

# Continuous objects detection and tracking in wireless sensor networks

Tarek R. Sheltami<sup>1</sup> · Shehryar Khan<sup>1</sup> · Elhadi M. Shakshuki<sup>2</sup> · Meshawi K. Meshawi<sup>3</sup>

Received: 28 February 2016 / Accepted: 8 April 2016 / Published online: 12 May 2016  
© Springer-Verlag Berlin Heidelberg 2016

**Abstract** Most research, in the area of target detection and tracking in wireless sensor networks (WSN), is focused on a single or multiple targets tracking. However, limited research is aimed at tracking and detection of continuous objects such as forest fires, biochemical materials and mudflows, etc. These continuous objects pose new challenges due to their nature and characteristics of changing in size and shape, shrinking and expanding, splitting into multiple objects, or merging of multiple objects into one object. Continuous objects tracking and detection require extensive communication, which consumes a considerable amount of network energy. To this end, this paper proposes a new algorithm named Continuous Object Detection and Tracking (CODAT). This paper also introduces a new data structure for reporting data. This new data structure reduces the communication cost of the overall algorithm without compromising the accuracy for reconstructing the boundary of a continuous object at the base station. A concept for differentiating between the holes in the phenomenon and overall phenomenon changes at the base station level is also introduced which provides additional information to the user as an added improvement while maintaining the high accuracy and efficiency. To demonstrate the feasibility and efficiency of this algorithm, it is implemented and compared its results with two known algorithms,

including Continuous Boundary Monitoring (COBOM) and Detection and Monitoring for Continuous Objects (DEMOCO). The simulation results show that CODAT outperforms COBOM and DEMOCO with dense WSNs.

**Keywords** Continuous objects detection · Phenomenon detection · Wireless sensor networks

## 1 Introduction

Today's wireless sensor networks can be deployed and used on large scale areas for variety of applications such as disaster monitoring, environment monitoring, health monitoring, traffic control and infrastructure security (Akyildiz et al. 2002; Chaudhary et al. 2008; Fuemmeler and Veeravalli 2010). A considerable amount of research is done in Object Tracking in Wireless Sensor Networks (OTSNs) (Amirjavid et al. 2012; Salamah and Zawaideh 2013; Ugolotti et al. 2011). In OTSNs, sensor nodes are deployed over a monitored area with predefined geographical boundaries. The base station issues the commands and collects the data of interest, and acts as an interface between the applications and OTSN. The sensor nodes track the objects that intrude its area of detection and report their status to the base station. This application of object tracking with the ability to detect anytime and anywhere became possible due to the advancement in embedded processors technology and low cost of sensor nodes. The OTSN has wide range of applications including business, military, environmental safety, etc. Traditional object detection and tracking focuses on one or several individual objects, such as intruder, tank, and/or vehicle (Salamah and Zawaideh 2013; Tang et al. 2014; Thangarajan et al. 2013). Recently, several researchers have considered continuous objects spreading in a very large

✉ Tarek R. Sheltami  
tarek@kfupm.edu.sa

<sup>1</sup> Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

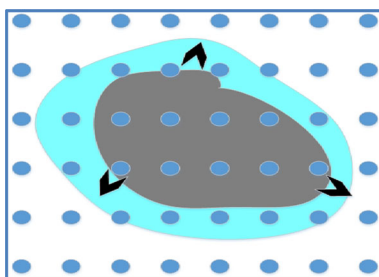
<sup>2</sup> Jodrey School of Computer Science, Acadia University, Wolfville, NS B4P 2R6, Canada

<sup>3</sup> Electrical Engineering Department, University of Benghazi, Benghazi, Libya

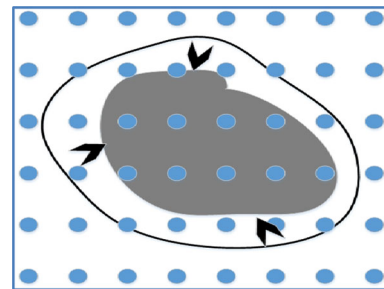
regions such as diffused noxious gases, biochemical, chemical liquids, and forest fires (Chang et al. 2008; Hong et al. 2013, 2015; Ji et al. 2004; Jung-Hwan et al. 2008; Park et al. 2012; Shen et al. 2015; Zhong and Worboys 2007). Due to sensors' ability to detect and monitor objects anytime and anywhere, continuous object (or phenomenon) tracking in sensor networks has the potential for being used in many application domains such as monitoring fire outbursts, nuclear explosions and hazardous biochemical diffusions. To monitor such spatial phenomena, it is necessary to understand the possible evolving situation of the phenomenon. These shapes can emerge when one phenomenon shrinks or expands, two phenomena merge together and form one phenomenon, one phenomenon splits into two or more new phenomena, and one or two holes may appear or disappear in phenomenon. For example, expanding and shrinking of phenomenon are demonstrated in Figs. 1 and 2 respectively. Figure 3 demonstrates holes in the phenomenon that may shrink or expand.

A continuous object usually has a size larger than that of an individual object, which implies that there are more event nodes in continuous object detection and tracking. If all event nodes report their location information to sink nodes, it will result in extremely high traffic load. An event node is defined as a reporter if it is required to report its location information to sink nodes. Thus, the main challenge in designing continuous object detection and tracking protocol is to reduce the number of reporters. Therefore, individual object detection and tracking protocols are not applicable to continuous object detection and tracking.

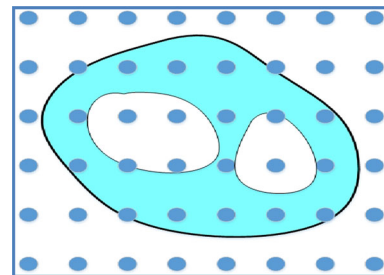
In WSNs, two main processes are involved in object detection and tracking. First process is the collaborative data processing that involves the sensors to collaborate and devise an accurate and concise mechanism for the determination of object location information. Second process is Object Location Reporting. This involves the transportation of the location information to the sink on time. On the basis of these two processes, the research in phenomenon detection and tracking are divided into two categories. Examples of collaborative data processing, such as in Dynamic Cluster Structure for Continuous Objects Detection and Tracking (DCSC) (Ji et al. 2004) and Continuous



**Fig. 1** Phenomenon expansion



**Fig. 2** Phenomenon shrinkage



**Fig. 3** Holes inside the phenomenon

Object Detection Algorithm (CODA) (Chang et al. 2008), are formed by dynamically grouping the boundary nodes. The boundary information is collected by the cluster head and is transmitted to the base station. Similarly, all the cluster heads send their information and then the global boundary is estimated at the base station. We argue that cluster formation itself consumes a considerable amount of energy. With regard to Object Location Reporting, Continuous Boundary Monitoring (COBOM) (Zhong and Worboys 2007) and Detection and Monitoring for Continuous Objects (DEMOCO) (Jung-Hwan et al. 2008) both use energy efficient algorithms for detection and tracking of continuous objects. Using these two algorithms, the energy consumption is reduced by selecting few nodes for reporting among large number of predefined boundary nodes. Energy consumption is less in this case by avoiding cluster formation, but the accuracy in determining the boundary is compromised. In this paper, we successfully addressed the aforementioned deficiencies by trading off between efficiency and accuracy.

## 2 Related work

Research in detection and tracking of individual or multiple targets is ample. These targets might include animals, vehicles and persons, etc. that are within sensor network proximity region. Applications using wireless sensor networks for habitat monitoring are discussed by (Mainwaring et al. 2002; Malik et al. 2011).

## 2.1 Individual and multiple target tracking schemes

A number of techniques are proposed for identifying targets and discussed in (Kaplan et al. 2001; Learned et al. 1992). It has been discussed by Krishnamachari and his co-workers (2001) discussed how to track and detect an object via three sensor nodes. In another research, different signal processing techniques are discussed and classified (Li et al. 2002). Zhao et al. in (2002) utilized information security to predict future sensing actions. Similar approach for target tracking is discussed in Dynamic Convoy Tree-Based Collaboration (Zhang and Cao 2004a, b). A tree structure is described in DCTC (Zhang and Cao 2004a), which is known by convoy tree. Addition and pruning of sensor nodes enable dynamic configuration of the tree as the object moves within the region of deployed sensor nodes. Prediction-based Energy Saving (PES) scheme and Dual Prediction-based (DPR) scheme for Object Tracking in Sensor Networks are described in (Xu et al. 2004a, b) respectively. Intelligent prediction of object movement is achieved via sensor nodes as well as the base station. Chen et al. proposed (2004) a dynamic clustering scheme for acoustic tracking with light weight and decentralized mechanism. In this proposed approach, a Cluster Head (CH) identifies the strength of an acoustic signal. When it exceeds its predetermined threshold, this CH broadcasts a message with the invitation to the sensors in its vicinity to join the cluster and provide their sensing information. This CH then identifies the acoustic target via localization techniques after it receives sufficient information from newly joined sensor nodes.

## 2.2 Continuous objects tracking schemes

The above-mentioned schemes are specifically designed for the detection and tracking of individual targets, e.g., animals, people, vehicles, etc. In many cases it is a necessity track a phenomena or a continuous object that spreads on a large area. Such type of phenomena might include toxic gases, wild fires, oil spills, among others. These continuous objects are different from multiple or individual targets in a manner that they are not discretely distributed over specific area. They are continuously distributed over a large region and can cover a massive area. They can usually diffuse, change their shape, merge into one another or split into comparatively smaller continuous objects. Hellerstein et al. (2003) proposed an Isobar scheme. In this scheme, spatially correlated data aggregation method is used for mapping purposes. The sensor nodes with the same readings status are collected by the Isobar into polygons as soon as the information from the nodes flow towards the sink. This scheme reduces the volume of transmitted data; however, every node has to

participate in data collection and passes data to the parent node which is inefficient.

Solis and Obraczka in (2005a, b) attempted to overcome Isobars drawback by proposing isolines scheme. In this scheme, if a node detects an isoline between itself and the surrounding neighbors then it reports to the sink; otherwise, it does not report to the sink. This way, Isoline based technique sends less bytes to the sink as compared to Isobars. However, in Isolines almost 50 % of the nodes report to the sink in each sampling round. This introduces communication cost. If the sensor nodes are deployed in dense settings, then only the sensor nodes that detected the Isolines report to the sink. Another scheme similar to Isoline is described in (Meng et al. 2004), which is a combination of spatial and temporal suppressions. Spatial suppression is defined as follows when a node has a similar reading compared to neighbors, it does not need to report to sink. Temporal suppression is defined as follows when nodes' reading remain the same compared to their previous reading, it does not need to report to the sink.

Another algorithm known as Constraint Chaining (COCH) is proposed by Silberstein et al. (2006). This algorithm makes use of spatial and temporal suppressions in a very efficient manner. According to this algorithm, only one node on each side of the boundary of the continuous object reports to the sink. COCH algorithm provides good performance when the shape of the continuous object does not change with regular movement. But, its performance is considerably degraded when the object's shape changes irregularly. Consequently, the number of nodes reporting to the sink increases that costs more energy. Chintalapudi and Govindan attempted to address the problem of boundary detection (Chintalapudi and Govindan 2003). In their approach, enough data is reported so that the sink can construct an accurate boundary. Since all the boundary nodes report to the sink, the communication cost becomes high. A fault tolerant algorithm based on data mining techniques is proposed in (Ding et al. 2005). Both approaches presented in (Chintalapudi and Govindan 2003; Ding et al. 2005) did not discuss how the nodes report their data to the sink. They also overlooked the situation when continuous object changes its shape. It would cost too much energy if all the nodes report their data to the sink simultaneously.

