Localization algorithm of preferred beacon nodes based on ZigBee network

Ying Ni[1,2] *

[1] *College of Electronic and Information, Nanjing University of Aeronautics and Astronautics, Nanjing, JiangSu, 210016, China*

[2] *College of Power and Electric Engineering, Nanjing Institute of Industry Technology, Nanjing, JiangSu, 210023, China*

Received 1 December 2014, www.cmnt.lv

Abstract

A new localization algorithm was proposed to enhance positional accuracy of ZigBee from improving distribution of beacon nodes. In this algorithm, beacon nodes were distributed as regular hexagon, with preferred appropriate beacon nodes in location calculation. RSSI was adopted as the distance reference. Moreover, Multimetering Averaging Method was used to reduce the impact of Gauss noise on RSSI.

Keywords: ZigBee Network; cellular network; RSSI; Guass noise.

1 Introduction

Localization technique is used for real-time detection on location information of an object or person, typically on occasions where location information of important objects or places are necessary for real-time control, such as cars, valuable goods, miners, and firemen in the rescue. Currently, satellites have been used in the more mature localization technology for localization, such as the American GPS localization System and Chinese Beidou Navigation System. However, their localization accuracy can only reach about 10m even in the military. Moreover, localization of indoor environment cannot be achieved due to poor received satellite signal. Therefore, an indoor localization technology with high precision is needed in practical applications. Lots of studies have focused on these aspects, including the combination of ZigBee and RFID for localization [1]. In ZigBee [2], localization was achieved through RSSI Localization algorithm [3], but the accuracy cannot reach theoretical value after test. Therefore, how to improve the localization accuracy of RSSI algorithm is the key to ZigBee localization technology, as well as the starting point of the work.

2 ZigBee localization technology

It is the first step to establish ZigBee network for realization of ZigBee location. ZigBee network is mainly consisted of beacon nodes and measured nodes [4]. Beacon nodes provide reference coordinates while measured nodes' position needs to be tested. Measured nodes are installed in object or person to be located, thus achieving real-time localization of the object or person. In addition, gateway node is necessary to transmit the detected information to host computer. Different from ordinary ZigBee nodes, gateway nodes need an interface for communication with the host computer, optimally selecting beacon nodes with algorithm. Meanwhile, position information of measured nodes is calculated by software, unfolded through man-machine interface software. ZigBee network consisted of beacon nodes forms a backbone network of ZigBee wireless location [5, 6]. To ensure the location, the backbone network of ZigBee wireless location should include at least three nodes, as shown in Figure 1.

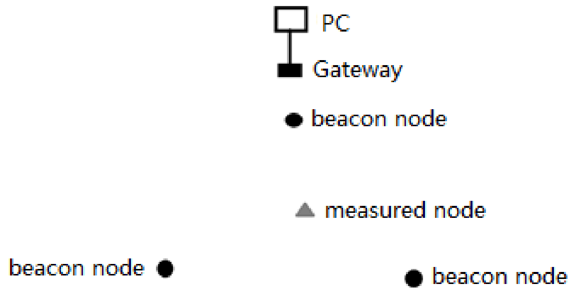

FIGURE 1 ZigBee location backbone network with 3 nodes

In Figure 1, the solid circle represents beacon node while the triangle represents measured node. In RSSI method, signal strength of beacon node is measured using measured node. Meanwhile, signal strength is converted into distance information through propagation model. Typically, the object's location can be determined with three reference points. However, the received signal strength of beacon nodes is unstable due to the complexity of indoor environment and mobility of object. Signal strength value can be different even at various times without movement of the object. Therefore, it is not

* ***Corresponding author*** e-mail: raonifei@126.com

enough relying on three beacon nodes. Generally, the more beacon nodes, the more accurate the measured location information will be, thus 4 to 8 beacon nodes are available. However, with the increasing number of beacon nodes, the computational workload will be very complicated. In the work, three of the six beacon nodes are preferably selected, obtaining location information of measured nodes by localization algorithms. PC and gateway were in wired connection, while wireless connections were adopted among beacon node, measured node and gateway. This eliminates the trouble of laying cable with saving time.

