

논문 2010-05-25

무선 센서 네트워크에서 동적 클러스터 유지 관리 방법을 이용한 에너지 효율적인 주기적 데이터 수집

(An Energy-Efficient Periodic Data Collection using Dynamic Cluster Management Method in Wireless Sensor Network)

윤상훈, 조행래*

(Sanghun Yun, Haengrae Cho)

Abstract : Wireless sensor networks (WSNs) are used to collect various data in environment monitoring applications. A spatial clustering may reduce energy consumption of data collection by partitioning the WSN into a set of spatial clusters with similar sensing data. For each cluster, only a few sensor nodes (samplers) report their sensing data to a base station (BS). The BS may predict the missed data of non-samplers using the spatial correlations between sensor nodes. ASAP is a representative data collection algorithm using the spatial clustering. It periodically reconstructs the entire network into new clusters to accommodate to the change of spatial correlations, which results in high message overhead. In this paper, we propose a new data collection algorithm, name EPDC (Energy-efficient Periodic Data Collection). Unlike ASAP, EPDC identifies a specific cluster consisting of many dissimilar sensor nodes. Then it reconstructs only the cluster into subclusters each of which includes strongly correlated sensor nodes. EPDC also tries to reduce the message overhead by incorporating a judicious probabilistic model transfer method. We evaluate the performance of EPDC and ASAP using a simulation model. The experiment results show that the performance improvement of EPDC is up to 84% compared to ASAP.

Keywords : Sensor network, Energy efficiency, Data collection, Probabilistic model

1. 서론

최근 무선 통신 기술과 전자 기기의 발전으로 작고 고성능인 센서 노드를 저비용으로 제작 가능해졌다. 센서 노드는 주위 환경 상태를 감지할 수 있는 센싱 장치, 센싱된 데이터를 처리하기 위한 처리 장치, 다른 센서 노드들과 근거리 무선 통신을 이용하여 데이터를 주고받기 위한 통신 장치, 그리고 센서 노드들의 운용을 위한 전원 장치 등을 갖

* 교신저자(Corresponding Author)

논문접수 : 2010. 08. 02., 수정일: 2010. 10. 11.,

채택확정 : 2010. 10. 27.

윤상훈, 조행래 : 영남대학교 컴퓨터공학과

※ 본 논문은 한국전자통신연구소 “영농 자동화를 위한 임베디드 시스템 구조 연구” 과제의 연구결과로 수행되었음.

추고 있다. 이러한 센서 노드들을 이용하여 무선 센서 네트워크(Wireless Sensor Network: WSN)를 구성할 수 있다 [1,2].

센서 노드의 전원 장치는 건전지와 같은 제한된 전원으로 구성되는 것이 일반적이다. 센서 노드의 전원을 교체 혹은 재충전하기 어려울 경우, 제한된 전원은 심각한 문제가 된다. 또한 센서 노드는 근거리 무선 통신을 멀티-홉 방식으로 연결하여 원거리 통신을 수행한다. 이러한 통신방식은 대역폭이 좁고 반응 속도가 느리며 패킷 소실과 전력 소비가 많은 문제가 있다 [3-8].

환경 감시를 위한 WSN을 구성할 경우 많은 센서 노드를 이용한다. 대규모 WSN에서 데이터 수집은 각 노드에서 환경 감지를 수행하는 과정과 감지된 데이터를 사용자에게 전달하는 과정에서 에너지 소비가 많다. 에너지 소비로 수명을 다한 센서 노드들은 그 기능을 상실하므로 에너지 효율적으로 구

동하여 노드의 수명을 늘이는 것이 중요하다. 또한 감지되는 데이터는 시공간적으로 변화하는 환경, 사용자의 요구에 따라 변화하는 작업, 에너지 수준 및 위치, 연결 상태에 따라 변화하는 네트워크를 고려하여 수집하는 데이터의 신뢰도도 보장해 주어야 한다.

WSN을 이용하여 데이터를 수집하는 방법은 이벤트 기반 데이터 수집 방법과 주기적인 데이터 수집 방법으로 분류된다. 이벤트 기반 데이터 수집은 각 센서 노드가 이벤트를 감지하여 베이스 스테이션(Base Station: BS)로 이를 알려주는 방법이다. 이 방법은 BS로 전송되는 데이터가 적다는 장점이 있지만, 센서 노드가 이벤트를 감지하기 때문에 노드에서 이벤트 감지를 위한 복잡한 알고리즘 수행이 요구되고, 노드에서 감지된 데이터의 전송이 이루어지지 않기 때문에 자세한 분석이 불가능하다는 단점이 있다. 반면 주기적 데이터 수집 방법은 주기적으로 수집된 데이터를 BS로 전송하기 때문에 보다 자세한 데이터의 분석이 가능하지만, 많은 데이터 전송으로 에너지 소비가 많다.

주기적 데이터 수집 방법의 단점을 극복하기 위해 데이터의 시공간적 인접성을 고려한 클러스터링 알고리즘들이 연구되었다 [3,4]. 클러스터링 알고리즘은 공간적으로 인접한 노드들은 동일한 시간에 유사한 값을 가진다는 내용을 기반으로 근거리의 노드들을 중심으로 클러스터를 구축하고 각 클러스터에서 데이터를 수집하여 BS로 전송하는 방법을 사용하였다. 이는 주기적 데이터 수집에서 데이터 전송량을 줄였으나 클러스터가 환경변화에 적응하지 못하여 시간흐름에 따라 데이터의 정확도가 떨어졌다.

