대규모 무선 센서 네트워크에서 노드 분포를 고려한 분산 위치 인식 기법 및 구현

정회원 한 상 진*, 이 성 진*, 종신회원 이 상 훈*, 정회원 박 종 준**, 박 상 준***

Node Distribution-Based Localization for Large-scale Wireless Sensor Networks

Sangjin Han*, Sungjin Lee* ***Regular Members***, Sanghoon Lee* ***Lifelong Member***,
Jongjun Park**, Sangjoon Park*** ***Regular Members***

요 약

대규모 센서 네트워크를 위한 분산 위치 인식 기법은 센서 네트워크의 운용에 있어서 반드시 필요한 기술이다. 센서가 획득한 데이터에 대한 지형적인 위치를 결정함으로써 그 데이터에 대한 가치가 중요해질 수도 그렇지 않을 수도 있기 때문이다. 본 논문에서는 초소형, 저가의 센서가 비교적 많은 수로 분포 되어 있고 또한 그 분포 특성이 대부분 균일할 때 기존에 사용되어 오던 삼각법 보다는 정확하고 최근에 제안된 MDS를 이용한 기법 보다는 간단하고 효율적인 분산 위치 인식 기법에 대해서 소개 한다. 이는 노드의 분포 특성을 이용함으로써 삼각법과 같은 매우 간단한 기법 보다는 보다 정확하고 많은 계산을 요구하여 시스템의 실용성에는 적합하지 않은 MDS보다는 보다 간단한 위치인식을 수행한다. 이는 MATLAB을 이용한 PC 실험에서 제안하는 기법의 성능을 검증하고 Crossbow사의 micaZ 모트에 TinyOS-2.x를 이용하여 제안하는 위치인식 기술을 구현함으로써 그 실용성을 입증하였다.

key Words : Wireless sensor networks, Distributed localization algorithm, Large-scale sensor networks, Trilateration, Multidimensional Scaling

ABSTRACT

Distributed localization algorithms are necessary for large-scale wireless sensor network applications. In this paper, we introduce an efficient node distribution based localization algorithm that emphasizes simple refinement and low system load for low-cost and low-rate wireless sensors. Each node adaptively chooses neighbor nodes for sensors, update its position estimate by minimizing a local cost function and then passes this update to the neighbor nodes. The update process considers a distribution of nodes for large-scale networks which have same density in a unit area for optimizing the system performance. Neighbor nodes are selected within a range which provides the smallest received signal strength error based on the real experiments. MATLAB simulation showed that the proposed algorithm is more accurate than trilateration and les complex than multidimensional scaling. The implementation on MicaZ using TinyOS-2.x confirmed the practicality of the proposed algorithm.

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 정보통신 선도기반기술개발 사업의 일환으로 수행하였음 [2005-S-038, UHF RF-ID 및 Ubiquitous 네트워킹 기술 개발]
* 연세대학교 전기전자공학과 무선네트워크 연구실 (sangjinhan@posdata.co.kr, elflee7@yonsei.ac.kr, slee@yonsei.ac.kr)
** 한국전자통신연구원 융합기술연구부문 RFID/USN 연구본부 USN 네트워킹팀 (juny@etri.re.kr)
*** 한국전자통신연구원 융합기술연구부문 RFID/USN 연구본부 감시정찰무선센서네트워크연구팀 (sangjoon@etri.re.kr)
논문번호 : KICS2008-02-076, 접수일자 2008년 2월 11일, 최종논문접수일자 : 2008년 8월 14일

I. 서 론

무선 센서 네트워크에서 위치 인식 기술은 공간적으로 서로 떨어져 있는 무선 노드끼리의 통신을 이용하여 각 노드의 위치를 절대적 또는 상대적인 좌표계로 나타내는 것으로서 언제 어디서나 개인이 필요한 정보를 얻기 위해 네트워크에 접속할 수 있는 환경을 만들기 위하여 무엇보다 선행되어야 하는 기술이다. 위치 인식 기술의 적용 환경에 있어서 무엇보다 먼저 무선 센서 네트워크에서 위치 인식 기술이 고려하여야 할 요소는 1) 위치 인식 기술의 신속성, 2) 위치 데이터의 정확도, 3) 위치 인식 기술이 전체 시스템에 미치는 가중도 이다. 위치 인식 기술의 신속성은 위치가 결정 되어야 할 센서 노드의 위치 값이 결과 값으로 주어지는 데 걸리는 수행 시간을 뜻한다. 위치 인식 기법의 수행 시간은 응용 서비스의 종류에 따라 다르지만 특히 실시간으로 운영되어야 하는 네트워크에서 보다 중요하게 취급될 수 있다. 왜냐하면 하나의 노드에 있어서 위치 인식이 수행되기 전에 생성된 데이터는 해당 노드가 위치 인식 수행을 마치기 전까지 그 중요도가 모호하기 때문이다[1]. 위치 데이터의 정확도가 낮은 위치 인식 기술은 그 필요성을 떨어트린다. 마지막으로 위치 인식의 전체 시스템에 대한 가중도라는 것은 하나의 노드에서 응용 서비스를 수행함에 있어서 위치 인식 기술이 그 노드의 시스템 자원을 얼마나 차지하는가 이다. 이는 응용 서비스를 구현함에 있어서 위치 인식 기술이 센서 노드의 자원 할당에 관한 이슈가 된다는 것을 말한다. 따라서 위치 인식 기술이 시스템 자원을 얼마나 사용하는지에 대한 점도 반드시 고려해야할 점들 중의 하나이다.

그림 1은 본 논문에서 가정하고 있는 네트워크의 구성을 나타내고 있다. 우선 대규모 센서 네트워크를 가정하므로 해당 네트워크는 많은 수의 노드를 갖는다. 노드의 종류는 자신의 위치를 모르는 일반 노드와 (Unknown node) 자신의 위치를 알고 있는 참조 노드로 (Anchor Node) 분류한다. 일반 노드의 분포는 대규모 센서 네트워크의 성능을 최적화할 수 있도록 단위 면적당 같은 숫자로 배치함을 가정한다. 이는 노드가 해당 영역 내에서 가급적 균일한 분포를 갖는 것으로 해석할 수 있다. 또한 해당 네트워크에서는 전체 노드 수에 비해 비교적 적은 수의 참조 노드가 분포하고 있다. 따라서 그림 8의 대부분의 노드들이 참조 노드들의 위치 정보와 그들로부터의 거리를 멀티 홉 환경에서 활용하여 위치를 결정해야 한다는 것을 의미한다.

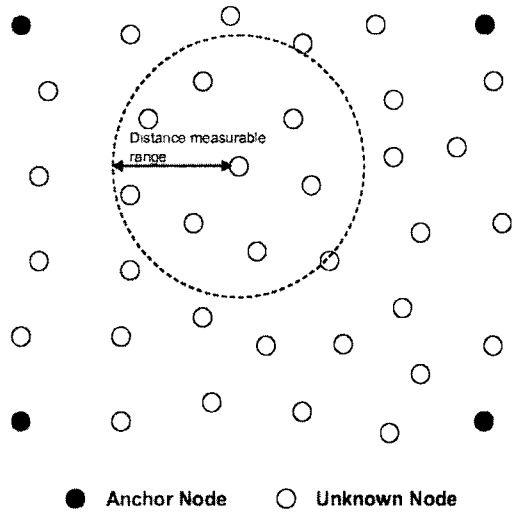


그림 1. A distance measurement example in multihop communication

두 센서 노드 사이의 거리를 측정함에 있어서 수신 신호 세기 (RSS)를 이용할 경우 신호 세기에 대한 패턴이 일정 거리 이상에서는 선형적이지 못하다. (see section 2.1) 따라서 RSS를 이용한 거리 측정에서는 일정 이상의 거리 정보는 믿을만하지 못하게 된다. 따라서 본 논문에서는 수신 신호 세기를 이용한 거리 측정의 정확도를 높이기 위해서 신호 세기에 대한 패턴이 거리에 따라 선형적으로 나타나는 거리를 거리 측정 가능 거리로 설정하고 이를 그림 1에서 보는 바와 같이 거리 측정 가능 영역(distance measurable range)이라고 정의한다. 이 영역을 벗어나는 노드와의 거리는 멀티 홉의 홉 수로 거리 측정을 하는 데, 여기에는 서로 거리 측정을 하고자 하는 노드들 사이에 배치된 중간 노드들이 일직선으로 배치되어 있는 것이 아니므로 single hop 거리 측정의 오차와 더불어 멀티 홉에서 홉 수에 대한 오차도 더해지게 된다.

