

논문 2008-5-23

모바일 애드-혹 네트워크에서 연결차수 기반 탐색 및 복제 기법

Connection-Degree Based Search and Replication in Mobile Ad Hoc Networks

강은영*, 임용순**

Eun-Young Kang, Yong-Soon Im

요 약 이동 애드-혹 네트워크에서 이동 노드들이 자신이 원하는 정보를 찾고 이를 복제하는 연구들은 유비쿼터스 환경에서 아주 필요한 응용이다. 본 논문에서는 연결 차수를 기반으로 하여 파일, 컨텐츠등 데이터 제공자의 경로 정보를 로컬 캐쉬에 저장하고, 파일의 복제본을 최소화하여 생성하는 기법을 제안한다. 이 기법에서는 질의가 많이 도착하는 클러스터 헤더 노드에 파일 제공자의 경로 정보를 저장하고, 클러스터 헤더 노드의 부하를 줄이기 위하여 그 이웃노드에 복제본을 생성한다. 이는 제한적이고 지역적인 정보만을 이용하는 이동 노드들중 클러스터 헤더를 판별하는 알고리즘을 제시하며, 동적인 네트워크에서 확장적이고 적응적이며 최신의 정보를 검색하고 활용할 수 있도록 한다. 이는 검색의 정확성과 응답 시간이 빨라지고 네트워크 통신 비용을 줄이며, 이동 노드들간의 연결 단절 상황에서도 데이터를 액세스 할 수 있어 효과적이다. 이의 효율성을 시뮬레이션을 통하여 보인다.

Abstract In MANET, it is important to efficiently manage data objects and traffic to improve network throughput and response time. This paper proposes a connection-degree based data object search and replication scheme in mobile ad-hoc networks. In this method, information about the path data object providers is stored at the cache of a cluster header at which lots of queries arrive, so that, to reduce the load of the cluster header, replicas can be created at its neighboring nodes. The method proposes an algorithm that picks up a cluster header among mobile nodes and makes it possible to search for and utilize adaptive and up-to-date information in MANET. The method is expected to be effective since it enables access to data objects in spite of broken links among mobile nodes with an enhancement in network response time of searching and a decrease in communication costs. The efficiency of this system was verified via simulation.

Key Words : Mobile, Ad Hoc Network, search, replication, connection-degree

1. 서 론

모바일 애드-혹 네트워크(MANET, Mobile Ad-Hoc Network)는 기존에 설치된 유선망이나 기지국의 도움없이 이동 노드들간에 자율적으로 구성되는 망으로서, 최근 산업 및 연구 단체에서 MANET에 대한 관심이 높아지고 있다[1,2]. 이는 MANET가 더 이상 단일의 독립적

인 네트워크로서의 기능을 수행할 뿐 아니라 기존 인프라 기술과 접목하여 유선 및 무선 네트워크가 결합된 네트워크 구성이 가능하기 때문이다. 또한, PDA, 핸드폰, 노트북과 같은 이동 디바이스들도 빠른 발전을 하고 있다. 그러므로 이런 다양한 이동 디바이스들이 상호 연결되어 네트워크상의 서비스나 파일과 같은 원하는 데이터를 이웃 노드들과의 협조를 통하여 자동으로 검색하고 이 데이터를 액세스 할 수 있게 하는 방법은 매우 중요하다[3,15].

서비스 검색 프로토콜은 네트워크에서 이용 가능한

*정회원, 성균관대학교 정보통신학부

**중신회원, 국제대학 방송학부

접수일자 2008.9.27, 수정일자 2008.10.10

정보를 동적으로 발견하고 네트워크 사용자가 원하는 서비스를 액세스 할 수 있도록 한다. 그러므로 필요한 데이터를 검색하는데 이 서비스 검색 프로토콜을 사용할 수 있다. 또한, 복제 프로토콜은 데이터 접근성을 높이기 위하여 데이터를 소유자 노드와 함께 그 데이터를 필요로 하는 다른 노드에 복제본을 가지고 있는 기법이다[4]. 이 서비스 검색 프로토콜과 복제 프로토콜 두 기법을 합하여 데이터를 검색하고 복제함으로써, 효율적인 검색을 위한 방법으로 제시되고 있다.

기존의 서비스 검색 프로토콜은 비구조적 P2P 와 구조적 P2P 로 연구되었다. 비구조적 P2P 는 Gnutella[7]와 KaZaA[8] 등이 있다. Gnutella는 중앙 집중형 디렉터리 서비스를 주로 이용한다. 이와 같은 비구조적 네트워크는 중앙서버와의 컨트롤이 있어야 하며 많은 수의 쿼리 메시지를 생성하는 문제점이 있다. 반면, Chord[9], CAN[10], Pastry[11]와 같은 구조적 P2P 는 분산해쉬 테이블(DHT)을 기반으로 하고 있다. 그러나 이동 애드-혹 네트워크에서 DHT를 사용하는 경우는 피어들이 이동하는 동적인 원리, 자원 제한, 무선 특성 때문에 분산해쉬 테이블을 유지하기 위해서는 어려움이 있다. 또한, 기존의 복제 프로토콜은 검색 속도와 효율을 높이기 위한 방법으로 임의 경로 검색(Random Walk Searching) 방법에 대하여 연구되고 있다[4]. 또 다른 복제기법으로 균등기법[12], 제곱근 복제 기법[13]등을 들 수 있다.

