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

NMFLRED: Neuro-Multilevel-Fuzzy Logic RED Approach for Congestion Control in TCP/IP Differentiated Services

Sunil Kumar Kushwaha[1], Dr. Suresh K. Jain [2]

Submitted: 16/09/2023 **Revised**: 18/11/2023 **Accepted**: 29/11/2023

Abstract: The problem of congestion is ubiquitous and will remain forever, which is not limited to the field of networking but also in another field. The rapid expansion of the internet especially after corona pandemic in addition to the development of new computer technology such as ChatGPT, real time video and audio have accelerated the exponential increase in high-speed computer networks. The number of computers supporting more and more applications using the network has led to a significant increase in the number of packets passing across those networks, which has resulted in resource contention and ultimately leads to congestion. Thus, a solution needs to be drawn out which works for different prioritize packet with different forwarding and dropping probabilities. A neural network for self-adaptive and learning while Fuzzy logic for multi-variable linguistic calculation is used to generate an algorithm which is proposed in this research paper. A method using Multilevel Dropping method for different types of packets has been proposed which clearly shows that overall performance has been largely increased.

Keywords: *Congestion, Fuzzy, Neural, DSCP, Delay, Packet loss, RED*

1. Introduction:

The problem of congestion had made its presence since the day of inception and will remain forever. The requirement of network has changed from time to time as it has been found that after corona pandemic data is more of video rather than of conventional data. Thus, the management of the network is not stagnant but is getting varied. In the efficient management of networks, flow control and congestion control are one of the important parameters to get it managed in an efficient manner. Also, it has been observed that Congestion and Flow both are ancestor of each other. Congestion occurs when the input port buffer capacity fails to manage the incoming packets, i.e. forwarding speed is less than the rate at which packet is coming. Even if the rate matches still Congestion will occur. Congestion lowers bandwidth consumption, worsens network performance, and increases delay and packet loss. There are multiple ways to manage network congestion, including random early detection (RED), that is widely popular and widely used with practically implemented ways to reduce the issues brought on by congestion. However, the linear behavior and parametrization issues plague RED and its further improvement. In order to address these issues, we put forth a novel technique in this study termed Neuro- Multilevel-Fuzzy Logic RED (NMFLRED), which builds on RED by including fuzzy logic and neural network.

In order to anticipate and prevent congestion at too early stage, the suggested NMFLRED approach depends on the instantaneous average queue length (aql) and the early predicted delay (DSpec). We simulate and assess NMFLRED using a discrete-time queue model. The findings shown that NMFLRED outperformed RED and effective RED (ERED), Gentle RED(GRED), Blue RED(BRED) by reducing latency and packet loss in the presence of significant congestion. In comparison to ERED and RED, NMFLRED reduced delay by up to 3.5 and 6.5% under extreme congestion and packet loss by up to 9 and 40%, respectively. These results imply that NMFLRED is a potential congestion technique that can increase performance while conserving network resources, especially when the traffic is more of real time rather than conventional old traffic.

To overcome the issue of router congestion brought on by the rise in network utilisation and achieve a high-speed transmission, the queue length and load rate should be watched carefully. To maintain good network performance, prior active queue management techniques is used to manage the queued packets in the router buffer. These techniques, however, rely on monitoring indicators that do not account for all congestion symptoms, causing packet loss and delay. In order to prevent global synchronization, loss, and delay, all congestion indications should be controlled by an algorithm that may drop packets randomly.

In this paper, a neural based multilevel fuzzy controlled Multilevel Random Early Detection (NMFLRED) is proposed to deal with the differential services and having different dropping probability as and what required to services. The threshold indicator for various services may be different which further depends on the packets arrival at

Medi-Caps University, Department of Computer Science and Engineering Indore(Madhya Pradesh), India
Sunil.kushwaha@medicaps.ac.in, suresh.jan@rediffmail.com

routers interface, the departure rate and the instantaneous queue length. Accordingly, a fuzzy inference engine needs to be developed to act on the arrival packets and further calculate the variable dropping probability in coordination with neural network for adaptive learning the behaviour of network. Simulation results on very large data clearly shows that NMFLRED improve loss and packet dropping probability (Dp) with high utilization of network especially when the network is bursty and more of real time video. While calculating the loss of packets the simulation shows that at low rate of traffic, there may be a chance of drop of packets while at heavy load compared with other existing or proposed methods such as RED, GRED, BRED, FRED, ERED, FCRED it has been reduced to 0.08. The throughput of channel has increased tremendously to 1.5 while dropping of packets has been reduced to 0.08, a 25% improvement to other algorithms. The suggested NMFLRED outperforms other implemented or proposed fuzzy-based approaches by dropping packets which are of less priority rather to high priority packets to prevent loss, which will be made only in the case is essential dropping, while achieving better results that are comparable to those of the latter.

The primary network resources, connections, and routers serve as the intermediaries for the data that is exchanged and transported from site to host over the internet. Congestion is defined as an increase in data rate that was too large in quantity for the network's resources to handle, especially in routers, firewalls as well as UTM. A host using the TCP protocol to send data avoids congestion by using a process called widowing, which modifies the sending speeds in accordance with the network's condition as judged by the timing and volume of acknowledgements received. Other protocols do not follow this method, such as User Datagram Protocol and protocols used in wireless as well as mobile and ad-hoc networks. Congestion at the network routers or firewalls may result from transmitting the packets from inside port to outside port by making queue, filtering, stacking and making variable queue even in using TCP protocols [1–5].

The well stabilized procedures already in applications for route-based congestion control is Active Queue Management (AQM). To minimize packet loss and delay, AQM techniques identify and manage congestion at primarily stage and begin dropping packets. In order to avoid congestion, these systems rely on estimating the dropping probability, ***Dp,*** for each arrival packet. ***Dp*** is then utilized to decide whether to accept or reject the packet. Random Early Detection (RED), Adaptive RED (ARED), Gentle RED (GRED), Adaptive GRED (AGRED), Dynamic GRED (DGRED), BLUE RED(BRED), dynamic threshold-based BLUE (DT BLUE), GREEN, Adaptive CHOKE, ERED, Subsidized RED (SubRED), Fuzzy Logic RED(FLRED) and others are examples of the current AQM approaches [6]. The average queue length (*avgaql*), which is an mark for the current queue state, is used by RED [7], which is the backbone of all AQM approaches, and other leading AQM methods. The mean value is assessed at various levels, each of which is constrained by a different threshold. If the position in the buffer designated by ***minThreshold*** is not surpassed by average, the buffer can be thought of as having a reasonable number of packets in the queue. When the ***maxThreshold*** parameter is achieved or surpassed by the average, congestion may be present. As a result, ***Dp*** rises when the ***maxThreshold*** for the queued packets is reached.

