Copyright © 2012 KSII

# EBCO - Efficient Boundary Detection and Tracking Continuous Objects in WSNs

**Sajjad Hussain Chauhdary[1], Jeongjoon Lee[2], Sayed Chhattan Shah[3] and Myong-Soon Park[1]**
[1]Department of Computer Science and Engineering
Korea University, Seoul –South Korea
[2]Convergence Technology R&D Center
LS Industrial Systems Co, LTD. Anyang- 431-080, South Korea
[3]Vehicle and Defense IT Convergence Research Division
(ETRI) Electronics and Telecommunications Research Institute Daejeon- 305-700, South Korea
[E-mail: sajjad@korea.ac.kr, myongsp@korea.ac.kr,: jjlee@lsis.biz, shah@etri.re.kr ]
*Corresponding author: Myong-Soon Park

*Received May 29, 2012; revised August 21, 2012; revised October 1, 2012; accepted October 8, 2012; published November 30, 201*

## Abstract

Recent research in MEMS (Micro-Electro-Mechanical Systems) and wireless communication has enabled tracking of continuous objects, including fires, nuclear explosions and bio-chemical material diffusions. This paper proposes an energy-efficient scheme that detects and tracks different dynamic shapes of a continuous object (i.e., the inner and outer boundaries of a continuous object). EBCO (**E**fficient **B**oundary detection and tracking of **C**ontinuous **O**bjects in WSNs) exploits the sensing capabilities of sensor nodes by automatically adjusting the sensing range to be either a boundary sensor node or not, instead of communicating to its neighboring sensor nodes because radio communication consumes more energy than adjusting the sensing range. The proposed scheme not only increases the tracking accuracy by choosing the bordering boundary sensor nodes on the phenomenon edge, but it also minimizes the power consumption by having little communication among sensor nodes. The simulation result shows that our proposed scheme minimizes the energy consumption and achieves more precise tracking results than existing approaches.

***Keywords:*** Wireless sensor network, Continuous Object Tracking, Boundary Detection of Phenomenon

**http://dx.doi.org/10.3837/tiis.2012.10.009**

## 1. Introduction

Wireless sensor networks are projected to be parts of key infrastructures in many applications, including crop monitoring, security surveillance, intelligent highway systems and emergent disaster response systems [1][2][3]. Due to the sensor's ability to detect and monitor at anytime and anywhere for little cost, sensor networks have great potential to be used in many domains where continuous monitoring and detection is needed, e.g., monitoring fires, nuclear explosions and hazardous bio-chemical diffusions [4][5][6][7][8]. Many sensor nodes are available on the market for gas detection **(wasp mote)** [12] and monitoring **(IRIS and MICA)** [13]. It is important to know the overall shape of such a spatial phenomenon. If we know its shape, size, or movement, we can understand its possible damage and take counter-measures to minimize this damage. It is necessary to understand the evolving circumstances of the phenomenon, which will help us understand the topic in greater detail. These possible situations are described as follows.

I. A phenomenon can expand in size and emerge into a new shape.
II. A phenomenon can shrink in size and emerge into a new shape.
III. Two or more phenomena can merge into one and emerge into a new shape.
IV. A phenomenon can split into two or more phenomena.
V. A phenomenon can have one or more holes inside.
VI. Holes inside a phenomenon can shrink, expand, merge, and split.

**Fig. 1** shows the possible evolving situation of a phenomenon (continuous object). **Fig. 1 (I) Phenomenon Expansion** shows that a phenomenon can expand on all sides of the boundary or, as in **Fig. 1 (II)**, some sides of the boundary remain static while expansion occurs on the other sides. **Fig. 1 (III) Phenomenon Shrinkage** shows that a phenomenon can shrink on all sides of the boundary or on some edges, while the other sides of the boundary remain static at the same time. **Fig. 1 (IV) Holes inside Phenomenon** shows a hole inside a phenomenon: there can be one or more holes at a time, and two or more holes can merge into one hole or one hole can split into two or more holes. These holes can also expand or shrink. **Fig. 1 (V) Phenomenon Splitting** shows one phenomenon splitting into two phenomena. **Fig. 1 (VI) Phenomenon Merging** shows two phenomena merging into one. In **Fig. 1**, issues relevant to **(I), (II), (III), (V) and (VI)** can be expressed (e.g., expanding, shrinking, merging, splitting) relevant to the outer boundary, and those related to **(IV)** are inner boundary-relevant.As **Fig. 1** shows, a phenomenon (continuous object) covers a large region in the network, and it usually requires an excessive amount of message exchanges among sensor nodes to collaboratively estimate the objects' movement, shape and location information. Excessive message exchanges among sensor nodes consume a huge amount of energy, which is a critical issue due to the battery power limitation of WSNs.
Accurate and low-cost continuous object tracking is a critical requirement in wireless sensor networks. Existing research efforts on continuous object tracking can be categorized into two parts: (1) Precise Tracking and Detection and (2) Energy Efficient Tracking and Detection. The first category addresses accurately estimating the boundary of the continuous object. DCSC [4] dynamically groups boundary sensor nodes into clusters, and a cluster head collects the boundary information from all boundary sensor nodes in the cluster and transmits this information to the base station. After the base station receives boundary information from each

dynamic cluster, it processes the global boundary formation. Dynamic cluster formation requires significant energy due to excessive data exchange and causes delays. The second category discusses energy efficiency to decrease the communication cost. In DEMOCO [6], sensor nodes communicate with each other to identify boundary nodes, thus consuming a significant amount of energy. Although this scenario reduces energy by selecting a Representing Node (RN) of boundary nodes to report to the sink node, it compromises boundary precision. In our study, we successfully handle the trade-off between accuracy and energy efficiency.

