

데이터 중심 저장 방식의 센서 네트워크를 위한 비균등 영역 분할 기법[†]

A Non-Equal Region Split Method for Data-Centric Storage in Sensor Networks

강홍구* / Hong-Koo Kang 전상훈** / Sang-Hun Jeon
홍동숙*** / Dong-Suk Hong 한기준**** / Ki-Joon Han

요약

데이터 중심 저장(Data-Centric Storage: DCS) 방식의 센서 네트워크는 같은 값의 데이터를 같은 노드에 저장하기 때문에 센서 네트워크가 확장되거나 같은 값의 데이터가 빈번하게 발생하면 특정 센서 노드에 저장 부하가 집중되어 에너지 효율성이 나빠지는 문제가 발생한다. 기존의 데이터 중심 저장 방식에 대한 연구들은 저장 데이터의 효율적인 관리에만 치우쳐 센서 네트워크의 확장에 따른 에너지 효율성 문제를 고려하지 않았다.

본 논문에서는 다차원 센서 데이터 저장의 효율적인 확장성(Scalability)을 지원하는 비균등 영역 분할(Non-Equal Region Spilt) 기법을 제안한다. 제안한 기법은 센서 네트워크를 센서 노드의 분포에 따라 같은 센서 노드 개수를 갖도록 서로 다른 크기의 영역으로 분할하고 분할된 각 영역 내에서 측정된 데이터를 해당 영역에서 저장 및 관리함으로써 센서 네트워크의 확장에 따른 저장 비용을 줄였다. 또한 분할 영역 개수를 센서 네트워크의 크기와 센서 노드 개수, 센서 데이터 발생량에 비례하게 증가시켜 센서 노드의 에너지 소모를 분산시킴으로써 센서 네트워크의 수명 연장과 확장성을 높였다.

Abstract

A sensor network which uses DCS(Data-Centric Storage) stores the same data into the same sensor node. Thus it has a hot spot problem when the sensor network grows and the same data arise frequently. In the past researches of the sensor network using DCS, the hot spot problem caused by growing the sensor network was ignored because they only concentrated on managing stored sensor data efficiently.

In this paper, we proposed a non-equal region split method that supports efficient scalability on storing multi-dimensional sensor data. This method can reduce the storing cost, as the sensor network is growing, by dividing whole space into regions which have the same number of sensor nodes according to the distribution of sensor nodes, and storing and managing sensor data within each region. Moreover, this method can distribute the

† 본 연구는 서울시 산학연 협력사업의 지원으로 수행되었음.

■ 논문접수 : 2006.10.17

■ 심사완료 : 2006.12.06

* 교신저자 건국대학교 대학원 컴퓨터공학과 박사과정 (khang@db.konkuk.ac.kr)

** 삼성전자 반도체총괄 MEMORY 사업부 선임연구원 (shjeon@db.konkuk.ac.kr)

*** 건국대학교 대학원 컴퓨터공학과 박사과정 (dshong@db.konkuk.ac.kr)

**** 건국대학교 컴퓨터공학부 교수 (kijhan@db.konkuk.ac.kr)

energy consumption of sensor nodes by increasing the number of regions according to the size of the sensor network, the number of sensor nodes within the sensor network, and the quantity of sensor data. Therefore it can help to increase the life time and the scalability of the sensor network.

주요어 : 센서 네트워크, 데이터 중심 저장, 영역 분할, 에너지 효율성, 확장성

Keyword : sensor network, DCS, region split, energy efficiency, scalability

1. 서론

최근 무선 통신의 급속한 발전과 통신 분야의 하드웨어 기술 발전으로 산업 현장, 자연 현상 관측, 군사 시설 뿐만 아니라 실생활과 밀접한 관계를 이루는 분야까지 센서 네트워크의 활용 분야가 확산되고 있다[1,2,3]. 센서 네트워크는 일반적으로 적게는 수백 개에서 많게는 수백만 개의 센서 노드들로 구성되며 하나의 센서 노드는 다양한 주변 환경의 데이터를 측정하기 위해 다수의 센서들을 가지고 있다. 이러한 센서 노드는 일반적으로 온도, 습도, 조도 등과 같은 다양한 스칼라 데이터를 측정한다[4,5,6].

센서 네트워크에서 측정된 데이터를 저장하는 방식으로는 센서 네트워크 외부 저장소에 데이터를 저장하는 외부 저장 방식(External Storage: ES), 측정한 센서 노드 자신이 데이터를 저장하는 지역 저장 방식(Local Storage: LS), 그리고 데이터의 값에 따라 데이터를 저장할 센서 노드를 결정하는 데이터 중심 저장 방식(Data-Centric Storage: DCS)이 있다[7,8].

본 논문은 위의 세 가지 데이터 저장 방식 중에서 데이터 중심 저장 방식의 센서 네트워크를 기반으로 한다. 데이터 중심 저장 방식은 측정된 데이터를 값에 따라 특정 키 값으로 분류하여 특정 센서 노드에 저장한다. 외부 저장 방식에서 외부 저장 공간에 가까운 센서 노드들에게 부하가 집중되거나 지역 저장 방식에서 질의를 수행할 때 필요치 않은 센서 노드가 질

의 처리에 관여되는 것을 데이터 중심 저장 방식에서는 방지할 수 있다[9]. 그러나, 데이터 중심 저장 방식의 센서 네트워크에서는 센서 노드의 추가로 센서 네트워크가 확장되면서 특정 센서 노드에 통신 부하가 증가되고 측정된 데이터의 위치와 데이터 저장 위치가 멀리 떨어지게 되면서 데이터 저장 시 통신비용이 증가한다. 또한, 값이 같은 데이터가 빈번하게 발생하면 이들 데이터를 저장하는 특정 노드에 저장하기 위한 통신비용이 가중되는 문제가 있다[10,11].

