org>

[4] K. Arnold, A. Wollrath, B. O'Sullivan, R. Schiefler, and J. Waldo, *The Jini specification*, Addison-Wesley, Reading, MA, USA, 1999.

[5] "Understanding UPnP: A White Paper," <http://www.upnp.org/resources/>, Jun. 2000.

[6] SUN Microsystems Inc., "JXTA v2.0 Protocols Specification," <http://www.jxta.org>, Feb., 2003.

[7] S. Motegi, K. Yoshihara and H. Horiuchi, "Service Discovery for Wireless Ad Hoc Networks," *The 5th International Symposium on Wireless Personal Multimedia Communication*, 2002.

[8] D. Chakraborty, A. Joshi, Y. Yesha and T. Finin, "GSD: A Novel Group-based Service Discovery Protocol for MANET," *IEEE Conference on Mobile and Wireless Communications and Networks*, Sep. 2002.

[9] A. Datta, "MobiGrid: P2P Overlay and MANET Rendezvous - a Data Management Perspective," *CAiSE 2003 Doctoral Symposium*, 2003.

[10] A. Klemm, C. Lindemann, and O.P. Waldhorst, "A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks," *Proc. IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall)*, Orlando, FL, Oct. 2003.

[11] 이세연, 장주욱, 조태경, "모바일 Ad hoc 네트워크에서의 P2P 네트워크 구축," *정보과학회지*, 제22권 제3호, Mar. 2004.

[12] Michael Klein, Birgitta Konig-Ries, Philipp Obreiter, "Lanes - A Lightweight Overlay for Service Discovery in Mobile Ad Hoc Networks," *Third Workshop on Applications and Services in Wireless Networks*, Bern, Swiss, 2003.

[13] Y. Charlie Hu, Saumitra M. Das, and Himabindu Pucha, "Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks," *HotOS-IX: Ninth Workshop on Hot Topics in Operating Systems*, Hawaii, May 2003.

[14] G. Ding, B. Bhargave, "Peer-to-Peer File-sharing over Mobile Ad hoc Networks," *The 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Mar. 2004.

[15] Clip2. The Gnutella Protocol Specification v0.4 (Document Revision 1.2) Jun. 2001. <http://www.clip2.com/GnutellaProtocol04.pdf>

[16] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *Proc. ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.

[17] I. Stoica, R. Morris, D. Karger, M.F. Kaashock, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications,"

*Proc. ACM SIGCOMM'01*, San Diego, CA, Aug. 2001.

[18] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *IFIP/ACM Middleware 2001*, Heidelberg, Germany, Nov. 2001.

[19] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault resilient wide-area location and routing," Technical Report UCB/CSD-01-1141, U.C.Berkeley, Apr. 2001.

[20] M. Castro, P. Druschel, Y.C. Hu, and A. Rowstron, "Topology-aware routing in structured peer-to-peer overlay networks," Technical Report MSR-TR-2002-82, Microsoft Research, 2002.

[21] L. Garces-Erice, E.W. Biersack, P.A. Felber, K.W. Ross and G. Urvoy-Keller, "Hierarchical Peer-to-peer Systems," *IFIP/ACM International Conference on Parallel and Distributed Computing(Euro-Par)*, 2003.

[22] K. Gummadi, R. Gummadi, S. Gribble, S. Ratanasamy, S. Shenker, and I. Stoica, "The Impact of DHT Routing Geometry on Resilience and Proximity," *Proc. ACM SIGCOMM'03*, Karlsruhe, Germany, Aug. 2003.

[23] K.W. Ross, E.W. Biersack, P. Felber, L. Garces-Erice, and G. Urvoy-Keller, "Topology-Centric Look-up Service," *Fifth International Workshop on Networked Group Communications*, 2003.

[24] C.G. Plaxton, R. Rajaram, and A.W. Richa, "Assessing nearby copies of replicated objects in a distributed environment," *The Ninth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, Jun 1997.

[25] K. Fall and K. Varadhan, "The ns Manual (formerly ns Notes and Documentation)," The VINT Project. <http://www.isi.edu/nsnam/ns/ns-documentation>.

[26] IEEE Computer Society LAN MAN Standards Committee, "IEEE Std 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) Specifications," The Institute of Electrical and Electronics Engineers, New York, 1999.

[27] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF RFC 3561, 2002.

국과학기술원 전자전산학과 초빙교수



이 은 주

1997년~2002년 고려대학교 컴퓨터학과(이학사). 2002년~2004년 한국과학기술원 전산학과(공학석사). 2004년~현재 삼성전자주식회사 정보통신총괄 무선사업부



정 현 구

1999년 한동대학교 전자전산 공학부(공학사). 2002년 한국과학기술원 전산학과(공학석사). 2002년~현재 한국과학기술원 박사과정



김 진 수

1991년 서울대학교 컴퓨터공학과(공학사) 1993년 서울대학교 컴퓨터공학과(공학석사). 1999년 서울대학교 컴퓨터공학과(공학박사). 1998년~1999년 IBM T.J. Watson Research Center 방문연구원 1999년~2002년 한국전자통신연구원 선임연구원. 2002년~현재 한국과학기술원 전자전산학과 조교수



윤 현 주

1988년 서울대학교 컴퓨터공학과(공학사). 1990년 한국과학기술원 전산학과(공학석사). 1997년 한국과학기술원 전산학과(공학박사). 1997년~1998년 ICASE, NASA LaRC 객원연구원. 1998년~1999년 이화여자대학교 컴퓨터학과 객원교수 1999년~2001년 ㈜브레인투엔티원 기술이사. 2002년~2003년 한국정보통신대학원대학교 연구교수. 2003년~2004년 한