International Journal of Hybrid Information Technology
Vol.8, No.5 (2015), pp.1-10
http://dx.doi.org/10.14257/ijhit.2015.8.5.01

Position Location Scheme Using Nonlinear Programming Based on RSSI and DV-Hop

Fengjun Shang, Lan Lan and Maojian Dong

College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
E-mail: shangfj@cqupt.edu.cn

Abstract

Localization is used in location-aware applications such as navigation, autonomous robotic movement, and asset tracking to position a moving object on a coordinate system. In this paper, a Nonlinear Programming algorithm is proposed based on RSSI and improved DV-Hop algorithm, called NPRDV-Hop. The algorithm makes four major contributions to the localization problem in the wireless sensor networks (WSNs). Firstly, a hop distance is improved, called Hop Distance. This scheme could assure that the most nodes receive the Hop Distance from beacon node who has the least hops between them. This practical localization scheme is relatively high accuracy and low cost for WSNs. Secondly, Heron's formula is introduced as objective function. Thirdly, Gauss distribution is introduced to select RSSI so that the error of distance is little. Lastly, the general problem is considered by the nonlinear programming to solve for the locations of the sensors. Simulation results show that the proposed method can improve location accuracy and coverage without increasing hardware cost of sensor node. The performance of this algorithm is superior to the original DV-Hop algorithm.

Keywords: *WSN, DV-Hop, RSSI, Nonlinear Programming, Location Accuracy*

1. Introduction

With the development of sensor techniques, low-power electronic and radio techniques, low-power and inexpensive wireless sensors have been put into application, then the wireless sensor networks have appeared. WSNs can be applied to many areas, such as military affairs, commerce, medical care, environmental monitoring, and have become a new research focus in computer and communication fields. Many applications of WSNs are based on sensor self-positioning, such as battlefield surveillance, environment monitoring, indoor user tracking and others, which depend on knowing the location of sensor nodes. Because of the constraint in size, power, and cost of sensor nodes, the investigation of efficient location algorithms which satisfy the basic accuracy requirement for WSNs meets new challenges.

Many localization algorithms for sensor networks have been proposed to provide per-node location information. Based on the type of knowledge used in localization, we divide these localization protocols into two categories: range-based and range-free. Range-based protocols use absolute point-to-point distance or angle information to calculate the location between neighboring sensors. The second class of methods, range-free approach, employs to find the distances from the non-anchor nodes to the anchor nodes. Several ranging techniques are possible for range measurement, such as angle-of-arrival [1], received signal strength indicator (RSSI) [2], time-of-arrival [3] or time-difference-of-arrival [4].Because of the advantages on power and cost on sensor node, this paper focuses the investigation on the range-free algorithms for WSNs [5]. Centroid algorithm [6] is a simple range-free

ISSN: 1738-9968 IJHIT

localization algorithm. The node receives signals of landmarks in its communication area and makes its coordinates as the centroid of these landmarks. A new algorithm which is based on the sequence-based algorithm and the three orthocenter method is proposed, called SATOM [7]. The new algorithm will bring some computing increases. However, it does not need additional improvement in hardware or complexity of nodes.

This paper makes four major contributions to the localization problem in WSNs. Firstly, a hop distance is improved. Secondly, Heron's formula is introduced as objective function. Thirdly, Gauss distribution is introduced to select RSSI so that the error of distance is little. Lastly, the general problem is considered by the nonlinear programming to solve for the locations of the sensors. Furthermore, it explored the influence of anchor nodes on localization performance of the NPRDV-Hop algorithm.

The rest of this paper is organized as follows. Section 2 presents the NPRDV-Hop Location Scheme. In Section 3, simulation results are shown and localization performances are discussed. Finally, the conclusions are presented in Section 4.

2. NPRDV-Hop Location Scheme

In this paper, we present a nonlinear programming algorithm based on RSSI and improved DV-Hop algorithm. The proposed method can improve location accuracy without increasing hardware cost of sensor node.

2.1. Improved DV-Hop Algorithm

Niculescu and Nath [8] have proposed the DV-Hop, which is a distributed, hop by hop positioning algorithm. The algorithm implementation is comprised of three steps. First, it employs a classical distance vector exchange so that all nodes in the network get distances, in hops, to the landmarks. And then, it estimates an average size for one hop, which is then deployed as a correction to the entire network. Finally, unknown nodes compute their location by trilateration [9].

