■ 2007년도 학생논문 경진대회 수상작

모바일 애드 혹 네트워크에서 분산 해쉬 테이블 기반의 서비스 탐색 기법

(Distributed Hash Table based Service Discovery in Mobile Ad Hoc Network)

정 재 훈 † 이 승 학 † 김 남 기 †† 윤 현 수 †††
(Jaehoon Jung) (Seunghak Lee) (Namgi Kim) (Hynsoo Yoon)

요 약 Ad hoc 네트워크에서 필요한 서비스를 사용하려면 먼저 원하는 서비스를 어떤 노드가 제공하는지, 또한 이런 서비스를 호출하려면 어떠한 방법을 사용해야 하는지 등의 정보를 알아내야 한다. 본 논문에서는 이러한 문제점들을 해결할 수 있는 DHT(Distributed Hash Table) 기반의 서비스 발견 프로토콜을 제안한다. 제안하는 프로토콜은 중앙 룩업 서버를 요구하지 않고 멀티캐스트나 플러딩을 사용하지 않기 때문에 확장성을 지닌다. 성능평가 결과, 제안하는 프로토콜은 확장성이 있고 기존의 서비스 탐색 프로토콜에 비해 나은 성능을 가짐을 알 수 있었다.

키워드 : 서비스 탐색, 분산 해쉬 테이블, 오버레이 네트워크

Abstract In order to get a desired service in such environments, we need a service discovery method for discovering a device providing that service. In this paper, we propose a service discovery protocol which is based on DHTs (Distributed Hash Tables) to solve these problems. Our protocol is scalable since it does not require a central lookup server and does not rely on multicast or flooding. Simulation results show that our protocol is scalable and outperforms existing service discovery protocols.

Key words : Service discovery, Distributed Hash Table, Overlay networks

1. 서 론

지난 수년간 노트북 컴퓨터, PDA 등 모바일 디바이스(device)의 사용이 급격히 증가하였고 무선통신 기술의 발전 및 확산으로 인하여 이들 모바일 디바이스들을 이용하여 무선 네트워크 환경을 구성할 수 있는 모바일 애드 혹 네트워크(mobile ad hoc network)[1] 기술이 함께 발전하고 있다. 모바일 애드 혹 네트워크의 가장 큰 특징은 기반 시설(infrastructure)이 따로 필요 없고 구성이 쉬우며 모바일 디바이스들이 모여 임시적으로 네트워크를 구성한다는 점이다. 따라서 네트워크의 토폴로지(topology)가 지속적으로 변하며, 노드가 동적으로 네트워크에 들어오고 나가게 된다.

유비쿼터스 환경은 이동성을 가지는 수많은 노드들로 구성된 대규모의 모바일 애드 혹 네트워크가 될 수 있다. 따라서 유비쿼터스 환경에서 서비스 탐색이 제대로 이루어지기 위해서는 대규모의 모바일 애드 혹 네트워크에 적합한 서비스 탐색 기술이 필요하다. 그러나 기존의 서비스 탐색 기술은 확장성을 지니지 못해 대규모의 모바일 애드 혹 환경에는 적합하지 않다. 따라서 대규모의

· 본 연구는 2007년도 경기대학교 경기도지역협력연구센터 (GRRC) 사업으로부터 연구비 지원으로 수행하였습니다.

† 학생회원 : 한국과학기술원 전자전산학부
jungjh@nslab.kaist.ac.kr
shlee@nslab.kaist.ac.kr
†† 정 회 원 : 경기대학교 컴퓨터과학과
ngkim@kyonggi.ac.kr
(Corresponding authror임)
††† 종신회원 : 한국과학기술원 전자전산학부
hyoon@nslab.kaist.ac.kr

논문접수 : 2007년 5월 29일
심사완료 : 2007년 11월 21일

정보과학회논문지 : 정보통신 제35권 제1호(2008.2)

모바일 애드 혹 네트워크를 위한 서비스 탐색 방법은 중앙 서버를 사용하지 않는 분산 방식이어야 하고 멀티캐스트나 플러딩을 사용하지 않는 것이 바람직하다. 본 논문에서는 대규모의 모바일 애드 혹 네트워크에 적합한 분산 해쉬 테이블(DHT, Distributed Hash Table) 기반의 서비스 탐색 프로토콜을 제안한다. 제안된 프로토콜은 중앙 서버를 이용하지 않고 멀티캐스트나 플러딩을 사용하지 않기 때문에, 노드 수가 많은 대규모의 모바일 애드 혹 네트워크에서도 잘 동작하는 확장성을 지닌다.

기존의 유선 네트워크에서는 DHT 기반의 서비스 탐색 기술이 보편화되어 있다. 피어 투 피어(Peer to Peer, P2P) 컴퓨팅 환경에서 데이타 저장을 위한 확장성을 지닌 메커니즘이 DHT를 이용하는 것이다. DHT를 사용할 때 중요한 것은 오버레이(overlay) 네트워크를 노드들의 물리적 위치를 반영해서 구성하는 것이다. 그렇지 않으면 오버레이 네트워크 상에서 라우팅 되어 전달되는 메시지가 물리적인 전달 경로 상으로는 비효율적으로 멀리 전달 될 수 있기 때문이다.

