A Cluster-based Congestion Avoidance Technique for Data Aggregation for the IoT Network Using the Heuristic Technique

Anitha P [a], Vimala H S [b], Shreyas J [c]

[a]Department of Computer Science and Engineering, Visvesvaraya College of Engineering (UVCE, IIT Model College), Bengaluru, Karnataka, India;
[b]Department of Computer Science and Engineering, Visvesvaraya College of Engineering (UVCE, IIT Model College), Bengaluru, Karnataka, India;
[c] Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, Karnataka, India

ARTICLE HISTORY
Compiled June 10, 2024

ABSTRACT
Congestion issues have arisen due to the rise of Internet of Things (IoT) devices and increased data traffic within IoT networks. Congestion can result in decreased reliability, higher latency, and performance degradation due to the overwhelming amount of data generated by various IoT devices. To optimize data flow, reduce collisions, and improve overall network efficiency, this paper looks into the root causes of congestion. It proposes a new adaptive algorithm called Congestion Avoidance through Cluster Head (CA-CH). A thorough examination of the current mechanisms for controlling congestion is provided to combat it effectively. The research assesses distributed and centralized methods, considering how well they apply to dynamic and diverse IoT environments. The cluster head is an essential strategy for ensuring reliable communication and effective data traffic management. The traffic management in the Internet of Things network is efficiently managed in the article by the BUG and Bully algorithms. Utilizing the Contiki OS Cooja simulator, the simulation is conducted. According to the results of the simulation, the CA-CH performs better than the previous methods in terms of packet delivery ratio, average delay of (10.6 ms), throughput with different cluster heads (0.98), average energy consumption of (6.6 mJ/packet), packet sending rate, and overall throughput of (30%).

KEYWORDS
6LoWPAN; BUG Algorithm; Bully Algorithm; Congestion; Cooja; Cluster; RSSI; QoS

1. Introduction

Global development necessitates quick adjustments to the ever-changing terrain of day-to-day business operations. Companies must consistently adopt intelligent technological advancements to guarantee expansion and competitiveness in international markets. Notably, recent years have seen a notable increase in the global adoption of state-of-the-art technological systems, with the Internet of Things emerging as a

CONTACT Anitha P Author. Email: murthy.ani@gmail.com

crucial sector that both boosts the world economy and raises people's standard of living.

The unique features of the Internet of Things, such as its remote access and control capabilities, have stimulated its applications in numerous fields, such as precision agriculture, military operations, smart health solutions, smart homes, futuristic cities, and environmental monitoring projects. A smooth and long-lasting IoT ecosystem that supports improved financial growth and stability is built on the interaction of IoT data management, interoperability, data analytics, and privacy (1).

The Internet of Things (IoT), is becoming a more useful tool in today's world and is still having a big influence on a lot of different businesses and everyday activities. IoT is shown to be beneficial in the following ways: Wearable devices, supply chain management, smart cities, healthcare, industrial internet, smart home automation, etc., IoT use has many advantages, but there are drawbacks(2). Need to consider and deal with issues like load balancing, privacy, security, and interoperability. IoT technologies are expected to become more and more integrated into daily life as they develop, opening up new possibilities and efficiency (3).

Proactive management and technology developments must be strategically combined to reduce network congestion. By putting Quality of Service (QoS) measures in place, important network traffic may be prioritized, mission-critical applications can be given the bandwidth they require, and the impact of congestion on vital services can be reduced.

Consequently, the suggested method for congestion control in networks is meant to enhance performance and prolong the life of network nodes. It manages and reduces congestion by arranging the path and rerouting the packets to the other node using the BUG algorithm. If there is any congestion in CH, using the BULLY search algorithm will assign a different CH. The BUG is a robust algorithm that considers every option before committing. The following are the primary contributions made to the work area:

(1) The cluster is created using the distributed Bully search algorithm, and communication is controlled by the cluster head (CH).
(2) To use the BUG algorithm to determine the path that will most likely get them to their destination while avoiding obstacles.
(3) To enhance network performance by load balancing, network topology optimization, QoS implementation, and reducing the number of hops needed for data transmission.
(4) The proposed work was simulated using the Cooja simulator and the Contiki operating system, which provides the necessary infrastructure for the IEEE 802.15.4 standard and the 6LoWPAN protocol stack.

