

STUDY, IMPROVE RED QUEUE STRATEGY BASED ON FINE-TUNING THE LOWER THRESHOLD

Vu Van Dien, Le Hoang Hiep*

TNU - University of Information and Communication Technology

ARTICLE INFO	ABSTRACT
Received: 19/10/2021 Revised: 19/4/2022 Published: 21/4/2022	Over the years, congestion has become a major problem affecting the Internet leading to increased packet loss rates and delay. Dynamic queue management (AQM) algorithms have been introduced to control congestion. RED (Random Early Detection) is the first dynamic queue management technique implemented for congestion avoidance control. RED is based on comparing the average queue length with upper and lower thresholds to mark or discard packets. Although, many researchers have come up with improved algorithms for RED, RED still continues to be researched to improve the performance of RED. In this paper, the authors propose an improved RED algorithm called ThRED (Theshold RED) based on lower threshold fine-tune. Through simulation evaluation on the NS2 simulator, the authors found that ThRED gave better results than RED in terms of packet loss and average queue delay.
KEYWORDS Active queue management Congestion Congestion avoid control RED algorithm Lower Theshold	

NGHIÊN CỨU, CẢI TIẾN CHIẾN LƯỢC HÀNG ĐỢI RED DỰA TRÊN VIỆC TÍNH CHỈNH NGƯỠNG DƯỚI

Vũ Văn Diện, Lê Hoàng Hiệp*

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

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 19/10/2021 Ngày hoàn thiện: 19/4/2022 Ngày đăng: 21/4/2022	Trong những năm qua, tắc nghẽn đã trở thành vấn đề chính ảnh hưởng tới Internet, dẫn đến làm tăng tỉ lệ mất gói và làm tăng độ trễ truyền gói tin. Các thuật toán quản lý hàng đợi động (AQM-Active queue management) đã ra đời nhằm điều khiển tắc nghẽn. RED (Random Early Detection) là kỹ thuật quản lý hàng đợi động đầu tiên được triển khai để điều khiển tránh tắc nghẽn. RED dựa trên việc so sánh chiều dài trung bình hàng đợi với ngưỡng trên và ngưỡng dưới để đánh dấu hoặc loại bỏ gói tin. Mặc dù có nhiều nghiên cứu đã công bố đưa ra các thuật toán cải tiến cho RED, nhưng RED vẫn tiếp tục được quan tâm nhằm cải tiến hiệu suất được tốt hơn nữa. Trong bài báo này, nhóm tác giả đề xuất một thuật toán RED cải tiến có tên gọi là ThRED (Theshold RED) dựa trên sự tính chỉnh ngưỡng dưới. Qua mô phỏng đánh giá trên bộ mô phỏng NS2, nhóm tác giả đã thấy được ThRED cho kết quả tốt hơn RED xét về số gói tin bị mất và độ trễ hàng đợi trung bình.
TỪ KHÓA Quản lý hàng đợi động Tắc nghẽn Điều khiển tránh tắc nghẽn Thuật toán RED Ngưỡng dưới	

DOI: <https://doi.org/10.34238/tnu-jst.5180>

* Corresponding author. Email: lhiep@ictu.edu.vn

1. Giới thiệu

Internet được xây dựng dựa trên IP cung cấp các dịch vụ vận chuyển đầu cuối - đầu cuối (end to end). Khi các gói tin được gửi đến một nút (node) mạng nào đó, gói tin đó sẽ được đưa vào bộ nhớ đệm của nút đó trước khi được lấy ra để xử lý. Tắc nghẽn mạng chỉ được phát hiện khi bộ nhớ đệm bị đầy và khi gói tin đến bị mất (drop). Tắc nghẽn mạng là vấn đề chính ảnh hưởng đến chất lượng dịch vụ (QoS) của mạng IP [1]. Tỷ lệ mất gói, trễ, thông lượng trung bình là các vấn đề chính của mạng. Giảm số gói tin bị mất và giảm trễ trung bình của mạng là mục tiêu quan trọng trong việc cải thiện chất lượng dịch vụ mạng. Quản lý hàng đợi đóng vai trò quan trọng trong việc điều khiển tránh tắc nghẽn. Có hai kỹ thuật được sử dụng: Một là ngăn ngừa tắc nghẽn được sử dụng trước khi có tắc nghẽn xảy ra; Hai là, điều khiển tắc nghẽn khi nó đã xảy ra. Kỹ thuật ngăn ngừa tắc nghẽn là kỹ thuật chủ động để duy trì mạng ở trạng thái có tỷ lệ mất gói thấp và thông lượng trung bình cao bằng việc duy trì kích thước hàng đợi trung bình ở mức thấp so với mức thay đổi lưu lượng mạng. Lúc đó, TCP sẽ giảm tốc độ truyền khi phát hiện ra gói tin bị mất [2], [3].