따라서 본 논문에서는 모바일 환경에서도 노드들의 물리적 위치를 반영한 오버레이 네트워크를 효과적으로 유지하는 메커니즘을 제안함으로써 DHT 기반의 서비스 탐색 기술이 모바일 애드 혹 네트워크에서도 가능하게 한다.

2. 관련 연구

2.1 기존의 서비스 탐색 프로토콜들

애드 혹 네트워크에 속해 있는 임의의 노드는 다른 노드에서 제공하는 서비스를 사용하기 위해 원하는 서비스를 어떤 노드가 제공하는지 알아내야 하며 자신이 제공하는 서비스를 다른 노드가 사용할 수 있도록 알려야 하는데, 이를 위해서는 적절한 서비스 탐색 메커니즘이 제공되어야 한다. 서비스 탐색 메커니즘은 크게 세 가지로 분류 된다. 첫째, 중앙 집중식으로, 자신이 속한 네트워크 상에서 특정 룩업(lookup) 서버를 찾아 여기에 자신의 서비스 정보를 등록하고 또한 다른 노드의 서비스 정보를 얻어오는 방법이 있다. 둘째, 중앙 서버가 없는 분산 방식으로, 멀티캐스트나 플러딩에 의존하여 자신이 제공하는 서비스 정보를 다른 노드에게 알리고 또한 다른 노드가 제공하는 서비스 정보를 알아내는 방법이 있다. 셋째, 분산 방식이면서도 멀티캐스트나 플러딩(flooding)을 의존하지 않고 유니캐스트나 브로드캐스트(broadcast)를 주로 사용하는 방식이 있다.

중앙 집중식으로 서비스 탐색을 하는 대표적인 프로토콜로는 Jini[2], UDDI[3], Salutation[4]이 있다. 이들은 중앙 룩업 서버를 유지하면서 이를 이용해 서비스 광고 및 서비스 탐색을 수행한다. 이러한 중앙 집중 방식은 중앙 룩업 서버 역할을 하는 특정 노드를 두어야 하므로, 특정 기반 시설 없이 동작해야 하고 네트워크 특성이 다이나믹한 모바일 애드 혹 네트워크에는 적합하지 않다. 또한 서버를 둘 수 있다 하더라도, 서버에서 병목 현상(bottleneck)이 발생하기 때문에 노드들이 많은 대규모의 모바일 애드 혹 네트워크에는 적합하지 않다. 게다가 중앙 서버가 고장 나면 네트워크 전체가 서비스 탐색을 수행 할 수 없게 되는 문제점도 있다.

멀티캐스트 및 플러딩 방식으로 서비스 탐색을 하는 대표적인 프로토콜로는 SLP[5], UPnP[6], Konark[7]이 있다. 이들은 중앙 서버를 두지 않고 분산된 방식으로 동작하지만 기본적으로 멀티캐스트나 플러딩에 의존해서 서비스를 등록하고 탐색한다. 이러한 멀티캐스트 및 플러딩 방식은 중앙 룩업 서버를 사용하지 않는 분산 방식으로 동작하여 중앙 집중식 보다는 모바일 애드 혹 네트워크에 적합하다. 하지만 멀티캐스트나 플러딩은 리소스가 유선 환경에 비해 상대적으로 제한적인 애드 혹 네트워크환경에서는 비효율적인 방법이다. 따라서 노드가 많은 대규모의 모바일 애드 혹 네트워크에는 부적합하다고 할 수 있다.

최근에는 중앙 서버를 사용하지 않는 분산 방식이면서도 멀티캐스트나 플러딩에 의존하지 않는 서비스 탐색 기술에 대한 연구가 활발하다. 대표적으로 확장성 있는 서비스 탐색 기법(SSD, Scalable Service Discovery)[8]과 그룹 기반의 서비스 탐색 기법(GSD, Group-based Service Discovery)[9]이 있다. SSD는 클러스터링(clustering)을 기반으로 하고 블룸 필터(Bloom filter)[10]를 이용하여 대규모의 모바일 애드 혹 네트워크에도 잘 동작하는 확장성 있는 서비스 탐색을 제공한다. SSD에서는 모든 노드가 자신의 서비스를 다른 노드에게 광고하는 대신 클러스터링을 기반으로 클러스터 헤더들끼리만 블룸 필터를 통해 압축된 서비스 정보를 광고 함으로써 서비스 탐색을 수행한다. 하지만 노드 수에 비례해서 영역의 크기가 커진다면 클러스터 수가 많아지게 되고 블룸 필터 정보를 교환하는데 오버헤드가 커지게 된다. 즉, 블룸 필터를 통해 전달할 메시지의 크기가 줄어 들긴 하였지만, 블룸 필터 정보가 플러딩을 통해 전달이 되므로 항상 확장성을 지닌다고 볼 수 없다.

