ITC 2/51

Information Technology and Control
Vol. 51 / No. 2 / 2022
pp. 252-267
DOI 10.5755/j01.itc.51.2.30194

Fuzzy Comprehensive Random Early Detection of Router Congestion

Received 2021/11/24 | Accepted after revision 2022/05/05

http://dx.doi.org/10.5755/j01.itc.51.2.30194

HOW TO CITE: Abu-Shareha, A., Al-Kasasbeh, B., Shambou, Q. Y., Abualhaj, M. M., Al-Khatib, S. N. (2022). Fuzzy Comprehensive Random Early Detection of Router Congestion. *Information Technology and Control*, 51(2), 252-267. http://dx.doi.org/10.5755/j01.itc.51.2.30194

Fuzzy Comprehensive Random Early Detection of Router Congestion

Ahmad Adel Abu-Shareha
Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan; e-mail: a.abushareha@ammanu.edu.jo

Basil Al-Kasasbeh
Faculty of Computer Studies, Arab Open University (AOU), Riyadh, Kingdom of Saudi Arabia; e-mail: bkasasbah@arabou.edu.sa

Qusai Y. Shambour
Department of Software Engineering, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan; e-mail: q.shambour@ammanu.edu.jo

Mosleh M. Abualhaj
Department of Networks and Information Security, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan; e-mail: m.abualhaj@ammanu.edu.jo

Sumaya N. Al-Khatib
Department of Computer Science, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan; e-mail: sumayakh@ammanu.edu.jo

Corresponding author: a.abushareha@ammanu.edu.jo

The queue length and the load rate should be monitored to overcome the problem of router congestion due to the increase in network utilization and achieve a high-speed transmission. Previous active queue management methods manage the queued packets in the router buffer to maintain high network performance. However, these methods depend on monitoring indicators that do not cover all the congestion signs, leading to packet loss and delay. Accordingly, all the congestion signs should be wrapped into these indicators and managed by an algorithm that randomly drops packets to avoid global synchronization, loss, and delay. In this paper, a fuzzy comprehensive random early detection (FCRED) is proposed to deal with the gap in network monitoring and congestion control at the router buffer. FCRED is built by using three indicators, which monitor the router's arrival, departure, and queue length. Accordingly, a fuzzy inference process is developed to manage these indicators and calculate the dropping probability (Dp). Simulation results show that FCRED improves loss and packet dropping under various network statuses compared with RED, BLUE, and ERED. In terms of loss, FCRED achieves zero loss at high congested status. For dropping, FCRED achieves an optimal rate of 0.47 with an arrival rate of 0.95. For the throughput and delay, FCRED achieves the best results. Accordingly, the proposed FCRED method achieves zero loss and reduces packet dropping from 0.28 to 0.21, a 25% reduction compared with the best performance of these methods. Compared with recent fuzzy-based methods, the proposed FCRED achieves comparable results and outperforms them by dropping more packets to avoid loss, which in such case is necessary dropping.

KEYWORDS: Congestion, Random Early Detection, Active Queue Management, Fuzzy Inference Process.

1. Introduction

Active queue management (AQM) methods are mechanisms used to monitor, control, and manage the queued packets at the router buffer. Accordingly, these methods are crucial to the network performance and quality of services (QoS). Random early detection (RED) method [16] was proposed to overcome the problem with firm and nonprediction-based method for queue management, the drop-tail (DT), which is an early approach for queue management [4]. RED has two important features, namely, 1) early detection and 2) random dropping. The early detection mechanism is achieved by monitoring the queue length at the router buffer and averaging the length over time of the so-called average queue length (AQL). AQL is calculated with each packet arrival as a weighted average of the instance queue time (IQL) and the previously calculated value of the AQL. Random dropping is achieved when RED calculates a dropping probability (Dp) based on the value of the AQL. The value of Dp is used for random packet dropping. With the increase of the value of Dp, the chance of packet dropping increases and vice versa. Besides, RED also implemented firm dropping under high load traffic [3].

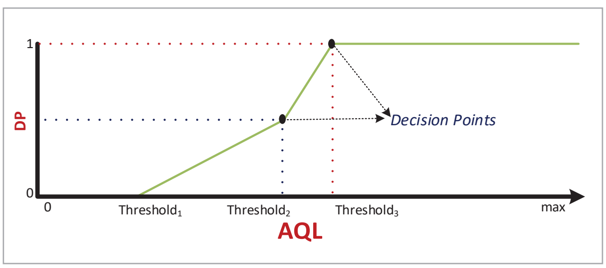
Monitoring indicators, such as the AQL, affect the AQM methods' performance, because they are the basis for calculating the Dp. An example of the relationship between the AQL and the Dp value is given in Figure 1, which is similar to the relationship embodied in the effective RED (ERED) method [1]. Apart from AQL, other monitoring indicators have been used in the existing AQM. A definition of the existing monitoring indicators is given in Table 1.

AQL and IQL have been used extensively in various methods. AQL has been used in RED, ERED, and fuzzy gentle RED (FGRED). IQL indicator has also been used by the stability RED (SRED) [31] and stabilized virtual buffer (SVB) [13], in addition to, RED and ERED. However, these methods used joint monitoring indicators rather than using AQL or IQL independently.

Various examples of using multiple monitoring indicators are presented in the existing AQM methods, such as using IQL and AQL jointly in ERED [1], using AQL and delay jointly in FGRED [10], using arrival

Figure 1

AQL vs. DP Packet Dropping Decisions



rate and IQL jointly in random exponential marking (REM) [22], rate-based AQM (REAQM) [39], SVB [13], stable rate-based AQM (RAQM) [38] and robust active queue management (RaQ) [35].

Table 1

Summary of the Existing Indicators

Indicator	Abb.	Description
Average Queue Length	AQL	Average queue length over a time frame
Instance Queue Length	IQL	The instance queue length at a specific time
The difference in Queue Length	ΔQ	The difference in the queue length between two subsequent time slots
Packet Loss	PL	Estimated packet loss at a specific time due to buffer saturation
Arrival Rate	AR	Estimated arrival rate at a specific time
Average Arrival Rate	AAR	Average arrival rate over a time frame
Load Rate	LR	Estimated load rate at a specific time
Average Load Rate	ALR	Average load rate over a time frame

The AQM methods are operated in multiple cases controlled by if-else, similar to RED, GRED, and ARED. Other AQM methods utilize a fuzzy system to manage the cases and convert the problem into a fuzzy inference process. The inference process eases the crisp decision made in the previous group of methods [40]. The advantage of the decision fuzziness, the fuzzy-based AQM methods enable an easy extension into many cases and straightforwardly use multiple indicators. Various fuzzy-based methods have been proposed [42, 45]. These methods have solved the parameterization problem, added more flexibility to the developed methods, and enhanced the results under certain traffic conditions. However, the problem of the fuzzy-based methods is inherited from the non-fuzzy methods, which is the inability to optimize the network performance using comprehensive congestion indicators [41].

Accordingly, all the congestion signs should be wrapped into the monitoring indicators, managed by an algorithm that randomly drops packets to avoid loss and delay. In this paper, a fuzzy comprehensive RED (FCRED) is proposed to deal with the gap in network monitoring and congestion control at the router buffer. For clarification, the problem investigated in this paper and the contributions are summarized in Table 2. The FCRED is built by using three indicators, which monitor the router's arrival, departure, and queue length. These indicators are the integrated incoming flow and the integrated departing flow. The indicators are calculated and then used as inputs for the fuzzy system, which produces the Dp.