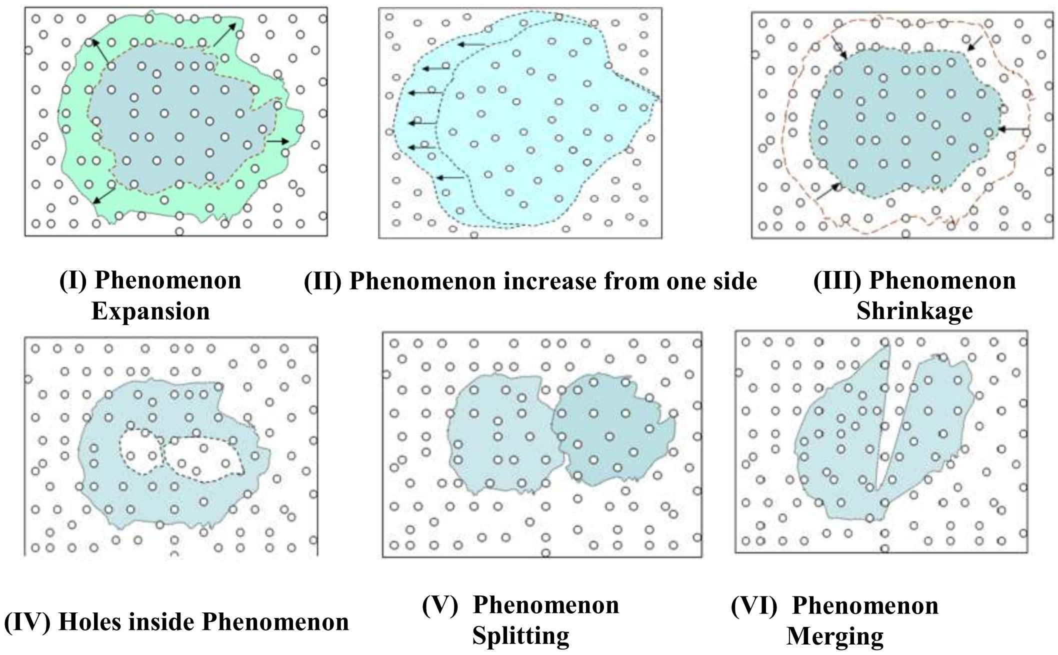


**Fig. 1.** Evolving Phenomenon Shapes

Our study focuses on scheming communication costs, which directly impact the network energy source. We effectively adjust the sensing range of sensor nodes to find the precise boundary nodes of continuous objects and minimize power consumption by controlling communication costs. The proposed scheme contains three processes: Outer Boundary Detection, Inner Boundary Detection and Representing Sensor Node Selection. These processes track the object movement and identify the boundary nodes of the phenomenon. Our approach identifies boundary nodes by automatically adjusting the sensing range of sensor nodes instead of communicating with neighboring sensor nodes. Avoiding data exchanges between sensor nodes during boundary node identification can significantly save energy. Our work makes the following contributions:1) Prolong network life by controlling communications among candidate boundary nodes, 2) increasing overall shape precision by detecting inner boundary nodes, outer boundary nodes near the phenomenon edge and the differences between them.

The rest of the paper is organized as follows. Section 2 discusses the related work. In section 3, we present the system model and assumptions. In section 4, we describe the proposed approach in detail. We analyze our idea and compare it to previous work using a simulation in section 5. In section 6, we conclude this paper.

## 2. Related Work

Significant research on single or multiple object detection and tracking has been performed with wireless sensor networks [2][3][14]. EOATR [2] tracks objects by automatically adjusting the sensor nodes' transmission range according to the speed of objects, with a better trade-off between tracking accuracy and energy efficiency. However, single or multiple objects are quite different from continuous objects. In the case of single or multiple objects, the shape and appearance of the objects are defined before the tracking time. Two works [10] and [11] analyze the detection of non-local events, which are closely related to continuous object detection and tracking. The main difference is that these references do not examine the situation in which phenomena change their shapes in real time.

In this techniques [17][18] the goal is to approximate the boundary by a polygon. The sensors use the sensed information to place interpolation points that define the polygon estimate. These points are distributed uniformly along the boundary and the boundary is approximated by the polygon formed by connecting the interpolated points. The approach assumes the knowledge of local tangent and curvature to the boundary and every sensor exchanges this information with other sensors in a ring-topology, which basis lots of communication. The sensors update and interpolate points on the polygon representing the boundary and move along the interpolated points. These techniques use mobile sensors, which is not cost effect to implement in real-time.

Other studies have focused on continuous object tracking [4][5][6][8][9] because of its possible potential applications, including detecting fire flames, nuclear explosions, oil spills and hazardous bio-chemical material diffusions. DCSC [4] proposes a dynamic cluster structure to detect and track a continuous object. When a sensor node senses the appearance of the object, it communicates with its one-hop neighboring nodes to learn their object detections. If the sensor node receives at least one different detection status from any neighbors, the sensor node becomes a boundary node. After the boundary node selection, the cluster formation process takes place among the boundary nodes. However, the cluster formation is not described clearly [4]. We argue that clustering is not an appropriate approach when the goal of forming clusters is to save energy because it requires much communication and causes delays.

In [15] algorithm is motivated by a study that holes in a sensor field create abnormality in hop count distances. Basically, in a shortest path tree rooted at one node, each hole is "hugged" by the paths in a shortest path tree. We identify the "cut", the set of nodes where shortest paths of distinct homotopy types terminate and touch each other, trapping the holes between them. The nodes in a cut can be easily recognized, since they have the property that their common ancestor in the shortest path tree is reasonably far away, at the other side of the hole. The detection of nodes in a cut can be performed independently and locally at each pair of adjacent nodes. The algorithm is based on flooding procedures. The algorithm is effective to find the holes and boundaries of a static network or a static phenomenon. If the object is continuous phenomenon and changing its shape over the time, flooding of messages can't be effective, because it consume much energy for each time of flooding.

CODA [5] proposes a hybrid clustering technique for continuous object tracking. At the time of network deployment, static clusters are formed. In a static cluster, all nodes that detect an object transmit their detection information to the cluster head, thus forming boundary nodes in a dynamic cluster. In this approach, boundary nodes are identified by the cluster heads of static clusters rather than a message exchange process among the local sensors. This process requires substantial communication, especially when the object overlays many nodes. Though this approach uses the static cluster head as a dynamic cluster to avoid the energy cost of selecting new cluster head, it still costs too much energy to group these nodes into a dynamic cluster.

DEMOCO [6] propose an energy-efficient algorithm to detect and tracking continuous objects. The sensor nodes send a message that includes their current object detection to their neighboring nodes, which detect the emergence or disappearance of the object. The sensor nodes that have received the message and have different object detections than that included in the message are boundary nodes. Boundary nodes are assigned different back-off times, which are inversely proportional to the number of received messages. Boundary nodes with short back-off times send data to the sink node and suppress messages from nearby boundary nodes as RNs. Only RNs send data to the base station, so the base station has less information to build phenomenon boundaries. The boundary precision is thus comparatively less than our approach and that of DCSC [4], in which the base station receives information from almost every boundary node. DEMOCO is not cluster-based and reduces the communication consumption and message size, but the boundary node identification process needs much communication among neighboring sensor nodes.

LEDTP [16] shows the improvement of TOCOB [8], but it is based on clustering and manages the array of boundary nodes and non boundary nodes. Whereas clustering itself cause of more energy consumption and it requires processing cost.
The ECOT sensor node [9] itself identifies whether it is a boundary node by automatically adjusting its sensing range instead of communicating with neighboring nodes because radio communications consume more energy than adjusting sensing range. Furthermore, only a subset of boundary nodes sends data to the sink node with a small message. Finally, the sink node can estimate the object's position and shape by locating object mechanism. The proposed scheme [9] does not consider the inner boundary issue of the continuous object and cannot distinguish a difference between the inner and outer boundaries, which affects the overall shape reorganization.
This study extends a previous work [9], the proposed scheme differentiates between an outer and inner boundary, which leads to more precise boundary detection than a previous method [9]. Our approach further handles the inner phenomenon boundary (i.e., a hole inside a phenomenon).