3 Algorithm model

3.1 WIRELESS SIGNAL PROPAGATION LOSS MODEL [7] [8]

In RSSI algorithm model, signal strength sent from beacon nodes is tested at measured node. Meanwhile, the received signal strength and corresponding coordinate information is transmitted to host computer through network, using localization algorithm and calculation software to compute the location information of measured nodes. The key issue of the algorithm is the transformation from signal strength to position information. Due to special wireless transmission environment between beacon nodes and measured nodes, wireless signal will be blocked by objects with decline, experiencing reflection, scattering and diffraction. Thus, there is a complex relationship between actual received signal strength and distance. A radio propagation model should be established to obtain the relationship between signal intensity and distance. Free Space Propagation Model is one of the common propagation path loss model, but it is not applied to ZigBee positioning system. Therefore, a modified model was adopted as Formula (1):

$$\frac{P_r(d)}{P_r(d_0)} = \left(\frac{d_0}{d}\right)^{\alpha} \quad , \tag{1}$$

Therefore,

$$d = \left(\frac{P_r(d_0)}{P_r(d)}\right)^{\frac{1}{\alpha}} \quad , \tag{2}$$

where Pr(d) is the receiving loss from the sending point to d; Pr(d0) the receiving power of transmitter at d0; d0 the reference distance, usually taking 1m indoors; α the path loss factor, usually taking different values in various environments.

The measured RSSI is a random component due to particularity of wireless transmission environment. However, it fluctuates around a stable value, in line with the Gaussian distribution model. Multimetering Averaging Method can be used to reduce the impact of random noise on measurement results.

3.2 LOCALIZATION ALGORITHM

In RSSI localization algorithm, measured nodes calculated transmission loss according to transmitted signal strength. Transmission loss was converted to distance using Formula (2), then calculating the location of node by algorithm. The calculation was not conducted at measured node in the work to ensure the accuracy. Instead, measured nodes sent signal strength and corresponding reference coordinates of beacon nodes to PC under ZigBee network, completing the calculation through software of PC.

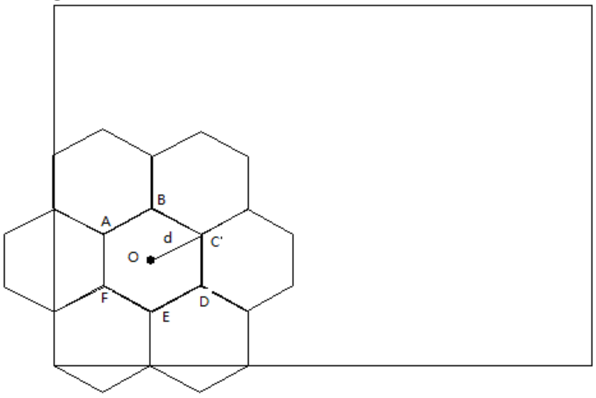

FIGURE 2 Cellular localization algorithm beacon node distribution

Firstly, the entire region should be separated into small areas. Equilateral triangle was used in reference [9]. Though with fewer beacon nodes, needed beacon nodes were not easily probably selected if in complex indoor environments. Therefore, regular hexagon was adopted as a targeted area according to the cellular concept of mobile communication network. The distribution of beacon nodes was shown in Figure 3.

Observe one cellular ABCDEF, setting its center point as O, the distance from O to endpoints as d. The value of d should not be too big or too small.

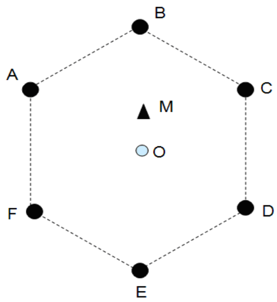

FIGURE 3 Hexagon location area

Generally, at least three preferred beacon nodes should be selected in location. The key issues of algorithms were how to determine the regular hexagon area of measured nodes and preferably selecting three points of the regular hexagon. In the work, information was sent from the received beacon node, calculating the distance from beacon node to measured node based on RSSI. Target location can be determined with three points due to the use of regular hexagon in localization. The three nodes with strongest signal strength were selected according to mathematical calculation, regardless of obstacles. The regular hexagon where measured nodes located was defined as location area, the outside area as non-location area. In order to accurately select three preferred beacon nodes in location area, the value of d was less than 5m [4]. Meanwhile, the transmission power of each node can drop to -25dBm.