In the first step, each anchor node broadcasts a beacon to be flooded throughout the network containing the anchors location with a hop-count value initialized to one. Each receiving node maintains the minimum hop-count value per anchor of all beacons it receives. Beacons with higher hop-count values to a particular anchor are defined as stale information and will be ignored. Then those not stale beacons are flooded outward with hop-count values incremented at every intermediate hop. Through this mechanism, all nodes in the network get the minimal hop-count to every anchor node.

In the second step, once an anchor gets hop-count value to other anchors, it estimates an average size for one hop, which is then flooded to the entire network. After receiving hop-size, blindfolded nodes multiply the hop-size by the hop-count value to derive the physical distance to the anchor. The average hop-size is estimated by anchor *i* using the following formula.

$$HopDistance = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} H_{ij}} \qquad (1)$$

where (x_i, y_i), (x_j, y_j) are coordinates of anchor *i* and anchor *j*, H_{ij} is the hops between beacon *i* and beacon *j*.

In the third phase, the unknown node locations can be estimated by the multi-alteration method when these nodes have the distance estimations to at least three reference nodes in the plane. Given a set of reference nodes $R_i = (x_i, y_i)^T$, *i*=1, 2… M, where M is the number of reference nodes, let the hop value between the unknown node $X = (x, y)^T$ and the *i*-th reference node is L_i. Then the distance between the unknown node and *i*-th reference node is given by $d_i = L_i \times HopDistance$. The unknown node location X can be obtained as follows.

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ \quad\quad\quad\quad \cdots \\ (x_M - x)^2 + (y_M - y)^2 = d_M^2 \end{cases} \tag{2}$$

In the above data structure, (x_i, y_i) are the two-dimensional coordinates of the *i*-th reference point, (x, y) are the coordinates of unknown node, and d_i is the measured ranged between the *i*-th reference point and the unknown. This data structure can be linearized by subtracting the last row and performing some minor arithmetic shuffling, resulting in the following relations [10]:

$$x_i^2 + y_i^2 - 2x_i x - 2y_i y + x^2 + y^2 = d_i^2 \Rightarrow d_i^2 - M_i = -2x_i x - 2y_i y + K \tag{3}$$

where $M_i = x_i^2 + y_i^2$, $K = x^2 + y^2$.

Let $Z = [x, y, K]^T$, $G = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \cdots & \cdots & \cdots \\ -2x_i & -2y_i & 1 \end{bmatrix}$ and $h = \begin{bmatrix} d_1^2 - M_1 \\ d_2^2 - M_2 \\ \cdots \\ d_i^2 - M_i \end{bmatrix}$.

By (3), we can have

$$GZ = h \tag{4}$$

2.2. Improved Hop Distance

The difference between estimated and actual distances, denoted by $e^{i,j}$, is expressed as:

$$e^{A,B} = d_{est}^{A,B} - d_{true}^{A,B}$$
$$e^{A,C} = d_{est}^{A,C} - d_{true}^{A,C}$$
$$e^{C,B} = d_{est}^{C,B} - d_{true}^{C,B}$$

We use the differential error $e^{i,j}$ as correction factor of original *Hop Distance* estimation represented by *Hop Distance* in Equation (1). The effective average *Hop Distance*, between anchor node *i* and *j* is defined as:

$$HopDistance^{i,j} = HopDistance - \frac{\frac{e^{A,B}}{hop_{AB}} + \frac{e^{A,C}}{hop_{AC}} + \frac{e^{B,C}}{hop_{BC}}}{\frac{1}{hop_{AB}} + \frac{1}{hop_{AC}} + \frac{1}{hop_{BC}}}$$

Each anchor node broadcasts its *Hop Distance* to network using controlled flooding. Unknown nodes receive *Hop Distance* information, and save the first one. At the same time, they transmit the *Hop Distance* to their neighbor nodes. This scheme could assure that the most nodes receive the *Hop Distance* from beacon node who has the least hops between them.

In the end of this step, unknown nodes compute the distance to the beacon nodes based hop-length and hops to the beacon nodes.

