

무선 센서 네트워크에서 칼만 필터를 이용한 잔여 힘-벡터 기반 Range-free 위치인식 알고리즘

준회원 이상우*, 정회원 이재우*

Range-free Localization Based on Residual Force-vector with Kalman Filter in Wireless Sensor Networks

Sang-Woo Lee* Associate Member, Chae-Woo Lee* Regular Member

요 약

기존에 연구된 위치인식 기법은 자신의 위치를 알고 있는 앵커노드만을 참조하여 일반노드의 위치를 계산한다. 그 외 노드들 간 관계를 고려하지 않기 때문에 참조하는 앵커노드의 수가 부족하거나 앵커노드로부터의 거리정보가 부정확한 경우에는 실제위치와 예측위치 간 위치오차 뿐만 아니라 노드 간 상대적인 위치오차가 크다. 본 논문에서는 일반노드가 노드 간 거리정보 없이 앵커노드와 한 홉 거리의 이웃노드의 정보만을 참조하여 모든 이웃노드와의 거리가 균등하도록 위치를 계산하는 알고리즘을 제안한다. 노드가 모든 이웃노드와 균등한 거리에 위치함으로써 노드 간 상대적 위치오차와 네트워크 전체의 평균적인 위치오차를 감소시킨다. 다양한 환경에서의 모의 실험을 통해서 제안하는 알고리즘의 성능을 평가하였다.

Key Words : Localization, Range-free, Kalman Filter, Wireless Sensor Network, Ad Hoc Network

ABSTRACT

Many localization schemes estimate the locations of radio nodes based on the physical locations of anchors and the connectivity from the anchors. Since they only consider the knowledge of the anchors without else other nodes, they are likely to have enormous error in location estimate unless the range information from the anchors is accurate or there are sufficiently many anchors. In this paper, we propose a novel localization algorithm with the location knowledge of anchors and even one-hop neighbors to localize unknown nodes in the uniform distance from all the one-hop neighbors without the range information. The node in the uniform distance to its all neighbors reduces the location error relative to the neighbors. It further alleviates the location error between its actual and estimated locations. We evaluate our algorithm through extensive simulations under a variety of node densities and anchor placement methods.

I. 서 론

무선 센서 네트워크(Wireless Sensor Networks)는 주변 환경 관찰 및 데이터 수집이 용이하여 다양한 분야에서 관심을 받고 있다^[1]. 무선 센서 네트워크를 응용한 대표적인 예는 환경 모니터링, 보건 의료, 군사

시설의 감시 및 추적 시스템 등이 있으며, 이와 같은 응용 서비스는 기본적으로 무선노드의 위치정보에 기반을 두고 있다. 무선노드의 위치는 관리자에 의한 수동적 구성 또는 GPS(Global Positioning System)^[2]를 통해서 쉽게 결정할 수 있다. 하지만 수동적 구성에 의한 위치부여 방법은 무선노드의 이동성에 대해 취

* 이주대학교 전자공학과 Multimedia Networking 연구실({sixthcube and cwlee}@ajou.ac.kr)
 논문번호 : KICS2009-11-549, 접수일자 : 2009년 11월 2일, 최종논문접수일자 : 2010년 3월 9일

약하며, 다수의 노드를 관리하기 어렵기 때문에 대규모의 네트워크에 부적합하다. GPS를 사용한 위치측정은 실내에서 그 기능이 제한되며, 비용 문제와 더불어 장치사용에 따른 에너지 문제를 야기한다. 이러한 문제로 인하여 무선 센서 네트워크에서는 무선노드가 직접 노드 간 상대적인 관계 정보를 이용하여 자신의 위치를 계산할 수 있는 위치인식 기법이 요구되고 있다.

자신의 위치를 모르는 일반노드는 수동적 구성 또는 GPS에 의해 실제 물리적 위치를 알고 있는 소수의 앵커노드의 위치정보와 자신과 앵커노드 간 거리 및 관계 정보에 기반을 두어 자신의 위치를 추정한다. 이런 위치인식 기술은 노드 간 거리 또는 방향과 같은 거리정보의 측정 여부에 따라 Range-based 방식과 Range-free 방식으로 구분한다. Range-based 방식은 일반노드가 수신하는 신호의 세기, 지연시간, 입사각의 측정을 통해서 자신과 송신노드 간 거리정보를 얻고 이를 바탕으로 자신의 위치를 추정한다. 그리고 노드 간 거리정보를 얻는 방법에 따라서 RSS(Received Signal Strength)^[3], TOA(Time of Arrival)^[4], TDOA(Time Difference of Arrival)^[5], AOA(Angle of Arrival)^[6]로 분류한다. Range-based 방식은 노드 간 거리정보의 측정을 통해서 일반노드의 위치를 계산하기 때문에 Range-free 방식보다 위치계산의 정확도가 높다. 하지만, 다중경로 페이딩(multipath fading)과 잡음(noise) 등의 요인에 인한 신호의 왜곡에 따라 그 정확도가 달라지기 때문에 Range-based 방식은 주위환경의 변화에 민감하다. 또한, 무선노드가 노드 간 거리정보를 얻기 위해서는 고가의 측정 장비 혹은 노드 간 동기화가 필요하기 때문에 Range-based 방식은 추가적인 비용 문제와 에너지 소비에 따른 네트워크 수명단축 문제를 야기할 수 있다.

이러한 문제의 대안으로 제안된 Range-free 방식은 노드 간 거리정보의 측정 없이 노드들 간 전송경로 혹은 연결 관계와 같은 토폴로지 정보를 이용하여 노드의 위치를 계산한다. 이 방식의 대표적인 예로는 Centroid^[7], APIT(Approximation Point in Triangulation Test)^[8], DV-Hop^[9] 등이 있다. Range-free 방식은 노드 간 거리정보를 사용하지 않고 일반노드의 위치를 계산하기 때문에 Range-based 방식과는 다르게 위치추정에 있어서 주위환경의 변화에 영향이 적으며, 추가적인 비용 문제와 에너지 소비 문제가 없다. 하지만 일반노드가 거리정보 없이 앵커노드만을 참조하여 위치를 계산하기 때문에 위치계산의 정확도가 낮을 뿐만 아니라 노드들의 예측위치가 실제위치와 서로 뒤바뀌는 플립 문제(flip ambiguity)^[10,11]와 노

드의 예측위치가 네트워크 또는 이웃노드로부터 고립되는 문제가 네트워크 전역에 빈번하게 발생한다.