Table 2

Purpose of the FCRED

Item	Description
Research Gap	Lack of comprehensive indicators with a suitable controlling process that optimizes the queue management at the router buffer to optimize the network performance.
Goal	Identify an integrated indicator, build up a framework that wraps these indicators, and use the fuzzy inference process to predict congestion and false congestion for optimizing network performance in terms of loss, delay, and dropping rate.
Methodology	Identify the indicators through three elements, queue length, arrival, and departure rate, over a time frame and use them as input to a fuzzy inference process with suitable fuzzy rules.

2. Previous Work

AQM methods are either crisp-based or fuzzy-based [3]. The crisp-based methods use a set of parameters in their crisp form to calculate the Dp value. An if-else mechanism with multiple cases controls the dropping. An example of the control cases for the GRED [15] is given in Figure 2(a). The fuzzy-based methods use fuzzy inference processes to calculate the Dp. A fuzzy version of the GRED (FGRED) method that is demonstrated in Figure 2(a) is illustrated in Figure 2(b) [10].

Fuzzy-based methods required fuzzification, rule evaluation, integration, and defuzzification process. These processes are standardized regardless of the input and output variables [42]. Existing fuzzy-based methods differ in the input variables, the rules utilized, and the membership functions. The input variables of the fuzzy-based methods are determined by the method's goal, similar to the monitoring indicators in the crisp-based methods. These inputs are commonly represented as mapped from the crisp-based method into fuzzy variables. An example of such mapping is mapping AQL used in RED into IQL and ΔQ input variables in the fuzzy-RED (FRED) [32].

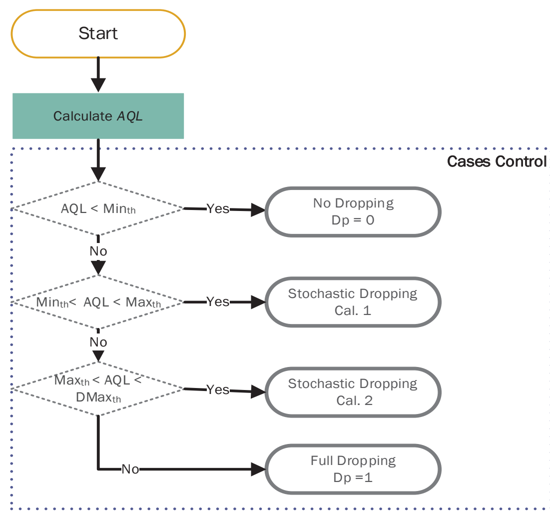
The existing fuzzy-based methods can be classified into queue-based, traffic-based, and hybrid-based. Queue methods utilize a variation of queue length indicators, such as AQL, Q, and ΔQ. FRED, fuzzy explicit marking (FEM) [12], and fuzzy BLUE (FB) [42] use the same indicators, which are IQL and ΔQ. The differences between these methods imply the fuzzification function and the rules utilized, which are commonly established through trial and error approach [8]. The advantage of FEM and FB is reducing packet loss. However, unnecessary packet dropping when light or false congestion appears is the disadvantage of these methods.

Fuzzy controller RED (FConRED) [36] use the difference between queue length and the target length and the change in this difference as input to a fuzzy system. The output of the FConRED is a value for the changes in Dp. Accordingly, FConRED follows an adaptive mechanism in which the value of the Dp is increased/decreased with each packet arrival rather than calculating a new value of Dp. The using of adaptive mechanism was first introduced by the BLUE method [14]. The using of adaptive Dp reduces packet loss and improves dropping in stable network statues. However, unnecessary packet dropping and loss occur when using adaptive Dp in common bursty networks.

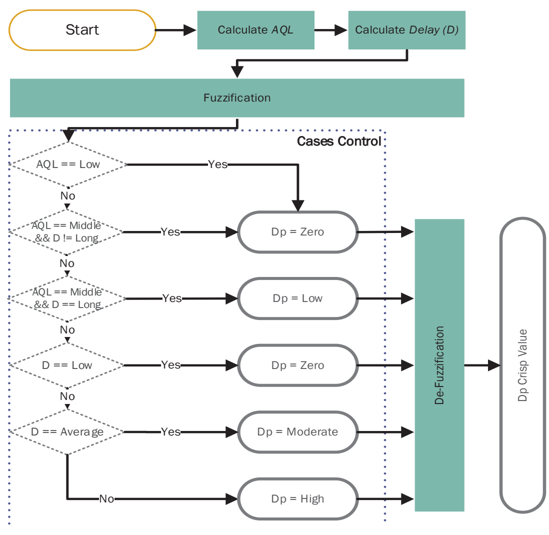
Fuzzy ERED [20] is a fuzzy version of the robust ERED proposed by Abbasov and Korukoglu in 2009 [1]. FGRED [10] and fuzzy logic-based RED (FLRED) [6] use AQL with delay. FConRED [2] use AQL with loss. Another method uses fuzzy RED with AQL, ΔQ, and delay [28]. The goal of these adaptive-based methods was to reduce packet loss. However, the slow adaptation of such adaptive technique leads to loss and unnecessary drop in bursty networks.

Figure 2

Controlled Cases in AQM Methods



(a) GRED Mechanisms



(b) FGRED Mechanisms

As different indicators are utilized, different membership functions and rules are required. Different functions and rules lead to a different output even with the same inputs, justifying the use of trial and error approach for setting up the fuzzy system components—even the indicators, which can be similar in the name or maybe different [26]. The differences are embodied in how these indicators are calculated and updated. Estimated loss and estimated delay, commonly used by current methods, differ in the definition and calculation [5].

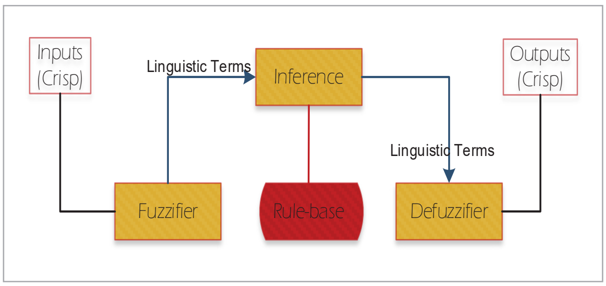
Traffic-based methods utilize a variation of traffic load indicators, such as AR and LR. However, these indicators are combined with delay and loss rather than using these variations [9]. Although, various non-fuzzy based methods, such as adaptive virtual queue (AVQ) [21], stable AVQ (SAVQ) [24], EAVQ [43], LUBA [7], SVB [13], RAQM [38], RaQ[35], PI [18] and Yellow [25], were proposed using such indicators, fuzzy-based methods that used such indicators are rare. Among these methods, fuzzy logic based AQM (Fuzzy-AQM) [30] utilizes AR and arrival factors related to the queue occupation. Traffic-based methods reduce unnecessary packet dropping by monitoring the traffic status and avoiding false congestion. However, traffic-based methods lead to packet loss and increased delay with the occurrence of sudden congestion.

Hybrid-based methods combine the queue-based and load-based attributes. Hybrid-based methods, such as REM [22] and REAQM [39], are proposed to obtain the advantages of the two approaches. However, the problem with these methods is how the indicators are combined, leadings to unnecessary packet dropping.

The fuzzy-based AQM methods are developed based on the Mamdani model [27]. The other model, Takagi–Sugeno–Kang [23], is not used because it requires training data and is characterized by its low interpretability compared with the Mamdani model. Given the problem of implementing an AQM based on explicit knowledge rather than training samples, the AQM model is best described by using the Mamdani model. The Mamdani model consists of four main components: fuzzifier, inference, defuzzifier, and rule-based, as illustrated in Figure 3.

Figure 3

Mamdani Model for Fuzzy System



Bio-inspired optimization algorithms have been used for the parameter optimization problem. Accordingly, ant colony optimization is used for tuning the proportional integral derivative (PID) method [11]. Particle swarm optimization is used to optimize the fuzzy version of the PID method [19]. Similarly, the genetic algorithm [34] and grey wolf [33] are used for tuning the AQM parameters. The advantages of these methods are not related to performance. Optimization is used to optimize the parameter setting, commonly implemented through trial and error, but, with more human labor. The trial and error produce similar or better results because optimization might be stuck in local optima.

