

USN 미들웨어에서 트라이 구조 쿼드 트리를 이용한 영역 질의 재구성 기법

Region Query Reconstruction Method Using Trie-Structured Quad Tree in USN Middleware

조속경* / Cho Sook Kyoung, 정미영** / Jeong Mi Young,
정현민*** / Jung Hyun Meen, 김종훈**** / Kim Jong Hoon

요약

유비쿼터스 센서 네트워크(USN) 환경에서 사용자 중심의 서비스를 하기 위해서는 영역 질의 처리가 필수적이다. In-network 질의에서 영역 질의를 처리하는 방법으로 R-tree를 이용하는 방법이 선호되었다. 유비쿼터스 센서 네트워크 환경하에서는 센서의 수명이 전체 네트워크의 수명을 좌우하기 때문에 질의가 전달될 센서를 정확하게 선별할 수 있어야 한다. 그러나, R-tree를 이용한 방법은 질의 영역과 MBR(Minimum Boundary Rectangle) 간의 교집합 연산에 의해 센서를 선별하기 때문에 실제 센서가 없는 영역임에도 불구하고 센서 값 측정을 요청하는 메시지를 전달하게 된다. 이러한 문제를 해결하기 위해 본 논문에서는 기지국의 미들웨어에 트라이 구조 쿼드 트리를 구축하여 이를 이용하여 정확하게 질의 영역에 포함되는 센서 노드들에게만 질의를 전달한다. 이 기법은 질의 수행시 R-tree를 이용한 방법보다 응답 시간이 늦어지지만, 필요한 센서 노드에만 정확히 질의를 전달하기 때문에 USN의 전력 소모를 감소시키는 장점이 있다.

Abstract

In ubiquitous sensor networks(USN) environment, it is essential to process region query for user-demand services. Using R-tree is a preferred technique to process region query for in-network query environment. In USN environment, USN middleware must select sensors that transfers region query with accuracy because the lifetime of sensors is that of whole sensor networks. When R-tree is used, however, it blindly passes the region query including non-existent sensors where MBR(Minimum Boundary Rectangle) of R-tree is intersected by region of query. To solve in this problem, we propose a reconstruction of region query method which is a trie-structured Quad tree in the base station that includes sensors in region of query select with accuracy. We observed that the proposed method delays

-
- 논문접수 : 2007.12.18 ■ 심사완료 : 2008.2.13
 - * 교신저자 (주)케이지아이 개발부(skcho@kgi.co.kr)
 - ** (주)KT 인프라연구소 책임연구원(myjeong@kt.com)
 - *** (주)KT 인프라연구소 수석연구원(hmjung@kt.com)
 - **** (주)케이지아이 대표이사(jhkim@kgi.co.kr)

response time than R-tree, but is useful for reducing communication cost and energy consumption.

주요어 : 유비쿼터스 센서 네트워크, 공간 인덱스, 질의 영역 재구성

Keyword : USN(Ubiquitous Sensor Networks), spatial index, reconstruction of query region

1. 서론

최근 정보화 사회가 유비쿼터스 사회로의 진입에 따라 언제 어디서나 사용자가 원하는 정보를 전달해줄 수 있는 사용자 중심의 서비스(user-demand services)에 대한 연구가 활발히 진행되고 있는데, 현재의 상태는 사용자가 원하는 정보를 공급자가 가공해서 일방적으로 전달해주는 방식을 취하고 있다[1]. 예를 들어 교통 정보의 경우 사람의 확인을 거쳐 어느 도로가 막힌다는 정도의 단편적인 정보만 전달하고 있다. 이러한 방식은 복잡하고 정밀한 답을 요구하는 사용자를 만족시키지 못하는 상황이지만 가까운 미래에 센서의 보급을 통해 도로 상황을 자동으로 수집, 분석하여 사용자가 원하는 형태로 정보가 제공될 것으로 기대하고 있다[1].

유비쿼터스 사회의 핵심 인프라로 주목받고 있는 센서[2]에 관한 연구는 저장 공간, 데이터 전송, 제한된 전원 등의 제약 사항을 고려한 센서 네트워크 구축 방법과 센서에서 수집한 방대한 양의 스트림 데이터를 관리하는 방법에 초점을 두고 있다. 대표적인 연구로 TinyDB, Cougar, STREAM, Telegraph, Aurora, SCADDS, UbiCore 등이 있다. 이러한 연구들은 센서 데이터들을 내부, 외부 저장 장치에 저장하여 웨어하우징 방식으로 처리하거나 내부 네트워크 질의(in-network query) 처리 방식을 사용하여 질의 대상이 되는 센서를 선정하여 센서들 사이의 데이터 전송량과 통신량을 줄여 통신 비용의 감소와 에너지 절약 효과를 가져오는 방식에 대한 연구들이다[3-9]. 이 기법들은 센서에 대한 가능성을 충분히 증명은 하였지만 원하는 영역에 대한 센서 데이터 값을 추출하는 영역 질의와 응답 시간에 대한 고려가 미비하다[10]. 예

를 들어, 교통 사고가 발생한 지점에서 센서가 이벤트 질의를 통해 사고 발생 상황 정보를 기지국(base station)으로 보내고, 기지국에서는 사고 주변의 센서 정보를 통합 분석하여 사용자에게 교통 상황을 보내주어 다른 도로를 선택하게 만들어주어야 한다. 교통사고 지점 주변의 센서 데이터를 가져오는 질의는 전체 센서를 대상으로 하지 않으므로 특정 영역의 센서를 통해 데이터를 전달받아야 센서의 에너지를 절약하며, 사용자에게 응답하는 시간을 감소시킬 수 있다. 특히, 사용자 중심 서비스에서는 사용자의 위치 정보와 연계한 센서 데이터 값을 추출하려는 질의가 빈번하게 발생하므로 영역 질의 처리와 응답 시간의 감소가 가능한 방법이 필요하다.

본 논문에서는 사용자 중심 서비스를 하기 위해 영역 질의가 가능한 MBR(Minimum Boundary Rectangle) 기반 센서 네트워크를 구축하고, 연속적으로 발생하는 센서 데이터를 지역적으로 가까운 센서들을 같은 버킷 내에 저장시킴으로서 향후 데이터 분석에 효과적으로 대처할 수 있는 저장 방법과 센서 노드들을 MBR로 표현하는 문제로 인하여 질의 영역 내에 포함되지 않는 센서들도 영역 질의 대상이 되는 문제를 해결하고자 한다. 사용자 중심의 서비스를 위해 USN(ubiquitous sensor networks) 미들웨어에 지역성을 고려하도록 MBR과 집계값을 갖는 퀴드 트리를 트라이 구조로 구축하고, USN 어플리케이션에서 요구하는 질의는 퀴드 트리에 보관된 집계값으로 응답을 하여 시간을 단축시킨다. USN에 구축된 라우팅 트리를 통해 특정 영역에 속한 센서 데이터 값을 실시간으로 요구하는 경우에는 퀴드 트리를 이용하여 질의 영역의 조정을 통해 실제로 센싱이 필요한 센서들에게만 질의를 전달하여 센서의 에너지를 절약하며 이를 통해 전체 USN

의 수명을 연장시키는 효과를 얻을 수 있다.