4 Achievement of preferred beacon nodes

RSSI information could be extracted from received packet using afIncomingMSGPacket_t structure with byte LinkQuality. In the protocol stack, LQI could be directly obtained from the structure after receiving data. For example, the structure of receiving data was defined as pkt: LQI = pkt-> LinkQuality;

Then conversion relationship between RSSI and LQI is as follows:

$$RSSI = - (81- (LQI * 91) / 255). \quad (3)$$

In the protocol stack, void SampleApp_MessageMSGCB (afIncomingMSGPacket_t * pkt) is receiving function of wireless message. The corresponding LQI in every wireless message should be added to T_Buffer and uploaded to PC for node screening through serial port. Specific code was as follows:

```
void SampleApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  uint8 i, j;
  memcpy(T_Buffer.data,pkt->cmd.Data,pkt->cmd.DataLength);

  //BINDING COMMAND ID EXECUTE
  if(pkt->clusterId == SAMPLEAPP_LIGHT_CLUSTERID)
    HalLedBlink(HAL_LED_2,1,50,500);

  if('&'==T_Buffer.packet_Struct.Head)
  {
    // network management
    if((T_Buffer.packet_Struct.cmd[0] == 'J') && (T_Buffer.packet_Struct.cmd[1] == 'O') && (T_Buffer.packet_Struct.cmd[2] == 'N'))// there are new nodes joining network
    {
      if((T_Buffer.packet_Struct.DataBuf[0] == 'R') && (T_Buffer.packet_Struct.DataBuf[1] == 'F') && (T_Buffer.packet_Struct.DataBuf[2] == 'D'))//RFD node
      {
        for(i=0; i<8; i++)
        {
          JoinNode.RfdAddr[JoinNode.RfdCount][i] = T_Buffer.packet_Struct.DataBuf[3+i]; // physical address
        }
        For (i=0; i<2; i++)
        {
          JoinNode.RfdAddr[JoinNode.RfdCount][8+i] = T_Buffer.packet_Struct.Saddr[i]; // network address
        }
        for(i = 0;i<10;i++)
        {
          JoinNode.RfdPaterAddr[JoinNode.RfdCount][i] = T_Buffer.packet_Struct.DataBuf[11+i]; // Parent node physical address+ network address
        }
        for(j=0; j<JoinNode.RfdCount; j++)// determine if there are repeated join nodes
        {
          HaveFlag = 1;
          for(i=0; i<8; i++)
          {
            if(JoinNode.RfdAddr[JoinNode.RfdCount][i] != JoinNode.RfdAddr[j][i])
            {
              HaveFlag = 0;
              break;// no
            }
          }
          if(HaveFlag == 0)continue;
          JoinNode.RfdCount--;// yes
          JoinNode.RfdAddr[j][8] = T_Buffer.packet_Struct.Saddr[0];
          JoinNode.RfdAddr[j][9] = T_Buffer.packet_Struct.Saddr[1];
          break;
        }
        JoinNode.RfdCount++;
      }
      else if((T_Buffer.packet_Struct.DataBuf[0] == 'R') && (T_Buffer.packet_Struct.DataBuf[1] == 'O') && (T_Buffer.packet_Struct.DataBuf[2] == 'U'))// routing node
      {
        // queue storage
        for(i=0; i<8; i++)
        {
          JoinNode.RouterAddr[JoinNode.RouterCount][i] = T_Buffer.packet_Struct.DataBuf[3+i];
        }
        for(i=0; i<2; i++)
        {
          JoinNode.RouterAddr[JoinNode.RouterCount][8+i] =T_Buffer.packet_Struct.Saddr[i];
        }
        for(i = 0;i<10;i++)
        {
          JoinNode.RouterPaterAddr[JoinNode.RouterCount][i] =T_Buffer.packet_Struct.DataBuf[11+i];
        }

        // test
        for(j=0; j<JoinNode.RouterCount; j++)// determine if there are repeated join nodes

        {
          HaveFlag = 1;
          for(i=0; i<8; i++) // compare physical address
          {
            if(JoinNode.RouterAddr[JoinNode.RouterCount][i] != JoinNode.RouterAddr[j][i])
            {
              HaveFlag = 0;
```

```
        break;// no
      }
    }
    if(HaveFlag == 0)continue;

    // if there are the same physical addresses
    JoinNode.RouterCount--; // count minus 1, final result plus 1.

    // update the network address after changing duplicate address
    JoinNode.RouterAddr[j][8] = T_Buffer.packet_Struct.Saddr[0];
    JoinNode.RouterAddr[j][9] = T_Buffer.packet_Struct.Saddr[1];
    break;
   }
   JoinNode.RouterCount++; // back++
  }
 }
 else // data transmitting
 {
  T_Buffer.packet_Struct.DataBuf[20]=pkt->LinkQuality;

  HalUARTWrite ( MT_UART_DEFAULT_PORT, T_Buffer.data, 32);// output from the serial ports
 }
}
else //RFID
{
 HalUARTWrite (MT_UART_DEFAULT_PORT, pkt->cmd.Data,pkt->cmd.DataLength);// output from the serial ports
}
}
```

