This is the pre-peer reviewed version of the following article: D3: Distributed Approach for the Detection of Dumb Nodes in Wireless Sensor Networks, which has been published in final form at https://doi.org/10.1002/dac.2913.
This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.

D3: Distributed Approach for the Detection of Dumb Nodes in Wireless Sensor Networks

Arijit Roy, Pushpendu Kar, Sudip Misra
School of Information Technology
Indian Institute of Technology Kharagpur
Kharagpur 721302, India
Email: {arijitr, pkar, smisra}@sit.iitkgp.ernet.in

Mohammad S. Obaidat
Fellow of IEEE, *Fellow of SCS*
Dept. of Computer Science and Software Enggineering
Monmouth University
West Long Branch, NJ. USA
Email: obaidat@monmouth.edu

Abstract

In this work, we propose D3 – a distributed approach for the detection of "dumb" nodes in a wireless sensor network (WSN). A dumb node can sense its surroundings, but is unable to transmit these sensed data to any other node, due to the sudden onset of adverse environmental effects. However, such a node resumes its normal operations with the resumption of favorable environmental conditions. Due to the presence of dumb nodes, the network is unable to provide the expected services. Therefore, it is prudent to re-establish connectivity between dumb and other nodes, so that sensed data can be reliably transmitted to the sink. Before the re-establishment of connectivity, a node needs to confirm its actual state of being dumb. Dumb behavior is dynamic in nature, and is, thus, distinct from the traditional node isolation problem considered in stationary WSNs. Therefore, the existing schemes for the detection of other misbehaviors is not applicable for detecting a dumb node in a WSN. Considering this temporal behavior of a dumb node, we propose an approach, D3, for the detection of dumb nodes. The propose scheme we uses *Cumulative Sum* (CUSUM) test, which helps in detecting the dumb behavior. The simulation results show that, there is 56% degradation in detection percentage with the increment in the detection threshold whereas energy consumption and the message overhead increases by 40% with the increment in detection threshold.

Index Terms

Dumb Node, Environmental Effect, Detection, CUSUM, Markov Chain.

I. Introduction

The rapid development of Micro-Electro-Mechanical Systems (MEMS) technology has made deployment of low-cost wireless sensor networks (WSNs) feasible. Currently, WSNs are used extensively, in various application domain such as surveillance [13], [25], [35], health care monitoring [26], disaster management [21], and fire detection [18]. The sensor nodes deployed over an area to sense data in a distributed manner and transmit these to a centralized unit, termed as the sink [3], [4]. A WSN consists

For Personal Use Only

of a set of low-power sensor nodes, and limited transmission range. Consequently, the intermediate nodes forward the sensed information to the sink. Communication in a WSN takes place over network with single- or multi-hop connectivity. Therefore, active participation and collaboration of each node in the network is essential. WSNs are resource-constrained and are vulnerable to various types of misbehaviors and attacks such as Denial of Service (DoS) attacks, environmental effects, and faults. To handle the issues of misbehaviors and faults, a number of schemes exist in literature [7], [10], [14], [15]. A newly explored type of misbehavior is the dumb behavior [23], [29], [30]. When a node exhibits dumb behavior, the node unable to transmit its sensed data packet to any other nodes, due the shrinkage in communication range in the presence of adverse environmental effects. Thus when a node behave as dumb, it can sense its surrounding but unable to transmit the sensed data packet. As the adverse environmental effects are temporal in nature, a node resumes its normal operation on the onset of favorable environmental conditions.

A. Motivation

The presence of a dumb node causes detrimental effects on the performance of the networks. Therefore, the detection of dumb nodes in the network is important. Dumb nodes continue their sensing operation in the presence of environmental effects, but are unable to communicate with other nodes. On the resumption of favorable environmental conditions, the dumb nodes start performing normal operations. As dumb behavior is not permanent in nature, it is infeasible to eliminate dumb nodes from the network permanently. Subsequently, the connectivity between the dumb and other nodes requires re-establishment, so that such nodes can transmit the sensed data to the sink. Before re-establishment of connectivity of a dumb node with other nodes, it is essential to detect whether a node is dumb. The temporal nature of dumb behavior of a node makes detection a non-trivial issue. We consider the activity of node's behavior in different time instants and provide a solution for dumb node detection.

B. Contribution

This paper centers around the newly proposed concept of the existence of dumb nodes. It is caused due to the sudden onset of adverse environmental effects, while the nodes behave normally with the resumption of these effects. The specific *contributions* in this work are summarized below:

- We propose a scheme for dumb node detection in WSN using the cumulative sum (CUSUM) approach.
- We analyze the detection problem using Markov chain.

For Personal Use Only

- The proposed solution has been rigorously theoretically characterized.
- The concept of aperiodic *HELLO* message has been introduced, considering the energy-constrained nature of WSNs.

The rest of the paper is organized as follows. Section II describes the related work done in this area. Section III includes the details of the problem description. Theoretical analysis of the solution approach is performed in Section IV. The performance of the proposed scheme is shown in Section V. Finally, we conclude the work in Section VI, giving directions for future research.

II. Related works

WSNs are vulnerable to different type of attacks, faults, and misbehaviors. Faults in WSNs is a common issue that impede a sensor node to perform normal operations. Different existing piece of literature [14], [17], [22], [31] have studied different aspects of this issue in WSNs. Sharma *et al*. [31] proposed a fault detection scheme with the help of a real test-bed. The authors focused on collecting faulty sensor readings. Subsequently, faults are detected through a combination of four method – Rule-based methods, Estimation methods, Time series analysis-based methods, and Learning-based methods. All of the methods are dependent on data measurement of the sensor. Luo *et al*. [22] proposed another fault detection scheme that primarily focuses on noise related measurement error and sensor fault. Krishnamachari *et al*. [17] proposed a distributed Bayesian algorithm for detecting sensor measurement, following which the correlation of those faults is computed. The authors detected faults using the heuristic that a faulty sensor node produces abnormal value, whereas a normal-behaved node produces low value for their sensing activity. A fault detection scheme called Sequence-Based Fault Detection (SBFD), was proposed by Kamal *et al*. [14]. According to the authors, SBFD is lightweight in terms of communication, the detection rate is high due to its distributed nature, and finally, SBFD detects fault with low latency. Guiyun *et al*. [12] have developed an indicator kriging estimator for predicting the unknown observation of data from fusion center in a WSN. Further the authors form an optimization problem in order to maintain the tradeoff between estimation performance and energy consumption.