거리 기반의 위치인식에서 가장 대표적인 삼선법 (Trilateration)은 고려하는 참조 노드 수만큼의 선형 방정식만 풀면 되므로 위치 인식 기법이 매우 간단하다. 따라서 계산 속도가 빠르고 시스템 가중도가 낮게 된다. 하지만 참조 노드들로부터의 거리 정보는 1차원인데 반해 위치인식을 하는 영역은 2차원 또는 3차원이므로 이에 따라 거리 측정에 대한 오차가 실제 위치 인식에의 오차로 반영되게 된다. (본 논문에서는 2차원을 가정으로 하나 더 높은 차원에도 언급하는 모든 위치인식 기법이 적용될 수 있다.) 그림 2는

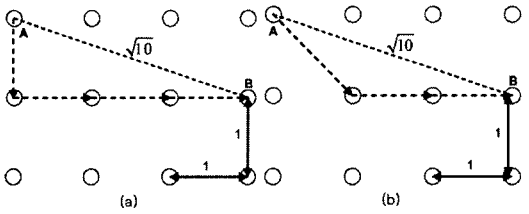


그림 2. Multihop distance measurement using hop counts between A and B ; (a) range is 1 : actual distance is $\sqrt{10}$ while hop count distance is 4. range is 1.5 : actual distance is $\sqrt{10}$ while hop count distance is $2+\sqrt{2}$

삼선법을 이용할 경우 균일한 분포를 갖는 영역에서도 오차가 발생할 수 있다는 것을 보여준다. 이는 실제 위치인식의 적용에 있어서 삼선법에 대한 정확성이 매우 낮다는 것을 보여준다.

다차원 측량법은 (Multidimensional Scaling, MDS) 하나의 노드에 있어서 그의 이웃노드들의 모든 짝에 대한 거리 정보를 갖는 행렬을 이용하여 위치 정보를 계산하므로 거리 측정 오차가 많은 상황에서도 정확도가 높다는 장점이 있다[2][3][4]. 그러나 계산에 이용되는 거리정보의 수가 너무 많아서 복잡도가 높고 따라서 계산량이 많다는 단점이 있다. 이는 시스템의 부하를 높이는 효과를 가져오므로 실제 구현에 있어서는 그 실용성이 많이 떨어진다.

대규모 센서 네트워크에서 실제 구현 가능한 위치 인식 기법을 위해서는 앞서 살펴본 삼선법보다 정확도가 높고 MDS보다는 빠르고 간단한 알고리즘이 필요하다. 따라서 본 논문에서는 노드의 분포 특성을 이용한 분산 위치 인식 기법을 제안한다. 제안하는 알고리즘은 가정하고 있는 대규모 센서 네트워크의 특징에서 출발한다. 규모가 크고 많은 수의 노드가 단위 면적당 비슷한 수로 분포하고 있다고 가정하므로 이러한 노드 분포의 특성을 이용한다. 이는 삼선법을 통한 초기화에 이어서 거리 측정 가능 영역에 있는 이웃 노드들의 떨어져 있는 거리를 하나의 평균을 갖는 확률 분포 함수 (Probability density function, pdf)로 모델링한다. 그리고 하나의 노드에 대해 그의 모든 이웃 노드들의 pdf로 Maximum likelihood 함수를 구성하여 이에 최적화된 위치로 해당 노드의 위치를 보정한다. 본 논문에서 제안하는 노드 분포를 고려한 분산 위치 인식 기법의 장점은 다음과 같다. 첫 번째로 초기화에 있어서 삼선법을 사용하고, 자신의 이웃 노드들만을 고려한 보정 알고리즘을 사용함으로써 계산량을 줄여 MDS보다 간단하고 신속한 위치인식이 이루어지도록 한다. 두 번째로 보정 알고리즘을 통해 위치 오차를 줄임으로써 삼선법보다 더욱 정확한 위치 정보를 제공한다. 세 번째로 위치 인식 기법이 각 노드에서 분산적으로 수행될 수 있도록 함으로써 시스템 전체의 부하를 줄이도록 한다.

본 논문의 구성은 다음과 같다. 먼저, Ⅱ장에서는 RSS를 이용한 거리 측정의 특징과 함께 앞서 언급한 종래의 위치 인식 기법들에 대해 자세히 알아본다. Ⅲ장에서는 대규모 센서 네트워크에 적합한 간소화된 계산을 통한 분산 위치 인식 기법에 대해 소개한다. Ⅳ장에서는 Ⅲ장에서 제안한 기법의 PC simulation의 결과를 다른 기법들과 비교한다. Ⅴ장에서는 Ⅲ장에서 제안한 기법의 실제 구현에 있어서 crossbow사의 MicaZ mote를 이용한 위치 인식의 실제 예를 살펴보고 Ⅵ장에서는 결론을 맺는다.

Ⅱ. RSS를 이용한 거리측정과 종래의 위치 인식 기술들

2.1 RSS를 이용한 거리측정

RSS는 수신 신호의 세기가 거리에 따라 변하는 특징을 이용하여 두 노드 사이의 거리를 측정하는 방법이다. 이론적으로는 거리 손실 모델에 따라 신호의 세기가 변하는 데 본 단원에서는 실제 구현에 있어서 수신 신호 세기가 거리에 따라 어떠한 특징을 갖고 있는지에 대해 살펴본다. RSS를 이용한 거리 측정은 구현 관점에 있어서 간단하여 실용성이 높다는 장점이 있다. 하지만 신호 세기의 패턴이 노드의 외관적인 모양과 주변의 지형적인 특성에 따라 많은 영향을 받는다는 단점이 있다. 이는 직접적으로 거리 측정에 대한 오차로 반영된다.

RSS 측정의 구현에 있어서 본 논문에서는 MicaZ를 기준으로 서술한다. MicaZ에는 무선 통신 방법으로 IEEE 802.15.4의 라디오 스택이 구현된 CC2420 칩이 장착되어 있다. TinyOS-2.x를 기반으로한 RSS 측정 프로그램은 다음 세 가지의 특징을 갖는다. 첫 번째는 송신 신호 세기 설정이다. 이는 다음과 같은 함수를 사용하여 설정할 수 있다.

call CC2420Packet.setPower(&packet, 31);

여기서 packet은 메시지 형식을 따르는 변수명이다. CC2420 데이터 시트를 살펴보면 신호 세기 오프셋 31은 0 [dBm]에 해당한다. 이와 같이 송신 신호를 고정하면 수신 측에서는 신호 세기의 경감

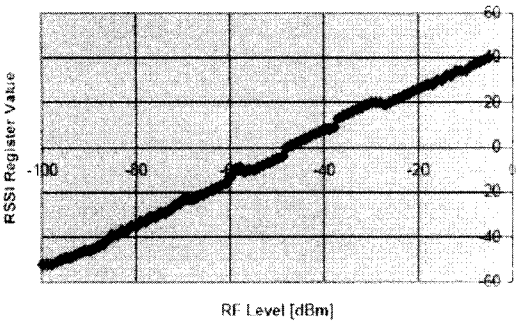


그림 3. Typical RSSI value versus input RF level in dBm

을 측정할 수 있다. 두 번째로 Received Signal Strength Indicator (RSSI) 값의 측정이다. RSSI는 수신 신호에 있어서 8 심벌동안의 신호 세기를 평균한 값으로 이는 신호를 수신할 때마다 metadata의 한 부분으로서 칩 내부의 레지스터에 기록되게 된다. 이는 다음과 같은 함수를 사용하여 레지스터 내부에 기록되는 값을 읽어올 수 있다.

call CC2420Packet.getRSSI(msg);

여기서 msg는 메시지에 대한 포인터이다. 위 함수의 반환 값은 실제 수신 신호 세기가 아닌 RSSI_VAL로 표현되는 값을 반환해 주는 데 MicaZ의 설명서를 보면 그림 3을 이용하여 실제 수신 신호 세기에 대한 값을 dBm 단위로 구할 수 있다.