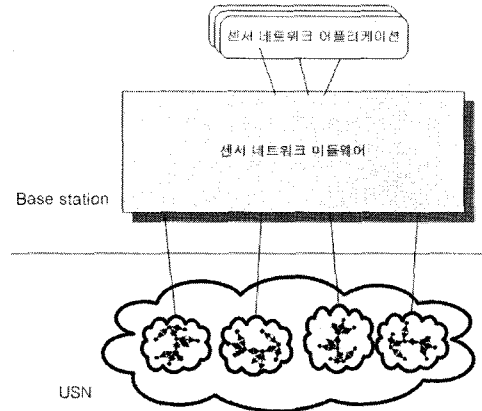
본 논문은 2장 관련 연구에서 USN 미들웨어와 기존의 영역 질의를 위한 라우팅 트리 구축 기법을 살펴본 후, USN을 위한 영역 질의에 가장 적합하다고 여겨지는 R-tree 기반 라우팅 트리의 문제점을 논의한다. 3장에서는 2장에서 도출한 문제점을 해결하기 위해 USN에 적합한 라우팅 트리를 선정하고, 이를 기반으로 USN 미들웨어에 트라이 구조 쿼드 트리의 구축과 질의 실행 과정, 센서들로부터 직접 데이터를 원하는 경우 질의 영역을 조정하여 라우팅 트리를 통해 질의를 전달하는 과정을 설명한다. 4장에서 영역 질의 재구성 기법과 기존의 기법을 성능 평가를 통해 비교하며 5장에서 결론을 맺는다.

2. 관련 연구

2.1 USN 미들웨어

사물에 센서를 부착하여 사용자에게 필요한 정보를 적시에 제공하기 위해서는 센서의 기능 확장, 센서 네트워크의 구축, 센서 어플리케이션의 연구와 더불어 USN과 센서 네트워크 어플리케이션을 연결해 줄 수 있는 USN 미들웨어의 연구 개발이 시급히 이루어져야 한다[10,11]. <그림 1>의 센서 네트워크의 구조에서 보듯이 센서 네트워크 어플리케이션은 미들웨어를 통해 USN과 연결되는데, 미들웨어는 sever-side 미들웨어와 in-network 미들웨어로 구성된다. 응용 서비스 지원을 위해 server-side 미들웨어는 서버 시스템에 설치되고, 다수의 USN 응용 서비스 관리, 응용 서비스에서 발생하는 다중 질의 처리, 센싱된 연속 데이터들과 센서들을 처리하기 위한 메타 정보의 관리 등의 일을 수행한다. 또한, 기존의 데이터들과 센싱 데이터들을 통합, 분석하여 앞으로 요구될 수 있는 상황 정보 생성 및 지능형 이벤트 처리의 수행이 필요하다. In-network 미들웨어는 센서 노드와 싱크 노드 수준에서의 질의 처리, 센서 노드 네트워크 위상 관계 관리, 센서 노드의 상태 관리 등을 수행해야

한다[11].



<그림 1> 센서 네트워크의 구조

USN에서 발생하는 질의 유형으로는 현재 센싱된 정보를 실시간 요청하는 snapshot 질의, 센싱한 데이터를 일정한 주기로 연속적으로 요청하는 continuous 질의, 특정 상황이 발생한 경우에만 센싱 데이터를 요청하는 이벤트 질의, 시간에 따른 위치 정보의 획득이 가능한 spatio-temporal 질의가 있다[11]. 영역 질의는 각각의 질의에 영역을 추가한 상태로 cougar[4]의 질의 모델을 참조하면 “위치 좌표 값이 (10, 20), (100, 200)인 A 고속도로의 오전 7시부터 9시 사이의 차량 대수의 합계를 10분 간격으로 구하십시오.” 라는 질문에 다음과 같은 영역 질의를 작성할 수 있다.

```
SELECT sum(value)
FROM sensor_nodes S
WHERE S.loc IN [(10, 20), (100, 200)]
DURATION [7:00, 9:00]
EVERY 10m;
```

센서 어플리케이션에서 보낸 질의들은 기지국인 루트 노드에서부터 <QID, Region, Attribute, Op, Time, Frequency> 형식의 질의 요소를 센서 네트워크로 전달한다. QID는 질의를 구분하기 위하여 기지국인 루트 노드에서 부여하는 유일한 식

별자이다. Region은 센싱하고자 하는 질의 영역이고, Attribute는 검색하려는 변수이다. Op는 집계 연산자 MAX, MIN, AVG, SUM, COUNT 등이 될 수 있다. Time은 질의 기간으로 센서 노드가 센싱해야 되는 <시작 시간, 종료 시간>을 나타내며, Frequency는 센서 노드가 센싱하는 주기이다.

2.2 영역 질의가 가능한 센서 네트워크 구축 기법

센서 네트워크의 구축 기법으로 영역 질의에 대한 관점을 가지고 연구된 기법으로는 SRT (semantic Routing Tree) 기법, DIFS 기법, DIM 기법과 SPIX 기법이 있다. SRT[3] 기법은 TinyDB에서 센서 네트워크에 대한 질의 처리 기법으로 제한한 집계 트리로서, 센서 노드 자신의 하위 노드들이 질의의 결과값을 구하는데 필요한 노드를 판단하는 기준을 갖는다[9]. 센서의 좌표값을 가지고 라우팅 트리를 구성하였기 때문에 좌표값 비교를 통하여 결과 값을 도출하는데 필요한 노드인지 아닌지를 구분할 수 있다. 질의의 결과를 도출하는데 필요한 노드를 결정하고 자식 노드로 질의를 전파하여 관련 노드들로부터 질의 전달의 역방향으로 센싱된 값을 전달하여 결과 값을 산출한다. 이 기법에서의 영역 질의는 센서 노드의 좌표 값을 기준으로 하기 때문에 기준이 되는 좌표의 경우 간단히 질의에 포함되는 센서인지 아닌지 구분가능하나 기준이 되지 않는 좌표는 구분 가능하지 않으므로 질의 영역에 포함되지 않는 센서들이 결과를 도출하는데 포함될 가능성이 있다.

DIFS[12]는 Quad tree[13]의 각 노드를 각각의 센서로 대응하는 방식의 토폴로지를 구성하여 영역 질의를 지원한다. 트리 구조 토폴로지의 가장 큰 단점이 상위 센서가 많은 메시지 전송을 담당해야 하는 것이므로 가급적 하위 센서에 많은 데이터 영역을 할당하는 방식으로 부하를 분산하도록 하였다. 이로써 네트워크 전체의 에너지 소모는 어느 정도 늘어나지만 메시지 전송과 저장 간의 균형을 맞추도록 하였다. 이는 단일 데이터에 대한 색인 구조

로서 한 개의 센서 노드에 여러 개의 센서가 포함된 경우에 대한 고려가 없는 문제점이 있으며[9], 기지국에서 질의가 전파될 때 한 번에 목적지 센서까지 갈 수 없고 여러 센서를 거쳐야 하는데, 최적의 경로 선정을 위한 고려가 없으므로 실시간 응답이 불가능한 경우가 발생한다.