그러므로 데이터 중심 저장 방식의 센서 네트워크에서는 센서 네트워크의 확장으로 인한 특정 센서 노드의 통신 부하를 줄이고 전체 데이터의 저장 비용을 줄이는 것이 필요하다. 특히 전체 통신비용을 지역적으로 분산시켜 모든 센서 노드들의 에너지 소비를 줄임으로써 센서 네트워크의 수명을 연장하는 것이 무엇보다 중요하다[11,12]. 이러한 요건을 만족시키기 위해 지리적인 위치를 이용한 인덱스 기법들이 연구되고 있다[4,7,8,9]. 그러나, 기존에 제시된 연구들은 저장된 데이터의 효율적인 관리에만 치우쳐 센서 네트워크의 확장에 따른 에너지 효율성을 고려하지 않고 있다[13,14].

따라서, 본 논문에서는 이와 같은 문제를 해결하기 위하여 센서 데이터 저장의 효율적인 확장성(Scalability)을 지원하는 비균등 영역 분할(Non-Equal Region Spilt) 기법을 제안한다. 제안한 기법은 센서 네트워크를 센서 노드의 분포에 따라 같은 센서 노드 개수를 갖도록 서로 다른 크기의 영역으로 분할하고 분할된

각 영역 내에서 측정된 데이터를 저장 및 관리함으로써 센서 네트워크의 확장에 따른 데이터 저장 비용 증가와 센서 노드의 통신 부하를 줄였다. 그리고 분할 영역 개수를 센서 네트워크 크기와 센서 노드 개수, 각 센서 노드가 발생하는 데이터 양에 비례하게 증가시킴으로 센서 노드의 에너지 소모를 최소화하고 센서 네트워크의 수명을 연장하였다.

본 논문의 구성은 다음과 같다. 제 2 장의 관련 연구에서는 센서 네트워크에서의 데이터 저장 방식에 대해 소개하고, 데이터 중심 저장 방식의 센서 네트워크에서 제안된 기존 연구에 대해 설명한다. 제 3 장에서는 본 논문에서 제시한 비균등 영역 분할 기법과 알고리즘에 대해 기술한다. 제 4 장에서는 비균등 영역 분할 기법을 적용한 센서 네트워크의 성능을 분석하고 결과를 보여준다. 마지막으로 제 5 장에서는 결론 및 향후 연구에 대해 언급한다.

2. 관련 연구

본 장에서는 외부 저장 방식, 내부 저장 방식, 데이터 중심 저장 방식으로 구성되는 센서 네트워크에서의 데이터 저장 방식에 대해서 살펴본다. 그리고 데이터 중심 저장 방식의 센서 네트워크에서 제안된 기존 연구들에 대해 설명한다.

2.1 데이터 저장 방식

센서 네트워크는 일반적으로 적게는 수백 개의 센서 노드부터 많게는 수만 개의 센서 노드로 구성되어 있다. 이러한 센서 노드는 온도, 습도, 속도 등의 다양한 스칼라 데이터를 측정하며 측정된 데이터를 저장하는 방식은 크게 외부 저장 방식, 지역 저장 방식, 데이터 중심 저장 방식의 3가지로 분류된다[7,8].

외부 저장 방식은 센서 네트워크에서 발생하는 데이터를 외부에 존재하는 데이터 저장소

에 저장하는 방식이다. 이 방식은 측정된 모든 데이터가 외부 저장소에 저장되고 처리되므로 측정된 데이터에 대한 질의 시 질의에 대한 추가비용이 없으며 질의에 대한 응답 속도가 빠르다. 그러나 측정된 모든 데이터는 외부에 저장되기 때문에 외부 저장소와 연결되어 게이트웨이 역할을 수행하는 센서 노드는 부하가 집중되어 에너지 소모가 높아져 수명이 짧아지는 문제가 있다.

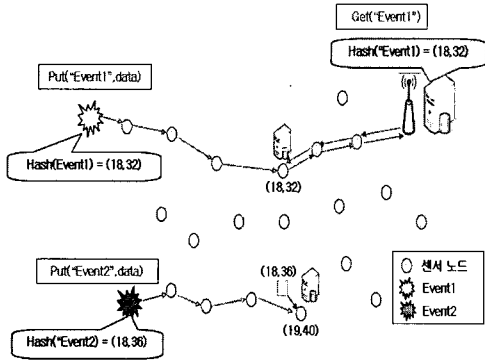
지역 저장 방식은 센서 노드 자신이 측정된 데이터를 센서 노드 자신이 저장하는 방식이다. 이 방식은 측정된 모든 데이터를 센서 노드 자신의 저장소에 저장하기 때문에 데이터의 저장을 위한 추가비용은 없으나 질의 시 결과 데이터를 얻기 위해서는 센서 네트워크를 구성하는 모든 센서 노드에게 질의를 전달해야 하는 문제가 있다. 따라서, 질의가 빈번한 환경에서는 전체 센서 네트워크의 수명이 빠르게 감소하게 된다.

데이터 중심 저장 방식은 측정된 데이터의 값에 따라 저장할 센서 노드를 결정하여 저장하는 방식이다. 센서 노드가 데이터를 측정하면 해당 데이터를 저장하는 센서 노드로 측정된 데이터를 전송하여 저장한다. 모든 데이터는 값에 따라 분류 저장되므로 질의 시 해당 값을 저장하고 있는 센서 노드에게 질의를 바로 전달할 수 있다. 따라서, 데이터 중심 저장 방식은 센서 네트워크의 크기가 크고 데이터의 저장량이 많이 발생하는 환경에 효율적이다.

