A Fairness Investigation on Active Queue Management Schemes in Wireless Local Area Network

Suhaidi Hassan, Atheer F. Hassan, Suki Arif
InterNetWorks Research Lab, School of Computing, Universiti Utara Malaysia,
06010 UUM Sintok, Kedah, Malaysia
atheer.f.h@ieee.org, {suhaidi,suki1027}@uum.edu.my

***Abstract*— Active Queue Management (AQM) is scheme to handle network congestion before it happened by deciding which packet has to be dropped, when to drop it, and through which port have to drop when it has become or is becoming congested. Furthermore, AQM schemes such as Random Early Detection (RED), Random Early Marking (REM), Adaptive Virtual Queue (AVQ), and Controlled Delay (CoDel) have been proposed to maintain fairness when unresponsive constant bit rate UDP flows share a bottleneck link with responsive TCP traffic. However, the performance of these fair AQM schemes need more investigation especially evaluation in WLANs environment. This paper provides an experimental evaluation of different AQM schemes in WLAN environment with presence of two different types of flows (TCP flows and UDP flows) to study the behavior of these AQM schemes which might punish some flows unfairly. The simulation method has conducted in this paper by using Network Simulation 2 (ns-2) with the topology of bottleneck scenario. The result has shown that REM and AVQ both obtain higher fairness value than RED and Codel. However, CoDel has given the lowest fairness comparing with RED scheme which have given a moderated value in terms of fairness in WLANs environment. Besides, AQM schemes must be chosen not only based on its performance or capability to indicate the congestion and recovering overflow situation but also considering fairness with different types of flows and the environment as well, such as WLANs environment.**

***Index Terms*— Experimental Evaluation; Congestion Control, WLANs.**

I. INTRODUCTION

The presence of unresponsive flows in the Internet leads to the problem of unfairness [1]. Responsive flows back off by reducing their sending rate on detecting congestion while unresponsive flows keep on injecting packets into the network incessantly. Whenever responsive and unresponsive flows compete in a best-effort network, unresponsive flows aggressively grab a larger share of bandwidth, thereby depriving the responsive flows of their fair bandwidth share.

In response to these problem, there has been a long history, dating back to [2], of realizing that routers play a significant role in fair bandwidth allocation. The technical report by Floyd and Fall [3] is, however, the first one to extensively demonstrate the danger of unfairness due to unresponsive flows. They also argue that the incentives for cooperative behavior can only come from the network itself, and therefore, routers inevitably need to deploy mechanisms to provide an incentive structure for applications to use end-to-end congestion control. The report also proposes AQM based techniques for identifying and restricting unresponsive flows. Since then, a number of fairness-driven queue management schemes have been proposed to shield responsive flows and to regulate unresponsive and aggressive flows.

Active Queue Management is scheme to indicate the congestion in advance before it happens besides overcome the full buffer situation by dropping or marking the packets [4]. AQM has three main mechanisms: (1) congestion indicator, (2) control function and (3) feedback mechanism [5]–[9]. The congestion indicator detects when the congestion occurs or near to occur whereas control function decides what have to be done when the congestions has been indicated, where the feedback mechanism is the signal that will be sent to notify the sources about the congestions states in order to reduce the sources sending rates. Fairness is one of the earlier goals of AQM while the main concept behind fairness in AQM is to provide an equal share of queue between different type of flows. A variant AQM schemes has been proposed to tackle fairness issue and many mechanisms have been designed to provide fair bandwidth share among different types of flows.

Wireless Local Area Network (WLAN) was introduced in response to the increasing demand of low cost, fast and simple to set up, and use technology in comparison with the previous generation of products. For accommodating these demands, WLAN gained increased interest from a communication trade perspective, and its importance was highlighted in providing easy wireless Internet access in public areas, such as libraries, airport halls, restaurants, and convention centers. The recent emerged standards of IEEE 802.11 and their added functionalities for satisfying the vast range of upcoming service requirements have been discussed by [10].

The fundamental access method in the IEEE 802.11 protocol is the Distributed Coordination Function (DCF) designed based on Carrier-Sense Multiple Access/Collision Avoidance (CSMA/CA) to procure equal opportunity for each competing wireless station to access the channel. However, the basic CSMA/CA method cannot guarantee fair resource allocation between wireless nodes with different features and may lead to an unfairness problem, known as Performance Anomaly of IEEE 802.11b. When there are stations with different data rates in the same wireless cell, the higher data rate station defers its frame transmission longer than that of the lower data rate station.

Consequently, the throughput of all stations transmitting at a high transmission rate is degraded to the level of the lower transmission rate [10].

The key question considered in this paper is therefore whether these fair AQM mechanisms actually provide fairness when multiple responsive flows compete with unresponsive traffic in WLANs environment. Therefore, this paper is to allocate a fair proportion of throughput among TCP flows coming from competing stations with different channel conditions. In order to provide per-rate fairness, we set the window size and packet size of the flows according to the available space of the Access Point (AP) buffer in IEEE 802.11 WLAN infrastructure.

In the next section of this paper we recall some previous surveys and studies conducted on congestion control and AQM. There we also outline the contribution this paper attempts to bring to the research community. In Section III we discussed variant AQM schemes that attempted to improve fairness in the networks. The performance evaluation and analysis is discussed in Section IV including the simulation settings and result discussion. Finally, conclusions are provided in Section V.

II. RELATED WORKS

There have been a number of studies on queue management (see, e.g., [6], [11]–[14]), which are either generic surveys and/or taxonomies of AQM schemes, and/or present simulation based comparative analyses of various AQM schemes. These studies, however, are not entirely focused on the problem of unfairness. In order to put the work in this paper in context, this section highlights only those very few closely related surveys that focus on the problem of unfairness and provide an in-depth review of only “fairness-driven” queue management schemes.

To the best of our knowledge, the first survey on the fairness-driven schemes was conducted in 2001 by Hasegawa and Murata [15]. The paper surveys several approaches including but not limited to queue management schemes. The paper first describes per-flow scheduling approaches and RED variants, which existed up to that time for enforcing fairness. Fair queuing (FQ) [16] and its variants are discussed under the per-flow scheduling; whereas, Flow Random Early Detection (FRED) [17], and Stabilized RED (SRED) [18] are discussed as the RED variants. The paper also investigates the fair-share of resources at end-systems.

The first survey dedicated in particular to fairness-driven queue management was conducted in 2004 by Chatranon et al. [19]. The taxonomy proposed by the authors divides the queue management schemes proposed until that time into two categories: (i) those requiring full per-flow state and (ii) those not requiring full per-flow state. The latter category is further divided into two subcategories: (ii-a) schemes based on the estimation of the number of active flows, and (ii-b) schemes that are not based on the estimation of the number of active flows. The survey includes the description and qualitative comparison for all the schemes. A quantitative comparison is also included in the survey to demonstrate the pros and cons of only those schemes that do not require full per-flow state information. The paper also describes various techniques for estimating the number of active flows traversing a router.

Another study by Chatranon et al. [20] provides the evaluation of fairness of various queue management schemes in presence of a number of TCP variants. The queue management schemes evaluated are Drop-tail, RED, CHOKe [21], CARE [22], and BLACK [23] modified with an improved technique to estimate the number of active flows.

Adamczyk and Chydzi´naski [24] have presented a simulation based comparative performance analysis. The paper studies the impact of seven TCP variants on the performance of seven fairness-driven queue management schemes, including Drop-tail, RED, FRED, CHOKe, and CARE. Particularly, the fairness index, throughput and queue size are analyzed. More recently, Doma´nski et al. [25] have also presented simulation based comparative analysis of CHOKe with four of its enhanced variants. The paper also presents comparisons in a real network operation.

Since 1998, RFC 2309 [26] has been approved by ITEF which is stated the recommendation on queue management and congestion avoidance in the Internet. It’s strongly recommended that AQM should avoid the lock-out phenomena which happens in the router when few flows monopolize all the queue space, preventing other connections from getting room in the queue. This phenomenon is one of the main reasons for the unfairness. During decades AQM has been an elegant and a promising technology that have been taken extensive discussion and debate in IETF meetings until the last RFC 7567 which has been published in July 2015 that stated the latest IETF recommendation regarding AQM. RFC 7567 [27] is clearly stated the presence of lock-out issue and has suggested the researchers to investigate deeply in the issue of unfairness.