In the DV-Hop algorithm, the unknown node localization is completely depend on its distances to at least three reference nodes, which are determined by *Hop Distance* and hop count values. When an unknown node X holds the hop value, equal to 1, to a reference node R_i, it should be noted that the distance between X and R_i must be less than the node communication range, that is $\|X - R_i\| \le D$, where $\|\bullet\|$ is the Euclidean distance, D is radio range. Similarly, if the X holds the hop value equal to 2, then $\|X - R_i\| \le 2 \bullet D$.

We improve the DV-Hop algorithm by using above observation.

For presentation simplicity but without loss of generality, we just discuss the localization of one of the unknown nodes denoted as $P = (x, y)^T$. For the reference node i, the hop value between P and reference node i is L_i and the average single hop distance is *Hop Distance*, i=1, 2… M and M is the number of the reference nodes. Then the distance between P and the i-th reference node is $d_i = L_i \times HopDistance$. Denote these reference nodes as A, B, C.

In order to estimate the distance, measuring hop-count is used just like DV-Hop. Each of the anchor nodes launches the DV-Hop algorithm by initiating a broadcast containing its known location and a hop count of 0. All of the one-hop neighbors surrounding the anchor hear this broadcast, record the anchor's position and a hop count of 1, and then perform another broadcast containing the anchor's position and a hop count of 1. Every node that hears this broadcast and did not hear the previous broadcasts will record the anchor's position and a hop count of 2 and then rebroadcast. This process continues until each anchor's position and an associated hop count value have been spread to every node in the network. It is important that nodes receiving these broadcasts search for the smallest number of hops to each anchor. This ensures conformity with the model used to estimate the average distance of a hop, and it also greatly reduces network traffic. One model for estimating the average hop distance between nodes for the entire network is to simply use the maximum radio range of each node. This simplistic approach is sufficient to generate satisfactory position results, and saves on communication costs relative to more complicated models [8]. The details are shown in Figure 1.

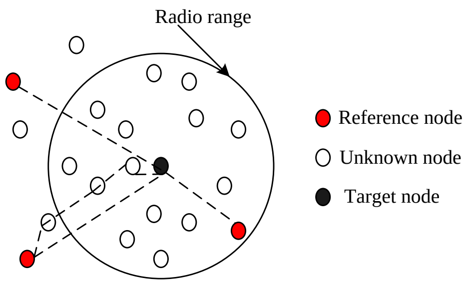

Figure 1. Illustration of Calculating Hop-Count

Once a node has received data regarding at least three (four) anchor nodes for a network existing in a two (three)-dimensional space, it is able to perform a NPRDV-Hop to estimate its location. If this node subsequently receives new data after already having performed a NPRDV-Hop, either a smaller hop count or a new anchor, the node simply performs another NPRDV-Hop to include the new data. This procedure is summarized in the following piece of pseudo code:

when a positioning packet is received,
if new anchor **or** lower hop count **then**
    store (hop count + 1) for this anchor.
    compute estimated range to this anchor.
    broadcast new packet for this anchor.
else
    do nothing.
if number of anchors≥(dimension of space + 1) **then**

NPRDV-Hop.
else
do nothing.

2.3. Introducing Objective Function

Assumed the triangle with sides *a*, *b*, and *c*, Heron's formula states that the area *(Area)* of a triangle whose sides have lengths *a*, *b*, and *c* is

$$Area = \sqrt{s(s-a)(s-b)(s-c)} \tag{5}$$

where *s* is the semiperimeter of the triangle:

$$s = \frac{a+b+c}{2} \tag{6}$$

In order to explain our idea, we introduce an example shown in Figure 2. Figure 2 shows an illustrative sensor network, where nodes A, B and C are the reference nodes and the node P is unknown.

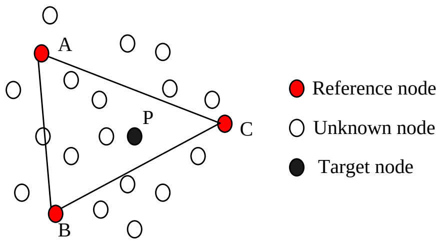

Figure 2. Illustration of our Idea

According to Figure 2, we have

$$Area_{ABC} = Area_{APC} + Area_{BPC} + Area_{APB} \tag{7}$$

When unknown node P is outside triangle ABC, there is a similar conclusion, so Equation (7) can be considered as an objective function.