이러한 기존의 논문에서는 고정된 유선 네트워크에 적용되고, 데이터 제공자(목적지노드)까지의 경로에 있는 모든 노드에 데이터를 복제함으로써, 동적이고 자원이 부족한 모바일 애드-혹 네트워크에 적용하기 위해서 많은 연구가 필요하게 되었다.

본 논문에서는 검색의 정확성과 속도를 높이고 비용을 효율적으로 줄일수 있는 복제기법을 제안한다. 앞에서 제시한 문제점을 해결하기 위한 방법으로 다중 홉 애드-혹 네트워크에서 이웃 노드의 여러 정보(데이터정보, 접속 정보, 전력 정보, 대역폭 정보, 위치정보)를 효율적으로 이용한다. 이 정보를 이용하여 허브(hub) 역할을 하는 클러스터 헤더와 일반 노드를 판별한다. 효율적으로 검색 성능을 향상시킬 수 있도록 클러스터 헤더에는 데이터의 경로 정보만을 캐시에 저장하고, 클러스터 헤더의 이웃 노드중 하나에 콘텐츠의 복사본을 저장하는 기법을 제안한다. 복사본은 네트워크의 연결이 단절되었을 때 필요한 데이터를 접근 할 수 있다.

이는 제한적이고 지역적인 정보만을 이용하여 이동 노드들 중 클러스터 헤더를 판별하는 알고리즘을 제시하며, 동적인 네트워크에서 적절하고 확장성이 있으며 최신의 정보를 검색하고 활용할 수 있도록 한다. 이에 대한 효율성은 시뮬레이션을 통하여 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 서비스 디스커버리를 위한 여러 가지 기법들을 살펴보고, 3장에서는 제안한 서비스 디스커버리 구조에 대하여 설명하고, 4장에서는 제안하는 탐색 및 복제 프로토콜을, 5장에서는 제안한 방법의 성능평가에 대하여 설명하고, 6장에서는 결론을 맺는다.

II. 관련 연구

기존 네트워크(인터넷)는 서비스 디스커버리를 제공하기 위해 비구조적 P2P 인 Gnutella[7]와 KaZaA[8] 등을 사용하였다. Gnutella[7]는 중앙 집중형 디렉터리를 주로 이용하였다. 각 서버는 중앙의 디렉터리의 위치를 알고, 자신이 가지고 있는 서비스의 정보를 등록하고, 클라이언트는 이 디렉터리에 서비스 요청 쿼리를 보냄으로서 서비스를 찾고 이를 사용한다. 또한 KaZaA와 같은 서비스 디스커버리 프로토콜은 일반적으로 많은 수의 쿼리 메시지를 생성하는 플러딩 서치 프로토콜을 사용한다. 이와 같은 비구조적 네트워크는 중앙서버와의 컨트롤이 있어야 하며 많은 수의 쿼리 메시지를 생성하는 문제점이 있다.

반면 구조적 P2P 네트워크인 Chord, CAN, Pastry와 같은 분산해쉬테이블(DHT)을 기반으로 구성되며, DHT에서 각 서비스는 키에 의해 ID가 부여된다. DHT는 오버레이 네트워크 피어들에게 각각의 키에 해당하는 파일을 담당하여 지정하게 함으로써 키들을 매핑하고, 파일을 찾기 위하여 주어진 키를 담당하는 현재의 피어를 찾으면 되므로 효과적이다. 그러나 이동 애드-혹 네트워크에서 DHT를 사용하는 경우는 피어들이 이동하는 동적인 원리, 자원 제한, 무선 특성 때문에 DHT를 유지하기 위해서는 어려움이 있다.

또한, 임의 경로 검색에서는 검색의 성능 향상을 위해 복제 기법을 사용한다. 검색시 용량이 큰 노드들에게 질의가 많이 전달되는 점을 이용하여 검색 속도와 그 정확도를 높였다. 하지만 이 방법은 복제에 드는 비용도 문제

이거나와 아무리 용량이 큰 노드라 할지라도 복제본이 집중되므로 적정 부하를 초과하기 쉬운 문제점을 가지고 있다. 한편, 참고문헌 [4]에서는 복제 기법으로 균등기법과 비례기법에 대해 분석하고, 최적에 가까운 제공근 복제 기법을 제시하였으며, 이를 구현하기 위하여 경로 복제 방법을 제시하였다. 하지만, 이 논문들에서는 고정된 유선 네트워크에 적용되고, 데이터 제공자 (목적지노드) 까지 경로에 있는 모든 노드에 데이터를 복제함으로써, 동적이고 자원이 부족한 모바일 애드-혹 네트워크에 적용하기 위해서는 연구가 더 필요하다.

III. 제안하는 서비스디스커버리 구조

이번 장에서는 하나의 모바일 디바이스에 포함하고 있는 구조에 대하여 기술한다. 그림1은 하나의 모바일 디바이스에 포함하고 있는 컴포넌트와 제안한 서비스디스커버리의 구조를 보여준다.

