Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

Clustering Based Congestive Traffic Aware Routing Algorithm (CBCTRA) For Effective Traffic Management And Congestion Control In Wireless Sensor Networks (WSN)

[1]M. Savitha, [2]Dr. B. Jayanthi

[1]Assistant professor, [2]Associate professor& Head
[1]Department of Computer Applications, [2]Department of Computer Science (PG)
[1] Kaamadhenu Arts and Science College, [2]Kongu Arts and Science College
[1]Sathy, Erode, [2]Erode
[1][2] Tamil Nadu, India.

Abstract – The Clustering Based Congestive Traffic Aware Routing Algorithm (CBCTRA) integrates the Leaky Bucket and Token Bucket algorithms for effective traffic management and congestion control in Wireless Sensor Networks (WSN). This hybrid approach considers both incoming and outbound traffic rates, utilizing Leaky Bucket at the network entrance and Token Bucket at intermediate nodes. Additionally, the algorithm introduces cluster-based flow control, empowering cluster heads to manage intra-cluster traffic. Its adaptability to dynamically adjust parameters based on congestion signals and intra-cluster traffic patterns ensures optimized performance. Experimental results affirm the algorithm's efficiency in achieving optimal traffic flow and congestion mitigation across the network.

Keywords: WSN, Clustering Based Congestive Traffic Aware Routing Algorithm, Leaky Bucket and Token Bucket algorithms, traffic management and congestion control;

1. Introduction

The WSN is a network that comprises a large network of sensor nodes that are spread over the large area. These sensor nodes are autonomous and send their sensing data in the form of the data packet to the base station (or sink).Traffic pattern of these sensor nodes is depending on the sensing data that research sense [5]. According to the event that research, a large set of data is generated which are transferred in the form of data packets to the sink and utilizing WSN a bunch of promising answers for guarantee the supportability of shrewd applications like Internet of Things (IoT) applications and military applications has been illustrated.

The high generation rate of data becomes uncontrolled to handle and cause congestion in the network [6]. The following two fundamental categories can contribute to WSN congestion: A. Link Collision: When multiple active nodes within the range start transmitting data to the sink at once, collisions on the link happen because there are too many data packets on the link [7]. B. Buffer Overflow –When a node is unable to handle the rate at which it receives the packet, it starts to drop the packet. This packet dropping leads to creating routing hole in the network and packet retransmission. This type of congestion is buffer overflow or node level congestion. The congestion avoidance is to try to avoid the congestion and congestion mitigation is to take a reactive action to alleviate it [10].

Congestion has emerged as a primary concern as the demand for wireless sensor networks (WSNs) due to their numerous applications has grown [8]. Therefore, using a novel cluster based method proposed as a more feasible scheme for such networks

1.1 Problem Statement

The most challenging congestion mechanisms are congestion Avoidance, detection and alleviation. This paper proposes a Clustering Based Congestive Traffic Aware Routing Algorithm (CBCTRA) that combines the strengths of the Leaky Bucket and Token Bucket algorithms for traffic control and congestion management [9]. Cluster-based flow control is an extension of the hybrid algorithm for traffic control and congestion management in networks, specifically taking into account the clustering inside the network.

2. Literature Survey

2.1 Congestion-Aware Clustering and Routing (CCR) protocol

Mohammed Farsi (2019) et.al proposed a congestion-aware clustering and routing (CCR) protocol to alleviate the congestion issue over the network [1]. This paper proposed a novel protocol for mitigating congestion and clustering in WSN. The proposed CCR protocol consists of two main phases; the setup phase and the transmission phase. A small-setup phase is used in rounds other than first round to remove dead nodes and solve dead clusters problem. The results of the experiments show how well the CCR protocol satisfies the quality of service (QoS) standards by extending network lifetime and sending more packets overall. Additionally, the CCR surpasses other cutting-edge methods in reducing data overflow, resulting in a decrease in network bandwidth utilisation.

