

센서 네트워크에서 효율적인 KNN 질의처리 방법

(An Efficient KNN Query Processing Method in Sensor Networks)

손인근[†] 현동준^{**} 정연돈^{***} 이은규^{***} 김명호^{****}
 (In Keun Son) (Dongjoon Hyun) (Yon Dohn Chung) (Eunkyu Lee) (Myoung Ho Kim)

요약 전기전자 기술의 발달로 센서의 기능이 더욱 강력해지면서, 센서 네트워크의 활용 분야는 더욱 다양해지고 있다. 센서 네트워크 어플리케이션을 사용하는 주 목적은 관심 지역(예, 공장 물품 창고, 재난 지역, 야생 서식지 등)에서 발생하는 현상들을 관찰하고, 유용한 정보를 얻기 위한 것이다. k-근접 노드(KNN: k Nearest Neighbor) 탐색 질의는 특정 위치에서 지리적으로 근접한 k개의 이웃 객체를 찾기 위한 질의로서, 센서 네트워크 환경에서도 중요한 어플리케이션 중 하나이다. 그러나 이전 방법들은 센서 네트워크 환경에서 사용하기 부적합하거나 에너지 효율성 문제를 가지고 있었다. 본 논문에서는 센서 네트워크 환경의 특성을 고려하면서, k개의 근접 노드를 에너지 효율적으로 탐색할 수 있는 방법을 제안한다. 제안하는 방법은 k개의 근접 노드를 찾을 때까지 탐색 영역을 점진적으로 확장하고, 영역 내 센서들을 선별적으로 방문하여 원하는 위치 정보를 얻어내는 것이다. 이를 통해 원하는 k개의 근접 노드를 찾아내면서도 에너지 소모를 줄일 수 있다. 본 논문에서는 제안하는 방법이 기존의 방법보다 효율적이라는 것을 다양한 조건의 실험을 통해 설명한다.

키워드 : 근접 노드 탐색, 센서 네트워크, 질의처리

Abstract As rapid improvement in electronic technologies makes sensor hardware more powerful and capable, the application range of sensor networks is getting to be broader. The main purpose of sensor networks is to monitor the phenomena in interesting regions (e.g., factory warehouses, disaster areas, wild fields, etc) and return required data. The k Nearest Neighbor (KNN) query that finds k objects which are geographically close to the given point is an important application in sensor networks. However, most previous approaches are either seem to be impractical or are not energy-efficient in resource-limited sensor networks. In this paper, we propose an efficient KNN query processing method in sensor networks. In the proposed method, we dynamically increase searching boundary, if necessary, and traverse nodes inside the boundary until finding k nearest neighbors. Since only the representative sensor nodes are visited, our algorithm reduces a number of messages. We show thorough experiments that the proposed method performs better than the existing method in various network environments.

Key words : KNN, Sensor networks, Query processing

1. 서론

센서 네트워크(sensor network)는 관심 지역에 많은 수의 센서들을 배치하여 원하는 데이터를 획득하는 시스템이다. 초기의 센서들은 미리 설정된 기능대로 특정 현상만을 모니터링(monitoring)하는 단순한 형태였다[1]. 그러나 전기전자 기술이 발달하면서 다양한 기능을 가진 센서가 등장하였다. 최근의 센서들은 여러 현상들을 감지할 뿐 아니라, 획득한 데이터에서 유용한 정보를 분석하고, 필요한 정보는 자체 저장소에 관리할 수 있다. 그리고 다른 센서나 네트워크 외부에서 주어지는 질의

· 본 연구는 대학 IT연구센터 육성 지원사업의 지원을 받아 수행되었음
[†] 비 회 원 : 육군본부 전산정보 ikson@dbserver.kaist.ac.kr
^{**} 비 회 원 : 한국과학기술원 전산학과 djhyun@dbserver.kaist.ac.kr
^{***} 종신회원 : 동국대학교 컴퓨터공학과 교수 ydchung@dgu.edu
^{****} 비 회 원 : 한국전자통신연구원 연구원 ekyulee@etri.re.kr
^{*****} 종신회원 : 한국과학기술원 전산학과 교수 mhkim@dbserver.kaist.ac.kr
 논문접수 : 2005년 2월 18일
 심사완료 : 2005년 5월 24일

를 처리할 수 있다[2,3]. 이러한 센서 기능의 발달로 인해 센서 네트워크는 다양한 어플리케이션에서 활용될 수 있을 것으로 예상된다. 예를 들어, 생산공장 창고의 물품 관리, 인간이 접근하기 어려운 재난지역 탐사, 야생 동식물의 생태계 조사[4], 기상 관측, 스마트 홈(smart home) 등에서 센서 네트워크를 활용할 수 있다.

최근 센서 네트워크를 활용하기 위한 다양한 연구가 이루어지고 있는데, 관심 지역에 위치한 센서로부터 원하는 데이터를 가져오기 위한 분산 질의 처리(distributed query processing)도 중요한 연구 분야가 되고 있다. 효율적인 분산 질의 처리에 대한 연구는 이미 데이터베이스 분야에서 활발하게 이루어지고 있다[5]. 그러나 센서 네트워크에서는 분산 질의 처리를 담당하는 노드들이 제한된 에너지와 통신 능력을 가진 센서들이다. 그리고 센서 네트워크는 수 천, 수 만에 이르기까지 많은 센서들로 구성이 된다. 따라서 센서 네트워크에서 질의 처리 방법에 대한 연구는 이러한 특성을 고려한 새로운 방법이 필요하다. 특히, 질의 처리를 통해 원하는 결과를 얻을 때까지 소모되는 에너지를 줄이는 것이 우선적으로 고려되어야 한다. 에너지 소모는 무선 통신 비용, 즉 센서 노드 간의 메시지 전달 회수와 밀접한 관련성을 가지고 있다. 실제 센서 네트워크 환경 기반의 질의 처리에 대한 많은 연구들이 에너지 절약을 목표로 하고 있다.

k-근접 노드(KNN: k Nearest Neighbor) 탐색은 주어진 위치로부터 지형적으로 근접한 k개의 객체를 찾는 것이다. 근접 노드 탐색은 지리 정보 시스템(GIS: Geographic Information System)에서 점 질의(point query) 및 영역 질의(range query)와 더불어 자주 사용되는 질의 형태이다[6].

본 논문에서는 센서 네트워크 환경의 특성을 고려하면서, k개의 근접 노드를 에너지 효율적으로 탐색할 수 있는 방법을 제안한다. 본 논문에서는 원하는 k개의 근접 노드를 찾을 때까지 탐색 영역(search space)을 질의 위치에서 점진적으로 확장하면서 탐색하고, 확장된 영역의 정보를 모두 알 수 있는 센서 노드들을 선별적으로 방문하여 에너지 소모를 줄인다. 이를 위해 탐색 영역을 얼마나 확장할 것인가, 확장된 영역에 위치한 노드 위치 정보를 어떻게 수집할 것인가 그리고 근접 노드를 어떻게 계산할 것인지에 대한 알고리즘들을 제안한다. 그리고 실험을 통하여 제안 방법이 다양한 센서 네트워크 환경에서도 에너지 효율적임을 보인다.

본 논문에서는 센서들은 움직이지 않으며, 자신의 통신 범위 내에 위치한 이웃 노드들의 위치를 아는 것으로 가정한다. 또한 두 개 이상의 네트워크로 분리되지 않는 것으로 가정하였다. 균등 분포되어 있는 무선 네트

워크에서 통신 범위 내에 위치하고 있는 노드의 평균 개수(네트워크 밀집도)가 6이상인 경우, 네트워크는 분리되지 않고 하나로 연결되어 있는 것으로 알려져 있다 [8]. 센서의 통신 반경은 수 십 미터 정도인 것이 대부분이며, 수 백 미터에서 10킬로미터 이상의 통신 반경을 가지는 센서들도 있다.

