International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Idiosyncratic Fuzzy Low-Energy Adaptive Clustering Hierarchy

R. Sridevi[1]* **V. Sinthu Janita Prakash[1]**

[1]*Department of Computer Science, Cauvery College for Women (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli Tamilnadu, India*
* Corresponding author's Email: sridevirajaram1981@gmail.com

Abstract: Clustering is one of the important functionalities of networking, which has a dominant impact on the performance in terms of Throughput, packet delivery ratio and communication delays. The challenge in performing the clustering process is more challenging for heterogeneous networks. The divergence between the member nodes is highly variable for a network that contains IoT nodes. As the result of vast availability of different IoT hardware, organizing them under one roof is a critical task which is very vital too. This work is intended to develop a new hierarchical based clustering to improve the performance and the lifetime of the IoT / Wireless sensor network nodes. Three legacy functional blocks such as idiosyncratic fuzzy energy estimator, cumulative fuzzy energy hierarchy builder and energy based cluster head exchanger are integrated in this work and submitted here as "Idiosyncratic Fuzzy Low-Energy Adaptive Clustering Hierarchy (IFLEACH)". An industrial standard network simulator is used to evaluate the performance of IFLEACH in terms of throughput, packet delivery rate, latency, end-to-end delay and energy consumption by simulating the real-worlds IoT network environment replica. An achievement around 11 kbps throughput and 4 % security level are obtained by the proposed method in high density network.

Keywords: Internet-of-things (IoT), Network clustering, Heterogeneous networks, Energy adaptive clustering, Fuzzy logic.

1. Introduction

Emerging communication developments open the gates of several newfangled manifestos towards the essential needs of this century such as responsible resource conservation, pollution controls, organized cultivation and smart city management [1]. The applications of state-of-the art communication technologies are merely limitless. The role of wireless sensor nodes and networks are inevitable in the mentioned applications [2, 3]. The introduction of IoT dramatically reduces the infrastructure cost in Wireless sensor network deployments. It is quite evident that the cost of wireless sensor nodes and other components are significantly reduced after the commercial presence of IoT devices in the market [4]. Now-a-days there is always a unique IoT device available for every application based on the needs.

The introduction of several IoT devices with different hardware architectures makes the networking omnipresent in all domains [5]. The word 'smart' can be preceded almost every electronic gadget available these days [6, 7] such as smart watch, smart TV, smart fridge, smart pen etc. Organizing these devices under a single network is a challenging job because excluding the uniformity in following IEEE 802.11 communication standard, remaining all other architectures such as embedded chips and their driver software are exceedingly different between the IoT Nodes. A tiny 8-bit brained device and a rigid non-stop work rig both can be a part of a single network at present [8, 9]. This multifariousness between the nodes has to be handled very carefully to avoid the performance glitches. The power sources of these devices also differ based on the purpose of the devices.

Clustering is an elementary operation in networking which is responsible for interconnect different devices into a single entity [10]. The clustering procedure should provide a smooth bridge between the configurational deviations between the devices. Energy efficiency is one of the most important characteristics of modern networks because most of the network member nodes are battery powered while considering wireless sensor networks in particular [11, 12]. A perfect clustering procedure should provide sufficient security to the network without increasing much power consumption as well as without bothering the performance of the network. The challenge behind network clustering is, parameters such as security, power consumption and network performance are tied together in most cases. For example, if an algorithm targets on security improvements, obviously bigger authentication keys are involved which has the impact on power consumption and also on performance of the devices with limited computations resources.

IFLEACH is designed in a way to constitute clusters based on exclusive multiparameter considerations to achieve higher performance without consuming more energy.

2. Existing methods

There are several existing methods in practice for IoT clustering. In this section, some of the best existing IoT clustering procedures are evaluated in terms of performance, adaptiveness and to perceive the merits, methodologies and limitations. Based on this evaluation study, the requirement for a new clustering method is legitimated. The methods taken into the evaluation are an affinity propagation-based self-adaptive clustering method for wireless sensor networks [13], EESRA: Energy efficient scalable routing algorithm for wireless sensor networks [14], energy-efficient selection of cluster headers in wireless sensor networks [15] and residual energy-based cluster-head selection in WSNs for IoT application [16].

2.1 An affinity propagation-based self-adaptive clustering method for wireless sensor networks [APSA]

K-medoid and affinity propagation methods are integrated in APSA work to improve the clustering performance. This integration is accustomed to reduce the probability of uneven cluster head distribution and unbalanced power consumption. Affinity propagation is used to calculate the number of cluster heads and their placements for K-medoid. ASPA consists of network model and energy model and the iterative process is used to optimize the parameters of the models. A dedicated algorithm is introduced in ASPA for initialization phase, setup phase and for communication phase. Experiments are carried out using MATLAB simulation based on different number of sensor nodes in the environment.

Table 1. Existing methods, advantages and limitations

Author	Work	Methodology	Advantages	Limitations
J. Wang [13]	An Affinity Propagation-Based Self-Adaptive Clustering Method for Wireless Sensor Networks	K-Medoid and Affinity Propagation	Energy efficiency	Lesser performance
E. F. Ahmed Elsmany [14]	EESRA: Energy Efficient Scalable Routing Algorithm for Wireless Sensor Networks	Three-layer hierarchical Algorithm	Load balancing, Energy Efficiency	Lesser Throughput and Packet delivery rate
AdemFanos Jemal [15]	Energy-efficient selection of cluster headers in wireless sensor networks	K-Means and Dispersed address assignment	Performance, Energy efficiency	Lower Range and less number of nodes
T. M. Behera [16]	Residual Energy-Based Cluster-Head Selection in WSNs for IoT Application	Cluster head selection and rotation	Performance, Energy efficiency	Lower Range and very less number of nodes

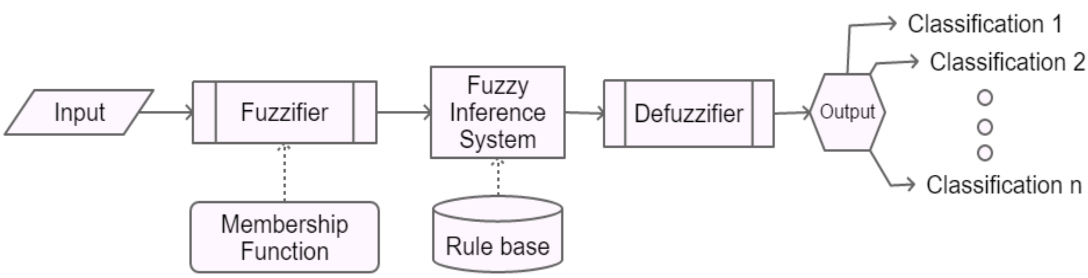


Figure. 1 Fuzzy logic flow

Energy efficiency is the observed advantage of ASPA method whereas declining performance parameters are identified as the limitations of this work.