본 논문에서는 노드 간 거리정보를 측정할 수 없고 소수의 앵커노드만 주어진 환경에서 한 홉 거리의 이웃노드 간 평균 거리오차를 감소시킴으로써 네트워크 전체의 평균적인 위치오차를 감소시키는 알고리즘을 제안한다. 이를 위하여 일반노드는 모든 이웃노드로부터의 예측거리 오차 제공의 합을 최소화하는 위치, 즉 모든 이웃노드로부터 예측거리의 평균에 위치하도록 자신의 위치를 계산한다. 제안하는 알고리즘에서 우선 일반노드는 앵커노드의 정보만을 이용하여 자신의 위치를 계산하고, 이후 주기적으로 이웃노드의 위치정보를 참조하여 일반노드와 모든 이웃노드로부터 거리가 평균 거리로 균등하도록 일반노드의 위치를 보정한다. 이때 이웃노드는 일반노드와의 예측거리와 평균 거리 간 오차(residual)에 따라 노드 간 예측거리가 평균 거리에 수렴하도록 일반노드에 힘-벡터를 가하고 일반노드는 모든 이웃노드로부터 받는 총체적인 힘-벡터에 기반을 두어 위치를 보정한다. 반복을 통해서 일반노드와 모든 이웃노드로부터의 거리가 평균 거리에 수렴하고 힘의 균형을 유지하는 위치를 계산한다. 하지만 매번 평균 거리의 변화로 인하여 이웃노드가 일반노드에 미치는 힘-벡터는 방향성이 뒤바뀌고 이에 따라 일반노드의 위치는 진동한다. 이러한 요인으로 인하여 일반노드가 안정한 위치를 계산하기 위해서는 오랜 소요시간이 필요하다. 따라서 일반노드는 칼만 필터(Kalman Filter)^[12]를 사용하여 과거와 현재의 평균 거리로부터 안정된 평균 거리를 추정한다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 위치인식 알고리즘에 대해 살펴보고, III장에서는 네트워크 모델과 본 논문에서 사용되는 표기법에 더불어 성능 평가 기준에 대해 설명한다. IV장에서는 제안하는 알고리즘을 설명하고, V장에서 그 성능을 모의실험을 통해 확인한다. 그리고 VI장에서 본 논문의 결론을 맺는다.

II. 관련 연구

기준에 연구된 위치인식 기술은 노드의 위치계산을 위한 거리정보 측정 여부에 따라 Range-based와 Range-free 방식으로 구분된다. 본 장에서는 두 방식에 대해 간략하게 설명한다.

2.1 Range-based

Range-based 방식은 무선노드가 앵커노드로부터

수신하는 신호를 측정하여 앵커노드와의 거리를 계산하고 그로부터의 상대적인 위치를 계산한다. 거리를 측정하는 방법에 따라서 RSS, TOA, TDOA와 AOA로 분류된다. RSS 방식은 거리에 따라 감쇠하는 신호의 세기를 측정하여 앵커노드와 일반노드 간 거리를 계산한다. RADAR^[13]와 SpotON^[14]은 RSS 방식을 사용한다. TOA는 이미 널리 사용되는 GPS의 방식으로 앵커노드로부터 일반노드까지 신호의 전달 지연 시간을 측정하여 두 노드 간 거리를 계산한다. 하지만 신호의 지연시간을 이용하여 노드 간 거리를 계산하기 위해서는 앵커노드와 일반노드의 동기화가 추가적으로 필요하다. TOA와 유사한 TDOA은 앵커노드에서 서로 다른 두 신호를 동시에 전송하고 일반노드에 도달하는 두 신호의 시간차를 이용하여 노드 간 거리를 계산하는 방식으로 예로는 Cricket^[15]이 있다. AOA 방식은 수신신호의 입사각을 통해서 앵커노드와의 상대적인 각을 측정한다. 그러나 신호의 입사각을 측정하기 위해서는 지향성 안테나(directional antenna)^[15]와 디지털 나침반(digital compass)^[16] 같은 고가의 장치가 필요하다. Range-based 방식은 측정된 거리정보를 사용하여 일반노드의 위치를 계산하기 때문에 Range-free 방식에 비해서 위치계산의 정확도가 높지만, 거리정보 측정을 위해 별도의 장비 또는 노드 간 동기화가 요구되기 때문에 비용 문제와 에너지 문제를 야기한다. 또한 무선채널 환경에서 신호는 다중경로 페이딩과 잡음 등의 요인에 민감하기 때문에 주위 환경에 따라 위치계산의 정확도가 불안정하다. 그렇기 때문에 최근에는 잡음 등의 요소를 고려하여 거리정보를 측정하는 기법^[17]과 노드 간 측정거리와 예측거리 정보 사이에 오차를 최소화하는 위치인식 알고리즘^[10,18]이 연구되고 있다.

2.2 Range-free

Range-free 방식은 Range-based 방식의 비용 문제와 에너지 소비 문제를 해결하고자 제안되었으며, 거리정보의 측정 없이 노드 간 연결 정보를 이용하여 노드의 위치를 계산한다. Range-free 방식의 대표적인 예로는 Centroid, APIT, DV-Hop 등이 있다. Centroid^[7]에서 일반노드는 자신과 이웃한 앵커노드의 무게중심을 자신의 위치로 계산한다. 그렇기 때문에 알고리즘은 간단하지만 대규모 네트워크에서 일반노드의 위치를 계산하기 위해서는 다수의 앵커노드를 요구한다. APIT^[8]는 서로 다른 조합의 앵커노드들이 이루는 삼각형 영역 중에서 가장 많이 중첩되는 영역의 무게중심을 일반노드의 위치로 계산한다. 네트워크

를 일정 크기의 격자(grid)로 나누어 영역을 구분하기 때문에 그 격자 크기에 따라 위치계산 정확도가 달라지며 Centroid 알고리즘과 마찬가지로 일반노드의 위치 계산을 위해 다수의 앵커노드를 요구한다. DV-Hop^[9]은 앵커노드의 위치정보와 홉 수 정보를 사용하여 앵커노드와 일반노드 간 거리를 계산하고 일반노드의 위치를 추정한다. 앵커노드는 다른 앵커노드의 위치와 그로부터의 최소 홉 수를 이용하여 한 홉의 평균 거리를 산출한다. 그리고 일반노드는 각 앵커노드의 홉 당 평균 거리와 그로부터의 홉 수 정보를 통해서 앵커노드와의 평균 거리를 계산한다. 이후 일반노드는 앵커노드로부터의 위치와 거리정보를 이용하여 자신의 위치를 추정한다. 노드 간 거리정보를 산출하여 일반노드의 위치계산에 사용하기 때문에 다른 Range-free 방식과는 다르게 적은 수의 앵커노드만으로도 일반노드의 위치를 계산할 수 있다. 그러나 다른 Range-free 알고리즘과 마찬가지로 정확한 위치계산을 위해서는 많은 수의 앵커노드와 균일한 노드 배치가 요구된다. 이렇게 Range-free 방식은 Range-based 방식의 제약으로부터 자유롭지만, 노드의 정확한 위치계산을 위해서는 균일한 노드 배치 또는 다수의 앵커노드를 필요로 하기 때문에 토폴로지의 정보를 충분히 사용하여 적은 수의 앵커노드만으로도 위치계산 정확도를 향상시킬 수 있는 기법이 요구되고 있다^[19].

III. 네트워크 모델과 성능 평가 기준

본 장에서는 논문에서 사용되는 용어와 알고리즘의 성능을 평가하는 기준 척도에 대해 정의한다.

3.1 네트워크 모델

$D(2$ 또는 $3)$ 차원 공간에 네트워크는 총 $S = M + N$ 개의 무선노드들을 통해서 구성된다. 각 노드는 별개의 아이디 i 을 갖는다. M 개의 노드 ($i = 1 \dots M$)는 관리자 또는 GPS를 통해서 자신의 정확한 위치 정보를 포함한 앵커노드이며, 그 외 N 개의 노드($i = M + 1 \dots S$)는 자신의 위치를 모르는 일반노드다. 여기서 모든 노드는 이동하지 않고 네트워크에 새롭게 참여하거나 네트워크로부터 이탈하는 노드가 없다. 또한, 모든 노드의 전송반경 T 은 동일하다. 무선노드의 실제위치와 예측위치는 각각 $\{L_i\}_{i=1}^S$ 와 $\{\hat{L}_i\}_{i=1}^S$ 이다. 단, $\{\hat{L}_i = L_i\}_{i=1}^M$ 로 앵커노드의 실제위치와 예측위치는 동일하다. 그리고 두 노드 i 와 j 의 실제거리와 예측위치에 의한 예측거리는

$d_{ij} = \|\mathbf{L}_j - \mathbf{L}_i\|$ 와 $\delta_{ij} = \|\hat{\mathbf{L}}_j - \hat{\mathbf{L}}_i\|$ 이다. 여기서, $\|\cdot\|$ 는 Euclidean norm이다. 이 때 일반노드 i 는 이웃노드와의 예측거리를 이용하여 모든 이웃노드와 평균 거리 x_i 을 만족하는 위치를 계산한다. 여기서 일반노드와 이웃노드 간 평균 거리는 이웃노드들 간 예측거리의 산술평균으로 계산되며 이는 일반노드와 모든 이웃노드 간 예측거리의 오차 제곱의 합을 최소화하는 거리다.

