

논문 2008-45TC-12-9

MANET에서 가상 위치 기반 라우팅을 위한 지역 분산 해쉬 테이블 적용 방법

(A Geographic Distributed Hash Table for Virtual Geographic Routing
in MANET)

고 석 갑*, 김 영 한*

(Seok Kap Ko and Young Han Kim)

요 약

본 논문에서는 MANET(Mobile Ad hoc Network)에서 가상 위치 기반 라우팅을 위한 GDHT(Geographic Distributed Hash Table)를 적용하는 방법을 제안한다. 기존의 GDHT는 저장하고자 할 데이터를 가상 2차원 직교좌표계에 균일하게 매핑 하였다. 그러나, 위치 기반 라우팅을 사용하는 네트워크에서 노드의 분포는 기존 GDHT의 가상 공간과 다르게 분포한다. 따라서, 기존의 GDHT를 바로 적용할 경우, 많은 데이터들이 네트워크의 외곽 또는 특정 노드에 위치하게 된다. 이는 데이터 또는 서비스 검색 거리 및 시간을 늘리는 문제를 발생시킨다. 또한, BVR(Beacon Vector Routing)이나 LCR(Logical Coordinate Routing) 좌표 요소 사이에는 서로 연관관계가 있기 때문에 일반적인 해쉬 함수를 적용할 경우, 존재할 수 없는 좌표에 매핑될 가능성이 높아진다. 이러한 문제를 해결하기 위하여, 데이터들의 분포를 제어할 수 있도록 하고, 좌표 요소 사이의 연관관계가 반영될 수 있는 “Geographic 해쉬 함수”를 제안한다. 분석과 시뮬레이션을 통하여 제안하는 방법이 데이터 분포를 제어할 수 있고, 데이터 검색 성능을 향상시킬 수 보인다.

Abstract

This paper presents a new geographic distributed hash table (GDHT) for MANETs or Mesh networks, where virtual geographic protocol is used. In previous work, GDHT is applied to a network scenario based on two dimensional Cartesian coordinate system. Further, logical data space is supposed to be uniformly distributed. However, mobile node distribution in a network using virtual geographic routing is not matched to data distribution in GDHT. Therefore, if we apply previous GDHT to a virtual geographic routing network, lots of DHT data are probably located at boundary nodes of the network or specific nodes, resulting in long average-delay to discover resource (or service). Additionally, in BVR(Beacon Vector Routing) or LCR(Logical Coordinate Routing), because there is correlation between coordinate elements, we cannot use normal hash function. For this reason, we propose to use “geographic hash function” for GDHT that matches data distribution to node distribution and considers correlation between coordinate elements. We also show that the proposed scheme improves resource discovery efficiently.

Keywords : Peer-to-Peer(P2P), MANET, GDHT, Geographic Routing

I. 서 론

최근, Peer-to-Peer(P2P) 네트워크 및 그 응용은 다양한 분야로 확장되고 있다. MANET 또는 무선 메쉬

네트워크는 무선 통신 기술을 이용하여 노드들 간에 멀리 흩어져 연결된 네트워크이다. 기존의 P2P 네트워크의 개념을 MANET에 적용하면 더욱 효율적이고 다양한 서비스를 제공할 수 있다. MANET에 P2P 기술을 적용할 때의 문제는 P2P 오버레이 네트워크가 MANET 토폴로지를 제대로 반영하지 못하는 것이다. 이러한 문제를 해결하기 위하여 접근성을 고려하여 오버레이를 구성하는 여러 가지 방법이 연구되었다^[1-3].

정희원, 숭실대학교
(Soongsil University)

※ 본 논문은 일부 숭실대학교 교내연구비과제로 지원된 것임

접수일자: 2008년7월1일, 수정완료일: 2008년12월10일

최근에는 네트워크 계층과 P2P 계층을 통합하려는 교차 계층적인 접근(Cross layer approach) 방법으로 연구가 진행되고 있다^[4].

MANET 라우팅 알고리즘은 최단 경로(Shortest Path) 방식, 계층적 주소(Hierarchical Addressing), 위치기반(Geographic) 라우팅으로 분류할 수 있다. 최단 경로 방식에는 DSR, AODV, DYMO, OLSR 등이 있지만, 확장성(Scalability)이 없다. 계층적 주소방식은 복잡하며 실제 적용에 어려움이 있다. 위치기반 라우팅은 노드의 현재 위치를 이용하여 자기 이웃 노드 중 목적지에 가장 가까운 노드로 메시지를 전달하는 방식으로 각 노드는 자기 이웃의 위치 정보만 유지하면 되므로 확장성이 뛰어나다^[5-6].

위치 기반 라우팅은 노드의 현재 위치를 얻기 위해 GPS 등의 부가 장치를 이용한다. 하지만, 실제 좌표를 이용하는 방법은 장치 추가에 따른 부가 비용뿐만 아니라 실내와 같은 장소에서 적용하기 어려운 문제, 장애물 등으로 인한 라우팅 실패 문제 등 성능 문제를 가진다. 그래서 노드들 간의 연결성을 이용하여 가상 좌표계를 구성하는 가상 위치 기반 라우팅이 연구되었다^[7-9].