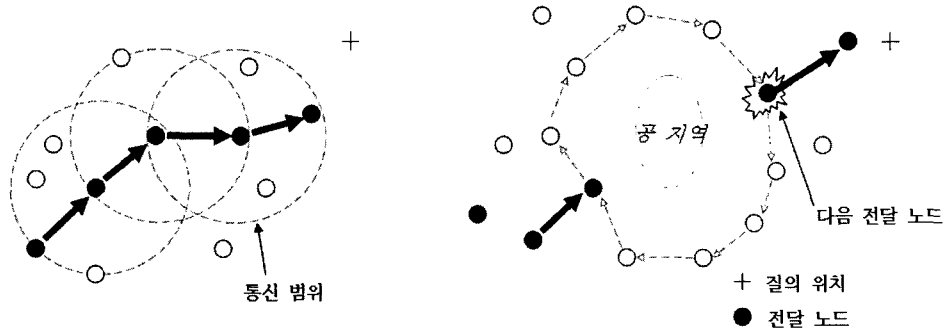
본 논문은 다음과 같이 구성되어 있다. 2장에서는 센서 네트워크에서 k-근접 노드 탐색방법에 대한 이전 연구들을 소개하고, 3장에서는 본 논문에서 제안한 근접 노드 탐색 방법에 대해 설명한다. 그리고 4장에서는 제안하는 방법의 에너지 효율성을 실험으로 보이고, 기존 연구와 비교 분석한다. 끝으로, 5장에서 결론을 맺도록 한다.

2. 관련 연구

2.1 센서 네트워크에서 분산 질의 처리

센서의 데이터 처리 능력을 활용하는 어플리케이션에서, 관심 지역에 위치한 센서로부터 원하는 데이터를 가져오기 위한 분산 질의 처리는 중요한 연구 분야이다. 분산 질의 처리에 대해서는 이미 데이터베이스 분야에서 활발하게 연구되었던 부분이기도 하다. 그러나 센서 네트워크 환경은 기존 네트워크 환경과는 다른 몇 가지 특성을 가지고 있어, 이에 적합한 방법이 연구되고 있다. 질의 처리에서 고려해야 할 센서 네트워크 특징은 다음과 같다[1-3,9].

- 센서들은 데이터 생성 및 저장, 무선 통신 그리고 질의 처리 능력들을 가지고 있다: 최근의 센서들은 자체의 컴퓨팅 능력을 가지고 있어, 이를 활용한 분산 질의 처리가 가능하게 되었다.
- 센서 네트워크는 에너지 제한적이다: 센서 네트워크는 배터리(battery)와 같은 제한된 에너지원을 가진 센서들로 구성되어 있다. 따라서 센서의 수명(lifetime)은 에너지 소모 정도가 결정한다. 예를 들어, 미국 버클리 대학에서 개발한 Mica Mote라는 연구용 센서의 수명은 사용 정도에 따라 1주일에서 약 1년까지 다양한 차이를 보이고 있다.
- 센서의 에너지 소모는 무선 통신 비용(wireless communication cost)이 가장 큰 비중을 차지한다: 센서에서 발생하는 에너지 소모는 데이터 생성 및 저장, 질의 처리 등의 내부 처리 비용과 이웃 센서에게 메시지를 주고 받은 무선 통신 비용으로 구분할 수 있다. 이 중 무선 통신 비용이 주된 에너지 소모 요인이 된다.
- 센서의 무선 통신 범위는 일반적인 통신 장비에 비해 작아서 멀티-홉(multi-hop) 통신 방식을 사용한다: 그러므로 목적지까지 원하는 메시지를 전송하기 위해서



(a) 그리디(greedy) 포워딩 (b) 퍼리미터(perimeter) 포워딩
그림 1 GPSR (Greedy Perimeter Stateless Routing)

는 센서 간의 통신을 여러 번 거쳐야 한다.

- 센서 네트워크를 구성하는 센서의 개수는 많다: 센서는 에너지가 제한적이고 통신 범위가 작아서, 특정 지역에 가용한 센서가 없거나 네트워크 분리가 발생할 가능성이 높다. 이러한 상황을 고려하여 많은 수의 센서를 배치한다.

정리하면, 센서 네트워크에서는 질의 처리를 통해 원하는 결과를 얻을 때까지 소모되는 에너지를 줄이는 것이 우선적으로 고려되어야 한다. 이때 에너지 소모는 무선 통신 비용, 즉 센서 노드 간의 메시지 전달 회수에 밀접한 관련성을 가지고 있다. 그리고 원하는 지점까지 메시지를 보내기 위해서는 멀티-홉 통신을 위한 프로토콜이 필요하다. 실제 센서 네트워크 환경 기반의 질의 처리에 대한 많은 연구들이 에너지 절약을 목표로 하고 있다.

2.2 멀티-홉 통신 방법

센서 네트워크에서 보내고자 하는 데이터의 목적지가 센서의 통신 범위를 벗어날 경우에는 멀티-홉 통신 방법을 사용해야 한다. 그림 1에서 보여주는 GPSR(Greedy Perimeter Stateless Routing)은 센서 네트워크에서 잘 알려진 라우팅 방법이다[10]. 센서 네트워크에서 노드는 센서를 의미한다. GPSR은 전체 네트워크 구성 정보 없이 각 노드가 알고 있는 이웃 정보만을 활용하여 원하는 목적지까지 메시지를 전달하는 방법이다. GPSR은 그리디(greedy) 포워딩 방법과 퍼리미터(perimeter) 포워딩을 사용한다. 그리디 포워딩은 목적지까지 메시지를 전달하기 위하여 기본적으로 사용하는 방법으로, 자신의 이웃 노드 중에서 목적지에 가장 가까운 노드에게 메시지를 전달하는 것이다. 퍼리미터 포워딩은 그리디 포워딩으로 메시지를 전달하지 못할 경우에 사용하는 방법으로, 여기서 퍼리미터(perimeter)란 노드들을 경유하면서 구성된 다각형 폐곡선을 의미한다. 퍼리미터 포

워딩에서는 목적지 방향으로 이웃 노드가 없는 공 지역(void area)를 만날 경우 해당 지역을 감싸는 퍼리미터를 구성한 후 퍼리미터에 있으면서 목적지에 가장 가까운 노드로 메시지를 전달하는 것이다. GPSR은 목적지까지 그리디 포워딩으로 메시지를 전달하고, 전달해야 할 이웃이 없는 공 지역을 만날 경우에는 퍼리미터 포워딩으로 메시지를 전달한다.

2.3 기존의 근접 노드 탐색 방법

주어진 위치에 근접한 k개의 객체를 찾는 것을 근접 노드 탐색이라 하며, 이때 주어진 위치를 “질의 위치”라 한다. 객체들의 위치 정보를 모두 알고 있는 경우에 대한 연구는 많이 있었다. 그러나 센서 네트워크에서 기존 방식을 사용하기에는 어려움이 있다. 센서는 제한된 에너지와 통신 능력을 가지고 있고 장애에 취약하여, 질의 시점마다 사용 가능한 센서 상태가 달라질 수 있기 때문이다. 따라서 센서 네트워크에서는 네트워크 전체 정보를 따로 유지하지 않고 탐색할 수 있는 방법이 필요하게 되었다.

지금까지 제시된 센서 네트워크에서 근접 노드 탐색 방법은 크게 두 가지 방법으로 구분된다. 첫 번째 방법은 네트워크를 구성하는 전체 센서들 간의 경로를 특정 트리 구조 형태로 구성하는 것이다. 이 방법은 트리 구조가 센서 네트워크 전체에 사전에 형성되어 있어, 전통적인 인덱스 기반의 탐색 방법들을 활용할 수 있는 장점을 가지고 있다. Peer-tree[12]는 센서 네트워크 상에 R-tree를 구성한 것이다. 각 센서 노드들을 객체로 생각하고, 센서 노드들을 정사각형 모양의 클러스터로 묶은 후, 그 클러스터들을 계층적인 구조를 연결하여 분산된 R-tree를 구성한다. DIFS(Distributed Index for Features in Sensor networks)[13]는 센서 네트워크 상에 Quad-tree를 구성한 것이다. 센서 네트워크 전체 영역을 재귀적으로 4등분하면서 Quad-tree 구조를 생성

하여 사용한다[6,14]. 그러나 이러한 방법들은 여러 문제점들을 가지고 있다. 먼저 트리의 상단에 해당하는 센서들은 사용 빈도가 높아 에너지 소모가 커지므로 다른 센서보다 수명이 짧아지게 된다. 전체 센서 네트워크의 수명은 네트워크가 분리되는 것과 밀접한 관련이 있다. 따라서 모든 노드들의 수명이 균등 분포를 이루지 않으면 전체 센서 네트워크의 수명이 짧아지게 된다. 그리고 수명이 다하거나 장애가 발생한 노드가 있을 경우 트리를 재구성하는 것도 심각한 오버헤드(overhead)다.

