

STUDY AND IMPROVE NETWORK PERFORMANCE USING RANDOM EARLY DETECTION (RED) BY COMBINING FINE-TUNING THE LOWER THRESHOLD AND AVERAGE QUEUE SIZE

Vu Van Dien

TNU - University of Information and Communication Technology

ARTICLE INFO		ABSTRACT
Received:	12/7/2022	Internet is expected to better support many applications such as multimedia applications with limited bandwidth, low delay and packet loss rate requirements. Therefore, there is a need for new mechanisms to control congestion in the network. Active queue management (AQM) algorithms play an important role in ensuring network stability. RED (Random Early Detection) is the first dynamic queue management technique implemented in TCP/IP networks for congestion avoidance control. RED is based on comparing the average queue length with lower and upper thresholds to mark or discard packets. RED strategies have been introduced with the adjustment of parameters of threshold, average queue size to obtain better network performance. In this paper, the author proposes an improved RED algorithm called TqRED (Threshold queue RED) to address the limitation of RED. Through the evaluation simulation on the NS2 simulator, the author found that TqRED gives better results than RED in terms of packet loss, average queue delay, and average throughput.
Revised:	05/8/2022	
Published:	05/8/2022	
KEYWORDS		
Active queue management		
Congestion		
Average Queue Size		
RED		
Lower Threshold		

NGHIÊN CỨU, NÂNG CAO HIỆU NĂNG MẠNG SỬ DỤNG PHÁT HIỆN SỚM NGẪU NHIÊN (RED) BẰNG CÁCH KẾT HỢP TÍNH CHỈNH NGUỒN DƯỚI VÀ KÍCH THƯỚC HÀNG ĐỢI TRUNG BÌNH

Vũ Văn Diện

Trường Đại học Công nghệ thông tin và Truyền thông – ĐH Thái Nguyên

THÔNG TIN BÀI BÁO		TÓM TẮT
Ngày nhận bài:	12/7/2022	Internet được mong đợi sẽ hỗ trợ nhiều ứng dụng tốt hơn chẳng hạn như ứng dụng đa phương tiện với băng thông giới hạn, yêu cầu độ trễ và tỉ lệ mất gói thấp. Do đó, yêu cầu đặt ra là cần có các cơ chế mới để kiểm soát sự tắc nghẽn trong mạng. Các thuật toán quản lý hàng đợi động (AQM-Active queue management) đóng vai trò quan trọng để đảm bảo sự ổn định của mạng. RED (Random Early Detection) là kỹ thuật quản lý hàng đợi động đầu tiên được triển khai trong các mạng TCP/IP để điều khiển tránh tắc nghẽn. RED dựa trên việc so sánh chiều dài trung bình hàng đợi với ngưỡng dưới và ngưỡng trên để đánh dấu hoặc loại bỏ gói tin. Các chiến lược RED đã được đưa ra với sự hiệu chỉnh các tham số về ngưỡng, kích thước hàng đợi trung bình để thu được hiệu năng mạng tốt hơn. Trong bài báo này, tác giả đề xuất một thuật toán RED cải tiến có tên gọi là TqRED (Threshold queue RED) để giải quyết hạn chế của RED. Qua mô phỏng đánh giá trên bộ mô phỏng NS2, tác giả đã thấy được TqRED cho kết quả tốt hơn RED xét về tỉ lệ mất gói, độ trễ hàng đợi trung bình và thông lượng trung bình.
Ngày hoàn thiện:	05/8/2022	
Ngày đăng:	05/8/2022	
TỪ KHÓA		
Quản lý hàng đợi động		
Tắc nghẽn		
Kích thước hàng đợi trung bình		
RED		
Ngưỡng dưới		