To address the high cost of reporting, Kotidis (2005) proposed an algorithm in which the representative nodes report data on behalf of neighboring nodes. Nodes ignore some queries due to efficiency and/or accuracy of snapshot queries that are considerably reduced. For the reduction of errors, update messages are sent at predefined intervals. Due to this approach, extra energy is consumed by snapshot queries and hence making it inefficient for the use of continuous boundary changes.

For detection and tracking of continuous objects, a dynamic cluster structure named Dynamic Cluster Structure for Continuous Objects Detection and Tracking (DCSC) is proposed (Ji et al. 2004). In this approach, when the emergence of an object is detected in sensor's local area during the current time slot, then it is interpreted as the object's boundary has moved through its area of detection during the previous time slot. Consequently, the object's boundary is likely close to the sensor. Then, communication takes place between the sensor and its one hop neighboring sensors to query, provided the neighboring sensors also detect the object. If there is no detection of an object by the neighboring sensors, then the sensor becomes the boundary sensor. Similarly, if the sensors in the current time slot detect the disappearance of object, then the boundary of object must have moved through its area of detection in the previous time slot. Then, the sensor node enquires its one-hop neighbors about the detection of the object. If the neighboring sensor nodes can detect the object, then those neighboring nodes become the boundary sensor nodes. The neighboring nodes in DCSC (Ji et al. 2004) reply back to the node that inquired whether they have different reading or the same reading compared to the inquirer node, and thus this process adds to the communication cost. Cluster is formed after the selection of boundary nodes. However, the description of cluster formation is not very clear. Also, arguable that cluster formation is not suitable when the purpose of its formation is to save energy. This is because considerable communication is required to form clusters which causes delay in application and hence it is not suitable for real time monitoring of unexpected diffusion or drift such as explosions, etc. Moreover, all boundary nodes along with cluster heads are directly or indirectly involved in routing the data to the sink. This increases the precision and accuracy in identifying the shape of the boundary, but at the expense of high traffic delays and overheads.

A hybrid dynamic/static clustering for the tracking of continuous object called Continuous Object Detection Algorithm (CODA) (Chang et al. 2008) is proposed. At the time of deployment, static clusters are formed and it is assumed that Cluster Head (CH) of each static cluster assigns a unique cluster ID to that cluster. When a new node joins the cluster, it exchanges messages with CH. During this process, a new node is informed about the cluster ID. The assumption is that every sensor node is aware of its location and this information is sent to CH when the cluster is joined for the first time. After the CH receives the location information of all the nodes in a cluster, the CH determines the boundary nodes and sends notify messages to inform them that they are the boundary sensor nodes of that particular cluster.

Whenever an object is detected, control messages are used for the transmission of detection information to CH. Control messages are classified into two types. (1) Sense message is used only by Static cluster Inner sensors (SIs). Once the SIs detect target object, sense message is sent to CH. (2) Report message is used only by Static cluster Boundary sensors (SBs). This message is in the form of bitmap and if detected object is identified within cluster  $n$  then  $n$ th bit is set to 1. Communication takes place between SB and its one-hop neighbors after the detection of object to inquire the neighboring nodes of their object detection status. When SB receives this information, the corresponding bit of the bitmap is set to 1. Then, a report message is sent to CH so that all the static clusters are identified in which the target object has spread.

All the nodes in the static cluster transmit their detection information to CH and dynamic cluster is then formed from the boundary nodes. In this case, the boundary nodes for the detected object are determined by cluster heads of the static cluster instead of the dynamic clusters in order to save energy. However, there is still high communication costs required in case of large number of nodes. Additionally, extra energy is consumed in grouping the boundary nodes into dynamic cluster.

An energy efficient algorithm for boundary detection and tracking named Continuous Boundary Monitoring (COBOM) is presented in (Zhong and Worboys 2007). In this algorithm, the nodes initially communicate with each other and send their respective Boundary Node Array (BN-Array), start node and its unique ID to the sink. The start node is selected randomly. The BN-Array contains detection readings of the neighboring nodes. Initially, the BN-Array readings are all zeros to mean any node does not detect any phenomenon. The sensor broadcasts its own ID and reading. If its current reading is different compared to its previous reading, i.e., detected the phenomenon or otherwise. The receiving node stores the reading in its BN-Array and the sensor becomes the BN given its BN-array contains at least one different reading. Figure 4 shows the BN-Array that stores the readings in counter clockwise direction with  $c$  as the start node.

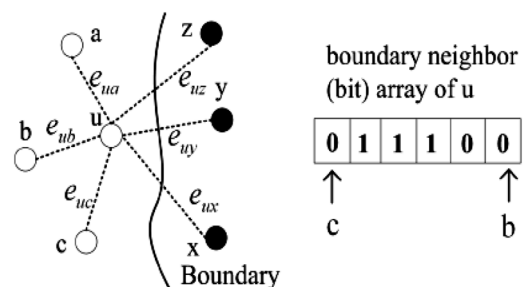


Fig. 4 COBOM BN-array



Among these boundary nodes few nodes are selected to become Reporting Nodes (RNs). The BN-Arrays with number of different detection readings, i.e., more number of 1's readings, more likely it will become a candidate for Reporting Node (RN), as shown in Fig. 4. The RNs are selected when a BN-Array of a particular node crosses a threshold set. As BN-Array reaches the threshold, a back off random timer starts. At the end of this timer, the node becomes a RN. The earlier the BN-Array of a particular node reaches the threshold, the early it becomes the RN and suppresses the nodes within its range from reporting. In COBOM (Zhong and Worboys 2007), it is reported that it is an energy efficient as few reporting nodes are selected. Also, the report message size is small as it only contains neighbors' detection readings as bits instead of keeping neighbors' IDs. The report message size contains the RN's ID, its reading and the node's BN-array. In COBOM (Zhong and Worboys 2007) approach, it is not clear how the base station determines the location of RN's neighbors since the BN-Array contains only the detection readings not the IDs. Moreover, the base station has RN's ID only which compromises the accuracy when the base station tries to reconstruct the boundary in the future. The other issue with COBOM (Zhong and Worboys 2007) is that the BNs and RNs are formed on both sides of the Boundary, as shown in the Fig. 4. In this figure, BN-Array of node *y* is similar to that of node *u* which indicates that node *y* is also the Boundary Node. This results in yielding more number of BNs and RNs; thus, increasing the communication cost.

For detection and tracking of continuous objects, an energy efficient algorithm Detection and Monitoring for Continuous Objects (DEMOCO) (Jung-Hwan et al. 2008) is proposed. A sensor node detects an object by comparing its current detection status to its previous one. Compare One Zero (COZ) messages containing current detection statuses are sent to the neighboring nodes. The BN for the continuous objects are those nodes received the COZ messages, and their object detection status are different from the status included in the messages. If the detection status is the same as that of neighboring nodes, the receiving node ignores this message. Different random back off timers are assigned to the BNs. Nodes with short back off timers send the data to the sink and suppresses the other boundary nodes from sending the data to the sink. These nodes are called as Representative Nodes (RN). There are few RNs that report data to the sink. The report back message includes the RN's own ID and neighboring BN's ID that has the most powerful signal strength. It should be noted that it is not clear how the signal strength is compared to the received signals. As the signal strength is in the form of 1's and 0's, the estimation of signal strength is not possible by simply comparing 1 and 0.

### 3 Continuous object detection and tracking (CODAT)

There are a number of ways to detect and track continuous objects. The simplest approach for the detection of boundaries is when every node sends its detection information to the sink node, which in turn forwards the information to the user via a network. The main disadvantage of this approach is that most of the energy is dissipated at the breakpoints, which considerably reduces the network lifetime. In DCSC (Ji et al. 2004), small number of nodes are selected at the boundary of the continuous object grouped into clusters, and the nodes at the boundaries are responsible for forwarding data to the sink nodes. This approach reduces the communication cost and consequently requires less energy.

In COBOM (Zhong and Worboys 2007), the selected boundary nodes (BN) are comparable to that of DCSC (Ji et al. 2004). However, only few reporting nodes (RN) are responsible for forwarding data to the sink. Our proposed approach is similar to DEMOCO (Jung-Hwan et al. 2008) approach with further reduction of the BNs as well as the RNs as compared to the previous algorithms. The main advantage of our proposed algorithm lies in the improvement in accuracy and efficiency of the detection and tracking of continuous objects. Our proposed approach is different with respect to the average report data size forwarded by the RNs to the sink.

In this paper, we analyze and compare our proposed solution with both COBOM (Zhong and Worboys 2007) and DEMOCO (Jung-Hwan et al. 2008) approaches on the basis of efficiency, accuracy and uniform random deployment. Table 1 summarizes the main features of COBOM and DEMOCO along with their drawbacks.

Before we describe our proposed energy efficient algorithm, Continuous Object Detection and Tracking (CODAT), the following discusses our assumptions and definitions:

- Every node has a unique ID.
- Static nodes are deployed in a large number along with sink nodes.
- All sensor nodes are uniformly distributed.
- All sensor nodes have the same functionality and capability.
- All sensor nodes have same the sensing range and communication range.
- The sink nodes know all sensor nodes' locations using unique IDs via GPS unicast messaging.
- All sensor nodes know their own locations and positions using GPSs or other triangulation or localization techniques.
- The possibility of any data loss is not considered.

**Table 1** Comparative analysis of COBOM and DEMOCO

Features	COBOM	DEMOCO
Detection	Based on BN array	Based on difference in detection readings
Boundary nodes	Both sides	One side
Number of BNs and RNs	N	N/2
Drawback	No neighboring node ID BN-Array readings only	Nearest ID based on signal strength

- Damages or destructions on any node are not considered.

### 3.1 Definitions

#### 3.1.1 Phenomenon detection status (PDS)

The sensor node's detection status changes from 'undetected' to 'detected' or vice versa. Number 1 represents 'detected' status and number 0 represents 'undetected' status.

#### 3.1.2 Phenomenon changed message (PCM)

The node that has different PDS compared to its previous timeslot broadcasts a message to its neighboring nodes. This message is called a phenomenon changed message. This message contains the node's ID, detection status and phenomenon tags.

#### 3.1.3 Boundary nodes (BN)

When a node receives PCM, it compares its own PDS compared to the PDS received in broadcasted PCM. If a node has at least one different 'detect' status compared to the received message then that node becomes the boundary node.

#### 3.1.4 Reporting nodes (RN)

The RN sends information to the sink on behalf of the surrounding neighboring nodes in order to conserve the network energy. In our proposed algorithm, a RN sends its own ID and a boundary node array.

### 3.2 The proposed continuous object detection and tracking (CODAT) algorithm

The sensor nodes are periodically activated to detect the phenomenon such as wild forest fire, gas leakage, oil spills, etc. depending upon the specific application. When no phenomenon is detected, the sensors go to idle state. If a sensor node detects a phenomenon and finds the current Phenomenon Detection Status (PDS) is different from the

previous PDS, then it broadcasts Phenomenon Changed Message (PCM). This PCM contains a node's unique ID and PDS and Phenomenon Tags.

When a sensor node  $x$  receives a PCM from a neighboring node within its communication range, it compares it with its own PDS. If they match, then the node ignores the received PCM. If the node finds at least one received PDS is different from its own PDS, then the node  $x$  becomes a Boundary Node (BN). This BN counts the number of received PCMs for setting different waiting time and use it for making future Reporting Nodes (RN). This BN also maintains a Boundary Node-Array (BN-Array), which contains neighboring nodes' detection readings and a couple of IDs along with Phenomenon Tags. The reporting nodes consequently send the boundary information to the sink along with the current tags of the nodes BN-Array is explained later.

