Journal of Institute of Control, Robotics and Systems (2016) 22(11):984-990
http://dx.doi.org/10.5302/J.ICROS.2016.16.0178 ISSN:1976-5622 eISSN:2233-4335

무선 메쉬 센서 네트워크에서 셔플드 로우 메이져 인덱싱 기법을 활용한 데이터 수집 방법

Data Aggregation Method using Shuffled Row Major Indexing on Wireless Mesh Sensor Network

문 창 주, 최 미 영, 박 정 근*
(Chang-Joo Moon[1], Mi-Young Choi[2], and Jungkeun Park[1,*])
[1]Department of Aerospace Information Engineering, Konkuk University
[2]CNTech Systems

Abstract: In wireless mesh sensor networks (WMSNs), sensor nodes are connected in the form of a mesh topology and transfer sensor data by multi-hop routing. A data aggregation method for WMSNs is required to minimize the number of routing hops and the energy consumption of each node with limited battery power. This paper presents a shortest path data aggregation method for WMSNs. The proposed method utilizes a simple hash function based on shuffled row major indexing for addressing sensor nodes. This allows sensor data to be aggregated without complex routing tables and calculation for deciding the next hop. The proposed data aggregation algorithms work in a fractal fashion with different mesh sizes. The method repeatedly performs gathering and moves sensor data to sink nodes in higher-level clusters. The proposed method was implemented and simulations were performed to confirm the accuracy of the proposed algorithms.

Keywords: wireless mesh sensor network, data aggregation, shuffled row major indexing

I. 서론

무선 메쉬 센서 네트워크(Wireless Mesh Sensor Network: WMSN)는 일반적인 무선 센서 네트워크(Wireless Sensor Network: WSN)에서 특화된 네트워크로 노드들이 메쉬 구조를 구성하는 대규모의 네트워크이다. WMSN은 넓은 지역에 분산된 다수의 센서들로부터 데이터를 수집하기 용이한 구조를 가지고 있어 센서들이 일정한 구조를 가지고 분산되어 있는 플랜트 시스템, 건물 자동화, 물류 시스템 등에서 활용도가 높다[1,2]. WMSN은 일정한 통신 거리 안에 4개의 이웃 노드로 망을 구성하며 $N = 2^p$ ($p >= 1$)인 N x N 구조의 노드들로 구성된다. 이러한 구조는 안정적이지 않은(stateless) 노드 위치를 기반으로 구성되는 일반 무선 센서 네트워크보다 양질의 데이터를 얻을 수 있어 비용대비 효과적이다[3,4].

WMSN은 노드들의 메쉬구조 때문에 소스 노드부터 목적지 노드까지 다양한 경로가 존재하므로 최소 홉(hop)에 목적지 노드까지 데이터를 수집해야 한다. 또한, 노드의 제한된 리소스 때문에 특정 노드가 배터리 소진으로 작동하지 않으면 데이터 수집에 영향을 준다. 따라서 데이터 수집 방법의 확장성, 강건성, 신뢰성, 재구성, 에너지 효율성 등의 특성을 만족시키는 방법들은 WMSN의 주요 연구 이슈들이다.

기존 연구들은 WMSN에서 데이터 수집을 위해 별도의 라우팅 테이블이나 검색 트리를 사용함으로써 노드들이 많은 연산을 수행하지만 본 논문에서 제안된 방법은 셔플드 로우 메이져 인덱싱(shuffled row major indexing) [5] 주소 체계를 기반으로 하는 간단한 해쉬 함수만을 사용하여 데이터를 수집한다. 수집된 중간값들은 WMSN 내부의 클러스터 헤더 노드들이 분산하여 수집함으로써 한 노드에 데이터 집계 부담이 가중되는 것을 방지하며 최종적으로는 최종 집계까지의 홉의 합이 N x N 메쉬구조에서의 일반적인 O(N)보다 적거나 같은 선형적인 T(N)의 값의 되도록 한다.

본 논문에서는 이전 논문[6]에서 언급했던 해쉬 함수를 이용하여 최소의 멀티 홉으로 데이터 집합화를 수행하는 알고리즘을 발전시키고 이를 구현하고 검증하였다. 기존 알고리즘에서는 프렉탈 특성을 가능하게 하는 8가지 타입의 알고리즘 사용을 제안했지만 구체적인 적용 조건에 대해서 언급하지 못했다. 그러나 보강된 알고리즘에서는 기존 8개 알고리즘을 WMSN에서 사용되는 각 조건에 일반화하여 제안함으로써 알고리즘의 정확성을 보강하였다. 또한 제안된 데이터 수집 알고리즘이 다양한 크기의 WMSN에서 작동되게 함으로써 제안된 알고리즘이 확장성을 가지게 하였다. 그리고 알고리즘을 구현하고 데이터 집합화 각 단계를 시뮬레이션 함으로써 다양한 크기의 WMSN과 다양한 조건에서도 제안한 알고리즘이 정확하게 작동하고 있음을 시각적으로 확인할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. II장에서는 관련연구에 대해서 언급하고 III장에서는 데이터 집합화에 사용되는 해쉬 함수들과 노드 인덱싱 방법에 대해서 설명한다. IV장에서는 데이터 집합화 알고리즘에 대해서 설명하고 마지막으로 V장에서 결론에 대해서 언급한다.