본 논문에서는 대규모 WSN에서 데이터 수집을 수행할 때 에너지 소비를 줄여 네트워크 수명을 늘리고 동적으로 변화하는 환경에 적응하여 데이터의 신뢰도를 보장할 수 있는 에너지 효율적인 주기적 데이터 수집(Energy-efficient Periodic Data Collection: EPDC) 알고리즘을 제안한다. EPDC 알고리즘은 네트워크 환경과 센서 노드의 상태에 따라 동적으로 선택된 노드들을 이용하여 환경을 감지하고 그 결과를 베이스 스테이션(Base Station: BS)으로 전송한다. 선택되지 않은 노드들의 값은 확률모델과 선택된 노드의 값을 이용하여 그 값을 추정한다. 확률모델은 센서 노드들의 시공간적 연관성을 이용하여 동적으로 구성되며 시스템 및 환경 변화에 맞추어 수정된다. 또한 환경 변화에 따라 추정 노드의 데이터 신뢰도를 향상시키기 위해 확률

모델을 동적으로 분할한다. 추가적으로 확률모델의 전송에서 에너지 효율적인 동작을 위해 이전 확률모델과 비교하여 변화가 많은 값들만 전송한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대하여 살펴본다. 3장에서 EPDC 알고리즘의 세부 동작을 자세히 설명한다. 4장에서는 EPDC 알고리즘의 이론적 성능에 대하여 논의하며, 5장에서는 EPDC 알고리즘의 성능을 실험을 통하여 분석한다. 마지막으로 6장에서 결론을 맺는다.

II. 관련연구

WSN에서 에너지를 절약하기 위한 많은 연구가 이루어졌다. 최근에는 공간적 연관성을 이용한 클러스터링 알고리즘이 연구되고 있다. 대표되는 알고리즘으로 Energy Efficient Data Collection (EEDC)[4]과 Adaptive Sampling Approach to Data Collection(ASAP) [3]가 있다. 두 알고리즘은 에너지 소비를 줄이기 위해 소수의 노드에서만 환경 감지를 수행한다. 환경 감지를 수행하지 않은 노드의 값을 알기 위해서는 감지를 수행한 노드의 값을 이용하여 확률적으로 유추한다. 노드의 값을 유추하기 위해 두 알고리즘은 확률모델을 필요로 한다. 이를 생성하기 위해 EEDC는 BS에서 모든 클러스터의 정보를 수집하고, 확률모델을 생성한다. 하지만, ASAP는 클러스터의 각 헤더에서 필요한 데이터를 수집하고, 확률모델을 계산한다. 그 후, 확률모델을 BS로 전송한다. EEDC와 같이 BS에서 모든 클러스터의 정보를 수집하고 확률모델을 생성하는 방법은 확률모델의 전송비용을 아낄 수 있지만, BS에 많은 작업을 부과하게 된다. ASAP와 같이 클러스터 헤더에서 확률모델을 생성하는 방법은 확률모델을 BS로 전송하는 비용이 필요하지만, 여러 노드에서 작업을 나누어 수행하므로 BS의 작업을 줄일 수 있다. 본 논문에서는 클러스터 헤더에서 확률모델을 생성하는 방법을 채택했다.

본 논문에서 채택한 ASAP는 확률모델을 BS로 전송하는데 값-단위로 하나씩 전송한다. 이는 많은 데이터 전송을 발생시킨다. 또한 ASAP의 확률모델은 고정된 주기마다 재구성하여 시시각각 변화하는 환경에 즉각적으로 반응하지 못하여 추정된 데이터의 정확도가 떨어진다는 단점이 존재한다. 본 논문에서 제안하는 알고리즘은 생성된 확률모델을 전송함에 있어 이전 확률모델과 비교하여 변화된 부분을 전송하여 통신비용을 줄이고, 시시각각 변화하는

환경에 적응하여 동적으로 클러스터 분할을 수행하여 데이터 수집에 정확도를 높일 수 있는 방법을 제안한다.

III. EPDC 알고리즘

본 절에서는 환경 변화에 적응하며 에너지 효율적으로 데이터를 수집하는 EPDC 알고리즘을 설명한다.

1. 시스템 모델

EPDC 알고리즘은 두 계층의 네트워크 모델을 사용한다. 첫 번째 계층은 N개의 센서 노드와 이를 연결하는 데이터 수집 트리 계층이다. 센서 노드는 BS로 데이터를 수집할 때 이 계층을 이용한다. 데이터 수집 트리는 [5,6]등의 알고리즘을 이용하여 쉽게 구성할 수 있다. 두 번째 계층은 WSN을 중복되지 않는 영역으로 분할하는 클러스터 계층이다. 각 노드는 하나의 클러스터에 포함되며 하나의 클러스터는 하나의 헤더를 가진다. 클러스터는 클러스터 트리를 구성하며 이는 헤더와 노드간의 통신에 사용된다. 클러스터 헤더는 클러스터 트리의 최상위 노드이다. 클러스터의 구성은 [3]의 알고리즘을 이용한다. 클러스터는 그 크기에 따라 하나 이상의 샘플러 노드를 가진다. 본 논문에서 WSN는 클러스터가 구성되어 있고, 샘플러 노드가 선정되어 있다고 가정한다.

2. 데이터 수집 방법

WSN은 일정 주기마다 클러스터를 생성한다. 클러스터는 가까운 거리에 있으며 서로 연관성 있는 데이터를 가지는 노드들로 구성되며, 구성된 클러스터 내에서 잔여 에너지가 많은 노드를 클러스터 헤더로 선정한다. 각 클러스터는 크기에 따라 잔여 에너지가 많은 하나 이상의 샘플러 노드를 선택한다. 샘플러 노드는 일정 주기마다 환경을 감지하고 데이터를 BS로 전송한다. BS는 이 데이터와 확률 모델을 이용하여 비-샘플러 노드의 값을 추정한다. 확률 모델은 모든 노드에 대하여 강제적으로 수집된 데이터를 기반으로 클러스터 헤더에서 생성한다. 강제 수집은 일정 주기마다 모든 노드에서 수행되며, 클러스터 헤더는 수집된 데이터들의 시계열을 저장한다. 생성된 확률 모델은 BS로 전송된다.