In summary, existing fuzzy AQM methods can be characterized as follows: 1) Fuzzy-based models create more flexibility in queue management than the non-fuzzy methods and ease the problem of parameterization. 2) The existing fuzzy-based AQM methods use the Mamdani model to implement the fuzzy system because explicit knowledge can be developed. 3) The existing fuzzy-based AQM methods focused on the queue-related indicators combined with delay and loss to overcome the shortage in monitoring and evaluating the network traffic. 4) Limitations are found in loss and dropping rate, which can be referred back to the utilized indicators because the controlling mechanism is adjusted to fit with the utilized indicator. Accordingly, explicitly integrated indicators, such as those related to queue and load-based, should be considered. Thus, a new mechanism is required to use multiple congestion indicators efficiently.

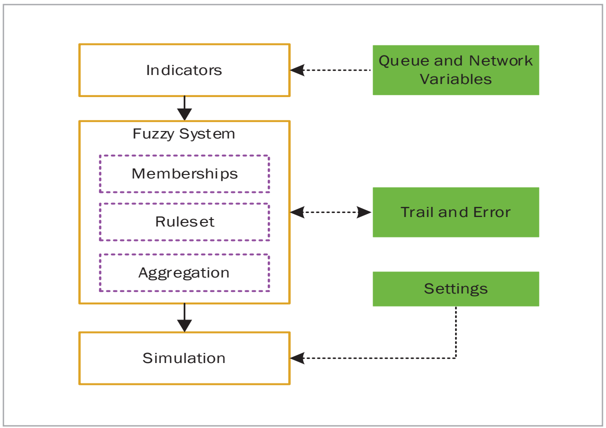
3. Proposed Work

The objectives of this paper are as follows: 1) to propose and identify integrated indicators, 2) to create a model that wraps these indicators, and 3) to utilize the fuzzy inference process to calculate Dp. These objectives are covered by a set of processing stages, which will be discussed in the following subsections.

The proposed method is built in three steps, as illustrated in Figure 4. These steps are as follows: 1) Identify the indicators by using the network parameters, which will be maintained during the execution of the proposed method. 2) Building the fuzzy system, which operates on the indicators identified in the previous step. The components of the fuzzy system are the fuzzy sets, the membership functions, the rules, and the output aggregation approach. The rules are built through trial and error approach to avoid the drawback of the optimization algorithms. 3) The simulation settings and parameters are identified concerning the related work in the domain.

Figure 4

Flowchart of the Proposed Work



3.1. Indicators

As a first step, the indicators are identified and calculated. Three criteria are identified to create integrated indicators and wrap them in a fuzzy system. This process is performed to control the process of identifying and utilizing these indicators, which are as follows:

1. The indicators shall be representative and demonstrate the queue's status and the network collectively.
2. The indicators shall be compatible to capture different aspects within identical parameters and formulations.
3. The indicators shall be standardized accordance with the covered time frame and the value range.

Accordingly, three average-based indicators are identified: queue-related, arrival-related, and departure-related. These indicators are calculated as a weighted moving average (WMA) over the current and the previously calculated value with identical weight values for all the indicators. These indicators are calculated with identical form, as defined in Equation 1.

$$WAV_t = Val_t{}^* w + WAV_{t-1}{}^* (1 - w), \tag{1}$$

where WAV_t is the weighted average value at time t, Val_t is the instance value value at time t, and w is the weight. The weight value is calculated as a portion of the buffer capacity, rather than a fixed value, as given in Equation 2. The WMA value is influenced by a time frame with a period length equal to the queue capacity.

$$w = 1 / Capacity. \tag{2}$$

All the values are normalized in the range of [0-1], which is influenced by normalizing the instance value to the same range. Accordingly, the instance queue is calculated as a portion of the capacity similar to the weight calculation in Equation 2. Given that these values are calculated with each network event (arrival or departure or both), the values of the instance arrival and departure will have the value of {0,1}. As such, the values of the indicators are calculated as given in Equation 3, Equation 4, and Equation 5.

$$AQL_t = (Q_t / Capacity) {}^* w + AQL_{t-1} (1 - w) \tag{3}$$

$$AAR_t = AR_t{}^* w + AAR_{t-1} (1 - w), \tag{4}$$

$$ADR_t = DR_t{}^* w + ADR_{t-1} (1 - w), \tag{5}$$

where AQL_t is the average queue length at time t, AAR_t is the average arrival rate at time t, ADR_t is the average departure rate at time t, Q_t is the instance queue value, AR_t is the instance arrival rate value, DR_t is the instance departure rate value, and w is the unified weight. Overall, the list of utilized indicators and their characteristics that fulfill the criteria mentioned above are given in Table 3.

Table 3

Proposed Indicators in the FCRED

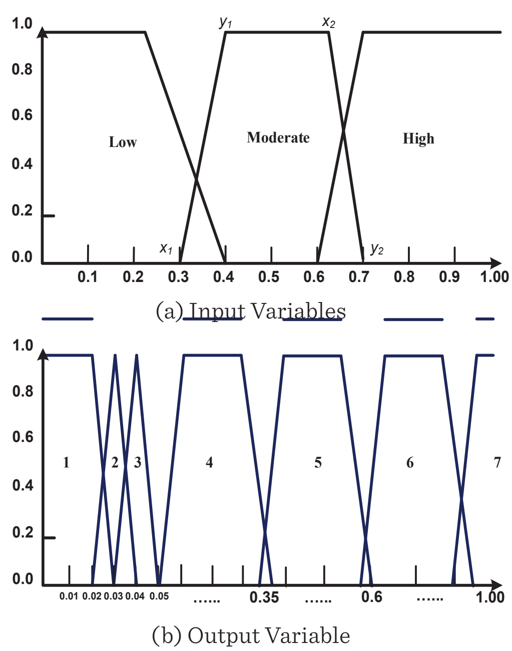
Indicator	Representative	Aspect	Parameters	Form	Time-Frame	Value-Range
AQL	√	Queue	w	WMA	= Queue Length	[0-1]
AAR	√	Traffic/ Load	w	WMA	= Queue Length	[0-1]
ADR	√	Traffic/ Load	w	WMA	= Queue Length	[0-1]

3.2. Fuzzification in the Fuzzy Model

Two components should be identified to implement the fuzzification step, the linguistic set, and the membership function. Commonly utilized linguistic sets in the existing AQM methods consist of three, four, or five terms. Given that the proposed method covers all aspects of the monitoring criteria, large variability is not required. Accordingly, the linguistic sets in the proposed FCRED are unified with three terms as follows {low, moderate, high} for the input variables. For the output variables, Dp, the set is established with more regular variability to give more flexibility to the responding action. Thus, the output set is consists of seven terms {$term_1$, ..., $term_7$}, where $term_1$ is the lowest, whereas $term_7$ is the highest in value. The linguistic set of the output variable and its membership function is not used in the fuzzification step. However, it is related to the identification of the input variables and is established in this step.

The membership function determines the boundaries of each term in each set. In FCRED, the membership function is set to equal ranges for the input variables following the equal space approach [23].

Figure 5
Fuzzy Membership Functions



For the output variable, the first three terms are set to occupy 0.05 of the space and the rest occupy the rest of the space, to avoid unnecessary dropping. Thus, for the input variables, as illustrated in Figure 5, the boundaries of the functions are defined as: low (0, 0, 0.3, 0.4), moderate (0.3, 0.4, 0.6, 0.7) and high (0.6, 0.7, 1.0, 1.0). The boundaries of output variable are defined as: $term_1$ (0, 0, 0.02, 0.03), $term_2$ (0.02, 0.03, 0.03, 0.04), $term_3$ (0.03, 0.04, 0.04, 0.05), $term_4$ (0.05, 0.1, 0.3, 0.35), $term_5$ (0.3, 0.35, 0.55, 0.6), $term_6$ (0.55, 0.6, 0.8, 0.85), and $term_7$ (0.8, 0.85, 1.0, 1.0).