2.2 Energy efficient scalable routing algorithm for wireless sensor networks [EESRA]

The standard LEACH algorithm suffers the issue of performance decay over network size. A new energy-efficient scalable routing algorithm is introduced in EESRA work to overcome the mentioned issue of default LEACH. EESRA improves the overall lifespan of the network by achieving optimal routing between the nodes. A three-layer hierarchical algorithm is used to construct the primary module of EESRA. There are two phases exist in the EESRA method, they are setup phase and steady-state phase. The setup phase is to initialize the genesis random routing and the steady-state phase is used to converge the routings towards the optimum. The performance of EESRA is evaluated by the simulation with 100, 200, 300 and 400 nodes and the performance of EESRA is promising in terms of load balancing and energy efficiency in large scale networks. Achievement of load balancing and energy efficiency in large scale networks is the learned advantage of EESRA method whereas, compromised network performance in terms of throughput and packet delivery rate is the noted disadvantage.

2.3 Energy-efficient selection of cluster headers in wireless sensor networks [EESCH]

EESCH prefers the placement of routers to improve the energy-efficiency. The cluster wise router placement selection is performed by K-means algorithm. Nodes with adequate battery power from each cluster are selected as the cluster heads and work as the inter-cluster communication hubs. Dispersed address assignment scheme is used in EESCH method in which the gateway is responsible for number of child nodes, number of routers and maximum number of hops to reach the cluster head of every cluster. OPNET simulator is used to evaluate the performance of EESCH in terms of throughput, end-to-end delay and packet delivery rate. Achievement of mentioned parameters for low range network environments is the noted advantage of this method whereas, applicability of EESCH in more than 500 square meters is not addressed – which is known as the limitation of this work.

2.4 Residual energy-based cluster-head selection in WSNs for IoT application [RECHS]

RECHS work focuses on the processes of cluster head selection and substitution. More than one cluster heads for a cluster are selected based on the initial energy. In the beginning a single cluster head in each cluster is active at a time based on the initial energy and the cluster head substitution or rotation process is performed based on the residual energy. Whenever a cluster head is about to drain a significant amount of power, the role of the cluster head is assigned to another eligible node to be a cluster head. MATLAB is used to evaluate the performance of RECHS for the simulation environment of 100 square meters with 100 nodes. Based on the observed results both Throughput and the lifetime of the network is improved by about 60% which is the advantage of the device. Application environment of 100 square meters is a short range for modern IoT applications and the performance of RECHS is limited to that range is the limitation of this method.

A brief summary of existing works, methodologies used, advantages and limitations are given in Table 1.

3. Related works

Some rudimentary concepts from fuzzy logic and round-robin scheduling are used as the endorsements to build the functional modules of the proposed system. A gentle introduction about fuzzy logic and round-robin scheduling is given here to support the explanation of the proposed modules apparently.

3.1 Fuzzy logic

Fuzzy logic is frequently used where boolean results are not applicable, at the same time, more accurate analog results are not required. High precision floating point readings take more measurement time which are not very essential in many real-world applications. So binary type measurement lacks the degrees of truth measurements. Fuzzy logic is a natural option by which it is possible to make measurements swiftly with certain degrees of truth.

A typical fuzzy model consists of four modules, they are fuzzifier, rule base, inference engine and defuzzifier. The input values are processed by the fuzzifier which applies a membership function to determine the elements category. Fuzzified results are fed to the fuzzy inference system that operates based on the rule base. Then the defuzzification process is performed to determine the output for the given input. The flow of Fuzzy Logic is given in Fig. 1.

3.2 Round-robin scheduling

Round-robin scheduling (RRS) is widely used for

Received: December 2, 2022. Revised: March 14, 2023.