By tagging or deleting packets at an early stage with a probability value ***Dp***, RED prevents congestion. The declining likelihood depends on the average. Therefore, when the average is below the minimum threshold, RED does not drop any packets. While, if the average exceeds *maxThreshold*, all incoming packets are dropped. When average oscillates between *minThreshold* and *maxThreshold*, true congestion management is put into place. The following are RED's benefits:

(i) As it responds to the increase in packet queuing in the buffer by discarding packets, RED forecasts congestion in the early stages. RED prevents global synchronization by haphazardly dumping packets. Red deals with *avg* as a congestion indicator rather than *q,* RED circumvent by deleting unnecessary packets when a brief period of bursty traffic is present (false congestion). When fake congestion occurs, *avg* does not grow quickly. The Internet Engineering Task Force (IETF) has successfully adopted RED in RFC 2309. However, as technology has advanced quickly, issues with RED have emerged. These issues can be summed up as follows [8]:

Calculated Average Queue Length A_{ql}
The Instantaneous average queue length calculated over a finite time frame I_{ql}

The variation in Queue Length calculated over a finite time period i.e. the differential value in the two consecutive time slots ΔQ

The Packet Loss calculated at a fixed time interval P_L

Predicted arrival rate over a fixed time A_R

In a calculated time frame the mean Arrival Rate A_{ar}

Predicted load rate over a specific time period L_r

The mean packet needs to be serviced over a calculated time frame A_{lr}

Due to RED's use of average rather than q, it is insensitive to the current queue status, which causes packet loss in the event of sudden congestion. Be aware that when there is a brief period of high traffic, taking q into account will cause packets to be dropped needlessly. The RED's parameter initialization issue, which has an impact on the programme's functionality.

Because RED does not account for arrival and departure rates, it could cause a delay. (traffic-load amount). In order to overwhelm the insensitivity to the present queue size, the Effective RED (ERED) [5] was devised. Along with the average, ERED employs the instantaneous queue size (q) as a congestion marking flag. The following scenarios are used to control congestion: When *q* is more than *minThreshold* and average is between *minThreshold* and *maxThreshold*, ERED removes arrival packets with *Dp* determined in RED. While ERED eliminates arrival packets with *Dp* as determined in RED when average is less than *minThreshold* and q is less than 1.75* *maxThreshold*.

Similar to RED, GRED, and ARED, the AQM methods operate in many situations controlled by if-else. Other AQM techniques manage the cases with a fuzzy system and transform the issue into a fuzzy inference procedure. The inference procedure facilitates the clear judgement made using the prior set of techniques [32]. Due to the benefit of decision fuzziness, fuzzy-based AQM systems can extend into several situations and incorporate different indicators. Different fuzzy-based methodologies have been put forth [32, 35]. These techniques have improved the outcomes under specific traffic conditions, solved the parameterization problem, and given the created techniques more flexibility. The inability to use thorough congestion indicators to optimize network performance is a drawback that the fuzzy-based methods share with the nonfuzzy methods [31].

In order to prevent loss and delay, all congestion indications should be maintained by a random packet-dropping mechanism that is integrated within the monitoring indicators. In order to close the gap in network monitoring and congestion control at the router buffer, a Neuro-Multilevel-Fuzzy Logic RED (NMFLRED) is presented in this research. Table 1 provides a summary of the issue this research looked into, and the contributions made. Three indicators that track the arrival, departure, and queue length of the router are used to construct the NMFLRED. The integrated arriving flow and the integrated departing flow are these indications. The indicators are computed, and the fuzzy system uses these as inputs to generate the *Dp*.

Item	Description
Research Gap	Lack of thorough indicators with an appropriate regulating procedure that improves queue management, minimum packet loss, high throughput and multilevel treatment of packets at the router buffer to improve network performance. Less research is carried out on real time video traffic specially the scenario generated after corona pandemic and usage of ChatGPT.
Goal	Make a variable length queue for various differential packets arriving over the same time period, mark the dropping length as a variable quantity. The dropping indicator may be selected, by using fuzzy inference engine to calculate the dropping probability with a self-learning and adaptive by using neural network. The goal is to maximize by keeping in overview of loss, delay, and dropping rate.
Methodology	A variable Queue length is allotted, for multivalued packets with variable arrival rate, and exit rate will be used to identify the indicators across time and to provide the input for a fuzzy inference procedure using the appropriate fuzzy rules, while neural will help in adaptive learning with the help of central node. A variable dropping equation is derived as per priority defined.

Table 1

Network bandwidth, latency, and packet loss rate—performances that are also of relevance to users—are used in the evaluation of a congestion control system. However, as the operation of the complete network system is tied to congestion control, we conduct our review while considering all components of the system.

We will mostly talk about the congestion control protocol assessment index in the sections that follow.

1) Stability. Assuming that each user is self-centered, the balancing point is stable from the perspective of congestion control after the network architecture reaches Nash Equilibrium [1]. If such a balance point is not unique, it will cause the balance point to vary and fail to meet the requirements for stability. The stability was ensured by Nash Equilibrium's existence and special characteristics.

2) Fairness: If there are n users sharing a resource, fairness requires that each user receive the same amount of the resource (unless the user sought less than its fair share). Using the following equation from Reference [2], fairness can be measured even if throughput is not perfectly equal. where xi is the throughput for user it.

$$Fairness = \frac{(\sum_{i=1}^{n} xi)^2}{n \sum_{i=1}^{n} xi^2}$$

3) Effectiveness. When the distribution of resources is Pareto optimum, no user can achieve better satisfaction through a different distribution of resources without also lowering the satisfaction of the other users. Therefore, Pareto optimum will ensure that congestion control algorithms are valid. Power can be used to measure how effectively resources are allocated in Reference [3].

$$\text{Utilization}=\frac{Goodput}{Response\ Time}$$

4) Stability. It indicates that even when a control system's parameters jitter, the system can continue to function normally. The scheme must function in an unreliable (random) environment in order to be robust. Therefore, plans that assume a specific distribution (exponential) for service times or plans that only function for deterministic service times were disregarded.

5. Scalability. With regard to bandwidth, access, user count, and other factors, the internet has advanced quite quickly. The physical expansion, such as from a local area network (LAN) to a wireless network link, must be supported by a congestion management protocol. Additionally, it must be able to receive a variety of expansion possibilities.

