http://dx.doi.org/10.14400/JDC.2016.14.8.261

센서 네트워크를 위한 Radical line을 기반으로 한 센서 노드간의 Range-free 지역화 알고리즘

신봉희*, 전혜경**
인천대학교 컴퓨터공학부*, 인하대학교 컴퓨터정보공학과**

Range-free localization algorithm between sensor nodes based on the Radical Line for Sensor Networks

Bong Hi, Shin*, Hye Kyoung, Jeon**
Dept. of Computer Science & Engineering, Incheon National Univ.*
Dept. of Computer Science & Information Technology, Inha Univ.**

요 약 본 논문에서는 센서 네트워크를 위한 Radical line을 기반으로 한 센서 노드간의 Range-free 지역화 알고리즘에 대해 연구한다. 무선 센서 네트워크에서 라우팅 기법은 센서 네트워크의 전체적인 에너지 소모량을 감소시키거나 모든 센서 노드들의 균등한 에너지 소비를 유도해야 한다. 특히 데이터가 전송할 데이터의 양이 많아지면 에너지 소모가 심해지는데 이를 극복하기 위한 새로운 방식들이 제안되었다. 그 결과 전체적인 에너지 소모량을 균등하게 조절할 수 있게 되었다. 이를 위해 논문에서도 적은 연산으로 주변 노드의 위치정보를 획득할 수 있는 지역화 알고리즘을 설계한다. 알고리즘의 연산을 위해 Radical Line을 적용한다. 실험환경은 운영체제는 윈도우 7, 플랫폼은 Visual C++ 2010으로 실험하였다. 실험결과 0.1837의 에러율로 지역화를 수행할 수 있었다.

주제어 : 센서 네트워크, IOT, 라우팅, Radical Line, 지역화, 융합

Abstract In this paper, we studied the range-free localization algorithm between sensor nodes based on the Radical Line for sensor networks. Routing in wireless sensor networks should reduce the overall energy consumption of the sensor network, or induce equivalent energy consumption of all the sensor nodes. In particular, when the amount of data to send more data, the energy consumption becomes worse. New methods have been proposed to address this. So as to allow evenly control the overall energy consumption. For this, the paper covers designing a localization algorithm that can obtain the location information of the peripheral nodes with fewer operations. For the operation of the algorithm is applicable Radical Line. The experimental environment is windows 7, the Visual C ++ 2010, MSSQL 2008. The experimental results could be localized to perform an error rate of 0.1837.

Key Words : Sensor Network, IOT, Routing, Radical Line, Localization, Convergence

* 본 논문은 인천대학교 2015년도 자체연구비 지원에 의하여 연구되었음
Received 2 July 2016, Revised 28 July 2016
Accepted 20 August 2016, Published 28 August 2016
Corresponding Author: Hye-Kyoung Jeon(In ha University)
Email: jhk7010@nate.com

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

미래의 기기들은 인간들이 생성하고 가공한 데이터, 정보 및 지식에서 뿐만 아니라 주변의 센싱과 액츄에이터 기능을 갖는 사물로부터 정보를 획득할 수 있다. 그리고 그 기기들은 자체 식별자를 지니고 연산과 통신 기능을 통해 외부 세계와 상호 작용 할 수 있을 것으로 예측된다. 이를 지원해 주는 개념을 사물인터넷(Internet Of Things)라 하며 사물인터넷은 인간과 사물, 사물과 사물, 사물과 서비스 사이에서 인간의 개입 없이 상호 협력적으로 센싱, 네트워킹, 정보처리 등의 스마트적인 관계를 형성하는 사물 공간 연결망으로 정의되고 있다. 이 연결망 내에서 정보의 공유가 가능해 지면 인간은 높은 수준의 물리적 환경 정보를 얻을 수 있으며 이를 토대로 자신의 행동에 대해 보다 최적화된 선택을 할 수 있게 된다[1,2,3,4,5,6].