3. 확률 모델

클러스터 헤더는 강제 수집된 데이터를 기반으로 확률 모델을 생성한다. 확률 모델은 평균과 공분산을 이용한 상관계수 매트릭스로 구성된다. 센서 노드 p_u 와 p_v 의 상관계수 $C[p_u, p_v]$ 는 다음 식(1)과 식(2)를 이용하여 계산한다 [3].

$$Y[p_u, p_v] = (D_i[p_u] - E[D_i[p_u]]) * (D_i[p_v] - E[D_i[p_v]])^T \dots\dots\dots \text{식(1)}$$

$$C[p_u, p_v] = \frac{Y[p_u, p_v]}{L * \sqrt{\text{Var}(D_i[p_u])} * \sqrt{\text{Var}(D_i[p_v])}} \dots\dots\dots \text{식(2)}$$

(E 는 평균, Y 는 공분산, L 은 시계열의 수, $D_i[p_u]$ 는 클러스터 i 의 노드 p_u 의 시계열 벡터)

EPDC 알고리즘은 계산된 확률모델을 BS로 전송할 때 에너지 효율적으로 동작한다. 구체적으로, 확률모델을 BS로 전송할 때 기존 확률모델과 새로 생성된 확률모델을 비교하여 임계치 $\alpha(0 < \alpha < 1)$ 이상 변화한 확률모델 값만 선택적으로 전송한다.

확률모델 전송 알고리즘(그림 1)은 현 주기에 생성된 확률모델(C_{new})과 이전 주기에 생성된 확률모델(C_{old})의 값을 비교하여 $\alpha\%$ 이상 변화가 있을 경우 D-매트릭스에 true를 기록한다(4,5번째 줄). 즉, D-매트릭스에 true가 기록된 위치의 C_{new} 값들이 전송될 대상이다.

확률모델의 특정 값을 전송하려면 그 값과 함께 어느 노드와 관련된 확률모델인지를 알려주기 위해 두 노드의 식별자가 함께 포함되어야 한다. 만약 하나의 노드와 관련된 여러 확률모델 값들이 변경되었다면 동일한 식별자를 중복으로 보내는 것보다 그 노드의 확률모델 값들을 묶어 동시에 보내는 것이 효율적이다. 예를 들면 노드 i 의 확률모델이 모든 노드에 대하여 임계치 $\alpha\%$ 이상 변경되었다고 가정하자. 이 값들을 각각 전송하면 ($i, x, value$)쌍을 N번 전송하여야 한다. 하지만 이를 노드 i 의 확률모델로 전송한다면 ($i, value[N]$)을 한번 전송하면 된다. 즉, 반복적인 식별자의 전송을 줄일 수 있다. 이 방법은 변화가 적은 확률모델 값도 전송될 가능성이 있으므로 임계치 $\beta(0 < \beta < 1)$ 를 사용하여 전송될 노드 수에 따른 적절한 전송 수준을 계산해야 한다.

```

Cold : Old Correlation Coefficient Matrix
Cnew : New Correlation Coefficient Matrix
D : Discrepancy between two matrix
N : Number of Nodes
α : Dissimilarity threshold ( 0 < α < 1 )
β : Update threshold ( 0 < β < 1 )

```

```

1. begin
2. for i = 1 to N
3.   for j = i to N
4.     if !(1-α < |Cold[i,j]/Cnew[i,j]| < 1+α)
5.       D[i,j] = D[j,i] = true
6.     else
7.       D[i,j] = D[j,i] = false
8.     end if
9.   end for
10. end for
11. end for
12.
13. for i = 1 to N
14.   sum = 0
15.   for j = 1 to N
16.     if ( D[i,j] == true )
17.       sum = sum+ 1
18.     end if
19.   end for
20.   if ( sum > β*N )
21.     send Cnew[i,•] to BS
22.     for j = 0 to N
23.       D[i,j] = D[j,i] = false
24.     end for
25.   end if
26. end for
27.
28. for i = 1 to N
29.   for j = i to N
30.     if ( D[i,j] == true )
31.       send Cnew[i,j] to BS
32.     end if
33.   end for
34. end for
35. end

```

그림 1. 확률 모델 전송 알고리즘
 Fig. 1. Probabilistic model transfer algorithm

효율적인 데이터 전송을 위하여 데이터의 전송은 두 가지 방법을 모두 사용한다. 한 노드의 환경이 급격히 변화할 경우 이 노드와 다른 노드사이의 상관계수가 동시에 급격히 변화한다. 이는 D-매트릭스의 특정 열에 기록된 true값의 개수로 유추할 수 있다. true값의 개수가 β*N개 이상일 경우 한 노드의 확률모델(C[i,•] : i번째 노드의 확률모델)을 모두 전송하는 열 전송을 사용하여 BS에 저장되어 있는 확률모델을 수정한다(13~26번째 줄). 그렇지 않고 소수 노드와의 상관계수가 변화되었을 때는 변화된 값만을 전송하여 확률모델을 수정한다

(28~35번째 줄).

4. 클러스터 분할

확률모델을 이용하여 추정된 데이터는 샘플러 노드와 비-샘플러 노드사이의 상관관계에 따라 정확도가 달라진다. 즉, 상관관계가 높은 노드를 이용할 경우 정확한 데이터를 추정할 수 있다. 그러므로 클러스터는 상관관계가 높은 노드들로 구성되어야 하며, 변화하는 환경에 맞추어 상관관계를 유지하여야 한다.

클러스터 내의 상관관계가 낮아졌을 때 이를 높이는 방법은 전체 네트워크에서의 클러스터 재구성과 클러스터 분할이 있다. 클러스터 재구성은 WSN 전 영역에 대하여 강제 수집 과정을 거쳐 클러스터를 새로이 구성하는 방법으로 수행비용이 비싸다. 특히, 하나의 클러스터에서만 상관계수가 낮아졌다면 클러스터 재구성은 적절하지 못하다. 이 경우에는 클러스터 내에서 상관계수를 높이는 방법이 요구된다.

본 절에서는 클러스터의 상관계수를 높이기 위한 클러스터 분할 알고리즘을 제안한다. 제안하는 알고리즘은 클러스터 헤더가 이미 구축된 확률모델을 이용하여 클러스터의 상관계수가 낮아지는 것을 동적으로 감지하여 클러스터를 분할한다. 클러스터 분할 알고리즘은 세부적으로 나누어진다. 먼저 클러스터의 분할 시점을 감지하는 과정과 상관계수가 높은 노드들을 모으는 과정, 마지막으로 가장 효과적인 분할 위치를 찾아 클러스터를 분할하는 과정이다.

