

# 센서 네트워크의 노드 위치 추정 시스템 설계

임철우<sup>†</sup>, 김영락<sup>\*\*</sup>, 강병욱<sup>\*\*\*</sup>

## 요 약

센서 노드의 위치에 따라서 수집되는 정보의 가치가 결정되기 때문에 센서 네트워크에서 센서 노드의 위치는 매우 중요하다. 본 논문에서는 센서 네트워크에서 인접 노드에 대한 정보와 수신되는 신호의 세기를 이용해서 노드의 위치를 추정하는 기법을 제안한다. 제안하는 기법을 이용하면 노드가 알고 있는 정보를 이용해서 쉽게 노드의 위치를 찾을 수 있다. 더욱이 위치를 알고 있는 앵커 노드를 기준으로 위치를 모르는 미지 노드에 대한 상대적인 위치를 찾을 수 있어서 전체적인 센서 노드의 분포를 쉽게 알 수 있다. 그리고 동적인 시스템인 센서 네트워크에서 제안한 기법을 구현하기 위해서 UML의 여러 다이어그램 중에서 유즈 케이스 다이어그램, 액티비티 다이어그램, 상태 머신 다이어그램을 이용해서 분석 및 설계를 하였다. UML을 이용하기 때문에 노드 위치 추정 시스템에서 제안 기법의 각 기능에 대한 흐름을 정확하게 파악할 수 있었다. 그리고 센서 노드들의 위치가 작은 오차를 가지고 추정됨을 확인하였다.

## Design of Node Position Estimation System for Sensor Networks

ChulWoo Rhim<sup>†</sup>, YoungRag Kim<sup>\*\*</sup>, ByungWook Kang<sup>\*\*\*</sup>

## ABSTRACT

The value of sensing information is decided according to positions of sensor nodes, which are very important in sensor networks. In this paper, we propose a method that estimates positions of nodes by using adjacent node information and received signal strength in a sensor network. With the proposed method, we can find positions of nodes easily because we use information that nodes have. Moreover, we can find distribution easily for all the nodes because we can measure a relative position for a node whose position is not known based on anchor nodes whose positions are already known. We utilized Use case diagram, activity diagram and State machine diagram among several diagrams of UML to implement proposed method in sensor networks that is dynamic system. We can understand exact flow for each function of the proposed method in node position estimation system can be implemented easily. And we can be confirmed that the position of estimated nodes has a little error.

**Key words:** Position Estimation(위치추정), Sensor Network(센서 네트워크), Adjacency(인접성), RSS (수신 신호 세기), UML(통합 모델링 언어)

## 1. 서 론

모든 사물과 공간에서 통신을 할 수 있다는 유비쿼터스 컴퓨팅(Ubiquitous Computing)이라는 개념

[1]이 제안된 이후 컴퓨팅 분야에서는 유비쿼터스 컴퓨팅 환경을 제공하기 위해서 많은 다양한 연구가 진행되고 있다. 센서 네트워크(Sensor Network)는 이러한 연구 중의 한 분야이다. 센서 네트워크는 크

※ 교신저자(Corresponding Author) : 강병욱, 주소 : 경상북도 경산시 대동 241-1번지(712-749), 전화 : 053)810-2552, FAX : 053)819-1976, E-mail : bwkang@yu.ac.kr  
접수일 : 2009년 4월 6일, 수정일 : 2009년 5월 18일  
완료일 : 2009년 7월 1일

<sup>†</sup> 정회원, 영남대학교 컴퓨터공학과  
(E-mail : brain95@ynu.ac.kr)

<sup>\*\*</sup> 준회원, 영남대학교 컴퓨터공학과  
(E-mail : yrkim@ynu.ac.kr)

<sup>\*\*\*</sup> 정회원, 영남대학교 전자정보공학부 컴퓨터공학 전공 교수

기가 작은 수많은 저전력 센서들을 사용자가 지정하는 공간에 설치하여 각각의 센서 노드가 감지하는 주변 환경에 대한 데이터를 수집하고 이를 중앙의 시스템으로 전달하도록 구성된 네트워크이다[2].

센서 네트워크에서 센서 노드들은 특정한 위치를 지정해서 설치하는 것보다는 임의적으로 설치하는 경우가 많다. 이때, 배치된 센서 노드의 위치를 모른다면 그 센서 노드가 수집한 데이터는 의미가 없게 된다. 즉, 데이터가 수집된 위치가 어디인지 알아야만 의미 있는 데이터가 되는 것이다. 그렇기 때문에 센서 노드의 위치를 찾는 것은 센서 네트워크상에서 가장 먼저 해야 될 과제이다[3,4]. 간단한 방법으로는 각 센서 노드에 GPS(Global Positioning System) 장비를 설치하면 되지만, GPS 장비는 센서 노드에 비해 크거나 전력소모가 크다. 그렇기 때문에 센서 네트워크에서 사용하기에는 부적합하다. 그러므로 별도의 장비가 없이 센서 네트워크상에서 노드의 위치를 찾는 방법이 필요하다. 센서 노드의 위치 추정은 센서 노드의 위치를 결정하는 것이며, 이를 위해서 다양한 기법들이 연구되고 제안되고 있다.

센서 노드의 위치를 추정하는 기법은 노드 간의 거리 정보를 기반으로 하는(range-based) 방법과 거리 정보를 기반으로 하지 않는(range-free)방법으로 크게 구분할 수 있다. 전자는 위치를 알고 있는 두 노드의 신호세기[5] 또는 전파 도달 시간[6]등을 사용해서 계산된 거리를 기반으로 위치를 추정한다. 그렇기 때문에 센서 노드의 위치를 정확하게 추정해낼 수 있다. 후자는 두 노드 간의 연결성[7]이나 홉-카운트[8] 정보를 이용해서 위치를 추정한다. 이러한 거리 정보에 기반을 하지 않는 방법은 앞에서 서술한 기법에 비해서 정확도 측면에서는 불리하나 훨씬 적은 정보를 이용해서 노드의 위치를 추정할 수 있다.

한편, 소프트웨어 모델링에 대한 OMG(Object Management Group)의 표준인 UML (Unified Modeling Language)은 소프트웨어 시스템의 구조와 행위를 분석하고 표현하기 위한 다양한 다이어그램을 제공하고 있다[9]. 본 논문에서는 여러 가지 다이어그램 중에서 시스템의 동적인 측면을 표현하기 위해서 시스템이 수행해야 할 일련의 활동들을 정의한 유즈 케이스(Use Case diagram)와 행위를 표현하기 위한 액티비티 다이어그램(Activity diagram) 그리고 객체가 수행하는 상태에 대해서 표현한 상태 머신 다이어

그램(State machine diagram)을 사용한다[10].

일반적으로 센서 네트워크에서 위치가 알려진 노드를 앵커 노드(anchor node) 그리고 위치를 알 수 없는 노드는 미지 노드(unknown node)라고 한다. 앵커 노드는 사용자에 의해서 배치되기 때문에 GPS를 이용하거나 수동으로 설정할 수 있다. 이러한 앵커 노드의 위치는 임의로 설정이 가능하기 때문에 이 노드가 이동하면 모든 센서 노드의 위치는 다시 추정해야 한다. 본 논문에서는 앵커 노드를 중심으로 노드의 인접성 정보와 수신하는 신호 세기에 의해 구해진 거리를 이용하여 센서 노드의 위치를 추정하는 기법에 대해서 제안을 한다. 이 제안된 기법을 사용하게 되면 시작 노드인 앵커 노드를 기준으로 미지 노드들이 제공하는 신호 세기와 인접성을 이용하여 각 미지 노드의 위치를 추정하며, 기준인 앵커 노드를 중심으로 하는 미지 노드들의 전체적인 분포를 찾아낸다. 제안한 기법을 이용하여 노드의 위치 추정하는 시스템을 설계하기 위해서 통합 모델링 언어인 UML을 이용해서 분석 및 설계를 한다.

