

이동 센서 네트워크에서 트리 기반의 배치 알고리즘

Tree-based Deployment Algorithm in Mobile Sensor Networks

문 종 천, 박 재 현*
(Chongchun Moon and Jaehyun Park)

Abstract : Sensor deployment is an important issue in the mobile wireless sensor network. In this paper, we propose a deployment algorithm for mobile sensor network to spread out mobile sensor nodes widely as well as regularly. Since the proposed algorithm uses tree topology in deploying the sensor nodes, calculating power as well as spreading speed can be reduced compare to other deployment algorithms. The performance of the proposed algorithm is simulated using NS-2 simulator and demonstrated.

Keywords : deployment algorithm, mobile wireless sensor network, wireless sensor network

I 서론

센서 네트워크(sensor network)는 정보를 수집 및 처리하고 서로 통신을 하는 센서 노드들(sensor nodes)이 통신망을 구성하여 이루어진 네트워크이다. 그 중, 이동 센서 네트워크(mobile sensor network)의 노드는 기본 능력 외에 이동 능력이 있기 때문에 조작을 통하여 환경 변화에 적절한 대응을 할 수 있기 때문에 수집하는 데이터의 질을 높여 줄 수가 있다. 그러나 이러한 모바일 노드가 어느 한곳에 집중적으로 모여 있다면, 그 지역에서 수집하는 데이터는 분해능이 좋은 데이터가 되겠지만 넓은 지역의 데이터를 포함 할 수 없다는 단점이 있고 각 노드의 데이터가 중복이 될 수도 있어 자원의 낭비를 불러올 수 있다. 따라서 노드가 서로의 위치를 파악하고 적절한 배치가 된다면 넓은 지역을 포함 하면서 각각의 노드에서 유용한 정보를 얻을 수 있다. 예를 들어 장기간의 밀접관찰이 필요한 생태계 감시나 직접 접근이 위험한 적진의 군사동향 감시를 위한 무인정찰을 임무로 하는 모바일 센서 네트워크에서 노드가 적절한 거리를 유지하면서 노드가 맡은 임무를 수행한다면 포함하고자 하는 지역을 넓게 고루 퍼지면서 각각의 노드가 센싱을 하기 때문에 원하는 지역을 최대한 포함할 수 있어 더 좋은 결과를 얻게 된다.

하지만 기존의 센서 네트워크에 관한 연구에서는 라우팅이나 에너지 소모, 서비스의 질 등에 관한 연구에 집중되어 온 반면, 이동 센서 노드(mobile sensor Node)의 배치(deployment)는 중요한 문제임에도 불구하고 시나리오들이 배치에 드는 비용과 시간과 같은 실질적인 이유 때문에 무작위 배치(random deployment)를 적용해서 사용했다. 그러나 사전 지식이 없는 지역에서 이용될 경우 무작위 배치는 이동 센서 네트워크에서 바람직하게 분포되어지길 원하는 관심 영역상에서의 일정한 분포(uniform distribution)를 제공하지 못하고, 네트워크 노드 간 단절이 발생하여 데이터 전송 지연이나 센서 노드의 메모리 제약에 따른 데이터 손실 등 여러 가지 문

제가 발생한다[1]. 따라서 이러한 무작위 배치에서의 문제점을 해결하기 위하여 일정하게 분산된 노드 위상의 구성을 통하여 네트워크 포함범위(network coverage)와 시스템 수명(system lifetime) 연장의 향상을 목적으로 한 자체 배치 방법들(self deployment methods)이 제안 되었다[2,3].

이동 센서 네트워크는 이동성(mobility) 자체만으로도 주어진 제한된 에너지로부터 에너지 소비를 하기 때문에 배치 설계(deployment scheme)는 노드들이 배치되는 동안 되도록 적은 에너지를 소비하도록 고려해야 하고 만족스러운 포함범위나 에너지 효율적인 노드 위상을 가지도록 고안 되어야 한다. 게다가 분산되는 노드가 상대적으로 간단한 하드웨어 구조를 가지기 때문에 각 노드는 배치, 조직화(organization), 네트워크 관리 등에 간단하고 효율적인 방법을 가져야 한다.

배치 알고리즘에 관한 기존의 연구 중에 DSSA(Distributed Self Spreading Algorithm)는 물리 법칙에서 핵간 인력과 반발력의 개념을 도입하여 P2P(Peer-to-Peer)방식으로 무작위로 배치된 노드를 배치하였다[4]. 이는 노드 간 거리에 작용하는 힘은 거리에 반비례하기 때문에, 밀접한 노드들은 큰 힘을, 멀리 떨어진 노드들은 서로 작은 힘을 발생시켜 무작위로 배치된 불규칙한 위상의 노드를 균일하게 분포된 무선 네트워크 형태로 만든다. 다른 배치 알고리즘으로 포텐셜 필드(potential field)기반의 배치 알고리즘도 있다[2]. 기본 개념은 노드와 장애물이 포텐셜 필드를 구성하고 각 노드는 다른 노드나 장애물에 의해 가상의 힘을 받는다. 힘의 작용으로 노드가 배치되고 네트워크는 결국에는 물리적인 힘의 평형에 도달하며 최대의 포함 범위를 갖게 된다.

