Journal of Advanced Zoology

ISSN: 0253-7214

Volume 44 Special Issue -02 Year 2023 Page 1039:1051

Implementation of Leaky Bucket with deep learning Algorithm to Avoid Congestion in DEC Protocol in medical applications

[1]**D. Shankar,** Department of CSE, Vishnu Institute of Technology, Vishnupur, Bhimavaram, Andhra Pradesh-534202
Corresponding author Email: shankar.d@vishnu.edu.in
[2]**P. Shyamala Madhuri**, Department of CSE, Vishnu Institute of Technology, Vishnupur, Bhimavaram, Andhra Pradesh-534202, shyamalamadhuri.y@vishnu.edu.in
[3]**Dr.P.Chandra kanth,** Associate Professor in the department of CSE at Audisankara College of Engineering & Technology ASCET (Autonomous), Gudur, Tirupathi(DT). India, 524101, chandrakanthc4u@gmail.com
[4]**VVS SASANK**, Department of computer science and Engineering, Koneru Lakshmaiah Education, vaddeswaram, AP, India, Email : sasank64@gmail.com
[5]**Dr. B. Sankara Babu**, Professor, Dept. of CSE, Gokaraju Rangaraju Institute of Engineering and Technology(Autonomous), Bachupally, Telangana, bsankarababu81@gmail.com
[6]**K saikumar,** School of engineering, department of CSE, Malla Reddy University, Maisammaguda, Dulapally, Hyderabad, Telangana 500043, saikumarkayam4@ieee.org

Article History

Received: 27Aug 2023

Revised: 28Sept 2023

Accepted: 06Oct 2023

Abstract: A wireless sensor network is a critical component in many disciplines. There are a large number of sensor nodes in it. These sensor nodes perform various tasks, including identifying, dispensing, communicating, and providing power. Data is sent from source to destination and plays an important role. Congestion will occur during data transfer =0>0from one node to another and in the cluster head. Congestion will emerge as a result of either traffic division or resource allocation. Energy will be wasted due to traffic division congestion, which results in packet loss and retransmission of deleted packets. As a result, it must condense. Congestion management will be handled by a few wireless sensor networks using various protocols. Deterministic Energy Efficient Clustering (DEC) protocol is considered to reduce energy consumption based on residual energy in which the leaky bucket algorithm is cast-off. In the event of congestion, our plan outlines a strategy for dealing with and resolving it using this manner. According to simulation testing, the suggested approach may significantly improve longevity, energy, throughput, and packet loss.

Keywords: Leaky bucket, Queue, Packet size, Base station, Residual energy, Congestion.

1. INTRODUCTION

Nowadays, the advancement in microchip design has led to the fabrication of lightweight sensor nodes. The deployment of these sensor nodes can be found in different applications where monitoring is required. Those applications include health-care, agricultural lands, disaster-prone areas, etc. Many times, deploying these wireless sensor networks depends on how the battery life is managed. The constraint in the capacity of batteries in wireless sensor networks would make them costlier, and it has become a challenge for us to deploy them on a large scale. Therefore, it has become a necessity to develop a protocol that results in the optimization of energy that is consumed. It has been recognized that we need to use an energy-aware protocol to control energy consumption by configuring itself.

Clustering can be used to manage Wireless Sensor Networks [1]. The sensors that are nearer to each other are formed as clusters. In each group, a leader is elected from the available sensor nodes [2]. This process is known as Clustering. After selecting the cluster heads, the data from the cluster members is given to cluster heads, and cluster heads will filter the gathered data using techniques like data compression and send the collected data to the Base station (BS) [3].

Anyway, the work done by cluster-head would consume more energy. Instead of fixing the same node as cluster-head, rotating cluster-head among the members would result in more significant gains [4]. Therefore, the design for distributive Wireless Sensor Networks on how far it can manage energy consumption has been an essential factor in determining a better protocol's success. Earlier, the cluster heads were rotated randomly, and there was no guarantee of optimizing the election of cluster heads. Hence, a protocol to elect a better cluster head should be employed to check the congestion phenomena. One such protocol is the DEC protocol. In this protocol, the criterion for selecting cluster heads is monitoring the residual energy of the sensor nodes [5].

Congestion occurs in a WSN when the traffic load on a specific sensor node exceeds the assigned buffer capacity. Congestion can also occur as a result of a network's resources being used in an inequitable manner [6]. Congestion usually arises at the sensor node due to the existing constraints, and there are two forms of congestion: node-level and link-level congestion. The buffer utilization, or the disparity between the traffic response time and departure rate, is usually examined in node-level congestion. The channel utilization, on the other hand, is used to track congestion at the link level. In both cases, early identification of congestion and mitigation from the network is essential [7].

Small sensor nodes in wireless sensor networks (WSNs) have their battery, memory, transceiver, CPU, and sensor(s) [8]. The nodes collect data locally and transmit it to a hub where it may be processed to provide valuable insights [9]. The data rate, bandwidth, memory, and battery life of sensor nodes are all limited somehow. Depending on the application's needs, these nodes may run on tiny batteries for days, months, or even years. Hence, efficient power use may extend the nodes' battery life [10] . Communication uses much more power than computing does in WS[11]Ns. Thus, it is necessary to design intelligent and efficient routing protocols to ensure that the sensor nodes share the energy burden equitably [12]. There are a few ways to categorize routing protocols: data-centric, hierarchical cluster, and location-based. Hierarchical cluster-based protocols best solve the energy restrictions of such networks. These protocols group the nodes into smaller groups called collections; in each cluster, one node serves as the cluster head (CH), which is responsible for gathering information from the other nodes in the group and sending it to a central location (BS). Eight times less energy is used with these protocols than with traditional routing techniques [13].

