International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

FBESSM: An Fuzzy Based Energy Efficient Sleep Scheduling Mechanism for Convergecast in Wireless Sensor Networks

Vishav Kapoor [1] and Daljeet Singh [2]

Submitted: 25/04/2023 **Revised**: 28/06/2023 **Accepted**: 07/07/2023

Abstract: The Wireless Sensor Networks (WSNS) of today are designed to collect and process vast amounts of data. Nodes near to the root have persistently high traffic levels and significant network congestion issues due to convergecast traffic. Congestion on the network and complicated procedures caused less efficiency even in the case of OSCAR method. Because of the close proximity of the nodes, collisions during transmission and energy consumption from duplicate data will be unavoidable. The process through which WSN gathers its data also contributes to its power usage. Clustering is superior to other data gathering methods for WSN because it manages duplicated data inside the network during in-network processing. To solve these challenges, a novel solution is provided in this work which is a fuzzy-based sleep scheduling mechanism (FBESSM) that would activate or sleep sensors as needed to decrease power usage. Performance indicators like as throughput, packet loss, the lifespan, energy consumption, and latency are being used to assess the effectiveness of the FBESSM approach as it has been implemented in NS-2. The FBESSM is more effective than OSCAR, as indicated from its superior performance across all of the relevant network metrics when it is being tested by varying the node count.

Keywords: *Convergecast, WSN, fuzzy, cluster head, network lifetime, energy consumption, OSCAR.*

1. Introduction

WSNs comprise of many tiny nodes. Each node is having a sensing component, a processing component, and a communicating component [1]. The sensing unit is equipped with several sensors for gathering data about the environment, including temperature, intensity of light and sound. The number of connected devices in use today has risen during the past decade [2]. Connected gadgets are pervasive in modern life, from automobiles to smart phones to watches and televisions [3]. By the use of the Time Slotted Channel Hopping (TSCH) [4] method developed in the IEEE 802.15.4, medium access for WSNs is coordinated in accordance with a time-frequency transmission cycle. In order for these networks to function, MAC protocols must be optimized for both less power usage and low delay or latency. There will be a dramatic rise in IoT data volume as the number of devices that may be connected grows. There is a high volume of information flowing through these nodes, which presents difficulties. Energy consumption and latency are two major problems in WSN, especially for industrial uses [5-6].

Reducing power consumption while maintaining a reliable wireless network and minimal data loss is a significant challenge in the field of wireless communications. As sensors often need to run for months or even years without human intervention or battery replacement, power consumption is a major challenge. In fact, users typically deploy hundreds or thousands of sensor nodes at once in an unplanned pattern throughout an inaccessible region, making repair either impossible or prohibitively expensive. Several applications, including industrial automation, rely on TSCH networks to be reliable and low-latency so that all data may be sent to and received from their intended destinations. Centralized scheduling methods [7], distributed scheduling algorithms [8-10], and autonomous scheduling algorithms [11-12] are some of the ways by which the allocation of resources is being done efficiently. With centralized scheduling, all of the network data is collected by one location, and from there the link schedules are determined. But, when the network's topology shifts, the root, which is already under heavy strain, takes longer to update the network's data, which can lead to inefficiency. The schedule in a distributed system is constructed by cooperative efforts of surrounding nodes.

The independent method was first used by Orchestra [13]. Orchestra has a number of advantages, such as a high delivery ratio and the adoption of simple scheduling rules; nevertheless, TSCH plans are determined for each node independently of traffic volume, which might lead to a significantly greater delay in communications. Each node's demand is distinct and fluctuates according on its position in the network. WSN applications use hierarchical cluster-based structure to efficiently gather data [14]. A coordinator, or Cluster Head (CH), manages

[1]*School of Electronics & Communication Engineering, Lovely Professional University, Phagwara, Punjab, India*
[2]*School of Electronics & Communication Engineering, Lovely Professional University, Phagwara, Punjab, India*
Vishavkapoor55@gmail.com

the cluster and sends data to the base station (BS). Changing CH selection balances network node energy utilization. With dense deployment, sensor nodes in nearby locations may show spatial and temporal correlations. While offering a fault-tolerant technique for data aggregation, transmission of same data would cause collisions and energy depletion, affecting network lifespan [15]. With identifiable objectives, evolutionary algorithms the cluster nodes comes with comparable monitoring outcomes as much as feasible. Lastly, cluster as many spatially-correlated sensor nodes as feasible. Hence, it can increase monitoring area data accuracy and lower CH transmission costs. But this procedure is complicated and time-consuming [16-17].

To reduce energy consumption and choose the fuzzy based clustering procedure, this paper present a similarity-dependent energy based sleep scheduling mechanism called Fuzzy Based Energy-Efficient Sleep Scheduling Mechanism (FBESSM). This new architecture is distinct from Orchestra and OSCAR in the manner that it utilizes fuzzy logic and energy metric to assign slots. The TSCH frequency for each node is really determined by Orchestra regardless of the traffic load, which has a significant influence on the latency and energy usage of interconnected communications. Because of their substantial load, nodes close to the root thus experience packet accumulation in their buffers, which may result in packet loss as a result of a buffer capacity overflow. The average OSCAR duty cycle is noted to be dependent on the node's location in the network, as does the latency. Because to the service cycle adjustment mechanism, this is less than Orchestra in both experimental configurations when traffic is light. As there is constant demand on the network and there are more operational nodes than that in Orchestra, its significance increases as network traffic load increases. But apart from slot allocation, better utilization of nodes is also important one. So, in this regard the proposed approach proves quite beneficial by using the concept of sleep mode with integration of fuzzy logic for better node selection and thus reduce the energy consumption to a larger extent.

The major highlights of this work are:

1. It highlights the process of existing OSCAR approach and the limitations associated with it.
2. It provides a new approach named FBESSM which is energy efficient one and based on sleep scheduling mechanism with integration of fuzzy.
3. It also highlights the comparative analysis of the FBESSM approach with the existing method OSCAR considering the network parameters.

2. Related Work

In a convergent network, every sensor node transmits data straight to the sink node. No data is being merged in the middle. Researchers find from these convergecast scenarios that in order for convergecast to be delay economical and data reliable in emergency circumstances, real-time data collecting from individual nodes is necessary, in opposed to data aggregation scenarios.