## 3. Preliminaries

In this section, we define the sensor modes the basic assumptions and definitions to ensure that our approach functions.

### 3.1. Assumptions

- Sensor nodes are randomly deployed in the area of interest.
- Sensor nodes have identical capabilities, e.g., sensing, energy, and computation.

- The sink node knows every node's ID and position.
- Each sensor node can automatically adjust its sensing range [7].

### 3.2. Modes of Sensor

A sensor node has three major components: (a) Sensor (b) CPU and (c) Transceiver. We define modes of sensor by keeping in mind these three major components of the sensor as follows.

***Idle Mode****:* In idle mode, three major components remain off, but the sensor component periodically turns on and makes local observations.
***Sensing Mode:*** In sensing mode, the Sensor and CPU components are turned on, and the Transceiver component remains off.
***Communication Mode:*** In communication mode, the Sensor component remains off, but the CPU and Transceiver components are turned on.

### 3.3. Definitions

**Candidate Boundary Node:** A sensor node can become a boundary node by adjusting its sensing range. There are two types of candidate boundary nodes:

**A: Undetected-to-detect Node (UTDN):** A node that could not detect the object in the previous time slot $P_{i-1}$ but can detect it in the current time slot $P_i$.

**B: Detect-to-undetected Node (DTUN)**: A node that could detect the object in the previous time slot $P_{i-1}$ but cannot detect it in the current time slot $P_i$.

**Boundary Node (BN):** A node that changes its detection status during the current time slot by changing its sensing range.

**Representative node (RN):** A node that sends a report to the sink node, called an RN, and represents other neighbor boundary nodes.

**Setting of Sensing Range:** We set a 30 meter radius (r) sensing range as the default sensing range for a sensor node and assume that a sensor node can increase its sensing range up to 60 meters and decrease it to 5 meters (r). This sensing range setting can be set according to specific application requirements, i.e., depending on object type.

## 4. Boundary Detection Of Continuous Object (Phenomenon)

In this section, we describe the proposed scheme "energy-efficiency continuous object tracking protocol (EBCO)". As shown in **Fig. 2** over all detection process in steps, which are describe as follows.

**Step 1:** The sensor node wakes up and periodically makes local observations.

- If there is no change in sensor node detection, the sensor node returns to its idle state.
- If a sensor node detects a different detection in sensing, go to step 2.

**Step 2:** The sensor node detects a different detection.

To understand the 2nd step, we have divided it in three major parts: Outer Boundary Detection, Inner Boundary Detection and the difference between the inner and outer boundaries, as in Fig. 2. In step 2, sensor nodes begin Sensing Mode and leave Idle Mode.

## 4.1 Outer Boundary Detection

**Case 1: Undetected-to-detect Node (UTDN):** This case occurs when a sensor node did not detect the phenomenon in the previous time slot Pi-1 but detects it in the current time slot, Pi. This change is an Undetected-to-detect Node (UTDN) outer boundary expansion. A sensor node records its radio signal strength for phenomenon detection. Signal strength is recorded using a 1 to 10 scale, where 10 is considered the highest scale level and 1 is the lowest. The node then decreases its default sensing range (DSR) to a predefined threshold. After decreasing the sensing range to this predefined threshold, a sensor identifies itself as a boundary node if it could not detect the phenomenon. If it still detects the phenomenon, then the sensor node is either inside the phenomenon or is close enough to the phenomenon to not be boundary node; it then goes to idle mode. All sensor nodes in this process tag themselves as PEs (Phenomenon Expansions) until the end of the next time period. A tagging process is used to differentiate between inner and outer boundary nodes, as explained in detail in section 4.3. The right side of Fig.3 shows that when a phenomenon expands, the sensor nodes that have not detected the phenomenon in a previous time period record their signal strengths to detect the phenomenon and begin to decrease their sensing ranges to a predefined threshold to become BNs. If a sensor node cannot detect the phenomenon after decreasing the sensing range, it declares itself a BN. Algorithm 1 expresses the UTDN procedure.

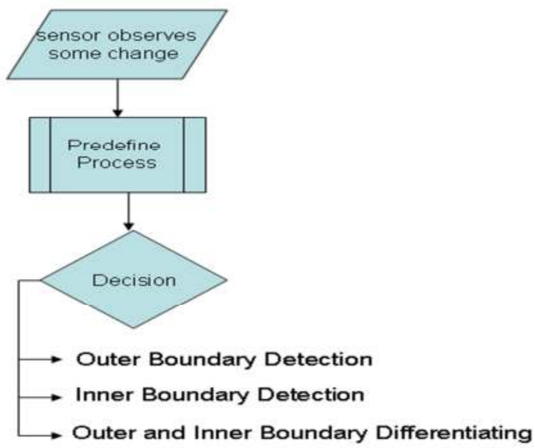


**Fig. 2.** Overall boundary detection process

Algorithm 1 starts with basic parameters: DSR, Default Sensing Range; PSR, Previous Sensor's Reading; CSR, Current Sensor's Reading; and N Tag, Node's tag status. If the CSR is 1 (i.e., it can currently detect the phenomenon), the PSR is 0 (i.e., it could not detect the phenomenon in the last time) and the N tag is none, which means it is a Case 1: UTDN process. The sensor node records its signal strength to detect the phenomenon and decreases its DSR to a predefined threshold. After decreasing its sensing range, if a sensor node cannot detect the phenomenon again, CSR=0, and it becomes a BN. If a sensor node detects the phenomenon again, the sensor node is still inside the phenomenon or close enough to not be a BN. At the end of this process, the sensor node keeps its current timestamp, records its signal strength and tags itself as a PE (Phenomenon Expansion) until the next time period.

**Case 2: Detect-to-undetected Node (DTUN):** This case occurs when a node detects the phenomenon in the previous time slot Pi-1 but cannot detect it in the current time slot Pi. This change is the detected-to-undetected node (DTUN); the node has a PE tag, and its shrinkage of outer boundary. The sensor node increases its SR to a predefined threshold. If it detects the phenomenon again, it identifies itself as a boundary node and records the signal strength of the phenomenon detection. Recording the signal strength aims to select the boundary node closest to the phenomenon, which helps to obtain a better boundary shape at the end. We discuss the detailed procedure to select the closest boundary node as an RN in section 5. If a sensor node cannot detect the phenomena, it is far from the phenomenon or not close to detects the phenomenon; it returns to idle mode until the next time period. The boundary node broadcasts to the one-hop nodes to tag them as PDs (Phenomenon Decreases) until the end of the next time period, keeps the recoded signal strength, and maintains its default sensing range. The left side of **Fig. 3** shows that when the phenomenon shrinks, the sensor nodes that have detected the phenomenon in the last time period start to increase their SRs to a predefined threshold to become BN. If a sensor node detects the phenomenon after increasing its SR, it declares itself a BN. Algorithm 2 explains the DTUN procedure.