논문은 다음과 같은 순서로 구성되어 있다. 2장에는 기존의 위치를 추정하기 위한 기법들과 UML에 대한 내용을 살펴보고, 3장에서는 각 노드의 신호세기를 이용하여 노드의 위치를 추정하는 기법을 제안한다. 4장에서는 제안하는 기법을 이용한 시스템을 UML 모델을 사용해서 분석 및 설계를 하고, 5장에서는 이에 대해서 시뮬레이션을 수행한다. 6장에서는 본 논문의 결론을 맺는다.

## 2. 관련연구

센서 노드의 위치 추정 기법은 거리에 독립적인 방법과 거리에 기반을 하는 방법으로 구분할 수 있다. 거리에 독립적인 방법은 센서 노드 간의 거리 대신 연결성 정보 등을 이용하는 방법이며, 거리에 기반을 하는 방법은 센서 노드 간의 거리를 계산하여 센서 노드의 위치를 추정하는 방법이다. 그리고 UML을 사용하여 분석 및 설계하기 위해서 유즈 케이스 다이어그램, 액티비티 다이어그램 그리고 상태 머신 다이어그램을 소개한다.

### 2.1 거리에 독립적인 위치 추정 방법

거리에 독립적인 방법은 센서 노드 간의 거리를

직접적으로 계산하지 않고 위치를 추정하는 방법이다. 대표적인 방법은 Centroid, APIT(Approximate Point In Triangle), DV-Hop(Distance Vector-Hop) 등이 있다.

### 2.1.1 Centroid

Centroid 방법은 앵커 노드의 위치가 알려져 있을 때 연결성만을 근거로 하여 미지 노드의 위치를 추정하는 방법이다. 각 앵커 노드들은 특정 무선 전송 범위를 가지고 중첩 배치되며, 주기적으로 자신의 위치 정보를 담은 비컨을 전송한다. 이때 미지 노드는 연결성이 있는 앵커 노드들로부터 비컨 신호를 수집한다. 미지 노드들은 인접한 앵커 노드들의 위치 정보가 포함된 비컨 신호를 중점 공식을 사용하여 위치를 추정한다[7].

### 2.1.2 APIT

APIT는 일반 노드 주변에 통신 가능한 기준 노드가 비교적 많을 경우에 사용 가능한 방법이다. 위치를 모르는 일반 노드는 무선 통신을 통해 주변에 있는 한 홉 기준 노드를 찾고, 찾은 노드들 중 3개의 노드를 선택해서 삼각형을 만든다. 이때 만들어진 삼각형에 대하여 일반노드가 삼각형의 내부 또는 외부에 있는지 판단하여 주변의 기준 노드들로 만들 수 있는 모든 삼각형에 대해 반복 작업을 수행한다. 그렇게 하면서 전체를 한 홉 네트워크 영역에서 일반 노드의 위치 범위를 줄여간다. 일반 노드의 최종 위치는 모든 삼각형의 영역에서 중첩된 부분이 가장 많은 영역이 된다[11].

### 2.1.3 DV-Hop

DV-Hop은 노드의 배치 분포가 균일하고 기준 노드의 밀도가 적은 경우에 사용할 수 있는 방법이다. 이 방법은 노드 사이의 연결성을 이용하여 노드 간의 거리를 추정한다. 각 기준 노드는 자신의 위치 정보를 다른 기준 노드로 멀티-홉 플러딩 기법으로 전송한다. 이때 다른 기준 노드로 전송되기 전까지의 가장 짧은 홉 카운트를 저장한다. 하나의 기준 노드가 다른 기준 노드로부터의 메시지를 받으면 두 기준 노드의 위치 좌표를 통해서 직선거리를 구하고 메시지가 전송되기까지의 가장 적은 홉 수로 나누어 노드 간 한 홉의 평균 거리를 추정한다. 이렇게 두 기준을 통해 구해진 노드 간 평균 거리는 다시 네트워크 전

체로 전송되어 전체 네트워크에서의 노드 간 평균 거리를 계산하는데 사용된다. 노드 간 평균거리가 구해지면 일반 노드의 위치는 기준 노드까지의 홉 수를 곱해 얻은 거리 값을 이용해 삼각측량법을 이용해서 구한다[12].

## 2.2 거리에 기반을 하는 위치 추정 방법

거리에 기반을 하는 방법은 센서 노드간의 거리를 계산하여 위치를 추정한다. 이러한 기법으로는 노드가 수신하는 신호의 도착 시간을 이용하는 ToA(Time of Arrival), TDoA(Time Difference of Arrival), 수신 신호의 각도를 계산해서 사용하는 AoA(Angle of Arrival), 수신 신호의 세기를 이용하는 RSSI(Received Signal Strength Indicator) 그리고 DV-Distance 방법 등이 있다.

### 2.2.1 ToA

ToA는 신호가 도착하는 시간을 이용해서 위치를 추정하는 알고리즘으로 3개 이상의 참조 센서에서 송·수신하는 신호를 이용해서 위치를 찾는다. 센서에서 목표까지의 거리는 신호의 종류에 따라서 적절한 상수를 곱해서 산출할 수 있으며, 센서와 목표 사이의 거리를 반경으로 하는 원들의 교점이 목표의 위치가 된다. 보다 정확하게 위치를 하기 위해서는 참조 센서 노드의 개수를 늘려야 하며, GPS(Global Positioning System)가 이 기법의 대표적인 예이다[6,13].

### 2.2.2 TDoA

TDoA는 목표와 두 개 이상의 참조 센서가 송·수신하는 신호의 도착 시간의 차이를 측정하여 센서 간의 거리차가 일정한 지점의 목표를 찾아내는 알고리즘이다. 즉, 참조 센서 노드를 초점으로 하는 쌍곡선을 사용하여 위치를 추정한다. 예를 들어 3개 이상의 센서가 존재할 경우 여기에서 구해지는 2개의 쌍곡선을 이용하여 그 교점을 찾아내서 목표의 2차원 위치를 찾는다. 4개 이상의 센서를 이용하게 되면 목표의 3차원 위치를 찾을 수 있다. 정밀한 위치 인식을 위해서는 목표와의 신호 동기화가 필요하다[14].

### 2.2.3 AoA

AoA는 각도를 측정하여 위치를 추정하는 대표적인 방법이다. 참조 센서에서 목표가 보내는 신호의

방향각을 이용하여 각을 추정하고 각 참조 센서와 목표 사이의 방향각의 교차점을 계산하여 목표의 위치를 추정하는 알고리즘이다. 목표의 위치를 추정하기 위해서는 최소 2개의 방향각이 필요하지만 보다 정확한 위치를 추정하기 위해서는 여러 개의 방향각을 사용한다[15].

### 2.2.4 RSSI

RSSI는 수신기에서 수신한 전파의 세기가 거리에 따라 달라지는 점을 이용하여 비컨과 수신기 사이의 거리를 측정하는 방식이다. 수신된 신호의 세기가 거리가 증가하게 되면 수신 신호의 세기는 로그 함수의 형태로 감소하게 된다. 이를 이용해서 위치를 추정하기 위해서는 통계적인 방법에 근거해서 확률 분포와 대조한 후에 위치를 추정하는 방법과 신호의 감쇄 특성을 이용해서 수학적으로 모델링한 식을 이용하는 방법이 있다. 이러한 수신 신호 세기는 전파의 전파 환경에 따라 달라지며, 이를 위해서 해당 환경에 대한 경로 손실 모델이 필요하다[6].

### 2.2.5 DV-distance

DV-distance는 레퍼런스 노드와의 홉 수를 전파하는 대신에 hop-by-hop으로 신호의 세기를 기반으로 거리를 측정하여 전파하는 방식으로 이전의 DV-Hop과 유사하다. 매트릭으로써 미터 단위의 누적 거리를 사용하고 있다. 모든 홉이 동일한 크기를 가지고 있지 않기 때문에 DV-Hop보다 측정 오차에 대해서는 민감하다[12].