클러스터 분할 시점을 계산하는 알고리즘은 그림 2와 같다. 클러스터 분할은 확률모델의 모델계수(Model Factor: MF)가 낮아졌을 때 수행한다. MF는 확률모델의 정확도를 나타내기 위한 값이다. 확률모델은 상관계수로 구성된다. 상관계수는 -1 ~ 1 사이의 값을 갖으며, 절대값이 1에 가까울수록 밀접한 관계이다. 이를 이용하여 MF는 각 값의 절대값을 더하여 표현한다. MF를 식으로 표현하면 식(3)과 같다.

$$MF = \sum_{i=1}^N \sum_{j=1}^N |C[p_i, p_j]| \quad \dots\dots \text{식(3)}$$

(p_i 는 i번째 노드, C는 상관계수 매트릭스, MF는 모델계수)

MF의 계산은 4~9번째 줄에서 수행한다. 특정

클러스터가 현재의 MF(Cnew의 MF)와 이전 전송 시점에서의 MF(Cold의 MF)와 비교하여 임계치-δ (0<δ<1)보다 더 많은 변화가 이루어졌다면 클러스터의 분할을 수행한다(11~13번째 줄).

Cold : Old Correlation Coefficient Matrix
 Cnew : New Correlation Coefficient Matrix
 N : Number of Nodes
 δ : Weight Factor (0 < δ < 1)

```

1. begin
2.   sumN = 0
3.   sumO = 0
4.   for i = 1 to N
5.     for j = 1 to N
6.       sumN = sumN + |Cnew[pi, pj]|
7.       sumO = sumO + |Cold[pi, pj]|
8.     end for
9.   end for
10.
11.   if ( 1-δ < sumO/sumN < 1+δ )
12.     cluster division time
13.   end if
14. end
    
```

그림 2. 클러스터 분할 시점
 Fig. 2. Cluster division time

확률모델에서 상관계수가 높은 노드들을 모으는 상관계수 클러스터링 알고리즘은 그림 3과 같다. 임의의 한 노드를 이미 배치된 노드들의 사이에 위치시킬 수 있는 모든 배치에 대하여 친밀도를 계산한다(6번째 줄). 그 중 가장 높은 친밀도를 갖는 위치(loc 변수)를 기억하여 배치할 준비를 한다(7~10번째 줄). 위에서 사용하는 노드간 친밀도는 다음 식(4)로 계산한다.

$$cont(p_i, p_k, p_j) = C[p_i] * C[p_k]^T + C[p_k] * C[p_j]^T - C[p_i] * C[p_j]^T$$

..... 식(4)

(p_i는 i번째 노드, C[p_i]는 p_i의 상관계수 벡터,

cont(p_i, p_k, p_j)는 p_i, p_k, p_j사이의 친밀도)

위의 식(4)는 pi와 pj사이에 pk를 배치시켰을 때, pi와 pk, pk와 pj는 인접 노드가 되어 친밀도가 높아지고 pi와 pj는 멀어져 친밀도가 낮아지는 것을 수식으로 표현한 것이다. 즉, 두 노드 사이에 다른 노드를 삽입하였을 때 서로간의 친밀도를 표현한다.

높은 친밀도를 갖는 노드들을 서로 인접시키기 위하여 loc 변수에 기억되어 있는 위치로 노드를 이동시킨다. 이 노드의 이동은 행과 열의 방향으로 수

행한다(13~17번째 줄, 19~23번째 줄, C[•,index]: index번째 열의 모든 원소, C[index,•]: index번째 행의 모든 원소).

C : Correlation Coefficient Matrix
 CC : Clustered Correlation Coefficient Matrix
 N : Number of Nodes

```

1. begin
2.   for index = 3 to N
3.     loc = index
4.     best = cont(ploc-1, ploc, 0)
5.     for i = 1 to index-1
6.       con = cont(pi-1, pindex, pi)
7.       if ( best < con )
8.         loc = i
9.         best = con
10.      end if
11.    end for
12.
13.    temp = C[•,index]
14.    for j = index to loc by -1
15.      CC[•,j] = CC[•,j-1]
16.    end for
17.    CC[•,loc] = temp
18.
19.    temp = C[index,•]
20.    for j = index to loc by -1
21.      CC[j,•] = CC[j-1,•]
22.    end for
23.    CC[loc,•] = temp
24.
25.    index = index + 1
26.  end for
27. end
    
```

그림 3. 노드 클러스터링 알고리즘
 Fig. 3. Node clustering algorithm

그림 4의 상관계수 분할 알고리즘은 확률모델을 앞선 알고리즘으로 클러스터링 하였을 때, 최적화된 분할 위치를 찾고 분할을 수행하는 알고리즘이다. 확률모델은 대각선상에 있는 임의의 위치를 기준으로 4영역으로 나눈다. 각 영역은 클러스터된 상단 사분위(Clustered Top Quarter: CTQ), 클러스터된 하단 사분위(Clustered Bottom Quarter: CBQ), 남은 사분위(Other Quarter: OQ)*2 이다. CTQ와 CBQ는 새로이 생성될 클러스터의 확률모델로 상위-원-사분위와 하위-오른-사분위에 위치한다. OQ는 클러스터의 분할로 서로 다른 클러스터의 노드가 되어 버려질 값으로 상위-오른-사분위와 하위-원-사분위에 위치한다.