DOI: <https://doi.org/10.34238/tnu-jst.6253>

Email: vvdien@ictu.edu.vn

<http://jst.tnu.edu.vn>

144

Email: jst@tnu.edu.vn

1. Giới thiệu

Internet được xây dựng dựa trên IP để cung cấp dịch vụ vận chuyển dữ liệu cho người dùng đầu cuối sử dụng giao thức TCP hoặc UDP. Internet đã phát triển rất nhanh chóng trong những năm qua, kéo theo sự tăng lên về tắc nghẽn xảy ra ở trong mạng. Khi tắc nghẽn xảy ra thì hiệu năng mạng giảm đi. Tắc nghẽn mạng được phát hiện khi bộ nhớ đệm đầy và các gói tin đến sau sẽ bị loại bỏ (drop), thông lượng mạng giảm đi. Tắc nghẽn mạng là vấn đề chính ảnh hưởng đến chất lượng dịch vụ (QoS) của mạng IP [1]. Số gói tin bị mất, trễ truyền và thông lượng trung bình là các vấn đề chính của mạng. Giảm tỉ lệ mất gói, trễ và tăng thông lượng trung bình là các mục tiêu quan trọng trong các kỹ thuật quản lý hàng đợi khác nhau để cải thiện chất lượng, dịch vụ mạng.

TCP là giao thức tầng giao vận được sử dụng phổ biến trên Internet. Đây là giao thức tin cậy, hướng kết nối. Nó cung cấp cơ chế ngăn ngừa và điều khiển tắc nghẽn trong mạng. Khi được sử dụng, TCP sử dụng một số kỹ thuật để đạt hiệu năng mạng cao và tránh tắc nghẽn [2], [3].

Để xử lý vấn đề tắc nghẽn trong mạng, đã có nhiều thuật toán được đề xuất như Drop Tail và các chiến lược hàng đợi động (AQM). Trong đó, Drop Tail xử lý gói tin ở hàng đợi theo nguyên tắc FIFO (First In First Out), khi hàng đợi bị đầy thì các gói tin đến sau sẽ bị mất. Các chiến lược hàng đợi động là các chiến lược được sử dụng để thay thế cho Drop Tail trên bộ định tuyến (Router). AQM cảm nhận tắc nghẽn có thể xảy ra đủ sớm và loại bỏ các gói tin dựa trên xác suất ngẫu nhiên, chứ không dựa vào hàng đợi đầy mới drop gói tin. Từ đó, Router thông báo cho nút nguồn điều chỉnh tốc độ phát thay vì đồng loạt giảm kích thước cửa sổ xuống, dẫn đến làm giảm số gói tin bị mất và tăng thông lượng trung bình trong mạng. Có nhiều chiến lược quản lý hàng đợi động đã được đề xuất từ RED, ARED, ERED,... cho đến DyRED. Chiến lược RED gốc được đề xuất bởi Floyd và Jacobson [4]. RED tránh tắc nghẽn sớm bằng cách sử dụng tham số avg cho biết kích thước hàng đợi trung bình. Tham số avg này được tính toán dựa trên trọng số hàng đợi w_q , kích thước hàng đợi hiện tại và avg trước đó. Sau đó, lấy avg để so sánh với ngưỡng dưới (*lower thresh*) và ngưỡng trên (*upper thresh*) để đưa ra quyết định xử lý gói tin. Nếu avg nhỏ hơn ngưỡng dưới thì không loại bỏ gói tin. Nếu avg nằm giữa ngưỡng dưới và ngưỡng trên thì gói tin sẽ bị loại bỏ theo xác suất được tính dựa trên avg , ngưỡng dưới và ngưỡng trên. Còn nếu avg lớn hơn ngưỡng trên thì loại bỏ gói tin.

Bằng việc phát hiện tắc nghẽn sớm, RED đã cho thấy ưu điểm của mình so với Drop Tail trong việc làm giảm số gói tin bị mất, giảm trễ truyền và tăng thông lượng trung bình. Tuy nhiên, khi lưu lượng mạng tăng đột biến thì RED lại tỏ ra không hiệu quả trong việc cải thiện hiệu năng mạng [5] - [9]. Trong nghiên cứu này, tác giả sẽ đưa ra phương pháp để cải thiện hiệu năng mạng nêu trên.