In the paper [18], radial coordination is suggested as a new way to make sure that convergecast in WSNs works well. The approach is designed to simplify the process by which several sensor nodes provide data to a sink node and the sink node obtains this data, all by selecting a single parent node for each sub tree in the WSN. The work in [19] provides a new algorithm for convergecast in WSNs called LEACH-C, which is created to lessen the latency and energy utilization of convergecast in WSNs. It is based on a cluster method to coordinate the gathering of data from multiple sensors and sending then to a sink node. The work in [20] proposes a new algorithm for Data Aggregation Convergecast (DAC) in WSNs called ST-DAC, which is based on a spanning tree approach to coordinate data aggregation and transmission from multiple sensor nodes to a sink node. This method targets DACs in WSNs to lessen their impact on latency and power consumption while maintaining steady data transmission. The authors in [21] propose a new scheduling algorithm for convergecast in WSNs called distributed minimal time convergecast scheduling (DMTCS). The algorithm is made to reduce the amount of time it takes to collect and send data from multiple sensor nodes to a sink node while making sure that no two nodes are in conflict

In [22], a technique called WirelessHART is proposed that allows the use of multiple packets and multiple channels. Unlike typical multi-channel TDMA protocols, WirelessHART uses channel hopping technology. The protocol Adaptive (A-TSCH) is proposed in [23] as an enhancement to the TSCH protocol used in IEEE 802.15.4e. To adapt to changing wireless circumstances, A-TSCH employs a hardware-based channel energy measurement-based dynamic blacklisting mechanism to prioritize less-interference communication channels. To maximize availability in PSNs used for IoT applications, the authors of [24] suggest the Data Collection and Aggregation with Selective Transmission (DGAST) method. Data from the sensors is collected on a regular basis by DGAST, which helps to preserve the sensors standby time. A Systematic Data Aggregation Model (CSDAM) for processing data in real-time is presented in [25]. At first, the network is organized into a cluster, consisting of both awake and inactive nodes, and a

Cluster-Head (CH) is taken depends on the sensors rankings in terms of their current energy status and their distance from the Base Station (BS). As discussed by the authors of [26], data aggregation is a useful method for extending the useful life of WSNs and decreasing their energy consumption. By merging and integrating data that is relevant and similar, unnecessary packets may be avoided and redundancy is reduced. In [27], the authors suggest an energy-efficient data-gathering format for use in WSNs. The sensors are gathered for probable rendezvous location selection using clustering techniques.

The authors offer multi-layer large data aggregation architecture and a Priority-Based Dynamic Data Aggregation (PDDA) approach in [28]. Since most current methods focus only on data aggregation at the central server level, the suggested PDDA methodology operates at the bottom layer of sensors. In [29], the authors offer a method named Hybrid Energy-Efficient Distributed Clustering (HEED) that chooses CHs on a periodic basis based on a combination of residual energy of node and a supplementary variable, like the node's closeness with respect to its peers or its grade. It provides a fairly consistent distribution of CHs over the network with little message overhead. Utilizing fuzzy logic, the distributed clustering method Cluster Head Election (CHEF) [30] is able to function with WSNs. At the beginning of each iteration, a list of possible CHs is compiled using a probability-based method, and from this list, a final CH is chosen using the nodes' residual energy and their local distance as deciding factors. More likely sensor nodes are chosen as CHs. Unfortunately, local distance, the input fuzzy variable, could not function for larger networks.

Specifically developed for wireless sensor networks, the Fuzzy Inference System (FIS) gives the basis for the computation of the competition radius of candidate CHs in the Multi Objective Fuzzy Clustering Algorithm (MOFCA) [31]. While making its calculations, the FIS factors in residual energy, sink distance, and density of node. Congestion is particularly bad for convergecast nodes close to the root. This issue is addressed in OSCAR [32], a revolutionary independent scheduling TSCH cell assignment technique built on Orchestra. OSCAR, in opposed to Orchestra, uses the nodes distance from the root to determine how many slots each node receives. A node's rank in the RPL routing protocol determines how many timeslots it receives, with higher-ranking nodes receiving more time slots overall. The main aim of the work [33] is to come up with a way for WSNs to collect data for as long as possible. They discovered that the energy utilization of two nodes is the same if they deliver the same amount of data packets, regardless of the length of the payloads within those packets. This is because most of the power is used for medium access control overhead and radio startup. By putting an emphasis on scheduling, e-TSCH-Orch [12] has also made an effort to lower Orchestra communication delay. A node notifies the receiver of the count of packets in its broadcast queue as it sends a data packet to the closest node. In [34], the authors demonstrate that Orchestra encounters issues while functioning under settings of constrained connection capacity, particularly in areas near to the network's root node.

In centralized systems, network data is sent to a central organization, which then schedules connections. In TASA, the schedule is made by a single node [35]. The network's layout and current traffic levels are taken into account to determine the schedule followed by each node. To reduce idle flow, CLS [7] distributes and discharges slots without altering the whole process at each CLS. DeTAS [37] deals with networks that have a lot of sink nodes. In [8], the authors suggest a method for scheduling that is based on waves. In [9], a proposal was made for DeAMON, which is a decentralized mechanism for 6TiSCH wireless networks. Later on, a distributed diverge cast scheduling method called DIVA [10] was made. Autonomous scheduling is better for the network than centralized and distributed scheduling because it uses less bandwidth. Orchestra [13] began auto-scheduling TSCH. Orchestra is a new way for TSCH to schedule jobs that uses basic criteria to speed up the rate of delivery. Escalator [11] is an autonomous scheduling method that uses Orchestra. It makes a sequential timeslot plan along the packet transmission pipeline to cut down on latency. Also, in [12], the authors tried to cut down on orchestra delays by changing the order in which they played. A distributed and scalable scheduling accessing approach was presented in [38], which can accommodate some mobility and reduces the amount of data lost in data-intensive sensor networks.

3. Oscar Approach

OSCAR [32] uses the rank idea presented through the routing mechanism for Low Power and Lossy Networks (RPL) [39] to compute proximity to the root. In a nutshell, RPL is an IPv6-based LLN that uses a proactive routing protocol. The RPL architecture is a DODAG, or a Destination Oriented Direct Acyclic Graph. The IPv6 edge router, that links nodes normally to outside community and from which it may receive management orders, is a common analogy for the DODAG root. A packet is sent to the root node from a node by way of a lower-rank neighbor node. An increasing rank is given to each node in the network as it moves away from the center.

Slots are allocated to nodes based on their needs, which is what the rescheduling algorithm is for. The busiest nodes are the ones closest to the root. This strategy, which is planned for incorporation into the scheduling of these nodes, proposes to regulate the number of time slots allocated to each node depending on the node's rank. Each network node's rank value is retrieved upon startup. The rank value in RPL is used to label classes.

OSCAR reallocates or releases up a slot if a rank's value shifts. As a result, as the node is further from the root, fewer slots are made available. On the other hand, when a node is brought closer to the root, additional slots are made available. On account of this, the class number and, by extension, the number of slots assigned to each node, will change as that node moves about in the network's topology. Each node awakens at the beginning of each time period to either send or receive packets in accordance with its class ID. Two pairs of neighboring layers also utilize distinct channels to reduce interference. Nodes away from the root, where rank matters, will utilize less power since fewer slots will be allotted to them. Those near the source will consume somewhat more than the average person, although even this increase is small. In addition to reducing latency, this approach has the added benefit of preventing network congestion at the root node.

Let C be the collection of scheduling cells in the network, where each cell has its own distinctive identifier. Let N signifies the overall count of nodes in the network, where each node has a distinct identifier. Let L be the collection of links between nodes, where each link is a tuple (i, j) indicating a link between nodes i and j.

The goal of the OSCAR Rescheduling Algorithm is to give each node i in N its own scheduling cell c in C so that the following conditions are met.

1. Each cell c in C is assigned to at most one node i in N.
2. Each node i in N is assigned exactly one cell c in C.
3. The number of cells in C is minimized.

There is a duty cycle that each node in the system follows. Sleep and waking times constitute each time period. The node's radio is disabled when it is in the sleep state, and it is activated for packet transmission or reception during the waking state. Orchestra's unicast slot frame has clearly defined regular receiving and broadcast time periods. Throughout their designated receiving time windows, all network nodes display a duty cycle of 100%. In terms of energy output, this is inefficient. There is a tradeoff among EED latency and power consumption brought by the slot frame length in OSCAR [32]. This method maintains a constant slot frame size to meet demand and minimize latency, even under severe traffic conditions.