이러한 기존의 방법들은 초기 배치가 무작위 하게 분포 되어 있어 노드들이 어느 정도 퍼져 있을 경우를 가정하여 각 노드들이 이웃 노드와의 거리나 장애물과의 거리 등을 계산하여 이동한다. 하지만 초기 배치 시 노드를 모아 놓고 한군데에서 배치하거나 특정 상황에서 노드가 모여 있을 경우, 각 노드가 모든 이웃 노드에 대해 계산을 하여 이동하게 된다면 노드의 확산이 느려지고 밀집된 노드들 사이에 있는 노드는 진동을 하는 경우가 발생한다. 게다가 계산 또한 많아지게 되어 센서 노드에 무리를 주는 단점을 가진다.

따라서 본 연구에서는 각 노드간에 부모-자식의 계층형 관계를 만들어 트리의 형태의 위상을 만드는 배포 알고리즘을

* 책임저자(Corresponding Author)

논문접수 : 2005. 12. 23., 채택확정 : 2006. 6. 30.

문종천 : 뉴센텔레콤(ccmoon@emcl.inha.ac.kr)

박재현 : 인하대학교 정보통신공학부(jhyun@inha.ac.kr)

※ 본 논문은 산업자원부지원 차세대신기술개발사업인 “지능성발현을 위한 진화적응 복합칩 개발” 과제의 지원으로 연구되었음.

제시한다. 여기서 부모-자식의 관계는 네트워크에서 통신의 위상이 아니라 노드들의 위치를 배치하기 위한 관계이다. 위와 같이 트리 형태를 가짐으로써 모바일 노드는 균일하게 관심 지역에 넓게 분포되면서 부모 노드나, 같은 부모를 갖는 자식 노드와의 거리만 신경 쓰면 되므로 적은 계산 능력을 가진 센서 노드에 적합하다. 본 논문에서 제시된 알고리즘은 네트워크 시뮬레이터를 이용하여 시뮬레이션하고 평가하였다.

II. 트리 기반의 배치 알고리즘

본 논문에서는 제시하는 알고리즘은 트리 형태의 배치를 통하여 센서 노드들을 포함 하고자 하는 영역에 골고루 배치하며 일정한 균일성을 가지면서 계산은 단순화하여 배치 시간을 줄이도록 하였다. 우선 본 논문에서 제시할 알고리즘에서 사용한 가정에 대해서 알아보고 트리 기반의 배치 알고리즘을 제안하도록 한다.

1. 가정

우리가 연구하는 알고리즘에서 모든 센서 노드는 센싱, 통신, 계산 그리고 이동성의 능력을 가지고 있다고 가정한다. 각 노드의 센싱 범위와 통신 범위는 이상적이라고 가정을 한다. 두 가지의 범위는 그림 1과 같이 원의 형태를 가진다.

각 노드의 센싱 범위는 그림 2처럼 이진 센서 모델(binary sensor model)[5] 이나 확률적 센서 모델(stochastic sensor model)[6] 을 가진다. 이진 센서 모델에서는 감지 범위 안에서의 검출하고자 하는 사건의 탐색은 감지 범위 안의 모든 사건은 거리에 상관 없이 모두 똑같다고 가정하기 때문에, 감지 범위 밖으로 나간다면 사건을 검출할 가능성은 없다. 그래서 이진 센서 모델을 사용하는 센서 네트워크는 각 센서 노드의 감지 거리의 합으로 결정될 수 있다. 확률적 센서 모델에서는 센서로부터 거리에 따라서 센서 감지 감소 함수를 이용하여 측정을 한다. 그 중 본 논문에서는 이진 센서 모델을 이용하기로 한다.

계산 능력은 각 노드에서 라우팅을 하거나 알고리즘을 추론하여 배치 알고리즘의 실행을 위하여 필요하다. 센서 노드에 일반적으로 많이 쓰이는 컨트롤러는 계산 능력이 떨어지기 때문에 알고리즘에서 계산이 적어 진다면 계산 속도도 빠르고 에너지 소모에서 이점을 가지게 된다. 본 연구에서 초기 배치는 무작위로 배치하지 않고 노드가 특정상황에서 모여 있거나, 어떤 상황에서 노드를 무작위로 배치 하지 못하고 한곳으로부터 배치를 해야 할 경우를 가정하여 노드를 한곳에 모아서 배치를 한다. 본 연구에서 기준이 되는 루트 노드는 배치를 시작하기 전에 사용자가 임의로 하나를 설정하여 노드의 배치 기준을 만든다. 이밖에 모든 노드는 수신 신호의 세기 등을 이용하여 자기 자신의 위치를 알 수 있다는 것과 또한 데이터 전송과 위치 계산에는 에러가 없다고 가정을 한다. 위치 추정은 배치 알고리즘에서 다음 움직일 위치를 알기 위하여 필요하다.