마지막 특징은 수신 신호 세기를 이용하여 실제 거리를 구하는 거리 손실 모델인데, 먼저, CC2420에서 제공하는 다음 수식을 이용하여 RSSI_VAL 값을 실제 수신 신호 세기로 바꿀 수 있다.

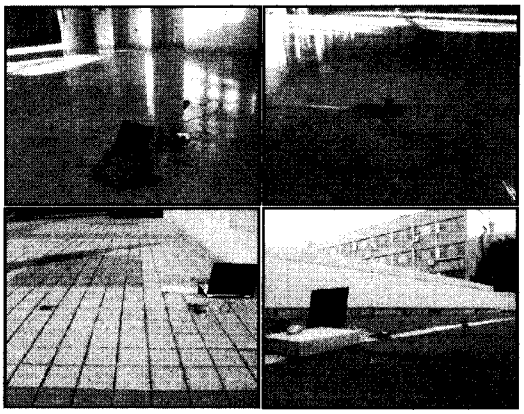
$$P = RSSI\_VAL - RSSI\_OFFSET \ [dBm]$$

여기서 P는 수신 신호 세기의 [dBm] 값이고, RSSI_OFFSET은 경험적으로(empirically found) 산출된 값으로 그림 3 에 근거한다. 다음으로, IEEE 802.15.4의 표준을 살펴보면 다음과 같은 수식을 사용하여 거리 값을 미터 단위로 구할 수 있다.

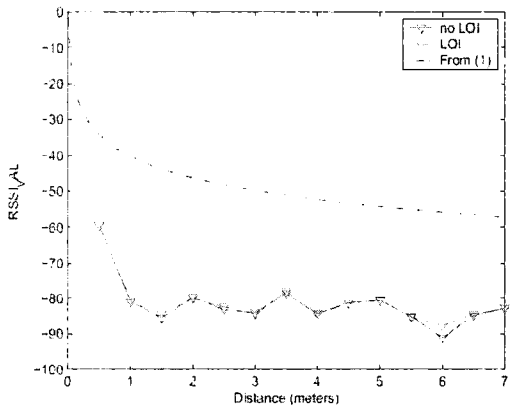
$$d = 10^{\frac{(P_t - P_r - 10.2)}{20}} \quad \text{for } d < 8m \qquad (1)$$

$$d = 8 \times 10^{\frac{(P_t - P_r - 58.5)}{33}} \quad \text{for } d > 8m$$

여기서 P_t는 송신 신호 세기이고, P_r은 수신 신호 세기이다. (8 m는 FCC에서 언급하고 있는 측정



(a) RSSI measurement experiment



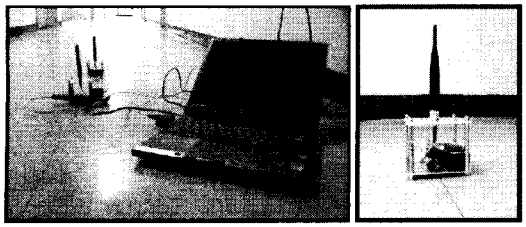
(b) RSSI experiment result

그림 4. RSSI measurement experiment photos and result graph

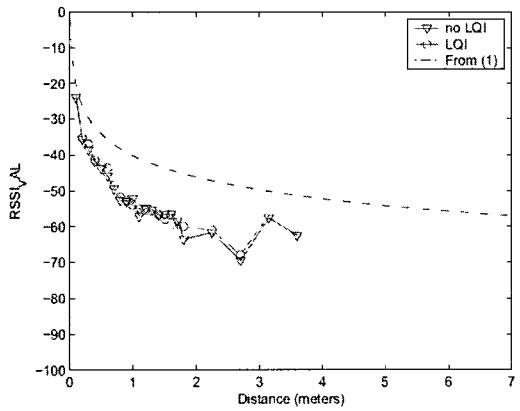
방식 : 주변에 아무런 장벽이 없는 트인 공간에서 지상 1 m 높이 위에서 측정할 때, 반사파가 생기지 않는 최대 거리이다. 즉, 수식 (1)은 8 m 까지는 direct line-of-sight를 가정함을 알 수 있다.)

본 논문에서 거리 측정을 위해서 두 번의 실험을 진행하였다. 첫 번째로 진행한 실험은 그림 4 에 도시되어있다. 그림에서 파란 선은 Link Quality Indication (LQI) 값을 고려하지 않은 것이고 빨간 선은 LQI 값을 고려한 수신 신호 세기이다. 그림에서 알 수 있다시피 RSSI 값은 거리에 따라 감소 추세를 보이긴 하나 그 패턴이 매우 불규칙하여 거리 측정에는 다소 적합하지 않아 보인다. 그럼에도 불구하고 매우 좁은 거리, 예를 들어 1 m 내의 거리에서는 다소 선형적인 신호 세기의 감소가 관찰되었다.

두 번째 실험은 앞서 진행된 실험에서 나타난 선형적인 거리 영역에 초점을 맞추었다. 본 실험의 목적은 이러한 선형적인 구간을 늘리고 다소 이론적인 수치에 비슷한 값을 얻고자 함이다. 따라서



(a) RSSI measurement experiment with cases



(b) RSSI experiment result with cases

그림 5. RSSI measurement experiment photos and result graph with cases

MicaZ 노드에 적합한 케이스를 설계하였다. 그림 5(a)는 MicaZ노드에 케이스를 제작하여 실험하는 모습이다. Crossbow사에서 제공하는 하드웨어 사용자 메뉴얼에 따라 그라운드 플래인을 가로 세로 각각 7 cm가 되도록 금속판을 깎고, 2.4 GHz에 공진을 맞춘 SMA 타입의 안테나를 해당 그라운드 플레인의 중앙에 설치하였다. 그림 5(b)는 제작한 케이스로 수신 신호 세기를 측정한 그래프이다. 본 실험은 MicaZ 노드가 지상에서 케이스의 높이만큼 올라간 곳에서의 실험이라고 할 수 있다. 따라서 FCC에서 제안하는 실험 높이인 1 m 보다 훨씬 낮은 높이이다. 그러므로 신호의 방사 패턴에 의한 반사파 형성이 보다 짧은 거리에서 생기므로 실제 실험 결과에서도 RSSI 값의 진동이 8 m 보다 훨씬 짧은 거리에서부터 나타나는 것을 알 수 있다. 하지만 RSSI 값의 변화가 1.5 m 까지 다소 규칙적으로 변화하였다는 것이 두 번째 실험의 의의라고 할 수 있다. 결론적으로 그림에서 보다시피 LQI를 고려한 빨간 색의 경우 선형적인 구간이 다소 늘어남을 볼 수 있다. 또한 수신 신호 세기의 값이 이론적인 수치에 조금 더 근접함을 볼 수 있다.

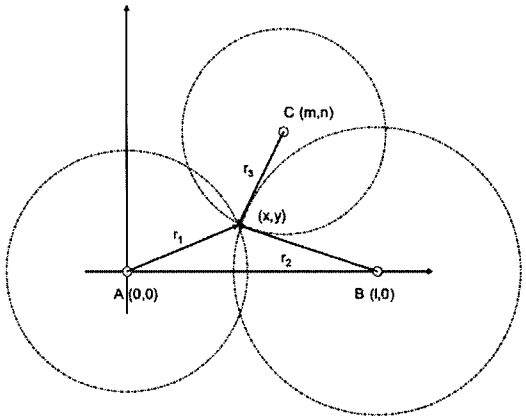


그림 6. A simple example of trilateration

2.2 종래의 위치인식 기술들

2.2.1 Trilateration

Trilateration, 즉, 삼선법은 세 개의 참조노드들로부터의 측정 거리를 이용하여 하나의 센서노드의 위치를 계산하는 기본적인 위치인식 기법이다. 이는 센서 필드를 2D, 즉, 2차원 평면으로 가정하였을 때의 방법이고 일반적으로 n차원에서 위와 같은 위치인식 기법을 수행하기 위해서는 자신의 위치를 알고 있는 참조노드의 수가 n+1개가 되어야 한다.