Equation (7) can also be written as:

$$t = (Area_{ABC} - Area_{APC} - Area_{BPC} - Area_{APB})^2 \tag{8}$$

2.4. Measuring Distance Using RSSI

In this section, we present a position estimation algorithm using RSSI that consider range measurement inaccuracies. Nodes in a sensor network can belong to two different classes, namely beacons and unknowns. We assume that the beacons have known positions, while the unknown nodes estimate their position with the help of beacons. The first step in RF-based localization is range measurement, *i.e.*, estimating the distance between two nodes, given the signal strength received by one node from the other. RF-based signal strength measurements are usually prone to inaccuracies and errors and, hence, calibration of such measurements is inevitable before using them for localization.

In [11], those statistics shows that: Each RSSI value corresponds to a distance scope, and high-intensity values has small probability, low-intensity values has large probability. So we can find the highest density peaks and filter out most wrong dates by doing Gaussian fitting. There is only one peak for each different RSSI measurement value, and the peak is steeper as the value is bigger, then the error is small, the peak is more slowly as the value is smaller, then the error become big. We get the fitting function:

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-(\frac{x-\mu}{\sqrt{2}\sigma})^2} \tag{9}$$

where $\mu=\frac{1}{N}\sum_{i=1}^{N}RSSI_i$ is the average value and $\sigma=\sqrt{\frac{\sum_{i}^{N}RSSI_i}{N-1}}$ is the standard deviation.

It is hard to find out the RSSI peak value of each measurement point. The value can be substituted into (9), when $0.7 \le f(x) \le 1$, we consider it is a large probability event and can be reserved. Moreover we can obtain the determined RSSI value by taking the average of the reserved RSSI values. N is the number of received beacon nodes.

Log Distance Path Loss Model is a basic way of estimating path loss as a function of distance between the nodes. The model is normally expressed as following equation (10).

$$RSSI(dB)=-10n\lg(d)+A \tag{10}$$

where the initial signal strength A describes the absolute value of RSSI with 1 m distance to the transmitting unit. The signal propagation coefficient n shows the damping of the signal. Both parameters must be determined empirically. In following sections, the determination approach and the experimental results will be given.

Consider a network with two types of nodes n_A anchor nodes with known location and n_S sensor nodes with unknown location, for a total of $n=n_A+n_S$ nodes. For simplicity, let the nodes lie on a plane such that mode i has location $x_i \in R^2$ indexed through i, $i=1\ldots n_A$ for the anchors and $i=n_A+1\ldots n$ for the sensors.

Neighboring nodes i and j measure the link distance $d_{ij}=\sqrt{(x_i-x_j)+(y_i-y_j)}$ between them through received signal strength.

Neighboring nodes i and j measure the link distance l_{ij} between them through received signal strength and Hop Distance. Where $l_{RSSI_{pj}}$ means the distance from anchor p to node j by RSSI and $l_{Hop_{pj}}=L_j\times HopDis\tan ce$ means the distance from anchor p to node j by hop, the estimated l_{ij} is written as:

$$l_{ij}=\frac{l^2_{RSSI_{pj}}+l^2_{Hop_{pj}}}{l_{RSSI_{pj}}+l_{Hop_{pj}}} \tag{11}$$

In formula (2), it is known that $d_i=l_{ij}$.

Given the locations of the anchor nodes and the measured distances between neighboring nodes in the network, the general problem considered by the nonlinear programming to solve for the locations of the sensors x_P, $p=n_A+1\ldots n$ follows:

$$\min\ t$$

$$s.t.\begin{cases} d_{ij}<L_jD & \forall i\in n_A,\ j\in n_s \\ d_{ij}>l_{ij} & \forall i\in n_A,\ j\in n_s \\ GZ=h \end{cases} \tag{12}$$

For this constrained nonlinear least square optimization problem, it can be solved by the Lagrange multiplier approach or just apply the '*fmincon*' function in Matlab optimization toolbox for convenience. In this paper, we use the '*fmincon*' function to resolve Formula (12).

Practically, the number of operations is typically bounded by $O(n^3)$ in solving a localization problem with n sensors.