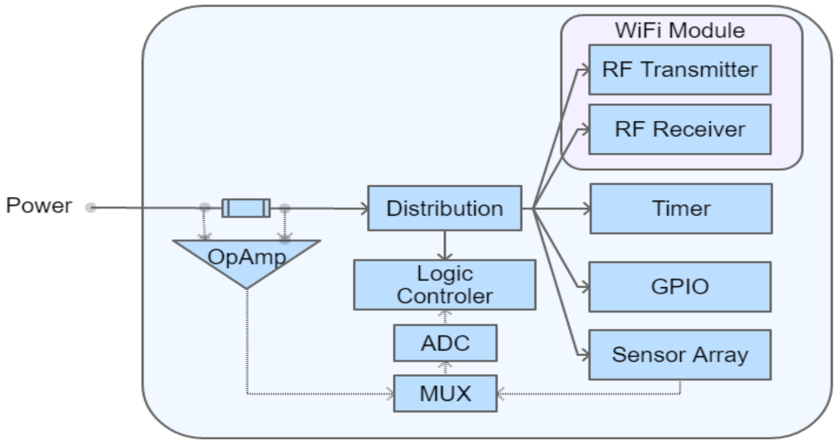


Figure. 2 IoT node power measurement architecture

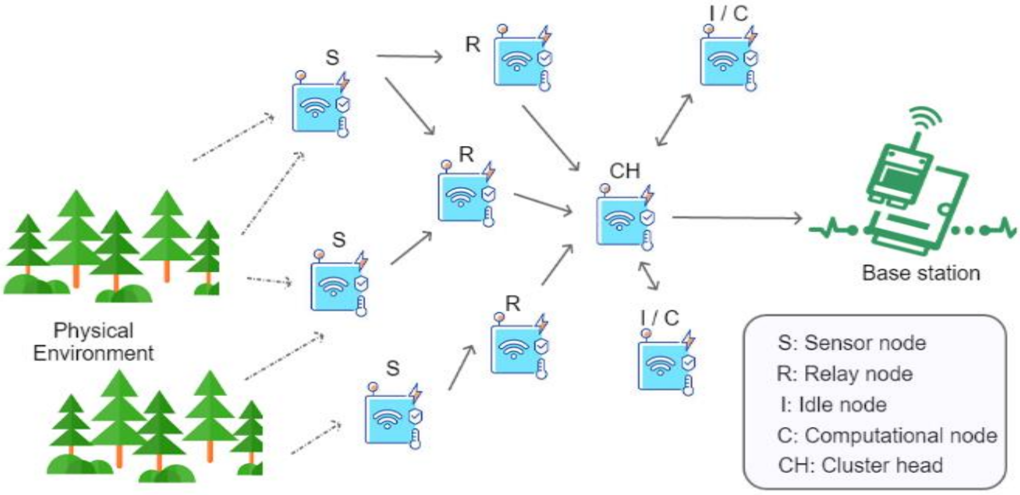


Figure. 3 Wireless sensor environment

resource management and load balancing. It is a pre-emptive algorithm established over the round-robin principle. In RRS, every task is assigned with an estimated time-to-complete (ETC) which is also represented as quantum. Once the task is accomplished within the corresponding ETC, the resource will be preempted for the successive task from the queue. Resistance against starvation and convoy effects, fair allocation of resources, burst time independence and applicability of context state switching are the well-known advantages of RRS over the period.

4. Proposed method

Proposed method IFLEACH is the conglomeration of three imperative functional modules, they are idiosyncratic fuzzy energy estimator, cumulative fuzzy energy hierarchy builder and energy based cluster head exchanger. The idiosyncratic fuzzy energy estimator module is used to measure the residual energy of each and every node in the cluster. The primary purpose of cumulative fuzzy energy hierarchy builder is to maintain the nodes in different hierarchical clusters. It is also given the ability to predict the energy utilization of the nodes from different hierarchical clusters and to shift nodes from one hierarchy to another accordingly. Energy based cluster head exchanger is used to designate the cluster heads periodically based on the residual energy, computational power and euclidean distance proximity. The basics and the processing procedures of these modules are submitted in this chapter.

4.1 Idiosyncratic fuzzy energy estimator (IFEE)

Most of the modern wireless sensor IoT nodes are powered by the microcontroller-based circuit boards. These circuit boards have tiny but powerful integrated circuits such as MCP32F511N which has the capability of built-in current and power consumption measurements. IFEE module makes use of this facility to measure the power consumption of the nodes in different operational states such as sense, idle, watchdog mode, computational mode,

 DOI: 10.22266/ijies2023.0630.44

Table 2. Node role and power consumption

Role of the Node	Purpose	Active Modules	Power consumption
Sensor	Physical parameter measurement	Sensor Array, ADC, Logic Controller, Transmitter	Moderate
Relay	communication bridge between nodes	Transmitter, Receiver, Logic Controller	High
Idle	Nil	Logic Controller	Low
Computational	To perform lightweight distributed processing	Logic Controller, Transmitter or Receiver (Intermittent)	Moderate
Cluster Head	Cluster management and base station communication	Transmitter, Receiver, Logic Controller	Higher

$$\forall i = 1 \rightarrow Role(N_i) \coloneqq \begin{cases} 'Idle' \ if \ \rho_t < \frac{1}{4}\rho_m \\ 'Sensor' if \ \rho_t \geq \frac{1}{4}\rho_m \ and \ \rho_t < \frac{1}{2}\rho_m \ and \ \delta_{M_s} = 1 \\ 'Compute' if \ \rho_t \geq \frac{1}{4}\rho_m \ and \ \rho_t < \frac{1}{2}\rho_m \ and \ \delta_{M_s} = 0 \\ 'Relay' if \ \rho_t \geq \frac{1}{2}\rho_m \ and \ \rho_t < \frac{3}{4}\rho_m \\ 'Cluster\ Head' otherwise \end{cases} \quad (2)$$

transmitter mode, receiver mode and transceiver mode. Each of these states consume different currents based on the operational hardware utilization. The power measurement architecture of a typical IoT wireless sensor node is illustrated in Fig. 2.

IFEE starts with a power-on-self-test (POST) phase in which the power consumption of the device at different modes are measured.

Let P be the superset of power consumption of an individual node with members $p_1, p_2 \dots p_n$ where n refers to the number of nodes. Each set p_i is initialized as follows $p_i = \{\rho_s, \rho_i, \rho_w, \rho_c, \rho_t, \rho_r, \rho_x\}$ where $\rho_s, \rho_i, \rho_w, \rho_c, \rho_t, \rho_r$ and ρ_x refers to the power consumption during sense, idle, watchdog, compute, transmit, receive and transceiver modes. Energy is the measurement of power and time, so that an energy estimation table (EET) is constructed during the initial phase of the network operation. Based on the energy utilization, EET is updated with the power sets $p_1, p_2 \dots p_n$ and the temporary role of every individual node is determined. A typical wireless sensor network environment is illustrated in Fig. 3.

Sensor nodes are the paramount nodes in dealing with the physical environment. They are used to collect information about the environment on purpose. The relay nodes are used to carry the collected data from the sensor nodes to the cluster heads in a single hop network. A relay node is permitted to connect with another relay node in multi-hop network. There can be some Idle nodes during the random deployment where the accessibility is not available for direct deployment. These nodes are also used optimally by employing fog computing. The cluster head collects the observatory data and transmits to the base station.

The power consumption of a particular node is closely bound to the role of the node in the entire network environment. The residual energy of these nodes is also very much correlated with the role of nodes in the network. In general, since both the transmitter and receiver modules are active during the transceiver mode of the relay nodes, they are more prone to power supply insufficiency. The role of the node and the active modules are given in Table 2.

In IFEE module, the EET table entry begins just after network initialization. Maximum consumable power ρ_m is required to determine the role of a node. ρ_m is calculated using the following equation.

$$\rho_m = \rho_s + \rho_c + \rho_x \quad (1)$$

The current consumption of every node is measured and updated in the EET. The role of each node is determined using conditional Fuzzy as given in Eq. (2).

Where N_i is the ith node in the cluster, ρ_t is the present power consumption, and δ_{M_s} refers to the state of the sensor module.

The EET has the fields to accommodate total energy, remaining energy during the previous timestamp, current residual energy, role of the node, and estimated residual energy for the subsequent timestamp. The organization of EET is given in Table 3.

Table 3. Energy estimation table

Node ID	Role	Total Energy (mAh)	Previous Residual Energy (mAh)	Current Residual Energy (mAh)	Estimated Residual Energy (mAh)
Node 1	Cluster Head	40600	22500	21100	19700
Node 2	..	..	..	..	..
…	..	..	..	..	..
Node n	..	..	..	..	..