위치 기반 라우팅은 노드의 현재 위치를 알아오는 과정이 필요하다. 이러한 서비스를 위해 정보를 네트워크에 분산시켜 저장하는 분산 해쉬 테이블(DHT, Distributed Hash Table)를 사용할 수 있다. P2P 데이터 역시 DHT를 사용하면 효율적으로 자료를 저장, 검색할 수 있다. 해쉬 함수를 이용하여 데이터들을 가상 위치로 매핑하고, 그 위치와 가장 가까운 위치에 있는 실제 노드에 저장한다. 데이터들이 지역적으로 분산되도록 하기 때문에 이를 GDHT(Geographical DHT)라고 한다^[10].

CAN(Contents Addressable Network)^[11]과 같은 기존의 GDHT에서는 균일한 데이터 분포를 정사각형의 가상 공간을 이용하였다. 그러나, 가상 위치 기반 라우팅에서의 노드들의 분포는 원형에 가까우며, 좌표 요소 간 연관 관계를 가지고 있다. 따라서 기존의 GDHT를 적용하는 경우, 잘못된 위치에 데이터를 매핑하거나 데이터가 네트워크의 외곽 노드 또는 특정 노드에 집중된다. 이는 P2P 데이터 검색 성공률을 낮추고, 평균 검색 거리 및 시간을 늘리는 문제를 가져 온다.

본 논문에서는 가상 위치 기반 라우팅의 특성을 고려한 GDHT 적용방법을 제안한다. 본 논문의 나머지 부분은 다음과 같이 구성된다. II장에서, 가상 위치 기반

라우팅에 대해 간단히 설명한다. III장에서, 가상 위치 기반 라우팅에 맞는 GDHT를 적용하는 방법에 대해 설명한다. IV장에서, 제안하는 방법을 간단히 분석하여 평가한다. V장에서 시뮬레이션을 통하여 제안하는 알고리즘을 사용했을 때의 성능향상을 보인다. 마지막으로 VI장에서 앞으로 할 일과 결론을 맺는다.

II. 가상 위치 기반 라우팅

1. NoGeo

NoGeo 논문^[7]에서는 시야계(Perimeter) 노드가 실제 위치를 알고 있는 경우와 실제 위치를 알지 못하는 경우로 나누어 가상 좌표계를 구성하는 방식을 제안하였다. GPS 등을 통해 시야계 노드들의 실제 위치를 알고 있는 경우, 일반 노드들은 시야계 노드 및 이웃 노드들의 좌표의 평균을 이용하여 자기의 좌표를 구한다. 시야계 노드가 실제 위치를 모를 경우, 중앙의 시야계 노드로부터 가장 멀리 떨어진 노드들이 시야계 노드가 되고, 시야계 노드 간 메시지 교환을 통하여 서로 간의 거리를 얻는다. 그리고, 삼각 측량법을 이용하여 네트워크의 중심을 원점으로 하는 (x, y) 좌표를 구한다. 일반 노드도 이 과정에서 대략적인 좌표를 얻은 후 이웃 노드들의 좌표의 평균을 이용하여 자기 좌표를 갱신한다.

라우팅은 이렇게 구한 좌표를 이용하여 자기 2 홉 이웃 중에 목적지와 가장 가까운 노드로 메시지를 전달하는 Greedy Routing 방식을 사용한다.

2. BVR(Beacon Vector Routing) 방식

BVR^[8]은 Beacon 노드들과의 거리 벡터를 이용하여 좌표처럼 사용한다. 각 Beacon 노드들은 주기적으로 Beacon 메시지를 플러딩하여 각 노드들은 Beacon 노드로부터의 거리를 알게 된다. Beacon 노드수가 b일 경우, 임의의 노드 q의 위치 P(q)는 다음과 같이 각 beacon 노드로부터의 거리로 표현된다.

$$P(q) = \langle q_1, q_2, \dots, q_b \rangle \quad (1)$$

BVR에서 메시지 라우팅은 Greedy Routing을 사용한다. 즉, 이웃 노드들 중 목적지에 가장 가까운 노드에게 메시지를 전달한다. 현재 노드 p로부터 목적지 노드 d까지의 거리는 다음 공식(2)~(4)를 통해 계산된다.

$$\delta_k^+(p, q) = \sum_{i \in C_k(d)} \max(p_i - d_i, 0) \quad (2)$$

$$\delta_k^-(p, q) = \sum_{i \in C_k(d)} \max(d_i - p_i, 0) \quad (3)$$

$$\delta_k = A\delta_k^+(p, q) + \delta_k^-(p, q) \quad (4)$$

여기서 $C_k(d)$ 는 노드 d 에 가장 가까운 k 개의 Beacon 노드들의 집합을 나타낸다. δ_k^+ 는 p 보다 d 에 가까운 Beacon과의 거리의 합을 나타내며, δ_k^- 는 d 보다 p 에 가까운 Beacon과의 거리의 합을 나타낸다. 상수 A 는 δ_k^- 에 가중치를 주는 역할을 하며 10을 사용하였다. 그래서 노드는 자기 이웃 노드들중 δ_k 를 최소화하는 노드를 다음 홉으로 선택한다. 만약 주변에 더 이상 δ_k 를 감소시켜주는 노드가 없는 경우, Fallback 모드로 전환되어 목적지와 가장 가까운 Beacon 노드로 전달한후, 플러딩을 수행한다. BVR은 이동성을 고려한 Fault Tolerant BVR^[12]로 더 보완되었다.