2. Phát hiện sớm ngẫu nhiên (RED)

Thuật toán RED [4] là chiến lược đầu tiên được đề xuất trong số các chiến lược quản lý hàng đợi động. Nó được sử dụng để điều khiển tránh tắc nghẽn dữ liệu tại các bộ định tuyến bằng cách kiểm tra độ dài trung bình hàng đợi với các gói dữ liệu đến và quyết định đánh dấu hoặc loại bỏ gói dữ liệu đến với xác suất tăng dần khi độ dài trung bình của hàng đợi vượt quá giá trị một ngưỡng xác định. RED có khả năng chống hiện tượng các nút nguồn đồng loạt giảm kích thước cửa sổ, duy trì khả năng đạt thông lượng cao qua hàng đợi RED, cũng như độ trễ thấp, cùng với việc đối xử công bằng giữa các kết nối TCP đi qua hàng đợi. Ý tưởng đằng sau của RED là, với mỗi gói tin đến bộ định tuyến, kích thước hàng đợi trung bình avg được tính toán sử dụng bộ lọc thông thấp trong các trường hợp hàng đợi trống và hàng đợi không trống. Sau đó, avg được tính toán đó đem so sánh với 2 ngưỡng (ngưỡng dưới min_{th} và ngưỡng trên max_{th}) trong bộ đệm của Router để quyết định khi nào loại bỏ gói tin. Các giá trị ngưỡng này là cố định, trong các mô phỏng đánh giá hiệu suất của RED [4], [5], nhóm tác giả lấy $max_{th} = 3$. Min_{th} .

RED gồm 2 giải thuật tách biệt: Tính kích thước hàng đợi trung bình và tính xác suất loại bỏ gói tin. Kích thước hàng đợi trung bình được tính toán dựa trên kích thước hàng đợi hiện thời q , trọng số hàng đợi và kích thước hàng đợi trung bình trước đó theo công thức sau:

$$avg = (1 - w_q).avg + w_q.q \quad (1)$$

Giải thuật này xác định mức độ bùng nổ cho phép trong hàng đợi tại gateway. Tính toán xác suất loại bỏ gói tin theo công thức sau [6]:

$$p_b = \begin{cases} 0, & avg < min_{th} \\ \frac{avg - min_{th}}{max_{th} - min_{th}} \cdot max_p, & min_{th} \leq avg \leq max_{th} \\ 1, & avg > max_{th} \end{cases} \quad (2)$$

Ở đây, max_p : giá trị lớn nhất cho p_b

Giải thuật này được sử dụng để tính toán đưa ra quyết định loại bỏ gói tin dựa vào mức độ tắc nghẽn hiện thời. Mục đích là để có sự công bằng trong việc đánh dấu các gói tin ở các khoảng thời gian đều nhau, để tránh sai lệch và tránh hiện tượng các nút nguồn đồng thời giảm kích thước cửa sổ, và để kiểm soát được kích thước hàng đợi trung bình.

Giải thuật tổng quát của RED gateway [4] được mô tả như sau:

```

For Với mỗi gói tin đến
    Tính toán kích thước hàng đợi trung bình  $avg$ 
    if  $min_{th} \leq avg < max_{th}$ 
        Tính toán xác suất  $p_a$ 
        Với xác suất  $p_a$ 
            Đánh dấu gói tin đến
    else if  $max_{th} \leq avg$ 
        Đánh dấu hoặc loại bỏ gói tin đến
    else
        Chấp nhận gói tin đến
  
```

RED đã giải quyết được vấn đề phát hiện sớm tắc nghẽn, tăng hiệu suất sử dụng đường truyền, tránh đồng bộ toàn cục. RED có các biến thể có xu hướng kiểm soát độ trễ hàng đợi trung bình, trong khi vẫn duy trì hiệu suất sử dụng đường truyền cao, giảm tỉ lệ gói tin bị mất, giảm đồng bộ toàn cục và các kết nối bùng nổ.

2.1. Adaptive RED (ARED)

ARED làm cho số tin bị mất và sự khác biệt trong trễ hàng đợi ở mức tối thiểu bằng việc duy trì avg không vượt quá trung bình cộng của ngưỡng trên và ngưỡng dưới. ARED có xác suất drop gói tin trên 1% và dưới 50% [2].

