IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 15, 2020, accepted September 19, 2020, date of publication September 25, 2020,
date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026939

Improving Energy Efficiency With Content-Based Adaptive and Dynamic Scheduling in Wireless Sensor Networks

MUHAMMAD NAWAZ KHAN[1], HASEEB UR RAHMAN[1], MOHAMMED AMIN ALMAIAH[2], (Member, IEEE), MUHAMMAD ZAHID KHAN[1], AJAB KHAN[1], MUSHTAQ RAZA[3], MOHAMMED AL-ZAHRANI[4], OMAR ALMOMANI[5], AND RAHIM KHAN[3]

[1]Network Systems and Security Research Group, Department of Computer Science & IT, University of Malakand, Chakdara 18800, Pakistan
[2]Department of Computer Networks and Communications, College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia
[3]Department of Computer Science, Abdul Wali Khan University Mardan, Mardan 23200, Pakistan
[4]College of Computer Science and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia
[5]Computer Network and Information Systems Department, The World Islamic Sciences and Education University, Amman 11947, Jordan

Corresponding author: Rahim Khan (rahimkhan@awkum.edu.pk)

This work was supported by the Deanship of Scientific Research from King Faisal University.

ABSTRACT Wireless Sensor Networks (WSNs) have revolutionized the era of conventional computing into a digitized world, commonly known as ''The Internet of Things''. WSN consists of tiny low-cost sensing devices, having computation, communication and sensing capabilities. These networks are always debatable for their limited resources and the most arguable and critical issue in WSNs is energy efficiency. Sensors utilize energy in broadcasting, routing, clustering, on-board calculations, localization, and maintenance, etc. However, primary domains of energy consumption at node level are three i.e. sensing by sensing-module, processing by microprocessor and communication by radio link. Extensive sensing, over-costs processing and frequent communication not only minimize the network life-time, but also affects the availability of these resources for other tasks. To increase life-time and provide an energy-efficient WSN, here we have proposed a new scheme called ''A Content-based Adaptive and Dynamic Scheduling (CADS) using two ways communication model in WSNs''. CADS dynamically changes a node states during data aggregation and each node adapts a new state based on contents of the sensed data packets. Analyzer module at the Base-Station investigates contents of sensed data packets and regulates functions of a node by transmitting control messages in a backward direction. CADS minimizes energy consumption by reducing unnecessary network traffic and avoid redundant message-forwarding. Simulation results have been shown that it increases energy-efficiency in terms of network life-time by 9.65% in 100 nodes-network, 11.36% in 150 nodes-network and 0.94% in 300 nodes. The proposed scheme is also showing stability in terms of increasing cluster life by 87.5% for a network of 100 nodes, 94.73% for 150 nodes and 53.9% in 300 nodes.

INDEX TERMS Adaptive, dynamic, wireless sensor networks (WSNs), two-way communication, analyzer, scheduling.

I. INTRODUCTION

Wireless Sensor Network (WSN) is a group of resource constraint minute sensors, distributed over an area for monitoring physical or environmental phenomenon. These sensors have limited processing capabilities in terms of computation, memory and communication. They are used to acquire any physical or environmental condition and convert these analog patterns into digital form. A *Sink/Base station* (BS) is a central point where all sensors send their sensed values for further analysis [1], [2]. BS is rich in resources and is considered to be the decision-center for all sensor nodes. WSNs are typically used in recording physical conditions (temperature, humidity, pressure) traffic monitoring, weather condition,

The associate editor coordinating the review of this manuscript and approving it for publication was Chan Hwang See.

intensities (vibration intensity, illumination intensity, sound intensity), and vital body functions [3]. They are widely used in medical, health-care, agriculture, and chemical concentrations [4]. Sensor nodes convert measurement in physical phenomenon into digital form and quantify these values for various calculations. Sensor nodes are deployed in a targeted area where data is collected periodically or continuously [5]. Sensors are programmable, easily controllable, and at the same time are used for plugging with other smart appliances. Due to its small size, low power, low cost and minimal user intervention, it is widely used in many practical applications. The frequent uses of sensors and smart devices have a greater impact on our day to day lives [6], [7].

Sensor nodes in WSNs are either static or mobile. In addition, they are deployed closely enough that a node's premises are detected by other nodes. Communication links in WSN are mostly fragile due to rapidly changing topology and random deployment [8], [9]. WSNs are usually deployed in an inaccessible and hazardous scenario where normal data collection is challenging. Now computing is mainly focused on distributed and ubiquitous services with wireless connectivity in verities of devices [10], [11]. Sensor nodes are the basic building block in providing these pervasive services due to their miniature size and easily pluggable with other appliances. Nearly all smart devices are using sensor nodes for obtaining data from real-world scenarios and provides services according to user mode and according to the context. WSN brings all these devices in a uniform layer and arrange them in a context [12], [13].

Due to the resource-constrained nature and unique characteristics, WSNs are always at risk of faults and failures [14]. The reason being, most of the sensor network's energy is exhausted in aggregation and forwarding of data. As a result, most of the devised solution focused on energy-aware computing with distributed data collection [15], [16]. Another widely used technique in WSNs for energy reservation is Clustering. In clustering, each zone selects a master node (Cluster Head (CH)) for the collection of data from all nodes and forwards to BS [17], [18]. Communicating of CH only with BS managed network structure and save a good amount of energy. Usually, sensor nodes in WSNs work in continuous sensing and processing manner. Furthermore, it performs data aggregation and data forwarding of the sensed data, which reduces energy level of nodes and effect network life-time. To minimize energy consumption and prolong the network life-time, numerous schemes have been proposed. Most of these schemes have mainly focused on energy dissipation [19], [20]. Different phases are defined in various protocols to control and monitor activities in the network.

Instead of usual energy dissipation techniques in WSN, actual dissipation at node level are sensing, processing and communication. Extensive sensing, over-cost processing and frequent communication in the network not only drains energy level of a node but also affects other resources (processing, bandwidth, buffer) at a great extent. By regulating and controlling of a node's components, energy consumption can minimize and it can increase life-time of network. Always needed a scheme which:

- ➢ Minimize Extensive Sensing, reduce over-cost Processing and decrease frequent communication.
- ➢ Avoids un-necessary message-forwarding (redundant sensing values).
- ➢ Minimizes Re-Clustering Overhead (stable topologies)
- ➢ Reduces overall Network Traffic (Control Messages & Data Packets).
- ➢ Prolongs network life-time (Consumed Minimum Energy).

The main contributions of the article include:

1) Four state model is proposed where States are defined based on internal components of the sensor.

2) An analyzer module at BS is implemented, which analyzes the contents of sensed data packets. The analyzer module directs nodes to acquire any state by sending control messages in backward communication.

3) A comparative study of CADS with other state-of-the-art using different parameters.

4) The overall energy consumption is minimal compared to other schemes after calculating different metric values.

The rest of the paper is organized as: Section-II presents a brief literature review, Section-III explains the proposed scheme with details, Section-IV is simulation details for CADS, Section-V is the comparison with existing schemes, and Section-VI concludes the article and provides future directions for further enhancing the work.

II. LITERATURE REVIEW

Energy dissipation in WSNs is always debatable and numerous schemes have proposed for efficient use of energy. Most of the schemes have focused on network establishment and topological structure of the network, while few of them have MAC level approaches for energy efficiency. Here we have categorized these approaches in Cluster-Based Models and State-based Models.

A. CLUSTER BASED MODELS

Cluster-based models have mainly included those schemes which network is managed and classifies into zone-based structures. Efficient CH selection has a greater impact on resource utilization. Low-Energy Adaptive Clustering Hierarchy (LEACH) [21] is the most famous clustering routing protocol for WSNs. In LEACH, the selection of CH is based on probability for the current round. Hence, CH selection is not uniformly distributed, which produces disconnected nodes. Similarly, CH selection is a multifarious process creating overhead in the network and consumes more energy. Multi-hop-LEACH [22] is the modification in LEACH protocol, operates in multi-hop inters cluster operation and multi-hop intra cluster operation. Data is sensed by sensors and sends it to CHs based on time out period. Forwarding annoying messages may lead to endless utilization of energy. The Quadrature-LEACH (Q-LEACH) [23] is another modified form of LEACH. Q-LEACH is optimized in a way for power management as well as for better services. Nevertheless, all modification in LEACH is based on hierarchical clustering while in WSNs hierarchy is not the optimum solution. Hybrid LEACH [24], [25] is used for better selection of CH depends on remaining energy. Using threshold condition for each round, CH is elected between high and low energy levels.

PEGASIS [26] has removed all drawbacks in LEACH. Only two messages will have required for a leader to transmit data to BS instead of twenty messages in LEACH for full coverage. However, in this scheme, some bulky and unnecessary messages are circulating in the network, which leads to curtailing energy levels of nodes. Another Multi-hop communication model is CEER [27], equally useful for CHs and for BS. This algorithm works in rounds, each node uses distance, residual energy, and probability to own fitness function. BS selects CH based on the Genetic algorithm. However, this scheme fails in managing the algorithm and in selecting the best path. For balance energy consumption in fog supported WSNs, New Stable Election Protocol (SEP) [28] has proposed. CH is elected based on remaining energy, distances with CHs. A threshold is set up for checking energy and other parameters for the primary and secondary CH election process. Prolong SEP [29] protocol is a successor for SEP. For efficient use of resources and minimize energy consumption, network is alienated in zones and each zone is monitor/command by a CH. However, in this scheme, the network structure is unstable because of node heterogeneity and frequent changing scenarios.

Location-based protocol like Minimum Energy Communication Network (MECN) [30] is used to calculate sub-networks using minimum energy on GPS interface. Failure occurs when the network is scattered and obstacles arose between neighbor nodes. Geographic Adaptive Fidelity (GAF) [31] is another location-based protocol, is initially designed for Mobile Ad hoc networks and later on modified for WSNs. In Virtual Grid, nodes communicate with each other in fixed zones. The scheme is looking uncertain when mobility of SNs is involved with dynamic changing topologies. Geographic and Energy Aware Routing (GEAR) [32] calculates geographic information and this information is used to select CH. GEAR is good for overall network management but not recommended for low energy communication networks like WSNs.

SPIN [2] is a Data-Centric approach for energy conservation, where each node works as BS. Based on negotiation for routing, redundant data packets are removed. However, a negotiation message produces congestion in the network. COUGAR [33] is another variant of the data-centric scheme which arranges the network in a distributed database system. On network layer, abstract query processing is running for data collection, aggregation and data forwarding. The COUGAR works on upper layer where more energy is acquired while WSNs need lower layer policies for aggregation and data forwarding. For balanced CH selection and equivalent clusters, other schemes suggest many parameters like energy [7], distance [2] and coverage, convergence areas [9], [34]. An energy-efficient routing protocol known as a Distributed Energy-Efficient Adaptive Clustering Protocol (DEACP) [25] has proposed. Using queue overflow in routing strategy among all cluster members, dropping probability is minimized. It adjusts energy dissipation among all nodes. The CH selection is optimized and all area is covered with full connectivity and minimum consumption of energy. The radio module of a node is turned off for a definite time according to a sleeping schedule. A comprehensive discussion of routing in WSNs and IoT is presented in [35]. Many factors affect performance during communication of data from sensor to sink. This section discussed all those parameters including delay, latency and throughput on layered basis.

For reducing packets loss during mobility with efficient CH selection, LEACH-MF [36] is proposed. It enhances the former clustering protocol for mobility patterns of nodes based on fuzzy interference system. Those nodes which having lesser mobility with better pause timing and more energy are best candidates to be the CH. They used the first order radio model with mobility patterns. Three phases algorithm are used i.e. the CH selection phase, Cluster formation phase and Data transmission phase. Analysis of this scheme proves enhancement in energy-efficiency and reduces packet loss due to mobility. Another clustering protocol based on fuzzy logic is SET-FL [37] to increase energy efficiency in distributed WSNs. Free-nodes are used for direct communication to BS and cluster-nodes are used for collecting data from leaf-sensors. The Four fuzzy metrics used are energy level of BS, energy level of nodes, node proximity and position of nodes with CH. For balancing energy utilization in large scale networks and resolve set cover problem, KMSPGA [38] is proposed. It basically schedules different sensors into complete disjoint sets. Dimensionality reduction is achieved by divide and conquer method. For refining more novelty, KMSPGA uses eight parameters for increasing efficiency. It improves coverage rate, success rate, scalability and is showing high robustness. For balance energy consumption for optimum network topology, ULGATE and ULGAT-GAF [39] is proposed. It uses physically adaptive confirmation for some node to become sleep. It mainly consists fitness function, selection, crossover and mutation processes and optimized problem formulation. Genetic algorithm and AHP uses reshape topology by initiating partial CMs.

B. STATE-BASED MODEL

Many state-based models have also proposed for energy efficiency, in which node acquires different transition states (Active/Sleep/Idle, etc.). The first node level sensing is PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks [40]. In this approach, a node works in three operating modes. Main drawback, when a node is in active mode, it will never change its state to sleep again. PECAS [41] removes main drawbacks in PEAS. Nodes wake-up and start probing. If no communication is found for an active node, it becomes active. However, active node can adapt sleep state after a specific time. In both approaches, nodes are deployed in a random and unplanned topology. Another improvement in PEAS for the deployment of sensors in a deterministic way is "Energy Efficient Coverage Guaranteed protocol (EECG)" [42]. Deterministic deployment of nodes minimizes the cascading effects in which near to BS node will frequently drain because of huge and multi-time usage of these nodes. EEGG minimizes the cascading effect by adding additional sleepy nodes. However, many nodes connected directly to BS in a deterministic way and dies frequently.

Random Back-off Sleep Protocol (RBSP) [43] is another improvement in PEAS. Active node information is used for probing and for changing states in three states. States are calculated from different parameters using back-off algorithm. The discharge curve back-off sleep protocol (DCBPS) [44] removes the drawback of RBSP and based on back-off sleep timing. The sleeping time of neighbors is calculated and based on the battery discharge curve. This algorithm only activates sleepy nodes when an active node is near to drain completely. Coverage and Energy Strategy for WSNs (CES) [45] is working in two stages and each node can adapt four conditions. In first stage, all nodes are working and interchange their location information to other neighbors. As active nodes provide coverage and connectivity, these nodes may die more frequently compared to other nodes. Maintaining sensing coverage and connectivity in large sensor networks (CCP) [46] removes the drawback of CES. In CCP, nodes are basically in three states: sleep, active, and listen with circular topology and location awareness. Based on sensing range, nodes deactivate and activate its states from sleep to active. Fewer active nodes may not cover full area and as a result, disconnect network. To remove the drawbacks of CCP, another scheme known as "An Enhanced Coverage Control Protocol (ECCP)" [47] has proposed. ECCP is used for covering full area in a specific region and a balance zone-based active sensing mechanism. More nodes are used for full-time coverage. ECCP is famous for deploying more active nodes for covering the boundaries of overlap areas. Another scheme for full coverage and with minimum number of active nodes is OGDC [46], [48]. Nodes in this scheme are synchronized, location-aware and coverage area is twice the sensing area. Node actives from an Undecided state if there is a minimum overlapping region. However, changing state after each round is time-consuming and result, more overhead.

Energy-efficient protocol for deterministic and Probabilistic Coverage (PCB) [49] is proposed for reducing overlapping zones. In PCB, nodes are activated by an activator, which working on probabilistic and deterministic ways. Many nodes work as an activator on the hexagonal structure and activate other nodes in Active and Sleep modes. All those nodes will be activating; whose energy time is minimum. Probabilistic Coverage Preserving Protocol (PCPP) [50] is an enhanced form of PCB. In PCPP, BS transmits a message for each node randomly to adapt active-state or remains in sleep. When the network is established and some uncovered intersection is discovered, roughly new and unused sensors are deployed there. Balanced Energy and Coverage Guaranteed protocol (BECG) [51] is another state-based scheme. BEGG works in three states, Sleeping, Checking and Working-state. Working and sleeping nodes exchange its states to each other. But the exact/proximate number of working nodes in target area is not possible known in prior. Coverage enhancement algorithms for distributed mobile sensors (EBC) [52] removes the problem of redundant covered areas by using mobility of the nodes. A sensor network is divided into many polygons using the Voronoi diagram. From all polygons, uncovered holes have identified. If area is increased, new sensor needs moves to that new area.

Connectivity preserving localized coverage algorithm (CPL) [53] is another state based scheduling scheme. In CPL, Area of Interest (AoI) is determined. Overlapping coverage areas are calculated by the Euclidian algorithm. Nodes receive Hello message and respond to CH and send their locations and IDs. For full connectivity and efficient use of energy resources, another scheme named, Connectivity and Energy-Efficiency (CEE) [54] has proposed. Based on probability theory, a relationship is obtained between connectivity and communication radius. Different nodes attain wake-up and sleep based on connectivity probability. Residual energy and distance are the two parameters used for energy efficiency. To increase the life-time in terms of reducing idle listening and overhearing issues, another scheme known as, An Energy Consumption Model for Multimodal Wireless Sensor Networks based on Wake-up Radio Receivers [55], [56] is proposed. The radio wakeup (WuRx) and Low Duty Cycling (LDC) with addressing mechanism is used in estimating the energy model. The performance is checked with addressing WuRx and without addressing it. For minimizing idle listening at radio component, the radio is activated on periodic time on receiving a message. It uses M2WSNs and multi-hop communications to analyze the behavior and performance of wakeup protocol.