OSCAR's Duty Cycle Adjustment Algorithm uses a mathematical representation to change the duty cycles of each node in the network so that there is less interference and less energy is used. The algorithm aims to assign duty cycles to nodes like that network throughput is maximized while maintaining a low level of interference. The mathematical representation of the Duty Cycle Adjustment Algorithm of OSCAR is as follows:

Input: N (set of nodes in the network), D (set of duty cycles that can be assigned to nodes).

Output: Di (duty cycle assigned to each node i)

Step 1: Initialization:

- Generate a set of initial duty cycles $D = \{d1, d2, ..., dn\}$, where each duty cycle is assigned to a node in N randomly.

Step 2: Interference Evaluation:

- Based on the number of collisions between transmissions, figure out how much interference is caused by transmissions from different nodes.
- Set C to be the set of collisions between transmissions in the network, where each collision is represented by a tuple (i, j) that shows a collision between nodes i and j.

Step 3: Duty Cycle Adjustment:

- Calculate the maximum available time for transmissions T in the network.
- For each node i in N, figure out the time needed for transmissions Ti and the sum(dj) of all other nodes' duty cycles (except for node i).
- Calculate the duty cycle assigned to node i, di, according to the following formula:

$$di = (Ti - sum(dj) + di) / T$$

where Ti is the time required for transmissions from node i, dj is the duty cycle assigned to node j, and di is the current duty cycle assigned to node i.

Step 4: Energy Consumption Evaluation:

- Calculate the energy used by each node *i* depends on the time required for transmissions *Ti* and the duty cycle assigned to node *i*, *di*.
- Set *E* is the energy consumed by nodes in the network, where each energy consumption is represented by a tuple (*i*, *ei*) indicating the energy consumed by node *i*.

Step 5: Termination:

- The algorithm terminates when a solution with the desired level of interference and energy consumption is found or a maximum count of iterations is reached.

3.1 Limitations of OSCAR

One of the primary challenges associated with OSCAR [32] is its complexity. The algorithm uses a sophisticated optimization technique to allocate scheduling cells for convergecast transmissions. The optimization process involves the use of a genetic algorithm that requires significant computational resources to execute. This complexity can be a major hurdle in the practical implementation of OSCAR, as it can limit the scalability of the algorithm and increase the overhead of the network.

Another challenge associated with OSCAR [32] is its dependence on accurate network information. The algorithm requires precise information about the arrangement of the network, including the location and availability of the nodes, the transmission range, and the quality of the wireless link between the nodes. Any errors or inaccuracies in this information can significantly impact the performance of OSCAR, leading to inefficient scheduling and suboptimal use of network resources. Another issue with OSCAR is related to its compatibility with existing protocols and standards. The algorithm was developed specifically for IEEE 802.15.4e TSCH networks, which may limit its applicability to other wireless network environments. Additionally, OSCAR requires modifications to the existing network infrastructure to enable the allocation of scheduling cells, which can be challenging and time-consuming to implement.

Another challenge associated with OSCAR is its vulnerability to network congestion and interference. The algorithm relies on a fixed set of scheduling cells for convergecast transmissions, which can become congested if multiple nodes tried to send data at the same time. It may cause packet loss, enhance latency, and decreased network performance. Additionally, interference from other wireless devices can impact the quality of the wireless link, which can further degrade the performance of OSCAR. Finally, the implementation of OSCAR requires significant coordination and collaboration among network nodes. The allocation of scheduling cells requires the exchange of information between nodes, which can introduce additional latency and communication overhead. Additionally, OSCAR requires synchronization among nodes to ensure that convergecast transmissions occur at the same time, which can be challenging in dynamic network environments.

4. Proposed Approach

The proposed approach has been presented in this section of the paper. First of all the CH selection process has been defined which is used in the proposed approach which is accompanied by the data transmission step as represented in the form of algorithms. The proposed approach is named as Fuzzy Based Energy-Efficient Sleep Scheduling Mechanism (FBESSM) where fuzzy logic is integrated for the CH selection based on sleep scheduling approach for efficient energy utilization. The proposed FBESSM method is having two algorithms in operation as depicted.

Algorithm CH_selection

Broadcast Hello message to all nodes

$[Y_i]_R = \{Y_j | r_{ij} = 1\}$ for individual Y_i , the element y_j is characterized into the same cluster when the fuzzy condition is satisfied

For each node N_i

Calculate the distance $D_n(i)$ based on RSSI, add Cluster Head Selection(CHS)

Elect CH based on D_{max}, $E_{res}(i)$

If ($[Y_i]_R$ and $Y_j | r_{ij} = 1$) are equivalent matrixes the nodes can be categorized directly;

Else

Matrix R being converted into an equivalent Boolean matrix by certain rules

Randomly select the candidate node (0,1) → X_0

If (X_0 < CHS (N_i)

Add candidate node to Cluster Head

$D_{CH}(i)$ = distance between Cluster head to candidate node

Broadcast (ID, E_{res} , D_{CH}, D_n) to other nodes;

Set $d_{CH}(i)$ to the candidate node

Else

Exit(0)

For each node N_i € neighbor_SET(S_i)

While receive CH_msg from node N_i

Add *di*;

If (receives all data samples)

Forward the aggregated result and wakeup the dormant nodes

If S_i € Candidate_CH && find minimum distance of different cluster head nodes

Add node N_i to CH_Set;

Else

If ($E_{res}(j) > E_{res}(i)$)

Broadcast highest residual energy node to candidate selection

Else

Send Control message to neighbor set using Matrix R

Modify the sleep timer's interval and the status flag, then send CHED_MSG_ACK

End If

End if

End if

End While

End for

The algorithm mentioned above is a CH selection algorithm for WSNs. The aim of the algorithm is to elect CHs that will efficiently manage data collection and aggregation tasks in a multi-hop convergecast communication scenario using the fuzzy based matrix R. The algorithm begins by broadcasting Hello messages to all nodes in the network, establishing communication among them. For every node Ni in the network, the algorithm calculates the distance $D_n(i)$ depends on the Received Signal Strength Indicator (RSSI) and adds a Cluster Head Selection (CHS) factor. The CHS factor is used as a threshold for candidate node selection in the later steps of the algorithm. The next step is to elect the CH depend on the maximum value of $D_n(i)$ and the residual energy $E_{res}(i)$ of the node. The CH for a given round is the node with the greatest total value.

The algorithm then checks if the nodes can be categorized directly using a fuzzy condition. If not, a Boolean matrix is created using certain rules. The algorithm then randomly selects a candidate node using a value between 0 and 1, denoted as X_0. If X_0 is less than the CHS value of node N_i, the algorithm proceeds with adding the candidate node to the CH. The distance $D_{CH}(i)$ between the CH and the candidate node is calculated, and the information (ID, E_{res}, D_{CH}, D_n) is broadcasted to other nodes, where ID represents the identifier of the candidate node. The algorithm also sets $d_{CH}(i)$ to be the candidate node. On the other hand, if X_0 is greater than or equal to the CHS value of node N_i, the algorithm exits and does not select any candidate node for the current round.