2.2 GHT(Geographic Hash Table)

GHT는 데이터 중심 저장 방식의 센서 네트워크에서 데이터의 값을 기반으로 지리적 위치를 생성하고 생성된 지리적 위치와 가장 가까운 센서 노드에 데이터를 저장하는 색인 기법이다[7]. GHT는 데이터의 저장을 위한 Put() 연산과 질의 처리를 위한 Get() 연산을 제공하며 데이터 전송과 질의 전송을 위해 GPSR[8]

을 사용한다. <그림 1>은 GHT에서의 데이터 저장과 질의 처리를 보여준다.

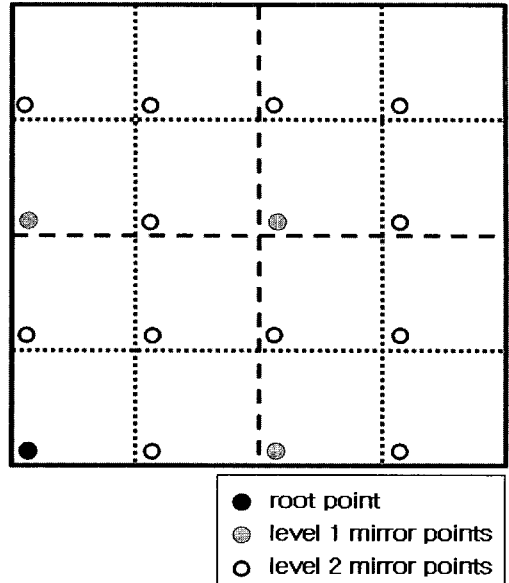


<그림 1> GHT에서의 데이터 저장과 질의 처리

<그림 1>에서 보는 것과 같이 센서 노드가 측정된 데이터 값이 "Event1"에 속하는 값일 경우 Put("Event1", data)를 호출한다. Put("Event1", data) 연산은 "Event1"을 해쉬하여 지리적인 위치 값(18, 32)를 생성한 후, GPSR을 통하여 생성한 지리적 위치에 있는 센서 노드에 data를 저장한다. 그리고 센서 노드가 측정된 데이터 값이 "Event2"인 경우에는 "Event2"를 해쉬하여 나온 지리적 위치 값 (18, 36)에 센서 노드가 없기 때문에 생성한 지리적 위치에서 가장 가까운 센서 노드에 데이터를 저장한다. 그리고 질의 처리시 사용자가 "Event1"에 대한 검색을 원할 경우 Get ("Event1")을 호출한다. Get("Event1") 연산은 입력된 "Event1"을 해쉬하여 지리적인 위치 값 (18, 32)를 생성하고, 생성된 지리적 위치에 있는 센서 노드로 질의를 전달한다.

GHT에서는 센서 네트워크의 확장 시 센서 노드의 에너지 효율성을 위해 분할 레벨을 d라 할 때, 전체 센서 네트워크의 영역을 4d(d≥0)개로 정분할하는 Structured Replication을 사용한다[7,8]. <그림 2>는 Structured Replication을 보여준다.

<그림 2>에서 보는 것과 같이 센서 네트워크 영역을 정분할한 후 각 영역에 mirror



<그림 2> Structured Replication

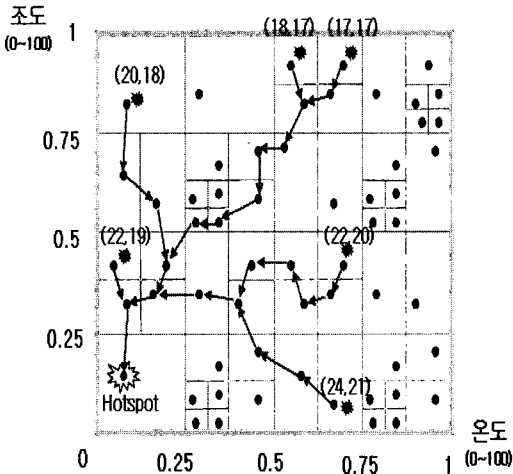
point라는 대표 센서 노드를 둔다. 질의는 root point로부터 level 1 mirror points에게 전달되고, level 1 mirror points는 level 2 mirror points에게 전달한다. 그러나 Structured Replication은 각 분할 영역에 대표 센서 노드를 두는 계층 구조 인덱스를 구성하기 때문에 통신이 상위 노드로 집중되어 대표 노드의 통신비용이 높아지는 문제가 있다.

2.3 DIM(Distributed Index for Multi-dimensional data)

DIM은 데이터 중심 저장 방식의 센서 네트워크에서 다차원 센서 데이터 값을 기반으로 센서 네트워크의 공간 영역을 서로 매핑하면서 지리적으로 인접한 센서 노드에 데이터를 저장하는 인덱스이다[4]. DIM은 센서 네트워크 영역을 x좌표와 y좌표로 번갈아 가면서 분할하면서 하나의 영역에 하나의 센서 노드만 남게 될 때까지 영역을 분할하는데 이 영역을 Zone이라 한다. 센서 노드에서 데이터가 발생하면 데이터를 해쉬하여 Zone 코드를 생성하고, Zone

코드가 나타내는 Zone에 있는 센서 노드에 GPSR을 사용하여 데이터를 저장한다. 만일 해당 Zone에 센서 노드가 위치하지 않을 경우 Backup Zone에 데이터를 저장한다. Backup Zone은 센서 노드가 없는 Zone의 데이터를 저장하는 이웃한 Zone이다. 따라서 특정 Zone에 센서 노드가 존재하지 않더라도 Backup Zone에 데이터를 저장할 수 있다.