A lot of research has been dedicated to developing queue management schemes for identifying and restricting unresponsive flows. Recently, there has been a renewed interest [27] at the IETF to re-emphasize the need for a concerted effort of research, measurement, and ultimate deployment of queue management schemes for protecting the Internet from unresponsive flows. To that end, this paper presents an experimental evaluation and a literature review of the fairness-driven queue management research from the pioneering proposal to most recent schemes, including the taxonomy of these schemes, their strengths and weaknesses, open issues and design guidelines. There is a lack of such a comprehensive recent survey on fairness-driven queue management, as indicated in the previous subsection.

To provide the reader a more complete perspective on fairness-driven queue management research, we have elaborated on the concepts of resource sharing and congestion control, on fundamentals of queue management, and on the notion of fairness. The remainder of this paper reviews eminent queue management schemes developed to address the unfairness problem and describes their strengths and weaknesses. We present a comparison and our analysis of these fairness-driven schemes, discuss open issues, and provide guidelines for future research in this area.

III. FAIRNESS IN AQM SCHEMES

Queue Management is a process to overcome the congestion before it happened by deciding which packet has to be dropped, when to drop it, when it has become or is becoming congested [28]. Simplicity in implementation is one of the properties for the queue management algorithm that make the implementation as simple as applying First-In-First-Out (FIFO) queuing for all the flows or maintaining pre-flow state [29].

Fairness is one of the earlier goals of AQM while the main concept behind fairness in AQM is to provide an equal share of queue between different types of flows. However, it's hard for AQM to differentiate between flows without any supporting information to define the flows type whether responsive, unresponsive, or short flows. Therefore, AQM with per-flow information has given a better result than no per-flow information AQM. Even though, keeping the buffer occupancy equal for each flow individually does not mean the output rate from the buffer will be equal [30]–[32]. The recommended way to guarantee fairness among multiple flows is by combining scheduling algorithms with AQM algorithms as suggested in [32]–[35]. But, the main difficulty behind this combination is a conflict between AQM algorithms objectives and scheduling algorithms objectives, since the AQM algorithms tries to keep the queue as short as possible, whereas scheduling algorithms required longer queues to gain more efficiency [36]. As concluded by [37], the rate-based AQM algorithms have stronger effect on fairness and QoS than scheduling algorithms. Many AQMs have been proposed with respect to fairness such as Fair RED (FRED) [17], Short-lived Flow Friendly RED (SHRED) [38], CHOKe [21], GREEN [39], Stochastic Fair BLUE (SFB) [40], and BLACK [23].

A. *Random Early Detection (RED)*

RED is the first formal AQM that have been proposed to be deployed instead of Droptail queue algorithm in TCP/IP networks [48]–[50]. RED is a queue-based AQM with no per-flow information that provide the network with a congestion avoidance mechanism, and RED can be considered under heuristic design which uses statistical probability packet drop when queue length reached a specific threshold value [44]–[46]. Besides of congestion avoidance and control, RED has been designed to achieve fairness among different bursty flows [47], [48], minimize queueing delay by controlling the queue lengths in low values [56], preventing the interconnection between global synchronization and packet dropping [57], [58], reduce the packet loss, and achieve high link utilization [51]. RED will be discussed and explained precisely due to its importance in this research study because it was the foundation to design many newer AQM schemes and was the most studied algorithms in the AQM researches so far as stated in [60].

RED uses Exponential Weighted Moving Average (EWMA) [54]–[57] for the queue length as congestion indicator and calculating congestion level at the queue. This average will be updated every packet arrival and estimated the actual queue length [46] and it can be calculated as:

$$\bar{q}(t_{i+1}) = (1 - w_q)\bar{q}(t_i) + w_q q(t_{i+1}) \tag{1}$$

where $q(t)$ is instantaneous queue length at time t, $\bar{q}(t)$ is average queue length at time t, w_q is EWMA queue weight, and t_i is arrival time of the i packet.

EWMA (wq) is a static parameter that have to be configured accurately. Beside that RED has three other parameters that used for RED control function which are: minimum threshold min_{th}, maximum threshold max_{th}, and maximum non-congestion probability P_{max}. If the average queue length is below the *minth* the RED will work normally without any changes, but if it increased between min_{th} and max_{th} the RED starts to drop incoming packets strained by proportional probability function to reduce average queue length. When the average queue length increase above the max_{th}, RED will drop all incoming packets without any exception as shown in Figure 1 [57]–[59]. The RED control function can be expressed mathematically as:

$$p(\bar{q}) = \begin{cases} 0 & 0 \le \bar{q} \le min_{th} \\ \frac{\bar{q} - min_{th}}{max_{th} - min_{th}} P_{max} & min_{th} \le \bar{q} \le max_{th} \\ 1 & \bar{q} \ge max_{th} \end{cases} \tag{2}$$

where p is packet dropping probability and p_{max} is maximum un-congestion probability of dropping at the max_{th}.

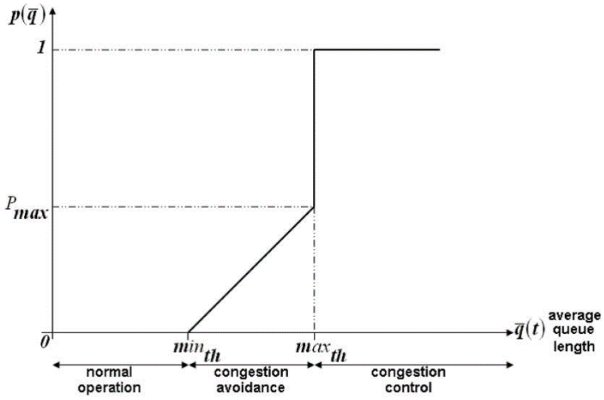


Figure 1: RED Control Function

RED, besides its importance, has many drawbacks. One of the major problems that RED suffer from is the parameter tuning [67], [68]. Whereas, RED have four important parameters which would be very sensitive depending on network conditions. Therefore, a set of parameters values might work perfectly with certain network load and delay but imperfect for a next load and delay which is not desirable due to Internet rapidly changing characteristic [69], [70]. Another huge problem with RED is using queue length as a performance measure and congestion indicator at the same time, this coupling has big deterioration in RED performance with increasing of traffic load in term of throughput and delay [71]. According to [38], when the number of flows increase, the marking probability should increase, but in RED case this means the queue length should be increased as well however it's still same at fixed value regardless increasing of the flows which will leads to instability. Thus, RED does not differentiate between different types of flows or TCP flows with different RTT which will penalize the stability as well. As mentioned in [66], the packet dropping probability in RED does not guarantee fair bandwidth share among the flows because RED will drop all incoming packets in same probability regardless of the number or type of flows, that will cause a higher packet loss for high sending rate flows as same as short-live flows even it does not reach its fair share of the bandwidth.

B. *Fair RED (FRED)*

FRED can easily define as RED with per-flow information state. This algorithm has been developed to overcome RED unfairness problem [73]. FRED has added two parameters which are *minq* and *maxq* to represent minimum and maximum packets that allowed to enter the queue for each flow. If the flow's buffer occupancy (represented by *qlen*) less than *minq* and the average queue length below max_{th}, the flow will not suffer from any loss. A flow will experience packet drop when its buffer occupancy *qlen* exceeds *maxq* or the average queue length becomes greater than *maxth* which is same as RED dropping policy. *minq* adjusted dynamically with the global variable *avgcq* which is the average per-flow queue length and can be calculated by dividing the average queue length with number of flows. FRED counts how many times *maxq* has been exceeded and keep the value in parameter called strike for each flow. The flow is not allowed to exceed its *avgcq* when it has higher strike value comparing with other flows, thus penalizing unresponsive flows from using up most of the queue space [17], [68].