GSD는 분산 방식의 서비스 탐색 기법으로, 서비스 광고 메시지를 피어 투 피어(P2P, peer-to-peer) 방식으로 저장하고, 그룹 정보에 기반한 서비스 탐색 메시지의 선택적 전달(selective forwarding)을 통해 서비스를 탐색한다. 하지만 GSD에서 서비스 탐색 메시지가 선택적으로 전달 되는 것은 캐쉬에 저장된 그룹 정보의 양에 따라서 그 효과가 달라진다. 캐쉬에 저장된 그룹 정보가 너무 많거나 너무 적을 경우에는 서비스 탐색 메시지가 거의 브로드캐스트 되어 전체적으로는 플러딩에 가깝게 전달되게 된다. 특히 노드 수가 많아져 이웃 노

드 수가 많아질 경우에는 캐쉬에 저장된 그룹 정보가 많아져 선택적 포워딩이 플러딩에 가깝게 형성되기 때문에, GSD는 대규모의 모바일 애드 혹 네트워크에는 적합하지 않다고 할 수 있다.

따라서 본 논문에서는 대규모 모바일 애드 혹 네트워크에서도 적용할 수 있도록 분산 해쉬 테이블(DHT, Distributed Hash Table)을 기반으로 한 새로운 서비스 탐색 기법을 제안한다.

2.2 분산 해쉬 테이블

DHT는 피어 투 피어(Peer-to-Peer, P2P) 환경에서 분산 컴퓨팅을 위해 안전한 탐색 메커니즘을 제공할 목적으로 분산된 위치에 해쉬 테이블들을 저장하는 기술이다. DHT는 크게 키 분할(Keyspace partitioning)과 오버레이 네트워크(Overlay network)의 두 부분으로 나뉘는데 해쉬(hash)의 키(key) 집합들을 DHT 시스템 내부의 각 노드에 분산(keyspace partitioning) 시키고 DHT의 엔트리 포인트에 상관없이 키의 위치를 찾아갈 수 있도록 라우팅하는 알고리즘을 이용해서 DHT 시스템 내부의 네트워크(overlay network)를 떠돌며 목적지를 찾아가게 된다. 물론 홉 스트레치(hop stretch) - 최단거리에서의 물리적 홉 수에 대한 오버레이 경로 상에서의 물리적 홉 수의 비율 - 가 커지지 않게 하기 위해 물리적 토폴로지를 반영하여 오버레이 네트워크를 구성한다. DHT는 현재 인터넷상의 구조적 P2P 네트워크를 구성하는 데 쓰이지만, 노드의 이동성이 있는 모바일 애드 혹 네트워크에서는 사용이 어렵다. 왜냐하면, 노드들의 물리적 토폴로지를 반영한 오버레이 네트워크를 유지하기가 힘들기 때문이다.

모바일 애드 혹 환경에서 DHT를 사용하기 위해서는 노드가 이동하더라도 토폴로지를 반영한 오버레이 네트워크를 유지해야 한다. 이를 위해 크게 근접 이웃 선택(PNS, Proximity Neighbor Selection) 방식[11]과 근접 아이디 선택(PIS, Proximity Identifier Selection) 방식[12]이 제안되었다. PNS는 오버레이 네트워크 상에서 오버레이 아이디는 랜덤하게 할당하지만, 오버레이 라우팅 시에 토폴로지를 고려하여 가장 가까운 노드에게 전달하는 방식이다. 하지만 해쉬 값이 포함되는 범위 내의 노드 중에 2홉 이웃이 있어야만 효과가 나타나기 때문에 밀도(density)가 높은 네트워크에서만 잘 동작하는 문제가 있다. PIS는 오버레이 네트워크 상에서 오버레이 아이디를 할당할 때 토폴로지를 고려하면서 할당하여 오버레이 라우팅이 물리적 토폴로지에 맞게 효과적으로 동작하도록 하는 방식이다. 하지만 랜드마크 노드들은 물리적 토폴로지를 고려하여 오버레이 아이디가 할당된 것이 아니기 때문에 클러스터 간의 오버레이 라우팅에서는 홉 스트레치가 커지는 문제점이 있다. 또한 노드의 이동성이 있어서 가장 가까운 랜드마크 노드를 주기적으로 확인해야 하므로 오버헤드가 크다.

