

# 센싱 뷰를 이용한 센서 네트워크에서의 질의 처리\*

김혜정<sup>○</sup>, 홍준표, 홍성범, 김창화, 김상경, 박찬경

강릉대학교 컴퓨터공학과

{kimsam72<sup>○</sup>, newsam202, hsb7410}@nate.com {kch, skkim98, cjpark}@kangnung.ac.kr

## Query Processing of Sensor Network using Sensing View

Kangnung National University Dept. of Computer Science & Engineering

Haejung Kim<sup>○</sup>, J.P. Hong, S.B. Hong, C.H. Kim, S.K. Kim, C.J Park

### 요 약

센서 네트워크에서 질의 처리 기술은 통신 비용을 절감하고 신뢰성을 향상하고 에너지 효율성을 향상시키는데 목적을 두고 있다. 현재 에너지의 효율성을 위해서 센서 노드가 수행하는 질의를 최적화하여 전력 손실을 줄이고자 하는 연구가 활발하게 수행 중이다. 본 논문에서는 실제 뷰와 인덱싱 뷰를 이용하여 센서 네트워크에 맞는 효율적인 새로운 센싱 뷰라는 기법을 제안한다. 제안하는 질의 처리 알고리즘은 실제 뷰와 인덱싱 뷰의 개념을 바탕으로 기존의 분산 데이터베이스의 질의 처리 방법을 센서 네트워크에 적용하였다. 제안된 센싱 뷰 알고리즘은 센싱 뷰를 싱크 노드에 위치시킴으로써 센서 노드와의 통신 횟수를 최대한 줄임으로서 통신 비용과 에너지의 소모량을 줄일 수 있었다. 성능은 시뮬레이션을 통해서 평균적으로 약 30%정도의 성능 향상 효과를 보였다.

## 1. 서 론

센서 네트워크는 최근 발전하고 있는 임베디드 시스템과 유비쿼터스 컴퓨팅 기술을 반영하기 때문에 최근 이와 관련된 폭 넓은 연구와 개발이 진행되고 있으며, 관심 지역에 분산되어 있는 많은 수의 센서 노드들로부터 감지된 상황인 데이터를 무선 네트워크를 통하여 데이터를 주고받고 있다[1].

센서 네트워크에서의 센서 노드들은 크기 및 가격 면에서 경쟁력을 위하여 컴퓨팅 능력, 메인 메모리 통신 대역폭 그리고 배터리 용량에서 제한된 성능을 갖고 있다. 특히 배터리 용량은 센서의 동작과 중지애 관련이 있으므로 매우 중요한 제한 요건이라 할 수 있다.[2] 또한 최근 센서들은 여러 현상들을 감지할 뿐 아니라, 획득한 데이터에서 유용한 정보를 분석하고, 필요한 정보는 자체 저장소에 관리할 수 있고, 다른 센서나 네트워크 내부에서 주어지는 질의를 처리할 수 있다[3].

최근 센서 네트워크를 활용하기 위한 다양한 연구가 이루어지고 있으며, 관심 지역에 위치한 센서로부터 원하는 데이터를 가져오기 위한 분산 질의 처리도 중요한 연구 분야가 되고 있다. 효율적인 분산 질의 처리에 대한 연구는 데이터베이스 분야에서 활발하게 이루어지고 있다. 그러나 센서 네트워크에서의 질의 처리 방법에 대

한 연구는 센서 노드들의 제한된 에너지와 통신 능력 등의 특성을 고려한 새로운 방법이 필요하다. 특히 질의 처리를 통해 원하는 결과를 얻을 때까지 소모되는 에너지를 줄이는 것이 우선적으로 고려되어야 한다. 에너지 소모는 센서 노드간의 통신 횟수와 밀접한 관련성을 가지고 있다.

본 논문에서는 센서 네트워크 환경의 특성을 고려하여 실제 뷰와 인덱싱 뷰를 이용하여 센서 네트워크에 맞는 효율적인 새로운 센싱 뷰의 기법을 제안한다. 제안된 센싱 뷰 기법은 임계값을 사용하여 센서 노드간의 통신량과 데이터 처리량을 최소화함으로써 에너지 소비를 줄인다. 그리고 시뮬레이션을 통하여 특정 센서 네트워크 환경에서 에너지가 효율적임을 확인한다.