FRED suffers from several drawbacks. According to [38] and [69], FRED cannot guarantee fairness in most cases and scenarios, however it is fairer than the RED. On the other hand, FRED has limited number of flows because of the queue size is limited, and needs large buffer space to sufficiently interpret unresponsive flows. It should be noted that FRED is memoryless, so that unresponsive flows will be considered as responsive once its packets cleared from the buffer, limiting its stability.

C. Stochastic Fair BLUE (SFB)

SFB is an enhancement to the BLUE algorithm in terms of fairness [75], [76]. SFB is per-flow information algorithm that uses multi-level *L* hash table with *N* number of bins in each table. Each bin assigned to certain flow and counts how many times that flow has been hashed, associated with each bin is dropping or marking probability *Pm*. For each *L* table there are independent hash function which assigned a bin for each flow based on the flow ID (which contains its source address, destination address, source port, destination port, protocol). Each packet will be hashed on its arrival, that will increase its occupancy bin and that will increase the *Pm* associated with that bin, vice versa. The flow with high transmission rate would increase its *Pm* value from 0 to 1 in all of its bins of *L* tables. Thus, this flow will be considered as non-responsive flow and its rate should be penalized as shown in Figure 2.

The novelty behind SFB scheme is to protect responsive flows by penalizing non-responsive flows and limiting their rate with their fair share, it is differentiating between responsive and non-responsive flows by using bloom filter with multi-level hash function. The main disadvantage for SFB is misclassification, it has been approved by [70] that responsive flows may occupy some bins associated with non-responsive ones which will cause penalizing responsive flows as well. Thus, the concept of fairness and stability will be punished in the network. Nonetheless, since SFB congestion indicator mechanism independent with queue occupancy, some of non-responsive flows will be penalized even there is empty buffer space available.

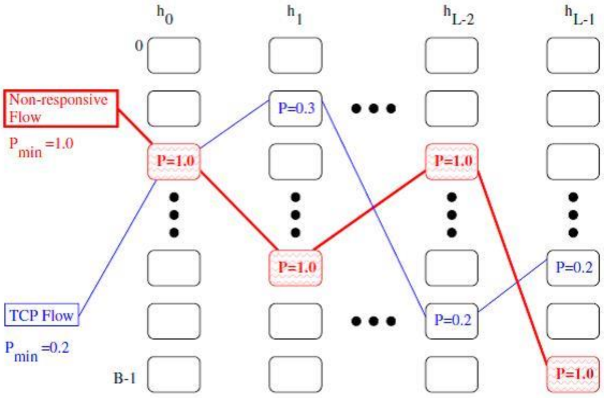


Figure 2: Multi-Level Hash Table in SFB

D. BLACK

The main idea behind BLACK (from BLACKlisting unresponsive flows) scheme is to control high bandwidth unresponsive flows with different types of flows to achieve fairness [23]. BLACK uses buffer occupancy fraction and bandwidth share as congestion indicator at the link. Based on FIFO queue concept, if the AQM scheme allocates the buffer among all the flows equally, fairness can be achieved at the link bandwidth. BLACK uses limited state information and sampling technique to estimate buffer occupancy fraction of only the large queue. To differentiate between flows, BLACK uses cache memory (HBF cache memory - from High Bandwidth Flows) to capture the flow ID for all the flows that occupied the queue. Upon each packet arrives to the queue, BLACK record this packet in the HBF cache memory. Whenever, the queue length exceeded the certain threshold, the packet will be randomly picked from the queue and compered its ID with the ID of the incoming packet. If these two packets have same ID then the ID will be recorded in HBF cache memory and change its "Hit" value to one (1). But if the ID not same then it will be recorded in the memory and keep sampling. After m sampling, the Hit Fraction for the flow, which is responsible for estimating the buffer occupancy of that flow, can be calculated by dividing Hit value with m. The flow with higher Hit Fraction will be considered as high bandwidth flow and it will be dropped according to probability function, which can be calculated as:

$$\widehat{p_\iota} = \frac{\overline{H_\iota} - \frac{1}{N_{act}}}{\left(\frac{1}{N_{act}}\right)} \tag{3}$$

where $\overline{H_\iota} = \frac{Hit_i + H_i m}{\acute{m} - m}$, H_i is hit fraction for i flow equal to $\frac{Hit_i}{m}$, Hit_i is the number of hits at the sampling time, m is the number of samples taken so far, N_{act} is the estimate of the number of active flows, and $\widehat{p_\iota}$ is the dropping probability.

This dropping probability will be scaled based on RED congestion avoidance as:

$$p_{final,i} = \widehat{p_\iota} \times \frac{\bar{q} - min_{th}}{max_{th} - min_{th}} \tag{4}$$

So that, the flows, that do not have any record in HBF cache memory, will be controlled totally by RED. When the high bandwidth flows will remain at HBF cache memory and penalized regarding how much greater than the fair share they occupied.

BLACK has been proved to perform fairly not only in high bandwidth unresponsive flows scenarios but also in scenarios with different TCP types and RTTs are competing in the link bandwidth [20]. The main drawback in the BLACK scheme is it's not accurate to estimate the number of flows in most cases such as when the queue size is not large compared with high bandwidth and delay applications, or when traffic transmission rate of the flows is very different. In addition, BLACK act like RED in some cases, as mentioned before, in that case BLACK will have same drawbacks as RED such as each packet arrival checking and probability function.

E. Adaptive Virtual Queue (AVQ)

AVQ [44] is a rate-based AQM that maintains arrival rate at a targeted utilization [77]–[79]. AVQ has been designed based on Kelly et al. [75] optimization approach by Kunniyur and Srikant in 2004 [44]. The probability function of AVQ has derived from M/M/1/B loss probability (see [76]) which is $\mathcal{P}_l(c_l, \mathcal{Y}_l) = \frac{(1-\rho)\rho^B}{1-\rho^{B+1}}$ where $\rho = \frac{\mathcal{Y}_l}{c_l}$ is link utilization, and B is the buffer limit. In this loss probability, B, link capacity (c_l), and arrival rate ($\mathcal{Y}_l$) are scaled by K factor and take the limit as $K \to \infty$, so the probability will be $\rho_l(c_l, \mathcal{Y}_l) = \frac{(\mathcal{Y}_l - c_l)^+}{\mathcal{Y}_l}$ where $[z]^+ = \max(0, z)$.

AVQ adapts a virtual queue with link capacity less than the physical link capacity that connected in the real queue [77]. The packet will be marked when the arrival rate to the queue exceeds the virtual capacity [76]. The actual number of the flows should be known by the queue in order to compute a virtual capacity and that will satisfy the probability function. This computation should occur when the network load is changing because the number of the flows is varying in time. However, the link capacity of virtual queue is calculated by differential equation to make the computation independent for each number of links, the differential equation is:

$$\tilde{C}(t) = \alpha\big(\gamma C - \mathcal{Y}(t)\big) \tag{5}$$

where $\tilde{C}$ is the link capacity in the virtual queue, C is the link capacity in the real queue, and $\mathcal{Y}(t)$ is the arrival rate to the system.

α and γ are AVQ parameters. γ is the desired link utilization whereas α is smoothing parameter [44] or step-size as mentioned in [76]. α considered as a key design issue for AVQ because it determines the adaption speed of the virtual link capacity, whereas γ represent how the system robust with presence of short-live flows [74]. As stated by [44], the stability of the system can be determined by both α and γ. It can be easily seen from the Eq (8) that AVQ tries to match the arrival rate with the link capacity of the virtual queue to achieve the desired utilization [66].

The size of real queue and virtual queue are same, and both queues will receive same arrival rate, but virtual queue will build up and overflow faster than real queue due to its parameters. Each time the virtual queue experience overflow, the same packet will be marked/dropped in the real queue [11]. This concept called deterministic which it is the opposite of probabilistic that used by RED [61]. In addition, it can be noted from Eq. (8) that when the virtual link capacity is greater than real link utilization, the marking probability will be so aggressive, vice versa [44].