* Corresponding Author
Manuscript received September 20, 2016 / revised September 30, 2016 / accepted October 3, 2016
문창주, 박정근: 건국대학교 항공우주정보시스템공학과
(cjmoon@konkuk.ac.kr/parkjk@konkuk.ac.kr)
최미영 : CNTech 시스템 연구소장(michelle@cntechsystems.com)

II. 관련연구

WMSN에서의 라우팅 프로토콜은 구조에 따라 크게 평면 라우팅(flat routing)과 계층적 라우팅(hierarchical routing)으로 분류된다[7,8]. 평면 라우팅에서는 모든 노드들이 동등하게 동작하며 각 센서 노드들은 라우팅 기준에 따라 접근 가능한 주변 노드들에 데이터를 전달한다. 반면, 계층적 라우팅에서는 인접한 센서 노드들을 클러스터링(clustering)하고 클러스터마다 헤드(head) 노드를 할당하여 헤드 노드가 클러스터의 데이터를 수집하고 이를 싱크(sink) 노드로 전달한다. 계층적 라우팅은 평면 라우팅에 비해 에너지 소모가 적을 뿐만 아니라 확장성(scalability) 및 강건성(robustness)이 좋아 WMSN에서의 라우팅에 적합하다.

WMSN의 각 센서 노드는 지역적(local) 또는 전역적(global)으로 유일한 주소를 할당 받는다. 주소 할당은 라우팅 알고리즘과 연관되며 WMSN의 동적 재구성 특성상 대부분 동적으로 이루어진다. 주소 할당은 센서 네트워크의 규모, 네트워크의 구조 등에 따라 다양한 방법이 사용된다[9,10].

트리 기반의 주소할당의 경우 센서노드들 간에 트리를 구성하고 각 노드는 부모(parent) 노드의 주소 공간으로부터 주소를 할당 받는다. 주소가 계층적으로 할당되므로 각 노드는 목적지 주소만으로 데이터를 부모 노드 혹은 자식(children) 노드에 전달할 지 결정할 수 있다. 따라서 트리 기반 주소할당에서는 별도의 라우팅 테이블 없이 데이터 전송이 이루어진다. 이러한 기법은 Zigbee의 분산 주소할당 알고리즘에 사용된다[10].

확률적(stochastic) 주소할당의 경우 주소는 랜덤하게 할당되며 주소 중복을 허용한다. 주소가 중복될 경우 주소 결정 프로토콜(address resolution protocol)에 의해 주소가 재할당된다[9]. 이러한 주소할당 기법은 메쉬 구조를 효과적으로 지원한다. 대표적으로 Zigbee Pro [11]에서 이를 지원한다. 확률적 주소할당에서는 주소할당이 랜덤하게 이루어지기 때문에 주소 간의 연관체계가 없어 AODV [12,13]와 같이 테이블 기반의 라우팅 알고리즘이 사용된다. 주소 결정 프로토콜과 라우팅 테이블은 WMSN의 각 센서노드의 메모리 요구량을 증가시키는 단점이 있다.

위치기반 라우팅에서는 센서들의 위치정보가 주소로 활용되며 거리가 가까운 노드에 데이터를 전송하게 된다[7,8]. 따라서 위치기반 라우팅에서는 데이터 전송에 의한 에너지 소모를 크게 줄일 수 있다. 반면 각 노드들은 GPS 위치 정보 등을 통해 센서의 위치정보를 교환해야 한다. 이밖에 센서들이 길게 연결된 구조(Long-Thin Network)에서는 센서들을 클러스터링하여 주소를 할당하고 각 클러스터의 헤드 노드와 브리지 노드 간에 데이터 전송이 이루어 진다[10].

한편 본 논문에서 가정하는 N X N 메쉬구조에서의 주소할당과 데이터 수집과정은 메쉬구조의 병렬 컴퓨터에서의 분산 처리와 데이터 수집과정과 유사하다[14,15]. 분산 처리에서 정렬과 기하 알고리즘들을 효율적으로 수행하기 위해서 메쉬 노드 간의 통신을 줄이고 인접한 노드 간의 결과를 수집할 수 있도록 메쉬 노드의 인덱스를 할당하고 있다. 본 논문에서 제안하는 노드 인덱스 방법은 스마트 그리드[16], 스마트 홈, 스마트 공장[1] 등 정적으로 노드가 설치된 WMSN에 적합하다. 제안된 방법은 N X N 메쉬구조를 클러스터로 나누고 정적으로 노드의 인덱스를 할당한다. 제안된 방법은 각 센서 노드가 데이터를 전송할 노드가 노드의 주소에 의해 정해져 있어 라우팅 테이블을 유지할 필요가 없어 라우팅 테이블에 의한 메모리 소모와 라우팅 프로토콜에 의한 통신 오버헤드가 없다.

III. 해쉬 함수를 이용한 데이터 집합화

이 장은 WMSN에서 노드 인덱싱, 싱크노드 주소계산, 헤드 노드의 주소 계산 그리고 데이터 집합화를 위한 센싱 데이터 전송경로 계산을 수행하는 해쉬 함수를 소개한다.

1. WMSN 주소체계와 주소해쉬 함수

WMSN는 그림 1과 같이 2^p x 2^p = 4 x (2^{p-1} x 2^{p-1}) 의 구조와 2p 비트 크기의 셔플드 로우 메이져 인덱싱의 주소 체계를 가진다. 이때 p는 1보다 큰 양의 정수이다.