Next, the algorithm iterates through the neighbor_SET(S_i) of each node N_i. While receiving CH_msg (Cluster Head message) from node N_i, the algorithm adds the distance d_i to the list of CCHs in *CH_Set*. If S_i is in the Candidate_CH and the algorithm finds the minimum distance among different CH nodes, it adds node N_i to the *CH_Set*. However, if S_i is not in the Candidate_CH, the algorithm compares the residual energy $E_{res}(j)$ of node N_i with that of the current Cluster Head node *Eres(i)*. If $E_{res}(j)$ is greater than $E_{res}(i)$, the algorithm broadcasts the *ID* of the node having highest residual energy to the candidate selection process. Otherwise, the algorithm sends a Control message to the neighbor set using the matrix *R*. The algorithm continues to iterate through the nodes in the neighbor_SET(S_i) and selects CHs based on the criteria mentioned above until a termination condition is met.

Algorithm Data_transmission

For each CH_i

    For each timeslot

        If (receive control message from different sequence of CH (data, E_{res})

            Send ACK to all candidate set nodes;

        Else

            Send control message to CH (*id*, sequence,data)

        End if

End for

End for

For each non-cluster

    Send ACK (*ID*, data)

    Wait

    Receive control_ACK to all non-CH;

End for

The algorithm above provided is a data transmission algorithm for WSNs in a multi-hop convergecast communication scenario. The algorithm aims to facilitate efficient data transmission from non-CH nodes to CH nodes for further aggregation and processing. The algorithm operates in a time-slotted manner.

For each CH_i, the algorithm iterates through each timeslot. In each timeslot, the algorithm checks for control messages received from different sequences of CHs, which contain data and residual energy information. If control messages are received from different CHs with different sequences, the algorithm sends acknowledgement (ACK) messages to all candidate set nodes. This is done to confirm the successful reception of data by the CHs.

Alternatively, if control messages are not received from different sequences of CHs, the algorithm sends control messages to the CHs containing the *ID*, sequence, and data. This is the actual data transmission process from non-CH nodes to the CHs for aggregation and processing.

After completing the transmission to the CHs, the algorithm moves on to non-Cluster nodes. For each non-CH node, the algorithm sends an ACK message containing the *ID* and data. The non-CH nodes then wait for the reception of control_ACK messages from all the non-CH nodes. The purpose of this is to guarantee that all nodes outside of the CHs have sent their information to the CHs and received confirmation from other nodes outside of the CHs.

5. Performance Analysis

We describe the FBESSM's performance in this section. We use measures that we'll go through in this part to compare the performance of FBESSM and OSCAR. We have created and taken into account scenario while changing the node count to observe how the number of nodes affects performance.

5.1 Simulation Results

The implementation of the proposed method is being done through the NS-2 which is simulator. The nodes are deployed in a random manner in 1000m * 1000m * 1000m area. The counts of nodes deployed are varying from 50 to 200. The count of mobile nodes is taken 10 and transmission range of each node which is considered is 250m. The energy of each node initially is kept as 100 J. The time for simulation is 50 seconds for getting the precise results and also the packet interval for sending the hello packets has also been kept 100 seconds. Routing mechanism considered here is AODV [40]. The other simulation factors that are taken for the simulation of the proposed algorithm are depicted in Table 1.

Table 1: Simulation parameters for the proposed FBESSM approach

Simulation Parameter Taken	Value Taken
Topology	Random (100m)
Area for Deployment	1000m * 1000m * 1000m
Node count	50- 200
Range	250 m
MAC Protocol	802. 15. 4
Energy	100 J
Radio Propagation Model	Two Ray Ground
Antenna model	Omni Antenna
Number of mobile nodes	10
Routing protocol	AODV
Energy Usage	2w, 0.75w and 10mw
Data Packet Size	64 byte
Time for Simulation	50 s

5.2 Energy Model

The energy model is required for determining the energy requirement for the successful transmission and reception of data by the nodes involved in data communication. The quantity of energy spent (E_{trans}) by the transmitting node for transmitting 'T' bits to the receiving node placed at a distance "*Dist*" is given by the equation 1 and 2 [41].

$$E_{trans}\ (T,\ Dist) = T * E_{cons} + T * E_{fsm} * Dist^2 \text{ when } Dist < Dist_{thres} \quad (1)$$

$$E_{trans}\ (T,\ Dist) = T * E_{cons} + T * E_{mam} * Dist^4 \text{ when } Dist_{thres} \le Dist \quad (2)$$

Where E_{cons} signifies the energy consumed while transmitting single bit

E_{fsm} signifies the energy utilized under the free space model

E_{mam} signifies the energy utilized under the multipath amplifier model

Also, $Dist_{thres}$ is the threshold value of the distance which is computed by equation 3.

$$Dist_{thres} = \sqrt{\ E_{fsm}/\ E_{mam}} \quad (3)$$

Thus on the basis of threshold value ($Dist_{thres}$) the equation 1 or 2 is selected by the node.

Similarly, the amount of energy spent while receiving the data by the node at the receiver end is shown by the equation 4.

$E_{receiv} = T * E_{trans}$ (4)

5.3 Metrics for Evaluation

The factors on the basis of which analysis of the proposed scheme and its comparison with the existing approaches has been done are defined below.

1. *Packet Delivery Ratio (PDR)* = Σ(Aggregate of packets received by all destination node) /Σ(Aggregate packets send by all source node) (5)

2. *End -to-End Delay (EED)* = 1 /*n* Σi=1 ($Tr_i - Ts_i$) * 1000 [ms] (6)

Where, i = packet identity

Tr_i = time of reception

Ts_i = time of sending

n = successfully delivered packets count

3. *Throughput (TP)=(*recvdSize/(stopTime-startTime))*(8/1000) (7)

4. *Residual Energy(RE)* = Total Energy (*TE*)- Energy Consumed (*EC*) (8)

All of the above mentioned factors are important to analyze the efficiency of the given scheme and its comparison with respect to other approaches.

5.4 Comparative Evaluation

The comparative analysis of the proposed FBESSM technique with the existing OSCAR method has been done based on the metrics as mentioned by varying the node count from 50 to 250 here. First of all the OSCAR method has been implemented on the simulator with the simulation factors as mentioned in table 1 and then it is tested under the metrics as defined in the table 2 while varying the node count.

Table 2: Metric based evaluation for the OSCAR method

Node count	Delay	PDR	TP	EC	Network lifetime (NLT)	Packet loss
50	27.6554	0.8456	338.98	21.4567	45.678	123
100	26.2345	0.8467	341.50	39.5563	34.577	134
150	25.3212	0.8532	347.62	57.3221	26.458	156
200	24.543	0.8598	349.35	76.4567	21.578	178
250	24.542	0.8623	355.98	100.43	17.0872	202

Similarly in table 3 the proposed method FBESSM has been tested under the same simulation environment as defined in table 1 and it's being evaluated with the metrics as defined in table 3.