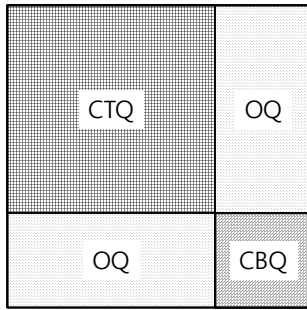


그림 5. 확률모델의 분할 형태
Fig. 5. Division pattern of probabilistic model

$$CTQ_k = \sum_{i=1}^k \sum_{j=1}^k C[p_i, p_j] \quad \dots\dots\dots \text{식(5)}$$

$$CBQ_k = \sum_{i=k+1}^n \sum_{j=k+1}^n C[p_i, p_j] \quad \dots\dots\dots \text{식(6)}$$

$$OQ_k = \sum_{i=1}^k \sum_{j=k+1}^n C[p_i, p_j] \quad \dots\dots\dots \text{식(7)}$$

$$= \sum_{i=k+1}^n \sum_{j=1}^k C[p_i, p_j]$$

그림 5는 확률모델의 분할 형태를 나타내고 있다. 분할된 각 영역을 수식으로 표현하면 식(5)~식(7)과 같다. 클러스터 분할은 상관계수를 높게 유지하기 위한 것이므로 CTQ와 CBQ의 값은 크게 만들고 OQ의 값은 작게 만들 수 있는 위치에서 수행되어야 한다. 이를 클러스터 분할 포인트(Cluster Divide Point: CDP)라 하며, 식(8)과 같이 표현한다.

$$CDP = CTQ * CBQ - OQ^2 \quad \dots\dots\dots \text{식(8)}$$

최적의 분할 위치를 찾기 위해 확률모델의 대각선상의 모든 위치에 대하여 식(8)을 계산한다(8-9 번째 줄, mp 변수 지정). 또한 shift matrix 함수를 이용하여 확률모델을 행/열 방향으로 한 칸씩 이동한 후 다시 대각선상의 모든 위치에 대하여 식(8)을 계산한다(16번째 줄, mf 변수 지정). 이 과정으로 확률모델은 가능한 모든 모양(mf 변수)과 위치(mp 변수)에서 최적의 분할 위치를 계산하고, 분할을 수행한다(18번째 줄). 분할이 이루어지면 BS로 클러스터의 분할을 알려주고, 분할된 확률 모델을 전송 알고리즘을 이용하여 전송한다.

CC : Clustered Correlation Coefficient Matrix
FC : Set of Fragments

```

1. begin
2. calculate CTQN-1, CBQN-1, OQN-1
3. best =
   (CTQN-1*CBQN-1)-(OQN-1*OQN-1)
4. mf = 1
5. mp = N-1
6. for k = 1 to N-1
7.   for i = 1 to N-1
8.     calculate CTQi, CBQi, OQi
9.     z = (CTQi*CBQi)-(OQi*OQi)
10.    if ( best < z )
11.      best = z
12.      mf = k
13.      mp = i
14.    end if
15.  for end
16.  shift matrix(CC)
17. end for
18. FC =
   { ∏i=1mf ∏j=1mf C[pi, pj], ∏i=mf+1N ∏j=mf+1N C[pi, pj] }
   at mp matrix position
19. end
    
```

그림 4. 클러스터 분할 알고리즘
Fig. 4. Cluster division algorithm

IV. 논 의

1. 확률모델 전송 알고리즘

클러스터 전송 알고리즘의 성능을 살펴보기 위해 N개의 노드로 구성된 클러스터를 고려해보자. 클러스터 헤더는 강제 수집과정을 통하여 확률모델을 생성한다. N개의 노드로 구성된 클러스터의 확률모델은 N*N 크기의 매트릭스로 구성된다. 확률모델은 대각선을 기준으로 서로 대칭인 매트릭스이다. 이 매트릭스를 값-단위로 전송하려면 각 값마다 관련된 두 노드 식별자를 추가하여 전송을 수행한다. 즉, 확률모델 전체를 전송하기 위해 (sizeof(value) + sizeof(id) + sizeof(id))*(N + 1)*N/2)의 데이터 전송이 발생한다. 확률모델의 값(value)의 크기를 1, 노드 식별자(id)의 크기를 1로 가정하면 1.5*N*(N + 1)의 전송이 발생한다. 이를 노드-단위 확률모델의 전송으로 전송한다면 한 노드의 확률모델을 N개의 벡터로 전송하므로 N*(sizeof(value)*N + sizeof(id))로 모두 전송할 수 있다. 이는 N*(N + 1)의 데이터 전송이 발생한다. 이 두 값을 비교하면 값-단위 전송이 노드-단위 전송보다 1.5배의 데이터 전송을 사용한다.

만약 확률모델에서 변화된 값만 전송한다면 값-단위로 전송되는 것이 더 이득일 수 있다(변화된 값이 소수일 경우). 한 노드에 관계된 모든 확률모델 전송에 사용된 데이터 전송은 $N + 1$ 이고, 이를 각 값별로 전송할 경우에는 $3N$ 이다. 한 노드의 확률모델을 전송함에 있어 $N/3$ 개의 값을 값별로 전송한다면 노드의 확률모델은 동시에 전송하는 것보다 이득을 볼 수 있다. 즉, 한 노드의 확률모델에서 변화된 값이 $N/3$ 개보다 적다면 값-단위 전송이 이득이다.

본 논문에서 제안하는 확률모델 전송 알고리즘은 위의 두 관점을 모두 고려하여 작성되었다. 확률모델의 값이 많이 변화된 노드의 확률모델은 노드-단위의 전송을 수행하며 값의 변화가 적은 노드는 값-단위의 전송을 수행하여 최소의 데이터 전송으로 확률모델의 전송을 수행한다.

2. 클러스터 분할 알고리즘

클러스터 분할 알고리즘의 성능을 살펴보기 위해 N 개의 노드로 구성된 클러스터를 고려해보자. 이 클러스터의 확률모델 전송에는 4.1절의 설명과 같이 $1.5 * (N * (N + 1))$ 의 데이터 전송이 필요하다(값-단위 확률모델 전송). 이 클러스터를 2개의 작은 클러스터로 나누어 확률 모델을 전송해 보자. 클러스터를 분할하는 여러 형태 중 두 클러스터의 크기를 균등하게 나누는 $(N/2, N/2)$ 의 분할과 크기의 차이가 가장 많이 나는 $(N-1, 1)$ 의 분할을 비교해 보자. 먼저, $(N/2, N/2)$ 분할의 확률 모델은 $N/2 * N/2 = N * N / 4$ 크기의 확률 모델 두 개의 전송과 동일하다. 그러므로 $1.5 * N / 2 * (N/2 + 1)$ 의 전송비용이 필요하다. 즉, 분할된 두 확률모델의 전송 비용 $1.5 * N * (N/2 + 1)$ 과 기존 확률모델 전송 비용 $(1.5 * N * (N + 1))$ 을 비교하면 약 50%에 해당하는 데이터 전송으로 확률모델을 전송할 수 있다. $(N-1, 1)$ 분할의 전송은 $1 * 1$ 크기와 $(N-1) * (N-1)$ 크기 확률 모델 2개를 전송하는 것이므로 $3 * 1 * 1$ 과 $1.5 * (N - 1) * N$ 의 전송 비용이 소비된다. 전체 전송 비용은 두 식의 합인 $1.5 * (N - 1) * N + 3$ 이 되고 이는 기존 전송비용과 비교하면 약 $3N - 3$ 만큼 데이터 전송을 줄일 수 있다. 이 과정에서 살펴보면 클러스터 분할은 확률모델의 전송시 소비되는 에너지를 아낄 수 있다는 것을 알 수 있다. 또한 분할된 클러스터의 크기가 비슷할 때 최대 50%의 이득이 있음을 알 수 있다.