주소 체계에 의해서 전체 메쉬 영역은 그림 2와 같이 4개의 클러스터로 분할 되고 각 분할된 클러스터는 또 다시 4개의 클러스터로 나누어지는 자기 복제의 성격을 갖는다. 그림 2의 경우 p가 3으로 메쉬 네트워크 크기는 8 x 8 이고 6 비트로 노드 안쪽에 주소를 표현했다.

메쉬 크기에 따라 주소가 정해지면 클러스터 헤드 및 싱크노드의 주소가 결정된다. k 비트(bit) 주소를 가지는 WMSN에서 싱크노드의 주소는 k번째, k-1번째 비트만 1이고 나머지는

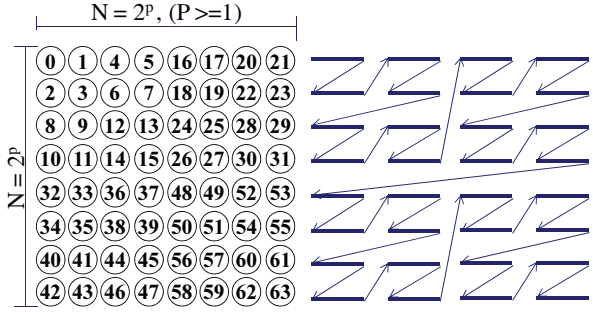


그림 1. N x N 셔플드 로우 메이져 인덱싱 주소체계.

Fig. 1. Address system of N x N shuffled row major indexing.

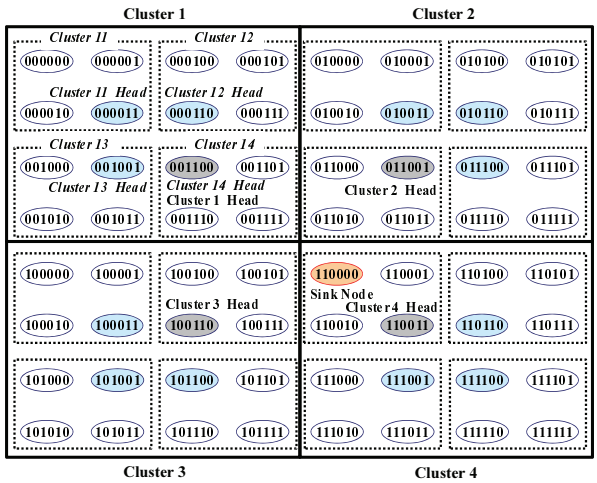


그림 2. 8 x 8 메쉬 구조.

Fig. 2. 8 x 8 mesh structure.

모두 0으로 채워지는 주소이다. 그림 2의 메쉬구조는 6비트이므로 11000(=48) 주소의 노드가 싱크노드이다.

그림 2의 8 X 8 메쉬 구조는 4 X 4 메쉬 구조를 가지는 4개의 클러스터(Cluster 1, Cluster 2, Cluster 3, Cluster 4)로 구성되며 각 4 X 4 클러스터의 데이터가 수집된 헤더 노드들(address: 12, 25, 38, 51)은 한 단계 높은 8 X 8 클러스터의 데이터 수집과 이동의 출발점이 된다. 이러한 과정은 16 X 16 클러스터에서도 동일하게 적용된다.

그림 2와 같이 WMSN은 다수의 데이터 노드와 하나의 싱크 노드로 구성된다. 데이터 노드에서 생성된 데이터는 주변 데이터 노드들을 경유하여 싱크 노드로 전달되며 이 과정을 데이터 집합화라 한다. 그림 2의 주소체계와 비트연산은 데이터 집합화 과정에서의 프렉탈 특성의 알고리즘을 가능하게 하며 라우팅 테이블과 같은 부가적인 정보 없이도 최소경로 데이터 수집이 가능하게 한다.

2. WMSN 구성

메쉬 구조로 WMSN을 구성할 때는 아래와 같은 전제 조건이 필요하다.

1) 네트워크

WMSN의 메쉬구조에서 대각선 경로의 홉보다 가로와 세로 두개의 홉이 더 에너지 효율적이며 시분할 다중접속을 기반으로 구성하여 각 노드는 자신에게 할당 받은 시간 슬롯에 데이터를 전송한다.

2) 센서노드

센서는 아래와 같은 상태를 가지며 상태 변화에 따른 전력 소비는 무시할 수 있다.

- Transmit : 센서는 목적지 주소와 데이터를 전송한다.
- Receive : 센서는 전송 받은 데이터의 주소가 자신의 주소인지 확인한다.
- Switch Off : 일단 데이터의 송신이 완료되면 대기 상태가 된다.

각 센서는 메쉬 구조 상에서 연결된 최대 4개의 이웃 노드의 주소를 알고 있다.

3) 싱크노드

센서 노드와 달리 H/W 제약을 받지 않는다.

3. 데이터 집합화 방법

그림 3과 같이 최소 단위 메쉬 네트워크는 4 x 4 메쉬 구조이고 최소 단위 클러스터는 2 x 2 메쉬 구조이다. 그림 3과 같이 최소 단위 클러스터의 헤더 위치에 따라 4종류의 최소 단위 클러스터가 가능하며 각 클러스터마다 다른 데이터 집합화 경로 step 1과 step 2가 존재한다. 최소 단위 메쉬 네트워크의 중앙에 있는 최소 단위 클러스터 헤더들의 데이터 집합화를 위해서는 최소 단위 클러스터에 사용되었던 4종류의 데이터 집합화 방법이 동일하게 사용된다.