2.2. Enhanced RED (ERED)

Một thuật toán cải tiến khác của RED là ERED [3]. ERED tập trung vào việc thay đổi kích thước hàng đợi trung bình để cải thiện hiệu năng mạng. Nó sử dụng hai tham số là α, β để thay đổi giá trị avg . Việc tính toán avg tương ứng với việc so sánh kích thước cửa sổ hiện thời với ngưỡng dưới và ngưỡng trên. Nếu kích thước cửa sổ hiện thời nhỏ hơn ngưỡng dưới hoặc lớn hơn ngưỡng trên thì tiến hành tinh chỉnh avg sử dụng tham số α, β . Trong mô phỏng, đánh giá, nhóm tác giả đã lựa chọn giá trị cho cả hai tham số này đều là 1.1. ERED đã làm giảm avg so với RED, từ đó biên độ của chiều dài hàng đợi trung bình của ERED lớn hơn so với RED và số gói tin bị mất giảm đi.

2.3. Enhanced Random Early Detection (ENRED)

Alshimaa cùng các cộng sự đã đề xuất ra ENRED [6]. ENRED sử dụng một tham số khác bên cạnh trọng số hàng đợi w_q , được gọi là hàng đợi mục tiêu qt (target queue). Tham số qt này được

xác định bằng hiệu của kích thước hàng đợi hiện thời và trung bình cộng của ngưỡng dưới và ngưỡng trên. Còn kích thước hàng đợi trung bình được tính theo công thức sau:

$$avg = qt(1 - w_q) + q.(qt - w_q) \quad (3)$$

ENRED làm giảm kích thước hàng đợi trung bình của RED, từ đó dẫn đến làm giảm độ trễ hàng đợi và giảm số gói tin bị mất.

2.4. UTRED

Một nghiên cứu khác tập trung vào phần hiệu chỉnh xác suất loại bỏ gói tin có tên gọi là UTRED [7]. Thay vì sử dụng 2 ngưỡng: ngưỡng trên và ngưỡng dưới, UTRED sử dụng 3 ngưỡng là: Ngưỡng trên, ngưỡng dưới và U_{th} (Upper Threshold RED). Nếu kích thước hàng đợi trung bình lớn hơn ngưỡng trên thì trao đổi giá trị giữa max_{th} và U_{th} , rồi tính xác suất loại bỏ gói tin. UTRED cho hiệu năng mạng tốt hơn so với RED xét về thông lượng, tỉ lệ mất gói tin, số gói tin nhận được và bị mất [7], [8].

2.5. DyRED

Một chiến lược khác cũng đã được đề xuất bởi Danladi và cộng sự, có tên gọi là DyRED. DyRED kế thừa gần như nguyên bản của RED, chỉ có một sự thay đổi nhỏ nằm ở việc tinh chỉnh ngưỡng trên. Khi giá trị của avg nằm ở khoảng giữa của ngưỡng dưới và ngưỡng trên thì tinh chỉnh lại ngưỡng trên dựa trên kích thước hàng đợi trung bình, việc tinh chỉnh này cần đảm bảo trễ hàng đợi đủ nhỏ. DyRED cho hiệu quả tốt hơn RED xét về các tham số hiệu năng mạng như: Thông lượng lớn hơn và tỉ lệ mất gói nhỏ hơn trong các trường hợp: Tắc nghẽn nhẹ, tắc nghẽn trung bình và tắc nghẽn nặng [9].

3. Phương pháp nâng cao hiệu năng mạng được đề xuất

Phương pháp được đề xuất ở đây là sự mở rộng của RED [10] – [14]. TqRED kết hợp tinh chỉnh ngưỡng dưới và kích thước hàng đợi trung bình để điều khiển tắc nghẽn trong bộ nhớ đệm của Router ở trạng thái sớm trước khi bộ nhớ đệm bị đầy. Mục đích của phương pháp đề xuất này là nhằm làm tăng thông lượng trung bình, làm giảm độ trễ hàng đợi trung bình và giảm tỉ lệ mất gói tin trong các trường hợp tắc nghẽn là: tắc nghẽn nhẹ, tắc nghẽn nghiêm trọng. TqRED mở rộng RED bằng cách kết hợp tinh chỉnh ngưỡng dưới dựa trên kích thước hàng đợi trung bình avg và tính toán lại avg khi đem kích thước hàng đợi hiện tại so sánh với các ngưỡng. Nó tính toán kích thước hàng đợi trung bình mỗi khi có gói tin đến dựa trên kích thước hàng đợi hiện tại và avg được tính toán trước đó. Ta tinh chỉnh ngưỡng dưới và kích thước hàng đợi trung bình theo biểu thức sau:

$$avg = (1 - w_q).avg/u + w_q.q, \text{ với } u > 1 \quad (4)$$

$$min_{th} = v.avg, \text{ với } v > 1 \quad (5)$$

Trong đó, u và v được lựa chọn hợp lý để đạt được độ trễ hàng đợi trung bình nhỏ.