2. 알고리즘

본 연구에서는 이동 센서 네트워크에서 노드의 배치를 하고자 트리 기반의 알고리즘을 제시 한다. 노드간 계층 구조를 만들어 계산을 단순화하고 알고리즘을 이용하여 노드를 배치하게 함으로써 노드가 포함하는 영역을 크게 하면서 전체 네트워크의 형태가 일정한 형태를 갖게 하도록 하는 것을

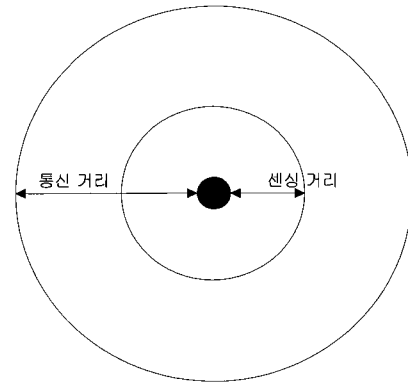


그림 1. 노드의 이상적인 통신범위와 센싱범위.
Fig. 1. Ideal communication range and sensing range of node.

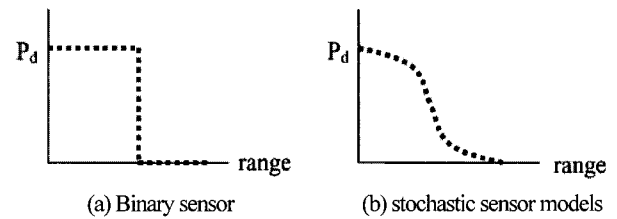


그림 2. 센서의 포함 모델[7].
Fig. 2. Sensor coverage models[7].

```

Procedure Tree-based Deployment Algorithm
1. Link setup
   Initial_node_locations p0;
   Initial_sensing_range sR; communication_range cR;
   Initial_child_num, current_child_num;
   Search_neighbor & make_neighbor_list;
   While(current_child_num < child_num)
     Choose_select_node from_neighbor_list;
     Check_select_node;
     If(not(bLinked))
       Current_child_num++;
       Child_addr = select_node_addr;
       Go_while_loop;
     Else
       Go_while_loop;
   Wait_until_command;
2. Distance maintenance
   Initialize P, D, D';
   While(not(In a region of stable))
     Calculate_distance, theta;
     Calculate_force f_n(D, theta);
     update_temporary_position p_{n+1}^i;
     If(threshold_min < |p^n - 1 - p^{n+1}| < threshold_max)
       Stop_node_i's_movement;
     Else
       Update_next_location_to_the_temporary_position;
       Go_to_while_loop;
    
```

그림 3. 트리 기반 배치 알고리즘의 가상코드.
Fig. 3. Pseudo code for tree-based deployment algorithm.

목표로 한다. 본 알고리즘에서는 루트 노드를 두게 되고 루트를 기준으로 하여 링크와 노드를 만들어 간다. 그래서 본 알고리즘상에서는 모바일 노드 중에 하나를 초기에 루트 노드로 설정하게 되고 그 루트 노드를 기준으로 하여 통신을 통하여 주변 노드를 검색하여 자식 노드를 만들어 나간다. 각각의 노드는, 상위 노드가 자기 자신의 자식 노드를 만들게 되면 하위 노드로 자식 링크 만드는 프로세스를 실행하도록 명령을 내려 보낸다. 명령을 받은 노드는 이웃 노드를 검색하여 또 다른 자식 노드를 만들어 나가게 된다.

트리 기반의 배치 알고리즘의 가상코드는 그림 3과 같다. 알고리즘은 크게 노드간 링크 만드는 부분과 거리 유지 실행 부분으로 나뉜다. 우선 노드간 링크 만드는 부분을 살펴 보면 그림상의 link setup의 순서를 따라 각 노드들은 자식 노드를 만들어 간다. 명령을 받으면 자식 노드의 수를 설정하고 현재 가지고 있는 자식 노드 변수를 초기화 한다. 그리고 이웃 노드 리스트를 만들게 되는데 여기서 이웃 노드는 통신 가능 거리 범위 안에 있는 모든 노드를 말한다. 부모 노드는 이 이웃 노드 리스트를 이용하여 자식 노드가 될 노드를 정하게 되는데, 부모 노드는 거리가 가까운 순으로 이웃 노드들에게 자식 노드 요청 패킷을 보내게 된다. 요청 패킷을 받은 노드는 자기 자신이 다른 노드와 연결되어 있는지 검사하고 링크가 만들어져 있다면 거부 메시지를 보내게 되고 링크가 없다면 요청 패킷을 보낸 송신 노드와 링크를 설정하고 수락 메시지를 보내며 수락 메시지를 받은 송신 노드 또한 링크를 설정하게 된다. 그 다음에 노드는 현재 가지고 있는 자식 노드의 수가 설정된 노드 수와 맞는지 비교 후에 아직 설정 값보다 모자라다면 링크 만드는 프로세스를 다시 실행하여 자식 노드를 만들게 된다.