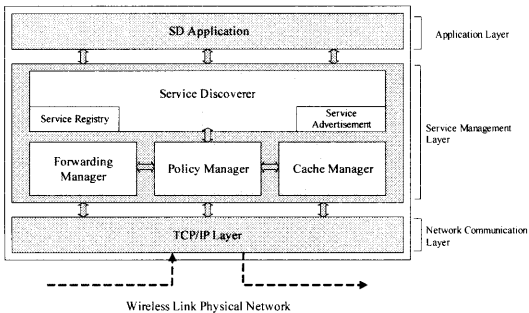


그림 1. 서비스 디스커버리 구조
Fig. 1. Service Discovery Architecture

그림1에서 이웃과 상호협조적인 서비스 디스커버리는 3개의 계층로 구성되며, 응용 계층(application layer), 서비스 관리 계층(service management layer), 네트워크 통신 계층(network communication layer)을 들수있다. 응용계층은 사용자에게 오디오-비디오 플레이어, 이벤트 알람, 파일 저장등과 같은 응용을 제공한다.

서비스 관리 계층은 탐색과 관련된 서비스를 제공한다. 서비스 관리 계층에서 네트워크에서 이웃 피어들로부터 제공된 서비스에 관한 정보를 등록하기 위해 캐쉬를 사용하고 캐쉬메니저를 사용한다. 서비스 제공자는 자신의 서비스를 로컬 캐쉬에 있는 로컬 서비스 디렉터리(Local Service Directory, LSD)에 저장하고, 자신의

서비스를 인접한 이웃 노드에게 주기적으로 광고한다. 네트워크에 있는 모든 디바이스들은 주어진 시간 동안 그들의 로컬 캐쉬에 수신된 정보를 저장하고, 서비스 제공자로부터 들려지는 광고 메시지를 저장하며, 여행중 경로하는 노드에도 서비스 정보를 캐쉬에 저장하고 관리한다. 사용자가 특정 서비스를 발견하기를 원할 때, 서비스 디스커버리는 먼저 그들의 로컬 캐쉬에 있는 LSD를 검색함으로써, 탐색을 시작한다. 만일 로컬에서 요청한 서비스를 발견하지 못하면, 서비스 디스커버리 요청 메시지를 전달 매니저(Forwarding Manager)로 보내어 서비스 디스커버리를 시작한다. 전달 매니저는 서비스 광고 메시지, 요청 메시지의 전달을 담당하는 역할을 한다.

서비스 디스커버리는 서비스 디스커버리 요청 메시지를 전달하기 위해 서비스 디스커버리 프로토콜을 사용하여 이웃노드, 여행한 경로에 있던 노드들 중, 가장 빠른 지름길을 찾는다. 선택된 노드에게 서비스 요청 메시지를 전달한다. 이후에 이들 컴포넌트들에 대해서 자세히 설명하도록 한다. 레이어에 있는 정책 매니저(Policy Manager)는 현재 노드의 정책을 통하여 노드를 컨트롤하는 역할을 담당하고 있다. 이런 정책들은 데이터 접근 횟수를 이용한 갱신 정책과 같은 캐쉬 우선권, ttl(time-to-live) 등을 나타낸다.

네트워크 통신 계층(Network communication layer)에서는 어떤 라우팅 프로토콜을 사용하여도 무관하지만, 본 논문에서는 인기있는 두 개의 프로토콜을 사용한다. 패시브 형태인 DSDV 프로토콜과 온디멘드 형태인 AODV를 사용한다.

IV. 제안하는 탐색 및 복제 프로토콜

이 장에서는 전체 시스템에 대한 개요와 제안한 탐색 프로토콜과 복제 프로토콜에 대하여 설명한다. 제한된 메모리, 배터리를 가진 모바일 디바이스, 대역폭, 통신범위를 가진 무선 통신 네트워크의 특성등을 고려하여 유연하고 적절한 모바일 프로토콜을 제안한다. 피어들이 랜덤하게 이동하여 빈번하게 네트워크 토폴로지가 변하는 상태를 고려한다. 본 논문에서는 탐색의 응답 시간과 데이터 접근성을 위하여 2가지 방법을 사용한다. 주소(경로) 정보가 실제 데이터에 대한 포인터 역할을 하는 연결차수 기반 탐색 프로토콜 방법과 실제 데이터를 직접 복제하는 복제 프로토콜 방법이다.

4.1 전체 시스템에 대한 개요

본 논문에서 제안하는 연결 차수 기반 데이터 검색 및 복제 기법에서 네트워크는 몇 개의 클러스터들로 구성될 수 있다. 각 클러스터는 3가지 종류로 구성되는데 각각 클러스터 헤더, 게이트웨이 노드, 일반 노드로 들 수 있다.

클러스터에서 각 노드는 3가지 중 한가지 역할을 한다. 각 이동 노드는 주기적으로 이웃 노드에게 Hello message를 교환하여 데이터 정보를 이 메시지에 같이 포함한다. 이 메시지를 통하여 이웃 노드들의 연결 차수, 파워차수에 대한 정보를 모아 이웃노드관리 테이블을 관리한다. 메시지를 전달하려고 시도하는 노드는 메시지를 전달할 노드로 주변 노드중 연결차수가 자신의 연결차수보다 크거나 같은 노드를 선택하고 메시지를 전달한다. 이러한 메시지 전달 정책으로 메시지는 연결 차수가 높은 노드들로 전달되며, 가장 연결 차수가 높은 노드에게 전달되면 더 이상의 메시지 전달은 발생하지 않는다.