Mỗi khi có gói tin đến, tùy theo hàng đợi đang không trống hay trống mà tính kích thước hàng đợi trung bình. Sau đó, đem so sánh avg với hai ngưỡng trong bộ nhớ đệm của Router để xác định mức độ tắc nghẽn tại hàng đợi. Nếu avg nhỏ hơn ngưỡng dưới thì ta tính toán lại avg và tinh chỉnh ngưỡng dưới rồi gọi thực thi RED. Còn nếu avg lớn hơn hoặc bằng ngưỡng trên thì gọi thực thi RED như thuật toán gốc, tức là nếu avg lớn hơn ngưỡng trên thì loại bỏ gói tin với xác suất là 1, còn avg nằm giữa ngưỡng dưới và ngưỡng trên thì tính xác suất loại bỏ gói tin và thực hiện loại bỏ gói tin theo xác suất vừa tính được đó.

Phương pháp đề xuất TqRED được mô tả chi tiết như sau:

```

Khởi tạo:  $avg = 0$ ;  $count = -1$ 
for mỗi gói tin đến
    Tính kích thước hàng đợi trung bình  $avg$ 
    if hàng đợi không rỗng
         $avg = (1 - w_q).avg + w_q.q$ 
        if ( $avg < min_{th}$ )  $avg = (1 - w_q).avg/u + w_q.q$ 
    else:  $m = f(time - q\_time)$ 
         $avg = (1 - w_q)^m.avg$ 
    if  $min_{th} \leq avg < max_{th}$ 
         $count++$ 
        Tính xác suất  $p_a$ :
        
$$p_b = \frac{avg - min_{th}}{max_{th} - min_{th}} \cdot max_p$$

        
$$p_a = \frac{p_b}{1 - count \cdot p_b}$$

        Với xác suất  $p_a$ :
        Đánh dấu gói tin đến;  $count = 0$ 
    else if  $max_{th} \leq avg$ :
        Đánh dấu gói tin đến;  $count = 0$ 
    else:  $count = -1$ ;  $min_{th} = v.avg$ 
    Khi hàng đợi trở lên rỗng:  $q\_time = time$ 

```

Trong đó:

q_time : điểm bắt đầu hàng đợi rỗng

$count$: số lượng các gói đến ngay sau gói cuối cùng bị đánh dấu

$time$: thời điểm hiện tại

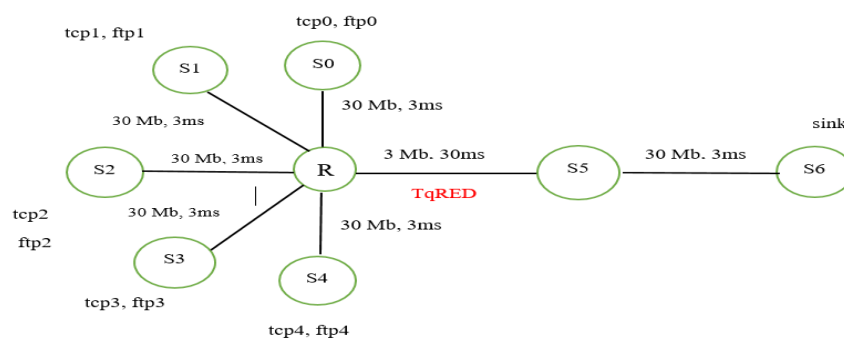
p_a : Xác suất đánh dấu gói tin hiện thời

$f(t)$: Hàm tuyến tính của thời gian t

4. Mô phỏng và đánh giá hiệu năng mạng

4.1. Môi trường mô phỏng

Trong nghiên cứu này, tác giả tiến hành mô phỏng RED và phương pháp đề xuất TqRED sử dụng công cụ mô phỏng mạng NS-2. Sơ đồ (topology) của mạng được thiết kế như trong hình 1.