이러한 사물 인터넷은 여러 가지 환경으로 구성이 되는데 그중 무선 센서 네트워크에서 라우팅 기법은 센서 네트워크의 전체적인 에너지 소모량을 감소시키거나 모든 센서 노드들의 균등한 에너지 소비를 유도해야 한다. 특히 데이터가 전송할 데이터의 양이 많아져 에너지 소모가 심해지는데 이를 극복하기 위한 새로운 방식들이 제안되어 전체적인 에너지 소모량을 균등하게 조절할 수 있다[7]. 센서 네트워킹 기술은 네트워크 구조와 라우팅 동작에 의하여 평면 라우팅 알고리즘, 계층적 라우팅 알고리즘, 위치기반 알고리즘, 실시간 방식 라우팅 알고리즘등이 있다[8]. 본 연구를 이용해서 사용하려고 하는 라우팅은 이 중 위치 기반 라우팅 알고리즘이다. 센서들 간의 라우팅을 위한 센서들의 최적의 수를 알아내어 지역화를 통한 라우팅으로 저전력 및 경로 설정 문제를 해결하려고 한다.

본 연구에서는 센서 노드간의 효율적인 라우팅을 위한 방법으로 Radical Line을 이용하여 센서 노드들 간의 지역화를 위한 알고리즘을 설계하는데 목표를 둔다.

2. 연구배경

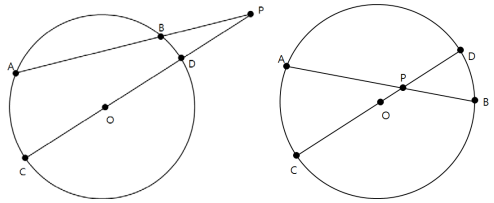
2.1 위치기반 라우팅 프로토콜

위치기반 라우팅 프로토콜은 부가적인 지리 위치정보를 이용하여 목적지 노드의 위치로 가는 경로를 계산하여 패킷을 전송하는 방식이다. 이러한 라우팅 프로토콜은 전송 패킷의 헤더에 포함된 목적지 노드의 위치정보와 1-Hop 범 내에 있는 이웃노드들의 위치정보만을 이용하여 경로결정을 한다. 따라서 각 노드들은 자신의 위치정보를 필요로 하며, 위치정보는 GPS, Active Badges, Cricket[9] 등의 위치 센싱 기법들을 이용하여 획득할 수 있다. 위치기반 라우팅 프로토콜의 특징은 다음과 같다. 첫째, 지역적인(localized) 정보를 이용하여 경로를 선정하고 오버 헤드를 줄일 수 있다. 위치정보를 이용하는 라우팅 프로토콜들은 일반으로 경로의 설정과 유지과정이 필요하지 않고 소스 노드가 주변 노드들의 위치정보를 이용하여 목지 노드의 위치로 가는 경로를 계산하여 패킷을 포워딩(forwarding)한다. 둘째, 확장성이 용이하다. 전체 네트워크의 토폴로지에 대한 정보와 유지과정이 필요하지 않기 때문에 네트워크의 크기에 구속받지 않는다. 위치기반 라우 프로토콜의 어려움 중의 하나가 목적지 노드와 주변 노드의 정확한 위치정보를 얻는 것이다. 목적지 노드와 주변 노드의 정확한 위치정보가 없으면 라우팅 패킷은 목적지 노드에 제대로 전달할 수 없다. 따라서 위치기반 라우팅을 수행하기 위해 선행으로 위치 서비스와 비콘 서비스가 제공되어야 한다. 위치 서비스는 위치기반 라우팅을 수행하기 위해 필요로 하는 목적지 노드의 위치정보를 제공하게 된다. 목적지 노드의 위치정보를 알게 되면 소스 노드는 자신과 주변 이웃 노드의 위치정보를 기반으로 라우팅 경로를 설정하게 된다. 비콘 서비스는 주변 이웃 노드의 위치정보를 제공 하는 기능으로써, 이는 주기적으로 자신의 ID, 치 등 정보를 포함하는 비콘 메시지를 주변에 브로드 캐스트 한다. 결과적으로 위치기반 라우팅은 위치서비스와 비콘 서비스를 사용함에 따라 발생하는 오버헤드, 위치 서비스에 따른 전송 지연 및 위치 정보의 정확도에 따른 라우팅 실패 등 불리한 요소들을 갖는다[10].