3.2 성능 평가 기준

본 논문에서 제안하는 알고리즘의 성능을 평가하기 위한 성능 평가 기준은 다음과 같다. 실제위치와 예측위치 간 오차에 따른 모든 노드들의 평균 위치오차 (average location error: ALE)와 노드 간 거리의 수렴 정도를 보이는 전체 링크의 분산(global variance of link: GVL)를 사용한다. 평균 위치오차는 앵커노드에 기반을 두어 일반노드의 실제위치를 추정하는 연구에서 위치인식 기법의 성능을 평가하는 지표로 사용되고 있다. 위치오차는 실제위치와 예측위치 간 오차를 의미하고 전체 일반노드의 위치오차에 대한 평균 위치오차는 다음과 같다.

$$ALE = \frac{1}{N} \sum_{i=M+1}^S \|\mathbf{L}_i - \hat{\mathbf{L}}_i\| \quad (1)$$

제안하는 알고리즘에서 일반노드는 모든 이웃노드와의 거리가 균등하게 노드 간 평균 거리에 수렴하도록 자신의 위치를 계산한다. 따라서 노드 간 거리의 균등 정도를 평가하기 위하여 전체 링크의 분산을 제시한다. 전체 링크의 분산은 모든 이웃노드 간 링크 길이, 즉 예측거리의 분산으로 네트워크 전체의 노드 간 링크의 평균 거리와 예측거리 사이의 오차 제곱의 평균으로 다음과 같이 산출한다.

$$GVL = \frac{\sum_{\forall i} \sum_{\forall j \neq i, d_{ij} \leq T} (E - \delta_{ij})^2}{\sum_{\forall k} n_k} \quad (2)$$

여기서 n_k 는 일반노드 k 와 한 홉 거리에 위치한 이웃노드의 수이고, E 는 전체 링크의 평균 거리로 다음과 같이 계산한다.

$$E = \frac{\sum_{\forall i} \sum_{\forall j \neq i, d_{ij} \leq T} \delta_{ij}}{\sum_{\forall k} n_k} \quad (3)$$

IV. 제안하는 알고리즘

제안하는 알고리즘은 크게 2 단계로 구성 동작한다. 초기 단계에서 일반노드는 앵커노드의 위치정보와 그로부터의 연결 정보에 기반을 두어 위치를 계산한다. 이후 단계에서는 반복적인 알고리즘을 통해서 일반노드의 위치를 계산한다. 앵커노드의 정보를 이용한 일반노드의 위치계산은 기존의 Range-free 방식과 같은 방법으로 이루어진다. 그리고 일반노드는 이웃노드와 위치정보를 주기적으로 교환함으로써 모든 이웃노드와의 거리가 평균 거리로 균등하도록 위치를 계산한다.

본 장에서는 일반노드가 이웃노드와의 평균적인 거리오차를 감소시키기 위하여 일반노드가 모든 이웃노드로부터 평균 거리에 위치하도록 위치를 계산하는 과정에 대하여 살펴본다. 일반노드는 이웃노드와의 예측거리가 평균 거리에 수렴하도록 두 거리 간 차이에 비례하는 힘을 받고 모든 이웃노드로부터 받는 힘의 합에 따라서 자신의 위치를 보정한다. 그러나 평균 거리는 매번 일반노드의 위치 변화에 따라 변하기 때문에 힘에 따른 일반노드의 위치는 진동할 수 있다. 이러한 문제를 해결하기 위해서 일반노드는 칼만 필터를 통해서 과거와 현재의 평균 거리로부터 위치변화에 안정된 평균 거리를 추정한다. 이와 같은 과정을 반복하여 일반노드는 모든 이웃노드로부터 칼만 필터를 통해 추정한 평균 거리에 위치하고 힘의 평형을 이루는 지점에 위치한다.

4.1 노드 간 평균 거리 추정

일반노드는 주기적으로 이웃노드의 위치정보를 활용하여 자신과 이웃노드 간 평균 거리를 계산한다. 그러나 매번 일반노드의 위치변화에 따라서 그 평균 거리 또한 따라 변한다. 이와 같은 불안정한 평균 거리를 이용하여 일반노드의 위치를 계산할 경우에는 모든 노드가 이웃노드로부터 평균 거리를 만족하는 위치를 계산하는데 필요한 소요시간이 증가할 수도 있으며, 평균 거리가 너무 작거나 커서 일반노드 자신뿐만 아니라 그 이웃노드의 위치계산을 방해할 수 있다. 이러한 요인을 완화시키기 위하여 일반노드는 칼만 필터를 사용하여 위치변화에 변동이 적은 안정된 평

균 거리를 추정한다.

시간 k 에서 일반노드 i 와 이웃노드 간 평균 거리를 $x_i^{(k)}$ 라 할 때, 평균 거리 추정을 위한 칼만 필터의 프로세스 모델(process model)과 관측 모델(measurement model)은 각각

$$x_i^{(k)} = Ax_i^{(k-1)} + Bu_i^{(k-1)} + w_i^{(k)} \quad (4)$$

$$z_i^{(k)} = Hx_i^{(k)} + v_i^{(k)} \quad (5)$$

이다. 여기서 평균 거리는 시간에 변함없이 일정하고 시간에 따른 입력이 없으므로 $A = 1$, 입력 $u_i^{(k)} = 0$ 이다. 측정(measurement) $z_i^{(k)}$ 은 시간 k 에서 일반노드 i 와 그 이웃노드 간 예측거리의 평균으로 프로세스 모델의 평균 거리와 상태가 동일하기 때문에 $H = 1$ 이다. 그리고 $w_i^{(k)}$ 와 $v_i^{(k)}$ 은 랜덤 변수로 각각 프로세스 노이즈(process noise)와 관측 노이즈(measurement noise)를 의미하며 서로 독립이고 다음의 정규분포를 따른다.

$$P(w_i^{(k)}) \sim N(0, Q_i^{(k)}) \quad (6)$$

$$P(v_i^{(k)}) \sim N(0, R_i^{(k)}) \quad (7)$$

여기서 $Q_i^{(k)}$ 와 $R_i^{(k)}$ 은 시간 k 에서 각 노이즈의 공분산(covariance)으로 각 모델에서 평균 거리의 잠재적인 변화량 또는 불확실성을 나타낸다. 프로세스 모델을 통해서 일반노드 i 와 이웃노드 간 평균 거리를 추정하는데 있어서 입력에 따른 상태 변화가 없으므로 $Q_i^{(k)}$ 은 필터의 조율을 위해 0에 근사한 상수를 갖는다. 반면, $R_i^{(k)}$ 은 일반노드 i 가 측정한 평균 거리와 이웃노드와의 예측거리를 통해 다음으로 정의한다.

$$R_i^{(k)} = \frac{1}{n_i} \sum_{\forall j \neq i, d_{ij} \leq T} (x_i^{(k)} - \delta_{ij}^{(k)})^2 \quad (8)$$

여기서 n_i 는 일반노드 i 와 한 홉 거리에 위치한 이웃노드의 수다.