그림 4는 링크 만드는 과정을 통하여 노드간 부모-자식을 만드는 과정을 보여준다. 노드 1을 기준으로 하여 노드 1은 통신을 통하여 자식 노드를 만들게 되는데 우선 주변 노드를 검사하여 이웃 노드 리스트를 만들게 되고 그림에서와 같이 노드 2로 자식 노드를 만들기 위한 요청 패킷을 보내게 된다. 그 다음 요청 패킷을 수신한 노드 2는 자기 자신과 관계를 맺은 노드가 없다면 링크관계를 허락하는 응답 패킷을 보내게 되고 parent_node 변수에 부모 노드의 주소를 넣는다. 응답 패킷을 받은 송신 노드는 수신된 패킷의 주소를 이용하여 자식 노드의 아이디를 left_node 변수에 기록하게 된다. 송신 노드는 아직 right_node 변수가 비어있기 때문에 추가로 링크를 만들기 위하여 주변 노드를 검색하여 노드 3에 요청 패킷을 보내게 된다. 노드 3은 현재 관계를 맺은 노드가 없기 때문에 요청 패킷에 응답을 하여 노드 1과 부모-자식 관계를 맺게 된다.

다음과 같이 링크 만드는 과정을 이용하여 모든 노드가 서로 연결되게 되면 거리 유지 과정을 실행 하게 된다. 거리 유지를 하기 위하여 노드는 수신한 전파 신호의 세기를 이용하여 거리와 위치를 측정하게 된다. 이 측정된 값을 이용하여 각 노드는, 노드가 같은 부모 노드나 이웃 노드 등을 이용하여 이동해야 할 위치를 계산하게 되고 노드의 이동력을 이용하여 목표 위치로 이동하게 된다.

거리 유지 실행 부분을 살펴보면 처음 시작에서는 프로세

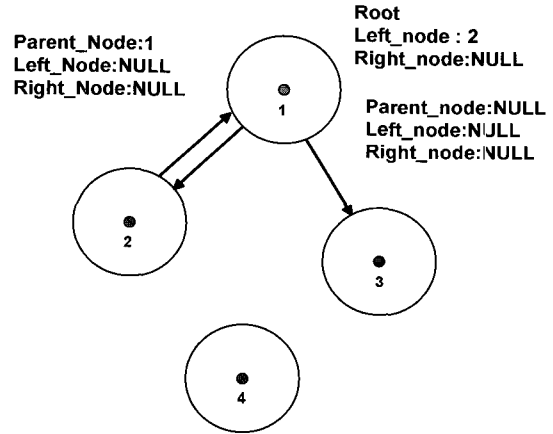


그림 4. 링크 만드는 과정.
Fig. 4. Link setup process.

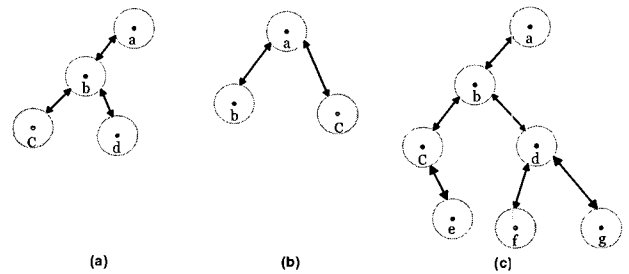


그림 5. 노드의 조건에 따른 계산대상.
Fig. 5. Calculation target depend on node's condition.

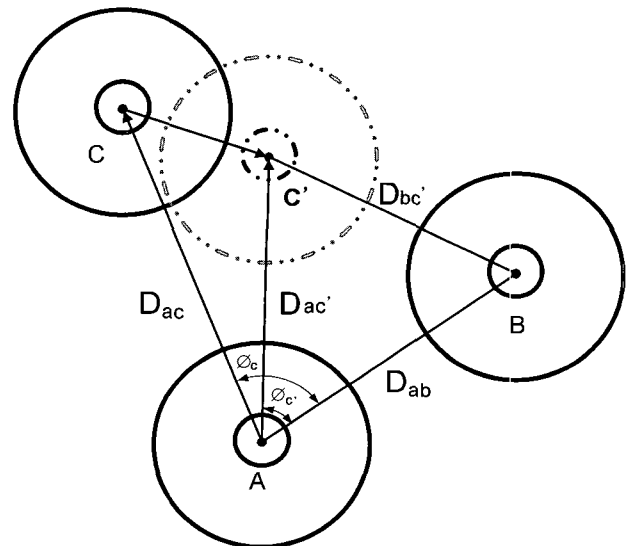


그림 6. 노드의 이동.
Fig. 6. Illustration of node's movement.

스에서 사용하게 될 각종 변수를 초기화하고 계산 하고자 하는 노드와의 거리를 측정한다. 각 자식 노드들은 현재 자신이 부모 노드의 왼쪽 노드인지 오른쪽 노드인지 등의 상황에 따라 계산하는 노드가 달라지게 되는데 각 상황에 따른 대상 노드들과의 벡터 합으로 이동해야 할 위치를 구하게 된다. 구해진 위치를 가지고 이동하기 전에 이동한 후의 위치가 임

계 범위 값의 위치라면 노드는 안정된 위치에 있다고 판단을 하고 이동을 멈추게 된다. 그렇지 않으면 조건에 맞는 위치에 다다를 때까지 이동을 하게 된다.