Table 3: Metric based evaluation for the proposed FBESSM method

Node count	Delay	PDR	TP	EC	Network lifetime (NLT)	Packet loss
50	26.6554	0.8489	352.432	21.321	46.995	120
100	25.2345	0.8543	352.554	38.454	35.665	122
150	24.3212	0.8632	353.443	56.4332	27.404	153

200	24.043	0.8690	355.667	75.433	21.5092	173
250	23.781	0.8721	358.443	99.445	18.455	197

In figure 1 the variation of EED when the node count is varied from 50 to 250 has been presented and the comparative analysis of FBESSM has been done with OSCAR. It has been depicted that FBESSM technique outperforms the OSCAR method for EED at all instances. Delay is reduced with OSCAR, and this is explained by the reason that the congested nodes are given extra time slots to clear out their congestion. But the EED in FBESSM is less on account of the fuzzy matrix used for the efficient CH selection based on sleep scheduling approach as less overhead while communication. FBESSM reduces EED by 0.63%, 2.78%, 3.94%, 2.03% and 3.10% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

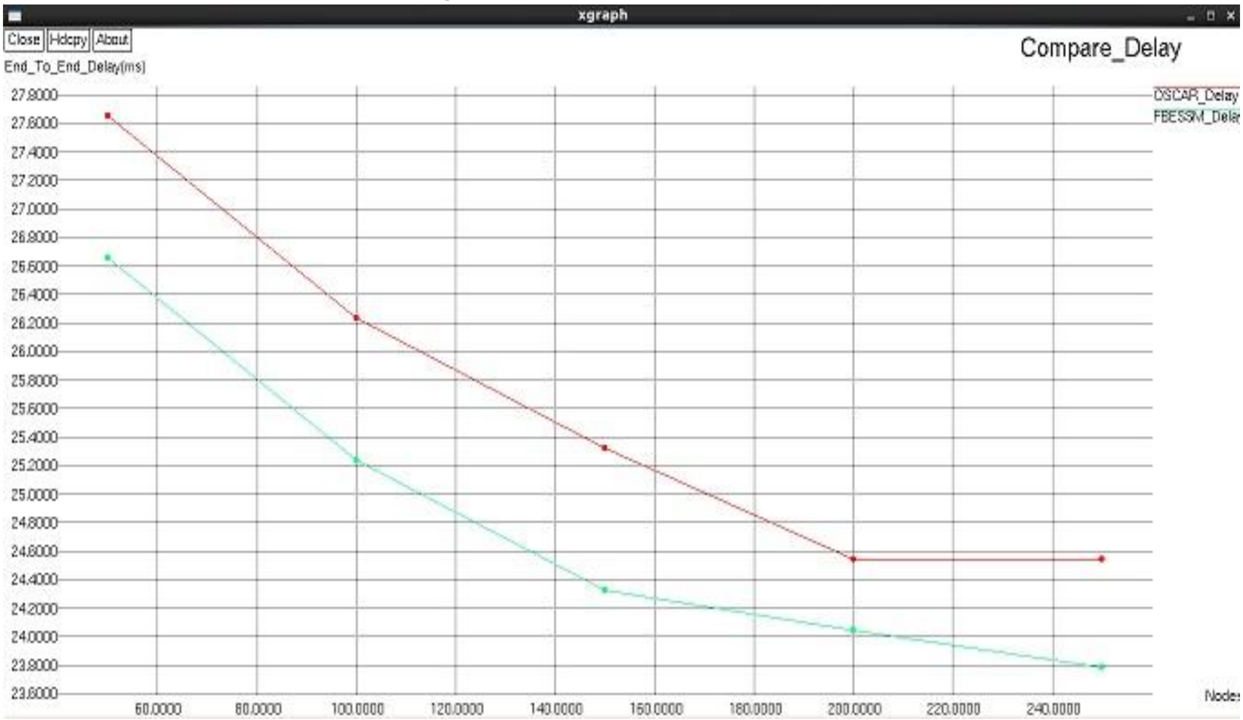

Fig 1: Variation of EED under the node count

In figure 2 the variation of EC when the node count is varied from 50 to 250 has been presented and the comparative analysis of FBESSM has been done with OSCAR. It has been depicted that FBESSM technique is performing slightly better than the OSCAR method. Since certain nodes are less engaged than others, OSCAR has a method for reducing the number of slots allotted to inactive nodes, which results in less EC overall. On the other hand FBESSM is using the concept of sleep scheduling of nodes which are not participating in the CH selection process and thus reduces the EC to large extent. FBESSM reduces EC by 3.61%, 3.81%, 1.55%, 1.33% and 0.98% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

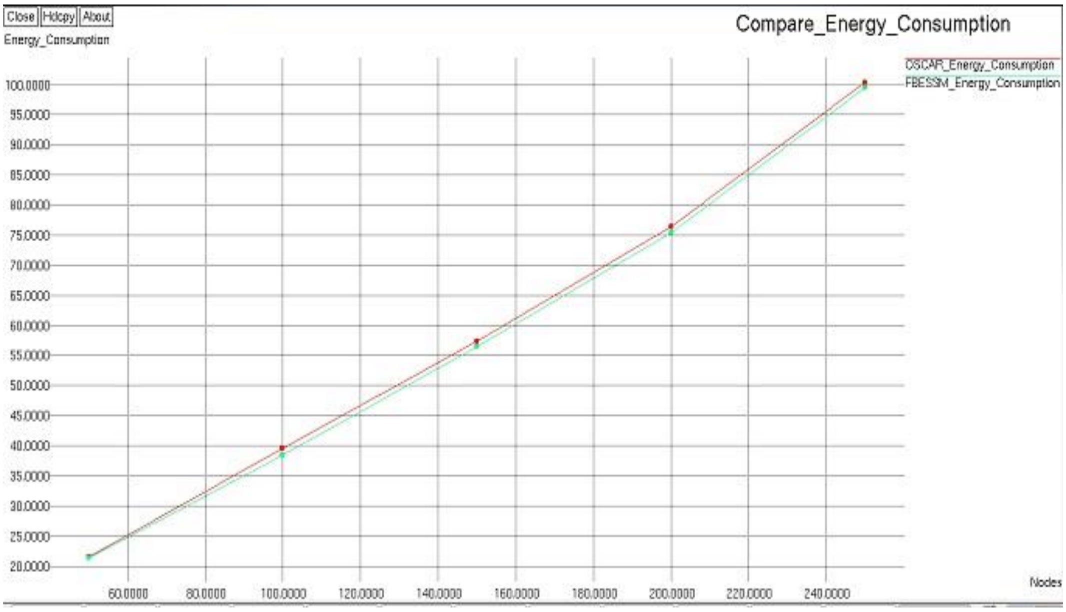

Fig 2: Variation of EC under the node count

In figure 3 the variation of Network Life Time (NLT) when the node count is varied from 50 to 250 has been presented and the comparative analysis of FBESSM has been done with OSCAR. Because it takes into account a number of factor including RE and distance to BS when choosing a CH node, FBESSM consistently outperforms competing algorithms across all test situations. FBESSM increases EC by 2.88%, 3.14%, 3.57%, 0.31% and 0.80% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