PC using the serial port to read the data uploaded from the bottom, specific code is as follows:

```
private void tmrReceive_Tick(object sender, EventArgs e) // data receiving
    {
      List<byte> datalist=new List<byte>();
      if (comm.BytesToRead > 30)
      {
        int n = comm.BytesToRead;
        byte[] data = new byte[n];
        comm.Read(data, 0, n);
        datalist.AddRange(data);
        duqu(datalist);// analyze the data
        comm.DiscardInBuffer();
        comm.DiscardOutBuffer();
      }
    }
```

Among them, the configuration code of serial port is as follows:

```
try
    {
      if (this.comm.IsOpen)
      {
        this.comm.Close();
        this.btnOpen.Enabled = false;
        this.cboxName.Enabled = true;
        this.cboxBaud.Enabled = true;

      }
      else
      {
        this.comm.PortName = this.cboxName.Text;// set the name of serial port
        this.comm.BaudRate = int.Parse(this.cboxBaud.Text);// set the baud rate of serial port
        this.comm.Open();
        this.btnOpen.Enabled = true;
        this.cboxName.Enabled = false;
        this.cboxBaud.Enabled = false;
      }
      this.btnOpen.Text = this.btnOpen.Text.Equals("open the serial port ") ? " close the serial port " : " open the serial port ";
    }
    catch (Exception e1)
    {
      MessageBox.Show(e1.ToString());
    }
```

PC data acquisition, start the timer for timing acquisition. Specific code is as follows:

```
private void tmrSend_Tick(object sender, EventArgs e) // Cyclically send instruction of reading the sensor's value of each node in the whole network
    {
      comm.DiscardInBuffer(); // clear serial data
      comm.DiscardOutBuffer();

      if (count >= allNet.Count)
      {
        count = 0;
      }
      int i = count++;
      List<byte> data = new List<byte>();
      data.Add(Convert.ToByte('&'));
      data.Add(Convert.ToByte('W'));
      data.Add(Convert.ToByte('S'));
      data.Add(Convert.ToByte('N'));
      data.Add(Convert.ToByte('R'));
      switch (allNet[i].JD) //&WANRWS``` The first byte, using 3 byte in showing part; the actual number is based on the nodes number
      {
        case 0x01:
          {
            data.Add(0x57);
            data.Add(0x53);
            break;
          }
        case 0x02:
          {
            data.Add(0x57);
            data.Add(0x53);
            break;
          }
        case 0x03:
          {
            data.Add(0x57);
            data.Add(0x53);
            break;
          }
      }
      data.Add(allNet[i].Wang1);
      data.Add(allNet[i].Wang2);
```

```
            for (int j = 0; j < 6; j++)
            {
              data.Add(Convert.ToByte('y'));
            }
            data.Add(Convert.ToByte('*'));
            comm.Write(data.ToArray(), 0, data.Count);
            this.tmrReceive.Interval = 1000;// 1s Start reading serial receive buffer for 1second
            this.tmrReceive.Enabled = true;
          }
```

