ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Distributed Region-Based Monitoring in Low-Power Listening Wireless Sensor Networks

Krita Pattamasiriwat[1] and Chaiporn Jaikaeo[2]

ABSTRACT

Advancement in IoT technology and the concept of Information-Centric Networking lead to less importance of node individuality since several nodes can work interchangeably. Multiple sensor nodes can be grouped into a *region* and monitored as one instance to guarantee sufficient coverage over the region. Therefore, a single node fault often does not need to be reported unless it is the last node in the region. In addition, applications focusing on detecting rare events rarely require nodes to transmit and often rely on a low-power listening MAC protocols, where nodes spend most of their time sleeping but require significantly more work during transmission. In such situations it is desirable to avoid periodic status reports transmitted to the central monitor station as usually found in a centralized monitoring scheme. A distributed region-based monitoring scheme, or *DRMON*, is then proposed to facilitate this circumstance. This approach designates a representative to each region so that it can be used as an indicator of the region's status with a mechanism to re-elect a new representative until all nodes in the respective region are dead, implying region inactiveness. We evaluate the suitability of DRMON over various scenarios in two aspects: centralized vs. distributed monitoring schemes and individual-based vs. region-based monitoring schemes, along with existing work in the literature. Simulation results indicate that region-based schemes outperform the individual schemes in terms of power consumption and scalability when the number of regions is low. The distributed schemes also yield better efficiency in terms of message overhead. Compared against the other schemes, DRMON's overall power consumption is reduced by 4%-10%, with 66%-88% reduction in packet transmissions, while maintaining fault detection precision and recall of greater than 90% and the detection delay within an acceptable range. This outcome suggests that in the case where existence of individual node is out of concern, distributed region-based fault monitoring scheme could be employed to reduce energy usage and lower message overhead while retaining acceptable detection accuracy.

Article information:

Keywords: Wireless Sensor Networks, Network Monitoring, Distributed Monitoring, Region-based Monitoring

Article history:
Received: December 2, 2020
Revised: May 10, 2021
Accepted: July 4, 2021
Published: February 26, 2022 (Online)

DOI: 10.37936/ecti-cit.2022161.243469

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of small-size, low-cost, low-power, multi-functional sensor nodes that collaborate with each other [1] to serve a variety of purposes. WSNs have recently been incorporated into the Internet of Things (IoT) to facilitate data acquisition and wireless communication. To guarantee the functionality of networks, nodes and communication links must be maintained in good condition. Therefore, network monitoring becomes a

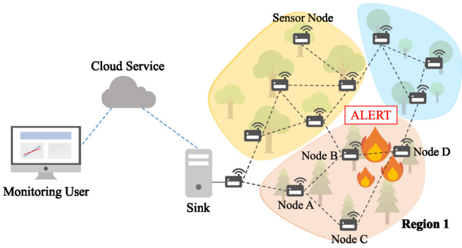


Fig.1: *Forest fire detection scenario.*

[1,2]The authors are with Department of Computer Engineering, Faculty of Engineering, Kasetsart University, Bangkok, Thailand 10900, Phone (+66) 2797-0999 Ext.1424, E-mail: krita.p@ku.th and chaiporn.j@ku.ac.th
[2]Corresponding author: chaiporn.j@ku.ac.th

crucial part of the system, especially for safety-critical applications. Traditional monitoring methods such as human inspection or wired network monitoring protocols are not preferable or even inapplicable. Therefore, several monitoring and fault diagnosis mechanisms specifically designed for WSNs have been proposed to address this problem. Network monitoring techniques can be classified into *centralized* and *distributed* approaches, depending on the location where the status of the nodes and links is determined.

Advances in IoT technology is facilitating the large-scale deployment of WSNs. The concept of Information-Centric Networking (ICN) [2] has been widely adopted in this area. In this scenario, node individuality is irrelevant as several nodes are eligible to work interchangeably. Identical observations may be collected and reported by more than one node. As a result, the network is still able to operate correctly even if some nodes malfunction, thereby lowering repair cost. For example, in forest fire detection application, the monitored area is populated with sensor nodes that are randomly deployed using an aircraft. The coverage areas of nearby nodes likely overlap, as shown in Fig. 1. The network then requires only one or a few nodes in the same region to detect an event. To minimize the cost of maintenance, the network administrator should be notified when all of the sensor nodes in a region are damaged or separated from the network.

Motivated by the concept of ICN, nodes could be grouped and monitored as one instance to guarantee the availability of specific content. To the best of our knowledge, existing monitoring approaches primarily focus on monitoring the individual node or link status. Although the awareness of the status of all nodes can be further processed to estimate the risk of network failure, redundant monitoring information is unnecessarily produced, resulting in the wasting of the nodes' energy and communication bandwidth. When relying on low-power listening mechanisms, unnecessary transmissions are even more costly with modern WSN transceivers that allow high power during transmission compared to reception. Our proposed approach, the distributed region-based monitoring scheme (DRMON), monitors the network status in terms of *regions*, each of which is a predefined group of adjacent nodes that can perform sensing tasks interchangeably. Faulty nodes in a region are acceptable as long as the region is still able to continue its operation by having at least one functional node representing the region. A region is considered faulty if all its nodes have failed or if there is no available path to the sink. This scheme provides a framework for monitoring status of WSN used for rare event detection or non-continuous data collection, where periodic sensor reports are not always assured. In addition, many of these rare-event detection scenarios often involve detection of life-threatening events, such as landslides. In these situations, a certain degree of redundancy would be preferable, especially when multiple sensors could reduce the chance of missing an event [3]. The main objective of this method is to increase energy efficiency by reducing the traffic overhead generated in the monitoring process while maintaining effectiveness in both fault detection accuracy and latency aspects.

To evaluate the efficacy of the DRMON scheme, four types of generic monitoring approaches, i.e., centralized individual scheme (CIMON), distributed individual scheme (DIMON), centralized region-based scheme (CRMON), and distributed region-based scheme (DRMON), including an existing distributed diagnosis protocol for WSN called WSNDiag [4], were implemented. The evaluation was conducted using the Cooja simulator in the Contiki operating system [5]. The simulation results show that region-based schemes outperform the individual monitoring schemes in terms of power consumption and message overhead in cases where the network has a low number of regions. However, maintaining region representatives can cause overhead, so individual monitoring approaches may be more suitable if the number of regions is high and/or the regions are of small sizes.

Here we review previous work related on fault monitoring of wireless sensor networks.

1.1 Network Fault Monitoring

Faults in WSNs can be classified into *data faults* and *function faults* [6]. A data fault is a sensor data report that is inconsistent with the actual behavior of the phenomenon of interest [7]. Several fault detection schemes have been proposed to address this problem [8] [9]. Nevertheless, this type of faults is beyond the scope of this investigation. Function faults are defined as network operation abnormalities, such as node crashes, node energy depletion, link failures, and traffic congestion.

Faults can be further categorized into hard faults and soft faults [4] [10]. *Hard faults*, also called *permanent* faults, often result from failures of hardware modules such as broken communication modules or battery depletion. A node experiencing a hard fault is not capable of communicating with the rest of the network. *Soft faults*, on the other hand, are temporary or intermittent faults. Nodes with soft faults may continue to operate, but potentially with undesirable behaviors. Therefore, soft faulted nodes are often difficult to detect. This work focuses on hard (i.e., permanent) faults only.

Data retrieval techniques used for network monitoring can be generally categorized into two categories: *active* and *passive*. Active monitoring [11] [12] results in the generation of extra probing packets or event report, in addition to the application of data traffic in the network. This method provides detailed information but usually results in high operation overhead if the probing frequency and monitoring parameters are not carefully selected.