본 논문에서 제안하는 클러스터 분할 알고리즘은 이러한 특성을 고려하여 클러스터의 분할시 두

클러스터의 크기를 균등하게 분할할 수 있도록 작성되었다.

V. 성능 평가

본 논문에서 제안하는 알고리즘의 성능을 분석하기 위하여 모의실험을 수행하였다. 모의실험은 C언어를 이용하여 WSN 실험 환경을 구현하였다. 본 절에서는 개발한 실험 환경과 실험 결과를 분석하도록 한다.

1. 실험 환경

센서 네트워크에서 데이터 수집은 다음 과정을 통하여 수행한다. 먼저 센서가 물리적 환경을 감지한다. 다음으로 센서에서 감지된 값을 노드(연산장치)에서 적절한 연산을 통하여 필요한 데이터를 생성한다. 마지막으로 연산과정으로 생성된 데이터를 멀티-홉 방식을 이용하여 BS로 전송한다. 이 과정에서 소비되는 에너지들 중, 센서가 환경을 감지하는데 소비하는 에너지와 노드에서 명령어를 수행하는데 소비하는 에너지는 데이터를 멀티-홉으로 BS에 전송하는데 소비하는 에너지보다 아주 적다 [1,2]. 그러므로 센서 네트워크의 동작에서 에너지를 아끼는 가장 효과적인 방법은 데이터 전송량을 줄이는 것이다.

본 논문에서는 제안한 EPDC 알고리즘의 성능 분석을 위해 가장 대표적인 클러스터형 데이터 수집 알고리즘인 ASAP과 비교하였다. 주요 성능 지수는 확률모델의 유지를 위해 전송되는 데이터의 양이다. 확률모델의 전송에는 확률모델의 값과 함께 그 확률모델이 발생한 노드의 정보를 함께 전송하여야 한다. 그러므로 하나의 확률모델 값은 두 노드의 식별자와 함께 전송된다. 실험에서 확률모델의 자료형과 노드 식별자의 자료형은 크기가 동일하다고 가정하였다. 노드의 수는 최소 25개의 노드부터 최대 900개의 노드까지 사용하였다. 강제 수집 수행시 데이터의 시계열 수를 30개로 정의하여 실험하였다. 즉, 30개의 데이터를 이용하여 평균과 분산 및 상관계수를 계산하였다. 확률모델은 1~10번의 강제 수집마다 생성하였으며, 20번의 확률모델 전송에 소비되는 메시지 전송 비용을 측정하였다. 실험에 사용된 데이터는 Global Precipitation Climatology Project [9]에서 제공하는 데이터이다.

2. 결과 분석

먼저 확률모델 전송 알고리즘의 성능을 평가하기 위하여 노드 수를 5*5부터 30*30까지 증가시켰다. 확률모델의 시계열은 30개로 설정하였다. 확률모델은 업데이트 주기(Tu)마다 생성하여 BS로 전송한다. Tu는 강제 수집 주기(Tf)의 배수로 지정된다. 본 실험에서는 Tu와 Tf의 비율(Tu/Tf)을 0.1~1까지 변화시키며 실험하였다. 다른 실험 환경은 5.1절의 환경을 따른다. 실험 결과는 그림 5와 같다.

4.1절에서 논의했던 것과 같이 확률모델 전송 알고리즘(EPDC-PMTA)은 ASAP보다 적은 데이터 전송량을 보인다. 먼저 ASAP는 확률모델을 모든 값을 전송하므로 노드의 수가 늘어날수록 제곱으로 전송량이 늘어났다. EPDC-PMTA 알고리즘도 노드의 수가 늘어남에 따라 전송량이 늘어나지만 ASAP와 비교하여 25개 노드일 때 10%, 100개 노드일 때 13%, 225개 노드일 때 15%, 400개 노드일 때 16%, 625개 노드일 때 16%, 900개 노드일 때 17%정도의 데이터 전송량을 보였다. 또한, Tu/Tf에 대하여 ASAP는 변화가 없었으나 EPDC-PMTA는 Tu/Tf가 0.1일 때 22%, 0.2일 때 19%, 0.33일 때 16%, 0.5일 때 14%, 1일 때 11%의 에너지 전송량을 보였다. 평균적으로 약 16%의 전송량으로 확률모델의 전송을 완료하였다.

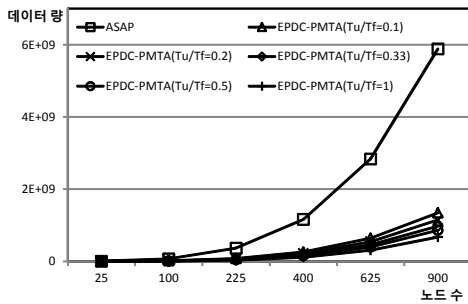


그림 5. 확률모델 전송 알고리즘의 노드수의 변화에 따른 성능

Fig. 5. Performance of probabilistic model transfer algorithm by varying the number of nodes

다음으로 확률모델 전송 알고리즘에서 α , β 의 영향을 살펴보기 위하여 α 와 β 를 0.1부터 0.1씩 증가시키며 실험을 수행하였고, 그 값들을 모두 합하여 하나의 그래프로 나타내었다. α 값에 대한 메시지 전송 비용을 확인하기 위한 실험에서는 β 값의 영향력을 배제하기 위하여 β 값을 균등하게 분포시

켰다. 반대로 β 의 영향력을 살펴보기 위한 실험에서는 α 값을 균등히 분포시켰다. 클러스터의 크기를 20*20로 설정하고, 확률모델을 생성하는 강제 수집 횟수는 5회로 지정하였다. 다른 환경 변수들은 5.1절의 설명을 따른다. 메시지 수의 측정은 20번의 확률모델 전송에서 발생한 데이터 전송량을 모두 합하였다. 그림 6은 α 값의 변화에 따른 전송량이고 그림 7은 β 값의 변화에 따른 전송량이다.