The dropping probability for AQM, as mentioned in [66], will be as:

$$p(t) = \left[1 - \frac{\tilde{C}(t)}{\mathcal{Y}(t)}\right]^+ \tag{6}$$

By making γ value equal one (1) and apply it in Eq. (8) with initial sittings $\tilde{C}(0) = C$ and q(0) = 0 (q(t) is the queue length at time), as mentioned in [66] the differential equation will be as:

$$\tilde{C}(t) = C - \alpha q(t) \tag{7}$$

which is indicating that AVQ matches the virtual capacity with the queue length, so when the queue length increases, the virtual capacity will be increased.

AVQ has shown a significant performance especially in high link utilization, low packet loss, and low delay [11]. The virtualization technique has given AVQ a great performance in terms of robustness, responsiveness, and stability. In fact, AVQ has been performed in stability analysis by the designers [44], the result from this analysis was some recommended rules to control AVQ parameters (α and γ) according to the number of flows to preform efficiently. AVQ could be considered as fair scheme due to its control function that can maximize the aggregation of source utilities in the network in the absence of feedback delays. The only disadvantages in AVQ scheme is AVQ cannot differentiate unresponsive flows from responsive one because it was not designed to deal with responsive flows.

F. Random Early Marking (REM)

REM has been proposed by Athuraliay and Low [52]. REM is an AQM scheme that designed to attain high link utilization with negligible loss and delay in stable and simple manner. The main difference between REM and RED is the congestion measure and the probability function for dropping/marking. The key idea behind REM is achieving its targeted queue length for low delay and targeted rate for high utilization independently of network congestion [78] by separating the congestion measure from the performance measure [82], [85], and reaching the global optimal performance point [59].

The congestion measure that been used by REM, known as 'price', is the weighted sum of the mismatch between the arrival rate and queue length with the targeted ones (the difference between arrival rate and link capacity with the difference between the queue length with the targeted length). The price is updatable individually for each link, when the aggregation of this weighted mismatch is positive, the price will increase, and otherwise it will be decremented. Whenever the arrival rate exceeds the link capacity or the queue length is greater than the target, the weighted sum for the mismatch will be positive, and otherwise it is negative. The incrementing of the price will push up the marking probability, thus it will send a strong signal to the sources to reduce their rates. Reducing the arrival rates will push down the price and hence the marking probability until eventually the weighted sum for the mismatches will be zero, in this point REM will achieve the highest link utilization and minimum delay and loss. The REM price can be updated as the following equation, according to [50]:

$$\mu_l(t+1) = \left[\mu_l + \gamma\left(q_l(t+1) - (1-\alpha_l)q_l(t) - \alpha_l q_{ref}\right)\right]^{+} \qquad (8)$$

where μ_l is the price of link l, q_l is the actual queue length, q_{ref} is the targeted queue length, α_l is the stability constant, and $[z]^{+} = max(z, 0)$.

And the exponential packet marking probability for the l link will be formed as:

$$p_l(t) = 1 - \emptyset^{-\mu_l(t)} \qquad (9)$$

where p_l is the marking probability for the link l and $\emptyset$ is constant less than 1.

Thus, the end-to-end marking probability can be expressed, according to [81], as:

$$1 - \prod_{i=1}^{L}\left(1 - p_l(t)\right) = 1 - \emptyset^{-\sum_l \mu_l(t)} \qquad (10)$$

which means that end-to-end marking probability will increase when the individual prices increased. REM has been designed to perform optimally in steady-state situation, but not so efficient in transit situation [67], [80]. However, the designers run the stability test on REM and they found that, the prior knowledge about the network parameters, such as the number of flows and RTTs, could guarantee the stability for responsiveness satisfaction, but these parameters are frequently changed in real networks. REM has given a poor performance in the wireless environment from the experimental result that been done by the REM designers, and they claimed that the reason behind this poor performance due to a TCP sources cannot differentiate between the overflow loss or wireless effects loss, and reduce the transmission rate on both events.

G. *Controlled Delay (CoDel)*

CoDel is the latest AQM proposed by Nichols and Jacobson in 2012 [46]. It has been built and designed to solve a full buffer problem “bufferbloat” in network by limiting the packet queue delay that happened in the network links (routers). CoDel tries to enhance overall performance of the network by reducing the delay and packet loss with high link utilization and throughput. According to [46], CoDel has a significant characteristics that make it better than the rest of AQM, such as:

- Parameterless: no pre-configured parameters required.
- Treating good queue (the queue that drains as fast as possible) and bad queue (the queue that fills up as same as transmission rate) differently.
- Queue delay controlling regardless to the RTT delay and traffic load.
- Maintaining dynamically changing send rate.
- Simple to implement in real router.

CoDel can be considered as delay-based AQM because it uses packet-sojourn time instead of arrival rate or queue length in its congestion indicator. Packet-sojourn time is the time that the packet spends in the queue, which can be found by adding a timestamp to each arrival packet to the queue that contains arrival time for that packet. When the packet is about to leave the queue, the packet-sojourn time can be simply calculated by subtracting the leaving time with the time that recorded in the timestamp (arrival time) for each packet independently. If the sojourn time is bigger than a pre-defined target, the algorithm will set timer for dropping packet at dequeueing (leaving the queue). This dropping will happened only when the sojourn time is bigger than the target and the packets at the queue is less than one Maximum Transmission Unit’s (MTU’s) of bytes. The time that indicated the next dropping event will be update periodically according to $Next\_drop\_time \mathrel{+}= {}^{interval}/_{\sqrt{count}}$.

The count represents the total number of dropped packet since the first drop event. Whereas, interval is the minimum value of sliding window that entered the queue and CoDel algorithm has to experience that by time, because it is varying with the time, and update it frequently. The dropping action will be stopped when the sojourn time goes below the target value.

It should be noted that, CoDel has two important parameters, target and interval. These parameters are needed to be configured wisely to get better performance. However, these parameters have been given a fixed values which have been chosen based on many simulations and experimental results, as stated in [7] as follow:

- Target: constant 5ms (acceptable queue latency)
- Interval: constant 100ms (in worst case of RTTs)

CoDel has shown a better result among many proposed studies that compare it with the previous AQM schemes. In [82], the authors have compared CoDel with RED and Adaptive RED (ARED), and they concluded that CoDel is independent to queue size, rate measurements, drop rate, and RTT delays, and they showed that CoDel has better performance in link utilization, queue length, and drop rate, but they suggested that CoDel needs more optimization and improvement to increase its robustness. According to [83], CoDel has a better performance than Droptail and RED algorithms in terms of queue delay, link utilization, and packet drop. But, it has been concluded that CoDel has higher packet loss than RED when increasing the number of flows. This issue is not acceptable in terms of network stability, CoDel needs to be improved to control the stability when increasing the number of flows. In terms of fairness, it has been claimed that CoDel has more fair than few of the RED, but it needs to be enhanced by combining CoDel algorithm with scheduling algorithm (such as Fair Queue) as suggested by [84].

IV. PERFORMANCE EVALUATION

As mentioned earlier, this paper has conducted a simulation approach as a method to study and evaluate the performance of various AQM schemes. The simulation has conducted in NS-2 version 2.35 on Linux operating system. Furthermore, the experiment scenario will be executed on bottleneck topology (dumbbell) which had been studied and approved as suitable topology for testing and evaluating the performance of different types of AQM [6]. The AQM schemes has been tested under a single wired link that contains one router node in the middle of two Access Points (AP) which will be connected wirelessly to 20 sources and one destination in each side as shown in the Figure 3. Two different type of flows will be applied on the sources. First, 15 FTP flows transferred over TCP-Reno links with 1000 bytes packet size. Second, 5 UDP (transmission rate 180 Kbps) flows will be implemented on CBR packets. This scenario has run for 100 seconds time long; the rest of parameters and AQM schemes configuration settings can be found in Table 1 and 2 respectively.

