Contiki Cooja Simulation for Time Bounded Localization In wireless Sensor Network

Conference Paper · April 2015

6 authors, including:

Mona Nasseri
University of California, Davis

Hussein Al-Olimat

Mansoor Alam
National University of Sciences and Technology

R.C. Green
Bowling Green State University

Contiki Cooja Simulation for Time Bounded Localization In wireless Sensor Network

Mona Nasseri, Hussein Al-Olimat, Mansoor Alam, Junghwan Kim
EECS Departent
Univ. of Toledo, Toledo, OH
Mona.Nasseri@rockets.utoledo.edu

Robert Green
Dept. of Comp. Sci.
Bowling Green Univ.
Bowling Green, OH
greenr@bgsu.edu

Wei Cheng
Dept. of Comp. Sci.
Virginia Commonwealth Univ.
Richmond, VA
wcheng3@vcu.edu

ABSTRACT

An attractive research topic in wireless sensor networks is the issue of localization – that is localizing an entire wireless sensor network accurately and within a reasonable time frame. A newly developed algorithm in this area is Time Bounded Essential Localization (TBEL), which allows for rapid, network-wide distribution of essential location information with full localization to occur at a later time. In order to further develop this line of research, this study evaluates the implementation of a variant of TBEL (TBL) on the Cooja simulation platform as it enables not only the simulation of TBL, but the eventual implementation of this algorithm on real world motes. The Simulation results show that in a limited four rounds of communication, 76.67% of nodes would be localized with an average localization error of 0.5 meters in a 200m by 200m simulation area including 30 blind nodes and 50 anchors.

Keywords

Wireless Sensor Network; Localization; Time bounded Essential Localization; Contiki- Cooja.

ACM Classification Keywords

C.2 **[Computer-Communication Networks]**: Wireless communication; D.4.8 [**Performance**]: Simulation

INTRODUCTION

A communication network includes several nodes, each with the computing capability to transmit and receive messages over various mediums [1, 13]. The development of low power and multi-functional wireless sensors which are able to communicate effectively over short distances provides the opportunity to take advantage of wireless sensor networks in indoor environments and to monitor and control conditions such as temperature, light, etc. Additionally wireless sensor networks have a wide spectrum of applications including search and rescue operations, particularly during the occurrence of natural disasters, tracking and monitoring patients in hospitals, road traffic monitoring, and military surveillance. In all of the mentioned applications, wireless sensors collect data and send it to the base station for the future actions. In order to achieve this purpose, one must be able to determine any individual node's position in both an accurate and timely manner. This is problematic, as highly accurate methods, such as using the Global Localization System (GPS), are not possible in many closed areas. GPS usage may also be limited due to some other issues such as size, cost, and energy consumption. Therefore, nodes cannot always access GPS to localize themselves.

To combat this issue, different localization algorithms have been introduced to calculate the locations of sensors [1, 4, 13, 15]. In many cases, the algorithms use the knowledge of a specific node called an anchor. An anchor is simply a node that can independently and accurately localize itself, typically using GPS [4, 15]. These algorithms use the anchor's position and other measurable parameters that are calculated by transferring signals between anchor nodes and un-localized/blind nodes to determine blind node's location. The most common signal parameters in WSN localization can be generally classified into four classes: Received Signal Strength (RSS), Time of Arrival (TOA), Time Difference of Arrival (TDOA), or Angle of Arrival (AoA).

This study implements a modified version of Time Bounded Essential Localization (TBEL) [5]. The study presented in [5] focuses on the newly proposed TBEL algorithm, where the concept of essential localization and localization within specific time-bounds are introduced. In the TBEL algorithm, each node initially constructs a Local Coordinate System (LCS) amongst neighboring nodes and localizes itself to a local coordinate system, while the algorithmic goal is the physical localization of the entire network to a global coordinate system. This paper uses the time constraint aspect of TBEL to achieve a time limited

SpringSim 2015, April 12 - 15, 2015, Alexandria, VA, USA

localization method (Time Bounded Localization: TBL) and implements the idea of TBL in the Cooja simulation environment with the end goal of porting the simulation to the real-world motes and extending the methodology to TBEL.