그림 4는 메쉬 네트워크의 p값이 증가하여 노드가 증가하는 경우 클러스터를 구분하는 중앙의 경계선을 기준으로 그림 3에서 설명된 데이터 집합화 방법이 대칭적이고 반복적으로 사용됨을 보여 준다. 중앙의 박스 내부에서는 싱크 노드로의 데이터 이동이 진행되다가 마지막의 step 7과 step8단계에서 데이터 집합화가 일어난다. 이러한 데이터 집합화와 이동은 p+1구조에서도 동일하게 적용된다.

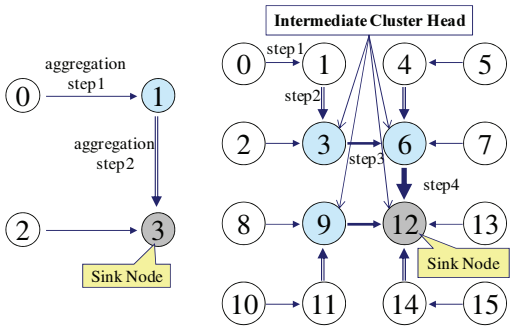


그림 3. 최소단위 클러스터와 최소단위 메쉬 네트워크.
Fig. 3. Minimum size cluster and mesh network.

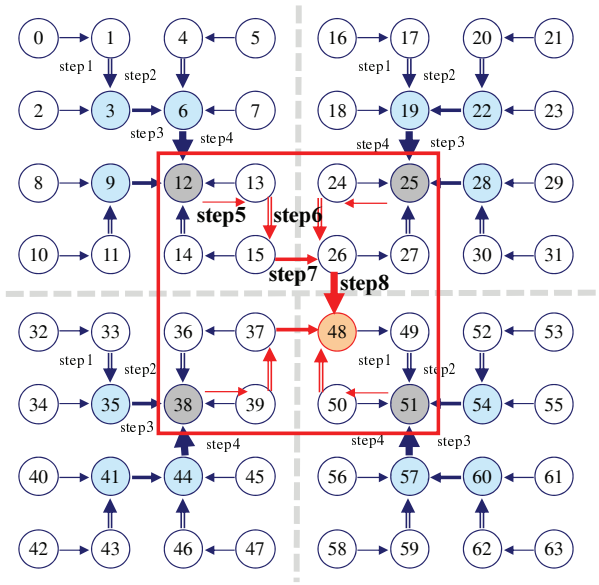


그림 4. 데이터 집합화 방법.
Fig. 4. Data aggregation method.

IV. 데이터 집합화 알고리즘

이 장에서는 제안된 WMSN에서의 주소부여 알고리즘, 싱크 노드로의 데이터 수집 알고리즘과 데이터 이동 알고리즘을 설명한다. 이 모든 알고리즘은 모두 확장성이 제공되어 p값이 증대되어 WMSN의 크기가 증대 되어도 정확하게 작동한다.

1. 주소부여 알고리즘

WMSN의 각 노드를 나타내는 SensorNode 클래스와 각 멤버 변수와 메소들에 대한 명세는 그림 5와 같다. 셔플드 로우 메이져 인덱싱 방법으로 WMSN의 각 노드에 주소를 부여하는 알고리즘은 그림 6과 같다. p값이 1인 경우 최소 클러스터를 구성하게 되므로 그 때 주어진 x, y 지점에서 주소를 부여한다. header 값은 4등분 되는 클러스터를 구분할 수 있게 한다. 재귀호출을 통하여 하위 클러스터로 이동하여 p값이 1이 될 때까지 header의 값을 이용하여 주소를 부여한다.

2. 데이터 집합화 알고리즘

데이터 집합화는 데이터 수집과 이동을 통틀어서 의미하며 데이터 수집은 데이터의 값이 싱크 노드에서 합쳐지는 것을 의미하고 데이터 이동은 클러스터 헤더의 데이터 수집 값이 상위단계 클러스터 싱크 노드의 수집을 위해서 이동하는

단계를 의미한다. 데이터 집합화 알고리즘은 그림 7과 같다. 주소부여 과정에서 정해진 p값에 따라 메쉬 구조의 가로 세로 값인 N이 결정이 되고 데이터 집합화 알고리즘의 입력으로 사용된다.

먼저 최소 단위 메쉬 네트워크에서 데이터 집합화를 위해 사용 변수들을 초기화한다. size는 클러스터의 크기로 최소 크기 4(=2x2)부터 증가하여 16(=4x4), 64(=8x8)을 거쳐 N x N

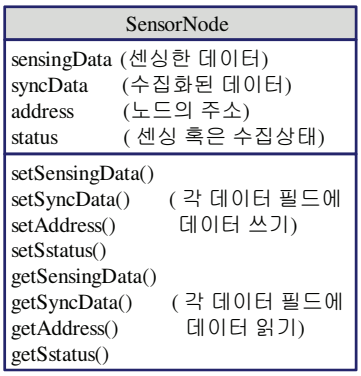


그림 5. SensorNode 클래스.

