Nigerian Journal of Physics (NJP)
ISSN online: 3027-0936
ISSN print: 1595-0611
DOI: https://doi.org/10.62292/njp.v35i1.2026.457
Volume 35(1), March 2026

A Refined AQM Approach for Providing an Improved Performance in Communication Networks

***[1]Samuel O. Hassan, [2]Fatimah Adamu-Fika, [3]Gloria Y. Amadi, [4]Vivian O. Nwaocha, [5]Adekunle A. Adeyelu, [1]Olalere A. Abass and [1]Olakunle O. Solanke**

[1]Department of Computer Sciences, Olabisi Onabanjo University, Ago-Iwoye, Ogun State, Nigeria
[2]Cyber Security Department, Air Force Institute of Technology, Kaduna, Nigeria
[3]School of Foundation Studies, Rivers State College of Health Sciences & Management Technology, Port Harcourt, Nigeria.
[4]Department of Information Systems & Technology, National Open University, Jabi, Abuja, Nigeria.
[5]Rev. Fr. Moses Orshio Adasu University, Makurdi, Nigeria.

*Corresponding Author's Email: samuel.hassan@oouagoiwoye.edu.ng

Keywords:
Active queue management,
Dropping probability,
Performance evaluation,
RED algorithm,
Network congestion.

ABSTRACT
Effective queue management for making certain of obtaining an improved network quality of service (QoS) is becoming a crucial challenge to the Internet owing to enormous increase in real-time communications. The widely reputed Random Early Detection (RED) and its offshoot amendment (Three-section RED, TRED) Active Queue Management (AQM) algorithms are considered inefficient for handling high congested network settings, leading to large end-to-end delay. In this paper, we propose a more sophisticated RED-based AQM design that leveraged TRED. We call this new scheme Ref-RED (an abbreviation of 'Refined Random Early Detection'). Experimental investigations in ns-3 simulator demonstrated that Ref-RED's packet dropping strategy is more belligerent as it actively drops packets at heavy-load congested network, leading to the appreciable performance improvement.

INTRODUCTION

It is beneficial ab initio to describe congestion. Typically, congestion has been portrayed as a critical phenomenon in the parlance of computer networks, which occurs when the available buffer capacity of the routing infrastructure is unable to support and accommodate the overall arriving data packets. This issue causes performance deterioration of the network quality of service, through reduced throughput, increased packet loss rate, and unfair sharing of resources among competing flows (Abdel-Jaber, 2025; Giacomoni *et al*, 2023; Kaur and Singhai, 2019; Abu-Shareha, 2019; Adamu *et al*., 2021; Thiruchelvi and Raja, 2008; Abdel-Jaber, 2015; Ryu *et al*., 2003; Adamu *et al*., 2020; Pei *et al*., 2021).

Adamu et al. (2020) categorically stated that purposeful reduction or avoidance of congestion is a fundamental factor for ensuring a more desirable network performance. Two major approaches are usually employed for managing network congestion. Typically, one is the transmission control protocol (TCP) end-to-end congestion control approach, while the second is the network device-enabled congestion control approach (Ali et al., 2024). This later approach can be classified into two forms: passive queue management (PQM) and active queue management (AQM) (Zhu et al., 2023; Jain et al., 2018; Pan et al., 2022; Gim´enez et al., 2020).

PQM often employs the FIFO (First-In, First-Out) queue policy as it only drops packets indiscriminately when queue overflow happens, resulting in a vast degraded QoS. PQM incurs shortcomings such as (Braden *et al*., 1998; Long *et al*., 2005): i) lock-out phenomenon, ii) global synchronization, iii) full queue (or buffer overflow), and iv) large delay. On the contrary, AQM policies are considered a far more effective approach as it attempt to manage congestion by purposefully and proactively dropping packets at routing nodes early prior to queue overflow as TCP packet senders gets notified of incipient congestion and in turn reduce their rates, bypassing the shortcomings of PQM (Zhu *et al*., 2023; Jain *et al*., 2018; Chydzinski and Adamczyk, 2023; Chen *et al*.,2004; Imputato *et al*., 2020).

AQM which has since become an active research area with explosive progression has triggered an extensive attention from scientists in both academia and industry, resulting in many recent proposed improved solutions. Among them, the Random Early Detection (RED) algorithm (Floyd and Jacobson,1993) (particularly its components) continues to stand as the most thoroughly studied and evaluated (Zhou *et al*., 2006).

Although RED can be singled (out) as the foremost AQM algorithm, the researchers in (Jain *et al*., 2023) strongly advocates the necessity for revisiting and leveraging upon such classical algorithm and indeed some other old-established/past designs for at least five reasons:

i. To serve as a primer model for upcoming researchers interested in the AQM research area;
ii. To minify the complexity in relations to their deployment;
iii. To enhance their robustness in relations to the demands of the current Internet traffic;
iv. To acquire initial insightful knowledge about the operations of forthcoming AQM algorithms, and
v. To design new algorithms by utilizing their merits.

The principal features of RED (Floyd and Jacobson, 1993) can be summarized as follows. At every packet arrival, the congestion level is first appraised by valuating the average queue size q_{Avg}, using EWMA (an abbreviation for 'exponential weighted moving average') which is a first-order low-pass filter as represented in Eq. (1).

$$q_{Avg} = (1 - q_w) \times q'_{Avg} + (q_w \times q) \quad (1)$$

where

q'_{Avg} designates the preexisted q_{Avg},

q expresses the current queue size,

q_w conveys a predefined weighting factor (in the interval [0, 1])

RED then takes a step further to determine the drop probability represented as

$$P_a = P_b \left[\frac{1}{1-(count \times P_b)}\right] \quad (2)$$

where

count expresses the number of undropped packets since the preexisted dropped packets,

P_b connote the intermediate drop probability,

P_a is the final drop probability,

The value for P_b in Eq. (2) can be obtained as follows:

$$P_b = \begin{cases} 0, q_{Avg} < Thld_{min} \\ P_{max}\left(\frac{q_{Avg}-Thld_{min}}{Thld_{max}-Thld_{min}}\right), Thld_{min} \le q_{Avg} < Thld_{max} \\ 1, q_{Avg} \ge Thld_{max} \end{cases} \quad (3)$$

Where

$Thld_{min}$ indicates the minimum queue threshold;

$Thld_{max}$ signifies the maximum queue threshold; and

P_{max} is expresses the maximum drop probability.

Equation (3) tells us that,

i. When $q_{Avg} < Thld_{min}$, then packet discarding will not occur.
ii. When $Thld_{min} \le q_{Avg} < Thld_{max}$, then a packet drop probability P_b is computed as follows

$$P_b = P_{max}\left(\frac{q_{Avg}-Thld_{min}}{Thld_{max}-Thld_{min}}\right) \quad (4)$$

iii. When $q_{Avg} \ge Thld_{max}$, then all incoming packets are rejected with a probability of one.