## 2.3 동적인 시스템 표현을 위한 UML 다이어그램

센서 네트워크는 정적인 시스템에서 사용하는 것이 아니라 동적인 시스템이다. UML의 여러 다이어그램 중에서 동적인 시스템을 표현하기 위해서 사용하기 위해서 유즈 케이스 다이어그램, 액티비티 다이어그램 그리고 상태 머신 다이어그램을 이용한다.

### 2.3.1 유즈 케이스 다이어그램

유즈 케이스 다이어그램은 시스템이 제공하는 기능과 시스템을 사용하는 사용자(또는 다른 시스템)를 표현하는 다이어그램이다. 사용자의 요구사항으로부터 행위자와 각각의 유즈 케이스를 도출하고 그 내부의 시나리오 및 이벤트의 흐름을 파악한다. 이

표 1. 유즈 케이스 다이어그램 구성요소

구성 요소	설 명
액터	시스템과 상호 작용하는 개체이다.
유즈 케이스	시스템의 행위 또는 기능을 정의한다.
연관	클래스들 사이의 의미적 관계를 정의한다.
일반화	일반적인 요소와 더 구체적인 요소를 연결한다.
의존	요소의 구현이나 기능을 위해 다른 요소의 존재가 요구된다.

다이어그램은 액터(Actor), 유즈 케이스(Use Case), 연관(Association), 일반화(Generalization), 의존(Dependency)이 있다. 표 1은 유즈 케이스 다이어그램에서 사용하는 요소를 설명하고 있다[6].

### 2.3.2 액티비티 다이어그램

액티비티 다이어그램은 순차적인 제어 흐름뿐만 아니라 병렬적으로 수행되는 활동과 분기가 이루어지는 것에 대해서 표현할 수 있다. 기본적으로 액티비티(Activity), 시작점(Initial State), 종료점(Final State), 판단(Decision, Branch), 동기화 막대(Synchronization Bar), 전이(Transition), Swim Lane으로 구성된다. 표 2는 액티비티 다이어그램의 구성요소에 대한 설명이다[6,10].

### 2.3.3 상태 머신 다이어그램

상태 머신 다이어그램은 전체 시스템을 대상으로

표 2. 액티비티 다이어그램 구성요소

구성 요소	설 명
액티비티	액티비티의 액션을 나타내는 것으로 행위나 작업을 나타낸다.
시작점	처리 흐름의 시작하는 곳을 의미한다.
종료점	처리 흐름이 종료하는 곳을 의미한다.
판단	분기가 일어나는 곳으로 하나의 입력에 두 개의 흐름으로 분기가 일어날 수 있다.
동기화 막대	하나의 흐름을 두 개로 분할하거나 제어의 흐름을 두개 또는 그 이상의 동시에 발생하는 제어의 흐름으로 동기화한다.
전이	하나의 액션이나 액티비티로부터 다른 액션이나 액티비티로 이동하는 경로를 나타낸다.
Swim Lane	액티비티에 대해서 책임 있는 조직적인 단위를 표현하는 각 그룹을 나타낸다.

표 3. 상태 머신 다이어그램 구성요소

구성요소	설 명
상태	객체나 시스템이 가질 수 있는 상태를 의미한다.
시작상태	상태 머신 다이어그램의 시작 상태를 의미한다.
종료상태	상태 머신 다이어그램의 종료 상태를 의미한다.
선택	조건에 따라 여러 가지 흐름 중 하나로 분기됨을 의미한다.
전이	상태가 다른 상태로 전환되는 것을 의미한다.
분할과 합류	상태의 전이가 여러 흐름으로 분할(Fork)되거나 여러 흐름이 합류(Join)됨을 의미한다.

하여 이벤트에 따른 객체의 상태 변화를 나타내는 다이어그램이다. 상태와 전이 그리고 이벤트 등이 표현되며 상태가 전이되기 위한 이벤트, 조건, 행동을 포함하고 있다. 이 다이어그램은 시작 상태(Initial State), 종료 상태(Final State), 상태(State), 선택(Choice), 전이(Transition), 분할(Fork)과 합류(Join)로 구성되어 있다. 표 3은 상태 머신 다이어그램의 구성요소에 대한 설명이다[6,16].

### 3. 노드의 인접성과 거리를 기반으로 하는 노드 위치 추정 기법

본 논문에서는 앵커 노드 이외의 미지 노드를 추정하기 위해서 축이 되는 두 노드와 미지 노드 간의 인접성 정보와 노드 간의 신호 세기를 이용하여 노드 간의 거리를 계산한다. 여기에서 제안하는 기법은 모든 센서 노드들이 평면상에 있고, 각 센서 노드들에서의 송·수신되는 신호의 세기는 균일함을 전제로 한다.

#### 3.1 노드간 거리 계산

두 노드 간의 신호 세기를 이용하여 거리를 구하기 위해서 Friis Free Space 공식을 거리에 대해서 정리한 식(1)을 사용한다.

$$\frac{P_r}{P_t} = G_t G_r \left( \frac{\lambda}{4\pi R} \right)^2 \quad (1)$$

여기서,  $G_t$ 는 송신 안테나의 이득,  $G_r$ 은 수신 안테

나의 이득,  $P_r$ 은 수신된 전력,  $P_t$ 는 송신된 전력 그리고  $\lambda$ 는 전파의 파장,  $R$ 은 거리이다. 식(1)에서  $\lambda$ 는  $c/f$ 로 바꾸어서 표현할 수 있다. 여기에서  $c$ 는 광속이고,  $f$ 는 무선 신호의 주파수이다. 이를 사용해서 식(1)을  $R$ 에 대해서 정리하면 다음의 식(2)이 된다. 본 논문에서는 이 식(2)을 이용하여 노드 간의 거리를 계산한다[17,18].

$$R = \frac{c}{4\pi f} \sqrt{\frac{G_t G_r P_t}{P_r}} \quad (2)$$

#### 3.2 노드 간의 인접성 확인

노드 간의 인접성이란 노드 간에 서로 통신이 가능하다는 것을 의미한다. 그림 1에서 보는 것과 같이 Origin 노드와 Axis member 노드가 미지 노드와의 신호 세기를 수집하여 미지 노드와 통신이 가능함을 확인하고 있다. 인접성이 확인이 된 노드의 위치는 두 노드에서 위치를 추정할 수 있다.

#### 3.3 앵커 노드에서 미지 노드의 위치 추정

본 논문에서는 앵커 노드를 중심으로 미지 노드의 위치를 찾아서 평면상에 배치된 센서 노드의 전체적인 분포를 찾는다. 이를 위한 첫 번째 단계로 앵커 노드를 기준점으로 미지 노드들의 상대적인 위치를 찾아야 한다. 미지 노드들의 상대적인 위치는 다음과 과정들을 통해서 찾는다.

##### 3.3.1 앵커 노드에서 기준 축 설정

사용자에 의해서 위치가 지정된 앵커 노드를 중심으로 미지의 센서 노드들의 상대적인 위치를 찾기 위해서 다음의 과정을 통해서 앵커 노드에서 센서 노드들의 위치를 찾는다.

원점 노드인 앵커 노드에서 각 센서 노드와의 인

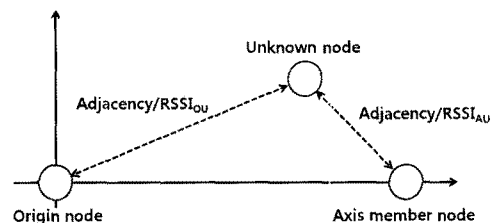


그림 1. 노드의 인접성 확인의 예

접성 여부를 확인하기 위해서 노드 간의 통신을 한다. 그리고 통신이 가능한 센서 노드와 식(2)를 이용하여 원점 노드와 인접한 미지의 센서 노드간의 거리를 계산한다. 앵커 노드에서 통신이 가능한 노드와 그 노드와의 거리를 인접성 정보 행렬로 구성한 다음에 거리가 원점 노드에서 최소인 노드를 찾아서 기준 축의 멤버 노드로 선택한다. 그림 2는 기준 축을 설정하는 알고리즘을 기술하고 있다.