3. LCR(Logical Coordinate Routing) 방식

LCR^[9]은 BVR과 유사하다. BVR에서는 20개 이상의 많은 Beacon 노드를 사용하는 것에 반해 LCR은 4개 정도의 Landmark 노드들을 선정하여 사용한다. Landmark 선정 방식은 후보자들 중에 가장 서로 멀리 떨어져 있는 Landmark를 선정하여 높은 정확도를 갖도록 한다. 또한 BVR에서는 Greedy 라우팅이 실패할 경우 제한된 범위 내에서 플러딩을 수행하여 목적지에 도착하는 데 반하여, LCR에서는 역추적(Backtracking) 알고리즘을 사용하여 플러딩 없이 실패한 경로를 우회하는 방법을 사용한다. LCR의 역추적 알고리즘은 경로가 없을 때 지나왔던 경로를 다시 되돌아가서 다른 경로로 시도하는 방식이다.

LCR에서 $V(V_1, V_2, \dots, V_n)$ 과 $W(W_1, W_2, \dots, W_n)$ 좌표를 갖는 두 노드 사이의 거리를 계산하는 방법은 다음과 같다. n 은 선정된 Landmark 노드의 수이다.

$$D = \sqrt[n]{\sum_{i=1}^n (|V_i - W_i|)^n} \quad (5)$$

여기서 N 은 실험을 통해 2를 사용했을 때 좋은 성능이 나타났다. BVR과 마찬가지로 LCR도 이웃 노드 중 목적지와 거리가 가장 가까운 노드를 선택하여 메시지를 전달한다.

LCR은 신뢰할 수 없는 링크를 고려하여, 링크의 품질을 거리 메트릭에 반영하여 불안한 경로를 덜 선택하도록 확장하였다.

III. 가상 위치 기반 라우팅을 위한 GDHT 적용 방법

1. GDHT 적용 방법

가상 위치 기반 라우팅 네트워크에 GDHT를 적용하는 방법은 다음과 같다. 먼저, 위치 정보를 저장할 노드의 ID 또는 저장하고자 할 P2P 데이터의 키를 해쉬 함수를 이용하여 논리적인 가상 좌표 공간에 어떤 점으로 매핑 시킨다. 이 좌표에 가장 가까운 위치에 있는 실제 노드가 이 데이터를 저장하도록 한다. 저장하고자 할 P2P 키와 데이터는 위치 기반 라우팅 알고리즘을 이용하여 매핑된 좌표에 가까운 노드까지 전달할 수 있다. 이 노드는 노드의 이동에 따른 데이터 손실을 방지하기 위하여 제한된 범위의 플러딩을 수행하여 사본을 전파한다. 동일한 해쉬 함수를 사용하여 검색하고자 하는 키에 대한 가상 좌표 공간에 매핑 되는 위치를 얻을 수 있으며, 위치 기반 라우팅 알고리즘을 통해 키를 저장하고 있는 노드에 접근후 확장 링 검색을 통해 저장된 데이터를 찾을 수 있다.

2. 가상 위치 기반 라우팅 특성 고려

가상 위치 기반 라우팅에 GDHT를 적용할 때, 다음과 같은 3가지 사항을 고려하여야 한다.

1. DHT를 위한 논리적 공간과 실제 공간의 크기. 논리적인 공간과 실제 공간의 크기가 일치하지 않을 경우, P2P 데이터의 분포가 네트워크 외곽 노드 또는 특정 노드에 집중되게 된다. 이는 평균 검색 길이를 늘리며, 로드 분산이 이루어지지 않게 한다.

2. 노드의 분포 특성. 라우팅 알고리즘에 따라 노드의 분포 특성이 다르다. 논리적 공간과 실제 공간의 노드 분포 특성이 일치하지 않을 경우에도 일부 노드들에 데이터가 집중되게 되어 성능저하를 가져온다.

3. 좌표계가 갖는 연관 관계. BVR이나 LCR 같은 경우에는 각 좌표요소 간에 상호 연관 관계를 가지고 있다. 따라서 임의로 좌표를 선정할 경우, 존재할 수 없는 좌표가 만들어지게 된다. 존재할 수 없는 좌표에 매핑된 데이터는 실제 가장 가까운 노드에 데이터를 저장할 수 없게 되며, 검색 성공률을 낮추게 한다.

NoGeo와 같은 직교 좌표계를 사용하는 경우, GDHT 적용은 비교적 간단하다. CAN과 같은 기존의 DHT는 정사각형 논리 공간을 사용한다. 하지만, NoGeo에서 실제 노드의 배치는 원형에 가깝다. 따라서 매핑 해쉬 함수의 결과가 원형의 논리 공간을 갖도록 하여야 한다.

BVR이나 LCR에서는 노드의 분포가 균일하지 않으며, 좌표 요소간의 연관관계를 가지고 있다. 따라서 매핑 함수는 이러한 특성을 반영하도록 해야 한다.