Over the last three decades, several scientists have intensified the quest for the development of vast and sophisticated models with advanced performance, leading to the emergence of newer alternative refined approaches for RED. A push factor in these derivative notable models is the redesigning of the deciding packet dropping distribution of RED, aiming at addressing the various deficiencies that plagued RED. As it has been rightly picked out in (Adamu *et al*., 2021; Pan *et al*., 2022; Gim´enez *et al*., 2022; Zhou *et al*., 2006; Zheng and Atiquzzaman,2006; Sumannapong *et al*., 2025; Patel and Hawakami, 2016; Hassan *et al*., 2022; Mahajan and Singh, 2014; Patel and Karmeshu, 2019; Kato *et al*., 2023; Hassan, 2022; Hassan *et al*., 2022; Bie *et al*., 2022; Hassan *et al*., 2024; Hassan *et al.*, 2023; Sumannapong and Khunboa,2019; Kumar *et al.*, 2021; Jafri *et al.*, 2022; Hassan and Rufai, 2023; Floyd, 2000; Hassan *et al.*, 2022; Abdel-Jaber, 2020; Suwannapong and Khunboa, 2019), among other shortcomings of RED, the most consequential is the linear (degree 1) function used in its packet dropping distribution deciding principle.

The DSRED (an abbreviation of 'Double Slope RED') model in Zheng and Atiquzzaman (2006) typically implemented double first order (linear) functions between the lower $Thld_{min}$ and upper $Thld_{max}$ threshold congestion levels of RED, aiming to enforce a better throughput performance. It should be noted that the development of DS-RED considered a mid-threshold between the lower and upper threshold limits, that is, mid-threshold, ($Thld_{min}$ + $Thld_{max}$)/2. The Double Slope Exponential-RED, abbreviated to DSE-RED model (Sumannapong *et al*, 2024) was expressly built upon the design of DS-RED. Specifically, DSE-RED triggers an exponential function as a substitute for the first order (linear) function used in the section between mid-threshold and $Thld_{max}$. Researchers in (Prabhavat and Varakulsiripunth, 2004) primarily disclosed low throughput as well excessive consecutive drop as two crucial challenges of RED. These shortcomings were bypassed by conferring a quadratic (second order) polynomial function between the upper threshold $Thld_{max}$ and the buffer limit instead of forcefully dropping packets by probability of 1. The modified scheme was called Extended DSRED (denoted ExRED). The RTRED (an abbreviation of 'Risk Threshold RED') by authors in (Hamadneh *et al.*, 2011) involves setting a new higher dropping level called the *risk threshold* above $Thld_{max}$ and dynamically setting the P_{max} parameter.

RTRED was able to reduces the packet loss rate but unable to obtain a lower average queue size than RED. In the work by Hassan *et al.*, (2022), the researchers suggested a modification that implements second-order (quadratic) shape function and a first-order (linear) function both implemented within the congestion avoidance segment of RED. IM-RED obtained a considerable delay reduction. Patel and Karmeshu (2019) proposed an optimized enhancement for RED. The authors suggested a procedure represented by Eq. (5) for achieving increased throughput. The model has a limitation as it increased the loss rate.

$$P_b = 1 - \left[p_1 \times \frac{-log(p_1)}{(1+count)}\right] \quad (5)$$

Such that

$$p_1 = \frac{P_{max} \times (q_{Avg} - Thld_{min})}{(Thld_{max} - Thld_{min})} \quad (6)$$

Likewise, in the research done by Patel and Hawakami (2016), the congestion avoidance section of RED was segmented into two subdomains. One part implements a second order (quadratic) drop function, while other segments confers a first order (linear) drop function. The modified scheme was called SmRED (an abbreviation for 'Smart RED'). SmRED claimed a trade-off between delay and throughput. Mahajan and Singh (2014) proposed a revised RED policy named modified Gaussian function-RED, MGF-RED for short. MGF-RED confers a modified Gaussian function on the congestion avoidance region of RED for deciding its dropping probability. MGF-RED was successful at offering an increased throughput. Kumar *et al.*, (2021) suggested a quadratic function (degree 2) when q_{Avg} varies from $Thld_{min}$ and the buffer limit, Q_{max} thresholds using dropping functions represented in Eqs. (7) and (8). The new idea was called *Quadratic RED* and it achieved a higher throughput gain than RED.

$$P_b = \left(\frac{q_{Avg} - Thld_{min}}{Q_{max} - Thld_{min}}\right)^2 \quad (7)$$

or

$$P_b = 1 - \left(\frac{Q_{max} - q_{Avg}}{Q_{max} - Thld_{min}}\right)^2 \quad (8)$$

Following the same research path, Jafri et al., (2022) chose to implement a nonlinear (sigmoid) for the congestion avoidance region of RED, leading to an outstanding delay reduction. The new model was tagged Aggressive-RED, AgRED for short. Floyd (2000) also named low throughput as a major challenge of RED, proposing Gentle RED (GRED). Specifically, in GRED, this deficiency was overcomed by suggesting $2 \times Thld_{max}$ as another threshold upper congestion level. Such that another first order (linear) function is conferred between the coordinates ($Thld_{max}, P_{max}$) and ($2 \times Thld_{max}$, 1). Zhou *et al.*, (2006), low throughput was spotted as a central difficulty of RED. To address this issue, the researchers suggested a quadratic function (degree 2) (as shown in Eq. (9)) as a potential substitute for the first order (linear) strategy of RED. In NLRED, the value of $P_{max}$was increased by $1.5 \times$. The new approach was named nonlinear RED (NLRED). NLRED was able to improve the throughput of RED at least to some extent.

$$P_b = \begin{cases} 0, q_{Avg} < Thld_{min} \\ P_{max}\left(\frac{q_{Avg} - Thld_{min}}{Thld_{max} - Thld_{min}}\right)^2, Thld_{min} \le q_{Avg} < Thld_{max} \\ 1, q_{Avg} \ge Thld_{max} \end{cases} \quad (9)$$

Continuing on a similar research track, Bie *et al.*, (2022), an alternative design that implements a 3rd degree polynomial function shape for the congestion avoidance region of RED, leading to an improved throughput. The new scheme was called Improved ARED (an abbreviation of 'improved adaptive RED'). The scheme is plagued with its inability to curtail the average queue size. Hassan *et al.*, (2022), picked out large delay as a central weakness of RED. This deficiency was addressed by allowing a quadratic shape function and a first order function in-between between the lower $Thld_{min}$ and upper $Thld_{max}$ threshold congestion levels of RED.