Passive monitoring [13] relies on the utilization of existing information or piggybacking of data to infer fault occurrences without generating extra packets. This technique seems to be preferable because it has a minimal effect on network performance. However, in some cases such as non-periodic event-driven data applications, existing information or data traffic might be insufficient for the effective analysis of network status.

Network monitoring techniques could also be classified based on the location where the diagnosis process is performed. The traditional method where the node's status information is separately delivered to a powerful sink node for diagnosis is described as *centralized monitoring*. This method gives the complete status of the network and high accuracy rate but suffers from packet collisions and the energy sink-hole problem as the number of nodes in the network increases [14] [11] [13].

In large-scale networks, centralized approaches may encounter several problems, such as detection delay, packet loss, and the energy sink-hole problem. To achieve better energy efficiency and scalability, *distributed monitoring* techniques have been investigated [15] [12] [16] [17]. Using this scheme, any node in the network may assume the responsibility of monitoring and could implement countermeasures or report the status to the sink. Based on this concept, information is not relayed to the sink from the sensor nodes during normal operation and a report is made only when an anomaly is detected.

In addition, several investigations have been pursued that focus on the observation of the condition of communication links [18] [19]. However, all these schemes mainly focus on the monitoring of individual nodes. With the recent trend of ICN, individual nodes might be less relevant. To the best of our knowledge, an approach specifically designed to monitor an information-centric network has not been developed as yet.

1.2 Region-based Monitoring Technique

Conceptually, a region or a cluster is simply a set of nodes located in the same neighborhood to collectively perform sensing tasks such as environmental observation. Regions can be dynamically formed using existing clustering methods such as *LEACH* [20] or *CHEF* [21]. Dividing nodes into clusters facilitates distributed computing such as routing and data aggregation. Several distributed fault diagnosis techniques previously proposed often rely on some forms of clustering mechanisms [22] [23] [24] [25] [26]. Zhang and Yuan [22] proposed a fault diagnosis algorithm that divides the whole network into several clusters so that fault management can be distributed into each cluster area. Nodes in each cluster cooperatively exchange information which is used to determine their faulty states. Titouna *et al.* [23] applies Bayesian classifiers to calculate fault probabilities at node level. Each node's probability distribution is then forwarded to its cluster head for further processing along with data from other nodes to determine faults at the cluster level. To avoid collisions and improve energy efficiency, some works employ a time-division multiple access (TDMA) mechanism by allocating time slots for each member within a cluster to report diagnosis information to its cluster head, and also among the cluster heads when reporting to the central monitoring station [26] [24]. However, TDMA requires relatively complicated clock synchronization among nodes and does not adapt well to network changes due to nodes' mobility or failures. Loss of synchronization results in interference from overlapping schedules, which can lead to high latency and power consumption [27].

Regions may also be predefined by an operator based on geographical location so that nodes in the same region can perform tasks on behalf of others. In this case, we do not need to know the status of each individual node, but for each region instead. The concept of regions may have been formerly used to describe the report generated by *eScan* [14], which is the integrated data of adjacent nodes that produce similar reported values. Nevertheless, the membership of nodes is not preserved in this method, which differs from the definition of the region in our scenario. The cluster-based fault diagnosis algorithms reviewed so far, while dividing nodes into regions for distributed fault diagnosis, still mostly aim to determine individual nodes' faulty status instead of the region as a whole. To the best of our knowledge, a monitoring scheme that monitors networks with predefined regions, especially for rare-event detection, has not been proposed in the literature.

1.3 Impact of Low-Power Listening MAC Protocols on Network Monitoring

Low-Power Listening (LPL) is a well-known technique used for scheduling the wake-up periods of nodes. Nodes that adopt LPL periodically wake up to sample the medium and remain in an active state if there are available data packets, otherwise they return to sleep mode until the next scheduled wake-up time. The power consumption rate of the network using LPL mechanism is thus drastically lower.

X-MAC [28] is an LPL-based MAC protocol that uses a series of short strobe frames with an embedded receiver ID to alleviate the overhearing problem and to increase scalability because non-target nodes return to the sleep mode as soon as they determine that there is no data available for them. The communication duration is also decreased due to the gap between preamble frames, which allows the recipient to send an early acknowledgment frame back to the

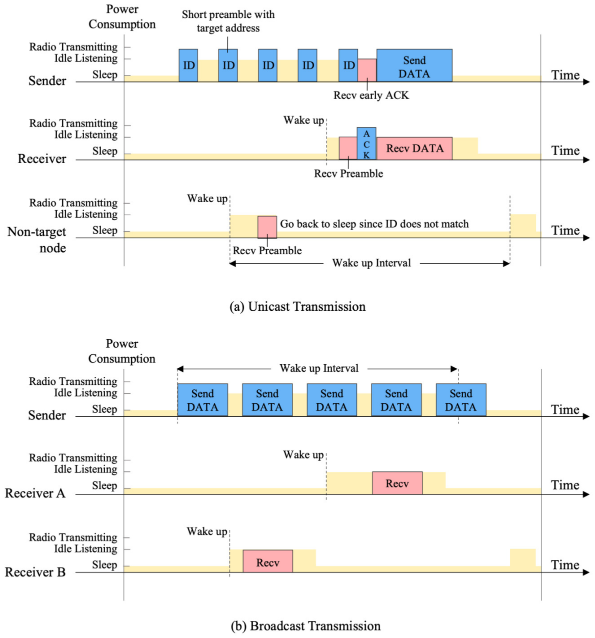


Fig.2: Duty-cycling mechanism of X-MAC protocol

sender and to initiate data transmission. This protocol is widely adopted in WSN applications because it works efficiently in terms of reducing energy usage and has no limitations in terms of radio hardware choice. The procedure involved in data transmission using the X-MAC protocol is depicted in Fig.2. Unicast transmission is performed when the sender transmits probing frames until an acknowledgment is received from the intended receiver. The maximum duration of the frame repetition is the full wake-up interval, which guarantees data reception of the receiver. In the case of broadcast transmission, the acknowledgment mechanism is not presented, thus, the sender must send the frame containing the entire data packet continuously throughout the full wake-up interval.

ContikiMAC [29], inspired by X-MAC, also relies on periodic wake-ups but employs additional optimization mechanisms to improve energy-efficiency such as a fast sleep optimization and a transmission phase-lock optimization. With these mechanisms, nodes learn the exact wake-up intervals of their neighboring nodes, allowing them to send packets to a receiver at the exact time it is awake. Therefore, wake-up duration and energy consumption can be greatly reduced.

As LPL can inflict substantial energy consumption on transmitting nodes due to the use of multiple transmissions per packet, and also on overhearing nodes by waking them up, it is suitable for scenarios in which nodes rarely transmit, such as detecting rare events. Typical network monitoring requires nodes to report their status periodically, which worsens the performance of LPL, especially for nodes located around the central monitoring station in a centralized monitoring scheme. The evaluation of LPL-based MAC protocols on the performance of network monitoring was discussed in [30]. However, only individual-based monitoring schemes were evaluated in this previous work. In this paper, region-based monitoring approaches are also investigated.

2. METHODS/EXPERIMENTS

Our proposed monitoring mechanism, distributed region-based monitoring scheme (DRMON), is a framework that could be flexibly configured to monitor individual nodes by considering each region to contain only one node. We assume that nodes are distributed in a 2-dimensional area without obstacles. All nodes have limited power resource and are not rechargeable, thus, radio duty cycle protocols could be utilized to prolong the network lifetime. There is exactly one powerful sink node located at the border of the network that can be easily accessed by the operator. Each node can communicate with the sink directly or indirectly via multi-hop communication.

A *region* is a group of adjacent nodes that can work interchangeably. It can be predefined by the operator or automatically arranged based on specific criteria. Faulty nodes in a region might result in lower performance but the region is still able to continue its operation if there is at least one functional node. A region is considered faulty if all its nodes fail or if there is no available path to the sink. Each sensor node must be aware of its region ID, which might be derived from a region information message propagated from the sink or a region map containing coordinate information.