Cooja is a Java-based simulation environment that simulates the work of the Contiki open source operating system on wireless motes. Contiki-OS is designed for sensor networks and other networked embedded devices. Instant Contiki includes the Contiki source code, compilers and tools for developing software. Its environment is an Ubuntu Linux installation which runs within the VMware Player (virtual machine execution environment) [6, 14]. Contiki provides TCP/IP communication, including loadable modules and radio networks, which are protocol independent and can be simulated with the Rime stack. Also, it provides protothreads and software-based power profiling.

The remainder of the paper is organized as follows: WSN localization techniques are examined first; simulation in Contiki is described, following by results and conclusion.

LOCALIZATION ALGORITHM FOR WIRELESS SENSOR NETWORK

Localization algorithms are iterative, range-based procedures that are typically started via an information flood that originates from anchor nodes. Upon initial transmission, all nodes within range receive the broadcast from the anchors and begin the localization process. In the first phase of localization, known as the ranging phase, each node measures the distances between itself and other nodes based on techniques like TOA, TDOA, or RSSI (received signal strength Index). In this research, we calculate the distances between neighboring sensors using the RSSI [17]. This requires no additional hardware and has a small effect on energy consumption, resulting in a lower overall cost. It should be noted that, in free space, RSS at the receiver varies as the inverse square of the distance between the transmitter and the receiver. To calculate distance (d), the RSS should be converted to power (P_r) that is in direct relation to the transmitter power (P_t). The received signal power P_r is calculated according to (1), where λ is the signal wavelength, and G_r and G_t are receiver and transmitter antenna gains [16];

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2} \tag{1}$$

In the second phase, known as the position estimation phase, each blind node estimates its position based on the distances found in the first phase. The localization procedure is thoroughly discussed and mathematically proved in [3].

Time Bounded Localization

TBEL [5] is a newly proposed algorithm for conditions that require limited time for localization, such as military applications where a node's location should be determined in a specific time in order to avoid attacks.

In the TBEL algorithm, each node localizes itself by constructing a Local Coordinate System (LCS) amongst those nodes closest to a given node. Each node can belong to several LCSs, therefore the corresponding coordinates can be transformed between the various LCSs in an iterative manner. For global localization purpose at each LCS, three nodes should be considered as either anchor or localized nodes in order to complete the local localization process. Anchor nodes are aware of their locations, using various methods such as GPS. In the case where there are a lower number of localized nodes in the neighborhood, un-localized nodes can use other methods to find their location by using more complicated calculations [2].

In the localization process nodes broadcast signal repeatedly. During signal transmission and reception process, nodes can estimate their location and broadcast it. The maximum needed time for a node to transmit its location and receive its neighbor locations, during global positioning, is the time taken for one-flooding round of communications [5].

In TBEL, essential and global localizations are two steps in the localization process. Several LCSs are formed during the initial k communication rounds, which is called essential localization. Then, global localization takes place through communications between the LCSs that have been formed. Information flooding continues until all possible LCSs formed and network get essentially localized, then communications among LCSs result in global localization.

In our simulation, anchors know their global locations, therefore blind nodes can globally localize themselves during k initial communication rounds, hence we take advantage of the time bounded phase, in order to complete localization in k rounds of flooding. We calculate an upper bound for k, (k_{upper}), based on the worst case scenario to connect all nodes. If two specific nodes have the highest possible distance - the area diameter - the number of hops to connect them can be considered as an upper bound for k that depends on the node transmission range.

In the proposed TBL simulation in Cooja, anchors are randomly distributed in the experimental area which is applicable for mobile networks, including mobile anchors that can freely move to any position. During the first-communication round, anchor nodes broadcast messages through the entire network. Blind nodes, which receive signals from at least three anchor nodes will then use three circle intersection formulas to estimate their location. In the next communication round, localized nodes will also broadcast their estimated locations to help blind nodes. This process is continued for predefined k-rounds, a value which can be smaller than k_{upper}, that is determined by observing the network and the environment in which the maximum number of nodes are localized. After k-rounds, nodes stop sending signals, but some nodes still do not connect to at least three localized/anchor nodes. Therefore, they are called isolated nodes and do not globally localize.