6. Feasibility. The technology requirements are all included in the feasibility. Due to the complexity of the hardware and software systems used by the Internet, congestion management protocols must incorporate a number of distinct technologies. Additionally, in order for an algorithm to be feasible, the congestion control protocol must be sufficiently straightforward and easy to implement.

2. Related Works

The main and most prominent issue in networks is congestion. As a result, various scientists have created and suggested improved congestion control techniques. In 1993, RED was put forth [14]. With the help of this process, the tail-drop method's drawbacks, such as congestion at the fast stage in the buffer, are resolved.

Before the router buffer fills up, the congestion is managed using the RED approach using a dropping policy. Therefore, the RED approach uses an *aql* computed value and has binary level thresholds (*minthreshold* and *maxthreshold*) established at the ingress port of buffer in the router. The following guidelines determine how the packet is dumped. All packets are forwarded when the *aql* is below the minimum threshold. The packets at ingress port of router start to be dropped with a low probability when the *aql* lies between the *minthreshold* and *maxthreshold*. The dropping value changes to unity and every packet which arrives at ingress port of router may overflow when the number of packets exceeds the *maxthreshold* [14]. As a result, the RED method yields superior performance results compared to the tail-drop method and reduces the bursty traffic issue.

One more algorithm named AQM technique called GRED [15] improves the root RED algorithm method, which fixes the parameterization issue, and reduces the bursty traffic nature [22, 23]. To minimize the congestion occurred by using two separate level dropping probabilities, the GRED technique, in contrast to the RED approach, uses three level thresholds (min*threshold*, *maxthreshold*, and *doublemaxthreshold*) at the ingress port of the router.

The dropping probability follows the guidelines as mentioned in the paragraph highlighted below. The instant active queue length a_{ql} value is below the minimum threshold, dropping doesn't happen. According to an equation identical to the RED method's, the GRED technique removes the packet if the a_{ql} value is between the *minthreshold* and *maxthreshold* [14]. In order to keep the router ingress buffer from getting filled up when the a_{ql} value is between the *maxthreshold* and *double maxthreshold*, GentleRED employs a different mathematical equation for dropping at a higher value of probability. Finally, the dropping grows until it hits 1 as with the RED technique [15] when the a_{ql} exceeds the *doublemaxthreshold*.

There will be possibility that all such packets which now enter the ingress port of router may suffer from overflows and hence may be dropped. As a result of the *doublemaxthreshold* addition, the result calculated clearly shows that the GRED technique performs much better than RED. In [10], the DGRED technique is proposed. DGRED uses the same number of thresholds as the GRED approach. As opposed to GRED, DGRED uses target a_{ql} allocated at specified levels to maintain the a_{ql} value between *minthreshold* and *maxthreshold* and leverages adaptive threshold locations to improve performance. As was evident, a_{ql} was used in all prior AQM techniques. Thus, although the actual queue length may sometimes be low, the a_{ql} value may occasionally be high.

Additionally, AQM procedures are framed on the a_{ql} value in accordance with the dropping policy. The true queue length lowers as soon as the processor of router can process traffic quickly. Even if the actual queue length is small, previously received packets are dropped if a_{ql} is high [17, 24]. The majority of AQM techniques, however, employ thresholds and maximum target a_{ql}, $T(a_{ql})$, which causes parameterization issues [18]. Fuzzy logic is accepted as a remedy for minimizing the congestion by generating multivalued threshold value in order to minimize the parameterization issues which arise in AQM approaches [25]. Parameters are not a concern for fuzzy logic [21]. The application of Fuzzy logic was first applied to ATM networks, through which it was found that the efficiency of network has increased while minimizing the packet drop.

Gentle Random Early Detection Fuzzy Logic (GREDFL), which is an improved AQM GRED technique, was proposed by Baklizi et al. Researchers of GREDFL's *aql* and D_L as fuzzy logic input variables. As a result, the GREDFL approach improves the GRED method while decreasing its high level of parameter dependence. The AQM GentleRED technique, that deploys a queue model based on discrete time and FIFO packet arrival probability, that uses the same simulation as GentleREDFL.

The router having variable buffer allocation to variable services and prioritizing queue is shown in fig .1 The tail drop mechanism is shown with *minthershold* and *maxthreshold* value shown with blue color. No priority is allocated to any service and thus the packet which is in tail of queue is dropped. This can be treated as simple mechanism of designing the algorithm. The packet drop in case *maxthreshold* is achieved is shown in fig 2 for RED.

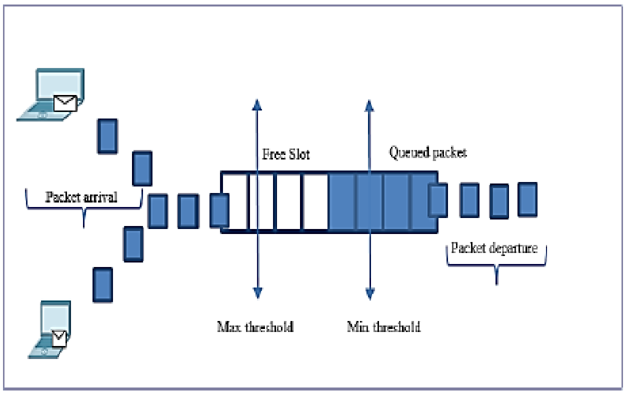

Fig.1 Router Buffer with Tail-Drop

Performance for the GREDFL approach is superior to the GRED method.

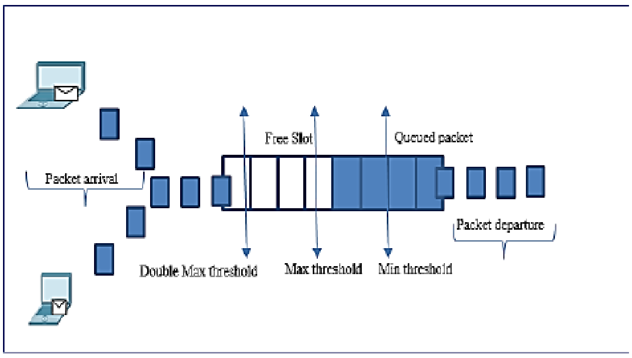

Fig. 2 Packet drop with RED

Fuzzy logic is used in many additional techniques to boost performance of network. Contrarily, as of now no such technique can manage the early stages of congestion to perform better than the present AQM techniques [3, 26, 27]. Unlike GentleREDFL, which uses *aql* and delay P_L, FLACC uses two valued input linguistic variables (PL and aql). One of the most crucial metrics that has a minimal impact on network performance is packet loss. Additionally, the suggested FLACC achieves superior performance results than existing AQM[32] approaches by avoiding parameterization issues and enhancing packet loss and delay performance measures.