메시지가 마지막으로 전달된 노드를 ‘클러스터 헤더’ 노드라 한다. 클러스터 헤더 노드는 네트워크에 여러 개가 존재할 수 있다. 어떤 노드가 자신의 연결 차수 보다 더 높은 연결 차수를 가지는 이웃 노드가 복수개가 존재하는 경우 클러스터 헤더는 중첩되고 이 노드는 두 클러스터에 속하게 된다. 이 노드를 ‘게이트웨이 노드’라 한다. 클러스터의 생성은 메시지 전달에 따라 자동으로 구성되며, 각 이동 노드는 자신의 연결 차수와 이웃노드관리 테이블에 있는 인접 노드의 연결 차수를 비교함으로써 자신이 클러스터 헤더 노드인지 게이트웨이 노드인지를 판단한다. 또한 노드의 이동에 따라 클러스터 헤더 노드와 게이트웨이 노드는 동적으로 변경될 수 있으며, 클러스터 역시 동적으로 변하게 된다. 그림 2는 애드-혹 네트워크 내에 생성된 다중 클러스터의 예를 보여준다.

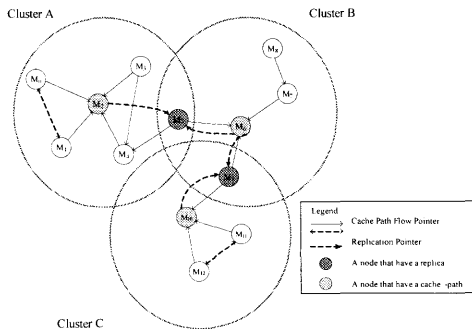


그림 2. 클러스터 구조
Fig. 2. Cluster Architecture

2의 연결차수를 가진 피어 M0, M1, M3는 자신의 연결차수보다 큰 4의 연결 차수를 가진 노드 M2에게 메시지를 전송한다. 3의 연결 차수를 가진 M4는 M2에게 메시지를 전송한다. 노드 M2는 자신의 연결차수보다 큰 이웃 노드가 없으므로 메시지 전달은 더 이상 발생되지 않는다. 그러므로 노드 M2는 클러스터 헤더가 되며, 클러스터A의 형태가 구성된다. 유사하게, 노드 M5, M6, M7, M8, M9는 클러스터B의 형태로, 노드 M9, M10, M11, M12는 클러스터 C의 형태를 구성한다. 이때, 양쪽 클러스터에 속한 노드 M5과 M9는 게이트웨이 노드로서, 그들은 한 개의 클러스터보다 많은 클러스터들에 있는 노드들과 통신할 수 있다.

4.2 연결차수 기반 탐색 프로토콜

본 논문에서 제안하는 연결 차수 기반 탐색 기법에서 로컬 캐시는 주소(경로) 정보를 가지고 있어 실제 데이터에 대한 포인터 역할을 한다. 즉, 데이터를 직접 복제하는 것이 아니고, 현재 노드로부터 실제 데이터에 대한 경로, 실제 데이터가 존재하는 노드의 주소, 데이터의 크기 등을 저장함으로써 질의 메시지가 쉽게 실제 데이터로 찾아갈 수 있도록 한다.

메시지를 전달하려고 시도하는 노드는 메시지를 전달할 노드로 이웃 노드중 연결차수가 자신의 연결차수보다 크거나 같은 노드를 선택하고 메시지를 전달한다. 메시지에는 데이터를 제공할 수 있는 노드의 IP 주소(경로) 정보가 포함된다.

아이디어는 그림 2를 사용하여 설명할 수 있다. M0가 데이터를 요청하고, M3이 데이터를 제공한다고 가정하자. M0는 먼저 자신의 로컬캐시테이블을 먼저 검색한다. 이 때 해당 키 값을 발견하였다면 이는 자신이 바로 데이터 제공자임을 의미한다. 만약 등록된 데이터가 로컬 정보에 없다면, 찾고자 하는 값을 포함하는 데이터요청 메시지를 만들어 자신의 연결차수보다 크거나 같은 4와 2의 연결 차수를 가진 노드 M2와 M1에게 요청 메시지를 전송한다. 데이터를 요청하는 메시지를 받은 M2와 M1은 자신의 로컬 캐시 테이블에서 요청한 데이터의 정보가 있는지를 검색한다. 데이터 요청 메시지의 중간 과정이나 데이터 제공자에서 요청한 서비스가 발견 되면, 노드는 발견한 데이터의 정보를 바탕으로 데이터 응답 메시지를 만든다. 데이터 응답 메시지는 데이터 요청 메시지에서 지정한 데이터 그리고 이 데이터를 제공할 수 있는

서버의 IP 주소 정보를 포함한다. 데이터 응답 메시지는 여러 개가 존재 할 수 있다. 이들 수신된 데이터 응답 메시지 중 데이터의 여유시간값이 가장 큰 값을 가지는 데이터제공자를 선택하고, 선택된 데이터제공자에게 데이터를 호출한다. 만약 자신의 노드에서 데이터가 발견되지 않았다면, 이웃노드 관리 테이블을 검색하여 이웃 노드들 중 자신의 연결 차수보다 크거나 같은 연결 차수 값을 가지고 있는 이웃들에게 데이터 요청 메시지를 전송한다.