3. DHT 기반의 서비스 탐색 프로토콜

3.1 제안하는 프로토콜의 동작 원리

그림 1은 DHT 기반 서비스 탐색 메커니즘을 나타낸다. 그림에서 나타나듯이 노드들의 물리적 위치를 반영하여 오버레이 네트워크를 구성한다. 이는 토폴로지를 반영한 오버레이 네트워크를 구성하는데 있어서 2차원 카테션(cartesian) 영역이 물리적 2차원 영역을 가장 잘 표현할 수 있기 때문이다. 물리적 토폴로지를 반영한 오버레이 네트워크를 구성하는 방법은 다음 절에서 설명하기로 하고, 여기서는 토폴로지를 반영하는 오버레이 네트워크가 구성되어 있다고 가정하고 DHT기반의 서비스 탐색 기법의 동작 원리에 대해서 먼저 설명한다. 서비스 정보는 서비스 이름, 서비스 제공자 ID 등으로 기술되고, 이들은 DHT를 이용하여 분산된 방식으로 저장된다. 특정 서비스 정보는 그 서비스 이름의 해쉬 값에 해당하는 노드(랑데뷰 노드, rendezvous node)에 저장된다. 서비스 제공자는 DHT를 이용하여 랑데뷰 노드에게 서비스 정보를 저장시키고, 서비스 요구자는 동일하게 DHT를 이용하여 랑데뷰 노드에게서 서비스 정보를 찾아간다. 예를 들어, 그림 1에서 보듯이, 노드 E가 빔 프로젝터 서비스를 제공하고 있다. 노드 E는 자신의 서비스 정보를 알리기 위해 서비스 이름(빔 프로젝터)의 해쉬 값을 구한다. 만일 그 해쉬 값이 (0.2, 0.3)이라면, 오버레이 네트워크 상에서 해당 해쉬 값을 담당하는 노드가 A이므로, 노드 E는 자신의 서비스 정보를 노드 A(랑데뷰 노드)에게 전달하여 저장한다. 유사한 방식으로, 노드 C가 빔 프로젝터 서비스를 찾는다면, 원하는 서비스 이름(빔 프로젝터)의 해쉬 값을 계산한다. 모든 노드는 동일한 해쉬 함수를 사용하므로, 그 해쉬 값은 마찬가지로 (0.2, 0.3)이다. 따라서 노드 C는 원하는 서비스를 노드 A(랑데뷰 노드)에게 묻게 되고, 노드 A는

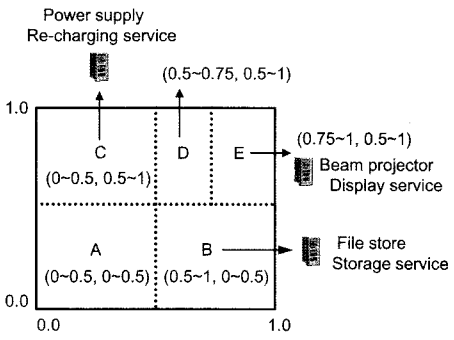

그림 1 DHT를 이용한 서비스 탐색

이미 노드 E에게서 받은 서비스(빔 프로젝터) 정보를 알려준다. 여기서, 서비스 광고 메시지나 서비스 요청 메시지가 랑데뷰 노드에게 전달되기 위해서는 오버레이 네트워크를 이용한 오버레이 라우팅이 사용된다.

3.2 토폴로지를 반영한 오버레이 네트워크

이번 절에서는 모바일 애드 혹 네트워크에서 노드들의 물리적 토폴로지를 반영한 오버레이 네트워크를 구성하는 방법을 설명한다. 제안된 방법의 기본 아이디어는 클러스터링(clustering)이다. 모바일 애드 혹 네트워크를 구성하는 노드들을 클러스터링을 통하여 물리적으로 가까운 노드를 묶어주고, 클러스터를 담당하는 헤더 노드를 선출한 다음, 클러스터 헤더들로만 오버레이 네트워크를 구성한다. 이 때, 그림 2에서 나타나듯이, 클러스터 헤더들이 그들의 물리적 위치를 반영하도록 오버레이 네트워크를 구성해야 한다. 노드들의 물리적 위치를 반영하기 위해서는, 새로 생성된 클러스터 헤더는 자신에게서 가장 가까운 클러스터 헤더를 찾아서 그 헤더가 담당하는 오버레이 네트워크에서의 키 영역을 분할한 뒤 그 중 하나를 할당 받아야 한다. 각 클러스터의 가장 자리에 있는 노드들은 이웃 클러스터의 헤더 주소를 알 수 있다. 클러스터 멤버 노드들이 주기적으로 이웃 클러스터의 헤더 정보를 자신의 헤더에게 알려 주면, 클러스터 헤더들은 자신의 이웃 클러스터 헤더가 누군지 알 수 있게 된다. 이 정보를 이용하여 새로 생성된 클러스터 헤더는 자신에게서 가장 가까운 클러스터 헤더를 찾을 수 있다.