In our proposed algorithm, when an object is expanding only the nodes in the outer region of the phenomenon become BNs. Similarly, when an object is shrinking only the nodes in the inner region become BNs.

```

1 Algorithm for Hazardous Gas Leakage Detection
2 do
3   Run the CODAT Algorithm
4   If (a node current PDS  $\neq$  previous PDS) Then
5     {Broadcast phenomenon change message// Containing Node's
      status, ID and Phenomenon Tags}
6   Else
7     {Set a node to idle}
8   If (no. of received PCM > 1) and (received PDS  $\neq$ 
      Current PDS) Then
9     { Update Bn-Array and Set node to boundary node}
10  Else
11    {drop current PCM message}
12  If (no. of received PCMs for BN > Threshold) Then
13    { Random waiting Timer countdown starts//Value depends upon
      different threshold levels}
14    If (Random Waiting Timer=0) and (BN=Highest Energy level) Then{
15      Set node to Reporting node
16    }

```

The behavior of algorithm is demonstrated in Figs. 5 and 6. Our approach is different from COBOM approach where the nodes in both the outer and inner region of the phenomenon become BNs regardless whether the object expands or shrinks. This is because COBOM selects the BNs based on the difference in detection readings within a node's BN-Array, which eventually results in more number of BNs as compared our approach.

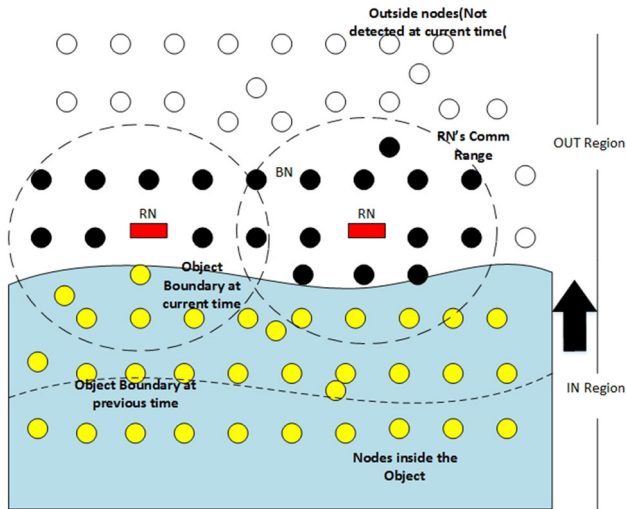


Fig. 5 Expansion of phenomenon

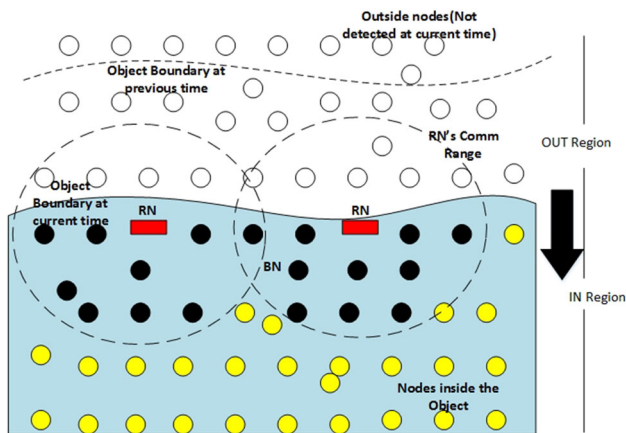


Fig. 6 Shrinkage of phenomenon

Although most of the algorithms can differentiate between the holes and overall phenomena, we propose additional information as an added improvement forwarded to the user along with the reported data with more accuracy and with less number of reporting nodes compared to COBOM. To differentiate between the outer boundary and the inner boundary by the base station, we used an approach similar to the approach presented in (Hussain et al. 2013). Where outer boundary refers to the boundary of the phenomenon such as expansion or shrinkage of the phenomenon, and the inner boundary refers to the boundary of the holes such as expansion or shrinkage of a hole inside the phenomenon. Every node performs the detection process based on a predefined time interval. If a node does not detect change in its PDS, then it goes to idle state. For outer boundary detection, the following two cases are considered. (1) If a node detects a phenomenon in a current time while it did not detect a phenomenon during the previous detection time, then it broadcasts a Phenomenon Changed Message

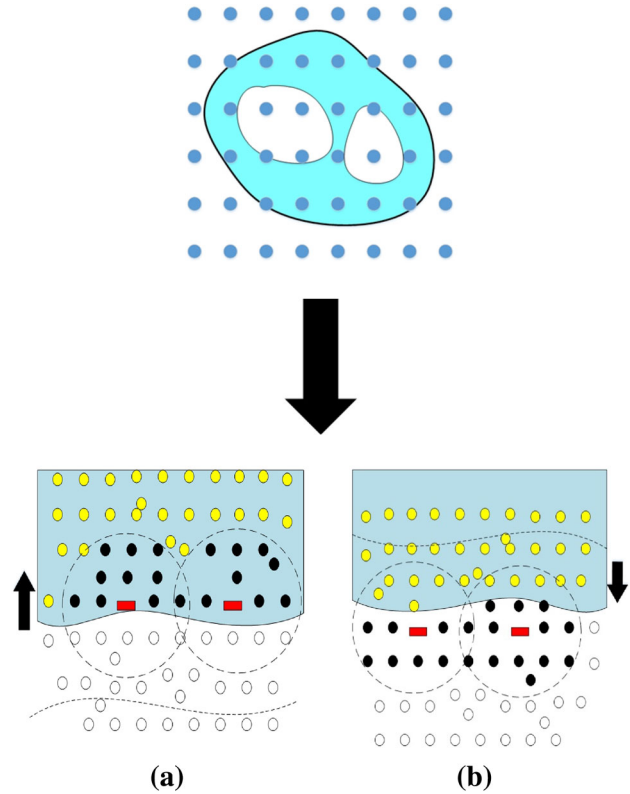


Fig. 7 Hole expansion and hole shrinkage

(PCM). The PCM identifies change in the detection status to its neighbors that are in range. This node also receives PCMs from other neighboring nodes. (2) If at least one of the received messages has the detection status different from the node itself, then the node becomes the boundary node. Otherwise, it ignores the received PCM. Based on the number of PCMs that this node receives, a back off timer is set for this node and makes it a reporting node. At the end of this period, the boundary nodes tag themselves as Phenomenon Expansion. The reporting nodes consequently send the boundary information to the sink along with the current tags of the nodes. Similarly, when the nodes did not detect a phenomenon in the current time but detected a phenomenon in the previous time slot while tagged as phenomenon expansion, then it broadcasts PCM and Phenomenon Shrinkage is represented by this change. The node becomes a boundary node if it receives a PCM from neighboring node having different detection status as compared to itself. On the basis of the number of received PCMs, back off timers are set and reporting nodes are selected. At the end of this process, all nodes tag themselves as Phenomenon Shrinkage. For inner boundary detection, the following two cases are considered. (1) When a node detects a phenomenon in the previous time slot, but does not detect a phenomenon in the current time slot that has no tag. This change represents the expansion of a hole, as shown in Fig. 7a. (2) When a node

detects a phenomenon in the current time slot, but it did not detect in the previous time slot while it has a tag of Hole Expansion. This change represents the shrinkage of a hole, as shown in Fig. 7b. In both cases, the node that detects the change, it broadcasts PCMs and receives PCMs from its neighbors. This node also becomes a boundary node provided that the status in at least one of the received PCMs is different from its own detection status. Reporting nodes are selected on the basis of back off timers. In case of Hole Expansion, nodes tag themselves as Hole Expansion (HE). In the case of Hole Shrinkage, nodes tag themselves as Hole Shrinkage. These tags are valid for single time period only. These tags create differences at the base station level between the Phenomenon Shrinkage and Hole Expansion and similarly between Phenomenon Expansion and Hole Shrinkage.

In situations when a change in the status of a phenomenon passes without detection (undetected to detect), there are two possibilities it is either the phenomenon is expanding or the hole is shrinking. It is assumed that a hole does not shrink unless it is expanding; as such, if the node detects a phenomenon during current time slot and did not detect it in the previous time slot and has a tag Hole Expansion, then it means that it is previously involved in finding the boundary nodes for Hole Expansion and the Hole is currently shrinking. Alternatively, in other case, if a node currently detects a phenomenon and did not detect it previously and has no tag, then this indicates that the phenomenon is expanding. In situations when no change in a phenomenon is detected while no change happened (detect to undetected), there are also two possibilities it is either the phenomenon is shrinking or the hole is expanding. It is assumed that a phenomenon shrinks unless it is expanded before. Therefore, whenever a node detects a phenomenon in the previous time slot and did not detect in current time slot provided it has a tag of phenomenon expansion, then it indicates that this node is previously involved in finding boundary nodes for expansion of phenomenon while currently the phenomenon is shrinking. Meanwhile, if the node has no tag then it indicates that currently the hole is expanding. These tags will expire after a single time period via timers. New tags will be formed in the next time period.

### 3.3 Reporting nodes

A simple approach to report the information to the sink, all the boundary nodes send their data to the sink. However, this approach is cost inefficient. Our main objective of this research work is to increase the network lifetime as well as to improve the accuracy in detection and tracking of phenomenon. In our approach, we select few REPORTING NODES (RN) among the existing BNs that will report data

to the sink on behalf of the neighboring nodes while maintaining the accuracy. The RNs are selected based on the number of PCMs. The BN that receives the highest number of PCMs will set a random waiting time. The more a BN receives PCMs, the shorter will be its random waiting time and likely to become a RN. This approach ensures high probability of the boundary nodes that are near the actual boundary of the phenomenon to become RNs. If two or more BNs have the same random waiting time, then the BN with the highest energy level is selected as the RN. The re-election of RN is performed after each time period. The waiting time as mentioned in DEMOCO (Jung-Hwan et al. 2008) can be defined in Eq. (1).

$$D = \left\{ \begin{array}{l} \frac{W}{PCM_{total}} - U \left( \frac{\frac{W}{PCM_{total}+1} - \frac{W}{PCM_{total}}}{2} \right), \quad \text{if } PCM_{total} > 2 \\ \frac{W}{PCM_{total}} + U \left( \frac{\frac{W}{PCM_{total}-1} - \frac{W}{PCM_{total}}}{2} \right), \quad \text{if } PCM_{total} = 2 \\ \frac{W}{PCM_{total}} + U \left( \frac{W}{PCM_{total}-1} - \frac{W}{PCM_{total}} \right), \quad \text{if } PCM_{total} = 1 \end{array} \right. \quad (1)$$

where, D is the back off timer and W is the maximum waiting time. The total number of PCMs that are received is denoted by PCMtotal. If PCMtotal is greater than 2, then the back off timer is considered the shortest, as shown in Eq. (1). If PCM total equals to 2 or to 1, then the back off timer is longer. Equation (1) demonstrates that the back off timer ensures that the backs off timers for boundary nodes within the communication range are different from one another.

As mentioned earlier, each node maintains a Boundary Node Array (BN-Array). This BN-Array contains detection status readings of the neighboring nodes that have been received through PCMs. BN-Array also contains the unique IDs of the first two nodes received through PCM. As shown in Fig. 8, a BN-Array has Phenomenon Detection Status's (PDS) along with unique IDs of the first two neighboring nodes received through PCM and Phenomenon Tags (PT). This data structure for BN-Array enables a sink to recognize neighboring nodes via first two unique IDs that are received through PCM. While the rest of the nodes are located through a series of the detection readings in BN-Array near to the first two IDs. This is possible because the sink has a map to all nodes in the network; however, this requires more processing power and memory. Our approach is different from DEMOCO (Jung-Hwan et al. 2008) in that the reporting node sends its own ID and the ID of the nearest neighbor based on signal strength. We

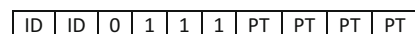


Fig. 8 BN-array of CODAT



argue that estimating the nearest node based on signal strength is not very accurate approach since the signal strength is estimated in the form of 1's and 0's.