데이터 요청 노드가 데이터를 요청할 때, 자신이 속한 클러스터에 속한 데이터만을 찾을 수 있다. 따라서 데이터 요청 노드가 전송한 데이터 요청 메시지가 클러스터 헤더까지 전달되고도 해당 데이터를 발견하지 못할 경우가 있다. 그림 2의 예에서 M0가 M7에 있는 데이터를 요청한다고 가정하자. M7의 데이터는 게이트웨이 노드 M5와 M9에 데이터의 복제를 가지고 있다. 노드 M0에서 요청한 데이터 요청 메시지가 클러스터 헤더 노드에까지 전달되고도 해당 데이터를 발견하지 못하면, 클러스터 헤더 노드가 파악하고 있는 게이트웨이 노드에게 데이터 요청 메시지를 보낸다. 클러스터 헤더 노드가 전송한 데이터 요청 메시지를 수신한 게이트웨이 노드는 자신에게 복제된 데이터가 있으면 이에 응답하고, 복제된 데이터가 없으면 다른 클러스터로 해당 데이터 요청 메시지를 전송한다. 결국 게이트웨이나 다른 클러스터에서 해당 데이터 요청 과정을 반복하게 되고 요청한 데이터를 발견하게 된다.

게이트웨이 노드는 이웃노드관리테이블에서 자신의 연결차수보다 크거나 같은 이웃노드에게 게이트웨이 노드 알림 메시지를 보낸다. 이 게이트웨이 노드 알림 메시지는 결국 클러스터 헤더에게 전달된다.

이 방법의 장점은 먼저 직접 복제본을 옮기지 않고 주소(경로)에 대한 정보만을 캐쉬함으로써 복제에 대한 비용을 줄일 수 있다. 또한 경로에 대한 정보만을 연결차수 기반으로 이웃 노드의 수가 많은 노드에 전파함으로써 검색시 검색 효율성을 높인다.

다음 그림 3은 데이터에 대한 탐색 요청 메시지를 받았을 때에 대한 처리 알고리즘을 나타낸다.

```

//데이터 요청 메시지가 도착 했을 때
Request_Search(Mi)
{
  For every data entry list in local cache
  {
    if requested data exists in local cache
      send response message to data requester
  }

  /* 연결 차수(cd)를 기준으로 이웃 노드중 전송할 이웃노드를 선택, 이웃노드관리테이블(NNMT) */
  while (k < size of NNMT) {
    For each neighbor n in NNMT {
      if requested data exists in NNMT[k]
        send response message to data requester
      else If NNMT[k]cd >= CurrentNodecd
        forward Request_Search(Mi) to neighbor node in NNMT[k]
      else drop packet
    }
    k = k + 1;
  }
}

```

그림 3. 탐색 알고리즘
Fig. 3. Search Algorithm

4.3 토폴로지 기반 복제 프로토콜

데이터 복제는 사용자들에게 데이터를 가깝게 하도록 하기 위하여 캐쉬하고 이동하는 훌륭한 방법이다.

MANET에서 가용성을 높이기에는 좋은 방법 중 하나는 복제이다. MANET에서 노드들의 잦은 이동으로 각 클러스터들의 연결이 단절되었을 때 또는 노드들에 장애가 발생하였을 때, 그에 속한 데이터는 접근 할 수 없다. 이런 경우 복제본은 아주 유용하게 사용될 수 있다. 더군다나 복제본은 네트워크에서 요청된 메시지에 응답할 확률을 높여 탐색 비용과 평균 응답 시간을 줄여준다. 또 복제는 접근 지연과 대역폭소비를 감소시킨다. 그것은 또 부하 밸런싱을 쉽게 하고 여러 개의 데이터 복사들을 생성함으로써 신뢰성을 증가시킨다[5,6,16].

그러므로 원하는 데이터 제공자가 네트워크를 떠나거나 이동하였을 때 복제본이 사용 될 수 있다.

복제본이 놓이는 노드를 선택하는 것은 특정 파일에 대한 요청 회수, 읽기/쓰기 통계, 그리고 네트워크 토폴로지, 데이터의 위치, 데이터의 크기와 같은 다양한 요소들과 관련이 있다. 복제본 생성 확률은 데이터 참조 횟수에 비례하고, 노드의 현재 용량, 데이터 크기에 반비례함을 알 수 있다. 즉 이웃 노드중 한 노드로부터 그 복제할 데이터를 검색하는 요청이 많이 전달된다면, 그 이웃 노드

에 복제본이 생성될 확률이 높다.