Followed by the section of data analysis, LQI was converted into RSSI with specific conversion statement as follows:

```
labone.Text += (-(81 - (Convert.ToInt32(data[30].ToString(), 16) * 91) / 255)).ToString();
```

```
r[0] = (-(81 - (Convert.ToInt32(data[30].ToString(), 16) * 91) / 255)) * -10;
Sum[0] = r[0];
```

The above analysis is an example based on a node data. Statements are similar for adding other nodes. In PC program, bubble sort was used in RSSI values of all nodes, taking the first three values. All RSSI data could be arranged in sum array with descending order. Knowing the specific node according to selected RSSI, thus the regional context can be determined. With experimental measurement, data relationship between RSSI and actual distance was shown in Table 1. When preferred beacon nodes were selected, the location of measured nodes could be estimated using ZigBee.

TABLE 1 Relation between measured RSSI and actual distance

Serial number	actual distance	RSSI	Noise
1	0.6m	-31dBm	-127dB
2	2.1m	-48dBm	-127dB
3	5m	-60dBm	-127dB
4	13m	-75dBm	-127dB
5	19m	-80dBm	-127dB
6	80m	-87dBm	-127dB

5 Calculation of measured nodes' location

Set six vertices of location area as A, B, C, D, E, F, respectively; the center as O; measured node as M; three preferred beacon nodes as A, B, C. The distance map set implied from the loss model Formula (2) is {dA, dB, dC}, the location of beacon nodes {(xA, yA), (xB, yB), xC, yC)}. Set location of measured node as (xM, yM). It is discussed according to different situations of {dA, dB, dC}.

(1) If 0≤dA, dB, and dC≤d, taking A, B, C as the center, respectively, we draw circles with the radius of dA, dB and dC. Then the three circles intersect at one point shown in Figure 4. This intersection is the location of measured node M. The measured nodes' location can be obtained with the equation (xM, yM), see Formula (4):

$$\begin{cases}(x_M - x_A)^2 + (y_M - y_A)^2 = d_A^2 \\ (x_M - x_B)^2 + (y_M - y_B)^2 = d_B^2 \\ (x_M - x_C)^2 + (y_M - y_C)^2 = d_C^2\end{cases} . \quad (4)$$

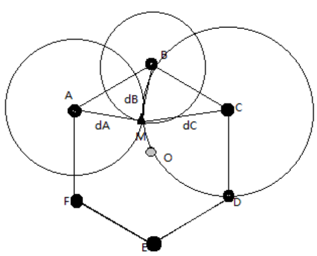

FIGURE 4 Situation 1

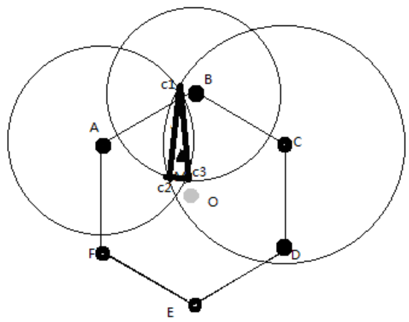

FIGURE 5 Situation 2

There is another situation given in Figure 5, where there are two coordinates calculated by Formula (5). We choose one closer to B as C1. The coordinates of C2 and C3 can be computed as (xC2, yC2), (xC3, yC3) using the same method.

$$\begin{cases}(x_{C1} - x_A)^2 + (y_{C1} - y_A)^2 = d_A^2 \\ (x_{C1} - x_C)^2 + (y_{C1} - y_C)^2 = d_C^2\end{cases} . \quad (5)$$

Therefore, coordinates of measured nodes can be calculated by Formula (6):

$$\begin{aligned}&(x_M - x_{C1})^2 + (y_M - y_{C1})^2 \\ &= (x_M - x_{C2})^2 + (y_M - y_{C2})^2 \\ &= (x_M - x_{C3})^2 + (y_M - y_{C3})^2\end{aligned} . \quad (6)$$

(2) If d≤dA (dB or dC) in Figure 6, then obstacles exist between measured node and beacon nodes. Therefore, the loss exponent αin Formula (2) should be corrected in calculation of dA (dB or dC), appropriately increasing the value of path index. Then dA (dB or dC) is recalculated to make 0≤dA, dB, and dC≤d, with calculations according to the first or second situation.