두 번째 방법은 질의 위치 지점까지 질의를 전달한 후, 질의 위치 주변의 부분적인 영역에 대해서만 특정 트리 구조를 구성하는 것이다. 이때 트리 구조는 사용자 질의가 질의 위치 지점에 도착한 시점부터 생성되며, 결과를 찾을 때까지 임시적으로 유지된다. 따라서 필요한 시점에 필요한 범위 내에서만 트리 구조를 생성하므로, 전체 트리 경로를 유지할 필요가 없다는 장점을 가지고 있다. 이때 어느 범위까지 트리를 구성해야 k개의 근접 노드를 찾을 수 있는지 트리 구성 범위를 추정(estimation)해야 한다. 구성해야 하는 최소 영역은 k개의 근접 노드 존재를 보장할 수 있는 범위까지가 된다. 기존 연구로는 KPT(k Perimeter Tree)[15] 방법이 있다. KPT는 그림 2와 같이 질의 위치에 근접한 노드를 연결하여 퍼리미터(perimeter)를 구성한 다음, 탐색 영역을 분할하여 퍼리미터에 있는 각 노드를 루트 노드로 하는 최소 확장 트리(minimum spanning tree)를 구성하는 것이다. 각 트리를 구성하는 범위의 반경은 평균 통신 거리에 k값을 곱한 값을 이용하기 때문에, 탐색 영역은 k값에 선형적(linear)으로 비례하여 증가한다. 센서 노드들이 질의 위치 주변 지역에 충분히 많이 분포되어 있더라도 트리가 구성되는 범위는 동일하다. 즉, 불필요한 탐색 영역이 커지는 단점이 있다. 그리고 트리 구성 및 질의 처리를 위해 범위 내에 있는 모든 노드에서 각 노드 별로 최소 2번의 메시지 전달이 필요하다.

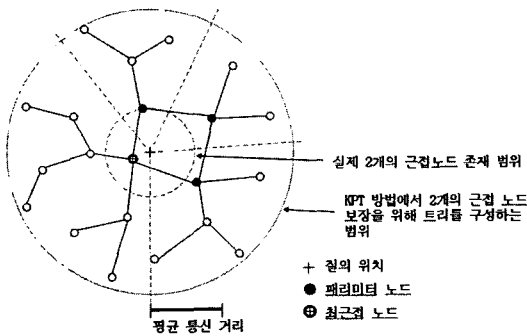


그림 2 KPT 방법(k = 2 일 때)

3. 제안하는 방법

3.1 제안 동기

k개의 근접 노드를 찾기 위해서는 질의 위치 주변에 위치한 노드들의 정보를 계산해야 한다. 이를 위해서는 전체 영역에서 근접 노드가 있을 영역을 줄여가면서 탐색하거나, 질의 위치로부터 근접 노드가 있을 만한 영역까지 확장하여 탐색하는 방법을 사용할 수 있다. 그러나 네트워크 전체 영역에 트리 구조를 구성하여 탐색하는 것은, 2절에서 설명했듯이 에너지와 통신거리가 제약된 센서 네트워크 특성에 적합하지 않다. 따라서 질의 위치로부터 어느 범위까지 탐색할 것인지를 정하는 방법이 중요하다.

KPT 방법은 근접 노드가 있을 수 있는 최대 범위를 고려하므로, 불필요한 탐색 영역이 커질 수 있는 문제점이 있다. 그리고 트리 구조를 구성하여 원하는 노드 정보를 수집하는 과정에서 범위 내의 모든 노드들을 방문해야 한다. 본 논문에서는 이러한 점을 고려하여 탐색 영역을 어떻게 설정하고, 영역 내 위치한 모든 노드들의 위치 정보를 어떻게 수집할 것인지에 대한 방법을 제안한다.

3.2 근접 노드 탐색 방법

본 장에서는 센서 네트워크 환경에서 활용 가능하며, 기존의 방법보다 에너지 효율적인 근접 노드 탐색 방법을 제안한다. 제안하는 방법은 크게 두 가지 특성을 가진다.

- 탐색 영역을 점진적으로 확장한다: 탐색 영역은 k개의 근접 노드를 찾을 때까지 질의 위치를 중심으로 점차 확장한다. 처음 설정된 탐색 영역에서 원하는 k개의 근접 노드를 찾는다면 결과를 반환하고, 부족할 경우 이전 탐색 범위에서부터 추가적인 탐색 범위를 정하고 범위 내 위치한 노드를 찾아내는 과정을 반복한다. 이를 통해 불필요한 탐색 영역을 줄일 수 있다.
- 필요한 노드들을 방문하여 탐색 영역에 위치한 노드를 찾아낸다: 센서들은 자신의 통신 범위 내에 있는 이웃 위치 정보를 알고 있기 때문에, 노드들을 선택적으로 방문해도 모든 위치 정보를 알 수 있다. 탐색 영역에 위치한 노드들은 방문한 노드가 가지고 있는 이웃 노드 정보를 종합하여 구한다. 이를 통해 메시지 수를 줄일 수 있다.

제안하는 방법에서는 탐색 영역을 '트랙(track)'으로 정의하였으며, 트랙에 위치한 일부 노드들만으로 구성된 순환경로, 즉 사이클을 '트랙 퍼리미터(track perimeter)'라 하였다. 이때 종합된 이웃 노드 정보들은 'FSR(Fully-Scanned circular Range)'로 정의된 영역에 속한 것만을 근접 노드 계산에 사용한다. 정의는 다음과 같다.

정의 1. 질의 위치 Q를 중심으로 n번째 생성되는 트랙(track) T_n 은 다음과 같이 정의된다.

$$T_n = \begin{cases} R(C_1), & \text{if } n=1 \\ R(C_n) - R(C_{n-1}), & \text{if } n>1 \end{cases}$$

C_n = 질의 위치 Q를 중심으로 한 동심원(concentric circle), $n \geq 1$

C_i ($i \geq 1$)의 반지름을 r_i 라 할 때 $r_i > r_{i-1}$ 을 만족한다, 단 $r_0 = 0$

이 때 트랙 T_n 의 폭은 $r_i - r_{i-1}$ 로 정의한다.

$R(C_n)$ = 동심원 C_n 내의 범위 □

트랙은 근접 노드를 찾기 위해 확장되는 영역으로 k개의 근접 노드를 찾을 때까지 반복해서 생성된다. 첫 번째 트랙(T_1)은 원형이며, 이후에는 그림 3의 3번째 트랙(T_3)과 같이 도넛츠 모양을 가진다. 이때 트랙의 폭을 어느 정도로 할 것인가가 중요한데, 자세한 내용은 3.3에서 설명한다.

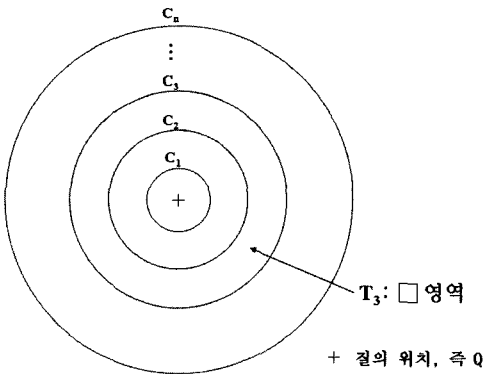


그림 3 트랙(track)의 예(T_3 의 경우)

정의 2. 트랙 퍼리미터(track perimeter)는 트랙 T_n 내에 위치하고 서로 통신 가능한 노드들을 경유하여 트랙을 한 바퀴 도는 순환 경로(circular path)이다. □

트랙 퍼리미터는 그림 4와 같이 트랙에 위치한 노드들을 사용하여 트랙을 한 바퀴 도는 순환 경로다. 이때, 트랙 퍼리미터를 구성하는 노드들을 트랙 퍼리미터 노드라 한다. (트랙에 위치한 노드들만으로 퍼리미터를 구성하지 못하는 경우에 대해서는 3.6에서 설명한다.) 트랙 퍼리미터를 구성하는 이유는 트랙에 위치한 노드들의 정보를 종합하기 위해서다. 예를 들어, 어떤 트랙에서 m개의 노드 v_i ($i = 1, 2, \dots, m$)들로 구성된 트랙 퍼리미터 $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_m \rightarrow v_1$ 가 있다고 하자. 이때 경유하면서 수집한 노드들의 집합은 다음과 같다.