1.1. *Organization of the Paper*

The remaining sections of this study are organized as follows: To address the issue of traffic congestion, Section 2 goes into further detail and examines some of the most significant and current research papers. Section 3 provides a detailed description of the architecture. The experimental data, including comparative and performance results, are displayed in Section 4. Part 5 concludes the work by providing space for additional investigation.

2. Related Work

To lower overhead, energy consumption, traffic congestion, and network lifetime, data aggregation techniques are crucial for gathering and combining information. In data aggregation scenarios for Internet of Things applications, it is difficult to develop dependable, energy-efficient, and delay-aware route planning. To overcome these obstacles, the current study suggests a Cluster-based Energy-aware Data Aggregation Routing (CEDAR) protocol in the Internet of Things by fusing fuzzy logic systems with the Capuchin Search Algorithm (CapSA). Cluster construction and intra/extra cluster routing are the two primary stages of the suggested hybrid routing algorithm (4).

Policies based on data aggregation are widely employed to ensure the desired service quality of the sensed data from the environment. These techniques collect and combine data packets in an effective way to improve network lifetime, data accuracy, and other factors while lowering power consumption, network overhead, and traffic congestion. This study proposes a hybrid system for Quality of Service-Aware Data Aggregation, or QADA. While addressing some of their significant drawbacks, the suggested system integrates some of the intriguing aspects of the tree-based and cluster-based data aggregation schemes (5).

As wireless service performance increases in Internet of Things (IoT) based wireless mesh networks (WMN), interference routing metrics and adaptive load balancing have drawn a lot of attention as the two main obstacles to be addressed. Furthermore, a high number of users' massive data creation causes network load, which negatively impacts IoT over WMN. Therefore, to reduce the current network problems, the author has suggested a clustering-based routing algorithm that takes into account interference and load-balancing routing measures. This study presents a plan that minimizes end-to-end latency while fully taking into account the quality of the complete path to the destination and the anticipated lifespan of IoT nodes that are experiencing bottlenecks due to heavy traffic (6).

An energy-efficient, congestion-aware resource allocation and routing protocol (ECRR) for Internet of Things networks is proposed in this research, which takes into account both resource allocation and routing difficulties. It is based on hybrid optimization approaches. To alleviate overall congestion between IoT gateways and large-scale devices, the first contribution of the proposed ECRR technique is to use data clustering and a metaheuristic algorithm for device allocation (7).

The lifespan of a network can be significantly shortened by imbalanced energy dissipation and cluster creation. As a result, for the Internet of Things running on the SDN architecture, this study suggests an effective clustering technique called C-LBCA that is based on load adjustment. The fundamental concept involves the utilization of cloud resources, including storage units and data centers, using a centralized SDN controller situated within the cloud to compute a load-balanced PSO clustering algorithm. The PSO is implemented by the SDN controller, which builds a clustering table (CT) by taking into account load-balancing, communication costs, and remaining energy considerations. An ideal collection of cluster heads (CHs) and cluster members (CMs) is among the data used in this CT for cluster formulation (8).

Cluster-based routing schemes improve the efficiency of data aggregation while using less energy. Therefore, if better clustering parameters are used to create a fitness function to reduce the network's energy consumption, genetic algorithms can be used as one of the optimization strategies to select a better cluster head without compromising the network's lifetime. In a similar spirit, the Genetic algorithm is frequently employed to choose cluster heads that make up the paths for transmitting data from the source to the destination or to optimize the path. This paper proposes the use of the Mobile Low Energy Adaptive Clustering Hierarchy (Mobile LEACH) protocol in an adaptive clustering method based on genetic algorithms (9).