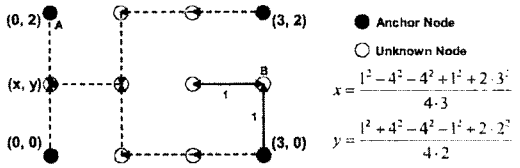
우선 세 개의 참조 노드를 각각 A, B, C라고 하였을 때, 첫 번째 참조 노드인 A를 2차원 평면의 (0,0)에 있다고 가정하자. 또한 두 번째 참조 노드인 B를 A로부터 x축으로 l만큼의 거리에 떨어져 있다고 하자. 그러면 마지막 참조 노드인 C는 (m, n)에 있게 된다. 여기서 자신의 위치를 알고자 하는 센서 노드의 위치를 (x, y)라고 하면 다음과 같은 간단한 선형 방정식을 세울 수 있다.

$$r_1^2 = x^2 + y^2 \tag{2}$$

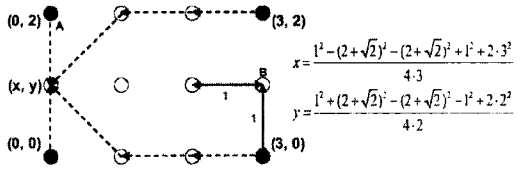
$$r_2^2 = (x-l)^2 + y^2 \tag{3}$$

$$r_3^2 = (x-m)^2 + (y-n)^2 \tag{4}$$

여기서 x의 값은 간단히 (2) - (3) 의 연산을 통해 구할 수 있다. 하지만 여기서 y의 해는 2개가 된다. 따라서 원하는 y의 값을 구하기 위해서는 수식 (4)를 이용하여 y에 대한 1차식을 만들고 이에 앞서 구한 x의 값을 대입시켜야 한다. 결과적으로 거리에 대한 값이 완벽히 정확하다고 가정할 때 2차원 평면에서 임의의 센서 노드의 위치는 3개의



(a) range of 1 : distances from 4 anchors are (1,4,4,1). Computed position is (-1, 1) which has an error to the actual position.



(b) range of 1.5 : distances from 4 anchors are $(1.2+\sqrt{2}.2 + \sqrt{2}.1)$. Computed position is (-0.2761, 1) which has an error to the actual position.

그림 7. Examples of trilateration error for various measurable range

참조 위치 값과 이에 이르는 거리를 이용하여 구할 수 있다.

그림 7은 삼선법을 멀티 홉 환경에서 홉 수에 기반을 둔 거리를 이용하여 계산할 경우 측정 가능 거리 영역에 따른 계산 오차를 보여준다. 그림 7(a)에서는 거리 영역이 1일 때이고 그림 7(b)는 거리 영역이 1.5일 때인데, 계산 결과에서 보다시피 보다 넓은 거리 영역을 갖는 환경에서 위치 오차가 적은 것을 관찰할 수 있다. 참고로 거리 영역이 매우 넓은 환경에서 거리 오차가 없는 경우에는 완벽한 위치 계산이 가능하다.

2.2.2 Multidimensional Scaling

MDS는 본래 수리심리학에서 사용되는 기법이다[2]. 많은 분야에서 이 기법을 사용하고 있지만 이를 통해 얻고자 하는 목적은 모두 같다. 어떠한 점들의 집합에 있어서, 그들의 위치는 모르나 모든 대칭점들에 대한 거리는 알고 있을 때, MDS는 이 점들의 상대적인 위치 좌표계와 그로 인한 점들의 위치를 결정한다. 이러한 기법은 무선 센서 네트워크의 위치 인식에 있어서 각종 하드웨어로부터 얻을 수 있는 거리 측정 결과를 적용하면 참조 노드 또는 Global Positioning System (GPS) 등과 같은 추가적인 정보 또는 장치를 이용하지 않아도 구성원들 간의 상대적인 위치를 구할 수 있는 것이다. 또한 이는 거리 측정에 있어서 피할 수 없는 오차에 대해서 상당히 정확한 위치좌표를 제공한다는 특징을 갖고 있다. 본 섹션에서는 "classical metric MDS"라고 하는 MDS의 기법에 대해 다룬다. 여기서 classical은 "dissimilarity" 또는 거리 정보를 담고 있는 하나의 행렬만을 이용하는 것을 의미하고 metric은 이러한 dissimilarity 정보가 거리 측정과 같은 정량적인 특성을 갖고 있다는 것을 의미한다[4]. 하나의 네트워크에 위치인식을 하고자 하는 센서 n개가 있다고 했을 때, classical metric MDS는 다음과 같은 방식으로 진행된다.

① 모든 대칭점에 대한 거리정보를 담고 있는 symmetric 행렬 $D = [d_{ij}]$를 만든다.
여기서 $d_{ii} = 0$ 이고, 모든 i, j, k 에 대해서 $d_{ij} + d_{ik} \geq d_{jk}$를 만족하여야 한다.

② Symmetric 행렬 J를 다음과 같이 만든다.

$$J_{n \times n} = I_{n \times n} - \frac{1}{n} e^T e$$
$$e_{1 \times n} = [1, \ldots, 1]$$

③ $D^2 = [d_{ij}^2]$ 일 때, 다음과 같이 B 행렬을 만든다.

$$B_{n \times n} = -\frac{1}{2} JD^2 J = XX^T$$

④ B를 eigen-decomposition 하여 $B = UVU^T$를 구한다.

⑤ V에서 d개만큼의 eigenvalue 들을 그들 중 값이 가장 큰 순서대로 뽑아내어 V_d라고 하는 정방향 행렬을 만든다. 여기서 d는 센서 노드들이 분포되어 있는 영역의 차원을 나타낸다.

⑥ U에서 V_d의 값들에 해당되는 d개의 eigenvector들을 골라내어 U_d를 만들어 낸다.

⑦ d차원의 위치 좌표 행렬을 X_d라고 할 때, X_d는 다음과 같이 구한다.

$$X_d = [X_1, X_2, \ldots, X_n]'$$
$$X_d = U_d V_d^{\frac{1}{2}}$$

여기서 $V_d^{\frac{1}{2}}$는 각 eigenvalue들의 제곱근으로 이루어진 벡터를 뜻한다.

Ⅲ. Node Distribution based Localization

대규모 센서 네트워크에서 적은 용량을 활용하여 분산 위치인식 기법을 신속하게 수행하기 위해서

노드의 분포 특성을 이용한 위치 인식 기법(NDBL)을 제안한다. 이는 더욱이 참조 노드의 개수가 적어 모든 노드가 참조 노드의 통신거리 안에 속할 수 없는 상황에서도 거리오차에 따른 위치 인식에의 영향을 줄일 수 있는 방법이다. 우선 해당 노드와 그의 이웃 노드들과 떨어진 거리 x를 수식 (5)와 같이 평균 μ, 분산 σ^2을 갖는 Gaussian pdf로 모델링 한다.