2.2 Hybrid Cluster Congestion Aware method (HCBCA)

V.Perumal (2018) et.al proposed Hybrid Cluster Congestion Aware method is concentrated on the buffer management and Packet retransmission in WSN [2]. The objective is providing high transmission of packet delivery ratio has been improved the network lifetime performance with respect to time, at the same time packet losses have reduced by packet retransmission. The hybrid cluster based congestion aware (HCBCA) technique that is being presented focuses primarily on traffic that impacts continuous data flow, arrival of data from the source to destination delay time, avoid packet losses, and energy consumption process. The process of transferring packets from the sensor node to the CH in a many-to-one way is mostly congested within the cluster. Congestion is primarily caused by communication paths, node energy levels, and node buffer sizes. When the aforementioned are completed successfully, there is no congestion; otherwise, there will be congestion.

2.3 Congestion-aware Load Balancing Scheme (CLS)

Azlan Awang (2014) et.al proposed Congestion-aware energy efficient and traffic Load Balancing Scheme (CLS) for routing in WSN [3]. This method makes use of the information that was overlooked during the route discovery process and takes into account a composite measure that includes the amount of energy E that was consumed, the node's level of participation P, and the signal strength S of the link between the nodes. In addition, a separate field is maintained in the packet for each routing metric in the case of multiple metrics that might overload the node. The proposed scheme compares the proposed routing metric over a new route discovery mechanism, using weighted additive composition approach and lexical approach. The optimum next hop is selected based on a combination of three different metrics such as energy E, participation level P and signal strength S during forward route formation. Using this approach, a least congested and an energy efficient route is discovered that maintains the minimum routing information. Furthermore, this approach increases the PDR, decreases the energy consumption and an ETE delay of the entire network

2.4 Hop-by-hop traffic-aware routing

Majid Gholipour (2015) et.al proposed a distributed traffic-aware routing scheme is proposed that multi-sink wireless sensor networks with the ability to change node data transmission rates be used to efficiently transfer traffic from source to sink nodes [4]. Our algorithm's architecture incorporates the creation of a hybrid virtual gradient field employing depth and normalised traffic loading for routing, balancing the best available paths with potential congestion on routes leading to those sinks. The simulation results show that the proposed scheme can enhance network resource utilisation, decrease needless packet retransmission, and greatly enhance WSN performance.

3. Proposed Methodology

This paper proposes a Clustering Based Congestive Traffic Aware Routing Algorithm (CBCTRA) that combines the strengths of the Leaky Bucket and Token Bucket algorithms for traffic control and congestion management. Cluster-based flow control is an extension of the hybrid algorithm for traffic control and congestion management in networks. A method of regulating the volume of traffic transmitted to the network is called traffic shaping. It sends a consistent stream of traffic to the network. The network capacity restrictions and the host's packet priority may affect this consistent rate. This constant rate, which will be used to transfer traffic to the network, can be negotiated by the host or sender with the network provider when the connection is formed. Every network guarantees each host a specific amount of bandwidth. We use this to establish this limit. One of the most popular methods for avoiding network congestion is traffic shaping. The network can manage traffic more effectively when it is aware of the intended type. There are two primary algorithms that can be used to achieve traffic shaping. Leaky Bucket and Token Bucket algorithms are those.

3.1 Leaky Bucket Algorithm

As a congestion-control method, the leaky bucket algorithm temporarily stores a large number of packets. A fixed rate is agreed upon by the sender and the network for the transmission of these packets to the network. In data networks, this method is used to achieve traffic shaping-based congestion control. Let's talk about the analogy of a leaking bucket first in order to grasp the method. Consider a bucket that has a tiny hole at the bottom. Imagine now that the bucket is filled with water sporadically. The amount of water thrown into the bucket at each interval is variable.