### 3.3.2 계산된 거리를 이용한 센서 노드 위치 계산

설정된 기준 축에서 미지 노드를 구하는 과정을 그림 3에서 보이고 있다. 그림 3에서 위치를 알고 있는 앵커 노드는 A이고 기준 축 설정 알고리즘에 의해 구해진 기준 축 멤버 노드는 B이다. 여기서 식(4)를 이용하여 미지 노드의 위치를 계산한다. 미지 노드의 위치를 계산할 때, 그림 3에서 보는 것과 같이 설정한 축의 두 센서 노드의 통신 범위 내에서 공통적으로 통신이 가능한 센서 노드에 대해서 위치를 찾고 있음을 볼 수 있다. 따라서 미지 노드의 위치를

<p>입력 값 : 신호 세기 출력 값 : 설정된 기준축</p> <ol style="list-style-type: none"> <li>1 : 앵커 노드와 각 노드간의 통신을 확인한다.</li> <li>2 : 수신 신호 세기로부터 앵커 노드와 각 미지 센서 노드 간의 거리를 계산한다.</li> <li>3 : 앵커 노드에서의 인접성 정보 행렬을 만든다. (이 행렬은 통신 가능 노드 정보와 계산된 거리로 구성된다.)</li> <li>4 : 앵커 노드와 통신 가능한 노드 중에서 최소의 거리를 가지는 센서 노드를 찾는다.</li> <li>5 : 찾아낸 센서 노드와 기준 축을 설정한다. (기준축 설정 좌표 : 원점(0,0), 축 멤버 노드(최소 거리, 0))</li> <li>6 : 설정된 기준 축을 반환한다.</li> </ol>
---

그림 2. 기준 축 설정 제한 알고리즘

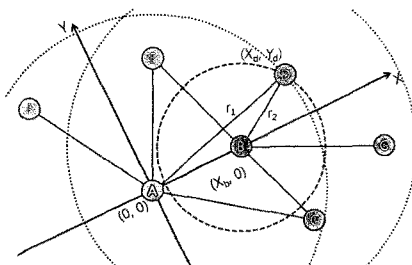


그림 3. 미지 노드 D를 찾기 위한 노드 정보

계산하기 위해서 설정된 축의 두 센서 노드가 공통적으로 통신을 하고 있는 센서 노드의 수와 공통적으로 통신을 하고 있는 노드를 찾아야 한다. 그림 3에서 센서 노드 A와 센서 노드 B가 위치를 추정하고 있는 센서 노드는 D이다.

$$\begin{cases} x^2 + y^2 = r_1^2 \\ (x - X_b)^2 + y^2 = r_2^2 \end{cases} \quad (3)$$

여기서,  $(X_b, 0)$ 은 노드 A에서 기준 축을 설정하기 위해서 선택된 센서 노드 B의 위치이고,  $r_1$ 은 앵커 노드 A에서부터 미지 노드 D까지의 거리,  $r_2$ 는 센서 노드 B로부터 미지 노드 D까지의 거리이다. 식(3)에서 센서 노드 D의 위치는 식(4)로 나타낼 수 있다.

$$(X_d, Y_d) = \left( \frac{r_1^2 - r_2^2 + X_b^2}{2X_b}, \pm \sqrt{r_1^2 - X_d^2} \right) \quad (4)$$

### 3.3.3 미지 노드의 위치 결정

식(4)에 의해 계산된 미지 노드인 D의 좌표 x값은 하나이나, 계산된 좌표 y값은 2개가 되고 있다. 즉, y좌표의 위치에 따라서 센서 노드의 위치가 달라지며, 그에 따라서 센서 노드의 분포가 달라진다. 그렇기 때문에 미지 노드인 D의 위치를 찾기 위해서 센서 노드 D에서 인접성 정보와 수신 신호 세기를 이용하여 계산된 거리 정보를 사용하여 해당 노드의 위치를 결정한다. 다음의 그림 4는 이러한 과정을 기술하고 있다.

## 3.4 기준 축 이후의 센서 노드에서 위치 추정

### 3.4.1 원점의 이동

기준 축을 중심으로 할 때 추정되지 않은 센서 노드들에 대해서는 원점을 이동시켜서 위치를 추정한다. 이때 원점으로 결정하는 센서 노드는 이전 축에서 원점 노드와 함께 축의 멤버로 설정했던 센서 노드를 이용한다.

### 3.4.2 생성되는 축의 회전각 계산

원점을 이동하면 센서 노드들의 위치를 추정하기 위해서 새로운 축을 설정해야 한다. 새로운 축을 설정하는 방법은 이전에 기술한 그림 2와 같다. 이 때, 새로운 축은 이전에 설정된 축에 대해서 회전을 하는 경우가 있는데, 식(5)의 회전 행렬을 이용해서 축을

입력 : 원점 노드의 정보 행렬  
출력 : 추정된 미지 노드의 위치

- 1 : 축 멤버 노드와 각 노드간의 통신을 확인한다.
- 2 : 수신 신호 세기로부터 축 멤버 노드와 각 미지 노드 간의 거리를 계산한다.
- 3 : 축 멤버 노드에서의 인접성 정보 행렬을 만든다. (이 행렬은 통신 가능 노드 정보와 거리로 구성된다.)
- 4 : 축으로 설정되는 노드들에서 공통으로 통신하는 노드의 수를 찾는다.
- 5 : 축으로 설정되는 노드들에서 첫 번째로 공통으로 통신하는 노드를 찾는다.
- 6 : 공통으로 통신하는 노드의 수만큼 루프를 수행한다.
- 7 : 인접성 정보 행렬에서 거리를 추출한다.
  - 거리1 = 원점 노드에서 공통 노드까지의 거리
  - 거리2 = 최소 거리 노드에서 공통 노드까지의 거리
- 8 : 식(4)를 이용해서 미지 노드에 대한 좌표를 계산한다.
- 9 : 계산된 좌표에서 y좌표 값이 (+)인 노드와 (-)인 노드의 노드 정보가 추정하고자 하는 센서 노드 정보와 일치할 경우
  - (+)/(-)인 위치에서 각 센서 노드와 거리 정보를 비교한다.
- 10 : 계산된 좌표에서 y좌표 값이 (+)인 노드와 (-)인 노드의 노드 정보가 추정하고자 하는 센서 노드 정보와 일치하지 않을 경우
  - (+)/(-)인 위치에서 노드들이 가지고 있는 노드 정보만 비교한다.
- 11 : 추정된 미지 노드의 위치를 반환한다.

