

해양 센서 네트워크 아키텍처 중심의 질의 최적화를 위한 데이터 병합 기법^o

김해정^o, 지경복, 김창화, 김상경, 박찬정
강릉대학교 컴퓨터공학과

{kimsam72^o, jkb23}@nate.com, {kch, skkim98, cjpark}@kangnung.ac.kr

Data Aggregation for Query Optimization Based on Ocean Sensor Network Architecture

Haejung Kim^o, Kyoungbok Ji, Changhwa Kim, Sangkyung Kim, Chanjung Park
Department of Computer Science & Engineering, Kangnung National University

요 약

최근 센서 네트워크에서 에너지 효율성을 위한 다양한 연구가 진행 중이다. 특히 센서 노드의 자전력을 위해서는 센서 네트워크에서 전송되는 데이터의 횟수나 전송량을 최소한으로 줄이면서 효율적이면서 신뢰성을 가지는 질의에 대한 결과를 얻을 수 있어야 한다. 본 연구에서는 해양 센서 네트워크 상에서 데이터의 전송량을 줄일 수 있는 SDMTree(Sensing Data Management Tree)를 제안한다. 제안된 SDMTree는 질의 최적화를 위해 질의 처리기 구성 요소로 도입 가능하다. 해양 센서 네트워크에서 in-network 각 4 레벨에서 하위 노드로부터 받은 데이터를 병합, 관리하기 위한 방법으로 데이터를 속성별로 구분하여 중복된 데이터를 제거하여 트리형태로 구성되기 때문에 질의에 대한 응답에 해당하는 데이터 검색시 정확하고 신속하게 처리할 수 있으며, 트리 구성 또한 중복 데이터 및 중복 영역을 배제하여 구성되므로, 상위 노드가 하위 노드로부터 센싱 데이터를 수집하여 저장하기 위한 에너지와 상위 노드에서 하위 노드로 질의를 전송시 질의에 해당하는 특정 영역에만 질의를 전송할 수 있기 때문에 데이터 저장 및 통신에 소모되는 불필요한 에너지를 최대한 줄일 수 있다.

1. 서 론

센서 네트워크는 유비쿼터스 컴퓨팅의 급속한 발전에 따라 최근 이와 관련된 많은 연구와 개발이 진행되고 있으며, 활용 분야도 더욱 다양해지고 있다. 예를 들어, 사탕이 살 수 없거나 접근이 불가능한 야생 동식물의 생태계 조사, 기상 관측, 재난 지역 탐사 등 관심 지역에 분산되어 있는 많은 센서로부터 상황이나 발생하는 현상의 데이터를 무선 네트워크를 통하여 얻고 있다[1].

최근 센서 네트워크를 이용하여 관심 지역에 위치한 센서 노드로부터 원하는 데이터를 가져오기 위한 분산 질의 처리(distributed query processing)도 중요한 연구 분야로 대두되고 있다. 그러나 센서 네트워크에서는 분산 질의 처리를 담당하는 센서 노드는 크기 및 가격 면에서 경쟁력을 위하여 컴퓨팅 능력, 메인 메모리 통신 대역폭 그리고 배터리 용량에서 제한된 성능을 갖고 있다. 특히 배터리 용량은 센서의 동작과 중지 및 센서 네트워크의 생존기간과 밀접한 관련이 있으므로 매우 중요한 요건이다[2]. 그러므로 센서 네트워크에서의 질의 처

리 방법에 대한 연구는 센서 노드의 제한된 에너지와 통신 능력의 특성을 고려한 새로운 방법이 필요하며, 특히 질의 처리를 통해 원하는 결과를 얻을 때까지 소모되는 에너지를 줄이는 것이 우선적으로 고려되어야 하고 에너지 소모는 센서 노드간의 통신 횟수와 밀접한 관련성을 가지고 있다.