Beacon 노드가 주기적으로 전송하는 Beacon 메시지의 TTL을 L 이라고 하자. 하나의 Beacon으로부터의 거리는 0부터 L 까지 값이 될 것이고, 전체 b 개의 Beacon 노드가 있으므로, BVR에서의 논리적인 공간의 크기는 $(L+1)^b$ 이 된다. 이 공간에 노드 ID 또는 P2P 데이터의 키를 해쉬하여 데이터를 매핑시킬 수 있다. 그러나, 실제 노드들의 분포는 Beacon 거리에 따라 균일하게 분포하지 않는다. 그림 1은 실제 물리적인 공간에서의 노드들의 분포 범위를 보인다. 모든 노드들은 반경 r 을 가진 원의 내부에 균일하게 분포한다. 또한 모든 노드들은 모든 Beacon 노드들로부터 L 거리 이내에 위치한다. 이는 모든 Beacon 노드들은 원의 중심으로부터 $L/3$ 범위 내에 위치함을 의미한다. 중심 Beacon으로부터의 최대 거리는 L 이 아닌 $2L/3$ 이 되며, 원의 반경 r 은 최대 $2L/3$ 이 된다. 그림을 통하여, Beacon 거리에 따른 노드의 분포는 $2L/3$ 이상의 영역에서 상대적으로 적게 분포함을 알 수 있다.

LCR이나 BVR에 GDHT를 적용할 때의 또 다른 문제는 좌표 요소 간의 연관성이다. 즉, 각 Landmark 또는 Beacon 노드로부터 임의의 거리를 선택할 경우, 그 위치는 존재할 수 없는 위치가 될 수 있다. 예를 들어, 3개의 Beacon 노드가 서로 10 홉 거리를 두고 위치한다고 할 때, 위치 $\langle 10,10,10 \rangle$ 은 존재할 수 없는 좌표가 된다. 그러므로 가상 위치를 선정할 때, Beacon 노

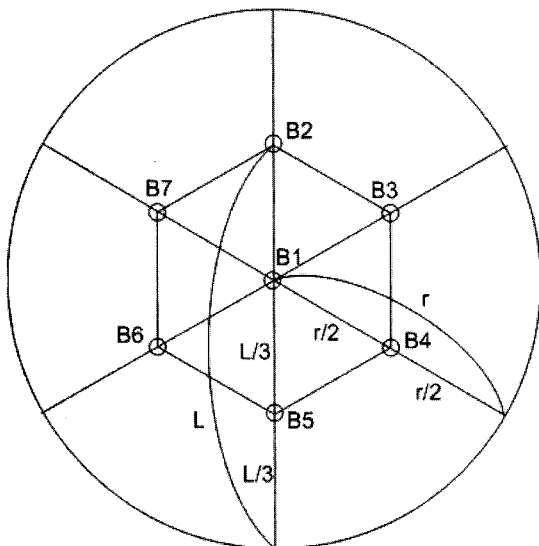


그림 1. 실제 공간에서의 BVR 노드 분포 범위
Fig. 1. BVR node distribution range in physical space.

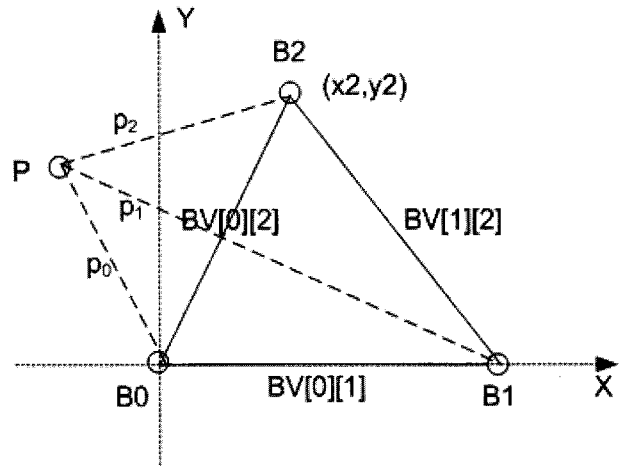


그림 2. G-해쉬를 위한 가상 2차원 공간
Fig. 2. Virtual 2D space for G-hash.

드들 간의 거리를 고려하여야 한다. 이 문제를 해결하기 위해, 본 논문에서는 Beacon 노드들 간의 거리를 고려한 가상 2차원 직교좌표계를 사용한다. 그림 2는 이러한 가상 직교좌표계를 보인다.