Fig. 5. SendorNode class.

```
procedure shuffledIndexing(x, y, p, header : integers)
   N := 1 << p
   if p == 1 then
      k := 0
      for i :=x to i < 2+x
         for j :=y to j < 2+y
            WMSN[i][j].setAddress((header << 2) | k)
            k++;
         end
      end
   else
      shuffledIndexing(x, y, p-1, (header << 2) | 0)
      shuffledIndexing(x, y + N/2, p-1, (header << 2) | 1)
      shuffledIndexing(x + N/2, y, p-1, (header << 2) | 2)
      shuffledIndexing(x + N/2, y + N/2, p-1, (header << 2) | 3)
```

그림 6. 주소부여 알고리즘.

Fig. 6. Address assignment algorithm.

```
size : 데이터 수집/이동 클러스터 크기
cycle : 데이터 수집/이동 단계.
head : 클러스터 별 데이터 수집을 시작하는 노드
aStep : 데이터 수집 횟수
startNode : 데이터 이동 시작 하는 노드
range : 데이터 이동을 수행하는 범위의 가로/세로 노드 수

procedure aggregation_move(N:integer)
   head:= 0, aStep:=1, mask:=3, startNode:=0, range:=1
   for cycle:=1 to cycle<((N/2)+1)
      size:= 1<< (2*aStep)
      if (cycle is power of two) then
         aggregation(cycle, N)
      else
         move(cycle, N)
   end
```

그림 7. 데이터 집합화 알고리즘.

Fig. 7. Data aggregation algorithm.

까지 증가한다. 즉 데이터 집합화가 발생하는 클러스터의 노드 수를 의미한다. cycle은 데이터 수집 혹은 이동이 이루어지는 2홉마다 1이 증가하며 최댓값은 N/2이다. cycle값은 데이터 집합화 과정에서의 홉 이동마다 데이터 수집을 수행할지 혹은 데이터 이동을 수행할지를 결정한다. 예를 들어 size가 64(=8x8) 클러스터의 싱크노드로 데이터가 집합화가 발생할 때 cycle의 최댓값은 4이다. cycle 3은 싱크노드로 데이터 수집이 수행되기 이전 단계인 데이터 이동 단계를 의미한다. head는 싱크 노드로 데이터 수집이 수행되는 단계 즉 cycle의 값은 N/2일 때 싱크 노드 대각선에 있는 노드로써 데이터 수집이 시작되는 노드를 의미한다. aStep은 cycle 값이 N/2일 때 1씩 증가한다. 즉 데이터 집합화 과정에서 서브 클러스터의 단계가 증가할 때 마다 1씩 증가한다. 최댓값은 p값과 동일하다. startNode는 크기가 size인 클러스터의 싱크 노드를 의미한다. 이 싱크 노드는 그 다음 단계 클러스터의 싱크 노드로 데이터 이동의 출발 노드가 된다. range는 헤더노드의 값들이 그 다음 단계의 싱크 노드로 데이터 집합화가 수행될 때 이전 단계 헤더 노드들 사이의 거리를 의미한다.

데이터 집합화 함수 aggration_move()가 시작되면 변수들이 초기화되고 cycle 값을 통하여 전체 데이터 집합화 과정이 제어된다. cycle값에 따라 aStep값이 결정되고 aStep값에 따라 size값이 결정된다. cycle값이 2의 제곱값 즉 N값과 같으면 클러스터 해더노드에 데이터가 수집되는 최종단계이므로 데이터 수집을 위하여 aggregation() 함수를 호출하고 그렇지 않은 경우에는

```
procedure aggregation(cycle, N)
   if cycle == 1 then
      startNode := 0x03
   else
      head:= (head << 2) | 0x03, startNode:= startNode << 2
   for i:=0 to i<N
      for j:=0 to j<N
         address = WMSN[i][j].getAddress()
         if (address % size) == head then
            Data_Aggregation(i, j, (address >> (2*aStep))%4)
      end
   end
   aStep++; range=range << 1, mask := (mask << 2) | 0x03

Data_Aggreation(x, y, type)
   x0 = x; x1 = x0;    x2 = x0+1; x3 = x0+1;
   y0 = y; y1 = y0+1; y2 = y0;    y3 = y0+1;
   switch(type)
      case A : Data[x1,y1] <- Data[x0,y0]
               Data[x3,y3] <- Data[x2,y2]
               Data[x3,y3] <- Data[x1,y1]
      case B : Data[x0,y0] <- Data[x1,y1]
               Data[x2,y2] <- Data[x3,y3]
               Data[x2,y2] <- Data[x0,y0]
      case C : Data[x1,y1] <- Data[x0,y0]
               Data[x3,y3] <- Data[x2,y2]
               Data[x1,y1] <- Data[x3,y3]
      case D : Data[x0,y0] <- Data[x1,y1]
               Data[x2,y2] <- Data[x3,y3]
               Data[x0,y0] <- Data[x2,y2]
```

그림 8. 데이터 수집 알고리즘.

Fig. 8. Data gathering algorithm.

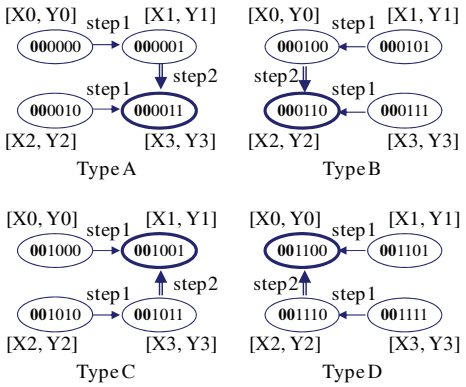


그림 9. 데이터 수집 타입.

Fig. 9. Data gathering type.