Another routing based scheme is based on relay selection joint consecutive packet routing (RS-CPR) [57] for energy efficiency has proposed. Each node is weighted on distance from the sink, remaining energy and number of packets in a queue. Nodes are selected from the forwarding node-set with high weight. Two thresholds are tested for each node, the packet queue length and maximum packet waiting time. Fewer nodes with data are selected as relay nodes and, as a result, minimize the contention conflict for the channel. Energy-aware scheduling with quality guarantee method (ESQG) [58] is a state-based scheme based on the degree of investigating the area and residual energy. Voronoi diagram is used for the scope of an active node and calculates importance in the form of residual energy. It uses awakening frequency and individual sensing for changing the state of a node. This scheme is useful where individual sensing is important (scattered network) but it creates overhead in calculating residual energy and multicasting that information to all neighbors.

For reducing the scheduling overhead in on-demand WSNs, overhearing (OH) based on Double Modulation (DM) [59] has proposed. Extra receiver for maintaining wakeup-sleep scheduling is used. An extra signal in the payload is broadcasted for wakeup. It minimizes the impact of sensed data packets on delay and size. This scheme combined the best features of OH and DM. Distributed Minimum-Delay Energy-efficient flooding Tree (MDET) [60] is a scheduling algorithm proposed for better scheduling and with reliable links in WSNs. Undetermined-delay-constrained minimum spanning tree is used for reducing the latency and delay causes by flooding. Very useful in NP-complete problems, but sometimes it creates extra overhead due to such a massive calculation.

Enhanced multimodal switching mechanisms for node scheduling and data gathering (ESMS) [61] is another state-based scheme. Time driven is useful where constant supervision is required while event-driven is used for the detection of specific events. Two switching mechanisms are implemented eHNS and eHNS based on state machine states. Changing of state based on outcomes of PED and PAD algorithms. Another approach for 3D heterogonous based on k-coverage probabilistic [62] has proposed. Sensing sphere of each sensor has detected and redundant points are calculated using k-coverage algorithm. For deployment of sensors, it does not need any terrestrial position.

Energy-efficient sleep scheduling mechanism with similarity measure (ESSM) [63] is another state based scheme. ESSM is based on changing state of node from Active to Sleep. The algorithm has focused on condensed sensor deployment and useful for congested WSNs. But in dispersed deployment, the correlation of data will be low and network will be used more energy. An Effective Scheduling Algorithm for Coverage Control in Underwater Acoustic Sensor Network (UASNs) [64] is state based approach for underwater acoustic sensor networks. Based on ESACC Active-Sleep strategy is deployed with redundant nodes in two portions. Memetic procedure is used for activating sleep state while some nodes will be Active state for sensing. The sleepy nodes are activated during network operations to become as Active/Live. A Method of Balanced Sleep Scheduling in Renewable Wireless Sensor Networks (BSSR) [65], [66] has proposed for introducing harvesting layer in WSNs. Energy are harvested by charging-points from harvesting to underlying layers. Energy harvesters are solar, light or electromagnetic waves. In congested network, sleep-wake up strategy has applied to preserved storage battery power. Many nodes are active for sensing and communication while redundant nodes are kept in sleep to save energy usage. Heuristic Algorithm for Clustering Hierarchy (HACH) [67] is a sleep-wake up scheduling algorithm. Many nodes remain active using stochastic selection of inactive nodes (SSIN). Boltzmann selection is used for sleeping and activating nodes by ensuring full coverage. To enhances clustering process, HEECHS algorithm is used. Advance heuristics is used global and local search process to improve quality of solution. Both clustering and sleep schedules improve optimization process and increase energy-efficiency.

Motivation: In all of the above schemes, states are changed in passive and proactive manner. They are saving energy by applying strategies on upper and lower layers. Clustering is the management of bunch of nodes at upper layer. While duty cycling is performed at lower layers. Most of the schemes used some definite criteria for changing state from active to sleep. Some dominant parameters for changing states are including: 1) Residual Energy, 2) Distances from BS/CH, 3) RSSI based sensing, 4) Periodic checking of beacons packets, 5) Broadcast packets on time intervals, 6) Location based Hello packets and many other parameters which have already been discussed in the literature. To accomplish certain criteria and meet some predefined measures, some calculation and assessment is needed. In total, these schemes create extra overhead because of changing states in an unintelligent style. A table of all schemes with merits and demerits have included in Appendix-B which mentioned main parameters and its applicability. There is no checking for contents of sensed data packets or inspection the frequency of data or analysis of repeating certain patterns. We always need a reactive mechanism which works on real-time sensing and change states according to contents/frequency/pattern of the sensed data packets. The scheme should avoid un-necessary message-forwarding (redundant sensed values) and minimizes re-clustering overhead. Here we have proposed a new scheme "A Content-based Adaptive and Dynamic Scheduling (CADS) using two-way communication model in WSNs". CADS controls the sensing capabilities by introducing state-based procedure. Four state model is derived on internal components of sensor node. The analyzer module at BS analyzes data patterns/contents/frequency and sends control message for adapting any specific state. CADS prolongs network life-time by consuming minimum energy and reduces overall network traffic.

III. CONTENT BASED ADAPTIVE AND DYNAMIC SCHEDULING (CADS) SCHEME USING TWO-WAY COMMUNICATION MODEL IN WSNs

CADS implements four state model in which each node adapts a specific state. Each state works on definite energy level. Nodes collect sensed data packets and send to BS for further analysis. Analyzer module at BS checks and analyzes the contents of sensed data packets. According to contents, analyzer module transmits different control messages for different nodes. Each node receives control message and adapts a specific state. Control messages regulate the functionality of each node while the analyzer module plays a vital role in changing state of a node.

Let sensors are deployed in a network topology ($T_{SN-work}$) such that $T_{SN-work}\ \varepsilon$ N where N=1, 2, 3, (Natural Numbers). $T_{SN-work}$ consists a number of immovable homogenous sensor nodes SN_1, SN_2, SN_3, ..., SN_k where K = 1, 2, 3,.... (Natural Numbers) Omni-directional with two-way communication with elected Cluster Heads (CH_i) and specific Base

Station (BS_i) where i = 1, 2, 3,.... (Natural Numbers). Three types of data is communicated in $T_{SN-work}$ i.e. Discovery Messages ($D_{Message}$), Sensing Data (S_{Data}) and Aggregated Frames ($A_{aggregated}$). $D_{Messages}$ are broadcast by BS to all SNs for discovering the position of SNs in $T_{SN-work}$ and for formation of clusters ($K_{Cluster}$). When $D_{Message}$ received from BS, SNs broadcast necessary information including its unique Identity (ID_k) and hop count (H_P) to reach the BS. With the help of $D_{Message}$ BS knows overall topology of network and select a suitable CH for each zone.

BS sends D_{CH} with a unique ID_{CH} to a specific SN based on degree of node and its location. The receiver node of this message now becomes a CH_i for a specific zone. BS also broadcasts a message embedded with same ID for all surrounding SNs which tells about the elected CH. Now all SNs sends its S_{Data} to that CH and CH aggregate sense data with related IDs of SNs and forward that $A_{aggregated}$ to BS. CADS works in two-way communications i.e. forward and backward communication between SNs, CH_k and BS. When SN sense an event, it sends collected packets to CH and CH aggregates and sends data packets to BS, the communication is known as Forward communication. while all Control Message ($C_{Messages}$) transmits by BS for CH and for SNs is known as Backward communication. The size of the $C_{Messages}$ is very small as compared to actual data packets. Following are changing states based on Control Messages.

$$\Delta State = \left\{ \begin{array}{c} Active - Live :: C\ Sleep - Live \\ Sleep - Live :: C\ Active - Sleep \\ Active - Sleep :: C\ Sleep - Sleep \\ Sleep - Sleep :: C\ Sleep - Live \end{array} \right\}$$

Two types of communication occur in CADS. "*Forward Communication*" is regular communication where data is sensed through leaf nodes and send to CH. While "*Backward Communication*" is newer way of communication where BS controls all activities of a node by sending control messages to leaf nodes. Analyzer module of BS analyzes and categories the contents and transmits control messages to those sensors which need to change the state.

A. CADS NETWORK MODEL

BS broadcasts $C_{Messages}$ for all SNs to broadcast their IDs and other information (remaining energy) to its surrounding nodes. When SNs received this message on its radio link, they also broadcast a message in its surrounding. Every SN received $C_{Messages}$ and transmits broadcast message. Consider the deployment of SNs in a Target Area (T_{Area}) as shown in FIGURE 1. Where SNi=1….n ϵ T SN-work with eight nodes. The CH1 is the Cluster Head for all nearby nodes with maximum dense node in the $T_{SN-work}$.

CH_1 is responsible for collection of sense data from these eight nodes, aggregates it, appends IDs of all concerned nodes and send $A_{Aggregated}$ to BS for further necessary action. BS controls the entire network by sending $C_{Messages}$ to CH_i and CH further sends to connected nodes. When $T_{SN-work}$ is formed and $C_{Messages}$ change the states of SNs, there are

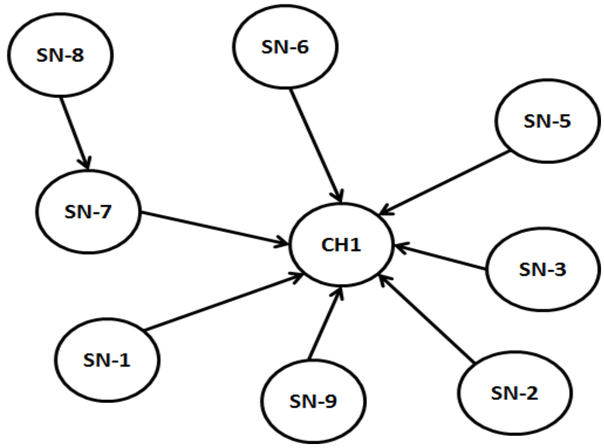

FIGURE 1. CADS network model.

approximate distribution of states transition in the $T_{SN-work}$. 25% of each state is maintain in the topology i.e. total of four states, each state maintains 25% of all 100 nodes. Sensor's components change their states from one to another (Active/Sleep), based on $C_{Messages}$ received from BS.

B. NODE'S COMPONENTS AND ITS ENERGY LEVEL

A typical sensor node consists of *micro-controller, magnetic sensor, communication link (radio link)* and a *built-in battery* (power source). *Microcontroller* is responsible for processing all onboard data calculations and aggregations. It uses operative method for minimum utilization of energy. Micro-controller is three operating states i.e. Idle, Sleep and Active. We have used only two states; either the microcontroller is in Active-state or in Sleep-state. The *Radio link* is used for communication and it has three states i.e. Idle, Sleep and Active but we have used only two Active and Sleep. The magnetic sensor has two states i.e. Active and Idle; we have used both active state and idle. In TABLE 1 [28], details of different components are shown. Each component of a node consumes a specific amount of energy at each transaction.

TABLE 1. Different states of SN & its energy consumption.

Component	State of the component	Parameters	Amount of Energy (m.A)
Micro controller	Active state	$\mu\ E_{Active}$	10
	Idle state	$\mu\ E_{Idle}$	9.5
	Sleep state	$\mu\ E_{Sleep}$	0.65
Radio link	Active state	$xb\ E_{Active}$	30
	Idle state	$xb\ E_{Idle}$	28
	Sleep state	$xb\ E_{Sleep}$	0.05
Magnetic Sensor	Active state	$\hat{s}\ E_{Active}$	0.5
	Idle state	$\hat{s}\ E_{Idle}$	0

C. CADS FOUR STATE TRANSITION MODEL

The SNs first install in target area for collecting information are in firing state. CH is any SN_i with more density as compared to other SNs in $T_{SN-work}$ as discussed in network

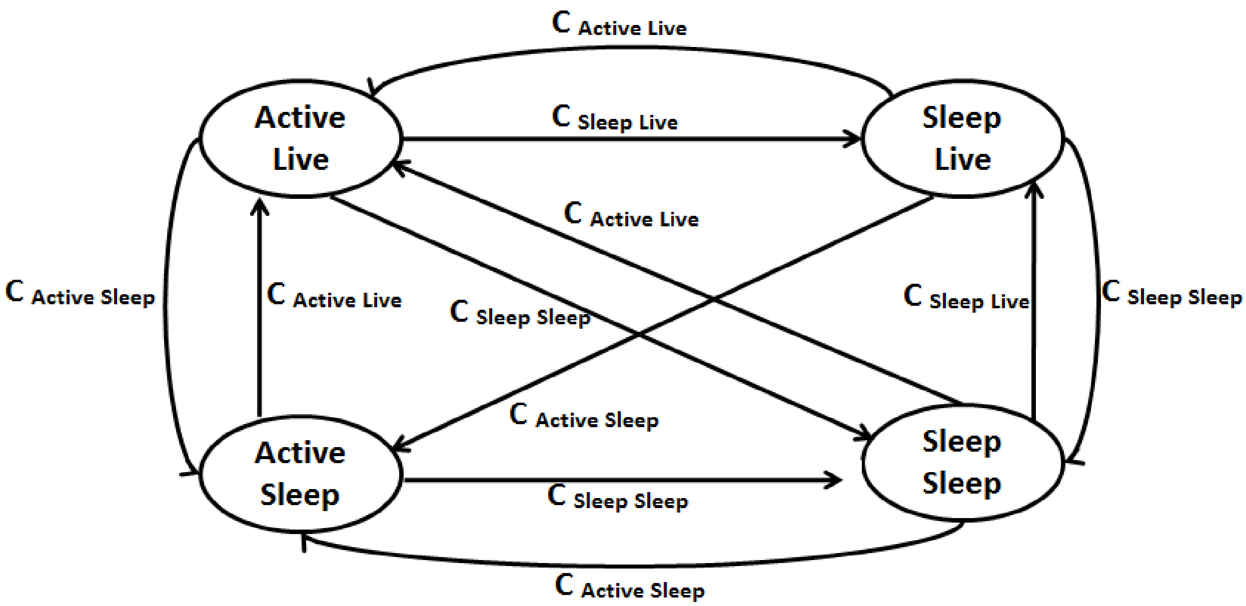

FIGURE 2. Four state transition model.

model. The states of the SNs are changing by receiving different types of C_{Messages} from BS. There are four states of any SN_i in $T_{SN-work}$ based on four C Messages. Four State Transition diagram has shown in FIGURE 2. State Transition diagram with three states is also proposed in Basic Energy Conservation Algorithm (BECA) [4] and in Geography-informed energy conservation for ad hoc routing (GAF) [68]. The analyzer module transits C_{Messages} and each node adapts the state accordingly. These states are based on the internal components of a node.

D. ENERGY STATE

Four states of CADS have defined based on the internal components. Different combination of microcontroller, communication-link and sensing components defined different states. Four states derived from internal structure of sensor has been shown in TABLE 2.

TABLE 2. Four states of CADS.

States of SNs	Energy at Sensing (§ E)	Energy at Microcontroller (µ E)	Energy at Radio link (xb E)	Total Energy at SN (m.A)
Active-Live	0.5 (m.A)	10 (m.A)	30 (m.A)	40.50
Sleep-Live	0.5 (m.A)	10 (m.A)	0.05 (m.A)	10.55
Active-Sleep	0.5 (m.A)	0.65 (m.A)	30 (m.A)	31.15
Sleep-Sleep	0.0 (m.A)	0.65 (m.A)	0.05 (m.A)	0.70

1) ACTIVE-LIVE STATE

In Active-Live state of SN, all components are Active (microcontroller + sensing + communication) with maximum consumption of energy 40.50 (m.A). Active-Live state is the initial state of a node where sensor in firing state with active all internal components. Micro-controller consumes 10 (m.A) of energy per transaction, sensing consumes 0.5 (m.A) and communication link consumes 30 (m.A) with total of 40.50 (m.A) of energy.

2) SLEEP-LIVE STATE

In Sleep-Live state, two components, microcontroller and sensing are active whereas communication is in sleep mode.

It utilizes 10.55 (m.A) energy. In this state the sensor only listens the sensing. Microcontroller consumes 10 (m.A) of energy per transaction, sensing consumed 0.5 (m.A) and communication-link consumed only 0.05 (m.A) with total of 10.55 (m.A) of energy.

3) ACTIVE-SLEEP STATE

It utilizes 10.55 (m.A) energy. In this state the sensor only listens the sensing. Microcontroller consumes 10 (m.A) of energy per transaction, sensing consumed 0.5 (m.A) and communication-link consumed only 0.05 (m.A) with total of 10.55 (m.A) of energy.

4) SLEEP-SLEEP STATE

In Sleep-Sleep state, all component (microcontroller + sensing + communication) are in sleep state, with total energy utilization of 0.70 (m.A). This state is hibernating state where all components are in sleep. Microcontroller consumes 0.65 (m.A) of energy per transaction, sensing consumes 0.0 (m.A) and communication link consumes only 0.05 (m.A) with total of 0.70 (m.A) of energy.

Initially all nodes are in Active-Live state. After a number of transactions, if same values are sensed consecutive times by a node and these redundant values are sent to analyzer module. It analyzes the contents and transmits different C_{Messages}. When analyzer discovers similarities in sensed data packets or observes same pattern of data for consecutive sensing cycles, it transmits C_{Messages} in backward direction.