먼저 가상 좌표계에서 Beacon 0를 선택하여 원점에 둔다. Beacon 0는 다른 Beacon 노드들과의 거리가 최소인 Beacon 노드이다. 다음에, 정확성을 위하여 Beacon 0로부터 가장 멀리 떨어진 Beacon 1을 선택하여 가상 좌표계의 $x > 0$ 인 X축 위에 위치하도록 한다. 그리고 그 다음 멀리 떨어진 Beacon 2를 선택하여 $Y > 0$ 위치에 둔다. Beacon 2의 위치는 삼각측량법을 통하여, Beacon 0 및 Beacon 1과의 거리에 따라 $Y > 0$ 인 위치 또는 $Y < 0$ 인 위치에 둘 수 있지만, 좌표계의 Y축 기준을 위하여 $Y > 0$ 에 두어, 다른 노드들의 위치가 $Y > 0$ 인지 $Y < 0$ 에 있는지 결정하는데 사용된다. 이 기준 Beacon 노드들과의 거리를 이용하여 다른 Beacon 노드들의 가상 2차원 공간상의 위치를 계산할 수 있다. 이 계산 결과는 Beacon 0가 주기적으로 전송하는 Beacon 메시지에 포함하여 모든 노드가 공유한다. 이제 가상 2차원 공간상의 임의의 점을 선택했을 때, Beacon 노드들과의 거리를 구할 수 있으므로, 유효한 BVR 또는 LCR 좌표를 얻을 수 있다.

무선 메쉬 네트워크에서는 고정된 무선 노드가 존재하며, 이를 Beacon 노드로 활용할 수 있고, 따라서 노드 분포를 예측하기 쉽다. 그러나, MANET에서는, 고정된 노드가 없기 때문에, Beacon 노드들이 자동으로 임의로 선택된다. 따라서 노드의 분포를 예측하기 어렵다. 본 논문에서는 Beacon 노드들이 네트워크 중앙에 위치하는 것으로 가정한다. 이것은 평균적인 Beacon 노

드 분포가 되므로 적절한 가정이 된다. 그림 1에서 보였듯이, 중심 Beacon 노드으로부터의 어떤 노드와의 거리는 0에서 $2L/3$ 이 된다. 노드 분포에 일치시키기 위해 논리적인 공간의 범위도 0에서 $2L/3$ 로 한다.

3. Geographic-Hash 함수 설계

본 절에서는 앞에서 설명한 가상 위치 기반 라우팅의 특성을 고려하여, 가상공간의 크기와 분포를 제어할 수 있고, 유효한 가상 좌표를 얻을 수 있는 비균일 해쉬 함수를 설계한다. 본 논문에서 이를 "Geographic 해쉬 함수(G-hash)"라고 정의한다. 이 함수 GH(key)는 key를 유효한 BVR 좌표 또는 LCR 좌표로 매핑시킨다. 각각의 좌표 요소는 h_i 로 표시된다.

$$GH(key) = \langle h_1, h_2, \dots, h_b \rangle \quad (6)$$

$$u_r(key) = \frac{Hash_r(key)}{MAX_HASH_VALUE} \quad (7)$$

$$u_\theta(key) = \frac{Hash_\theta(key)}{MAX_HASH_VALUE} \quad (8)$$

$$r_i = R_h(u_r(key))^\sigma, \quad \theta_i = 2\pi u_\theta(key) \quad (9)$$

$$x_i = r_i \cos \theta_i, \quad y_i = r_i \sin \theta_i \quad (10)$$

$$h_i = \left[\sqrt{(x_i - x_{bi})^2 + (y_i - y_{bi})^2} \right] \quad (11)$$

$Hash_r(key)$ 와 $Hash_\theta(key)$ 는 SHA-1과 같은 일관 해쉬 함수를 사용할 수 있으며, 두 함수는 서로 다른 함수이다. 예를 들어, $Hash_r(key)$ 로서 SHA-1(key)를 사용하고, $Hash_\theta(key)$ 로는 SHA-1(key+c)과 같이 사용하여 키 값에 어떤 상수 c를 더하여 SHA-1을 사용할 수 있다. u_r 과 u_θ 는 키 값에 따라 0~1사이의 균일한 임의의 값이 된다. 여기에 반지름과 각도의 범위를 곱하여 r_i 와 θ_i 를 얻는다. r_i 를 구할 때, 지수 σ 를 사용하여 반지름에 따른 데이터의 분포를 조절할 수 있다. 본 논문에서는 이를 분포 계수라고 한다. 분포 계수 0.5를 사용할 경우, 유니폼 분포를 거리에 비례하는 분포로 바꿀 수 있으며, 데이터는 원내에서 균일하게 분포하게 한다. 분포 계수가 클수록, 데이터는 중심에 많이 집중된다. R_h 는 논리적인 공간의 크기를 제어한다. 앞 절에서 설명한 것과 같이 BVR의 경우, R_h 값으로 $2L/3$ 을 사용한다. LCR의 경우 중심으로부터 가장 먼 Landmark 노드와의 거리를 이용할 수 있다. 삼각함수

를 이용하여 r_i 와 θ_i 로부터 x_i 와 y_i 를 구할 수 있다. 마지막으로, 각 Beacon 노드의 가상 2D 좌표 x_{bi} , y_{bi} 와의 거리를 계산하여 BVR 또는 LCR 좌표인 h_i 를 구한다.

일반적인 분포에 대해 이와 같은 G-Hash를 통하여 키 값에 따른 좌표를 얻을 수 있다. 만약 네트워크의 노드 분포를 알고 있고 일반적인 상황과 다른 경우, G-Hash를 일부 수정하여 쉽게 적용할 수 있다. NoGeo의 경우에는 좌표 변환이 필요 없기 때문에 G-Hash 공식중 (7)~(10)만 사용하면 된다.