칼만 필터는 앞의 두 모델에 기반을 두어 예측(predict)과 보정(correct) 단계로 나뉘어 동작한다. 일반노드는 예측 단계에서 프로세스 모델을 통해 과거의 평균 거리로부터 현재의 평균 거리를 예측하고, 보정 단계에서 평균 거리의 측정을 통해서 앞서 예측한 평균 거리를 보정함으로써 예측한 평균 거리보다 정확하고 안정한 평균 거리를 추정할 수 있다. 시간 k 에서 일반노드 i 와 그 이웃노드 간 평균 거리의 추정은 $\hat{x}_i^{(k)}$ 로 표기하고, 시간 k 에서의 예측 단계는 시간 k^- 로 표기한다. 프로세스 모델을 통한 시간 k 에서 평균 거리의 예측은 다음과 같다.

$$\hat{x}_i^{(k^-)} = \hat{x}_i^{(k-1)} \quad (9)$$

$$P_i^{(k^-)} = P_i^{(k-1)} + Q_i^{(k)} \quad (10)$$

여기서 $P_i^{(k)}$ 은 일반노드 i 와 이웃노드 간 평균 거리의 추정 오류 공분산(estimate error covariance)으로 시간 k 에서 예측한 평균 거리의 잠재적인 변화량 또는 불확실성을 나타낸다. 그리고 평균 거리의 예측과 측정을 통한 시간 k 에서의 평균 거리 보정은 다음으로 계산된다.

$$K_i^{(k)} = P_i^{(k^-)} (P_i^{(k^-)} + R_i^{(k)})^{-1} \quad (11)$$

$$\hat{x}_i^{(k)} = \hat{x}_i^{(k^-)} + K_i^{(k)} (z_i^{(k)} - \hat{x}_i^{(k^-)}) \quad (12)$$

$$P_i^{(k)} = (1 - K_i^{(k)}) P_i^{(k^-)} \quad (13)$$

여기서 $K_i^{(k)}$ 은 칼만 이득(Kalman gain)으로 예측과 측정한 평균 거리의 불확실성에 따라 예측과 측정 평균 거리의 가중치를 달리한다. 결과적으로 일반노드 i 는 수식 (12)에서 측정한 평균 거리를 자신의 위치계산에 사용한다.

이와 같이 일반노드는 칼만 필터를 이용하여 과거와 현재의 평균 거리의 잠재적인 변화량을 고려하여 위치변화에 보다 안정된 평균 거리를 추정할 수 있다. 초기에는 평균 거리의 추정 오류 공분산이 크기 때문

에 평균 거리는 결과적으로 측정치에 따라 결정된다. 그러나 평균 거리의 추정 오류 공분산은 수식 (10), (11)과 (13)를 통해 시간에 따라 일정한 값에 안정적으로 수렴해나간다. 시간이 증가할수록 추정 평균 거리의 잠재적인 변화량이 감소하기 때문에 평균 거리의 예측치는 그 측정치에 비해 큰 가중치를 갖는다. 따라서 일반노드와 이웃노드 간 평균 거리는 시간에 따라 평균 거리의 변화량이 감소한다.

4.2 일반노드의 위치 보정

칼만 필터를 이용하여 이웃노드와의 평균 거리를 추정된 일반노드는 이후 힘 벡터의 방법에 기반을 두어 평균 거리와 이웃노드와의 예측거리 사이의 오차를 제거하여 모든 이웃노드로부터 평균 거리에 위치하고 힘의 균형을 유지하는 위치를 계산한다. 그림 1은 이웃노드로부터 받는 힘에 따른 일반노드의 위치 이동을 보인다. 일반노드 U 를 중심으로 한 원은 일반노드의 평균 거리를 보인다. 이때 이웃노드 $N1$ 은 일반노드의 평균 거리보다 가까이 위치하기 때문에 일반노드는 이웃노드 $N1$ 으로부터 멀어지는 방향의 힘 F_{N1} 을 받는다. 그리고 일반노드와 이웃노드 $N2$ 간 거리는 그 평균 거리보다 크기 때문에 일반노드는 이웃노드 $N2$ 방향의 힘 F_{N2} 을 받는다. 따라서 일반노드는 모든 이웃노드로부터 받는 힘의 합 F 에 의해 이동한다.

시간 k 에서 무선노드 i 의 예측위치 $\hat{L}_i^{(k)}$ 이고, 그 이웃노드 j 간 예측거리는 다음과 같다.

$$\delta_{ij}^{(k)} = \left\| \hat{L}_j^{(k)} - \hat{L}_i^{(k)} \right\| \tag{14}$$

이때 일반노드 i 가 이웃노드 j 에 의해 받는 힘

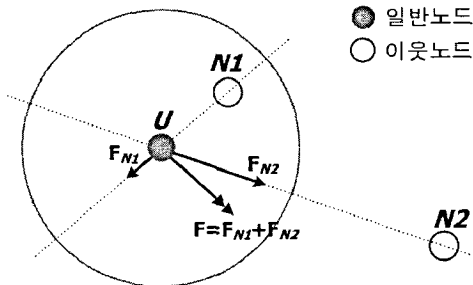


그림 1. 이웃노드의 힘에 의한 일반노드의 위치 이동

$F_{ij}^{(k)}$ 은 평균 거리 $\hat{x}_i^{(k)}$ 와 두 노드 간 예측거리 $\delta_{ij}^{(k)}$ 의 차이, 즉 추정치와 측정치 간 오차(residual)에 따라 다음으로 정의한다.

$$F_{ij}^{(k)} = \frac{(\delta_{ij}^{(k)} - \hat{x}_i^{(k)})}{\eta_j} \mathbf{u}_{ij}^{(k)} \tag{15}$$

여기서 $\mathbf{u}_{ij}^{(k)}$ 은 시간 k 에서 두 노드 간 단위 벡터로 다음과 같다.

$$\mathbf{u}_{ij}^{(k)} = \frac{\hat{L}_j^{(k)} - \hat{L}_i^{(k)}}{\delta_{ij}^{(k)}} \tag{16}$$

그리고 수식 (15)에서 η_j 는 이웃노드 j 에 따라 힘을 나타내는 계수로 이웃노드가 앵커노드인 경우에 크기가 1인 값을 갖고, 반면에 일반노드의 경우에 크기가 2인 값을 갖는다. 앵커노드는 마치 벽과 같이 위치가 고정적이기 때문에 이웃노드로부터 받는 힘은 없다. 따라서 일반노드와 앵커노드로 구성되는 계(system), 즉 두 노드 간 링크에 미치는 힘은 일반노드가 받는 힘과 동일하다. 하지만 두 개의 일반노드가 서로 이웃하는 경우에는 각 노드가 서로의 이웃노드로부터 힘을 받는다. 그렇기 때문에 결과적으로 하나의 계에 대하여 두 힘이 작용하여 일반노드뿐만 아니라 이 계는 기대이상의 힘이 발생한다.