거리 유지 실행 과정에서 각각의 노드는 다음과 같은 세 가지의 경우에 따라 계산하는 노드의 대상이 달라지게 되어 서로의 거리를 조절하게 된다.

- 같은 깊이에서 맨 왼쪽에 있는 자식 노드일 경우(그림 5(a)): 트리 형태의 링크를 갖는 노드들 중에 루트 노드를 기준으로 가장 왼쪽에 있는 자식 노드(c)일 경우 기준 노드가 되어 부모 노드(b)와 그 부모 노드의 부모 노드(a)와 180도가 되는 방향으로 거리를 유지한다.
- 오른쪽 자식 노드일 경우(그림 5(b)): 트리 형태의 링크를 갖는 노드들 중에 어느 한 노드가 오른쪽 자식 노드(c)일 경우에 그 노드의 부모 노드(a)와 같은 부모를 갖는 왼쪽 노드(b)와 거리를 유지한다.
- 처음 경우를 제외한 왼쪽 자식 노드일 경우(그림 5(c)): 트리 형태의 링크를 갖는 노드들 중에 조건 1)이 아닌 왼쪽 노드(f)일 경우 같은 깊이의 바로 이웃하는 오른쪽 노드(e)를 찾고 부모 노드(d)와 그 이웃하는 오른쪽 노드(e)와 거리를 유지한다.

노드의 계산할 대상이 구해지면 노드는 대상 노드들과의 거리를 이용해서 움직이게 된다. 그림 6과 같이 노드 A, B, C가 있을 경우, 노드 C가 C'로 이동할 때 3개의 노드간 거리를 이용해서 노드를 움직인다. 노드는 힘의 세기와 방향을 이용하여 움직이게 되는데 이는 $F(\underline{f}_i, \theta)$ 를 이용해서 표현할 수 있다. \underline{f}_i 는 힘의 세기로 이동해야 할 거리가 멀수록 큰 힘이 작용하며, θ 는 이동해야 할 방향을 나타낸다. 그림 6의 예시 그림을 이용하여 식으로 표현하면 기존의 직교 좌표 계의 X축을 선 AB 축으로 회전변환하고 A는 원점이라 가정할 때 아래 (1)~(5) 같이 나타낸다. 현재 노드 C의 위치 P_c 를 구하면 (1)과 같고 노드 C가 이동해야 될 위치 $P_{c'}$ 를 구하면 (2)와 같다. 힘이 작용하는 세기의 기준이 되는 거리는 $D_{cc'}$ 가 되는데 아래 (5)와 같이 구할 수 있다. (4)에서 구한 θ 를 다시 직교좌표 계의 형태로 변환시키면 노드가 이동하여야 하는 방향을 구할 수 있다. 아래 식을 이용해서 θ 와 거리를 구하고 노드는 목표 지점을 향하게 된다.

$$P_c = (d_{ac} \cos \theta_c, d_{ac} \sin \theta_c) \quad (1)$$

$$P_{c'} = (d_{ac'} \cos \theta_{c'}, d_{ac'} \sin \theta_{c'}) \quad (2)$$

$$\overline{CC'} = (d_{ac'} \cos \theta_{c'} - d_{ac} \cos \theta_c, d_{ac'} \sin \theta_{c'} - d_{ac} \sin \theta_c) \quad (3)$$

$$CC'\text{의 } \theta = \arctan\left(\frac{d_{ac'} \sin \theta_{c'} - d_{ac} \sin \theta_c}{d_{ac'} \cos \theta_{c'} - d_{ac} \cos \theta_c}\right) \quad (4)$$

$$D_{cc'} = \sqrt{(d_{ac'} \cos \theta_{c'} - d_{ac} \cos \theta_c)^2 + (d_{ac'} \sin \theta_{c'} - d_{ac} \sin \theta_c)^2} \quad (5)$$

III. 성능평가

본 논문에서 제안한 알고리즘의 성능을 평가 하기 위하여 NS2 (Network Simulator)[8,9]를 이용하여 모델링을 하고 시뮬레이션을 하였다.

NS2의 환경아래에서 알고리즘을 추가하고 시뮬레이터가 제공하는 각종 클래스를 이용하여 시뮬레이션 환경을 구축

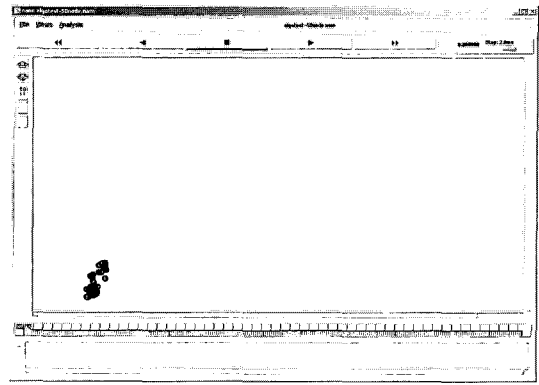


그림 7. 노드의 초기 배치: 노드가 50개일 경우.

Fig. 7. Initial distribution: 50 nodes.