The BNs with the shortest waiting time will send a broadcast message to its neighboring nodes to indicate that it will become RN. At the same time, the neighboring nodes don't need to send data to the sink node. This RN will send a report data to the sink on behalf of all the neighboring nodes that are within its communication range. The RN will also include its own unique ID with the reported data, a BN-Array and the phenomenon tags as explained earlier. In our approach, we consider each unique ID is approximately 2 bytes in size. Later, we will discuss how the average report size of our algorithm does not exceed the previous algorithms while considering improvements in accuracy and efficiency. Figure 9 shows the flowchart for CODAT algorithm.

### 4 Simulation Results

In this section, we evaluate and compare our proposed algorithm CODAT against both DEMOCO (Jung-Hwan et al. 2008) and COBOM (Zhong and Worboys 2007) algorithms for continuous moving phenomena. To test the performance of these algorithms, we developed a simulator in Java. It should be noted that we consider neither possible

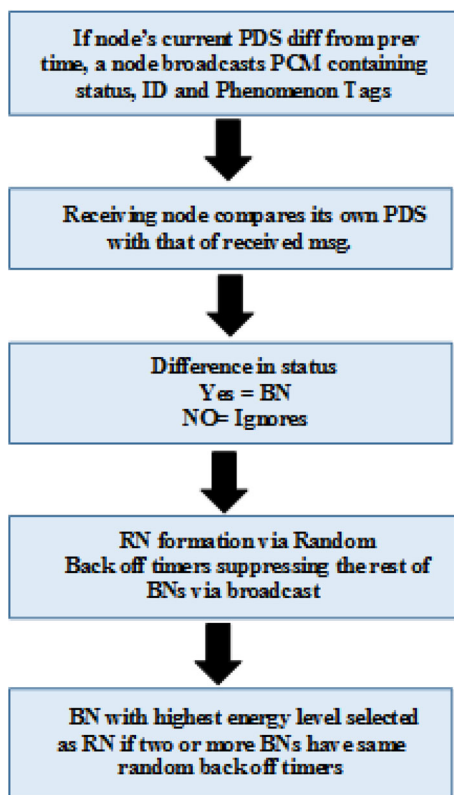


Fig. 9 CODAT mechanism

contention nor concerned with the routing approach used by nodes to send data to the sink. We ran each simulation for 300 times with a node's unique ID of 2 bytes.

The number of BNs and RNs generated were depending on the density of sensor nodes. In our simulation model, the area is fixed while the number of nodes may vary. Wireless sensor nodes are deployed in an area of  $500 \times 500 \text{ m}^2$ . The number of sensor nodes is set to 1500 while it is set to 5000 for sparse and dense setting respectively. The uniform distribution of 5000 sensor nodes is shown in Fig. 11. In order to simulate real life scenario, an unsmoothed object is simulated. The object expands like a staircase structure from the bottom left corner, as shown in Fig. 10. The object expands in each time period (each time period is 5 s) up to 25th time period with low diffusion rate (expanding 3 meters in each time period). The communication range for sensor nodes is set to 15 and 25 m.

In Figs. 12 and 13, the phenomenon expansion for a total of 5000 nodes is shown for both CODAT and COBOM algorithm at 5th time period with 25 m communication range for all sensor nodes. Similar phenomenon expands at the 10th time period with 25 m communication range for all sensor nodes, as shown in Figs. 14 and Fig. 15. We can clearly see in these figures that in case of COBOM the Boundary Node strip (Blackish nodes) is more in width as compared to the boundary node strip of CODAT. This is because in case of COBOM boundary nodes (BNs) are formed on both sides of the boundary of the phenomenon, whereas in case of CODAT boundary nodes are formed on a single side of the boundary of phenomenon resulting in overall less number of boundary nodes (BNs) and reporting nodes (RNs) compared to COBOM. Among these blackish nodes, reporting nodes are shown in red color.

In Figs. 16 and 17, the phenomenon at 20th time period is shown which has further expanded.

In Figs. 18 and 19, the phenomenon has further expanded up to 25th time period. At this point, maximum number of boundary nodes and reporting nodes are generated because the phenomenon has reached to the diagonal of square sensor field.

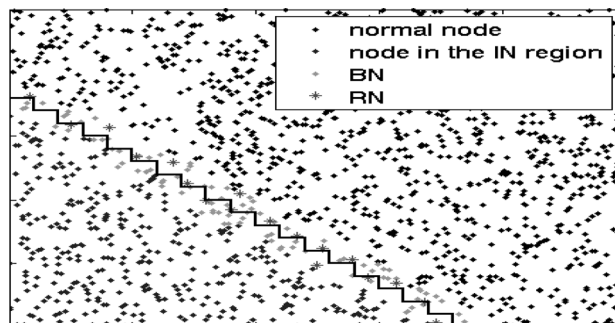
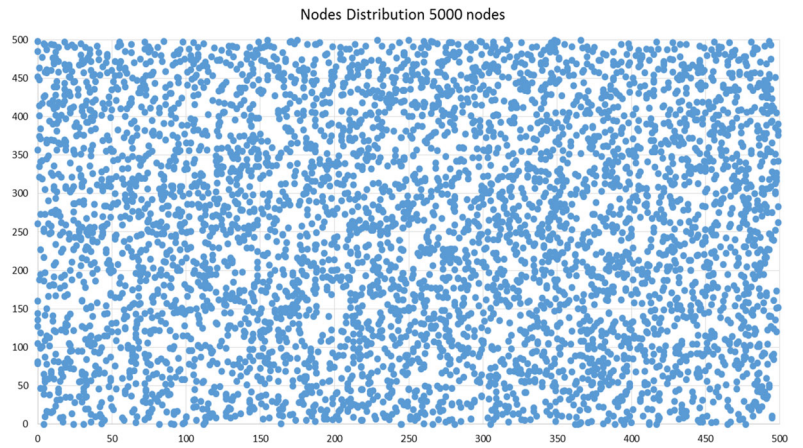
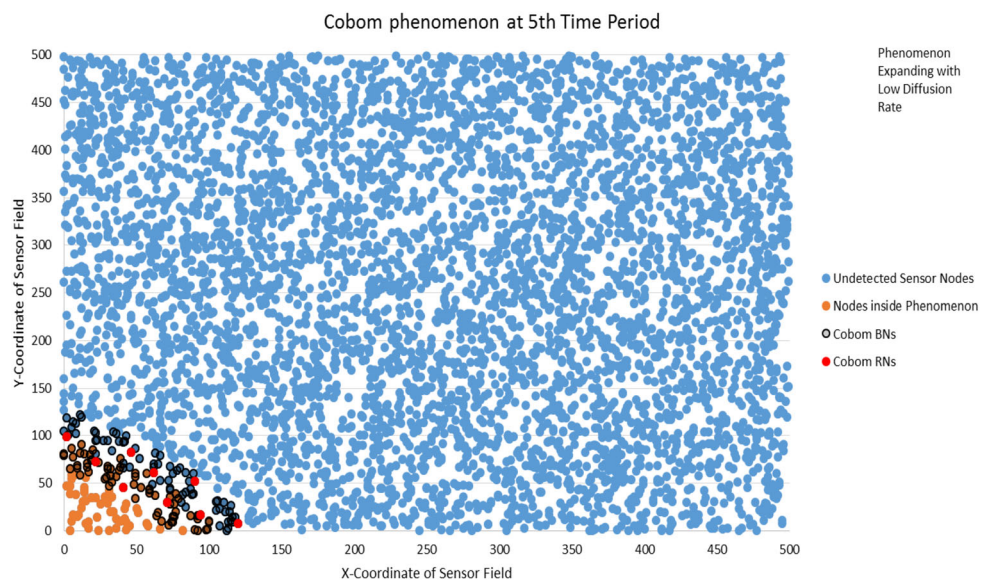


Fig. 10 Expansion of object in staircase pattern

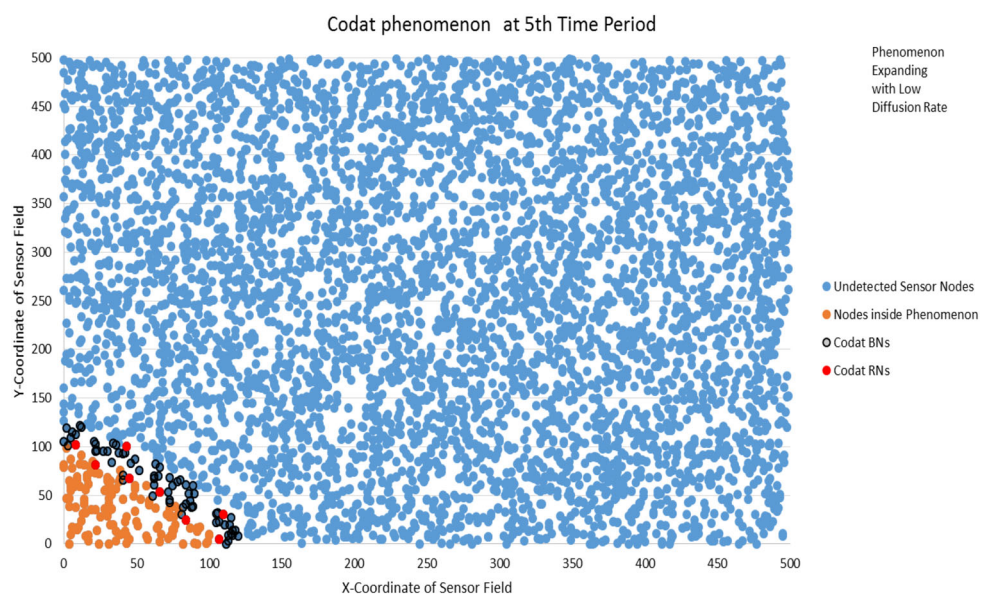
**Fig. 11** 5000 sensor nodes distribution in sensor field