There are two reasons behind the introduction of idiosyncratic nodes, they are selection of nodes to determine the repeated role change, identification of faulty or security wise compromised nodes. The repeated role changing idiosyncratic nodes are common in which a redetermination of current role by applying Eq. (2) is enough to handle. Identification of faulty or compromised nodes is an essential process of the IFEE module, which helps to prevent the performance degradation and the security of the entire network. The residual energy values are calculated by subtracting the used energy from total energy. The Idiosyncratic nodes are identified using the following equation.

$$\forall i = 1 \rightarrow N_i \coloneqq \begin{cases} Idiosyncratic\ if |\varepsilon_E - \varepsilon_A| > \varphi \\ Normal\ otherwise \end{cases} \tag{3}$$

In which ε_E is the estimated residual energy of Node N_i, and ε_A is the actual residual energy, φ is the permitted volatile threshold. The repeated roll changing idiosyncratic nodes are included to the subsequent entry update of EET whereas, faulty or compromised nodes are discarded from EET where an alternate neighbourhood node is selected to serve the purpose.

Algorithm: IFEE

Input: Network Information

Output: EET (Timestamp based update)

Step 1: Initialize EET with Node ID and total energy of the power source

Step 2: $\forall \mathrm{i} = 1 \rightarrow N$ apply POST for N_i to get p_i

Step 3: Initialize timestamp by affixing the values of set p_i in EET

Step 4: Estimate residual power based on the current role

Step 5: wait for next timestamp

Step 6: update residual energy in EET

Step 7: Identify Idiosyncratic nodes by applying Eq. (3)

Step 8: Determine current role of all idiosyncratic nodes by Eq. (2)

Step 9: If Roll Change, update EET, else discard node

Step 10: Update timestamp

Step 11: Repeat from 4

By this way the proposed IFEE module eliminates the faulty and compromised nodes and maintains the EET for the adjacent modules.

4.2 Cumulative fuzzy energy hierarchy builder (CFEHB)

The purpose of the CFEHB module is to categorize the network nodes based on the energy level hierarchy. The standard cluster head selection procedure of LEACH is optimized to achieve swifter cluster head selection. The nodes are split into 3 categories such as Highly recommended, Recommended and Not recommended to be the cluster heads. CFEHB maintains the Highly recommended category in the top layer, from which the descended cluster heads are selected frequently. If all the nodes in the Highly category are busy or not available to take the cluster head role, then the first node from the Recommended category though the probability of occurrence of this situation is a typical. The classification process is performed using the following equation.

$$\forall i = 1 \rightarrow N \coloneqq N_i = \begin{cases} Highly\ Recomended\ if\ = \varepsilon_A > \frac{3}{4}\varepsilon_{\max} \\ Recomended\ if\ \varepsilon_A \leq \frac{3}{4}\varepsilon_{max}\ and > \frac{1}{2}\varepsilon_{max} \\ Not\ Recomended\ otherwise \end{cases} \tag{4}$$

where ε_A is the current residual energy of node N_i and ε_{max} is the maximum residual energy available among all nodes N_1 to N_n.

Then the nodes belonging to highly recommended and recommended categories are arranged in descending order pursuant to the following algorithm.

Algorithm: CFEHB

Input: Node Categories

Output: Arranged Node Categories

$$N_{CH} = \begin{cases} \Gamma_{HN2} \text{ if } \Gamma_{HN1} \text{ is the current cluster head and } \varepsilon_{A\Gamma_{HN1}} - \varepsilon_{A\Gamma_{HN2}} < \omega \\ \Gamma_{HN1} \; otherwise \end{cases} \quad (5)$$

Step 1: Let Γ_H be the set of members $\{\gamma_{H_1}, \gamma_{H_2}, \dots \gamma_{H_n}\} \in$ 'Highly Recommended' $Category$
Step 2: Let Γ_R be the set of members $\{\gamma_{R_1}, \gamma_{R_2}, \dots \gamma_{R_n}\} \in$ ' Recommended' $Category$
Step 3: $\forall i = 1 \rightarrow H_n - 1 : \forall j = i + 1 \rightarrow H_n \coloneqq swap\left(\gamma_{H_i}, \gamma_{H_j}\right) if \left(\gamma_{H_i} < \gamma_{H_j}\right)$
Step 4: $\forall i = 1 \rightarrow R_n - 1 : \forall j = i + 1 \rightarrow R_n \coloneqq swap\left(\gamma_{R_i}, \gamma_{R_j}\right) if \left(\gamma_{R_i} < \gamma_{R_j}\right)$
Step 5: Return Γ_H and Γ_R

4.3 Energy based cluster head exchanger (ECHE)

ECHE module is useful to improve the overall network performance and lifetime by efficiently switching cluster heads. The output CFEHB gives two usable categories "Highly Recommended" and "Recommended" with a set of nodes. ECHE prevents using the same node again and again as a cluster head. Though the nodes are sorted in the CFEHB output categories, there is a chance of selecting the same cluster head for more than one timestamp. A novel strategy is introduced in the ECHE module which operates based on Eq. (5).

Where N_{CH} is the new cluster head, Γ_{HN1} is the first node element of "Highly Recommended" category and Γ_{HN2} is the next element in the same category, $\varepsilon_{A\Gamma_{HN1}}$ is the residual power of Γ_{HN1}, $\varepsilon_{A\Gamma_{HN2}}$ is the residual power of Γ_{HN2} and ω is the energy threshold calculated as $\frac{\rho_{max} \times d_t}{2}$, d_t is the duration in seconds between the timestamps. ECHE will proceed with Γ_{RN1} if Γ_{HN2} is not available.

Computational power (CP) and euclidean distance proximity (EDP) are also used as corroborating parameters in ECHE cluster head selection. A CP index set is introduced in ECHE as $\{cp_1, cp_2, \dots, cp_n\}$ where n is the number of nodes in the cluster. The computational power index is calculated as follows

$$\forall i = 1 \rightarrow n \coloneqq cp_i = \frac{f_i + M_i}{2} \quad (6)$$

Where f_i refers the processor clock frequency and M_i is the random-access memory (in bytes) provided to the node.

The elements of CP are arranged from highest CP index to lowest CP index as in the CFEHB algorithm. Instead of residual energy, here the computational power index is used to sort the nodes.