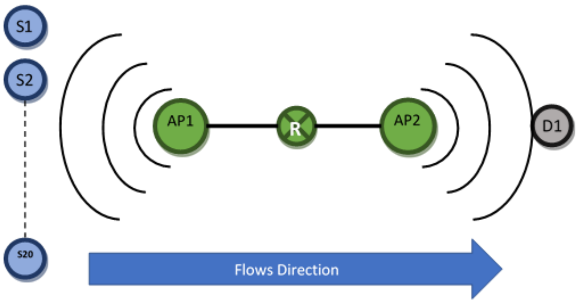


Figure 3: Single Link Bottleneck (Dumbbell) Topology

Four AQM schemes under evaluation are: RED, AVQ, REM, and CoDel. RED will be chosen in the evaluation because of two reasons. Firstly, as mentioned earlier, RED is the first formal AQM. Secondly, RED have been implemented in some of the real devices now days (such as: Alcatel-Luncent OmniAccess 5510/5740 unified services gateway; Brocade MLX Series – Multiservice IP/MPLS Routers; Cisco ASR 9000 System Aggregation Services Routers; HP MSR50 Series: and Juniper Networks M7i/ M10i Multiservice Edge Routers) [6]. Optimization approach has proved its ability to enhance fairness in many different studies (such as [44], [63], [85], [86]). Due to that, AVQ and REM have been chosen in evaluation stage. CoDel is the latest proposed AQM. It has shown significant results in terms of fairness and stability in wireless environment. Besides, CoDel is the first AQM scheme that uses queue-delay as congestion indicator.

V. SIMULATION RESULTS

After running the simulation, the results have collected from many tracing files and analyzed by specific AWK script that been used to analyze NS-2 results. The experimental results have divided in to multiple subsection based on the performance matrices to get a better understanding of each AQM schemes behavior.

Table 1
Simulation Parameters

Parameters	Value
Simulation Topology	Dumbbell
Flow Type	Responsive (FTP over TCP-Reno) Unresponsive (CBR rate 180 Kbps)
Packet Size	1000 Bytes
MAC Protocol	Ethernet 10 Mbps IEEE 802.11
Queue Size	100 Packets
Simulation Time	100 Seconds

Table 2
AQM Schemes Configuration Settings

QM Scheme	Parameter	Value
RED	max_{th}	80
	min_{th}	20
	max_p	0.1
	w_q	0.002
AVQ	γ	0.98
	α	0.8
REM	γ	0.001
	α	0.1
	Φ	1.001
CoDel	*target*	0.005
	interval	0.1

Queue Size - the average of the packet that occupied by the aggregations of the flows, which consider as a direct indicator of router resource utilization. Minimizing the queue size is one of the key issues for designing AQM schemes by which can affect overall network resources and can also show the different characteristics of different schemes. In this paper, queue size has been monitored and recorded every second by measuring the number of packets in the buffer. Figure 4 shows the result of queue size for RED, REM, AVQ, and CoDel schemes. from the figure we can see that RED and CoDel has a better result than REM and AVQ due to the parameters characteristics of RED and CoDel which gives the algorithm the ability to keep the queue size as low as possible, its differs from AVQ that considered as adaptive algorithm that can maintain the queue size to suits the virtual queue and maintain the congestion indicator regularly. On the other hands, RED have max_{th} and min_{th} which can affect directly on the queue size.

Queue Lost: is the packets that dropped in the queue mostly because of the congestion collapse or algorithmic reasons. Queue lost is very important matric to study the reaction of AQM scheme to the near overflow situation and congestion indication. The queue lost has been monitored and calculated by number of packets every second. From Figure 5, it can be concluded that REM and AVQ has better reaction in terms of congestion indication because they have low queue loss and that will increase the throughput and outgoing link utilization due to its optimization mathematical design. However, CoDel and RED have a fixed parameter that controls the amount of packets in the buffer which lead to drop down the rest of the packets.

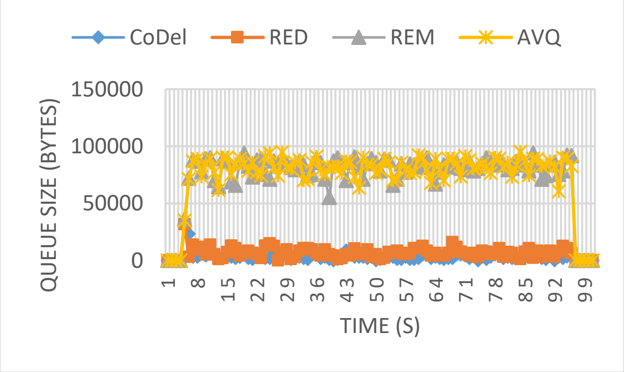


Figure 4: Queue Size for AQM Schemes

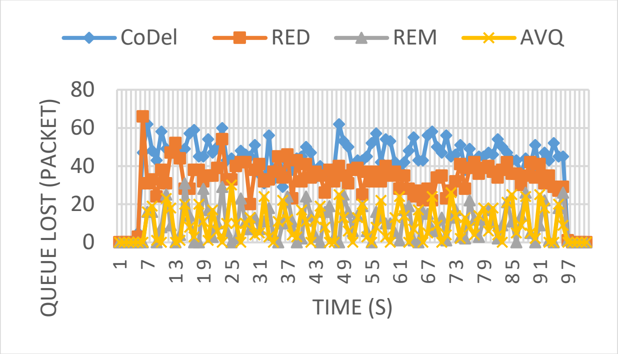


Figure 5: Queue Packet Loss for AQM Schemes

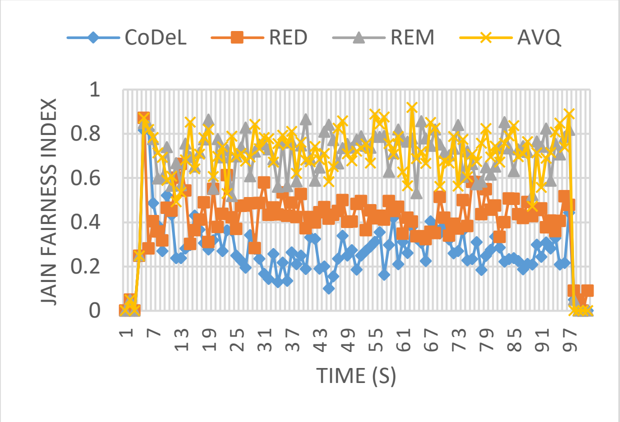
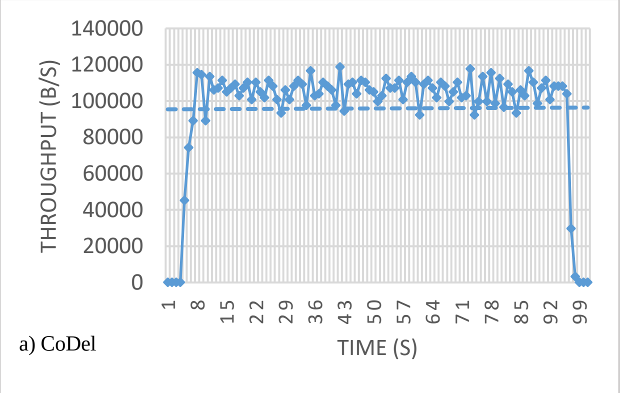
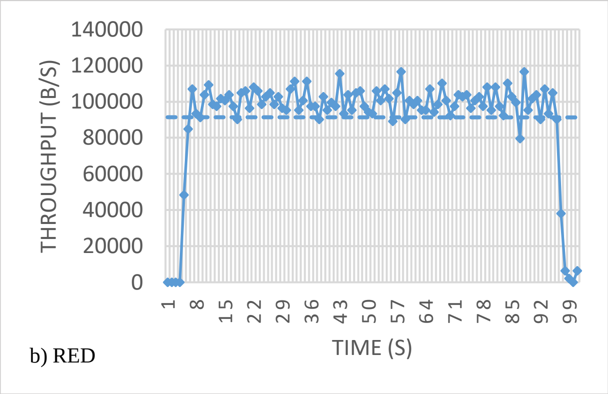