그림 3은 가장 가까운 클러스터 헤더를 찾는 예를 나타낸다. 새로운 클러스터 헤더 f의 이웃 클러스터들은 {b, c, d, e}이고 이웃 클러스터 헤더 중에서 f의 이웃 클러스터와의 교집합의 크기가 가장 큰 d가 가장 가까운 클러스터 헤더로 선택되고 있다. 만일 가장 가까운 클러스터 헤더로 선택된 노드가 다수라면, 각 노드까지

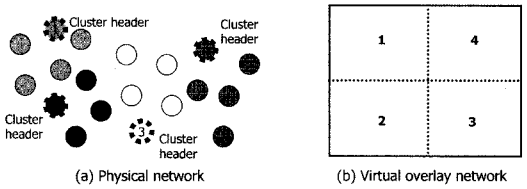

그림 2 클러스터 헤더로 이루어진 오버레이 네트워크

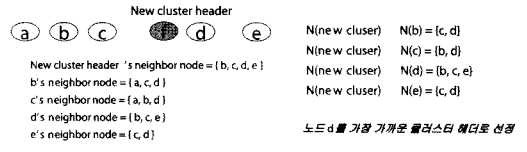

그림 3 가장 가까운 클러스터 헤더 선출의 예

의 홉 수를 계산하여 홉 수가 작은 노드가 가장 가까운 클러스터 헤더로 선택된다.

가장 가까운 클러스터 헤더를 선택 한 후에는, 선택된 헤더가 담당하는 키 영역을 수직이나 수평으로 이등분한다. 수직으로 나눌지 수평으로 나눌지는 한 번씩 번갈아 가면서 나누도록 한다. 중요한 것은 이등분된 키 영역을 노드의 물리적 위치를 고려해서 할당하는 것이다. 이를 위해서 선택된 헤더의 이웃 헤더 노드를 참조(reference) 노드로 이용한다. 만일 키 영역이 수직으로 나누어 졌다면, 선택된 헤더의 왼쪽 이웃 헤더나 오른쪽 이웃 헤더를 참조 노드로 한다. 키 영역이 수평으로 나누어졌다면, 선택된 헤더의 아래쪽 이웃 헤더나 위쪽 이웃 헤더를 참조 노드로 한다. 새로운 헤더와 선택된 헤더는 각각 참조 노드까지의 홉 수를 계산하고, 이 홉 수 정보를 바탕으로 이등분 된 키 영역을 노드의 위치에 맞게 담당하게 된다. 만일 참조 노드가 없다면, 이등분된 키 영역은 노드의 위치를 고려하지 않고 임의로 할당된다. 하지만 이러한 상황은 초기 상황에만 발생하게 되고, 오버레이 네트워크가 물리적 토폴로지의 대칭된 모습으로만 구성될 뿐, 홉 스트레치(hop stretch)가 증가하지는 않는다.

이상의 방법을 통해 노드의 물리적 토폴로지를 반영한 오버레이 네트워크를 구성할 수 있다. 또한, 클러스터링 기법에 의해 노드가 동적으로 움직이더라도 클러스터의 토폴로지 변화는 거의 없으므로 오버레이 네트워크가 물리적 토폴로지를 반영하도록 유지하는 오버헤드도 작다. 따라서 모바일 애드 혹 네트워크에서 노드들의 물리적 토폴로지를 반영하는 오버레이 네트워크를 효과적으로 구성할 수 있다.

3.3 DHT 기반의 서비스 탐색 프로토콜

서비스 제공자가 자신의 서비스 정보를 광고하기 위해 서비스 제공자는 자신의 서비스 정보를 자신의 클러스터 헤더에게 전달한다. 이를 받은 클러스터 헤더는 앞서 설명한 것과 마찬가지로 DHT를 기반으로 랑데뷰(rendezvous) 노드에게 서비스 정보를 전달한다. 이 때, 제안된 방식으로 구성된 오버레이 네트워크를 이용하여 오버레이 라우팅이 이루어진다. 랑데뷰 노드는 받은 서비스 정보를 저장해 둔다. 서비스 요구자가 원하는 서비스를 탐색하고자 할 경우, 서비스 요구자는 원하는 서비스 정보를 자신의 클러스터 헤더에게 전달한다. 이를 받은 클러스터 헤더는 앞서 설명한 것과 마찬가지로 DHT를 기반으로 랑데뷰 노드에게 서비스 정보를 요청한다. 이때, 마찬가지로, 제안된 방식으로 구성된 오버레이 네트워크를 이용하여 오버레이 라우팅이 이루어진다. 랑데뷰 노드는 해당 서비스를 제공하는 노드가 누구인지 서비스 요구자에게 알려준다.

4. 성능 평가

성능 평가는 NS-2 시뮬레이터[13]를 사용한 실험을 통해 수행하였다. 비교 대상인 기존 서비스 탐색 프로토콜은 2장에서 설명한 플러딩 기반의 서비스 탐색 프로토콜(FSD, Flooding-based Service Discovery), 확장성 있는 서비스 탐색 프로토콜(SSD, Scalable Service Discovery), 그리고 그룹 기반의 서비스 탐색 프로토콜(GSD, Group-based Service Discovery)이다. 성능 평가 기준에서 네트워크 로드는 서비스 탐색을 위해 전달된 모든 메시지의 총 합이고, 서비스 탐색 시간은 서비스 요구자가 서비스 탐색 요청 메시지를 보낸 시간부터 서비스 탐색 결과 메시지를 받은 시간까지 걸린 시간이다. 확장성을 평가하기 위해 사용한 실험 환경은 표 1과 같다.