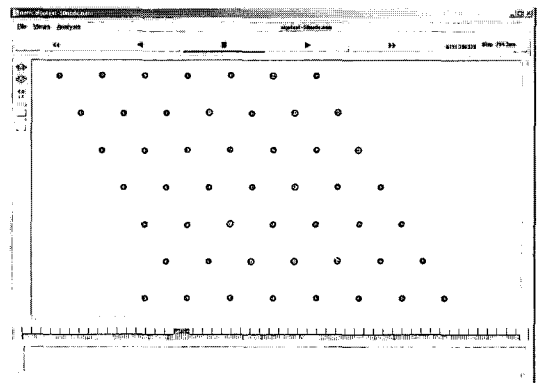


그림 8. 배치 알고리즘 수행 후: 노드가 50개일 경우.

Fig. 8. Final distribution after running algorithm: 50 nodes.

하고 노드가 50개일 경우 시뮬레이션을 실행하여 이동 센서 노드가 알고리즘 적용 후 배치 되는 모습을 살펴본다. 전파의 도달 가능 거리는 35m이고 노드가 배치 되는 영역은 300m*300m이다. 그림 7은 노드 50개가 초기 배치된 상태이다. 여기에서도 초기 배치는 노드가 한곳에 모여서 배치하게 된다. 노드가 배치 되는 영역은 300X300이다. 그림 8은 알고리즘 수행후의 모습이다. 노드가 기준 노드를 기준으로 해서 균일하게 분포 되어 있는 모습을 보이고 있다. 알고리즘상의 트리의 위상에서 맨 왼쪽 노드들은 기준선을 만들게 된다. 그림에서는 아래쪽 노드들이 해당된다. 기준이 되는 노드를 기준으로 해서 노드를 배치해 나간다.

이동 센서 네트워크에서 알고리즘의 성능 평가를 위해 적절한 측정 방법을 선택하는 것은 중요한 문제이다. 포함범위(coverage), 시간(time), 거리(distance)는 모바일 센서 네트워크에서 성능 척도로써 고려 된다[10]. 포함범위는 센서들이 배치가 완료된 다음의 센서 네트워크의 성능에 연관된다. 시간과 거리는 배치 설계의 성능과 직접적으로 연관 된다.

본 논문에서 포함범위는 전체 관심 영역과 각 노드가 포함하고 있는 영역의 합에 대한 비율로, 각 노드의 포함 영역은 센싱 반경 R을 포함하는 원으로 정의한다. 또한 센서가 포함하는 영역에서의 모든 사건은 완전히 검출된다고 가정한다. 만약 노드가 관심 영역 안쪽에 배치된다면 완전한 원이 관심 영역 상에 놓이게 된다. 원의 완전한 영역(즉 πR^2)이 포함

영역으로써 여겨지게 되는 것이다. 일반적으로 노드들은 센싱 거리와 통신 거리는 다르게 여기는데, 본 논문에서도 노드의 센싱 거리와 통신 거리는 구분 된다. 센싱 포함 범위의 확장은 센서 노드가 무선 링크로 다른 노드에 연결되면 발생한다.

배치하는 동안 진행된 시간은 탐색과 구조, 재해 복구 시행 같은 시간 제한적인 응용에서는 중요하다. 대부분 요구되는 시간은 알고리즘의 계산, 추론의 복잡성과 노드가 움직이는 물리적 시간에 달려있다. 총 소요되는 시간은 노드가 최종 목적 위치에 도달 하기까지로 정의 된다. 본 논문에서는 일반적으로 소스 노드부터 목적지 노드 사이에서의 네트워크 성능 평가나 서비스의 질을 위하여 측정되는 데이터 전달 지연 등을 고려하면서 배치 본연에 소모되는 시간에 초점을 맞춘다.

각 노드가 이동한 평균 거리는 노드의 이동에 필요한 에너지와 연관되어 있다. 그렇기 때문에 각 노드가 제한된 에너지를 가지고 있을 때 이동할 거리를 예상하는 것은 이동 시에 필요한 에너지를 가늠하기 때문에 중요하며, 이동 거리의 차이 또한 배치 알고리즘의 공정성을 결정하고 시스템 에너지 이용하기 때문에 중요하다.

그림 9에서 11까지는 앞서 말한 성능 지표인 포함범위, 시간, 거리 등의 배치 알고리즘 성능 지표를 이용하여 평가한 결과를 보여준다. 평가를 하기 위해 노드를 5개에서 50개까지 이용하여 나온 결과 값을 이용하였다. 각각의 시뮬레이션 환경에서 노드는 400mX300m의 영역에 배치되었다고 설정하였으며, 각각의 노드의 통신 가능 거리는 40m로, 노드의 이동 속도는 1m/s로 설정하였다.

그림 9는 트리 기반의 배치 알고리즘(TBDA)의 성능 평가를 위하여 노드 개수 당 포함범위를 보여 주고 있으며, 앞의 성능지표에서 설명한, 노드가 배치 되는 전체 영역상에서 배치된 노드가 포함하는 영역의 비율로 나타낸다. 시뮬레이션에서 초기 배치는 노드가 모여있다고 가정하기 때문에 DSSA는 노드가 증가 하여도 약간의 증가만을 보이고 있는 반면, 트리 기반의 배치 알고리즘을 적용한 후에는 노드 수가 많아 질수록 포함 범위가 월등히 증가 하는 것을 보인다.