앞에서 언급한 바와 같이 이웃 노드의 수가 많은 클러스터 헤더 노드는 많은 데이터 요청 메시지와 데이터 응답 메시지를 받으므로 부하가 다른 노드에 비해 높다. 그러므로 이 클러스터 헤더에 콘텐츠를 복사하는 것은 해당 노드를 과부하로 만들기가 쉽다. 클러스터 헤더 노드는 클러스터에 있는 모든 노드들과 통신하므로 클러스터 헤더가 과부하 상태에 빠지면 주고 받는 메시지의 지연과 같은 심각한 문제가 발생한다.

따라서 이웃 피어가 많은 클러스터 헤더에는 이웃 노드들의 주소 정보를 캐쉬에 저장하고, 이들을 관리하기만 하며, 클러스터 헤더의 이웃 노드중에 하나를 선택하거나 게이트웨이 노드에 복제본을 저장한다.

게이트웨이 노드는 클러스터 헤더에게 자신이 가지고 있는 복제본에 대한 정보를 메시지를 통해 보냄으로써 클러스터 헤더는 항상 최신의 게이트웨이 노드에 대한 정보를 가지고 있다.

데이터 요청자는 서비스 검색을 통하여 데이터 제공자나 복제본의 위치를 알고 있으므로 데이터 복제를 위한 복제 요청 전송 메시지를 보낸다. 요청된 데이터의 복사가 완료되면 해당 노드는 자신이 또한 데이터 제공자가 될 수 있다.

4.4 캐쉬 일관성 유지기법

캐쉬 일관성을 유지하기 위하여 본 논문에서는 타임스탬프 기법을 기반으로 한다. 이 모델에서는 라우팅 노드는 만일 그것의 사용 가능 시간이 초과되지 않았다면 적절한 값으로 간주한다. 만일 사용 가능 시간값이 초과되었다면 데이터가 유효하지 않으므로 그 노드는 테이블에서 제거된다. 이 모델에서 만일 지나가는 노드가 같은 값을 가지고 있다면 새로운 값으로 수정하여 가장 최근의 정보를 유지한다. 만일 새로운 데이터가 더 최근의 시간정보를 가지고 있다면, 그 노드는 단지 시간정보 필드만 수정하고 그렇지 않다면 삭제한다. 또, 만일 캐쉬 사이즈가 꽉 차서 새로운 서비스를 추가할 수 없다면 자주 참조되는 참조성을 기반으로 하여 LRU 삭제정책에 따라 자주 참조되지 않는 데이터의 정보는 제거된다.

V. 성능 평가

이번 장에서는 본 논문에서 제안하는 탐색과 복제 프

로토콜(neighbor)의 성능을 평가한다. 제안한 프로토콜을 기존의 플러딩 방식, DHT 기반의 방식과 비교한다.

5.1 실험 환경

MANET 환경에서의 탐색 및 복제의 시뮬레이션을 위해 대표적인 네트워크 시뮬레이션 도구인 NS2 v2.32[18]를 사용하였다.

네트워크 영역의 크기는 1,500m x 1,500m를 사용하였고, 라우팅 프로토콜로 DSDV와 AODV프로토콜[14]을 사용하였다. 노드의 수는 20개에서 100개까지 변화를 주어가며 실험하였다. MANET에서 시뮬레이션 성능에 가장 큰 영향을 미치는 것 중 하나가 이동 노드들의 이동 패턴이다. 이동 노드의 이동 시나리오는 평균 정지 시간 2초와 (평균 이동 속도, 최대 이동 속도)를 각각 (3-20, 5-30) 으로 변화하여 이들을 성능평가에 사용하였다. 표1은 본 논문의 알고리즘을 위한 시뮬레이션에 사용된 환경 변수 및 값들을 나타낸다.

표 1. 성능 평가를 위한 실험 환경 변수
Table 1. Simulation parameter

환경 변수	값
노드 수	20 ~ 100
영역 크기	1,500m x 1,500m
평균 쿼리 요청 시간	5s
라우팅 프로토콜	DSDV, AODV
평균 이동 속도	3 ~ 20(m/s)
최대 이동 속도	5 ~ 30(m/s)
이동성 모델	Random Way Point ^[17]
무선 대역폭	1Mbps
전송 거리	250m
MAC 프로토콜	Mac/802_11

5.2 실험 결과

캐쉬 적중률(cache hit ratio)은 캐쉬 사용의 효율성을 평가하기 위한 측정 기준으로, 검색에 적중한 캐쉬데이터의 수를 전체 캐쉬 데이터의 수로 나누어 구한다. 여기에서 캐쉬 데이터란 복제본과 캐쉬 모두를 가리킨다. 만일 캐쉬 적중률이 낮다면, 그만큼 사용되지 않는 캐쉬 데이터의 비율이 높다는 의미이므로 알고리즘의 효율성이 떨어진다고 평가할 수 있다. 캐쉬 적중률은 식(1)과 같이 구할 수 있다.

$$\frac{\text{검색에 사용된 (복제본 + 로컬 캐쉬)}}{(\text{복제본} + \text{로컬 캐쉬})\text{의 수}} \quad (1)$$

그림 4는 데이터 요청자가 데이터를 찾을 때 데이터에 대한 정보가 로컬에 있을 확률을 나타낸다. 이 로컬 캐쉬에 저장된 데이터 정보들은 다른 노드들이 이 데이터 정보를 필요로 할 때 유용하게 사용될 수 있다. 왜냐하면 데이터를 필요로 하는 노드가 데이터를 제공하는 데이터 제공자에게까지 도착하지 않고도 이 캐쉬된 정보를 이용하여 응답하게 함으로써, 여행하는 경로의 수를 줄여, 전체 네트워크 로드를 줄일 수 있다. 또한 더 많은 응답 메시지를 응답할 수 있으므로 성공률도 늘릴 수 있다.