Furthermore, to combat the low throughput at low workload and large delay at high workload deficiencies of RED, authors in Feng *et al.*, (2017) came up with an improved model that implements three dropping function: i) third order (cubic), ii) first order (linear), and iii) another cubic (third order). These functions were implemented between $Thld_{min}$ and $Thld_{max}$ congestion levels of RED. The new policy was named Three-section RED (TRED). Abdel-Jaber (2020) established that P_{max} could be circumvented from the number of control parameters needed by RED for determining the dropping probability. As such the Exponential RED (RED_E) was suggested. In RED_E, an exponential function that grow quickly between the coordinates ($Thld_{min}$, 0) and ($Thld_{max}$, 1) was developed to bypass the large delay limitation of RED. Progressing on the same line of reasoning, Kato *et al.*, (2023), initiated an exponential drop function in the packet dropping distribution for RED. In particular, an exponential drop function was conferred on the congestion avoidance section of RED. The model developed uses a gamma (γ) parameter for exploiting the tilting extent of the exponential function. The suggested model was tagged RED with nonlinearity. An increased throughput was reported.

In this paper, we have resorted to revisit RED by proposing a new minimal enhancement to its inefficient design. Unlike existing works that springs from RED, the proposed Refined Random Early Detection (Ref-RED) which incorporates a fully optimized packet dropping distribution uses three dissimilar dropping functions aiming at increasing the overall network performance during various traffic workloads.

MATERIALS AND METHODS

Refined Random Early Detection (Ref-RED) Algorithm

In this section, we present a detailed design of our proposed algorithm called Refined Random Early Detection (Ref-RED). In Ref-RED, the gateway buffer queue is segmented into three subdomains between $Thld_{min}$ and $Thld_{max}$ thresholds similar to TRED as follows:

i. A subdomain that supports when q_{Avg} varies from $Thld_{min}$ to $Thld_{min} + \Delta$
ii. A subdomain that supports when q_{Avg} lies between $Thld_{min} + \Delta$ and $Thld_{min} + 2\Delta$.
iii. A subdomain that supports when q_{Avg} varies from $Thld_{min} + \Delta$ to $Thld_{max}$.

It should be noted that Δ is obtained as follows

$$\Delta = \left[\frac{Thld_{max} - Thld_{min}}{3}\right] \quad (10)$$

Cubic function is known to grow slowly at the beginning and eventually accelerates as the magnitude of inputs gets larger. This feature makes the cubic function suitable to sustain low network traffic load. Therefore, when the valuated q_{Avg} is situated within 0 and $Thld_{min}$, Ref-RED would certainly enqueue every incoming packets with a probability of zero. When q_{Avg} appears in the first subdomain (in the sense $Thld_{min} \leq q_{Avg} < Thld_{min} + \Delta$), a cubic function that exhibits a relaxed and less belligerent drop function using the coordinate points ($Thld_{min}$, 0) and ($Thld_{min} + \Delta$, $\frac{P_{max}}{3}$) is utilized for computing the packet dropping probability as in

$$P_b = 9P_{max}\left(\frac{q_{Avg} - Thld_{min}}{Thld_{max} - Thld_{min}}\right)^3 \quad (11)$$

Square root function exhibits a more sharply increased growth at the beginning and slows down as the input gets larger. This attributes makes the square root function suitable to support a moderate network traffic load. Thus, if the appraised value of q_{Avg} shows up to exist in the second subdomain (in the sense $Thld_{min} + \Delta \leq q_{Avg} < Thld_{min} + 2\Delta$), Ref-RED began to utilize a more belligerent frop action. In this region, a square root function (having the coordinates ($Thld_{min} + \Delta$, $\frac{P_{max}}{3}$) and ($Thld_{min} + 2\Delta$, P_{max}) that accelerates higher is required for computing the packet dropping probability, as in

$$P_b = \frac{2P_{max}}{3}\left[\frac{3\times[q_{Avg} - (Thld_{min} + 2\Delta)]}{Thld_{max} - Thld_{min}}\right]^{\frac{1}{2}} + \frac{P_{max}}{3} \quad (12)$$

Exponential function growth is believed to be significantly faster than square root function. This property makes the exponential function fit to sustain a heavy network traffic load. Hence, if q_{Avg} is found to be stationed in the third subdomain (as in $Thld_{min} + 2\Delta \leq q_{Avg} < Thld_{max}$) Ref-RED utilizes a higher belligerent structure of packet dropping following an exponential function (with coodinate points ($Thld_{min} + 2\Delta$, P_{max}) and ($Thld_{max}$, 1) for computing the dropping probability. It follows that

$$P_b = e^{log(P_{max})\left[\frac{3\times(Thld_{max} - q_{Avg})}{(Thld_{max} - Thld_{min})}\right]} \quad (13)$$

When the appraised value of q_{Avg} equals or goes beyound the bounds of $Thld_{max}$, Ref-RED drops every incoming packets just as RED would perform, as in

$$P_b = 1 \quad (14)$$

More formally, depending on the value of q_{Avg}, the packet dropping distribution for Ref-RED can be represented as shown in Equation (15).

$$P_b = \begin{cases} 0, q_{Avg} < Thld_{min} \\ 9P_{max}\left(\frac{q_{Avg} - Thld_{min}}{Thld_{max} - Thld_{min}}\right)^3, Thld_{min} \leq q_{Avg} < Thld_{min} + \Delta \\ \frac{2P_{max}}{3} \times \left[\frac{3\times[q_{Avg} - (Thld_{min} + 2\Delta)]}{Thld_{max} - Thld_{min}}\right]^{\frac{1}{2}} + \frac{P_{max}}{3}, Thld_{min} + \Delta \leq q_{Avg} < Thld_{min} + 2\Delta \\ e^{log(P_{max})\left[\frac{3\times(Thld_{max} - q_{Avg})}{(Thld_{max} - Thld_{min})}\right]}, Thld_{min} + 2\Delta \leq q_{Avg} < Thld_{max} \\ 1, q_{Avg} \geq Thld_{max} \end{cases} \quad (15)$$

The dropping function for Ref-RED shown in Figure 1 was obtained courtesy of the values presented in Table 1. As can be observed from the fore-going description, Ref-RED differs from TRED in some sense. Such that:

i. Ref-RED utilizes a nonlinear square root function in the second subdomain that runs through two coordinates: ($Thld_{min} + \Delta$, $\frac{P_{max}}{3}$) and ($Thld_{min} + 2\Delta$, P_{max}), while TRED implements a linear function that operates through the coordinates: ($Thld_{min} + \Delta$, $\frac{P_{max}}{3}$) and ($Thld_{min} + 2\Delta$, $\frac{2P_{max}}{3}$) in the same second subdomain.
ii. Ref-RED model uses an exponential function in the third subdomain that operates from the coordinates ($Thld_{min} + 2\Delta$, P_{max}) and ($Thld_{max}$, 1), while TRED model uses a cubic function that operates from($Thld_{min} + 2\Delta$, $\frac{2P_{max}}{3}$) to ($Thld_{max}$, P_{max}).