센서 네트워크에서 센서 노드의 배치는 매우 조밀하게 하여 같은 지역을 여러 센서 노드가 중복해서 감시하는 특성을 가지고 있다. 이는 하나의 이벤트 또는 속성에 대해 여러 노드가 동시에 감지하여 중복된 데이터가 발생할 수 있고 특히 인접한 노드는 같은 데이터를 반복적으로 전송함으로써 불필요한 에너지 소모를 하게 된다 [3]. 그러므로 이러한 불필요한 데이터 통신으로 인한 에너지 소모를 줄이기 위한 방법이 필요하다. 이를 해결하기 위해서는 센서 네트워크에서는 네트워크 내부에서 데이터를 병합하여 중복된 데이터를 배제하여 구분되는 데이터만을 상위 노드로 전송할 수 있는 데이터 병합 방법 및 질의에 대한 응답을 신속하고 정확하게 하기 위한 질의 최적화 방법이 필요하다.

본 연구에서는 데이터 병합 방법 및 질의 최적화 방법은 발생하는 센싱 데이터와 응용에 따라 다르며 센서 네트워크의 환경적 요인과 센서 노드의 특성을 고려하여 그룹이나 클러스터를 구성하는 내부에서 발생하는 데이터를 수집하여 하위 노드에서 상위 노드로 전송되는 경

^o 본 연구는 정보통신부 및 정보통신연구진흥원의 2007년도 대학 IT연구센터 지원사업의 연구결과로 수행되었음(ITA-2007-C1090-0701-0044)

로 상에서 통신비용 절감 및 에너지 효율성을 고려하여 데이터 병합 방법 및 질의 최적화 문제를 해결하기 위해 데이터 병합 및 질의 최적화 트리를 구성하는 방법으로 SDMTree(Sensing Data Management Tree)를 제안한다.

본 연구의 구성은 2장에서는 센서 네트워크에서 데이터 병합 및 질의 최적화에 대해 설명하고, 3장에서는 질의 처리 시스템과 SDMTree의 구성 및 관리 방법을 기술하고, 4장에서는 SDMTree의 알고리즘을 기술하며, 마지막으로 5장에서는 결론 및 향후 연구 과제를 기술한다.

2. 관련 연구

2.1 데이터 병합

센서 네트워크는 대규모 숫자의 노드가 광범위한 지역에 감시 임무를 위해 무작위로 배포되고 센서를 통해 센싱된 데이터는 네트워크 내부에서 데이터 처리를 통해 하위 노드에서 상위 노드로 전송되어 베이스스테이션의 사용자에게 전달된다. 센서 노드는 배터리로 동작하는 저전력의 저가의 시스템이므로 에너지 사용에 많은 제약을 받게 된다. 즉 센서 네트워크의 센서 노드들은 일회용 소모품의 성격을 가진다. 그러므로 에너지 사용을 효율적으로 하여 네트워크의 생존 시간(lifetime)을 최대화하는 것은 매우 중요한 요소이다[4]. 또한 센서 네트워크에서 센서 노드의 배치는 사람에 의해 배포되기 보다는 비행기 또는 다른 운반수단을 이용하여 무작위로 배치되기 때문에 노드의 밀집도가 높게 되어 같은 지역을 여러 노드가 중복하여 센싱하게 되어 중복된 데이터를 발생시킨다. 이러한 문제점을 해결하기 위해 데이터 병합에 대한 기술은 매우 중요하며 센서 네트워크에서 통신 에너지를 보존하기에 효과적인 기술이며, 센서로부터 수집된 가공되지 않은 데이터의 초기의 중복 때문에 in-network의 데이터 병합은 에너지 소비를 감소시키고 네트워크의 생존기간을 확장할 수 있기 때문에 센서 네트워크가 데이터 병합을 지원하는 것은 매우 중요하다. 또한 데이터 병합에 대한 연구 초점은 다른 노드로부터 어떻게 데이터를 병합할 것인가, 데이터를 병합을 쉽게 하기 위한 구조를 어떻게 할 것인가, 그 구조를 유지하기 위한 방법, 그리고 관련이 있는 데이터를 고려해서 데이터를 효과적으로 압축하고 병합에 관한 방법이다[5].

2.2 질의 최적화

센서 네트워크에서의 질의 최적화의 주요 목적은 센서 네트워크 환경에 흠어져있는 센싱 데이터에 대하여 센서 노드들의 제한된 자원의 저성능의 CPU, 제한된 메모리 용량, 낮은 통신 대역폭을 고려하면서 효율적으로 사용자가 원하는 질의에 대한 정확하고 신속하게 질의에 대한 응답을 만족시켜 주는 것이다.