2.2 Radical Line

두 유향 성분의 곱을 방멱(power)라고 한다. 방멱에 대한 정의는 다음과 같이 정의된다.

정의 1. 어느 점 P와 반지름이 r인 원 O가 주어져 있다고 하자. 여기서 원 O에 대한 P의 방멱은 $OP^2 - r^2$ 이다[14,15].



[Fig. 1] The Power of Circle

[Fig. 1]에서 왼쪽은, 선분 P를 지나 선분 A, B에서 만나도록 직선을 그리면 유향선분 PA와, PB는 같은 방향이다. 따라서 $PA \times PB > 0$ 이어야 한다. 선분 OP와 원의 교점을 C, D라고 했을 때 원의 성질에 따라 $PA \times PB = PC \times PD = (OP + r)(OP - r) = OP^2 - r^2$ 이 된다. 한편 오른쪽의 경우, 유향성분 PA와 PB의 방향이 다르기 때문에 $PA \times PB < 0$ 이 되고, 원의 성질에 따라 $PA \times PB = PC \times PD = (OP + r)(OP - r) = r^2 - OP^2$ 이 된다. 방멱(Power)은 $f(x,y) = (x-a)^2 + (y-b)^2 - r^2 = 0$ 과 점 $P(x',y')$에 대해 $f(x,y) = (x'-a)^2 + (y'-b)^2 - r^2$ 로 표현할 수 있다. 여기서 P의 원 $f(x,y) = 0$ 에 대한 방멱이 원 $g(x,y) = (x-a')^2 + (y-b') - r'^2 = 0$ 에 대한 방멱과 같다고 가정하면, 아래 식을 도출한다.

$$(x'-a)^2 + (y'-b)^2 - r^2 = (x-a')^2 + (y-b') - r'^2$$
$$\Leftrightarrow$$
$$[(x'-a)^2 + (y'-b)^2 - r^2] - [(x-a')^2 + (y-b')^2 - r'^2 = 0 \quad (1)$$

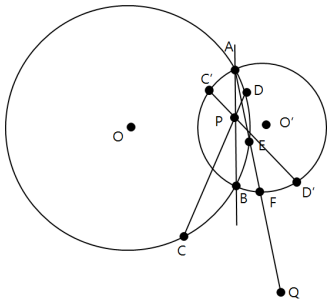
결국 두 원의 방정식을 뺏을 때 얻어지는 직선은 두 원으로부터 방멱이 같은 점들의 집합이 된다. 그리고 이것을 근축(Radical Axis, Radical Line)이라고 정의한다[12].

정의 2. 동심원이 아닌 두 원에 대해 방멱(Power)이 같은 점들의 집합을 근축이라고 정의한다.

위 직선식은 $(a_1 - a_2)x + (b_1 - b_2)y = k$ 형태이다 그리고 두 원의 중심을 이은 직선의 방향 벡터는 $(b_1 - b_2, -(a_1 - a_2))$ 이다. 따라서 원의 방정식의 차를 통해 얻은 직선은 원의 중심끼리 이은 선분에 수직이다. 그러므로 아래와 같이 정리할 수 있다.

정리 1. 근축은 두 원의 중심을 잇는 선분에 수직인 직선이다[13].

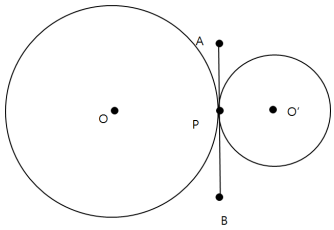
[Fig. 2]에서처럼 두 원의 두 점이 만나는 경우는 아래와 같다.



[Fig. 2] Intersections of Two Circle

두 원이 O, O'의 교점 A, B를 지나는 직선 위에 있는 점들은 모두 두 원에 대한 방멱이 같다. 그림에서처럼 직선 위에 임의의 점 P를 잡고, O와는 C, D에서 만나고 O'와는 C' D'에서 만난다고 하면 원의 성질에 따라 $PC \times PC = PA \times PB = PC' \times PD'$ 이다. 그리고 P가 원 O의 안에 있으면 O'의 안에도 있고 원 O의 밖에 있으면 O'의 밖에 있다. 따라서 직선 AB의 모든 점에서 O에 대한 방멱과 O'에 대한 방멱은 같다.