IV. 분석

이 장에서는 논리적인 공간이 물리적인 공간과 차이가 있을 경우, 평균 검색 길이가 어떻게 바뀌는지 살펴본다.

길이 L을 갖는 일직선 상의 한 쪽 끝에 Beacon 노드가 위치하고, L 개의 노드들이 균일하게 위치한다고 하자. 여기에 데이터 n 개의 데이터를 균일하게 배포할 경우, 노드당 n/L 개씩 데이터가 저장된다. 임의의 노드가 임의의 데이터를 검색할 때 그 평균 거리는, L 길이를 갖는 선 상에서 임의의 두 점을 뽑았을 때 거리의 평균과 같다. 분석의 편의를 위해 L=1로 두고, 첫 번째 뽑히는 점의 위치를 랜덤 변수 X, 두 번째 뽑히는 점의 위치를 랜덤 변수 Y라고 하면, 기대치를 다음과 같이 계산할 수 있다.

$$E\{g(X, Y)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) f_{XY}(x, y) dx dy \quad (12)$$

랜덤 변수 X, Y는 독립적이며, 균일한 분포를 가지므로, 다음과 같이 정리할 수 있다.

$$f_{XY}(x, y) = f_X(x) f_Y(y) \quad (13)$$

$$f_X(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{elsewhere} \end{cases}$$

$$f_Y(y) = \begin{cases} 1, & 0 \leq y \leq 1 \\ 0, & \text{elsewhere} \end{cases}$$

$$E\{|X - Y|\} = \int_0^1 \int_0^1 |x - y| dx dy = \frac{1}{3} \quad (14)$$

따라서 노드 분포 공간과 데이터의 분포 공간이 일치되어 데이터가 균일하게 분포된 경우, 평균 검색 길이는 $\frac{1}{3}L$ 이다.

만약 데이터의 분포 공간이 $2L$ 로 노드의 분포 공간보다 클 경우, 0부터 $L-1$ 위치의 노드에는 노드당 $n/2L$ 개의 데이터가 저장된다. L 위치의 노드에는 $n/2L+n/2$ 개의 데이터가 저장된다. 즉 데이터의 절반이 L 위치의 노드에 집중된다. 이 경우, L 을 1로 본다면, 데이터의 분포 함수는 다음과 같다.

$$f_Y(y) = \begin{cases} \frac{1}{2}, & 0 \leq y < 1 \\ \frac{1}{2}, & y = 1 \\ 0, & \text{elsewhere} \end{cases} \quad (15)$$

평균 검색 길이는 다음과 같이 구할 수 있다.

$$\begin{aligned} E\{|X - Y|\} &= \int_0^1 \int_0^1 |x - y| \frac{1}{2} dx dy + \frac{1}{2} \int_0^1 (1 - x) dx \\ &= \frac{5}{12} \end{aligned} \quad (16)$$

즉, 평균 검색 길이는 $\frac{5}{12}L$ 이 된다. 이는 데이터 분포 공간과 노드 분포 공간을 일치시켰을 때보다 25% 증가된 값이다.

BVR에서 노드의 분포는 중심 Beacon으로부터의 거리에 비례하여 증가한다. 네트워크가 반지름 R 을 갖는 원에 노드가 균일하게 위치하고 있다고 가정하자. 간단한 시뮬레이션을 통해, 원 내에서 임의의 두 점을 선택했을 때 거리의 평균은 $0.9R$ 이다. 이는 데이터를 노드의 분포와 동일하게 균일하게 분포하도록 하였을 때의 평균 거리가 된다. 즉, G-Hash의 분포 계수 σ 가 0.5로 데이터를 분포 시켰을 경우, 평균 검색 길이는 $0.9R$ 이 된다. 만약 데이터를 중심 Beacon에 더 모이도록 분포 계수 σ 를 1로 한 경우, 평균 검색 거리는 $0.85R$ 이 된다. 노드의 분포를 중심 Beacon에 집중 시킬수록 평균 검색 길이는 줄어들지만, 부하 불균형이 발생하는 문제가 생긴다.

V. 시뮬레이션

이 장에서는 시뮬레이션 결과를 보인다. 시뮬레이션 공간은 200×200 단위 면적을 갖는 정사각형 공간과 112.8 단위 길이의 반지름을 갖는 원형 공간을 사용한다. 이는 홉수로는 14.1이고 $2L/3=14.1$ 이므로, L 은 22이다. 두 공간의 면적은 동일하다. 노드의 통신 범위는 8 단위 길이이다. Beacon 노드들은 공간의 중심으로부터

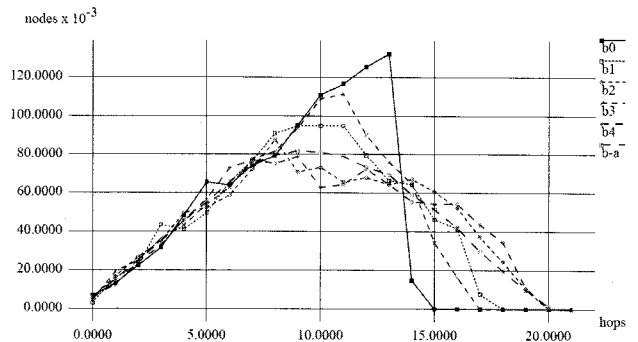


그림 3. BVR에서 노드의 분포
Fig. 3. Node distribution in BVR.