본 논문의 구성은 1장에서는 서론, 2장에서는 센서 네트워크의 관련 연구, 3장에서는 센싱 뷰 시스템, 4장에서는 센싱 뷰 시스템에서의 질의 처리 절차 및 방식과 제안하는 방법의 에너지 효율성을 시뮬레이션을 통해 확인하고, 마지막으로 5장에서는 결론과 향후 연구 내용을 기술한다.

## 2. 관련 연구

### 2.1 센서 네트워크에서의 질의 처리

센서 네트워크에서의 질의 처리를 위한 데이터베이스

\* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IIITA-2005-C1090-0501-0010)

시스템의 주요 목적은 센서 네트워크 환경에 흩어져 있는 센싱 정보들에 대하여 센서 노드들의 제한된 자원의 저성능의 CPU, 제한된 메모리 용량, 낮은 통신 대역폭을 고려하면서 효율적으로 사용자의 질의를 만족시켜 주는 것이다. 다시 말하면, 각 센서 노드들에서의 에너지 소모율을 최소화시키면서 질의에 대한 정확성 및 신속성을 최대화 시킬 수 있는 질의 처리기를 만드는 것이다.

센서 네트워크에서의 질의 처리에서의 질의 최적화 방법은 하나의 상황에 여러 노드가 동시에 감지하여 중복된 데이터를 발생시킬 수 있으며, 특히 인접한 노드들은 같은 데이터를 반복적으로 전송할 수 있다. 이러한 문제가 발생하게 되면 센서 네트워크내의 데이터 전송량이 증가되어 에너지 소모가 많아질 수 밖에 없게 된다. 이러한 문제를 효과적으로 해결하기 위해서는 각 노드들이 데이터를 병합하여 중복된 데이터를 배제하여 구분되는 상황만을 상위 노드로 전송하여야 한다.

## 2.2 실체 뷰와 인덱싱 뷰의 관리 기법

데이터웨어하우스는 사용자의 의사 결정에 필요한 정보를 제공하여 효율적인 데이터 마이닝을 통하여 질의 처리를 할 수 있게 한다. 데이터웨어하우스에서 중요하게 다루어지는 실체 뷰(materialized view)는 뷰의 연산 결과를 미리 저장된 형태로서 가상의 테이블인 뷰와 달리 실제 값을 저장해 두고 있어 훨씬 빠르게 접근할 수 있어 질의 처리의 성능을 향상시킬 수 있으며 특히 분산 데이터베이스에서 질의를 실행하고 그 실행 결과를 응답할 때 유용하게 쓰여진다. 그러나 데이터웨어하우스 환경에서 소스 테이블의 변경 시 실체 뷰에 바로 적용될 수가 없다는 문제점이 있으며 이는 소스테이블과 실체 뷰의 일관성 유지 문제가 생긴다. 데이터웨어하우스 응용 프로그램에서 자주 발생하는 복잡한 보고 시나리오의 경우, 질의가 수행되는 동안 데이터베이스 서버에 대한 요구 사항이 높다. 이러한 질의가 뷰에 액세스할 때 데이터베이스는 기본 테이블 데이터로부터 완전한 질의 결과 집합을 작성하는 데 필요한 논리를 동적으로 병합하기 때문에 성능이 감소될 수 있다. 이러한 작업의 오버헤드는 특히 대형 데이터 집계나 다중 테이블 조인과 같은 대규모 행을 처리하는 복잡한 작업이 포함된 뷰의 경우 더욱 커질 수 있으며, 결과 집합은 데이터베이스에 영구적으로 저장되지 않기 때문에 뷰에 대한 이후의 액세스 시도에서도 질의가 실행될 때마다 결과 집합을 작성하는 비용이 계속해서 발생할 것이다. 인덱싱 뷰(indexing View)를 이용할 경우 기존 질의는 기록될 필

요 없이 인덱싱 뷰로부터 데이터를 검색하여 효율성을 향상시킬 수 있다. 또한, 특정 유형의 뷰일 경우 이러한 방식은 성능을 매우 크게 향상시킬 수 있다.