Có nhiều thuật toán khác nhau đã được đề xuất để xử lý vấn đề tắc nghẽn từ Drop Tail cho đến các chiến lược hàng đợi động (AQM). Trong đó, các chiến lược hàng đợi động là các chiến lược phổ biến được sử dụng trên bộ định tuyến (Router). Chiến lược hàng đợi động cảm nhận tắc nghẽn có thể xảy ra đủ sớm và loại bỏ các gói tin dựa trên xác suất drop gói tin. Từ đó, Router thông báo cho nút nguồn điều chỉnh tốc độ phát, dẫn đến số gói tin bị mất giảm đi, tăng thông lượng trung bình. Có rất nhiều các chiến lược hàng đợi động từ RED, BLUE, FRED,... cho đến DyRED. Chiến lược RED (Random Early Detection) được đề xuất bởi Floyd và Jacobson [4]. Đây là chiến lược hàng đợi động đầu tiên và được sử dụng phổ biến để tránh tắc nghẽn. 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), cùng với đó là sự so sánh với ngưỡng trên (*maximum thresh*) và ngưỡng dưới (*minimum thresh*). Nếu *avg* nhỏ hơn ngưỡng dưới thì không loại bỏ gói tin. Nếu *avg* lớn hơn ngưỡng trên thì loại bỏ gói tin. Còn nếu *avg* nằm giữa ngưỡng trên và ngưỡng dưới thì các gói tin sẽ bị loại bỏ ngẫu nhiên theo xác suất được tính toán dựa trên *avg*, ngưỡng trên và ngưỡng dưới.

Bằng việc phát hiện tắc nghẽn sớm, RED đã cho thấy lợi thế của mình so với Drop Tail trong việc giảm số gói tin bị mất và tăng thông lượng, giảm độ trễ truyền gói tin. Tuy nhiên, khi lưu lượng mạng tăng đột ngột thì RED tỏ ra chưa thực sự hiệu quả trong việc giảm số gói tin bị mất [5]-[9] và giảm độ trễ hàng đợi trung bình. Nghiên cứu này sẽ đưa ra phương pháp giải quyết vấn đề nêu trên.

2. Thuật toán RED và một số chiến lược quản lý hàng đợi động khác

Thuật toán RED [4] 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 (để loại bỏ sau này nếu cần thiết) 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 sẽ tính toán kích thước hàng đợi trung bình dựa trên bộ lọc thông thấp (*LowPass Filter*), giá trị trung bình này còn được gọi là trung bình trượt có trọng số tăng theo hàm mũ - EWMA (Exponential Weighted Moving Average). Kích thước hàng đợi trung bình được so sánh với hai giá trị ngưỡng: ngưỡng dưới min_{th} và ngưỡng trên max_{th} để quyết định việc đánh dấu hoặc loại bỏ các gói tin trong hàng đợi. Các ngưỡng này là cố định, trong các mô phỏng đánh giá hiệu suất của RED [5], nhóm tác giả lấy $max_{th} = 3 \cdot min_{th}$.

RED gồm 2 giải thuật tách biệt: Tính kích thước hàng đợi trung bình và tính xác suất loại bỏ gói tin. Tính kích thước hàng đợi trung bình theo công thức sau:

$$avg = (1 - w_q) \cdot avg + w_q \cdot q \quad (1), \text{ với } q \text{ là kích thước hàng đợi hiện tại.}$$

Giải thuật này quyết định mức độ bùng nổ cho phép trong hàng đợi tại gateway. Nó đưa ra bản mô tả các chu kỳ khi hàng đợi rỗng (chu kỳ rỗi) bằng cách đánh giá số lượng m của các gói

nhỏ có thể được truyền trong suốt chu kỳ rồi bởi Router. Sau mỗi chu kỳ rồi, Router lại tính toán kích thước hàng đợi trung bình như thể m gói đã đến được hàng đợi rỗng trong suốt chu kỳ đó. Tính toán xác suất loại bỏ gói tin theo công thức sau [6]:

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

Ở đây, max_p : giá trị lớn nhất cho p_b

Giải thuật tính xác suất loại bỏ gói tin phải đảm bảo sao cho các gói tin được đánh dấu tại những khoảng không gian đều nhau, mục đích để tránh hiện tượng đồng bộ toàn cục các luồng TCP, trong khi vẫn giữ kích thước hàng đợi trung bình ở một giới hạn nhất định. Độ chiếm giữ hàng đợi lớn thì xác suất loại bỏ gói càng cao, độ chiếm giữ hàng đợi càng gần giá trị max_{th} thì xác suất loại bỏ gói dần tiến tới giá trị max_p .

```

For each packet arrival
    Calculate the average queue size avg
    if  $min_{th} \leq avg < max_{th}$ 
        Calculate propability  $p_a$ 
        With propability  $p_a$ 
            mark or drop the arriving packet
    else if  $max_{th} \leq avg$ 
        mark or drop the arriving packet
    else
        accept the arriving packet
  
```

Giải thuật tổng quát của RED được mô tả như sau:

Khi kích thước hàng đợi trung bình nằm trong khoảng giá trị min_{th} và max_{th} thì mỗi gói đến đều được đánh dấu hoặc loại bỏ bằng một xác suất p_a , đây là một chức năng của kích thước hàng đợi trung bình. Tại mỗi thời điểm có một gói bị đánh dấu và xác suất gói bị đánh dấu từ một kết nối điển hình tỉ lệ với băng thông chia sẻ kết nối tại mỗi router.

Giải thuật chi tiết của RED được mô tả như sau:

```

Khởi tạo:
     $avg = 0$ 
     $count = 0$ 
for mỗi gói tin đến
    Tính kích thước hàng đợi trung bình avg
    if hàng đợi không rỗng
         $avg = (1 - w_q) \cdot avg + w_q \cdot q$ 
    else
         $m = f(time - q\_time)$ 
  