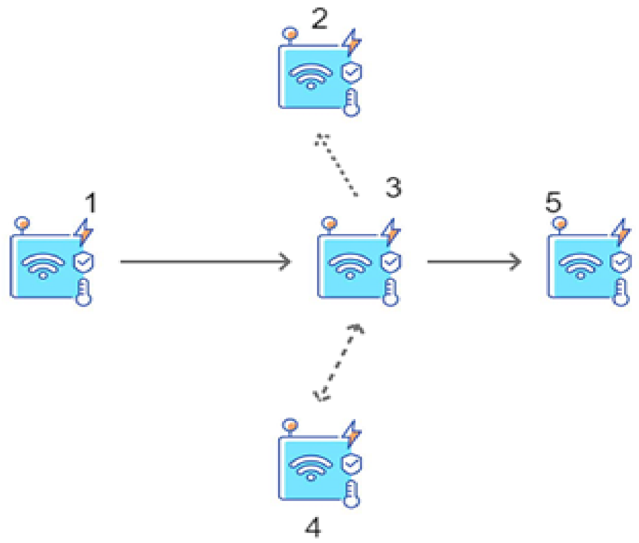
Similarly, the Euclidean distance-based set $E = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$ is also formed based on the K-nearest neighbour method. The nearest proximity nodes are placed in the first place and the longer proximity nodes are listed at the end of the set. That is, unlike Γ and CP set, the Euclidean distance set E is sorted in the Ascending order to prioritize the lowest proximity node as cluster head.

Algorithm: ECHE
Input: Γ_H, Γ_R, CP and E
Output: Cluster head N_{CH}
Step 1: Load data from Γ_H, Γ_R, CP and E
Step 2: Select Γ_H such that $\forall i = 1 \rightarrow n \coloneqq \gamma_{H_i} \in CP \cap E$
Step 3: Similarly select Γ_R such that $\forall i = 1 \rightarrow n \coloneqq \gamma_{R_i} \in CP \cap E$
Step 4: If $\Gamma_H = NULL \; and \; \Gamma_R = NULL$, declare End of Network lifetime
Step 5: If $\Gamma_H = NULL \; and \; \Gamma_R \neq NULL$, Append Γ_R to Γ_H
Step 6: If $n(\Gamma_H)$>1, Select cluster head by using Eq. (5), go to step 10
Step 7: Else if $\Gamma_R = NULL$ $N_{CH} = \Gamma_{HN1}$, go to step 10
Step 8: Else Append Γ_R to Γ_H
Step 9: Select cluster head using Equation 6
Step 10: return N_{CH}

Sometimes there is a possibility to have multiple nodes with the same available energy levels. In this case ECHE applies round-robin technique to select the eligible nodes one by one. Different example network scenarios and cluster selection scenarios are explained using the following Figures.

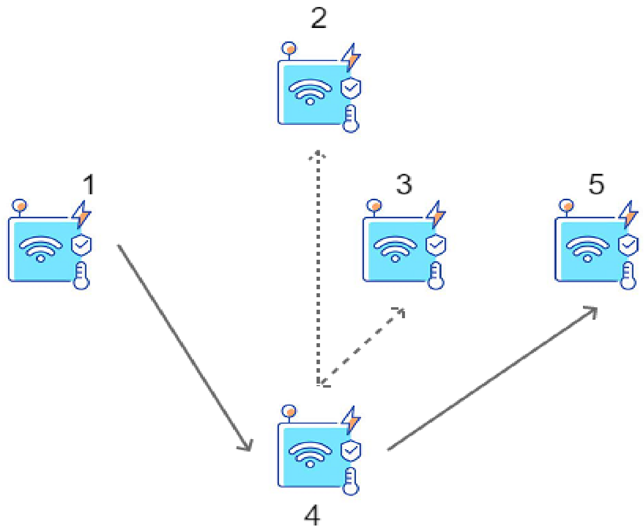
As in Fig. 4, node 3 is selected as the cluster head based on the initial residual energy calculation. At a typical energy consumption of 270 mAh during it's role as a cluster head, Node 3 exists with a remaining residual energy of 2230 mAh. At this situation, Node 4 with 2250 mAh remaining energy is selected as the cluster head during t+1.

During timestamp t+2, both nodes 2 and 4 are with the same residual energy of 1980 mAh. Since node 4 is already working as the cluster head, node 2 takes over the position as per the round-robin approach of proposed ECHE algorithm. Thus node 4 is released from the cluster head position to avoid the stress heating problem of the batteries [17]. By this way the composition of the functional modules IFEE, CFEHB and ECHE ensures the proper cycling of cluster heads



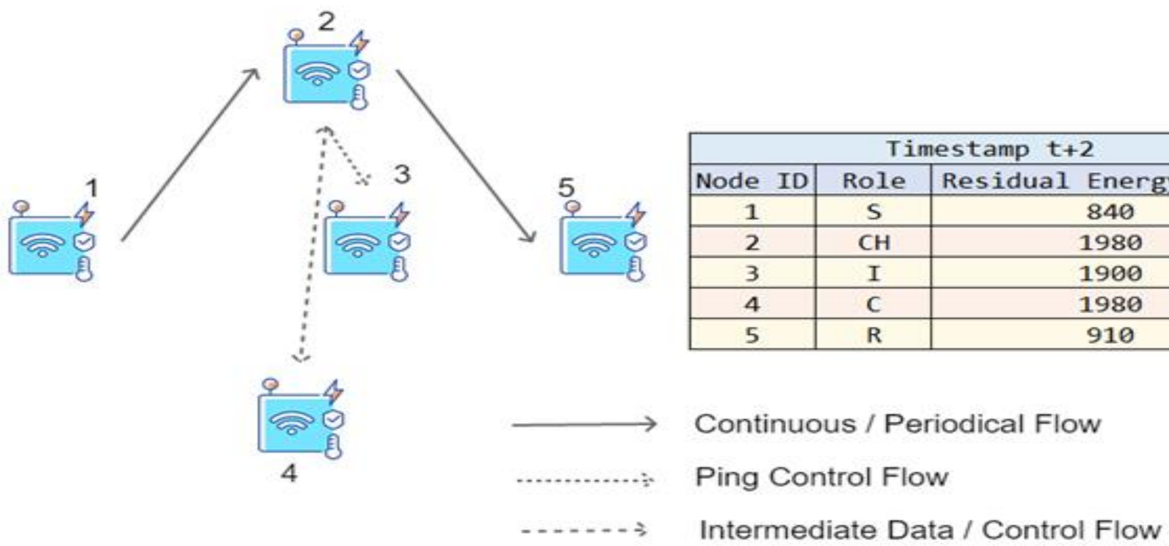
Timestamp t		
Node ID	Role	Residual Energy (mAh)
1	S	1000
2	I	2000
3	CH	2500
4	C	2400
5	R	1050