Hình 1. Sơ đồ topology của mạng mô phỏng

Bảng thông và độ trễ của các liên kết là 30 Mbps và 3 ms, trừ liên kết R-S5 có băng thông và độ trễ là 3 Mbps và 30 ms. Kích thước hàng đợi liên kết R-S5 là 50 gói tin. Liên kết song công giữa nút R và S5 sử dụng kiểu hàng đợi là RED hoặc TqRED, $w_q = 0.002$. Ngoài các tham số về băng thông, độ trễ, kích thước hàng đợi thì các tham số khác về thời gian tuân lưu lượng mạng và thời gian ngừng tuân, thời gian mô phỏng,... phải đảm giống nhau trên các mô phỏng so sánh.

4.2. Số gói tin bị loại bỏ

Tỉ lệ mất gói tin (drop) được xác định bằng tổng số tin bị mất chia cho tổng số gói tin đến.

Số gói tin bị loại bỏ được xác định trong hai trường hợp tương ứng với hai mức độ tắc nghẽn khác nhau trong mạng.

4.2.1. Trường hợp 1: Tắc nghẽn nhẹ

Các giá trị về tổng số gói tin đến và tổng số gói tin bị loại bỏ (drop) ở cả hai chiến lược trong trường hợp tắc nghẽn nhẹ được thể hiện như trong Bảng 1.

Bảng 1. So sánh các gói tin bị drop trong trường hợp tắc nghẽn nhẹ

Chiến lược RED			Chiến lược TqRED		
Tổng số gói tin đến	Tổng số gói tin bị drop	Tỉ lệ % các gói tin bị drop	Tổng số gói tin đến	Tổng số gói tin bị drop	Tỉ lệ % các gói tin bị drop
287340	1635	0,569	290442	1018	0,350
287090	1736	0,605	292058	1016	0,347
290098	1693	0,584	294146	1009	0,343
287596	1692	0,588	290462	993	0,341
289240	1697	0,587	294098	1073	0,365

4.2.2. Trường hợp 2: Tắc nghẽn nghiêm trọng

Các giá trị về tổng số gói tin đến và tổng số gói tin bị drop ở cả hai chiến lược trong trường hợp tắc nghẽn nghiêm trọng được thể hiện như trong Bảng 2.

Bảng 2. So sánh các gói tin bị drop trong trường hợp tắc nghẽn nghiêm trọng

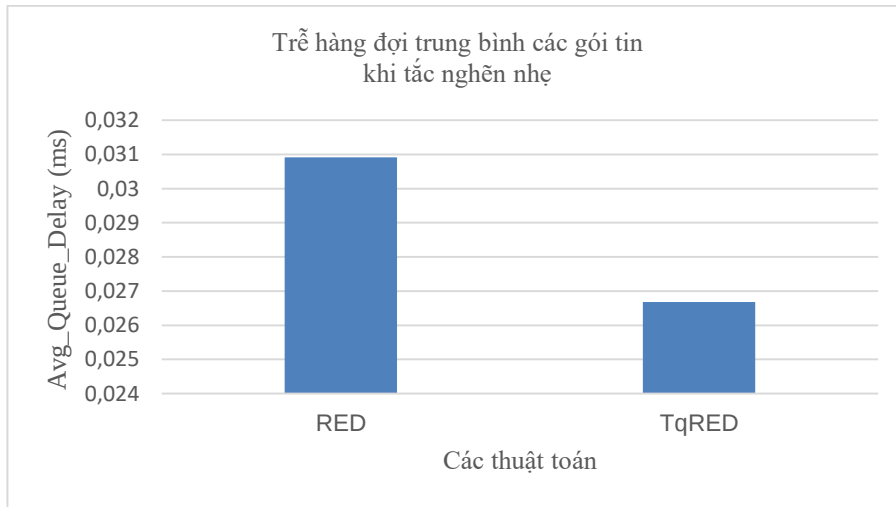
Chiến lược RED			Chiến lược TqRED		
Tổng số gói tin đến	Tổng số gói tin bị drop	Tỉ lệ % các gói tin bị drop	Tổng số gói tin đến	Tổng số gói tin bị drop	Tỉ lệ % các gói tin bị drop
345438	4011	1,158	338526	2256	0,666
346076	4011	1,159	337130	2254	0,669
345044	3989	1,156	334320	2203	0,659
345580	4022	1,164	334306	2205	0,659
345334	4403	1,275	337744	2257	0,668