3. Simulation Results

To validate our improved method, we consider an experiment region of square area of 50m×50m and sensor nodes are assumed to be randomly distributed in that area. The number of sensor nodes and the radio range of sensor nodes will be varied. We have implemented a number of experiments to cover a wide range of algorithm configurations including varying the ratio of anchor nodes, the number of unknown nodes, and the radio range.

Figure 3 shows the variation of the average localization errors as the reference ratios. In this experiment, the number of sensor nodes is fixed to 100. Suppose that the total estimation error is the summation of the Euclidean errors between true positions and estimated positions of all unknown nodes. Here the average localization error is defined as ratio between the total error and the number of the unknown nodes. Figure 3 is obtained by averaging over 100 dependent network simulations. It can be seen from Figure 3 that the average localization error by the NPRDV-Hop algorithm is obviously less than the DV-Hop method in all considered conditions. For example, with 20 anchor nodes (10%), our NPRDV-Hop has an average error of about 2m, whereas the DV-Hop has an average error of about 6m.

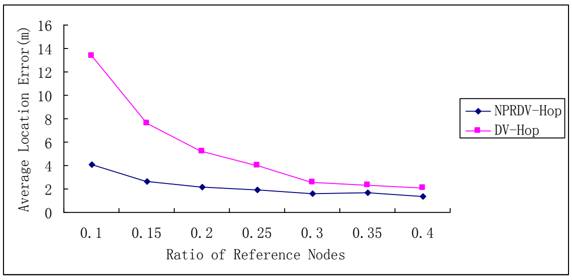

Figure 3. Average Location Errors vs. Ratio of Reference Nodes

Figure 4 shows the variation of the average localization errors as the reference ratios. In this experiment, the parameter of NPRDV-Hop is same as [7]. Figure 4 is obtained by averaging over 100 dependent network simulations. It can be seen from Figure 4 that the average localization error by the NPRDV-Hop algorithm is slightly worse than the SATOM method in all considered conditions. But the complexity of SATOM is about $O(n^5)$.

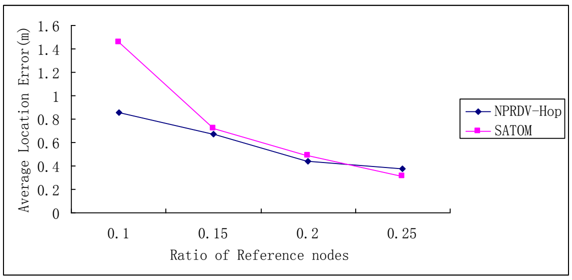

Figure 4. Average Location Errors vs. Ratio of Reference Nodes

The number of unknown nodes affects the NPRDV-Hop algorithm. In this experiment, the number of anchor nodes is fixed to 20. We can see from Figure 5 that the location error of these two algorithms is decreased with increasing the number of unknown nodes. This is because with the increase of unknown nodes, the node density in networks is increased; consequently the average number of neighbors is also increased. Thus, the network will be well connected and has a higher connectivity. This increases probability that there exist unknown nodes located on the line between anchor node *i* and *j* in each broadcast of hop count. Then the average *Hop Distance* estimated by any pair of anchor nodes will be accurate and thus the estimated distance between the unknown node and the anchor node using average *Hop Distance* will be closer to the true distance between the unknown node and the anchor node. So the location error of the algorithm is slightly decreased with increasing the number of unknown nodes. Our NPRDV-Hop algorithm also achieves better performance than the DV-Hop in the scenario.