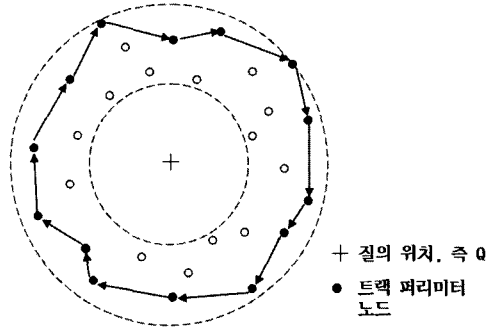


그림 4 트랙 퍼리미터(track perimeter)의 예

$$\{u \mid u \in V \wedge (u = v_i \vee isNeighbor(u, v_i)) \mid 1 \leq i \leq m\}$$

V = 전체 노드 집합

v_i = 트랙내 위치한 노드

m = 경유한 노드 개수

$$isNeighbor(u, v_i) = \begin{cases} true, & \text{if } u \text{ is one of } v_i\text{'s neighbors} \\ false, & \text{otherwise} \end{cases}$$

트랙 퍼리미터를 구성하기 위해서는 자신의 이웃 노드 중에서 어느 노드를 경유할 것인지를 정해야 한다. 이에 관해서는 3.4에서 설명한다.

정의 3. FSR(Fully-Scanned circular Range)은 트랙 퍼리미터를 통해 확인된 영역에 내접하는 원 영역이다. 이때 원은 질의 위치 Q를 중심으로 한다. □

FSR은 수집된 노드의 집합 중에서 근접 노드 계산에 적합한 노드들을 선택하기 위해 사용된다. 그림 5에서 음영 지역은 트랙 퍼리미터 노드 4개의 통신 영역을 나타낸다. 이때, 근접 노드 계산에 이용될 수 있는 영역 FSR은 질의 위치 Q에서 점선으로 된 동심원까지의 범위이다. 그림 5에서 노드 v_a 와 v_b 는 트랙 퍼리미터에서 수집한 노드에 해당한다. 만약 수집된 노드들 전부를 질의 위치 Q와의 거리를 기준으로 근접 정도를 정렬할 경우, FSR 내에 위치한 v_a 의 근접 순서는 보장할 수 있다. 그러나, v_b 는 근접 순서를 계산하더라도 아직 검색되지 않은 영역에 더 근접한 노드(예. v_c)가 있을 가능

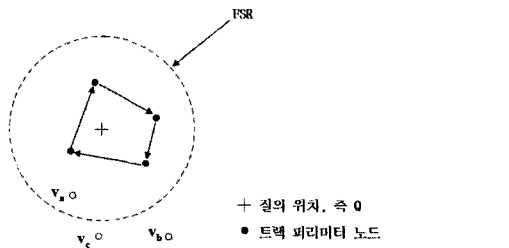


그림 5 FSR(Fully-Scanned circular Range)의 예

```

/* 입력: 질의 위치 좌표, 근접 노드 개수 k
출력: k개의 근접 노드 ID와 위치 집합 */

/* 질의한 노드에서 질의 위치의 트랙 T1까지 질의 메시지를 전달하는 과정 */
1. GPSR을 이용하여 T1에 진입할 때까지 질의 메시지를 전달한다.
2. T1에 처음 진입한 노드를 트랙 퍼리미터 시작 노드로 정한다.
3. n = 1; /* n: 탐색할 트랙 번호 */

/* 트랙 퍼리미터를 구성하고 수집된 위치 정보를 이용하여 근접 노드 계산*/
4. while (true) {
    5. if (n != 1){ /* 탐색할 트랙이 T1이 아닌 경우 */
        6. 이전 트랙에서 계산한 FSR로부터 검색할 트랙의 폭을 정한다.
        7. 이전 트랙 퍼리미터 시작 노드에서부터 질의한 노드 방향으로 메시지를 전달한다. 메시지 전달은 계산할 트랙에 처음 진입할 때까지 진행한다.
        8. 트랙에 처음 진입한 노드를 트랙 퍼리미터 시작 노드로 정한다.
    }
    9. 트랙 퍼리미터를 구성하면서 경유하는 노드들이 가지고 있는 이웃 정보를 모은다.
    10. FSR을 계산한다.
    11. FSR에 위치한 센서 위치 정보를 질의 위치와의 거리 값으로 정렬하여 근접 노드를 계산한다.
    12. if(k개 이상의 근접 노드를 찾았을 경우)
        13. break;
    14 else /* 근접 노드를 더 찾아야 할 경우 */
        15. n = n + 1;
}

/* 근접 노드 탐색 결과를 질의한 노드에게 돌려주는 과정 */
16. 질의한 노드에게 GPSR을 이용하여 결과를 전달한다.

```

그림 6 근접 노드 탐색 절차

성이 있기 때문에 근접 노드 계산에 포함하는 것은 부적합하다. 트랙 퍼리미터를 구성한 이후 FSR을 계산하는 방법에 대해서는 3.5에서 설명한다. FSR은 그림 3에서 트랙을 정의하기 위해 사용된 동심원 C_i 와 다른 것이다.

제안하는 근접 노드 탐색 절차는 다음과 같다. 질의 조건으로는 질의 위치 좌표 값과 근접 노드 개수 k 가 주어진다. 질의 메시지는 GPSR을 이용하여 질의한 노드에서부터 질의 위치를 중심으로 한 첫 번째 트랙, T_1 에 진입할 때까지 전달된다. 이때 T_1 에 처음 진입한 노드를 트랙 퍼리미터 구성을 위한 시작 노드로 정한다. 제안하는 트랙의 폭은 3.3에서 자세히 설명한다. 트랙 퍼리미터 시작 노드를 정해지면 퍼리미터를 구성하면서 노드 정보를 모은다. 퍼리미터 구성이 완료되면 트랙 퍼리미터 시작 노드에서 근접 노드를 계산하는데 FSR 범위 내에 속하는 노드 정보들을 이용한다. 만약 계산된 근접 노드 개수가 k 개 이상인 경우에는 질의한 노드에게 결과를 돌려주게 되고, 아니면 다음 트랙의 폭을 정하고 이전 과정을 반복하게 된다. T_2 부터는 이전 트랙

퍼리미터 시작 노드에서부터 질의한 노드 방향으로 메시지를 전달하면서, 트랙에 처음 진입한 노드를 트랙 퍼리미터 시작 노드로 정한다.

트랙의 폭을 어느 정도로 정하여 탐색 영역을 확장할 것인가, 트랙에 위치한 노드 정보를 모으기 위해 어떠한 방법으로 퍼리미터를 구성할 것인가 그리고 근접 노드를 계산하기 위해 FSR을 어떻게 계산할 것인지에 대해 다음 절부터 자세히 알아본다.

3.3 트랙(track)의 폭 설정

제안하는 방법은 트랙에 존재하는 모든 노드들의 정보를 수집하기 위해서 트랙 퍼리미터를 구성한다. 이때 트랙의 폭은 트랙 퍼리미터가 어떤 경로로 구성되더라도, 트랙에 있는 모든 노드들을 찾을 수 있는 범위로 설정해야 한다. 트랙의 폭은 센서의 무선 통신 반경과 밀접한 관련이 있다. 트랙 퍼리미터 노드들은 자신의 통신 범위에 있는 이웃 정보만을 알기 때문이다.