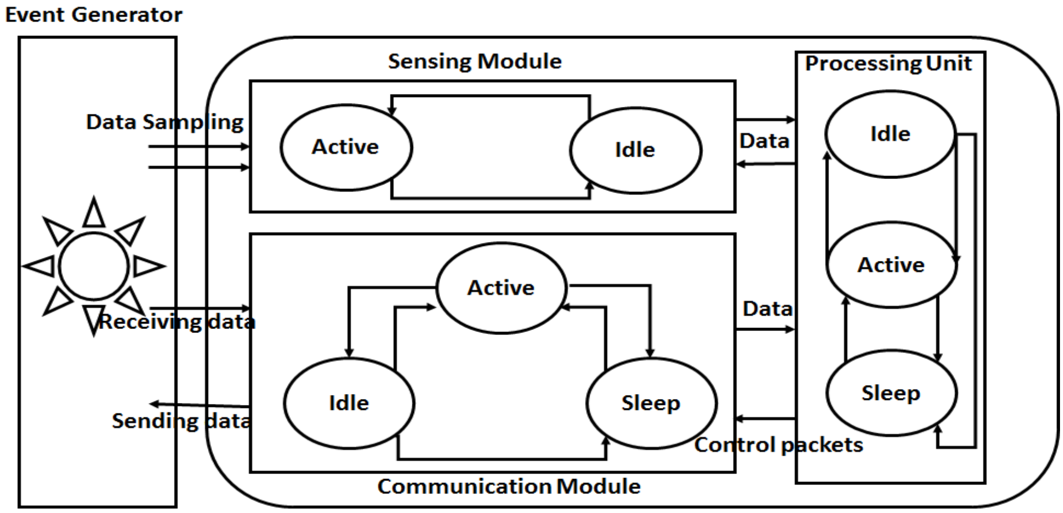

FIGURE 3. Overall Sensor Model for activating different components.

Normally, analyzer module sends $C_{Messages}$ in a sequence. The node state is switched from Active-Live to Sleep-Live. In Sleep-Live, a node is only Listening the sense values and microcontroller make aggregation of packets only. The communication is in sleep-state where no communication take place.

Based on these values, node state is changed to Sleep-Sleep if these values remain persistent for last three times. Sleep-Sleep state is a condition in which node consumes minimum energy. Active-Sleep is rare-time state which are used for maintaining communication paths and is useful only for routing nodes. Routing nodes are path establishing nodes which take part in communication only but have no concerned with assessment of values.

All other schemes have also defined different states as discussed in literature review but in CADS, states of node are defined based on actual amount of energy consumed by each unit. Other schemes passively sense the phenomena and have no comparisons between data. But in CADS, contents of sensed data packets are checked and based on the assessment, states of nodes are changed.

E. MODULE BASED EVENT DRIVEN MODEL

CADS works in different operating phases where analyzer module streamlines all nodes in specific states. Four states model categorizes nodes in four operating states based on activating different components inside a sensor node. When an event occurs, sensing module senses the event, it proceeds that information to microprocessor. Microprocessor aggregates these packets, append identities and forwards that information to communication module to send it to CH or BS. Each component in a node have different operating states, depends upon the state of the node, these modules can transit from one to another. Sensing module have two states (Active/Idle), processing module have three states (Idle/Active/Sleep) and communication module have also three operating states (Active/Idle/Sleep). These components are actually utilizing certain amount of energy when performing their tasks. Active state always consumed more energy while idle-state have moderate consumption of energy and sleep have its minimum level of energy. In CADS, the analyzer module changes these components from one to another (Active → Idle → Sleep) for different energy levels. Here different energy level means different types of tasks performing by these components. There are sixteen possible cases when these internal component are combined but we have only derived four state. These four state are useful states which helps in reducing energy consumption and avoids to missed any important event. In TABLE 3, CADS states for a node by combining these components have been shown and components-based activities of CADS are shown in FIGURE 3.

TABLE 3. Components states of SN at different operations.

State of Node	Sensing	Microcontroller	Radio link
Active-Live	Active	Active	Active
Sleep-Live	Active	Active	Sleep
Active-Sleep	Active	Sleep	Active
Sleep-Sleep	Sleep	Sleep	Sleep

F. ANALYZER MODULE

The sensed data packets are sent to analyzer unit of BS for further assessment and necessary action. Analyzer module investigates the contents of data and takes necessary decision for a node to obtained different states. This module is the important part and sensing are minimized due to transmitting $C_{Messages}$. The contents of received data packets are examined and keeps it in buffer. If consecutive three values remain the same or if some pattern is repeated, the analyzer transmits

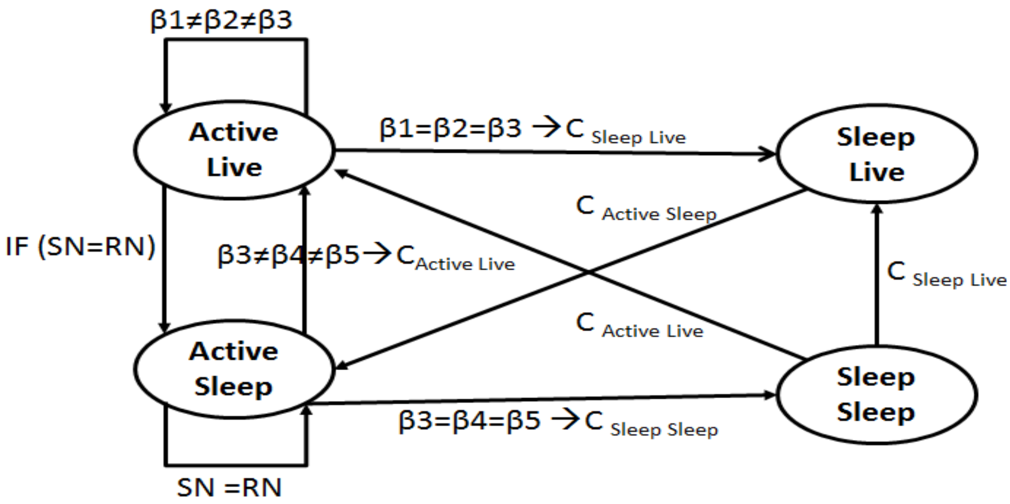

FIGURE 4. Analyzer module at base station.

different $C_{Messages}$ to different nodes while maintaining the overall network in active state. In FIGURE 4, the analyzer module has shown, in which different $C_{Messages}$ are transmitted for switching states. When sensed data packets are received for the first time (T1), the analyzer checks the contents and retain one copy/signature in buffer. In second time (T2) of receiving data, analyzer checks the contents again and retain one copy in buffer. In same way, third times (T3), contents are checked and examined for similarities. If consecutive three times, contents remain the same then state is changed to another i.e. $\beta 1 = \beta 2 = \beta 3$, where β n is the contents of data. The analyzer module repeats the sensing cycle for each state, consecutive three times (T4, T5, T6) and state change accordingly. The analyzer module also enforces strategy for routing nodes. Any node which is path established-node will never adapt the sleep state. Acquiring sleep state for routing node, will disconnect certain node and results as network partitions. In same figure, an analyzer module has been shown in which β n are compared and different, $C_{(messages)}$ are transmitted while RN is routing nodes

G. COMMUNICATION MODELS

For mathematical model, $T_{SN-work}$ is constructed including seven nodes namely SN1, SN2, SN3, SN5, SN6, SN7 and CH1 as in FIGURE 2. These nodes sense the events and sends the sensing data packets (S_{Data}) to CH1, where it aggregates S_{Data} into aggregated frames ($A_{aggregated}$) and sends to BS. For first time, these nodes in firing state with full active mode. When they receive $C_{Messages}$ from the BS it triggers itself according to the message and configure their states. After selecting any node as CH_i based on $D_{Messages}$ in CH selection process, the BS now knows the overall network $T_{SN-work.}$ The BS locates actual position of CH and location of all SNs. For the first time when SNs are deployed, the $D_{Messages}$ are broadcast by BS. i.e.

$$BS \rightarrow SN_i\{SN_1, SN_2, SN_3, \ldots, SN_{k-1,} SN_{k,} SN_{k+1}\} \quad (1)$$

1) FORWARD COMMUNICATION

Forward communication is regular communication of data packets which are transmitted from SNs to analyzer module of BS. This communication starts with first time sensing and ends when sensed data packets is reached to BS. Node deployed in $T_{SN-Work}$ is in firing state, where all components are active of node. In firing state, all the SNs respond to this message on time slot T_t where "t" is interval t=1,2, 3…., n. SNs advertises message to its surrounding for other SNs to know its location.

$$SN_i \rightarrow SN_j\{SN_1, SN_2, SN_3, \ldots, SN_{k-1,} SN_{k,} SN_{k+1}\} \quad (2)$$

SNs also send their identities (ID_i) to CH_i. CH forwards these ID_i to BS for recorded in network database. With the help of these IDi, the overall network structure are mapped at BS. FIGURE 4, showing the S_{Sense} in the $T_{SN-Work}$ from SN_8 to SN_7 and then to CH1.

$$ID_1 + ID_2 + \cdots + ID_{k-1} + ID_k + ID_{k+1} \rightarrow CH_j \quad (3)$$

When these SNs responds for the first time to the CH, it aggregates the packets into a single packet, appends the ID_i of these SNs and sends to BS. The SNs in this stage is in Active-Live state with maximum utilization of energy (40.50 mA) where all components of SN are in active mode. After sensing in the T_{Area}, the SNs senses the data packets uniformly and sends the D_{Sense} to CH i.e.

$$ID_1\beta 1 + ID_2\beta 1 + ID_3\beta 1 + \cdots + ID_k\beta 1 \rightarrow CH_j \quad (4)$$

Where $\beta 1$ is the D_{Sense} is sensed uniformly in T_{Area}. All SNs send the D_{Sense} to CH. The CH aggregates the same data into a single packet with one copy of D_{Sense} and ID_i of all participating nodes. For time being the data is same ($\beta 1$).

$$SN1\beta 1+SN2\beta 1+SN3\beta 1+SN5\beta 1+SN6\beta 1+SN7\beta 1 + SN8\beta 1+SN9\beta 1 == \sum SN(1\ldots 9)[CH1+\beta 1] \quad (5)$$

The CH appends its ID_{CH-1} to the packet and sends to BS for further necessary action.

$$CH1 + \beta 1 \rightarrow BS \quad (6)$$

After some interval of time when system re-sense certain event with same data packets, the S_{sense} is collected at CH1 and sends to BS and this cycle is repeated. After receiving data packets, the analyzer module examines the contents of data packets and take necessary decision. It is also possible that after sometime S_{sense} changed ($\beta 2$) but sense same data in over $T_{SN-Work}$, i.e.

$$SN1\beta 2 + SN2\beta 2 + SN3\beta 2 + SN5\beta 2 + SN6\beta 2 + SN7\beta 2 + SN8\beta 2+SN9\beta 2 == \beta 2 \sum_{(1\ldots 9)} SnSN == CH1+\beta 2 \quad (7)$$

It is also possible that sensed data is varied and diverse. For example, two different data are sensed in $T_{SN-Work}$. i.e. $\beta 1$ and $\beta 2$. CH categorizes the events and arranges the S_{Sense} according to its contents. The procedure is repeated with categorizing in different segments i.e.

$$SN1\beta 2 + SN2\beta 1 + SN3\beta 1 + SN5\beta 2 + SN6\beta 1 + SN7\beta 2 + SN8\beta 1 + SN9\beta 1 == \beta 1 \sum(SN2, SN3, SN6, SN8, SN9)||\beta 2 \sum(SN1, SN5, SN7) = CH1 + \beta 1 + \beta 2 \quad (8)$$

where || is used for concatenation.

2) BACKWARD COMMUNICATION

Then analyzer module responds with different $C_{Messages}$ in the form of *backward communication.* These messages also include the first message transmits by BS for CH selection process as in equation-(1). $C_{Messages}$ are transmitted in response of sensed data packets received at BS [50]. The analyzer unit checks the contents of S_{Sense} and responds to nodes in the form of different $C_{Messages}$. When analyzer receives same data again and again from $T_{SN-Work}$, the analyzer transmits C Messages to change the states of the SNs. Different types of C Messages in backward communication are shown in FIGURE 2 and in FIGURE 4. First, the SNs are in Active-Live state, it senses data and transmits to CH and onward to analyzer unit. If same data is received consecutive three times, $C_{Sleep-Live}$ is transmitted and node change state from Active-Live to Sleep-Live. In some situations, there are rapid variations in sensed data, the Active-Live will remains the same (state is not changing). In Sleep-Live, communication is in sleep while processor and sensing is active. The analyzer transmits $C_{Active-Sleep}$ to node, it switches to Active-Sleep. In this state, processor is in sleep while communication and sensing is active. When analyzer module receives data with variations, it switched to Active-Live from Sleep-Live. While for consistent and same pattern data, $C_{Sleep-Sleep}$ is transmitted. When a node acquires the Sleep-Sleep state, all the components are in sleep mode. For routing node, the Active-Sleep is defined and these nodes will never have switched to Sleep-Sleep state. In Sleep-Sleep state, when node receives $C_{Active-Live}$, the node is switched to Active state. The analyzer module checks the contents, frequency or pattern and responds accordingly. The analyzer responds in form of $C_{message}$. i.e.

$$BS \rightarrow SN_i \quad (9)$$

H. MEDIUM ACCESS IN CADS

In CADS, TDMA is used in synchronous form with adoptive cycling of Sensor-MAC (S-MAC) protocol [69]. S-MAC is RTC-CTS based MAC protocol, used for fixed time-slots. It works in three modes, adaptive listening, periodic sleeping and virtual clustering. Here adaptive sleeping is an ideal method of availing the channel. Adaptive sleeping reduces sleeping delay and minimize latency at node level. Fixed slots alternative scheduling mechanism has used when the BS creates S-MAC for equal time slot for each node. One-time slot is assigned to two nodes at the same time. If one node is in Sleep-Sleep state and not in a position to transmit data, then another node transmits its sensed value. Therefore, one slot is alternatively used by others. The same slot can be used for more than two nodes as well but here we have implement it only for two nodes as in TABLE 4 has shown [70]. T1 slot is assigned to SN1 and SN8 at same time if SN1 is in Sleep state then SN8 will be use the same T1 slot. In the same way in T2 slot, SN2 and SN7 can use the same slot alternatively.

TABLE 4. S-MAC slot > 1 Nodes.

SMAC T1	T2	T3	T4	T5	T6	T7	T8
SN1	SN2	SN3	SN4	SN5	SN6	SN7	SN8
SN8	SN7	SN6	SN5	SN4	SN3	SN2	SN1

I. CADS ALGORITHM

CADS *[SN_{i-j} ID_{i-j}, CH_{i-j}, $C_{Messages}$, $L_{Message}$, D_{Sense}, $A_{Aggregated}$, States (Active-Live, Sleep-Live, Active-Sleep, Sleep-Sleep)].*

1: Base Station (BS) broadcast a $C_{Message}$ to all sensor nodes (SNs) to form a network and make selection for Cluster Heads (CH_i).

BS *broadcasts* $C_{Message}$ *(Hello + IDs)*

For *each* SN_i *to* SN_j

2: SNs broadcast $L_{Message}$ its IDs and Hello packets in response of $C_{Message}$.

SNs *broadcast L (Hello + IDs)*
For *each SN_i to SN_j*
3: BS transmits a $C_{Message}$(SNi = CHi) for a specific node as CH (CH election) on certain criteria.
BS *broadcasts $C_{Message}$ (SNi = CHi)*
if *density of SN_i> density of SN_j*
Set *SN_i is CH_i*
3: Now all SNs send their IDs to CH_i
For *each SN_i to SN_j*
Send IDs to CHi
4: The CH collects data from all SNs.
CH receives D_{Sense}
For *each SN_i to SN_j*
5: CHi aggregates the sensed data packets into aggregated Frames $A_{Aggregated}$
For *each $D_{Sense}(\beta n)$*
Receives SN_i to SN_j
$A_{Aggregated}(i)$ ++
6: CHi appends the IDs of all SNs and its own ID before transmitting to BS.
$L_{Messge} = \sum A_{Aggregated}(i) \,||\, \sum ID_{(i\ to\ j)} \,||\, ID_{CH}$
$L_{Message}$ *sends to BS*
7: Analyzer module at BS checks the contents (βn) of received data packets for similarities, either to receive same data or different
If *($ID_i||\beta_i = ID_j|| \beta_j$)*
For *each $D_{Sense}(\beta n)$*
8: According to the contents (βn) of sensed data received at analyzer, different $C_{Message}$ are transmitted to SNs.
For *each $L_{Message}$*
BS sends $C_{Message}$
9: Checking the contents at consecutive three sensing cycles (T_1, T_2, T_3) and the current state is Active-Live
If($ID_i||\beta_1 = ID_i|| \beta_2 = ID_i||\beta_3$) **&&** *State = Active-live*
For *each SN_i to SN_j*
BS transmits $C_{Sleep-Live}$ to SN_{i-j}
10: If the contents are different at the sense at same time (T_1, T_2, T_3) (no change in state) then
Else($ID_i||\beta_1 \neq ID_i\ ||\beta_2 \neq ID_i\ ||\beta_3$) **&&** *State = Active-live*
For *each SN_i to SN_j*
BS transmits $C_{Active-Live}$ to SN_{i-j}
11: Checking the contents after forth cycle (T_4, T_5, T_6),
If($ID_i||\beta_3 = ID_i|| \beta_4 = ID_i||\beta_5$) **&&** *State = Sleep-live*
For *each SN_i to SN_j*
BS transmits $C_{Active-Sleep}$ to SN_{i-j}
12: Checking the contents after seventh cycle (T_7, T_8, T_9),
If($ID_i||\beta_5 = ID_i|| \beta_6 = ID_i||\beta_7$) **&&** *State = Active-Sleep*
For *each SN_i to SN_j*
BS transmits $C_{Sleep-Sleep}$ to SN_{i-j}
13: When the contents are different at these cycles (T_7, T_8, T_9),
Else($ID_i||\beta_5 \neq ID_i\ ||\beta_6 \neq ID_i||\beta_7$) **&&** *State = Active-Sleep*
For *each SN_i to SN_j*
BS transmits $C_{Active-Live}$ to SN_{i-j}
14: Checking the node status for routing,
if *SN_i = Routing-Node*
For *each SN_i to SN_j*
BS transmits $C_{Active-Sleep}$ to SN_{i-j}
15: Repeat if needed
Go To Step-5
Else Exit
13: *End If*
End If
End