[Fig. 3]에서 볼 수 있듯이 두 원이 한 점에서 만나는 경우는 아래 그림처럼 직선 AB만 공통 접선으로 바꿔주면 된다.



[Fig. 3] Common Intersection of Two Circle

3. RFLAbasedRaL

본 논문에서는 위치기반 라우팅 프로토콜에 Radical Line을 적용한 알고리즘을 제안 및 실험한다. 위치기반 라우팅을 적용할 센서 네트워크는 앵커로부터 수신된 비콘 신호에 따라 앵커 노드가 전송 범위 내에 있는지를 결

정한다. 앵커는 자신으로부터 R 미터 범위 내에 있는 센서와 통신할 수 있다 라고 가정한다. 왜냐하면 실제 환경에서는 전파 조건에 좌우되기 때문이다. 적용 범위 내에서 변화량의 측정은 불규칙의 정도(DOI - Degree Of Irregularity)로 삼는다.. DOI는 전파 방향에 따라 변화하는 장치 당 최대 범위 변화량을 나타낸다. [11]에서는 앵커와 센서의 성공적인 연결에 요구되는 조건을 보였다. a_i로부터 센서에 전송된 전력량을 $W(a_i)$로 하고 Q는 확인되지 않은 노이즈의 양으로 하고 S는 무선 센서 네트워크에서 추론된 전력량이다.

$$\frac{W(a_i)}{Q+S} \geq TH \tag{2}$$

위 수식에서 TH 는 하드웨어에 의존하는 임계치이다.

앵커에 인접한 센서의 위치를 알아내기 위해 앵커 및 센서의 전파 반경을 Radical Line에 적용한다. 센서 P가 알려지지 않은 위치 $P=[x, y]^T$ 에 있다고 하면 그 센서의 위치는 알려진 위치 $a_i=[x_i, y_i]^T$를 갖는 N개의 앵커 a_i와 연결되어 있고 Radio 범위 R_i를 갖는다. 그러므로 센서 P는 반드시 N개의 원의 RI 내에 있게 된다. N에 따라서 고려해야 할 사항들이 몇 가지 있다.

3.1 $N > 3$

N개의 원에는 $\frac{N!}{2!(N-2)!}$개의 RL이 있다. 연산량을 줄이기 위해서, 알고리즘은 N개의 원 사이의 거리 중 가장 큰 거리에 있는 중심을 갖는 두 원의 RL만을 선택한다. 이유는 그 두 원의 RL이 가장 작기 때문에 RL은 모든 N개의 원들의 RI에 있을 확률이 높기 때문이다.

$$d_{ij} = \| a_i - a_j \|^{1/2} = d_{ji},\ i, j = 1, \ldots, N \tag{3}$$

a_i와 a_j의 중점 사이의 거리를 $d_{ij}=d_{ji}$ 로 둔다. d_{qk}는 수식 (4)의 최대값이다. 간단하게 하기 위해, $q=1$, $k=2$ 로 한다. [Fig. 4]에서 보는 것처럼, RL의 끝점은 $I_a=[x_a, y_a]^T$, $I_b=[x_b, y_b]^T$ 로 하고, $O=[x_o, y_o]^T$ 는 RL과 a_1과 a_2를 연결하는 선 사이의 교차점이다. O와 a_1, a_2사이의 거리는 아래 식으로 구할 수 있다.

$$d_{o1} = \| O - a_1 \|^{1/2} = [(x_o - x_1) + (y_o - y_1)]^{1/2} \tag{4}$$

$$d_{o2} = \| O - a_2 \|^{1/2} = [(x_o - x_2) + (y_o - y_2)]^{1/2} \tag{5}$$

반경 R은 아래 수식을 통해 구할 수 있다.

$$d_{o1}^2 + m^2 = R_1^2 \tag{6}$$

$$d_{o2}^2 + m^2 = R_2^2 \tag{7}$$

수식 (7)에서(8)을 빼면 아래와 같은 수식을 얻을 수 있다.

$$2(x_2 - x_1)x_o + 2(y_2 - y_1)y_o = R_1^2 - R_2^2 + k_2 - k_1 \tag{8}$$

여기서

$$k_i = x_i^2 + y_i^2 \tag{9}$$

이 된다.