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}exp-\frac{(x-\mu)^2}{2\sigma^2} \tag{5}$$

하나의 노드에 있어서 그의 위치를 정할 때, 해당 노드의 이웃 노드들과의 거리에 관한 probability가 가장 높은 위치로 자신의 위치를 정한다. 이는 다음과 같이 Maximum likelihood로 표현할 수 있다.

$$\begin{aligned}\hat{\boldsymbol{x}_i}&=\arg\max_{\boldsymbol{x}_i}\log f(\boldsymbol{x}_i\mid \boldsymbol{x}_j,\ j\in N(i)) \\ &=\arg\max_{\boldsymbol{x}_i}\log\Big(\prod_{j\in N(i)} Gaussian(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|,\mu_i,\sigma_{ij}^2)\Big) \\ &=\arg\min_{\boldsymbol{x}_i}\sum_{j\in N(i)}\frac{1}{2\sigma_{ij}^2}(\|\boldsymbol{x}_i-\boldsymbol{x}_j\|-\mu_i)^2\end{aligned} \tag{6}$$

1) $\boldsymbol{x}_i$: a coordinate of the desired node i.
2) $\boldsymbol{x}_j$: a coordinate of the neighbor node j.
3) $N(i)$: a set of neighbor nodes of node i.
4) $\hat{\boldsymbol{x}_i}$: estimated coordinate of node i.
5) $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|$: Euclidean distance between a coordinate of node i and a coordinate of node j.
6) μ_i : mean distance of node i from neighbor node $j\in N(i)$.
7) σ_{ij}^2 : variance of distance between node i and j.

이를 다음과 같이 다시 나타낼 수 있다.

$$\hat{\boldsymbol{x}_i}=\arg\min_{\boldsymbol{x}_i}\sum_{j\in N(i)}\omega_{ij}(d_{ij}(X)-\mu_i)^2 \tag{7}$$

1) $\omega_{ij}\triangleq\frac{1}{2\sigma_{ij}^2}$: weight between node i and j

2) $d_{ij}(X)\triangleq\|\boldsymbol{x}_i-\boldsymbol{x}_j\|=\sqrt{(\boldsymbol{x}_i-\boldsymbol{x}_j)^T(\boldsymbol{x}_i-\boldsymbol{x}_j)}$ where X is a coordinate matrix of all nodes and T is transpose.

3.1 The NDBL Cost Function

제안하는 분산 위치 인식 기법에 있어서 가장 중요한 요소는 하나의 노드 주변에 있는 이웃 노드들의 위치와 이에 따른 분포 특성을 결정하는 것이다. 또한 이런 분포 특성을 이용하여 해당 노드의 위치 좌표를 결정하는 방법도 포함한다. 따라서 이러한 특징들을 반영하는 위치 인식 기법을 수행하기 위한 global cost function을 다음과 같이 정의한다.

$$S=\sum_{i=1}^{n}\sum_{j\in N(i)}\omega_{ij}(d_{ij}(X)-\mu_i)^2 \tag{8}$$

위의 식은 노드 분포의 특성을 Gaussian으로 모델링 한다는 의미를 갖는다. 노드의 위치좌표를 Gaussian 확률 모델에 기반을 두어 결정하는 것은 대규모 센서 네트워크에서 노드의 분포가 asymptotically uniform할 수 있다는 가정을 갖는다. 식 (8)을 분산 위치 인식 기법으로 구현하기 위해 다음과 같은 식으로 다시 쓸 수 있다.

$$S=\sum_{i=1}^{n}S_i$$

여기서 S_i는 local cost function으로 다음과 같이 재정의 할 수 있다.

$$S_i=\sum_{j\in N(i)}\omega_{ij}(d_{ij}(X)-\mu_i)^2 \tag{9}$$

3.2 Minimizing the NDBL Cost Function

앞서 다룬 classical metric MDS와는 달리 식 (8)과 식 (9)에는 closed form의 해가 존재하지 않는다. 위치인식을 하고자 하는 각각의 노드가 이웃 노드들로부터 추정된 위치를 넘겨받는다는 가정 하에 SMACOF(Scaling by MAjorizing a COmplicated Function[4])에서와 같은 quadratic majorizing function을 이용하여 $S_i=S(\boldsymbol{x}_i)$를 최소화 할 수 있다. 이 방법은 일련의 nonincreasing STRESS 값들을 만들어낸다는 특징을 갖는다. 다시 말하면 하나의 majorizing function을 정의하고 이를 최소화함으로써 하나의 iterative minimization scheme을 만들어낼 수 있다는 것이다. 식 (9)를 다시 쓰면 다음과 같다.

$$\begin{aligned}S(\boldsymbol{x}_i)=&\sum_{j\in N(i)}\omega_{ij}d_{ij}^2(X)-2\mu_i\sum_{j\in N(i)}\omega_{ij}d_{ij}(X)\\&+\mu_i^2\sum_{j\in N(i)}\omega_{ij}\end{aligned}$$

다음과 같은 Cauchy-Schwartz 부등식을 사용하여

$S(x_i)$를 majorize 할 수 있는 $T(x_i, y_i)$를 정의할 수 있다.

$$T(x_i, y_i) = \frac{d_{ij}(X) d_{ij}(Y)}{d_{ij}(Y)} \geq \frac{(x_i - x_j)^T (y_i - y_j)}{d_{ij}(Y)}$$

$$\begin{aligned} T(x_i, y_i) \triangleq & \sum_{j \in N(i)} \omega_{ij} d_{ij}^2(X) \\ & - 2\mu_i \sum_{j \in N(i)} \frac{\omega_{ij}}{d_{ij}(Y)} (x_i - x_j)^T (y_i - y_j) \\ & + \mu_i^2 \sum_{j \in N(i)} \omega_{ij} \end{aligned}$$

이제 majorizing 기법을 통해 $S(x_i)$를 최소화 하는 것은 $T(x_i, y_i)$의 최소값을 찾는 것으로 간소화 되었다.

$$\frac{T(x_i, y_i)}{x_i} = 0 \qquad (10)$$

$$\begin{aligned} & = 2 \sum_{j \in N(i)} \omega_{ij} x_i - 2 \sum_{j \in N(i)} \omega_{ij} x_j \\ & - 2\mu_i \sum_{j \in N(i)} \frac{\omega_{ij}}{d_{ij}(Y)} y_i + 2\mu_i \sum_{j \in N(i)} \frac{\omega_{ij}}{d_{ij}(Y)} y_j \end{aligned}$$

이제 식 (10)을 이용하여 노드 i에서의 위치 추정에 대한 update formula를 만들 수 있다.

$$x_i^{(k+1)} = a_i (X^{(k)} b_i^{(k)})$$

여기서 $x_i^{(k+1)}$는 $k+1$번째 iteration에서 노드 i의 위치좌표 벡터이고 $X^{(k)}$는 k 번째 iteration에서 노드 i와 그의 이웃 노드들의 위치좌표 벡터 열로 구성된다. 또한 a_i는 다음과 같이 표현할 수 있다.

$$a_i^{-1} = \sum_{j \in N(i)} \omega_{ij} \qquad (11)$$

$b_i^{(k)}$는 다음과 같이 구성된다.

$$b_i = \mu_i \sum_{j \in N(i)} \left(\frac{\omega_{ij}}{d_{ij}(X^{(k)})} \right), \qquad (12)$$

$$b_j = \omega_{ij} \left(1 - \frac{\mu_i}{d_{ij}(X^{(k)})} \right), \quad j \in N(i)$$

3.3 Algorithm

본 논문에서 제안하는 NDBL 알고리즘의 수행은 각 노드에 있어서 분산적으로 이루어진다. 따라서 하나의 노드는 자신이 설정한 주변 노드에 대한 정보만 있으면 자신의 위치를 업데이트할 수 있다. 알고리즘의 실제적인 구현에 있어서 iteration에 대한 개략적인 동기는 이웃 노드의 좌표에 대한 정보를 수신함에 있어서 iteration 수를 비교함으로써 달성할 수 있다. 하지만 개별적인 노드에 있어서 순서가 정해져 있는 것은 아니다. 다음은 전체 네트워크 관점에서 NDBL 알고리즘의 수행을 초기화, 순환 루프, 종료의 순서로 설명한다.

3.3.1 초기화

① 이웃 노드 탐색 및 평균 거리 설정 : 각 노드에서 통신 거리 내에 있는 노드들은 모두 이웃 노드의 후보가 될 수 있다. 하지만 실제 이웃 노드 집합에 들어올 수 있는 노드들은 그 노드들로부터 해당 노드로 신호를 보냈을 때 해당 노드에서 수신 신호의 세기를 판단하여 거리측정 가능 범위에 들어오는 노드들만 해당 노드의 이웃노드로 정한다. 거리 측정 가능 범위에 있어서는 알고리즘의 정확도를 높이기 위해 비슷한 거리에 떨어져 있는 노드의 수가 많아지도록 하는 거리로 정한다. 또한 이 비슷한 거리를 평균 거리 μ_i로 설정한다.