DIM은 공간 영역을 데이터 값과 매핑하여 데이터를 저장하므로 다차원 센서 데이터에 대한 저장과 질의 처리가 가능하다. 그러나 유사한 값의 데이터가 빈번하게 발생할 경우 특정 센서 노드에 통신이 집중되고 지리적으로 멀리 위치한 센서 노드 간의 통신비용이 많이 된다. <그림 3>은 DIM에서의 데이터 저장을 보여준다.



<그림 3> DIM에서의 데이터 저장

<그림 3>에서 보는 것과 같이 DIM에서는 유사한 값의 데이터가 다수 발생할 경우 특정 센서 노드에 통신이 집중되어 Hotspot이 발생되고 센서 노드의 수명이 단축된다. 또한 센서 네트워크 크기의 증가로 데이터를 발생한 센서 노드와 저장할 센서 노드와의 거리가 멀어질 경우 데이터 저장을 위한 통신비용이 높아지는 문제가 있다.

3. 비균등 영역 분할 기법

본 장에서는 본 논문에서 제시한 비균등 영역 분할 기법에 대해서 설명한다. 비균등 영역 분할 기법은 전체 센서 네트워크 영역을 센서 노드의 개수에 따라 서로 다른 크기의 Region으로 분할하고, 분할된 각 Region 내에서 측정된 데이터는 각 Region 내에서 저장 및 관리한다.

3.1 Region 생성

전체 센서 네트워크 내의 모든 센서 노드의 센싱 영역을 포함하는 사각형을 $A(x-y$ 평면)라고 할 때, Region은 사각형 A 를 x 축 또는 y 축으로 분할하여 얻어진 영역을 말한다. Region 생성에서 센서 네트워크에 있는 센서 노드 개수가 짝수 개이면 분할된 두 Region이 갖는 센서 노드 개수를 동일하게 하고 홀수 개이면 분할된 두 Region 중 영역 크기가 큰 Region에 센서 노드 개수를 하나 더 많도록 한다.

센서 네트워크에서 센서 노드의 분포가 고르지 않은 상태에서 Region은 센서 노드 개수가 같거나 최대 한개 차이만 나도록 분할함으로써 생성되기 때문에 각 Region의 크기는 서로 다르고 분할 횟수가 i 번일 때, 생성되는 Region 개수는 $2^i(i>0)$ 개가 된다. 그리고 각 Region은 서로 다른 Region과 구별되는 Region 코드를 갖는다. Region 코드는 영역 분할 시 분할 영역이 좌측 또는 하단일 경우에는 비트 0을 추가하고 반대로 분할 영역이 우측 또는 상단일 경우에는 비트 1을 추가한다.

또한 생성된 각 Region에 대해서는 다차원 범위 질의를 위한 색인 기법인 DIM을 적용하여 Zone을 생성한다. 여기서 Zone은 Region을 x 축과 y 축을 번갈아가며 같은 크기의 영역으로 분할하고 각 분할된 영역에 하나의 센서 노드가 남을 때까지 분할을 반복하여 얻어진

사각형을 말한다. Zone은 서로 다른 Zone과의 구별을 위한 Zone 코드를 갖는다. Zone 코드는 x축 또는 y축으로 분할시마다 추가되는 비트로 구성된다. 분할된 Zone이 좌측 또는 하단일 경우에는 비트 0을 추가하고 우측 또는 상단일 경우에는 비트 1을 추가한다. Region 생성 알고리즘은 <그림 4>와 같다.

Algorithm : CreateRegion(area, sensors, regionFactor)

```

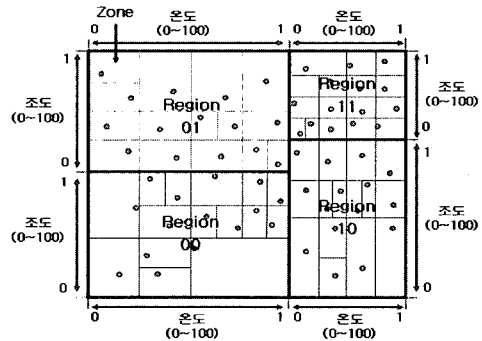
1: Begin
2:   counter = 1;
3:   while(1)
4:     if(!(counter mod 2)) then
5:       SortSensor(xAxis);
6:       newX = sensors[sensorNum/2].x;
7:       newRegion[k++] = SplitRegion(area, newX);
8:     else
9:       SortSensor(yAxis);
10:      newY = sensors[sensorNum/2].y;
11:      newRegion[k++] = SplitRegion(area, newY);
12:     end if
13:     UpdateRegionCode(newRegion[k]);
14:     if(regionFactor == counter) then
15:       exit;
16:     end if
17:     counter++;
18:   end while
19: End
    
```

<그림 4> Region 생성 알고리즘

<그림 4>의 Region 생성 알고리즘에서 입력 인자로서 area는 전체 센서 네트워크 영역이며, sensors는 센서 네트워크를 구성하는 센서 노드들의 배열이고, regionFactor는 생성할 Region 개수이다. 라인 2는 생성된 Region의 개수를 비교할 counter를 설정하고, 라인 3은 모든 Region 생성을 마칠 때까지 Region 생성을 반복 수행한다. 라인 4부터 라인 12까지는 counter의 값에 따라 Region 생성을 위한 영역 분할 기준 축을 결정한다. counter의 값이 홀수일 경우에는 x축을 기준으로 영역을 분할하고, counter의 값이 짝수일 경우에는 y축을 기준으로 영역을 분할한다. 라인 13은 생성한 Region에 대한 Region 코드를 생성하고, 라인 14는 생성된 Region 개수가 counter 값과 같은지를 검사한다. 만약 개수가 같지 않으면 라인 17에서 counter를 하나 증가시킨 후 Region

생성을 반복한다. 만약 개수가 같으면 Region 생성을 종료한다.