일반노드 i 가 이웃노드 j 에 의한 힘 $F_{ij}^{(k)}$ 에 따라 모든 이웃노드로부터 받는 총체적인 힘 $F_i^{(k)}$ 은 다음과 같다.

$$F_i^{(k)} = \sum_{\forall j \neq i, d_{ij} \leq r} F_{ij}^{(k)} \tag{17}$$

일반노드는 모든 이웃노드로부터 받는 총체적인 힘에 의해 현재의 위치에서 그 힘의 방향으로 이동한다. 하지만 그 힘에 따라 일반노드가 이동하는 크기가 너무 크면 시간에 따라 일반노드는 진동할 수 있기 때문에 하나의 안정된 위치를 계산하기 어렵다. 반면, 일반노드의 위치 이동량이 너무 작은 경우에는 전체적으로 노드 간 거리가 균등한 위치를 계산하는데 필요한 소요시간이 길어질 수 있다. 그렇기 때문에 일반노

드는 힘에 따라 적절히 이동해야 한다. 따라서 모든 이웃노드에 의한 힘에 따른 일반노드 i 의 위치 이동 $\Delta_i^{(k)}$ 은 다음과 같다.

$$\Delta_i^{(k)} = \alpha \mathbf{F}_i^{(k)} \quad (18)$$

여기서 α 는 총체적인 힘에 따른 일반노드의 위치 이동을 나타내는 계수로 0과 1 사이에서 너무 크거나 작지 않은 크기의 값을 갖는다. 이로써 일반노드 i 는 다음과 같이 현재의 위치에서 $\Delta_i^{(k)}$ 만큼 이동한 지점을 자신의 위치로 계산한다.

$$\hat{\mathbf{L}}_i^{(k+1)} = \hat{\mathbf{L}}_i^{(k)} + \Delta_i^{(k)} \quad (19)$$

일반노드는 각자의 평균 거리에 기반을 두어 위치를 계산하기 때문에 모든 노드 간 거리가 균등해질 수 없지만, 반복적인 과정을 통해서 일반노드는 이웃노드로부터 힘의 평형을 이루는 위치를 계산할 수 있다. 이를 통해서 네트워크 부분적으로 노드 간 거리가 균등해지는 네트워크 토폴로지를 추정할 수 있다.

4.3 예제

본 절에서는 4개의 예를 통해서 제안하는 알고리즘을 이용한 위치계산의 정확도를 살펴본다. Range-free 방식은 노드 간 연결 관계와 앵커노드의 위치정보에 기반을 두어 일반노드의 위치를 계산하기 때문에 토폴로지의 형태, 즉 노드의 배치에 따라 다른 성능을 보인다. 이를 알아보기 위해서 토폴로지를 균일 배치와 불규칙 배치로 구분하고, 각 경우에 대해서 정규 배치와 랜덤 배치로 나누어 그림 2와 같이 총 4개의 예를 보인다.

정규 균일 배치(regular uniform)에서의 모든 노드

는 격자 형태로 배치되어 그 이웃노드들과의 거리와 노드의 연결 정도(node degree)가 전체적으로 균일하다. 랜덤 균일 배치(random uniform)의 노드는 국소적으로 이웃노드들과 서로 불규칙한 거리에 위치하지만 네트워크 전체에 균일하게 위치한다. 정규 C형 배치(regular C-shaped)의 노드는 이웃노드들과는 균일한 배치를 이루지만 알파벳 C형태의 배치로 대다수의 노드 간 전송경로는 우회한다. 랜덤 C형 배치(random C-shaped)에서 노드는 C형태의 배치를 따르고 불규칙하게 위치한다.

모든 경우에서 네트워크 크기는 $10 \times 10m^2$, 그리고 노드의 최대전송반경 T 는 $1.5m$ 로 동일하다. 노드 배치 방법에 따라 노드의 밀도와 노드의 연결 정도를 고려하여 서로 다른 수의 노드를 배치한다. 정규 균일 배치에서는 총 81개의 노드를 배치하고 그 중 임의의 5개 노드를 앵커노드로 지정하여 일반노드의 위치계산을 돕는다. 정규 C형 배치에서는 총 63개의 노드를 배치하고 그 중 7개를 앵커노드로 지정한다. 반면, 랜덤 균일 배치와 랜덤 C형 배치에서는 각각 총 200개의 노드, 8개의 앵커노드 그리고 총 150개의 노드, 14개의 앵커노드를 배치한다.

제안하는 알고리즘에서 일반노드의 초기 위치는 Range-free 방식 중에서 적은 수의 앵커노드만으로도 일반노드의 위치를 계산할 수 있는 DV-Hop 알고리즘을 통해서 계산한다. 힘에 따른 일반노드의 위치 이동 계수 α 의 크기는 이웃노드의 평균 개수(node degree)의 역수를 갖는다. 그리고 100번의 위치보정을 통해서 초기의 위치오차와 보정 이후의 위치오차를 살펴본다. 그림 3과 4는 각각 정규 배치와 랜덤 배치에 따른 제안하는 알고리즘을 이용한 일반노드의 위치오차를 보여준다. 그림에서 원과 점은 일반노드와 앵커노드의 실제위치를 나타내고, 선은 일반노드의 실제위치와 예측위치 간 위치오차를 보인다. 선의 길이가 짧을수록 일반노드가 예측위치가 실제위치에 근사