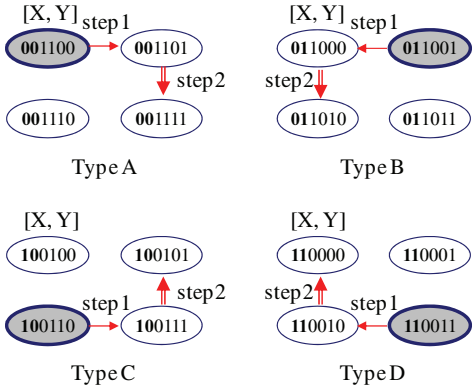


그림 11. 데이터 이동 타입.

Fig. 11. Data move type.

데이터 이동을 위하여 move() 함수를 호출한다. cycle값이 최댓값인 N/2에서 WMSN의 싱크 노드에 데이터 수집을 수행하고 데이터 집합화가 종료된다.

그림 8은 데이터 수집 알고리즘을 나타낸다. aggregation() 함수가 호출된다는 것은 데이터 집합화 과정에서 증가하는 클러스터의 크기에 따른 해당 헤더 노드로 데이터가 수집되는 것을 의미한다. head 값을 왼쪽으로 2비트 쉬프트 시키고 최하위 2비트를 모두 1로 만들어 한 단계 증가된 1사분면 클러스터의 헤더노드(head 값)를 찾는다. 그 다음 WMSN 모든 노드를 순회하면서 1사분면 클러스터 head 값을 이용하여 나머지 클러스터들의 헤더노드들을 찾고 Data_Aggregation() 함수를 호출한다. 그리고 데이터 수집 후 클러스터의 크기가

```
procedure move(cycle, N)
   moveStep := cycle - (range>> 1) - 1
   for i:=0 to i<N
      for j:=0 to j<N
         address = WMSN[i][j].getAddress()
         if ((address % mask) == startNode) then
            Data_Migration (I, j, range, moveStep)
      end
   end

Data_Migration(xi, yi, range, moveStep)
   int x, y
   // case type A
   x = xi + moveStep, y = yi + moveStep
   Data[x,y+1]    <- Data[x,y]
   Data[x+1,y+1] <- Data[x,y+1]
   // case type B
   x = xi + moveStep, y = yi + range -1- moveStep
   Data[x,y-1]    <- Data[x,y]
   Data[x+1,y-1] <- Data[x,y-1]
   // case type C
   x = xi + range-1-moveStep, y = yi + moveStep
   Data[x,y+1]    <- Data[x,y]
   Data[x-1,y+1] <- Data[x,y+1]
   // case type D
   x = xi + range-1-moveStep, y = yi + range -1- moveStep
   Data[x,y-1]    <- Data[x,y]
   Data[x-1,y-1] <- Data[x,y-1]
```

그림 10. 데이터 이동 알고리즘.

Fig. 10. Data move algorithm.

한 단계 증가하므로 데이터 이동이 시작하는 startNode의 주소값을 왼쪽으로 2비트 쉬프트하여 클러스터 크기와 일치하도록 한다. 데이터 수집이 완료되면 aStep, range, mask 값을 그 다음 단계 클러스터에 적합하도록 증가시킨다.

size값에 따라 head 값이 정해지면 address >> (2*aStep) 연산에 따라 주소의 비트수를 오른쪽으로 쉬프트하고 4로 나머지 연산을 수행하면 그림 9와 같은 데이터 수집 타입이 결정된다. 값이 0이면 A타입, 1이면 B타입, 2이면 C타입, 3이면 D타입이다. 수집 타입이 결정되면 Data_Aggregation() 함수의 case문에서 타입에 해당하는 데이터 수집이 수행된다.

그림 10는 데이터 이동 알고리즘을 나타낸다. cycle값이 2의 제곱승이 아니면 그림 4의 step 5, step6과 같은 헤더노드로의 데이터 수집을 위한 이동하는 단계이므로 move()함수가 호출 된다. moveStep은 startNode에서 데이터 수집을 위해서 head 노드로의 이동하는 단계를 의미한다. moveStep이 계산되면 WMSN 모든 노드를 순회하면서 startNode 노드를 찾고 그 위치에서 range와 moveStep 정보를 이용하여 수집 데이터를 이동한다.

데이터 이동은 startNode, range, moveStep이 정해지면 최종 싱크노드로의 이동 출발점과 이동 범위가 결정된다. 이동 범위 안에서 그림 11의 A타입, B타입, C타입, D타입의 이동이 실행된다. cycle값이 변함에 따라 moveStep이 결정되고 이동 데이터가 있는 노드 주소를 다시 계산하여 다시 4가지 타입의 이동을 각각 수행한다. Data_Migration()함수가 호출 되면 각 타입의 이동이 데이터 수집 바로 전 단계까지 수행된다.

V. 알고리즘 구현과 검증

이 장에서는 제안된 알고리즘의 성능을 분석하고 알고리즘을 구현하여 제안된 알고리즘이 올바르게 수행되는지 검증한다.