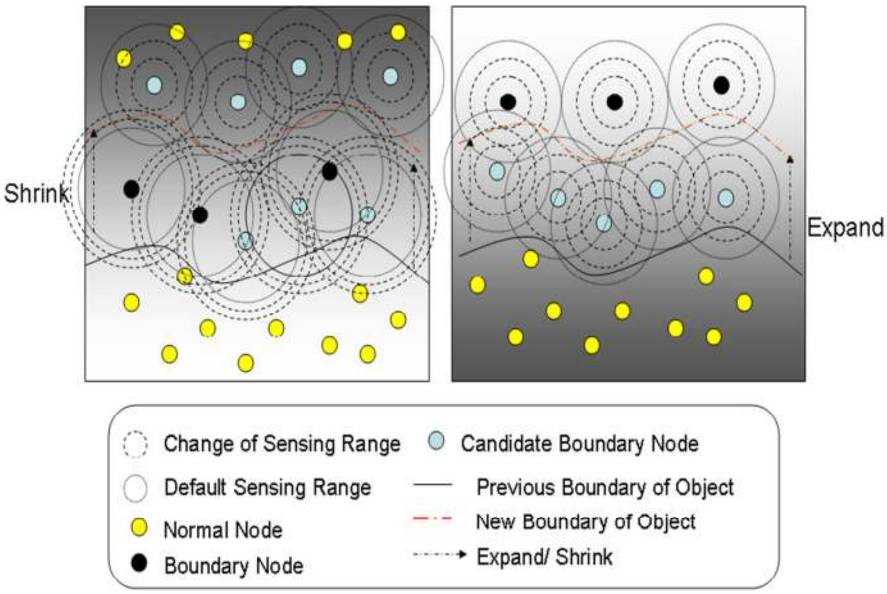


**Fig. 3.** Boundary Node identification during phenomenon shrinking and expanding

**Algorithm 1: Case 1: Undetected-to-detect Node (UTDN)**

**Parameters:**
*DSR*: Default Sensing Range;
*PSR*: Previous Sensor's Reading;
*CSR*: Current Sensor's Reading;
*N Tag*: Node Tag Status
*SR:* Sensing Range

00 **Input** *DSR, PSV, CSV*; and *N Tag*
01 **if** *CSR=1, PSR =0, DSR=30 m and N Tag =0* **then** { // Undetected to Detect Node-Expansion of outer

```
boundary
Rec  current RSSI   // Record radio signal strength of  phenomenon detection from (1~ 10) scale level as predefined
     while (DSR --   until SR= 1/3 * DSR) {   //  decrease  DSR to predefine threshold to identify BN
                               // Boundary node
          input CSR;
          if CSR =0 then {  //sensor's reading changes
             The node is identified as boundary node;
             Keep current timestamp, record signal strength and tag itself  as PE (Phenomenon Expand)
             Return;
          }
     } // end of while
} else  {
set  SR= DSR //  node sets its sensing  as  DSR (default sensing range )
     Node enters sleep mode;
}
```

Algorithm 2 initiates with basic parameters: DSR, Default Sensing Range; PSR, Previous Sensor's Reading; CSR, Current Sensor's Reading; and Tag Status. If the CSR is 0 (i.e., it can detect the phenomenon in the current time), the PSR is 1 (i.e., it could not detect the phenomenon in the last time), and the N Tag is PE (Phenomenon Expands), it is a Case 2: DTUN process, and the sensor node increases its DSR to a predefined threshold. After increasing its SR, the sensor node becomes a BN if it can detect the phenomenon and records its signal strength detection. If the sensor node does not detect the phenomenon, it is far from the phenomenon or not close enough not to be a BN. At the end of this process, the sensor node keeps the current timestamp, records its signal strength and tags itself as a PS (Phenomenon Shrinkage) until the next time period.

**Algorithm 2: Outer Boundary Detection Case 2: Detect-to-undetected Node (DTUN)**

```
Parameters:
DSR: Default Sensing Range;
PSR: Previous Sensor's Reading;
CSR: Current Sensor's  Reading;
N Tag: Node Tag Status
SR: Sensing Range
Input DSR, PSR, CSR; PE
if  CSR=0, TAG= PE, PSR =1 and N Tag = PE then { // Detect to Undetected
Rec  current RSSI   // Record radio signal strength of  phenomenon detection from (1~ 10) scale level as predefined
     while ( DSR ++   until SR>=2*DSR) {  // increase DSR to identify boundary node until predefined threshold

          input CSR;
          if CSR=1 then {  //sensory value changes
             The node is identified as boundary node;
     Rec RSSI >0 < 10  // Record radio signal strength or current sensor reading
             Keep current timestamp and TAG itself as PS (Phenomenon Shrinkage )
             Return;
          }
     } // end of while
} else  {
set  SR= DSR //  node sets its sensing  as  DSR (default sensing range )
     Node enters sleep mode;
}
```

## 4.2 Inner Boundary Detection

**Case 1: Detect-to-undetected Node (DTUN):** If the change is detected-to-undetected Node (DTUN) and the node has no tag, this change represents the appearance or expansion of a hole inside the phenomenon, and the sensor node increases its SR to a predefined threshold. If the sensor detects the phenomenon, then it identifies itself as a BN of the hole and records its signal strength to detect the phenomenon. If it could not detect the phenomena, the node is far from the phenomenon or not closes enough to be a BN; it returns to idle mode. All nodes in this process tag themselves as HEs (Hole Expansions) until they obtain new readings and maintain their default sensing range. Fig. 4 explains the DTUN situation for internal boundaries, where the black node becomes a BN after following the procedure in algorithm 3.

Algorithm 3 commences with basic parameters: DSR, Default Sensing Range; PSR, Previous Sensor's Reading; CSR, Current Sensor's Reading; and N Tag: Node Status. If the CSR is 0 (i.e., it can detect the phenomenon in the current time), the PSR is 1 (i.e., it could not detect the phenomenon in the last time), and Node Tag is none, it is a Case 1: DTUN process, and the sensor node increases its DSR to a predefined threshold. After increasing its sensing range, a sensor node becomes a BN for the inner boundary if it can detect the phenomenon, and the sensor node records its signal strength detection. If the sensor node does not detect the phenomenon, it is far from the phenomenon or not close enough not to be a BN. At the end of this process, the sensor node keeps the current timestamp, records its signal strength and tags itself as an HE (Hole Expansion) until the next time period**.**