Mathematical Analysis

As mentioned before, to complete the localization process, a connection to three anchor nodes is essential. In this section, we will discuss how the anchor-based distance measurement is used to find a node's location.

In Figure 1, the location of three nodes, S_1, S_2 and S_3 are known, and the location of S_0 should be estimated. At first, the two intersection points (P_1 ($estX_1$, $estY_1$), P_2 ($estX_2$, $estY_2$)), of two circles with center points, S_1 and S_2, and radius of d_1 and d_2, are found. Then, the third circle, centered at S_3, determines the correct intersection point (here P_2) which is the unknown location (x_0, y_0). Therefore, the first step is designated to find the distance between the blind node and anchors (d_1, d_2 and d_3), which is done by applying RSSI measurement.

To find intersection points the following formulas are used:

$$\begin{cases} x_0 = x_1 \pm h \times \dfrac{y_2 - y_1}{|s_1 s_2|} \\ y_0 = y_1 \pm h \times \dfrac{x_2 - x_1}{|s_1 s_2|} \end{cases} \tag{2}$$

where,

$$h^2 = d_1^2 - a^2$$
$$a = \frac{d_1^2 - d_2^2 + |s_1 s_2|^2}{2 \times |s_1 s_2|} \tag{3}$$

The third circle, which is centered at S_3, can eliminate the point which does not fit. But in Contiki applying complicated equations to find a solution does not lead to the correct answers due to its limitations in memory size and memory addresses. This is due to the use of several nonlinear functions, such as the square root, which do not work properly in the Contiki environment due to the lack of support for floating point numbers. Thus, it is necessary to decrease the number of calculations as much as possible. In order to achieve this, the method shown in the flowchart in Figure 2 is proposed. In this method a comparison of distances is made that lead to a usable estimate.

SIMULATION IN CONTIKI

Overview of Contiki Operating System

To set up a vast network of physical nodes, simulators are applied to develop and test systems. One of them is Cooja, a Java-based simulator for Contiki-OS, which allows real hardware platforms to be simulated. It is designed for memory-constrained environments, such as the nodes used in WSNs. Contiki consists of an event-driven kernel and features contain dynamic loading and unloading of individual programs [6]. Additionally, Contiki supports pre-emptive multithreading, which performs through a library and it can be associated with programs. It also supports Protothreads, a stackless type of thread designed for memory-constrained systems [7]. Protothreads are programming concept used on top of the systems, to simplify executions of high-level functionality.

Contiki supports a lightweight implementation of the TCP/IP stack through the uIP (micro IP) library. uIP was implemented to be used in 8- and 16-bit microcontrollers and has the minimum set of features of a full TCP/IP stack [8]. It also provides tailored wireless networking stack called Rime (radio communication stack of Contiki) [9]. It supports simple and complex operations, including sending a message to all neighbors or to a specified neighbor.

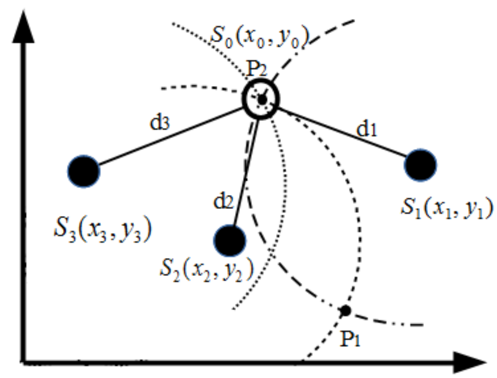

Figure 1. Blind node is connected to 3 anchors/localized nodes.