정리 1. 트랙 T_n 에서 트랙 퍼리미터를 구성했을 때, T_n 의 폭이 δ_n 보다 작거나 같을 때는 반드시 T_n 에 존재하는 모든 노드들의 정보를 찾을 수 있다.

$$\delta_n = \begin{cases} \frac{\sqrt{3}}{2} \times R, & \text{if } n=1 \\ \frac{\sqrt{2}}{2} \times R, & \text{if } n>1 \end{cases}$$

R: 센서의 무선 통신 반경

증명:

(1) n = 1인 경우

트랙 퍼리미터 노드 $v_i (i = 1, 2, \dots, m)$ 들이 $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_m \rightarrow v_1$ 형태로 트랙 퍼리미터를 구성하여 노드 정보를 수집하였다고 하자. 이때 T_1 의 트랙 퍼리미터 노드들은 질의 위치 Q를 통신 범위에 포함하기 위해 질의 위치 Q와의 거리가 R보다 작은 위치에 있다. 두 지점 A와 B를 양 끝으로 하는 선분은 \overline{AB} 로, 선분의 길이는 $|\overline{AB}|$ 로 표시하겠다.

그림 7의 (a)는 통신 반경이 R인 트랙 퍼리미터 노드 v_i 와 v_j 의 위치와 트랙 T_1 의 폭과의 관계를 보여주고 있다. 그림에서 v_i 와 v_j 사이에서 모든 노드를 검색할 수 있는 트랙의 폭은 $|\overline{QP_1}| (= |\overline{QP_2}| + |\overline{P_1P_2}|)$ 이 된다. 이때

$|\overline{P_1P_2}|$ 는 $|\overline{v_i v_j}|$ 가 R일 때 $\frac{\sqrt{3}}{2} \times R$ 로 최소값을 가진다.

그리고 $|\overline{QP_2}|$ 는 그림 7의 (b)와 같이 $\overline{v_i v_j}$ 사이에 질의 위치 Q가 있는 경우 0으로 최소값을 가진다. 따라서

$|\overline{QP_1}|$ 은 $\frac{\sqrt{3}}{2} \times R$ 보다 항상 크거나 같다. 따라서 트랙

T_1 의 폭을 $\frac{\sqrt{3}}{2} \times R$ 로 정하고 트랙 퍼리미터가 폭의 범위 내에서 구성된다면, T_1 에 위치한 모든 노드들을 항상 찾을 수 있다.

(2) n > 1인 경우

T_2 부터 트랙은 도너츠 모양으로 내선 경계(inside line of track)와 외선 경계(outside line of track)를 가진다. 이때, 트랙의 폭은 트랙 퍼리미터 노드가 내선 경계에 위치할 때 외선 경계까지 알 수 있어야 하고, 역으로도 가능해야 한다. 따라서 이 두 조건을 만족할 수 있는 폭을 정하면 된다. 만약 트랙 퍼리미터 노드 $v_i (i = 1, 2, \dots, m)$ 들이 $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_m \rightarrow v_1$ 형태로 트랙 퍼리미터를 구성하여 노드 정보를 수집하였다고 할 때, 연결된 두 트랙 퍼리미터 노드 v_i 와 v_j 사이의 거리, $|\overline{v_i v_j}|$ 와 트랙의 폭과의 관계는 다음과 같다.

(가) v_i 와 v_j 가 트랙의 내선 경계 위에 위치할 경우

그림 8과 같이 v_i 와 v_j 가 트랙의 내선에 위치할 경우, 외선 경계의 위치는 내선 경계와의 폭이 $|\overline{P_1P_3}| (= |\overline{P_1P_2}| + |\overline{P_2P_3}|)$ 이 되도록 정할 수 있다. 이때 $|\overline{P_1P_2}|$ 는 $|\overline{v_i v_j}|$ 가 R일 때

$\frac{\sqrt{3}}{2} \times R$ 로 최소값을 가진다. 그리고 $|\overline{P_2P_3}|$

$(= |\overline{QP_3}| - |\overline{QP_2}|)$ 는 v_i 와 v_j 가 T_2 의 내선에 위치하고

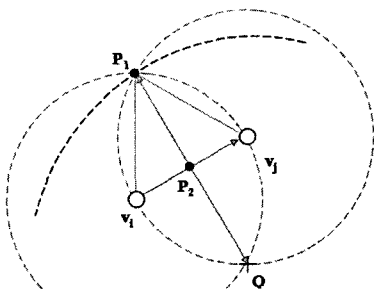
$(|\overline{Qv_1}|$ 와 $|\overline{Qv_2}|$ 가 $\frac{\sqrt{3}}{2} \times R$ 이요), $|\overline{v_i v_j}|$ 가 R일 때

$\frac{\sqrt{3}-\sqrt{2}}{2} \times R$ 이 되는데, T_3 부터 $|\overline{P_2P_3}|$ 는 v_i 와 v_j 를 지

나는 원이 커지므로 $\frac{\sqrt{3}-\sqrt{2}}{2} \times R$ 보다 작은 값을 가진

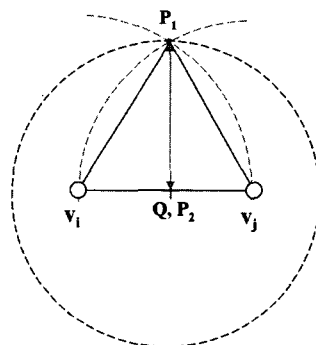
다. 그러므로 $|\overline{P_1P_3}|$ 는 T_2 이상이면 항상 $\frac{\sqrt{2}}{2} \times R$ 보다

크거나 같다. 따라서 트랙의 내선 경계에 위치한 트랙



P₁: v_i 와 v_j 를 중심으로 각각 반경이 R인 두 원의 교점중 Q에 먼 지점
P₂: $\overline{QP_1}$ 과 $\overline{v_i v_j}$ 가 만나는 교점

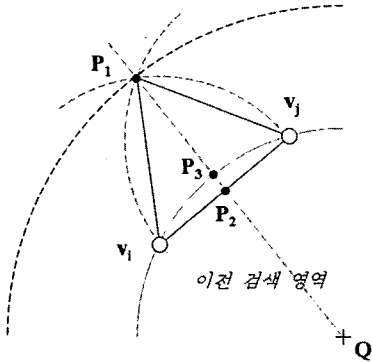
(a) 일반적인 경우



(b) $|\overline{QP_2}| = 0$ 인 경우

그림 7 T_1 의 폭 $|\overline{QP_1}| (= |\overline{QP_2}| + |\overline{P_1P_2}|)$ 계산

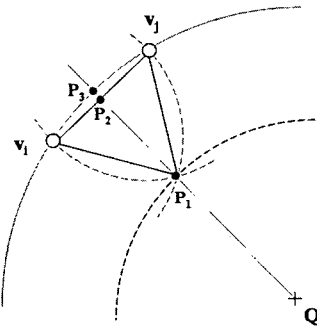
퍼리미터 노드가 외선 경계까지 알 수 있는 폭을 $\frac{\sqrt{2}}{2} \times R$ 로 정하면 이는 항상 만족한다.



P₁: v₁와 v_j를 중심으로 각각 반경이 R인 두 원의 교점중 Q에 먼 지점
 P₂: Q와 P₁을 지나는 직선과 v₁v_j가 만나는 교점
 P₃: Q를 중심으로 v₁와 v_j를 지나는 원과 Q와 P₁을 지나는 직선이 만나는 점

그림 8 트랙의 내선 경계에 위치한 경우

(나) v₁와 v_j가 트랙의 외선 경계 위에 위치할 경우
 (가)와 유사한 방법으로 트랙의 외선 경계에 위치한 트랙 퍼리미터 노드가 내선 경계까지 알 수 있는 폭을 계산하면 $\frac{\sqrt{3}}{2} \times R$ 가 된다(그림 9 참조).