However, both Ref-RED and TRED implements the same dropping function in the first subdomain using the same coordinates. The pseudocode for implementing the Ref-RED model is displayed in Algorithm 1.

Table 1: Parameter settings

Parameter	Value
P_{max}	0.1 (as endorsed by Floyd and Jacobson (1993))
$Thld_{max}$	90 packets
$Thld_{min}$	30 packets (configures to one third of as endorsed by Floyd and Jacobson (1993))

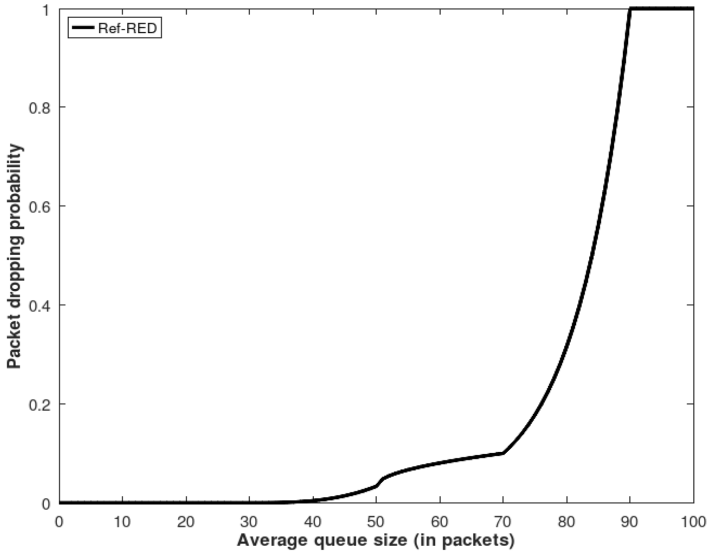


Figure 1: Dropping function for Ref-RED

Algorithm 1: Ref-RED Algorithm

1: Set preconceived control parameters: $Thld_{max}$, P_{max}, $Thld_{min}$, q_w

2: Set $q_{Avg} = 0$

3: Set $\Delta = \left[\frac{Thld_{max}-Thld_{min}}{3}\right]$

Upon every single packet appearance at the router's buffer do:

Appraise the congestion level by solving for q_{Avg} (with the formula in Eq. (1))

if ($q_{Avg} < Thld_{min}$)then

Enqueue the packet

else if ($Thld_{min} \leq q_{Avg} < Thld_{min} + \Delta$)then

Work out the drop probability P_b, by using the formula that follows:

$$P_b = 9P_{max}\left(\frac{q_{Avg} - Thld_{min}}{Thld_{max} - Thld_{min}}\right)^3$$

Perform packet dropping based on the valuated probability

else if ($Thld_{min} + \Delta \leq q_{Avg} < Thld_{min} + 2\Delta$) then

Figure out the drop probability P_b in support of the following formula:

$$P_b = \frac{2P_{max}}{3} \times \left[\frac{3 \times \left[q_{Avg} - (Thld_{min} + 2\Delta)\right]}{Thld_{max} - Thld_{min}}\right]^{\frac{1}{2}} + \frac{P_{max}}{3}$$

Perform packet dropping based on the valuated probability

else if ($Thld_{min} + 2\Delta \leq q_{Avg} < Thld_{max}$)then

Find out the drop probability P_b via the formula listed as follows:

$$P_b = e^{log(P_{max})\left[\frac{3\times(Thld_{max}-q_{Avg})}{(Thld_{max}-Thld_{min})}\right)}$$

Perform packet dropping based on the valuated probability

else if ($q_{Avg} \geq Thld_{max}$) then

Drop packet

end if

RESULTS AND DISCUSSION

Performance Assessment

In this study, simulation experiments were conducted using the ns-3 discrete-event network simulation platform. Details regarding the configuration of the simulation topology (depicted in Figure 2) is displayed in Table 2. In the figure, network nodes labelled S_1 to S_N are implemented as TCP packet sendets in which the specific value of *N* is varied to reflect different levels of a congested network. The bottleneck in the network topology is formed between the two routing/intermediary nodes, Router1 and Router2.

Three experimental scenarios were conducted in the study. Firstly, a scenario that implements 50 nodes as packet senders to signify a light network traffic load. Secondly, a simulation experiment that sets 100 as the number of packet generator nodes to signify a mid network traffic load. Lastly, implementing 200 nodes at the sender side representing a heavy network traffic load.

The proposed Ref-RED algorithm together with the other AQM algorithms, namely, NLRED and TRED are sequentially injected in the queue of Router1. More importantly, we opted for these two algorithms so serve as sufficient benchmark algorithms for the simple reason that they both do not require a labor-intensive imlplementation task and have been reported to yield improved performance than RED. Furthermore, latter-day AQM schemes, such as Controlled Delay (CoDel) introduced by Nichols and Jacobson (2012), Common Applications Kept Enhanced (CAKE) developed by Høiland-Jørgensen *et al*. (2018), and Proportional Integral controller Enhanced (PIE) suggested by Pan *et al*. (2013) that uses queuing latency rather queue length as congestion indicator were not considered as benchmark models. Evaluation metrics assessed include average queue size, throughput, and delay.

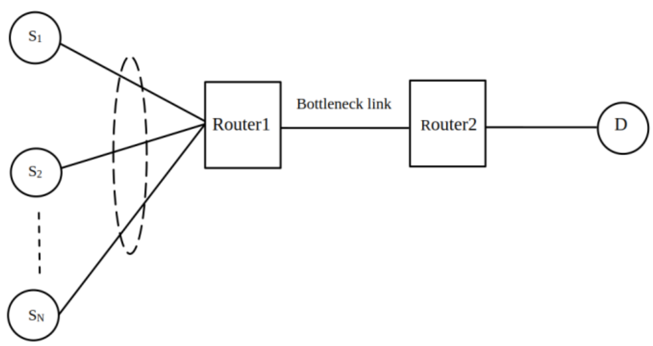


Figure 2: Simulation topology

Simulation Experiment 1

This simulation scenario implements a total of 50 sender nodes configured to generate packets to be sent to the sink, namely, *D*. This scenario showcases a light loading of the network traffic.