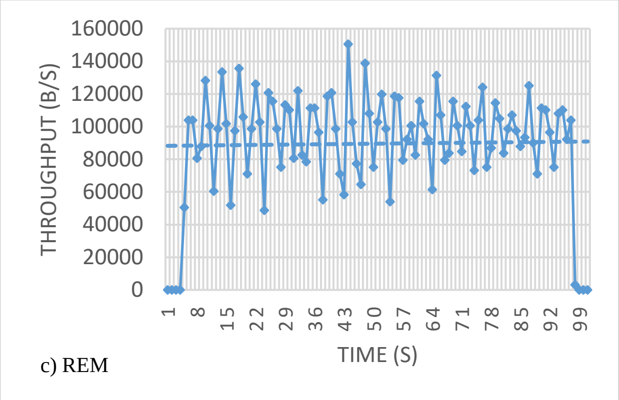
Figure 6: Jain Fairness Index for AQM Schemes



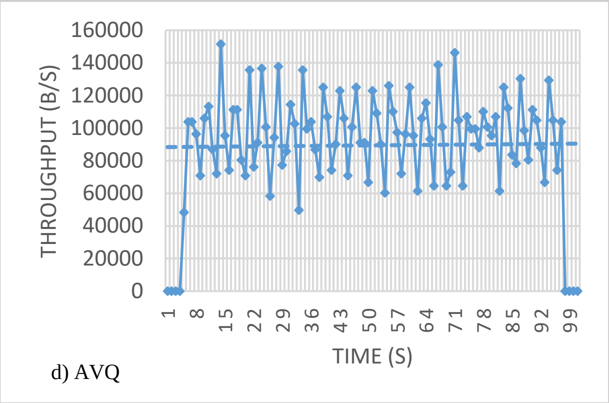
a) CoDel



b) RED

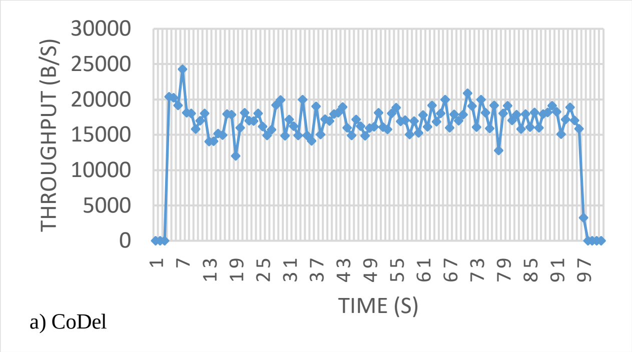


c) REM

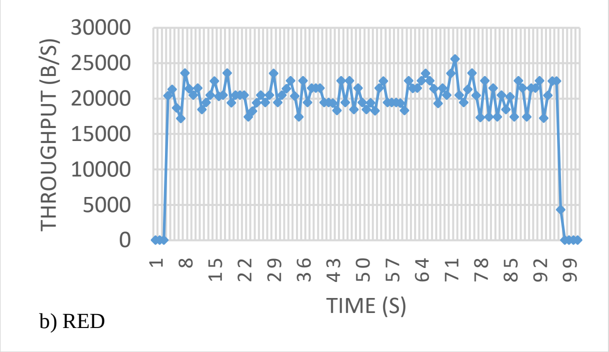


d) AVQ

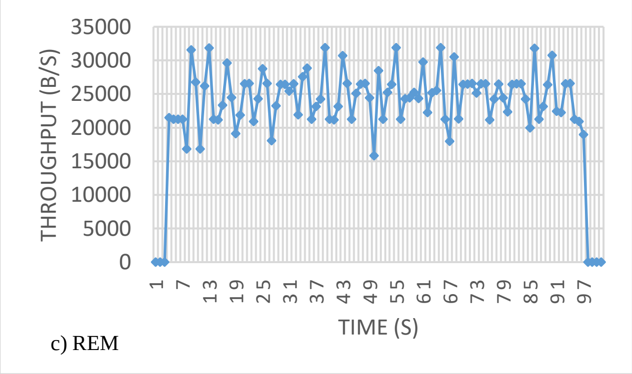
Figure 7: TCP Throughput for AQM Schemes



a) CoDel

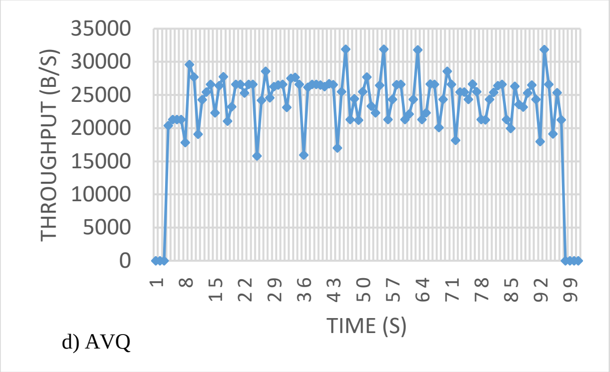


b) RED



c) REM

]



d) AVQ

Figure 8: UDP Throughput for AQM Schemes

Throughput: is the total number of received bytes by the destination in the time unit (i.e., megabytes per second). It is approved that any AQM algorithm should have significant increasing of the throughput at the end node. From the perspective of this paper, the throughput has been measured into two parts depending on the flow type (TCP and UDP), the aggregation of each throughput can be seen in Figure 7. Moreover, all algorithms have nearly the same amount of average throughput of TCP flows, but it can be differentiating AVQ and REM have slightly higher throughput than RED and CoDel because AVQ and REM have lower queue packet loss than the rest.

However, in terms of UDP flows throughput, from the Figure 8, it's a bit higher than TCP flows which is normal due to the TCP slow start characteristic and congestion control mechanism. Whereas, it's also clear that AVQ and REM have a better throughput than RED and CoDel.

Fairness: this paper has computed Jain's fairness index as a measure of fairness among the individual bandwidth share in the aggregate. Jain's fairness index is the mathematical formulation to measure the fairness among number of links by calculating the received throughput for each link, it is originated by Raj Jain in 1984 [116] and formulated as $\mathcal{J}(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot (\sum_{i=1}^{n} x_i^2)}$. If all the users got same throughput then the fairness index is 1 and the system is totally fair. Figure 6 presents the fairness values obtained for the scenario. It can be observed that REM and AVQ both obtain higher fairness value than RED and Codel. However, CoDel has given the lowest value comparing with RED scheme which have given a moderated value in terms of fairness.

From the above result, we can conclude that AVQ and REM have better adaptation in WLANs environment due to the congestion indicator mechanisms for AVQ and REM have considered the aggregation of transmission rate among all flows with the ability to penalize the unresponsive flows. Thus, AVQ and REM tries to equalize all flows transmission rate together with capability of congestion indicator mechanism. Unlike RED and CoDel, both have used queue size and queue delay respectively in their congestion indicator mechanism and that is not enough to equalize the throughput of the flows which lead to unequal bandwidth share in the outgoing link.

VI. CONCLUSION AND FUTURE WORK

In this paper, we provide a performance evaluation of the fairness for different fair AQM schemes under the presence of responsive TCP and unresponsive UDP flows. The results show the better performance of AVQ and REM in terms of fairness even when they have a higher queue size. On the other hand, CoDel has a lower queue size comparing with the rest of the schemes.

Based on the results of the simulation model, it can be concluded that AQM with rate-base congestion indicator can achieve higher level of fairness than delay-base and queue-base. Besides, rate based congestion indicator have better throughput and link utilization than the rest. In order to maintain high level of fairness in the network, AQM schemes needs to be implemented with high queue size. This queue size helps AQM to take a better decision to achieve fairness among different types of flows. However, AQM schemes have different behavior in WLANs environment than Wired Network environment due to the transit, dynamic and rapidly changing of the wireless nodes. Therefore, the fairness among wireless users having different numbers and directions of TCP and UDP flows can be assured even with diversity of transmission rate.

Moreover, fairness is one of early goals of AQM schemes and yet there are no schemes have considered as achieved a total fair situation. We highly recommended researchers to do more investigation and design new schemes since it is still a rich area to dig on. Moreover, the hybridized congestion indicators have given a better result than the others we recommended designers to hybridize between transmission rate and other parameter that increase fairness in the network such as queue lost or queue delay. Besides, a standardized model for fairness evaluation in AQM schemes in WLAN environment has not designed yet, therefore we highly recommended researchers to do more investigation in this part.

REFERENCES

[1] G. Abbas, Z. Halim, and Z. H. Abbas, "Fairness-driven queue management: A survey and taxonomy," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 324–367, 2016.