DIM[14]은 k-d Tree 방식으로 센서 데이터에 대한 분산 색인을 구성한다. 각각의 센서는 위치에 따라 2차원 영역을 분할하여 관리하며, x와 y 좌표를 번갈아 균등 분할하는 방식으로 각 센서에게 좌표 영역을 할당한다. 각 센서는 k-d Tree의 단말 노드에 해당하는 다차원 데이터 영역을 담당하며, 센서 데이터의 지역성을 보장하기 위한 위치기반 해쉬 기법을 사용하여 적은 수의 라우팅에 의해 영역 질의를 처리할 수 있다. 그러나 데이터의 지역성에 치중하여 특정 센서들에 부하가 집중됨으로써 네트워크의 전체 동작시간을 단축시킬 수 있는 문제점이 있다[9].

SPIX 기법[15]은 rectangle 기반의 라우팅 트리 구축 기법으로 공간 질의를 위한 인덱스 기법인 R-Tree 구조로 라우팅 트리를 구축한다. 각 센서 노드들을 R-tree의 부모, 자식 노드로 구성하여 트리를 구축하였으므로 영역 질의에 SRT 기법보다 우수한 성능을 보인다. 이 기법에서는 영역 질의시 적은 수의 센서 노드에 질의를 전파하기 위하여 센서의 MBR과 영역 질의의 MBR을 교집합하여 질의 전달 여부를 판단한다. 센서 노드들의 MBR 구성시 멀티 홉을 가능하게 구성하는가 1 홉만을 부모 자식 관계로 구성하는가에 따라 질의 영역에 포함되는 센서의 수가 달라지므로, 좀 더 적은 수의 정확한 센서 노드들만 질의에 참가시키기 위한 SPIX 트리의 변형 기법[16]이 연구되고 있다. R-tree 기반의 라우팅 기법들은 SRT 기법 보다 제한된 센서 노드의 저장공간에 저장할 메타데이터의 양이 크다는 단점이 있다.

2.3 문제 정의

센서 데이터는 상태 기반 데이터로서 센서 어플

리케이션에서 요구하는 데이터를 요구하는 시간 내에 사용자에게 정확한 데이터를 전달해야 한다.

외부 저장 장치에 저장된 센서 데이터를 이용하려는 기존의 연구는 센서 네트워크를 구축할 때와 동일하게 R-tree 기반의 색인을 구축하여 센서의 위치 정보를 영역 질의에 이용하고 R-tree의 각 노드마다 집계값(MAX, MIN, SUM, AVG, COUNT 등)을 저장하여 응답 시간을 감소시키려는 시도를 하고 있다. 센서 데이터 관리 시스템에서 영역 질의를 위해 R-tree 기반의 기법을 선호하는 이유로 성능의 향상과 센서 영역의 확장에 따라 트리의 유연한 확장성을 들 수 있다. 하지만, 센서의 부모 노드와 그 부모에 속한 자식 노드들을 하나의 MBR로 구성하기 때문에 영역 질의시 센서가 포함되지 않은 MBR이 후보 영역으로 선택될 가능성이 있다.

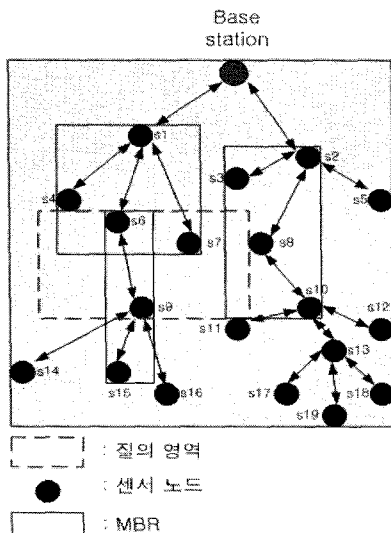
<그림 2>에서 점선으로 표시한 부분이 질의 영역이라 할 때 s2 노드가 부모인 rectangle이 선택되어 잘못된 집계값을 사용자에게 전달하는 문제가 발생하며, 선택된 센서들의 위치 값을 전부 확인해야 영역 질의에 포함되는 센서를 찾을 수 있다. 또한, 질의 영역에 포함된 센서들에게서 데이터를 추출하려면 해당 센서까지 트리 토폴로지에 따라 질

의를 전달해야 하므로 MBR의 잘못된 선택은 센서의 에너지를 낭비하는 결과를 초래한다. 기지국에서 s1, s2로 전달된 질의는 자신의 저장 공간의 메타 데이터를 참조하여 질의 영역과 교집합 결과가 공집합이 아닌 MBR에 속한 자식 센서 노드들에 질의를 전파한다. 센서 s1을 통해서는 s4, s6, s7로 질의가 전달되며, 실제 질의 영역에 속한 센서가 하나도 존재하지 않는 부모 센서 노드 s2를 통해 자식 노드들에 질의가 전달되는 문제가 발생한다.

3. 영역 질의 재구성을 위한 트라이 구조 쿼드 트리

사물에 센서를 부착하여 사용자에게 필요한 정보를 적시에 제공하기 위해서는 센서 기능의 확장, 센서 네트워크의 구축, 센서 어플리케이션의 연구와 더불어 USN과 센서 어플리케이션을 연결해 줄 수 있는 USN 미들웨어가 필수적이다.

이 장에서는 미들웨어에 라우팅 트리에 대한 정보를 저장한 후 영역 질의 처리를 위한 트라이 구조 쿼드 트리를 구축하는 방법, 쿼드 트리를 이용한 영역 질의 처리 방법과 라우팅 트리를 통해 센서 데이터를 추출하려 할 때 질의 영역을 최적화 하는 방법을 설명한다.



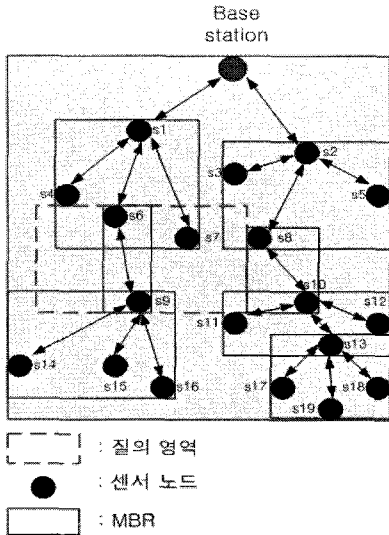
<그림 2> SPIX 기법에서 영역 질의시 문제점

3.1 In-network 영역 질의를 위한 라우팅 트리 구축

센서들로부터 데이터 값을 주기적으로 측정하거나 특정 센서로부터 값을 추출하기 위해 센서 네트워크를 구축해야 한다. 본 논문에서는 영역 질의에 좋은 성능을 보이는 것으로 증명된 SPIX 기법을 변형한 라우팅 트리 기법인 [16]에서 제안한 라우팅 트리를 기반으로 하여 센서 네트워크를 구축한다.