Given the linguistic sets and the membership function, the fuzzification transfers each crisp input into terms with confidence values. The boundaries of each term are identified with the points (x_1, y_1, x_2, y_2) (Figure 5). The fuzzification function resulted in all the terms with membership degrees greater than zero on the basis of Equation 6.

$$\mu(x) = \begin{cases} 0 & , x < x_1, x > y_2 \\ x - x_1 / y_1 - x_1 & , x_1 \le x \le y_1 \\ 1 & , y_1 \le x \le x_2 \\ y_2 - x / b_2 - y_1 & , x_2 \le x \le y_2 \end{cases}. \quad (6)$$

3.3. Rule Evaluation in the Fuzzy Model

In the rule evaluation, the inputs are the linguistic terms of the input variables with confidence degree as calculated in Equation 6. The output is a linguistic term(s) from the output set with a confidence value. Thus, the first step is to create the rules by which the input and output are combined. The set of rules is created in IF-THEN form. Twenty-seven different possible rules are found, with each input sets consisting of three terms (i.e., 3*3*3). The list of rules is formed in a table (Table 4).

The AND operation is used to produce the confidence value of the output term. As such, the confidence value of the output term of each rule is the minimum confidence value of the input terms, as calculated in Equation 7.

$$\mu^F(m) = \max(\mu^A(x), \mu^B(y), \mu^C(z)), \quad (7)$$

where $\mu^A(x)$ is the confidence value for the term associated with the first input variable, $\mu^B(y)$ is the confidence value for the term associated with the second input variable, $\mu^{C}(z)$ is the confidence value for the term associated with the third input variable, and $\mu^{F}(m)$ is the confidence value for the term associated with the output variable.

Table 4
Proposed Rule-based in the FCRED

AQL	Low			Moderate			High		
AAR / ADR	Low	Mod.	High	Low	Mod.	High	Low	Mod.	High
High	1	1	1	2	3	3	5	5	6
Mod.	1	1	1	2	3	3	5	6	7
Low	1	1	2	3	3	4	6	6	7

3.4. Aggregation in the Fuzzy Model

The output of rule-evaluation may include one or more redundant terms. A term presented as an output of different rules may be associated with a different confidence value. The redundancy is cleared by maintaining a single-term presentation with an aggregated confidence value. The aggregated value is produced by using the AND operation between the confidence values of the underlying term, which produce the minimum confidence value among all, as calculated in Equation 8.

$$\mu^{FA}(t) = \max(\mu^{F1}(t), \mu^{F2}(t)), \tag{8}$$

where $\mu^{F1}(t)$ and $\mu^{F2}(t)$are the first and second confidence values for the redundant term t, respectively, and $\mu^{FA}(t)$ is the produced confidence value for the redundant term t.

3.5. Defuzzification in the Fuzzy Model

The **defuzzification** step produces a crisp value by converting the terms and their confidence using defuzzification calculation. The center of gravity (COG) method is used with the input terms because it has similar capabilities to crisp averaging values [29]. COG is given in Equation 9.

$$COG = \frac{\sum_{x=a}^{b} \mu^{A}(x)x}{\sum_{x=a}^{b} \mu^{A}(x)} \ . \tag{9}$$

4. Simulation and Measurements

The simulation process is conducted by using the JAVA programming language and a single router. The simulation uses an edge and single buffered router with small buffer size to evaluate the proposed method under critical circumstances. The buffer is simulated as First-In-First-Out (FIFO) queuing model [17, 20]. The network is monitored by using a discrete-time queue similar to the previous work [10, 37, 44]. Compared with the continuous-time model, the discrete model accurately calculates and evaluates the performance by analyzing the network statuses and the AQM responses. In such a model, the running time is divided into an equal period, called a slot. The slot is characterized by having a packet arrival or departure or both of them. The running simulation consists of a 2 million time slot and the first 40% of which is used for the warm-up period to reach a steady-state, in which no performance is calculated.

Table 5
Parameter Settings

Par.	Discussion	Utilized Value(s)
α	Indicate the probability of packet arrival at each time slot. No packet arrival occurs at value 0, and the value 1 indicates certain arrival at each time slot. The value range (0-1) indicates a different probability for packet arrival. The higher the value, the more packet arrival occurs.	[0.30-0.95]
β	Indicate the probability of packet departure at each time slot. No packet departe occurs at value 0, and the value 1 indicates a certain packet departure at each time slot. The value range (0-1) indicates a different probability for packet departure. The higher the value, the more packet departure occurs.	0.5
weights	The weighted parameters of the compared methods, as utilized in literature [10, 37, 44]. These parameters are used for RED, ERED, and BLUE.	Dmax =0.1, MinThr = 3, MaxThr= 9

The arrival (α) and departure (β) rates are controlled with probabilities, which are changed in accordance with the required circumstances. As such, if the departure (β) rate is set to be 0.5 and the arrival rate (α) is set to any value in the range [0.5-1.0], then congestion circumstances will be enforced because the arrival (α) is higher than the departure (β) and vice versa. Accordingly, the arrival rate is set to values in the range of [0.30-0.95], and the departure is set to 0.5.

Table 5 list the parameters and the values utilized in the simulation. The evaluation of the proposed method is implemented on the basis of a set of performance measures, loss, dropping, delay, and throughput in packets per slot (PPS) manner, as summarized in Table 6.

Table 6

Performance Measure

Measure		Calculation	Comments
Packet Loss	PL	$PL = \frac{\#lost}{\#arrived}$	The number of lost packets due to queue overloading is normalized by dividing the lost number with the number of arrived packets
Packet Dropping	PD	$PD = \frac{\#dropped}{\#arrived}$	The number of dropped packets by the AQM response is normalized by dividing the dropped number with the number of arrived packets
Delay	D	$D = \sum \frac{\#queuedSlots}{\#arrived}$	The total number of slots that packets remained in the queue normalized by the number of arrived packets
Throughput	Th_{pps}	$Th_{pps} = \sum \frac{\#delivered}{\#arrived}$	The number of delivered packets (other than lost and dropped) normalized by the number of arrived packets

5. Results

The proposed method is compared with the core AQM methods, which are RED [16], ERED [1] and BLUE [14]. RED [16] is the first AQM method, which has been used as the core for all the subsequent AQM methods. ERED [1] improved the performance of RED in term of packet dropping. BLUE [14] is the first and the core for the adaptive approach for Dp calculation. Accordingly, these methods are used as the baseline for AQM comparison in the literature [20]. The proposed method is compared with related fuzzy-based AQM methods, which are FRED [36], FERED [20], FGRED [10], FBLUE [42] and FLRED [6], which have reported improvements over the core methods.

The results of the proposed and compared methods in terms of packet loss are illustrated in Figure 6, with varying arrival rates (α) and a departure rate (β) equal to 0.5. The results are provided at different arrival rates to distinguish between low and high traffic, resulting in different scenarios ranging from non-congestion to heavy congestion. The proposed method and BLUE lose no packets, which outperform RED (0.01 loss) and ERED (0.21 loss) in congested and heavily congested statuses. In non-congested status, all compared methods perform equally (zero loss). In a light congestion state, with α equal to 0.5 and β equal to 0.5, the RED starts to lose packets (0.02 loss), and ERED starts losing packets at α equal to 0.65 and β equal to 0.5 (0.01 loss). By contrast, FCRED and BLUE lost no packets at different α values.