## 3. 센싱 뷰 시스템

### 3.1 센싱 뷰 시스템의 구조

일반적인 센서 네트워크의 구조에 각 싱크 노드에 DBMS를 탑재한 구조의 센싱 뷰(sensing view)라고 하는 기법을 제안한다. 싱크 노드는 노드들의 대표 노드로서 임의로 선정될 수 있지만, 전력량이나 라우팅 기법에 의해 변경될 수도 있다. 센싱 뷰 시스템의 기본 구조는 아래 그림 1과 같다.

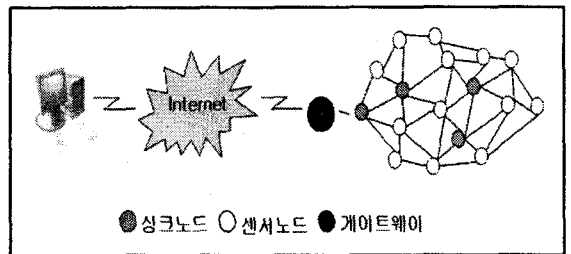


그림 1. 센서 네트워크 구성도

임의의 노드들로부터 헤더 노드가 선정되고 각 헤더 노드는 헤더 노드들끼리 통신이 가능하고, 인터넷을 통해 최종 사용자까지 연결된다. 또한 노드들의 연결 부분은 전력량이나 선출된 헤더 노드에 따라 달라질 수도 있다.

기존의 센서 네트워크 구성도와 특징적인 차이점은 미들웨어 구성에 있어서 DBMS의 기능이 확장되어진 것이다. 또한 고정적인 싱크 노드 구성과는 달리 특정 노드의 전력 소모를 적게함으로써 결과적으로 더 오랜 시간 센서 네트워크를 지속시킬 수 있고, 사람의 간섭 없이 오랜 기간 네트워크를 유지해야 하는 경우에도 유용하다.

### 3.2 싱크 노드 선정

센서 네트워크에서 인접한 노드간의 유사한 정보의 중복 전달로 인한 에너지 낭비를 줄이기 위한 "데이터 모음(data aggregation)"이 필요하다는 특성을 고려할 때 클러스터링에 기반한 계층적 라우팅 기법이 많은 장점을 가진다. 즉, 로컬 클러스터를 형성함으로써 인접한 지역에서 발생한 사건에 대한 유사한 정보를 클러스터 헤드

로 전송하고 클러스터 헤드가 데이터 모음을 수행하여 보다 에너지 효율적인 라우팅을 가능하게 하며, 요청된 질의에 대한 클러스터 헤드에 의한 전달로 비효율적인 질의의 플래딩을 막을 수 있다. 클러스터링 기반의 계층적 라우팅 기법에서는 다음과 같은 가정을 전제로 한다.

지속적인 전력이 공급되는 싱크 노드가 모든 센서 노드에 직접 데이터를 전달할 수 있고, 수집 데이터의 저장 및 데이터 가공 등의 에너지 집약적인 기능을 수행한다. 센서 필드내의 센서 노드들은 같은 특성을 가지며, 초기에 같은 에너지가 부여된다.

### 3.3 센서 노드의 구조

센서 노드들은 이들웨어 부분에 해당하는 각각의 DBMS를 구성하고 있다.