```

```

        avg = (1 - wq)m.avg
    if minth ≤ avg < maxth
        count++
        Tính xác suất pa :
        pb =  $\frac{avg - min_{th}}{max_{th} - min_{th}} \cdot max_p$ 
        pa =  $\frac{p_b}{1 - count \cdot p_b}$ 
        Với xác suất pa :
        Đánh dấu gói tin đến
        count = 0
    else if maxth ≤ avg :
        Đánh dấu gói tin đến
        count = 0
    else: count = -1
    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

f(t): một hàm tuyến tính của thời gian *t*

RED đã giải quyết được vấn đề phát hiện tắc nghẽn sớm, hiệu suất sử dụng đường truyền, đồng bộ toàn cục. Các nghiên cứu sau này đã tìm cách cải tiến nhằm khắc phục những hạn chế của RED. Alshima cùng các cộng sự đã đề xuất ra Enhanced Random Early Detection (ENRED) để giải quyết vấn đề khởi tạo bộ tham số trong RED. ENRED sử dụng một tham số mới bên cạnh trọng số của hàng đợi là w_q , được gọi là hàng đợi mục tiêu *qt* (target queue). Tham số *qt* được xác định bằng hiệu của kích thước cửa sổ hiện thời và trung bình cộng của ngưỡng trên và ngưỡng dưới. ENRED làm giảm kích thước hàng đợi trung bình của RED, từ đó dẫn đến làm giảm trễ và tỉ lệ gói tin bị mất. Một nghiên cứu khác, cũng tập trung vào vấn đề khởi tạo bộ tham số, đó là UTRED [7]. UTRED sử dụng 3 ngưỡng là: min_{th} , max_{th} và U_{th} (Upper Threshold RED). Nếu kích thước hàng đợi trung bình lớn hơn max_{th} thì tính toán xác suất *p*, trao đổi giá trị giữa max_{th} và U_{th} . UTRED cho hiệu năng tốt hơn so với RED xét về tỉ lệ phân phối gói tin và thông lượng [8]. Một chiến lược khác cũng đã được đề xuất bởi Danladi và cộng sự, đó là DyRED. DyRED tập trung vào việc thay đổi ngưỡng trên. Khi *avg* nằm khoảng giữa ngưỡng trên và ngưỡng dưới thì tính chỉnh lại ngưỡng trên dựa trên *avg*. DyRED cho hiệu quả tốt hơn RED xét về số gói tin bị mất ít hơn và thông lượng lớn hơn trong các trường hợp tắc nghẽn ở các mức độ khác nhau [9].

Một cải tiến khác của RED là ERED. Ý tưởng cơ bản của ERED là kết hợp chiều dài hàng đợi hiện thời và *avg* trước đó để tính *avg*. Việc tính toán *avg* tương ứng với việc so sánh chiều dài cửa sổ hiện thời với ngưỡng trên và ngưỡng dưới. ERED đã làm giảm *avg* so với RED, từ đó làm giảm số gói tin bị mất. FWMRED [10] cũng đã được đề xuất nhằm tăng băng thông, giảm tỉ lệ

mất gói và độ trễ. FWMRED khác các thuật toán khác ở chỗ: Thay đổi giá trị w_q , sử dụng $\max_p$ để duy trì giá trị avg nằm giữa ngưỡng trên và ngưỡng dưới, và bộ đệm có thể tính toán xác suất loại bỏ gói tin động p_d .

3. Phương pháp cải tiến chiến lược hàng đợi RED

Cải tiến được đề xuất ở đây là sự mở rộng của RED [11]-[15]. ThRED tinh chỉnh ngưỡng dưới để điều khiển tắc nghẽn trong bộ nhớ đệm của Router ở trạng thái sớm trước bộ nhớ đệm bị tràn. Cải tiến này nhằm làm giảm số gói tin bị mất và giảm độ trễ hàng đợi trung bình của các gói tin trong mạng. Mục đích của cải tiến chiến lược hàng đợi RED nhằm làm giảm số gói tin bị mất và giảm độ trễ hàng đợi trung bình. Hiệu suất của chiến lược cải tiến được hiển thị bằng kết quả của việc gói tin bị drop và độ trễ hàng đợi trung bình trong các trường hợp tắc nghẽn là: tắc nghẽn nhẹ, tắc nghẽn nghiêm trọng. ThRED cải tiến RED bằng cách hiệu chỉnh ngưỡng dưới dựa trên kích thước hàng đợi trung bình avg . Nó tính toán kích thước hàng đợi trung bình mỗi khi có gói tin đến, mà avg lại được tính toán dựa trên avg trước đó và chiều dài hiện tại của hàng đợi. Ta tinh chỉnh ngưỡng dưới theo biểu thức sau :