<그림 5>는 Region 생성 알고리즘에 의한 Region 생성의 예를 나타낸다.



<그림 5> Region 생성의 예

<그림 5>에서 보는 것과 같이 생성된 Region의 개수는 4개이며 대상 데이터로는 조도와 온도를 설정하였고 데이터의 범위는 0~100으로 하였다.

3.2 데이터 저장

센서 네트워크를 비균등으로 영역 분할한 후 데이터는 생성된 각 Region 내에 저장된다. Region 내에 발생한 데이터를 저장하는 알고리즘은 <그림 6>과 같다.

Algorithm : StoreData(a, d)

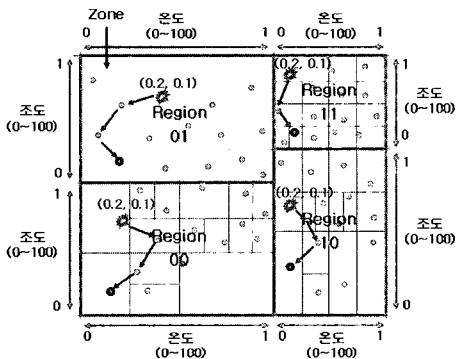
```

1: Begin
2:   c = ZondEncode(d);
3:   if(Contain(a, c) and IsInternal()) then
4:     Store(d);
5:   else
6:     b = CountNeighbor(c);
7:     for i from 0 to b do
8:       y = GetNeighbor(c);
9:       if(Closer(y, a, c)) then
10:        o = y; i++;
11:      end if
12:    end for
13:    if(o == a) then
14:      r = RightHandRouting(o, Code(d));
15:      StoreData(r, d);
16:    else
17:      SendData(o, d);
18:      StoreData(o, d);
19:    end if
20:  end if
21: End
    
```

<그림 6> 데이터 저장 알고리즘

<그림 6>의 데이터 저장 알고리즘에서 입력 인자 a는 저장할 데이터를 생성한 센서 노드의 Zone 코드이고 d는 저장할 데이터이다. 데이터 d는 자신이 발생한 Region 내의 해당 Zone에 존재하는 센서 노드에 저장된다. 먼저 데이터가 발생하면 발생한 데이터를 저장할 센서 노드의 위치를 알기 위해 Zone 코드를 생성한다. 라인 2는 데이터 발생 시 ZoneEncode() 수행하여 해당 Zone 코드 c를 생성하고, 라인 3은 생성한 Zone 코드 c가 데이터를 발생시킨 노드의 Zone 코드 a와 동일한지 검사한다. 만약 a와 c가 동일하면 데이터가 발생한 Zone 내의 센서 노드에 데이터를 저장하고 알고리즘을 종료한다. 그러나 a와 c가 다르면 데이터를 저장할 센서 노드를 찾기 위해 라인 6을 실행하여 데이터가 발생한 센서 노드의 이웃 센서 노드 개수를 가져온다. 라인 7부터 라인 12는 이웃 센서 노드들 중 하나에 데이터를 저장하기 위해 센서 영역과 가장 가까운 센서 노드 o를 찾는다. 이때, 만일 센서 노드 o가 데이터 발생 노드 a보다 목적 Zone 코드 c에 더 가까우면 이웃 센서 노드 o에 GPSR을 이용하여 데이터를 전달하고 그렇지 않으면 RightHandRule()을 호출하여 라우팅 경로를 우회한 후 데이터 저장 알고리즘을 다시 수행한다. 이후 목적 센서 노드를 찾으면 데이터를 저장하고 알고리즘을 종료한다.

<그림 7>은 센서 네트워크에 4개의 Region을 생성 후 데이터를 저장하는 예를 보여준다.



<그림 7> Region 생성 후 데이터 저장

<그림 7>에서 굵은 선으로 그려진 사각형은 Region을 나타내고, 가는 선으로 그려진 사각형은 Zone을 나타낸다. 각 Region 안의 00, 01, 10, 11은 Region 코드를 나타낸다. 데이터는 데이터가 발생한 Region 내에 분산하여 저장되기 때문에 센서 네트워크가 확장되어도 데이터 저장을 위한 통신비용과 Hotspot 통신비용을 효율적으로 줄일 수 있다.

3.3 질의 처리

질의 처리의 경우 질의는 생성된 모든 Region로 전달된다. 질의 처리를 위한 알고리즘은 <그림 8>과 같다.

```

Algorithm : QueryData(a, q)
1: Begin
2: s = SplitQuery(q);
3: for i from 0 to sCount do
4: c = ZoneEncode(si);
5: if(Contain(a, c) and IsInternal(Resolve(si))) then
6: SendResponse(si); break;
7: else
8: b = CountNeighbor(c);
9: for j from 0 to b do
10: v = GetNeighbor(c);
11: if(Closer(v, a, c)) then
12: o = v; j++;
13: end if
14: end for
15: if(o == a) then
16: r = RightHandRouting(o, c);
17: QueryData(r, si);
18: else
19: SendQuery(o, si);
20: QueryData(o, si);
21: end if
22: end if
23: end for
24: End
    
```