그림 4. 미지 노드 선택 제안 알고리즘

기준 축에 대해서 회전한다.

$$\begin{pmatrix} X_r \\ Y_r \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \end{pmatrix} \quad (5)$$

여기서,  $(X_c, Y_c)$ 은 이전에 추정된 좌표이고,  $(X_r, Y_r)$ 은 회전된 좌표이다. 식 (5)에서 회전각  $\theta$ 가 있는데, 회전각을 구하는 알고리즘은 그림 5와 그림 6에서 보이고 있다. 찾아낸 노드의 위치에서 축 멤버 노드의 위치 유무에 따라서 인접성 정보가 많은 경우와 그렇지 않은 경우로 나눈다. 그림 5는 원점으로 인접성 정보가 많은 경우 회전각을 구하는 과정이다. 센서 노드가 가지고 있는 인접성 정보가 많으면 그림 5의 과정을 통해서 회전각을 계산하고 해당되는 노드를 쉽게 찾을 수 있다. 그러나 인접성 정보가 적은 경우 즉, 찾아야 하는 노드와 인접된 정보가 2개인 경우에는 미지 노드를 쉽게 찾기가 어렵다. 이를 위해서 그림 6과 같은 알고리즘을 제안한다. 찾아낸 노

입력 : 찾아낸 노드의 위치  
출력 : 계산된 회전각

- 1 : 이전에 설정된 축에서 찾아낸 노드의 위치에서 현재 축의 멤버로 설정된 센서 노드가 있는지 확인한다.
- 2 : 이전에 설정된 축에서 찾아낸 노드의 위치에서 축 멤버로 설정된 센서 노드가 있으면 계산된 위치를 이용해서 회전각을 구한다.
- 3 : 원점으로 선택된 노드를 중심으로 축 멤버로 설정된 노드의 위치에 따라서 회전각을 계산한다.
  - 1 사분면과 2 사분면에 있는 경우 : -계산된 회전각
  - 3 사분면에 있는 경우 : pi-계산된 회전각
  - 4 사분면에 있는 경우 : pi+계산된 회전각
- 4 : 계산된 회전각을 반환한다.

그림 5. 회전각 계산을 위한 제안 알고리즘(인접성 정보가 많은 경우)

입력 : 찾아낸 노드의 위치  
출력 : 계산된 회전각

- 1 : 이전의 축에서 찾아낸 노드의 위치에서 축 멤버로 설정된 노드의 위치가 없는 경우에는 축으로 설정된 센서 노드와 원점 노드와의 공통적으로 통신을 하고 있는 센서 노드를 찾는다.
- 2 : 공통으로 통신을 하고 있는 센서 노드와 원점 노드와 새로운 축을 만들어서 축 멤버로 설정된 노드의 위치를 찾는다.
- 3 : 찾아낸 축 멤버로 설정된 노드의 위치를 이용해서 회전각을 구한다.
- 4 : 원점으로 선택된 노드를 중심으로 축 멤버로 설정된 노드의 위치에 따라서 회전각을 계산한다.
  - 1 사분면과 2 사분면에 있는 경우 : -계산된 회전각
  - 3 사분면에 있는 경우 : pi-계산된 회전각
  - 4 사분면에 있는 경우 : pi+계산된 회전각
- 5 : 계산된 회전각을 반환한다.

그림 6. 회전각 계산을 위한 제안 알고리즘(인접성 정보가 적은 경우)

드의 위치에서 축 멤버 노드의 위치가 없는 경우에는 미지 노드와 원점으로 설정된 노드가 공통적으로 통신이 가능한 노드를 찾은 후에 원점 노드와 공통 노드를 축으로 설정하여 축 멤버 노드의 위치를 찾은 후에 새로운 축을 생성한다.

### 3.5 기준 축에 대한 좌표로 변환

새로운 원점에서 구해진 센서 노드의 위치를 역회

전을 통해서 기준 노드 중심의 좌표계로 환원한다. 이동된 원점에서 구해진 회전각을 식(5)에 대입하는데, 역회전에 사용되는 회전각은 그림 5와 6의 알고리즘에 의해 구한 값에서 음의 값을 가지는 경우에는  $\pi - \theta$ 를 하고 그렇지 않으면  $-\theta$ 를 식(5)에 대입해서 이전 축 중심으로 변환한다. 그러면 이전 원점에 대한 좌표 계산의 결과를 얻을 수 있다. 그리고 축 멤버 노드를 원점으로 사용했기 때문에 기준 축에 대해서 이동한 거리를 알고 있어서 구한 좌표를 전체적으로 기준 축에 대해서 이동하면 앵커 노드를 기준으로 한 노드의 전체적인 분포를 알 수 있다.

#### 4. UML을 이용한 위치 추정 시스템에 대한 분석

3장에서 센서 네트워크에서 센서 노드 위치 추정 기법을 제안하였다. 본 장에서는 제안된 기법을 구현하기 위해서 UML을 이용하여 분석 및 설계를 한다. 센서 네트워크는 정적인 시스템이 아니기 때문에 UML 다이어그램 중에서 동적인 시스템을 표현하는데 사용되는 유즈 케이스 다이어그램, 액티비티 다이어그램, 상태 머신 다이어그램을 이용해서 알고리즘에 대한 분석 및 설계를 한다. 이와 같은 분석과 설계를 하는 이유는 센서 네트워크에서 위치 추정 시스템을 구현할 때 응용 프로그램을 빠르고 쉽게 구현이 가능하기 때문이다.

##### 4.1 유즈 케이스 다이어그램을 이용한 분석

그림 7은 제안된 센서 네트워크에서 노드 위치 추정 기법을 사용하는 시스템을 유즈 케이스 다이어그램을 이용해서 분석한 것이다. 여기서 분석된 액터는 위치를 알고 있는 앵커 노드(AnchorNode), 축 멤버로 설정되는 노드(AxisMemberNode) 그리고 거리

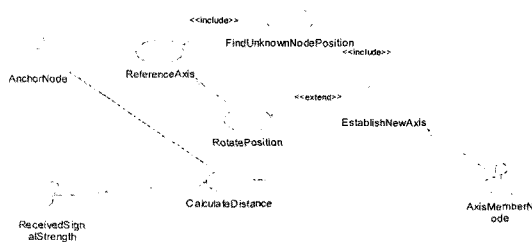


그림 7. 유즈 케이스 다이어그램으로 분석한 위치 추정 시스템

를 계산할 때 사용하는 수신 신호 세기(Received SignalStrength)이다. 유즈 케이스로 분석된 기능들은 수신 신호 세기를 이용해서 거리 계산(Calculate Distance), 기준 축을 설정(Reference Axis), 미지 노드의 위치 찾기(FindUnknown NodePosition), 다음 노드들에서 새로운 축 설정(EstablishNewAxis), 기준 축 중심으로 변환 및 이동을 하기 위한 회전(RotatePosition)이다.

##### 4.2 액티비티 다이어그램을 이용한 분석

그림 8은 기준 축을 설정 및 미지 노드의 위치를 찾는 기법을 액티비티 다이어그램을 이용해서 나타낸 것이다. 그림 2와 그림 4의 내용을 액티비티 다이어그램으로 표현하여 위치 추정에 대한 흐름을 쉽게 알아볼 수 있다. 수신 신호의 세기를 이용해서 거리를 측정하는 다음에 기준 축을 설정하기 위해 앵커 노드와 가장 가까운 센서 노드를 축의 멤버로 설정하며, 앵커 노드와 설정된 멤버 노드에서 공통적으로 통신되는 노드를 파악하여 각 센서 노드의 위치를 추정하게 된다. 센서 노드의 위치를 추정할 때, 센서 노드가 가지고 있는 인접성 정보를 이용해서 정확한 위치를 추정하게 된다. 인접성 정보를 이용하지 않게 되면 해당 센서 노드의 위치를 잘못 추정하는 경우가 발생하게 될 수도 있고, 그로 인해서 센서 노드의 전체적인 분포가 변화하게 된다.

그림 9는 회전각 계산 및 기준 축 중심으로 추정된 센서 노드들을 이동하는 과정을 액티비티 다이어그램을 이용해서 나타낸 것이다. 그림 9(a)의 액티비티 다이어그램은 그림 5와 그림 6의 내용을 표현하고 있고, 그림 9(b)의 액티비티 다이어그램은 3.5절의 이동에 관한 내용을 나타내고 있다. 그림 9(a)는 회전각을 계산할 때, 인접성 정보가 많은 경우에는 원점이 되는 센서 노드를 중심으로 축이 되는 센서 노드가 위치한 사분면의 위치에 따라 회전각을 계산하고, 인접성 정보가 적은 경우에는 즉, 축이 되는 센서 노드의 위치를 모를 경우에는 공통적으로 통신이 되는 센서 노드와 임시로 축을 생성하여 축이 되는 센서 노드의 위치를 찾은 후에 회전각을 계산하도록 한다. 그림 9(b)는 계산된 회전각을 이용해서 기준 축에 대해서 회전을 하고 앵커 노드에 대해서 원점 노드를 이동시켜서 앵커 노드 중심의 센서 노드 분포를 확인할 수 있게 한다.