**Algorithm 3: Inner Boundary Detection Case 1: Detect-to-undetected Node (DTUN)**

```
Parameters:
DSR: Default Sensing Range;
PSR: Previous Sensor's Reading;
CSR: Current Sensor's  Reading;
N Tag: Node Tag Status
SR: Sensing Range
Input DSR, PSR, CSR; N TAG
if  CSR=0, TAG= 0, PSR =1 and NTag =none then { // Detect to Undetected
Rec  current RSSI    // Record radio signal strength of  phenomenon detection from (1~ 10) scale level as predefined

    while DSR ++ ( SR=DSR*2{  // increase DSR to identify boundary node until predefine threshold
        input CSR;
        if CSR=1 then {  //sensory value changes
            The node is identified as boundary node;
            Keep current timestamp and TAG itself as HE (Hole Expansion )
            Return;
        }
    } // end of while
} else  {
set  SR= DSR //  node sets its sensing  as  DSR (default sensing range )
    Node enters sleep mode;
}
```

**Case 2: Undetected-to-detected Node (UTDN):** If the change is Undetected-to-detected Node

(UTDN) and was tagged as HE in the previous time slot, this result shows that the hole shrinks. The sensor node records its signal strength to detect the phenomenon and decreases its SR to a predefined threshold. If it does not detect the phenomenon, then it identifies itself as a hole BN. If it could detect the phenomenon, the node returns to the phenomenon, deletes its HE tag, goes to idle mode, and maintains the default sensing range. Fig. 4 explains the situation when the hole shrinks, and algorithm 4 explains the UTDN procedure.

Algorithm 4 begins with basic parameters: DSR, Default Sensing Range; PSR, Previous Sensor's Reading; CSR, Current Sensor's Reading and N Tag. If the CSR is 1 (i.e., it can detect the phenomenon in the current time), the PSR is 0 (i.e., it could not detect the phenomenon in last time) and the Node tag is HE, it is a Case 2: UTDN process. The sensor node records its signal strength detection and decreases its DSR to a predefined threshold. After decreasing its sensing range, a sensor becomes a BN for a potential inner boundary if it cannot detect the phenomenon. If a sensor node detects the phenomenon, it is back inside the phenomenon, or it is close enough to phenomenon to not be a BN. At the end of this process, the sensor node keeps the current timestamp and records its signal strength until the next time period.

**Algorithm 4**: **Inner Boundary Detection Case 2: Undetected-to-detect Node (UTDN)**

**Parameters**:
*DSR*: Default Sensing Range;
*PSR*: Previous Sensor's Reading;
*CSR*: Current Sensor's Reading;
*N Tag*: Node Tag Status
*SR:* Sensing Range

```
Input DSR, PSR, CSR; N TAG
if  CSR=1;  Tag = HE and PSR =0 then { // Undetected to detect
Rec  current RSSI    // Record radio signal strength of  phenomenon detection from (1~ 10) scale
level as predefined

     while (DSR --   until SR= 1/3 * DSR) {   //  decrease  DSR to predefine threshold to identify BN
                                    // Boundary node
          input CSR;
          if CSR =0 then {  //sensor's reading changes Node is identified as boundary node;
               Keep current timestamp and delete HE Tag
             Return;
          }
     } // end of while
} else  {
     Node enters sleep mode;
set  SR= DSR //  node sets its sensing  as  DSR (default sensing range )

}
```

### 4.3 Outer and Inner Boundary Differentiating

In this section, we discuss how to differentiate between inner and outer boundaries. There are two probable situations of such cases.

**Situation A: Undetected-to-detected Node (UTDN):** For UTDN, there are can be two possibilities: phenomenon expansion or hole shrinkage inside the phenomenon. A node itself cannot observe this difference. DEMOCO [6] and ECOT [9] also do not confirm this difference in their approaches. We assume that a hole inside a phenomenon does not shrink before its appearance and expansion, which self-evident. When node v did not detect the phenomenon in the previous time slot Pi-1 but detects it in the current time slot P and has an HE tag, the node was previously part of the process to find HE BNs and tags itself as HE, as explained in section 4.2. When node v did not detect the phenomenon in the previous time slot Pi-1 but detects in the current time slot Pi and there is no tag, this change represents phenomenon expansion, and all nodes involved in this process tag themselves as PEs at the end of this process. Our scheme clearly differentiates between phenomenon expansion and hole shrinkage inside the phenomenon. The base station clearly differentiates between the inner and outer boundaries using reports from the RNs.

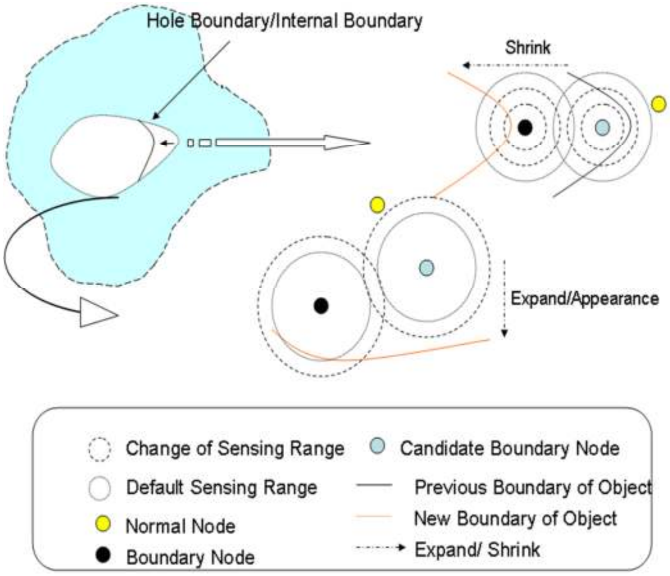


**Fig. 4.** Inner boundary node detection of phenomenon

**Situation B: Detect-to-undetected Node (DTUN):** DTUN also has two possibilities: phenomenon shrinkage or hole expansion inside the phenomenon. A node itself cannot observe this difference. We again assume that a phenomenon does not shrink before its appearance and expansion. When node v detected the phenomenon in the previous time slot Pi-1 but does not detect in the current time slot P and has a PE (Phenomenon Expansion) tag, the node was previously involved in the process to find PE BNs. Section 4.2 describes how a node tags itself as a PE. When node v detected the phenomenon in the previous time slot Pi-1 but does not detect in the current time slot Pi and there is no tag, this change represents the hole expansion inside the phenomenon, and all nodes involved in this process tag themselves as HE at the end of this process. Our proposed scheme clearly differentiates between phenomenon shrinkage and hole expansion inside the phenomenon. Using the RNs' reports, the base station can clearly differentiate between inner and outer boundaries. DEMOCO [6] and COBOM [9] cannot differentiate between inner and outer boundaries.

## 5. Representative Node Selection

The RNs are used to report to the base station or sink node. One RN represents two or more boundary nodes. EBCO uses the back-off time to select the RNs from the boundary nodes. All BNs record their phenomenon detection signal strengths. Depending on the recorded signal strength of each BN, they obtain a time (back-off time) to be an RN. BNs with higher signal strengths wait less time to be selected as RNs, and BNs with lower signal strengths wait longer. When the waiting time ends, a BN declares itself an RN and broadcasts a message to prevent its neighbors from becoming RNs. The BNs that receive the broadcast message reply with their IDs and tags to the relevant RN. The inclusion of tags in the report helps to differentiate between the inner and outer boundaries at the base station level. The report data include the RN's own ID as well as the neighboring BNs' IDs and tags.