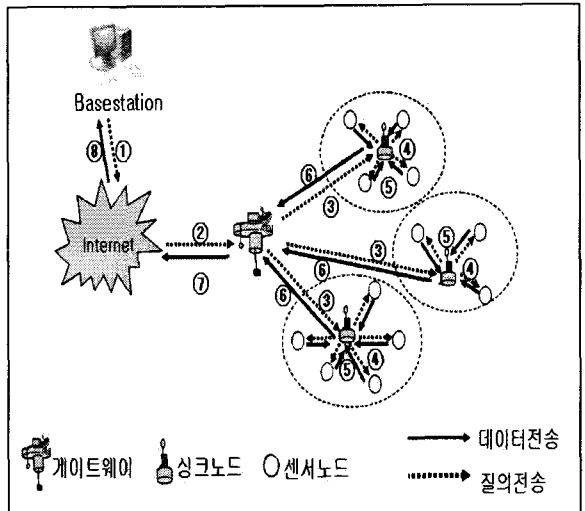
다시 말하면, 센서 네트워크에서의 질의 처리 기술은 통신비용을 절감하고 신뢰성과 에너지 효율성을 향상시키는데 목적을 두고 있으며, 현재 에너지의 효율성을 위해서 센서 노드가 수행하는 질의를 최적화하여 전력 손

실을 최대한 줄일 수 있는 질의 처리기를 만드는 것이다.

3. 질의 처리 시스템

3.1 질의 처리 과정

센서 네트워크의 질의 처리 시스템의 구조는[그림 1]과 같으며, 그림에서 보듯이 구성 요소로는 최상위에는 베이스스테이션이 있고, 그 아래에 게이트웨이가 위치하고, 그 하위 노드에는 싱크 노드가 있으며, 싱크 노드의 하위 노드에는 센서 노드로 구성되어, 4레벨로 데이터 전송을 하게 된다. 또한 각각의 구성 요소들은 질의 최적화를 위한 질의 분석과 처리 그리고 질의 분배 및 수집을 각각의 질의 처리기를 가진다.



[그림 1] 질의 처리 시스템 구조

[그림 1]의 질의 처리 시스템의 구조에서 질의 처리 과정을 좀 더 자세히 살펴보면, ①번에서 ④번까지는 질의 전송 과정을 순서대로 표현한 것이고, ⑤번에서 ⑧번까지는 사용자가 요구하는 질의에 응답하기 위해 해당하는 센싱 데이터의 전송 과정을 순서대로 나타낸 것이며, 각 단계에서 질의 처리 시스템 내의 구성 요소들은 질의 최적화기를 통해서 아래와 같은 기능을 하게 된다.

[단계 ①] 질의 처리의 첫 번째 단계로서 사용자 또는 응용프로그램에서 원하는 특정 데이터나 센서 네트워크의 센싱 데이터의 요청이 있을 때 받은 질의를 베이스스테이션에서 분석 처리하는 단계이다.

[단계 ②] 인터넷과 무선망을 통하여 베이스스테이션에서 게이트웨이로 질의를 전송하는 단계이다.

[단계 ③] 게이트웨이는 전송받은 질의를 다시 한 번 분석 처리하여 싱크 노드에게 질의를 전송하는 단계이다.

[단계 ④] 싱크 노드는 전송받은 질의를 다시 분석 처리하여 자신의 클러스터에 해당하는 모든 센서 노드에게

질의 응답에 해당하는 센싱 데이터를 요청하는 단계이다.

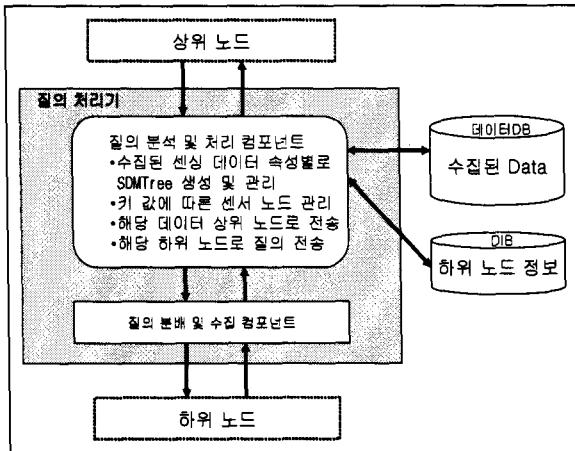
[단계 ⑤] 센서 노드에서 수집한 데이터 중에 상위 노드인 싱크 노드의 질의에 해당하는 센싱 데이터만 전송하는 단계이다.