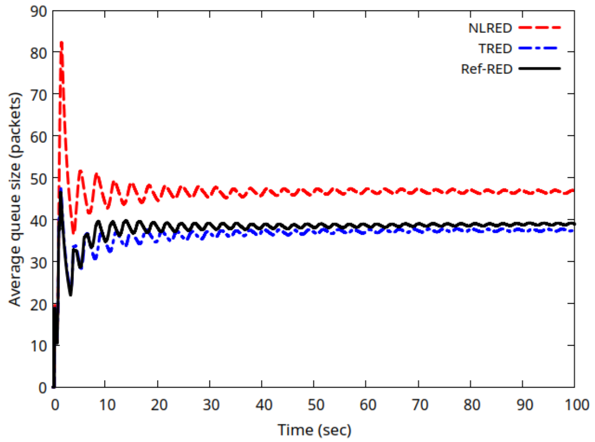
Figure 3 presents the results for average queue size, delay, and throughput. As illustrated in Figure 3(a), Ref-RED more thoroughly kept shorter the average queue size than NLRED, while TRED performed better than Ref-RED. As presented in Figure 3(b), TRED provides the lowest delay performance, while Ref-RED still outperformed NLRED. In Figure 3(c), the plot of the throughput is illustrated. It can be observed that Ref-RED is more effective than others in having a high throughput. Table 3 presents the analysis of these results whereby values for the better performing algorithm is displayed in bold.

Table 2: Simulation configuration

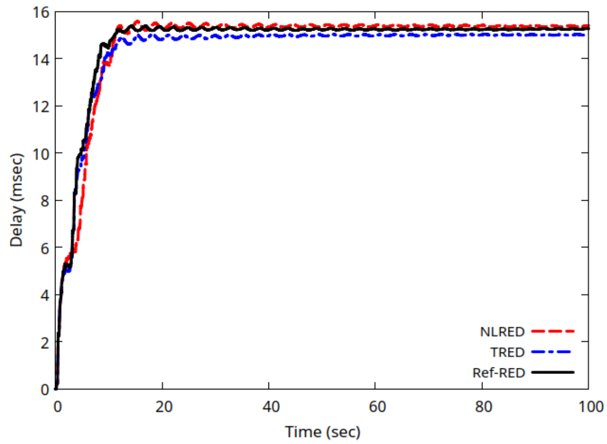
Parameter	Value
Simulation topology model	Double router dumbbell
Bandwidth of access links	100 Mb/sec
Propagation delay of access links	2 msec
Bandwidth of bottleneck link	10 Mb/sec
Propagation delay of bottleneck link	100 msec
Mean packet size	1000-bytes long
Buffer size	200 packets
Implemented TCP	New Reno
P_{max}	0.1 (as endorsed by Floyd and Jacobson (1993))
$Thld_{max}$	90 packets

$Thld_{min}$	30 packets (configured to one third of as endorsed by Floyd and Jacobson (1993))
q_w	0.002 (as endorsed by Floyd and Jacobson (1993))
Simulation time set	100 sec

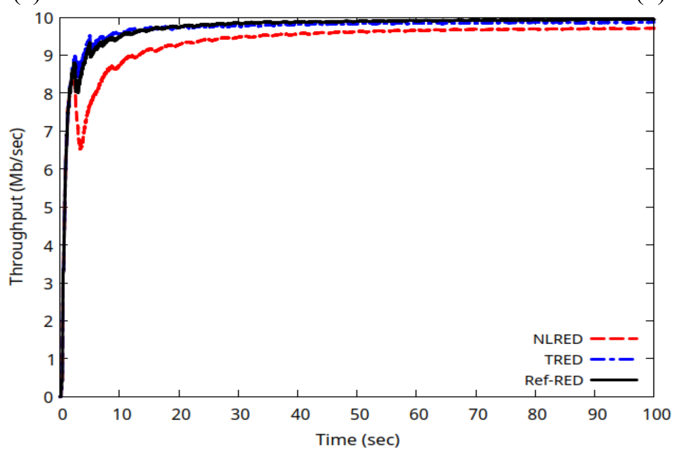
Simulation Experiment 2
Under this experimental scenario, a total of 100 nodes were configured at the sender side for the purpose of generating data packets to be transmitted to the packet receiver, *D*. This simulation scenario showcases a moderate loaded network traffic.



(a)



(b)



(c)

Figure 3: (a) Average queue size, (b) delay, and (c) throughput comparison for different AQM policies during light network traffic congestion

Table 3: Mean value of AQM policies under light network traffic workload

Criteria ↓	**NLRED**	**TRED**	**Ref-RED**
Delay (msec)	14.6494	14.3477	14.6423
Throughput (Mb/sec)	9.3335	9.6667	9.7043
Average queue size (packets)	46.4472	36.2781	37.8106

Figure 4 shows a comparison of the average queue size for the three algorithms. It can be seen that Ref-RED outperformed TRED to a greater degree. Also, NLRED performed better than TRED. As results presented in Figure 4(a) verifies the observation in Figure 4(b). Meaning that Ref-RED more competently decreases delay than TRED, while we can see that NLRED gives the best performance. Figure 4(c) presents the throughput results, where we can observe that Ref-RED more attractively improves throughput than NLRED, while TRED gives the highest throughput. Table 4 presents the analysis of these results whereby values for the better performing algorithm is displayed in bold.

Simulation Experiment 3

Under this experimental scenario, a total of 100 nodes were configured at the sender side for the purpose of generating data packets to be transmitted to the packet receiver, *D*. This scenario showcases a heavy loading of the network traffic.

As shown in Figure 5(a), a considerable performance improvement was achieved by Ref-RED in stabilizing and maintaining a lesser average queue size compared to NLRED and TRED. Although TRED offered a better result than NLRED. This observation is further verified by the plot in Figure 5(b) showing that Ref-RED more advantageously results in lower delay compared to NLRED and TRED. These results are due to the aggressive exponential phase of Ref-RED. Figure 5(c) shows that TRED outperformed Ref-RED, while Ref-RED in turn significantly improves throughput compared to NLRED. Table 5 presents the analysis of these results whereby values for the better performing algorithm is displayed in bold.