3. Proposed Nmflred Method

The suggested **NMFLRED**'s concept and implementation are presented in this section. In NMFLRED, the concept of Neuro-fuzzy has been applied to address and lessen the congestion occurred in the network. In most current AQM techniques, fuzzy logic is mostly used to identify and prevent parameterization issues. Additionally, the suggested solution uses *aql* and P_L as a mark point for the occurrence of congestion in order to improve the performance of AQM[33]. The major factors in predicting buffer congestion are *aql* and PL. The major objective is to use a fuzzy inference process (FIP) to determine the D_P for each packet at ingress port in the router buffer. *Aql* and P_L are utilised to compute DP.

The thresholds used in AQM methods were not considered when designing the FLACC router buffer. As a result, Fuzzy Inference Process is used to calculate the dropping probability, which uses the inputs *aql* and P_L to produce the output D_P (see Fig. 3).

The algorithm proposed named as NMFLRED is created and put into use to get over the drawbacks of existing AQM techniques and further concluded that satisfactory performance outcomes has been occurred[35]. This objective is accomplished in six steps. The settings and performance metrics are shown.

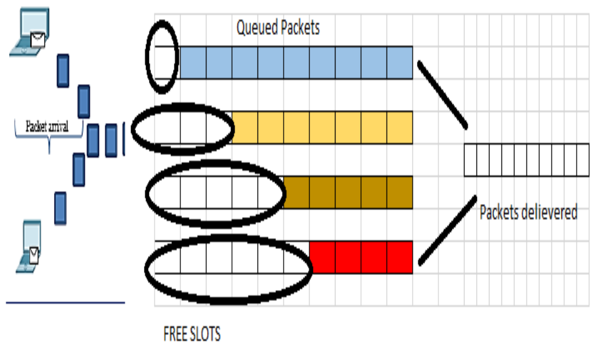

Fig. 3 Packet Drop with NMFRED

The parameters set for matching packets with DSCP value which is programmed is shown below in table 2.

	Assured Forwarding value		DSCP VALUE
af11	AF11	Assured Forwarding	001010
af12	AF12	Assured	001100

		Forwarding	
af13	AF13	Assured Forwarding	001110
af21	AF21	Assured Forwarding	010010
af22	AF22	Assured Forwarding	010100
af23	AF23	Assured Forwarding	010110
af31	AF31	Assured Forwarding	011010
af32	AF32	Assured Forwarding	011100
af33	AF33	Assured Forwarding	011100
af41	AF41	Assured Forwarding	100010
af42	AF42	Assured Forwarding	100100
af43	AF43	Assured Forwarding	100110
cs1	CS1	Precedence 1	001000
cs2	CS2	Precedence 2	010000
cs3	CS3	Precedence 3	011000
cs4	CS4	Precedence 4	100000
cs5	CS5	Precedence 5	101000
cs6	CS6	Precedence 6	110000
cs7	CS7	Precedence 7	111000
default		default dscp	000000
ef	EF	expedite forwarding	101110

Table 2: DSCP value assigned.

The router buffer in the suggested method begins to produce packets that are shaped using a discrete time queue to gauge performance results during a particular time period. The router has a FIFO queue for all packets created by input. The performance metrics and packet loss are computed at stage three. Whether or not the router buffer is empty determines the *aql* value. Equation 1 is applied when the buffer is empty to determine the *aql* value.

$$aql_{t+1} = aql_t + aql(1 - qinst)^{n-1} \quad \text{eq.1}$$

where n stands for the number of packets in idle time.

Equation 2 is used to determine the *aql* value when the router buffer is partially empty, where *q_inst* is the router buffer's current queue length.

$$aql_{t+1} = aql_t + aql(1 - qinst)^{n-1} + qinst \times qcal \quad \text{eq 2}$$

When the router buffer fills up completely or overflows, the value of packet loss is determined. In bursty traffic, as arriving packets enter at ingress port of router, the egress port buffer begins to grow on its own until it meets the router's buffer size threshold. As a result, every packet that arrives can be dropped immediately because no arriving packet can fit in the buffer. The packet loss is demonstrated in Equation 3. where P_b represents the likelihood that the ingress port buffer is filled, and beta represents the likelihood that a packet will leave or depart.

$$Pb = calculated\ value\ of\ x_{mn}\left(\frac{avg - present\ value\ of\ x_{mn}}{x_{mn} ma\, x_{mn} - min\, n_{th}}\right) m^2 \quad \text{eq 3}$$

A scenario is created using Cisco Packet Tracer 8.1.1 simulated on Windows 11 platform and shown in Fig 5. Different types of devices such as conventional Desktop, IP telephony, Video Conferencing devices, server and others are included which will generate multi valued packets as mentioned in table 2.

The time on which arriving packets reach the buffer, the egress port buffer begins to grow on its own until it meets the router's buffer size threshold. Now if the *maxthreshold* is reached as a result, every packet that arrives can be dropped immediately because no arriving packet can fit in the buffer. The packet loss is demonstrated in Equation 4, where P_b represents the likelihood that the buffer is filled and *mn* represents the likelihood that a packet will leave or depart.

$$Pb = calculated\ value\ of\ x_{mn}\left(\frac{avg - present\ value\ of\ x_{mn}}{x_{mn} ma\, x_{mn} - min\, n_{th}}\right) m^2 \quad \text{eq 4}$$

The proposed FLACC fuzzification begins in step four. Fuzzification, rule evaluation, aggregation, and defuzzification are the four phases that make up FIP.

Three variables are used by fuzzification in step 1 to produce the membership degree. These are the variables: *aql, PL, and DP*.