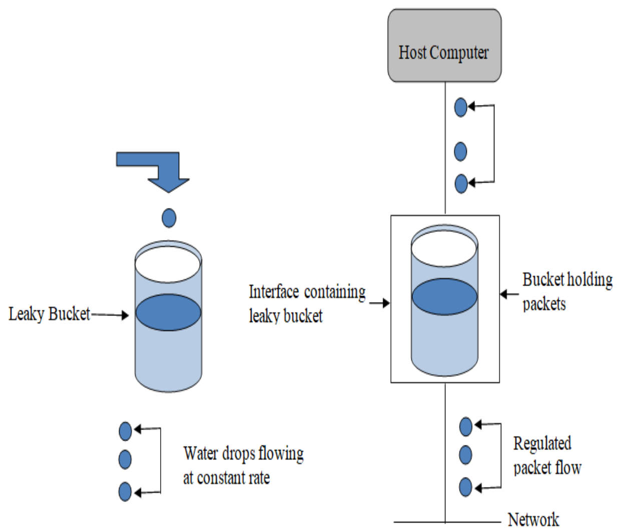

Figure 1. Leaky bucket

The volume of water in the bucket has no bearing on how quickly water leaks out (known as the leak rate). The water spilled will be lost if the bucket fills up. The data packets can be thought of in the same way as a leaky bucket. Think about how each network interface is like a bucket with a leak. Now, packets are dropped into the bucket when the sender wants to transmit them. The bucket located at the network interface fills up with these packets. When the bucket is filled, the packets are lost when the buckets throw them away. This bucket will continue to leak at the same rate. This indicates that the rate at which the packets are sent to the network will remain constant. The Leak Rate or Average Rate is the name given to this continuous rate. The leaking bucket works to transform erratic traffic into steady, smooth flow. The performance of the network is improved by queuing and releasing the packets at periodic intervals.

3.2 Token Bucket Algorithm

In contrast, the token bucket method permits a variable output rate for data packets. The rate at which packets are passed to the network depends on the size of the packets (i.e. the rate or size of bursts into the network). Tokens are stored in the buffer, an abstraction of the buckets; in order for the host node to transfer data, it must first obtain the token, after which the token is destroyed. In order for the host to be able to send huge amounts of data, it remains idle and capture and save up tokens.

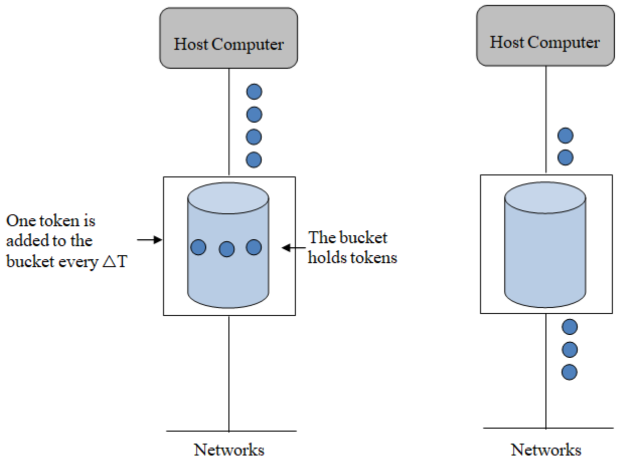

Figure 2. Token Bucket

When considering the implementation of traffic shaping techniques into computer network systems, it is paramount to have an understanding of network system architecture. The algorithm is first embedded into the network architecture.

3.3 Proposed Clustering Based Congestive Traffic Aware Routing Algorithm (CBCTRA)

Leaky Bucket algorithm and the Token Bucket algorithm can be used together in a network to provide comprehensive traffic control and congestion management. The Leaky Bucket algorithm can be applied at the source or network entrance point to regulate the average rate of incoming traffic. It ensures that the overall traffic entering the network does not exceed a predetermined limit, preventing congestion at the network ingress. On the other hand, the Token Bucket algorithm can be implemented at intermediate nodes or routers within the network to control the rate of outgoing traffic. Each node or router can maintain a token bucket for each outgoing connection or queue, allowing for bursts of traffic while still enforcing an overall rate limit. By combining these algorithms, it can achieve a more fine-grained control of the traffic flow in the network. A technique to regulate the average rate of traffic entering the network is provided by the Leaky Bucket algorithm, preventing an excessive load. The Token Bucket algorithm complements this by regulating the rate of outgoing traffic at each intermediate node, ensuring fair sharing of network resources and preventing congestion within the network. Using both algorithms in conjunction allows for a more effective management of traffic and congestion control in the network, providing a balance between controlling the average traffic rate and accommodating short-term bursts of data transmission.