P₁: v₁와 v_j를 중심으로 각각 반경이 R인 두 원의 교점중 Q에 가까운 지점
 P₂: Q와 P₁을 지나는 직선과 v₁v_j가 만나는 교점
 P₃: Q를 중심으로 v₁와 v_j를 지나는 원과 Q와 P₁을 지나는 직선이 만나는 점

그림 9 트랙의 외선 경계에 위치한 경우

(가)와 (나)에서 트랙의 폭을 $\frac{\sqrt{2}}{2} \times R$ 로 정하면, 트랙 퍼리미터 노드가 내선 경계에 위치할 때 외선 경계까지 알 수 있고 역으로도 가능하다. 그러므로 n>1에서 트랙의 폭을 $\frac{\sqrt{2}}{2} \times R$ 로 정하고 트랙 퍼리미터가 폭 범위 내

에서 구성된다면, 트랙 내 위치한 모든 노드들을 항상 찾을 수 있다.

위의 (1)과 (2)에 의해 정리 1을 만족한다. □

3.4 트랙 퍼리미터 구성 방법

본 절에서는 트랙 퍼리미터를 구성하는 방법으로 OSTP(Outer-Selective Track Perimeter) 포워딩 방법을 제시한다. 제안하는 방법은 트랙의 외선 바깥에 있는 노드 정보들까지 최대한 많은 정보를 모으는 방법이다. 트랙의 폭이 정리 1과 같을 때 트랙 퍼리미터를 구성하고 FSR을 계산하면, 트랙 내 모든 노드 정보는 알 수 있으므로 트랙의 외선 경계는 반드시 포함한다. 이때 트랙 퍼리미터를 질의 위치 Q에서 멀리 위치한 노드들을 이용하여 구성한다면, FSR은 트랙의 외선원 영역보다 커지게 된다. FSR이 커지게 되면 근접 노드 계산에 적합한 노드의 개수가 많아지게 된다. 그리고 k개의 근접 노드를 찾지 못했을 경우, 다음 트랙의 내선원을 현재 트랙의 FSR로 정하여 좀 더 바깥 영역을 탐색할 수 있다. 이를 위해 OSTP 포워딩 방법은 다음 트랙 퍼리미터 노드로 질의 위치와 자신으로부터 먼 이웃 노드를 선택하는 것이다. 그림 10은 OSTP 포워딩의 예를 보여 주고 있다.

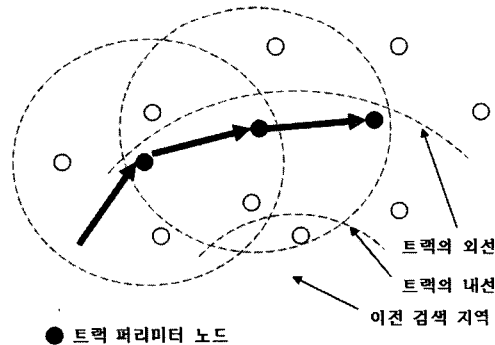


그림 10 OSTP(Outer-Selective Track Perimeter) 포워딩

3.5 근접 노드 계산

근접 노드 계산은 OSTP 포워딩을 통해서 수집한 노드 정보를 이용한다. 계산은 FSR(Fully-Scanned circular Range)내에 위치한 노드들을 대상으로 한다. 만약 계산된 근접 노드 개수가 k개 이상인 경우에는 질의 한 노드에게 결과를 다시 돌려주게 되고, 아니면 현재 트랙까지의 FSR을 다음 트랙의 내선 경계로 하고 다음 트랙을 정하여 탐색한다.

정리 2. T_n까지 퍼리미터를 구성했을 때 계산되는 FSR의 반지름은 다음과 같다. 이때, T_n의 트랙 퍼리미터 노드들의 집합 V_n 중에서 질의 위치 Q와 가장 가까

은 거리를 $MINDIST(V_n, Q)$ 라고 표기한다.

T_n 에서 FSR 반지름 =

$$\begin{cases} MINDIST(V_n, Q) + (\sqrt{3}-1) \times R, & \text{if } MINDIST(V_n, Q) \geq R \\ T_n \text{의 외선 경계 반지름,} & \text{otherwise} \end{cases}$$

R : 센서의 무선 통신 반경

V_n : T_n 에서 트랙 퍼리미터 노드의 집합

증명. T_n 에서 트랙 퍼리미터 노드 $v_i (i = 1, 2, \dots, m)$ 들이 $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_m \rightarrow v_1$ 형태로 트랙 퍼리미터를 구성하여 노드 정보를 수집하였다고 하자. 두 지점 A와 B를 양 끝으로 하는 선분은 \overline{AB} 로, 선분의 길이는 $|\overline{AB}|$ 로 표시하겠다.

(1) $MINDIST(S_n, Q) < R$ 인 경우

(1)의 경우는 트랙 퍼리미터 노드 중 질의 위치 Q와의 거리가 가장 작은 값이 R보다 적은 경우로 T_1 과 T_2

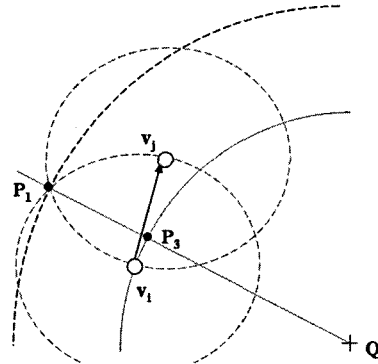
에서 생긴다. T_1 은 정리 1에 의해 항상 $\frac{\sqrt{3}}{2} \times R$ 내에 트랙 퍼리미터가 구성되고 T_1 에 위치한 모든 노드를 알 수 있으므로, 질의 위치 Q에서 T_1 의 외선 경계까지 FSR을 정할 수 있다. T_2 는 질의 위치 Q에 가장 근접한 노드가 R보다 적은 경우 적용되는데 T_2 에서도 외선 경계까지는 항상 알 수 있으므로, 질의 위치 Q에서 T_2 의 외선 경계까지 FSR을 정할 수 있다. 그러므로 $MINDIST(S_n, Q) < R$ 인 경우 해당 트랙의 외선 경계 반지름을 FSR로 정할 수 있다.

(2) $MINDIST(S_n, Q) \geq R$ 인 경우

(2)의 경우는 트랙 퍼리미터 노드 중 질의 위치 Q와의 거리가 가장 작은 값이 R보다 크거나 같은 경우로 T_2 이상에서 생긴다. 트랙 퍼리미터 v_i 가 질의 위치가 가장 근접한 노드이고, 연결된 트랙 퍼리미터 노드 간의

선분 중 $\overline{v_i v_j}$ 가 질의 위치 Q에 가장 근접할 때, 그림 11의 (a)와 같이 $|\overline{QP_1}| (= |\overline{QP_3}| + |\overline{P_1 P_3}|)$ 을 FSR의 반지름으로 정할 수 있다. 이때 $|\overline{QP_3}|$ 는 $|\overline{Qv_i}|$, 즉 $MINDIST(V_n, Q)$ 가 된다. 그리고 $|\overline{P_1 P_3}|$ 는 그림 11의 (b)와 같이 T_2 에서 $|\overline{Qv_i}|$ 와 $|\overline{Qv_j}|$ 그리고 $|\overline{v_i v_j}|$ 가 R일 때, $(\sqrt{3}-1) \times R$ 로 최소값을 가진다. T_3 부터 $|\overline{P_2 P_3}|$ 는 v_i 와 v_j 를 지나는 원이 커지므로 T_2 경우보다 작은 값을 가진다. 따라서, $|\overline{QP_1}|$ 은 $MINDIST(V_n, Q) + (\sqrt{3}-1) \times R$ 보다 항상 큰 값을 가지므로, $MINDIST(V_n, Q) + (\sqrt{3}-1) \times R$ 을 FSR로 정할 수 있다.