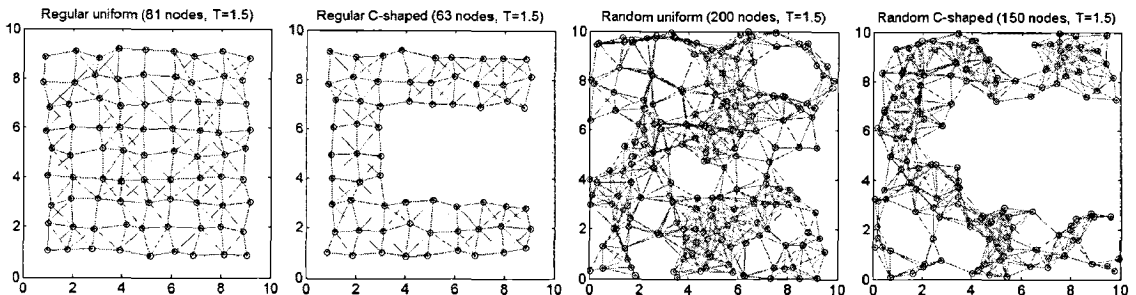


그림 2. 균일랜덤 배치에 따른 네트워크의 물리적 토폴로지

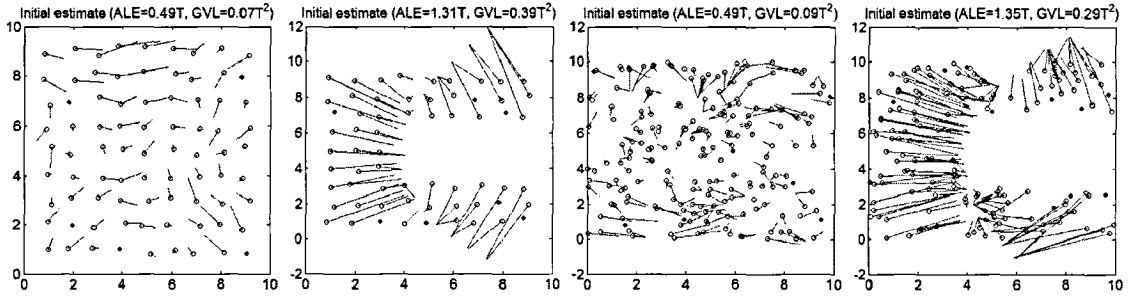


그림 3. 제안하는 알고리즘을 이용한 초기위치의 위치오차

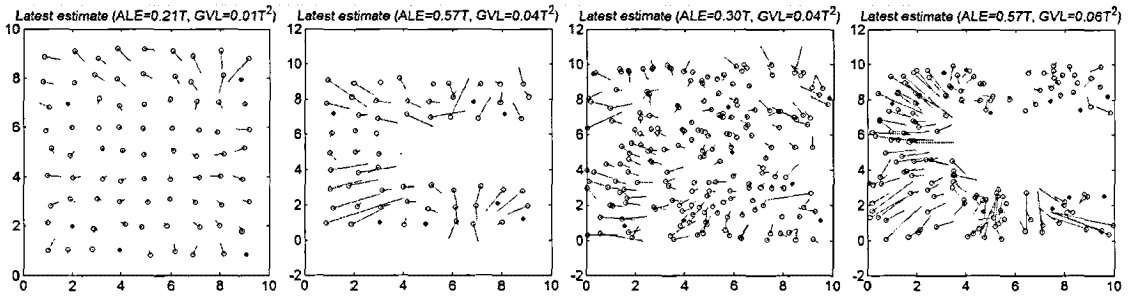


그림 4. 제안하는 알고리즘을 이용한 최종위치의 위치오차

함을 의미하고, 반면 길이가 길수록 두 위치 사이에 큰 차이가 있음을 의미한다.

정규 균일 배치에서는 네트워크 내 홀(hole)의 크기가 작고 그 수가 적어 다른 배치보다 정확한 위치계산을 보인다. 네트워크의 초기 평균 위치오차는 $0.49T$ 이고, 일반노드가 이웃노드와 평균 거리를 유지하기 위한 위치보정 이후의 평균 위치오차는 $0.21T$ 로 감소하였다. 그에 따른 네트워크 전체 링크의 분산은 초기 $0.07T^2$ 에서 위치보정을 통해서 $0.01T^2$ 로 감소하였다. 반면 동일한 정규 배치지만 네트워크 내에 큰 홀을 가진 정규 C형 배치의 경우는 대다수의 노드 간 전송경로는 우회한다. 이때 초기 위치에 따른 일반노드의 평균 위치오차는 $1.31T$ 로 위치계산의 정확도가 매우 낮았지만 위치보정을 통해서 $0.57T$ 까지 그 위치오차가 감소하였다. 그리고 초기 $0.39T^2$ 의 전체 링크의 분산은 위치보정을 통해서 $0.04T^2$ 로 감소함을 보였다.

랜덤 균일 배치의 경우는 네트워크에 곳곳에 다수의 소형 홀이 존재하기 때문에 일부의 노드들은 그 외의 노드들에 비해서 상대적으로 큰 위치 오차를 갖는다. 초기의 위치와 위치보정에 따른 평균위치오차는 각각 $0.49T$ 와 $0.30T$ 이다. 그리고 초기에 $0.09T^2$ 의 값을 보이던 이웃노드 간 링크의 분산은 위치보정을

통해서 $0.04T^2$ 까지 감소함을 보였다. 큰 홀과 더불어 다수의 소형 홀이 존재하는 랜덤 C형 배치에서는 일반노드의 위치계산 정확도가 매우 낮다. 초기에는 $1.35T$ 의 평균 위치오차와 $0.29T^2$ 의 링크 분산으로 위치계산의 정확도가 매우 낮고 노드 간 링크가 불규칙하였지만, 위치보정을 통해서 평균위치오차와 전체 링크의 분산은 각각 $0.57T$, $0.06T^2$ 까지 감소하였다.

본 장에서는 4개의 예제를 통해서 제안하는 알고리즘의 성능을 간략히 살펴보았다. 제안하는 알고리즘의 반복적인 위치보정을 통해서 일반노드의 위치계산의 정확도가 향상되었다. 또한 초기에 일부 인접 노드들은 서로의 위치를 구별하지 못하고 모두 동일한 지점에 위치하였지만, 위치보정을 통해서 이러한 문제를 해결하였다. 다음 장에서는 다양한 환경에서의 모의실험을 통해 보다 정확한 성능 분석을 보인다.

V. 성능 분석

본 장에서는 다양한 환경에서의 모의실험을 통해 제안하는 알고리즘의 성능을 분석한다. 제안하는 알고리즘은 앵커노드를 통해 일반노드의 초기위치를 계산하는 단계와 일반노드가 이웃노드의 위치정보를 참조하여 위치를 반복적으로 보정하는 단계로 나뉜다. 따라서 제안하는 알고리즘을 통해서 매 단계별 평균 위

치오차와 노드 간 링크의 분산을 살펴본다. 제안하는 알고리즘에서 일반노드의 초기위치는 예제와 동일한 방법으로 DV-Hop을 이용하여 계산하고, 힘에 따른 일반노드의 이동 계수 α 의 크기는 이웃노드의 평균 개수의 역수를 갖는다.

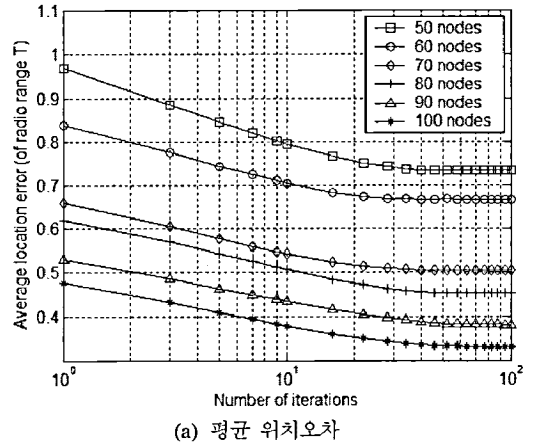
노드의 위치계산 정확도는 전체 노드의 밀도, 앵커노드의 밀도, 그리고 노드의 배치 방법에 영향을 받는다. 각 요인이 위치계산의 정확도에 미치는 영향을 살펴보기 위해 다양한 환경에서 모의실험 수행하였다. 모의실험을 위한 네트워크의 환경은 표 1과 같다. 네트워크의 크기가 일정한 2차원 공간에서 일반노드, 앵커노드의 밀도와 앵커노드의 배치 방법을 달리하여 제안하는 알고리즘의 성능을 살펴본다. 모든 모의실험은 동일한 조건에서 총 100번 반복했다. 제안하는 알고리즘의 성능은 평균 위치오차와 전체 링크의 분산을 각각 노드의 전송반경과 전송반경의 제곱으로 정규화한 수치다.