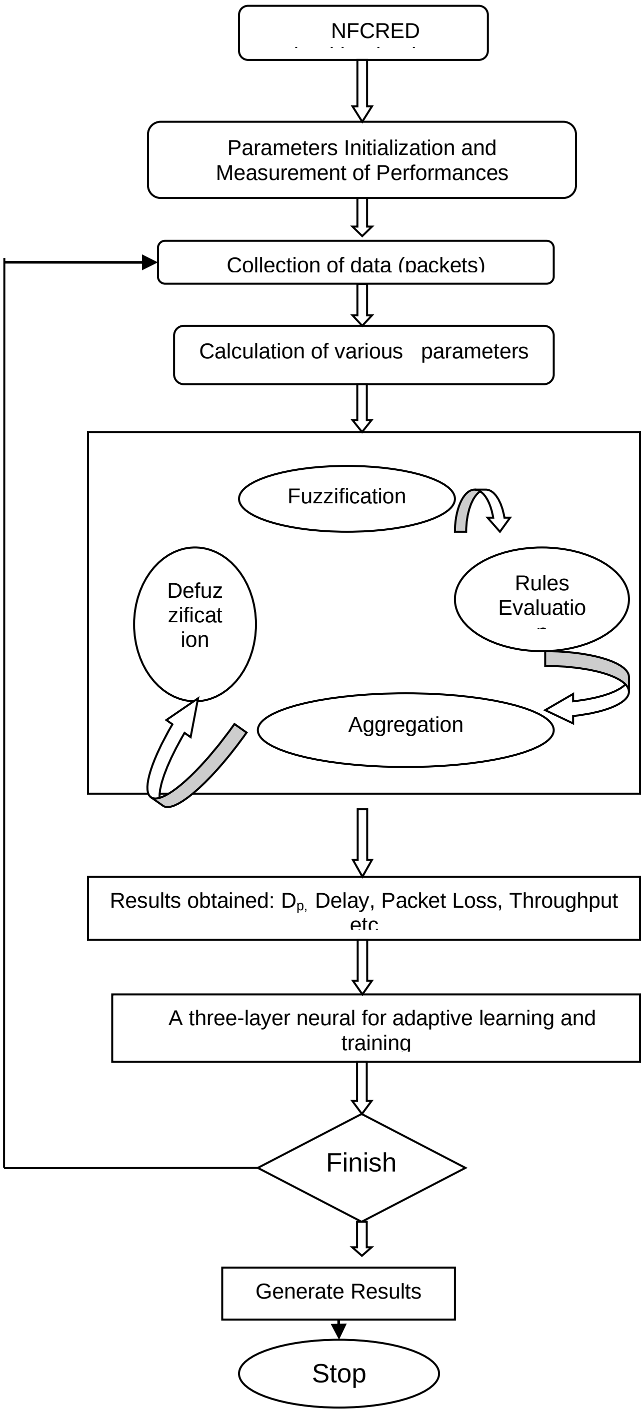

Fig. 4 Buffer of Router in NMFRED

The following fuzzy set is present in each variable:

aql: {below normal, normal, average, extreme, beyond extreme}

PL: {minimum, acceptable, Medium, More, extreme}

DP: {Low, Slight, Medium, Further, beyond}.

$$Pb = calculated\ value\ of\ x_{mn}\left(\frac{avg - present\ value\ of\ x_{mn}}{x_{mn}\, ma\, x_{mn} - min\, n_{th}}\right) m^2 \quad \text{eq 4}$$

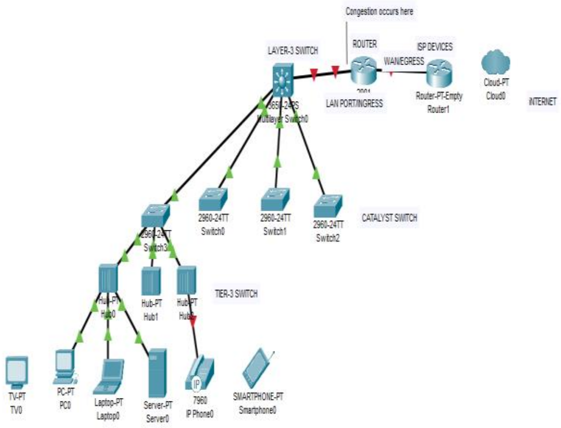

Fig. 5 Scenario for real time packet generation.

The input and output variables are shown in a trapezoidal shape in Fig. 6. The membership degrees for the two input variables and one output variable are shown, accordingly, in Figs. 7 and 8.

For every input linguistic variable, the fuzzy set is computed after the membership degrees are set. The region for each input linguistic variable is formed as the primary objective to calculate the steps for the fuzzification. At stage two, regulations are assessed. The evaluation of the FLACC rules is shown in Table II. Every component is a process that goes through rule assessment to determine its membership in the Dp output.

All the rules which come as output are combined into a unity output rule known as a fuzzy set-in step 3, which also combines the membership degrees from step 2.

Rule 1: if packets is in application 1 queue and queue is in x_{11} status then drop-probability is dr_{pz}

Rule 2: if packets is in application 1 queue and queue is in x_{12} status then drop-probability is dr_{pz}

Rule 3: if packets is in application 1 queue and queue is in x_{13} status then drop-probability is dr_{pz}

Rule 4: if packets is in application 2 queue and queue is in x_{11} status then drop-probability is dr_{pz}

Thus, the rule will be different for different applications based on above tabular method.

Defuzzification is the last stage of FIP and is found in step 4. All of the output variables' precise values are produced in the set value used by fuzzy rules implemented.

In the next step [ref mos], the centre of gravity (COG) is utilized. The primary objective of COG is to locate the fuzzy set's centre for each output variable. Officially, Equation (6) can be used to set COG.

$$COG = \sum_{x}^{y} A(n) \times n \Big/ \sum_{x}^{y} A(n) \quad \text{eq. 6}$$

The Proposed NMFLRED Algorithm

1. Set the parameters for Fuzzy sets and its values

2. Apply Fuzzy Rules and implement

3. For every packet in the queue

   I. Make the busty packet smooth by leaky bucket or token leaky bucket.
   II. Mark the packet and categorise as per dscp value
   III. Assign the bandwidth as per priority.
   IV. Mark the threshold value to each queue.
      (i) This threshold value is different for different dscp value.
      (ii) Mark a number of interval threshold and corresponding dropping value for the assigned queue.
   V. Apply the algorithm as mentioned in the fuzzified statements mentioned below

$dr_{pz_{1axi}}$

is the dropping probability when Application is of 1 level and instantaneous queue length is at level x_{11}.

$dr_{pz_{2axi}}$

is the dropping probability when Application is of 1 level and instantaneous queue length is at level x_{12}

$dr_{pz_{3axi}}$

is the dropping probability when Application is of 1 level and instantaneous queue length is at level x_{13}.

Similarly other dropping probability may be evaluated. Azi =1,2,……m (various dscp levels).

The dropping probability may be calculated as per equation mentioned below:-

(i) The inbound packet will be added to the queue if average is discovered to exist between 0 and the min_{th} threshold(different for different class of service), which is calculated as Pb = 0

The equation will be calculated based on the rules mentioned below:-

(i) min ≤ 0 , no drop

(ii) (x_{13} 〚≤ x_{12}〛 ≥ min) drop is $dr_{pz_{2axi}}$)

(iii) (x_{15} 〚 ≤ x_{14} 〛 ≥ x_{13}) drop is $dr_{pz_{2axi}}$) ……… eq.7

Rest formulation will be calculated similarly based on the below algorithms.