Figure. 4 Network scenario at timestamp t



Timestamp t+1		
Node ID	Role	Residual Energy (mAh)
1	S	920
2	I	1990
3	C	2230
4	CH	2250
5	R	980

Figure. 5 Network scenario at timestamp t+1



Timestamp t+2		
Node ID	Role	Residual Energy (mAh)
1	S	840
2	CH	1980
3	I	1900
4	C	1980
5	R	910

Figure. 6 Network scenario at timestamp t+2

Received: December 2, 2022. Revised: March 14, 2023.

Table 4. Simulation network environment

Element	Details
Simulation Area	5000 x 5000 square meters
Simulation Time	1 hour real-world
Evaluation Time Stamp	6 minutes
Number of Nodes	1000
Type of Nodes	Heterogeneous
Data Size	Max 2048 bits
Communication Mode	Wireless
Number of Wi-fi Routers	30
Node Placement	Random distribution
Traffic Type	Random real-world replica
Node movement	Random real-world replica

Table 5. Throughput

Throughput (kbps)					
Nodes	**APSA**	**EESRA**	**EESCH**	**RECHS**	**IF LEACH**
100	31903	33407	31455	32549	34065
200	29978	32102	27888	30064	34373
300	27936	30266	26061	28390	32226
400	23943	27651	24758	25142	32385
500	21805	24025	22399	24029	30480
600	18997	22607	18559	20645	30491
700	16358	20136	16809	17865	28184
800	12183	17710	14414	16051	28132
900	10450	16193	10940	14040	26445
1000	6605	14938	8525	11420	26175

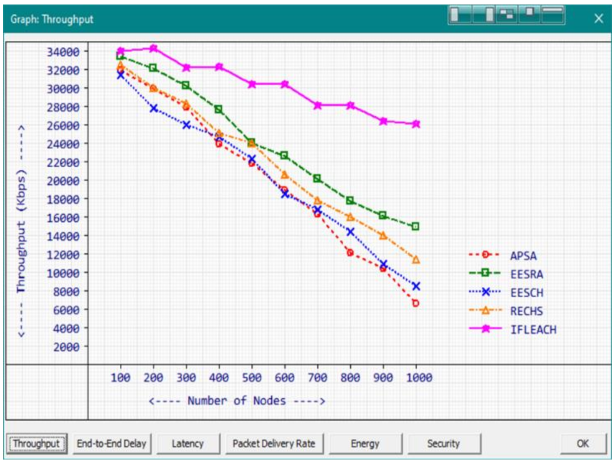


Figure. 7 Throughput

to prevent cluster head failures.

5. Experimental setup

IFLEACH is a network architecture and protocol-based work which needs low level access to the elementary parameters. Therefore, the most machine friendly programming language C++ [18] is used to develop the scripts and network snippets. Visual Studio 2019 [19] is used to develop the user interface (UI) which works as a launchpad for the network simulator. Visual Studio is selected due to its member language C++ 20.0 [20], and OPNET [21] is selected for simulation here because of its ability to simulate core level network simulation. OPNET has the capability of simulating a given network architecture by emulating the hardware network components. The results of OPNET with defined timestamps are more accurate and a real-world replica in most studies [22]. The user interface has the provision to load the existing and proposed method network architecture snippets and protocol source codes to the OPNET. The UI is responsible for collect the simulation periodical simulation results from OPNET. Then the UI exports the results to the report file and generates graphs based on the observed results from OPNET. Simulated IFLEACH environment details are given in Table 4.

6. Results and analysis

The purpose of introducing proposed IFLEACH is to enhance the user experience and quality of service in IoT based wireless sensor networks in comparison to the existing system such as APSA [13], EESRA [14], EESCH [15] and RECHS [16]. Thus, the benchmark performance parameters such as throughput, latency, end-to-end delay, packet delivery ratio, energy and security are obtained for existing and proposed method.

6.1 Throughput

The throughput refers to the communication quality in a network. Higher throughput means the higher quality of the network schema whereas the lower throughput refers to the disadvantage of the network. Based on the observed results, proposed IFLEACH achieved higher value throughputs. The observed values are given in Table 5 and the comparison graph is provided as Fig. 7.

6.2 Latency

Latency is one of the communication delays which causes performance degradation. The lower value of latency refers to the higher efficiency of the network. The measured latency values are given in Table 6 and comparison graph is given in Fig. 8.

6.3 End-to-end delay

The complete travelling time of a data packet is referred as end-to-end delay which is the accumulated values of all communication delays. The simulation observed End-to-End values are given below in Table 7 and comparison chart is given in Fig. 9.

A good network architecture should keep the end-

Received: December 2, 2022. Revised: March 14, 2023.

Table 6. Latency

Latency (mS)					
Nodes	**APSA**	**EESRA**	**EESCH**	**RECHS**	**IF LEACH**
100	21	27	31	33	25
200	27	26	39	34	23
300	29	23	35	30	22
400	26	30	40	35	27
500	32	33	37	35	29
600	31	28	40	35	29
700	40	35	43	42	29
800	35	33	47	46	26
900	40	38	50	44	33
1000	39	38	53	47	35

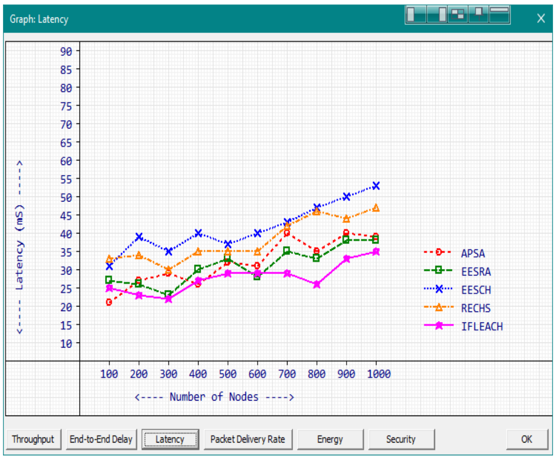


Figure. 8 Latency

to-end delay in control. Lesser end-to-end delay refers to the higher quality of the network. As per the observed results, it is understood that the proposed IFLEACH is having very less end-to-end delay values.