Algorithm	**Centralized**	**Distributed**	**Notes**
QADA			Tree-Based
LEACH		✓	
CEDAR	✓		
ECRR	✓		
C-LBCA	✓		
ECFPSP	✓		
Proposed CA-CH		✓	

Table 1. Comparison of Various Clustering Selection Process

Algorithm	**Resource-based**	**Randomness**	**Conscious mode**
QADA	YES	YES	
LEACH	YES		
CEDAR		YES	
ECRR	YES		
C-LBCA	YES		
ECFPSP	YES		
Proposed CA-CH			YES

Table 2. Comparison of Various Techniques in creating CLUSTER HEAD Selection

3. Problem Statement

To propose a method for handling congestion by enhancing Quality of Service (QoS) in the Internet of Things network by utilizing the priority-based Bully and BUG algorithm. The objectives are:

(1) To develop a mechanism to manage congestion within individual cluster heads.
(2) To enhance data transfer efficiency within the IoT network by implementing obstacle avoidance strategies.
(3) To minimize energy consumption while effectively controlling the congestion.
(4) To reduce the average queue loss ratio, packet loss rate, and throughput with different cluster heads.

4. Proposed Work

The recommended clustering algorithm is explained in this section. All of the nodes are thought to be stationary, and the global positioning system sends their coordinates to the cloud (GPS).

4.1. ***System Design***

The primary traditional goal of clustering is load balancing. To move data from nodes to base stations, clustering algorithms often employ a divide-and-conquer strategy. An excessive amount of data to transfer can affect QoS support and cause uneven resource use. Generally, the methods and approaches that are now in use to address load balancing are Densified clusters, Balanced clusters, Congestion control mechanisms, and Balanced energy consumption. Our proposed CA-CH technique implements a congestion control mechanism, which assures no packet loss, no delays, and increased QoS.

Generally, in a network, there are primarily two ways to create clusters: One way to determine clusters to group nodes and then choose one or more of them as the CHs. Another method is to choose the CHs first, then invite other nodes to join neighboring CHs. The proposed technique implements the second method. Here, nodes are primarily joined to a cluster-based highest priority. The highest priority is based on the value of RSSI. In the second method itself, there are several procedures for choosing a CH, some of which are listed below.

(1) Resource-rich nodes: These are set up to be fixed CHs for the duration of the network. This approach is inefficient in networks when the nodes are homogeneous or have limited resources. Even in a heterogeneous network, a prolonged period of CH will rapidly deplete the node's power, eventually resulting in node death. Furthermore, mobile nodes and network dynamics in the case of fixed CHs may cause an imbalance in the number of members and/or cluster load, leading to network congestion and wasteful resource usage (10).
(2) Randomness: It is used in some clustering approaches as a way to distribute the CH duty among nodes. While this is advantageous in homogenous networks, in certain CHs unintentional unfair randomness might result in chronic energy usage and network congestion (11).
(3) Conscious: This selection, wherein CHs are chosen according to the nodes' and the network's conditions, it is the most widely used remedy for congestion problems. Using criteria like resource availability, neighbor count, and geographic position, suitable nodes are actively chosen to become CHs (12; 13). But it takes time to calculate all these parameters

Therefore, in situations where there is an urgent need to transmit high-priority or emergency packets, the proposed technique implements Congestion Avoidance through the Cluster Head (CA-CH) technique that uses the BULLY SEARCH algorithm that chooses the cluster head dynamically if there is any congestion in cluster head and the BUG algorithm that avoids congestion and transmits the packets.

The Bully Algorithm is typically employed for leader election in distributed systems whenever the current leader node fails or becomes unavailable. The concept is that the remaining nodes in the system must select a new leader and begin uninterrupted network communication.

4.1.1. Congestion Model

Network congestion is likely to occur when multiple sensor nodes send data to a cluster head simultaneously. The primary causes of network congestion include a finite network capacity and somewhat restricted bandwidth availability. Figure 1 illustrates this phenomenon.

High device density, limited bandwidth, data-intensive applications, network interference, inefficient routing protocols, scarce resources on edge devices, security attacks,