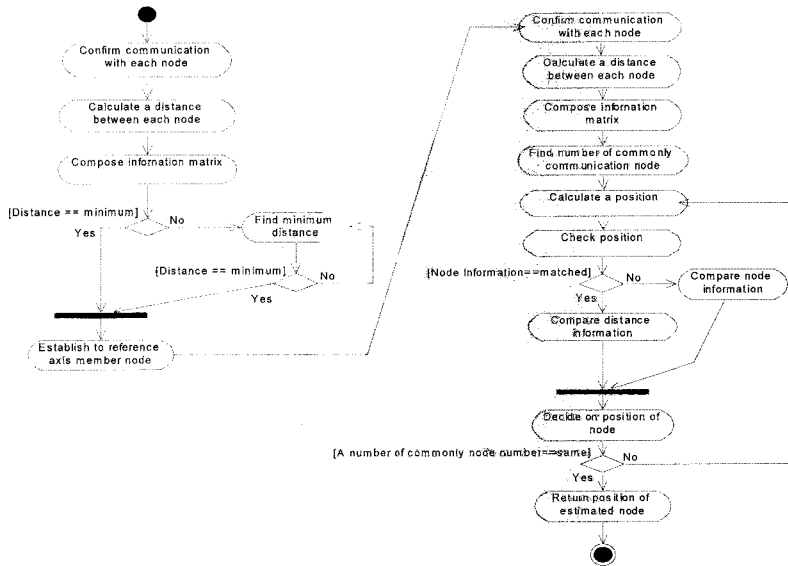


그림 8. 액티비티 다이어그램으로 표현한 기준 축 설정 및 미지 노드 설정

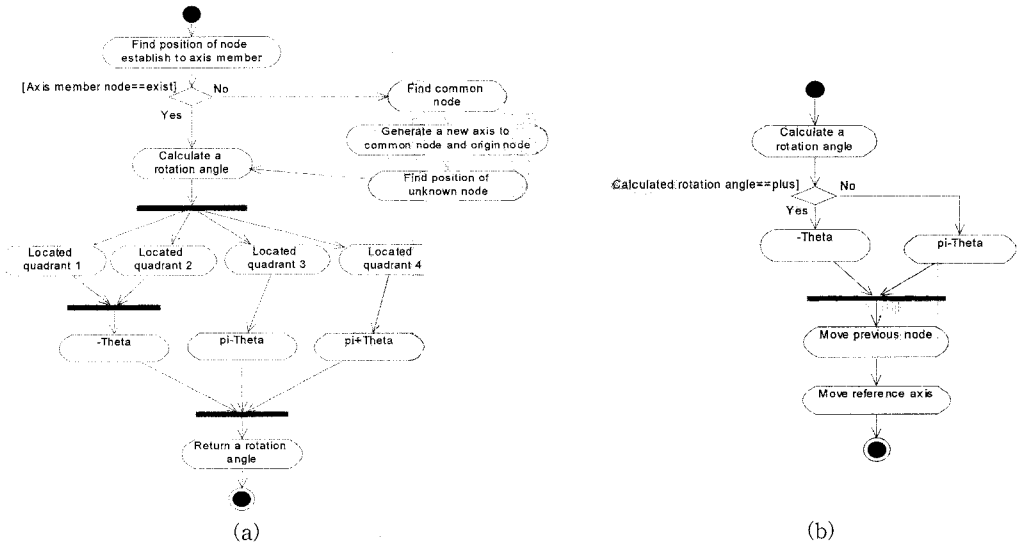


그림 9. 액티비티 다이어그램으로 표현한 회전각 계산 및 위치 이동

4.3 상태 머신 다이어그램을 이용한 분석 및 설계

그림 10은 제안한 센서 네트워크에서 노드 위치 추정 시스템을 상태 머신 다이어그램을 이용해서 분석한 것이다. 상태는 크게 Calculate Distance 상태, ReferenceAxis 상태, Find Unknown Node Position 상태, Establish New Axis 상태, Confirm node distribution 상태 5가지로 분석을 하였다. 여기에서 몇몇 상태들은 서버 머신들을 가지고 있다.

Calculate Distance 상태에서는 수신 신호 세기를 이용해서 거리를 계산하게 된다. 여기에서 계산된 거리는 각 축을 생성 및 센서 노드의 위치를 추정할 때 사용된다. 그리고 ReferenceAxis 상태에서는 기준 축을 설정하기 위한 서버 머신을 포함하고 있다. 거리를 계산하여 정보 행렬을 구성하고, 구성된 정보 행렬에서 거리가 최소인 센서 노드를 찾는다. 이 때 앵커 노드와 거리가 최소인 센서 노드가 축 멤버 노

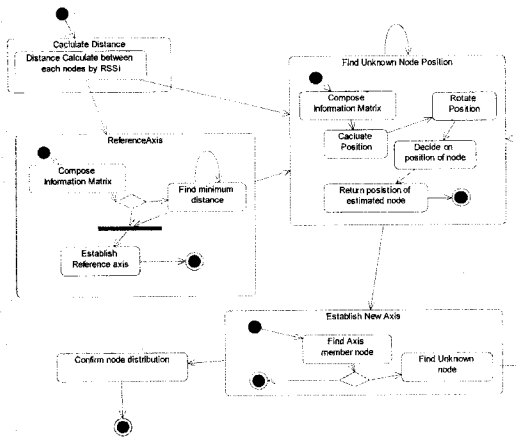


그림 10. 상태 머신 다이어그램으로 나타낸 제한 위치 추정 시스템

드로 선택된다. 축 멤버 노드가 설정되면 그 이후에는 미지 노드들의 위치를 추정해야 한다. 이 작업을 수행하는 상태가 Find Unknown Node Position 상태이다. 여기에서도 정보 행렬을 구성하고 각 센서 노드의 위치를 추정한다. 만일 축이 회전된 경우에는 위치를 회전시키게 되며, 센서 노드가 가진 인접성 정보를 이용해서 센서 노드의 위치를 추정하게 된다. 그리고 추정된 센서 노드의 위치는 축으로 설정된 센서 노드들에게 반환한다. 위치가 추정되지 않은 센서 노드를 위해서 축 멤버로 설정했던 센서 노드를 원점 노드로 설정하여 새로운 축을 생성하게 된다. 이 상태가 Establish New Axis이다. 여기에서는 이전의 기준 축 설정과 유사한 과정을 수행하고 추정해야 할 노드의 위치가 있으면 Find Unknown Node Position 상태로 전이해서 미지 센서 노드의 위치를

추정하며, 추정할 센서 노드가 없을 경우에는 작업을 종료한다. Confirm node distribution 상태에서는 기준 축에 대해서 즉, 앵커 노드를 기준으로 추정된 센서 노드의 분포를 확인한다.

### 5. 시뮬레이션

4장에서는 센서 네트워크에서 노드 위치 추정 시스템을 UML 다이어그램을 이용하여 분석을 하였다. 본 장에서는 이를 구현하여 제안하는 시스템의 성능을 확인하고자 한다. 이를 확인하기 위해서 다음의 그림 11과 같이 노드를 임의적으로 배치한다. 여기서 1번 노드는 위치를 알고 있는 앵커 노드로 한다. 2번부터 14번 노드는 위치를 모르고 있는 미지 노드이다. 시뮬레이션에서 각 센서 노드의 송신 안테나의 이득, 수신 안테나의 이득, 수신된 전력 및 송신된 전력은 각 센서 노드마다 일정하고, 모든 노드들은 평면상에 배치되어 있음을 전제로 한다.

$$RSSI = -(10 \log_{10} d + A) \tag{6}$$

식(6)은 시뮬레이션에서 거리 측정을 위해서 사용한 수신 신호의 세기를 나타내는 것으로 이때 생성된 RSSI값에 백색 가우시안 잡음(White Gaussian Noise)을 더해 사용하였다. 여기서,  $n$ 은 신호 전파 상수,  $d$ 는 송신지로부터의 거리,  $A$ 는 1미터의 거리에서 수신된 신호의 세기(dBm)를 나타낸다[19].