Figure 7 illustrates the packet dropping of the proposed and compared methods with varying arrival rates and a departure rate (β) equal to 0.5. The results are provided at different arrival rates to distinguish between low, high, and extremely high traffics and non-congestion to heavy congestion statuses. The proposed method (with average dropping equal to 0.21) outperforms the BLUE (with average dropping equal to 0.28), which has comparable performance to the proposed method in terms of loss. RED performs equally with the FCRED in dropping, and ERED has a better dropping rate (with average dropping equal to 0.15), resulting in a massive loss for ERED, as illustrated in Figure 6.

Figure 8 illustrates the delay of the proposed and compared methods with varying arrival rates and a departure rate (β) equal to 0.5. BLUE outperforms the compared methods in congested and heavy congested statuses in terms of delay with an average of 7.18 compared with 14.76, 16.74, and 21.96 for RED, FCRED, and ERED, respectively. In non-congested status, all compared methods perform equally with slight variation.

Figure 6

Packet Loss –based Comparison at β=0.5

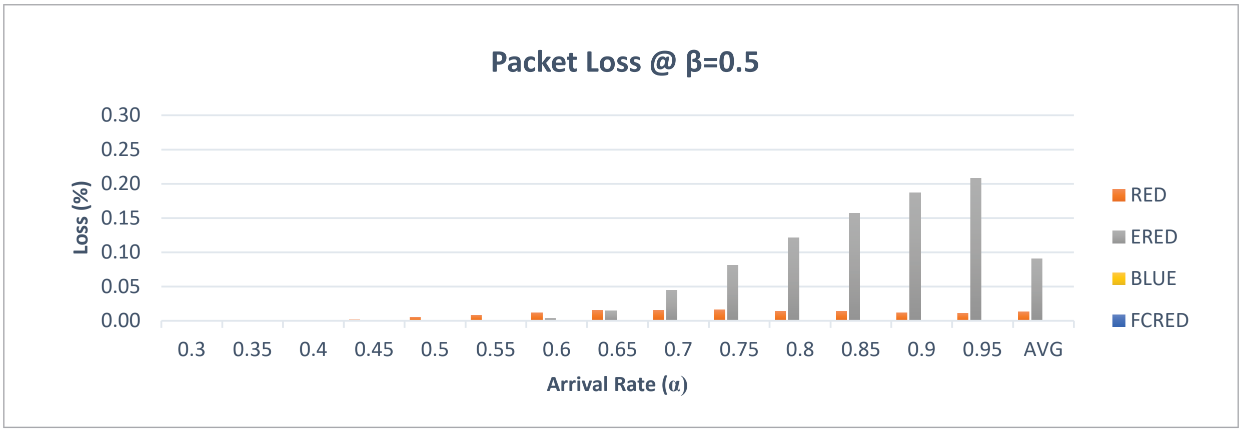


Figure 7

Packet Dropping–based Comparison at β=0.5

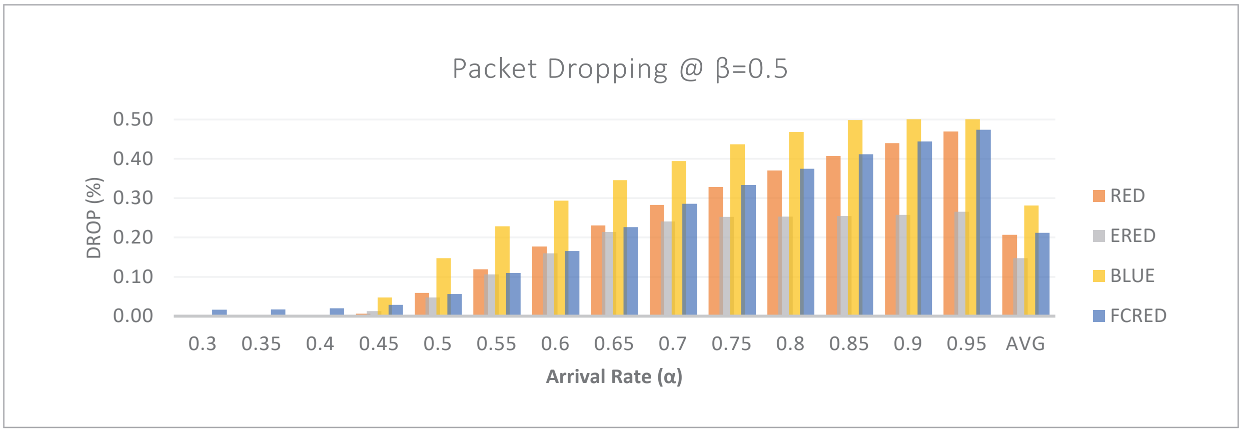


Figure 8

Delay–based Comparison at β=0.5