According to how the nodes interact, WSNs may be categorized as either stationary or mobile. Static deployment, which is used by the vast majority of apps, has several limitations [14]. To begin, a static deployment cannot provide ideal sensor field coverage. Consequences are severe if all key events happen beyond the designated area, and even massive deployments of nodes cannot ensure perfect coverage. Second, a communication gap between sensor nodes is created when static nodes expire or malfunction [15]. As a result, the connectivity suffers, leading to dropped packets and decreased overall network quality. Gateway nodes, just one hop away from their BS, are another critical issue with the static deployment [16]. Every network traffic must pass via these nodes on its way to the BS. Therefore they use a lot of power in the process [17].

In contrast, mobile nodes reposition themselves throughout the field to generate a rotating collection of gateway nodes [18]. Hence, all the nodes serving as gateways might share the energy burden. Mobile nodes guarantee full coverage by snatching up events and sending them to the BS [19]. Quality of Service (QoS) measures such as coverage, connection, energy usage, and others are enhanced by mobile WSNs [20]. The sensor nodes may remain stationary for numerous uses. Yet, they need data mules to collect and send information to the BS [21]. The data mule has a dual purpose: it transmits information to the BS and keeps the network connection at all times [22]. The number of nodes in a particular cluster may exceed the maximum threshold limit when the nodes travel throughout the cluster field [23]. Packet loss, delay, the blocking of new connections, and a drop in quality of service are all possible outcomes of congestion. Both stationary and moving WSNs suffer from congestion [24]. As time-stamped, data must be sent to the BS instantly, and congestion causes significant impediments for time-critical applications [25]. Even brief interruptions in transmission might render the data unusable or obsolete. Increasing the connection capacity or data rate control at each node may help reduce network congestion [26].

2. MATERIAL AND METHODS

We employed a probabilistic-based approach to regulating energy usage in wireless sensor networks. The primary goal of this protocol is to increase the lifetime of a Wireless Sensor Network by utilizing global data obtained from it rather than local data [27]. The residual energy of each node is what constitutes local information. The drawback of such protocols is that the number of cluster heads (CHs) who will be elected, or the elected Cluster-head, will need more energy to lead the process [28]. We may also use a deterministic cluster-head selection method instead of a probabilistic-based technique because the deterministic cluster-head method outperforms the probabilistic model [29]. This statement is made by considering energy consumption. Eq. (1) represents a generic probabilistic model which is given by making use of these protocols,

$$T(n) = \begin{cases} \frac{p}{1-\left(r mod \frac{1}{(P_x)}\right)*P} \times Q if n_x \in G; \\ 0 \qquad\qquad Otherwise, \end{cases} \qquad (1)$$

Q is a function of each node's residual energy ratio [30]. It may be regarded as a constant number, where x represents a node and nrm, int, or adv stands for normal, intermediate, or advanced nodes [31]. Think about the situation when Q equals one. [8][9]. For each iteration r, each sensor node will arbitrarily choose a value between 0 and 1 to serve as the cluster head, in line with the threshold value function in Eq.(1). The sensor node will become a Cluster-head if its value is smaller than the threshold value T for node n. (n) [32]. G And P represents a set of candidates for Cluster-head who were not elected and their chances of being selected (CMs) [33].

Clustering occurs when the energy spent per bit in the transceiver circuit, denoted by E (T X), exceeds the distance threshold for switching amplification models, characterized by d to CH, as defined by the DEC protocol [34]. At the first stage, known as "setup," all nodes will use the indication function to choose CHs [35]. The selected Cluster-heads broadcast will get a K-bit message. The non-persistent Carrier Sense multiple access (CSMA MAC) protocol will be used to assign radio resources in the Advertising message, also known as ADV. An ID for the Cluster leaders and a header is provided in the notification. Cluster members (CMs) are unelected nodes [10, 11], [12, 13], [14], and [15]. Members will choose a cluster and submit join requests to the Cluster-head with the lowest communication cost depending on the received signal strength of the advertising message. The request header, CM-ID, CHID, and other information are all included in this message (cluster head-ID). The Cluster-heads will implement TDMA for use in internal communications. The setup process is complete at this point. The system enters a steady-state condition after collected data has been sent from CMs to CHs and then from CHs to the Base-Station. Inter-cluster communication through direct sequence spread spectrum is possible (DSSS).

Several different data transmission issues might lead to network congestion. Over flowing buffers, time-varying channels, etc. Unusual traffic patterns may be identified using "congestion detection." In other words, when a packet is sent from one node to another using a particular [16]. Several packages are lost because of congestion on individual nodes. Energy consumption and link exploitation both drop as packet loss rises. When several sensor nodes try to use the channel simultaneously, it becomes congested. When link-level congestion occurs, all nodes simultaneously attempt to transmit traffic over the connection. This leads to packet collisions. Congestion at the link level also decreases link use. The problems mentioned above may be avoided if congestion is well managed or avoided. One example of such a method is the leaky bucket algorithm [17, 18].