In all of these works, the authors considered faults in the sensing unit of a node. In these works, after the recovery of actual data by a node itself using some algorithm, a node needs to transmit the data to the sink for further processing. Another process to recover actual data by the sink from the faulty data received from sensor nodes. Therefore, here the communication gets equal importance as data sensing. If

For Personal Use Only

in the existing work, we consider the existence of dumb nodes in the network, it is difficult to transmit such data to the sink or to any other node. This issue promogulates the significance of the problem of detection of dumb nodes, so that within those nodes, the connectivity re-establishment algorithm can be executed in order to recover the actual data.

Misbehaviors [7], [15] affect the performance of WSNs in the same way as faulty nodes do. Bao *et al*. [7] proposed a cluster-based hierarchical trust management scheme for WSNs. The scheme is applicable to selfish and malicious nodes. An analytical framework was proposed by Kannan *et al*. [15] for quantifying the impact of energy misbehavior on other nodes. In this work, the authors minimize the power consumption in two steps. First, they propose strategic power optimization, in which the nodes act as agents and work strategically to minimize the power consumption. Second, a joint power optimization scheme, in which a node jointly reduces the power consumption of the network. Soltanmohammadi *et al*. [32] proposed a solution for the detection of malicious nodes, using the theory of binary hypothesis testing. In the proposed solution, honest node transmits a binary decision to the fusion center, whereas a malicious node transmits fictitious messages to the fusion center. Further, the fusion center is used to identify the misbehaving nodes. Rajasegara *et al*. [28]. identified different types of anomalies, and thereafter, developed a statistical mode using real data. Conti *et al*. [11] studied clone attack, in which replicated nodes affect network activities. The authors proposed a solution for the detection of these replicated nodes in a distributed manner. Liu *et al*. [20] proposed a scheme for the detection of nodes under attack using spatial correlation, which assume there exist no prior knowledge about the node. The scheme is applicable in large-scale sensor networks. A work has been propose by Ahmed *et al*. [2] using Dempster Shafer Theory for the detection of internal attacks. Sometimes nodes act in a non-cooperative manner by not transmitting data packets sent by others. This type of selfish behavior and different solutions are reported in the existing literature [19]. Abid *et al*. [1] proposed a game theoretic scheme that encourages a node to cooperate in the network with goal of detection and prevention of selfish behavior of a node. The existing detection schemes of misbehaving, faulty, and selfish nodes are incapable for use with dumb nodes. This is because, due to its dynamic nature, it is difficult to communicate continuously with a node which is dumb.

Environmental impacts cause disruption in communication. The factors responsible for breaking of link among nodes, as reported in the literature [5], [6], [27] are temperature, rainfall, and fog. Boano *et*

For Personal Use Only

al. [9] showed how communication range gets affected due to increase in temperature. In the existing literature, it is reported that due to the presence of such environmental impacts, the communication range of a sensor node gets reduced, and it is unable to communicate with the other nodes. This affected node is characterized as dumb [23]. A node gets dumb when it can sense its surrounding, but is unable to transmit the sensed data. This misbehavior is not considered in existing literature. This makes the problem challenging. Therefore, the existing misbehavior and fault detection schemes are inapplicable for dumb node detection.

III. Problem Description

A. Objectives

In WSN, sensor nodes work in a collaborative manner to transmit data to the sink with single- or multi-hop connectivity. Due to the sudden onset of adverse environmental effects, the communication range of the sensor nodes decreases, and as a consequence of which they may not be able to communicate with the nearest active neighbor node. This results in a node being vulnerable to getting isolated from the network. It is a major concern to re-establish connectivity between dumb and other nodes, in order to get efficient services from the network. The connectivity re-establishment algorithm needs to starts either from an affected node or from a sink. Therefore, it is prudent to identify whether or not a node is dumb. The temporal nature of dumb behavior makes it difficult to identify an affected node and, thus, its detection is a challenge. This work attempt to address this challenge by providing a solution for the detection of dumb nodes in a WSN.

B. System Model

1) Network Architecture: We consider a static WSN consisting of homogeneous sensor nodes, i.e., each node having the same sensing and transmission characteristics. These nodes are GPS enabled and are deployed randomly over a terrain. We consider adverse environmental changes resulting in sever affects on the communication range of sensor nodes. Examples of environmental adversities include increased temperature, rainfall, and fog, which results in the shrinkage of communication range of sensor nodes.

2) Dumb Node: This work focus on the existence of dumb behavior [23] in sensor nodes. We propose a scheme to detect these nodes. A sensor node that can sense physical phenomena in its surroundings, and cannot transmit the sensed data to any other nodes at a certain instant of time due to the presence

For Personal Use Only

of adverse environmental conditions, but is able to transmit at a later instant with the resumption of favorable environmental conditions, is termed as a dumb node [23]. Such behavior is denoted by Ψ_d. Mathematically,

$$\Psi_d = \begin{cases} 1, & \{(0 < d_{min} \leq r_c(t_i) \leq R)\} \wedge \{0 \leq r_c(t_j) < d_{min} < R)\} \quad \forall t_i \forall t_j \quad t_i \neq t_j \\ 0, & otherwise \end{cases}$$

When the communication range of a node shrinks below the distance to its nearest neighbor node, it continues its sensing operation, but is unable to transmit the sensed data to any other node. Let the specified communication range of a node be R. At time instant t_i, the communication range is $r_c(t_i)$. Due to the presence of adverse environmental effects at a latter time instant t_j, the communication range becomes $r_c(t_j)$, such that $r_c(t_j) < d_{min}$. Let d_{min} be the distance to the nearest active neighbor of a node. In such a situation, the node is unable to transmit the sensed data to any other node. Consequently, the node becomes dumb.

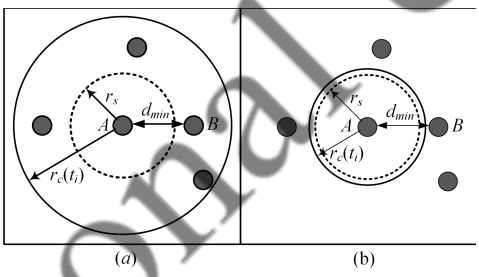


Fig. 1: Occurrence of dumb node

Fig. 1 pictorially depicts the occurrence of dumb node in a WSN. In Fig. 1(a), let node B be the nearest active neighbor of node A. The distance between A and B is d_{min}. In this case, the adverse effects of environment is not present. Consequently, node A can sense as well as communicate with its active neighbor nodes. However, Fig. 1(b) shows the situation when the adverse effect of environment is present. In this situation, the sensing range of node A is not affected due to the adverse environmental effects. On the other hand, the communication range gets affected and becomes $r_c(t_j)$. Thus, $r_c(t_j) < d_{min}$. Subsequently, node A cannot communicate with any of its active neighbor nodes. As a result, node A becomes dumb.