그림 4는 각 프로토콜에 대해서 네트워크 로드를 나타낸다. FSD에서는 서비스 요청 메시지가 플러딩(flooding)이 되고, DHTSD에서는 서비스 요청 메시지가 유니캐스트(unicast)된다. 이러한 차이로 인해 그림 4에 나타나듯이 기본적으로 DHTSD의 네트워크 로드가 FSD보다 현저히 작게 나타난다. GSD에서는 서비스 요청 메시지가 플러딩과 유니캐스트의 중간 정도로 발생한다. 실험 결과, 한번의 서비스 요청에 대해 전체 노드의 1/2 정도의 노드가 메시지를 포워딩(forwarding)하였다. 하지만 노드 수가 많아질수록 각 노드의 캐쉬에 저장된 그룹 정보가 많아져서 그룹 정보를 바탕으로 한 선택적 전달(selective forwarding)이 플러딩에 가깝게 형성된다. 따라서 그림 4에 나타나듯이 노드 수가 많아질수록 네트워크 로드가 플러딩에 가까워진다. SSD에서는 DHTSD와 마찬가지로 서비스 탐색 메시지는 유니캐스트된다. 하지만 DHTSD에서는 노드가 네트워크에 들어오거나 나갈 때만 서비스 광고가 필요하지만 SSD에서는 주기적인 서비스 광고와 클러스터 헤더간의 블룸 필터(Bloom filter) 정보 전달이 필요하기 때문에 SSD가 DHTSD보다 네트워크 로드가 많게 나타났다. 이러한 차이는 노드 수가 많아질수록 더 커진다. 전체적으로 FSD의 성능이 가장 나쁘고 DHTSD의 성능이 가장 좋음을 알 수 있다. 또한 노드 수가 많아질수록 DHTSD의 성능이 SSD나 GSD에 비해 보다 더 좋아짐을 알 수 있다.

표 1 성능 평가를 위한 실험 환경 변수

실험 시간	360s
노드 수	50~200
영역 크기	1400m × 1400m
전송 거리	250m
무선 대역폭	1Mbps
클러스터 크기	2 hop
서비스 요청 간격	10s
서비스 광고 간격	10s
클러스터링 간격	5s
이동성 모델	Random Way Point
정지 시간	3s
노드 속도	3 m/s
라우팅 프로토콜	DSR [14]
MAC 프로토콜	IEEE 802.11

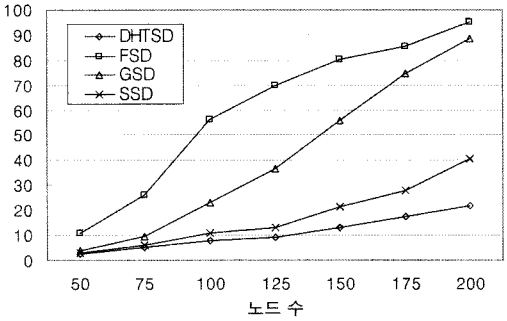

그림 4 네트워크 로드

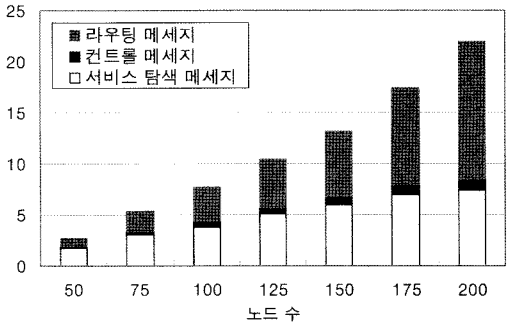

그림 5 메시지 별 네트워크 로드

그림 5는 DHTSD에서의 네트워크 로드를 발생된 메시지 별로 나타낸 것이다. 발생된 메시지는 클러스터링 및 오버레이 네트워크를 유지하기 위한 컨트롤(control) 메시지, 서비스 광고 및 요청에 필요한 서비스 탐색 메시지, DSR에 의해 발생된 라우팅 메시지다. 컨트롤 메시지는 다른 메시지에 비해 크기가 작기 때문에 전체에서 차지하는 비중이 작아 큰 영향을 미치지는 않는다. 서비스 탐색 메시지는 DHTSD에서 유니캐스트 되기 때문에 노드 수 N이 커질수록 sqrt(N)만큼 증가한다. 따라서 노드 수가 많아질수록 증가는 하지만 증가 정도는 오히려 줄어든다. 라우팅 메시지는 노드 수가 많아질 수록 전체에서 차지하는 비중이 커진다. 하지만 이는 실험을 라우팅 경로 설정이 전혀 안된 상태에서 수행한 것으로, 일반적인 애드 혹 환경에서 라우팅 정보가 이미 있는 경우에는 그 비중이 작아질 것이다.