2.1. Leaky Bucket

This adapt procedure is used to handle incoming packets at any rate as part of a traffic flow strategy rendering approach. This technique uses a waiting period while buffering observations across time. Packages are discarded when the bucket is full and their arrival is indicated. The FIFO system, as seen in Figure 1, is described here. DEC protocol has embraced this method.

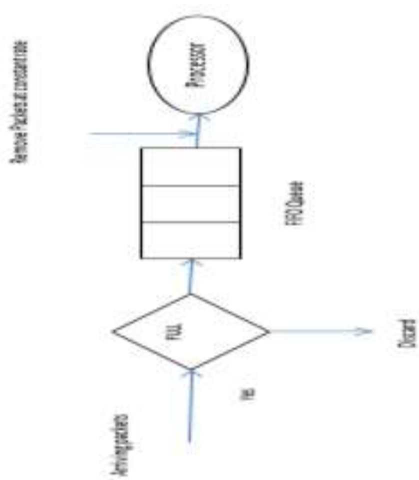


Figure:1 Working of Leaky bucket algorithm.

After the queue has reached its limit, its size must be evaluated. As long as storage space is available, all newly created packets are made public. Installing this method at the network's access point safeguards the network against packet bursts. Therefore, congestion is avoided, and QoS is improved by keeping traffic levels at a suitable range throughout the system.

2.2 Algorithm for Leaky Bucket

Begin
Input the packet size
Input the buffer size
Set buffer Size, interval, second=0
For(i=0; i<number of packet, i++)
If (packet size[i] +p_sz_rm)>buffer size
If(packet size [i] >buffer size)
Drop packet
Else
Add tokens to bucket: rate * time passed
Store it in bucket;
Increment the interval and second;
Send packet
Available tokens = available tokens – 1
}}
End.

3. EXPERIMENTAL

To avoid congestion in the channel process and at nodes, leaky bucket algorithmic concept is taken and implemented in an energy efficient protocol called DEC to avoid packet drops and enhance smooth data delivery process.

3.1Energy Calculation:

The energy required for transmission across a bit packet distance is expected to be as follows:

$$E_{t_x}k,d = E_{elect} * k + \varepsilon f_s * k * d * dif(d < d_0)(2)$$

$$E_{elect} * k + \varepsilon amp * k * d * d * d * dif(d > d_0)(3)$$

Where $\varepsilon f_s$is allowed space, $\varepsilon amp$is a multipath loss, d is a distance among starting and ending nodules and $d_0$is limit distance

$$d_0 = squareroot\ (\frac{\varepsilon f_s}{\varepsilon amp})(4)$$

The receiving radio energy

$$E_{Rx}(k) = k * E_{elect}(5)$$

3.2Implementation of Leaky Bucket concept in DEC protocol

Phase 1: Sensor hosts can be arranged in any order by using a common method.

Phase2: For clustering groups, the DEC protocol computation is used.

Phase 3: Apply the leaky bucket method on the nodes available in a various cluster. Sink will issue the CH-ticket, a better, unmatched type of response towards cluster members.

Phase 4: Assess then evaluate cluster heads those with tickets. The limit is described as follows:

$T(n) = \{{}^{t}/_{1-t} * (r mod\ {}^{1}/_{t}\}$(6)

Where t is preferred CHs number

Phase5: Through adjacent CHs, a subordinate section of nodes act as cluster members.

Phase6: Ticket will be shared to the Cluster Members by the Cluster Head.

Phase 7: Nodes that are part of a ticket, share their data to respective sensed CH.

Phase 8: Put on data in order to acquire information.

Phase 9: The information is then sent to the sink by the Cluster Head.

Phase 10: Estimate as well as inform about energies.

Phase 11: In every cluster the dead nodes are verified and tallied.

Phase 12: If all nodes in stage are dead, the lifespan is known; otherwise, go to step 12.

CNN model for congestion handling

In this section a brief computations of CNN model for congestion handling mechanism has been disscssed.

Mathematical computations of proposed model:

$$KL = -\frac{1}{2}\sum_{i=1}^{n}(1 + log(\sigma_i) - \mu_i^2 - \sigma_i) \quad (1)$$

The equation 1 is calculating the divergence loss and dividing the each components. In this µ =mean, σ = standard deviations of Google net distribution. In this i is varied depending on training data. The equation 2 demonstrate that individual elements prediction in this f(x) is a individual vector of selected image.

$$f(x_0) = (v^1, v^2, v^3, \dots, v^n,), where = 128, \quad (2)$$

$$f(x_{id}^i) = (v_{id}^1, v_{id}^2, v_{id}^3, \dots, v_{id}^n,), \quad (3)$$

Equation 3 explains about particular identity of individual elements in the selected packet in DEC protocol. Here separate model file is saved in the form of vector “V”, after extracting the vectors the pass function extracts the original vector with important values.

$$val, ind = \sum_{i}^{n} f(v_0^i, index_0^i), \qquad where x_0^i > 0, \quad (4)$$

Equation 4 explains about validation and indices values of corresponding positions in the face, if x>0 then it is an indices value, otherwise not considered in the ROI. Here n= number of features, v= vector original image, i is the index in the vector. Coming to equation 5 the ID is differentiating the validation function and giving the features in each packet.