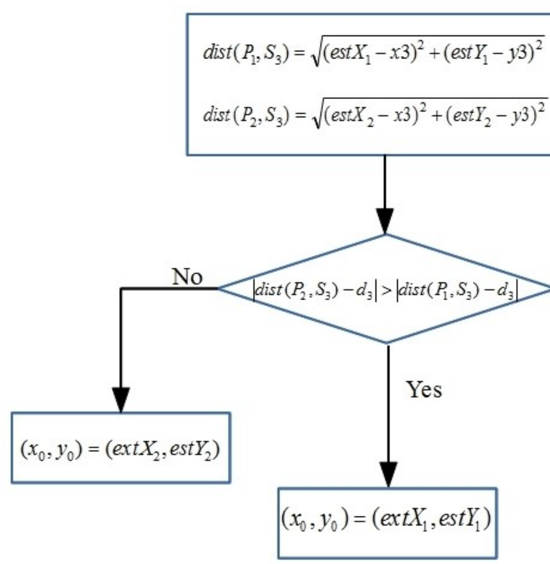

Figure 2. Suggested flowchart to decide about the estimated location

The messages are sent while the RF output power of the device is programmable and is controlled by the Power Amplifier Level; Transmitter PA_LEVEL register, which for CC2420 ZigBee, 8 values are defined between 3-31 corresponding to output power from -25 dBm to 0 dBm.

Wireless Sensor Network Simulation in Cooja

Cooja simulation contains multiple nodes being simulated. It supports several types of wireless sensor nodes (aka. motes), including Exp5438, Z1, TelosB and Tmote Sky [12]. Tmote Sky motes that are used in our simulation are typically powered by an external battery pack (containing 2 AA batteries) and operate in the range of 2.1 to 3.6V DC. However, the voltage must be at least 2.7V during programming the microcontroller flash or external flash [12]. In this study, the output power level is set to 31 [11].

In localization simulation, each node should be connected to a node type (here Blind or Anchor). Later, when running the simulation, all nodes act in turn. When all of them have acted once, the simulation time is updated and the process is repeated.

Additionally, to estimate the energy consumption, Contiki has built-in timer modules including, timer, stimer, ctimer, etimer, and rtimer [10]. We use the Contiki rtimer library that provides scheduling and execution of real-time tasks. So that, when the mote enters a specific mode, the energy estimation module is called and produces a time stamp. When the mote exits that mode, the time is recorded, and the time difference between on and off states is added to the table entry of the mote. To find energy consumption, the following code is added to the localization code:

```
P=(unsigned long) (Time × current × voltage)/ RTIMER_SECOND
```

where "RTIMER_SECOND" is the function and specifies the number of ticks per second. In our simulation, it has the value of 32,768 Hz clock that, divided by 8, produces 4,096 clock ticks per second. Table 1 includes more detailed simulation parameters.

Localization Process

As mentioned before, two kinds of motes are considered for simulations: anchor nodes and the blind/un-localized nodes, which the Sky mote type is chosen for both of them in simulations. The pseudo code in Algorithms 1 and 2 shows the process which nodes go through, to complete localization. Anchor nodes broadcast their positions *k* times as shown in Algorithm 1, and after *k* rounds, they stop broadcasting.

Anchor nodes send an array message, including four components. The first item should be 1 to show anchor's capability to collaborate in the localization and, the second one is the unique node ID, following by two dimensional location of the node.

Algorithm 2 shows the blind node's framework. Each blind node can localize itself if it receives signals from three anchors or localized nodes including their locations. Here, each node finds its distances from other localized nodes using RSS. Therefore, it can estimate its location according to (2), (3) and suggested flowchart in Figure 2. When a blind node becomes localized, it changes its localization condition from '0' to '1', indicating that it can cooperate to find blind node's position. To determine the localization accuracy, localization error is used which is calculated according to (4) where (X_{real}, Y_{real}) and (X_{est}, Y_{est}) show the real and estimated positions, respectively.