Based on the definition of a region, only one node per region is required to be monitored at a time. These nodes are called region representatives, or *region reps* for short. To achieve energy efficiency and scalability, region reps are monitored by their one-hop neighbors. At the sink, the status of each region reported to the operator is assumed to be normal unless there is a fault notification from the monitoring nodes.

The DRMON mechanism consists of two phases, region representative election, and monitoring process. Each node participating in a region representative election generates *chance* based on related information readily available, such as hops to sink, remaining energy, and link quality, then broadcasts its chance to all nodes in the same region. The node with the highest chance is appointed as the region rep and its parent derived from the existing routing scheme will automatically become the monitoring node. In case the routing scheme has assigned multiple parents to a node, any one of the parents will be selected. Algorithms 1 and 2 comprise the region representative election process. The calculation of *chance* given in Algorithm 1 is one example that takes into account only hops to sink, where the more hops to sink, the lower the chance. The random number r is introduced for tie breaking. The $T_{backoff}$ is then calculated accordingly so that the lower the chance, the more time the node will wait before broadcasting its chance to the neighbors.

Algorithm 1 Initialization

```
procedure INITIALIZE
    regionRep ← self ID
    r ← random number between 0 and 1
    ▷ hops-to-sink is used as an example, assuming
    ▷ the network diameter does not exceed 10 hops
    chance ← 10 − HOPSTOSINK + r
    T_backoff ← 10 − chance + 1
    Wait for T_backoff seconds
    Broadcast chance to neighbors
    Wait for T_elect seconds
    if regionRep = self ID then
        parent ← routing upstream node
        call REPHEARTBEAT
    end if
end procedure
```

Algorithm 2 Processing chance message

```
procedure ONCHANCERECEIVED(msg)
    if msg has been seen then
        return
    end if
    if msg.regionId not match self region ID then
        return
    end if
    if maxChance < msg.chance then
        maxChance ← msg.chance
        regionRep ← msg.source
        Rebroadcast msg
    end if
end procedure
```

Algorithm 3 Region rep heartbeat

```
procedure REPHEARTBEAT
    repeat
        Broadcast HEARTBEAT
        Wait for T_hb seconds
    until regionRep ≠ self ID
end procedure
```

Algorithm 4 Monitoring node process

```
procedure MONITOR
    RepSet ← ∅
    while true do
        if receives HEARTBEAT then
            rep.lastHeard ← now()
            rep.regionId ← HEARTBEAT's region ID
            RepSet ← RepSet ∪ {rep}
        end if
        ▷ check for region timeout
        for rep in RepSet do
            if now() − rep.lastHeard > T_mon then
                send rep failure report to sink
            end if
        end for
    end while
end procedure
```

After region reps have been selected, the monitoring process begins. We use a simple monitoring method based on heartbeat messages. A region rep periodically broadcasts heartbeat messages embedded with its region ID every T_{hb} seconds to inform its monitoring node on the region status. The absence of heartbeat messages can be interpreted as a region failure. The monitoring node then notifies the sink of the region fault status. Algorithms 3 and 4 comprise the monitoring process. Monitoring timeout, T_{mon}, can be adjusted and needs to be carefully selected to balance the detection time and false alarm rate.

Algorithm 5 Observer node process

```
procedure OBSERVE
    lastHeard ← now()
    while true do
        if receives HEARTBEAT from region rep then
            lastHeard ← now()
        end if
        if now() − lastHeard > T_obs then
            ▷ trigger region rep election
            call INITIALIZE
        end if
    end while
end procedure
```

However, from the region fault definition, the death of a single node in a region does not always imply region failure. A mechanism to maintain the region-monitoring structure is required if it is possible. The substitute region's representative should be reelected within an appropriate time to reduce false alarm. To achieve this goal, the rest of neighbors of the region rep, except the monitoring node, assume the role of the *observers*, whose process is listed in Algorithm 5. It's not necessary for observers to be in the same region as the region rep. Each observer monitors heartbeat messages from a region rep and trig-

Table 1: All types on monitoring approaches

	Centralized	**Distributed**
Individual	CIMON	DIMON
Region-based	CRMON	DRMON

gers a region representative election process by broadcasting a reelect message in the absence of heartbeats instead of reporting to the sink. When nodes in that particular region receive a reelect message, each of them will generate a new chance message and repeat all the election procedure. The parent node of the new region rep, derived from the existing routing information, will also become the next monitoring node for that region. The timeout of observer nodes, T_{obs}, should be shorter than T_{mon} to prevent false alarms at the sink since the report of absent previous representative node will be discarded.

Finally, this monitoring framework reports two possibilities of region status, R_t, region *active* and *inactive* at time t, to the operator. The sink initially assumes that all regions are active until it receives region inactive reports from monitoring nodes and vice versa. However, region inactive reports are discarded once a new region active report is received for the same region.

To evaluate the performance of DRMON, four types of monitoring schemes that adopt different techniques for monitoring approaches and instances were implemented. Monitoring approaches are classified into centralized and distributed and monitoring instances are grouped as individual nodes and regions. All the implemented schemes are presented in Table 1.

In the CIMON approach, all nodes in the network directly send heartbeat messages to the sink every T_{hb} seconds. Fig.3(a) shows a snapshot of this mechanism. The sink is responsible for monitoring the status of all nodes. If heartbeat messages of any node are not presented within T_{mon} seconds, then that node is considered inactive.

In the case of DIMON, parent nodes, derived from existing routing information, assume the responsibility of monitoring their child nodes, as seen in Fig.3(b). Heartbeat messages are no longer forwarded to the sink and a delay longer than T_{mon} seconds causes the parent node to send an inactive message for this node to the sink. Node status information for both CIMON and DIMON will be further interpreted into region status to compare the results with region-based approaches.

The mechanism of CRMON is similar to that of DRMON. It uses the same region representative election process, but after region reps emerge, only heartbeat messages from region reps are sent to the sink. The sink monitors the status of every region. The region status will become inactive if a heartbeat message from that region does not arrive within T_{mon} seconds. To avoid false reports, the sink broadcasts a region status confirmation message to every node in the network after it detects an inactive region. This message will trigger the region representative election process in that region. If a new region rep can be mediated, a heartbeat message sent by the new region rep will resume the active status of that region. Figures 3(c) and (d) illustrate snapshots of CRMON and DRMON, respectively.

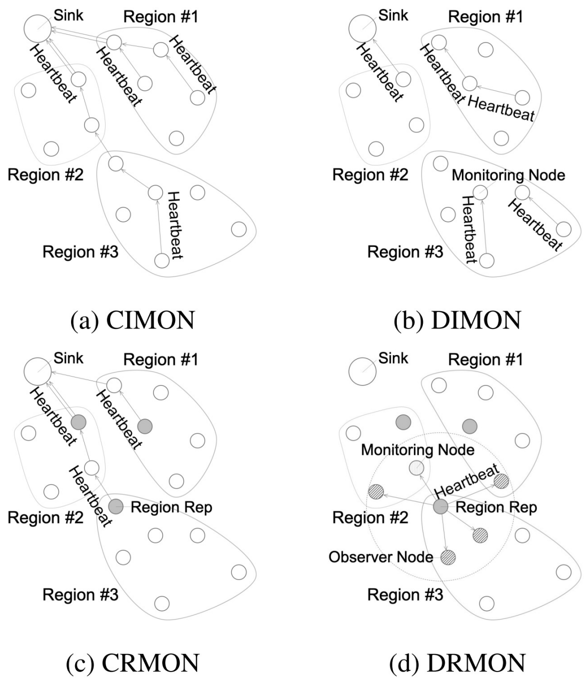


Fig.3: Heartbeat message snapshots of all the four network monitoring schemes

The performance of all network monitoring schemes was evaluated using Cooja simulation platform provided by Contiki Operating System version 3.1. GNU Parallel [31] was employed to execute multiple simulation instances simultaneously. The hardware used in this simulation includes Tmote Sky, a sensor node with an MSP430 processor and a CC2420 transceiver. This platform has 10 kB of RAM and 48 kB of flash memory.