O와 a_1의 기울기와 a_1과 a_2 사이 선의 기울기는 같다. 그러므로 아래의 식을 산출할 수 있다.

$$\frac{y_2 - y_0}{x_2 - x_0} = \frac{y_2 - y_1}{x_2 - x_1} \tag{10}$$

또한 이 식은 아래와 같은 전개를 만들 수 있다.

$$(y_2 - y_1)x_0 - (x_2 - x_1)y_0 = x_2(y_2 - y_1) - y_2(x_2 - x_1) \tag{11}$$

(9)식과 (12)의 수식을 이용해 $O(x_0, y_0)$을 구한다. $d_{12}=D$ 치환하여 아래 식으로 정리해 보면

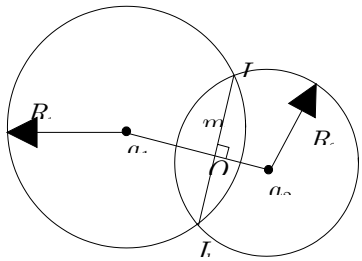
$$R^2 - d_{o1}^2 = m^2 = R^2 - (D - d_{o1})^2 \tag{12}$$

$$d_{o1} = \frac{R_1^2 - R_2^2 + D^2}{2D} \tag{13}$$

을 얻을 수 있고, 여기서

$$m = \sqrt{(R_1^2 - d_{o1}^2)} \tag{14}$$

를 얻을 수 있다.



[Fig. 4] The representation of coordinates associated with the RL

[Fig. 4]는

$$\frac{x_o - x_a}{m} = \frac{y_o - y_1}{d_{o1}} \tag{15}$$

와

$$\frac{y_o - y_a}{m} = \frac{x_o - x_1}{d_{o1}} \tag{16}$$

같은 삼각관계식을 내포하고 있다.

수식 (16)과 수식 (17)로부터 I_a을 위한 좌표를 아래와 같이 구할 수 있다.

$$x_a = x_o - \frac{m}{d_{o1}}(y_o - y_1) \tag{17}$$

$$y_a = y_o - \frac{m}{d_{o1}}(x_o - x_1) \tag{18}$$

I_b를 위한 좌표도 이와 같이 구할 수 있다.

$$x_b = x_o - \frac{m}{d_{o1}}(y_o - y_1) \tag{19}$$

$$y_b = y_o - \frac{m}{d_{o1}}(x_o - x_1) \tag{20}$$

다음으로, 알고리즘은 RL상에 있는 테스트 포인트 노드 $t_l = [x_l, y_l]^T, l = 1, ..., L$ 를 갖는 L을 선택한다. I_a와 I_b 사이에 일정한 증가를 통해 다음의 식을 얻는다.

$$x_l = x_a + \frac{l(x_b - x_a)}{L+1} \tag{21}$$

$$y_l = y_a + \frac{l(y_b - y_a)}{L+1} \tag{22}$$

L은 사용자 파라미터이다. 각 t_l에 있어 알고리즘은 t_l이 RI내에 있는지와 RI로부터 얼마나 떨어졌는지를 에러를 계산하여 검사한다.

$$\epsilon_{li} = \| t_l - a_i \|^{1/2} - R_i = \begin{cases} \epsilon_{li} & \text{if } \epsilon_{li} > 0 \\ 0 & \text{if } \epsilon_{li} \le 0 \end{cases} \tag{23}$$

그런 다음 주어진 모든 a_i 상에 에러를 더한다.

$$S_l = \sum_{i=1}^{N} \epsilon_{li} \tag{24}$$

만약 $S_l = 0$ 이면, 대응하는 t_l은 RI 내에 있고 P에 대한 측정치가 된다. 만약 $S_l > 0$ 이면 RL은 N개 원의 RI 안에 없다. 그렇기 때문에 CA의 에러를 계산할 필요가 있다.