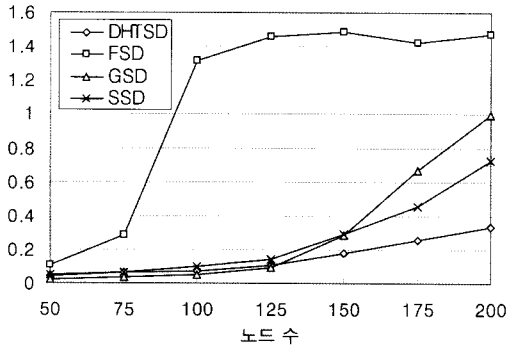

그림 6 서비스 탐색 시간

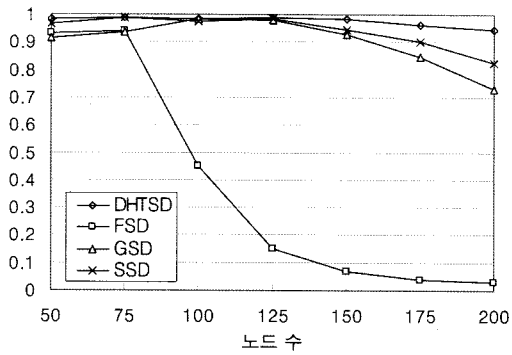

그림 7 서비스 탐색 성공 확률

그림 6은 각 프로토콜의 평균 서비스 탐색 시간을 나타낸다. 서비스 탐색 시간은 네트워크 로드에 영향을 받는다. 따라서 그림 4의 네트워크 로드에 영향을 받아 FSD의 서비스 탐색 시간이 가장 길다. 나머지 세 프로토콜에 있어서, 노드 수가 125까지는 서비스 탐색 시간이 거의 유사하다. 하지만 노드 수가 150 이상이 되면 SSD와 GSD의 서비스 탐색 시간이 DHTSD보다 확실히 길어짐을 알 수 있다. 이는 그림 4에서 알 수 있듯이 노드 수가 많아 질 수록 네트워크 로드의 차이가 커지기 때문이다. 결과적으로 DHTSD가 다른 프로토콜보다 노드 수가 많은 네트워크에서 더 좋은 성능을 보이므로 보다 확장성을 가진다. 한편 GSD에서 네트워크 로드가 큰데 비해 서비스 탐색 시간이 상대적으로 짧은 것을 알 수 있는데, 이는 GSD에서는 라우팅 경로 설정 과정이 없이 메시지를 전달하기 때문이다.

그림 7은 각 프로토콜 별로 노드 수에 따른 서비스 탐색 성공 확률을 나타낸다. 서비스 탐색 성공 확률도 서비스 탐색 시간과 마찬가지로 네트워크 로드에 영향을 받는다. 따라서 그림 4의 네트워크 로드가 원인이 되어 FSD의 서비스 탐색 성공 확률이 가장 낮다. 나머지 세 프로토콜에 있어서, 노드 수가 150까지는 세 프로토콜 모두 90% 이상의 성공 확률을 보이며 큰 차이는 없다. 하지만 노드 수가 150이 넘어가면서 DHTSD는 네트워크 로드가 크게 증가하지 않아서 여전히 95% 정도의 성공률을 보이지만 SSD와 GSD는 네트워크 로드가 크게 증가하여 성공률이 현저히 떨어진다. 한편, GSD가 FSD만큼의 네트워크 로드를 가지더라도 서비스 탐색 확률이 상대적으로 높은 것을 알 수 있다. FSD는 주로 라우팅 경로 설정 메시지로 인해 네트워크가 포화 상태가 되지만, GSD는 주로 서비스 탐색 메시지에 의해 네트워크가 포화가 되기 때문에 상대적으로 서비스 탐색 성공률은 높게 나타난다.

5. 결론 및 향후 연구 과제

유비쿼터스 환경은 이동성을 가지는 수많은 노드들로 구성된 대규모의 모바일 애드 혹 네트워크 (mobile ad hoc network) 가 될 수 있다. 따라서 유비쿼터스 환경에서 서비스 탐색이 제대로 이루어지기 위해서는 대규모의 모바일 애드 혹 네트워크에 적합한 서비스 탐색 기술이 필요하다. 그러나 기존의 서비스 탐색 기술은 확장성을 지니지 못해 대규모의 모바일 애드 혹 환경에는 적합하지 않다. 본 논문에서는 대규모의 모바일 애드 혹 네트워크에 적합한 분산 해쉬 테이블(DHT, Distributed Hash Table) 기반의 서비스 탐색 프로토콜을 제안하였다. 제안된 프로토콜은 중앙 룩업(lookup) 서버를 사용하지 않는 분산 방식으로서 멀티캐스트(multicast)나 플러딩(flooding)을 사용하지 않기 때문에 대규모의 모바일 애드 혹 네트워크에도 잘 동작하는 확장성을 지닌다. 표 2는 제안하는 방식과 기존 연구와의 차이점을 정리