구축된 센서 네트워크에서는 기지국을 루트 노드로 하고 센서 노드를 각각의 내부 노드, 터미널 노드로 하여, 노드들 사이 또는 노드와 기지국 사이의 무선 통신 연결로 트리 구조를 구성한다. 트리를 구성하는 센서들은 통신 가능 범위에 있는 노드들끼

리 양방향 링크로 서로를 연결하며 부모 노드와 자식 노드의 관계를 가진다. 하나의 노드는 하나의 부모만을 가지며, 부모 노드와 자식 노드들을 포함하는 노드들을 하나의 MBR로 구성하며, 그 정보는 부모 노드에 저장된다. 따라서, 하나의 MBR 내에서 부모 노드와 자식 노드는 한 홑 통신 거리를 가지게 되나, 구축된 트리 구조는 완전 이진 트리가 아닌 상위 레벨에 터미널 노드가 존재할 수 있는 구조가 된다. 센서 네트워크를 구성하는 라우팅 트리 구조는 다음 <그림 3>과 같다.



<그림 3> 질의 영역이 표시된 변형 SPIX 기법

라우팅 트리의 구축 후 각 센서 노드에 저장된 메타 데이터 구조는 아래 <그림 4>와 같으며, 여기에는 자신의 정보를 가지는 센서 노드 정보, 자신의 부모 노드 정보, 자신의 모든 자식 노드들의 정보를 저장하고 있다.

센서 노드 정보에서는 자신의 ID와 x,y 좌표로 구성된 Loc, 기지국인 루트 노드로부터 상대적인

거리를 나타내는 Level을 가진다. 기지국을 루트로 하여 Level 0이라 할 때, 기지국으로부터 직접 질의를 전달받는 센서 노드들은 Level 1이 된다. 부모 노드의 Level이 n이면 그 자식 노드들은 모두 Level n+1이 된다. 부모 노드 정보로는 부모 ID와 위치 정보인 PID, Loc를 저장하고 있다. 부모 노드의 정보는 질의가 터미널 노드까지 전달되고 추출된 결과 값을 기지국으로 돌려보낼때 질의가 전파된 역순으로 결과값을 전달하기 때문에 필요한 정보이다. 자식 노드 정보에는 하나의 센서 노드에 1 홑 거리로 연결된 자식 노드들의 개수만큼 자식 노드 엔트리를 만들어 저장한다. 자식 노드 정보에 저장된 데이터들은 각 자식 노드의 ID, 위치 값 Loc, 자신과 후손들을 모두 포함하는 MBR로 구성되어 있다. 자식 노드들의 정보는 영역 질의가 전달될 때 MBR의 값과 비교하여 질의 영역과 교집합이 있는 경우 추출 값을 받아오기 위해 질의를 전달하고 교집합이 없는 경우에는 질의를 전달하지 않는다.

3.2 질의 영역 재구성을 위한 MBR을 갖는 쿼드 트리

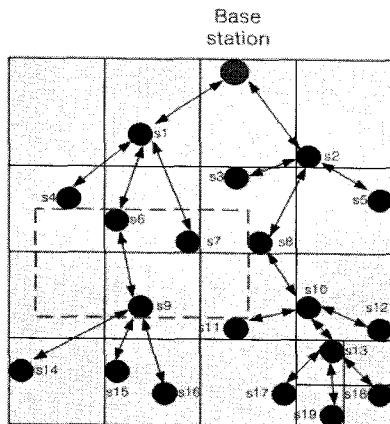
기지국에 구축하는 색인은 센서 노드에서 발생하는 방대한 양의 데이터들과 사용자 중심 서비스를 위해 지역성을 고려해야 한다. 지역성이란 사용자의 위치에서 근접한 영역에 대한 질의가 빈번하게 발생하고, 센서가 분포한 전 지역에 대한 질의 보다는 일부 영역에 대한 질의가 빈번하게 발생한다는 의미이다. 방대한 양의 센서 데이터에 대한 영역 질의를 효율적으로 수행하기 위해서는 지역적으로 근접한 데이터들을 함께 저장하는 것이다. 제안 기법에서는 '지리적으로 근접한 센서 데이터들을 하나의 버킷에 함께 저장하며, 함께 저장될 센서 노드들의

센서 노드 정보			부모 노드 정보		자식 노드 정보						
NodeID	Loc	Level	PID	Loc	ID 1	Loc	MBR	ID n	Loc	MBR

<그림 4> 센서 노드 정보

데이터를 위해 영역을 사분분할(quad-split)하여 일정 개수의 센서 노드들의 데이터만을 하나의 버킷에 저장한다.

USN의 라우팅 트리 구축시 각 센서 노드에 저장된 자신의 정보, 부모 노드의 정보, 자식 노드들의 정보를 기지국에 알려주어 메타 데이터를 구축한다. 색인 구축의 처음은 메타 데이터로 저장된 센서 노드들의 정보를 관독하여 Level 1인 센서 노드들을 색인에 추가한다. 색인 영역에 추가된 센서 노드들의 개수가 버킷에 들어갈 개수보다 크면 색인 영역을 사분분할 한다. 분할된 각 영역을 셀(cell)이라고 하고, 셀에 분포한 센서의 개수가 버킷에 들어갈 개수보다 작거나 같을 때까지 계속 사분분할을 수행하여 버킷에 적정 개수의 센서 노드들만 유지하도록 한다. 마지막 레벨 센서 노드들까지 위 단계를 반복한다.



- [- - -] : 질의 영역
- : 센서 노드
- : MBR

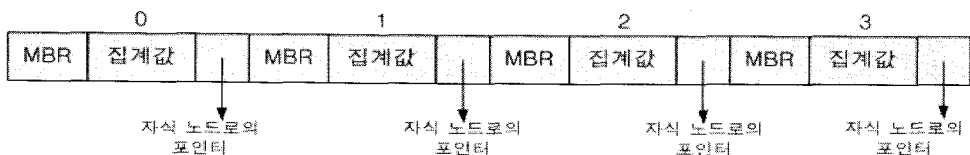
<그림 5> 센서 영역의 사분분할

<그림 5>는 버킷에 들어가는 센서 노드의 수를 두 개로 했을 때 마지막 레벨의 센서 노드까지 색인 영역에 추가한 후 센서 영역이 사분분할이 된 모습을 나타낸다. 색인 구조는 사분분할 특성에 맞고, 검색의 효율을 높이기 위하여 트라이로 구성한다. 트라이의 노드 구조는 다음 <그림 6>과 같다. 색인 영역의 사분분할 후 각각의 셀을 구분하기 위해 z-order로 번호를 좌하(left bottom)-우하(right bottom)-좌상(left up)-우상(right up) 순으로 준다. 각 셀을 표현하는 노드는 현재 셀에 분포하는 센서 노드들을 모두 포함하는 MRB, 그 셀 내의 센서 노드들의 집계값, 해당 셀이 다시 분할되면 만들어지는 노드(자식 노드)를 가리키는 포인터로 구성된다. 예를 들어 센서 s10은 셀 번호 13에 속하며, 이 번호는 센서의 데이터 값을 직접 접근하기 위한 버킷의 주소로 이용된다. 센서 s10은 위치 값이 (13,6), 측정 값 15이고 s12의 위치 값이 (16,5), 측정 값 7일 때 집계값으로 합계를 구한다고 하면 셀 번호 13에 대한 노드의 구성은 <((13,5),(16,6)),(22),(버킷13)>이 된다.