표 1. 모의실험 파라미터

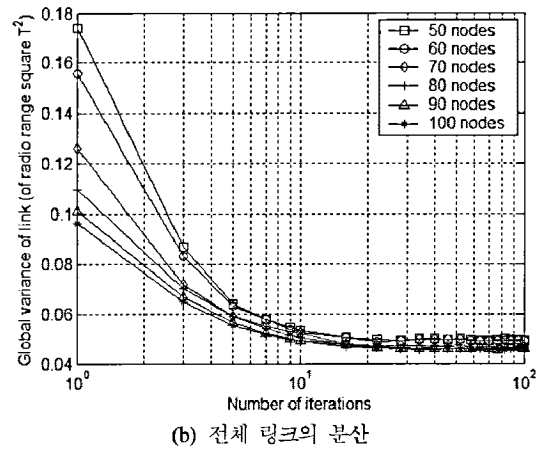
| 항목 | 값 |
|---------|-----------|
| 네트워크 크기 | 20m × 20m |
| 전송반경 | 4m |
| 노드의 수 | 50~100 |
| 앵커노드 비율 | 5~20% |
| 앵커노드 배치 | 랜덤 / 균일 |

4.1 노드의 밀도

본 절에서는 전체 노드의 밀도에 따른 제안하는 알고리즘의 평균 위치오차를 살펴본다. 표1과 같이 일정한 크기의 네트워크에서 앵커노드는 네트워크에 균일하게 위치하며 그 비율은 전체 일반노드의 10%로 일정하다. 이때 일반노드의 수를 50, 60, 70, 80, 90과 100으로 달리하고 랜덤하게 배치한다. 그림 5는 전체 노드의 밀도에 따른 제안하는 알고리즘의 성능을 보인다. 그림의 가로 축은 위치계산의 횟수를 나타내며 명확한 관찰을 위하여 log scale로 표시한다. 전체적으로 일반노드가 이웃노드의 위치정보를 사용하여 위치를 보정함으로써 평균 위치오차가 감소함을 확인할 수 있다. 평균 위치오차는 초기에 급속하게 감소하였지만, 일정 시간 이후에 평균위치오차는 일정한 값으로 수렴했다. 이는 전체 링크의 분산을 통해서 알 수 있듯이 이웃노드 간 거리가 균등해지고 일반노드가 힘의 균형을 이루는 위치에 있음을 의미한다. 위치보정을 통해서 평균 위치오차는 대체로 초기의 평균 위



(a) 평균 위치오차



(b) 전체 링크의 분산

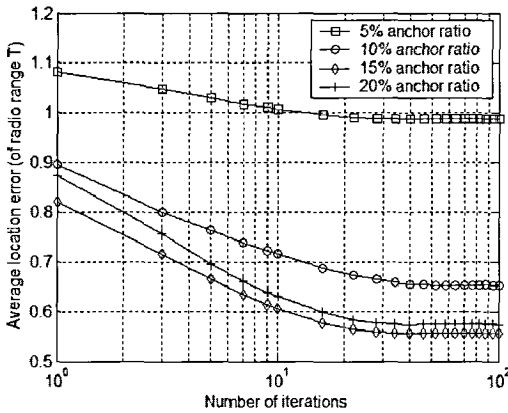
그림 5. 앵커노드의 비율이 10%인 경우에 전체 노드의 밀도에 따른 제안하는 알고리즘의 성능 평가

치오차보다 0.2 T 정도 감소하였다. 전체 노드의 밀도가 높을수록 평균 위치오차는 더 낮았는데, 그 이유는 네트워크 내 홀이 감소하였기 때문이다. 전체 링크의 분산은 초기에 노드의 밀도가 높을수록 그 분산의 크기는 낮았지만, 반복된 위치계산을 통해서 밀도에 관계없이 비슷한 분산 값을 가졌다. 따라서 제안하는 알고리즘을 통해서 모든 노드가 네트워크 전역에 균일하게 위치하였음을 가늠할 수 있고, 이로써 일반노드의 평균적인 위치오차가 감소하였음을 알 수 있다.

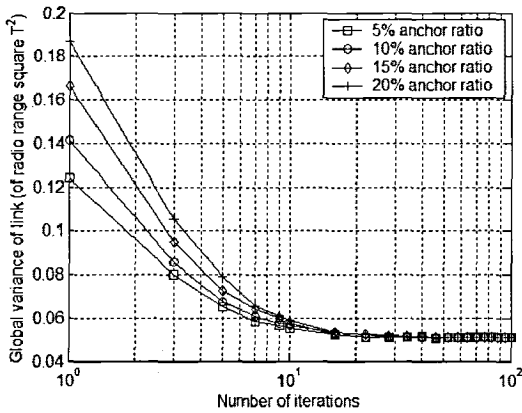
4.2 앵커노드의 비율

앞에서도 언급하였듯이 Range-free 방식은 앵커노드의 정보에만 기반을 두어 일반노드의 위치를 계산하기 때문에 앵커노드의 비율은 Range-free 방식의 성능에 가장 큰 영향을 준다. 이 모의실험은 앵커노드의 비율에 따른 제안하는 알고리즘의 성능을 확인한다. 일정한 공간에 60개의 일반노드가 랜덤하게 위치

할 때, 앵커노드의 비율을 일반노드의 5~20%로 달리 하고 앵커노드를 네트워크에 균일하게 배치한다. 그림 6은 앵커노드의 비율에 따른 제안하는 알고리즘의 성능을 보인다. 앵커노드의 비율이 높을수록 일반노드의 위치계산 정확도가 높았고, 위치보정에 따른 평균 위치오차의 감소율이 증가하였다. 앵커노드의 비율이 5%인 경우에는 약 0.1T의 감소를 보인 반면에 앵커노드의 비율이 전체 노드의 10~20%인 경우에는 평균적으로 0.2T 이상의 감소를 보였다. 제안하는 알고리즘은 앵커노드의 비율에 따라 평균 위치오차가 변화하였지만, 전체 링크의 분산은 앵커노드의 비율에 관계없이 일정한 값으로 수렴하였다. 앵커노드의 비율이 높을수록 오히려 초기의 노드 간 링크의 분산은 높았지만 이후 위치보정 과정에서 링크가 수렴하는 속도가 빨랐다. 결과적으로 전체 링크의 분산은 노드의 밀도에 관계없이 일정한 값으로 수렴하나 수렴하는 속도는 앵커노드의 비율에 비례함을 보였다.



(a) 평균 위치오차

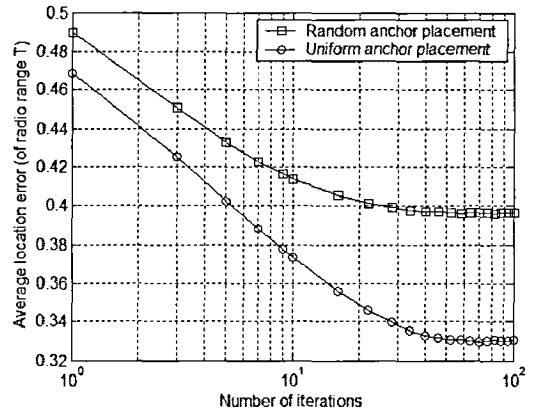


(b) 전체 링크의 분산

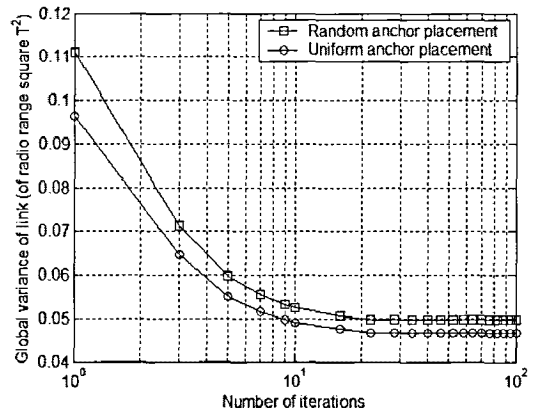
그림 6. 총 60개의 일반노드가 배치한 경우에 앵커노드의 비율에 따른 제안하는 알고리즘의 성능 평가