$$val_{id}^{i} = \sum_{i=1}^{nid} f\left(v_{id}^{i}, index_{id}^{i}\right), \; for\ each\ index_{id}^{i} = index_{0}^{i}, \tag{5}$$

$$iden = \min\left(val - \begin{pmatrix} val_{id}^{1} \\ val_{id}^{2} \\ val_{id}^{3} \\ \vdots \\ val_{id}^{i} \end{pmatrix} \right), \tag{6}$$

Here, for each identification in the tuple, we do not need to pick values greater than zero; rather, we only take the values referring to the indices of the largest values in the final photo. This step is very important since for instance, the features of an eye can be stored in a specific index; therefore, in each request in the database, we need to take the function of that packet. Designers would measure the distance between the filtered values of the original vector and the corresponding values of each vector of the identification in the dataset to identify the identities. Both would have the same identities at the lowest difference here between processed values of the main packet and a particular identity packet.

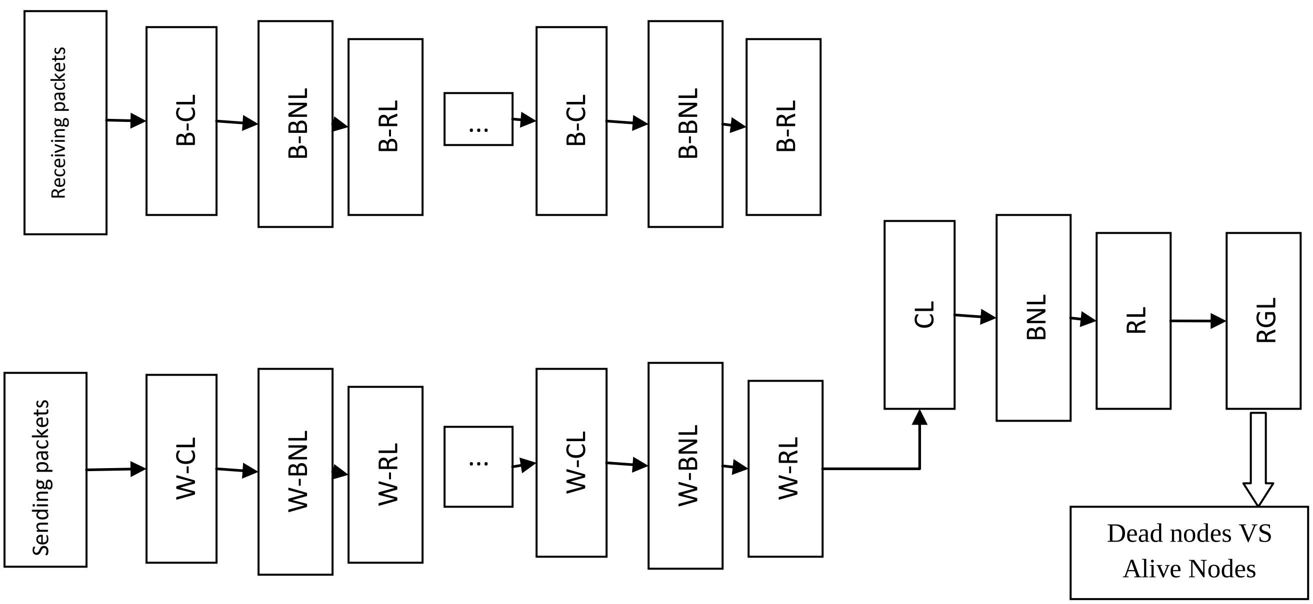


Figure: 2 CNN deep model

In the above Figure 2, three phases of information—green, yellow, and red rectangular bounding boxes—are shown. Each of these boxes extracts characteristics and sends them to a categorization block. Statistics numbers between 0 and 1 represent the retrieved characteristics. Although there are other feature extraction models, the ResNet approach yields findings that are more accurate than those from prior models. Features include all of the gathered features. a csv file that has a source code directory. The OS, SYS, Date-time, and Threading Python modules must be called in order to extract the features. Content analysis, directory changes, and the import of underlying packages are all crucial for the OS's performance. The SYS module offers many function variables that can be interpreted differently to affect Python run time. The focus of the date-time package is on the functions of the calendar, time, zone-info, and details. In this challenge, each major piece of information is derived from a local server using a specific geographical address. With the exception of the functions get ident(), get native id(), and main thread(), threading is a task for which inputs must be provided.

$$ind = total \times \frac{weights}{128}. \tag{7}$$

$$TW = \sum_{i1=0}^{128} W_{idn}^{o_{i2}}, if\, o_{i2} = idn_i, \tag{8}$$

Equation 7& 8 explains about original FER identification, Here the extracted packet passes the all blocks and giving the exact information on selected packet. In this o= index filter weight, Tw = feature weight shown in figure 3.