$$\min_{th} = b \cdot avg, \text{ với } b > 1 \quad (3)$$

Trong đó, b được lựa chọn hợp lý để đạt được độ trễ hàng đợi nhỏ.

Như vậy, mỗi khi có gói tin đến, tùy theo hàng đợi đang trống hay là không mà ta 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. Nếu avg nhỏ hơn ngưỡng dưới thì ta tinh chỉnh ngưỡng dưới rồi gọi thực thi RED. Còn nếu avg lớn hơn hoặc bằng ngưỡng dưới thì ta gọi thực thi RED, tức là nếu avg lớn hơn ngưỡng trên thì drop gói tin, còn avg nằm giữa ngưỡng trên và ngưỡng dưới thì tính xác suất drop gói tin và thực hiện drop gói tin theo xác suất đó.

Chiến lược cải tiến RED là ThRED có thể được mô tả như sau:

if $avg < \min_{th}$:

Tinh chỉnh $\min_{th}$ theo công thức (3)

Gọi thực thi RED

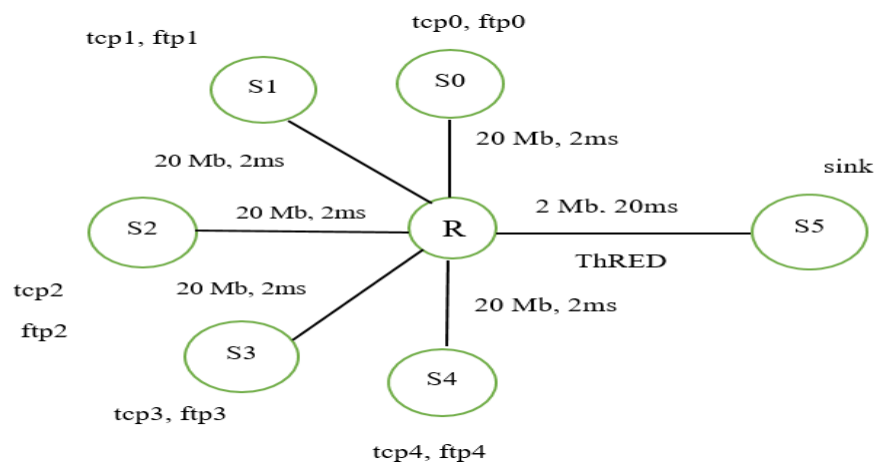
else :

Gọi thực thi RED

4. Mô phỏng và các kết quả

4.1. Môi trường mô phỏng

Trong nghiên cứu này, nhóm tác giả tiến hành mô phỏng chiến lược RED và chiến lược ThRED 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 20Mbps và 2ms trừ liên kết R-S5 có bảng thông và độ trễ là 2Mbps và 20 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 ThRED.

4.2. Số gói tin bị loại bỏ

Số gói tin bị drop được xác định trong hai trường hợp tương ứng với hai mức độ tắc nghẽn khác nhau:

4.2.1. Trường hợp 1: Tắc nghẽn nhẹ

Các giá trị về tổng số gói tin đến và gói tin bị 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 ThRED		
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
102160	711	0,696	113525	346	0,305
99626	695	0,698	110807	341	0,308
100117	679	0,678	111217	357	0,321
99536	693	0,696	110898	371	0,335
103653	804	0,776	113743	366	0,322

4.2.2. Trường hợp 2: Tắc nghẽn nghiêm trọng

Các giá trị về tổng số gói tin đến và 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

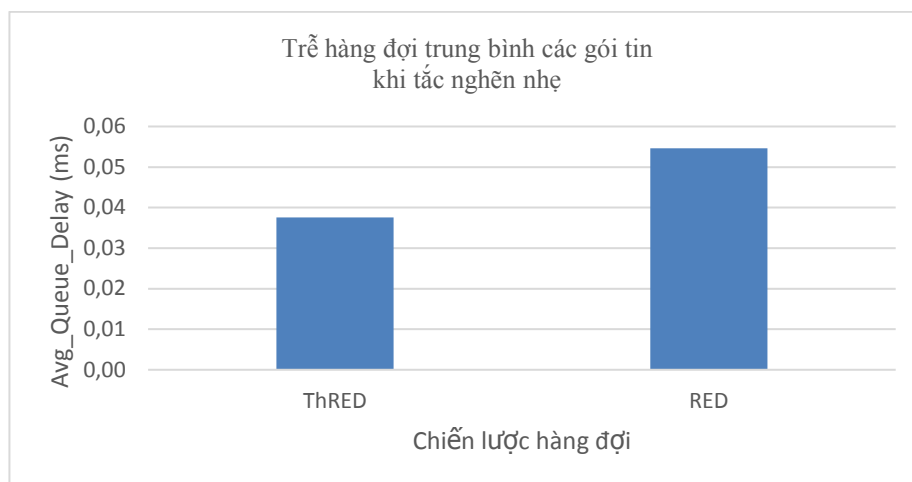
Chiến lược RED			Chiến lược ThRED		
Tổng số gói tin đến	Tổng số gói tin bị drop	Tỉ lệ % các gói tin bị drop	Tổng số gói tin đến	Tổng số gói tin bị drop	Tỉ lệ % các gói tin bị drop
126086	7408	5,876	118926	2632	2,213
125908	7276	5,779	118930	2621	2,204
126137	7392	5,860	118919	2640	2,220
126054	7390	5,863	118924	2648	2,227
125914	7275	5,778	118964	2608	2,192

Trong trường hợp tắc nghẽn nghiêm trọng, số gói tin bị drop của ThRED giảm đi rất nhiều so với RED. Còn trong trường hợp tắc nghẽn nhẹ thì tổng phát các gói tin của ThRED đều lớn hơn so với RED, trong khi số gói tin bị drop nhỏ hơn rất nhiều so với RED.

4.3. Trễ hàng đợi trung bình của các gói tin

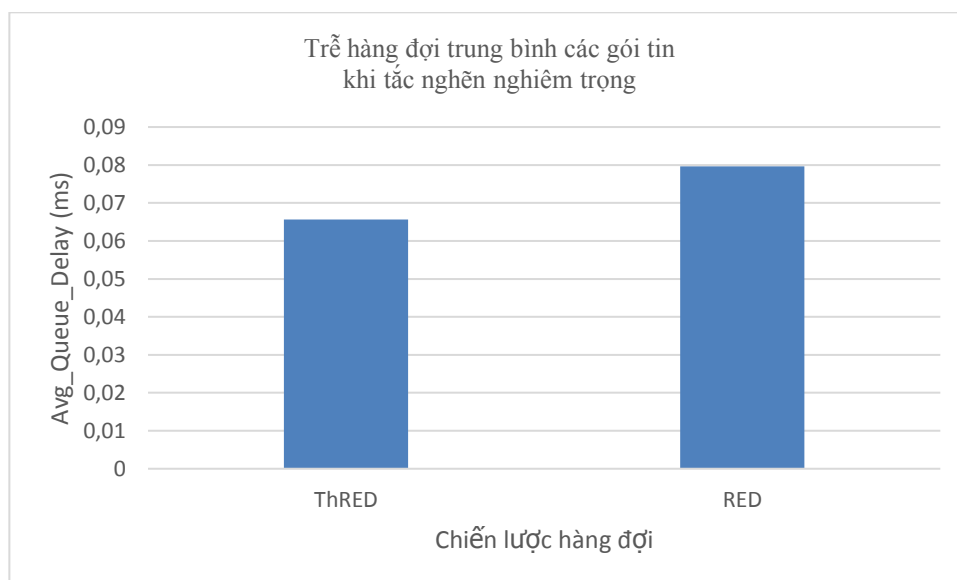