The network consists of several homogeneous sensor nodes and a powerful sink. Nodes are deployed in a two-dimensional area with uniform random placement. Heartbeat messages are transmitted every 20 seconds with random delays to avoid collisions and the monitoring interval is set to $3T_{hb}$. As the topology could change in real situations, all nodes are configured to operate in both TX and RX modes, even if they are leaf nodes. For DRMON, the observing interval used by observers in the region to trigger the region rep reelection process is set to $2T_{hb}$. The simulation was executed for 1 hour. Table 2 summarizes the parameters used in this simulation, where TX

Table 2: Simulation parameters

Parameters	Values
Number of seeds	10
Simulation time	3600 seconds
Number of sinks	1
TX range (R_{TX})	50 meters
Max terrain size	200x200 m^2
Interference range (R_{Int})	100 meters
Heartbeat Interval (T_{hb})	20 seconds
Observing Interval (T_{obs})	40 seconds
Monitoring Interval (T_{mon})	60 seconds

Table 3: Varied simulation parameters

Parameters	Values
Number of regions	3, 5, 9, 25
Number of nodes	16, 25, 49

and interference ranges are typical values for Tmote Sky nodes, and time-related parameters reflect time-critical applications such as landslide monitoring.

We assumed that each node is aware of its route to the sink if there are available routes. Each node selects the nearest node with the smallest number of hops to the sink as its parent node. The re-routing mechanism is triggered immediately after the occurrence of a node fault. We did not choose to enable a dynamic routing protocol to avoid the effect of routing overhead, which would dominate the comparison results. The application data was omitted for the same reason.

We evaluated the Cooja-based simulation for the scenario in which a random region failure was generated at a fixed point in time. Node faults in the defective region occur sequentially every 5 minutes. All faults are permanent, which indicates that nodes are completely destroyed and cannot recover.

As a comparison benchmark, a distributed diagnosis protocol for sensor networks called *WSNDiag* [4] has been implemented to run on Contiki OS. WSNDiag's diagnosis process starts with a designated initiator that broadcasts an "I'm alive" (IMA) message to its neighbors. A node that receives an IMA message for the first time will assign the sender as its parent and rebroadcast the message containing the parent's ID. A parent also keeps track of its children by checking whether a rebroadcast IMA message contains its own ID as the parent. In addition, IMA messages serve as indications that neighbors are fault-free. If a node does not hear an IMA message from a neighbor after a specific amount of time, that neighbor will be diagnosed as faulty. IMA messages are disseminated throughout the entire network in a flood-like manner, forming an aggregation tree via parent-child relationships. After broadcasting an IMA message, each node waits for its children in the aggregation tree to send back diagnosis messages, each of which contains a list of faulty nodes in the corresponding subtree. The node then aggregates all these diagnoses into a single diagnosis message of its own and sends back to its parent. Eventually, the initiator, which is the root of the aggregation tree, will acquire a complete list of faulty nodes. To support region-based diagnosis used in the comparison, faulty nodes are mapped into their designated regions. Any region whose members are all faulty is then diagnosed as a faulty region. In our simulation, the sink serves as the initiator which triggers the diagnosis process every T_{hb} time interval.

3. RESULTS AND DISCUSSION

We first compared the results between ContikiMAC, which represents LPL-based, energy-efficient MAC protocols, and the typical IEEE802.15.4 with no LPL (nullMAC). NullMAC in the Contiki platform denotes the standard CSMA/CA protocol without the duty cycling mechanism. In other words, the RX duty cycle is set to 100%. The numbers of regions and nodes were varied to observe the flexibility and scalability of the monitoring schemes. The parameters used for the evaluation are described in Table 3.

We evaluated the monitoring schemes for both efficiency and effectiveness. The efficiency evaluation metrics include the average power consumption per node and message complexity. In the case of effectiveness, we examined the fault detection accuracy with precision and recall the results and detection delay. All results are shown with error bars representing 95% confidence intervals. In addition, the sink node has the LPL mechanism disabled and is excluded from the results because it is generally attached to an external power source.

3.1 Power Consumption

Low power consumption is crucial for WSN applications to prolong the network's lifetime. The average power consumption can be calculated from the ratio of the CPU ticks for each state of the sensor node within a sampled time interval. The CPU ticks can be retrieved from the Energest module in the Contiki OS. The node's operating states consist of transmitting (TX), receiving (RX), CPU idle, and sleep mode. In sleep or Low Power Mode (LPM), a node consumes considerably less energy compared to other states. Tmote Sky which operates at 3V has different current consumption rates for different modes of operation, as shown in Table 4.

The comparison of average power consumption per node in a network using different MAC protocols is shown in Fig.4. By adopting the energy-efficient MAC protocol, the network monitoring process using ContikiMAC clearly outperforms the nullMAC

Table 4: Current consumption of different operating states

Operating States	Tmote Sky	LoRa Feather
Transmitting (Radio TX)	17.7 mA	120.5 mA
Receiving (Radio RX)	20 mA	11.3 mA
CPU Idle	1.8 mA	9.2 mA
Low Power Mode (LPM)	54.5 μA	0.8 mA

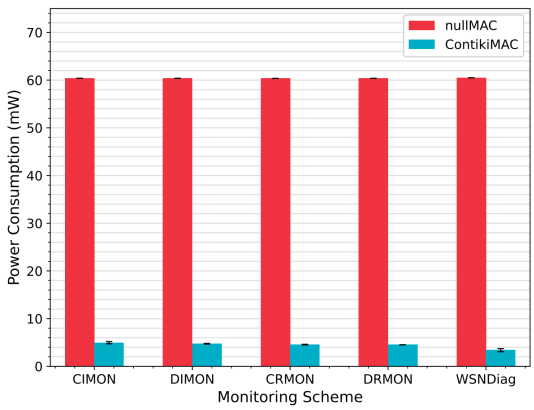


Fig.4: Average power consumption per node using different MAC protocols and network monitoring schemes with 25 nodes divided into 5 regions

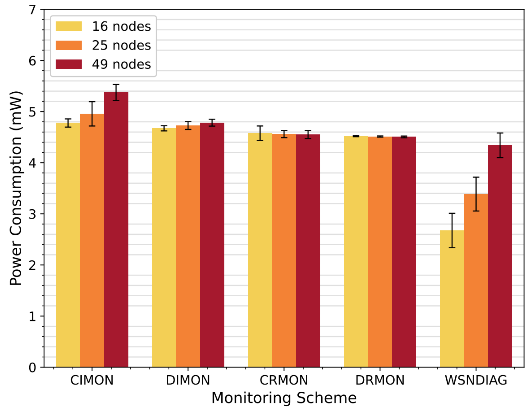


Fig.5: Average power consumption per node using different network sizes and network monitoring schemes, where nodes are divided into 5 regions

process. It can be seen that the nullMAC's power consumption remains the same regardless of monitoring schemes being used because nodes rarely transmit and generally waste most of their energy in the idle state. This suggests that ContikiMAC is favorable in WSN operations with infrequent transmissions, as found in rare event detection applications. Therefore, from now on we will only focus on the results based on ContikiMAC because of its superior power-efficiency. ContikiMAC will also give us clearer contrast among different monitoring approaches.