[단계 ⑥] 싱크 노드는 전송받은 데이터를 게이트웨이로 질의 결과를 전송하는 단계이다.

[단계 ⑦] 게이트웨이에서 베이스스테이션으로 질의 결과를 전송하기 위해 인터넷 무선 통신망을 거치는 단계이다.

[단계 ⑧] 질의 처리의 마지막 단계로서 게이트웨이에서 전송된 질의 결과가 인터넷 무선 통신망을 거쳐 베이스스테이션으로 전송되는 단계로 사용자 또는 응용프로그램에서 원하는 특정한 데이터나 질의에 응답에 대한 최종적인 결과를 전송하게 된다.

센서 네트워크에서 질의 처리를 위한 질의 최적화 시스템을 만들기 위해서는 특정 지역이나 특정 클러스터에서만 데이터를 받을 수 있어야 하고, 특히 병합 함수가 지원되어야 한다. 이러한 기능의 궁극적인 목표는 에너지 효율성을 높이고 신속하고 정확한 데이터를 얻기 위한 것이다. 그러기 위해서는 센싱 데이터의 중복된 값을 배제하고 병합하는 방법과 병합을 쉽게 하기 위한 구조 및 유지하는 방법이 필요하다. 본 연구에서는 [그림 2]와 같이 각각의 구성 요소에서 데이터의 중복성을 배제하고 데이터의 검색시 정확성과 신속성을 높이기 위해 질의 처리기를 SDMTree 형태로 구성하고 유지된다.



[그림 2] 질의 처리기 구성 요소

3.2 SDMTree 구성

질의 처리기에서 SDMTree 구성은 가정 먼저 싱크 노드에서 이루어지게 된다. 질의 처리 과정에서 싱크 노드는 자신의 해당 센서 노드에게 질의를 전송하며 센서 노드는 해당 싱크 노드로부터 받은 질의를 분석하여 환경으로부터 센서를 통해 센싱된 데이터를 해당 싱크 노드에서 데이터를 전송하게 된다. 각각의 싱크 노드는 센서

노드로부터 전송받은 데이터를 수집하여 데이터 속성별로 구성하고 중복된 값을 제거한 후 속성별로 SDMTree로 구성하게 된다.

싱크 노드에서의 SDMTree 구성은 다음과 같은 방법으로 구성하게 된다. 먼저 싱크 노드가 자신의 해당 센서 노드로부터 수집된 데이터를 병합(Aggregation)하게 되며, 병합된 데이터 중에 중복 데이터를 제거하기 위해서 속성별로 Distinct 함수를 사용하여 중복된 데이터를 제거하게 되고, SDMTree의 키 값은 센싱 데이터의 속성값이 되며, 루트는 각 속성의 데이터 중에 Median 함수를 사용하여 선택하게 된다. 루트를 기준으로 좌측 서브 트리는 루트보다 작은 속성의 값이 위치하게 되고, 우측 서브 트리는 루트보다 큰 속성의 값이 위치하게 된다. 그리고 트리의 각 키 값은 중복된 데이터를 가진 노드의 ID를 테이블 형태로 저장하고 관리하게 된다.

싱크 노드의 상위 계층인 게이트웨이는 각각의 싱크 노드로부터 트리의 정보를 포함한 모든 데이터를 받은 후 다시 한 번 중복된 데이터를 제거한 후 속성별로 트리를 재구성하고 관리하게 되며, 베이스스테이션도 게이트웨이와 같은 방법으로 게이트웨이에서 받은 데이터를 통해서 SDMTree를 형성하게 된다.