$$S_c = \sum_{i=1}^{N} \epsilon_{ci} \tag{25}$$

ϵ_{ci}는 수식 (24)에서 도출된다. 또한 t_l을 대신해서 수식 (22)로부터 $c = [\hat{x}, \hat{y}]^T$ 을 갖는다. P에 대한 마지막 측정치 $\hat{P}$는 t_l 또는 c을 선택함으로써 생기는데 t_l과 c는 S_l과 S_c에 대응한다.

3.2 $N = 2$ 과 $N = 3$

$N = 2$ 일 때 $\hat{P}$은 $O(x_o, y_o)$와 같다. $N = 3$ 일 때 알고리즘은 세 RL들의 교차점을 계산한다. 교차점을 $I = [x_I, y_I]^T$ 라고 한다. [Fig. 4]를 세 개의 원으로 확장해서 $h^2 \le m^2$ 인 아래식의 결과를 만들어 낸다.

$$(x_I - x_i)^2 + (y_I - y_i)^2 + h^2 = R_i^2, \quad i = 1, 2, 3 \tag{26}$$

4. 실험 및 결과

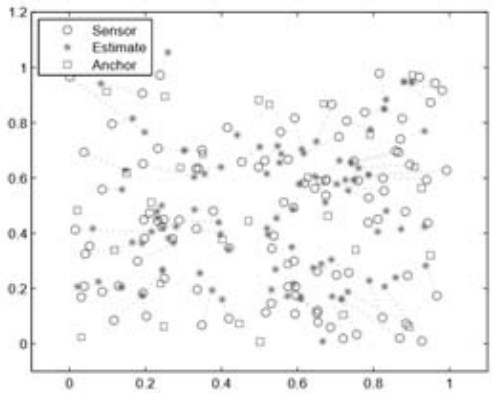
4.1 실험

실험을 위한 센서 네트워크 환경은 100m × 100m 의 영역을 갖고, 100개의 센서를 랜덤하게 배치하고 NA의 앵커는 모두 랜덤하며 초기 설정에서 위치를 알려져 있지 않다고 가정한다. 주변의 임의의 센서와 연결된 앵커의 수는 2개부터 NA개만큼 주어진다. 어떤 앵커는 $R=R_{\max}=45m$이고 어떤 앵커는 $R=0.5R_{\max}$이다. 지역화시에 발생하는 에러는 NA가 증가하는 것에 반비례한다. 100회의 독립시행동안, $R_{\max}$를 분모로 에러는 아래 식과 같다.

$$e(NA)=\frac{1}{100}\sum_{j=1}^{100}\left\{\frac{\sum_{i=1}^{100}\| p_j(i)-\hat{p}_j(j)\|^{1/2}}{100R_{\max}}\right\} \quad (27)$$

수식 (27)에서 $p_j(i)$는 j번째 시도에서 I번째 센서의 위치이고 $\hat{p}_j(i)$는 그 위치값이다. 실험사양은 운영체제는 윈도우 7, 플랫폼은 Visual C++ 2010, 데이터베이스는 MS SQL 2008로 실험하였다. 실험에서 $DOI=0$ 으로 한다. 앵커의 R_i 내에 있는 센서는 해당 앵커와 연결되어 있다. $DOI\neq 0$ 인 경우는 실제 R_i는 더 작다.

[Fig. 5]는 RFLAbasedRaL이 적용된 지역화모습과 에러율을 나타낸다.



[Fig. 5] Localization and Error of RFLAbasedRaL (DOI = 0.1, error = 0.1837)

4.2 결과

위치기반 라우팅을 기본으로 하는 범위가 자유로운 지역화는 범위기반인 만큼 정확하지는 않지만, 연산이 비교적 간결하다는 장점을 가지고 있다. 즉 TOA나 TDOA를 측정하기 위한 특별한 요구사항이 필요없다. 이것은 저전력 환경에서 동작하는 센서 네트워크에서는 무척 중요하다. 특정한 어플리케이션에서는 위치가 정확하지 않아도 수행될 수 있기 때문에 RFLAbasedRaL는 의미가 있다고 할 수 있다. 향후 무선 센서 네크워크 분야에서 활용도가 있을 것으로 생각된다.