In a network setting, the hybrid algorithm combining the Leaky Bucket and Token Bucket approaches can be implemented to control congestion and manage the flow of traffic. Cluster-based flow control is an extension of the hybrid algorithm for traffic control and congestion management in networks. In this approach, the network is divided into clusters, with each cluster having a designated cluster head responsible for managing intra-cluster traffic. The algorithm initializes the parameters for the Leaky Bucket and Token Bucket components as before, while also establishing the cluster structure. The Leaky Bucket algorithm is applied at the network entrance to regulate the average rate of incoming traffic, while the Token Bucket algorithm is implemented at intermediate nodes and cluster heads to control the rate of outgoing traffic. The below figure shows the proposed method workflow,

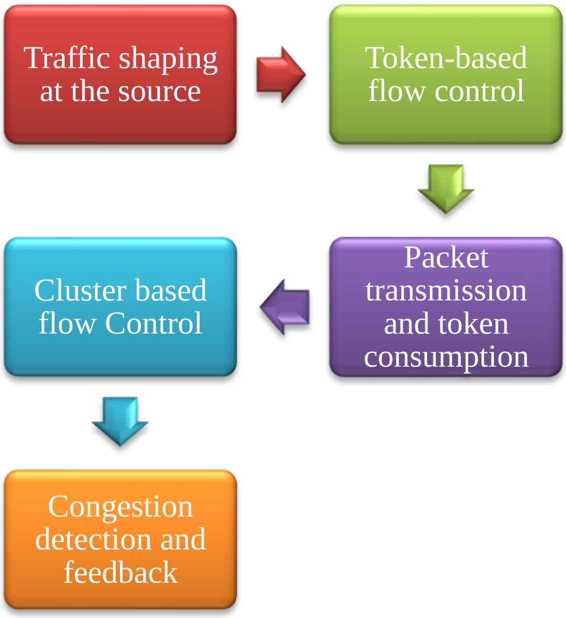

Figure 3. Proposed CBCTRA workflow

Here's how it can work in a network:

Traffic shaping at the source: The Leaky Bucket component of the algorithm is applied at the source node or network gateway. It regulates the rate at which packets are sent into the network. If the incoming traffic exceeds a certain threshold, the Leaky Bucket ensures that packets are transmitted at a controlled rate, preventing sudden bursts of traffic that could lead to congestion.

Token-based flow control: The Token Bucket component is implemented at intermediate nodes or routers in the network. Each node maintains a token bucket associated with each outgoing connection or queue. The size of the token bucket and the token generation rate are configured to control the maximum rate of outgoing traffic from each node.

Packet transmission and token consumption: When a packet is ready to be transmitted from a node, it needs to consume tokens from the corresponding token bucket. The number of tokens required for transmission depends on the size of the packet. If the token bucket has enough tokens, the packet is sent.

Cluster based flow Control: The network is divided into clusters, with each cluster having a designated cluster head responsible for managing intra-cluster traffic. The algorithm initializes the parameters for the Leaky Bucket and Token Bucket components as before, while also establishing the cluster structure. The Leaky Bucket algorithm is applied at the network entrance to regulate the average rate of incoming traffic, while the Token Bucket algorithm is implemented at intermediate nodes and cluster heads to control the rate of outgoing traffic. With cluster-based flow control, cluster heads have additional control over the flow of traffic within their respective clusters, allowing them to adjust the algorithm's parameters based on intra-cluster traffic patterns and congestion signals.