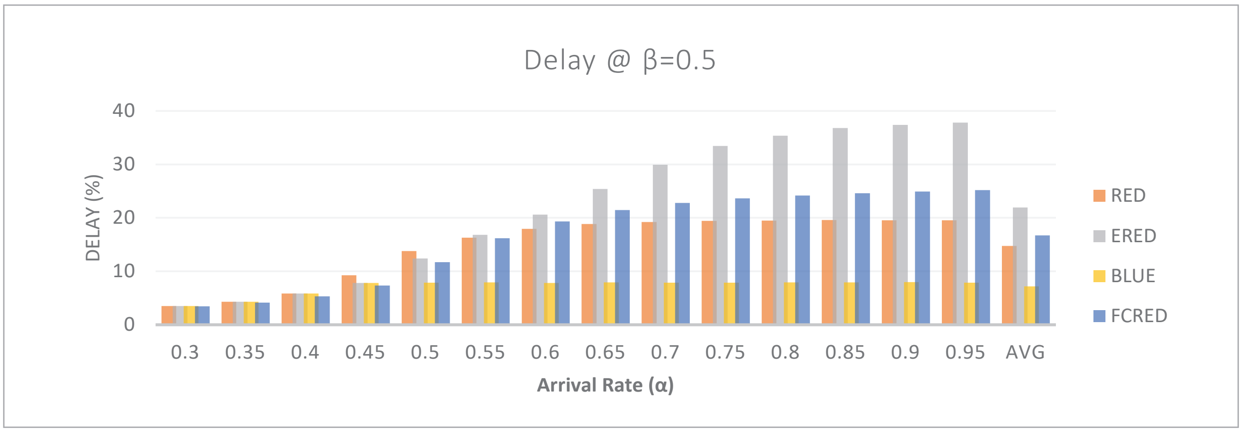


Figure 9

Throughput-based Comparison at β=0.5

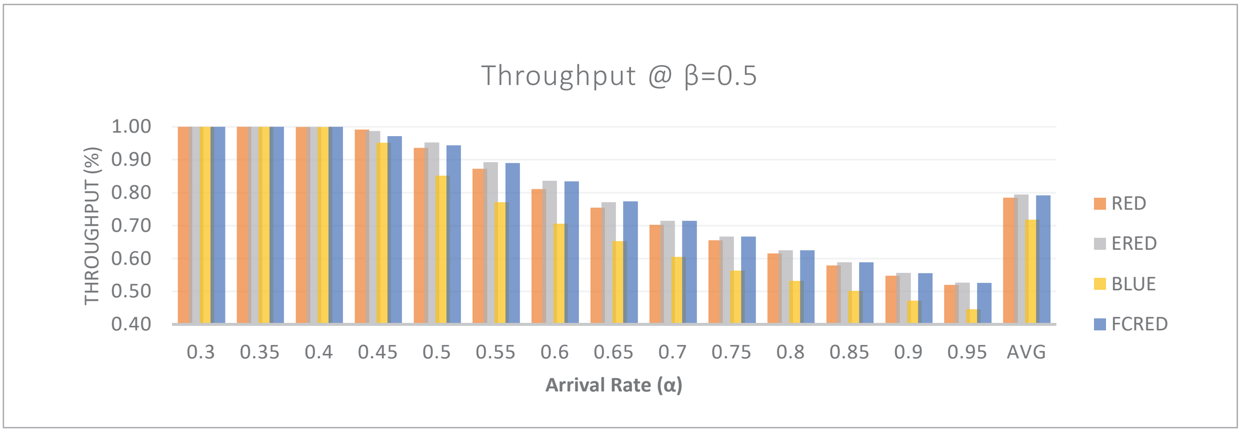


Figure 10

Packet Loss-based Comparison at β=0.3

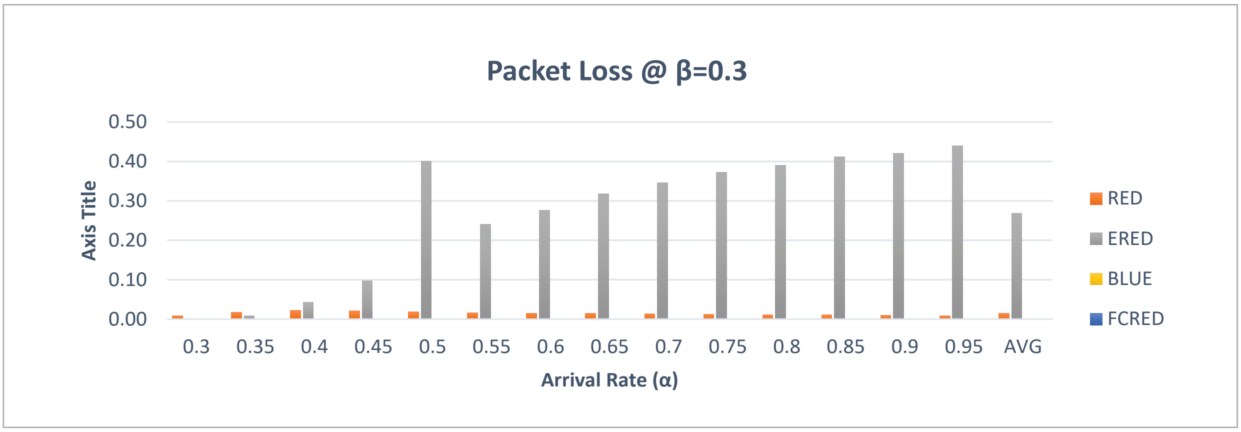


Figure 11

Packet Dropping-based Comparison at β=0.3

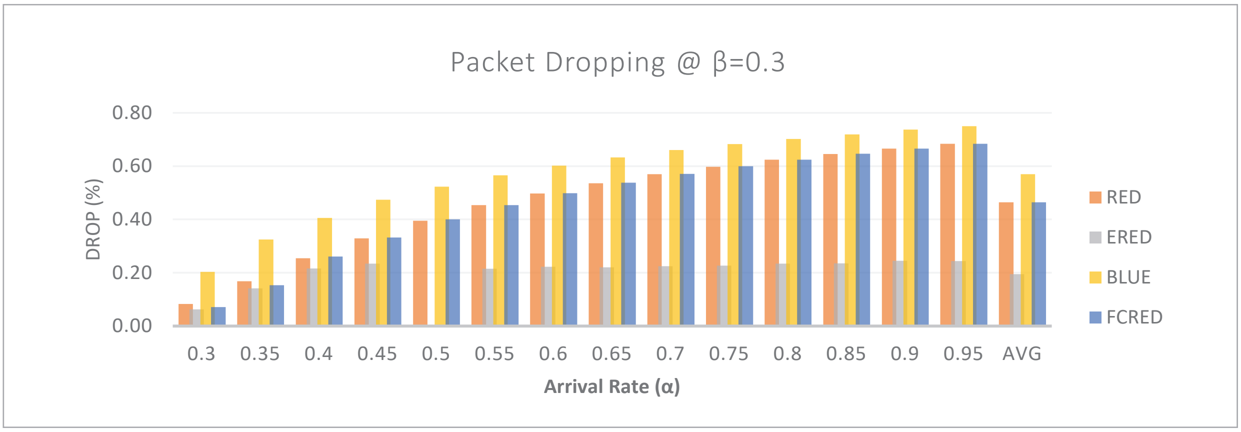


Figure 12

Delay–based Comparison at β=0.3

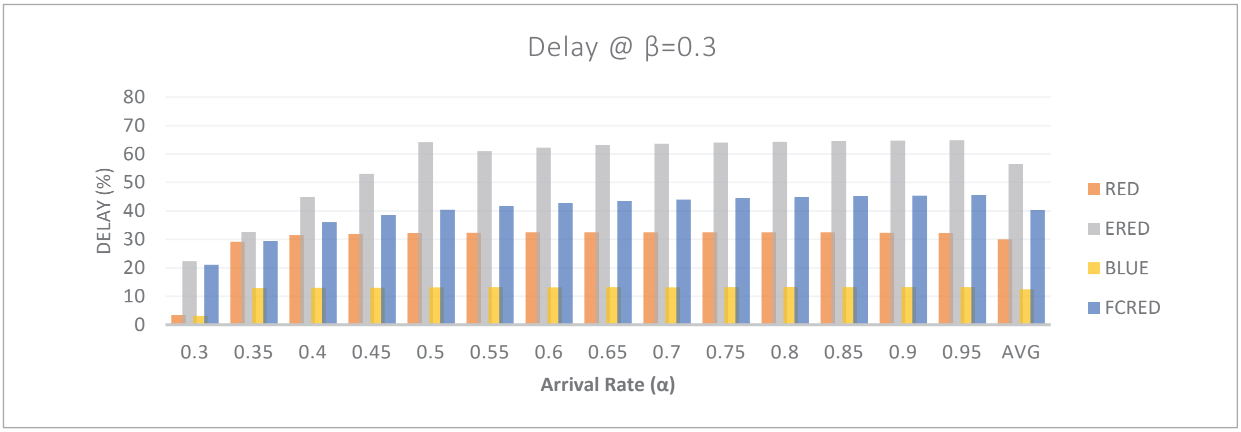


Figure 13

Throughput–based Comparison at β=0.3

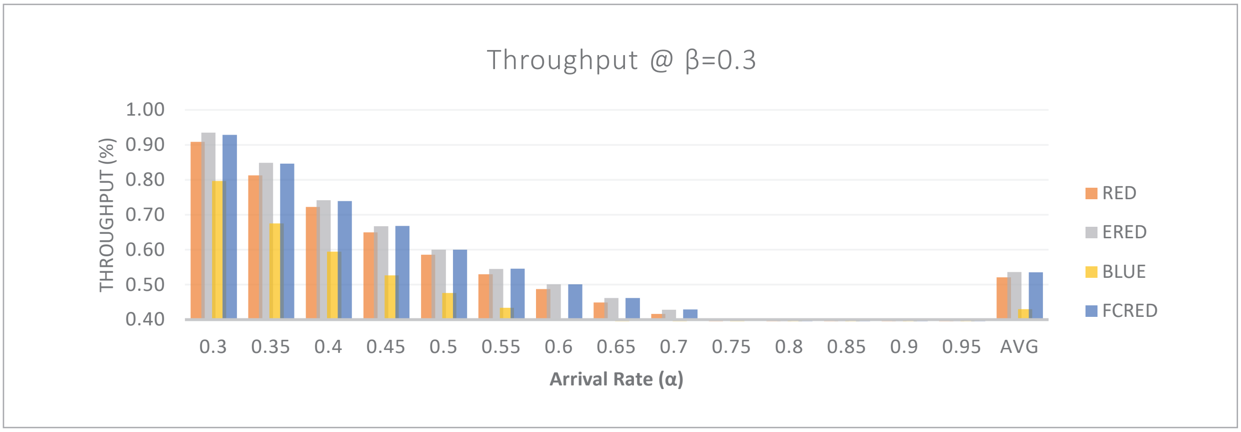


Figure 9 illustrates the throughput of the proposed and compared methods with varying arrival rates and a departure rate (β) equal to 0.5. The results show a throughput of 0.79, 0.79, 0.78 and 0.72, for FCRED, ERED, RED, and BLUE, respectively. The proposed method outperforms the BLUE and RED, ERED performs equally with the FCRED in terms of throughput, and FCRED is better than ERED in terms of delay and loss, as illustrated in Figure 6, Figure 7, and Figure 8. The proposed method outperforms the compared methods in congested and heavy congested statuses in terms of throughput. In non-congested status, all compared methods perform equally with slight variation. In such a scenario, the RED and BLUE result in less throughput than the proposed method, and FCRED results in better delay and loss than ERED.