$$E = \sqrt{(X_{real} - X_{est})^2 + (Y_{real} - Y_{est})^2} \qquad (4)$$

Parameter	Value/Model
Micro Controller Unit (MCU)	MSP430
Radio	CC2420
MAC driver	ContikiMAC
TX/RX success ratio	100%
Transmission range	50m
INT range	100m
PACKETBUF_SIZE	128 B
PACKETBUF_HDR_SIZE	48 B
Current consumption: Radio transmitting at 0 dBm	17.4 mA
Current consumption: Radio receiving	19.4 mA
Current Consumption: MCU on, Radio off	1800 µA
Current Consumption: MCU idle, Radio off	54.5 µA

Table 1. Simulation Parameters

```
Define counter=0
Set anchors to broadcast
While (counter < k) {
  Set a message:
  {Localization condition, node ID, X
coordinate, Y coordinate};
  Set the power amplifier level
  cc2420_set_txpower(31);
  Broadcast the message;
  Counter++;
}
End process
```

Algorithm 1. Anchor node framework

```
If the node is localized
  While (counter < k){
    Broadcast the massage:
    {Localization condition,node ID, X
  coordinate, Y coordinate};
  }
If the node is not localized
  While (i < 3){
    Receive broadcasted information (i)
    If localization condition = 1 {
      Save the received information:
      {Node ID, X coordinate, Y coordinate,
    RSSI};
      i++;
    } Else {
        Ignore the message;
      }
  }
      If i==3
    Calculate the node location;
    End
```

Algorithm 2. Blind node framework

RESULTS

In this study, a 200 m × 200 m area is considered in which 30 blind nodes and 50 anchor nodes are distributed randomly. An example of such an area is shown in Figure 3 (blind and anchor nodes were numbered from 1-30 and 31-80 respectively). TBL algorithms are implemented in Sky motes as blind and anchor nodes with the highest defined PA level, 'cc2420_set_txpower(31)', with the Constant path-loss model. The simulations are run for five times, and the average values are reported. As mentioned before, in the TBL method localization would be completed in *k* or less than *k* rounds of flooding. Figure 4, Shows the nodes' estimated and real locations after 3 rounds. The distance errors are mentioned next to the nodes and the total average error is 0.34 m. TBL method is compared with Multilateration Newton-Raphson localization method [18], in which an initial guess for node's location is optimized in an iterative manner according to node's distance to anchors and it shows the 7.8 meter total average error for 50% localized nodes. Figure 5 shows how different values for k influence on localization time and the number of localized nodes. The upper bound for k can be obtained by dividing the diameter of the area by the transmission range (here 50 m), and the result is rounded to 6 for this simulation. During the first round, which takes 9.98 seconds, 18 out of 30 blind nodes are localized. By increasing the k value, more nodes are able to estimate their location, but after four rounds the number of localized nodes remains constant at 23 (76% localized). In general, after six rounds of broadcasting, nodes can stop the transmission or, act in the localization process. Additionally, the total time spent on localization would grow by increasing the rounds, which necessitates it to stop the localization process after six rounds to decrease energy consumption.

In Figure 6, the average power consumption for 80 nodes in the simulation was calculated according to the mentioned current consumption for different states in Table 1. To find the power consumption in Contiki, the 'powertrace.h' file is added to our simulation in order to gain time measurements for different modes, including, transmission, reception, and active and idle CPU times for each node during simulation. The values in Figure 6 belong to a transmission or reception of one packet or the radio off states during one round of broadcasting. Figure 6 shows that a mote consumes more power in its receiving mode.

Figure 7 demonstrates the localization error in terms of distance from the real location (eq. 4) and the percentage of localized nodes after 6 rounds of flooding. The total number of nodes is 80, but the number of anchors varies between 20 to 50. By increasing the number of anchors error decreases and higher percentage of blind nodes are localized.