Table 4: Mean value of AQM policies under mid network traffic workload

Criteria ↓	NLRED	TRED	Ref-RED
Delay (msec)	30.0893	30.0893	29.4777
Throughput (Mb/sec)	9.8583	9.8583	9.8444
Average queue size (packets)	58.2044	58.2044	55.6836

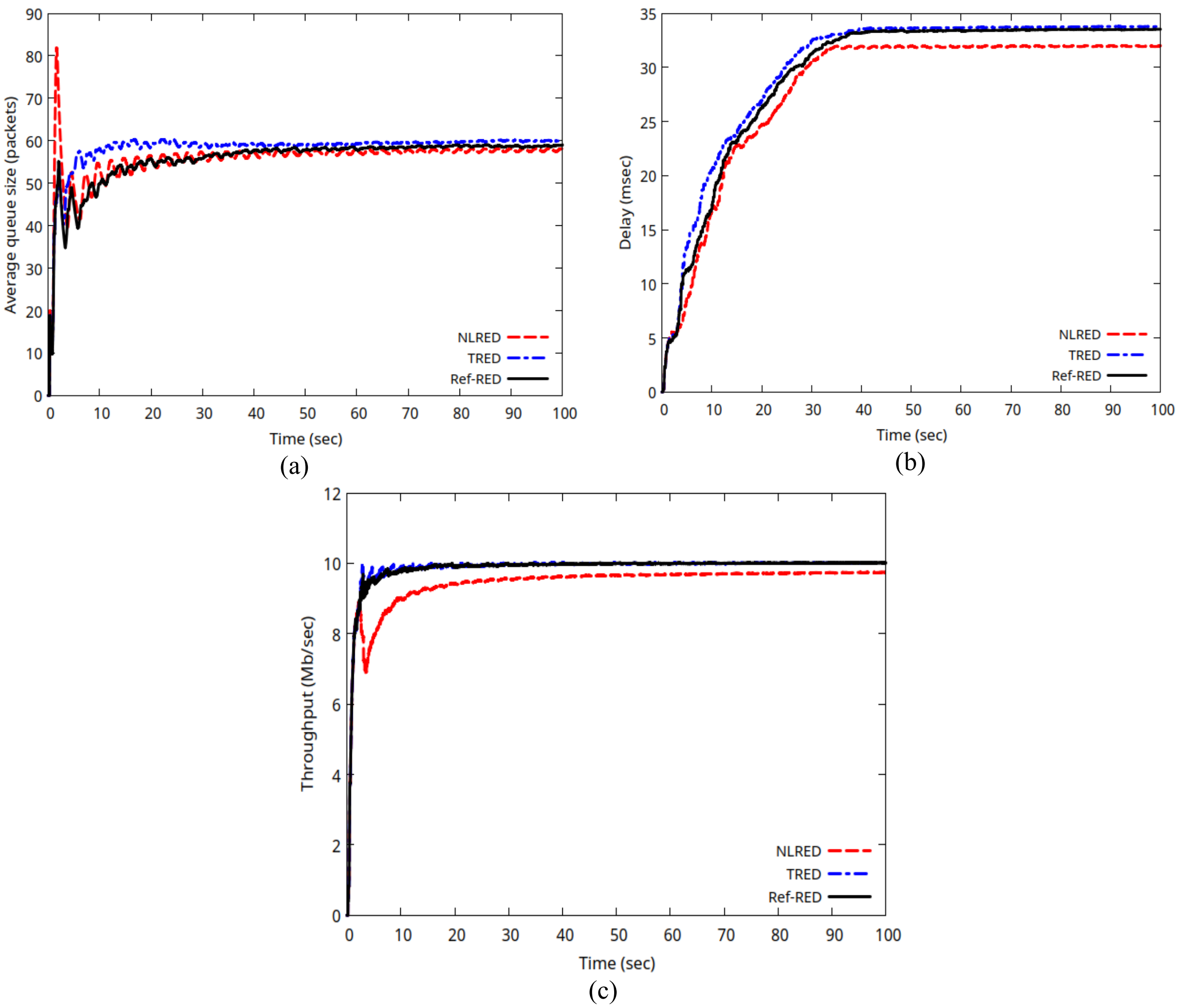
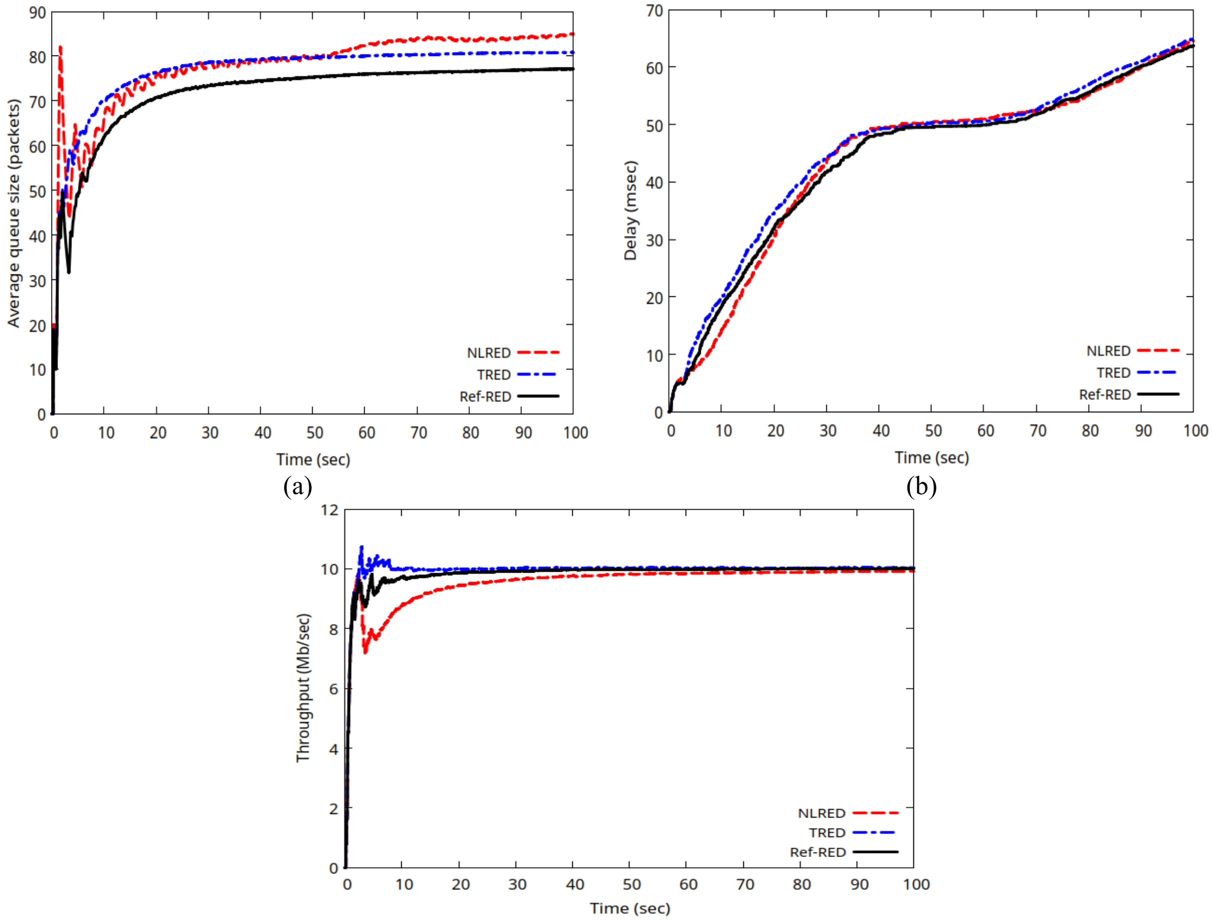