② 초기 좌표 계산 : 위치 인식 알고리즘의 수행을 위해 참조 노드의 수는 2차원 평면에서 3개 이상, 3차원 공간에서 4개 이상이 있다고 가정한다. Dijkstra 알고리즘 등을 사용하여 각 참조노드와 해당 노드와의 hop count를 계산하고 여기에 평균 거리를 곱한 값으로 그 참조노드로부터 떨어진 거리를 추정한다. 이러한 거리 정보들을 바탕으로 tri- lateration 또는 multilateration을 수행함으로써 초기 위치 좌표를 계산한다.

③ Iteration 수 결정 : 이론적인 iteration 수는 global cost function의 값의 변화가 일정 범위 내에 들어오는 criterion을 사용해야 한다. 하지만 대규모 무선 센서 네트워크에서 실용적인 구현을 위해서는 보다 직관적이고 결정적인 방법으로 iteration의 수를 결정하는 것이 알맞다. 따라서 iteration 수는 해당 네트워크 시스템의 성능 및 시스템이 요구하는 위치인식 알고리즘의 수행 시간에 맞추어서 정하는 것이 바람직하다.

3.3.2 Iteration

① Coefficient 계산 : 각 노드에 대해서 식 (11)과 식 (12)를 사용하여 다음 iteration에서 사용할

노드 i의 위치 좌표를 계산한다. 이전 단계의 위치좌표에 있어서 첫 번째 iteration에서는 초기 좌표 값을 사용한다.

② Iteration 수 증가 : iteration 횟수 k를 1만큼 증가 시킨다.

3.3.3 Termination Policy

매 iteration 마다 k의 값을 검사하여 주어진 모든 iteration이 끝나면 알고리즘 수행을 종료한다.

위와 같은 알고리즘의 수행에 따라 대규모 무선 센서 네트워크에서 효율적인 방법으로 위치인식을 수행할 수 있다.

Ⅳ. Simulation on MATLAB

본 장에서는 MATLAB을 이용하여 분산 위치 인식 알고리즘의 성능을 검증하고 측정 오차에 대한 효과를 살펴본다. MATLAB을 이용한 실험에서는 다음과 같은 실험 환경을 가정한다.

① 참조 노드 배치 : Anchor Node는 총 4개이고, 각 코너에 배치되어 있다.

② 분포 영역 : 노드 분포 영역은 가로 90, 세로 90의 정사각형으로 설정한다.

③ 노드 수 : 분포 영역 내에 참조 노드를 제외한 일반 노드의 수는 60 개로 설정한다.

④ 통신 가능 거리 : IEEE 802.15.4의 표준을 따르는 cc2420 radio chip의 실내에서 통신 가능한 영역을 참조하여 각 노드의 통신 가능 거리는 20으로 설정한다.

⑤ 측정 방식 : 본 논문에서 초기화 방법은 모두 멀티 홉 환경의 삼선법으로 동일하다. 즉, 하나의 노드는 각 참조노드로부터 떨어진 거리를 그 노드로부터의 hop count와 이웃 노드들과의 평균 측정 거리에 근거하여 계산한다. 이 거리정보와 참조노드의 위치 정보를 활용하여 자신의 초기 위치 값을 계산한다. 후에, 자신의 주변에 있는 이웃노드들의 위치정보를 참조하여 보정단계를 수행하는 데, 다음과 같은 방법을 사용한다.

(1) Trilateration : 자신의 위치 좌표에 대한 보정에 있어서 앞서 설명한 Trilateration기법을 사용한다.

(2) NDBL : 본 논문에서 제안하는 방법을 이용하여 자신의 위치 좌표를 Update 한다.

(3) dwMDS : [3]에서 제안한 방법을 이용하여 위치 좌표를 보정한다.

⑥ 수행 시간 : 본 논문에서는 위 3가지 측정, 보정 방식에 있어서 알고리즘수행 시간을 측정하였다. 위치 좌표를 계산하는 모든 노드에 있어서 한 번의 계산을 하나의 cycle로 설정하였고 알고리즘 수행시 매 cycle마다 수행 시간을 초 단위로 측정하여 알고리즘의 수행시간을 각각의 보정방법에 대해 비교할 수 있도록 하였다.

4.1 Simulation Results

그림 8에서는 Grid 분포를 가정하고 거리측정에 대한 오차가 없는 환경에서 각각의 측정 방식에 대한 위치좌표 계산 결과를 나타낸 것이다. 그림에서 o는 노드의 실제 위치를, x는 추정된 위치를 나타낸다. 따라서 o 와 x 사이의 직선거리는 추정 위치의 오차이다. (a),(b),(c)는 각각의 알고리즘에 대한 실제 노드 배치와 위치 계산에 따른 보정 정도를 보여준다. (d)에서는 각각의 알고리즘 수행시 전체 오차를 노드 수로 나눈 평균 거리 오차의 값을 매 cycle마다 걸리는 시간을 초 단위로 계산하여 보여준다.

그림 9에서는 완벽한 Grid가 아닌 분포를 가정하고 거리측정에 대한 오차가 없는 환경에서 각각의 측정 방식에 대한 위치좌표 계산 결과를 나타낸 것이다. 그림 8과 9에서는 거리 측정에 대한 오차가 없는 것을 가정 하였으므로 Trilateration의 경우 보정 계산에 대한 이득이 없다. 따라서 Trilateration은 한번의 cycle만 수행하고 알고리즘을 마친다. 여기서 주목할 만한 점은 NDBL의 경우 dwMDS보다 알고리즘을 통한 최종 위치 보정에 따른 이득이 좋지는 않지만 초기 알고리즘 수행시 보다 더 간단한 계산을 이용하므로 오차를 줄이는 속도가 탁월하게 좋다. 만약 위치 인식 알고리즘이 수행되어야 하는 시간제한이 매우 짧다면, 예를 들어 1 ~ 2분의 시간 안에 알고리즘 수행이 끝나야 한다 고하면 NDBL의 알고리즘 수행 이득이 dwMDS의 그것보다 더 좋게 나타날 것이라고 생각할 수 있다.

다음으로는 거리 측정에 대한 오차가 있을 때를 가정하고 실험하였다. 그림 10에서는 Grid 분포를 가정하고 거리측정에 대한 오차가 실제 거리의 50%인 환경에서 각각의 측정 방식에 대한 위치좌표 계산 결과를 나타낸 것이다. 또한 그림 11에서는 완벽한 Grid가 아닌 분포를 가정하고 거리측정에 대한 오차가 50%인 환경에서 각각의 측정 방식에 대한 위치좌표 계산 결과를 나타낸 것이다. 거리 측정에 대한 오차가 포함될 때에는 위에서 언급한 NDBL의 장점이 더욱 두드러진다.