[2] J. Nagle, "Congestion control in IP/TCP internetworks," *RFC 896*, 1984. [Online]. Available: http://www.ietf.org/rfc/ rfc896.txt.

[3] S. Floyd and K. Fall, "Router mechanisms to support end-to-end congestion control," 1997. [Online]. Available: http: http://www-nrg.ee.lbl.gov/floyd/papers.htm.

[4] M. Baklizi, H. Abdel-Jaber, A. A. Abu-Shareha, M. M. Abualhaj, and S. Ramadass, "Fuzzy logic controller of gentle random early detection based on average queue length and delay rate," *Int. J. Fuzzy Syst.*, vol. 16, no. 1, pp. 9–19, 2014.

[5] J. P. Yang, "Rotating preference queues: An efficient queue management scheme for fair bandwidth allocation," *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 420–423, 2013.

[6] R. Adams, "Active Queue Management: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3Adams, R. (2013). Active Queue Management: A Survey. IEEE Communications Surveys & Tutorials, 15(3), 1425–1476. doi:10.1109/SURV.2012.082212.00018, pp. 1425–1476, 2013.

[7] T. Jain, M. P. Tahiliani, and others, "Performance Evaluation of CoDel for Active Queue Management in Wired-Cum-Wireless Networks," in *2014 Fourth International Conference on Advanced Computing & Communication Technologies (ACCT)*, 2014, pp. 381–385.

[8] I. Jarvinen and M. Kojo, "Evaluating CoDel , PIE , and HRED AQM Techniques with Load Transients," in *39th Annual IEEE Conference on Local Computer Networks*, 2014, pp. 159–167.

[9] I. M. T. T. Bretagne, N. Kuhn, E. Lochin, and O. Mehani, "Revisiting Old Friends : Is CoDel Really Achieving What RED Cannot ?," vol. 2014, no. August, 2014.

[10] S. Banerji and R. S. Chowdhury, "Recent developments in IEEE 802.11: WLAN Technology," *Int. J. Mechatronics, Electr. Comput. Technol.*, vol. 3, no. 9, pp. 1001–1013, 2013.

[11] R. Seungwan, R. Christopher, and Q. Chunming, "Advances in Active

Queue Management (AQM) Based," *Telecommun. Syst.*, vol. 25, no. 3–4, pp. 317–351, 2004.

[12] V. Kushwaha and R. Shwer, "A Review of Router based Congestion Control Algorithms," *Int. J. Comput. Netw. Inf. Secur.*, vol. 6, no. 1, pp. 1–10, Nov. 2013.

[13] J. Koo, S. Ahn, and J. Chung, "Performance analysis of active queue management schemes for ip network," in *International Conference on Computational Science*, 2004, pp. 349–356.

[14] F. Li *et al.*, "A comparative simulation study of TCP/AQM systems for evaluating the potential of neuron-based AQM schemes," *J. Netw. Comput. Appl.*, vol. 41, pp. 274–299, 2014.

[15] G. Hasegawa and M. Murata, "Survey on fairness issues in TCP congestion control mechanisms," *IEICE Trans. Commun.*, vol. 84, no. 6, pp. 1461–1472, 2001.

[16] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *ACM SIGCOMM Computer Communication Review*, 1989, vol. 19, no. 4, pp. 1–12.

[17] D. Lin and R. Morris, "Dynamics of random early detection," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 4, pp. 127–137, Oct. 1997.

[18] T. J. Ott, T. V Lakshman, and L. H. Wong, "Sred: stabilized red," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1999, vol. 3, pp. 1346–1355.

[19] G. Chatranon, M. Labrador, and S. Banerjee, "A survey of TCP-friendly router-based AQM schemes," *Comput. Commun.*, vol. 27, no. 15, pp. 1424–1440, Sep. 2004.

[20] G. Chatranon, M. Labrador, and S. Banerjee, "Fairness of AQM schemes for TCP-friendly traffic," in *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, 2004, vol. 2, pp. 725–731.

[21] R. Pan, B. Prabhakar, and K. Psounis, "CHOKe-a stateless active queue management scheme for approximating fair bandwidth allocation," in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2000. IEEE*, 2000, vol. 2, pp. 942–951.

[22] M.-K. Chan and M. Hamdi, "An active queue management scheme based on a capture-recapture model," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 4, pp. 572–583, 2003.

[23] G. Chatranon, M. A. Labrador, and S. Banerjee, "BLACK: detection and preferential dropping of high bandwidth unresponsive flows," in *IEEE International Conference on Communications, ICC'03.*, 2003, vol. 1, pp. 664–668.

[24] H. Nski, "On the optimization of the inter-flow fairness in the Internet," *Theor. Appl. Informatics*, vol. 24, no. 3, pp. 205–225, 2012.

[25] A. Domanski, J. Domanska, and J. Klamka, "Analysis of CHOKe-family active queue management," *Theor. Appl. Informatics*, vol. 25, no. 1, p. 49, 2013.

[26] B. Braden *et al.*, "Recommendations on queue management and congestion avoidance in the Internet," 1998.

[27] F. Baker and G. Fairhurst, "IETF Recommendations Regarding Active Queue Management," 2015.

[28] S. Akhtar, A. Francini, D. Robinson, and R. Sharpe, "Interaction of AQM schemes and adaptive streaming with Internet traffic on access networks," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, 2014, pp. 1145–1151.

[29] F. Abraq and M. Mikki, "Enhanced RED Protocol Using Multipath Routing," The Islamic University - Gaza, 2015.

[30] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.

[31] D. Bauso, L. Giarré, and G. Neglia, "About the stability of active queue management mechanisms," in *Proceedings of the American Control Conference.*, 2004, vol. 4, pp. 2954–2959.

[32] T. Eguchi, H. Ohsaki, and M. Murata, "On control parameters tuning for active queue management mechanisms using multivariate analysis," in *Symposium on Applications and the Internet, 2003. Proceedings. 2003*, 2003, pp. 120–127.

[33] C. Zhu, O. W. W. Yang, J. Aweya, M. Ouellette, and D. Y. Montuno, "A Comparison of Active Queue Management Algorithms Using the OPNET Modeler," *IEEE Commun. Mag.*, vol. 40, no. 6, pp. 158–167, 2002.

[34] F. Ren, C. Lin, and B. Wei, "Design a robust controller for active queue management in large delay networks," in *Proceedings. ISCC 2004, Ninth International Symposium on Computers and Communications, 2004.*, 2004, vol. 2, pp. 748–754.

[35] Y. Chait, C. V Hollot, V. Misra, S. Oldak, D. Towsley, and W.-B. Gong, "Fixed and adaptive model-based controllers for active queue management," in *Proceedings of the 2001 American Control Conference, 2001.*, 2001, vol. 4, pp. 2981–2986.

[36] R. Pletka, A. Kind, M. Waldvogel, and S. Mannal, "Closed-loop congestion control for mixed responsive and non-responsive traffic," in *IEEE Global Telecommunications Conference. GLOBECOM'03.*, 2003, vol. 7, pp. 4180–4185.

[37] B. Suter, T. V. Lakshman, D. Stiliadis, and A. K. Choudhury, "Design considerations for supporting TCP with per-flow queueing," in *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1998, vol. 1, pp. 299–306.

[38] M. Claypool, R. Kinicki, and M. Hartling, "Active queue management for web traffic," in *IEEE International Conference on Performance, Computing, and Communications*, 2004, pp. 531–538.

[39] W. Feng, A. Kapadia, and S. Thulasidasan, "GREEN: proactive queue management over a best-effort network," in *IEEE Global Telecommunications Conference, GLOBECOM'02.*, 2002, vol. 2, pp. 1774–1778.

[40] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "Stochastic fair blue: A queue management algorithm for enforcing fairness," in *Proceedings. IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2001*, 2001, vol. 3, pp. 1520–1529.

[41] R. Zhu, H. Teng, and J. Fu, "A predictive PID controller for AQM router supporting TCP with ECN," in *Proceedings. The 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications, 2004 and the 5th International Symposium on Multi-Dimensional Mobile Communications*, 2003, vol. 1, pp. 356–360.