We studied the scalability of each monitoring scheme by varying the number of nodes in the network. The result presented in Fig.5 shows that CIMON and WSNDiag are the most affected by the increase in network size. As CIMON requires participation of all nodes in the monitoring process and all of the heartbeat messages need to be sent to the sink, the growth of the network size leads to higher power consumption. WSNDiag, which operates in a similar fashion as CIMON but with in-network aggregation of faulty node information, appears to consume the least power in most network sizes. However, the power consumption seems to increase significantly as the network size increases due to the use of network-wide flooding during the broadcast of IMA messages. It is worth noting here that the seemingly low power consumption, but with high variation, of WSNDiag is also due to the fact that in certain scenarios, IMA and diagnosis messages failed to be received as a result of high packet collisions. This caused the monitoring process to abruptly halt before it finished, resulting in lower-than-expected power consumption at the cost of very low monitoring accuracy, as will be discussed later in Section 3.3. The distributed or region-based schemes, on the other hand, consume approximately the same amount of energy regardless of network size, which implies that these schemes are scalable.

The comparison depicted in Fig.5 is based on radio hardware whose power consumption for transmitting and receiving are not much different, i.e., 17.7 mA and 20 mA, respectively. However, certain radio modules allow much higher transmission power. As an example, we additionally calculated the average power consumption of the nodes using current consumption profiles measured from an Adafruit Feather M0 RFM95 LoRa Radio module [32] with identical network topologies. Current consumption rates of LoRa-enabled nodes are shown in Table 4. Fig.6 shows that when operating on a hardware platform requiring high transmission current, WSNDiag would consume the most energy, followed by CIMON and DIMON, whereas CRMON and DRMON would consume the least energy. The reason WSNDiag's consumption for transmission is outstandingly high is due to the heavy use of broadcasts, which result in a large number of repeated transmissions of full packets, as already discussed in Section 1.3.

In conclusion, with respect to power consumption, it is found that region-based approaches (i.e., CRMON and DRMON) are marginally affected by the increase of network size, especially when the cost of transmission is high. By comparing to CIMON, CRMON, and DIMON over a network of 25 nodes, the overall power consumption of DRMON can be reduced by 10%, 2%, and 4%, respectively, while consumes 30% more power than WSNDiag.

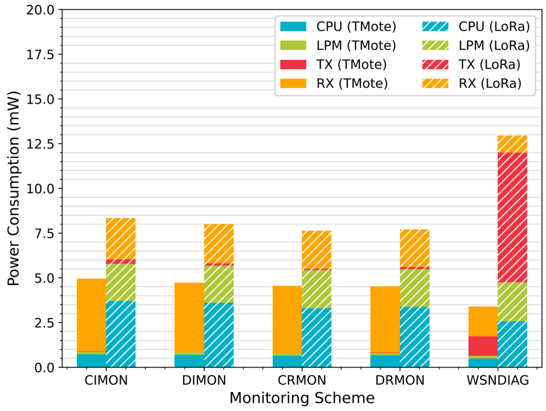


Fig.6: Average power consumption per node using different node models and network monitoring schemes with 25 nodes divided into 5 regions

3.2 Message Overhead

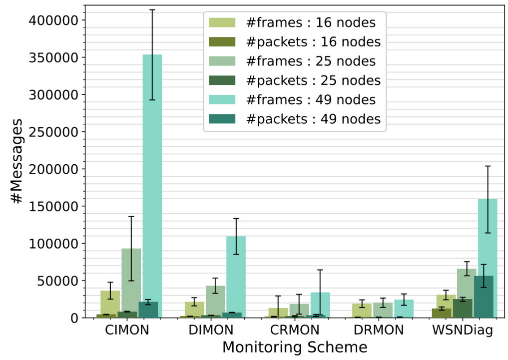
Control message overhead has been evaluated by recording the number of packets and frames used during the entire process. The number of packets is measured at the application level by counting how many times the application invokes the packet send (TX) function and is triggered by packet reception (RX) callbacks. In contrast, the number of frames is measured at the MAC layer by counting how many times MAC frames are transmitted and received. For LPL-based MAC protocols such as ContikiMAC, the number of frames is expected to be larger than the number of packets in both transmission and reception processes due to the use of multiple short strobe frames for unicast and repeated full packet transmission for broadcast, as discussed in Section 1.3. In general, the number of messages is indicative of the magnitude of the network's traffic, which has a direct impact on the actual application's data delivery. Thus, it is desirable to keep the number of messages low for the monitoring process.

Table 5 summarizes all the control packet types, total sizes at the data link layer, and modes of transmission used by all the schemes. It can be seen that their sizes are not much different and none of them are particularly large. In addition, the bandwidth used by each message transmission is dominated by the LPL mechanism rather than the message payload itself. Hence, only the overhead in terms of control message counts is evaluated.

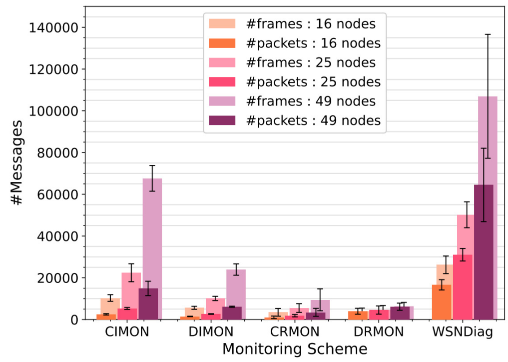
Fig.7 illustrates the message transmission and reception overhead for the five monitoring schemes for various network sizes, where the number of regions is fixed at 5. The addition of nodes clearly increases the number of messages generated in individual-based monitoring schemes, i.e., CIMON, DIMON, and WSNDiag. The region-based mechanisms (CRMON and DRMON) attempt to minimize the participation of nodes in the monitoring process. Thus, the number

Table 5: Sizes of control messages in bytes

Message type	Size (bytes)	Mode
Chance advertisement	22	broadcast
Heartbeat (DRMON)	15	broadcast
Heartbeat (others)	15	unicast
Node/region status report	16	unicast
Region rep election	15	broadcast
WSNDiag's IMA	11	broadcast
WSNDiag's diagnosis	26	unicast



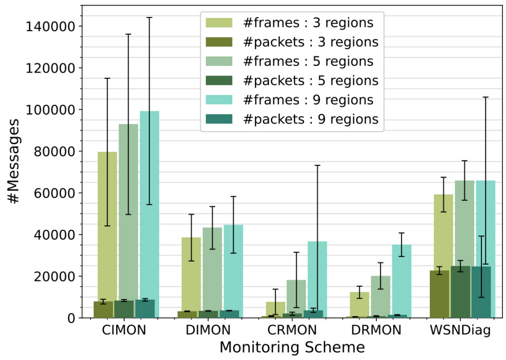
(a) TX



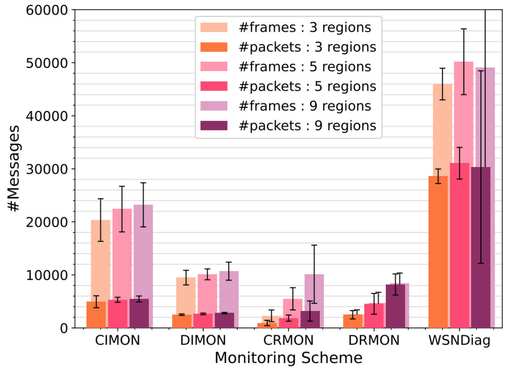
(b) RX

Fig.7: Message overhead using different network sizes, where nodes are divided into 5 regions

of messages is not obviously related to the network size if the number of regions remains the same. It is interesting to see that CIMON shows the largest number of frames transmitted, while WSNDiag shows the largest number of frames received. The reason is every node in CIMON periodically transmits its heartbeats to the sink using unicast transmission. During each transmission, other nodes in vicinity, except the parent, would wake up to one of the strobe frames and find out that the packet is not destined to themselves, so they would go back to sleep immediately, resulting in the low RX count. WSNDiag, in contrast, floods the network with IMA messages using broadcast transmission. All nodes hearing a broadcast would have to stay awake for the entire packet transmission duration, resulting in the very high RX count.