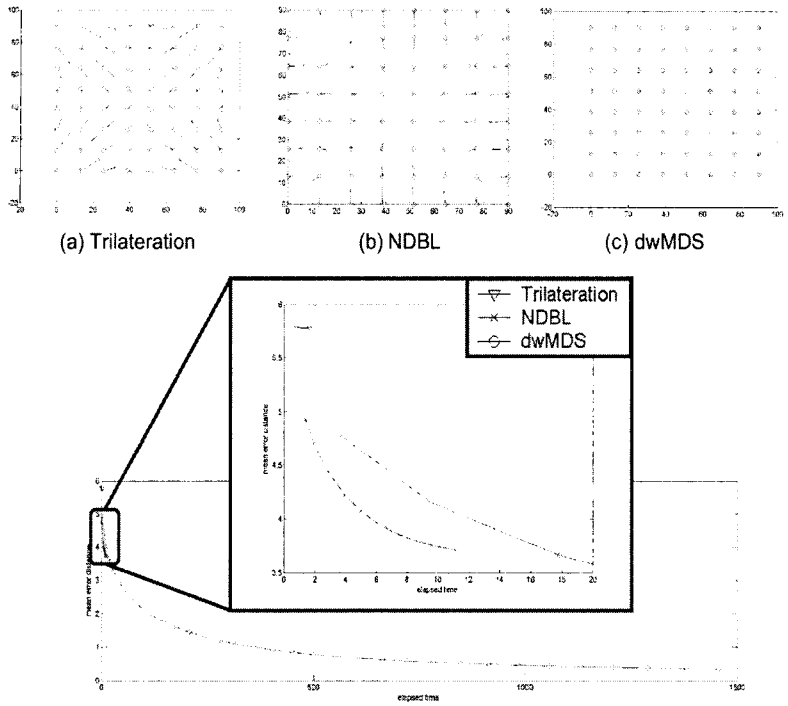


그림 8. Comparison of three localization algorithms in grid deployment and no distance error : (a) shows Trilateration, (b) shows NDBL algorithm, (c) shows dwMDS algorithm and (d) shows the elapsed time for each algorithm

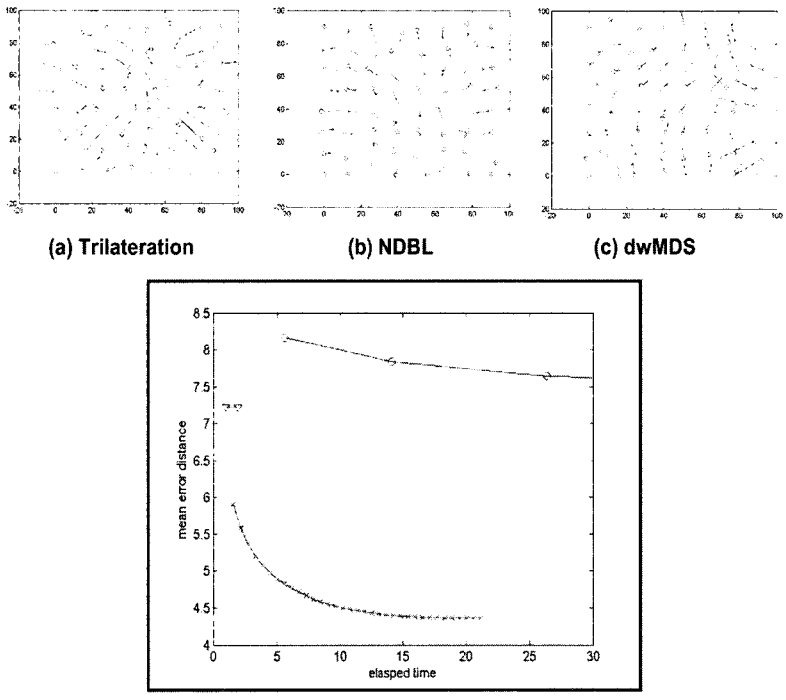


그림 9. Comparison of three localization algorithms in semigrid deployment and no distance error : (a) shows Trilateration, (b) shows NDBL algorithm, (c) shows dwMDS algorithm and (d) shows the elapsed time for each algorithm

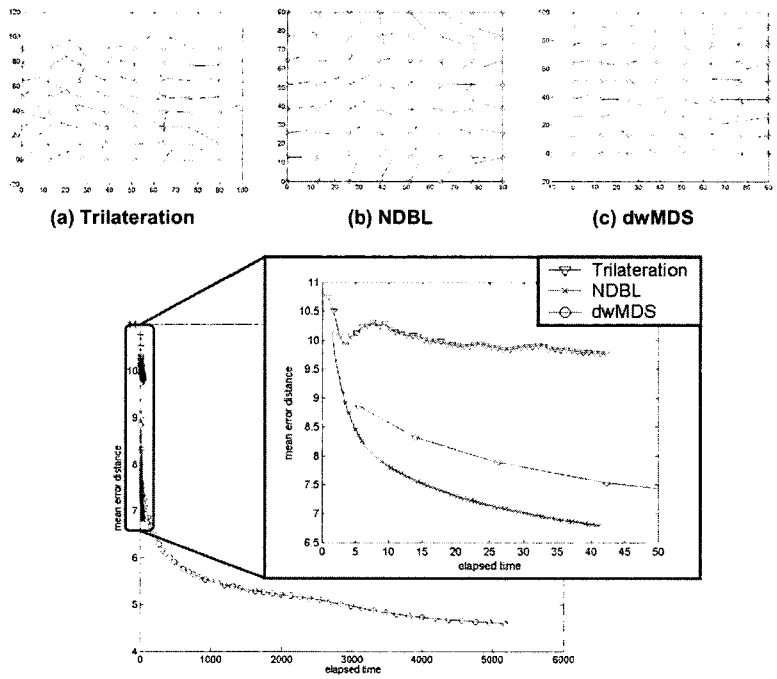


그림 10. Comparison of three localization algorithms in grid deployment and 50% distance error : (a) shows Trilateration, (b) shows NDBL algorithm, (c) shows dwMDS algorithm and (d) shows the elapsed time for each algorithm

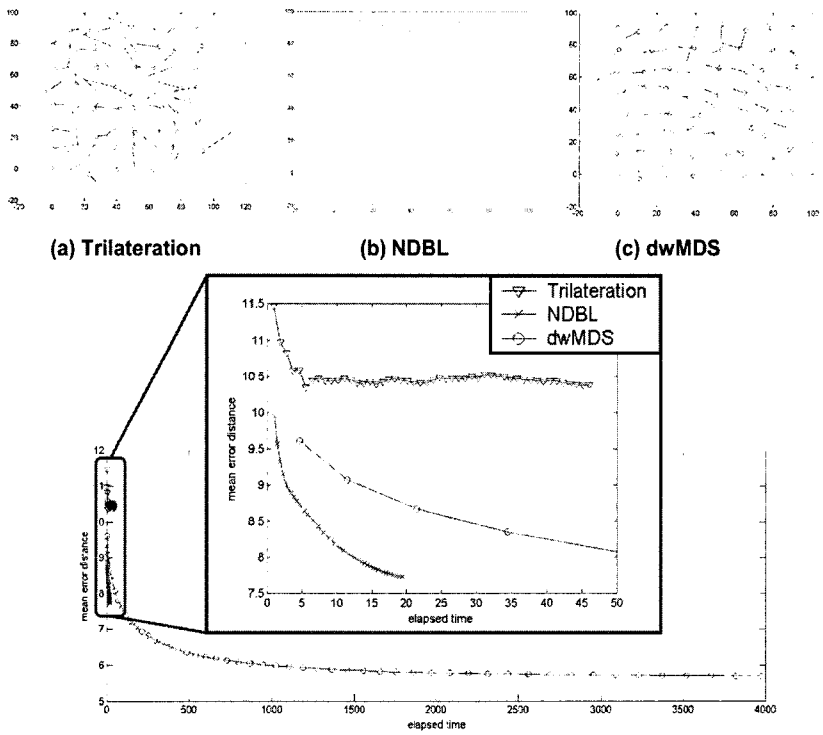


그림 11. Comparison of three localization algorithms in semigrid deployment and 50% distance error : (a) shows Trilateration, (b) shows NDBL algorithm, (c) shows dwMDS algorithm and (d) shows the elapsed time for each algorithm

무선 센서 네트워크에서 위치 인식은 초기 분포시 수행되어야하는 작업이다. 노드의 분포에 있어서 초기화 작업의 하나라고 할 수 있다. 이러한 초기화 작업에는 라우팅, 클러스터링과 같은 여러 작업이 포함되어 있으므로 위치인식을 위한 하드웨어적인 여분은 상당히 부족하다고 할 수 있다. 이러한 점에서 볼 때, dwMDS의 수행시간은 거의 비현실적임을 볼 수 있다. 따라서 본 논문에서는 위치 인식에 대한 신속성과 효율성의 특징으로 하는 NDBL을 제안하였다.