**Fig. 12** COBOM phenomenon at 5th time period

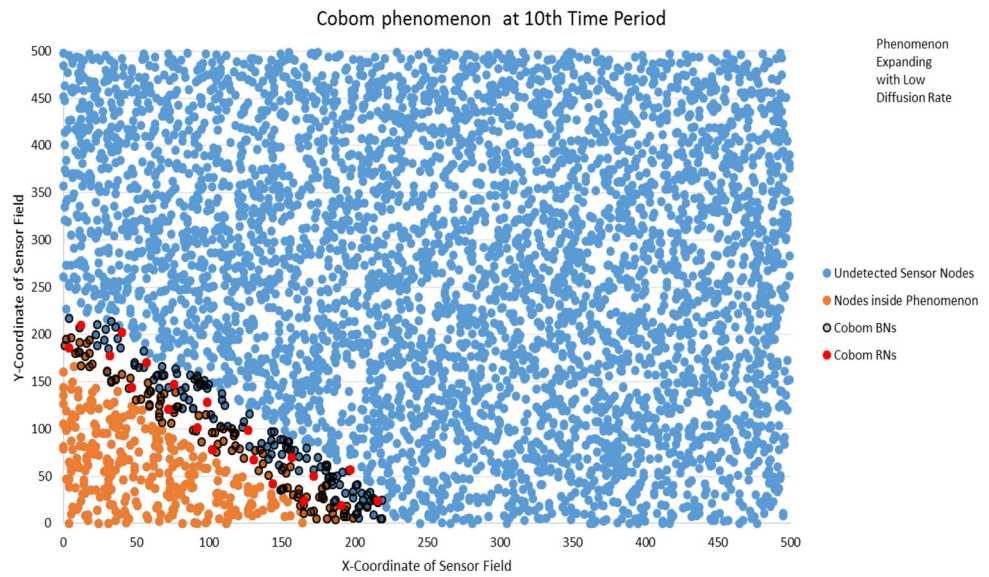


**Fig. 13** CODAT phenomenon at 5th time period

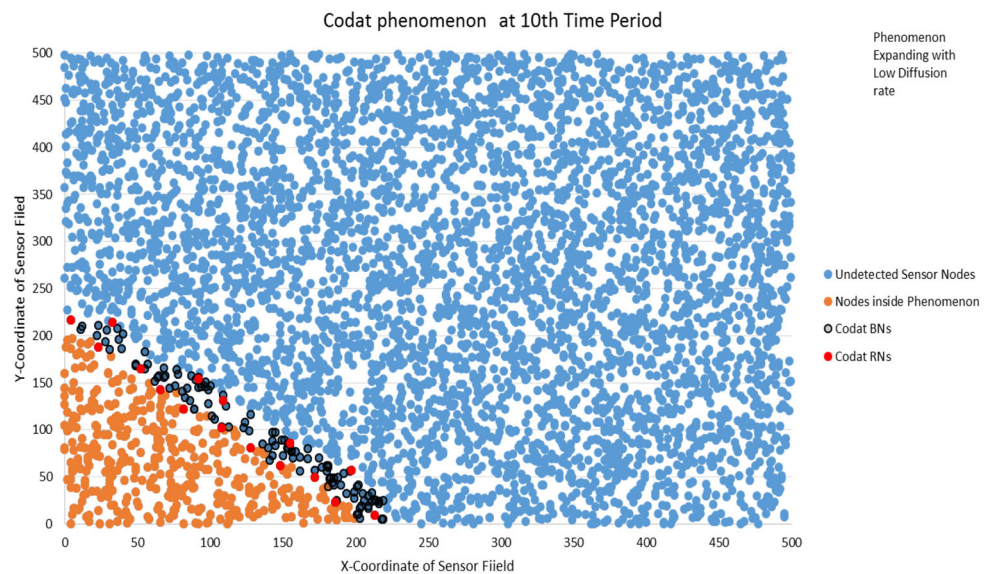




**Fig. 14** COBOM phenomenon at 10th time period



**Fig. 15** CODAT phenomenon at 10th time period



**4.1 Comparative analysis based on number of BNs**

The number of BNs generated is shown in the Figs. 20, 21, 22 and 23 with varying density along with variations in communication range of sensor nodes. The number of BNs increases with each period up to the 25th period for continuous phenomenon expansion. The higher the number of BNs generated by the algorithm, the more energy consumed by the sensor network.

The Figs. 20 and 21 show that the number of BNs generated by COBOM are far more compared to CODAT and DEMOCO for both 15 and 25 m communication range respectively for dense WSNs.

Similarly, Figs. 22 and 23 show that with the increase in expansion of phenomenon more number of BNs are

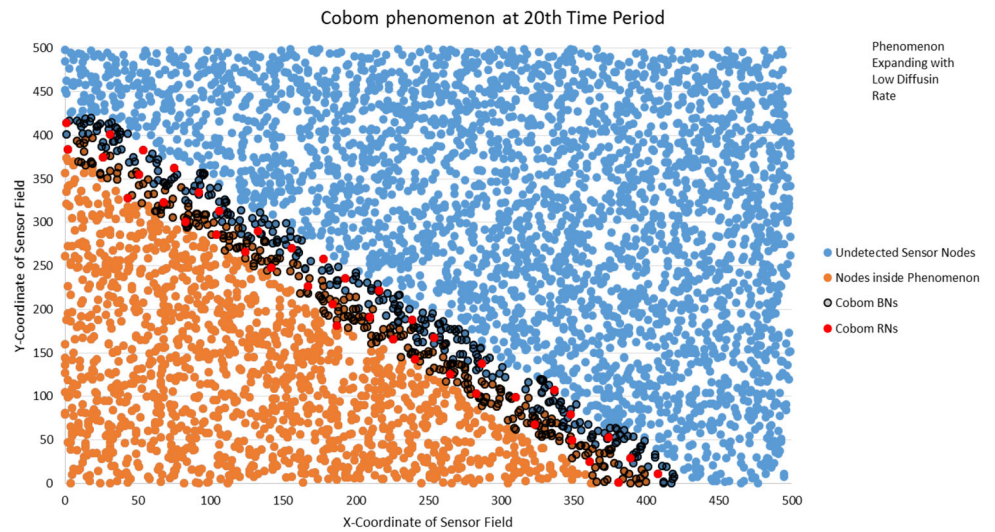
generated and COBOM produces more number of BNs compared to CODAT and DEMOCO for sparse WSNs.

**4.2 Comparative analysis based on number of RNs**

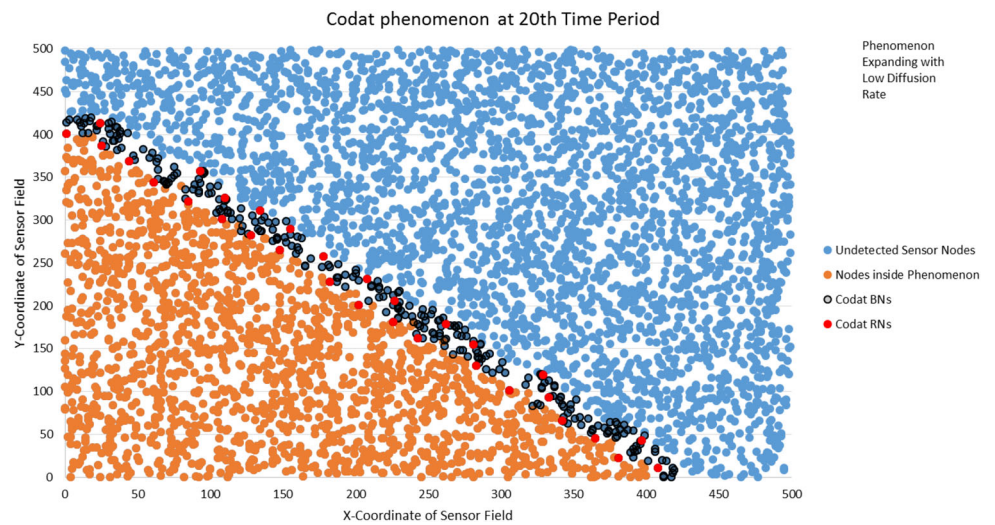
In our simulation, the number of RNs generated is demonstrated in Figs. 24, 25, 26 and 27 for both dense and sparse WSN.

Higher the number of RNs generated by the algorithm, higher will be the energy consumed by the network. In Figs. 24, 25, 26 and 27, it is clearly that demonstrated that the number of BNs and RNs generated by the COBOM is more as compared to DEMOCO and our proposed algorithm CODAT. This is because COBOM generated BNs and RNs on both sides of the boundary. Our proposed

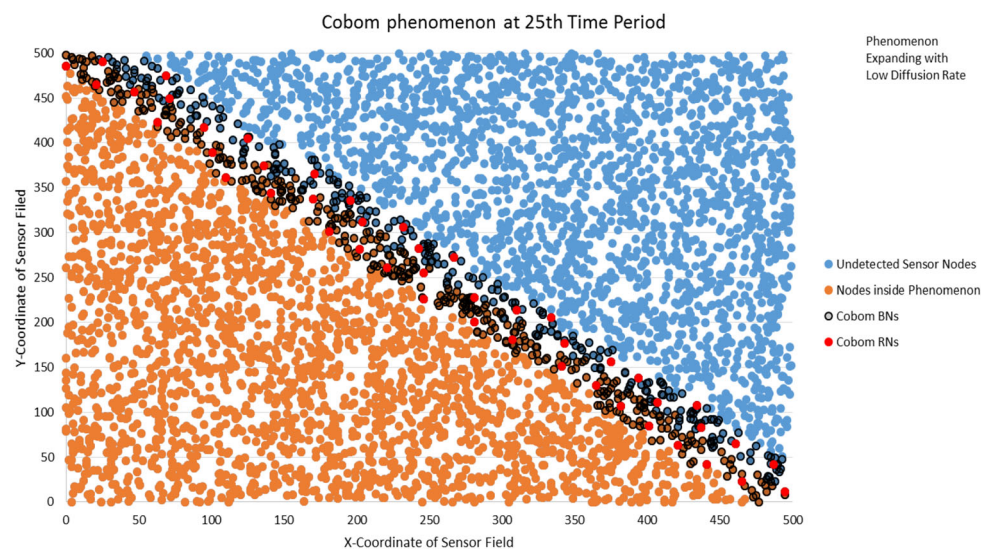
**Fig. 16** COBOM phenomenon at 20th time period



**Fig. 17** CODAT phenomenon at 20th time period

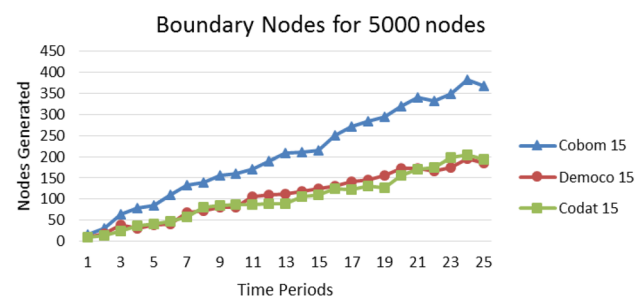
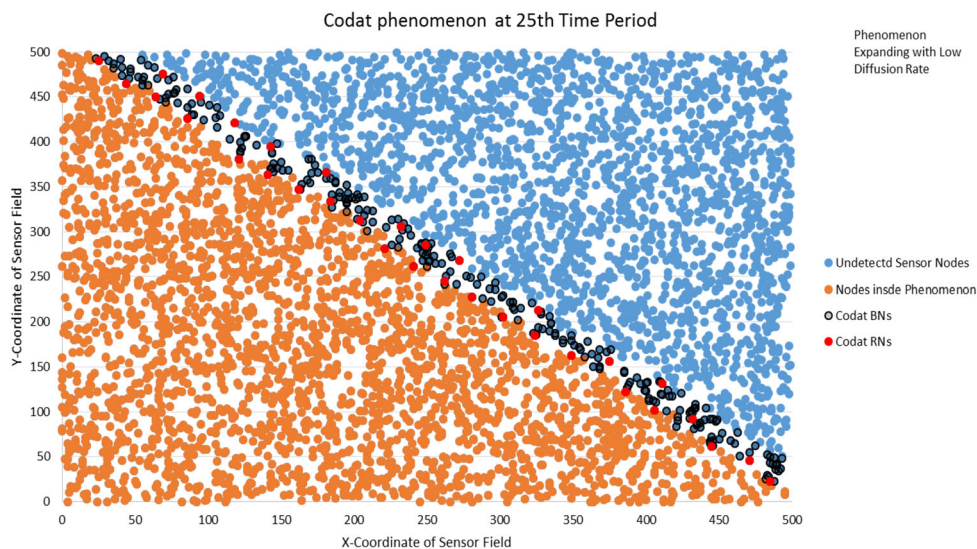