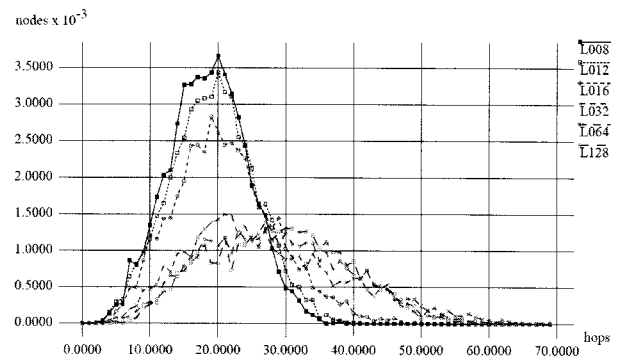


그림 4. 논리 공간 크기별 검색 길이의 분포
Fig. 4. Resource discovery length distribution per logical space size.

$1/2$ 거리 내에 위치한다. 전체 노드 수는 2500개이다. 따라서 노드의 밀집도는 통신 범위당 12.5 노드이다. 라우팅 성공률을 높이기 위하여 각 노드들은 2홉 이웃 테이블을 유지하고 사용한다. 원래 BVR에서 기본 라우팅에 실패한 경우 플러딩을 시도하도록 되어 있으나, 시뮬레이션에서는 성공률을 측정하기 위하여 플러딩을 사용하지 않는다.

그림 3은 원형 공간에서의 Beacon 노드들로부터의 거리에 따른 노드 분포를 보인다. 정사각형 공간의 시뮬레이션 결과도 이와 유사한 형태를 보인다. b_0 는 중심 Beacon 노드로부터의 거리에 따른 노드 분포를 보인다. $b_1 \sim b_4$ 은 해당 Beacon 노드부터의 거리에 따른 노드 분포를 보인다. $b-a$ 는 100개 Beacon들로부터의 평균적인 노드 분포를 보인다. 노드 분포는 $2L/3$ 홉 거리까지는 증가하지만 그 이후로는 줄어들고 있음을 알 수 있다. 특히 중심 Beacon으로부터의 노드 분포는 $2L/3$ 홉 거리 이후에는 노드가 존재하지 않는다.

그림 4는 논리적 공간의 크기에 따른 검색 길이의 분포를 보인다. 논리적 공간의 크기는 G-Hash의 R_h 값을 변경하여 조절한다. 시뮬레이션에서는 5000개의

표 1. 논리적 공간 크기 및 분포 계수에 따른 평균 검색 길이

Table 1. Average discovery length per logical space size and distribution parameter

Rh(Hops) ($\sigma=1.0$)	(1)		(2)	
	Average Discovery Length (Hops)	σ (Rh=16)	Average Discovery Length (Hops)	
8	18.567	0.5	22.818	
12	19.285	1.0	21.040	
16	21.040	2.0	19.178	
24	25.348	4.0	17.792	
32	27.382			
64	28.391			

표 2. Hash 함수에 따른 성공률 및 평균 검색 길이
Table 2. Success ratio and average discovery length per hash function

Function	Rh ($\sigma=0.5$)	Success ratio (%)	Average Discovery Length (Hops)
G-Hash	8	99.04	19.228
	12	99.18	20.125
	16	99.24	22.818
Random	8	97.92	29.231
	12	98.48	27.473
	16	96.48	29.261

데이터를 제안하는 알고리즘을 사용하여 각 노드에 배포한다. 데이터 저장시 목적지에서 2홉 이웃까지 데이터를 저장하도록 하였다. L008 및 L012, L016, L032, L64, L128은 R_h 값을 각각 8, 12, 16, 32, 64, 128로 했을 때의 검색 길이의 분포를 보인다. R_h 값이 16일때까지는 큰 차이가 없다가 그 이후에는 검색 길이가 늘어남을 알 수 있다.

표 1의 (1)부분은 R_h 에 따른 평균 검색 길이를 보인다. 이때 분포 계수 σ 는 1.0으로 하였다. R_h 가 커질수록 데이터가 외곽 노드에 집중 되어 평균 검색 길이가 증가하는 것을 볼 수 있다. 표 1의 (2)부분은 R_h 가 16으로 고정되어 있을 때, 분포 계수 σ 에 따른 평균 검색 길이를 보인다. 분포 계수가 클수록 데이터가 네트워크의 중심에 집중되어 평균 검색 길이가 줄어드는 것을 볼 수 있다. 분포계수를 변경하여 데이터의 분포를 제어할 수 있음을 보인다. 표 2는 G-Hash를 사용한 경우와 일반적인 균일 해쉬 함수를 사용한 경우를 보인다. 일반적인 균일 해쉬 함수를 사용할 경우, 3장에서 설명

한 Beacon 벡터 요소간의 연관 관계를 반영할 수 없다. 따라서 대부분의 데이터가 BVR 라우팅 중 Fallback 모드로 Beacon 노드들에 저장된다. 그러므로, 데이터의 분포를 제어할 수 없게 된다. 표의 결과는, G-Hash를 사용하였을 때 높은 검색 성공률을 보이며, 데이터 분포를 제어하여 더 짧은 검색 길이를 갖도록 할 수 있음을 보인다. R_h 가 8인 경우 평균 검색 길이는, 일반적인 균일 해쉬 함수를 사용한 경우 약 29홉이고, G-Hash를 사용한 경우 약 19홉으로 약 50%의 성능향상을 보였다.