Figure 4: (a) Average queue size, (b) delay, and (c) throughput comparison for different AQM policies during mid network traffic congestion



(c)

Figure 5: (a) Average queue size, (b) delay, and (c) throughput comparison for different AQM policies during heavy network traffic congestion

Table 5: Mean value of AQM policies under heavy network traffic workload

Criteria ↓	NLRED	TRED	Ref-RED
Delay (msec)	44.0443	45.3987	43.9421
Throughput (Mb/sec)	9.5389	9.9418	9.8255
Average queue size (packets)	77.6210	76.6324	71.5660

CONCLUSION

In this study, we described and suggested the Ref-RED (an abbreviation of 'Refined Random Early Detection') algorithm. By leveraging the design principles of TRED algorithm, Ref-RED implements a packet dropping distribution that uses different functions: cubic (not so aggressive), square root (a lot more aggressive), and exponential (far more aggressive), bypassing the limitations of an indiscriminant linear drop function of the original RED scheme. Evaluations in ns-3 simulation environment confirms that the proposed Ref-RED algorithm more attractively improves performance by mitigating network congestion under different traffic congestion settings. Ref-RED effectively reduces the average queue size compared to TRED under the mid and heavy traffic congestion levels. In our future work, we could explore the possible implementation and experimental study of Ref-RED in a real network environments (instead of resorting to simulation experimental environments). We will also study the performance of Ref-RED against some well-known queuing-delay based AQM schemes.

REFERENCES

Abdel-Jaber, H. (2025). Performance Analysis of Diverse Active Queue Management Algorithms. *International Journal of Networked and Distributed Computing*, 13(1), 1–32, https://doi.org/10.1007/s44227-025-00056-1

Abdel-Jaber, H., Thabtah, F., and Woodward, M. (2015). Modeling discrete-time analytical models based on random early detection: Exponential and linear. *International Journal of Modeling, Simulation, and Scientific Computing*, 6(3), 1550028-1– 1550028-22, https://doi.org/10.1142/S1793962315500282

Abdel-Jaber, H. (2020). An exponential active queue management method based on random early detection. *Journal of Computer Networks and Communications*, 2020, 1–11, https://doi.org/10.1155/2020/8090468.

Abu-Shareha, A. A. (2019). Controlling delay at the router buffer using modified random early detection. *International Journal of Computer Networks and Communications (IJCNC)*, 11(6), 63–75, https://doi.org/10.5121/ijcnc.2019.11604.

Adamu, A., Surajo, Y. and Jafar, M. T. (2021). Sared: Self-adaptive active queue management scheme for improving quality of service in network systems, *Journal of Computer Science*, 22(2), 253–267, https://doi.org/10.7494/csci.2021.22.2.4020.

Ali, I., Hong, S. and Cheung, T. (2024). Quality of Service and Congestion Control in Service-Defined Networking Using Policy-Based Routing. *Applied Sciences*, 14, 9066, https://doi.org/10.3390/app14199066.

Bie, Y., Li, Z., Hu, Z. and Chen, J. (2022). Queue Management Algorithm for Satellite Network Based on Traffic Prediction. *IEEE Access*, 10, 54313–54324, https://doi.org/10.1109/ACCESS.2022.3163519

Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Patridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and Zhang, L., "Recommendations on Queue Management and Congestion Avoidance in the Internet" *RFC 2309*, (1998).

Chen, L., Wang, X., and Han, Z. (2004). Controlling Chaos in Internet Congestion Control Model. *Chaos, Solitons and Fractals*, 21(1), 81–91, https://doi.org/10.1016/j.chaos.2003.09.037

Chydzinski, A., and Adamczyk, B. (2023). On the Influence of AQM on Serialization of Packet Losses. *Sensors*, 23(4), 2197, https://doi.org/10.3390/s23042197

Feng, C. W., Huang, L. F., Xu, C., and Chang, Y. C. (2017). Congestion control scheme performance analysis based on nonlinear red. *IEEE Systems Journal*, 11(4), 2247–2254, https://doi.org/10.1109/JSYST.2014.2375314.

Floyd, S. (2000). Recommendation on using the gentle_variant of red. Available on: https://www.icir.org/floyd/red/gentle.html.

Floyd, S., and Jacobson, V. (1993). Random early gateway for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4), 397–413, https://doi.org/10.1109/90.251892.

Foundation, N. S. The network simulator - ns-3, Available on: https://www.nsnam.org. Last accessed: 25 October 2025

Giacomoni, L., Benny, B. and Parisis, G. (2023). RayNet: A Simulation Platform for Developing Reinforcement Learning-Driven Network Protocols, *arXiv:2302.04519v1[cs.NI]*, pp. 1–24, , https://doi.org/10.48550/arXiv.2302.04519.

Gim´enez, A., Murcia, A., Amig´o, J. M., Mart´ınez-Bonastre, O., and Valero, J. (2022). New red-type tcp-aqm algorithms based on beta distribution drop functions, *arXiv:2201.01105v1[cs.NI]*, https://doi.org/10.48550/arXiv.2201.01105

Hamadneh, N., Dixon, M., Cole, P. and Murray, D. (2011). A Third Drop Level for TCP-RED Congestion Control Strategy. *World Academy of Science, Engineering and Technology*, 8(1), 892–896,.

Hassan, S. O. (2022). RED-LE: A Revised Algorithm for Active Queue Management. *Journal of Telecommunications and Information Technology*, (2), 91–97, https://doi.org/10.26636/jtit.2022.160022.

Hassan, S. O., Nwaocha, V. O., Rufai, A. U., Odule, T. J., Enem, T. A., Ogundele, L. A., and Usman, S. A. (2022). Random Early Detection-Quadratic Linear: An Enhanced Active Queue Management Algorithm. *Bulletin of Electrical Engineering and Informatics*, 11(4), 2262–2272, https://doi.org/10.11591/eei.v11i4.3875

Hassan, S. O., Rufai, A. U., Ajaegbu, C., and Ayankoya, F. (2022). DL-RED: a RED-based Algorithm for Routers. *International Journal of Computer Applications in Technology*, 70(3/4), 244–253, https://doi.org/10.1504/IJCAT.2022.10053495

Hassan, S. O., Rufai, A. U., Agbaje, M. O., Enem, T. A., Ogundele, L. A. and Usman, S. A. (2022). Improved Random Early Detection Congestion Control Algorithm for internet routers. *Indonesian Journal of Electrical Engineering and Computer Science*, 28(1), 384–395, https://doi.org/10.11591/ijeecs.v28.i1.pp384-395.