V. Implementation on MICAZ

본 장에서는 분산 위치 인식 기법의 실제 구현에 있어서 crossbow사의 MicaZ mote에 TinyOS를 이용한 nesC 코드에 대한 설명과 위치 인식 기법의 수행 결과를 다룬다.

5.1 TinyOS and Crossbow MicaZ

TinyOS는 하드웨어로 제작된 무선 센서 장비를 구동하는 데 필요한 OS를 nesC라는 언어로 제작할 수 있는 program이다. Sourceforge에서 working group으로 출발한 TinyOS는 현재 2.x 버전이 출시된 상태이고 웹페이지 www.tinyos.net 에 해당 프로그램과 이를 이용하기 위한 문서 및 각종 자료들이 게시되어 있다.

MicaZ는 IEEE 802.15.4를 구현한 chipcon사의 cc2420을 radio stack으로 사용하는 wireless sensor mote이다. MCU로는 ATmega128을 사용하고 programable memory로는 128KB, RAM은 4KB를 지원한다.

5.2 The NDBL Algorithm

그림 12는 MicaZ를 사용한 분산 위치 인식 기법의 구현에 있어서 사용된 Topology이다. 여기서 Anchor node는 참조 노드이고 Sink는 매 iteration 마다 각 노드의 위치 좌표를 수신하여 PC로 그 값을 report하는 역할을 하는 노드이다.

그림 13은 실제 실험 장면이다. 그림에서 보다시피 바닥에 MicaZ모트를 놓고 PC에 연결된 sink mote는 mib510 보드를 통해 신호를 전달하게 된다.

그림 14는 위치 인식 수행에서 첫 번째 iteration과 두 번째 iteration에서의 결과 값을 좌표로 나타낸 것이다. 1번 노드와 6번 노드의 오차가 보정 된 것을 볼 수 있고, 2,5,7,8번 노드들의 오차가 조금 줄어든 것을 볼 수 있다.

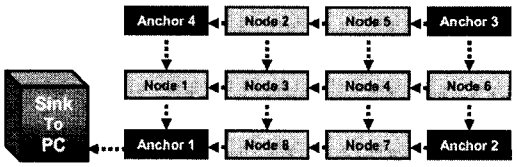


그림 12. Geographical configuration of the demonstration of the proposed algorithm

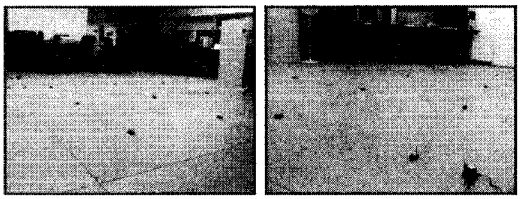
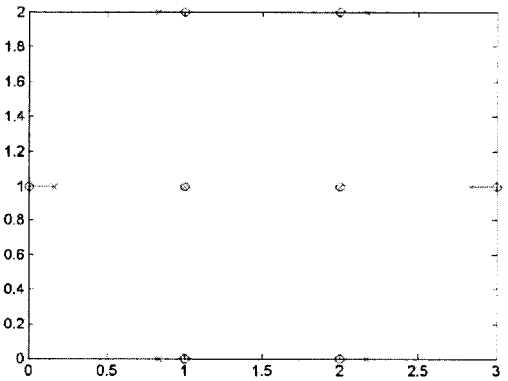
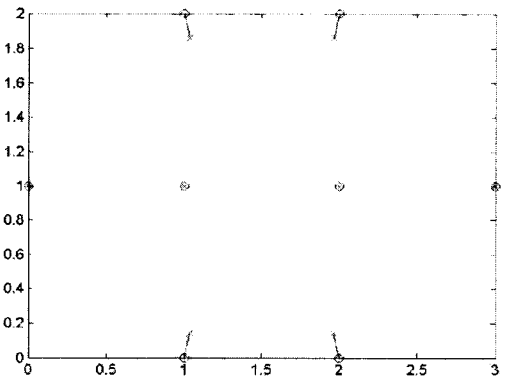


그림 13. The demonstration of the proposed algorithm. This demo uses 13 micaz motes and 1 mib510 board



(a) 1st Cycle



(b) 2nd Cycle

그림 14. Implementation result of the NBDL algorithm

VI. 결 론

본 논문에서는 대규모 무선 센서 네트워크에서 보다 실용적인 위치 인식을 위해 노드의 분포 특성을 이용한 분산 위치 인식 기법을 제안하고 MATLAB simulation과 실제 센서 노드에의 구현을 통한 성능 검증에 대한 결과를 게시하였다. Matlab simulation에서는 dwMDS algorithm과의 비교를 통해 효율성을 검증하였고, TinyOS 구현을 통해 실제 센서 노드에서의 구현 가능성을 보여주었다. 무선 센서 네트워크가 대규모로 구성될 때, 위치 인식은 센서 네트워크의 활용에 있어서 무엇보다 중요하다. 또한 저가의 센서 노드로 네트워크를 구성함에 있어서 위치 인식의 구현이 매우 적은 용량을 차지해야 함은 당연하다. 그러므로 실시간으로 변화하는 무선 센서 네트워크 응용에 있어서 본 논문이 제안하는 알고리즘은 그의 신속성과 간단함에 있어서 효율적인 위치 인식 기법이 될 수 있음을 주장할 수 있다.

참 고 문 헌

[1] Neal Patwari et al, "Relative location estination in wireless sensor networks," *IEEE Trans. Sig. Proc.* 51, pp.2137-2148, August 2003.

[2] Yi Shang et al, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel and Distributed systems,* Vol.15, No.11, pp.961-974, November 2004.

[3] Jose A. Costa et al, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Networks,* Vol.2, No.1, pp.39-64, 2006.

[4] Groenen, "The majorization approach to multidimensional scaling: Some problems and extensions," *DSWO Press.*

한 상 진 (Sangjin Han) 정회원

2005년 2월 연세대학교 전자공학과 학사

2008년 2월 연세대학교 전기전자공학과 석사

2008년 3월~현재 포스데이타 센서네트워크팀 연구원

<관심분야> 무선센서네트워크, 분산위치인식 및 무선자원관리

이 성 진 (Sungjin Lee) 정회원

2005년 2월 숭실대학교 전자공학과 학사

2007년 2월 연세대학교 전기전자공학과 석사

2007년 3월~현재 연세대학교 전기전자공학과 박사과정

<관심분야> 무선통신네트워크, 센서네트워크, 다중홉셀룰러네트워크

이 상 훈 (Sanghoon Lee) 종신회원

1989년 2월 연세대학교 전기전자공학과 학사

1991년 2월 한국과학기술원 전기전자공학과 석사

1991년 3월~1996년 8월 한국통신 연구원

1996년 6월~1999년 1월 미국텍사스오스틴주립대 전기전자공학과 박사

2000년 3월~2002년 12월 미국 루슨트 테크놀로지, 연구원

2003년~현재 연세대학교 전기전자공학과 부교수

<관심분야> 무선네트워크, 센서네트워크, 무선멀티미디어통신

박 종 준 (Jongjun Park) 정회원

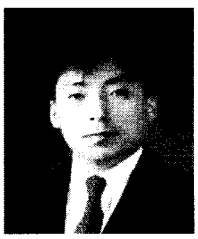
2004년 2월 포항공과대학교 전기전자공학부 학사

2006년 2월 포항공과대학교 전기전자공학부 석사

2006년 3월~현재 전자통신연구소 연구원

<관심분야> 디지털신호처리, 무선센서네트워크, 위치인식 및 미들웨어 이슈

박 상 준 (Sangjoon Park) 정회원



1988년 2월 경북대학교 전기전자공학과 학사

1990년 2월 경북대학교 전기전자공학과 석사

1990년 3월~2001년 국방과학연구소 연구원

2006년 2월 미국 놀스캐롤리나주립대학교 박사

2006년 3월~현재 전자통신연구소, 감시정찰센서네트워크연구팀 팀장

<관심분야> 무선센서네트워크, 차세대 임베디드센서네트워크, 다중센서데이타퓨전 및 타겟 트래킹