6.4 Packet delivery rate (PDR)

Packet delivery rate is calculated using the formula $\frac{P_R}{P_T}$, where P_R is the number of packets received at the destination whereas P_T is the total number of packets transmitted towards the destination by the sources. A good network architecture / protocol should secure high PDR scores since it is directly proportional to the quality of the network. Observed PDR results for existing and proposed methods are given in Table 8. The comparison graph for PDR is given in Fig. 10.

6.5 Energy

Energy is one of the prime consideration factors in IoT based wireless sensor network since most of the devices are battery powered and a constant power source access is almost not possible. These kinds of nodes are really used to monitor the physical environment where the access is much more restricted

Table 7. End-to-end delay

End-to-End Delay (mS)					
Nodes	**APSA**	**EESRA**	**EESCH**	**RECHS**	**IF LEACH**
100	125	117	165	139	98
200	136	148	188	164	108
300	154	172	206	186	107
400	158	197	228	204	113
500	178	234	253	228	112
600	189	261	270	246	115
700	196	291	291	272	116
800	214	321	320	294	123
900	224	351	337	314	127
1000	238	374	362	330	126

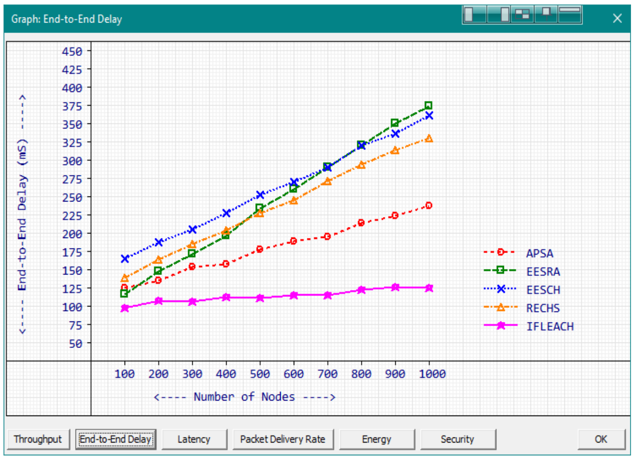


Figure. 9 End-to-end delay

Table 8. Packet delivery rate

Packet Delivery Rate (%)					
Nodes	**APSA**	**EESRA**	**EESCH**	**RECHS**	**IF LEACH**
100	93	97	94	99	100
200	93	94	92	97	98
300	91	92	89	94	97
400	90	90	88	92	96
500	89	89	87	91	96
600	88	85	85	89	95
700	87	83	81	86	93
800	87	81	80	85	92
900	86	81	79	81	92
1000	85	78	77	79	91

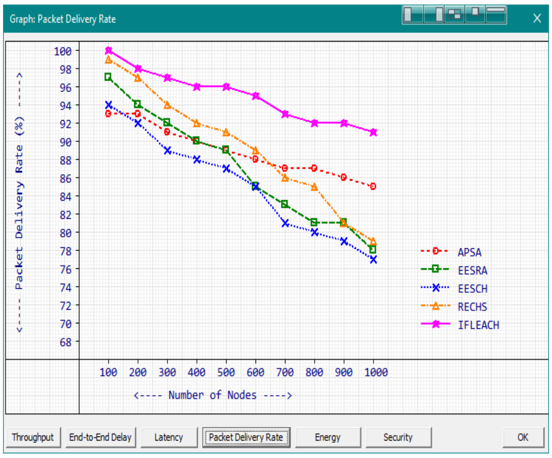


Figure. 10 Packet delivery rate

Received: December 2, 2022. Revised: March 14, 2023.

Table 9. Energy

Energy (J)					
Nodes	**APSA**	**EESRA**	**EESCH**	**RECHS**	**IF LEACH**
100	880	727	687	902	358
200	923	748	721	936	368
300	964	771	727	969	403
400	972	802	771	982	414
500	1015	806	778	1005	436
600	1047	825	813	1052	412
700	1085	841	834	1059	457
800	1111	862	847	1077	466
900	1144	913	891	1102	487
1000	1171	910	917	1138	470

Figure. 11 Energy

to humans. Energy is measured in Joules which is very proportional to the battery power backup which is measured in mAH. Measured energy levels for the networks with various numbers of nodes is given in Table 9. A comparison graph also provided in Fig. 11.

6.6 Security

Security is an inevitable entity in any private communication. Security is measured by OPNET by triggering predefined network security threat models such as brute force attack, dictionary attack and burrows-abadi-needham logic (BAN Logic) [23]. These attacks are used to expose the secure data transmission during the entire simulation. Security is calculated as $100-(P_c\,/\,P_t\ \times 100)$ where P_c is the compromised data packets and P_t is the total number of packets transmitted. While trying to increase the performance of a network by reducing the computational complexity of cryptography procedures, obliviously the level of security will decrease.

Sustaining the security level during the performance enhancement process is essential for a quality network protocol. Computed security values of existing and proposed methods are given in Table 10 and the comparison graph is provided in Fig. 12.

Table 10. Security level

Security (%)					
Nodes	**APSA**	**EESRA**	**EESCH**	**RECHS**	**IF LEACH**
100	89	92	92	95	99
200	89	92	92	94	99
300	89	92	91	95	99
400	88	92	92	94	99
500	89	92	91	95	99
600	88	93	92	95	98
700	88	92	92	94	99
800	88	92	91	94	99
900	88	93	92	95	98
1000	89	93	92	94	98