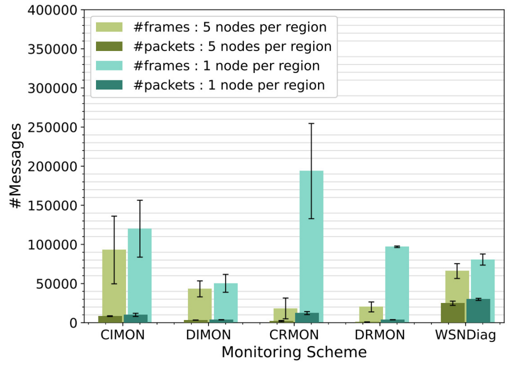


(a) TX

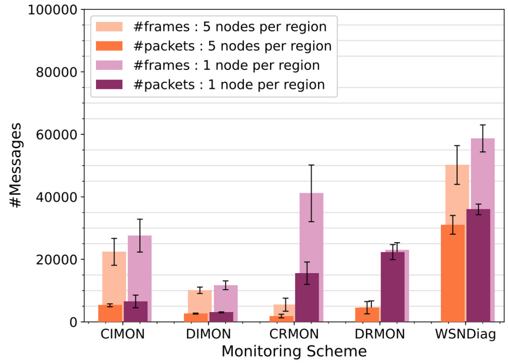


(b) RX

Fig.8: Message overhead using different numbers of regions with total of 25 nodes



(a) TX



(b) RX

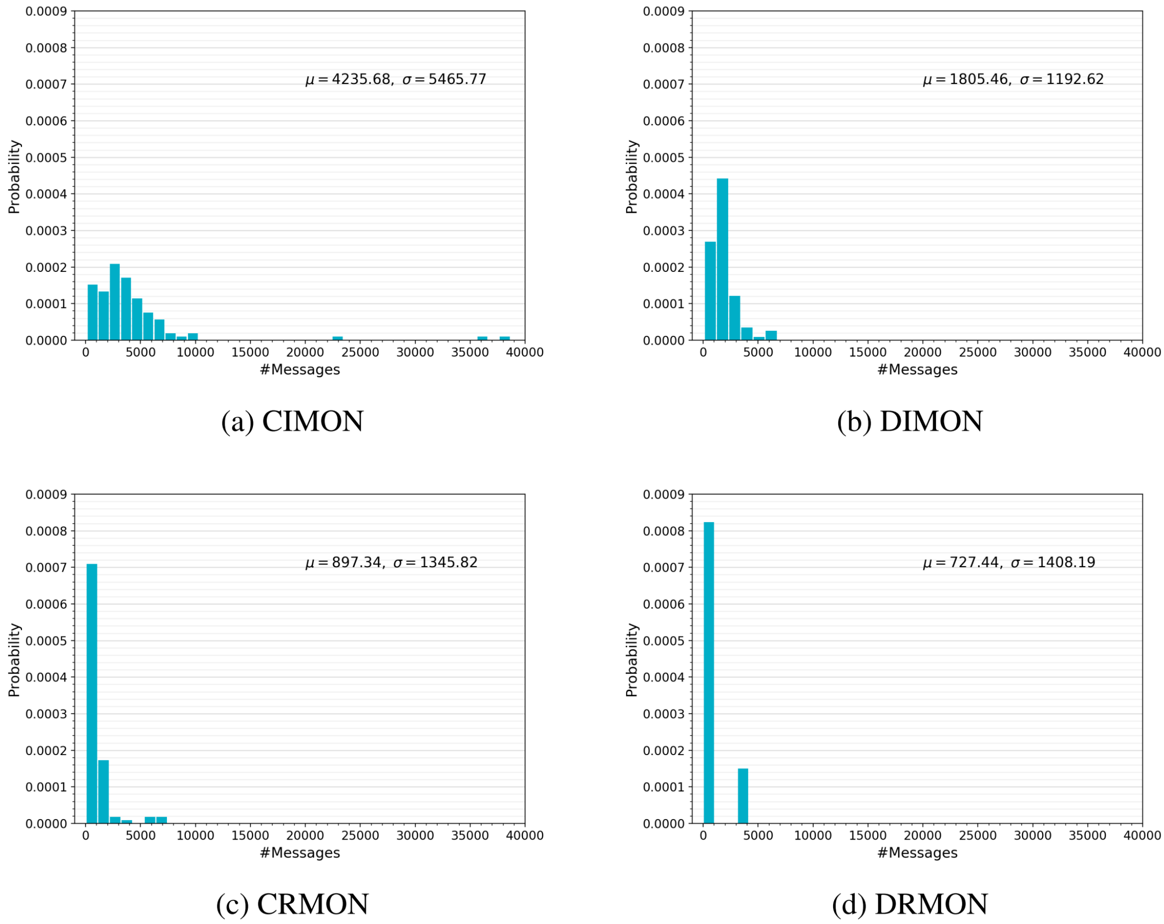
Fig.9: Message overhead using different numbers of nodes per region with total of 25 nodes

Fig.8 compares message overhead from a network of 25 nodes with different numbers of regions. As expected, when the number of regions increases, the individual-based schemes are mostly unaffected, while the region-based schemes' overhead grows almost linearly with the number of regions. Even so, the overhead in terms of packets transmitted is the lowest of all. On the receiving side, DRMON appears to have higher packet reception overhead compared to the other schemes, except WSNDiag, due to the more frequent use of broadcast for heartbeats and the region rep election process.

We also investigated the case where every node in the network is to be monitored individually. In this case, the region-based monitoring schemes consider each single node as a separate region. The result displayed in Fig.9 indicates that the schemes meant for individual monitoring, i.e., CIMON, DIMON, and WSNDiag, perform better than the region-based ones when the region size is reduced to one. It can be seen that the message overhead of CRMON and DRMON drastically increase. This implies that the performance of region-based schemes tends to deteriorate if the number of regions increases due to the overhead of the unnecessary region representation election process during the initializing state.

Networks with evenly distributed processing load among several nodes are preferable to those that tend to concentrate processing load on a few nodes, which leads to their energy quickly depleted. We visualize the distribution of message transmission counts among all nodes in the network in Fig.10. Fig.10(a) and (b) show high overhead across many nodes in the individual-based schemes due to periodic heartbeat transmission. The situation is worse in CIMON because all heartbeats must be relayed to the sink, inducing unevenly large load over a few nodes near the sink. With CRMON, as shown in Fig.10(c), the overall overhead is reduced but there is still a small set of nodes that process a lot of messages. For DRMON, as shown in Fig.10(d), most of the nodes have no monitoring load at all, except some region reps that are distributed over the area but their message counts are relatively low. This suggests that region reps should be periodically re-elected to achieve better balance in message distribution.

In summary, while it is obvious that the fewer messages reported to the sink, the better the performance, we also found out that a LPL-based MAC has a significant impact on the aspect of message overhead, especially with heavy use of broadcast transmission. The region-based approaches (CRMON and DRMON) help lower the number of messages because only region representatives are responsible for send-



(a) CIMON (b) DIMON

(c) CRMON (d) DRMON

Fig.10: Distribution of message overhead among all nodes for different monitoring schemes with 25 nodes divided into 5 regions

ing periodic heartbeats. The distributed approaches (DIMON and DRMON) help prevent transmitting of periodic heartbeats all the way to the sink. Hence, DRMON yields the best performance in this regard. WSNDiag, while attempting to reduce the number of messages by aggregation, shows the worst message overhead due to the heavy use of broadcast in the LPL-based MAC environment. DRMON avoids this problem by limiting broadcast transmission to only the region rep election process, which happens just once at the very beginning or when the current region rep becomes faulty in a region. Broadcast is later employed for transmitting heartbeats by the region reps, which are only a small fraction of nodes. In terms of overall message overhead, DRMON could reduce the number of packet transmissions by 88%, 66%, 69%, and 96% compared to CIMON, CRMON, DIMON, and WSNDiag, respectively.