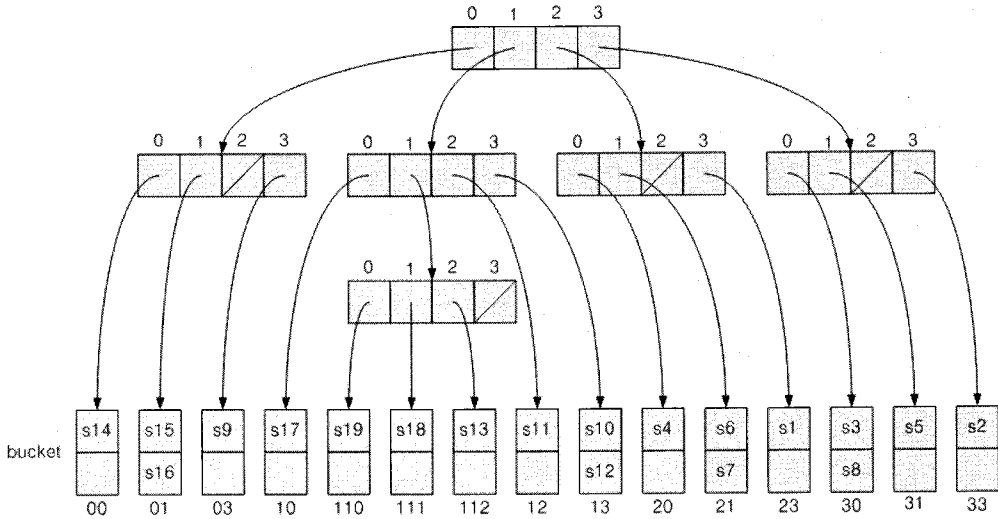
<그림 7>은 <그림 5>를 가지고 구성한 색인 파일의 구조이다. 사분분할 후 해당 영역에 센서 노드가 존재하지 않은 경우에는 자식 노드로의 포인터를 null로 처리한다.

3.3 질의 영역 재구성을 이용한 영역 질의 처리 기법

USN에서 영역 질의를 처리하기 위해서는 기지국 내 버킷에 저장된 센서 데이터를 대상으로 할 것인지 센서 네트워크를 대상으로 할 것인지를 결정해야 한다. 버킷에 저장된 센서 데이터들은 정규



<그림 6> 트라이 노드 구조



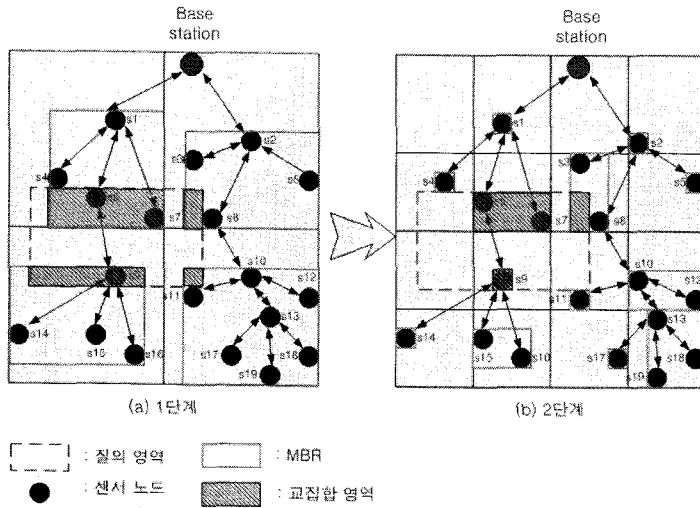
<그림 7> 트라이 구조로 표현한 쿼드 트리

적 또는 이벤트 질의에 의해 이미 센싱된 데이터로 현재 영역 질의의 유효 시간 내에 포함된 데이터들 이라면 영역 질의의 결과값으로 반환해도 된다. 그러나, 사용자가 현재 시간의 데이터를 요구하는 영역 질의를 하거나 이벤트 질의에 대한 트리거로서 영역 질의가 발생하는 경우에는 센서 네트워크에 질의를 전달하여 센서로부터 데이터를 센싱해야 한다.

구축된 색인을 이용하여 영역 질의를 처리하는데 셀의 MBR과 질의 영역은 다음과 같은 세가지 교집합 관계를 가진다.

- 포함(contains) 관계: 질의 영역에 셀의 MBR이 포함되는 경우로 해당 셀 영역의 모든 센서가 질의 대상이 된다.
- 배타(disjoint) 관계: 질의 영역과 셀의 MBR이 교집합이 공집합인 경우로 해당 셀에 있는 센서들은 질의의 대상이 아니므로 검색하지 않는다.
- 겹침(overlaps) 관계: 질의 영역과 셀의 MBR이 겹치는 경우 해당 셀 영역의 센서가 질의 영역에 포함될 수도 있고 아닐 수도 있으므로 포함 관계나 배타 관계가 나타날 때까지 검색을 반복한다.

예를 들어 <그림 5>에서 점선으로 표시된 영역 질의를 처리하는 과정을 논리적으로 표현한 것이 <그림 8>이다. 질의 영역과 루트 노드의 트라이 0, 1, 2, 3번의 MBR과 교집합을 구하면 첫 번째 부분분할에서 0, 1, 2, 3번 영역 모두에서 <그림 8(a)>에서 처럼 겹침 관계를 발견할 수 있으므로 자식 노드를 가리키는 포인터를 따라 다음 노드로 가면서 각각 MBR과 질의 영역의 교집합을 구한다 (<그림 8(b)>). 셀번호 00, 01, 02, 03에 해당하는 MBR을 비교하면 00, 01은 배타 관계이므로 질의를 전달할 센서가 없다는 의미이므로 더 이상 교집합을 구하지 않는다. 02의 경우 교집합은 존재하나 자식 노드의 포인터가 null이므로 해당 영역에는 센서가 존재하지 않으므로 마찬가지로 교집합을 구하지 않는다. 03의 경우 교집합이 존재하고 MBR이 포함 관계이므로 자식 포인터가 가리키는 모든 버킷에서 질의 결과 값을 받아온다. 모든 셀에 동일한 연산을 수행하여 선택된 셀 번호는 03, 21이 선택되어 영역 질의에 포함된 센서는 s6, s7, s9임을 알 수 있다.