Trong trường hợp tắc nghẽn nghiêm trọng, số gói tin bị drop của TqRED giảm đi rất nhiều so với RED với tỉ lệ giảm hơn 40%. Còn trong trường hợp nhẹ thì tổng phát các gói tin của TqRED đều lớn hơn so với RED, trong khi số gói tin bị drop nhỏ hơn rất nhiều so với RED. Lý do là bởi TqRED đã thực hiện giảm giá trị của *avg* mỗi khi thấy nó nhỏ hơn ngưỡng dưới. Đây là trường hợp mà gói tin sẽ không bị loại bỏ.

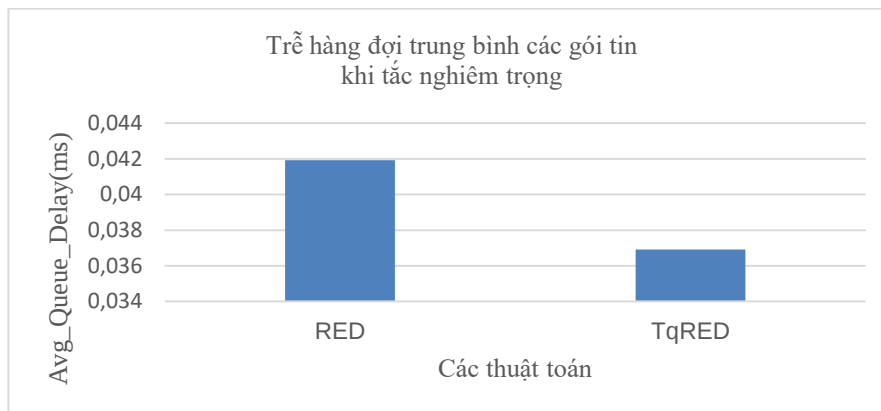
4.3. Trễ hàng đợi trung bình của các gói tin

Trễ hàng đợi trung bình được xác định bởi tổng độ trễ của các gói tin trong hàng đợi chia cho số gói tin vào hàng đợi. Nó được xác định trong hai trường hợp: Tắc nghẽn nhẹ và tắc nghẽn nghiêm trọng. Kết quả so sánh thu thập được thể hiện như hình 2 và hình 3.

Trong cả hai trường hợp, độ trễ hàng đợi trung bình của TqRED đều nhỏ hơn so với RED: Với tắc nghẽn nhẹ là 13,8%, còn với tắc nghẽn nghiêm trọng là 12%. Tắc nghẽn càng nghiêm trọng thì độ trễ hàng đợi trung bình của các gói tin càng lớn.



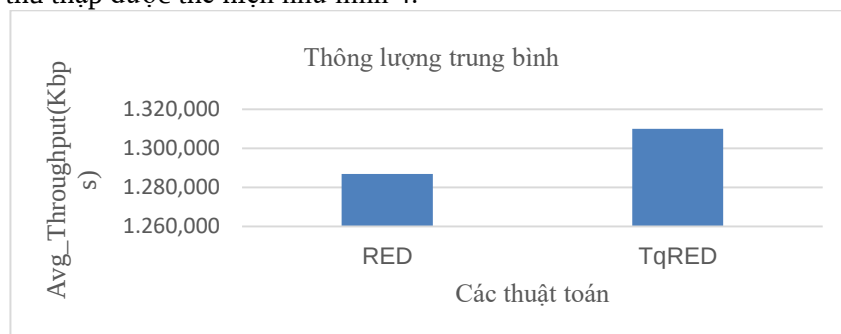
Hình 2. So sánh độ trễ hàng đợi trung bình của các gói tin trong trường hợp tắc nghẽn nhẹ



Hình 3. So sánh độ trễ hàng đợi trung bình của các gói tin trong trường hợp tắc nghẽn nghiêm trọng

4.4. Thông lượng trung bình

Thông lượng trung bình được xác định bằng tổng kích thước các gói tin nhận được chia cho hiệu của thời gian nhận được gói tin cuối cùng trừ đi thời gian nhận được gói tin đầu tiên. Kết quả so sánh thu thập được thể hiện như hình 4.