(i) If the average value is discovered to be higher than the min_{th} threshold but lower than the level target threshold, the packet will likely be dropped in the ratio calculated based on equation as:

$$Pb = calculated\ value\ of\ x_{mn} \left(\frac{avg - present\ value\ of\ x_{mn}}{x_{mn} max\, x_{mn} - min\, n_{th}} \right) m^2 \quad \text{eq-8}$$

4. Fuzzified the (aql)

5. Fuzzified the probability (PL)

6. Apply the rules (Fuzzy aql, Fuzzy PL)

7. Aggregattion

8. Calculation of the dropping probability based on above equations.

9. Drop the packet with Dp based on the equation mentioned

4. Simulation

This section compares the proposed NMFLRED with previous approaches and explains the basic simulation process. Most AQM techniques use discrete-time queues to calculate performance results at instantaneous moment [28, 29]. The phrase "discrete time slot" was used, which denotes that performance metrics might be computed at any moment.

In the simulation environment, the suggested NMFLRED, FLACC, GRED and FLGRED algorithms were implemented on the ingress buffer port of the router. The router's buffer capacity was oscillated from *minvalue* to *maxvalue* in order to get accumulated and get merge variable length of packets. All packets were routed and queued by using conventional such as FIFO implementations, which caused significant congestion in the router buffer. There was total 5.5 lakh samples collected by using Wireshark Version 4.0.3 (v4.0.3-0-gc552f74cdc23). The first 20 thousand were rejected as initials were in transition stage and found busty in nature and hence remaining were taken for simulation. The data collected were exercised using MATLAB R2019a Neuro-Fuzzy designer. There were two input variables used to train the NFDM. The first one is shown in Figure 7 and describes the DSCP values' membership functions, which combine a traffic class and its priority.

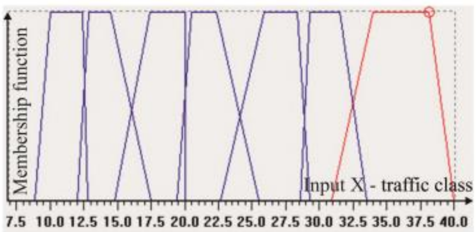

Fig 6: Membership functions used for DSCP values allocating the threshold values

The actual DSCP values of the corresponding range of the standard AF classes are set as the upper angles of the trapezoidal membership functions. There are twenty designated ranges, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3 ……………1. The trapezoidal shapes' overlapping regions are so predetermined that the membership function values are insignificant because no packets with DSCP values above the norm are anticipated. Figure 8 shows representation of the second input variable shows the four ranges that are commonly suggested [17] for selection of minimum and maximum threshold functions

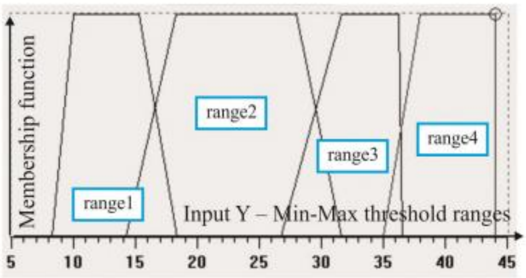

Fig. 7: Membership functions defined for the mostly used values

5. Performance Results

This section highlights the performance assessment of the suggested NMFLRED, FLRED, GREDFL, and GRED techniques. The proposed method is also thoroughly explained in terms of the membership functions mentioned in the fuzzy rules, the various perimeter values, and rules implemented as mentioned above within a set.

To find the best approach, the performance of NMFLRED, FLRED, GREDFL, and GRED was evaluated and compared using four distinct performance measures (mql, throughput, delay, Packet loss , and Dp).

A. Average Queue Length

The effective outcome of the calculated average length of the queue is assessed in this section. The output performances of the suggested NMFLRED, FLRED, GREDFL, and GRED techniques are shown in Fig. 9. Different packets collected with different arrival probabilities are used in the experiment to produce congestion and non-congestion at the router buffer.

In the event that the packet arrival probability which is different for different services is always at higher side in compared to the departure rate probability calculated which may be always different for different values, the overall accomplishment of metric mql discovers the similar outcomes for the proposed as well as contrasted approaches. However, NMFLRED outperforms GREDFL and GRED techniques when the rate likelihood of the arriving packets is high, such as 0.73, 0.88, and 0.98. Therefore, NMFLRED performs better than the all type of existing AQM comparable approach when there was a huge packets occurred at ingress of router buffer which causes a severe congestion.

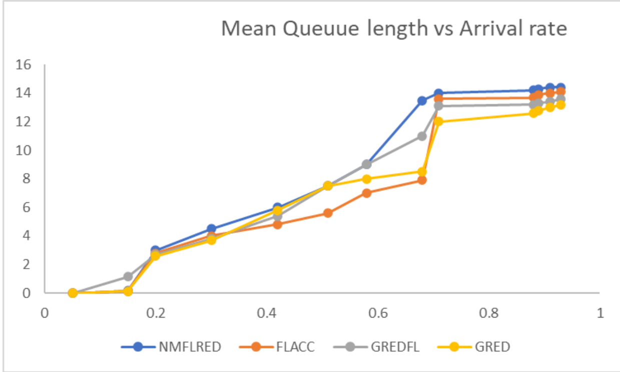

Fig.9: Mean queue length vs Arrival of packets

B. Throughput

Throughput and more of Goodput is the total number of successfully required number of packets that pass the ingress router at a given moment. One key standard for computer networks is the throughput performance metric. The analysis of the results calculated for various algorithms such as NMFLRED, FLRED, GREDFL, and GRED techniques is shown in Fig. 10.

The packets at ingress of router was either less than or greater than the minimum buffer provided for various dscp values in all cases when the probability of alpha was used. However, all of the strategies yielded the same throughput result, which was 0.7. As was noted earlier in the study, this outcome was caused by the initialization of the various dropping probability length for the egress to make the successful packet departure.