[42] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *ACM SIGCOMM Computer Communication Review*, 2000, vol. 30, no. 4, pp. 151–160.

[43] C. V Hollot, V. Misra, D. Towsley, and W.-B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings.Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2001. IEEE*, 2001, vol. 3, pp. 1726–1734.

[44] S. S. Kunniyur and R. Srikant, "An Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 286–299, Apr. 2004.

[45] T. Ziegler, "On averaging for active queue management congestion avoidance," in *IEEE Symposium on Computers and Communications (ISCC)*, 2002, p. 867.

[46] K. Nichols and V. Jacobson, "Controlling queue delay," *Commun. ACM*, vol. 55, no. 7, pp. 42–50, 2012.

[47] S. De Cnodder, K. Pauwels, and O. Elloumi, "A rate based RED mechanism," in *The 10th International Workshop on Network and Operating System Support for Digital Audio and Video, Chapel Hill, North Carolina*, 2000, pp. 1–9.

[48] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED's active queue management," *IOSR J. Comput. Eng.*, vol. 2, no. 6, pp. 20–24, 2001.

[49] C. V. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A control theoretic analysis of RED," in *Proceedings. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2001.*, 2001, vol. 3, pp. 1510–1519.

[50] M. Kisimoto, H. Ohsaki, and M. Murata, "On transient behavior analysis of random early detection gateway using a control theoretic approach," in *Proceedings of the 2002 International Conference on Control Applications*, 2002, vol. 2, pp. 1144–1146.

[51] Y. Jiang, M. Hamdi, and J. Liu, "Self adjustable CHOKe: an active queue management algorithm for congestion control and fair bandwidth allocation," in *Proceedings. Eighth IEEE International Symposium on Computers and Communication, 2003. (ISCC 2003).*, 2003, pp. 1018–1025.

[52] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: active queue management," *Network, IEEE*, vol. 15, no. 3, pp. 48–53, 2001.

[53] S. H. Low, S. Member, and D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 1–16, 1999.

[54] J. Chen, C. Hu, and Z. Ji, "Self-tuning random early detection algorithm to improve performance of network transmission," *Math. Probl. Eng.*, vol. 2011, 2010.

[55] W. Feng, "Improving Internet congestion control and queue management algorithms," IBM Research, 1999.

[56] G. F. Ahammed and R. Banu, "Analyzing the performance of Active Queue Management Algorithms," *Int. J. Comput. Networks Commun.*, vol. 2, no. 2, pp. 1–19, 2010.

[57] H. Chandra, A. Agarwal, and T. Velmurugan, "Analysis of active queue management algorithms & their implementation for TCP/IP networks using OPNET simulation tool," *Int. J. Comput. Appl.*, vol. 6, no. 11, 2010.

[58] J. Sun, K.-T. Ko, G. Chen, S. Chan, and M. Zukerman, "PD-RED: to improve the performance of RED," *IEEE Commun. Lett.*, vol. 7, no. 8, pp. 406–408, 2003.

[59] M. Kwon and S. Fahmy, "Comparison of load-based and queue-based active queue management algorithms," in *ITCom 2002: The Convergence of Information Technologies and Communications*, 2002, pp. 35–46.

[60] B. Zheng and M. Atiquzzaman, "DSRED: improving performance of active queue management over heterogeneous networks," in *IEEE International Conference on Communications, 2001. ICC 2001.*, 2001, vol. 8, pp. 2375–2379.

[61] C. Wang, B. Li, Y. Thomas Hou, K. Sohraby, and Y. Lin, "LRED: a robust active queue management scheme based on packet loss ratio," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2004, vol. 1.

[62] M. Arpaci and J. A. Copeland, "An adaptive queue management method for congestion avoidance in TCP/IP networks," in *Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE*, 2000, vol. 1, pp. 309–315.

[63] G. Thiruchelvi and J. Raja, "A survey on active queue management mechanisms," *Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 12, pp. 130–145, 2008.

[64] T. Alemu and A. Jean-Marie, "Dynamic configuration of RED parameters," in *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, 2004, vol. 3, pp. 1600–1604.

[65] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, vol. 3, pp. 1435–1444.

[66] E.-C. Park, H. Lim, K.-J. Park, and C.-H. Choi, "Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks," *Comput. Networks*, vol. 44, no. 1, pp. 17–41, Jan. 2004.

[67] M. K. Agarwal, R. Gupta, and V. Kargaonkar, "Link utilization based AQM and its performance," in *IEEE Global Telecommunications Conference, GLOBECOM '04. 2004.*, 2004, vol. 2, pp. 713–718.

[68] A. Kamra *et al.*, "SFED: a rate control based active queue management discipline," in *IBM India Research Laboratory Research Report*, 2000.

[69] F. M. Anjum and L. Tassiulas, "Fair bandwidth sharing among adaptive and non-adaptive flows in the Internet," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1999, vol. 3, pp. 1412–1420.

[70] W. Feng, K. G. Shin, D. D. Kandlur, and D. Saha, "The BLUE active queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 513–528, 2002.

[71] C. Long, B. Zhao, X. Guan, and J. Yang, "The Yellow active queue management algorithm," *Comput. Networks*, vol. 47, no. 4, pp. 525–550, Mar. 2005.

[72] S. S. Oruganti and M. Devetsikiotis, "Analyzing robust active queue management schemes: a comparative study of predictors and controllers," in *IEEE International Conference on Communications, 2003. ICC '03.*, 2003, vol. 3, pp. 1531–1536.

[73] X. Deng, S. Yi, G. Kesidis, and C. R. Das, "Stabilized virtual buffer (SVB) - an active queue management scheme for Internet quality-of-service," in *Global Telecommunications Conference, 2002. GLOBECOM . IEEE*, 2002, vol. 2, pp. 1628–1632.

[74] S. S. Kunniyur and R. Srikant, "Stable, scalable, fair congestion control and AQM schemes that achieve high utilization in the internet," *IEEE Trans. Autom. Control.*, vol. 48, no. 11, pp. 2024–2028, 2003.

[75] F. Kelly, "Mathematical modelling of the Internet," in *The Fourth International Congress on Industrial and Applied Mathematics*, 1999, vol. 33, no. July, pp. 1–20.

[76] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: Utility functions, random losses and ECN marks," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 689–702, 2003.

[77] G. Feng, A. K. Agarwal, A. Jayaraman, and C. K. Siew, "Modified RED gateways under bursty traffic," *IEEE Commun. Lett.*, vol. 8, no. 5, pp. 323–325, 2004.

[78] Y. Hong and O. W. W. Yang, "Design of TCP traffic controllers for AQM routers based on phase margin specification," in *Workshop on High Performance Switching and Routing, HPSR.*, 2004, pp. 314–318.

[79] T. Wu, H. Xu, and S. Tian, "End to end congestion control and active queue management," in *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, 2003, no. November, pp. 2–5.

[80] L. Le, J. Aikat, K. Jeffay, and F. D. Smith, "The effects of active queue management on web performance," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, 2003, p. 265.

[81] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, 1999.

[82] D. M. Raghuvanshi, B. Annappa, and M. P. Tahiliani, "On the effectiveness of CoDel for active queue management," in *2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT)*, 2013, pp. 107–114.

[83] T. Sharma, "Controlling Queue Delay (CoDel) to counter the Bufferbloat Problem in Internet," *Int. J. Curr. Eng. Technol.*, vol. 4, no. 3, pp. 2210–2215, 2014.

[84] N. Khademi, D. Ros, and M. Welzl, "The New AQM Kids on the Block : Much Ado About Nothing ?," *Tech. Rep. 434*, pp. 1–24, 2013.

[85] P. Mrozowski and A. Chydzinski, "On the stability of AQM algorithms," in *Proc. of Applied Computing Conference*, 2009, pp. 276–281.

[86] S. Athuraliya and S. Low, "Simulation Comparison of RED and REM," in *Proceedings. IEEE International Conference on Networks, 2000.(ICON 2000).*, 2000, no. 2, pp. 68–72.