## 6. Simulation Results And Analysis

In this section, we evaluate the performance of EBCO using simulation results. We compare our results to our previous work on DEMOCO [6] and ECOT [9]. We also compare our results to DCSC [4] as initial work in this research area.

We first consider holes inside objects, unlike DEMOCO [6], ECOT [9] and DCSC [4], which do not concern possible data loss or contention between nodes or how to route data to the sink. The simulations show that EBCO is usually more efficient than previous work because it tracks objects precisely with considerably lower communication costs.

### 6.1. Simulation Model

Sensor nodes are uniformly distributed over a field of 500 × 500 m2. We use MICA sensor motes for a simulation environment. In each experiment, 1500 sensor nodes are deployed arbitrarily in the field to simulate a sparse or dense setting. All sensor nodes have similar initial energy levels and signal ranges at the start of the simulation. NS2 is used for the simulation.

**Table 1**. Simulation Parameters

| Parameters | Default Setting |
|---|---|
| Terrain | 500 × 500 m2 |
| Number of Nodes | 1500 |
| Total time periods | 10 |
| Initiated point of low diffusion | (250, 250) |
| Low diffusion rate of phenomenon | 1~3 meter /each time slot |
| Initiated point of Average diffusion | (200,200) |
| Average diffusion rate of phenomenon | 3~7 meter / each time slot |
| Initiated point of Average diffusion | (150,150) |
| High diffusion rate of phenomenon | 7~11 meter / each time slot |

We set 3 different types of objects for the simulation: low, average and high diffusion rate objects. For the low diffusion rate object, which is initially centered at (250, 250), it continually expands by increasing its radius by 1 to 3 meters in each time slot randomly. The second object type has an average diffusion rate, initiates at (200, 200), and increases its radius by 3 to 7 meters in every time slot randomly. The third object type, with a high diffusion rate, initiates at (150, 150) and expands its radius by 7 to 11 meters in every time slot. We use 10 time slots with an average diffusion rate, and every time slot has same time period. All sensors in the network activate and make local observations in each time slot. Each time slot has three time periods, and RN selection process occurs at every two time slots.

### 6.2. Boundary Accuracy with Different Phenomena Diffusion Rates

In this part, we evaluate how the diffusion speed of continuous objects affects boundary accuracy. **Fig. 5.A** shows a low diffusion rate object that is initially centered at (250, 250), and it continually expands with its radius increasing by 3 meters in each time slot. The x-axis represents the number of time periods, and the y-axis shows the boundary accuracy in as percentage. In the DEMOCO and ECOT mechanisms, RNs of BNs report to the base station. DEMOCO has more RPs in its approach than ECOT, and TOCOB, thus showing more reports to the base station, which indicates a more accurate boundary node.

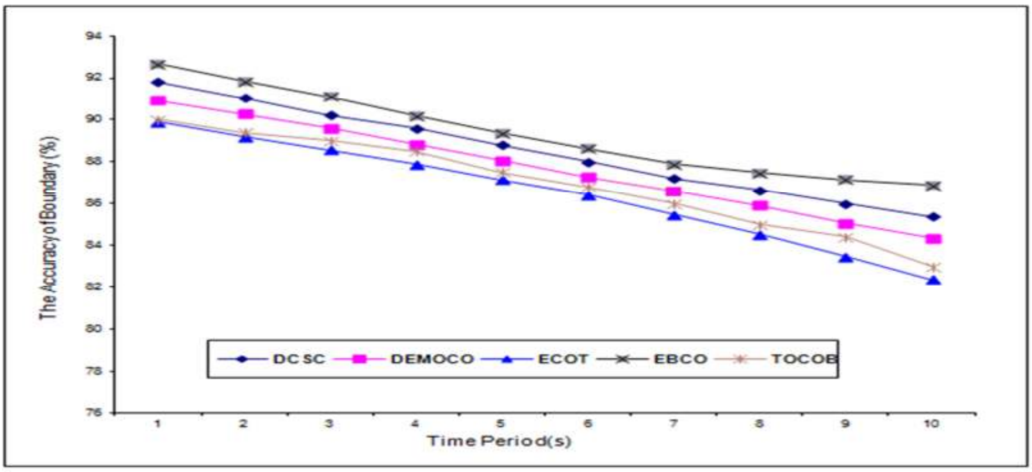


**Fig. 5.A.** Boundary accuracy with low phenomenon diffusion rate

In DCSC and TOCOB, each sensor node reports to cluster head and cluster head reports to the base station. In case of DCSC the BN, which is not a member of any cluster, reports directly to the base station. In TOCOB, it takes time to process the array and aggregation within the cluster, after aggregation and array process cluster head report back to base station, which affect the overall accuracy of boundary detection. The base station has information from more BNs than the base station in DEMOCO or ECOT, which makes the phenomenon boundaries more accurate in EBCO. In the EBCO approach, every BN reports to the RN, and the RN returns data to the base station. The base station thus has information from almost every BN, which makes the boundary shape more accurate than in DCSC, DEMOCO and EBCO. **Fig. 5.B** shows an average object diffusion rate, initiated at (200, 200), which expands with its radius increasing by 2 meters in every time slot. **Fig. 5.C**, with a high diffusion rate initiated at (150, 150), expands with its radius increasing by 3 meters in every time slot. In last scenario, where phenomenon expands with its radius increasing by 3 meters in every time slot, influence most to TOCOB, because of its clustering, array process and aggregation process. These processes cause much delay to send back to data to base station and in a given time slot it decreses boundary detection accuracy.

These results show that for low, average and high phenomenon diffusion rates, our approach provide a more precise boundary shape for the phenomenon because it differentiates between outer and inner boundaries. The base station also receives the IDs of every node from the RNs, so the base station has enough information to build a precise phenomenon boundary.