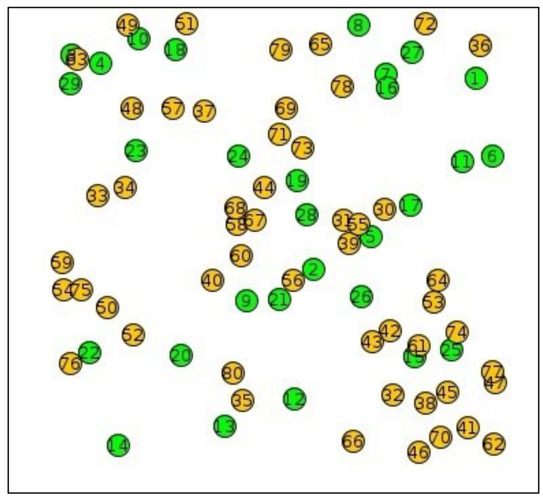

Figure 3. A random network topology including 50 anchors and 30 blind nodes

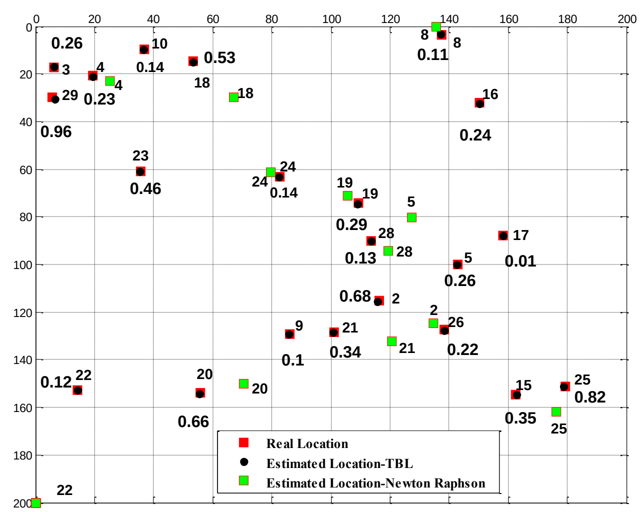

Figure 4. Real and estimated locations after localization

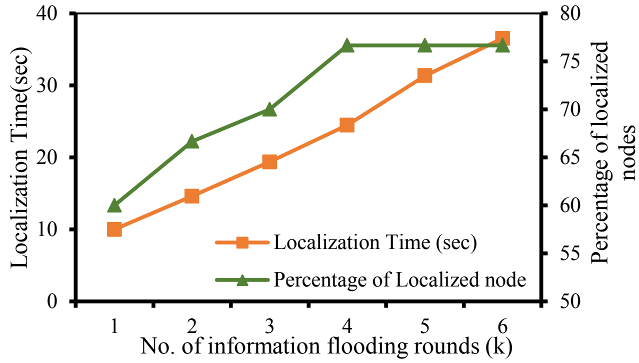

Figure 5. Examination of the effect of k on localization time and the number of localized nodes

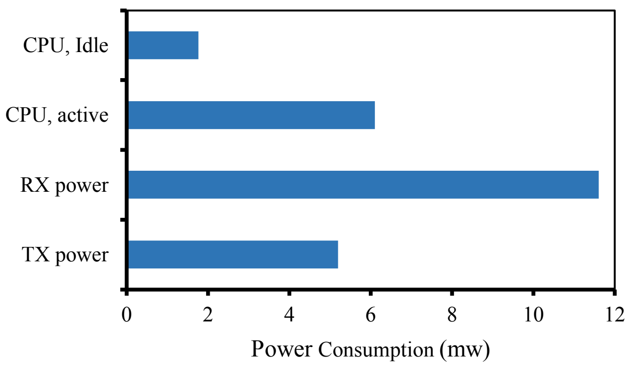

Figure 6. The estimated average power consumption of each node running the Contiki localization simulation