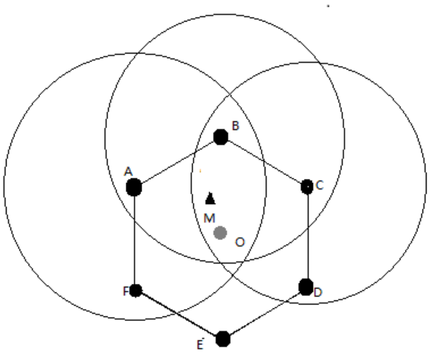

FIGURE 6 Situation 3

6 Conclusions

The location indoors or outdoors can be achieved by localization algorithm of preferred beacon nodes based on ZigBee. Influence of obstacle was effectively eliminated in preferred beacon nodes, thereby improving the accuracy of localization. However, RSSI values greatly change with the environment, varying in the same location at different times. Therefore, appropriate loss model remains to be further studied, which is also a difficulty of the research directions.

Acknowledgements

The work was supported by Open Fund Project of Intelligent Sensor Network Engineering Development Center in Jiangsu Province—"Research and Application Development of Multi-source Information Fusion Technology ", with Grant Number: ZK12-04-07.

References

[1] Zhang Ying, Yang Zhen 2012 Local Positioning System Based on RFID and ZigBee Technology *Electronic Design Engineering* **5** 175-8
[2] Chen J 2007 The Realization of ZigBee Wireless Sensor Networks and The Rresearch on Node Localization *Master Thesis* Harbin Institute of Technology (HIT) *(in Chinese)*
[3] Benkic K 2008 Using RSSI Value for Distance Estimation in Wireless Sensor Networks Based on ZigBee, Systems, Signals and Image Processing *IWSSIP 2008 15th International Conference* 303 -6
[4] Patwari N, Ash J N, S. Kyperountas A O Hero III, Moses R L, Correal N S 2005 Locating The Nodes: Cooperative Localization in Wireless Sensor Networks *IEEE Signal Process. Mag.* **22**(4) 54-69
[5] Chen Weike, Li Wenfeng, Shou Heng, Yuan Bing 2006 Weighted Centroid Localization Algorithm of Wireless Sensor Networks Based on RSSI *Journal of Wuhan University of Technology* **30**(2) 265-91
[6] Li Zhanbo, Liu Huiling 2012 Positioning of ZigBee-based Dynamic WSN Nodes. *Computer Engineering and Design* **3** 431-4
[7] Roxin A, Gaber J, Wack M., and Nait-Sidi-Moh A 2007 Survey of Wireless Geolocation Techniques *IEEE Globecom Workshops* 1-9
[8] Whitehouse K, Karlof C, Culler D E 2007 A Practical Evaluation of Radio Signal Strength for Ranging-based Localization. *ACM Mobile Computing and Communications Review* **11**(1) 41-52
[9] Zhu Jian, Zhao Hai, Sun Peigang, Bi Yuanguo 2007 Localization algorithm of Equilateral Triangle Based on Mean RSSI *Journal of Northeastern University (Natural Science)* **28**(8) 1094-7

Author

Ying Ni, 1979.02., Dantu, Jiangsu Province, China

Current position, grades: Ph. D Candidate of Nanjing University of Aeronautics and Astronautics, Lecturer of Nanjing Institute of Industry Technology
University studies: Data Communications
Scientific interest: Data Communications
Publications: 15 papers published in the international or national journals
Experience: She is a Ph. D Candidate of Nanjing University of Aeronautics and Astronautics and a lecturer of Nanjing Institute of Industry Technology, China. Her research interests include data communications and computer science.