J. RADIO MODEL

For implementing of CADS, here we have used first order energy model. First Order Energy Model is used in many other benchmark schemes [21], [25], [71]–[76] but here we have assumed the same model with necessary changes and many derived notations. The amount of energy used in transmission or receiving is simply (Radio dissipation) as follows in equation (10)

$$E_{(Transmit)} = E_{(Receive)} = E_{(Electric)} = 50 \text{ nJ/bit}, \quad (10)$$

This amount is used to run an electric circuit for receiving and transmitting while for permittivity in free space the amount is used as

$$\varepsilon_{amp} = 100 \text{ J/bit/m}^2 \quad (11)$$

For transmitting "L" number of bits to "D" distance we can write it as

$$E_{(Transmit)}(L, D) = E_{(Transmit)}L + E_{(Transmit)}\varepsilon(L, D)$$
$$E_{(Transmit)}(L, D) = E_{(Electric)}{}^*L + \varepsilon_{amp}{}^*L^*D^2. \quad (12)$$

For receiving "L" length of message

$$E_{(Receive)}(L) = E_{(Receive)}{}^*L. \quad (13)$$

After applying the concerned amount of each parameter, we have known that these processes consumed more energy. To reduce the energy consumption, we should have minimized these messages. For intermediate nodes routing overhead is covered as

$$E_{(Transmit)}(L, D = D_{SN1\to SN2}) + E_{(Transmit)}(L, D = D_{SN2\to SN3}) < E_{(Transmit)}(L, D = D_{SN1\to SN3}). \quad (14)$$

$$D^2_{(SN1\to SN2)} + D^2_{(SN2\to SN3)} < D^2_{(SN1\to SN3)} \quad (15)$$

In DTMA based scheduling techniques, each SN transmits at constant rate. The intermediate node transmitting is more reasonable as direct communication for "n" sensors at "d" equal distance, we can express it as

$$E_{(Direct)} = E_{(Transmit)}(L, D = n^*d) = E_{(Electric)}{}^*L + \varepsilon_{amp}{}^*L^*(nd)^2$$
$$E_{(Direct)} = L(E_{(Electric)} + \varepsilon_{amp}{}^*L^*n^{2*}d^2) \quad (16)$$

Now in this way each node will transmit data for its near node and each node will receive "n-1" data at a distance of "nd", we can express it as

$$\begin{aligned} E_{(Electric)} &= n^{*}E_{(Transmit)}(L, D = n^{*}d) + (n-1) \\ &\quad {}^{*}E_{(Receive)}(L) \\ E_{(Electric)} &= n(E_{(Electric)}{}^{*}L + \varepsilon_{amp}{}^{*}L^{*}d^{2}) + (n-1) \\ &\quad {}^{*}E_{(Electric)}{}^{*}L \\ E_{(Electric)} &= L((2n-1)E_{(Electric)} + \varepsilon_{amp}{}^{*}nd^{2}) \end{aligned} \tag{17}$$

In Hierarchical cluster formation, first CH is randomly selected while after ten rounds CH is reselected on bases on remaining energy. A threshold has been defined and calculated after hundred rounds. This CH selection is initial and after checking the contents and state-based policy, this threshold is calculated after greater number of rounds. Threshold is defined as

$$\begin{aligned} &T_{(Threshold)} \\ &= PROB/[1 - PROB^{*}(d \bmod (1/PROB))] \\ &\quad \times [R^{*}PROB + 1 - R^{*}PROB]E_{(Residual)}/E_{(Supply)} \\ &T_{(Threshold)} \\ &= 0; \quad \text{otherwise.} \end{aligned} \tag{18}$$

While distance and remaining energy is related in

$$\varepsilon = E_{(Residual)}/D_{S\rightarrow BS} \tag{19}$$

For inter and intra cluster communication, the energy usage is

$$\begin{aligned} E_{(Transmit)}(L, D) &= L^{*}E_{(Electric)} + E_{(Final)}{}^{*}D^{2} \\ &\quad {}^{*}L \leftrightarrow D <= D_{(Threshold)} \\ E_{(Transmit)}(L, D) &= L^{*}E_{(Electric)} + E_{(Final)}{}^{*}D^{2} \\ &\quad {}^{*}L \leftrightarrow D >= D_{(Threshold)}. \end{aligned} \tag{20}$$

For CADS, four states model is defined and implemented in first order radio model. The equations (11) and (12) are redesigned and the distances are checked for $D_{(Threshold)}$. The energy is now calculated normal with a factor of any defined state. There are eight possible conditions, four for inter-cluster and four for intra-cluster communication. In TABLE 5, four states with eight conditions have shown.

While other metrics will remain the same,

$$E_{(Receive)}(L) = L^{*}E_{(Electric)} \tag{21}$$

While $D_{(Threshold)}$ is calculated as;

$$D_{(Threshold)} = \sqrt{}[E_{(Final)}/\varepsilon_{amp} \tag{22}$$

For each node energy consumption will be

$$E_{(Residual)} = E_{(Supply)} - (E_{(Transmit)}(L, D) + E_{(Receive)}(L)). \tag{23}$$

Now total energy of the Network will be

$$E_{(Total)} = \sum n(i, j)(E_{(Supply)}) \tag{24}$$

TABLE 5. Four states and threshold.

Condition	State	Metrics used	State in energy
$D >= D_{(Threshold)}$	Active-Live	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Active-Live	:: Active-Live = 40.50 (m.A)
$D >= D_{(Threshold)}$	Sleep-Live	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Sleep-Live	:: Sleep-Live = 10.55 (m.A)
$D >= D_{(Threshold)}$	Active-Sleep	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Active-Sleep	::Active-Sleep = 31.15 (m.A)
$D >= D_{(Threshold)}$	Sleep-Sleep	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Sleep-Sleep	:: Sleep-Sleep = 0.70 (m.A)
$D <= D_{(Threshold)}$	Active-Live	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Active-Live	:: Active-Live = 40.50 (m.A)
$D <= D_{(Threshold)}$	Sleep-Live	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Sleep-Live	:: Sleep-Live = 10.55 (m.A)
$D <= D_{(Threshold)}$	Active-Sleep	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Active-Sleep	::Active-Sleep = 31.15 (m.A)
$D <= D_{(Threshold)}$	Sleep-Sleep	$\Rightarrow L * E_{(Electric)} + E_{(Final)} * D^2 * L +$ Sleep-Sleep	:: Sleep-Sleep = 0.70 (m.A)

While average energy will be

$$E_{(Average)} = \sum n_{(i,j)}E_{(Residual)}/n \tag{25}$$

And for each round, we can calculate energy as

$$\begin{aligned} E_{(Round)} &= L_{(total)}[(\sum n_{(i,j)}E_{(Aggregated)}) \\ &\quad + (2\sum n_{(i,j)}E_{(Electirc)}) + (\sum n_{(i,j)})\varepsilon_{amp}{}^{*}D^{2}) \\ &\quad + \varepsilon_{amp}{}^{*}D^{4})] \end{aligned} \tag{26}$$

In CADS the value of L=1000 bits, D will be calculated dynamically and the value of $E_{(Electric)}$ will be calculated form predefined values of $E_{(Transmit)}$ and $E_{(Receive)}$.

K. COMPLEXITY OF CADS

Lemma 1: In CADS, the algorithmic complexity of each node is *O(n)*.

Proof: CADS processes each node SN_i, in each sensing cycle for checking the contents of sensed data packets. Processing every node for changing the state of a node need "n" number of iterations and the worst case is the "n". The analyzer module (BS) elects any SN as CH among all SN_i with complexity *O (1)* and each node except the CH is now working as Active-Live nodes, with (n-1) worse complexity. Accordingly, the Algorithmic Complexity of CADS is *O(n)*.

Lemma 2: Average Time complexity of CADS is *O[log(n)]*.

Proof: CADS starts functioning when first time SNs are deployed. BS broadcast a $C_{Message}$ to send position and IDs of each node to BS and neighbors. The BS elects the suitable CH and after that CH communicates with BS with time complexity *O*(*n*). After a number of sensing cycle. The analyzer module transmits different $C_{Message}$ to change the state of each SN. After applying the proportion ratio, nearly 25% of

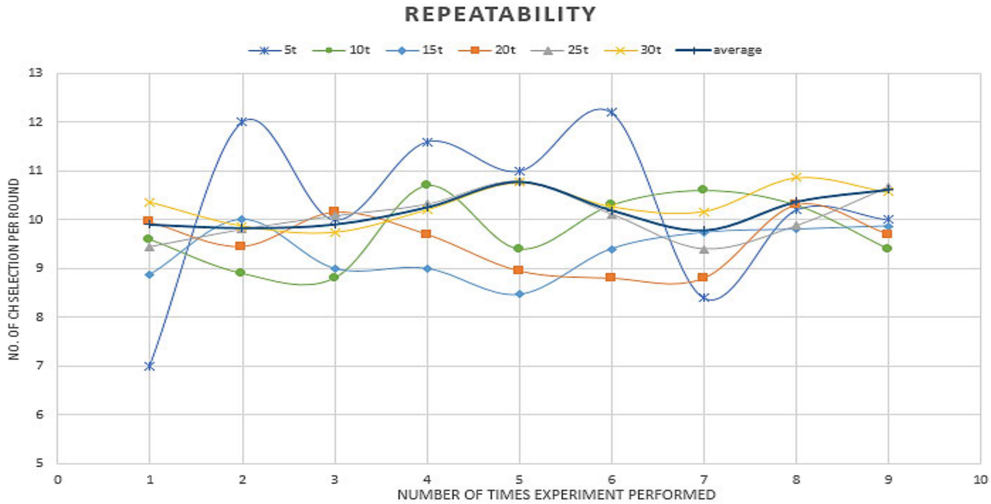

FIGURE 5. Repeatability of experiments where 30t is closer to the average.

each state is utilized. It means that Active-Live SN = 25%, Sleep-Live = 25%, Active-Sleep = 25% and Sleep-Sleep = 25% of the total. This forward communication with variation is time complexity like a binary search complexity [77], [78] and calculated as *O[log(n)]*. Complexity of CADS is calculated in both direction forward communication, from SN to BS and backward communication from BS to SN.

IV. SIMULATION SETUP FOR CADS

CADS is evaluated with different parameters and its performance is checked in different scenarios. The contents-based strategy, enforces nodes to adapt a specific energy level. Defining different energy levels, not only improves the energy-efficiency but also prolongs the network life-time.

A. SIMULATION PARAMETERS AND NETWORK SETUP

CADS four state model has implemented in MATLAB (V.R 2018a) using parameters set of "off-shelf" product "CC2420" [79]. Network topology has established with 100 to 300 nodes with area of 200 x 200 with variations and with initial energy of 0.5 joules for each node. Broadcast packet size is 512 bits ($C_{Message}$) while actual data packet (L) size is 1000 bits. After applying different operations, energy consumption is calculated at each point. Contents of sensed data packets are compared at analyzer and subsequently it regulates the functionality of each node which enhanced the functionality of the system. For uniform analysis, the experiments repeatability was checked. The experiments were repeated from 5 times to 30 times. After 25 times, the values become uniform and causes very minor deviation from its means value. This argument is verified by the minimization in the difference in the error of the consecutive values of the number of times the experiments were repeated as shown FIGURE 6 and TABLE 7, where X represent the former and Y the latter value in the sequence (e.g. 5t-10t). Hence, we believe that the expected results will be more stable. However, to be on safe-side, repeatability for each experiment was taken as 30 times. The result of these experiments are given in FIGURE 5. Different metrics, symbolic presentations and concerned values have been shown in TABLE 6 for simulation and these values are derived from product CC2420.

TABLE 6. Simulation Metrics, symbolic representation and values.

Simulation Parameters	Variable Used	Mathematical Values
Area of Simulation	A x A	200 x 200 m^2 (Maximum)
Number of Nodes	N	100-300
Initial Energy	E_o	0.5 Joules per node
Number of Rounds	Rmax	1000
Probability of CH selection	P	0.1
Transmitter energy per transaction	ETX	19.7 Ma
Receiving energy per transaction	ERX	17.4 Ma
Data buffering at transmission side	BTX	128 byte
Data buffering at Receiving side	BRX	128 byte
Date Packets	L	1000 Bits
Broadcast Packets	Cp	512 Bits
Energy per transaction at Sensing	ŝ E_{Active}	0.5 m.A
Energy per transaction at Microcontroller in Active	µ E_{ctive}	10
Energy per transaction at Microcontroller at Sleep	µ E_{Sleep}	0.65
Energy per transaction at Radio link at Active	xb E_{Active}	30
Energy per transaction at Radio link at Sleep	xb E_{Sleep}	0.05
Repeatability of experiment	--	30

B. NETWORK TOPOLOGY

For experiments, CADS has implemented in various topological forms. The experiments are started from 100 nodes and expended to 150 and 300 nodes. CADS performed different from network to network. Different metrics values have obtained in scattered and congested networks. Various values on different combination of SNs depicts the behavior

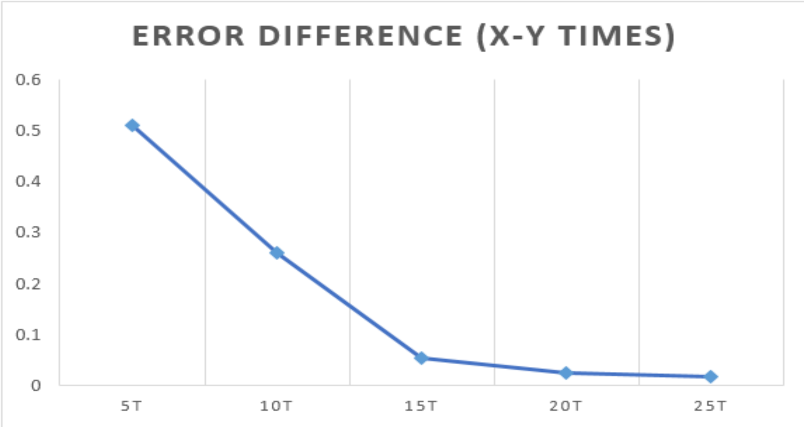

FIGURE 6. Error difference (stable results in 25 times).

TABLE 7. Difference of Errors in 5, 10, 15, 20 and 25 times.

Rounds	5times	10 times	15 times	20 times	25 times
Standard Average Error	0.510163	0.24911	0.195113	0.170652	0.152169
Error difference (X-Y) times)	---	0.261054	0.053996	0.024461	0.018483

of CADS. Based on network requirements, it works better in congested and scattered network. Network topology is established in Simulink (MATLAB) in same area of 100×100 m^2 with 100, 150 and 300 nodes. This implementation is shown in FIGURE 7. The number of CH and cluster size is varying from network to network. (a) 100 nodes network is scattered network with 0.1 CH probability. Cluster size is larger and fewer nodes covered all area. In scattered network, fewer nodes are in Sleep-Sleep state while maximum nodes are in Active-Live state. This is because CADS are trying to never miss any crucial event. (b) 150 nodes' network is moderate network where CH and cluster size is large and same area is covered by more nodes. Here probability of CH is same as 0.1 but large than 100 nodes. (c) 300 nodes, size of network is same while cluster size is small. Changing state between different nodes are more exercised here in 300 nodes with same probability of 0.1 CH selection.

C. DATA SET USED IN CADS

CADS scheme is tested with *Live Nodes, Node state position* and *Number of CH selection*, however, we would like to explain the experimental setup to justify our analysis. For the analysis of CADS, a standard and widely used dataset [80]–[83] is obtained from Intel Lab experimental setup [84] as shown in FIGURE 8. In this experimental setup 54 SNs nodes were deployed to monitor and cover the entire building. The data aggregated from the deployed sensors was stored in a file. However, tailored to our needs, we only used the temperature data/values. Different sensed values were obtained on periodic basis having repetition in sensed data packets. Data collection in the WSN is a continuous process and many sensing cycles are executed in one hour which produce a large file. However, the dataset shown in TABLE 8, a sample of data for 10 nodes obtained from the original file produced. The experiment spanned over 12 hours and

TABLE 8. Data set obtained from sensor temperature at different time.