그림 10은 트리 기반의 배치 알고리즘의 성능 평가를 위하여 노드 개수 당 배치에 소요되는 시간을 보여 주고 있다. 시간은 알고리즘 수행 시간과 모든 노드가 목적지에 도달하기까지의 합으로 표현된다. 노드가 증가할수록 배치 시 소요 시간이 증가하는 것을 알 수 있는데, 이는 알고리즘의 수행 시간은 노드가 많아 짐에 따라 링크를 만드는 시간이 길어지고 데이터 패킷의 충돌 또한 증가하여 시간이 증가하기 때문이다. 게다가 노드가 모여서 배치되기 때문에 노드가 증가할수록 배치 알고리즘상에서 먼 쪽에 위치해야 될 노드는 이동 거리가 길어지기 때문에 수행 시간이 길어진다.

그림 11은 트리 기반의 배치 알고리즘에서 노드 개수 당 노드의 평균 이동 거리를 보여준다. 노드가 증가할수록 멀리 이동해야 되는 노드가 많아지기 때문에 총 이동 거리는 증가하게 된다. 그래서 트리 기반의 배치 알고리즘에서는 노드가 많아 질수록 이동해야 되는 평균 이동 거리가 길어진다.

그림 12는 트리 기반의 배치 알고리즘의 성능 평가를 위하여 DSSA(Distribute Self Spreading Algorithm)에 대하여 포함 범

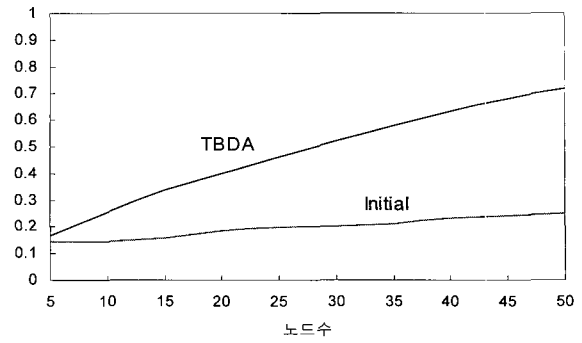


그림 9. 노드 개수당 포함 범위.
Fig. 9. Coverage versus node size.

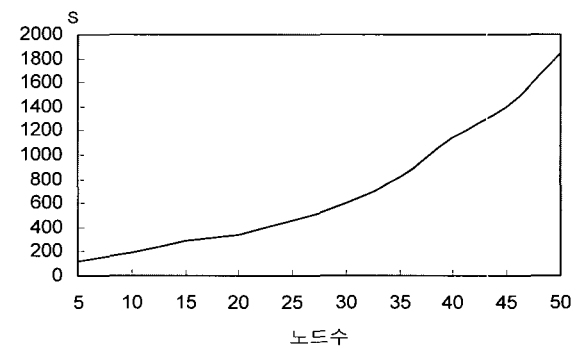


그림 10. 노드 개수당 수행 시간.
Fig. 10. Execution time versus node size.

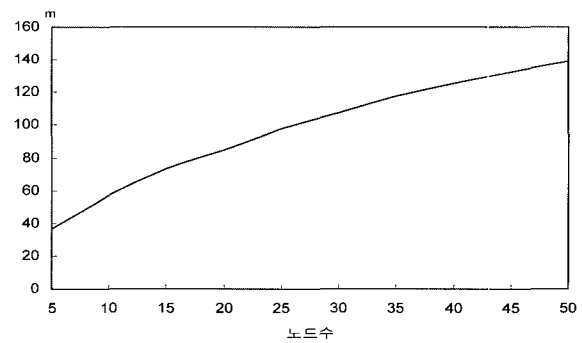


그림 11. 노드 개수당 평균 이동 거리.
Fig. 11. Mean traveled distance versus network size.

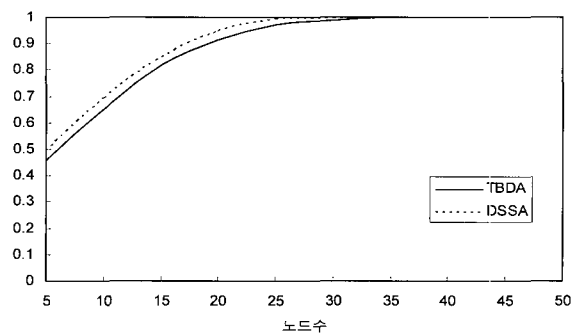


그림 12. 노드 개수당 포함 범위 비교.
Fig. 12. Coverage comparison versus network size.