1. 알고리즘 성능 분석

N x N 메쉬 네트워크에 대해 제안된 알고리즘에서 데이터가 싱크 노드에 최종 집계되는 최대 홉을 T(N)이라고 할 때 $N=2^p$인 경우 $T(2^p)$는 다음과 같이 표현될 수 있다.

$$T(2^p) = T(2^{p-1}) + 2(2^{p-2} - 1) + 2$$

우변에서 $T(2^{p-1})$은 2^{p-1} x 2^{p-1} 서브 클러스터에서 싱크 노드

로 데이터가 수집되는 홉의 수를 나타내고 두번째 항은 서브 클러스터의 헤드 노드에서 상위 클러스터 싱크 노드의 가장 가까운 노드로 데이터를 이동하는 홉의 수를 나타내며 마지막 항은 최종 데이터 수집을 위한 홉의 수를 나타낸다. 최소 단위 메쉬 네트워크인 2 x 2 네트워크에 대해 $T(2^1) = 2$ 이므로 위 식을 풀면 다음과 같다.

$$\begin{aligned} T(2^p) &= T(2^{p-1}) + 2(2^{p-2} - 1) + 2 \\ &= T(2^{p-1}) + 2^{p-1} \\ &= T(2^{p-2}) + 2^{p-1} + 2^{p-2} \\ &= \dots \\ &= T(2^1) + 2^{p-1} + 2^{p-2} + \dots + 2^1 \\ &= 2(2^{p-1} - 1) + 2 \\ &= 2^p \end{aligned}$$

따라서 T(N) = N 으로 표현되며 N x N 메쉬 네트워크에서 데이터 수집에 필요한 최대 홉은 O(N)의 선형적인 특성을 지닌다.

2. 알고리즘 구현

제안된 데이터 집합화 알고리즘은 그림 12와 그림 13과 같이 구현되고 검증되었다. 입력으로 p값을 3으로 하고 Create Random Network 버튼을 입력하면 $N = 2^p$ 이므로 $8 = 2^3$ 인 8 x 8 센서 네트워크가 생성된다. Show Node Address 버튼을 누르면 생성된 네트워크의 노드들의 주소를 확인할 수 있고 Show Sensing Data 버튼을 누르면 각 노드들의 센싱된 데이터 값을 확인할 수 있다. 데이터 집합화 단계를 쉽게 확인 할 수 있도록 각 센서의 센싱 데이터 값은 1로 단일화 하였다.

Execute 버튼을 누르면 데이터 이동과 수집의 각 단계가 데이터 뷰에 보여지며 hop의 값이 표시된다. Execute 버튼을 누르면서 데이터 집합화를 수행하면 각 단계에서의 데이터 수집과 이동을 확인할 수 있다. 그림 13(a)는 8 x 8 메쉬구조에서 최종집계까지의 홉의 합이 선형적인 T(N)의 값인 8홉안에 싱크 노드에 모든 센서 데이터의 값의 합이 수집된 결과를 보여준다. 그림 13(b)는 p값을 6으로 했을 때 64 x 64 메쉬 구조에서 데이터가 수집된 결과를 보여준다. 따라서 제안된 알고리즘이 p값이 증가 하여도 정확히 작동을 한다.

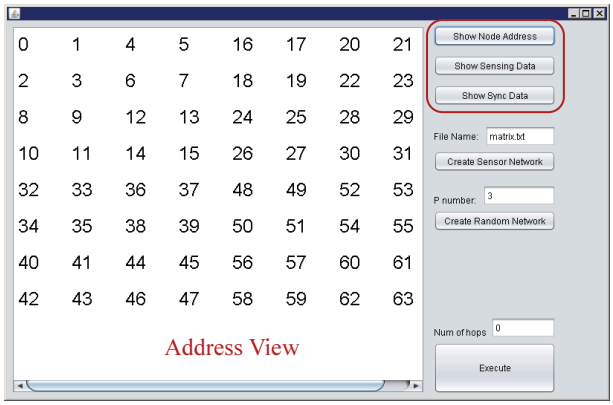
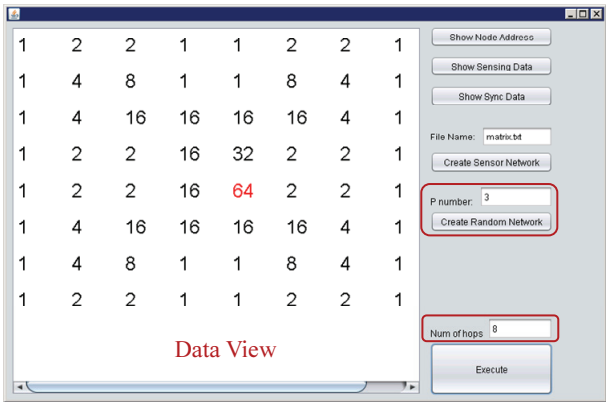
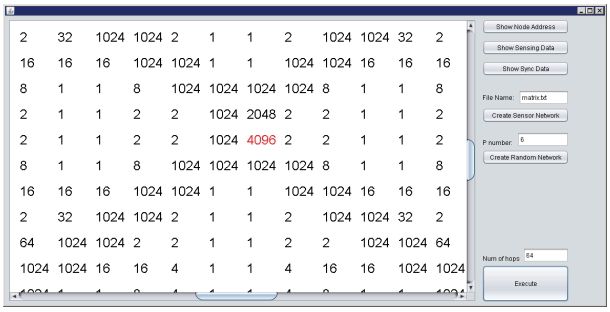


그림 12. 노드 주소 할당 알고리즘 수행 결과.
Fig. 12. Result of node address assignment.



(a) 8 x 8 mesh network.



(b) 64x64 mesh network.

그림 13. 데이터 집합화 알고리즘 수행 결과.
Fig. 13. Result of data aggregation algorithm.