For Personal Use Only

C. Dumb Node Detector

To re-establish connectivity, it is essential to identify the dumb nodes in the network. In a saturated network, a node always has data packets to send. In this situation, a dumb node causes major impacts on the performance of the network. When a node exhibits dumb behavior, it gets isolated from the network. Consequently, the sink node is unable to communicate with the dumb node. In such a scenario, it is difficult to detect a dumb node using a centralized approach. Therefore, a node detects itself whether it is dumb or not.

1) Observation: The communication between a node and its neighbor nodes depends on the links present between them. The presence of different types of interference causes link breaks. We assume that there is no interference present, apart from the environmental interference in the network, due to which the links between any two nodes break. Let the number of neighbors of a node n be represented as N. In the proposed approach, *Observation* is an indicator O_n, whether a node n receives any acknowledgment against aperiodic *HELLO* message. Node n detects itself whether or not it is dumb. Node n broadcasts aperiodic *HELLO* messages, and it receives acknowledgments from each of its neighbor nodes. Let the probability of not receiving an acknowledgment from any of its neighbors be p_n. When the node becomes dumb, it is able to broadcast *HELLO* message, but is unable to reach any of its neighbor node, and consequently, a dumb node does not receive any acknowledgment from any of its neighbor nodes. The packet format of the *HELLO* message is shown in Fig. 2.

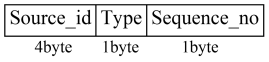


Fig. 2: Packet format of *HELLO* message

The probability distribution of O_n is given by:

$$P\{O_n = \rho\} = \begin{cases} p_n, & \text{if} \quad \rho = +1 \\ 1 - p_n, & \text{if} \quad \rho = -1 \end{cases} \tag{1}$$

Discussion. In WSNs, the sink plays a crucial centralized role for controlling network functionality. Therefore, centrally controlled algorithms run on the sink. On the other hand, distributed algorithms run on individual nodes and control the overall network operations collaboratively. As a dumb node isolates from the network, the detection of such nodes using a centralized approach is non-trivial. Therefore, a

For Personal Use Only

node detects itself as dumb and initiates the distributed connectivity re-establishment procedure. A node uses a *Detector*, which is mentioned in the following section

2) Detector: A node observes the value of O_n at each time, when it broadcasts a *HELLO* message. The *Detector* is based on the well-known method, *Cumulative Sum* (*CUSUM*) [8], [16], [33]. D is initialized with 0, and then it updates its own value by adding the observation value O_n.

Definition 1. *Detector: Detector is a counter that is initialized with 0, and then updates its value by adding O_n. The detection of dumb nodes is decided, when the value of the detector hits the detection threshold. The detector of a node is denoted by D.*

The behavior of the detector D is mathematically expressed as:

$$\begin{aligned} D_{n+1} &= (D_n + O_n)^+ \\ D_0 &= 0 \end{aligned} \tag{2}$$

In the proposed work, we assume that $D \geq 0$. Thus, if the detector value is already 0, and then the value does not decrease despite the node receiving all the acknowledgments against the *HELLO* message. Thus, the value of D always fluctuates around a low value in a normal situation. Correspondingly, if a node behaves abnormally, i.e., it does not receive any acknowledgment from any of its neighbor nodes, it accumulates a large value.

In Fig. 3, three cases, which describe the different possibilities of receipt of acknowledgments, are shown receiving. In Case 1, the communication range of node A is denoted by r_c^1, and all the neighbor nodes of node A are in this communication range. So, in an ideal network condition, node A receives all the acknowledgments corresponding to the broadcasted *HELLO* messages. In Case 2, the communication range reduces and becomes r_c^2 (where, $r_c^1 > r_c^2$). In such a case, only one neighbor node (node B) is in the communication range of node A. So, in an ideal condition, node A receives the acknowledgment from node B, against the broadcasted *HELLO* message. Finally, in Case 3, the communication range of node A reduces to r_c^3 (where, $r_c^2 > r_c^3$), and all the neighbor nodes are outside the communication range r_c^3. Consequently, node A is unable to receive any acknowledgment form any of its neighbors.

Definition 2. *Detection Threshold: When a node broadcasts a HELLO message, the observation indicator is set to $+1$ or -1. Accordingly, the value of the detector D_n changes. Continuing this process of broadcasting the HELLO message, the value of D_n reaches a value which determines whether the node*

For Personal Use Only

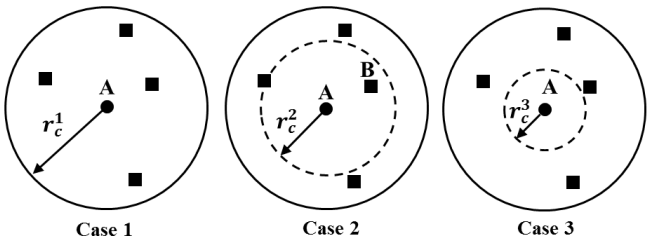


Fig. 3: Possibilities of receipt of acknowledgments

is dumb. This value is known as the Detection threshold, which is denoted by k.

Clearly, Equation (2) is a non-parametric CUSUM detector. Let k be the detection threshold. The detector decides at step n whether the node is dumb. We have,

$$\gamma = \begin{cases} 1, & \text{if,} D_n \geq k, \\ 0, & \text{if,} D_n < k, \end{cases} \tag{3}$$

Where, γ is the indicator function of whether the detection event occurs or not. The detection procedure starts over again when D reaches the value k, and thereafter, D is reset to 0.

Aperiodic *HELLO* message: For updating the value of D_n, a *HELLO* message is needed to be broadcasted at certain intervals of time. In our system, this interval is dynamic and is dependent on whether or not a node receives an acknowledgment. In an energy-constrained WSN, the unnecessary periodic broadcasting of *HELLO* messages is unacceptable. This signifies the importance of an aperiodic *HELLO* message. The time when the *HELLO* message is required to be broadcasted depends on the frequency at which a node becomes dumb and this is measured in the system with the parameter β_n. β_n increases its value by unity, only when a node does not receive an acknowledgment from any of its neighbor nodes, otherwise remains unchanged.

$$\beta_n = \begin{cases} \beta_{n-1} + 1 & \text{if, } \textit{ACK} \text{ not received from any} \\ & \text{of its neighbor nodes,} \\ \beta_{n-1} & \text{if, at least one ACK is received,} \end{cases} \tag{4}$$

Total number of *HELLO* messages broadcasted is counted by M_n. The time for sending the *HELLO* message is dependent on the time when the previous *HELLO* message was sent, and against the current

For Personal Use Only

HELLO message, if the node has received an *ACK* or not. Mathematically,

$$t_{n+1} = t_n + \left(\frac{M_n}{\beta_n}\right)\alpha \tag{5}$$

where α is a constant and the value of α = 1. Analytically, a plot is shown describing how the value of the next time interval changes with M_n and β_n.