**Fig. 18** COBOM phenomenon at 25th time period

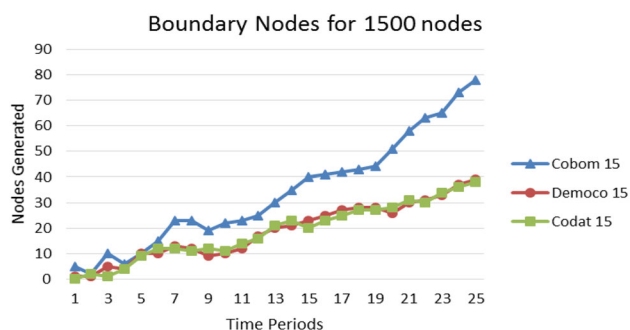




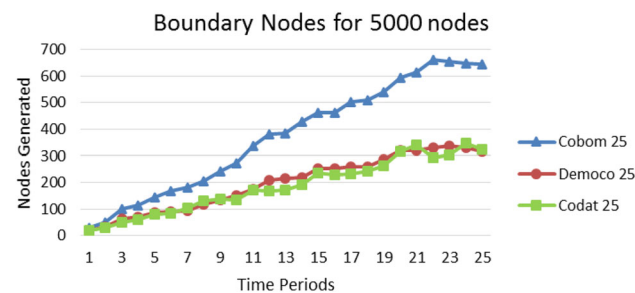
**Fig. 19** CODAT phenomenon at 25th time period



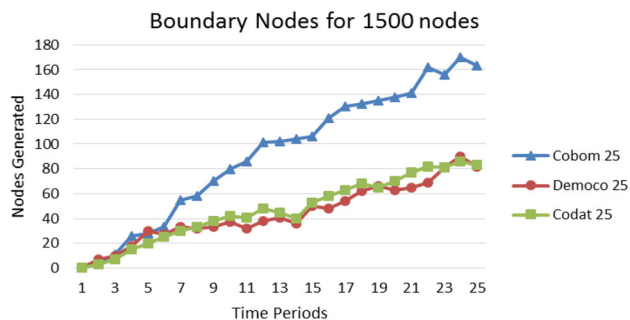
**Fig. 20** BNs for 5000 nodes with 15 m communication range



**Fig. 22** BNs for 1500 nodes with 15 m communication range



**Fig. 21** BNs for 5000 nodes with 25 m communication range



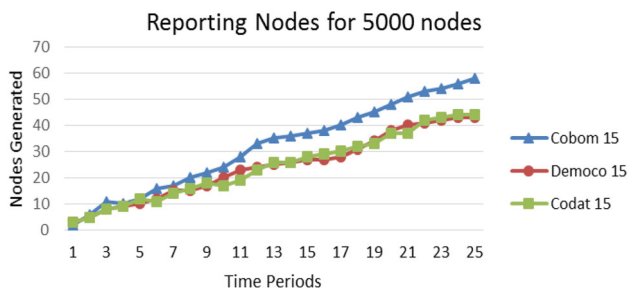
**Fig. 23** BNs for 1500 nodes with 25 m communication range

algorithm CODAT is similar to DEMOCO algorithm that generated BNs and RNs on the outer region only when objects are expanding. The CODAT also generated BNs and RNs on the inner region only when the objects are shrinking. The difference, in BNs and RNs, between COBOM and CODAT becomes apparent when the density of the sensor nodes is 5000 and the communication range is 25 m. These facts are demonstrated in Figs. 21 and 25. We also observed that the number of RNs generated is increased with sparse setting and communication range of 25 m. The number of RNs generated is also increased with

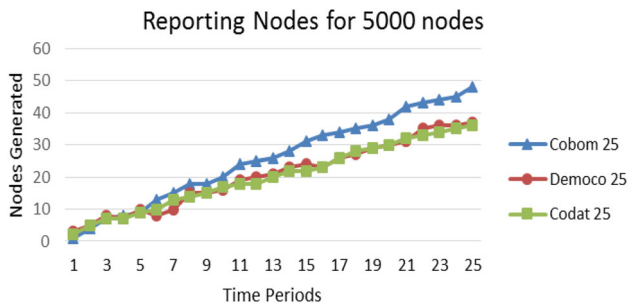
dense setting and communication range of 15 m. We noticed that there is no strong relationship between communication range and RNs.

### 4.3 Simulating the holes along with overall phenomenon

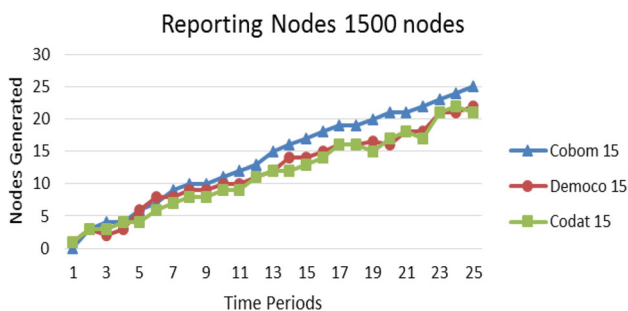
Although most of the algorithms can differentiate between the holes and overall phenomena. Our algorithm can



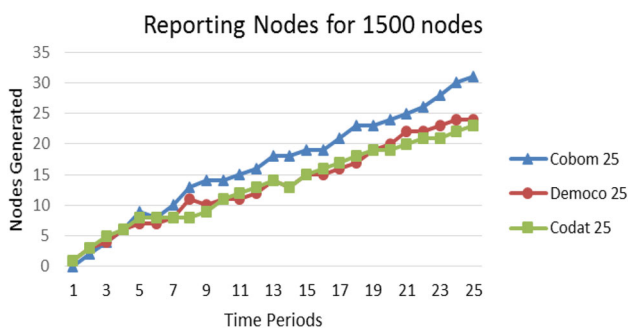
**Fig. 24** RNs for 5000 nodes with 15 m communication range



**Fig. 25** RNs for 5000 nodes with 25 m communication range



**Fig. 26** BNs for 1500 nodes with 15 m communication range



**Fig. 27** BNs for 1500 nodes with 25 m communication range

provide an additional information as an added improvement forwarded to the user along with the reported data with more accuracy and with less number of reporting nodes compared to COBOM. In order to analyze that our

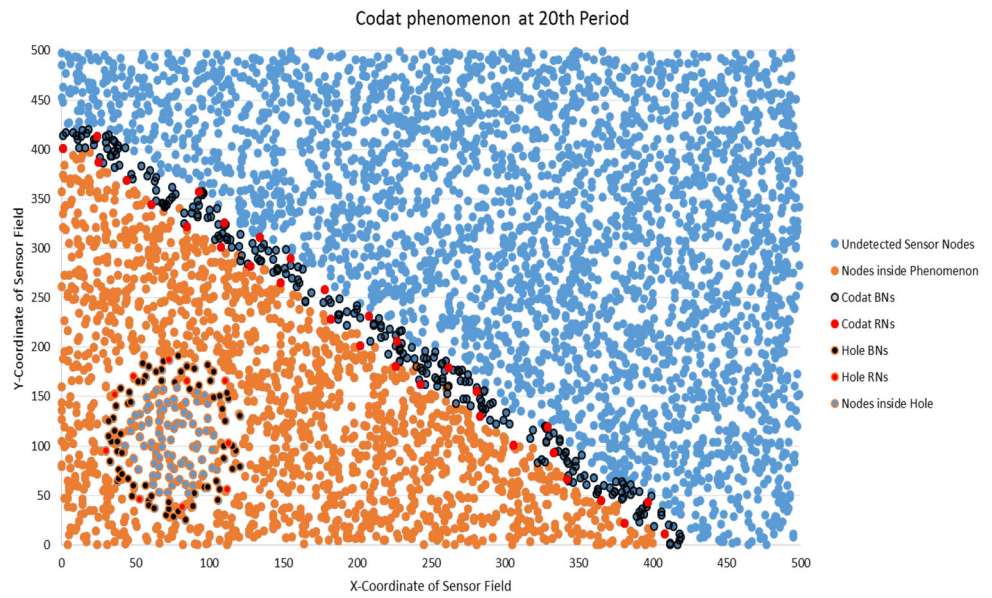
proposed algorithm can differentiate at the sink level between the holes and the overall phenomena, we are considering a special case in which staircase structure and hole both are expanding. In case of Staircase structure expansion, nodes move from undetected state to detected state and in case of Hole Expansion, nodes move from detected state to undetected state. We initiated a single hole (perfect circle in java) at 15th time period. The hole is centered at (76,106). The hole expands with each passing period with increase in size of 3 m in each time period (each time period of 5 s). Along with the hole, the staircase structure is also simulated which expands up to 25th time period. The center of the hole is chosen at (76,106) so that the Hole and the staircase phenomena do not merge into each other. Now both the hole and the staircase structure after reaching to 20th period, the total number of reporting nodes sent to the sink are calculated shown in Figs. 28 and 30. After that both the staircase structure and the hole further expand up to 25th period and again the number of reporting nodes sent to the sink are calculated shown in Figs. 29 and 30.

As shown in the Fig. 30, our algorithm CODAT can provide additional information as an added improvement forwarded to the user to differentiate between the reported data sent by the outer boundary (OB) or overall phenomenon and the reported data sent by the inner boundary (IB) or hole due to Phenomena Tags. Similarly, same results can also be deduced for shrinkage of staircase phenomenon and hole.

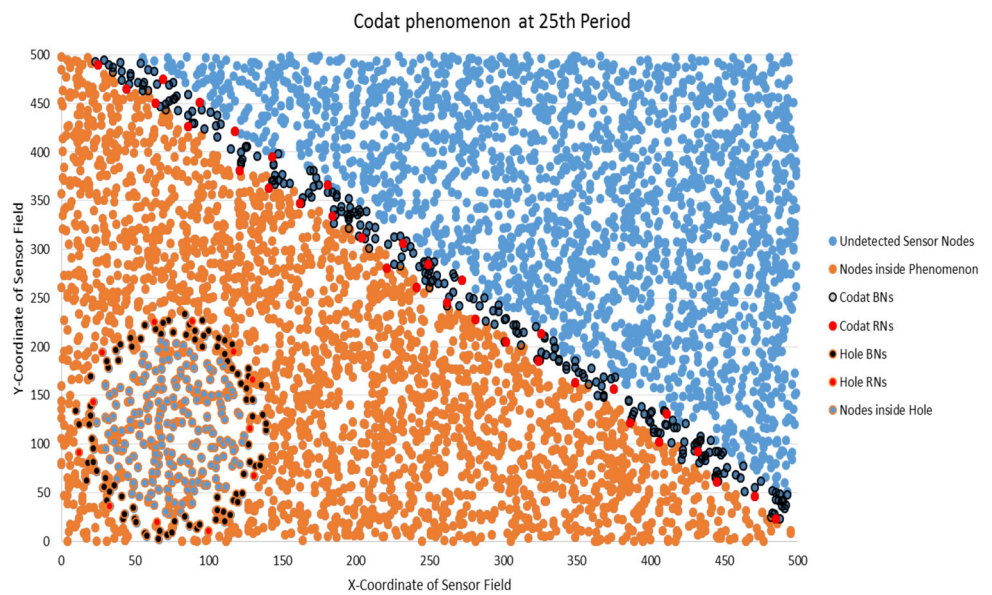
#### 4.4 Comparative analysis based on average report data size

Before calculating the size of average report data for each algorithm, some assumptions are made for simplicity purposes. The results are collected for both sparse (1500 nodes) and dense settings (5000 nodes) with a communication range of 25 m. We assumed that the unique ID of each sensor node is 2 bytes in size. In our proposed algorithm CODAT, the RN sends its own unique ID, first 2 BNs unique IDs received through PCM, and a BN-Array (containing status readings and tags). The RN in COBOM sends its own unique ID and BN-Array. Since the size of BN-Array depends on the RN's communication range and the density of the deployed nodes; therefore, the greater the density of the sensor field the larger will be the BN-Array. In our algorithm CODAT, the number of BNs and RNs are almost half as compared to BNs and RNs in COBOM. Table 3 shows the average report data size of CODAT that does not exceed COBOM in dense settings with 5000 nodes. The report data size of CODAT is greater than DEMOCO, because DEMOCO does not use BN-Array. However, the main drawback of DEMOCO lies in sending