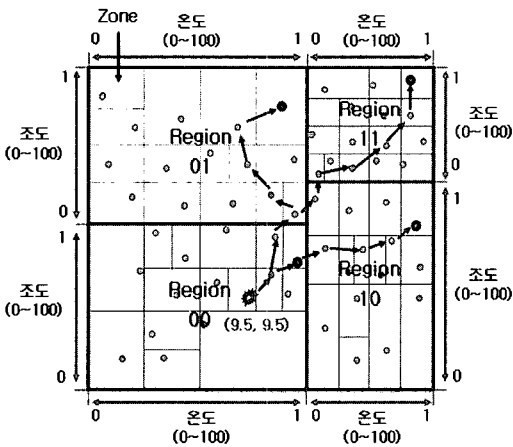
<그림 8> 질의 처리 알고리즘

<그림 8>의 질의 처리 알고리즘에서 입력 인자 a는 질의를 생성한 센서 노드의 Zone 코드이고 q는 질의이다. 라인 2는 질의 범위가 여러 Zone을 포함할 경우 각 Region의 Zone 구성에 따라 질의를 분할한다. 라인 3부터 라인 23은 분할된 질의의 개수만큼 질의를 처리한다. 라인 4는 분할된 질의에 대한 목적 센서 노드의 Zone 코드 c를 생성한다. 만약 c와 질의를 발생한 센서 노드의 Zone 코드 a가 같고 해당 데이터를 가지고 있으면 결과를 돌려준

다. 만약 c와 a가 다른 경우 라인 7부터 라인 23은 GPSR을 통해 각 Region의 해당 Zone에 질의를 전달한다. 질의가 해당 Zone의 센서 노드에 도달하면 SendResponse()를 통해 목적 센서 노드에 도달한 경로의 반대 경로로 질의에 대한 결과를 돌려준다. 분할된 모든 질의에 대한 결과를 모두 돌려주고 알고리즘을 종료한다.

<그림 9>는 센서 네트워크에 4개의 Region을 생성 후 질의를 전달하는 예를 보여준다.

<그림 9>에서 굵은 선으로 그려진 사각형은 Region을 나타내고, 가는 선으로 그려진 사각형은 Zone을 나타낸다. 각 Region안의 00, 01, 10, 11은 Region 코드이다. 질의 전달은 질의가 발생한 Region 뿐만 아니라 다른 Region의 해당 센서 노드에도 질의를 전달하기 때문에 Region 생성 전과 비교할 때 발생된 질의 개수는 생성된 Region의 개수만큼 증가한다.



<그림 9> Region 생성 후 질의 전달

4. 성능 평가

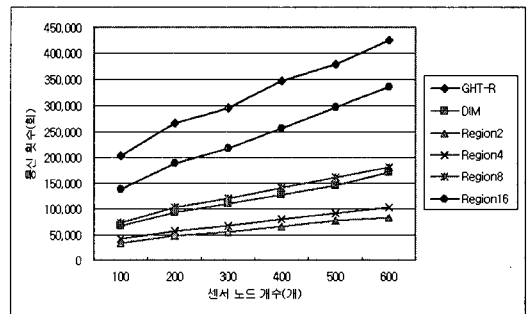
본 장에서는 본 논문에서 제시한 비균등 영역 분할 기법에 대한 성능 평가를 살펴본다. 비균등 영역 분할 기법에 대한 성능 평가를 위하여 사용된 시스템 사양은 Intel Pentium4 CPU 2.0GHz, 512MB RAM이고, 운영체제는

Windows XP Professional을 사용하였다. 개발 언어로는 C#을 사용하였고, GHT-R과 DIM, 그리고 본 논문에서 제시한 비균등 영역 분할 기법을 적용한 인덱스를 구현하여 전체 센서 네트워크 통신비용과 Hotspot 통신비용을 비교 평가하였다. 여기서 GHT-R은 범위 질의를 수행할 수 있도록 확장한 GHT이다.

4.1 전체 센서 네트워크 통신비용 평가

센서 네트워크에 있는 전체 센서 노드에서 발생한 데이터를 저장하기 위한 통신비용과 질의를 처리하기 위한 통신비용을 모두 고려한 전체 센서 네트워크 통신비용을 비교 평가하였다. 실험 조건으로 센서 노드의 통신 범위를 50m로 설정하고 센서 노드 개수를 100개부터 600개까지 100개씩 증가시켰다. 그리고 센서 노드에서 데이터와 질의의 발생 비율이 10:1일 경우 전체 센서 네트워크 통신비용을 비교 평가 하였다. 또한, 데이터와 질의의 발생 비율을 1:1에서부터 1:100까지 비율을 변화시키면서 전체 통신비용을 비교 평가하였다.

먼저 데이터와 질의의 발생 비율이 10:1일 경우(고정비율)의 전체 센서 네트워크 통신비용에 대한 실험 결과는 <그림 10>과 같다.