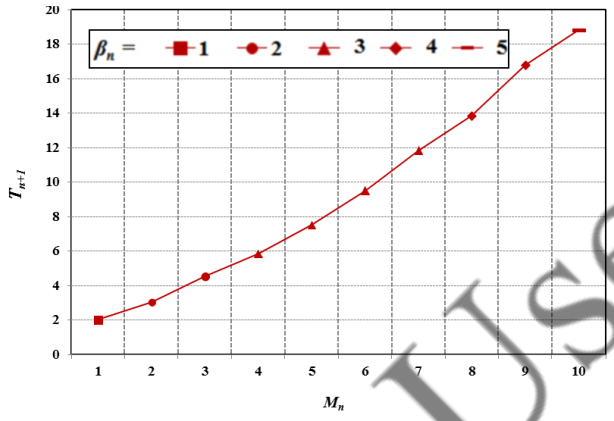


Fig. 4: Time instants of *HELLO* message broadcasting

D. Markov Chain-Based Model of the Detector

Let us consider the sequence $\{O_t, t = 0, 1, 2, \cdots\}$ as a discrete random process taken from a finite set $X = \{0, 1, 2, \cdots, k\}$. Let t and $(t+1)$ be two consecutive time instant when the broadcasting of *HELLO* messages takes place. The process is said to be in state i at time instant t, if $D_t = i$, and in state j at time instant $(t+1)$, if $D_{t+1} = j$, where $i, j \in X$. According to equation (2), the conditional distribution of future state D_{t+1} depends only on the current state D_t, given that $D_0, D_1, \cdots, D_{t-1}$ represent the past states. Thus, the random process $\{D_t\}$ satisfies the Markov property and can be modeled as a discrete-time Markov Chain.

The state transition probability from state i to j, of the Markov Chain, is:

$$P_{ij} = P\{D_{t+1} = j | D_t = i, D_{t-1} = i_{t-1}, \cdots, D_1 = i_1, D_0 = i_0\}$$

The Markov-Chain is then described as a $(k+1) \times (k+1)$ transition probability matrix, where k is the threshold of the detector D.

For Personal Use Only

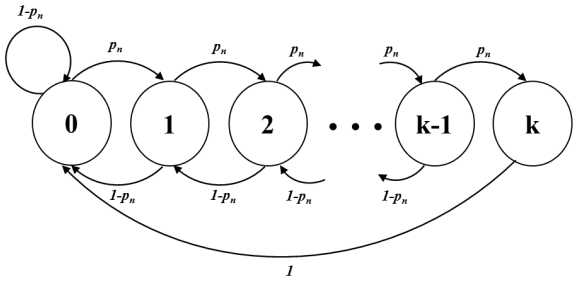


Fig. 5: State transition of the detector

$$\begin{pmatrix} P_{00} & P_{01} & P_{02} & \cdots & P_{0k} \\ P_{10} & P_{11} & P_{12} & \cdots & P_{1k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{k0} & P_{k1} & P_{k2} & \cdots & P_{kk} \end{pmatrix}$$

We divide all the transition probabilities into three distinct groups, based on the functionality of the proposed CUSUM-based dumb node detection scheme.

- Group 1: According to Equation (2), the transition probability P_{ij} is defined, where, $i = 0$ and $j = 1$. The state transition happens only when a node does not receive any acknowledgment from any of its neighbor node. Another possibility is that the state remains the same, i.e., it remains in state 0, and which possible only when it receives at least one acknowledgment from any of its neighbor nodes. Thus, these state transition probabilities are expressed as:

$$P_{0j} = \begin{cases} (1 - p_n), & \text{if } \ j = 0, \\ p_n, & \text{if } \ j = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

- Group 2: This group consists of the state transition probability P_{ij}, where $i \in (1, k-1)$ and $j = (i-1)$ or $(i+1)$. In this case, the state transition happens when the node receives or does not receive any acknowledgment from any of its neighbor nodes. However, in this case, a state cannot return to the

For Personal Use Only

same state. Thus, these state transition probabilities are expressed as:

$$P_{ij} = \begin{cases} (1-p_n), & \text{if } j = (i-1) \text{ and } i > 0 \\ p_n, & \text{if } j = (i+1) \text{ and } i > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

- Group 3: This group consists of state transition probability P_{k0}, where $i = k$ and $j = 0$. In this case, the state transition happens because D reaches the maximum threshold k. Thus, these state transitions probabilities are expressed as:

$$P_{k0} = \begin{cases} 1, & \text{if } i = k, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Algorithm 1 shows the proposed detection scheme of dumb nodes.

Algorithm 1 D3:Distributed approach for detection of dumb nodes

Inputs:
$n_i \leftarrow i^{th}$ active node, $[i = 1, 2, 3, \cdots, N_A]$, where N_A is the total number of active nodes
$\mathcal{N}(n_i) \leftarrow$ neighbor list of the i^{th} active node $[j = 1, 2, 3, \cdots, |\mathcal{N}(n_i)|]$
$\tau_O \leftarrow$ timeout
$D_n \leftarrow 0$ // `detector`
$M_n \leftarrow 1$ // `counter for` $HELLO$ `message`
$\beta_n \leftarrow 1$ // `counter if any` ACK `is not received from any of the` $\mathcal{N}_j(n_i)$
$\alpha \leftarrow 1$ // `constant`
$k \leftarrow x$ // `detection threshold`
$\mathcal{T}_i \leftarrow$ Lifetime of node n_i
Output:
Predict if node n_i is dumb
Begin
while true **do**
  Node n_i starts to broadcast *HELLO* message at time t_n
  $M_n = M_n + 1$
  if ACK received by n_i from any $\mathcal{N}(n_i)$ before τ_O **then**
    3. $T_n = T_n + \left(\frac{M_n}{\beta_n}\right)\alpha$
    if $D_n \neq 0$ **then**
      $D_n = D_n - 1$
    end if
  else
    $\beta_n = \beta_n + 1$
    $T_n = T_n + \left(\frac{M_n}{\beta_n}\right)\alpha$
    $D_n = D_n + 1$
    if $D = k$ **then**
      node n_i is dumb
      break
    end if
  end if
  if $t_n > \mathcal{T}_i$ **then**
    break
  end if
end while
End

For Personal Use Only

IV. Theoretical Analysis

In this section, the theoretical analysis of the detector is described. In a realistic situation, the transition probability between states S and $S+1$ depends on many factors such as the strength of impact of adverse environmental effects, distance between a node and its nearest neighbor node, and the residual energy of the node. The state transition probability from a state to the next state is computed by considering these factors.