Congestion detection and feedback: The network continuously monitors various congestion indicators such as packet loss, delay, or buffer occupancy. If congestion is detected in the network, congestion signals or feedback are sent to the nodes. Based on this feedback, the parameters of the hybrid algorithm, such as the leak rate, token generation rate, or bucket sizes, can be dynamically adjusted to adapt to the changing network conditions and alleviate congestion.

Algorithm: Clustering Based Congestive Traffic Aware Routing Algorithm (CBCTRA)

Step 1: Set the leak rate (R) for the Leaky Bucket algorithm.

Step 2: Set the token generation rate (G) and token bucket size (B) for the Token Bucket algorithm.

Step 3: Configure the network's maximum allowable traffic rate (M).

Step 4: Establish the cluster structure, including cluster heads and member nodes.

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

Step 5: Track the rate of incoming traffic at the network entrance.
Step 6: If the rate exceeds the maximum allowable traffic rate (M), initiate congestion control measures.
Step 7: At the source or network entrance point:
Step 8: Initialize a leaky bucket with a capacity equal to the leak rate (R).
Step 9: As packets arrive, check if the leaky bucket can accommodate them.
Step 10: If the bucket has sufficient space, add the packet to the bucket.
Step 11: If the bucket is full, mark the packet for potential future transmission.
Step 12: At each intermediate node or router:
Step 13: Initialize a token bucket capacity of B & a token generation rate of G.
Step 14: Periodically generate tokens based on the token generation rate.
Step 15: When a packet is ready to be transmitted:
Step 16: Check if the token bucket has enough tokens to accommodate the packet.
Step 17: If sufficient tokens are available, transmit the packet and consume the corresponding tokens.
Step 18: If the token bucket is empty, delay or queue the packet until enough tokens become available.
Step 19: Assign a cluster head to each cluster, responsible for managing intra-cluster traffic.
Step 20: Allow the cluster heads to regulate the flow of traffic within their respective clusters.
Step 21: Implement the hybrid algorithm (Leaky Bucket and Token Bucket) at the cluster heads and member nodes.
Step 22: The cluster heads can adjust the parameters of the algorithm based on the intra-cluster traffic patterns and congestion signals.
Step 23: Continuously monitor network conditions for signs of congestion, such as packet loss or buffer occupancy.
Step 24: If congestion is detected:
Step 25: Transmit congestion signals or feedback to the relevant nodes or cluster heads.
Step 26: Adjust the parameters of the hybrid algorithm dynamically based on the congestion feedback, both at the cluster level and node level.
Step 27: This could involve reducing the leak rate (R) or token generation rate (G) to alleviate congestion and manage the traffic flow effectively.
Step 28: Repeat steps 2-6 as needed to maintain traffic control and manage congestion in the network, considering both the overall network traffic and intra-cluster traffic.

4. Experimental Results

4.1 Packet Delivery Ratio (PDR)

Table 1. Comparison Table of Packet Delivery Ratio (PDR)

No of Nodes	CCR	HCBCA	Proposed CBCTRA
100	60	72	81
200	63	74	83
300	65	76	86
400	73	79	89
500	75	82	94

The comparison table 1 of Packet Delivery Ratio (PDR) addressed the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and proposed method values are higher than the existing method. The existing values start from 60 to 75 and 72 to 82 and proposed CBCTRA values start from 81 to 94. The proposed CBCTRA gives the best result.

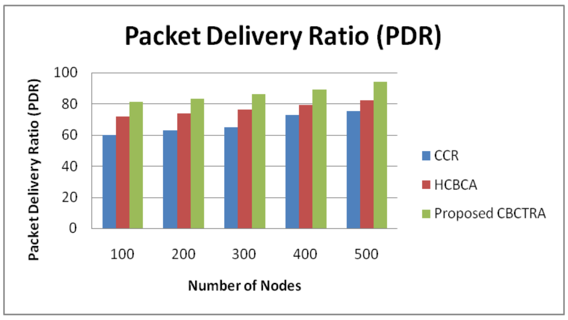

Figure 4. Comparison chart of Packet Delivery Ratio (PDR)

The figure 4 data Packet Delivery Ratio (PDR) describes the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and the proposed method values are higher than the existing method and No of Nodes in x axis and Packet Delivery Ratio (PDR) in Y axis. The existing values start from 60 to 75 and 72 to 82 and proposed CBCTRA values start from 81 to 94. The proposed CBCTRA gives the best result.