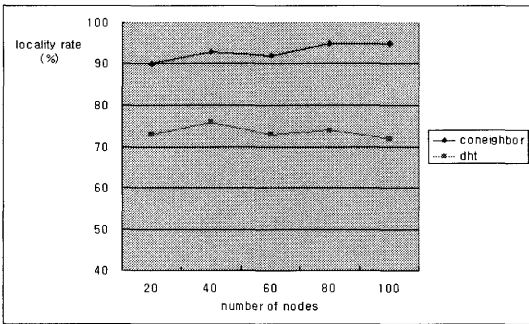


그림 4. 노드의 캐쉬 히트율
Fig. 4. Locality ratio of nodes

평균검색거리는 알고리즘의 성능을 평가하기 위한 측정기준으로, 검색에 소요된 평균 홉 수로 구한다. 이는 데이터요청자가 데이터 요청 메시지를 보낸 후 데이터 응답 메시지를 받을때까지의 거리로 거리가 짧을수록 빨리 검색을 성공했으며, 주고 받는 메시지 수도 작아 질 수 있다. 평균 검색 길이는 식(2)과 같이 구할 수 있다.

$$D_{avg} = \frac{\sum \text{hop of first success}}{\text{the number of query}} \quad (2)$$

그림 5는 평균 탐색 거리를 보인다. 제안한 방식이 플러딩 방식이나 DHT에 비해 현저히 적은 탐색 거리를 가진다. 데이터 제공자의 이웃 노드들은 데이터 제공자가 제공하는 서비스 정보를 알고 있다. 그러므로 데이터 요청 메시지는 더 적은 거리(홉)를 여행해 탐색된다. 원하는 서비스를 찾게 되면 데이터 요청 메시지는 더 이상 다른 노드에게 전송하지 않는다.

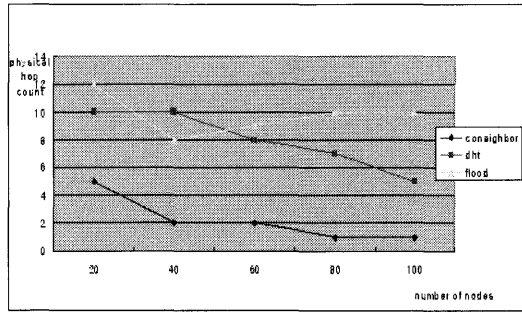


그림 5. 평균 탐색 거리
Fig. 5. Average Search Distance

검색 성공률은 알고리즘의 성능과 적응성을 평가하기 위한 측정 기준으로, 생성된 전체 질의중 데이터 검색을 성공한 질의의 비율이다. 검색 성공률은 식(3)과 같이 구할 수 있다. 만일 데이터 광고시 데이터에 대한 정보가 네트워크상에 효율적으로 배치된다면, 검색 성공률이 높아질 것이다.

$$\text{Success rate} = \frac{\text{the number of success query}}{\text{the number of total query}} \quad (3)$$

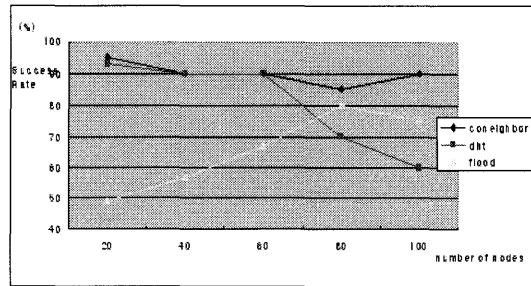


그림 6. 성공률
Fig. 6. Success rate

그림 6은 노드수의 변화에 따른 성공률이다. 노드의 수가 많아질수록 주고 받는 메시지의 수가 많아지므로 중간에 메시지가 삭제되어, 성공률이 낮아진다. 그러나 비교하는 프로토콜에 비해 현저히 성공률이 높음을 알 수 있다.

네트워크 로드는 알고리즘의 성능을 평가하기 위한 측정기준으로 데이터 검색 메시지 수를 말한다. 이는 데이터 제공자가 데이터 광고 메시지를 보낸후, 이를 받은 d-홉 이웃 노드들은 이를 캐쉬하여 데이터 검색 요청을

받았을때, 유용하게 사용될 수 있어 네트워크 로드를 줄일 수 있다. 그림 7은 주고 받은 메시지를 나타내는 그림이다. 플러딩 기법에서는 주고받은 메시지의 수가 탐색을 위해 이웃 노드에게 브로드캐스팅함으로써 노드의 수가 많아질수록 급격하게 늘어나다. DHT-기반 방식에서는 논리적 이웃 노드로 메시지를 전송함으로써 물리적 노드를 찾기 위한 오버헤드가 발생함으로써 많은 메시지를 주고 받게 된다. 두 가지 비교 방식에 비해 본 논문에서 제안하는 방식은 적은 수의 메시지만을 주고받는다.