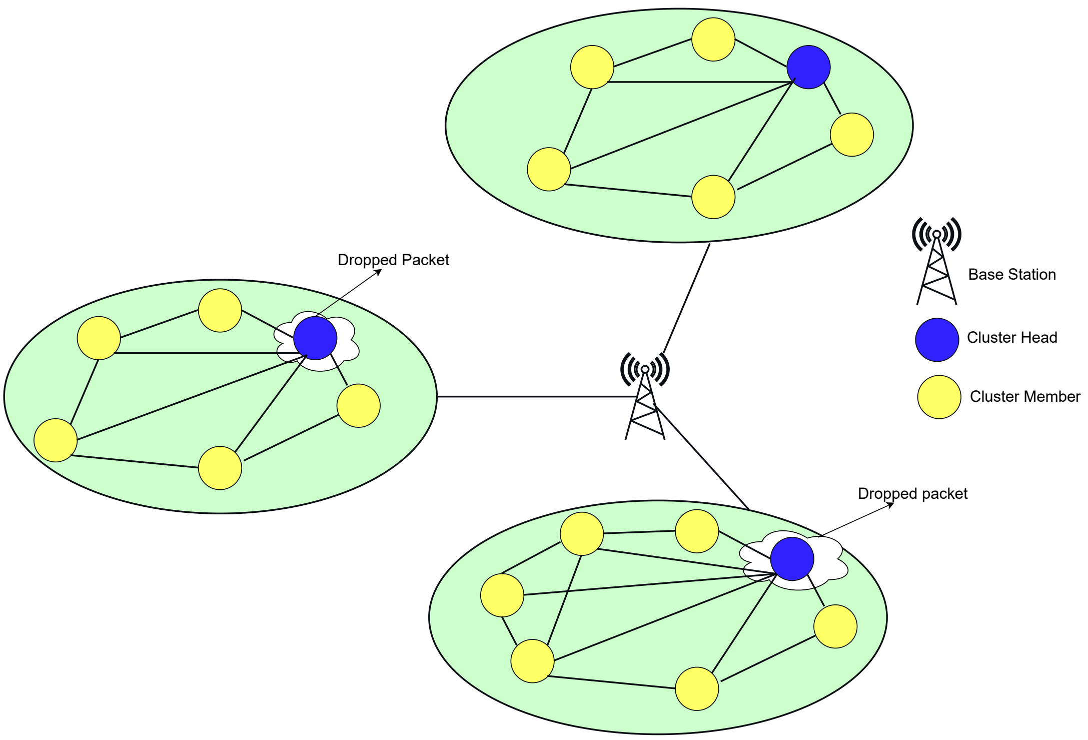

Figure 1. Network Model

absence of quality of service mechanisms, load balancing issues, and inadequate network planning are the primary causes of congestion. Therefore, to maximize the lifespan of the network, protocols created for the Internet of Things must be scalable and lightweight.

4.1.2. Congestion Control

The proposed method is broken down into two parts: the first part assigns the cluster head the task of clearing the congestion path and forms a cluster. To avoid delays and choose an alternate cluster head as soon as the current cluster head migrates away, a modified bully search algorithm is used to select a new cluster head. While implementing this algorithm these are some of the assumptions to be made

(1) Priority Assignment: To ascertain the preference order during the leader election process, it gives each IoT node a distinct priority. This priority is identified based on their RSSI Value.
(2) Communication: Ensure that election messages are correctly sent and received across nodes by using a 6LoWPAN protocol.
(3) Security: Take into account security factors to safeguard the communication channels and stop unauthorized nodes from interfering with the leader election process.

The proposed CA-CH algorithm is essentially the same as the bully search algorithm with extra features such as assigning a priority to each node based on its RSSI value, which will automatically activate if congestion occurs. This RSSI value indicates the signal strength. A good signal indicates that the device can stay in the network for an extended time and send packets without experiencing congestion.

Algorithm 1. initiates that all nodes will calculate and store their initial RSSI values. The node with the highest RSSI values will initiate communication with the other nodes in the network by making an announcement.

Initially, the cluster is created by the cluster head, the cluster head is elected based on the priority of each node. These priorities are assigned based on RSSI value. In the initial state, node communicates with each other, if any congestion occurs in the cluster head, the next highest node will become the cluster head and divert all the packets to that node. The next highest node is decided by implementing the BULLY algorithm. When the cluster knows that a new cluster head is formed all packets will start transmitting towards it. During this process, it should also be seen that there is no congestion in the channel. To decide this the BUG algorithm is used to decide whether there are any obstacles in the channel or not. If any obstacles that is congestion are found then it takes a different path to reach its destination.