4. SDMTree 구현

4.1 SDMTree 구조

SDMTree는 병합된 데이터를 속성별로 구분하여 병합한 후 중복된 데이터를 제거하고 병합 데이터의 속성의 수와가 트리 수가 같게 된다. 이렇게 속성별로 트리를 생성하는 이유는 질의에 대한 응답으로 Join연산과 같은 연산을 필요할 때 사용하기 위해서이며 트리의 키 값 또한 데이터의 속성의 값이 된다. 키 값을 데이터 속성값으로 설정한 이유는 센서 네트워크에서 하나의 클러스터에서 측정되는 데이터는 거의 유사하다고 추정하기 때문이다.

예를 들어, 만약 센서 노드로부터 싱크 노드가 받은 데이터가 노드 ID, 온도, 조도이고 아래와 같다.

[표 1] 싱크 노드에서의 수집 데이터

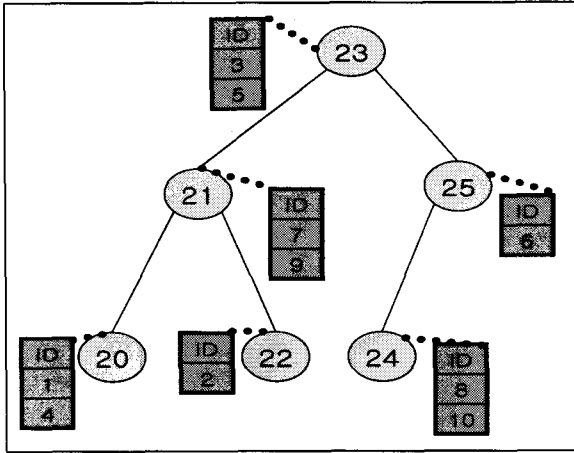
노드 ID	온도	조도
1	20	48
2	22	50
3	23	48
4	20	51
5	23	47
6	25	49
7	21	51
8	24	53
9	21	54
10	24	47

먼저, 데이터가 속성별로 두 개이므로 싱크 노드에서는 트리가 두 개가 생성될 것이며, 온도에 대한

SDMTree는 중복된 값을 제거하여 키 값 20, 21, 22, 23, 24, 25로 6개이며, 조도에 대한 SDMTree는 47, 48, 49, 50, 51, 53, 54로 7개의 키 값을 갖게 된다.

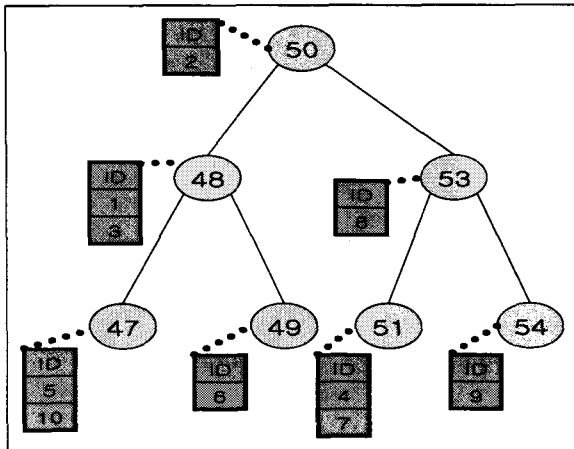
그리고 각각의 SDMTree는 루트 키 값으로 Median 함수를 사용하여 온도 SDMTree는 23, 조도 SDMTree는 50을 갖게 된다.

온도 SDMTree트리는 [그림 3]과 같이 구성되며, 조도 SDMTree는 [그림 4]와 같이 구성된다.



[그림 3] 온도 SDMTree

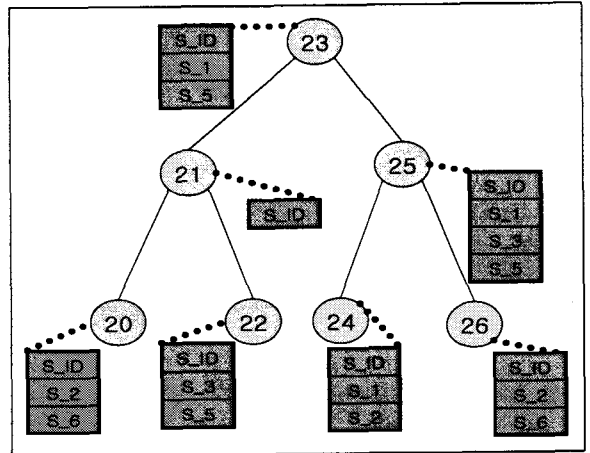
[그림 3] 온도 SDMTree의 구성을 살펴보면, 트리의 키 값은 온도 데이터의 속성값을 이루어져 있고 키 값에 따라 중복된 속성값을 가진 노드 ID를 테이블 형태로 구성하여 저장하고 관리된다. [그림 4] 조도 SDMTree도 온도 SDMTree와 마찬가지로 트리의 키 값은 조도에 대한 속성값으로 이루어져 있고 중복된 조도를 가진 노드 ID값을 저장, 관리하기 위해 키 값에 따른 별도의 테이블을 유지하게 된다.