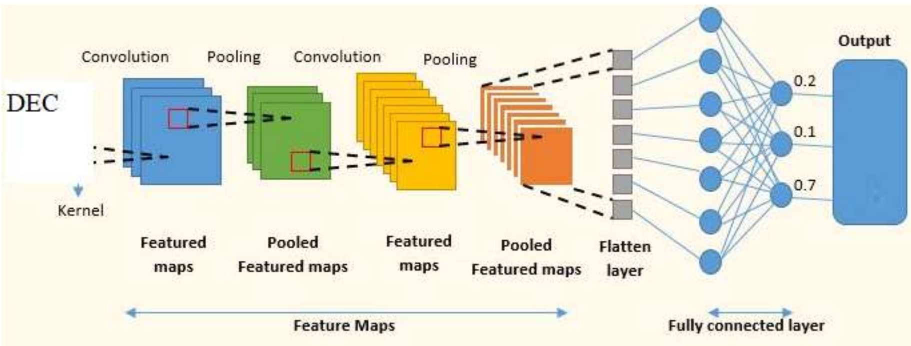


Figure: 3 CNN deep learning architecture

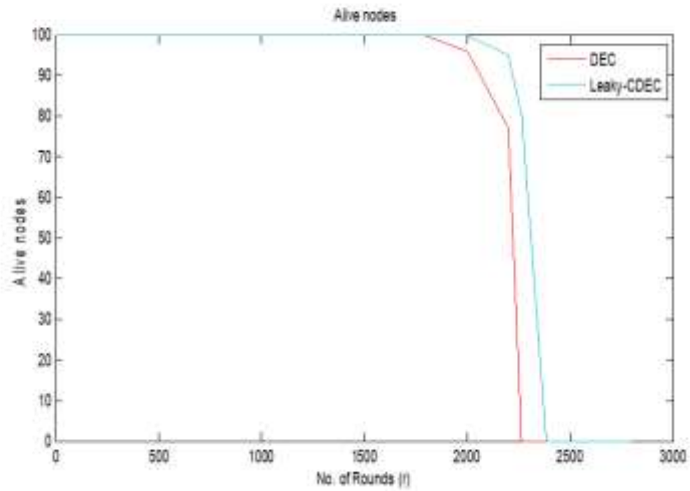
4. RESULTS AND ANALYSIS

The below given table 1 specifications are used as working parameters for testing results. For simulation results MATLAB platform is used.

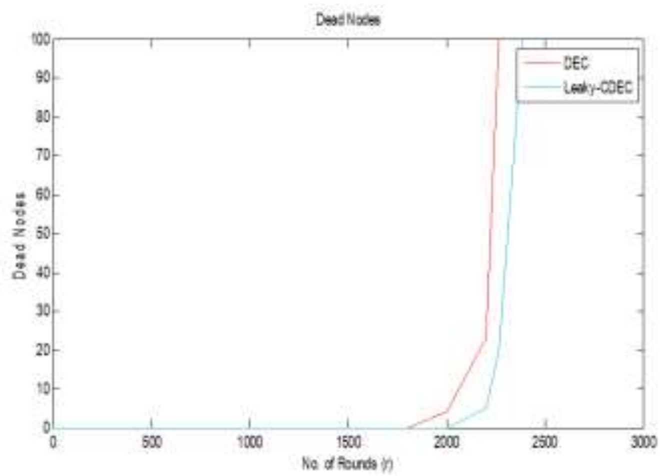
Table 1. Design Specification.

Parameter	Values
E_{elec}	50nJ/bit
E_{DA}	5nJ/bit/message
E_0	0.5J
k	4000
p_{opt}	0.1
ϵ_{fs}	$10pJ/bit/m^2$
ϵ_{mp}	$0.0013pJ/bit/m^4$
n	1000

Stability Period: Time of Reliability during Leaky Congestion in the DEC Network (Including Mortality Rate Study) (Leaky-CDEC). Figure 5 displays the number of iterations a network can endure before it succumbs to a catastrophic loss of life. The Leaky-CDEC protocol checks the RE of each node after each cycle to establish which CHs will act as cluster leaders. Nodes having a higher RE have a better chance of becoming cluster leaders. The current CH may remain the cluster head if its RE exceeds the preceding CHs. Consider that the present CH has a lower RE than the rest of the cluster. If this occurs, the cluster member with the highest RE automatically assumes the CH role. The CH for a collection collects information from its members and relays it to the hub. Figure 4 (a) shows that compared to DEC protocols, Leaky-CDEC has a more extended stability period when traffic flow strategy rendering is applied to packets arriving at an arbitrary rate (b). A DEC lost its first node around 1860, whereas a Leaky-CDEC lost its first node around 1972. Thus, Leaky-CDEC has a stability period of 5.9 percentage points longer than DEC.



(a)



(b)

Figure 4 (a) &(b). Stability Period analysis in terms of Live and Dead nodes

Table 2. Analysis of Live nodes

Rounds	Live nodes	
	DEC	Leaky- CDEC
0	1000	1000
500	1000	1000
1000	1000	1000
1500	1000	1000
2000	982	1000

CH's analysis: The lines in Figure 5 show a deployment of 100 CH nodes in a given networking zone. It reveals that the line dips dramatically at round 2265 for DEC and in Leaky-CDEC 2396, indicating that all of the nodes' energy has been used up (stage of falling to dead). Hence, cluster head nodes revert to regular nodes after each cycle and continue conversing with the sink. Thus, they cease at the network's conclusion.