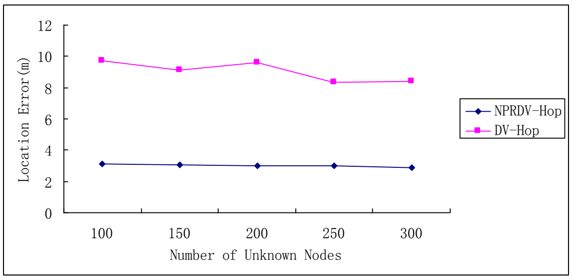

Figure 5. Location Errors vs. Number of Unknown Nodes

4. Conclusions

In this paper, we present a nonlinear programming algorithm based on RSSI and improved DV-Hop algorithm, called NPRDV-Hop. The algorithm makes four major contributions to the localization problem in the wireless sensor networks. Firstly, a hop distance is improved. This scheme could assure that the most nodes receive the *Hop Distance* from beacon node who has the least hops between them. Secondly, an improved Heron's formula is introduced as objective function. Thirdly, Gauss distribution is introduced to select RSSI so that the error of distance is little. There is only one peak for each different RSSI measurement value, and the peak is steeper as the value is bigger, then the error is small, the peak is more slowly as the value is smaller, then the error become big. Lastly, the general problem is considered by the nonlinear programming to solve for the locations of the sensors. Furthermore, it can be solved by the Lagrange multiplier approach or just apply the '*fmincon*' function in Matlab optimization toolbox for convenience. It explored the influence of anchor nodes on localization performance of the NPRDV-Hop algorithm. The proposed method can improve location accuracy and coverage without increasing hardware cost of sensor node. Simulation results show that the performance of this algorithm is superior to the original DV-Hop algorithm. Compared with DV-Hop, it is more available for WSNs.

Acknowledgment

The author would like to thank the Chongqing Natural Science Foundation under Grant No. cstc2012jjA40038, cstc2013jcyjA40023 and the Science and Technology Research Project of Chongqing Municipal Education Commission of China. The work presented in this paper was supported in part by the Ministry of Industry and Information Technology of the Peoples Republic of China for the special funds of Development of the Internet of things (2012-583).

References

[1] T. D. J, "Statistical theory of passive location systems", IEEE Trans. On AES, vol. 20, no. 2, **(1984)**, pp. 183-198.
[2] T. S. Rappapport, "Wireless Communications: Principles and Practice", Prentice Hall: New Jersey **(1996)**, pp. 50-143.

[3] G. L, E. D, “Robust range estimation using acoustic and multimodal sensing”, // IEEE International Conference on Intelligent Robots and Systems **(2001)** October 29 2001 - November 3, vol. 3, pp. 1312-1320.
[4] X. Cheng, T. A, G. Xue, D. Chen, “TPS: a time-based positioning scheme for outdoor wireless sensor networks, // IEEE INFOCOM, **(2004)** March 7-11, Hong Kong, China, pp. 2685-2696.
[5] J. Chen, K. Yao, and R. Hudson, “Source Localization and Beam forming”, IEEE Signal Processing Magazine, vol. 19, **(2002)**, pp. 30-39.
[6] C. S, H. M, H. J., “Gps-Free positioning in mobile Ad Hoc networks”, //Proc. of Hawaii Int'l. Conf, System Sciences, **(2001)**, pp.3481-3490.
[7] Z. Liu, J. Chen and X. Chen, “A New Algorithm Research of Sequence-Based Location Technology in Wireless Sensor Networks”, Chinese Acta Electronica Sinica, vol. 38, no. 7, **(2010)**, pp. 1552-1556.
[8] D. Niculescu, B. Nath, “Ad Hoc Positioning System (APS)”, // Proc. of the IEEE GLOBECOM, San Antonio, **(2001)**, pp. 2926-2931.
[9] L. Doherty, K. Pister, L. E. Ghaoui, “Convex position estimation in wireless sensor networks”, // IEEE INFOCOM, Anchorage, AK, vol.3, **(2001)**, pp.1655-1663.
[10] Y. T. Chan and K. C. Ho, “A simple and efficient estimator for hyperbolic location”, IEEE Transactions on Signal Processing, vol. 42, **(1994)**, pp. 1905-1915.
[11] V. Ramadurai and M. L. Sichitiu, “Localization in Wireless Sensor Networks: A Probabilistic Approach”, // Proceedings of the International Conference on Wireless Networks, **(2003)** June 23 - 26, Las Vegas, Nevada, USA. pp. 275-281.

Author

Fengjun Shang (1972), male, finished his Ph.D. degrees in Instrument Science and Technology at the College of Opto-electronic Engineering, Chongqing University, China, in 2005. Since then he works at the Institute of Computer Network Engineer in Chongqing University of Posts and Telecommunications, China. He was a visiting scholar in University of Wollongong, Australia, from November 2007 to November 2008. His research interests include sensor network, IOT, network optimization and Cloud Computing.