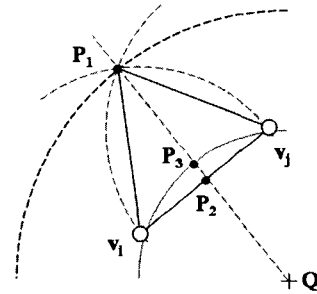
위의 (1)과 (2)에 의해 정리 2를 만족한다. \square



P_1 : v_i 와 v_j 를 중심으로 각각 반경이 R인 두 원의 교점중 Q에 먼 지점

P_2 : $\overline{Qv_i}$ 과 Q를 중심으로 하고 v_i 를 지나는 원과 만나는 교점

(a) 일반적인 경우



P_1 : v_i 와 v_j 를 중심으로 각각 반경이 R인 두 원의 교점중 Q에 먼 지점

P_2 : $\overline{Qv_i}$ 과 $\overline{v_i v_j}$ 가 만나는 교점

P_3 : Q를 중심으로 v_i 와 v_j 를 지나는 원과 $\overline{Qv_i}$ 이 만나는 점

(b) $|\overline{Qv_i}| = |\overline{Qv_j}| = |\overline{v_i v_j}| = R$ 인 경우

그림 11 FSR(Fully-Scanned circular Range)의 반지름 계산

3.6 트랙 퍼리미터 구성시 공 지역(void area) 처리

앞서 트랙 내에 위치한 노드를 경유하여 트랙 퍼리미터를 구성하면, 트랙에 위치한 모든 노드들을 찾을 수 있는 트랙의 폭을 제한하였고, 트랙 퍼리미터를 구성하는 방식으로 OSTP 포워딩 방법을 제안하였다. 일반적으로 센서 네트워크를 구성하는 센서의 개수는 충분히 많은 것이 특징이기 때문에, 트랙 퍼리미터 구성이 가능할 것으로 예상된다. (기존 연구에서는 통신 범위 내 위치하는 노드의 평균 개수, 즉 네트워크 밀집도는 10-20개 정도이다.) 그러나 노드가 존재하지 않는 공 지역이 있을 수 있다.

본 논문에서는 공 지역으로 인해 다음 퍼리미터 노드를 찾을 수 없을 경우에는 GPSR의 퍼리미터 포워딩 방식을 사용하여 해결한다. 그림 12는 OSTP 포워딩 방

식으로 트랙 퍼리미터를 구성할 때 다음 노드를 찾지 못할 경우, GPSR의 퍼리미터 포워딩 방식으로 이를 해결하는 방법을 설명하고 있다. GPSR의 퍼리미터 포워딩은 그리디 포워딩으로 메시지를 전달하는 도중에 다음 노드를 선택할 수 없는 공 지역을 만났을 경우 이를 해결하기 위한 방법이다.

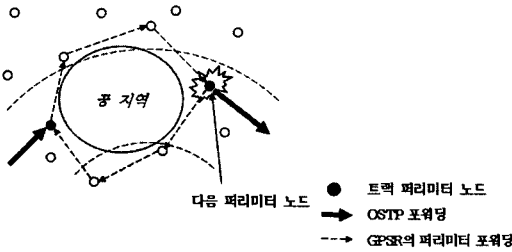


그림 12 공 지역에서 퍼리미터 포워딩

4. 실험 및 분석

4.1 실험 환경

실험에서는 센서 네트워크를 시뮬레이션하여 질의 처리에 소요된 메시지 수를 구한다. 이때 센서의 전체 개수는 네트워크 전체 영역과 네트워크 밀집도에 의해 계산하였다. 본 실험에서 통신 범위 내 센서의 평균 개수, 즉 네트워크 밀집도는 5~20개로 가정하였다. 그리고 예외적인 상황으로 센서가 배치될 수 없는 영역이 존재하는 경우도 고려하였다. 이러한 영역을 공 지역(void area)이라고 한다. 공 지역은 센서들의 통신 영역에 속하지 않는 영역이며 네트워크 전체 영역의 30%까지 랜덤(random)하게 생성시켰다. 센서들의 배치는 공 지역을 제외하고는 균등 분포(uniform distribution)되어 있다고 가정하였다.

4.2 성능 평가 기준

성능 평가는 k개의 객체를 찾기 위해 필요한 메시지 수의 평균 값 M_k (메시지 비율: message ratio)를 기준

으로 하였다. 평가 기준은 다음과 같이 계산한다.

$$M_k = \frac{\text{필요한 총 메시지 수}}{k}$$

4.3 실험 결과 및 분석

실험은 제안하는 방법과 KPT 알고리즘을 비교하였다. 그리고 실험 방법은 근접 노드 개수 k, 통신 범위 내 센서 평균 개수인 네트워크 밀집도, 그리고 통신 영역에 속하지 않은 공 지역 정도에 따른 메시지 수의 평균 값의 변화를 살펴보았다.

● k값에 따른 메시지 비율 변화

그림 14는 k값에 따른 메시지 비율의 변화를 표시한 그래프이다. 그래프에서는 KPT 알고리즘의 경우 k값이 증가하면 메시지 비율이 선형적으로 증가한다는 것을 알 수 있다. 즉, 총 메시지 수는 k값에 제곱으로 비례한다. KPT의 경우 k개의 노드를 찾는 것이 보장되는 탐색 영역을 지나치게 크게 설정한다. 이에 반해 본 논문에서 제안한 방법은 탐색 영역을 점진적으로 확장하여 탐색 영역이 불필요하게 커지는 것을 방지한다.

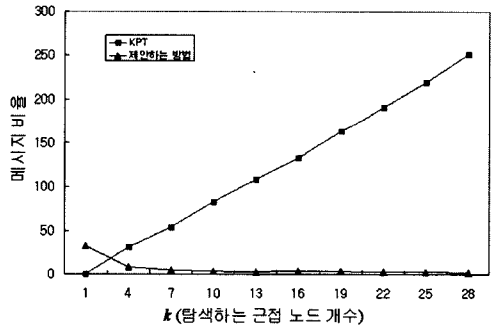


그림 14 k값에 따른 메시지 비율 변화(밀집도=12, 공 지역=30%)

● 네트워크 밀집도 값에 따른 메시지 비율 변화

그림 15는 네트워크 밀집도에 따른 메시지 비율 변화

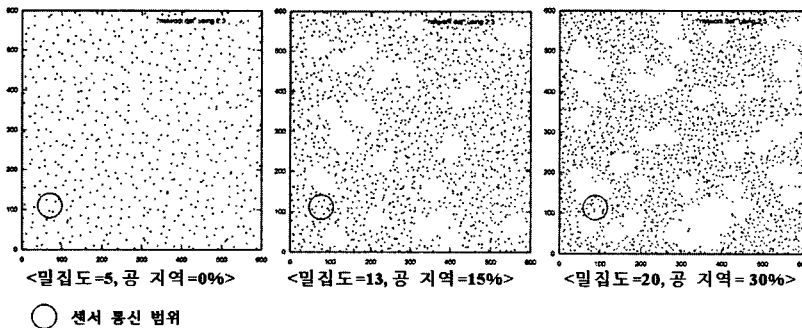


그림 13 네트워크 밀집도와 공 지역에 따른 노드의 배치

를 표시한 그래프이다. 이 경우 k 가 고정되어 있으므로 그래프의 변화는 총 메시지 수의 변화를 의미한다. 그래프에서는 동일한 k 개의 근접 노드를 탐색하더라도 네트워크 밀집도에 따라 필요로 하는 메시지 수가 차이가 난다는 것을 알 수 있다. KPT의 경우에는 네트워크 밀집도와 무관하게 단지 k 값에 따라 탐색 영역을 설정함으로써, 밀집도가 올라갈수록 탐색 영역 내에 있는 노드 수가 증가하여 필요로 되는 메시지 수가 늘어나는 결과를 보여준다. 이에 반해 본 논문에서 제안한 방법은 밀집도가 높을수록 트랙 퍼리미터 구성시 더욱 많은 센서 정보를 가지고 오며, 점진적으로 확장되는 트랙들의 수도 줄어들게 되므로 메시지 수가 줄어드는 것을 알 수 있다. 그리고 제안하는 방법은 그림 14에서와 같이 k 값이 변하더라도 메시지 수의 변화가 적은 것을 알 수 있다. 그림 15에서 k 가 10일 때와 15일 때 제안하는 방법은 거의 차이가 없어서 그래프가 겹쳐서 보이고 있다.