[그림 4] 조도 SDMTree

싱크 노드의 상위 노드인 게이트웨이의 SDMTree 구성을 살펴보면, 게이트웨이는 자신의 하위 노드인 도는 싱크 노드로부터 SDMTree에 대한 정보, 즉 싱크 노드의 ID와 SDMTree의 키 값을 수집하여 싱크 노드와 같은 방법으로 SDMTree를 구성하게 된다. 게이트웨이에서의 SDMTree의 예를 보면, [그림 5]와 같다.

[그림 4] 게이트웨이 온도 SDMTree의 구성을 살펴보면, 트리의 키 값은 싱크 노드의 온도 SDMTree의 정보에서 얻은 모든 키 값이며, 중복되는 키 값을 가진 싱크 노드를 저장하고 관리하기 위해 테이블을 구성, 관리한다.



[그림 5] 게이트웨이 온도 SDMTree

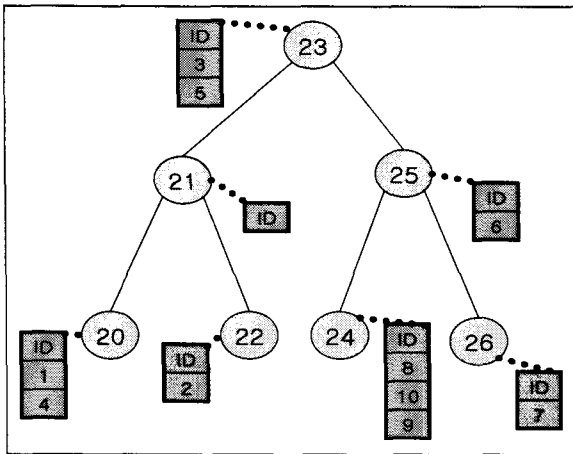
[그림 5]에서 S_ID는 싱크 노드의 ID를 의미하며, 논리적인 주소이며, 키 값 23은 싱크 노드 1번과 5번에 해당되는 데이터가 있다는 것을 나타내며, 키 값 21은 해당 싱크 노드가 없다는 것을 의미한다. 이러한 방법으로 게이트웨이의 질의 처리기를 구성하여 관리하게 됨으로써 얻는 이점은 사용자가 만약 온도가 22도 이하인 노드의 ID를 모두 원한다면, 네트워크 전역에 모든 싱크 노드에게 질의를 전송하지 않고, 키 값 22도 이하를 갖고 있는 해당 싱크 노드 2, 3, 5, 6에게만 질의를 전송하면 되므로, 질의 전송에 소모되는 통신 에너지뿐만 아니라, 센서 노드에서의 센싱에 소모되는 에너지도 줄일 수 있으므로 센서 네트워크 시스템 전체를 고려한다면 에너지 효율성 측면에서는 큰 이점을 갖게 될 것이다.

4.2 SDMTree 갱신

SDMTree의 갱신은 센서 노드로부터 수신된 데이터가 각각의 속성별 SDMTree에 없는 키 값이면 트리의 삽입이 필요하게 되며, 만약 키 값의 테이블에 저장된 노드의 ID가 모두 다른 속성값을 가져서 다른 키 값의 테이블로 이동한다거나 해당하는 노드의 ID가 없어서 테이블이 비어있는 상태가 되더라도 그 키 값은 삭제하지 않는다. 왜냐하면 센서 네트워크의 특징상 하나의 클러스터 내에 있는 센서 노드들은 근접한 지역에 위치하게 되고,

환경에 따라 센싱 데이터가 연속적으로 변화하게 되므로, 한 번 센싱된 데이터는 나중에 다시 그 값을 센싱할 수 있는 확률이 높으므로 트리의 키 값을 삭제하였다가 다시 삽입하는 비용보다 키를 항상 유지하고 있다고 나중에 테이블의 값만 갱신하는 것이 더 효율적이기 때문이다.