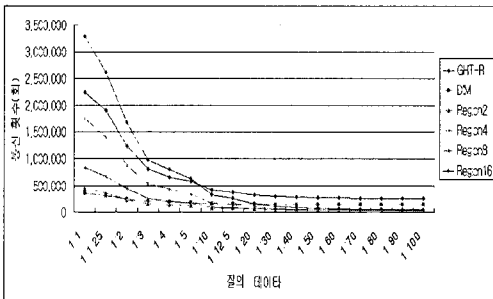


<그림 10> 전체 센서 네트워크 통신비용(고정비율)

<그림 10>에서 보는 것과 같이 전체 센서 네트워크 통신비용은 비균등 영역 분할 기법의 Region 개수가 2개, 4개, 8개, 16개일 때 각각

GHT-R보다 최대 83%, 79%, 64%, 32%가 감소하였다. 그리고 Region 개수가 2개, 4개일 때 각각 DIM보다 최대 52%, 40%가 감소하였고, Region 개수가 8개, 16개일 때 각각 최대 11%, 106%가 증가하였다. 이처럼 비균등 영역 분할 기법을 적용한 인택스의 통신비용이 효율적인 것은 전체 센서 네트워크에서 발생하는 데이터가 비균등 영역 분할로 생성된 각 Region에 분산 저장되어 라우팅 횟수가 크게 줄었기 때문이다. 그러나 Region 개수가 8개, 16개일 때 DIM보다 통신비용이 높아진 것은 전체 센서 네트워크에서 발생한 질의가 비균등 영역 분할로 생성된 모든 Region으로 전달되어야하므로 Region 개수가 증가할수록 질의를 위한 라우팅 횟수가 증가했기 때문이다.

센서 노드 개수가 600개일 때 센서 노드에서 데이터와 질의의 발생 비율이 1:1에서부터 1:100까지의 비율 변화(가변비율)에 따른 전체 센서 네트워크 통신비용의 실험 결과는 <그림 11>과 같다.



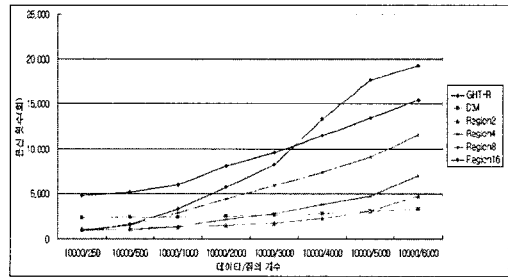
<그림 11> 전체 센서 네트워크 통신비용(가변비율)

<그림 11>에서 보는 것과 같이 전체 센서 네트워크 통신비용은 데이터와 질의 발생 비율이 1:1일 때 DIM이 가장 효율적이고 데이터와 질의의 발생 비율이 2:1부터 12.5:1까지는 Region 개수가 2개인 경우가 가장 효율적이다. 또한, 데이터와 질의의 발생 비율이 20:1과 80:1일 때 각각 Region 개수가 4개, 8개인 경우가 가장 효율적인 것으로 나타났다. 이러

한 실험 결과를 통하여 질의 처리보다 데이터 저장이 빈번한 센서 네트워크 환경에서 비균등 영역 분할 기법을 적용하면 효율적으로 통신비용을 줄일 수 있음을 알 수 있다.

4.2 Hotspot 통신비용 평가

전체 센서 네트워크에서 데이터 저장과 질의 처리를 수행하면서 가장 높은 통신비용을 갖는 센서 노드의 통신비용인 Hotspot 통신비용을 비교 평가하였다. 실험 조건으로 전체 네트워크 크기를 600m*600m, 센서 노드의 범위를 50m, 그리고 센서 노드 개수를 600개로 설정하고 센서 노드에서 데이터와 질의의 발생 비율이 40:1에서부터 5:3까지의 비율 변화에 따른 Hotspot 통신비용을 비교 평가하였다. Hotspot 통신비용에 대한 실험 결과는 <그림 12>와 같다.



<그림 12> Hotspot 통신비용

<그림 12>에서 보는 것과 같이 Hotspot 통신비용은 데이터와 질의 발생 비율이 40:1일 때 DIM이 가장 효율적이고, 데이터와 질의의 발생 비율이 20:1일 때 Region 개수가 4개인 경우가 가장 효율적이다. 또한, 데이터와 질의의 발생 비율이 10:1일 때 Region 개수가 2개인 경우가 가장 효율적인 것으로 나타났다. 데이터와 질의의 발생 비율이 2:1일 때 Hotspot 통신비용은 DIM이 효율적인 것으로 나타났다. 그러나, 질의 발생량보다 데이터 발생량이 큰 환경을 고려한 데이터 중심 저장 방식의 센서

네트워크에서는 데이터와 질의 발생 비율이 2:1인 경우는 평가 대상에서 큰 의미를 갖진 않는다. 따라서, 데이터 중심 저장 방식의 센서 네트워크에서 데이터 발생량이 질의 발생량보다 두 배 이상 큰 경우에 비균등 영역 분할 기법을 적용하면 효율적으로 통신비용을 줄일 수 있음을 알 수 있다.

5. 결론 및 향후 연구과제

무선 센서 네트워크 환경에서 센서 노드는 제한된 에너지와 저장 공간을 가지고 있으므로 센서 노드의 과다한 에너지 소비는 센서 노드의 기능을 상실시켜 전체 센서 네트워크에 치명적인 영향을 미친다. 그러므로 무선 센서 네트워크에서 센서 노드의 에너지 효율성을 높이는 것은 매우 중요한 문제이다. 특히, 데이터 중심 저장 방식의 센서 네트워크에서는 센서 네트워크의 확장에 따른 데이터 저장 비용의 증가와 동일한 값을 갖는 데이터의 빈번한 발생에 따른 특정 센서 노드의 통신 집중으로 전체 네트워크 수명에 상당한 영향을 미치는 문제점이 있다.