A. Average False Positive Rate

We define the average false positive rate, FPR, as the rate that the detector value D_n hits the state k, even if the node is not dumb. We have formulated this rate with the help of the steady-state probability distribution of the Markov chain model. According to the theory on the discrete-time Markov chain, the rate FPR is equal to the steady state probability that the Markov chain describing the detector stays in state k, when the node is not dumb.

According to Equation (1), the probability of not receiving acknowledgment against the *HELLO* message is p_n, whereas $1-p_n$ is the probability of receiving the acknowledgment (at least from one of the neighbor nodes. Further, the transition probability matrix P follows Equations (6)-(8).

The steady state probability of the Markov chain is denoted as $(\varsigma_0, \cdots, \varsigma_k)$, and thus, it can be solved as:

$$\varsigma_j = \sum_{i=0}^{k} \varsigma_i P_{ij}, \qquad j \in \{0, \cdots, k\}, \tag{9}$$

$$\sum_{j=0}^{k} \varsigma_j = 1. \tag{10}$$

We have, the average false positive rate is:

$$FPR = \varsigma_k \tag{11}$$

In Fig. 6, the results of the variation of FPR with k are shown analytically. how the FPR changes with k. For simplicity, we consider the communication range and the distance d_{min} to its nearest neighbor for calculating the state transition probability. As an example, the initial communication range of a sensor

For Personal Use Only

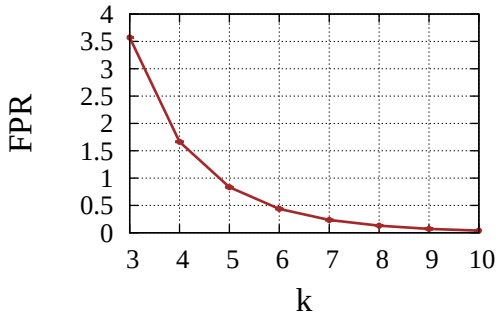


Fig. 6: False positive rate (*FPR*) vs detection threshold (*k*)

node r_c is taken as 2m, and d_{min} as 5m. The state transition probability is calculated as:

$$P_n = \left(1 - \frac{r_c}{d_{min}}\right) \tag{12}$$

Equation 12 indicates that the probability of not receiving the acknowledgment depends on the current communication range of the sensor node. Further, the result is plotted in Fig. 6. We observe that larger value of k yields smaller false positive rate, as expected.

B. Complexity analysis of the dumb node detection process

Lemma 1. *The best and the average case time complexity of the dumb node detection algorithm is* $O(kf(dist(n, h_{max})))$

Proof. ***Best Case***: Consider a node n having h^n number of neighbors, where $h^n = \{h_1, h_2, h_3, \cdots, h_m\}$. The distance from node n to a neighbor node h_i is denoted by $dist(n, h_i)$. Let h_{max} be a node, which is the farthest neighbor node of n. Thus,

$$h_{max} = max(dist(n, h_i)), \quad i = 1, 2, 3, \cdots, m \tag{13}$$

The time required to receive an acknowledgment from a neighbor h_i to n depends on the distance between them. The time to receive an acknowledgment from h_{max} is expressed as $f(dist(n, h_{max}))$. The total time required to hit k is $kf(dist(n, h_{max}))$.

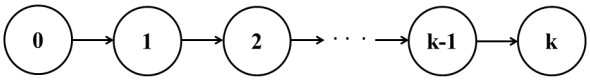


Fig. 7: Condition for best case time complexity

For Personal Use Only

In the best case, that a node broadcasts *HELLO* messages all the time, if it does not receive an acknowledgment and the detector hits the detection threshold. The condition for the best case is shown in Fig. 7. Therefore, the total time complexity expressed as $O(kf(dist(n, h_{max}))) \simeq Of(dist(n, h_{max}))$.

Average Case: The average case occurs when the detector of a node returns to its previous state and then it proceeds towards the next state. Specifically, the detector at state S goes to the next state $S+1$, because of not receiving acknowledgment from any of its neighbor nodes. Again, when the detector is at state $S+1$, it receives at least one acknowledgment from any of the neighbor nodes after broadcasting the *HELLO* messages. Thus, the detector returns to state S. Further, for two consecutive broadcast *HELLO* messages if it does not receive an acknowledgment from any of its neighbor nodes, it shifts to state $S+2$. This state transition continues for each state and it hits the detection threshold. The pictorial illustration for average case is shown in Fig. 8.

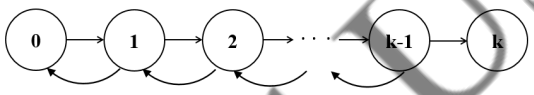


Fig. 8: Condition for average case time complexity

The time required to hit k is the same as the best case, i.e., $kf(dist(n, h_{max}))$. At each iteration, the state of the detector returns to its previous state, the time complexity being $(k-1)f(dist(n, h_{max}))$. Therefore, the total time complexity for the average case is $O(((k-1)+k)f(dist(n, h_{max}))) \simeq O(f(dist(n, h_{max})))$

□

C. Iterations required for detection

Theorem 1. *A node can detected as a dumb node if and only if, the node does get any acknowledgment from any of its neighbor nodes for equal times or higher than the numeric value of 'k'.*

Proof. Each node broadcasts a *HELLO* message periodically. Let a node broadcast a *HELLO* message p times, and let it receive an acknowledgment for $(p-q)$ times. The dumb detector, D_n, reaches k, which is dependent on how many times a node receives acknowledgment against *HELLO* message. Thus, the node does not receive the acknowledgment against the *HELLO* message for q times. For successfully dumb node detection, the value of D_n must be equal to k. Therefore, we have,

$$q - (p - q) = k \tag{14}$$

For Personal Use Only

Thus,

$$p = (2q - k) \tag{15}$$

For successful detection of dumb node, the number of times the *HELLO* message are to be broadcasted must be greater than or equal to the threshold value k.

$$p \geq k \tag{16}$$

By combining equations (15) and (16), we have,

$$q \geq k \tag{17}$$

Hence, from Equation (17), the statement of the theorem is proved. □

Lemma 2. *With the increase in numeric value k, the time elapsed for a node to be detected as dumb node increases.*

Let us consider k_1 and k_2 to be two detection thresholds, where, $k_2 > k_1$. Let the time required to change the detector from one state to the next be t.