Apart from controlling the congestion, this algorithm works best in

(1) Node Failure Detection
(2) Handling Multiple Initiations
(3) Initiating the Election

Algorithm 1 : Cluster Formation

---

input : Set of IoT devices
output: Highest priority node will be considered as cluster head
begin

Initialization: All nodes will be assigned priority. Priority is assigned based on RSSI value;
Node makes local decisions to determine where they should become cluster head;
Cluster Head is decided based on the highest priority

Cluster Formation:
Nodes that decide to become cluster head broadcast this information to other nodes
Based on signal strength the non-cluster head node connects to this node

Data aggregation:
Cluster head will aggregate data from its member nodes and transmit it to the Base station

Rotation of cluster head:
Once the cluster head reaches the threshold value or if the cluster head dies, the next highest priority cluster node will be considered as Cluster Head

end

To prevent and manage congestion, this algorithm 2 is made to dynamically modify transmission power and cluster configurations in a network of Internet of Things devices. Here is a detailed breakdown of the algorithm

(1) Initialization:
Priority Assignment: Every IoT device, or node, has a priority assigned to it. This priority is determined by the RSSI value.
Total Load Calculation: By adding up the individual loads of each device in a

cluster, the algorithm determines the overall load of the network.

(2) Load Calculation: Current Load Calculation (CurrentLoadCH): By adding up the loads of all the devices in the cluster, the current load of the CH is determined.
CurrentLoadCH ← sum(devices.load for all devices in the cluster)

(3) Threshold Check and Congestion Detection: Threshold Value: The threshold value of 100 is the default value.
Congestion Detection: Congestion within the cluster is detected by the algorithm if the CurrentLoadCH is greater than the threshold (100).
If the CurrentLoadCH is greater than the threshold value then
Implement Bully Algorithm: To deal with the congestion, the Bully algorithm is implemented. To choose a new cluster head or leader in the IoT network, the Bully algorithm is commonly employed.
Cluster Formation: By using Algorithm 1, the algorithm creates clusters using a predetermined technique, potentially rearranging the network to more evenly distribute the load.
Path Finding Using the BUG Algorithm: To ensure effective communication within the network, the BUG algorithm is used to find a non-congested path for data transmission.

(4) Communication Within the Cluster
Within the Cluster Communication: Devices within the cluster communicate normally if congestion is not detected or once it has been addressed.

(5) Load Adjustment
Transmission Power Adjustment: A five-unit reduction in the cluster's transmission power occurs if the CurrentLoadCH is greater than 90 but less than the 100 threshold. By taking this proactive action, the load will be decreased and the congestion threshold won't be reached.
In our findings, The algorithm computes the current load on the network and identifies possible congestion to maintain effective communication and load balancing within an IoT network.
   (a) If congestion is found, selecting a new cluster head via the Bully algorithm.
   (b) Finding different, less-crowded routes for data transmission with the help of the BUG algorithm.
   (c) proactively managing the load and avoiding congestion by modifying the cluster's transmission power.
   (d) Based on real-time load calculations, the algorithm dynamically modifies transmission power and cluster configurations to guarantee optimal network performance even in the face of fluctuating load conditions.

5. Simulation And Performance Evaluation

The suggested algorithm's performance is examined with simulations in the Cooja and Contiki operating systems. Additionally, plots numerical assessments and discuss the rationale behind the enhancements in the suggested work. Nodes 5, 6, 8, 20, and 30 have all had their priorities set to 1 for illustration as their RSSI values are high; all other modes are taken to have a standard priority mode. Table 3 provides the corresponding simulation parameters and protocol stacks used in the experiment. The results are compared with the CEDAR technique. According to (4) author, two types of networks are used to estimate the metrics:

Algorithm 2 : keeping track of the communication load

input : Set of IoT devices
output: Load calculation of the network
begin
(6) **Initialization:** All nodes will be assigned priority. the total load will be calculated;
(7) threshold value is default which is assigned to ;
CurrentLoadCH ← sum(devices.load for all devices in the cluster)
(8) threshold ← 100
if *CurrentLoadCH > threshold* **then**
Devices detect congestion within its cluster and initiate the BULLY algorithm;
Cluster Formation happen using this Algorithm 1;
BUG algorithm is invoked to find the non-congested path;
end
Within the cluster, communication occurs
end
if *CurrentLoadCH > 90* **then**
clust.transmission_power= -5;
clust.data_rate=-2;
end

time-driven and event-driven. A WSN with event-driven acquisition only gathers packets in response to network events; in contrast, a WSN with time-driven acquisition gathers packets continuously regardless of external events that take place in the network. The author (4) also expresses that time-driven acquisition uses more energy to continuously acquire packets, CEDAR in event-driven acquisition can withstand a longer lifespan because it only acquires packets when an event occurs. Furthermore, the sink node preserves its energy during events whereas time-driven acquisition continuously depletes it. But, based on a predetermined schedule, processes or actions are initiated. Actions are synchronized according to time intervals rather than always being driven by events. Hence, most priority should be given to time-driven activities.

Table 3. Cooja simulation parameters

Parameter	**Value**
Node Type	Sky mode
Transmission range	50 m
Radio medium	UDGM-Distance loss
Transmission ratio	100 %
Receiving ratio	95%
Inference range	100m
Operating system	Contiki 3.0

The quantity of data that can be transferred in a specific amount of time between two points is known as the network throughput, or zero. The rate of data transmission can be calculated by dividing the number of packets (Pi) acquired during simulation time (T) by the total number of packets acquired. Eq. 1 provides a mathematical representation of this.

The average rate of packet transmission or reception over the network during the computing period is measured by this formula. Usually, the outcome is

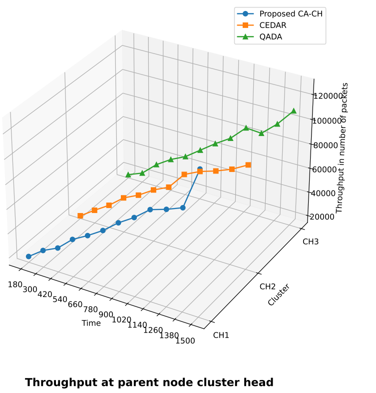

Figure 2. Throughput with varying simulation time

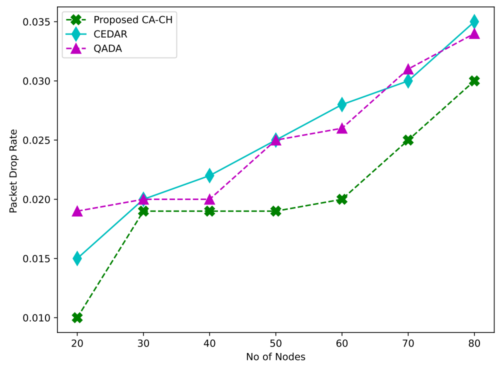

Figure 3. Packet loss rate according to various numbers of sensors

stated in packets per time interval, like packets per second. Figure. 2 illustrates the throughput of the proposed method for different simulation times when compared to the current method.

$$Throughput = \frac{|Pi|}{T} \tag{1}$$

The time it takes to receive a packet (p) from S1 to Sn is referred to as the packet's latency L. The objective of latency is to reduce the overall amount of time required for a packet to travel from its origin to its destination. Eq. 2 provides a mathematical representation of latency where d_i is the latency defined at each hop from S1 to Sn, and n represents the number of intermediate nodes between source to destination.

$$\mathcal{L} = \mathcal{L}_t(p) + \mathcal{L}_q(p) + \mathcal{L}_{pd}(p) \tag{2}$$

Numerous factors, including transmission $\mathcal{L}_t$, queuing $\mathcal{L}_q$, and processing delays $\mathcal{L}_{pd}$ at each network intermediate point, can have an impact on latency. Hence, average latency is computed as shown in Eq. 3

$$\mathcal{L}(a) = \sum_{i=1}^{n} (\sum_{p=1}^{S} (\mathcal{L})) * d_i \tag{3}$$

The aggregate energy consumption of each node and the network's overall performance are both determined by the Energy Loss (EL). The Average Energy Loss (AEL) is used to find the network's most energy-efficient nodes and decide which cluster head node should be used more frequently. This lowers the total energy consumption and guarantees that the network operates at maximum efficiency. Eq. 4 formula for calculating average energy drain, typically it is expressed in Watts (W) or milliWatts (mW).