<그림 8> 쿼드 트리를 이용한 질의 영역 처리 단계

위에서 설명한 질의 영역 처리 단계를 위한 센서 데이터 추출 알고리즘은 다음과 같다. 쿼드 트리의 루트 주소 trie와 질의 영역 r을 입력으로 받아 각 노드가 가진 집계값을 반환하는 알고리즘이다.

```

    } //end of switch
  } //end of for
} while(trie[i].child != NULL);
}

```

[알고리즘 1] 영역 질의에 대한 센서 데이터 추출 알고리즘

입력 : 트라이 쿼드 트리의 루트 trie, 질의 영역 r
 출력 : 센서들의 집계값 agg
 Algorithm selSensorVal(trie, r)

```

{
do
{
for(i=0; i<4; i++) {
switch(intersect(trie[i].MBR, r) {
case OVERLAP :
selSensorVal(trie[i].child, r);
case CONTAINS :
return trie[i].agg;
case DISJOINT :
break;
case default :
searchBucket(r);
}
}
}
}

```

트라이 노드의 네 개의 엔트리에 저장된 MBR과 질의 영역 r을 교집합 연산 intersect를 하여 OVERLAP 관계이면 자식 노드 trie[i].child와 질의 영역 r을 입력 값으로 하여 재귀적으로 동일 함수를 호출한다. CONTAINS 관계이면 그 트라이 노드 엔트리의 모든 센서가 질의 영역에 포함되는 센서이므로 저장된 집계값 trie[i].agg을 반환한다. DISJOINT 관계의 경우 교집합이 존재하지 않으므로 검색을 멈춘다. 그 이외의 경우는 트라이 노드를 모두 거쳐 버킷까지 도달한 경우로 버킷에 저장된 센서들의 위치와 질의 영역 비교가 일어나야 하므로 searchBucket() 함수를 호출한다. 알고리즘 searchBucket()은 다음과 같다.

[알고리즘 2] 버킷내의 센서 값 추출

입력 : 질의 영역 r
 출력 : 센서 데이터 값 agg
 변수 : 버킷 내의 센서를 가리키는 sensor

```

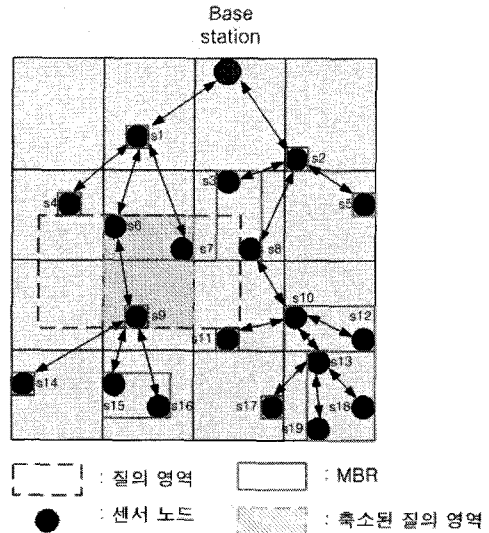
algorithm searchBucket(r)
{
  do {
    switch(intersect(sensor.Loc, r)) {
      case OVERLAP :
      case CONTAINS :
        return sensor.agg;
      case DISJOINT :
        sensor = next sensor;
    } //end of switch
  } while(no more sensor);
}
    
```

입력으로 질의 영역 r이 필요하고, 출력은 질의 영역에 포함된 센서의 집계값이 된다. 버킷 내에 저장된 각 센서 위치와 질의 영역의 교집합 연산이 필요하며 알고리즘 1에서와 동일한 관계가 발생한다. OVERLAP과 CONTAINS 관계이면 질의 영역 내에 포함되는 센서이므로 센서가 가지고 있는 집계값을 반환한다.

사용자가 현재 시점의 센서 데이터 값을 요구하거나 이벤트의 발생에 따른 트리거로서 영역 질의를 발생시키면 센서 네트워크 내 라우팅 트리를 통해 센서 데이터 값을 반환받아야 한다. 기존의 기법은 영역 질의가 발생하면 기지국과 바로 통신할 수 있는 Level 1인 센서 노드들에게 질의를 전달하고, 센서 노드 정보 내의 자식 노드들의 MBR과 질의 영역의 교집합 관계를 비교하여 겹침이나 포함 관계이면 질의를 반복적으로 각 Level 1의 센서 노드들로 전파한다. 이 경우 자식 노드의 MBR이 자신의 후손을 모두 포함하는 영역이므로 실제 질의 대상에 포함되지 않는 MBR도 선택되어 질의를 전파하므로 센서 노드의 에너지를 불필요하게 소모시킨다. 터미널 노드의 데이터를 센싱하기 위해서는 기지국부터 시작하여 자식 노드들에게로 차례로 질의가 전파되어야 한다는 센서 네트워크의 특성 때문이다.

이러한 문제를 해결하고자 제안하는 기법은 기지국 내에 구축된 트라이 구조 쿼드 트리를 이용해

실제 질의 영역에 포함되는 쿼드 트리의 셀을 파악한 후 각 MBR을 재구성하여 질의 영역을 축소시킨다. 이렇게 최적화된 질의 영역을 가지고 기지국 내에 저장된 Level 1 센서 노드의 MBR과 비교하여 포함, 겹침 관계를 가진 센서 노드를 선별하여 질의를 전파한다. 아래 <그림 9>는 <그림 8>의 과정을 거쳐 재구성된 질의 영역을 나타내고 있다.



<그림 9> 재구성된 질의 영역

재구성된 질의 영역을 얻기 위한 질의 영역 재구성 알고리즘은 다음과 같다. 입력으로 트라이 쿼드 트리 루트 trie와 질의 영역 r을 받아 재구성된 질의 영역 rr을 결과값으로 구하는 알고리즘이다.

[알고리즘 3] 질의 영역 재구성 알고리즘

입력 : 트라이 쿼드 트리의 루트 trie, 질의 영역 r
 출력 : 재구성된 질의 영역 rr

```

algorithm reRegion(trie, r)
{
  rr=0;
  do {
    for (i=0; i<4; i++) {
      switch(intersect(trie[i].MBR, r)
      case OVERLAP :
    
```

```

    reRegion(trie[i].child, r);
case CONTAINS :
    rr=rr+trie[i].MBR;
    break;
case DISJOINT :
    break;
case default :
    rr = rr + reRegionBucket(r);
} //end of switch
} //end of for
} while(trie[i].child != NULL);
return rr;
}

```

```

do {
    switch(intersect(sensor.Loc, r)) {
    case OVERLAP :
    case CONTAINS :
        rr = rr + sensor.Loc;
        sensor = next sensor;
        break;
    case DISJOINT :
        sensor = next sensor;
    } //end of switch
} while(no more sensor);
return rr;
}

```

알고리즘 1과 마찬가지로 트라이 노드의 네 개의 엔트리에 저장된 MBR과 질의 영역 r을 교집합 연산 intersect를 하여 OVERLAP 관계이면 자식 노드 trie[i].child와 질의 영역 r을 입력 값으로 하여 재귀적으로 동일 함수를 호출한다. CONTAIN 관계이면 그 트라이 노드 엔트리의 모든 센서가 질의 영역에 포함되는 센서이므로 라우팅 트리로 질의를 전달할 때 트라이 노드의 엔트리가 가진 MBR은 질의 영역의 부분 집합이 된다. 그러므로, 재구성 질의 영역 rr에 해당 MBR을 추가하여 질의 영역을 재구성한다. DISJOINT 관계의 경우 교집합이 존재하지 않으므로 현재 위치한 트라이 노드부터는 검색을 하지 않는다. 그 이외의 경우는 트라이 노드를 모두 거쳐 버킷까지 도달한 경우로 버킷에 저장된 센서들의 위치와 질의 영역 비교가 일어나야 하므로 reRegionBucket() 함수를 호출한다. 알고리즘 reRegionBucket()은 다음과 같다.