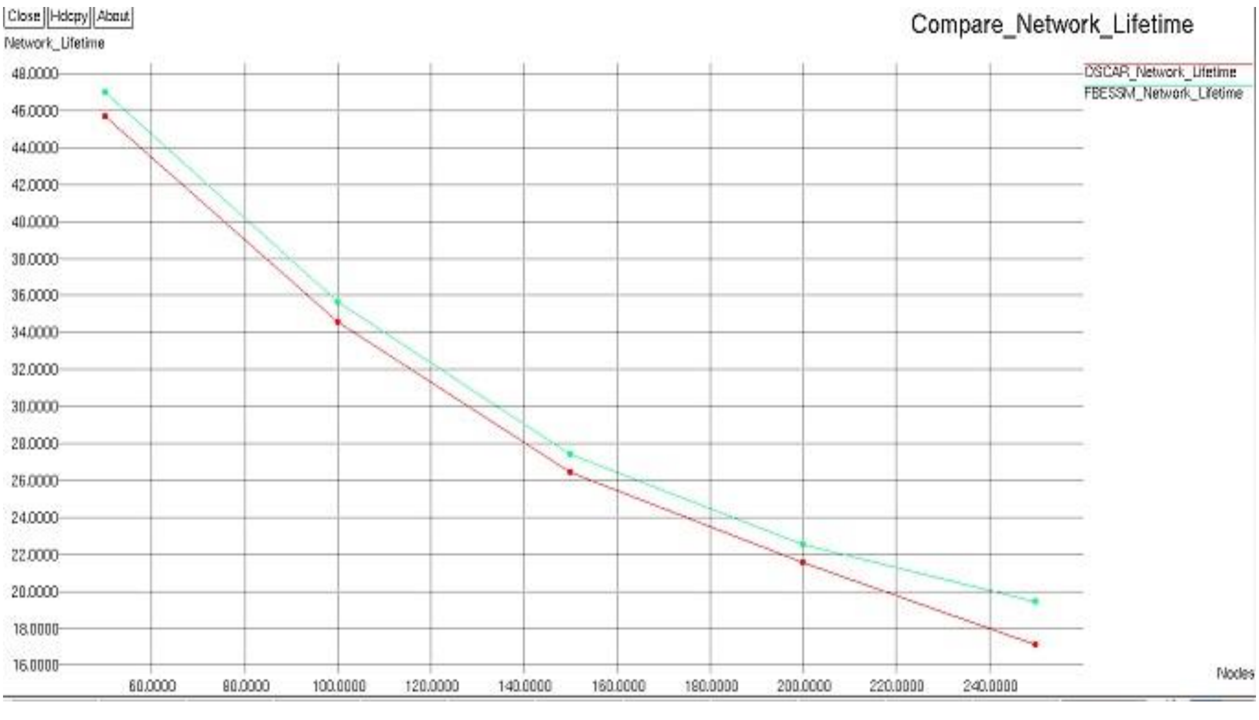

Fig 3: Variation of NLT under the node count

From the figure 4 it is depicted that the PDR of FBESSM is quite high in contrast to OSCAR method when the node count is increased from 50 to 250. The reason is that the FBEESSM incorporates fuzzy logic-based decision-making mechanisms to optimize energy consumption by controlling the sleep/wake cycle of the nodes. This allows the nodes to stay in sleep mode for longer periods of time, which reduces the amount of communication overhead and collisions in the network. As a result, FBEESSM may achieve higher PDRs by reducing the probability of packet collisions and interference. FBESSM has higher PDR of 0.39%, 0.89%, 1.17%, 1.07% and 1.13% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

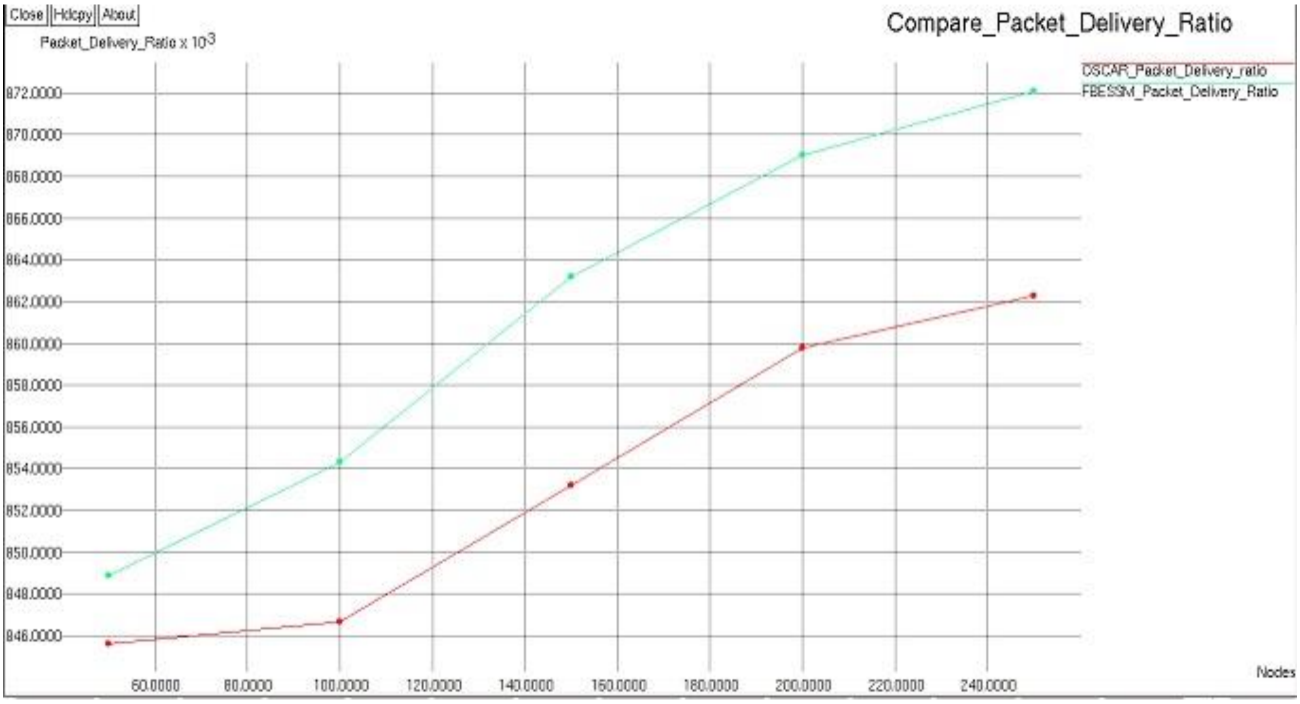

Fig 4: Variation of PDR under the node count

In the similar fashion the variation of packet loss with respect to the node count increment has been shown in figure 5 for the FBESSM in contrast to OSCAR. FBEESSM involves organizing the nodes into clusters, with each cluster having a CH responsible for communication with other clusters. This approach reduces the communication overhead and EC in the network, which can help reduce the probability of packet loss. FBESSM has lower packet loss by 2.4%, 8.9%, 1.92%, 2.8% and 2.4% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