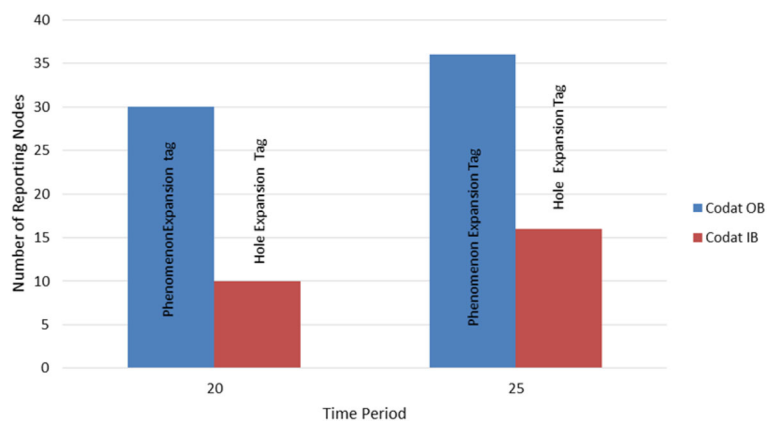
**Fig. 28** CODAT phenomenon and hole at 20th time period



**Fig. 29** CODAT phenomenon and hole at 25th time period



**Fig. 30** Differentiating between the inner boundary and outer boundary





**Table 2** Original results comparison average report data size for 5000 nodes

Object's time period	DEMOCO original bytes	DEMOCO bytes	COBOM Original bytes	COBOM bytes
1	1.692	2.622	3.653	3.645
2	6.546	6.432	13.999	14.002
3	11.526	13.563	24.631	25.432
4	14.376	15.624	30.736	30.431
5	16.503	17.364	35.209	36.252
6	18.243	19.328	38.782	38.521
7	20.304	21.651	43.304	44.652
8	22.368	22.632	47.602	48.524
9	24.435	25.563	52.308	53.152
10	26.61	27.462	56.673	57.342
11	28.668	29.473	61.052	62.452
12	30.657	31.462	65.293	66.373
13	32.685	33.564	69.606	70.352
14	34.902	35.736	74.289	75.262
15	36.741	37.631	78.148	83.463
16	38.637	39.621	82.185	86.983
17	40.671	41.431	86.596	92.146
18	42.807	43.651	91.123	96.473
19	44.739	44.982	95.137	100.183
20	46.803	46.961	99.52	104.454
21	48.668	49.243	103.432	108.474
22	50.673	51.432	107.617	113.464
23	52.851	53.781	112.305	116.464
24	54.51	56.193	115.869	119.474
25	56.511	57.453	120.06	121.632
Total	803.127	823.855	1708.85	1769.60

of report data that includes the nearest neighbor's ID. This nearest neighbor is estimated based on the signal strength that does not provide accurate results.

The average report data size for CODAT is higher as compared to DEMOCO and COBOM, as shown in Table 5. This is due to sparse settings with 1500 nodes. In sparse settings, COBOM performs better as compared to CODAT because its BN-Array does not contain any neighboring IDs. The average report data size is calculated using following Eq. 2.

$$\tau = \sum_{i=2}^n i.U_i \quad (2)$$

Where,  $i$  and  $U$  determine the average report data size.  $i$  refers to the number of bytes of report data and  $U$  refers to the average number of RNs corresponding to the  $i$ .  $\tau$  refers to the average report data size for each period. The total report data size is calculated by summing up the average report data size for all periods shown in Tables 3 and 5.

**Table 3** Average report data size for 5000 nodes

DEMOCO Bytes	COBOM Bytes	CODAT Bytes
2.622	3.645	3.543
6.432	14.002	10.736
13.563	25.432	18.363
15.624	30.431	24.374
17.364	36.252	27.873
19.328	38.521	31.736
21.651	44.652	34.733
22.632	48.524	37.373
25.563	53.152	40.635
27.462	57.342	43.625
29.473	62.452	48.732
31.462	66.373	52.262
33.564	70.352	56.367
35.736	75.262	59.362
37.631	83.463	63.362
39.621	86.983	68.373
41.431	92.146	72.273
43.651	96.473	76.363
44.982	100.183	80.373
46.961	104.454	83.272
49.243	108.474	87.372
51.432	113.464	90.373
53.781	116.464	93.272
56.193	119.474	97.387
57.453	121.632	100.102
823.855	1769.602	1401.236

Also the original results are replicated from the paper (Jung-Hwan et al. 2008) and compared with our simulated results shown in Tables 2 and 4. Our simulated results for average report data size for 5000 nodes for DEMOCO and COBOM can be compared with original results in Table 2. Our simulated results are almost 97 % closed to that of the original paper for both DEMOCO and COBOM. Similarly, for average report data size for 1500 nodes, we compared our simulated results with the original results. We achieved about 98 % similarity to that of original results for both DEMOCO and COBOM as shown in Table 4.

#### 4.5 Boundary accuracy with different diffusion rates of object

In order to analyze the accuracy of our algorithm in detecting the boundary of the continuous object, we simulated about 5000 sensor nodes uniformly distributed over  $500 \times 500 \text{ m}^2$  with 25 m communication range for each sensor node. Accuracy is computed by comparing the reported data at the boundary with that of sink at same time so that we can know that either the actual boundary of the



**Table 4** Original results comparison average report data size for 1500 nodes

Object's time period	DEMOCO original bytes	DEMOCO bytes	COBOM original bytes	COBOM bytes
1	0.405	2.404	0.404	2.421
2	1.776	3.767	1.798	3.71
3	4.038	4.04	4.053	4.05
4	6.864	6.922	6.826	6.67
5	9.318	9.55	9.196	9.811
6	11.091	11.095	10.954	10.391
7	12.522	12.571	12.332	12.491
8	14.004	14.244	13.791	13.998
9	15.639	15.537	15.422	15.501
10	17.436	17.692	17.149	17.718
11	19.05	19.127	18.714	18.621
12	20.611	20.73	20.215	20.321
13	22.2	22.491	21.782	21.971
14	24.06	24.81	23.64	23.86
15	25.503	25.525	25.048	25.029
16	27.902	28.13	26.752	26.964
17	28.902	29.3	28.39	28.36
18	30.267	30.393	29.724	29.721
19	31.956	31.965	31.379	31.357
20	33.552	33.635	32.951	32.957
21	35.151	35.602	34.512	34.691
22	36.831	36.846	36.151	36.158
23	38.427	38.018	37.73	37.7
24	39.78	39.8	39.02	39
25	41.352	41.435	40.628	40.621
Total	547.962	551.698	538.566	544.09

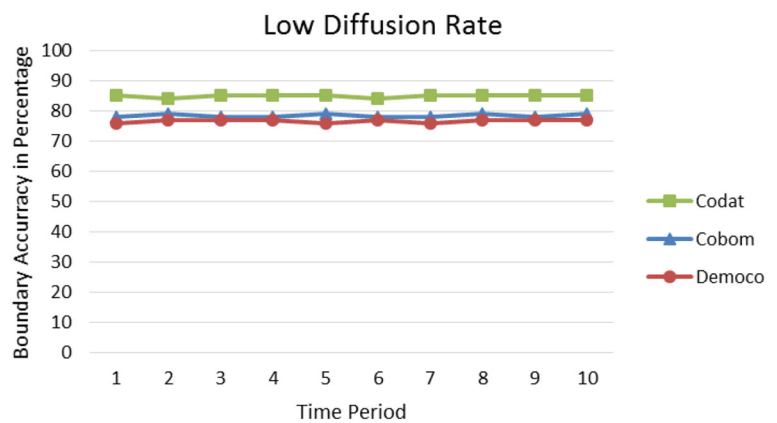
**Table 5** Average report data size for 5000 nodes

DEMOCO bytes	COBOM bytes	CODAT bytes
2.404	2.421	3.405
3.767	3.71	3.772
4.04	4.05	4.09
6.922	6.67	6.98
9.55	9.811	9.89
11.095	10.391	11.192
12.571	12.491	12.862
14.244	13.998	14.465
15.537	15.501	15.865
17.692	17.718	17.834
19.127	18.621	19.432
20.73	20.321	20.789
22.491	21.971	23.211
24.81	23.86	24.258
25.525	25.029	25.512
28.13	26.964	28.341
29.3	28.36	29.262
30.393	29.721	30.321
31.965	31.357	32.291
33.635	32.957	33.532
35.602	34.691	35.783
36.846	36.158	36.971
38.018	37.7	38.957
39.8	39	40.761
41.435	40.621	41.89
551.698	544.09	556.666

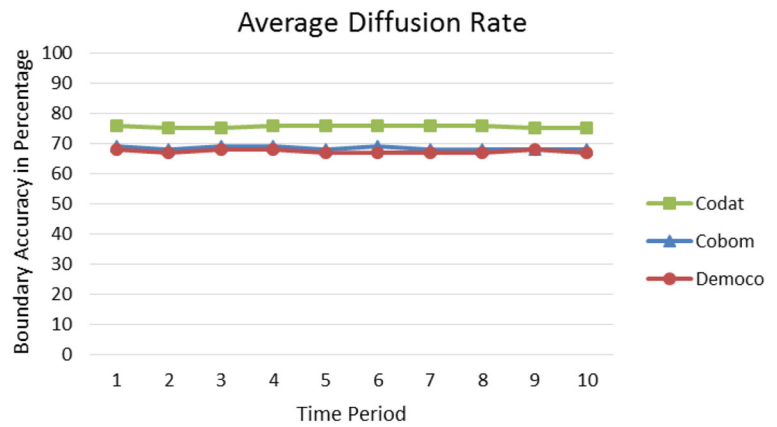
object exists at the specific location or not, when the sink receives the reported data. We set three different speeds for the continuous object (staircase structure) with low diffusion rate, average diffusion and high diffusion rate. Each time period is of 5 s and the reported data reaches to the

sink in 4 s. For low diffusion rate the object expands about 3 m in each time period. For average diffusion rate, the object expands about 7 m in each time period and for high diffusion rate, the object expands about 11 m in each time period. The first 10 time periods are used in order to calculate the preciseness of the boundary. At the end of each time period, boundary preciseness is calculated.

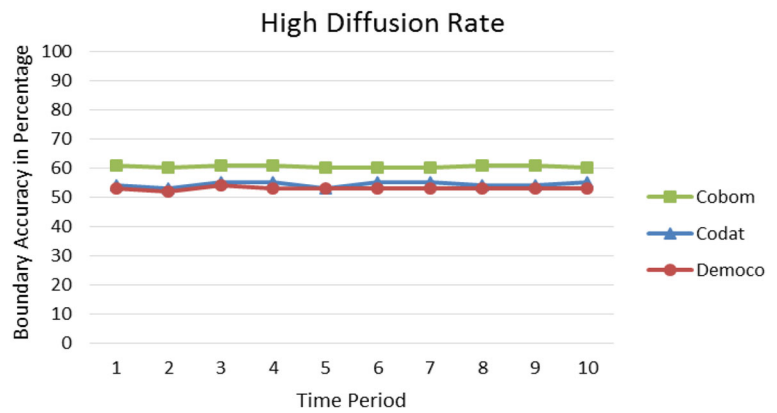
**Fig. 31** Boundary accuracy with low diffusion rate of object



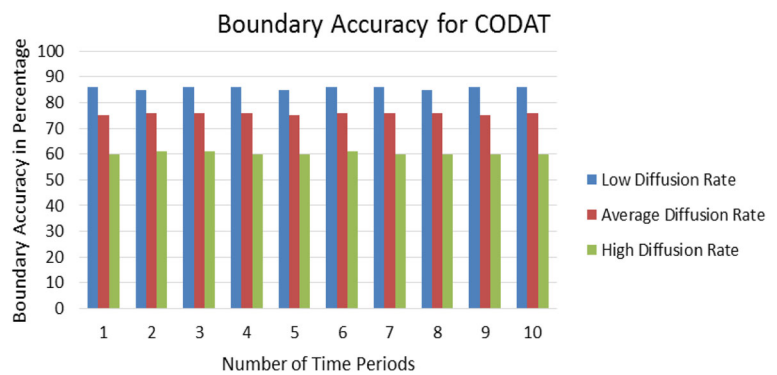
**Fig. 32** Boundary accuracy with average diffusion rate of object



**Fig. 33** Boundary accuracy with high diffusion rate of object



**Fig. 34** Boundary accuracy of CODAT with different diffusion rates of object



4.5.1 Boundary accuracy with low diffusion rate of object

Figure 31 shows the accuracy of the boundary detection of the object expanding at low diffusion rate i.e., expands 3 meters in each time period. The x axis of the Fig. 31 represents the number of time periods and the y axis represents the accuracy in percentage. As we can see that CODAT has high accuracy of about 86 % because of more number of IDs compared to COBOM and DEMOCO. COBOM’s and DEMOCO’s accuracies are almost 80 %.