4.3 앵커노드의 배치

일반노드의 위치계산 정확도는 노드의 배치 방법에 따라 달라질 수 있다. 제안하는 알고리즘에서 일반노드는 앵커노드의 정보를 사용하여 일차적인 위치를 계산한 이후에 이웃노드의 위치정보를 활용하기 때문에 이때 앵커노드를 이웃하는 일반노드가 많을수록 일반노드의 위치계산 정확도는 향상될 수 있다. 따라서 이 모의실험에서는 앵커노드의 배치 방법을 균일 배치와 랜덤 배치로 나누어 각 방법에 따른 제안하는 알고리즘의 성능을 살펴본다. 일정한 공간에 총 100개의 일반노드를 랜덤하게 배치하고, 그 중 10%의 비율로 앵커노드를 각 배치 방법에 따라 균일 또는 랜덤하게 지정한다. 그림 7은 앵커노드의 배치 방법에 따른 제안하는 알고리즘의 성능을 보인다. 이미 충분한 수의 노드가 배치되어 있기 때문에 초기의 위치오차는 앵커노드의 배치 방법에 관계없이 비슷하였다. 그러나 위치보정 과정에서 앵커노드의 배치 방법에 따



(a) 평균 위치오차



(b) 전체 링크의 분산

그림 7. 총 100개의 일반노드와 앵커노드의 비율이 10%인 경우에 앵커노드 배치에 따른 제안하는 알고리즘의 성능 평가

라 큰 차이를 보였다. 앵커노드가 균일하게 배치한 경우 평균 위치오차의 감소율이 랜덤 배치에서 평균 위치오차의 감소율보다 컸다. 초기에 두 방법에 따른 평균 위치오차는 약 0.02 T의 차이를 보였지만, 위치보정 이후에는 대략 7%의 차이를 보였다. 노드 간 링크의 분산은 앵커노드가 균일하게 위치하였을 때 랜덤하게 위치한 경우보다 높았다. 모의실험을 통해서 노드의 밀도와 노드의 배치 방법에 따른 제안하는 알고리즘의 성능을 살펴보고, 이웃노드 간 거리를 균등하게 함으로써 위치계산의 정확도가 향상됨을 보였다.

VI. 결 론

기존의 Range-free 방식에서 일반노드는 앵커노드와의 거리정보와 앵커노드 외의 노드 간 관계를 고려하지 않고 오직 앵커노드의 정보에만 기반을 두어 위치를 계산하기 때문에 그 정확도가 매우 낮다. 이러한 문제를 극복하기 위해서 본 논문에서는 일반노드가 앵커노드의 정보 외에도 자신의 이웃노드의 정보를 참조하여 위치를 계산하는 알고리즘을 제안하였다. 일차적으로 앵커노드의 정보를 이용하여 자신의 위치를 계산한 일반노드는 이후 이웃노드의 위치정보를 활용하여 자신의 모든 이웃노드와 평균 거리에 균등하도록 위치를 계산한다. 이때 일반노드는 칼만 필터를 통해서 안정된 평균 거리를 계산하고 이웃노드와 평균 거리에 위치할 수 있도록 힘-벡터 방법에 기반을 두어 위치를 보정한다. 제안하는 알고리즘의 성능을 살펴보기 위해 다양한 환경에서 모의실험을 수행하였고, 이를 통해서 소수의 앵커노드만을 이용하여 높은 위치 계산 정확도를 보임을 확인하였다. 제안하는 알고리즘에서는 단일 홉에 위치한 이웃노드의 위치정보만을 이용하여 위치를 보정하기 때문에 위치보정에 소요되는 시간이 다소 길다. 그렇기 때문에 단일 홉 이상의 다수 홉에 위치한 노드의 정보 혹은 앵커노드의 정보를 지속적으로 활용하여 보다 빠르고 정확한 위치계산이 추가 연구 사항으로 남는다.

참 고 문 헌

[1] B. Rao and L. Minakakis, "Evolution of Mobile Location-based Services," *Communications of the ACM*, Vol.46, No.12, pp.61-65, December 2003.

[2] I. Getting, "The Global Positioning System," *IEEE Spectrum*, Vol.30, iss.12, pp.36-47,

December 1993.

[3] M. Robinson and I. Psaromiligkos, "Received Signal Strength Based Location Estimation of a Wireless LAN Client," *IEEE WCNC 2005*, pp.2350-2354, March 2005.

[4] X. Li and K. Pahlavan, "Super-resolution TOA estimation with diversity for indoor geolocation," *IEEE Trans. Wireless Communication*, Vol.3, pp.224-234, January 2004.

[5] N. B. Priyantha, A. Chakraborty and H. Balakrishnan, "The Cricket Location-Support System," *Proc. ACM MobiCom 2000*, pp.32-43, August 2000.

[6] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," *INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol.3, No.30, Mar.-Apr. 2003.

[7] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications Magazine*, Vol.7, iss.5, pp.28-34, October 2000.

[8] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *ACM MobiCom'03*, pp.81-95, September 2003.

[9] D. Niculescu and B. Nath, "Ad Hoc Positioning System(APS)," in *Proc. of the IEEE International Conference on INFOCOM*, pp.2926-2931, November 2001.

[10] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in *2nd ACM Conf. on Embedded Networked Sensor Systems*, pp.50-61, Nov. 2004.

[11] A. A Kannan, B. Fidan, and G. Mao, "Robust Distributed Sensor Network Localization Based on Analysis of Flip Ambiguities," *IEEE GLOBECOM 2008*, pp. 1-6, Nov.-Dec. 2008.

[12] M. S. Grewal and A. P. Andrews, "Kalman Filtering: Theory and Practice Using Matlab," *John Wiley & sons, Ltd*, 2001.

[13] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based used location and tracking

system," *IEEE Trans. Wireless Communication*, Vol.3, pp.224-234, January 2004.

- [14] J. Hightowe, R. Want and G. Borriello, "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength," *UW CSE '00*, Univ. of Washington, February 2000.
- [15] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel and S. R. Das, "MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access," *ACM MobiSys '07*, pp.192-205, June 2007.
- [16] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *J. Telecomm. Systems*, Vol.22, pp.267-280, Jan.-Apr. 2003.
- [17] L. Girod and D. Estrin, "Robust Range Estimation using Acoustic and Multimodal Sensing," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vol.3, pp.1312-1320, 2001.
- [18] N. Priyantha, H. Balakrishnan, E. Demaine and S. Teller, "Anchor-Free Distributed Localization in Sensor Network," *Technical Report TR-892*, MIT LCS, Apr. 2003.
- [19] 이상우, 이동열, 이채우, "무선 애드 혹 네트워크에서 향상된 DV-Hop 기반 위치인식 알고리즘", *전자공학회 논문지 CI편*, Vol.46, No.4, pp.31-41, 2009년 7월.

이삼우 (Sang-Woo Lee)

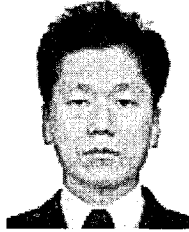
준회원



2009년 2월 아주대학교 전자공학부 학사
 2009년 3월~현재 아주대학교 전자공학과 석사과정
 <관심분야> Localization, Ad-hoc Network, Sensor Network

이채우 (Chae-Woo Lee)

정회원



1985년 2월 서울대학교 제어계측 학사
 1988년 8월 한국과학기술원 전자공학과 석사
 1995년 8월 University of Iowa 박사
 1985년 1월~1985년 12월 (주) 금성통신 연구원
 1988년 9월~1999년 3월 한국통신 선임연구원
 1999년 3월~2001년 9월 Lucent Technologies Korea 이사
 2001년 9월~2002년 2월 한양대학교 겸임교수
 2002년 3월~현재 아주대학교 전자공학과 부교수
 <관심분야> 광대역 통신망, Traffic Engineering, Ubiquitous Networking