[알고리즘 4] 버킷 내의 센서 영역 추출
 입력 : 질의 영역 r
 출력 : 센서들의 영역 rr
 변수 : 버킷 내의 센서를 가리키는 sensor
 algorithm reRegionBucket(r)
 {
 rr = 0;

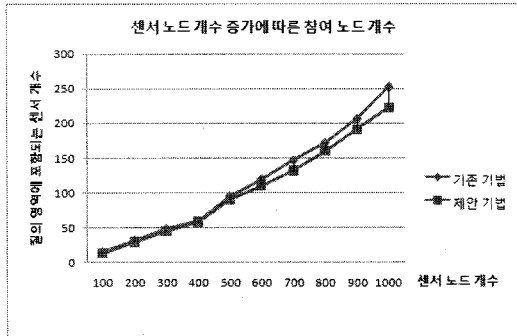
입력으로 질의 영역 r이 필요하고, 출력은 질의 영역에 포함된 센서의 위치 값이 추가된 재구성 질의 영역이 된다. 버킷 내에 저장된 각 센서 위치와 질의 영역의 교집합 연산이 필요하며 알고리즘 1에서와 동일한 관계가 발생한다. OVERLAP과 CONTAINS 관계이면 질의 영역 내에 포함되는 센서이므로 센서가 가지고 있는 위치 값을 재구성 질의 영역 rr에 추가한다. 버킷 내의 센서들에 대한 위치 값 비교가 종료되면 재구성된 영역 rr을 반환한다.

4. 성능평가

본 장에서는 제안된 질의 영역 축소 기법의 성능을 평가한다. R-Tree 구성과 영역 질의를 위해서 (주)KT의 전파 분석 도구인 CellTREK/OPT의 하부 저장 구조를 수정이용하였으며, windows xp, pentium 4, 1GB RAM, cpu 2.8 Ghz 환경하에서 시뮬레이션을 구축하였다. 센서 네트워크를 위한 환경으로는 기지국 1개, 센서 노드는 100m * 100m 영역에 랜덤으로 배치했으며 균일 분포를 이루도록 조절했다. 라우팅 트리는 한 개의 MBR 내에 최대 8개 까지의 센서 노드를 가지도록 구축했다.

첫 번째 실험은 센서 노드를 100~1000개 까지

100개씩 증가시키면서 질의 영역의 크기가 전체 영역의 크기의 약 10% 정도인 30m*30m인 상황 하에서 질의 전달에 참여하는 센서 노드의 개수를 측정한다. 다음 <그림 10>은 센서 노드 개수의 증가에 따른 질의 영역에 포함되는 센서 노드의 개수를 영역 질의가 발생하자마자 기지국에서 센서 노드들로 질의를 전파하는 기존의 기법과 기지국 내에서 질의 영역 재구성을 거친 후 센서 노드들로 질의를 전파하는 제안 기법을 비교 평가한 그래프이다.

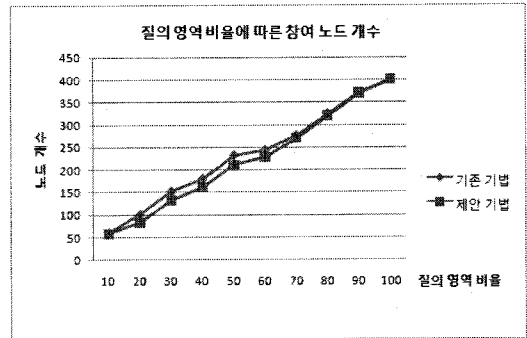


<그림 10> 센서 노드 개수 증가에 따른 참여 노드의 개수

영역 질의에 포함되어 질의 전파에 참여하는 센서 노드의 개수가 많을수록 전체 센서 네트워크의 전력 소모가 커지게 된다. 기존의 기법에 비해 제안된 기법은 기지국에서 질의 영역을 축소시켜 센서 노드들로 질의를 전파하기 때문에 참여하는 센서 개수가 감소하는 결과를 보이고 있다. 센서 네트워크 영역 내의 센서 노드들의 개수가 적은 경우 질의 영역과 MBR 겹침 현상이 적어지므로 기존의 기법과 제안 기법이 차이가 없지만 센서 노드의 개수가 많아지면 MBR 개수가 증가하면서 질의 영역과 겹침이 많이 발생하게 된다. 이러한 겹침 현상은 제안 기법인 질의 영역 재구성을 통하여 제거되므로 질의 전파에 참여하는 센서 노드가 적어지고 결과적으로 통신 회수를 감소시켜 전체 센서 네트워크의 수명을 연장시키는 효과를 가져온다.

두 번째 실험은 센서 노드 개수가 400개일 때 질의 영역을 전체 네트워크 영역의 10%에서 100%

까지 10%씩 증가시키면서 질의에 참여하는 센서 노드의 개수를 비교하였다. 그 결과는 <그림 11>에 나타나있다.



<그림 11> 전체 영역에 대한 질의 영역 비율에 따른 참여 센서 노드의 개수

영역 질의의 범위가 센서 네트워크의 전체 영역에 차지하는 비율에 따른 성능을 비교해 본 결과 질의 영역 비율이 아주 작은 경우 MBR과 겹치는 경우도 적어 기존의 기법과 차이가 나지 않았지만 20%~60% 정도에서는 제안 기법이 참여하는 노드의 수를 감소시키고 있다. 그러나, 질의 영역이 전체 센서 네트워크 영역의 70% 이상이 되면 기존의 기법과 거의 유사한 개수의 노드를 방문하게 된다. 센서 네트워크의 라우팅 트리 구성시 반드시 부모 노드를 거쳐야만 자식 노드로 질의가 전파되도록 했기 때문에 영역 질의에 포함되지 않는 부모 노드라도 질의 전파에 참여해야 하기 때문이다.

위의 두 실험을 통해 제안된 질의 영역 재구성 기법이 기존의 기법보다 질의에 참여하는 센서 노드의 수를 감소시켰음을 확인할 수 있었다. 그러나, 기존의 기법은 질의가 발생하자마자 기지국에서 센서 노드들로 전파하는 것이고, 제안된 기법은 실제 질의 영역에만 포함된 센서에만 질의를 전파할 목적으로 질의 영역을 재구성하여 센서 노드에 질의를 보내므로 기지국에서 질의 재구성하는 시간이 더 필요하다. 기지국에서 질의 영역 내에 속하는 제일 하단의 센서 노드까지 질의 전파 시간은 기존의 기법과 동일하므로 제안된 질의 영역 재구성 기법

을 적용하는 경우 기지국에서 질의 영역을 재구성하는 만큼의 시간이 더 요구된다. 이 결과로 두 번째 실험에서 질의 영역 비율이 70% 이상인 질의의 경우 제안된 질의 영역 재구성 기법을 적용하지 않고 센서 노드로 질의를 전파하는 것이 시간적인 이득이 발생한다고 추론 할 수 있다. 제안된 질의 영역 재구성 기법은 센서 노드의 개수가 많아질수록 또한 질의 영역 비율이 전체 20%~60% 사이의 경우에는 적용하는 것이 센서 네트워크의 전력 소모를 감소시키려는 노력에 부합된다고 할 수 있다.