The second set of results is obtained with varying arrival rate and a departure rate equal to 0.3. Compared with the first set, congestion is expected with a lower arrival rate with the decrease in the departure rate. The results of the proposed and compared methods in terms of packet loss are illustrated in Figure 10, separated by vertical lines to separate between low, high, and extremely high traffic. Figure 11, Figure 12, and Figure 13 illustrate the dropping, delay, and throughput, respectively. The results of the second set of experiments confirm the findings of the first set.

A comparison of the proposed FCRED with recent fuzzy-based methods is given in Table 7. FCRED and

FERED outperform the other compared methods. However, FCRED drops less packets compared with FERED and improves the network performance. A time comparison is given in Table 8. The proposed method consumes more time compared with RED, ERED, and BLUE, which are crisp-based. The proposed method outperforms the other fuzzy-based methods, except for FBLUE. Although, FBLUE consumes more time, FCRED produces better results, as given in Table 7.

Table 7

Results of the proposed FCRED method with Fuzzy-methods

α	Measure	FRED	FERED	FGRED	FBLUE	FLRED	FCRED
0.3	**Loss**	**0**	**0**	**0**	0	**0**	**0**
	Drop	**0**	**0**	**0**	0.04	**0**	0.02
	Drop & Loss	**0**	**0**	**0**	**0**	**0**	0.02
	Delay	3.90	3.92	3.90	3.5	3.90	**3.43**
0.5	**Loss**	0.03	**0**	0.03	0.03	0.03	**0**
	Drop	**0.06**	0.06	0.04	**0.03**	0.04	0.06
	Drop & Loss	**0.09**	**0.06**	0.07	**0.06**	0.07	**0.06**
	Delay	24.24	21.66	23.7	16.5	21.32	**11.73**
0.9	**Loss**	0.12	**0**	0.24	0.42	0.18	**0**
	Drop	0.44	0.53	0.20	0.02	**0.26**	0.44
	Drop & Loss	0.56	0.53	**0.44**	**0.44**	**0.44**	**0.44**
	Delay	28.32	28.25	39.7	39.0	28.23	**24.39**

Table 8

Time comparison of the proposed FCRED method and the existing methods

Method	Time (In Millisecond)
RED	**121**
ERED	**92**
BLUE	**93.75**
FRED	**7658**
FERED	**7652**
FGRED	**7560**
FBLUE	**5520**
FLRED	**7476.25**
FCRED	**6501.23**

6. Conclusion

This paper proposes a fuzzy-based AQM method based on network analysis, inferencing, and simple and comprehensive indicators. The contributions of FCRED can be summarized as follows: 1) identify a set of criteria for the AQM indicators and use them for collecting the comprehensive indicators. 2) Developing a fuzzy-based model to use these indicators for actively managing the queue at the router buffer. Accordingly, FCRED outperforms the existing methods in terms of packet loss, dropping, delay, and throughput. Future work will use other indicators that fit the set of criteria as determined in this paper. As given in the results, FCRED reduces loss to zero, and dropping is reduced from 0.28 to 0.21, a 25% reduction compared with BLUE, which achieves the steadiest results among the compared methods. Compared with recent fuzzy-based methods, the proposed FCRED and FERED outperform the other compared methods. FCRED drops less packets compared with FERED and improves the network performance.

Acknowledgment

This research was supported by Al-Ahliyya Amman University/Jordan, and Arab Open University (AOU)/KSA.

References

1. Abbasov, B., Korukoglu, S. Effective Red: An Algorithm to Improve Red's Performance by Reducing Packet Loss Rate. Journal of Network and Computer Applications, 2009, 32(3), 703-709. https://doi.org/10.1016/j.jnca.2008.07.001
2. Abdel-Jaber, H., Mahafzah, M., Thabtah, F., Woodward, M. Fuzzy Logic Controller of Random Early Detection based on Average Queue Length and Packet Loss Rate. Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Edinburgh, UK, June 16-18, 2008, 428-432.
3. Abu-Shareha, A. A. Controlling Delay at the Router Buffer using Modified Random Early Detection. International Journal of Computer Networks & Communications (IJCNC), 2019, 11(6), 63-75. https://doi.org/10.5121/ijcnc.2019.11604
4. Abu-Shareha, A. A. Enhanced Random Early Detection Using Responsive Congestion Indicators. International Journal of Advanced Computer Science and Applications (IJACSA), 2019, 10(3), 358-367. https://doi.org/10.14569/IJACSA.2019.0100347
5. Abu-Shareha, A. A. Integrated Random Early Detection for Congestion Control at the Router Buffer. Computer Systems Science and Engineering, 2022, 40(2), 719-734. https://doi.org/10.32604/csse.2022.018369
6. Abualhaj, M. M., Abu-Shareha, A. A., Al-Tahrawi, M. M. Flred: An Efficient Fuzzy Logic Based Network Congestion Control Method. Neural Computing and Applications, 2018, 30(3), 925-935. https://doi.org/10.1007/s00521-016-2730-9
7. Agarwal, M., Gupta, R., Kargaonkar, V. Link Utilization based AQM and Its Performance. Proceedings of IEEE Global Telecommunications Conference, 2004, 713-718.
8. Alcalá, R., Cordón, O., Casillas, J., Herrera, F., Zwir, S. Learning and Tuning Fuzzy Rule-Based Systems for Linguistic Modeling. Knowledge-Based Systems, Elsevier, 2000, 889-941. https://doi.org/10.1016/B978-012443875-0/50030-6
9. Alshinwan, M., Abualigah, L., Kim, C.-S., Alabool, H. Development of a Real-Time Dynamic Weighting Method in Routing for Congestion Control: Application and Analysis. Wireless Personal Communications, 2021, 118(1), 755-772. https://doi.org/10.1007/s11277-020-08041-3
10. Baklizi, M., Abdel-Jaber, H., Abu-Shareha, A. A., Abualhaj, M. M., Ramadass, S. Fuzzy Logic Controller of Gentle Random Early Detection Based on Average Queue Length and Delay Rate. International Journal of Fuzzy Systems, 2014, 16(1), 9-19.
11. Chebli, S., Elakkary, A., Sefiani, N. Ant Colony Optimization Based on Pareto Optimality: Application to a Congested Router Controlled by Pid Regulation. Systems Science & Control Engineering, 2018, 6(1), 360-369. https://doi.org/10.1080/21642583.2018.1509395
12. Chrysostomou, C., Pitsillides, A., Hadjipollas, G., Sekercioglu, A., Polycarpou, M. Fuzzy Explicit Marking for Congestion Control in Differentiated Services Networks. Proceedings of Eighth IEEE International Symposium on Computers and Communication, 2003, 312-319.
13. Deng, X., Yi, S., Kesidis, G., Das, C. R. Stabilized Virtual Buffer (Svb)-an Active Queue Management Scheme for Internet Quality-of-Service. Proceedings of IEEE Global Telecommunications Conference, 2002, 2(1), 1628-1632.
14. Feng, W.-c., Kandlur, D. D., Saha, D., Shin, K. G. Blue: A New Class of Active Queue Management Algorithms. 1999, University of Michigan, Ann Arbor, MI, Technical Report.
15. Floyd, S. Recommendations on Using the Gentle Variant of Red. 2000. http://www.aciri.org/floyd/red/gentle.html.
16. Floyd, S., Jacobson, V. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, 1993, 1(4), 397-413. https://doi.org/10.1109/90.251892
17. Guizani, M., Rayes, A., Khan, B., Al-Fuqaha, A. Network Modeling and Simulation: A Practical Per-