Hình 4. So sánh thông lượng trung bình giữa TqRED và RED

Thông lượng trung bình của TqRED lớn hơn so với RED là 1,8%. Có được kết quả này là do mỗi khi kiểm tra mà $avg < min_{th}$ thì ta lại điều chỉnh min_{th} lớn hơn, và avg nhỏ hơn.

5. Kết luận

Bài báo đã đề xuất nâng cao hiệu năng mạng với TqRED kết hợp hiệu chỉnh ngưỡng dưới và kích thước hàng đợi trung bình. Thông qua mô phỏng và đánh giá, ta thấy được TqRED cho kết quả tốt hơn RED xét về các tham số hiệu năng như tỉ lệ mất gói, trễ hàng đợi trung bình của các gói tin và thông lượng trung bình. Từ đó, TqRED nâng cao đáng kể hiệu năng hệ thống mạng.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] S. B. Danladi and F. U. Ambursa, "DyRED: An Enhanced Random Early Detection Based on a new Adaptive Congestion Control," *15th International Conference on Electronics Computer and Computation*, Abuja, Nigeria, 2019.
- [2] R. J. La, P. Ranjan, and E. H. Abed, "Analysis of Adaptive Random Early Detection (ARED), Networking," *IEEE/ACM Transaction*, vol. 12, pp. 10791092, 2004.
- [3] D. Que, Z. Chen, and B. Chen, "An Improvement Algorithm Based on RED and Its Performance Analysis," *9th International Conference on Signal Processing*, Beijing, China, 2008.
- [4] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397-413, August 1993.
- [5] M. Khatari and G. Samara, "Congestion Control Approach based on Effective Random Early Detection and Fuzzy Logic," *MAGNT Research Report*, Jordan, 2015.
- [6] A. H. Ismail, A. EL-Sayed, I. Z. Morsi, and Z. Elsaghir, "Enhanced Random Early Detection (ENRED)," *International Journal of Computer Applications (0975 – 8887)*, vol. 92, no. 9, pp. 20-24, April 2014.
- [7] R. Sharma and G. Dixit, "Experimental study of RED Performance by regulating Upper Threshold Parameter," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 5, pp. 6202-6204, 2014.
- [8] A. M. Alkharasani, M. Othman, A. Abdul, and K. Y. Lun, "An Improved Quality of Service Performance Using RED's Active Queue Management Flow Control in Classifying Networks," *IEEE Access*, vol. 5, pp. 24467 - 24478, 2017.
- [9] H. P. Uguta and L. N. Onyejebu, "An Intelligent Fuzzy Logic System for Network Congestion Control," *Circulation in Computer Science*, vol. 2, no. 11, pp. 23-30, December 2017.
- [10] K. K. Chandulal, "A Survey on Red Queue Mechanism for Reduce Congestion in Wireless Network," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 1, pp. 99-103, 2018.
- [11] M. M. Abualhaj, A. A. Abu-Shareha, and M. M. Al-Tahrawi, "FLRED: an efficient fuzzy logic based network congestion control method," *Neural Computing and Applications*, vol. 30, no. 3, pp. 925-935, November 2016.
- [12] L. M. A. Sup, R. M. de Moraes, and A. Bauchspiess, "Explicit non-congestion notification: A new AQM approach for TCP networks," *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, Spain, 2017.
- [13] J. Song and Z. Zhixue, "Research on the Improvement of RED Algorithm in Network Congestion Control," *Applied Mechanics and Materials*, vol. 713, pp. 2471-2477, 2015.
- [14] Z. Yuhong, M. Zhonggui, Z. Xuefeng, and T. Xuyan, "An Improved Algorithm of Nonlinear RED Based on Membership Cloud Theory," *Chinese Journal of Electronics*, vol. 26, no. 3, pp. 538-543, May 2017.