When the detection threshold is k_1, the total time (T_1) taken to detect the dumb node is:

$$T_1 = (k_1 - 1)t \tag{18}$$

Again, T_2 is the total time taken to detect the dumb node when the detection threshold is k_2. Mathematically:

$$T_2 = (k_2 - 1)t \tag{19}$$

As $k_2 > k_1$, from Equations (18) and (19), we conclude that $T_2 > T_1$.

V. Performance Evaluation

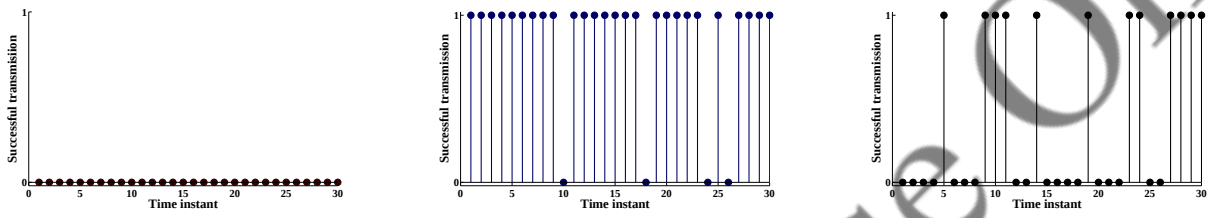
A. Simulation Design

In this Section, we evaluate the performance of the proposed scheme for the detection of dumb nodes in WSNs. To simulate our scheme, we consider that the sensor nodes are deployed randomly over in the terrain. This work is one of the first attempts of its kind that detects dumb nodes. Thus, comparative

For Personal Use Only

TABLE I: Simulation Parameters

Parameter	Value
Number of nodes (N)	100-350
Simulation area	500 m $\times$ 500 m
Sensing range	25 m
Communication range	20-60 m
Data rate	250 kbps
Constant value (ξ)	0.0005
Power consumption of transmitting circuitry (P_{T0})	$15.9mW$
Power consumption of receiving circuitry (P_{R0})	$22.2mW$
Drain efficiency (η)	15.7 %
Path loss exponent (α)	2.5



(a) Dead node (b) Selfish node (c) Dumb node

Fig. 9: Difference in packet delivery in dead, selfish, and dumb nodes

analysis with prior work is out of scope. However, for the sake of completeness, we present few results as the justification of the same. Fig. 9 depicts how the scenario of a dumb node is different from selfish and dead nodes. In figure, different time instants are shown along X-axis. Along Y-axis two points, 0 and 1, are shown. 1 signifies that a data packet from a node can reach the destination, and 0 signifies otherwise. Fig. 9(a) depicts the scenario of a dead node. In this figure, we observe that at each time instant, the node sends the data, but is unable to reach to the destination. Therefore, the dead nodes cannot communicate with any other node and it is permanent. Fig. 9(b) shows the scenario of a selfish node. We observe only few data packets at different time instants $(10, 18, 24, 26)$ that were unable to be transmitted to the destination. These unsuccessful data transmissions are quite normal in a communication system. However, in case of a selfish node, we reach it very easily, and decides that the node is selfish. The case of a dumb node is shown in Fig. 9(c). In this figure, we observe the dynamic situations in data packet delivery at different time instants. The packet delivery to the destination in case of dumb nodes depends on the intensity of the adverse environmental effects. Therefore, dumb behavior is completely distinct from the existing behavior. Consequently, the detection scheme of a dumb node is not comparable to any other misbehavior detection scheme.

We provide the results in terms of the following performance evaluation parameters:

For Personal Use Only

- *Percentage of dumb nodes*: The number of occurrences of dumb nodes per 100 nodes in the network, due to the shrinkage in communication range in the presence of adverse environmental effects.
- *Percentage of detection*: The number of dumb nodes detected per 100 dumb nodes. Mathematically, Percentage of detection = $\frac{N_d}{N_d^{tot}} \times 100$ where,
  N_d: Total number of dumb nodes detected
  N_d^{tot}: Total number of dumb nodes present in the network
- *Message overhead*: Number of bytes required to detect (all possible detection) the dumb nodes present in the network. The message overhead includes *HELLO* and *ACK* messages.
- *Energy consumption*: The amount of energy required to detect (all possible detection) dumb nodes in the network.

  Energy consumption model: The proposed algorithm uses the same energy consumption model as was used in [24], [34]. In this model, the energy consumption (E_T) required for transmitting a packet of N bits from one sensor node to another at a constant data rate R is given by:

$$E_T = \frac{P_T \times N}{R} \tag{20}$$

  where,

$$P_T = P_{T0} + \frac{\xi \times d^{\alpha}}{\eta} \tag{21}$$

- *Simulation time*: This parameter indicates the simulation time required to run the proposed algorithm.

B. Results

Fig. 10 indicates that with the increasing communication range, the percentage of dumb nodes decreases in the network. Again, if the number of nodes in the network is more in number, the possibility of getting neighbor nodes is more. Thus, with the increase in the total number of nodes in the network, the percentage of dumb nodes decreases.

Fig. 11 depicts how the percentage in detection of dumb node changes with the detection threshold, by considering total number of nodes in the network, $150, 250,$ and 350. The detection threshold is plotted along the X-axis from 10 to 20, with the step of 1. In this figure we observe that with the increasing value of detection threshold k, the percentage of detection decreases. The reason behind this degradation of the percentage of detection is that the possibility of hitting a lower value of detection threshold is more