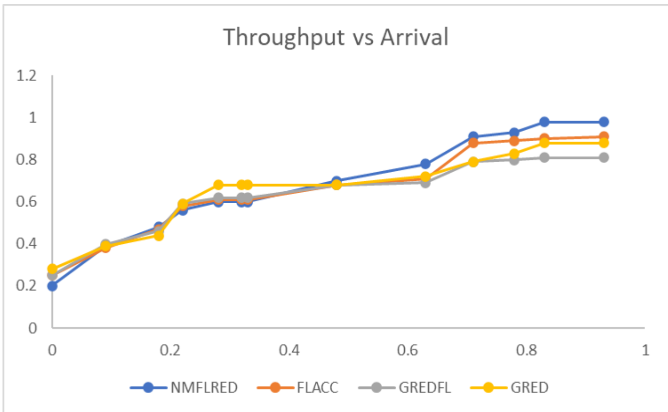

Fig.10: Throughput vs Arrival of packets

C. Delay

Fig. 11 explains the performance of various tested algorithms which result in evaluation of queueing delay. The calculation and evaluation of delays heavily relies on the average queue length calculated as mentioned in previous sections. According to mql and T, the delay is estimated (see Equation 5).

The equation shows that the value of delay is also modest when the value of mql is small.

In both congested and uncongested conditions, NMFLRED beats GREDFL and GRED AQM, as shown in Fig.11. The graph is plotted for the arrival rates of packet approaches 0.6, 0.8, and 0.9, as the suggested technique outperforms the compared methods in terms of performance.

D. The Packet Loss

Every packet that arrives may be dropped immediately and equally irrespective of class it belongs if there is no room left in the router ingress buffer when the received packets reach it, causing the buffer to overflow.

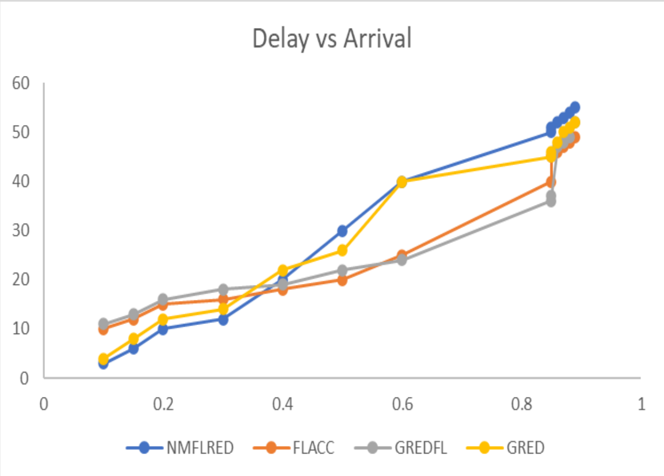

Fig.11: Delay vs Arrival of packets

The throughput in terms of outcomes of the NMFLRED, FLRED, GentleRED Fuzzy Logic, and GentleRED techniques are shown in Fig. 12. When there is significant congestion, the suggested NMFLRED has produced good and minimum P_L performance results. When FLRED, GREDFL and GRED are compared, NMFLRED router buffer overflows less frequently than those of those approaches.

Fig.12 : Packet Loss vs Arrival of packets

E. Dropping Probability

Fig. 13 demonstrates that the NMFLRED approach drops less packets at the router input buffer when it is congested as compared to FLRED, GREDFL or GRED methods. As a result, the throughput has been increased to all types of services which makes more quick service in less service time.

Notably, compared to the FLRED, GREDFL and GRED methods described in the preceding section, the NMFLRED approach drops minimal packets at the ingress port of router buffer.

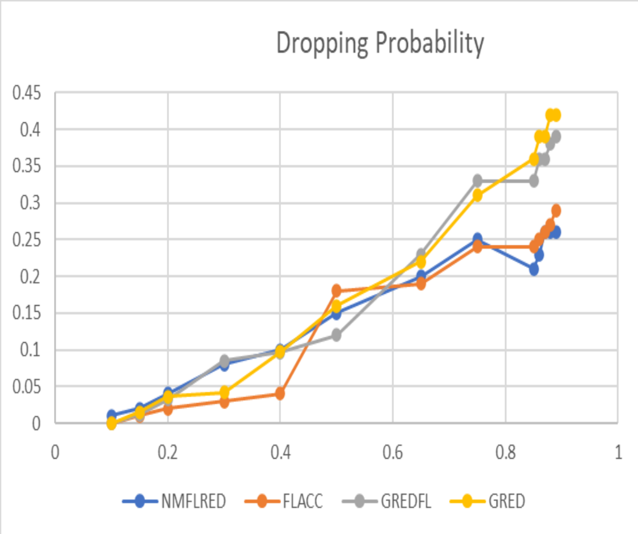

Fig.13: Packet Drop vs Arrival of packets

6. Conclusions

Congestion, which reduces network performance and resources, is one of the key problems that slow down performance and further effects the networks. The consequences include a larger queuing buffer at transmitter end, specially at router which may cause protracted delays, and finally packet loss. This problem has raised exponentially after the corona pandemic and use of OTT platform and ChatGPT. We hereby suggest NMFLRED as a fresh approach to avoiding congestion. NMFLRED is a development of the FLRED AQM approach. Since parameterization is a drawback of existing AQM techniques, NMFLRED seeks to identify and prevent this issue. The use of multilevel queue, with various dropping rules with different marking indicators allows for the early detection and prevention of congestion, before the router buffer is completely used up. The suggested NMFLRED, FLRED, GREDFL, and GRED were tested on a large database through a simulator by employing well-known processes in order to compare and assess their performance outcomes. When there is significant congestion, NMFLRED outperforms FLRED, GREDFL and GRED techniques, according to the evaluation of the simulation findings. Although all the compared approaches acquired the same T in congested or uncongested scenarios, the mql, D, and P_L dropped with the NMFLRED seems much better. Additionally, compared to the planned NMFLRED, the FLRED, GREDFL and GRED techniques dropped more packets (Dp) at initial stage at the input port of router buffers.

7. Future Work

The existing NMFLRED approach will be used in high usage of networks as demand is increasing exponentially day by day in wired as well as wireless areas in future work to mitigate congestion. Making for multilevel may take some time to calculate, but with increasing performance of hardware may compromise with this factor by improving the latency and packet loss in particular to boost the performance of the network. Additionally, in coming days most traffic will be on real time very high density video which will be bursty traffic and such issue is resolved by implementing the multi class queue using the suggested NMFLRED approach.

Author contributions

Sunil Kumar Kushwaha: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, Software, Validation., Field study

Dr. Suresh K. Jain: Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Chandra, E. and B. Subramani, A Survey on Congestion Control. Global Journal of Computer Science and Technology 2010. 9(9): p. 82-87.

[2] Thiruchelvi, G. and J. Raja, A Survey On Active Queue Management Mechanisms. IJCSNS International Journal of Computer Science and Network Security, 2008. 8(12): p. 130-145.