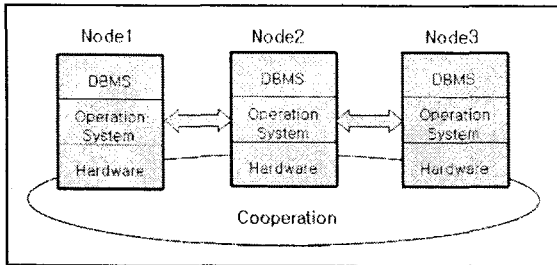


그림 2. 센서 노드 구조

그림 2와 같이 각각의 노드는 기본적인 하드웨어를 가지고 있으며, 운영체제와 DBMS를 가지고 있다. 어느 노드라도 싱크 노드에 선출될 수 있기 때문에 모든 노드는 동일한 형태의 구조를 가지고 있어야 저전력 상태가 발생했을 때, 자연스럽게 다른 노드가 싱크 노드로 선출되기 용이하다. DBMS를 각 노드에 탑재하면 싱크 노드 한 개에만 탑재할 경우보다 질의 처리 속도가 증가하고, DBMS간의 유기적인 협동이 가능하다. 또한, 센서 노드들은 컴퓨팅 능력의 제약으로 버퍼가 충분하지 못하다. 그래서 버퍼가 다 차기 전에 수집된 정보들을 게이트웨이로 전송해야 한다. 만약 오버플로우 상황이 발생한다면 정보들의 우선 순위를 효율적으로 관리함으로써 고품질 데이터의 게이트웨이에게 전달하는 역할을 DBMS가 수행하고, 주어진 질의를 분석하고 그 질의에 해당하는 센싱 뷰를 생성한다. 센싱 뷰는 클러스터된 고유 인덱스를 포함한 뷰를 말하며, 또한 센싱 뷰는 한 번 이상 비슷한 질의의 작업을 수행할 때 처리 속도와 DB의 액세스 속도를 현저하게 줄여줄 뿐 만 아니라, 그림 2와 같은 구조에서는 네트워크 상에 데이터 흐름의 양을 줄여

주는 역할도 한다. 이것으로 인해 전력의 소모와 CPU의 오버헤드를 줄일 수 있다.

### 4. 질의 처리 절차 및 방식

본 논문에서는 다음 그림 3과 같은 DBMS의 구조를 제안한다. 각각의 질의에 대해 센싱 뷰를 생성하고 그 센싱 뷰의 비용을 산출하여 각각의 센싱 뷰에 대한 비용 평가를 하고, 임계값을 설정하여 질의가 임계값을 벗어났을 때는 무조건 데이터 통신이 일어나도록 하고, 그 이외의 경우에는 일정 시간동안 랜덤하게 통신이 일어나도록 설계하였다.

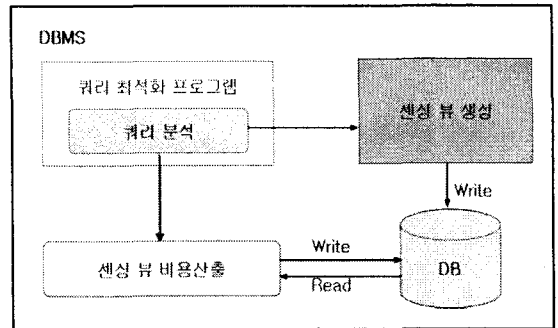


그림 3. 센싱 뷰를 적용한 DBMS 구성

질의 최적화 프로그램은 실행될 질의를 분석하여 센싱 뷰를 생성하고 비용을 산출한다. 또한 산출한 비용을 DB에 저장하게 된다.

#### 4.1 질의 최적화 프로그램

질의 최적화 프로그램은 주어진 질의를 분석하여 비용을 산출하고 센싱 뷰를 생성하는 역할을 한다. 그렇지만, 모든 질의에 대해 센싱 뷰를 생성하는 것은 아니다.

일반적으로 질의 빈도와는 상관없이 답변에 많은 시간이 소용되고 신속한 답변이 무엇보다 중요한 질의라면 센싱 뷰를 사용하는 것이 좋을 것이다.

즉, 전역적으로 연계해서 질의를 하는 경우에는 많은 시간이 소요되는 경우가 발생한다. 그런 경우에는 그 비용을 분석하여 센싱 뷰를 생성한다. 하지만, 본 논문에서는 센서 네트워크라는 특수한 경우이다. 이런 경우에는 전력의 소모량을 고려하지 않을 수 없다. 그래서, 추가적으로 사용 빈도 또한 센싱 뷰를 생성하는데 고려되어야 할 요소이다. 센싱 뷰를 생성할 때는 질의 비용을 계산한다. 질의 비용을 계산하는 방법은 아래와 같다.

- ① 전역 질의 : 2, 로컬 질의 : 0
- ② 빈도수 ≤ 10회 : 0, 10회 < 빈도수 : 1
- ③ 실행시간 ≤ 1초 : 0, 1초 < 실행시간 : 1

즉, 총합이 2이상인 경우에는 질의 최적화 프로그램에서 센싱 뷰를 생성한다.