그림 12는 앵커 노드로부터 센서 노드의 위치를 찾기 위해서 설정된 기준 축 및 사용된 축과 각 축에

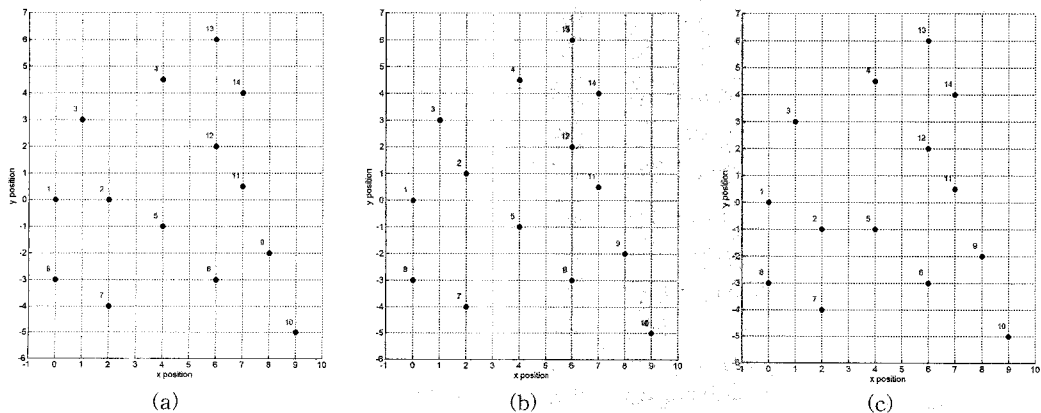


그림 11. 제안 기법 검증을 위한 센서 노드 위치

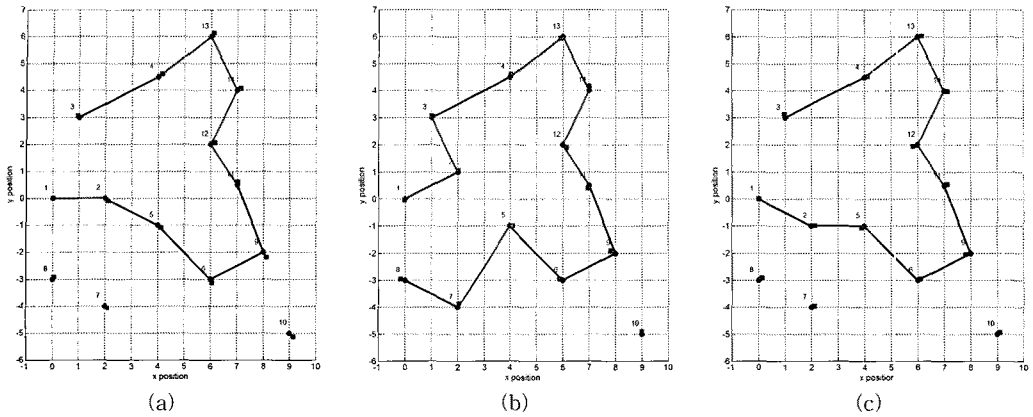


그림 12. 설정된 축과 추정된 센서 노드의 위치

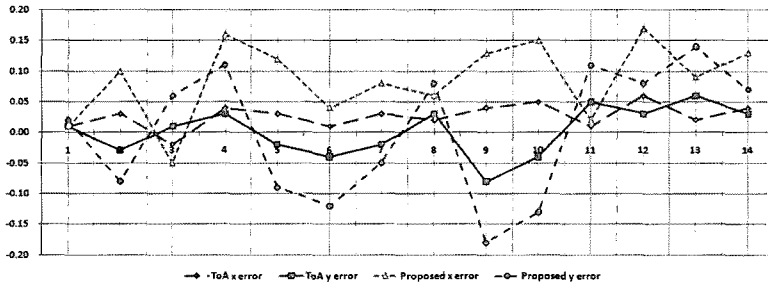


그림 13. 그림 11(a)의 토폴로지에 대한 ToA와 제안 시스템의 오차 비교

서 추정된 센서 노드들의 위치를 나타내고 있다. 분석된 시스템을 이용해서 추정한 센서 노드들이 대부분의 시물레이션을 위해서 설정된 위치를 근사하게 추정해내고 있음을 알 수 있다.

그림 13은 그림 11(a)의 토폴로지에 대해서 비컨 노드를 3개 이상 사용하는 ToA 알고리즘을 이용해서 추정된 위치 오차와 제안하는 시스템을 이용하여 추정된 위치 오차를 보이고 있다. 제안하는 시스템은 하나의 비컨 노드를 이용했지만 3개 이상의 비컨을 이용하는 ToA 알고리즘에 의한 위치 오차와 크게 차이가 나지 않으면서 유사한 형태를 보이고 있다.

표 4는 제안하는 시스템을 이용해서 각 축별로 추정된 노드의 위치와 검증용 위해 사용한 수신 신호의 세기에 대해서 20회의 시물레이션을 반복한 결과이다. 여기에서 추정된 위치를 보면 기준 축이 설정되는 위치에 따라서 센서 노드들의 위치가 달라지고 있음을 알 수 있다. 즉, 앵커 노드를 중심으로 각 센서의 절대적인 위치가 아닌 상대적인 위치를 나타내고

있다. 표 4에서 추정된 센서 노드의 위치의 표준편차를 보면 대부분 소수점 둘째자리 이하의 값들을 보이고 있다. 그렇기 때문에 제안하는 시스템에서 추정하고 있는 센서 노드들의 위치는 시물레이션을 위해 배치한 센서 노드들의 위치를 근접하게 추정해내고 있다고 할 수 있다.

위치를 추정하는데 있어서 위치 오차들이 발생하는데 이러한 오차가 발생한 이유로 볼 수 있는 것은 먼저 수신 신호 세기를 이용해서 거리를 계산할 때 잡음이 더해진 수신 신호 세기를 이용했기 때문에 정확한 거리를 계산해내지 못해서 발생한 것이 첫 번째 원인으로 볼 수 있으며, 두 번째 이유는 삼각 함수를 이용하는 회전 행렬에 의해서 발생한 것이라 볼 수 있다. 그리고 축을 회전하면서 발생하는 회전 행렬 오차 및 앵커 노드 중심으로 모든 센서 노드를 이동할 때 발생하는 누적 오차에 의해서 시물레이션을 위해 배치한 센서 노드들의 위치에 대해서 오차를 가지고 센서 노드의 위치가 추정된 것으로 분석된다.