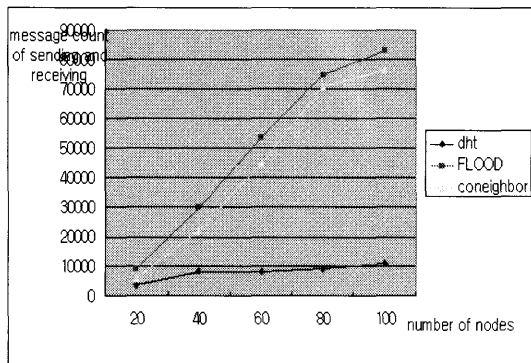


그림 7. 주고 받은 메시지 수
Fig. 7. Message count of sending and receiving

VI. 결론

기존의 방법에서는 플러딩이 주고받는 메시지가 많아 모바일 애드-혹 네트워크에서 많은 비용을 지불해야하는 어려움이 많았다.

본 논문에서는 어려움을 개선하기 위하여 지역적인 정보만을 이용하여 서비스를 효과적으로 찾을 수 있는 서비스 탐색 기법을 제안한다. 제안한 방법은 높은 이동성, 자원 제약성이 있는 모바일 디바이스들이 고정 기반 없이 무선 네트워크로 이루어진 모바일 애드-혹 네트워크에 적합하다고 본다. 평균탐색거리를 줄인다는 것은 응답 시간을 줄이는 것이며, 응답 지연을 줄이는 것이 되고, 주고받는 메시지 수를 줄인다는 것은 전체 네트워크 통신 비용과 노드의 부하를 줄여 효과적이다.

제안한 방법에서 얻은 실험 결과는 평균 탐색 거리를 줄이고, 성공률을 높여줌으로써, 주고받는 메시지의 수를 줄여 높은 효과를 보여주게 되었다.

참고 문헌

- [1] C.E. Perkins, Ad Hoc Networking, Addison-Wesley, 2001
- [2] C.K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall, 2002
- [3] Perich, F. Joshi, A.; Finin, T. and Yesha, Y., "On data management in pervasive computing environments", Knowledge and Data Engineering, IEEE Transactions on, Volume 16, Issue 5, pp. 621 - 634, May 2004.
- [4] Edith Cohen and Scott Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks" Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, Vol. 32, Issue 4, pp. 177-190, August, 2002.
- [5] Qin Lv, Pei Cao, Edith Cohen, Kai Li and Scott Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks", Proceedings of SIGMETRICS 2002, pp.258-289, June, 2002
- [6] Saroiu, S., Gummadi, P. K., and Gribble, S.D.A. "Measurement Study of Peer-to-Peer File Sharing Systems." Proceedings of Multimedia Computing and Networking, 2002
- [7] The Gnutella web site, www.gnutella.com
- [8] The KaZaA web site, www.kazaa.com
- [9] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., Balakrishnan, H. "Chord: A scalable peer-to-peer lookup service for internet applications.", SIGCOMM' 01: Proceeding of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, ACM Press (2001) pp. 149-160
- [10] Ratnasamy, S., Francis, P. Handley, M., Karp, R., Shenker, S., "A scalable content addressable network" SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer

- communications, NewYork, NY, USA, ACM Press(2001) pp.161-172
- [11] Rowstron, A.I.T., Druschel, P. "Pastry:Scalable, decentralized object location, and routing for large-scale peer-to-peer systems" *Middleware*. (2001) 329-350
- [12] Gopalakrishnan, V., Silaghi, B., Bhattacharjee, B., and Keleher, P., "Adaptive Replication in Peer-to-Peer Systems", *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pp. 360 - 369, 2004
- [13] Sanjay Madria, and Sanjeev Agarwal, "Adaptive replication and access control of multimedia data in a P2P environment", *Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA'05)*, pp. 764 - 768, 22-26 Aug. 2005
- [14] C.E. Perkins, E.M Royer, and S.Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing." RFC3561, IETF, Jul. 2003
- [15] Klemm. A., Lindemann. C., and Waldhorst. O. P., "A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks" *Vehicular Technology Conference, 2003, VTC 2003-Fall, 2003 IEEE 58th*, vol.4, pp. 2758-2763
- [16] Pradeep K. Sinha, *Distributed Operating Systems Concepts and Design*, IEEE Press, 1997
- [17] Karen Q. Tian and Donald C. Cox. *Mobility management in wireless networks - Data Replication Strategies and Applications*, Kluwer Academic Publishers, 2004
- [18] NS2 web site, www.isi.edu/nsnam/ns

저자 소개

강 은 영(정회원)

제2권 제1호 참조

- 현재 성균관대학교 정보통신학부 박사과정 수료

<주관심분야> : 모바일애드혹네트워크, 서비스디스커버리, 유비쿼터스 헬스케어 등

임 용 순(종신회원)

제2권 제1호 참조

- 현재 국제대학 방송학부 방송영상제작전공 교수

<주관심분야> : 방송영상통신, 영상처리, 모바일시스템 등