3.3 Fault Detection Accuracy

To evaluate the effectiveness of network monitoring schemes in terms of accuracy, the reported status of each region is compared against its ground truth status for every point in time. The *ground truth status*, $G_{r,t}$, is the actual state of a region r at time t, which is 1 if region r is active and has at least one route to the sink at time t; otherwise, it is 0. The *reported status* of a region r at time t as reported by the sink is denoted as $R_{r,t}$. It is 1 if the sink reports that region r is currently active to the operator, or 0, otherwise. Fig.11 illustrates how the ground truth and reported statuses are used to determine true/false positive and negative rates of fault detection over the time period T. Between the time t_{start} and $t_{reportone}$, both ground truth and reported statuses indicate no fault for this region ($G_{r,t}$ and $R_{r,t}$ are both 1), indicating a true negative duration. Between the time $t_{reportone}$ and $t_{reporttwo}$, $G_{r,t}$ is still 1 but $R_{r,t}$ becomes 0 (fault reported), indicating a false positive duration, which could happen due to losses of heartbeats, for example. Between the time t_{fault} and $t_{reportthree}$, $G_{r,t}$ becomes 0 (actual fault occurring) but $R_{r,t}$ is 1 (fault not reported), indicating a false negative duration. This duration is also interpreted as *fault detection delay* which measures the amount of time elapsed after a fault occurs until the fault is detected. And finally, between the time $t_{reportthree}$ and t_{end}, $G_{r,t}$ and $R_{r,t}$ are both 0, indicating a true positive duration.

In this work, we use *precision* and *recall* as the metrics to indicate the fault detection accuracy. Precision is defined as the ratio of the amount of the time during which faults are correctly reported (i.e., $G_{r,t}$ and $R_{r,t}$ are both 0) to the total amount time faults are reported (i.e., $R_{r,t}$ is 0). Given a set of discrete points in time T, the precision with respect to the region r is then defined as follows:

$$Precision(r) = \frac{|\{t \in T \ : \ G_{r,t} = 0 \wedge R_{r,t} = 0\}|}{|\{t \in T \ : \ R_{r,t} = 0\}|} \tag{1}$$

Recall is the ratio of the amount of time during which faults are correctly reported (i.e., $G_{r,t}$ and $R_{r,t}$ are both 0) to the total amount of time faults actually occur (i.e., $G_{r,t}$ is 0). The recall with respect to the region r is defined as follows:

$$Recall(r) = \frac{|\{t \in T \ : \ G_{r,t} = 0 \wedge R_{r,t} = 0\}|}{|\{t \in T \ : \ G_{r,t} = 0\}|} \tag{2}$$

Simulation experiments on a network of 25 nodes with 5 regions were conducted with the same parameters shown in Table 2. The nodes were uniformly distributed over the terrain of size 150x150 m^2, so the network diameter was approximately 4-5 hops. One of the regions was randomly chosen for triggering fault. Nodes in the region were made faulty one at a time, with an interval of 300 seconds apart between two consecutive faults, until the region became completely faulty. Precision, recall, and detection delay averaged across all the regions for all five schemes are shown in Table 6. Overall, CIMON gives the best performance in terms of both accuracy and delay as each node sends heartbeats directly to the sink, at the expense of highest power consumption. As expected, the region-based approaches (CRMON and DRMON) gives the highest detection delay because their mechanisms involve a few more steps to be certain that all nodes in a particular region is faulty. Their lower precision values also suggest that some regions are incorrectly identified as faulty while they are not. In case of CRMON, a few consecutive lost heartbeats from a region rep would trigger the fault detection while CIMON likely receives heartbeats from other region members. In case of DRMON, an observer may incorrectly report a fault of its monitored region when the current region rep is gone and the region is undergoing a new region representative election process.

The situation is quite unusual for WSNDiag. Its precision and detection delay are very low and its recall is very high. After further investigation, it was found that WSNDiag reported a lot of false positives (i.e., wrong fault reports) due to losses of IMA and diagnosis report messages caused by packet collisions. A node whose IMA message has not been heard before timeout, or whose diagnosis information

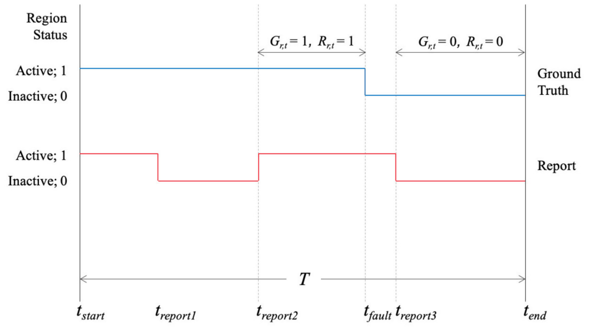


Fig.11: Diagram of ground truth ($G_{r,t}$) and fault report ($R_{r,t}$)'s status

Table 6: Detection accuracy using different monitoring approaches

	Precision	Recall	Delay
CIMON	97.01%	99.44%	13.41 s
DIMON	86.98%	88.84%	29.39 s
CRMON	86.79%	98.54%	35.51 s
DRMON	91.61%	97.94%	39.78 s
WSNDiag	43.72%	99.97%	0.64 s

never reaches the sink, will be immediately reported as faulty. The chance of this happening due to packet collisions is very high as a result of heavy use of broadcast during the flooding of IMA messages, which gets worsened even further by the LPL mechanism.

In summary, with both the precision and recall of greater than 90% and the fault detection delay within the guaranteed range of $T_{hb}+T_{mon}$ (80 seconds), DRMON seems to be the most suitable approach when power consumption and message overhead are taken into account. Typical responses when receiving a fault report would be to send staff over the area to investigate and possibly replace the faulty nodes in order to keep the sensor network readily operational. Therefore, fault detection latency within the range of a few minutes should generally be tolerable. Nevertheless, the latency could also be adjusted by setting T_{hb}, T_{obs}, and T_{mon} to preferred values.

4. CONCLUSIONS

Low-power listening MAC protocols significantly assist in reducing power consumption of WSNs by preventing idle listening and lowering transmission process loads. Traditional network monitoring, which requires periodic updates of the node status, can cause an increase in transmission, which greatly reduces network lifetime. In this work, we propose *DRMON*, a distributed region-based monitoring scheme that can be used to monitor wireless sensor nodes that are physically divided into groups, called regions. This approach increases energy efficiency and reduces the message overhead generated during the monitoring process while maintaining comparable detection accuracy.

Evaluations of centralized individual-based (CIMON), distributed individual-based (DIMON), and centralized region-based (CRMON) monitoring schemes, along with an existing distributed diagnosis protocol for WSN called WSNDiag, were conducted to compare their performance with DRMON. The simulation results show that, by comparing to CIMON, CRMON, and DIMON, the overall power consumption of DRMON can be reduced by 10%, 2%, and 4%, respectively, while consumes 30% more power than WSNDiag. However, WSNDiag was later found to cause a lot of packet losses due to collisions, leading to very poor monitoring accuracy and unusually low power consumption. In terms of message overhead, DRMON outperforms the other monitoring schemes in cases where the network has a low number of regions. However, if the number of regions is high, individual monitoring approaches are recommended as there is no message overhead for electing and maintaining the region's representatives (i.e., region reps). Overall, DRMON could reduce the number of packet transmissions by 88%, 66%, 69%, and 96% compared to CIMON, CRMON, DIMON, and WSNDiag, respectively. It can be concluded that, with both the precision and recall of greater than 90% and the fault detection delay within an acceptable range, DRMON seems to be the most suitable approach when power consumption and message overhead are taken into account.