Date	Time	Epoch	Mote-id	Temperature
2019-10-27	11:23:51.454312	18	10	21.1452
2019-10-27	11:09:09.812628	22	08	21.1631
2019-10-27	10:51:22.112821	23	03	21.8934
2019-10-27	10:21:42.121880	24	02	21.1612
2019-10-27	10:03:51.002347	26	06	21.1612
2019-10-27	09:37:51.002347	27	04	21.1612
2019-10-27	09:23:17.566742	29	05	21.0912
2019-10-27	09:11:39.891231	30	01	21.1652
2019-10-27	08:02:03.128932	31	10	21.2341
2019-10-27	08:12:22.454312	32	08	21.1452
2019-10-27	08:01:23.899928	33	03	21.1631
2019-10-27	07:57:34.280891	34	02	21.8431
2019-10-27	07:21:01.952817	36	06	21.1623
2019-10-27	06:03:34.709901	38	04	21.1645
2019-10-27	05:50:19.988921	39	05	21.1655
2019-10-27	05:18:23.892311	40	01	21.1652
2019-10-27	05:02:55.128932	41	01	21.3411
2019-10-27	04:33:17.458112	42	03	21.1112
2019-10-27	04:27:23.878319	43	10	21.1631
2019-10-27	04:19:34.080991	44	09	21.8744
2019-10-27	04:05:01.109897	45	08	21.3323
2019-10-27	04:03:14.098921	46	02	21.3322
2019-10-27	03:50:29.910923	47	04	21.3321
2019-10-27	03:45:31.109822	48	07	21.1652
2019-10-27	03:43:55.892221	49	05	21.6722
2019-10-27	03:38:26.223456	50	06	21.1452
2019-10-27	03:31:45.810009	52	10	21.1631
2019-10-27	03:19:34.563121	53	08	21.8912
2019-10-27	03:05:34.189767	54	02	21.1612
2019-10-27	03:03:45.985191	55	04	21.1612
2019-10-27	03:01:29.935311	57	07	21.1612

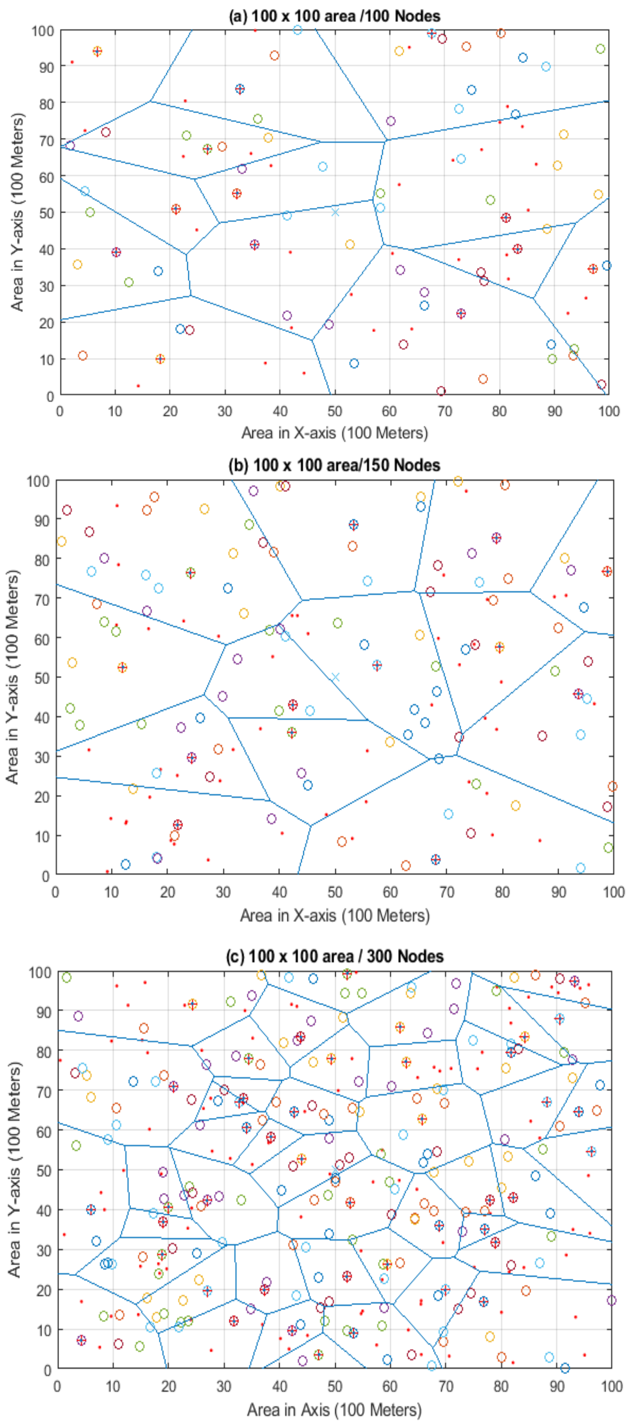

FIGURE 7. Topology of CADS (a) 100 Nodes, (b) 150 Nodes, (c) 300 Nodes.

frequent data collection is ignored. Random values are collected and compared. These values are imported to CADS model for broadcasting different control messages and changing states of the nodes. The analyzer module observes repetitions of values, and broadcasts control messages accordingly. The dataset was obtained from experiments that were performed in the Lab in a closed-environment and indoor scenario. Hence, we believe, that the circumstances parameters will have a minimum effect on the results. This argument is further supported by the already performed experiment that in a closed environment the environmental parameters have a least effect on temperature values [85]–[87]. Many classes have defined in implementing CADS in real scenario. Each class has their own methods and variables bound by object of the classes. SNs is abstract class while four derived classes (Active-Live, Sleep-live, Active-Sleep, Sleep-Sleep) are derived from it. Three methods (sense (), Process (), communicate ()) is defined inside each class with other variables. Class diagram of proposed scheme has included in Appendix-A at end of the paper.

V. COMPARISON OF CADS WITH OTHER SCHEMES

System performance of CADS has experimented with other recent and benchmark schemes including "*An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks*" (ESSM) [63], "*A Method of Balanced Sleep Scheduling in Renewable Wireless Sensor Networks*" (BSSR) [65], [66], "*An Effective Scheduling Algorithm for Coverage Control in Underwater Acoustic Sensor Network*" (UASNs) [64], "*HACH: Heuristic Algorithm for Clustering Hierarchy protocol in wireless sensor networks*" [67]. These benchmark schemes have their own state-level implementation while in CADS, an analyzer module have implemented. The analyzer checks the sending packets for similarities. The analyzer module plays a vital role in changing state of a node. These experiments demonstrate that CADS is more energy-efficient and the results has been shown the prolonging of network life-time.

For checking the performance and observing network behavior, three most important metrics are used. These parameters are *Live-Node*, *status of each sensor's component* and number of *CH selection* procedure. For each experiment, these parameters are mapped in graphs and analyzed for different observations. These parameters satisfy our network's requirements which includes no-disconnections in network, full coverage, long-live clusters and certainly not missing an event.

We have experimented with standard metrics because these values are: (1) practically applied for implementation, (2) due to its low cost in-terms of complexity and reliability, (3) high acceptance in WSNs and (4) for formal validity of such schemes which working in energy-efficiency. We have used these parameters for high precision and without biasness. CADS is fixed with these parameters for manipulating performance in congested and scattered networks. We have tried to find out the linear and non-linear effect of these parameters on CADS performance. We have tried to find-out how these parameters affect performance on larger and smaller values. In fact, we have used and fixed a response surface methodology [88] for setting different parameters for WSN.

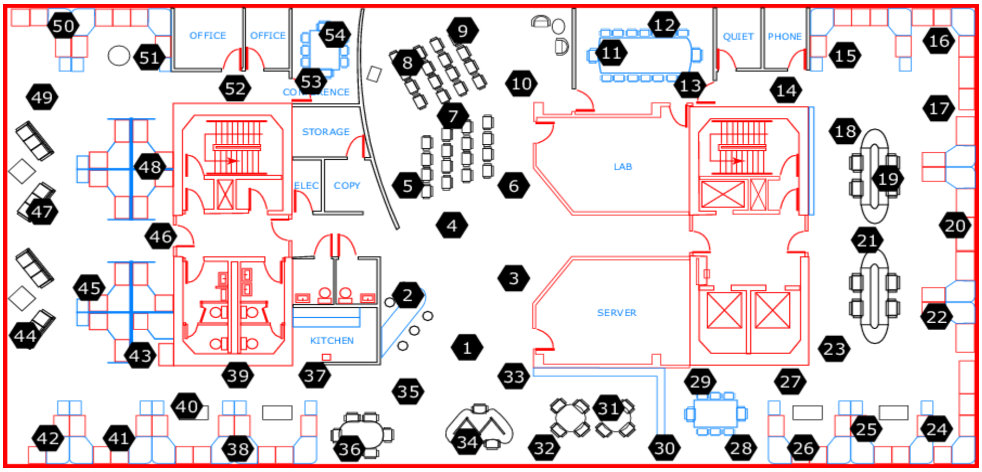

FIGURE 8. Experimental Setup of Intel Lab.

A. STABILITY THROUGH LIVE-NODES

Stability period of SNs (SPN) is the time when network start operation till the first node died (FSND) due to energy depletion. While instability period (ISN) is the time length when the first node dies till last node dies (LSND). While network life-time (NLT) is the time period start from first node died to last node died. In fact, network divided into partitions when 90% of nodes is dead. At this stage, many SNs and BS is live but breakup into different partitions, no communication towards BS. The performance of CADS with other schemes (HACH, UASNs, BSSR, ESSM) is equated in terms of SPN, ISN and LSND.

In scattered network ($T_{SN-Work}$ = 100 Nodes), CADS maintains network stability in terms of first node dead is 602 rounds. This stability is maximum in all other schemes as shown in FIGURE 9 (100 Nodes network). Other schems i.e, HACH, UASNs, BSSR and ESSM are maintaining 401, 399, 403 and 206 rounds respectfully. CADS maintains ISN is 380 rounds while HACH, UASNs, BSSR and ESSM maintain 774, 467, 470 and 466. The ISN is minimum among all other schemes. CADS also maintains LSND is 982 rounds while HACH, UASNs, BSSR and ESSM completes 980, 970, 869 and 867 respectively. The LSND of CADS is matching with ESSM but it is better than HACH, UASNs and BSSR.

In moderate network ($T_{SN-Work}$ = 150 Nodes) with 50% more nodes, CADS lasts in 698 rounds SPN which is better than HACH, UASNs, BSSR and ESSM. CADS verifies better results in SPN and it is concluded that CADS is more stable than other schemes. In order of ISN, CADS performed better than ESSM while rest of the scheme have shown low ISN values, i.e. 282 (CADS), HACH (196), UASNs (198), BSSR (202) and ESSM (498). For network life-time, CADS completes 980 rounds (maximum) while HACH, UASNs, BSSR and ESSM complete 966, 885, 872 and 797 respectively. HACH performed better in network life-time but HACH is longer ISN period while the most stable ISN is CADS.

In congested network ($T_{SN-Work}$ = 300 Nodes), with 100% increase in nodes, CADS completes 1097 rounds in SPN and here CADS covers longer stability period than HACH, UASNs, BSSR and ESSM. In this experiment, it is mentioned that compared HACH, UASNs, BSSR and ESSM, CADS is more stable in terms of long live clusters. CADS also covers 190 ISN value which is smallest and 1287 network life-time value, which is maximum. All these parameters have been shown in TABLE 9. The average is calculated and compared with other scheme. The performance of CADS with other schemes are mapped in graphs (a) 100 Nodes network (b) 150 Nodes network (c) 300 Nodes network as shown in FIGURE 9.

For refinement and optimal analysis, each scheme (CADS, ESSM, BSSR, UASNs and HACH) has been experimented for more than 200 times. First, a substantial population of results is obtained from these 200 experiments. These experiments are repeated 25 times for 100, 150 and 300 nodes with various number of rounds ranging from 100-1000. Second, averages were calculated from all these obtained values to get optimal results. Finally, based on the obtained data, the standard average errors for number of live nodes were calculated which are mentioned in TABLE 9. The results achieved from analysis of average error rate, verifies that these values have minimum effect on original experiment. Although, every scheme performed differently and show divert values from its mean position but did not show any abrupt change in their performance. Hence, it is concluded that these experiments show an optimum scenario and all schemes show how they are close to actual models.

B. CHANGING STATES AT NODE LEVEL

CADS maintains a topology of different SNs in different energy levels. These energy levels are states and each node is retained one of the state. Attaining a specific state for node, mainly depends on the contents of sensed data packets and subsequently the $C_{Messages}$ from the analyzer module. Here CADS has implemented for 100 nodes and checked the changing policy for all state with the constraints: any state < 10 % && any state > 50%, it means that for a particular time,

TABLE 9. Stability of CADS in SPN, ISN and LSND parameters.

Density of $T_{SN\text{-}Work}$	Scheme	Performance Metrics (NLT)			
		SPN (R)	ISN	LSND	Standard Average Error
$T_{SN\text{-}Work}$ = 100 Nodes	HACH	401	466	867	70.60667 ± 0.0316
	UASNs	399	470	869	71.72000 ± 0.1736
	BSSR	403	467	870	68.80333 ± 0.3974
	ESSM	206	774	980	57.95667 ± 0.6059
	Average	352.25	544.2	895.5	
	CADS	602	380	982	79.5266 ± 0.2874
$T_{SN\text{-}Work}$ = 150 Nodes	HACH	498	299	797	77.7812 ± 0.1760
	UASNs	202	670	872	74.3412 ± 0.8532
	BSSR	198	687	885	69.9901 ± 0.6097
	ESSM	196	770	966	59.0921 ± 0.7555
	Average	273.5	606.5	880.0	
	CADS	698	282	980	79.3321 ± 0.6761
$T_{SN\text{-}Work}$ = 300 Nodes	HACH	793	412	1205	79.1202 ± 0.3118
	UASNs	915	378	1293	80.3422 ± 0.8812
	BSSR	934	357	1291	72.9910 ± 0.6800
	ESSM	1015	296	1311	61.2221 ± 0.7629
	Average	914.25	360.7	1275	
	CADS	1097	190	1287	80.2311 ± 0.6911

any specific state cannot acquire by less than 10% nodes and cannot be acquire more than 50% nodes. The behavior of all nodes in different states has shown in FIGURE 10. First all nodes (100) in Active-Live state, after a number of sensing cycles, the contents are checked by analyzer and instructs the nodes to acquired different state positions. Four states are distributed in 100 random values for equalizing the states levels i.e. Active-Live = 50 nodes, Sleep-Live = 10 nodes, Active-Sleep = 30 nodes and Sleep-Sleep = 10 nodes. Continuing in same manner, in other sensing cycle, other values will acquire and same procedure will continuous for each cycle. The figure (a) is the *Active-Live* node positions in 100 nodes, figure (b) is *Sleep-Sleep* nodes in 100 nodes, figure (c) is the position of *Active-Sleep* in 100 nodes and figure (d) is the positions of *Sleep-Live* positions in 100 nodes.

C. STABILITY USING NUMBER OF CLUSTER HEADS

Frequent CH selection (re-clustering) requires more calculations for comparisons and more $C_{Messages}$ is broadcasted in the network, which creates extra traffic and consequently more energy is consumed. To minimize the network overhead (excessive sensing and broadcasting) reclustering is minimized and as a result, number of $C_{Messages}$ is reduced which ultimately prolong network lifetime. Frequent re-clustering means that more energy is consumed and this network management is less efficient. CADS works in different phases for reducing the overall sensing cycles. The contents of sensed data packets are analyzed and checked. Different $C_{Messages}$ are broadcasted to SNs to acquire different transition states. Traffic in network is adjusted and controlled by these $C_{Messages}$. Different states of nodes minimize the energy consumption and it also ensures less disconnection in the network. Contents of sensed data packets are checked by analyzer module and multicast $C_{Messages}$ to SNs for changing current state to another state. Regulating SNs in different transition states, minimizes traffic inside network and it reduces energy consumption at node level. SNs in Sleep-Live and Sleep-Sleep states, do not send data to analyzer while SNs in Active-Live and Active-Sleep sends data to analyzer. In few consecutive sensing cycles, all SN states are changed dynamically. The overall topology of the network remains active while different nodes achieves different transition states.

CADS is analyzed and compared for number of CHs and re-clustering with other schemes (BSSR, UASNs, ESSM, HACH). In 1000 rounds and 100-300 nodes, CADS performs better in CH selection and it ensures more stable network topology compared to other schemes.

In first scenario, 100 SNs ($T_{SN-Work}$ = 100) are examined for re-cluster ratio in 100 rounds. CADS have selected 10 number of CHs for total of 100 SNs, with a ratio of 1/10 (0.1 probability). Other schemes have different ratios i.e. BSSR = 23, UASNs = 27, ESSM =23 and HACH =18 in 1000 rounds in 100 nodes. In TABLE 10, CHs (re-cluster) in each scheme has been calculated after 100 rounds and average is determined for comparison. CADS reduces CHs selection/re-clustering up to 50% as shown in FIGURE 11 (a).

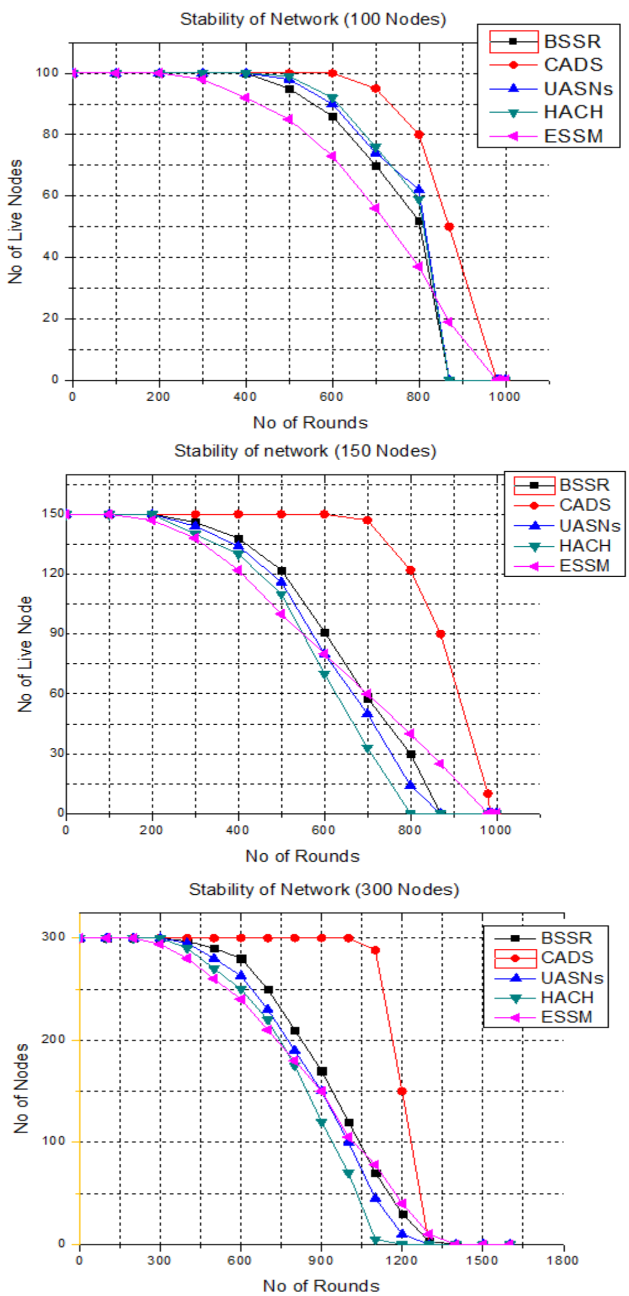

FIGURE 9. Stability of CADS in (a) 100 nodes, (b) 150 nodes, (c) 300 nodes with HACH, ESSM, UASNs, BSSR.