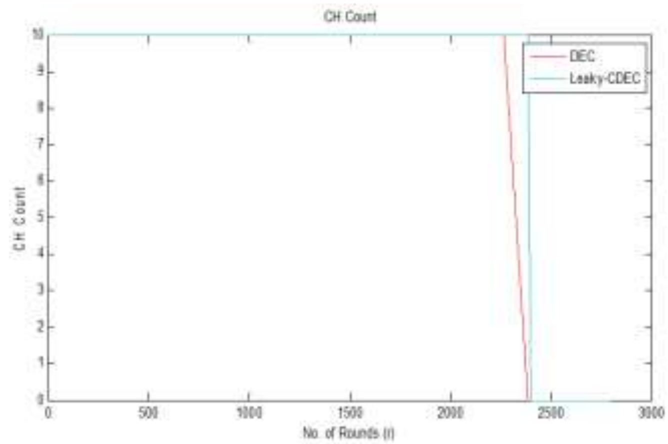


Figure 5. Analysis of Cluster head count

Packet Delivery Ratio: The number of packets transmitted to the received packets is the ratio. It suggests that the proposed method is more efficient than the existing one as shown in Figure 4.

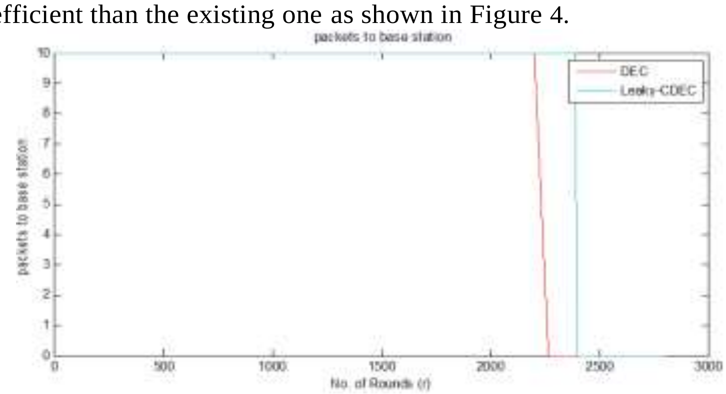
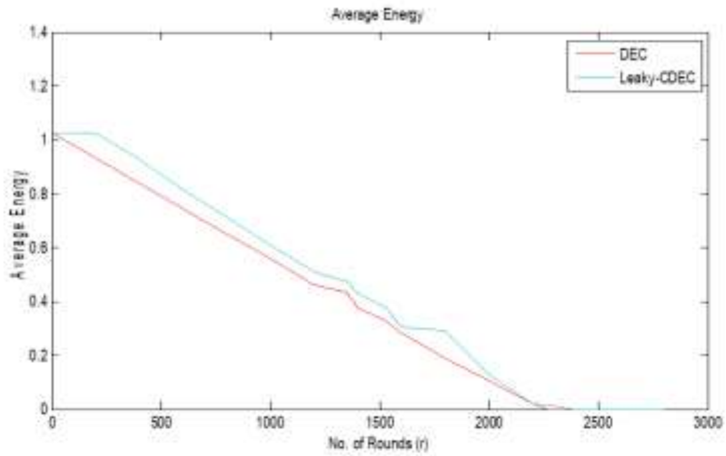


Figure 6 Analysis of packet to base station

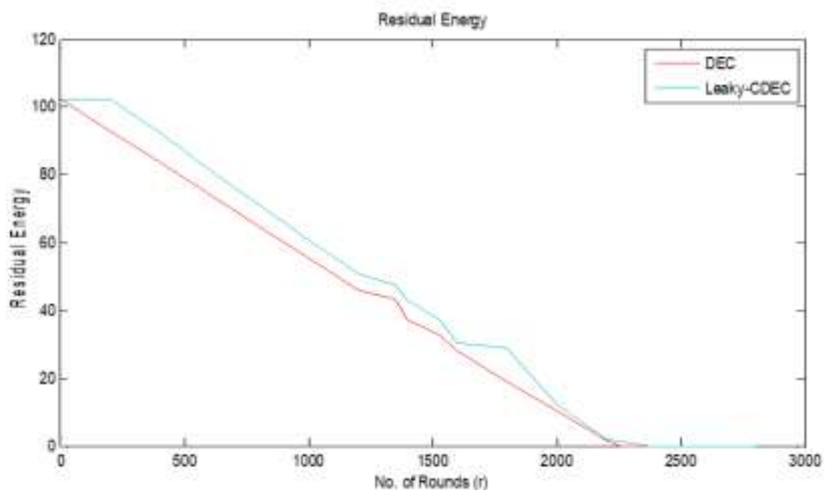
Average Energy and Residual Energy Performance Analysis in DEC and Leaky-CDEC: The assessment of DEC and Leaky-residual CDECs and average energies are shown in Figures 6 (a) and (b). Simulation results and a comparison between the DEC protocol and Leaky-CDEC are shown in Table II. According to the data, the total energy for round 2265 in the DEC protocol is zero; however, in the case of Leaky-CDEC, it is 1.09J. Around the year 2385, Leaky-CDEC services begin to dwindle. Leaky- CDEC outperforms DEC when the number of rounds is increased. Leaky- CDEC has been proven to improve performance by up to 5.34 percent compared to the current DEC protocol shown in figure 7.