Hassan, S. and Rufai, A. (2023). Modified Dropping - Random Early Detection (MD-RED): a Modified Algorithm for Controlling Network Congestion. *International Journal of Information Technology*, 15(3), 1499–1508, https://doi.org/10.1007/s41870-023-01201-1.

Hassan, S. O., Solanke, O. O., Odule, T. J., Adesina, A. O., Usman, S. A. and Ayinde, S. A. (2024). AmRED and RED-QE: Redesigning Random early Detection Algorithm. *Telecommunication Systems*, 85(2), 263–275, https://doi.org/10.1007/s11235-023-01082-6.

Hassan, S. O., Rufai, A. U., Nwaocha, V. O., Ogunlere, S. O., Adegbenjo, A. A., Agbaje, M. O. and Enem, T. A. (2023). Quadratic Exponential Random Early Detection: a New Enhanced Random Early Detection-Oriented Congestion Control Algorithm for Routers. *International Journal of Electrical and Computer Engineering*, 13(1), 669–679, https://doi.org/10.11591/ijece.v13i1.pp669-679

Høiland-Jørgensen, T., Täht, D., and Morton, J. (2018). Piece of Cake: A Comprehensive Queue Management Solution for Home Gateways. In Proceedings of 2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pp. 37-42

Imputato, P., Avallone, S., Tahiliani, M. P., and Ramakrishnan, G. (2020). Revisiting design choices in queue disciplines: The PIE case. *Computer Networks*, 171(22), 107136, https://doi.org/10.1016/j.comnet.2020.107136

Jafri, S. T. A., Ahmed, I., and Ali, S. (2022). Queue-Buffer Optimization Based on Aggressive Random Early Detection in Massive NB-IoT MANET for 5G Applications, *Electronics*, 11(18), 1–18, https://doi.org/10.3390/electronics11182955

Jian, V., Mittal, V., Shravya, K. S., and Tahiliani, M. P. (2018). Implementation and Validation of BLUE and PI Queue. *Simulation Modelling Practice and Theory*, 84(22), 19–37, https://doi.org/10.1016/j.simpat.2018.01.002

Kato, K., Kato, H., Asahara, H., Ito, D., and Kousaka, T. (2023). Effects on Random Early Detection of the Packet Drop Probability Function with an Adjustable Nonlinearity. *Nonlinear Theory and Its Applications (NOLTA)*, 14(2), 193–206, https://doi.org/10.1587/nolta.14.193.

Kaur, N., and Singhai, R. (2019). Congestion control scheme using network coding with local route assistance in mobile adhoc network. *International Journal of Computer Applications in Technology*, 60(3), 242–253, https://doi.org/10.1504/ijcat.2019.100298

Kumhar, C., Kumar, A. and Kewat, A. (2021). QRED: an Enhancement Approach for Cngestion Control in Network Communication. *International Journal of Information Technology*, 13(1), 221–227, https://doi.org/10.1007/s41870-020-00538-1.

Long, C., Zhao, B. M, Guan, X. and Yang, J. (2005). The Yellow Active Queue Management Algorithm. *Computer Networks*, 47(4), 525–550, https://doi.org/10.1016/j.comnet.2004.09.006.

Mahajan, M., and Singh, T. P. (2014). The Modified Gausian Function for Congestion Avoidance in Mobile Ad Hoc Networks. *International Journal of Computer Applications*, 91(6), 39–44, https://doi.org/10.5120/15889-5112

Nichols, K. and Jacobson, V. (2012). Controlling Queue Delay. *Communications of the ACM*, 55(7), 42-50.

Pan, C., Zhang, S., Zhao, C., Shi, H., Kong, Z., and Cui, X. (2022). A Novel Active Queue Management Algorithm Based on Average Queue Length Change Rate. *IEEE Access*, 10, 75558–75570, https://doi.org/10.1109/ACCESS.2022.3189183

Pan, R., Natarajan, P., Piglione, C., Prabhu, M. S., Subramanian, V., Baker, F., and VerSteeg, B. (2013). PIE: A Lightweight Control Scheme to Address the Bufferbloat Problem. In Proceedings IEEE 14th International Conference on High Performance Switching and Routing (HPSR), pp. 148-155.

Patel, S. and Karmeshu. (2019). A New Modified Dropping Function for Congested AQM Networks. *Wireless Personal Communications*, 104(1), 37–55, https://doi.org/10.11007/s11277-018-6007-8

Paul, A. K., Kawakami, H., Tachibana, A., and Hasegawa, T. (2016). An aqm based congestion control for enb rlc in 4g/lte network. *Proceedings of the IEEE Canandian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–5, https://doi.org/10.1109/CCECE.2016.7726792.

Pei, L., Wu, F., and Wang, S. (2021). Periodic, quasi-periodic and chaotic oscillations in two heterogeneous aimd/red network congestion models with state-dependent roundtrip delays. *International Journal of Bifurcation and Chaos*, 31(6), 2150124-1– 2150124-21, https://doi.org/10.1142/S0218127421501248.

Prabhavat, S. and Varakulsiripunth, R. (2004). Performance Improvement on RED Based Gateway in TCP Communication Network. *In Proceedings of the International Conference on Control, Automation Systems*, 782–787.

Ryu, S., Rump, C., and Qiao, C. (2003). Advances in Internet Congestion Control. *IEEE Communications Surveys and Tutorials*, 5(1), 28-39, https://doi.org/10.1109/COMST.2003.5342228

Sumannapong, S. Atsawaraungsuk, K. Somsuk and P. Chaichitwanidchakol. (2025). FaCoCo-RED: A Fast Response Congestion Control Mechanism for Constrained Application Protocol. *Electronics*, 14, https://doi.org/10.3390/electronics14010028.

Suwannapong, C., and Khunboa, C. (2019). Congestion Control in CoAP Observe Group Communication. *Sensors*, 19(3433), 1–14, https://doi.org/10.3390/s19153433

Thiruchelvi, G., and Raja, J. (2008). A Survey On Active Queue Management Mechanisms. *International Journal of Computer Science and Network Security*, 8(12), 130–145.

Zhou, K., Yeung, K. L., and Li, V. O. K. (2006). Nonlinear red: A simple yet efficient active queue management scheme. *Journal of Computer Networks*, 50(18), 3784–3794, https://doi.org/10.1016/j.comnet.2006.04.007.

Zhu, H., Sun, H., Jiang, Y., He, G., Zhang, L. and Lu, Y. (2023). A Sketch-Based Fine-Grained Proportional Integral Queue Management Method. *Axioms*, 12, 814, https://doi.org/10.3390/axioms12090814.