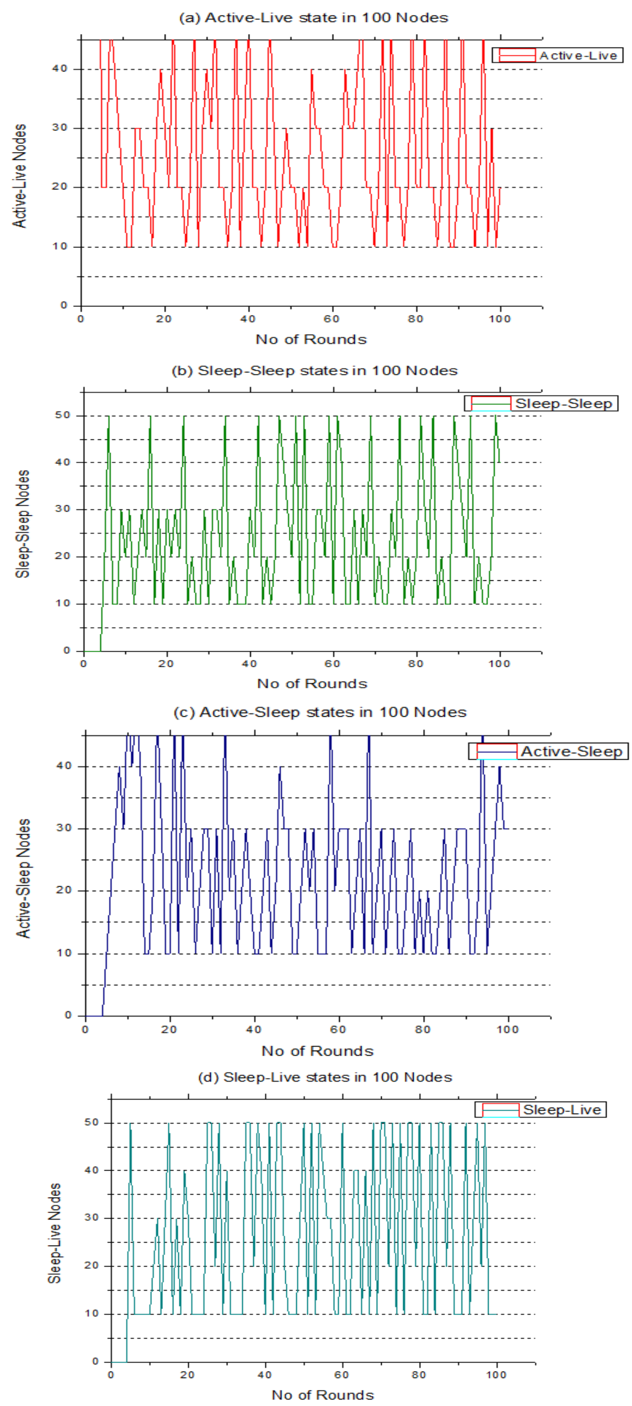

FIGURE 10. Four State levels in CADS (a) Active-Live (b) Sleep-Sleep (c) Active-Sleep (d) Sleep-Live.

Total number of re-clustering in each scheme with number of rounds have been shown in TABLE 10.

In second scenario, 150 nodes are deployed with 50% increase in nodes. CADS selects 19 CHs (number of re-clustering) in 1000 rounds in 150 nodes. While BSSR chooses 36, UASNs selects 50, ESSM selects 36 and HACH selects 29 CHs in 1000 rounds in 150 nodes. In TABLE 10, number of re-clustering/CHs are shown in 100-1000 rounds. In FIGURE 11 (b), number of re-clusters and its averages are shown with comparison with CADS. Number of re-clustering in 1000 rounds with 150 of each scheme is also determined in TABLE 10.

In third scenario, where 300 nodes are deployed, CADS performed better and it results more stable topology. CADS maintains 28 times CHs in 1000 rounds in 100 nodes. BSSR selects 54 CHs, UASNs selects 80 CHs, ESSM selects 60 CHs and HACH selected 40 CHs in 1000 rounds. The average of all these schemes (BSSR, UASNs, ESSM, HACH) is compared with CADS. CADS is more stable than all bench-marked schemes (28 by CADS and 58.5 average of all

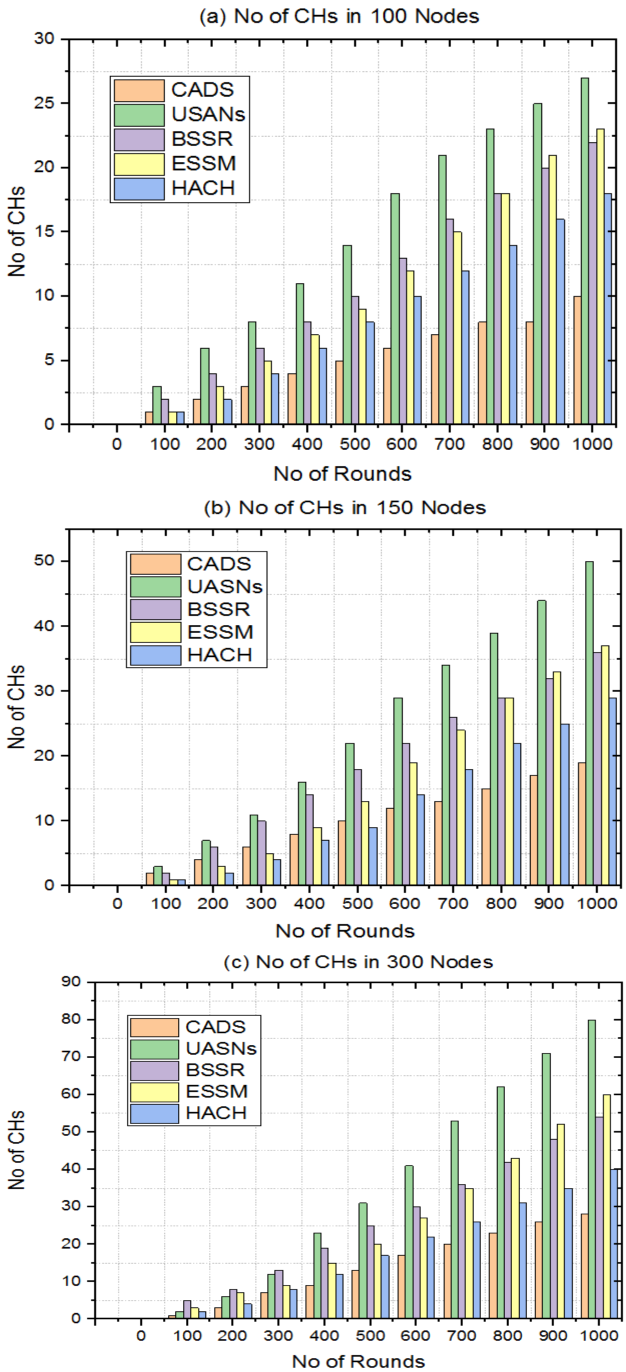

FIGURE 11. **No of CHs in (a) 100 nodes, (b) 150 nodes and (c) 300 nodes networks.**

in 300 nodes). In FIGURE 11 (c) the stability of CADS with other scheme has been shown in 100-1000 rounds with 300 nodes while for comparison all these values are mapped into TABLE 10.

We further calculated the standard average error for the total number of reclusterings/CHs in an experiment. The error values were calculated for each network scenario of 100, 150 and 300 nodes. The experiments were performed 25 times for rounds ranging from 100-1000 for each of the schemes mentioned in the TABLE 10. The standard average error for each scheme does not show extreme change in the obtained results which shows the precision of the experiments performed.

Hence, these errors are recorded in their mean (average) values of last iterations of experiments (1000 rounds), their effect on overall results is minimum. The standard error and error margin in averages has been shown in TABLE 9 and TABLE 10. It is observed that these values are not sufficient large that affect the obtained results.

D. COMPARATIVE STUDY OF PUBLISHED AND PRESENTED RESULTS

The published results of ESSM and UASNs are compared with the presented results of ESSM, UASNS and CADS.

We observed that ESSM and UASNs published results are correlating with minor differences. While ESSM and UASN presented results are correlating with minor differences. Hence, we conclude that the simulation setup and assumptions of ESSM and UASNs (published), ESSM and UASNs (presented) are same. Moreover, for the validation of the correctness of presented Vs published, we performed few more experiments. From the resultant graph as shown in FIGURE 12 and TABLE 11, we also can see a visible difference between the graph lines for the schemes under consideration. We believe that the difference between the published and presented results are due to the following reasons:

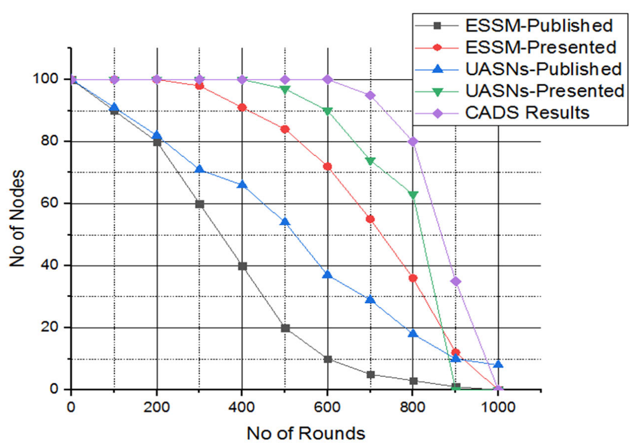

FIGURE 12. **Comparison of Published and Presented results of ESSM, UASNs and CADS.**

- The published results of ESSM and UASNs might have wrong assumptions and missing parameters in the experiment.
- Another notable reason could be the use of custom-build simulator and network setup in ESSM. In contrast, we believe that our results are more reliable and authentic due to the MATLAB simulator, which is widely-used for more realistic network simulation.
- Furthermore, our results are valid, because our simulation setup was carefully designed while taking care all the required parameters.

TABLE 10. Comparison of CADS with UASNs, BSSR, ESSM, HACH in No of CHs in 100 nodes, 150 nodes and 300 nodes networks.

Density of $T_{SN\text{-}Work}$	Scheme	No of CHs										
		100	200	300	400	500	600	700	800	900	1000	Standard Average Error
$T_{SN\text{-}Work}$ = 100 Nodes	HACH	1	2	4	6	8	10	12	14	16	18	18 ± 0.321
	UASNs	3	6	8	11	14	18	21	23	25	27	27± 0.419
	BSSR	2	4	6	8	10	13	16	18	20	22	22 ± 0.358
	ESSM	1	3	5	7	9	12	15	18	21	23	23 ± 0.345
	Average	1.75	3.75	5.75	8	10.25	13.25	16	18.25	20.5	22.5	
	CADS	1	2	3	4	5	6	7	8	8	12	12 ± 0.386
$T_{SN\text{-}Work}$ = 150 Nodes	HACH	1	2	4	7	9	14	18	22	25	29	29 ± 0.407
	UASNs	3	7	11	16	22	29	34	39	44	50	50 ± 0.573
	BSSR	2	6	10	14	18	22	26	29	32	36	36 ± 0.479
	ESSM	1	3	5	9	13	19	24	29	32	36	36 ± 0.469
	Average	1.75	4.5	7.5	11.5	15.5	21	25.5	29.75	33.25	37.75	
	CADS	2	4	6	8	10	12	13	15	17	19	19 ± 0.426
$T_{SN\text{-}Work}$ = 300 Nodes	HACH	2	4	8	12	17	22	26	31	35	40	40 ± 0.508
	UASNs	2	6	12	23	31	41	53	62	71	80	80 ± 0.763
	BSSR	5	8	13	19	25	30	36	42	48	54	54 ± 0.463
	ESSM	3	7	9	15	20	27	35	43	52	60	60 ± 0.513
	Average	3	6.25	10.5	17.25	23.25	30	37.5	44.5	51.5	58.5	
	CADS	1	3	7	9	13	17	20	23	31	38	38 ± 0.489

TABLE 11. Comparison of Published and Presented results.

Rounds	100	200	300	400	500	600	700	800	900	1000
ESSM-Published	90	80	60	40	20	10	5	3	1	0
ESSM-Presented	100	100	98	91	84	72	55	36	12	0
UASNs-Published	91	82	71	66	54	37	29	18	10	8
UASNs-Presented	100	100	100	100	97	90	74	63	0	0
CADS	100	100	100	100	100	100	95	80	35	0

- Finally, to find the reasons behind such difference needs further investigation and experimentations.

E. PERFORMANCE ANALYSIS

The results obtained from above experiments is clearly mentioning the effectiveness of CADS. In experiment for stability of network, CADS is depicting more stable network because of the implementation of four state model. The main advantage of four state model is that it never missed an event while different nodes have different states. The overall topology of CADS is active whereas many nodes (25%) is in sleep state (Sleep-Sleep). Other nodes adapt sleep state dynamically and maintains active topology all the times. The above experiments clarify how four state model affect the performance of CADS. The above experiments also prove that other schemes have states based on direct energy depletion while in CADS, the nodes maintain the states based on internal components. CADS has implemented at node level and different calculations have mapped in different diagrams. In first experiments (100 nodes, 150 nodes, 300 nodes) as in FIGURE 9, longer stability period means that CADS is more energy-efficient due to reducing the sensing cycles. The nodes maintain 25% of each state and each state consumes a specific amount of energy. When a node acquires a specific state, it operates at that energy level. Some nodes at lowest energy level (Sleep-Sleep), with no processing, sensing and communication. While in other transition state (Sleep-Live), the communication is in sleep and other components are active. Using these transition states, the sensing is controlled at node level. Only useful sensed data packets are sent to analyzer module. All this process reduces sensing cycles which eventually leads to minimize the over-cost processing and ultimately, reduces extensive sensing. In first experiment, metric values i.e. SPN, FSND, ISN, last LSND and NLT are calculated for observing behavior of CADS. Out-come of experiment clarify that how proposed scheme improve energy-efficiency compared with other bench-mark schemes. It is clear from the statistics that CADS performed better than other schemes and show more stable network.

In the second experiments, in FIGURE 10, sensing is controlled and analyzer module imposes four state policy.

The $C_{Messages}$ transmitted by analyzer module, it categories all SNs in different transition states. Nodes adapt different states after directing by analyzer and each node now consumes a definite amount of energy. When SNs only sensed useful data packets and dropped redundant values, the traffic inside network is reduced. As a result, the number of $C_{Messages}$ and $L_{Messages}$ are minimized and consequently, cluster maintain its shape for longer period and the process of re-cluster is minimized. Which ultimately results in prolonging cluster life and reducing re-cluster overhead. All this process reduced frequent communication and avoid congestion. Therefore, the overall traffic of the network has decreased and as results, it prolongs the network life-time. This experiment has shown actual positions of each sensor component at a specific time. It is the generic four state model where each node adapts any transition state. While in last experiment in FIGURE 11, network stability is calculated with selecting number of CH. Frequent re-clustering/CH selection procedure requires more operations and it consumes more energy. It means a scheme with frequent CH selection consume more energy. CADS minimizes overall network traffic including $C_{Messages}$ and $L_{Messages}$. In 100, 150 and 300 nodes network, proposed scheme has lesser number of CH and it confirms more stability in terms of CH selection/re-cluster.

Energy-efficiency in WSNs is also affected and correlated with security mechanisms. As these networks need light-weight, massive and heavy protocols are hard to implement. Limited energy and constraints on other resources, always needed a new architecture where security of network resources are ensured with minimum consumption of energy [89], [90]. Some schemes can improve the efficiency and security of CADS. For example, efficient resource management, an architecture for monitoring and collecting data of patient's health condition is proposed for mutual authentication [91]. Mutual authentication is confirmed by using five step procedure. Hash function, session key and BAN logic is mainly used for authentication of mobile users. Another energy efficient, light-weight and privacy-preserving mutual authentication protocol for industrial WSNs [92] has proposed. They have used XOR operation, light-weight one-way hash function and physical un-clone-able function for physical secure authentication process. No secret credentials are hosted in communicating devices. To minimize energy consumption and reduce communication delay, another scheme known as ''GA enabled distributed zone approach'' [93] has proposed. In this scheme, first calculate shortest route among all paths. Genetic algorithm (less complex, better performance) is used for decreasing these two factors. Complete genetic algorithm is mapped and an optimized solution is resultant which is based on chromosome, selection, crossover and mutation. All parameters (delay, energy consumption, computational cost and full connectivity) are analytically derived and tested with other algorithms including DIR, MFR, RRDLA, Dijkstra and Ahn-Ramakrishna [94]. To minimize communication and computation overhead in WSN, 3-factor user authentication based on Elliptic curve cryptography has proposed [95]. These factors are password, smart card and biometric. The BAN method is used for both user and device authentication. Another scheme is which authors have implemented SHA-3 in hardware for WBSN [96]. The Random-access memory, logic gates and finite state machines are combined in establishing the structure of SHA-3. FPGA is used for implementation and checking the effectiveness of the SHA-3 for WBSN. To minimize the transmission delay and resistance to all known security attacks with efficient energy management, security disjoint routing-based verified message (SDRVM) [97] is proposed. Two sets are created for data are; data CDS and message CDS. Depends upon the remaining energy of a node, data packets are retransmitted for effective data delivery. Data packets are marked with ID information and these ID are used by nodes for logging. These ID are updated probabilistically when data is received with same IDs. Depends on energy, if remaining energy is minimizing, the marking probability will be decreased. And if remaining energy is greater, then marking probability is increasing [98], [99].