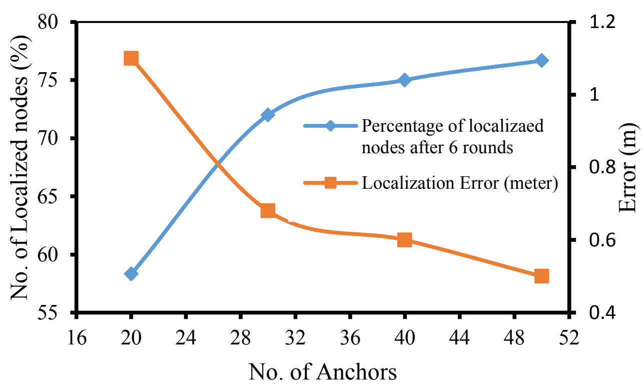

Figure 7. The effect of increasing the no. of anchor nodes on the localization error and the total no. of localized nodes

CONCLUSION

In this paper, the applicability of the time bounded essential localization method was examined by simulation using the Cooja simulator for Contiki-OS, which is comparable to real environments. Two algorithms are proposed for blind and anchor nodes to perform time bounded localization. The proper number of information flooding rounds has been chosen to get accurate localization for the highest possible number of localized nodes in the network in a limited time. Applying TBL can lead to the limiting of power consumption, particularly as a result of pre-defining the number of information flooding rounds. According to the simulation results for a random network topology, after 6 rounds of flooding, around 76% of the nodes are localized with an average error of 0.5 meters in a 200 m by 200 m area.

ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under Grant No. CNS-1248381.

REFERENCES

1. Akyildiz, I.F., et al., Wireless sensor networks: a survey. *Computer Networks,* 2002. 38(4): p. 393-422.
2. Alrajeh, N.A., M. Bashir, and B. Shams, Localization Techniques in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, (2013), 9 pages.
3. Aspnes, J., et al., A Theory of Network Localization. *IEEE Trans. on Mobile Computing,* (2006), 1663-1678.
4. Boukerche, A., et al., Localization systems for wireless sensor networks. *IEEE Wireless Communications*, (2007), 6-12.
5. Cheng, W., et al., Time-Bounded Essential Localization for Wireless Sensor Networks. *IEEE-ACM Trans. on Networking,* (2013), 400-412.
6. Dunkels, A., B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, (2004), 455-462.
7. Dunkels, A., et al., Protothreads: simplifying event-driven programming of memory-constrained embedded systems. In *Proc. of the 4th international conference on Embedded networked sensor systems*, ACM, (2006), 29-42.
8. Dunkels, A., Full TCP/IP for 8-bit architectures. In *Proc. of the 1st international conference on Mobile systems, applications and services*, (2003) ACM, 85-98.
9. Dunkels, A., F. sterlind, and Z. He, An adaptive communication architecture for wireless sensor networks. In *Proc. of the international conference on Embedded networked sensor systems*, ACM, (2007), 335-349.
10. Dunkels, A., The ContikiMAC Radio Duty Cycling Protocol. *Swedish Institute of Computer Science*, (2011).
11. Inc, T.I., CC 2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transciever Applications. *Technical Report*, (2014).
12. Johnson, M., et al. A comparative review of wireless sensor network mote technologies. In *Sensors, 2009 IEEE*, (2009), 1439-1442.
13. Mao, G.Q., B. Fidan, and B.D.O. Anderson, Wireless sensor network localization techniques. *Computer Networks,* (2007) 2529-2553.

14. Osterlind, F., et al. Cross-Level Sensor Network Simulation with COOJA. In *Local Computer Networks, Proc. 2006 31st IEEE Conference on*, (2006), 641-648.
15. Pandey, S. and P. Agrawal, A survey on localization techniques for wireless networks. *Journal of the Chinese Institute of Engineers,* (2006), 1125-1148.
16. Rappaport, T., Wireless Communications: Principles and Practice. 2001, Prentice Hall PTR. 736.
17. Yeong-Sheng, C., C. Tai-Lin, and H. Yi-Chen. Collaborative localization in Wireless Sensor Networks based on dependable RSSI. In *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, (2012), 341-347.
18. Chen, C., and T., Chun Lin. A Low-Cost Anchor Placement Strategy for Range-Free Localization Problems in Wireless Sensor Networks. In *International Journal of Distributed Sensor Networks*, (2013), 12 pages.