[3] Bazaz, Y., S. kumar, and S. Anand, Congestion Control Mechanismusing Fuzzy Logic. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2013. 2(3).

[4] M, K. and S. G, Congestion control approach based on effective random early detection and fuzzy logic. MAGNT, 2015. 3(8): p. 180-193.

[5] BAKLIZI, M., et al., Markov-Modulated Bernoulli Dynamic Gentle Random Early Detection. Journal of Theoretical and Applied Information Technology, 2018. 96(20).

[6] Abdel-jaber, H., F. Thabtah, and M. Woodward. Traffic Management for the Gentle Random Early Detection using Discrete-Time Queueing. in Information Management in Modern Organizations: Trends & Challenges 2008.

[7] Negnevitsky, M., Artificial Intelligence: A Guide to Intelligent Systems. Second Edition ed. 2005, England.

[8] Stanojevic, R., R.N. Shorten, and C.M. Kellett, Adaptive tuning of drop-tail buffers for reducing queueing delays. Communications Letters, IEEE, 2006. 10(7): p. 570-572.

[9] Brandauer, C., et al., Comparison of Tail Drop and Active Queue Management Performance for Bulk-Data and Web-Like Internet Traffic, in Proceedings of the Sixth IEEE Symposium on Computers and Communications. 2001, IEEE Computer Society.

[10] Baklizi, M., et al., DYNAMIC STOCHASTIC EARLY DISCOVERY: A NEW CONGESTION CONTROL TECHNIQUE TO IMPROVE NETWORKS PERFORMANCE. International Journal of Innovative Computing, Information and Control, 2013. 9(4).

[11] Welzl, M., Network Congestion Control: Managing Internet Traffic. 1 ed. 2005.

[12] Kalav, D. and S. Gupta, Congestion Control in Communication Network Using RED, SFQ and REM Algorithm. International Refereed Journal of Engineering and Science (IRJES), 2012. 12 (2): p. 41-45.

[13] Fan, X., et al. QBLUE: A New Congestion Control Algorithm Based on Queuing Theory. in High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on. 2012.

[14] Floyd, S. and V. Jacobson, Random early detection gateways for congestion avoidance. IEEE/ACM Trans. Netw., 1993. 1(4): p. 397-413.

[15] Floyd, S. Recommendations On Using the Gentle Variant of RED. http://www.aciri.org/floyd/red/gentle.html 2000.

[16] Baklizi, M., et al., Performance Assessment of AGRED, RED and GRED Congestion Control Algorithms. Information Technology Journal, 2012. 11(2): p. 255-261.

[17] Baklizi, M. and J. Ababneh, Performance Evaluation of the Proposed Enhanced Adaptive Gentle Random Early Detection Algorithm in Congestion Situations International Journal of Current Engineering and Technology 2016. 6(5).

[18] Baklizi, m., et al., Fuzzy Logic Controller of Gentle Random Early Detection Based on Average Queue Length and Delay Rate International Journal of Fuzzy Systems, 2014. 16(1).

[19] Abu-Shareha, A.A., Enhanced Random Early Detection using Responsive Congestion IndicatorsInternational Journal of Advanced Computer Science and Applications(IJACSA), 2019. 3(1): p. 358-367.

[20] Abualhaj, M.M., A.A. Abu-Shareha, and M.M. Al-Tahrawi, FLRED: an efficient fuzzy logic based network congestion control method. The Natural Computing Applications, 2016.

[21] Mulla, A.S. and B.T. Jadhav, Fuzzy Based Queue Management Policies–An Experimental Approach. International Journal of Current Engineering and Technology 2014. 4(1).

[22] Baklizi, M., J. Ababneh, and N. Abdallah. Performance Investigations of Flred and Agred Active Queue Management Methods. in Proceedings of Academicsera 13 th International Conference. 2018. Istanbul, Turkey.

[23] Seifaddini, O., A. Abdullah, and A.H. Vosough, Red, Gred, Agred Congestion Control Algorithms in Heterogeneous Traffic Types, In International Conference on Computing And Informatics. 2013.

[24] Baklizi, M., J. Ababneh, and A Survey in Active Queue Management Methods According to Performance Measures. International Journal of Computer Trends and Technology (IJCTT), 2016. 38 (3): p. 145.

[25] Tassiulas, L., Y.C. Hung, and S.S. Panwar, Optimal buffer control during congestion in an ATM network node. Networking, IEEE/ACM Transactions on, 1994. 2(4): p. 374-386.

[26] Kusumawardani, M., Active queue management (aqm) and adaptive neuro fuzzy inference system (anfis) as intranet traffic Control. Academic Research International 2013. 4(5).

[27] Ingoley, S.N. and M. Nashipudi, A Review: Fuzzy Logic in Congestion Control of Computer Network in International Conference in Recent Trends in Information Technology and Computer Science 2012.

[28] Abdel-Jaber, H., et al., Performance evaluation for DRED discrete-time queueing network analytical model. J. Netw. Comput. Appl., 2008. 31(4): p. 750-770.

[29] Woodward, M.E., Communication and Computer Networks: Modelling with discrete-time queues. 1993: Wiley-IEEE Computer Society Press.

[30] Ababneh, j., et al., Derivation of Three Queue Nodes Discrete-Time Analytical Model Based on DRED Algorithm, in The Seventh IEEE International Conference on Information Technology: New Generations (ITNG 2010),USA.2010.2010.

[31] Guan, L., et al., Discrete-time performance analysis of a congestion control mechanism based on RED under multi-class bursty and correlated traffic. Journal of Systems and Software, 2007. 80(10): p. 1716-1725.

[32] D. M. Lopez-Pacheco, C. Pham, "Robust Transport Protocol for Dynamic High-Speed Networks: enhancing the XCP approach" ICON, 2022. http://web.univ-pau.fr/~cpham/Paper/icon05.pdf

[33] E. Altman, K. Avrachenkov, C. Barakat, A.A. Kherani, B.J. Prabhu "Analysis of MIMD congestion control algorithm for High Speed Networks" Computer Networks: The International Journal of Computer and Telecommunications Networking Volume 48 , Issue 6, pp.: 972 – 989, 2021

[34] Sumitha Bhandarkar, Saurabh Jain and A. L. Narasimha Reddy, "Improving TCP Performance in High Bandwidth High RTT Links Using Layered Congestion Control", International Workshop on Protocols for Fast Long-Distance Networks, February 2022. http://whitepapers.silicon.com/0,39024759,60303395p,00.htm

[35] Cheng Jin David X. Wei Steven H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance" IEEE Infocom 2004. http://www.ieee-infocom.org/2019/Papers/52_2.PDF