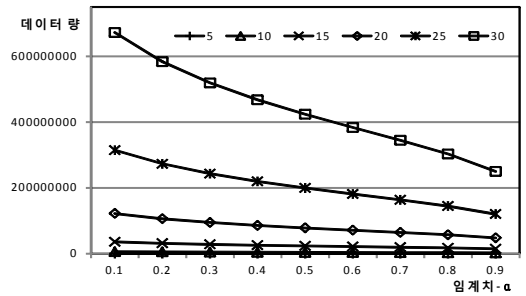


그림 6. 확률모델 전송 알고리즘의 α 값에 따른 성능
Fig. 6. Performance of probabilistic model transfer algorithm by α parameters

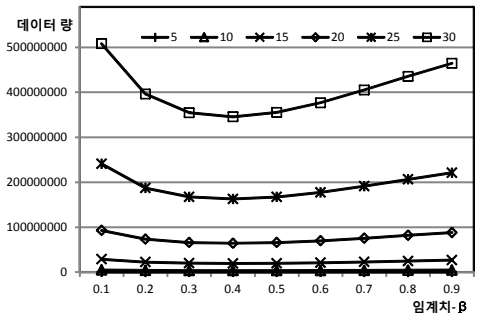


그림 7. 확률모델 전송 알고리즘의 β 값에 따른 성능

Fig. 7. Performance of probabilistic model transfer algorithm by β parameters

임계치- α 는 값이 증가할수록 확률모델에서 값의 변화에 둔감하게 된다. 즉, 많은 변화가 일어났음에도 확률모델의 전송이 이루어지지 않는 것이다. 그러므로 α 의 값은 커지면 전송되는 전송량은 줄어들어야 한다. 실험 결과 역시 α 의 값이 커질수록 전송량이 줄어드는 것을 확인할 수 있다. 임계치- β 는 4.1절에서 논의한 것과 같은 결과를 보인다. 낮은 임계치는 노드-단위의 전송을 발생시키고, 높은 임계치는 값-단위의 전송을 늘어나게 한다. β 가 낮을

때에는 소수의 값만 변화가 일어나더라도 노드-단위로 확률모형을 전송하게 되고, β 가 높을 때에는 변화된 수가 많더라도 노드-단위의 전송은 일어나지 않고 값-단위의 전송을 사용하게 된다. β 값이 낮거나 높으면 노드-단위 혹은 값-단위의 전송만 발생하게 된다. 하지만 적절한 β 값은 노드-단위의 전송과 값-단위의 전송을 최적화하여 보다 적은 전송량으로 확률모형을 전송할 수 있게 한다. 이 실험에서는 최소의 전송량 β 가 0.4일 때 최소의 전송량을 보였다.

클러스터 분할 알고리즘의 성능을 살펴보기 위해 5번의 강제 수집마다 확률 모형을 생성하며, 20번의 확률모형은 전송하는 동안 데이터 전송량을 측정하였다. 클러스터의 크기는 10*10이다. 확률모형 전송 알고리즘(EPDC-PMTA)을 위한 임계치 α 와 β 는 각각 0.8과 0.4로 설정하였다. 클러스터를 분할했을 때 발생하는 평균 데이터 전송량을 확인하기 위해 δ 를 0.1~0.3까지 균등하게 분포시켰다. 실험에서 δ 의 값이 0.4이상일 경우 클러스터의 분할이 이루어지지 않았기 때문에 그 이상은 실험결과에서 배제하였다. ASAP, 클러스터 분할 알고리즘(EPDC-CDA) 그리고 클러스터 분할 알고리즘과 확률모형 전송 알고리즘을 함께 적용한 방법(EPDC)을 비교하였다. 실험 결과는 그림 8과 같다.

그림 8에서 초기 4번째 확률모형 전송까지는 ASAP와 EPDC-CDA는 동일한 값을 보인다. 이는 확률모형이 임계치 δ 보다 낮은 변화를 유지하고 있음을 알 수 있다. 4번째 확률모형의 전송이 이루어지고 확률모형의 정확도가 낮아졌고, 그 결과 분할이 이루어졌다.

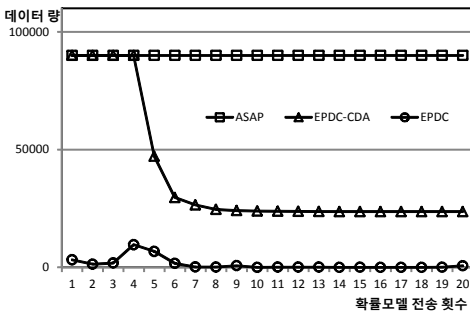


그림 8. 확률모형 분할 알고리즘의 성능
Fig. 8. Performance of probabilistic model division algorithm

확률모형의 정확도가 낮아짐은 EPDC의 전송량

이 많아지는 것으로 유추할 수 있다. 그 이후 확률모형은 안정된 전송을 보여주고 있다. 실험에서 EPDC-CDA는 ASAP와 비교하여 약 26%의 전송량으로 확률모형의 전송을 완료하였다. EPDC는 ASAP와 비교하여 약 0.7%의 전송량, EPDC-CDA에 비교하여 약 2.8%의 전송량으로 확률모형 전송을 완료하였다.