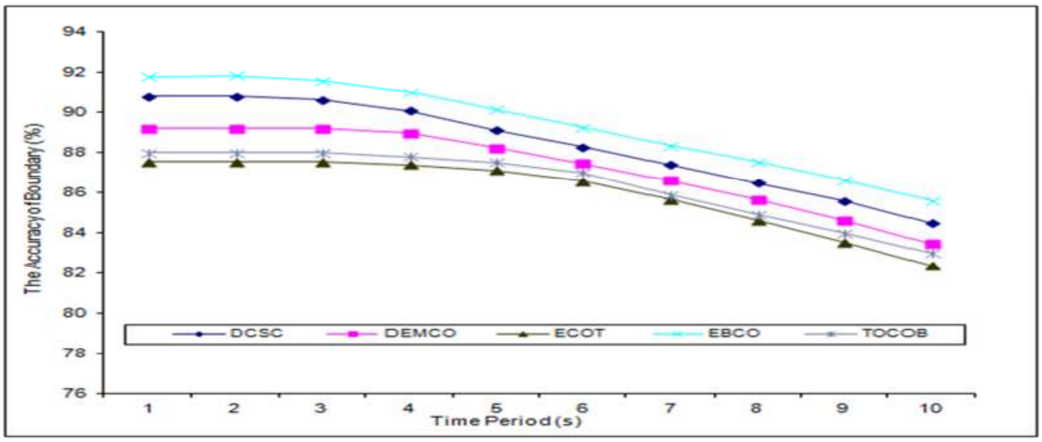


**Fig. 5.B.** Boundary accuracy with average phenomena diffusion rate

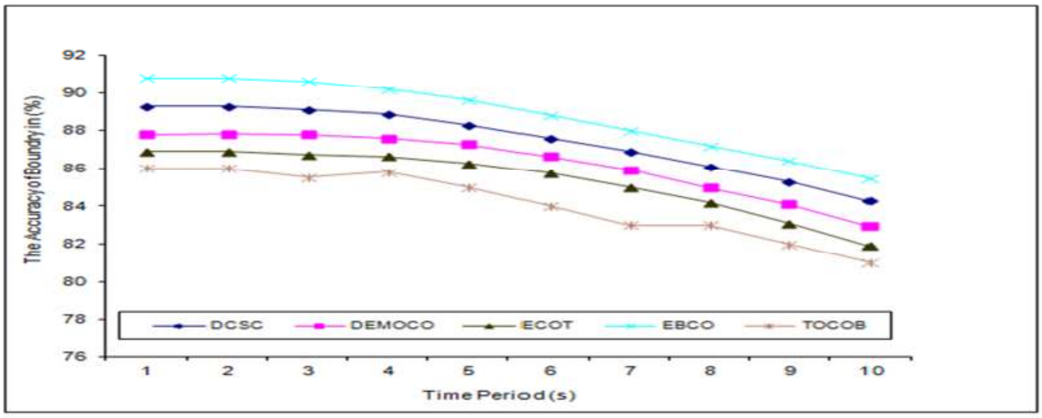


**Fig. 5.C.** Boundary accuracy with high phenomena diffusion rates

### 6.4. Comparing Overall Energy Consumption

**Fig. 6.** shows that when the object's movement status changes smoothly and its running time increases, the total energy consumption decreases compared to DCSC, DEMOCO, ECOT and TOCOB; as the running time increases, the power consumption ratio also increases because neighboring sensor nodes do not communicate with each other during boundary detection. In ECBO, candidate boundary nodes only control their sensing range, which consumes less power than the communication costs for neighbor nodes.

### 6.5. Comparing the Number of RNs

Comparing the number of RNs is important because RNs are responsible for reporting events to the sink node; however, the communication cost between the base station and RNs is outside the scope of this paper. The number of RNs is an important factor in energy consumption. DEMOCO, ECOT EBCO uses the same method to choose RNs, i.e., back-off time. DCSC and TOCOB uses their cluster heads as representing nodes. When the size of phenomenon increases the number of RNs increases. **Fig. 7.** shows that the number of RNs in DEMOCO is approximately 1.25 times greater than in EBCO and 1.1 times more than ECOT.

DEMOCO uses a complicated method to obtain BNs, which results in more RNs, but EBCO and ECOT have fewer BNs than DEMOCO. Whereas ECOT ignores the difference between outer and inner boundaries, EBCO efficiently differentiates between the inner and outer boundaries, which means that EBCO has more BNs than ECOT, which leads to more RNs. EBCO contains only 1.1 times more RNs than ECOT and 1.25 times fewer than DEMOCO. Particular results show a better trade-off than previous approaches.

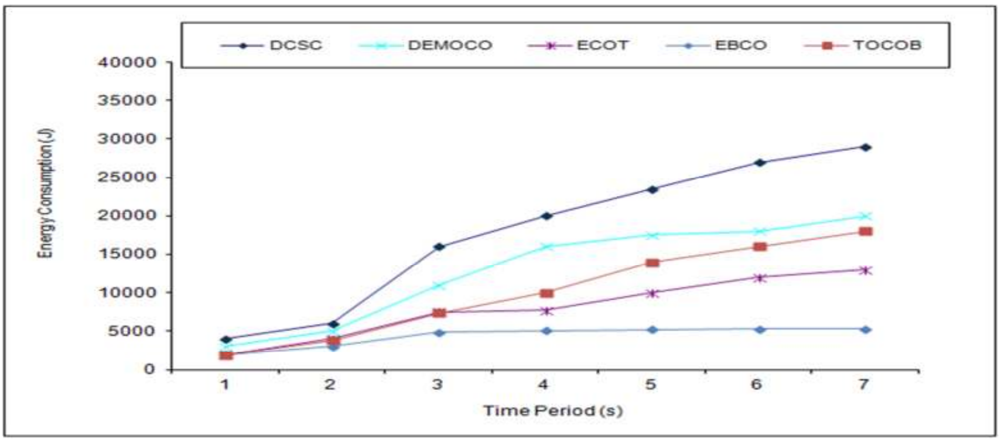


**Fig. 6.** Comparison of overall energy consumption

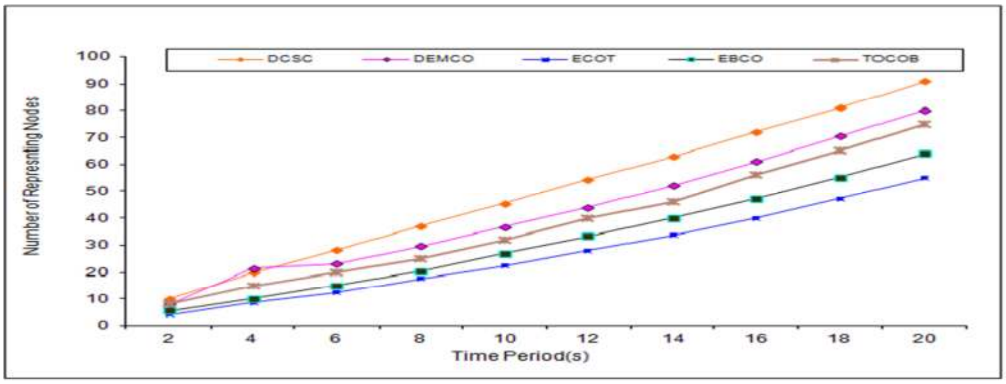


**Fig. 7.** Comparing the number of RPs

## 7. Conclusion And Future Work

In this paper, we propose EBCO – the Efficient Boundary Detection and Tracking of Continuous Objects in Wireless Sensor Network. EBCO effectively detects and tracks the boundary of a phenomenon when it expands or shrinks in size. It also efficiently differentiates between inner and outer boundaries. Using EBCO, we achieve precise continuous object tracking by adjusting the sensing range of the sensor nodes with low communication costs, which minimizes power consumption.