표 4. 추정된 센서 노드의 위치와 표준편차

노드	(a)				(b)				(c)			
	X	Y	표준편차		X	Y	표준편차		X	Y	표준편차	
1	0.01	0.02	0.007	0.014	-0.04	-0.02	0.028	0.014	-0.02	0.03	0.014	0.021
2	2.10	-0.08	0.071	0.057	2.28	0.03	0.028	0.021	2.36	0.07	0.085	0.049
3	0.95	3.06	0.035	0.042	2.27	2.29	0.028	0.035	-0.53	3.24	0.064	0.078
4	4.16	4.61	0.113	0.078	5.67	2.31	0.064	0.035	1.67	1.55	0.057	0.064
5	4.12	-1.09	0.085	0.064	3.24	-2.75	0.071	0.057	3.97	0.79	0.064	0.255
6	6.04	-3.12	0.028	0.085	3.94	-5.27	0.071	0.057	6.79	0.09	0.057	0.078
7	2.08	-4.05	0.057	0.035	0.09	-4.38	0.049	0.064	3.65	-2.61	0.057	0.064
8	0.06	-2.92	0.042	0.057	-1.46	-2.57	0.092	0.085	1.42	-2.54	0.064	0.106
9	8.13	-2.18	0.092	0.127	6.15	-5.21	0.092	0.092	7.91	1.67	0.106	0.064
10	9.15	-5.13	0.106	0.092	5.87	-8.41	0.021	0.049	10.36	-0.34	0.057	0.099
11	7.02	0.61	0.014	0.078	6.40	-2.76	0.064	0.071	6.12	3.68	0.049	0.085
12	6.17	2.08	0.120	0.057	6.35	-1.06	0.064	0.134	4.35	4.36	0.099	0.071
13	6.09	6.14	0.064	0.099	7.96	2.67	0.057	0.028	2.82	8.14	0.078	0.071
14	7.13	4.07	0.092	0.049	8.13	0.59	0.057	0.078	4.61	6.75	0.078	0.042

## 6. 결 론

센서 네트워크 시스템은 정적인 시스템이 아니라 매 시간 변화하는 동적인 시스템이다. 그리고 센서 노드의 위치에 따라서 수집되는 정보의 가치가 결정되기 때문에 센서 노드의 위치를 찾는 것은 중요하다.

본 논문에서는 센서 네트워크에서 이미 위치를 알고 있는 앵커 노드를 기준으로 위치를 알지 못하는 미지 노드들의 위치에 대해서 노드에서 수신하는 신호의 세기를 이용해서 계산된 거리와 센서 노드의 인접성을 사용하여 위치를 추정하는 기법을 제안하였다. 그리고 동적인 시스템인 센서 네트워크에서 위치 추정 시스템을 분석하기 위해서 UML의 유즈 케이스 다이어그램을 이용해서 액터와 각 기능을 분석하였으며, 액티비티 다이어그램을 이용하여 제안 기법의 동적인 흐름을 분석하였고, 상태 머신 다이어그램을 제안 기법의 상태를 정의하고 상태 간의 흐름 그리고 서버 머신 상태에 대한 분석하였다.

본 논문에서 제안한 위치 추정 기법은 GPS 장비를 이용하지 않고도 센서 노드의 인접성 정보와 수신 신호 세기만을 이용해서 센서 네트워크의 전체적인 분포를 확인할 수 있다. 그리고 노드의 위치가 이동하여 노드의 분포가 변화하는 경우에도 쉽게 노드의 위치를 앵커노드 중심으로 상대적으로 추정해낼 수 있다. 기존의 기법에서는 최소 2개 이상의 참조 노드

를 이용해서 위치를 찾아내야 했다. 그렇지만 본 논문에서 제안하는 기법은 앵커 노드 하나의 위치만 알아도 센서 노드의 전체적인 분포를 확인할 수 있음을 확인하였다. 또한 UML을 이용해서 제안 기법을 분석 및 설계를 하였기 때문에 제안 기법의 각 기능에 대한 흐름을 정확하게 파악할 수 있고, 이를 쉽게 구현할 수 있다. 그리고 시뮬레이션 결과를 보면 설계된 시스템이 아주 작은 오차를 가지고 센서 노드의 위치를 찾아내고 있음을 확인할 수 있다.

그러나 본 논문에서는 제안하는 기법은 모든 센서 노드가 평면상에 있다고 가정을 하였다. 3차원적인 환경에서는 제안한 기법이 오류를 나타낼 수 있다. 그리고 노드 간의 인접 정보가 부족할 경우 즉, 단독으로 연결된 노드의 위치를 추정하는데 문제가 발생할 수 있다. 향후에는 센서 네트워크에서 이동하는 노드에 대한 노드의 위치를 추정 및 예측하는 방법과 수신하는 신호의 세기에 대한 필터링을 통해서 신호 세기에 의해 거리를 정확하게 계산하여 노드의 위치를 추정할 수 있는 방법에 대한 연구가 필요로 할 것이다.

## 참 고 문 헌

[1] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American*, pp.

- 94-103, 1991.
- [2] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Communications*, Vol. 8, Issue 4, pp. 10-17, 2001.
- [3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Elsevier Computer Networks*, Vol. 38, Issue 4, pp. 393-422, 2002.
- [4] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing(LAR) in Mobile Ad Hoc Networks," *MOBICOM 98*, pp. 66-75, 1998.
- [5] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M.B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," *INFOCOM 2001*, Vol. 3, pp. 1380-1387 2001.
- [6] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," *INFOCOM 2000*, pp. 775-784, 2000.
- [7] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less Low-cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, Vol. 7, Issue 5, pp. 28-34, 2000.
- [8] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *IEEE Telecommunication Systems*, Vol. 1, pp. 267-280, 2003.
- [9] Object Management Group, "OMG UML Infrastructure," 2009, <http://www.omg.org/spec/UML/2.2/Infrastructure>
- [10] R.M. Bastos and D.D.A. Ruiz, "Extending UML activity diagram for workflow modeling in production systems," *HICSS 2002*, pp. 3786-3795, 2002.
- [11] T. He, C. Huang, B.M. Blum, J.A. Stankovic and T. Abdelzاهر, "Range-free Localization Schemes for Large Scale Sensor Networks," *MOBICOM 2003*, pp. 81-95, 2003.
- [12] D. Niculescu and B. Nath, "Ad hoc Positioning System (APS)," *IEEE Global Telecommunications Conference*, Vol. 5, pp 2926-2931, 2001.
- [13] G.P. Yost and S. Panchapakesan, "Improvement in Estimation of Time of Arrival (TOA) from timing advance (TA)," *ICUPC 98*, Vol. 2, pp. 1367-1372, 1998.
- [14] L. Zhu and J. Zhu, "A New Model and Its Performance for TDOA Estimation," *VTS2001*, Vol. 4, pp. 2750-2753, 2001.
- [15] D. Niculescu and B. Nath, "Ad hoc Positioning System (APS) using AOA," *INFOCOM 2003*, pp. 1734-1743, 2003.
- [16] A.J. Ramirez and B.H.C. Cheng, "Verifying and Analyzing Adaptive Logic Through UML State Models," *ICST 2008*, pp. 529-532, 2008.
- [17] C.A. Balanis, *Antenna Theory : Analysis and Design*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
- [18] V.K. Garg, *Wireless Communications and Networking*, Morgan Kaufmann Publishers, San Francisco, 2007.
- [19] K. Aamodt, "CC2431 Location Engine," *Application note AN042 Texas Instruments*, 2007.



임철우

1999년 2월 제주대학교 통신공학과 공학사  
 2002년 2월 제주대학교 통신공학과 공학석사  
 2009년 8월 영남대학교 컴퓨터공학과 공학박사

관심분야 : 소프트웨어공학, UML, Mobile Ad-hoc Network, Ubiquitous Sensor Network



김 영 락

2000년 2월 금오공과대학교 응용  
수학과 이학사  
2004년 2월 영남대학교 컴퓨터공  
학과 공학석사  
2009년 2월 영남대학교 컴퓨터공  
학과 공학박사

관심분야 : Mobile Ad-hoc Network, Ubiquitous Sensor  
Network, 무선 모바일 네트워크



강 병 옥

1970년 2월 영남대학교 전기공  
학과 공학사  
1977년 2월 영남대학교 전자공  
학과 공학석사  
1994년 2월 경북대학교 전자공  
학과 공학박사  
1973년~1976년 영남대학교 전자  
계산연구소(SE)

1977년~1978년 영남전문대학 전자과 전임강사

1979년~현재 영남대학교 전자정보공학부 컴퓨터공학  
전공 교수

관심분야 : 소프트웨어공학, UML, OCL, Ubiquitous Sensor  
Network