SDMTree의 갱신 과정을 설명하면, 트리에 키 값을 가진 노드가 없다는 것을 확인하고 키 값을 기본으로 하여 트리의 노드 탐색을 한다. 만약 이 때 키 값이 있어야 할 위치에 키 값이 없으면 종료된 위치에 키 값을 삽입하게 된다. 예를 들어, 앞의 [그림 3] 온도 SDMTree에서 온도 변화가 생겼다가 가정하자. 노드 ID가 7인 센서 노드가 온도의 변화로 인해 센싱 데이터의 온도 속성값이 21도에서 26로 바뀌고, 노드 ID가 9인 센서 노드도 21도에서 24도로 변했으면, 키 값 21은 해당하는 센서 노드가 없으므로 테이블이 비게 되며, 키 값 26도는 없기 때문에 새로운 노드를 추가하게 된다. 그러므로 온도 SDMTree는 [그림 3]에서 [그림 6]과 같이 갱신된다.



[그림 6] 온도 SDMTree 갱신

SDMTree 알고리즘은 다음과 같다.

[검색 알고리즘]

```
SearchSDT(T, d_key)
  p <- T;
  if( p = null) then
    return null;
  if(p.key = d_key) then
    return p;
  if(p.key < d_key) then
    return SearchSDT(p.right,
  d_key);
  else return SearchSDT(p.left, d_key);
End searchSDT()
```

[삽입 알고리즘]

```
InsertSDT(T, new_key)
```

```
p <- T;
while(p ≠ null) do {
  if( new_key = p.key) then return;
  q <- p;
  if(new_key < p.key) then p <- p.left
  else p <- p.right
}
newNode <- getNode();
newNode.key <- new_key
newNode.right <- null;
newNode.left <- null;
if (T=null) then T <- newNode
else if(new_key < q.key) then
  q.left <- newNode
else
  q.right <- newNode
return;
End insertSDT()
```

5. 결론

본 연구에서는 해양 센서 네트워크 분야에서 네트워크 생존 시간과 관련하여 중요한 요소인 에너지 효율성과 사용자의 질의에 대한 응답을 정확하고 신속하게 처리는 물론 에너지 소모에서도 효율적인 질의 처리기의 구성 방법으로 SDMTree를 제안하였다. 상위 노드에서 하위 노드의 데이터를 효율적으로 수집하고 관리하기 위하여 제안된 SDMTree는 싱크 노드에서는 센서 노드로부터 받은 센싱 데이터를 속성별로 구분하고 중복된 센싱 데이터를 제거한 후에 트리 형태로 구성하고 상위 노드의 게이트웨이, 베이스스테이션에서도 하위 노드의 트리 정보를 수신한 후에 트리 형태로 질의 처리기를 구성하게 되므로 질의에 대한 응답을 정확하고 신속하게 검색할 수 있으며, 베이스스테이션이나 게이트웨이, 싱크 노드에서는 질의가 수신하게 되면 질의를 분석 처리하고 네트워크 전역에 모든 노드에게 질의를 전송하는 것이 아니라 해당하는 특정 노드에서 질의 및 질의에 대한 응답을 전송할 수 있으므로 데이터 저장 및 데이터 전송에 소모되는 불필요한 에너지를 줄일 수 있다.

향후 연구 과제로는 제안한 SDMTree를 구현하여 데이터의 성능 평가와 실험을 실시하고, 트리의 갱신과 관련된 사항을 연구할 필요가 있다.

참고문헌

- [1] I.F. Akyildiz, S. Wdilian, 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 학술발표 연구집 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] A. Murphy and W. Heinzelman, "MiLAN: Middleware Linking Applications and Networks," TR-795, University of Rochester, Computer Science, Nov. 2002
- [10] 변창우, 박석, 손상혁, "센서 네트워크 환경에서의 데이터 관리", 2004. 12. 정보과학회지 제22권 제12호