spective. John Wiley & Sons, 2010. https://doi.org/10.1002/9780470515211

18. Hollot, C. V., Misra, V., Towsley, D., Gong, W. B. On Designing Improved Controllers for Aqm Routers Supporting Tcp Flows. Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, 2001, 3(1), 1726-1734.
19. Karam, Z. A. Hybrid Fuzzy Congestion Controllers for Computer Networks Tuned by Modified Particle Swarm Optimization. International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, 2018, 7(2), 17-26. https://doi.org/10.11601/ijates.v7i2.250
20. Khatari, M., Samara, G. Congestion Control Approach Based on Effective Random Early Detection and Fuzzy Logic. arXiv preprint arXiv:1712.04247, 2017.
21. Kunniyur, S., Srikant, R. End-to-End Congestion Control Schemes: Utility Functions, Random Losses and Ecn Marks. IEEE/ACM Transactions on Networking, 2003, 11(5), 689-702. https://doi.org/10.1109/TNET.2003.818183
22. Lapsley, D., Low, S. Random Early Marking: An Optimisation Approach to Internet Congestion Control. Proceedings of IEEE International Conference on Networks (ICON '99), 28 Sept.-1 Oct. 1999, 1999, 67-74. 10.1109/icon.1999.796161
23. Lavesson, N., Davidsson, P. Evaluating Learning Algorithms and Classifiers. International Journal of Intelligent Information and Database Systems, 2007, 1(1), 37-52. https://doi.org/10.1504/IJIIDS.2007.013284
24. Long, C.-N., Zhao, B., Guan, X.-P. Savq: Stabilized Adaptive Virtual Queue Management Algorithm. IEEE Communications Letters, 2005, 9(1), 78-80. https://doi.org/10.1109/LCOMM.2005.1375248
25. Long, C., Zhao, B., Guan, X., Yang, J. The Yellow Active Queue Management Algorithm. Computer Networks, 2005, 47(4), 525-550. https://doi.org/10.1016/j.comnet.2004.09.006
26. Ma, H., Xu, D., Dai, Y., Dong, Q. An Intelligent Scheme for Congestion Control: When Active Queue Management Meets Deep Reinforcement Learning. Computer Networks, 2021, 200(1). https://doi.org/10.1016/j.comnet.2021.108515
27. Mamdani, E. H. Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. Proceedings of the Institution of Electrical Engineers, 1974, 121(12), 1585-1588. https://doi.org/10.1049/piee.1974.0328
28. Nassar, K. A., Abdullah, A. A. Fuzzy Red to Reduce Packet Loss in Computer Network. Journal of AL-Qadisiyah for Computer Science and Mathematics, 2016, 8(1), 107-114.
29. Negnevitsky, M. Artificial Intelligence: A Guide to Intelligent Systems. Second ed., England, 2005.
30. Nyirenda, C. N., Dawoud, D. S. Multi-Objective Particle Swarm Optimization for Fuzzy Logic Based Active Queue Management. Proceedings of International Conference on Fuzzy Systems, 2006, 2231-2238. https://doi.org/10.1109/FUZZY.2006.1682010
31. Ott, T. J., Lakshman, T. V., Wong, L. Sred: Stabilized Red. Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, March 21-25, 1999, 3(1), 1346-1355. 10.1109/infcom.1999.752153
32. Rossides, L., Sekercioglu, A., Pitsillides, A., Vasilakos, A., Kohler, S., Tran-Gia, P. Fuzzy Red: Congestion Control for Tcp/Ip Diff-Serv. Advances in Computational Intelligence and Learning, Springer, 2002, 343-352. https://doi.org/10.1007/978-94-010-0324-7_24
33. Sabry, S. S., Kaittan, N. M. Grey Wolf Optimizer Based Fuzzy-Pi Active Queue Management Design for Network Congestion Avoidance. Indonesian Journal of Electrical Engineering and Computer Science, 2020, 18(1), 199-208. https://doi.org/10.11591/ijeecs.v18.i1.pp199-208
34. Sulttan, M. Q., Jaber, M. H., Shneen, S. W. Proportional-Integral Genetic Algorithm Controller for Stability of Tcp Network. International Journal of Electrical and Computer Engineering (IJECE), 2020, 10(6), 6225-6232. https://doi.org/10.11591/ijece.v10i6.pp6225-6232
35. Sun, J., Zukerman, M. Raq: A Robust Active Queue Management Scheme Based on Rate and Queue Length. Computer Communications, 2007, 30(8), 1731-1741. https://doi.org/10.1016/j.comcom.2007.02.005
36. Sun, J., Zukerman, M., Palaniswami, M. Stabilizing Red Using a Fuzzy Controller. Proceedings of International Conference on Communications (ICC'07), 2007, 266-271. https://doi.org/10.1109/ICC.2007.52
37. Tsavlidis, L., Efraimidis, P., Koutsiamanis, R.-A. Prince: An Effective Router Mechanism for Networks with Selfish Flows. Journal of Internet Engineering, 2016, 6(1).
38. Wang, C., Li, B., Hou, Y. T., Sohraby, K., Long, K. A Stable Rate-Based Algorithm for Active Queue Management. Computer Communications, 2005, 28(15), 1731-1740. https://doi.org/10.1016/j.comcom.2004.12.017
39. Wang, J., Song, M., Yang, H. Rate-Based Active Queue Management for Congestion Control over Wired and Wireless Links. Proceedings of First International Conference on Communications and Networking in China, October 25-27, 2006, 1-6. https://doi.org/10.1109/CHINACOM.2006.344692

40. Xu, Y., Ni, H., Zhu, X. An Effective Transmission Scheme Based on Early Congestion Detection for Information-Centric Network. Electronics, 2021, 10(18), 2205. https://doi.org/10.3390/electronics10182205

41. Yaghmaee, M. H., AminToosi, H. A Fuzzy Based Active Queue Management Algorithm. SIMULATION SERIES, 2003, 35(4), 458-464.

42. Yaghmaee, M. H., Menhaj, M., Amintoosi, H. A Fuzzy Extension to the Blue Active Queue Management Algorithm. IAEEE, Journal of Iranian Association of Electrical and Electronics Engineers, 2005, 1(3), 3-14.

43. Yanping, Q., Xiangze, L., Qi, L., Wei, J. A Stable Enhanced Adaptive Virtual Queue Management Algorithm for Tcp Networks. in Control and Automation, 2007. ICCA 2007. IEEE International Conference on. 2007. IEEE.

44. Yu-hong, Z., Xue-feng, Z., Xu-yan, T. Research on the Improved Way of Red Algorithm S-Red. International Journal of u-and e-Service, Science and Technology, 2016, 9(2), 375-384. https://doi.org/10.14257/ijunesst.2016.9.2.36

45. Zargar, S. T., Yaghmaee, M. H., Fard, A. M. Fuzzy Proactive Queue Management Technique. Proceedings of Annual IEEE India Conference, 2006, 1-6. https://doi.org/10.1109/INDCON.2006.302841