Trễ hàng đợi trung bình của các gói tin được xác định bởi tổng độ trễ của các gói tin trong hàng đợi chia cho số gói tin vào hàng đợi. Nó được xác định trong hai trường hợp: Tắc nghẽn nhẹ và tắc nghẽn nghiêm trọng. Kết quả so sánh thu thập được thể hiện như hình 2 và hình 3:

4.3.1. Trường hợp 1: Tắc nghẽn nhẹ



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

4.3.2. Trường hợp 2: Tắc nghẽn nghiêm trọng



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

Trong cả hai trường hợp, độ trễ hàng đợi trung bình của ThRED đều nhỏ hơn so với RED: Với tắc nghẽn nhẹ là 31,1%, còn với tắc nghẽn nghiêm trọng là 17,6%. Tắc nghẽn càng nghiêm trọng thì độ trễ hàng đợi trung bình càng lớn. Như vậy, chúng ta thấy khi sử dụng chiến lược hàng đợi ThRED cho kết quả tốt hơn chiến lược RED xét về số gói tin bị drop và độ trễ hàng đợi trung bình.

5. Kết luận

Bài báo đã đề xuất việc cải tiến RED với chiến lược hàng đợi ThRED. Thông qua mô phỏng và đánh giá ta thấy được ThRED không chỉ cho kết quả là số gói tin bị drop giảm đi so với RED mà còn cho kết quả tốt hơn về độ trễ hàng đợi trung bình, dẫn đến làm giảm độ trễ truyền gói tin giữa các nút trong mạng. Từ đó ThRED cải thiện 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, 2019.
- [2] 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.
- [3] H. P. Uguta and L. N. Onyejegbu, "An Intelligent Fuzzy Logic System for Network Congestion Control," *Circulation in Computer Science*, vol. 2, no.11, pp. 23-30, December 2017.
- [4] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *Institute of Electrical and Electronics Engineers(IEEE)*, pp. 1-22, August 1993.
- [5] M. Khatairi and G. Samara, "Congestion Control Approach based on Effective Random Early Detection and Fuzzy Logic," MAGNT Research Report, Jordan, 2015.
- [6] D. Que, Z. Chen, and B. Chen, "An Improvement Algorithm Based on RED and Its Performance Analysis," 9th International Conference on Signal Processing, 2008.
- [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," *Institute of Electrical and Electronics Engineers*, vol. 4, pp. 1-12, 2016.
- [9] 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.
- [10] A. M. Alakharasani, M. Othman, A. Abdullah, and K. Y. Lun, *An Improved Quality-of-Service Performance Using RED's Active Queue Management Flow Control in Classifying Networks*, 2017.
- [11] M. M. Abualhaj et al., "FLRED: an efficient fuzzy logic based network congestion control method," *Neural Computing and Applications*, vol. 30, no. 3, pp. 925-935, November 2016.
- [12] M. Khatairi and G. Samara, "Congestion Control Approach based on Effective Random Early Detection and Fuzzy Logic," MAGNT Research Report, 2015.
- [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 et al., "An Improved Algorithm of Nonlinear RED Based on Membership Cloud Theory," *Chinese Journal of Electronics*, vol. 26, no. 3, pp. 538-543, May 2017.
- [15] M. Abdulkareem et al., "EFRED: Enhancement of Fair Random Early Detection Algorithm," *International Journal of Communications Network and System Sciences*, vol. 8, pp. 282-294, July 2015.