위를 비교한 것을 보여 준다. 각각의 노드는 100mX100m 크기의 공간에 배치되고, 노드의 개수는 5에서 50으로 하였으며, 통신가능 거리와 센싱 가능거리를 각각 40m, 20m로 설정하였다. 실험 결과 두 알고리즘 모두 충분한 노드 개수가 주어진다면 포함 하고자 하는 영역을 모두 포함하는 좋은 결과를 보여준다. 하지만 DSSA가 적은 노드 수 일 때 좀 더 넓은 포함범위를 보였는데, 이는 TBDA의 노드의 배치 모양이 노드가 증가할수록 좌우로 퍼지면서 포함범위를 포함하여 가기 때문에 발생하는 차이이다.

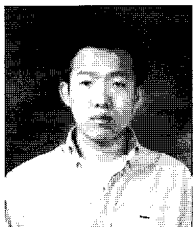
IV. 결론

본 연구에서는 무선 센서 네트워크에서 노드의 배치 문제에 대해서 알아 보았다. 센서 네트워크는 제한된 통신 및 센싱 거리를 갖는 노드를 사용하기 때문에 관심 영역이 포함될 필요가 있지만, 몇몇 연구는 배치의 비용과 시간 같은 실질적인 이유 때문에 무작위 배치를 사용하였다.

하지만 무작위 배치를 이용하게 되면 라우팅 경로를 보장하지 못하게 되고 넓은 지역을 포함하지 못하여 원활한 임무 수행을 할 수 없게 된다. 그래서 본 논문에서는 모바일 센서 노드의 이동 능력을 이용하여 정보 획득을 원하는 지역에 네트워크는 가능하면서도 넓게 퍼뜨려 불규칙한 초기 노드의 배치 위상을 항상 시키기 위한 트리 기반의 배치 알고리즘을 제안하였다. 트리 기반의 배치 알고리즘은 자료 구조에 쓰이는 트리의 개념을 이용하여 노드간 계층 구조를 형성하여 계산을 단순화하고, 알고리즘을 이용하여 노드를 배치하게 함으로써 노드가 포함하는 영역을 최대화하면서 전체 네트워크가 일정한 형태를 갖는 것을 목표로 하였다. 트리 기반의 알고리즘 수행 후 각 노드는 계층형 관계를 맺게 되고, 각각 노드는 맺어진 관계를 기반으로 하여 관심 영역상에서 모바일 노드의 이동력을 이용하여 균일하게 배치된다. 앞의 성능 평가와 시뮬레이션 결과값처럼 노드개수가 증가할수록 노드의 포함 범위가 넓어 지는 모습을 보여주고 있어 배치 알고리즘의 성공적인 수행 결과를 보여 주고 있다. 하지만 노드가 모여서 배치 되기 때문에 노드의 개수가 많아 질수록 알고리즘 수행 시간과 노드의 이동 거리가 늘어남을 보여주고 있다.

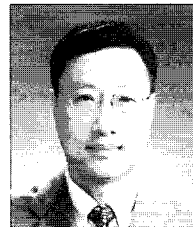
참고문헌

- [1] A. F. T Winfield, "Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots," *Distributed Autonomous Robotic Systems 4*, Springer-Verlag, pp. 273-282, 2000.
- [2] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. of the 6th Int'l Conf. on Distributed Autonomous Robotic Syst. (DARS02)*, Fukuoka, Japan, pp. 299-308, 2002.
- [3] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proc. of the IEEE INFOCOM Conference*, 2003.
- [4] N. Heo and P. K. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," in *Proc. of IEEE Wireless Communications and Networking Conference, WCNC 2003*, 2003.
- [5] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, pp. 472-476, 2001.
- [6] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. IEEE Infocom*, 2001.
- [7] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," in *IEEE Transaction on Systems, Man, and Cybernetics*, Jan, 2005.
- [8] NS-2 Documentation, Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC.
- [9] NS Tutorials, <http://www.isi.edu/nsnam/ns/tutorial/index.html>
- [10] D. W. Gage, "Command control for many-robot systems," *AUVS-92, the Nineteenth Annual AUVS Technical Symposium*, Huntsville AL, 22-24 June 1992. Reprinted in *Unmanned Systems Magazine*, vol. 10, no. 4, pp. 28-34, fall, 1992.
- [11] G. Wang, G. Cao and T. L. Porta. "Movement-assisted sensor deployment," *IEEE INFOCOM*, Hong Kong, March, 2004.
- [12] N. Heo and P. K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," *Systems, Man and Cybernetics, IEEE International Conference 2003*, vol. 5, Pages: 4576 - 4581, 5-8 Oct, 2003.
- [13] J. Wu and S. Wang, "Smart: a scan-based movement-assisted deployment method in wireless sensor networks," in *Proc. IEEE INFOCOM*, Miami, March 2005.



문 종 천

2003년 인하대학교 컴퓨터공학과 졸업.
2006년 인하대학교 정보통신공학과 석사 졸업. 2006년~현재 뉴젠텔레콤(주) 근무. 관심분야는 실시간 내장형 시스템, 로봇 제어.



박 재 현

1994년 서울대학교 제어계측공학과 졸업(공학박사). 1995년~현재 인하대학교 정보통신공학과 교수. 관심분야는 임베디드 시스템, 실시간 시스템, 컴퓨터 네트워크, 무선 센서 네트워크.