Table :3 Residual energy analysis

Rounds	Residual energy		Average Energy	
	DEC	**Leaky- CDEC**	**DEC**	**Leaky- CDEC**
0	102.5	102.5	1	1
500	79.15	82.53	0.791	0.799
1000	55.42	57.79	0.560	0.575
1500	33.13	34.54	0.331	0.352
2000	9.647	10.05	0.102	0.125



(a)



(b)

Figure 7 (a) & (b). Residual energy and average energy analysis for DEC and proposed Leaky-CDEC.

Figure 7 clearly explains about Residual energy and average energy analysis for DEC and proposed Leaky-CDEC. In this proposed model attains more improvement compared to DEC.

Performance measures

In this section discussing performance measures related to DEC protocol efficiency

$F_1 \text{ score} = \frac{2.TP}{2.TP+FN+FP}$ ------ (9)

(F_1 score : worst value = 0; best value = 1)

$\text{sensitivity} = \frac{TP}{TP+FN}$ ------ (10)

(sensitivity: worst value = 0; best value = 1)

$\text{Accuracy} = \frac{TN}{TN+FP}$ ------ (11)

specificity: worst value = 0; best value = 1)

$\text{Recall} = \frac{FN}{TN+FP}$ ------ (12)

specificity: worst value = 0; best value = 1)

CONCLUSION

The application-based DEC protocol known as "leaky bucket" might be utilized to avoid network congestion during data transmission. Congestion issues in outlying community clusters are the target of this strategy. In this article, we replace traditional queues with the Leaky Bucket method and analyze its effect on various network performance indicators, including lifetime, throughput, packet delivery ratio, and power consumption. Imitation demonstrates that the new approach is more effective than the current way.

REFERENCES

[1] F. A. Aderohunmu, J. D. Deng, and M. K. Purvis. “Enhancing Clustering in Wireless Sensor Networks with Energy Heterogeneity”, International Journal of Business Data Communications and Networking-(Accepted, to appear in the forthcoming issue, 2011.

[2] F. Comeau. “Optimal Clustering in Wireless Sensor Networks Employing Different Propagation Models And Data Aggregation Techniques”, PhD thesis, Dalhousie University, Halifax, Nova Scotia, 2008.

[3] S. Gamwarige and C. Kulasekere. “An algorithm for energy driven cluster head rotation in a distributed wireless sensor network”, In Proceeding of the International Conference on Information and Automation, pages 354–359, Dec 2005

[4] M. Haase and D. Timmermann. “Low energy adaptive clustering hierarchy with deterministic cluster-head selection”, In IEEE Conference on Mobile and Wireless Communications Networks (MWCN), pages 368–372, 2002.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. “Energyefficient communication protocol for wireless microsensor networks”, In Proceeding 33rd Hawaii International Conference on System Sciences, 2000.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. “An Application-Specific Protocol Architectures for Wireless Networks”, IEEE Transactions on Wireless Communications, 1:660–670, 2002.

[7] C. Li, M. Ye, and G. Chen. “An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks”, In Proceeding of IEEE MASS05, pages 3–17, 2005.

[8] L. Qing, Q. Zhu, and M. Wang. “Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks”, Computer Communication, 29:2230–2237, 2006.

[9] G. Smaragdakis, I. Matta, and A. Bestavros. “SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks”, In Proceeding of the International Workshop on SANPA, 2004.

[10] F. Xiangning and S. Yulin. “Improvement on LEACH Protocol of Wireless Sensor Network”, In SENSORCOMM ’07: Proceedings of the 2007InternationalConferenceonSensorTechnologiesandApplications, 2007.

[11] Habibulla Mohammad, A. S. Chandrasekhara Sastry, Implementa-tion of C-DEC Protocol along with sectorization concept for wire-less sensor networks. Journal of Advanced Research in Dynamical and Control Systems, Vol 9.Sp-18/2017: p. 223-234.

[12] Habibulla Mohammad, A. S. Chandrasekhara Sastry, “ACNM:Advance coupling networkmodelsleep/wake mechanism for wireless sensor networks”, International Journal of Engineering and Technology(UAE), Vol 7(1.1): p. 350-354, 2018.

[13] Habibulla Mohammad, A. S. Chandrasekhara Sastry, “Implementation of Multihop technique in DEC Protocol”, International Journal of Simulation systems, Science &Technology(UAE), Vol19, Number 4, Page 1, p. 20.1-20.9, August 2018.

[14] Habibulla Mohammad, A. S. Chandrasekhara Sastry, ”Implementation of Three-Tier Multihop Technique in Advance Coupling Network Model-Deterministic Energy-Efficient Clustering of Wireless Sensor Networks”, Journal of Computational and Theoretical Nanoscience, Vol 16,Page 2581-2589,2019.

[15] Habibulla Mohammad, A. S. Chandrasekhara Sastry ,“DSWS- Distributed Sleep/Wake Scheduling Scheme for DEC Protocol in Wireless Sensor Networks”,International Journal of Recent Technology and Engineering, Vol 8,Isuue 02,July 2019,Page 2695-2701

[16] Koppula, N., Rao, K. S., Nabi, S. A., & Balaram, A. (2023). A novel optimized recurrent network-based automatic system for speech emotion identification. Wireless Personal Communications, 128(3), 2217-2243.