In future work, region reps rotation process should be implemented to further improve the performance of DRMON. Further evaluation of these monitoring schemes could be facilitated by the investigation of additional parameters such as heartbeat and monitoring intervals, as well as other energy-efficient MAC protocols.

ACKNOWLEDGMENTS

This research was supported in part by the Faculty of Engineering, Kasetsart University, Bangkok, Thailand. Computing facilities for this research were supported by the Advancing Co-design of Integrated Strategies with Adaptation to Climate Change (ADAP-T) of the Japan International Collaboration Agency (JICA)/Japan Science and Technology Agency (JST) Science and Technology Research Partnership for Sustainable Development (SATREPS).

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol.38, pp.393-422, March 2002.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher and B. Ohlman, "A survey of information-centric networking," in*IEEE Communications Magazine*, vol.50, no.7, pp.26-36, July 2012.

[3] T. Khampeerpat and C. Jaikaeo, "Mobile Sensor Relocation for Nonuniform and Dynamic Coverage Requirements," *IEICE Transactions on Information and Systems*, vol.E100D, no.3, pp.520-530, 2017.

[4] S. Chessa and P.Santi, "Crash faults identification in wireless sensor networks," *Computer Communications*, vol.25, no.14, pp.1273-1282, September 2002.

[5] A. Dunkels, B. Gronvall and T. Voigt, "Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors," *29th Annual IEEE International Conference on Local Computer Networks*,pp. 455-462,. 2004

[6] S. Guo, Z. Zhong and T. He, "FIND: Faulty Node Detection for Wireless Sensor Networks," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems. SenSys '09*, New York, NY, USA: ACM, pp.253-266, 2009.

[7] K. Ni, N. Ramanathan,M.N.H. Chehade, L. Balzano, S. Nair, S. Zahedi, et al. "Sensor Network Data Fault Types," *ACM Trans Sen Netw*, vol.5, no.3, pp.25:1-25:29, Jun. 2009.

[8] J. Chen, S. Kher and A. Somani. "Distributed Fault Detection of Wireless Sensor Networks," in *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks. DIWANS '06.*, New York, NY, USA: ACM; pp.65-72, 2006.

[9] Y. Zhang and X. Yuan, "Fault diagnosis in clustering WSN based on neighbor cooperation," in *2016 Chinese Control and Decision Conference (CCDC)*, pp. 1803-1807, 2016.

[10] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang and M. Mukherjee, "A Survey on Fault Diagnosis in Wireless Sensor Networks," in *IEEE Access*, vol.6, pp.11349-11364, 2018.

[11] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler and D. Estrin. "Sympathy for the Sensor Network Debugger," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems. SenSys '05.*, New York, NY, USA: ACM; pp.255-267, 2005.

[12] A. Meier, M. Motani, H. Siquan and S. Künzli, "DiMo: Distributed Node Monitoring in Wireless Sensor Networks," in *Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems. MSWiM '08.*, New York, NY, USA: ACM, pp.117-121, 2008.

[13] Y. Liu, K. Liu and M. Li, "Passive Diagnosis for Wireless Sensor Networks," in *IEEE/ACM Transactions on Networking*,vol.18, no.4 pp.1132-1144, Aug. 2010.

[14] Y. J. Zhao, R. Govindan and D. Estrin, "Resid-

ual energy scan for monitoring sensor networks," *2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002 (Cat. No.02TH8609)*, vol.1, pp.356-362, 2002.

[15] C.F. Hsin and M. Liu, "A Distributed Monitoring Mechanism for Wireless Sensor Networks," in *Proceedings of the 1st ACM Workshop on Wireless Security. WiSE'02.*, New York, NY, USA: ACM, pp. 57-66, 2002.

[16] K. Liu, Q. Ma, X. Zhao and Y. Liu, "Self-diagnosis for large scale wireless sensor networks," *2011 Proceedings IEEE INFOCOM*, pp. 1539-1547, 2011.

[17] Q. Ma, K. Liu, X. Miao and Y. Liu, "Sherlock is around: Detecting network failures with local evidence fusion," *2012 Proceedings IEEE INFOCOM*, pp. 792-800, 2012.

[18] Q. Ma, K. Liu, X. Xiao, Z. Cao and Y. Liu, "Link Scanner: Faulty link detection for wireless sensor networks," *2013 Proceedings IEEE INFOCOM*, pp. 2688-2696, 2013.

[19] D. Dong, X. Liao, Y. Liu, C. Shen and X. Wang, "Edge Self-Monitoring for Wireless Sensor Networks," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 514-527, March 2011.

[20] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan "Energy-efficient communication protocol for wireless microsensor networks," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol.2, pp.10 pp., 2000.

[21] J. Kim, S. Park, Y. Han and T. Chung "CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks," *2008 10th International Conference on Advanced Communication Technology*, pp.654-659, 2008.

[22] Y. Zhang and X. Yuan, "Fault diagnosis in clustering WSN based on neighbor cooperation," in *2016 Chinese Control and Decision Conference (CCDC)*, pp.1803-1807, 2016.

[23] C. Titouna, M. Aliouat and M. Gueroui, "FDS: Fault Detection Scheme for Wireless Sensor Networks," *Wireless Personal Communications: An International Journal*, vol.86, pp.549-562, 2016.

[24] F. Zhen, F. JingQi and S. Wei, "The gateway anomaly detection and diagnosis in WSNs," in *2016 Chinese Control and Decision Conference (CCDC)*, pp.2401-2406, 2016.

[25] S. Shao, S .Guo and X. Qiu, "Distributed fault detection based on credibility and cooperation for WSNs in smart grids," *Sensors*, vol.17, no.5:983, pp.1-14, April 2017.

[26] R.R. Swain, P.M. Khilar and S.K. Bhoi, "Heterogeneous fault diagnosis for wireless sensor networks," *Ad Hoc Networks*, vol.69, pp.15-37, 2018.

[27] V. Cionca, T. Newe and V. Dadârlat, "TDMA Protocol Requirements for Wireless Sensor Networks," in *2008 Second International Conference on Sensor Technologies and Applications (SENSORCOMM 2008)*, pp.30-35, 2008.

[28] M. Buettner, G.V. Yee, E. Anderson and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems. SenSys '06.*, New York, NY, USA: ACM, pp.307-320, 2006.

[29] A. Dunkels, *The contikimac radio duty cycling protocol*, Swedish Institute of Computer Science; 2011.

[30] K. Pattamasiriwat and C. Jaikaeo, "Evaluation of low power listening MAC protocol on network monitoring in wireless sensor networks," in *2017 9th International Conference on Knowledge and Smart Technology (KST)*, pp.260-265, 2017.

[31] O. Tange, "GNU Parallel - The Command-Line Power Tool," *login: The USENIX Magazine.*, vol.36, no.1, pp.42-47, Feb. 2011.

[32] Adafruit. Adafruit Feather M0 Radio with LoRa Radio Module;. [Online; accessed 5-December-2019]. `https://learn.adafruit.com/adafruit-feather-m0-radio-with-lora-radio-module`.

Krita Pattamasiriwat received her B.Eng. degree in Computer Engineering from Kasetsart University in 2014. She is now an Upstream SAP Financial analyst at ExxonMobil, Ltd.

Chaiporn Jaikae received his B.Eng. degree in Computer Engineering from Kasetsart University, Thailand, and his M.S. and Ph.D. degrees in Computer and Information Sciences from the University of Delaware in 1996, 1999, and 2004, respectively. He is currently a faculty member in the Department of Computer Engineering at Kasetsart University. His research interests include mobile wireless ad hoc and sensor networks, topology control, energy harvesting, and Internet of Things.