4.5.2 Boundary accuracy with average diffusion rate of object

Figure 32 shows the accuracy in boundary detection of the object expanding with average diffusion rate i.e., 7 m in each time period. The x axis represents the number of periods and the y axis represents the boundary accuracy in percentage. It can be noticed in the Fig. 32 that accuracy for all the algorithms have decreased compared to the low diffusion rate. Our proposed approach CODAT’s accuracy has decreased to

76 % compared to low diffusion rate. COBOM's and DEMOCO's accuracies have decreased to 68 %.

#### 4.5.3 Boundary accuracy with high diffusion rate of object

The Fig. 33 shows the accuracy in boundary detection of the object expanding with high diffusion rate i.e., object expanding 11 m in each time period. The x axis represents number of time periods and the y axis represents the boundary accuracy in percentage. Accuracy for all the three algorithms COBOM, CODAT and DEMOCO is calculated at each time period. It is noticed that with the increase in diffusion rate, the overall accuracy of all the algorithms have decreased. Our proposed approach CODAT's accuracy has decreased to 60 % due to high diffusion rate of phenomenon. COBOM's and DEMOCO's accuracies have decreased to 52 %.

#### 4.5.4 Boundary accuracy of CODAT with different diffusion rates of object

Figure 34 shows the overall picture of the performance of CODAT with different diffusion rates. The x axis represents the time periods and the y axis represents the boundary accuracy in percentage. If we notice the results in detail we can see that with low diffusion rate of object, CODAT's accuracy is up to 86 %, with average diffusion rate of object, CODAT's accuracy is up to 76 % and with high diffusion rate of object, CODAT's accuracy is up to 60 %. This shows that the diffusion rate of object has direct impact on the accuracy. Higher the diffusion rate of the phenomenon, lower will be the accuracy. Similarly, lower the diffusion rate of the phenomenon, higher will be the accuracy.

## 5 Conclusion and future work

In this paper, we proposed an energy efficient algorithm named CODAT for the detection and tracking of continuous phenomenon. More specifically, it is developed for continuous expansion and shrinkage of a phenomenon, splitting and merging of phenomenon and monitoring holes inside the phenomenon. Moreover, we introduced the concept of phenomenon tags to differentiate between phenomenon changes and holes inside the phenomenon as an added information forwarded to the user while maintaining the high accuracy and efficiency. CODAT is an improved hybrid algorithm integrating the detection techniques of both previous algorithms COBOM and DEMOCO. CODAT outperforms COBOM in terms of efficiency and outperforms both COBOM and DEMOCO in terms of accuracy. In the future, we plan to test our

algorithm in a real sensor environmental setup in order to analyze it under different environmental conditions.

## References

- Akyildiz IF, Weilian S, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. *Commun Mag IEEE* 40:102–114. doi:10.1109/MCOM.2002.1024422
- Amirjavid F, Bouzouane A, Bouchard B (2012) Activity modeling under uncertainty by trace of objects in smart homes. *J Ambient Intell Humaniz Comput* 5:159–167. doi:10.1007/s12652-012-0156-5
- Chang WR, Lin HT, Cheng ZZ (2008) CODA: a continuous object detection and tracking algorithm for wireless ad hoc sensor networks. In: Consumer communications and networking conference, CCNC 2008. 5th IEEE, pp 168–174
- Chaudhary SH, Bashir AK, Park MS (2008) [ETCTR] efficient target localization by controlling the transmission range in wireless sensor networks. In: Networked computing and advanced information management, 2008. NCM'08. Fourth international conference. IEEE, pp 3–7
- Chen W-P, Hou JC, Sha L (2004) Dynamic clustering for acoustic target tracking in wireless sensor networks. *Mob Comput IEEE Trans* 3:258–271
- Chintalapudi KK, Govindan R (2003) Localized edge detection in sensor fields. *Ad Hoc Netw* 1:273–291
- Ding M, Chen D, Xing K, Cheng X (2005) Localized fault-tolerant event boundary detection in sensor networks. In: INFOCOM 2005. 24th annual joint conference of the IEEE computer and communications societies. Proceedings IEEE, pp 902–913
- Fuemmeler JA, Veeravalli VV (2010) Energy efficient multi-object tracking in sensor networks. *Signal Process IEEE Trans* 58:3742–3750
- Hellerstein JM, Hong W, Madden S, Stanek K (2003) Beyond average: toward sophisticated sensing with queries. In: Information processing in sensor networks. Springer, pp 63–79
- Hong H, Oh S, Lee J, Kim SH (2013) A chaining selective wakeup strategy for a robust continuous object tracking in practical wireless sensor networks. In: Advanced information networking and applications (AINA), 2013 IEEE 27th international conference, IEEE, pp 333–339
- Hong S-W, Ryu H-Y, Park S, Kim S-H (2015) Reliable continuous object tracking with cost effectiveness in wireless sensor networks. In: Ubiquitous and future networks (ICUFN), 2015 seventh international conference, IEEE, pp 672–676
- Hussain CS, Park M-S, Bashir AK, Shah SC, Lee J (2013) A collaborative scheme for boundary detection and tracking of continuous objects in WSNs. *Intell Autom Soft Comput* 19:439–456
- Ji X, Zha H, Metzner JJ, Kesidis G (2004) Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks. In: Communications, 2004 IEEE international conference. IEEE, pp 3807–3811
- Jung-Hwan K, Kee-Bum K, Chaudhary SH, Wencheng Y, Myong-Soon P (2008) DEMOCO: energy-efficient detection and monitoring for continuous objects in wireless sensor networks. *IEICE Trans Commun* 91:3648–3656
- Kaplan LM, Le Q, Molnár P (2001) Maximum likelihood methods for bearings-only target localization. In: Acoustics, speech, and signal processing, 2001. Proceedings (ICASSP'01). 2001 IEEE International Conference. IEEE, pp 3001–3004
- Kotidis Y (2005) Snapshot queries: towards data-centric sensor networks. In: Data engineering. ICDE 2005. Proceedings 21st international conference. IEEE, pp 131–142

- Krishnamachari B, Wicker SB, Bejar R (2001) Phase transition phenomena in wireless ad hoc networks. In: Global telecommunications conference. GLOBECOM'01. IEEE, pp 2921–2925
- Learned RE, Karl WC, Willsky AS (1992) Wavelet packet based transient signal classification. In: Time-frequency and time-scale analysis. Proceedings of the IEEE-SP international symposium. IEEE, pp 109–112
- Li D, Wong KD, Hu YH, Sayeed AM (2002) Detection, classification, and tracking of targets. *Signal Process Mag IEEE* 19:17–29
- Mainwaring A, Culler D, Polastre J, Szewczyk R, Anderson J (2002) Wireless sensor networks for habitat monitoring. In: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications. ACM, pp 88–97
- Malik H, Malik AS, Roy CK (2011) A methodology to optimize query in wireless sensor networks using historical data. *J Ambient Intell Humaniz Comput* 2:227–238. doi:[10.1007/s12652-011-0059-x](https://doi.org/10.1007/s12652-011-0059-x)
- Meng X, Li L, Nandagopal T, Lu S (2004) Event contour: an efficient and robust mechanism for tasks in sensor networks. Proceedings of Technical report. pp 1–13
- Park H, Oh S, Lee E, Park S, Kim S-H, Lee W (2012) Selective wakeup discipline for continuous object tracking in grid-based wireless sensor networks. In: Wireless communications and networking conference (WCNC). IEEE, pp 2179–2184
- Salamah M, Zawaideh F (2013) Optimal object tracking via wireless sensor networks. In: Electronics, computer and computation (ICECCO), 2013 International Conference. IEEE, pp 273–276
- Shen J, Han G, Jiang J, Sun N, Shu L (2015) An energy-efficient tracking scheme for continuous objects in duty-cycled wireless sensor networks. In: Consumer electronics-Taiwan (ICCE-TW), 2015 IEEE international conference. IEEE, pp 150–151
- Silberstein A, Braynard R, Yang J (2006) Constraint chaining: on energy-efficient continuous monitoring in sensor networks. In: Proceedings of the 2006 ACM SIGMOD international conference on management of data. ACM, pp 157–168
- Solis I, Obraczka K (2005a) Efficient continuous mapping in sensor networks using isolines. In: Mobile and ubiquitous systems: networking and services, MobiQuitous 2005. The second annual international conference. IEEE, pp 325–332
- Solis I, Obraczka K (2005b) Isolines: energy-efficient mapping in sensor networks. In: computers and communications. ISCC 2005. Proceedings 10th IEEE symposium. IEEE, pp 379–385
- Tang J, Wang Z, Du C, Zhou Z, Sun Y (2014) Object tracking in wireless sensor networks using an itinerary-based method. In: Communications and networking in China (CHINACOM), 9th international conference. IEEE, pp 38–43
- Thangarajan T, Sakthivel P, Padmanaban JB (2013) An energy efficient technique for object tracking in wireless sensor networks. In: Communication systems and network technologies (CSNT), 2013 international conference. IEEE, pp 316–321
- Ugolotti R, Sassi F, Mordonini M, Cagnoni S (2011) Multi-sensor system for detection and classification of human activities. *J Ambient Intell Humaniz Comput* 4:27–41. doi:[10.1007/s12652-011-0065-z](https://doi.org/10.1007/s12652-011-0065-z)
- Xu Y, Winter J, Lee W-C (2004a) Dual prediction-based reporting for object tracking sensor networks. In: Mobile and ubiquitous systems: networking and services. MOBIQUITOUS 2004. The first annual international conference. IEEE, pp 154–163
- Xu Y, Winter J, Lee W-C (2004b) Prediction-based strategies for energy saving in object tracking sensor networks. In: Mobile data management. Proceedings. 2004 IEEE international conference. IEEE, pp 346–357
- Zhang W, Cao G (2004a) DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks. *Wirel Commun IEEE Trans* 3:1689–1701
- Zhang W, Cao G (2004b) Optimizing tree reconfiguration for mobile target tracking in sensor networks. In: INFOCOM 2004. Twenty-third annual joint conference of the IEEE computer and communications societies. IEEE, pp 2434–2445
- Zhao F, Shin J, Reich J (2002) Information-driven dynamic sensor collaboration. *Signal Process Mag IEEE* 19:61–72
- Zhong C, Worboys M (2007) Energy-efficient continuous boundary monitoring in sensor networks. Technical report