For Personal Use Only

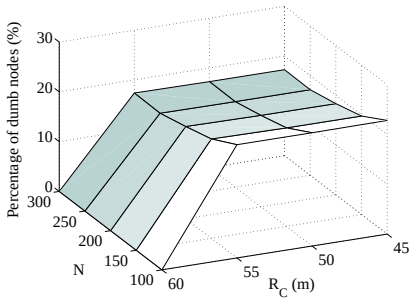


Fig. 10: Percentage of dumb nodes with Communication range

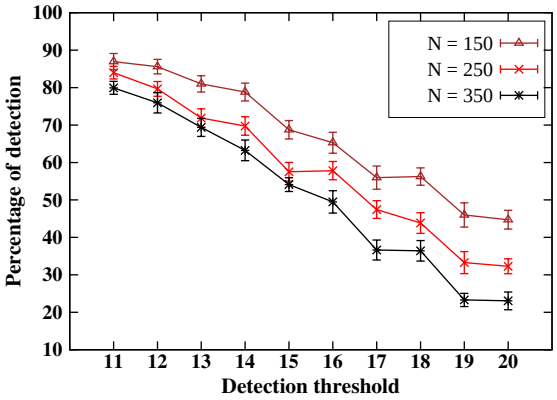


Fig. 11: Percentage of detection with k

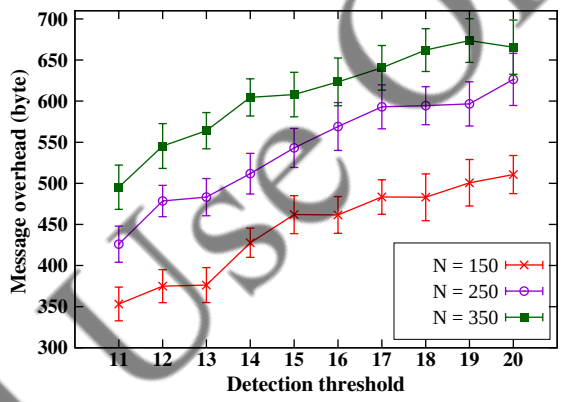


Fig. 12: Message overhead with k

than that of a higher value of detection threshold within the life-time of a node. The plot also depicts that the increasing number of nodes in the network, the detection percentage decreases. The probable reason for this pattern is that the possibility of getting a neighbor node increases with increase in total number of nodes in the network. Consequently, the probability of a node behaving as dumb also decreases. The detection percentage degrades by 56% with increase in the detection threshold value.

The variation in the message overheads with the detection threshold for varying number of nodes is shown in Fig. 12. In this figure, the message overhead increases gradually with the increase in the detection threshold. Also, we observe that the message overhead increases with higher value of detection threshold. The reason behind this nature of the plot is that for a higher value of detection threshold a node need to broadcast the *HELLO* message for more number of times than a lower detection threshold value. The plot also depicts that the overhead increases with the increase in total number of nodes in the network. With the increase in total number of nodes in the network, the number of broadcasted *HELLO* messages also increases, which, in turn, increases the overhead in the network. With the increase of detection

For Personal Use Only

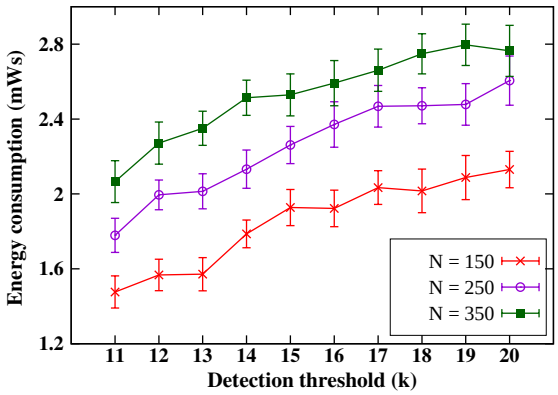


Fig. 13: Energy consumption with k

threshold, the message overhead in the network increases by 40%. Fig. 13 depicts the variation in the energy consumption for detecting the dumb nodes in the network with detection threshold for varying number of nodes using the proposed scheme D3. The detection threshold is shown along the X-axis. The total number of nodes considers $150, 250$, and 350. In each of the cases the energy consumption increases with increase in detection threshold. The energy consumption in the network depends on the percentage of dumb node detected using our scheme. The plots also shows that the increase in total number of nodes in the network increases energy consumption of the network. Increasing detection threshold and total number of nodes in the network increases the overhead in the network as shown in Fig. 12. Consequently, the energy consumption of the network also increases due to increase of the number of transmitted and received packets. However, the energy consumption increases by 40% with the increase of detection threshold.

In each of the plots, it is observed that different network parameters dependent on the value of the detection threshold k. Therefore, the value of detection need to set as per the user requirement. For example, detection of dumb node is very much crucial for a certain situation then the value of k should be chosen smaller.

We examine the performance of the simulation for the proposed scheme D3. We simulate an environment consisting of $100 - 400$ nodes with an interval of 50 nodes in the network. Fig. 14 depicts the variation in simulation time with the total number of node (N) in the network. In this plot, we consider three detection thresholds ($k = 10, 15, 20$). We observe that in each of the cases the simulation time increases with total number of nodes in the network. The simulation time also increases with increasing

For Personal Use Only

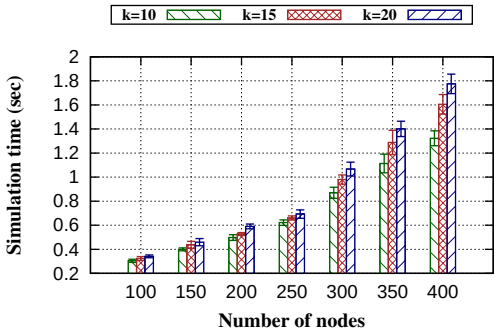


Fig. 14: Simulation time with number of nodes

value of detection threshold, k. However, the average, maximum, and minimum simulation times are $0.2701, 0.282717093, 0.257482907$ seconds, respectively, irrespective of k.

VI. Conclusion

In this work, we have considered a newly identified misbehavior of sensor nodes — dumb behavior [23]. Due to the shrinkage in the communication range, a dumb node is unable to transmit its sensed data to any other node. Consequently, the re-establishment of connectivity between dumb and the other nodes is essential. Prior to the re-establishment of connectivity, a node has to detect itself as dumb. The detection of dumb behavior of a node is challenging because of the dynamic nature of "dumbness". In this work, we have proposed a scheme for the detection of dumb nodes in a WSN using the cumulative sum (CUSUM) approach.

In the future, we plan to extend this work by enabling the detection of dumb nodes by the sink. Further, we want to propose an optimized connectivity re-establishment scheme for dumb nodes. In order to re establish the connectivity between dumb and other nodes, we plan to propose a node placement algorithm.

References

[1] I. B. Abid and N. Boudriga, "Game Theory for Misbehaving Detection in Wireless Sensor Networks," in *Proceedings of International Conference on Information Networking*, January 2013, pp. 60–65.

[2] M. Ahmed, X. Huang, D. Sharma, and L. Shutao, "Wireless Sensor Network Internal Attacker Identification with Multiple Evidence by Dempster-shafer Theory," in *Proceedings of the* 12^{th} *International Conference on Algorithms and Architectures for Parallel Processing - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 255–263.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[4] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, November 2002.

For Personal Use Only

[5] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance Measurements of Mote Sensor Networks," in *Proceedings of the* 7^{th} *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, New York, USA, 2004, pp. 174–181.

[6] K. Bannister, G. Giorgetti, and S. Gupta, "Wireless Sensor Networking for Hot Applications: Effects of Temperature On Signal Strength, Data Collection and Localization," in *Proceedings of the* 5^{th} *Workshop on Embedded Networked Sensors*, Virginia, USA, June 2008.