마지막으로 클러스터의 분할에서 사용자가 요구하는 확률모형 정확도를 유지하는지 확인해 보았다. 클러스터의 크기는 10*10으로 하였고, 사용자가 요구하는 확률모형의 정확도(δ)는 0.3으로 설정하였다. 확률모형의 전송을 위한 α 와 β 는 0.8과 0.4로 설정하였다. 5번의 강제수집마다 확률모형을 전송하였고, 20번의 확률모형을 전송하는 동안 실험을 수행하였다. 각 단계별 클러스터의 정확도와 분할 여부를 확인하였다. 실험 결과는 그림 9와 같다.

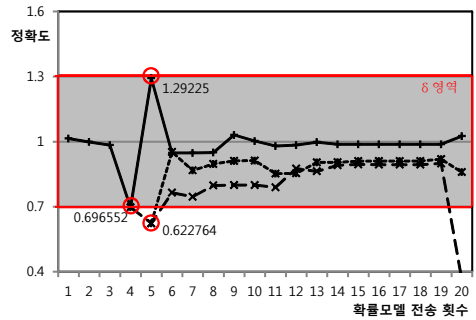


그림 9. 확률모형의 정확도와 확률모형의 분할
Fig. 9. Probabilistic model accuracy and probabilistic model division

실험에서 100개의 노드는 하나의 클러스터로 구성되어 있다. 클러스터는 하나의 확률모형을 생성하여 BS로 전송한다. 그래프에서 각 값은 확률모형의 정확도이다. 여기서 정확도는 이전에 전송된 확률모형과 현재 생성된 확률모형간의 모델계수(MF)값을 비교한 것이다. 실험에서 초기 3회의 확률모형 전송까지는 높은 정확도를 유지하고 있다. 4번째 전송에서 확률모형이 0.69로 떨어졌다. 이는 0.3으로 지정된 δ 값의 범위를 벗어난 결과로 클러스터를 분할을 수행하여야 하며, 실험에서도 분할을 수행하였다. 분할된 두 클러스터는 5번째 전송에서 각 1.29와 0.62의 정확도를 나타내었다. δ 값에 의해 1.29는 범위를 벗어나지 않았으므로 클러스터를 유지하고, 0.62는 범위를 벗어났으므로 다시 분할하여 클러스터는 총 3개가 되었다. 그 후로는 확률모형의 정확

도가 δ 의 값을 벗어나지 않고 유지되고 있다. 실험 결과 클러스터의 확률모델은 지정된 정확도를 유지하고자 노력하였다.

VI. 결 론

본 논문에서는 WSN에서 동작하는 에너지 효율적인 주기적 데이터 수집 방법을 제안하였다. 제안하는 알고리즘은 선택된 노드에서만 환경을 감지하여 통신 및 감지 비용을 줄이며, 감지하지 않은 노드의 데이터 추정을 위해 확률 모델을 사용한다. 확률모델은 클러스터 헤더에서 생성하며 에너지 효율적인 확률모델 전송 알고리즘을 이용하여 주기적으로 수정한다. 또한, 추정된 데이터의 정확도를 높이기 위하여 서로 연관성이 높은 노드들로 클러스터링하고, 환경 변화에 적응하여 클러스터 레벨에서 동적으로 클러스터 분할을 수행한다. 이러한 과정에서 제안하는 알고리즘은 주위 환경 감지와 데이터 통신에서 에너지 소비를 줄여 네트워크의 수명을 늘이며, 확률모델의 모델계수를 일정수준으로 유지함으로써 데이터의 질을 보장한다.

본 논문에서 제안하는 알고리즘에 대하여 다양한 환경에서 성능을 분석하였으며, 실험결과 확률모델의 전송 알고리즘은 기존 알고리즘과 비교하여 약 16%의 데이터 전송으로 확률모델을 전송하였으며, 클러스터 분할 알고리즘은 약 26%의 데이터 전송을 사용하여 확률모델을 전송하였다. 두 알고리즘을 함께 사용하였을 때는 약 0.7%의 데이터 전송으로 확률모델을 전송하였다. 또한 클러스터 분할 알고리즘은 사용자가 요구한 확률모델의 모델계수를 유지하며 클러스터 분할을 수행함을 확인하였다.

in sensor networks”, IEEE Transactions on Parallel and Distributed Systems, 18(12), 2007.

[4] C. Liu, K. Wu, J. Pei, “An Energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation”, IEEE Transactions on Parallel and Distributed Systems, 18(7), 2007.

[5] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “Tag: a tiny aggregation service for Ad Hoc sensor networks”, Proc. of the Fifth Symposium on Operating Systems Design and Implementation, 2002.

[6] S. Madden, R. Szewczyk, M. Franklin, and D. Culler, “Supporting aggregate queries over Ad Hoc wireless sensor networks”, Proc. of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, 2002.

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, “Wireless sensor networks for habitat monitoring”, Proc. of the First ACM Workshop on Wireless Sensor Networks and Applications, 2002.

[8] Y. Yao and J. Gehrke, “Query processing in sensor networks”, Proc. of the First Biennial Conference on Innovative Data Systems Research, 2003.

[9] Global Precipitation Climatology Project, <http://www.ncdc.noaa.gov/oa/wmo/wdcamet-ncdc.html>, Dec. 2004.

참 고 문 헌

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A Survey on sensor networks”, IEEE Communications Magazine, 40(8), 2002.

[2] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, “Energy conservation in wireless sensor networks: a survey”, Ad-Hoc Networks, 7(3), 2009.

[3] B. Gedik, L. Liu, P. S. Yu, “ASAP: an adaptive sampling approach to data collection

저 자 소 개

윤 상 훈



2006년 : 영남대학교 컴퓨터공학과 공학사.
 2008년 : 영남대학교 컴퓨터공학과 공학석사.
 2010년 : 영남대학교 컴퓨터공학과 박사수료.

관심분야 : 무선 센서네트워크, 분산데이터베이스.
 Email : sh_yun99@ynu.ac.kr

조행래



1988년 : 서울대학교
컴퓨터공학과 공학사.
1990년 : 한국과학기술원
전산학과 공학석사.
1995년 : 한국과학기술원
전산학과 공학박사.

1995년~현재 : 영남대학 컴퓨터공학과 교수.

관심분야 : 분산/병렬 데이터베이스, DBMS 개발,
트랜잭션 처리.

Email : hrcho@yu.ac.kr