이에 본 논문에서는 센서 네트워크의 확장에 따른 센서 노드 에너지의 효율적인 소비를 위한 비균등 영역 분할 기법을 제안하였다. 비균등 영역 분할 기법은 센서 네트워크를 센서 노드 개수에 따라 서로 다른 크기의 Region들로 분할하고 분할된 각 Region 내에서 측정된 데이터를 해당 Region 내에 저장 및 관리함으로써 센서 네트워크의 확장 시 데이터 저장 비용을 줄일 수 있다. 또한, 분할 Region 개수를 네트워크 크기, 센서 노드 개수, 각 센서 노드가 발생하는 데이터 양에 비례하게 증가시켜 전체 센서 노드의 에너지 소모를 최소화함으로써 센서 네트워크의 수명과 확장성을 향상시켰으며 실험을 통해 성능의 우수함을 보였다. 향후 연구과제로는 센서 네트워크에서 불규칙하게 분포되어 있는 센서 노드를 고려하여 다양

한 센서 노드 분포에 따른 효율적인 영역 분할 방안에 대한 연구가 필요하다.

참고문헌

1. Draper. C., and Wornell, G., "Side Information Aware Coding Strategies for Sensor Networks," IEEE Journal on Selected Areas in Communications, Vol.22, No.6, 2004, pp.966-976.
2. Karp, B., and Kung, T., "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proc. of the 6th Annual International Conference on Mobile Computing and Networking, 2000, pp.243-254.
3. Madden, S., Franklin, M., Hellerstein, J., and Hong, W., "TAG: A Tiny Aggregation Service for Ad-hoc Sensor Networks," Proc. of the 5th Symposium on Operating System Design and Implementation, 2002, pp.131-146.
4. Li, X., Kim, Y., Govindan, R., and Hong, W., "Multi-Dimensional Range Queries in Sensor Networks," Proc. of the 1st International Conference on Embedded Networked Sensor Systems, 2003, pp. 63-75.
5. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson, J., "Wireless Sensor Networks for Habitat Monitoring," Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, 2002, pp.88-97.
6. 전상훈, 홍동숙, 김동오, 김정준, 한기준, "텔레메틱스를 위한 센서 데이터 서비스 시스템 개발," 한국공간정보시스템학회 추계학술대회 논문집, 31권2호, 2005, pp.141-146.

7. Ratnasamy, S., Karp, B., Yin, L., Yu, F., Estrin, D., Govindan, R., and Shenker, S., "GHT: A Geographic Hash Table for Data-Centric Storage in Sensornets," Proc. of the ACM Workshop on Sensor Networks and Applications, 2002, pp.78-87.

8. Ratnasamy, S., Estrin, D., Govindan, R., Karp, B., and Shenker, S., "Data-Centric Storage in Sensornets," Proc. of the 1st ACM SIGCOMM Workshop on Hot Topics in Networks, Vol.33, No.1, 2003, pp.137-142.

9. Ratnasamy, S., Karp, B., Shenker, S., Estrin, D., Govindan, R., Yin, L., and Yu, F., "Data-Centric Storage in Sensornets with GHT, A Geographic Hash Table," Proc. of the Mobile Networks and Applications, Vol.8, No.4, 2003, pp.427-442.

10. Greenstein, B., Estrin, D., Govindan, R., Ratnasamy, S., and Shenker, S., "DIFS: A Distributed Index for Features in Sensor Networks," Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, 2003, pp.163-173.

11. Sharifzadeh, M., and Shahabi, C., "Supporting Spatial Aggregation in Sensor Network Databases," Proc. of the 12th ACM International Symposium on Advances in Geographic Information Systems, 2004, pp.166-175.

12. 이해준, 고양우, 이동만, "센서 네트워크를 위한 휴간 순서 번호를 이용하는 신뢰성 있는 데이터 전송 지원 기법," 한국정보과학회 학술발표 논문집, 32권2호, 2005, pp. 178-180.

13. 임용훈, 정연돈, 김명호, "데이터 기반 센서 네트워크에서 다차원 영역 질의를 위한 동적 데이터 분산," 한국정보과학회 논문지, 33권 1호, 2006, pp.32-41.

14. 강홍구, 김정준, 홍동숙, 한기준, "센서 네트워크에서 동적 영역 분할을 이용한 다차원 범위 질의 인덱스," 한국정보과학회 학술발표 논문집, 31권2호, 2006, pp.52-54.

강홍구

2002년 건국대학교 컴퓨터공학과(공학사)
 2004년 건국대학교 대학원 컴퓨터공학과(공학석사)
 2004년~현재 건국대학교 대학원 컴퓨터공학과 박사과정
 관심분야: 공간 데이터베이스, GIS, LBS, USN, 센서 데이터베이스

전상훈

2001년 한남대학교 전자공학과(공학사)
 2001년~2002년 양재 마이크로 사원
 2002년~2004년 i-TEC Technology 광전자연구소 사원
 2006년 건국대학교 대학원 컴퓨터공학과(공학석사)
 2006년~현재 삼성전자 반도체총괄 MEMORY사업부 Flash S/W그룹 선임
 관심분야: 공간 데이터베이스, GIS, LBS, 센서 데이터베이스, Hardware Platform

홍동숙

1999년 건국대학교 컴퓨터공학과(공학사)
 2001년 건국대학교 대학원 컴퓨터공학과(공학석사)
 2000년~2003년 쌍용정보통신 모바일/GIS 기술팀
 2003년~현재 건국대학교 대학원 컴퓨터공학과 박사과정
 관심분야: 데이터베이스, 이동체 데이터베이스, 유비쿼터스 컴퓨팅

한기준

1979년 서울대학교 수학교육학과(이학사)
 1981년 한국과학기술원(KAIST) 전산학과(공학석사)
 1985년 한국과학기술원(KAIST) 전산학과(공학박사)
 1990년 Stanford 대학 전산학과 Visiting Scholar
 1985년~현재 건국대학교 컴퓨터공학부 교수
 2004년~현재 한국정보시스템관리사협회 회장
 관심분야: 공간 데이터베이스, GIS, LBS, 텔레매틱스, 정보시스템 감리