[7] F. Bao, I. R. Chen, M. Chang, and J. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–183, June 2012.

[8] D. Bissell, *Statistical Methods for SPC and TQM*. Chapman & Hall, 1994.

[9] C. A. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig, "The Impact of Temperature on Outdoor Industrial Sensornet Applications," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 451–459, August 2010.

[10] R.-C. Chen, C.-F. Hsieh, and Y.-F. Huang, "A new method for intrusion detection on hierarchical wireless sensor networks," in *Proceedings of* 3^{rd} *International Conference on Ubiquitous Information Management and Communication*, 2009, pp. 38–45.

[11] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "Distributed Detection of Clone Attacks in Wireless Sensor Networks," *IEEE Transactions on Dependable and Secure Comp.*, vol. 8, no. 5, pp. 685–698, October 2011.

[12] L. Guiyun, X. Bugong, and C. Hongbin, "An indicator kriging method for distributed estimation in wireless sensor networks," *International Journal of Communication Systems (Wiley)*, vol. 27, no. 1, pp. 68–80, 2014.

[13] X. Ju, H. Zhang, and D. Sakamuri, "NetEye: a user-centered wireless sensor network testbed for high-fidelity, robust experimentation," *International Journal of Communication Systems (Wiley)*, vol. 25, no. 9, pp. 1213–1229, 2012.

[14] A. R. M. Kamal, C. J. Bleakley, and S. Dobson, "Failure Detection in Wireless Sensor Networks: A Sequence-Based Dynamic Approach," *ACM Transactions on Sensor Networks*, vol. 10, no. 2, Article 35, pp. 1–29, January 2014.

[15] R. Kannan, S. Wei, V. Chakravarthi, and G. Seetharaman, "Analysis of Communication Vulnerability through Misbehavior in Wireless and Sensor Networks," in *Proceedings of Military Communications Conference*, October 2005, pp. 1040–1046.

[16] M. Khatua and S. Misra, "CURD: Controllable reactive jamming detection in underwater sensor networks," *Pervasive and Mobile Computing*, vol. 13, pp. 203–220, 2014.

[17] B. Krishnamachari and S. Iyengar, "Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, March 2004.

[18] P. Kulakowski, E. Calle, and J. L. Marzo, "Performance study of wireless sensor and actuator networks in forest fire scenarios," *International Journal of Communication Systems (Wiley)*, vol. 26, no. 4, pp. 515–529, 2013.

[19] K. Lee, H. M. Kwon, Y. Ding, Y. Ibdah, and Z. Wang, "Noncooperative Distributed MMSE Relay Schemes under Jamming Environment and Node Geometry in Wireless Relay Networks," in *Proceedings of Wireless Telecommunications Symposium*, April 2011, pp. 1–5.

[20] F. Liu, X. Cheng, and D. Chen, "Insider Attacker Detection in Wireless Sensor Networks," in *INFOCOM*, May 2007, pp. 1937–1945.

[21] L. Liu, N. Antonopoulos, J. Xu, D. Webster, and K. Wu, "Distributed service integration for disaster monitoring sensor systems," *Communications, IET*, vol. 5, no. 12, pp. 1777–1784, August 2011.

[22] X. Luo, M. Dong, and Y. Huang, "On Distributed Fault-Tolerant Detection in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 58–70, January 2006.

[23] S. Misra, P. Kar, A. Roy, and M. S. Obaidat, "Existence of Dumb Nodes in Stationary Wireless Sensor Networks," *Journal of Systems and Software*, vol. 91, pp. 135–146, 2014.

For Personal Use Only

[24] S. Misra, G. Mali, and A. Mondal, "Distributed topology management for wireless multimedia sensor networks: exploiting connectivity and cooperation," *International Journal of Communication Systems (Wiley)*, vol. 27, no. 3, March 2014.

[25] S. Misra and S. Singh, "Localized Policy-Based Target Tracking Using Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 8, no. 3, Article 27, pp. 1–30, July 2012.

[26] S. Misra and S. Chatterjee, "Social choice considerations in cloud-assisted WBAN architecture for post-disaster healthcare: Data aggregation and channelization," *Information Sciences*, vol. 284, pp. 95–117, 2014.

[27] F. Nadeem, E. Leitgeb, M. S. Awan, and S. Chessa, "Comparing the Life Time of Terrestrial Wireless Sensor Networks by Employing Hybrid FSO/RF and Only RF Access Networks," in *Proceedings of* 5^{th} *International Conference on Wireless and Mobile Communications*, 2009, pp. 134–139.

[28] S. Rajasegarar, J. C. Bezdek, C. Leckie, and M. Palaniswami, "Elliptical Anomalies in Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 6, no. 1, Article 7, pp. 1–28, December 2009, article 7.

[29] A. Roy, P. Kar, and S. Misra, "Detection of Dumb Nodes in a Stationary Wireless Sensor Network," in *Proceedings of* 11^{th} *IEEE India Conference*, 2014, (Accepted).

[30] A. Roy, A. Mondal, and S. Misra, "Connectivity Re-establishment in the Presence of Dumb Nodes in Sensor-Cloud Infrastructure: A Game Theoretic Approach," in *Proceedings of* 6^{th} *International Conference on Cloud Computing Technology and Science*, 2014, (Accepted).

[31] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor Faults: Detection Methods and Prevalence in Real-World Datasets," *ACM Transactions on Sensor Networks*, vol. 6, no. 3, pp. 1–38, June 2010.

[32] E. Soltanmohammadi, M. Orooji, and M. Naraghi-Pour, "Decentralized Hypothesis Testing in Wireless Sensor Networks in the Presence of Misbehaving Nodes," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 205–215, January 2013.

[33] J. Tang, Y. Cheng, and W. Zhuang, "Real-Time Misbehavior Detection in IEEE 802.11-Based Wireless Networks: An Analytical Approach," *IEEE Transactions on Mobile Computing*, vol. 13, no. 1, pp. 146–158, Jan 2014.

[34] Q. Wang, H. Mark, and Y. Woodward, "A realistic power consumption model for wireless sensor network devices," in *Proceedings of Sensor and Ad Hoc Communications and Networks*, vol. 1. $3rd$ Annual IEEE Communications Society, 2006, pp. 286–295.

[35] Y. Wang, P. Shi, K. Li, and Z. Chen, "An energy efficient medium access control protocol for target tracking based on dynamic convey tree collaboration in wireless sensor networks," *International Journal of Communication Systems (Wiley)*, vol. 25, no. 9, pp. 1139–1159, 2012.

For Personal Use Only