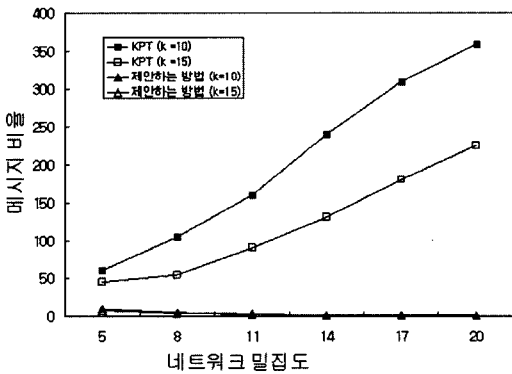


그림 15 밀집도에 따른 메시지 비율 변화($k=10$ 또는 15, 공 지역=0%)

● 공 지역 증가에 따른 메시지 비율 변화

그림 16은 공 지역을 네트워크 전체에서 최대 30%까지 랜덤하게 생성하고, 이에 따른 메시지 비율 변화를

보여주는 그래프다. 이 경우 k 가 고정되어 있으므로 그래프의 변화는 총 메시지 수의 변화를 의미한다. 오른쪽 그래프는 제안한 방법의 성능 변화를 보기 위해 왼쪽에서 확대한 것이다. 실험 결과를 보면 동일한 k 개를 탐색하더라도 공 지역으로 인해 메시지 수에 차이가 있음을 알 수 있다. KPT의 경우에는 공 지역의 존재와 무관하게 탐색 범위가 정해지므로 공 지역이 증가할수록 탐색 영역에 포함되는 센서의 수가 적어지고, 따라서 메시지 수가 줄어드는 것을 알 수 있다. 제안한 방법은 공 지역이 있을 경우 이러한 상황을 해결하기 위한 퍼리미터 포워딩을 하게 되므로, 공 지역이 없을 경우보다 메시지 수가 증가하게 된다. 그래서 공 지역이 많은 경우 필요한 메시지 수가 어느 정도 증가한다. 하지만 KPT 알고리즘보다 메시지 수가 훨씬 적다는 점은 다른 실험과 동일하다. 공 지역의 비율이 증가됨에 따라 공 지역의 위치와 형태가 랜덤하게 생성되기 때문에 그래프에서 다소간의 변동을 보이고 있지만, 양쪽 그래프의 전반적인 경향을 확인할 수 있다.

5. 결론

센서 네트워크에서 분산 질의 처리는 센서의 컴퓨팅 능력을 활용하는 장점을 가지고 있지만, 센서의 제한된 에너지를 가능한 적게 사용하도록 고려되어야 한다. 무선 통신을 사용하는 센서에서 에너지 소모량은 송수신되는 메시지 수가 결정적인 영향을 미치는 것으로 알려져 있다. 그리고 근접 노드 탐색 질의는 센서 네트워크 환경에서도 중요한 질의 중 하나이다

본 논문에서는 센서 네트워크 환경에 적합한 새로운 k 개의 근접 노드 탐색 방법을 제안하였다. 제안된 방법은 k 개의 근접 노드를 찾을 때까지 탐색해야 할 영역을 점진적으로 확장하여 적은 수의 메시지 만을 사용하여 원하는 k 개의 근접 노드를 찾을 수 있는 방법이다. 이를 위해 탐색 영역의 폭과 퍼리미터를 구성하기 위한 새로운 포워딩 방법 그리고 FSR(Fully-Scanned circular

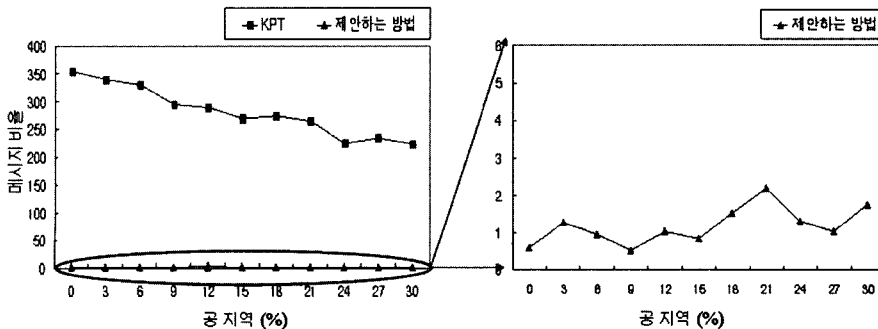


그림 16 공 지역에 따른 메시지 비율 변화 (밀집도=12, $k=25$)

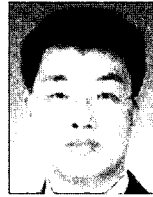
Range) 계산 방법을 제안하였다. 그리고 실험을 통해 제안된 방법이 다양한 센서 네트워크 환경에서 에너지 효율적이라는 것을 보였다.

참 고 문 헌

- [1] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," In IEEE Communications Magazine, 40(8), pp. 102-114, Aug. 2002.
- [2] P. Bonnet, J. Gehrke, and P. Seshadri, "Querying the physical world," Personal Communications, IEEE, 7(5):10-15, 2000.
- [3] J. Gehrke and S. Madden, "Query processing in sensor networks," IEEE Pervasive Computing, 3(1), pp. 46-55, January-March 2004.
- [4] Alan M. Mainwaring, David E. Culler, Joseph Polastre, Robert Szewczyk, and John Anderson, "Wireless sensor networks for habitat monitoring," In ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), pp. 88-97, 2002.
- [5] M.Tamer Ozsu, "Principles of distributed database systems (second edition)," Prentice Hall, 2003.
- [6] Nick Roussopoulos, Stephen Kelley, and Frédéric Vincent, "Nearest Neighbor Queries," Proceedings of the 1995 ACM-SIGMOD Intl. Conf. on Management of Data, pp. 71-79, 1995.
- [7] A. Coman, M.A. Nascimento, and J. Sander, "A Framework for Spatio-Temporal Query Processing over Wireless Sensor Networks," in Proceedings of the 1st International Workshop on Data Management for Sensor Networks (DMSN 2004) in conjunction with 30th VLDB, pp. 104-110, August 2004.
- [8] Leonard Kleinrock and John Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," In Proceedings of National Telecommunications Conference, 1978.
- [9] G.Potte, and W.Kaiser, "Wireless Integrated Network Sensors(WINS): Principles and Approach," Communications of the ACM. 43(5), pp. 551-558, 2000.
- [10] Brad Karp and H.T.Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Mobicom, pp. 243-254, 2000.
- [11] I. Stojmenovic, "Position based routing in ad hoc networks," IEEE Communications Magazine, 2002.
- [12] M. Demirbas and H. Ferhatosmanoglu, "Peer-to-peer spatial queries in sensor networks," In 3rd IEEE International Conference on Peer-to-Peer Computing (P2P '03), pp. 32-39, September 2003.
- [13] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker, "DIFS: A Distributed Index for Features in Sensor Networks," First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA 2003), 2003.
- [14] Neal Sample, Matthew Haines, Mark Arnold and Timothy Purcell, "Optimizing Search Strategies in

k-d Trees," 5th WSES/IEEE World Multi-conference On Circuits, Systems, Communications & Computers (CSCC), May 2001.

- [15] Julian Winter and Wang-Chien Lee, "KPT: A Dynamic KNN Query Processing Algorithm for Location-aware Sensor Networks," in International Workshop on Data Management for Sensor Networks, Toronto, Canada, pp. 119-125, August 2004.



손 인 군

1997년 육군사관학교 정보공학과 학사
2005년 한국과학기술원 전자전산학과 석사.
1998년~현재 육군 전산장교. 관심분야는 시공간 데이터베이스, 센서 네트워크, XML



현 동 준

1999년 한국과학기술원 전산학과 학사
2000년 한국과학기술원 전자전산학과 석사.
2000년~현재 한국과학기술원 전자전산학과 박사과정. 관심분야는 데이터베이스, 데이터마이닝, 센서 네트워크

정 연 든

정보과학회논문지 : 데이터베이스
제 32 권 제 3 호 참조



이 은 규

2000년 동국대학교 전자공학과 학사
2002년 한국정보통신대학교 컴퓨터공학과 석사.
2002년~현재 한국전자통신연구원 텔레메틱스&USN연구단 연구원. 관심분야는 텔레메틱스, 센서 네트워크, Mobile GIS

김 명 호

정보과학회논문지 : 데이터베이스
제 32 권 제 3 호 참조