5. 결론

본 논문은 USN 환경에서 사용자 중심의 영역 질의를 처리하기 위해 트라이 구조 쿼드 트리를 제안하였고, 이를 통해 센서 데이터 값을 검색하는 방법과 전체 센서 네트워크의 수명을 연장시킬 수 있도록 질의 영역 재구성 방법을 제안하였다. 제안된 기법은 기존의 기법이 센서 데이터 값을 가져오기 위해 센서 네트워크의 특징에 맞춰 라우팅 트리 구축과 센서 네트워크에서 발생하는 데이터 스트림에 대한 연구를 각각 수행한 것을 USN 환경에서 미들웨어의 중요성이 점점 강조됨에 따라 미들웨어 관점에서 영역 질의를 처리하는 방법을 제시하였다.

제안된 기법은 영역 질의가 수행될 때 현재 데이터 값을 요구하는 snapshot 질의나 이벤트 질의의 경우 기지국에 구축된 트라이 구조 쿼드 트리를 이용하여 질의 영역을 재구성하여 센서 네트워크로 질의를 전파한다. 질의 영역을 축소하여 재구성하는 방법은 질의에 참가하는 센서의 수를 감소시켜 전체 센서 네트워크의 수명을 연장시킬 수 있는 방법이라는 것을 실험을 증명하였다. 센서 노드의 개수가 많아질수록 질의에 참여하는 센서 노드의 수를 최대 10% 정도 감소시켰으며, 전체 센서 네트워크에 대한 질의 영역 비율이 20%~60% 일 경우에도 최대 10% 정도 감소시켰다

또한 제안된 기법은 지리적으로 가까운 센서 데이터 값들을 하나의 버킷에 저장함으로써 센서 데이터로부터 분석을 요구하는 영역 질의에는 즉각적

인 응답을 가능하게 한다. 기존의 기법에서는 센서 노드 내에 저장된 데이터들을 요구하여 분석을 행하던 것이 기지국 내의 서버에서 수행을 하므로 응답 시간의 감소를 가져올 것으로 기대할 수 있다.

향후 연구로는 지속적으로 발생하는 센서 데이터들을 버킷에 저장하는 방법을 연구해야 한다. 버킷이 가득 찬 경우의 해결 방법과 하나의 버킷에 저장되는 동일한 센서의 여러 데이터들이 공존해 있기 때문에 버킷 내의 센서 데이터들의 저장 형식에 대한 연구가 이루어져야 한다. 또한 이를 바탕으로 한 continuous 질의 처리에 대한 연구가 병행되어야 한다.

참고문헌

1. 장정아, 최정단, 장병태, "USN 기반 텔레매틱스 서비스 및 기술개발 동향," 전자통신동향분석, ETRI, 제22권, 제3호, 2007, 6, pp.1-11.
2. Akyildiz, I.F., Su, W., and Sankarasubramanian, Y., "A Survey on Sensor Networks," Communications Magazine, IEEE, 2002, 8.
3. Madden, S., Franklin, M.J., and Hellerstein, J. M., "Tiny DB: An Acquisitional Query Processing System for Sensor Network," Journal of ACM Transactions on DataBase Systems, Vol.30, No.1, 2005, pp.122-173.
4. Bonnet, P., Gehrke, J., and Seshadi, P., "Querying the Physical World," Journal of IEEE Personal Communications on Special Issue Networking Physical World, 2000, pp.10-15.
5. Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J., "Model and Issues in Data Stream Systems," In Proceedings of PODS, 2002.
6. Chandrasekaran, S., and Franklin, M. J.,

“Streaming Queries over Streaming Data,” In Proceeding of VLDB, 2002.

7. Abadi, D. J., Carney, D., Cetintemel, U., Cheniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., and Zdonik, S., “Aurora: A New Model and Architecture for Data Stream,” In Proceedings of VLDB, 2003.

8. Intanagonwiwat, C., Govindan, R., and Estrin, D., “Directed Diffusion for Wireless Sensor Networking,” In Journal of IEEE Transactions on Networking, 2003.

9. 박순영, 센서 네트워크 환경에서 실시간 영역 질의 처리를 위한 센서 데이터 관리 시스템, 박사학위논문, 인하대, 2006, 2.

10. 김대영, 김재언, 성종우, 이강우, “센서 네트워크 운영체제/미들웨어 기술동향,” 주간기술동향, IITA, 통권1221호, 2005, 11.

11. 김민수, 이용준, 박종현, “USN 미들웨어 기술 개발 동향,” 전자통신동향분석, ETRI, 제22권, 제3호, 2007, 6, pp.67-79.

12. Greenstein, B., Estrin, D., Govindan, R., Ratnasamy, S., and Shenker, S., “DIFS: A Distributed Index for Features in Sensor networks,” in Journal of Elsevier Journal of Ad Hoc Networks, 2003.

13. Finkel, R. A., and Bently, J. L., “Quadtree: A Data Structure for Retrieval on Composite Keys,” ACTA Informatica, Vol.4, No.1, 1974.

14. Li, X., Kim, Y. J., Govindan, R., and Hong, W., “Multi-dimensional Range Query in Sensor Networks,” In Proceedings of SenSys, 2003.

15. Soheili, A., Kalogeraki, V., and Gunopulos, D., “Spatial Query in Sensor Networks,” Proceedins of the 13th Annual ACM International Workshop on Geographic Information Systems, 2005,

pp.61-70.

16. 이연, 이상훈, 조속경, 이순조, 배혜영, “USN 환경에서 효율적인 공간영역질의를 위한 적응형 영역 집계 인덱스 기법,” 한국공간정보시스템학회 논문지, 제9권, 제2호, 2007., pp.93-107.

조속경

1990년 인하대학교 전자계산학과(이학사)
 1994년 인하대학교 대학원 전자계산공학과(공학석사)
 2002년 인하대학교 대학원 컴퓨터공학과(공학박사)
 2003년~2006년 인천대학교 강의전담 교수
 2006년~2007년 인하대학교 지능형 GIS 연구센터 연구원
 2007년~현재 (주)케이지아이 개발부
 관심분야 : 데이터베이스, 실시간 데이터베이스 시스템, GIS, 센서 네트워크

정미영

1996년 연세대학교 대학원 전산학과(공학석사)
 1998년~현재 (주)KT 인프라연구소 책임연구원
 관심분야 : 무선망설계/최적화 시스템, GIS

정현민

1996년 연세대학교 대학원 전자공학과(공학박사)
 1986년~1992년 한국통신 연구개발본부
 1996년~현재 (주)KT 인프라연구소 수석연구원
 관심분야 : 이동통신, 무선망설계/최적화 시스템

김종훈

1991년 인하대학교 전자계산공학과(공학사)
 1993년 인하대학교 대학원 전자계산공학과(공학석사)
 1998년 인하대학교 대학원 전자계산공학과(공학박사)
 1996년 인하대학교 전자계산공학과 전임대우
 1998년~현재 (주)케이지아이 대표이사
 관심분야 : 데이터베이스, 멀티미디어 데이터베이스 시스템, GIS, 센서 네트워크