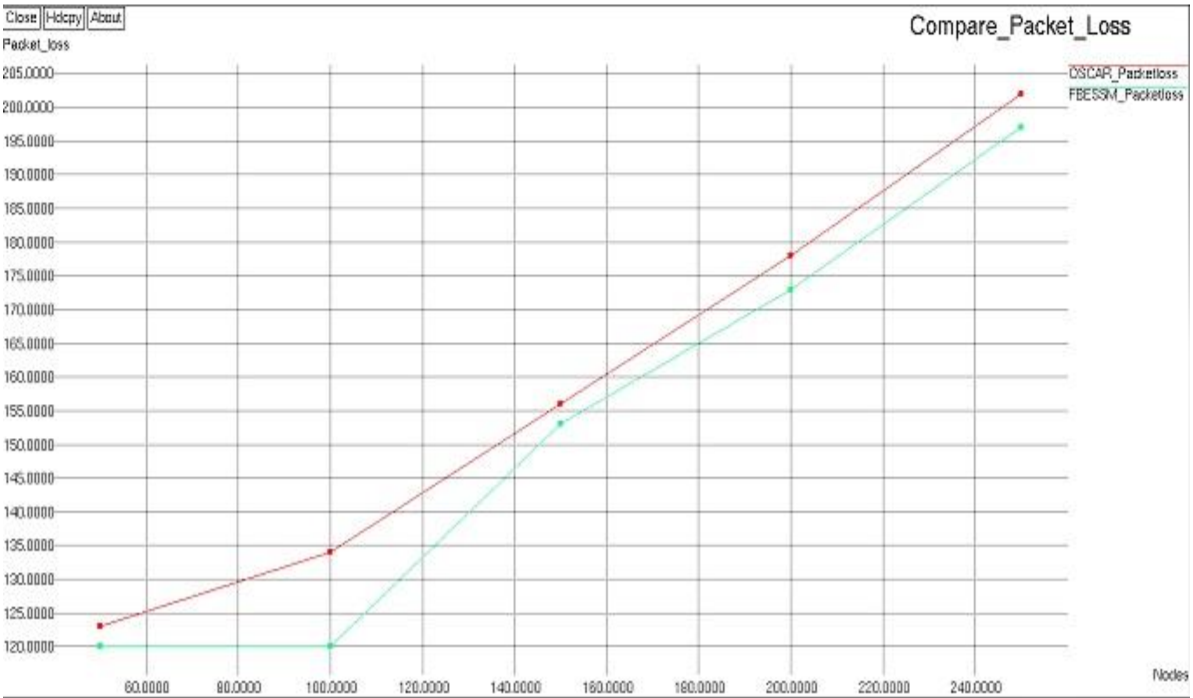

Fig 5: Variation of packet loss under the node count

Also, in figure 6 throughput of OSCAR is compared with FBESSM by varying the node count from 50 to 250. It is clearly depicted that FBESSM is having higher throughput in contrast to OSCAR method. This is because the combination of fuzzy-based decision-making mechanisms and clustering, in FBEESSM can contribute to higher throughput compared to OSCAR, which relies on a different set of mechanisms to manage communication in the network. FBESSM has higher throughput by 3.9%, 3.2%, 1.6%, 1.8% and 0.69% in comparison to OSCAR under node count 50, 100, 150, 200 and 250 respectively.

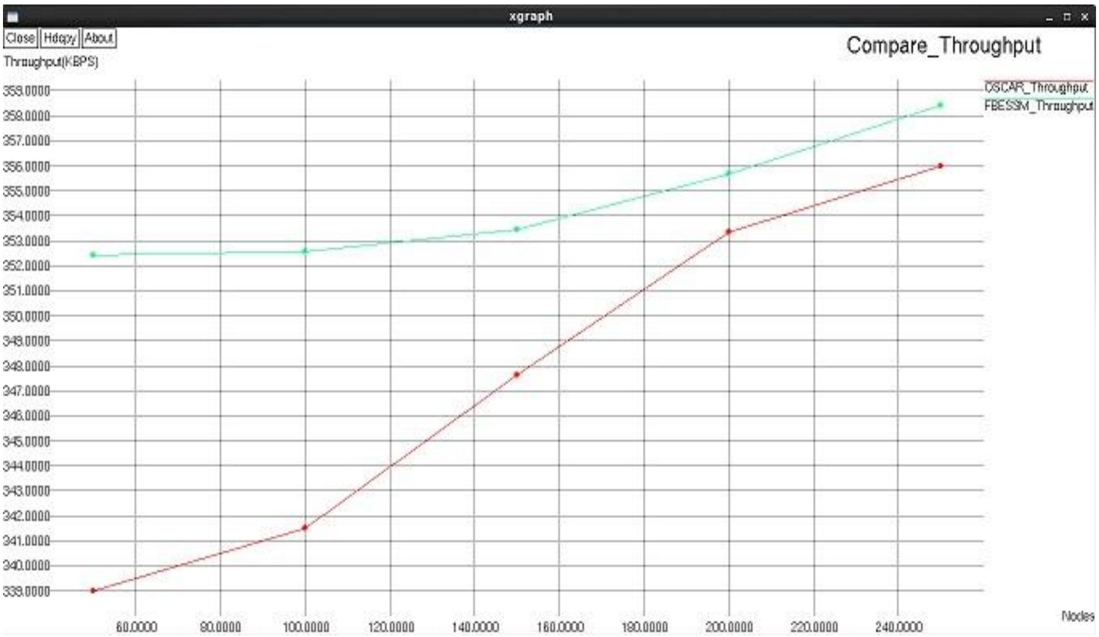

Fig 6: Variation of throughput under the node count

6. Conclusion

This paper proposed a fuzzy based cluster selection approach which is energy efficient also through sleep mode mechanism for convergecast WSN. The method has majorly two prominent phases. The first phase is the utilization of fuzzy matrix to selects a CH based on residual energy and distance. The second phase is the data transmission method which involves the control messages. This phase facilitates data gathering and aggregation by transmitting data efficiently in a multi-hop convergent communication environment. When comparing FBESSM with OSACR on EED, performance analysis highlighted that FBESSM is superior. Configurations of the FBESSM and OSCAR under different metrics and with node counts ranging from 50 to 250 have been explored to get insight into their performance. While both FBESSM and OSCAR assign slots, the experimental findings reveal that FBESSM's allocation of slots based on a fuzzy matrix leads in greater overall performance. In addition, simulations with varying numbers of nodes have shown FBESSM's superior performance on large networks. Increases in network capacity and associated traffic load lead to improved network lifetime and packet delivery under FBESSM compared to OSCAR. It is also observed that the FBESSM significantly decreases the EC by using the notion of sleep scheduling for nodes that are not involved in the CH selection process. When comparing EC usage with OSCAR, FBESSM achieves 3.61 %, 3.81 %, 1.55 %, 1.33 %, and 0.98 % lower EC against the node counts of 50, 100, 150, 200, and 250, respectively. Similar kind of results has been inferred for the other metric evaluation like packet loss, throughput etc. In the future integration of machine learning techniques for efficient CH selection and data transmission will be worked upon on the proposed technique.

Reference

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, *38*(4), 393-422.

[2] Balazka, D., & Rodighiero, D. (2020). Big data and the little big bang: an epistemological (R) evolution. *Frontiers in big Data*, *3*, 31.

[3] Ibarra-Esquer, J. E., González-Navarro, F. F., Flores-Rios, B. L., Burtseva, L., & Astorga-Vargas, M. A. (2017). Tracking the evolution of the internet of things concept across different application domains. *Sensors*, *17*(6), 1379.

[4] Astrin, A. (2012). IEEE Standard for Local and metropolitan area networks part 15.6: Wireless Body Area Networks. *IE EE Std 802.15. 6*.

[5] Harb, H., & Makhoul, A. (2017). Energy-efficient sensor data collection approach for industrial process monitoring. *IEEE Transactions on Industrial Informatics*, *14*(2), 661-672.

[6] Qin, Z., Wu, D., Xiao, Z., Fu, B., & Qin, Z. (2018). Modeling and analysis of data aggregation from

convergecast in mobile sensor networks for industrial IoT. *IEEE Transactions on Industrial Informatics*, *14*(10), 4457-4467.