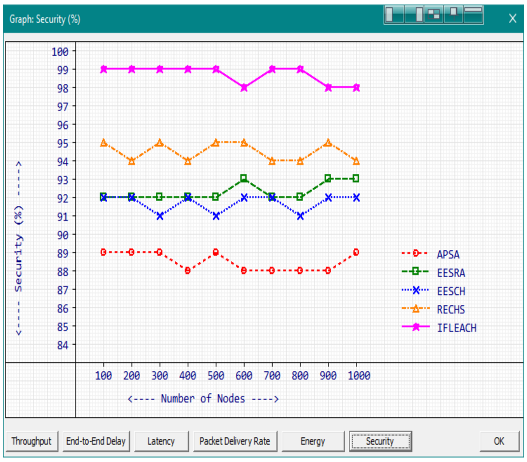


Figure. 12 Security level

7. Conclusion

A novel IFLEACH method which incorporates idiosyncratic fuzzy energy estimator, cumulative fuzzy energy hierarchy builder and energy based cluster head exchanger module is conceived and evaluated in this work to improve the performance of IoT based wireless sensor networks. Based on the evaluation results, it is realized that the proposed method provides the overall performance enhancement in terms of throughput, end-to-end delay, latency, packet delivery ratio, energy consumption and security. The first functional module IFEE is used to improve the performance as well as it is also improving the security of the overall network security by identifying and eliminating non-role changing idiosyncratic nodes. Based on the performance improvements of 11 kbps throughput and 4% security, the proposed IFLEACH method can be recommended for the real-time IoT based Wireless sensor network applications.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

The Contribution of authors are as follows: Ms. R. Sridevi, Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation and writing— original draft preparation. Dr. V. Sinthu Janita Prakash, Validation, Supervision and Project Administration.

Acknowledgment

The authors are grateful to the Management and Principal for their motivation and constant support. This research has been supported by the grant obtained under the scheme of Seed Money for Research Projects from Cauvery College for Women (Autonomous), Tiruchirappalli – 620 018, India.

References

[1] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, "A Survey and Future Directions on Clustering: From WSNs to IoT and Modern Networking Paradigms", *IEEE Transactions on Network and Service Management*, Vol. 18, No. 2, pp. 2242-2274, 2020.

[2] S. S. Gill, S. Tuli, M. Xu, I. Singh, K. V. Singh, D. Lindsay, and P. Garraghan, "Transformative Effects of IoT, Blockchain and Artificial Intelligence on Cloud Computing: Evolution, Vision, Trends and Open Challenges", *Internet of Things*, Vol. 8, 2019.

[3] K. Lakhwani, H. Gianey, N. Agarwal, and S. Gupta, "Development of IoT for Smart Agriculture a Review", In: *Proc. of In Emerging Trends in Expert Applications and Security: ICETEAS 2018 Springer Singapore*, pp. 425-432, 2019.

[4] K. Bajaj, B. Sharma, and R. Singh, "Integration of WSN with IoT Applications: A Vision, Architecture, and Future challenges", *Integration of WSN and IoT for Smart Cities*, pp. 79-102, 2020.

[5] Z. Chi, Y. Li, H. Sun, Y. Yao, and T. Zhu, "Concurrent Cross-Technology Communication Among Heterogeneous IoT Devices", *IEEE/ACM Transactions on Networking*, Vol. 27, No. 3, pp. 932-947, 2019.

[6] X. Li, N. Zhao, R. Jin, S. Liu, X. Sun, X. Wen, ... and Y. Zhou, "Internet of Things to Network Smart Devices for Ecosystem Monitoring", *Science Bulletin*, Vol. 64, No. 17, pp. 1234-1245, 2019.

[7] M. An, Y. Kim, and W. K. Cho, "Effect of Smart Devices on the Quality of CPR Training: A Systematic Review", *Resuscitation*, Vol. 144, pp. 145-156, 2019.

[8] O. L. A. López, N. H. Mahmood, and H. Alves, "Enabling URLLC for Low-Cost IoT Devices via Diversity Combining Schemes", In: *Proc. of. 2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1-6, 2020.

[9] S. Saroj, M. Yadav, S. Jain, and R. Mishra, "Performance Analysis of Q-Leach Algorithm in WSN", *International Journal of Inventive Engineering and Sciences*, Vol. 5, No. 10, pp. 1-4, 2020, doi:10.35940/ijies. I0977.0651020.

[10] S. Rani, S. H. Ahmed, and R. Rastogi, "Dynamic Clustering Approach Based on Wireless Sensor Networks Genetic Algorithm for IoT Applications", *Wireless Networks*, Vol. 26, pp. 2307–2316, 2020.

[11] S. Zhao, P. V. Rengasamy, H. Zhang, S. Bhuyan, N. C. Nachiappan, A. Sivasubramaniam, and C. Das, "Understanding Energy Efficiency in IoT App Executions", In: *Proc. of 2019 IEEE 39th International Conference on Distributed Computing Systems*, pp. 742-755, 2019.

[12] J. Sheth and B. Dezfouli, "Enhancing the Energy-Efficiency and Timeliness of IoT Communication in WiFi Networks", *IEEE Internet of Things Journal*, Vol. 6. No. 5, pp. 9085-9097, 2019.

[13] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S. J. Lim, "An Affinity Propagation-Based Self-Adaptive Clustering Method for Wireless Sensor Networks", *Sensors*, Vol. 19, No. 11 p. 2579, 2019.

[14] E. F. A. Elsmany, M. A. Omar, T. C. Wan, and A. A. Altahir, "EESRA: Energy Efficient Scalable Routing Algorithm for Wireless Sensor Networks", *IEEE Access*, Vol. 7, pp. 96974-96983, 2019,

[15] A. F. Jemal, R. H. Hussen, D. Y. Kim, Z. Li, T. Pei, and Y. J. Choi, "Energy-Efficient Selection of Cluster Headers in Wireless Sensor Networks", *International Journal of Distributed Sensor Networks*, Vol. 14, No. 3, 2018.

[16] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, "Residual Energy-Based Cluster-Head Selection in WSNs for IoT Application", *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 5132-5139, 2019.

[17] S. Kambampati, J. S. Gray, and H. A. Kim, "Level Set Topology Optimization of Load Carrying Battery Packs", *International Journal of Heat and Mass Transfer*, Vol. 177, 2021.

[18] F. Zehra, M. Javed, D. Khan, and M. Pasha, "Comparative Analysis of C++ and Python in

Terms of Memory and Time", *Preprints [Google Scholar]*, 2020.
[19] https://visualstudio.microsoft.com/
[20] https://www.geeksforgeeks.org/features-of-c-20/
[21] https://opnetprojects.com/opnet-network-simulator/
[22] M. Pahlevan and R. Obermaisser, "Evaluation of Time-Triggered Traffic in Time-Sensitive Networks Using the OPNET Simulation Framework", In: *Proc. of 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing, IEEE*, pp. 283-287, 2018.
[23] I. Leuştean and B. Macovei, "DELP: Dynamic Epistemic Logic for Security Protocols", In: *Proc. of 2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing,* pp. 275-282, 2021.