We tested CADS with 100, 150, 300 nodes. These may be applying in any case either for scattered or congested network. The out-come (Probability density function) may be vary in other scenarios. On the other hand, network experiences extra overhead in arranging all nodes unison because of different transition states of nodes. Although, energy consumption decreases in CADS, but due to changing states, extra overhead generates which also effects network efficiency. We have experimented CADS with Live nodes in the form of network life-time NLT (SPN, FSND, ISN, LSND), nodes states at different energy levels at different time slots and number CH selection. CADS can be tested for delay, latency and throughput in the form of $C_{Messages}$ and $L_{Messages}$. These factors can affect the performance because overall traffic decreases but $C_{Message}$ creates extra overhead and it can affect the normal procedures of CADS.

The authenticity of presented results is verified by comparing them with the benchmark published results. We created a table and mentioned published results with presented results and also calculated the difference. The experiments comprise of 100, 150 and 300 nodes and with three parameters i.e. Stability Period, Instability period and Last node died. We calculated these parameters for HACH, UASNs, BSSR, ESSM and compared CADS with the average of all these values as shown in TABLE 9.

The presented and published results show variance in the difference of both values. It means, in some cases proposed CADS performs close to other scheme whereas in other scenario, CADS performed diverse values. Presented results validate published results in some cases but not in all cases. There is no correlation between all difference values of proposed CADS and other benchmark schemes. Proposed CADS validates presented results by implementing node operations in simulation and with closed environment in Lab. In simulations, complete network is testified with extreme values while in Lab, we have only ten sensors and a limited

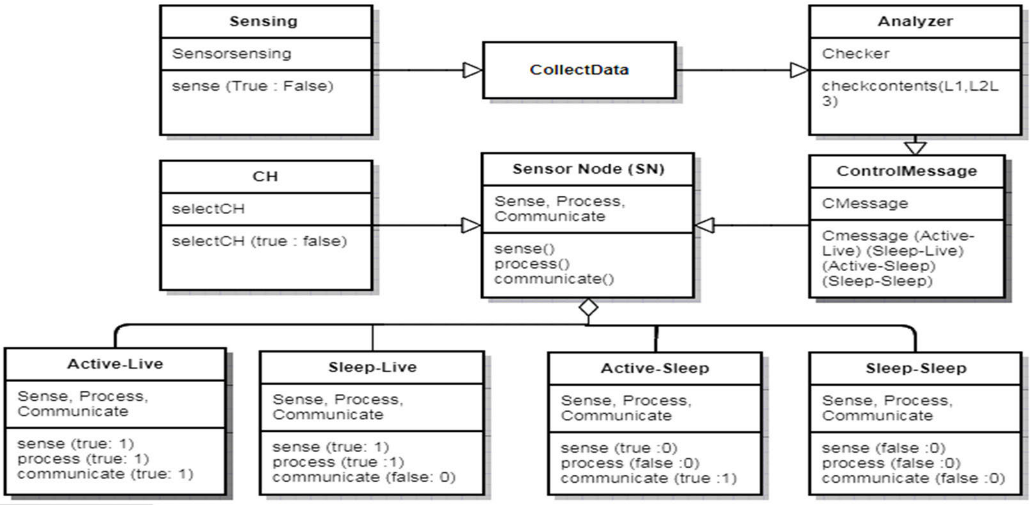

FIGURE 13. Class Diagram of CADS.

data set. Both in simulated values and real time data set are coincided.

For reliability, in the proposed CADS solution, the network is simulated in real time scenario. All reliability measures have been checked, For instance, established connected links, operating environment, broadcast and multicast communications, optimum state transitions (FIGURE 10) and re-clustering (FIGURE 11). Furthermore, for stability, CADS is found to be more consistent in configuring node states by transmitting control massages. These messages dynamically manage network structure and maintain a stable network topology. Experiments proved that CADS is comparatively more stable as shown in FIGURE 9 and TABLE 9 which shows the stability and instability period. The figure also shows that CADS is scalable where it performs better by increasing the number of nodes in network.

VI. CONCLUSION AND FUTURE WORK

Wireless Sensor Networks (WSNs) have numerous applications ranging from civil to military domains. They are usually deployed in inaccessible and hostile environment where normal data collection is impossible. Since WSNs are highly resource constrained therefore, efficient resource management is always desired. From the overall management perspective, for better power management, in this paper we have proposed a novel scheme "Content-based Adaptive and Dynamic Scheduling (CADS) using two ways communication model in WSNs". CADS is used to avoid redundant data values and reduces forwarding of un-necessary data packets. Four states (Active-Live, Sleep-Live, Active-Sleep, Sleep-Sleep) have defined and controlled by using $C_{Messages}$ transmitted by analyzer module of BS. States are derived on different combination of internal components of a node. Different states of a node are different energy levels and each state consumed a specific amount energy. $C_{Messages}$ in backward direction plays important role in changing these states either to aggregate or to avoid redundant data. CADS has implemented at component level for prolonging the life-time of WSNs and save precious energy at node level. CADS has experimented in both scattered and congested networks and the obtained results have proved that it performs better in both cases compared to other state-of-the-art schemes. Simulation results demonstrates that it increased energy-efficiency in terms of network life-time by 9.65% in 100 nodes-network, 11.36% in 150 nodes-network and 0.94% in 300 nodes. On the other hand, CADS has longer stability in terms of increasing cluster life by 87.5% in 100 node-network, 94.73% in 150 nodes-network and 53.9% in 300 nodes-network.

Although, CADS has implemented in four state model and a smart analyzer module in BS. In future, we are trying to implement CADS in deferent scenario and want to check its behavior in different mobility models, effect of pattern matching and frequency distribution, error analysis of different circumstances parameters and effect of security mechanism. It can be interesting by merging the main idea with authentication schemes like RAPM, LPPMA, 3-Factor authentication. The ESPDA is very near approach and the data gathering method can be implemented on CH in CADS. For secure path establishing between SNs and CH, between CH and BS we can merge Relay Selection Joint Consecutive Packet Routing and Adaptive Data and Verified Message Disjoint Security Routing. Energy-efficiency can be increased by combining the current scheme with other factors including energy and time. The performance may be different if CADS is tested in those scenarios where nodes/CH are mobile.

As per the circumstances parameters are concerned, our short time experiments having very little or negligible effect. However, it will be interesting to see that how in open real scenario, the performance is influenced by these parameters. We are committed to undertake this task in the future. Moreover, we also intend to formally verify and validate the proposed scheme using validation and verification tools

TABLE 12. Different of Scheme with Pros and Cons.

Scheme	Parameter used	Control Packets	Messages	Merits	Demerits
PEAS 2002	Probability Density function (PDF)	Yes,	Probing Random	Useful for Dense WSN, 1st node based activities	Active node never sleep, routing overhead probe, frequent disconnection
PECAS 2004	Random deployment, PDF	Yes, broadcast	Random probes, time based	change state after period, dense network, controllable,	Increase latency, message overhead probes, random deployment,
EECG 2007	Location awareness	Yes, deterministic	Random probes	location aware, minimize cascading effect,	Near to BS will dead soon, additional nodes have deployed
RBSP 2014	Back-off algorithm, PDF	Yes, random	Probing	State change by back-off algorithm, energy based strategy, backup nodes in active forms.	Overhead in calculating energy and probing. Active nodes remains live for long period
DCBPS 2015	Battery discharge curve	Yes, random	Hello packets	Batty discharge curve, back-off sleep period , fully connected, full coverage	Sleepy nodes only activate when near to drain completely, congestion due to hello packets, increase latency
CES 2015	Distance test, perimeter and center	Yes, Random & Deterministic	Probing, periodic	Location aware, connectivity conditions and coverage area, Distance tests, perimeter and center are used, useful for limited data traffic	Extended waiting time, redundant nodes are needed, active nodes die more frequently
CCP 2005	Connectivity configuration	Yes, deterministic	Broadcast, periodic	Location aware, based on sensing range, intersection are easily revealed	Overhead of calculation for neighbors and broadcast messages, not fully connected, may missed events
ECCP 2009	Overlapping zones	Yes, deterministic	Broadcast, periodic	Cross layered, Balance zones with full coverage, useful in overlapping zones	Extra nodes required than CCP, calculating overlap utilize more energy, increase latency in linear topology
OGDC 2005	Geographic density	Yes, random	Probing	Synchronized, location aware, minimum overlapping, useful in congested network	Frequent state change, more overhead due to fixed preamble sampling, each rounds operates in two states
PCB 2010	Activator nodes	Yes, probabilistic & deterministic	Random probing	Good coverage area in hexagonal structure, Low energy consumption when traffic is low	More energy consumed in developing and maintaining hexagonal structure, Activator needed, calculation for determining energy is also tedious process
PCPP 2007	Centralized algorithm	Yes, probabilistic	Random probing	Uncovered areas are discovered, based on three conditions low ID node is selected as Active, useful in small networks	Centralized approach, not working in distributed network, increase latency when load is high
BECG 2013	RSSI	Yes, Random	Check messages	Checking nodes transmit, determine energy level and status of node, node having less amount energy will automatically sleep	Coverage redundancy, uncovered area, not known exact number of nodes in target area.
EBC [13]	Voronoi Algorithm	Yes, random	Check Messages	Remove redundant nodes by voronoi algorithm, holes are discovered and remove, data collision is minimum	Mainly focused on holes, if two active nodes in same will be results as coverage redundancy, mobility of nodes also create congestion
CPL 2011	Euclidian algorithm	Yes, random	Hello Messages	Area of Interest, Euclidian algorithm calculates overlaps, zone wise CH based structures, CH change state of Node, update time to time.	Unconnected regions between zones, both inter and intra cluster communication utilize more energy, portioned and increase latency
CEE 2019	Distance, energy	Yes, probabilistic	Broadcast packets	Probability theory, better selection process of Active node, Message passing to reduce contention latency	Fixed duty cycle not adaptive to dynamic traffic loads, not recommended for congested network
ECMM 2018	M2WSN, LDC, WuRx	Yes, deterministic	Control packets	Minimize overhead, save energy in overhearing, dynamic load balanced	Increasing latency in calculations, before time sleeping, not useful in huge network
RS-CPR 2018	Distance, energy, memory queue	Yes, Time based	Broadcast packets	On-Demand transmission, weight on each distance, adaptive synchronous	Needs extra buffer, energy used in maintaining extra calculation for weighting and live buffer for long time
ESQG 2018	Voronoi diagram, residual energy	Yes, random	Awaking frequency	Useful in scattered networks, high throughput, low latency	Increasing latency if network load is high, maintain voronoi diagram needs extra energy.
MDET 2018	Spanning three,	Yes, time based	Broadcast messages	Undetermined-delay is calculated by minimum spanning tree, it reduces latency and delay causes by flooding. Very useful in NP-complete problems	Creates extra overhand in calculations, greater latency due to periodic sleep, sometime small listening periods effects performance

ESSM 2017	Fuzzy Metrix	Yes, Random	Hello Messages	Fuzzy matrix have developed, nodes have categories on correlation function, redundant nodes have change its change from active to sleep, useful for critical application, high accuracy of data.	In disperse environment, the correlation of data will be low and the network will be used more energy, randomly changing state,
UASNs 2018	Mimetic algorithm	Yes, Random, Deterministic	Probing messages	Redundant nodes underwater, memetic algorithm is used for activate nodes, memetic algorithm reduce network congestion by equally distributed sensors, full coverage	Need calculation for calculating energies at dying stage of active nodes, passive approach, increase latency due to high traffic
BSSR 2019	Solar energy harvesting mechanism	Yes, Random, Deterministic	Probing Messages	Having Fixed duty cycles, at lower traffic energy consumption is minimum, high throughput in scattered network	Latency in dense network, packets feels delay in congested network
HACH 2019	stochastic selection of inactive nodes, Boltzmann selection	Yes, Random	Hello Packets	HEECHS algorithm and Advance heuristics is used global and local search process to improve quality. Both clustering and sleep schedules improve optimization process.	Massive calculation decrease efficiency and increases latency, difficult to maintain in congested networks

such as Petri net. Furthermore, a mathematical and statistical analysis will also be carried out. Additionally, we intend to test the proposed scheme in a real time testbed scenario and then compare all the obtained results to find the authenticity of the results and behavior of the proposed scheme.

APPENDIX A

CADS has been implemented in MTALB with various topological structures both in a scattered and congested networks. Moreover, various matrices are used to evaluate the performance of CADS. The Figure, given below, shows the implementation of the CADS algorithm with the First-Order Radio Model with a class diagram. Main classes and flow of control has been shown. There are main sensing class with a method (sensing), Analyzer class with a method of check contents with three parameters (L1, L2, L3), ControlMessage class with a method and parameters Active-Live, Sleep-live, Active-Sleep and Sleep-Sleep. There is another class CH which is used for selection of CH with a method of selectCH. The main sensor Node class is abstract class and four child classes have been derived from it. These four classes are the states of that a node can acquire during sensing. Each of the derived classes has their own implementation each with three parameters.

APPENDIX B

All schemes are changing states based on some criteria in passive and proactive manner. In the following table, all those schemes have been argued with merits and demerits. The table also mentions how different parameters play vital role in changing states. Here type of control messages and strategy of transmission of control message are also stated.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] J. A. Stankovic, A. D. Wood, and T. He, "Realistic applications for wireless sensor networks," in *Theoretical Aspects of Distributed Computing in Sensor Networks* (Monographs in Theoretical Computer Science), S. Nikoletseas and J. Rolim, Eds. Berlin, Germany: Springer, 2011, pp. 835–863. [Online]. Available: https://doi.org/10.1007/978-3-642-14849-1_25

[4] D. Puccinelli and M. Haenggi, "Wireless sensor networks: Applications and challenges of ubiquitous sensing," *IEEE Circuits Syst. Mag.*, vol. 5, no. 3, pp. 19–31, 2005.

[5] S. B. Alla, A. Ezzati, and A. Mohsen, *Hierarchical Adaptive Balanced Routing Protocol for Energy Efficiency in Heterogeneous Wireless Sensor Networks*. Rijeka, Croatia: InTechOpen, Oct. 2012.

[6] B. Qiao and K. Ma, "An enhancement of the ZigBee wireless sensor network using Bluetooth for industrial field measurement," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jul. 2015, pp. 1–3.

[7] J. Hill, "A wireless embedded sensor architecture for system-level optimization," UC Berkeley, Berkeley, VA, USA, Tech. Rep., 2002.

[8] M. Yuxing, Z. Huiyuan, and Y. Dongmei, "Weak node protection to maximize the lifetime of wireless sensor networks," *J. Syst. Eng. Electron.*, vol. 29, no. 4, pp. 693–706, Aug. 2018.

[9] G. Mehmood, M. Z. Khan, S. Abbas, M. Faisal, and H. U. Rahman, "An energy-efficient and cooperative fault-tolerant communication approach for wireless body area network," *IEEE Access*, vol. 8, pp. 69134–69147, 2020.

[10] A. K. Idrees, W. L. Al-Yaseen, M. A. Taam, and O. Zahwe, "Distributed data aggregation based modified K-means technique for energy conservation in periodic wireless sensor networks," in *Proc. IEEE Middle East North Afr. Commun. Conf. (MENACOMM)*, Apr. 2018, pp. 1–6.

[11] K. Kifayat, M. Merabti, Q. Shi, and S. Abbas, "Component-based security system (COMSEC) with QoS for wireless sensor networks," *Secur. Commun. Netw.*, vol. 6, no. 4, pp. 461–472, Apr. 2013.

[12] E. Aguirre, P. Lopez-Iturri, L. Azpilicueta, A. Redondo, J. J. Astrain, J. Villadangos, A. Bahillo, A. Perallos, and F. Falcone, "Design and implementation of context aware applications with wireless sensor network support in urban train transportation environments," *IEEE Sensors J.*, vol. 17, no. 1, pp. 169–178, Jan. 2017.

[13] L. B. Bhajantri, P. H. C, and M. Patil, "Context aware topology control in wireless sensor networks," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Jul. 2016, pp. 825–830.

[14] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sensors J.*, vol. 18, no. 1, pp. 340–347, Jan. 2018.

[15] A. Ahmed, K. A. Bakar, M. I. Channa, A. W. Khan, and K. Haseeb, "Energy-aware and secure routing with trust for disaster response wireless sensor network," *Peer–Peer Netw. Appl.*, vol. 10, no. 1, pp. 216–237, Jan. 2017.