4.2 Throughput

Table 2. Comparison Table of Throughput

No of Nodes	CCR	HCBCA	Proposed CBCTRA
100	52	63	70
200	55	65	75
300	60	70	80
400	65	75	85
500	70	80	90

The comparison table 2 of Throughput describes the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and proposed method values are higher than the existing method. The existing values start from 52 to 70, 63 to 80 and the proposed CBCTRA values start from 70 to 90. The proposed CBCTRA gives the best result.

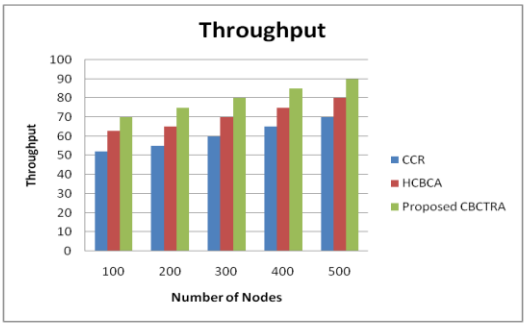

Figure 5. Comparison Chart of Throughput

The figure 5 data Throughput describes the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and the proposed method values are higher than the existing method and No of Nodes in x axis and throughput in Y axis. The existing values start from 52 to 70, 63 to 80 and proposed CBCTRA values start from 70 to 90. The proposed CBCTRA gives the best result.

4.3 End to End Delay

Table 3. Comparison Table of End to End Delay

No of Nodes	CCR	HCBCA	Proposed CBCTRA
100	56	43	32
200	65	70	50
300	75	80	65
400	88	83	73
500	93	85	67

The comparison table 3 of End to End Delay describes the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and proposed method values are higher than the existing method. The existing values start from 56 to 93 and 43 to 85 and proposed CBCTRA values start from 32 to 73. The proposed CBCTRA gives the best result.

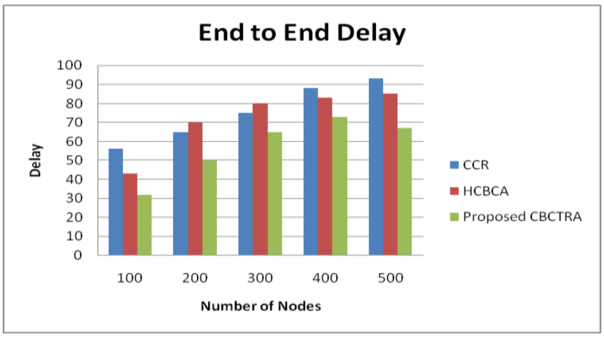

Figure 6. Comparison Table of End to End Delay

The figure 6 data Delay describes the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and the proposed method values are higher than the existing method and No of Nodes in x axis and End to End Delay in Y axis. The existing values start from 56 to 93 and 43 to 85 and proposed CBCTRA values start from 32 to 73. The proposed CBCTRA gives the best result.

4.4 Remaining Energy

Table 4. Comparison Table of Remaining Energy

No of Nodes	CCR	HCBCA	Proposed CBCTRA
100	100	100	100
200	80	85	90
300	60	75	80
400	40	62	70
500	20	45	60

The table 4 comparison of Remaining Energy describes the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and proposed method values are higher than the existing method. The existing values start from 100 to 20, 100 to 45 and proposed CBCTRA values start from 100 to 60. The proposed CBCTRA gives the best result.