## References


[1] Akyildiz LF, Su WL, Sankarasubramaniam Y, Cayirc E. "A survey on sensor networks." *IEEE Communicatyion megazine*, vol. 8, pp.102-114, 2002. Article (CrossRef Link).

[2] Chauhdary SH, Bashir AK, Shah SC, Park MS, "EOATR: energy efficient object tracking by auto adjusting transmission range in wireless sensor network." *Journal Applied Sciences*, vol.9, pp.4247-4252, 2009. Article (CrossRef Link)

[3] Chauhdary SH, Bashir AK and Park MS, "ETCTR: Efficient target localization by controlling the transmission range in wireless sensor networks." *IEEE 4th International Conference on Networked Computing and Advanced Information Management*, vol.1, pp.3-7. 2008. Article (CrossRef Link).

[4] Xiang JI, Hongyuan Z, Metzner JJ and Kesidis G. "Dynamic cluster structure for object detection and tracking in ad-hoc sensor networks. *IEEE International Conference on Communication*, pp/3807-3811, 2004. Article (CrossRef Link).

[5] Wang RC, Hui TL and Zong ZC. "CODA: A continuous object detection and tracking algorithm for wireless ad hoc sensor networks." *IEEE Consumer Communications and Networking Conference* , pp.168-174, 2008. Article (CrossRef Link).

[6] Jung HK, Kee BK, Chauhdary SH, Wenchen Y and Myong SP. "DEMOCO: Energy-efficient detection and tracking for continuous objects in wireless sensor networks." *IEICE Trans. on Communication,* vol.11, pp.3648-3656, 2008.

[7] Dhawan A, Vu C, Zelikovsky A, Li Y, Prasad SK. "Lifetime of Sensor Networks with Adjustable Sensing Range. Software Engineering," *Artificial Intelligence, Networking, and Parallel/Distributed Computing,*pp. 285-289, 2006. Article (CrossRef Link).

[8] Jung HK, Kee BK, Chauhdary SH, Park MS." Energy-Efficient tracking of continuous objects in wireless sensor networks." *5th international conference on Ubiquitous Intelligence and Computing*, pp. 323-337, 2008. Article (CrossRef Link).

[9] Dong XJ, Chauhdary SH, Xu J, Yi YZ, Dong HL, Park MS. "Energy-Efficiency continuous object tracking via automatically adjusting sensing range in wireless sensor network." *4th International Conference on Convergence and Hybrid Information Technology,*pp.122-127, 2009. Article (CrossRef Link).

[10] Chintalapudi K. and Govindan R. "Localized edge detection in sensor fields". *Ad-hoc Networks Journal*, , pp. 273–291, 2003

[11] Liu J, Cheung P, Guonidas L, and Zho F. "A dual-space approach to tracking and sensor management in wireless sensor networks." *ACM International Workshop on Wireless Sensor Network and Applications Workshop*, 2002. Article (CrossRef Link).

[12] http://www.libelium.com/products/waspmote

[13] http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html

[14] Olule E, Wang G. J, Guo M. Y and Dong M. X, "RARE: An Energy-efficient target tracking protocol for wireless sensor networks," In prce. of ieee international conference on parallel processing workshops, pp. 76-82, Sep. 2007.

[15] Dattagupta. S, Ramamritham. K, Kulkarni. P and Moudgalya. K, "Tracking dynamic boundary fronts using range sensors," In *Proc. of Wireless Sensor Networks, in: Lecture Notes in Computer Science*, vol. 4913 , pp. 125–140, 2008.

[16] Tiwari. R, Thai M. T, and Helal. S, "Localized energy efficient detection and tracking of dynamic phenomena," in *Proc. of the IEEE Global Communication Conference*, GLOBECOM, 2010

[17] Susca .S, Martinez .S, Bullo. F, "Monitoring environmental boundaries with a robotic sensor network," *Transactions on Control Systems Technology,* vol. 16, no. 2, pp. 288–296, 2008.

[18] Jin.Z, Bertozzi. A, "Environmental boundary tracking and estimation using multiple autonomous vehicles", in *Proc. of IEEE Conference on Decision and Control*, pp. 4918–4923, 2007.

**Sajjad Hussain Chauhdary** is currently working as Research Engineer in Convergence Technology Research & Development Center, LS Industrial Systems Ltd. He is PhD candidate in Department of Computer Science and Engineering Korea University- South Korea. He completed his Master Degree in 2006 from Department of Computer Science and Electrical Engineering, Ajou University – South Korea. He is member of SAE International, ICST, IAENG, AICIT and IACSIT. He participated in IEEE IDC 2009, IEEE INC 2009, IEEE NISS 2009, IEEE NCM 2009 and IEEE ASN 2010 as program committee member. His research interests include Smart Grid, AMI (Advanced Metering Infrastructure), communication infrastructure for electric vehicle and its charging and renewable energies, power line communications, industrial wired/wireless communication networks.

**Jeongjoon Lee** received the B. S. and the M. S. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 1988 and 1990, respectively, and the Ph. D. degree in the School of Electrical and Computer Engineering from Purdue University, U.S.A., in 2005. Dr. Lee is currently General Manager and Research Fellow of LS Industrial Systems, Co. LTD, Korea, leading Smart Network Research Team in the Smart Grid Research Division. His research interests include Smart Grid, AMI(Advanced Metering Infrastructure) and smart metering, communication infrastructure for electric vehicle and its charging and renewable energies, ZigBee Smart Energy Profile(SEP) and power line communications, industrial wired/wireless communication networks, RFID, and USN(Ubiquitous Sensor Networks)

**Sayed Chhattan Shah** is a Senior Researcher at Electronics and Telecommunications Research Institute in Daejeon, South Korea. His current research interests include parallel and distributed computing systems, ad hoc computational Grids, and mobile ad hoc networks. He received his Ph.D. in Computer Science from Korea University in 2012 and his M.S. in Computer Science from National University of Computer and Emerging Sciences in 2008. Dr. Shah is also a member of IEEE, IEEE Communications Society, and International Association of Engineers. He has been on program committees of various international conferences including IEEE ISIEA, IEEE ICCSII, and IEEE PECON.

**Myong-Soon Park** is Professor of Computer Science and Engineering at Korea University in Seoul, South Korea. He received his B.S. in Electronics Engineering from Seoul National University, M.S. in Electrical Engineering from University of Utah in 1982, and Ph.D. in Electrical and Computer Engineering from University of Iowa in 1985. During 1988–1989, he was an assistant professor of Computer Science, Electronics and Electrical Engineering at POSTECH. He has also worked with Agency for Defense Development Korea. Professor Park was a chair of the SIG on parallel processing of KIISE (1997-2000) and has been on program committees of various international conferences. His research interest includes sensor networks, internet computing, parallel and distributed systems, and mobile computing.