[16] M. N. Khan, H. U. Rahman, and M. Z. Khan, "An energy efficient adaptive scheduling scheme (EASS) for mesh grid wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 146, pp. 139–157, Dec. 2020.

[17] X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, and Y. L. Xu, "Energy efficient clustering for WSN-based structural health monitoring," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2768–2776.

[18] B. Aoun and R. Boutaba, "Clustering in WSN with latency and energy consumption constraints," *J. Netw. Syst. Manage.*, vol. 14, no. 3, pp. 415–439, Sep. 2006.

[19] M. K. Watfa, H. AlHassanieh, and S. Selman, "Multi-hop wireless energy transfer in WSNs," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1275–1277, Dec. 2011.

[20] S. Dutta, M. S. Obaidat, K. Dahal, D. Giri, and S. Neogy, "Comparative study of different cost functions between neighbors for optimizing energy dissipation in WSN," *IEEE Syst. J.*, vol. 13, no. 1, pp. 289–300, Mar. 2019.

[21] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2000, p. 10.

[22] N. Israr and I. U. Awan, "Multihop clustering algorithm for load balancing in wireless sensor networks," *Int. J. Simul.*, vol. 8, no. 3, pp. 13–24, 2007.

[23] B. Manzoor, N. Javaid, O. Rehman, M. Akbar, Q. Nadeem, A. Iqbal, and M. Ishfaq, "Q-LEACH: A new routing protocol for WSNs," *Procedia Comput. Sci.*, vol. 19, pp. 926–931, 2013.

[24] E. László, K. Tornai, G. Treplán, and J. Levendovszky, "Novel load balancing scheduling algorithms for wireless sensor networks," in *Proc. 4th Int. Conf. Commun. Theory, Rel., Qual. Service, Budapest*, 2011, pp. 49–54.

[25] C. Gherbi, Z. Aliouat, and M. Benmohammed, "A novel load balancing scheduling algorithm for wireless sensor networks," *J. Netw. Syst. Manage.*, vol. 27, no. 2, pp. 430–462, Apr. 2019.

[26] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proc. IEEE Proc. Aerosp. Conf.*, vol. 3, Mar. 2002, p. 1–3.

[27] A. Singh, K. Babbar, and A. Malik, "Clustering and energy efficient routing protocol for wireless sensor network using genetic algorithm," *Int. J. Comput. Appl.*, vol. 119, no. 7, pp. 40–43, Jun. 2015.

[28] H. Ali, W. Shahzad, and F. A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *Appl. Soft Comput.*, vol. 12, no. 7, pp. 1913–1928, Jul. 2012.

[29] F.-E. Bai, H.-H. Mou, and J. Sun, "Power-efficient zoning clustering algorithm for wireless sensor networks," in *Proc. Int. Conf. Inf. Eng. Comput. Sci.*, Dec. 2009, pp. 1–4.

[30] K. Ahmed and M. Gregory, "Integrating wireless sensor networks with cloud computing," in *Proc. 7th Int. Conf. Mobile Ad-Hoc Sensor Netw.*, Dec. 2011, pp. 364–366.

[31] X. Meng, X. Shi, Z. Wang, S. Wu, and C. Li, "A grid-based reliable routing protocol for wireless sensor networks with randomly distributed clusters," *Ad Hoc Netw.*, vol. 51, pp. 47–61, Nov. 2016.

[32] T. J. Saleem, "A detailed study of routing in Internet of Things," *Int. J. Eng. Sci. Innov. Technol.*, vol. 5, no. 3, May 2016.

[33] S. Yadav and R. S. Yadav, "A review on energy efficient protocols in wireless sensor networks," *Wireless Netw.*, vol. 22, no. 1, pp. 335–350, Jan. 2016.

[34] S. Khan, A.-S. K. Pathan, and N. A. Alrajeh, *Wireless Sensor Networks: Current Status and Future Trends*. Boca Raton, FL, USA: CRC Press, Nov. 2016.

[35] H. El Alami and A. Najid, "Optimization of energy efficiency in wireless sensor networks and Internet of Things: A review of related works," in *Nature-Inspired Computing Applications in Advanced Communication Networks*, ed. Hershey, PA, USA: IGI Global, 2020, pp. 89–127.

[36] J.-S. Lee and C.-L. Teng, "An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 1095–1103, Aug. 2017.

[37] H. El Alami and A. Najid, "A new fuzzy clustering algorithm to enhance lifetime of Wireless Sensor Networks," in *Proc. Int. Afro-Eur. Conf. Ind. Advancement*, 2016, pp. 68–76.

[38] X.-Y. Zhang, J. Zhang, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, and Y. Li, "Kuhn–Munkres parallel genetic algorithm for the set cover problem and its application to large-scale wireless sensor networks," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 695–710, Oct. 2016.

[39] Y. Chang, X. Yuan, B. Li, D. Niyato, and N. Al-Dhahir, "A joint unsupervised learning and genetic algorithm approach for topology control in energy-efficient ultra-dense wireless sensor networks," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2370–2373, Nov. 2018.

[40] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Proc. 23rd Int. Conf. Distrib. Comput. Syst.*, May 2003, pp. 28–37.

[41] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2004, pp. 129–143.

[42] Y.-S. Yen, S. Hong, R.-S. Chang, and H.-C. Chao, "An energy efficient and coverage guaranteed wireless sensor network," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 2923–2928.

[43] A. More and V. Raisinghani, "Random backoff sleep protocol for energy efficient coverage in wireless sensor networks," in *Advanced Computing, Networking and Informatics* (Smart Innovation, Systems and Technologies), vol. 28, M. Kumar Kundu, D. Mohapatra, A. Konar, and A. Chakraborty, Eds. Cham, Switzerland: Springer, 2014. [Online]. Available: https://doi.org/10.1007/978-3-319-07350-7_14

[44] A. More and V. Raisinghani, "Discharge curve backoff sleep protocol for energy efficient coverage in wireless sensor networks," *Procedia Comput. Sci.*, vol. 57, pp. 1131–1139, 2015.

[45] N.-T. Le and Y. M. Jang, "Energy-efficient coverage guarantees scheduling and routing strategy for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, Aug. 2015, Art. no. 612383.

[46] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Ad Hoc Sensor Wireless Netw.*, vol. 1, nos. 1–2, pp. 89–124, 2005.

[47] S. Zhang, Y. Liu, J. Pu, X. Zeng, and Z. Xiong, "An enhanced coverage control protocol for wireless sensor networks," in *Proc. 42nd Hawaii Int. Conf. Syst. Sci.*, Jan. 2009, pp. 1–7.

[48] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Trans. Sensor Netw.*, vol. 1, pp. 36–72, Aug. 2005.

[49] M. Hefeeda and H. Ahmadi, "Energy-efficient protocol for deterministic and probabilistic coverage in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 579–593, May 2010.

[50] J.-P. Sheu and H.-F. Lin, "Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 2631–2636.

[51] N. T. Le, N. Saha, R. K. Mondal, S. Chae, and Y. M. Jang, "Balanced energy and coverage guaranteed protocol for wireless sensor networks," in *Proc. Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Feb. 2014, pp. 1–6.

[52] M. S. Aliyu, A. H. Abdullah, H. Chizari, T. Sabbah, and A. Altameem, "Coverage enhancement algorithms for distributed mobile sensors deployment in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 2016, p. 29, Mar. 2016.

[53] S. Misra, M. Pavan Kumar, and M. S. Obaidat, "Connectivity preserving localized coverage algorithm for area monitoring using wireless sensor networks," *Comput. Commun.*, vol. 34, no. 12, pp. 1484–1496, Aug. 2011.

[54] L. Wang, J. Yan, T. Han, and D. Deng, "On connectivity and energy efficiency for Sleeping-Schedule-Based wireless sensor networks," *Sensors*, vol. 19, no. 9, p. 2126, May 2019.

[55] J. Aranda, M. Scholzel, D. Mendez, and H. Carrillo, "An energy consumption model for MultiModal wireless sensor networks based on wake-up radio receivers," in *Proc. IEEE Colombian Conf. Commun. Comput. (COLCOM)*, May 2018, pp. 1–6.

[56] J. Aranda, M. Schölzel, D. Mendez, and H. Carrillo, "Multimodal wireless sensor networks based on wake-up radio receivers: An analytical model for energy consumption," in *Revista Facultad de Ingeniería Universidad de Antioquia*. 2019, pp. 113–124.

[57] M. Peng, W. Liu, T. Wang, and Z. Zeng, "Relay selection joint consecutive packet routing scheme to improve performance for wake-up radio-enabled WSNs," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–32, Jan. 2020.

[58] K. Xiao, R. Wang, H. Deng, L. Zhang, and C. Yang, "Energy-aware scheduling for information fusion in wireless sensor network surveillance," *Inf. Fusion*, vol. 48, pp. 95–106, Aug. 2019.

[59] N. Tamura and H. Yomo, "Wake-up control adapting to Destination's Active/Sleep state for on-demand wireless sensor networks," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.

[60] L. Cheng, J. Niu, C. Luo, L. Shu, L. Kong, Z. Zhao, and Y. Gu, "Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks," *Comput. Netw.*, vol. 134, pp. 66–77, Apr. 2018.

[61] J. Aranda, H. Carrillo, and D. Mendez, "Enhanced multimodal switching mechanisms for node scheduling and data gathering in wireless sensor networks," in *Proc. IEEE Colombian Conf. Commun. Comput. (COLCOM)*, Aug. 2017, pp. 1–6.
[62] H. P. Gupta, S. V. Rao, and T. Venkatesh, "Sleep scheduling protocol for k-coverage of three-dimensional heterogeneous WSNs," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8423–8431, Oct. 2016.
[63] R. Wan, N. Xiong, and N. T. Loc, "An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks," *Hum.-Centric Comput. Inf. Sci.*, vol. 8, no. 1, p. 18, Dec. 2018.
[64] H. Wang, Y. Li, T. Chang, and S. Chang, "An effective scheduling algorithm for coverage control in underwater acoustic sensor network," *Sensors*, vol. 18, no. 8, p. 2512, Aug. 2018.
[65] M. Song, W. Lu, H. Peng, Z. Xu, and J. Hua, "A method of balanced sleep scheduling in renewable wireless sensor networks," in *Proc. Int. Conf. Mach. Learn. Intell. Commun.*, 2018, pp. 293–302.
[66] M. Song, W. Lu, H. Peng, Z. Xu, and J. Hua, "A method of balanced sleep scheduling in renewable wireless sensor networks," in *Proc. Mach. Learn. Intell. Commun., 3rd Int. Conf. (MLICOM)*, Hangzhou, China, Jul. 2018, p. 293.
[67] M. O. Oladimeji, M. Turkey, and S. Dudley, "HACH: Heuristic algorithm for clustering hierarchy protocol in wireless sensor networks," *Appl. Soft Comput.*, vol. 55, pp. 452–461, Jun. 2017.
[68] G. Wang, D. Turgut, L. Bölöni, Y. Ji, and D. C. Marinescu, "Improving routing performance through m-limited forwarding in power-constrained wireless ad hoc networks," *J. Parallel Distrib. Comput.*, vol. 68, no. 4, pp. 501–514, 2008.
[69] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 21st Annu. Joint Conf. IEEE Computer Commun. Societies*, Jun. 2002, pp. 1567–1576.
[70] S. A. Awwad, C. K. Ng, N. K. Noordin, M. F. A. Rasid, and A. H. Alhawari, "Mobility and traffic adapted Cluster Based Routing for Mobile Nodes (CBR-Mobile) protocol in wireless sensor networks," in *Proc. Int. Conf. Ad Hoc Netw.*, 2010, pp. 281–296.
[71] S. H. Choi and K. O. Lee, "A cluster based energy efficient location routing protocol in wireless sensor networks," Tech. Rep., 2011.
[72] S. K. Singh, M. Singh, and D. K. Singh, "Energy-efficient homogeneous clustering algorithm for wireless sensor network," *Int. J. Wireless Mobile Netw. (IJWMN)*, vol. 2, no. 3, pp. 49–61, 2010.
[73] M. G. Calle Torres, "Energy consumption in wireless sensor networks using GSP," Ph.D. dissertation, Univ. Pittsburgh, Pittsburgh, PA, USA, Apr. 2006.
[74] N. Ghazisaidi, C. M. Assi, and M. Maier, "Intelligent wireless mesh path selection algorithm using fuzzy decision making," *Wireless Netw.*, vol. 18, no. 2, pp. 129–146, Feb. 2012.
[75] F. He, H. Huang, R. Wang, and L. Jiang, "Data fusion-oriented cluster routing protocol for multimedia sensor networks based on the degree of image difference," *CCF Trans. Netw.*, vol. 1, pp. 65–77, Feb. 2019.
[76] P. G. V. Naranjo, M. Shojafar, A. Abraham, and E. Baccarelli, "A new stable election-based routing algorithm to preserve aliveness and energy in fog-supported wireless sensor networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 002413–002418.
[77] T. Amgoth and P. K. Jana, "Energy-aware routing algorithm for wireless sensor networks," *Comput. Electr. Eng.*, vol. 41, pp. 357–367, Jan. 2015.
[78] A. Chamam and S. Pierre, "A distributed energy-efficient clustering protocol for wireless sensor networks," *Comput. Electr. Eng.*, vol. 36, no. 2, pp. 303–312, Mar. 2010.
[79] *CC2420 Errata Note 001, Rev. 1.0, Revision Date Description/Changes 2.0 2005-03-18*, Problem Fixed on Lot Codes Higher Than WB8341.00, 1.0 2004-10-18 Initial Release, Mar. 2013.
[80] D. Vengertsev and H. Thakkar, "Anomaly detection in graph: Unsupervised learning, graph-based features and deep architecture," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2015.
[81] C. Truong, K. Romer, and K. Chen, "Sensor similarity search in the Web of things," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2012, pp. 1–6.
[82] A. R. M. Kamal, C. Bleakley, and S. Dobson, "Packet-level attestation (PLA): A framework for in-network sensor data reliability," *ACM Trans. Sensor Netw.*, vol. 9, pp. 1–28, Apr. 2013.
[83] B. Stojkoska, D. Solev, and D. Davcev, "Data prediction in WSN using variable step size LMS algorithm," in *Proc. 5th Int. Conf. Sensor Technol. Appl.*, 2011, pp. 1–6.
[84] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux. (2004). *Intel Lab Data (2004)*. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html
[85] B. Risteska Stojkoska, A. Popovska Avramova, and P. Chatzimisios, "Application of wireless sensor networks for indoor temperature regulation," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 5, May 2014, Art. no. 502419.
[86] T. Islam, Z. Uddin, and A. Gangopadhyay, "Temperature effect on capacitive humidity sensors and its compensation using artificial neural networks," *Sensors Transducers*, vol. 191, no. 8, p. 126, 2015.
[87] K. M. Awan, P. A. Shah, K. Iqbal, S. Gillani, W. Ahmad, and Y. Nam, "Underwater wireless sensor networks: A review of recent issues and challenges," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–20, Jan. 2019.
[88] H. M. Hasanien, A. S. Abd-Rabou, and S. M. Sakr, "Design optimization of transverse flux linear motor for weight reduction and performance improvement using response surface methodology and genetic algorithms," *IEEE Trans. Energy Convers.*, vol. 25, no. 3, pp. 598–605, Sep. 2010.
[89] Z. Zhang, W. Wu, J. Yuan, and D.-Z. Du, "Breach-free sleep-wakeup scheduling for barrier coverage with heterogeneous wireless sensors," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2404–2413, Oct. 2018.
[90] Y. Liu, A. Liu, N. Zhang, X. Liu, M. Ma, and Y. Hu, "DDC: Dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 131, pp. 16–27, Apr. 2019.
[91] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "A robust and anonymous patient monitoring system using wireless medical sensor networks," *Future Gener. Comput. Syst.*, vol. 80, pp. 483–495, Mar. 2018.
[92] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4957–4968, Sep. 2019.
[93] S. Kumar, V. Kumar, O. Kaiwartya, U. Dohare, N. Kumar, and J. Lloret, "Towards green communication in wireless sensor network: GA enabled distributed zone approach," *Ad Hoc Netw.*, vol. 93, Oct. 2019, Art. no. 101903.
[94] H. Çam, S. Özdemir, P. Nair, D. Muthuavinashiappan, and H. Ozgur Sanli, "Energy-efficient secure pattern based data aggregation for wireless sensor networks," *Comput. Commun.*, vol. 29, no. 4, pp. 446–455, Feb. 2006.
[95] A. K. Sutrala, A. K. Das, N. Kumar, A. G. Reddy, A. V. Vasilakos, and J. J. Rodrigues, "On the design of secure user authenticated key management scheme for multigateway-based wireless sensor networks using ECC," *Int. J. Commun. Syst.*, vol. 31, no. 8, p. e3514, May 2018.
[96] Y. Yang, D. He, N. Kumar, and S. Zeadally, "Compact hardware implementation of a SHA-3 core for wireless body sensor networks," *IEEE Access*, vol. 6, pp. 40128–40136, 2018.
[97] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, and Z. Cai, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, Jan. 2020.
[98] A. Latha, S. Prasanna, S. Hemalatha, and B. Sivakumar, "A harmonized trust assisted energy efficient data aggregation scheme for distributed sensor networks," *Cognit. Syst. Res.*, vol. 56, pp. 14–22, Aug. 2019.
[99] T. Shu, W. Liu, T. Wang, Q. Deng, M. Zhao, N. N. Xiong, X. Li, and A. Liu, "Broadcast based code dissemination scheme for duty cycle based wireless sensor networks," *IEEE Access*, vol. 7, pp. 105258–105286, 2019.