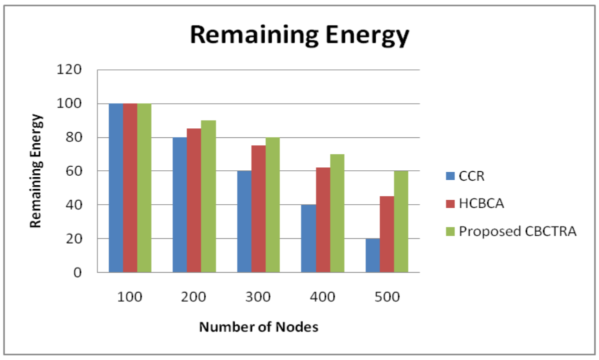

Figure 7. Comparison Chart of Remaining Energy

The figure 7 data Remaining Energy describes the different values of existing (CCR, HCBCA) and proposed CBCTRA. While comparing the existing and the proposed CBCTRA method values are higher than the existing method No of Nodes in x axis and Remaining Energy in Y axis. The existing values start from 100 to 20, 100 to 45 and proposed CBCTRA values start from 100 to 60. The proposed CBCTRA gives the best result.

5. Conclusion

The Leaky Bucket and Token Bucket algorithms are combined in this paper to create the Clustering Based Congestive Traffic Aware Routing Algorithm (CBCTRA). In order to take into account the network's cluster structure, the proposed algorithm is further improved with cluster-based flow control. The algorithm offers a complete solution for traffic management and congestion control by regulating the pace of incoming traffic at the network entry and managing the rate of outgoing traffic at intermediate nodes and cluster leaders. The experimental results show how well the suggested algorithm performs in achieving optimized traffic flow and congestion avoidance. By including cluster-based flow control and taking into account intra-cluster traffic patterns and congestion signals, localized traffic management is elevated to a new level. The hybrid method with cluster-based flow control improves resource utilisation and network performance overall, making it an important tool for traffic management and congestion control in network systems.

References

[1] Mohammed Farsi, Mahmoud Badawy , Mona Moustafa, Hesham Arafat Ali, And Yousry Abdulazeem (2019), “A Congestion-Aware Clustering and Routing (CCR) Protocol for Mitigating Congestion in WSN” 2169-3536 2019 IEEE. Translations and content mining are permitted for academic research, Digital Object Identifier 10.1109/ACCESS.2019.2932951.

[2] V.Perumal and Dr.K.Meenakshi Sundaram (2018), “Hybrid Cluster Based Congestion Aware (HCBCA) Approaches in Wireless Sensor Network”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 16, No. 3, March 2018.

[3] Azlan Awang, Omer Chugtai, Nasreen Badruddin “A Congestion-Aware and Energy efficient Traffic Load balancing Scheme for Routing in WSNs” Conference paper in Telecommunication System, IEEE oct-2014.

[4] Gholipour M, Haghighat AT, Meybodi MR. Hop-by-hop traffic-aware routing to congestion control in wireless sensor networks. EURASIP Journal on Wireless Communications and Networking. 2015 Dec;2015(1):1-3.

[5] Alkharashi, A. (2016). Wireless & Telecommunication. 2nd International Conference and Business Expo . Saudi Arabia: Open Access Jorunal.

[6] Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H., & Jacobson, V. (2016). Congestion-Based Congestion Control. acmqueue.

[7] Hu, X., & Guo, W. (2016). Congestion Control in Wireless Software Defined Networks with Propagation Delay and External Interference: A Robust Control Approach. Hindawi Publishing Corporation.

[8] Hussain, M. W., Jamwal, S., & Zaman, M. (2015). Congestion Control Techniques in a Computer Network. International Journal of Computer Applications.

[9] Jan, Mian Ahmad, Priyadarsi Nanda, Xiangjian He, and Ren Ping Liu. "PASCCC: Priority-based application-specific congestion control clustering protocol." Computer Networks 74 (2014): 92-102.

[10] Kafi, Mohamed Amine, DjamelDjenouri, Jalel Ben Othman, AbdelraoufOuadjaout, MiloudBagaa, NoureddineLasla, and NadjibBadache. "Interference-aware Congestion Control Protocol for Wireless Sensor Networks." Procedia Computer Science 37 (2014): 181-188.