[7] Choi, K. H., & Chung, S. H. (2016). A new centralized link scheduling for 6TiSCH wireless industrial networks. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 16th International Conference, NEW2AN 2016, and 9th Conference, ruSMART 2016, St. Petersburg, Russia, September 26-28, 2016, Proceedings 16* (pp. 360-371). Springer International Publishing.

[8] Soua, R., Minet, P., & Livolant, E. (2016). Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15. 4e TSCH networks. *Transactions on Emerging Telecommunications Technologies*, *27*(4), 557-575.

[9] Aijaz, A., & Raza, U. (2017). DeAMON: A decentralized adaptive multi-hop scheduling protocol for 6TiSCH wireless networks. *IEEE Sensors Journal*, *17*(20), 6825-6836.

[10] Demir, A. K., & Bilgili, S. (2019). DIVA: a distributed divergecast scheduling algorithm for IEEE 802.15. 4e TSCH networks. *Wireless Networks*, *25*, 625-635.

[11] Oh, S., Hwang, D., Kim, K. H., & Kim, K. (2018). Escalator: An autonomous scheduling scheme for convergecast in TSCH. *Sensors*, *18*(4), 1209.

[12] Rekik, S., Baccour, N., Jmaiel, M., Drira, K., & Grieco, L. A. (2018). Autonomous and traffic-aware scheduling for TSCH networks. *Computer Networks*, *135*, 201-212.

[13] Duquennoy, S., Al Nahas, B., Landsiedel, O., & Watteyne, T. (2015, November). Orchestra: Robust mesh networks through autonomously scheduled TSCH. In *Proceedings of the 13th ACM conference on embedded networked sensor systems* (pp. 337-350).

[14] Arunraja, M., Malathi, V., & Sakthivel, E. (2015). Energy conservation in WSN through multilevel data reduction scheme. *Microprocessors and Microsystems*, *39*(6), 348-357.

[15] Tyagi, S., & Kumar, N. (2013). A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks. *Journal of Network and Computer Applications*, *36*(2), 623-645.

[16] Vanus, J., Belesova, J., Martinek, R., Nedoma, J., Fajkus, M., Bilik, P., & Zidek, J. (2017). Monitoring of the daily living activities in smart home care. *Human-centric Computing and Information Sciences*, *7*(1), 1-34.

[17] Zheng, H., Guo, W., & Xiong, N. (2017). A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *48*(12), 2315-2327.

[18] Huang, Q., & Zhang, Y. (2004, November). Radial coordination for convergecast in wireless sensor networks. In *29th annual IEEE international conference on local computer networks* (pp. 542-549). IEEE.

[19] Upadhyayula, S., Annamalai, V., & Gupta, S. K. (2003, December). A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks. In *GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489)* (Vol. 6, pp. 3525-3530). IEEE.

[20] Upadhyayula, S., & Gupta, S. K. (2007). Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks. *Ad Hoc Networks*, *5*(5), 626-648.

[21] Gandham, S., Zhang, Y., & Huang, Q. (2006, July). Distributed minimal time convergecast scheduling in wireless sensor networks. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)* (pp. 50-50). IEEE.

[22] Zhang, H., Soldati, P., & Johansson, M. (2009, June). Optimal link scheduling and channel assignment for convergecast in linear WirelessHART networks. In *2009 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks* (pp. 1-8). IEEE.

[23] Du, P., & Roussos, G. (2012, September). Adaptive time slotted channel hopping for wireless sensor networks. In *2012 4Th computer science and electronic engineering conference (CEEC)* (pp. 29-34). IEEE.

[24] Rhee, I., Warrier, A., Min, J., & Xu, L. (2006, May). DRAND: distributed randomized TDMA scheduling for wireless ad-hoc networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing* (pp. 190-201).

[25] Wang, Y., & Henning, I. (2007, September). A deterministic distributed TDMA scheduling algorithm for wireless sensor networks. In *2007 International Conference on Wireless*

Communications, Networking and Mobile Computing (pp. 2759-2762). IEEE.

[26] Lee, W. L., Datta, A., & Cardell-Oliver, R. (2008). FlexiTP: a flexible-schedule-based TDMA protocol for fault-tolerant and energy-efficient wireless sensor networks. *IEEE transactions on parallel and distributed systems*, *19*(6), 851-864.

[27] Testa, A., Cinque, M., Coronato, A., De Pietro, G., & Augusto, J. C. (2015). Heuristic strategies for assessing wireless sensor network resiliency: an event-based formal approach. *Journal of Heuristics*, *21*, 145-175.

[28] Zeng, B., & Dong, Y. (2014). A collaboration-based distributed TDMA scheduling algorithm for data collection in wireless sensor networks. *Journal of Networks*, *9*(9), 2319.

[29] Younis, O., & Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*, *3*(4), 366-379.

[30] Kim, J. M., Park, S. H., Han, Y. J., & Chung, T. M. (2008, February). CHEF: cluster head election mechanism using fuzzy logic in wireless sensor networks. In *2008 10th international conference on advanced communication technology* (Vol. 1, pp. 654-659). IEEE.

[31] Sert, S. A., Bagci, H., & Yazici, A. (2015). MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks. *Applied Soft Computing*, *30*, 151-165.

[32] Osman, M., & Nabki, F. (2021). OSCAR: An optimized scheduling cell allocation algorithm for convergecast in IEEE 802.15. 4e TSCH networks. *Sensors*, *21*(7), 2493.

[33] John, J., Kasbekar, G. S., & Baghini, M. S. (2021). Maximum lifetime convergecast tree in wireless sensor networks. *Ad Hoc Networks*, *120*, 102564.

[34] Piyare, R., Oikonomou, G., & Elsts, A. (2020). TSCH for long range low data rate applications. *IEEE Access*, *8*, 228754-228766.

[35] Palattella, M. R., Accettura, N., Dohler, M., Grieco, L. A., & Boggia, G. (2012, September). Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks. In *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC)* (pp. 327-332). IEEE.

[36] Choi, K. H., & Chung, S. H. (2016). A new centralized link scheduling for 6TiSCH wireless industrial networks. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 16th International Conference, NEW2AN 2016, and 9th Conference, ruSMART 2016, St. Petersburg, Russia, September 26-28, 2016, Proceedings 16* (pp. 360-371). Springer International Publishing.

[37] Accettura, N., Palattella, M. R., Boggia, G., Grieco, L. A., & Dohler, M. (2013, June). Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things. In *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)* (pp. 1-6). IEEE.

[38] Lin, C. K., Zadorozhny, V., Krishnamurthy, P., Park, H. H., & Lee, C. G. (2010). A distributed and scalable time slot allocation protocol for wireless sensor networks. *IEEE Transactions on mobile computing*, *10*(4), 505-518.

[39] Tsvetkov, T. (2011). RPL: IPv6 routing protocol for low power and lossy networks. *Sensor nodes–operation, network and application (SN)*, *59*(2).

[40] Khem, D. ., Panchal, S. ., & Bhatt, C. . (2023). An Overview of Context Capturing Techniques in NLP. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 193–198. https://doi.org/10.17762/ijritcc.v11i4s.6440

[41] Prof. Nitin Sherje. (2017). Phase Shifters with Tunable Reflective Method Using Inductive Coupled Lines. International Journal of New Practices in Management and Engineering, 6(01), 08 - 13. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/50