[17] Ali, F., Kumar, T. A., Loganathan, K., Reddy, C. S., Pasha, A. A., Rahman, M. M., & Al-Farhany, K. (2023). Irreversibility analysis of cross fluid past a stretchable vertical sheet with mixture of Carboxymethyl cellulose water based hybrid nanofluid. Alexandria Engineering Journal, 64, 107-118.

[18] Pittala, C. S., Vijay, V., & Reddy, B. N. K. (2022). 1-Bit FinFET carry cells for low voltage high-speed digital signal processing applications. Silicon, 1-12.

[19] Saran, O. S., Reddy, A. P., Chaturya, L., & Kumar, M. P. (2022). 3D printing of composite materials: A short review. Materials Today: Proceedings.

[20] Dasari, K., Anjaneyulu, L., & Nadimikeri, J. (2022). Application of C-band sentinel-1A SAR data as proxies for detecting oil spills of Chennai, East Coast of India. Marine Pollution Bulletin, 174, 113182.

[21] Rao, A. D., Chaitanya, A. K., Seshaiah, T., & Bridjesh, P. (2022). An Integrated Approach by Using Various Approaches for a Green Supplier Selection Problem. In Recent Advances in Manufacturing, Automation, Design and Energy Technologies: Proceedings from ICoFT 2020 (pp. 909-919). Springer Singapore.

[22] Yakaiah, P., & Naveen, K. (2022). An Approach for Ultrasound Image Enhancement Using Deep Convolutional Neural Network. In Advanced Techniques for IoT Applications: Proceedings of EAIT 2020 (pp. 86-92). Springer Singapore.

[23] Kumar, C. A., & Haribabu, K. (2022). A Great Adaptive SNR Assumed Low Power LDPC Decoder. In Advanced Techniques for IoT Applications: Proceedings of EAIT 2020 (pp. 443-451). Springer Singapore.

[24] Thottempudi, P., Dasari, V. S. C. B., & Sista, V. S. P. (2022). Recognition of Moving Human Targets by Through the Wall Imaging RADAR Using RAMA and SIA Algorithms. In Advanced Techniques for IoT Applications: Proceedings of EAIT 2020 (pp. 544-563). Springer Singapore.

[25] Arun, V., Reddy, D. L., & Rao, K. N. (2022). A Novel Analysis of Efficient Energy Architecture in Cryptography. In Advanced Techniques for IoT Applications: Proceedings of EAIT 2020 (pp. 339-345). Springer Singapore.

[26] Amareswer, E., & Raju Naik, M. (2022). Smart Erobern of Vehicles on Crosswalks. In Advanced Techniques for IoT Applications: Proceedings of EAIT 2020 (pp. 489-497). Springer Singapore.

[27] Saikumar, K., Rajesh, V., Babu, B.S. (2022). Heart disease detection based on feature fusion technique with augmented classification using deep learning technology. Traitement du Signal, Vol. 39, No. 1, pp. 31-42. https://doi.org/10.18280/ts.390104

[28] Kailasam, S., Achanta, S.D.M., Rama Koteswara Rao, P., Vatambeti, R., Kayam, S. (2022). An IoT-based agriculture maintenance using pervasive computing with machine

learning technique. International Journal of Intelligent Computing and Cybernetics, 15(2), pp. 184–197.

[29] Saikumar, K., Rajesh, V. A machine intelligence technique for predicting cardiovascular disease (CVD) using Radiology Dataset. Int J Syst Assur Eng Manag (2022). https://doi.org/10.1007/s13198-022-01681-7.

[30] Shravani, C., Krishna, G. R., Bollam, H. L., Vatambeti, R., & Saikumar, K. (2022, January). A Novel Approach for Implementing Conventional LBIST by High Execution Microprocessors. In 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 804-809). IEEE.

[31] Kiran, K. U., Srikanth, D., Nair, P. S., Ahammad, S. H., & Saikumar, K. (2022, March). Dimensionality Reduction Procedure for Bigdata in Machine Learning Techniques. In 2022 6th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 836-840). IEEE.

[32] Srinivas Rao, K., Divakara Rao, D. V., Patel, I., Saikumar, K., & Vijendra Babu, D. (2023). Automatic Prediction and Identification of Smart Women Safety Wearable Device Using Dc-RFO-IoT. Journal of Information Technology Management, 15(Special Issue), 34-51.

[33] Sreelakshmi, D., Sarada, K., Sitharamulu, V., Vadlamudi, M. N., & Saikumar, K. (2023). An Advanced Lung Disease Diagnosis Using Transfer Learning Method for High-Resolution Computed Tomography (HRCT) Images: High-Resolution Computed Tomography. In Digital Twins and Healthcare: Trends, Techniques, and Challenges (pp. 119-130). IGI Global.

[34] Saikumar, K., Rajesh, V., & Rahman, M. Z. U. (2022). Pretrained DcAlexnet Cardiac Diseases Classification on Cognitive Multi-Lead Ultrasound Dataset. International Journal of Integrated Engineering, 14(7), 146-161.

[35] Maddileti, T., Sirisha, J., Srinivas, R., & Saikumar, K. (2022). Pseudo Trained YOLO R_CNN Model for Weapon Detection with a Real-Time Kaggle Dataset. International Journal of Integrated Engineering, 14(7), 131-145.