$$AEL = \frac{\text{Total Energy Consumption}}{\text{Time Period}} \tag{4}$$

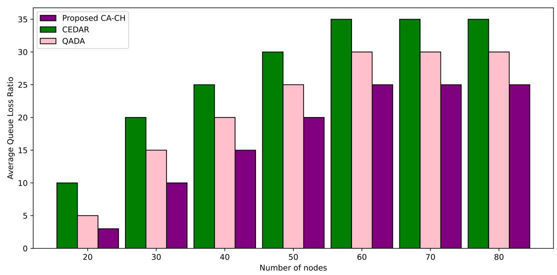

Figure 4. Average Queue Loss Ratio Vs Number of nodes

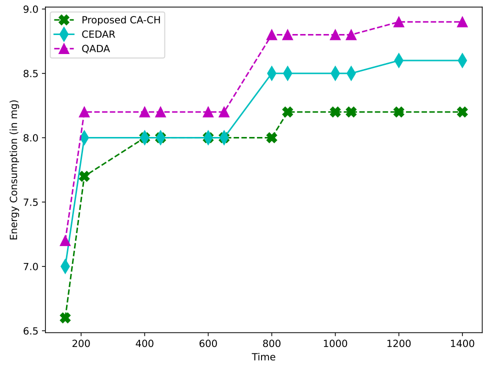

Figure 5. Energy consumption with varying simulation time

Total Energy Consumption is calculated as the summation of CPU energy E(CPU), Low Power Mode (LPM), energy during transmit operation (T), and energy during listen operation (L). Energy is measured using Eq. 5. Figure. 5 displays a graph illustrating the evolution of energy consumption.

$$\text{Total Energy Consumption} = E_{(CPU)} + E_{(LPM)} + E_{(T)} + E_{(L)} \tag{5}$$

Figure. 4 displays the buffer overflow results with the node variable present. Buffer overflow is reduced as the number of nodes rises. As the number of nodes increases, traffic is dispersed among them, and buffer overflow is decreased because of the constant amount of traffic. The findings indicate that when compared to CEDAR, the percentage of data loss resulting from buffer overflow is 3.2% lower in CA-CH. CA-CH offers methods to identify congestion and change between parents in addition to controlling it. The creation of a balanced DODAG is also successful.

$$\text{Buffer Overflow} = \frac{\sum \text{Number of packet dropped}}{\sum \text{number of packet sent}} * 100 \tag{6}$$

Figure 3 illustrates the variation in the packet drop rate with node count. The result shows the recommended CA-CH achieves the lowest packet drop rate (i.e., the lowest congestion) in comparison to CEDAR. This is because, when delivering data content, the suggested CA-CH technique transmits data to a congestion-less cluster head.

Figure 6 shows how throughput varies over time with a different cluster head. Comparing it to two other techniques, namely CEDAR and QADA, reveals that the proposed CA-CH technique has a higher throughput in both cases because it implements priority assignment, which assigns priorities based on RSSI values. Figure 7 shows a flat trend line that indicates a stable and well-performing network. It shows consistently low delay values when compared to the other two

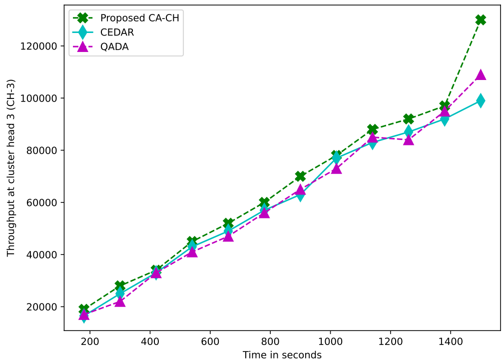

Figure 6. Throughput with different cluster head

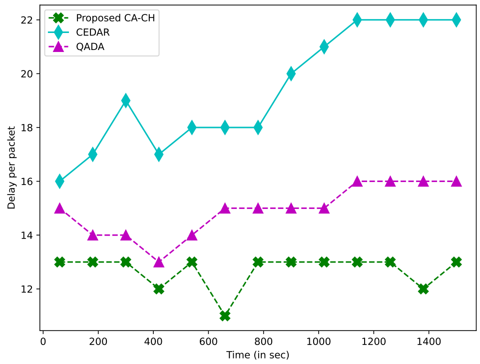

Figure 7. Average delay per packet

techniques. Spikes or sudden increases in latency are signs of network instability or congestion. These may be brought on by factors like increased traffic. The recommended algorithm has flipped the cluster head during that time and can observe a sudden decrease in network latency at times 650 and 1400. Figure **??** represents that all nodes are communicating through cluster head to sink node.