예를 들어, 만약 빈도수가 10회 이상이고, 로컬 질의 이면서 실행 시간이 1초 미만일 때, 질의를 많이 실행하여 실행 시간이 증가될 경우에는 센싱 뷰가 생성될 수 있으며, 실행 속도가 빨라지는 효과와 시스템 오버헤드를 줄일 수 있다. 센싱 뷰가 실행될 때 센서 노드가 사용한 CPU 통신량의 평균을 Query\_Cost로 설정한다. 또한, Sending\_Point 값은 확장 질의를 통해 설정할 수 있다.

#### 4.2 센싱 뷰

본 논문에서 제안하고 설계된 모든 센싱 뷰 구조에서는 Sending\_Point와 Query\_Cost가 존재한다. 이 두 가지 속성의 값은 각각 임계값과 질의 비용이다. 즉, 비슷한 내용의 질의에 대한 각각의 센싱 뷰가 생성될 수 있다.

질의는 사용자가 작성하는 것이 일반적이기 때문에 각각의 질의에 대한 최적화가 되어 있지 않을 경우가 있다. 그런 경우에는 질의 비용이 커지게 된다. 결과적으로 최적화된 질의보다 전력 소모가 많고 시스템 부하가 커진다.

앞에서 언급했듯이 Sending\_Point는 임계값을 나타낸다. 데이터 통신을 줄이기 위해 사용하고, 임계값에 도달할 때까지는 통신을 하지 않고, 임계값을 초과할 때만 데이터 통신이 이루어진다.

#### 4.3 센싱 뷰를 위한 확장 질의

본 논문에서 제안한 센싱 뷰를 위해서는 기존의 SQL 문을 그대로 사용하여 질의를 처리하는 데는 문제가 있다. 따라서 센싱 뷰에 추가된 내용과 기능을 수행하기 위해서는 확장된 질의의 정의가 필요하다. 기본적으로 SQL 질의를 기반으로 확장된 질의를 정의한다.

##### <센싱 뷰 생성>

```
Create sensing #sensing_name
AS Select select_list from_clause
```

- sensing\_name : 센싱 뷰 이름
- select\_list : table의 속성(rowset)
- from\_clause : 센싱 뷰로 만들어지는 table

##### <센싱 뷰 선택 및 추출>

```
Select select_list | * From #sensing_name
[where_clause] [order_by_clause]
```

- select\_list : table의 rowset
- sensing\_name : 센싱 뷰의 이름
- where\_clause : 조건문을 통해서 조건에 맞는 내용을 추출을 위한 조건문
- order\_by\_clause : 조건에 맞는 레코드의 정렬을 위한 순서를 만드는 조건문
- select\_list는 전체를 선택하거나 일부를 선택하여 참조할 table에서 불필요한 속성을 줄여서 센싱 뷰에서 필요한 데이터만 얻을 수 있으며, order\_by\_clause는 지정한 속성의 오름차순이나 내림차순으로 원하는 값을 쉽게 볼 수 있도록 정리해 준다.

##### <센싱 뷰의 수정>

```
Alter #sensing_name ADD element_clause
Alter #sensing_name MODIFY element_clause
Alter #sensing_name DROP element_clause
```

- sensing\_name : 센싱 뷰의 이름
- element\_clause : 튜플의 이름과 속성

##### <시간의 개념>

```
Time during_time
```

- during\_time : 일정 시간 후 이벤트나 일정 시간동안의 데이터 수집, 산출이 가능하도록 한 시간의 개념

##### <sending\_point>

```
Sending_Point number
```

- number : 임계값의 수치

<query\_cost>

Query\_Cost number

number : CPU 통신량의 평균

이렇게 새로 도입되는 센싱 뷰의 컨트롤을 위해서 센싱 뷰의 생성과 삭제, 수정이 가능하도록 해주는 부분을 정의해 주었다. 그리고 질의문에 새로 도입된 시간의 개념과 sending\_point라는 임계값과 query\_cost라는 CPU와의 통신량의 평균값을 이용하는 부분이 추가되었다.