VI. 결 론

본 논문은 가상 위치 기반 라우팅을 사용하는 네트워크에서 노드의 위치 정보 저장 및 P2P 서비스를 위해 GDHT를 적용하는 방법을 제안하였다. BVR이나 LCR 좌표계에서 노드 분포 특성은 일반적인 직교 좌표계에서의 노드 분포와 다르다. 본 논문에서는 가상 위치 좌표계의 특성을 고려하여 비균일 해쉬 함수인 G-Hash 함수를 설계하였다. 분석과 논리적인 시뮬레이션을 통하여 노드의 분포와 좌표계의 특성을 고려한 G-Hash 함수를 적용했을 때 약 50%의 성능향상을 확인할 수 있었다. 이는 위치 정보 및 P2P 데이터를 저장하기 위한 DHT를 가상 위치 기반 라우팅을 사용하는 네트워크에 적용할 때 노드 공간 크기와 모양, 노드 분포 특성 뿐만 아니라 좌표계의 특성을 고려하여야 함을 보였다.

본 논문에서는 무선 특성을 고려하지 않고 논리적인 시뮬레이션을 수행하였다. 향후 실제 무선 환경을 고려하고 노드의 이동을 고려한 추가 연구가 필요하다.

참 고 문 헌

- [1] M. Castro, P. Druschel, YC. Hu, and A. Rowstron, "Topology-aware routing in structured peer-to-peer overlay networks, Technical Report", *Proc. Intl. Workshop on Future Directions in Distributed Computing*, 2003.
- [2] HJ. Yoon, EJ. Lee, H. Jeong, JS. Kim, "Proximity-Based Overlay Routing for Service Discovery in Mobile Ad Hoc Networks", *Proceedings of the 19th International Symposium on Computer and Information Sciences*, 2004.
- [3] J. Eberspacher, R. Schollmeier, S. Zols, and G. Kunzmann, "Structured P2P Networks in Mobile

and Fixed Environments”, *2nd International Working Conference on Performance modelling and Evaluation of Heterogeneous Networks (HET-NETs'04)*, 2004.

- [4] J. Eriksson, M. Faloutsos, S.V. Krishnamurthy, “DART: Dynamic Address Routing for Scalable Ad Hoc and Mesh Networks”, *Networking, IEEE/ACM Transactions on*, Volume 15, Issue 1, Feb. 2007.
- [5] YB. Ko, NH. Vaidya, “Location-Aided Routing (LAR) in mobile ad hoc networks”, *Wireless Networks*, 2000.
- [6] B. Karp, HT. Kung, “GPSR: greedy perimeter stateless routing for wireless networks”, *In Proc. ACM/IEEE MobiCom*, August 2000
- [7] A. Jadbabaie, “Geographic Routing without Location Information”, *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, Volume 5, 14-17 Dec. 2004.
- [8] R. Fonseca, S. Ratnasamy, J. Zhao, C.T. Ee, D. Culler, S. Shenker, and I. Stoica, “Beacon vector routing: Scalable point-to-point routing in wireless sensornets”, *2nd Symposium on Networked Systems Design and Implementation*, ACM Press, 2004.
- [9] Q. Cao, T. Abdelzaher, “Scalable logical coordinates framework for routing in wireless sensor networks”, *ACM Transactions on Sensor Networks(TOSN)*, 2006.
- [10] S. Ratnasamy, B.Karp, L.Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, “GHT: a geographic hash table for data-centric storage”, *International Workshop on Wireless Sensor Networks and Applications*, 2002.
- [11] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A Scalable Content-Addressable Network (CAN)”, *Applications, Technologies, Architectures, and Protocols for Computer Communications*. 2001.
- [12] L. Demoracski, “Fault-tolerent beacon vector routing for mobile ad hoc networks”, *Parallel and Distribution Processing Symposium*, 19th IEEE international, 2005.

저 자 소 개



고 석 갑(정회원)

1997년 숭실대학교 정보통신
공학과 학사 졸업.
2002년 숭실대학교 정보통신
공학 석사 졸업.
2005년 숭실대학교 정보통신
박사 수료.

2004년~2008년 다산네트웍스 책임연구원
2008년~현재 ETRI 광통신연구센터
광융합서비스연구팀
<주관심분야 : 네트워크, P2P, VOIP>



김 영 한(정회원)

1984년 서울대학교 전자공학
학사 졸업.
1986년 한국과학기술원 전기전자
공학 석사 졸업.
1990년 한국과학기술원 전기전자
공학 박사 졸업.

1987년~1994년 디지콤정보통신연구소
데이터통신연구부장.
1994년~현재 숭실대학교 정보통신전자공학부
주교수.
<주관심분야 : 컴퓨터 네트워크, 인터넷 네트워
킹, 이동 데이터 통신망>