VI. 결론

본 논문에서는 별도의 라우팅 테이블이나 검색 트리를 사용하지 않고 셔플드 로우 메이져 인덱싱 주소기반의 데이터 집합화 알고리즘을 발전시키고 구현을 통하여 검증하였다. 데이터 집합화에 사용되는 8가지 타입의 프렉탈 특성을 가지는 알고리즘의 적용조건을 일반화함으로써 알고리즘의 정확성을 보강하였으며 다양한 크기의 WMSN에서 작동되게 하여 알고리즘이 확장성을 가지게 하였다. 알고리즘을 구현하고 데이터 집합화 각 단계를 시뮬레이션 함으로써 제안된 알고리즘의 정확한 실행을 시각적으로 확인하였다.

향후 연구로는 네트워크의 구조가 정방형이 아니거나 일부 노드가 작동 하지 않는 경우에도 데이터 수집화를 수행할 수 있는 방안에 대한 연구가 필요하다.

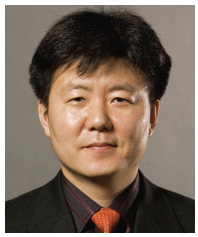
REFERENCES

[1] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," *Proc. of the 19th International Conference on Software, Telecommunications and Computer Networks*, pp. 1-6, Sep. 2011.

[2] H. Ha, Y. Hwang, K. Jung, H. Kim, B. Lee, and J. Lee, "TPC algorithm for fault diagnosis of CAN-based multiple sensor network system," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 22, no. 2, pp. 147-152, Feb. 2016.

[3] M. Singh, V. K. Prasanna, J. Rolim, and C. S. Raghavendra, "Collaborative and distributed computation in mesh-like wireless sensor arrays," *IFIP International Federation for Information Processing*, 2003.

[4] D. Wang, "Clustering mesh-like wireless sensor networks with an energy-efficient scheme," *Proc. of third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2006)*, Oct. 2006.
[5] C. D. Thompson and H. T. Kung, "Sorting on a mesh-connected parallel computer," *Communications of the ACM*, vol. 20, no. 4, pp. 263-271, Apr. 1977.
[6] M.-Y. Choi, C.-J. Moon, and D.-K. Baik, "The shortest path data aggregation using hash function on mesh wireless sensor network," *Lecture Notes in Electrical Engineering*, vol. 181, pp. 409-416, 2012.
[7] R. Devika, B. Santhi, and T. Sivasubramanian, "Survey on routing protocol in wireless sensor network," *International Journal of Engineering and Technology*, vol. 5, no. 1, pp. 350-356, Feb.-Mar. 2013.
[8] K. Pavai, A. Sivagami, and D. Sridharan, "Study of routing protocols in wireless sensor networks," *Proc. of 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, Kerala, India, pp. 522-525, Dec. 2009.
[9] D. Rodenas-Herraiz, A.-J. Garcia-Sanchez, F. Garcia-Sanchez, and J. Garcia-Haro, "Current trends in wireless mesh sensor networks: a review of competing approaches," *Sensors*, vol. 13, no. 5, pp. 5958-5995, May 2013.
[10] N. Saholia and S. Joshi, "Review on address assignment mechanism in ZigBee wireless sensor networks," *International Journal of Engineering Research & Technology*, vol. 2, no. 11, pp. 2067-2072, Nov. 2013.
[11] Zigbee Alliance, Zigbee specification. http://www.zigbee.org.
[12] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on demand distance vector (AODV) routing," *IETF RFC 3561*, 2003.
[13] N. H. Phong and M.-K. Kim, "Reliable message routing protocol for periodic messages on wireless sensor networks," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 17, no. 2, pp. 190-197, Feb. 2011.
[14] H. M. Alnuweiri and V. K. P. Kumar, "Optimal VLSI sorting with reduced number of processors," *IEEE Transactions on Computers*, vol. 40, no. 1, pp. 105-110, Jan. 1991.
[15] R. Miller and Q. F. Stout, "Mesh computer algorithms for computational geometry," *IEEE Transactions on Computers*, vol. 38, no. 3, pp. 321-340, Mar. 1989.
[16] P. Wang, H. Hou, X. He, C. Wang, T. Xu, and Y. Li, "Survey on application of wireless sensor network in smart grid," *Procedia Computer Science*, vol. 52, pp. 1212-1217, 2015.



문 창 주

1997년 고려대학교 컴퓨터학과(이학사). 1999년 고려대학교 컴퓨터학과(이학석사). 2004년 고려대학교 컴퓨터학과(이학박사). 2005년 고려대학교 정보보호대학원 연구교수. 2006년 건국대학교 컴퓨터응용과학부 컴퓨터시스템전공. 2007년~현재 건국대학교 항공우주정보시스템 공학과 교수. 관심분야는 데이터엔지니어링, 분산시스템, 빅데이터.

최 미 영

1990년 숭실대학교 전자계산학과(공학사). 1992년 포항공과대학교 전자계산학과(공학석사). 2013년 고려대학교 컴퓨터학과(이학박사). 2015년~현재 CNTech 시스템 연구소장. 관심분야는 병렬처리, 빅데이터.

박 정 근

1997년 서울대학교 전기공학부 졸업. 1999년 동 대학원 석사. 2004년 서울대학교 전기·컴퓨터공학부 박사. 2008년 3월~현재 건국대학교 항공우주정보시스템공학과 교수. 관심분야는 임베디드 실시간 운영체제, 실시간 시스템.