기존 질의에서 시간에 따른 이벤트나 분류를 하지 못했지만 시간 개념을 도입해서 보다 다양한 방식의 질의를 처리할 수 있다. 그리고 query\_cost와 sending\_point로 인해서 센싱 뷰의 업데이트나 생성 시기를 정해서 보다 효율적으로 센싱 뷰가 구축되고 사용되어질 수 있도록 하였다.

제한된 센싱 뷰의 기법의 성능을 확인하기 위하여 간단한 농작물 관리를 하는 시뮬레이션 프로그램을 실행시켜 보았다. 프로그램은 운영 체제는 LINUX, 작성 환경은 P4 2.0G, 언어는 C를 사용하였다. 프로그램은 센서 노드에서 측정된 데이터를 싱크 노드에 보고하는 데이터 통신량을 비교하여 보았다. 시작 온도는 20도이고, 시간은 24시간(1시간 단위 증가), 온도는 임의적으로 ±5도가 되도록 설계하였다. 또한 15도 미만일 때는 히터를 작동하고, 25도 초과일 때는 쿨러를 작동하도록 하였다. 실험 결과 평균적으로 약 30% 정도의 성능 향상 효과를 보였다.

## 5. 결론

본 논문에서는 센서 네트워크 환경의 특성을 고려하여 실제 뷰와 인덱싱 뷰를 이용하여 센서 네트워크에 맞는 효율적인 새로운 센싱 뷰의 기법을 제안하였다. 제안된 센싱 뷰 알고리즘은 센싱 뷰를 싱크 노드에 위치시킴으로서 센서 노드와의 통신 횟수와 시스템 오버헤드를 최대한 줄임으로서 통신 비용과 에너지의 소모량을 줄일 수 있다. 시뮬레이션 결과는 약 30% 정도 향상 되었지만, 센서 노드와 싱크 노드와의 통신의 예만 적용해 보았다. 향후 연구 방향은 센서 네트워크에서의 모든 노드와의 관계도 더 연구해야 할 과제일 것이다.

## 참고문헌

[1] I.F. Akyildiz, S. WOHian, Y.Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks." IEEE

Communications Magazine. Vol. 40, No. 8, pp. 102-114, 2002.

- [2] 김대영, "센서 네트워크." INFOTHE, 2004. 7. 9.
- [3] Samuel Madden, Michael J. Franklin, and Joseph M. Hellerstein Wei Hong. "TAG: a Tiny Agregation Service for Ad-Hoc Sensor Networks," ACM SIGOPSI
- [4] P.Bonnet, j. Gehrke, and P. Seshadir, "Querying the Physical world," Personal Communications, IEEE, 7(5):10-15, 2000.
- [5] M. Tamer ozsu, "Principles of distributed database systems (second edition)," Prentice Hall, 2003.
- [6] 강홍민, 장재우 "공간 네트워크 데이터베이스를 위한 저장 및 색인 구조의 설계" 한국정보과학회 가을 학술발표논문집 제31권 제2호 pp133-136, 2004
- [7] S.Shekhar et al. "Spatial Databases Accomplishments and Research Needs," IEEE Tran. on Knowledge and Data Engineering, Vol. 11, No. 1, pp45-55, 1999
- [8] Kay Romer, Oliver Kasten, Friedemann Mattern, "Middleware Challenges for Wireless Sensor Networks," Mobile Computing and Communications Review, Vol.6, No.4, 2002.
- [9] Jason Hill, Robert Szewczyk, Alex Woo, Seth Hollar, David Culler, Kristofer Pister, "System Architecture Directions for Networked Sensors," International Conference on Architectural Support for Programming Languages and Operating Systems, 2000. 11.
- [10] A. Murphy and W. Heinzelman, "MiLAN: Middleware Linking Applications and Networks," TR-795, University of Rochester, Computer Science, Nov. 2002
- [11] C.Shen, C. Srisathapornphat, C. Jaikeo, "Sensor Information Networking Architecture and Applications," IEEE Personal Communications, Vol.8, No.4, Aug. 2001, pp.52-59.
- [12] M.Angels Moncusi and Alex Arenas, Jesus Labarta, "A Modified Dual-Priority Scheduling Algorithm for hard real-time systems to improve energy savings." Compilers and Operating System for low power, Kluwer Academic Publishers. pp.17-36, Boston, USA, 2003