표 2 기존 연구와의 차이점

	서비스 관리 방식	서비스 탐색 방식	모바일 지원	MANET 지원	네트워크 부하	대규모 지원
Jini, UDDI, Salutation	중앙집중	서버-클라이언트	부적합	부적합	크다	부적합
SLP, UPnP, Konark	분산	멀티캐스트, 플러딩	가능	가능	크다	부적합
SSD	분산	유니캐스트	가능	가능	작다	부적합
GSD	분산	브로드캐스트	가능	가능	크다	부적합
PNS	분산	DHT	가능	가능	크다	부적합
PIS	분산	DHT	가능	가능	크다	부적합
제안된 방식	분산	DHT	가능	가능	작다	적합

한 것이다.

실험 결과, 제안된 프로토콜에 의해 발생하는 메시지 중에서 라우팅 경로 설정을 위한 메시지의 비중이 크다는 것을 알 수 있었다. 이러한 라우팅 메시지를 줄일 수 있도록 제안된 프로토콜을 개선하는 것이 향후 연구 과제이다. 한편, 에너지 측면에서 클러스터 헤더의 에너지 소비가 크다. 따라서 에너지 측면에서의 로드 균형(load balancing)을 최적화 하는 기법에 대한 연구도 필요하다. 또한, 보다 넓은 영역에 더 많은 수의 노드를 두고 실험을 해볼 필요성이 있으며, 실험뿐 아니라 실제 구현을 통해 실험을 해 볼 필요가 있다.

참 고 문 헌

[1] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations," RFC2501, Jan. 1999.

[2] K. Arnold, et al., "The Jini Specification," Addison-Wesley, Jun. 1999.

[3] "Universal Description Discovery and Integration Platform," http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, Sept. 2000.

[4] The Salutation Consortium Inc., "Salutation Architecture Specification Part 1, Version 2.1 Edition," http://www.salutation.org, 1999.

[5] E. Guttman, C. Perkins, and J. Veizades, "RFC 2165: Service Location Protocol," Jun. 1997.

[6] R. John, "UPnP, Jini and Salutaion - A Look at Some Popular Coordination Frameworks for Future Network Devices," technical report, California Software Labs, http://www.cswl.com/whiteppr/tech/upnp.html, 1999.

[7] S. Helal, N. Desai, and C. Lee, "Konark-A Service Discovery and Delivery Protocol for Ad-Hoc Networks," IEEE WCNC, pp.2107-2113, Mar. 2003.

[8] F. Sailhan and V. Issarny, "Scalable Service Discovery for MANET," IEEE PERCOM, pp.235-244, 2005.

[9] D. Chakraborty, A. Joshi, Y. Yesha, and T. Finin, "Toward Distributed Service Discovery in Pervasive Computing Environments," IEEE Tran. on Mobile Computing, Vol.5, No.2, pp. 97-112, 2006.

[10] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," CACM, 13(7), Jul. 1970.

[11] C. Cramer and T. Fuhrmann, "Proximity Neighbor Selection for a DHT in Wireless Multi-Hop Networks," IEEE P2P Computing, pp. 1143-1148, Mar. 2003.

[12] R. Winter, T. Zahn and J. Schiller, "DynaMO: A Topology-aware P2P Overlay Network for Dynamic, Mobile Ad-Hoc Environments," Kluwer Telecomm. System Journal 27:2-4, pp. 321-345, 2004.

[13] NS-2, http://www.isi.edu/nsnam/ns.

[14] D.B. Johnson, D.A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," in Ad Hoc Networking, ed. C.E. Perkins, ch.5, pp. 139-172, Addison-Wesley, 2001.

정 재 훈

2005년 경북대학교 컴퓨터공학과 학사 2007년 한국과학기술원 전산학 석사. 2007년~현재 포스데이타 연구원. 관심분야는 무선 센서 네트워크

이 승 학

2000년 한국과학기술원 전산학 학사. 2003년 한국과학기술원 전산학 석사. 2003년~현재 한국과학기술원 전산학과 박사과정. 관심분야는 무선 센서 네트워크, 애드혹 네트워크, peer-to-peer 네트웨크

김 남 기

1997년 서강대학교 전산과 학사. 2000년 한국과학기술원 전산학 석사. 2005년 한국과학기술원 전산학 박사. 2005년~2007년 삼성전자 책임연구원. 2007년~현재 경기대학교 컴퓨터학과 전임강사. 관심분야는 네트워크, 무선 통신 시스템

윤 현 수

1979년 서울대학교 전자공학과 학사. 1981년 한국과학기술원 전산학과 석사. 1981년~1984년 삼성전자 연구원. 1988년 오하이오 주립대학 전산학 박사. 1988년~1989년 AT&T Bell Labs. 연구원. 1989년~현재 한국과학기술원 전산학 교수. 관심분야는 Adhoc망, 암호학, 상호연결 네트워크, 병렬 컴퓨터