6. Conclusion

Utilizing clustering algorithms is becoming essential for accomplishing the overarching objectives of sustainability, efficiency, and dependability and a strategy for mitigating congestion as the Internet of Things develops. Proposed CA-CH, a novel IoT technique that uses the BUG and Bully algorithm, is proposed in this paper. Two stages make up CA-CH. The cluster is formed in the first phase using the priority-based bully algorithm, and if congestion arises in the second phase the BUG algorithm is used to be redirected to the non-congested path, and the cluster head dynamically switches. By implementing these two heuristic techniques the proposed method assures that there is a successful data transmission in the IoT network. The Cooja simulator in Contiki OS is used to test this technique. The obtained results suggest that the proposed algorithm CA-CH has reduced delay and increased performance, as well as greater use of network resources. The CA-CH algorithm has a variable simulation time but increases throughput while using less energy. In a dynamic environment where real-time obstacle avoidance is required, the suggested techniques find non-congested paths for every node to reach its destination. Future research will consider identifying the most optimal paths in this dynamic environment.

References

[1] H. Allioui and Y. Mourdi, "Exploring the full potentials of iot for better financial growth and stability: A comprehensive survey," *Sensors*, vol. 23, no. 19, 2023.

[2] P. Anitha, H. Vimala, and J. Shreyas, "Pqtba: Priority queue based token bucket algorithm for congestion control in iot network," in *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, pp. 1–7, IEEE, 2023.

[3] P. Anitha, H. Vimala, and J. Shreyas, “Comprehensive review on congestion detection, alleviation, and control for iot networks,” *Journal of Network and Computer Applications*, p. 103749, 2023.

[4] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, “Cedar: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic,” *Peer-to-Peer Networking and Applications*, vol. 16, no. 1, pp. 189–209, 2023.

[5] H. Rahman, N. Ahmed, and M. I. Hussain, “A qos-aware hybrid data aggregation scheme for internet of things,” *Annals of Telecommunications*, vol. 73, pp. 475–486, 2018.

[6] J. Li, B. N. Silva, M. Diyan, Z. Cao, and K. Han, “A clustering based routing algorithm in iot aware wireless mesh networks,” *Sustainable cities and society*, vol. 40, pp. 657–666, 2018.

[7] K. Praveen and P. J. Prathap, “Energy efficient congestion aware resource allocation and routing protocol for iot network using hybrid optimization techniques,” *Wireless Personal Communications*, vol. 117, no. 2, pp. 1187–1207, 2021.

[8] T. A. Al-Janabi and H. S. Al-Raweshidy, “Optimised clustering algorithm-based centralised architecture for load balancing in iot network,” in *2017 International Symposium on Wireless Communication Systems (ISWCS)*, pp. 269–274, 2017.

[9] H. M. Hassan and R. Priyadarshini, “Optimization techniques for iot using adaptive clustering,” in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 766–771, 2021.

[10] Z. Zhang, M. Ma, and Y. Yang, “Energy-efficient multihop polling in clusters of two-layered heterogeneous sensor networks,” *IEEE Transactions on Computers*, vol. 57, no. 2, pp. 231–245, 2008.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pp. 10–pp, IEEE, 2000.

[12] D. Jia, H. Zhu, S. Zou, and P. Hu, “Dynamic cluster head selection method for wireless sensor network,” *IEEE Sensors Journal*, vol. 16, no. 8, pp. 2746–2754, 2015.

[13] M. Gavhale and P. D. Saraf, “Survey on algorithms for efficient cluster formation and cluster head selection in manet,” *Procedia computer science*, vol. 78, pp. 477–482, 2016.