ACKNOWLEDGMENTS

This work was supported by the Incheon National University Research Grant in 2015.

REFERENCES

[1] ITU Internet Reports, The Internet of Things, Nov. 2005.

[2] Charith Perer, Dimitrios Georgakopoulos, "Contextaware computing for the internet of things : A survey", IEEE Communications Surveys & Tutorials, 2013.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey", Computer Networks, Vol. 54, No. 15, pp. 2787-2805, 2010.

[4] Jeong-Ick Lee, "Convergent Case Study of Research and Education: Internet of Things Based Wireless Device Forming Research", Journal of the Korea Convergence Society, Vol. 6, No. 4, pp. 1-7, 2015.

[5] Onechul Na, Hyojik Lee, Soyoung Sung, Hangbae Chang, "A Study on Construction of Optimal Wireless Sensor System for Enhancing Organization Security Level on Industry Convergence Environment", Journal of the Korea Convergence Society, Vol. 6, No. 4, pp. 139-146, 2015.

[6] Yong-Wook Nam, Yong-Hyuk Kim, "Speed estimation of sound-emitted objects through convergence of sound information analysis and smart device technology", Journal of the Korea Convergence Society, Vol. 6, No. 5, pp. 233-240, 2015.

[7] Bong Soo Kim, So Young Hwang, Hoon Jung, Sung Soon Joo, “Ubiquitous Sensor Network Routing Technology”, Review of Korean Society for Internet Information Vol. 9, No. 2, pp 13-21, 2008.

[8] P. Biswas, T.-C. Lian, T.-C.Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. TOSN, Vol. 2, No. 2, pp 188 - 220, 2006.

[9] J. Hightower and G. Borriello, “Location Systems for Ubiquitous Computing,” IEEE Computer, Vol. 8, No. 34, pp. 57 - 66, 2001.

[10] Youngsoo Park, Jaeil Jung, "Performance Analysis of Position Based Routing Protocol for UAV Networks", The Journal of The Korean Institute of Communication Sciences, Vol. 37, No. 2, pp 188-195, 2012.

[11] http://mathworld.wolfram.com/CirclePower.html

[12] http://mathworld.wolfram.com/RadicalLine.html

[13] http://mathworld.wolfram.com/ChordalTheorem.html

[14] T. Moscibroda, R. Wattenhofer, and A. Zollinger. “Topology control meets sinr: The scheduling complexity of arbitrary topologies”. In MobiHoc, pp 310-321, 2006.

[15] Young-Ho Kim, Joong-Soo Lim, Gyoo-Soo Chae, Kichul Kim, “An investigation of the Azimuth Error for Correlative Interferometer Direction Finding”, Journal of the Korea Convergence Society, Vol. 6, No. 5, pp. 249-255, 2015.

신 봉 희(Shin, Bong Hi)

- 1977년 2월 : 인하대학교 전자공학과 공학사
- 1981년 2월 : 인하대학교 전자공학과 공학석사
- 1995년 2월 : 단국대학교 전자공학과 공학박사
- 2010 ~ 현재 인천대학교 컴퓨터공학부 교수
- 마이크로프로세서, 임베디드시스템 등
- E-Mail : bhshin@incheon.ac.kr

전 혜 경(Jeon, Hye Kyoung)

- 1995년 2월 : 인하대학교 일문과(문학사)
- 1999년 8월 : 인하대학교 정보공학과(공학석사)
- 2002년 9월 ~ 2005년 8월 : 인하대학교 컴퓨터정보학과 박사수료
- 2009년 4월 ~ 2015년 2월 : 이스트림 선임연구원
- 2016년 2월 ~ 현재 : YM나을텍 선임연구원
- 관심분야 : 상황인식, 센서 네트워크, 유비 쿼터스, 사물인터넷
- E-Mail : jhk7010@nate.com