

ĐÁNH GIÁ HIỆU NĂNG CỦA MỘT SỐ CƠ CHẾ QUẢN LÝ HÀNG ĐỢI TÍCH CỰC DỰA TRÊN KÍCH THƯỚC HÀNG ĐỢI VÀ TẢI NẠP

Nguyễn Kim Quốc¹, Võ Thanh Tú²

¹Trường Đại học Nguyễn Tất Thành

²Trường Đại học Khoa học, Đại học Huế

Tóm tắt: Internet phải đối mặt với sự bùng nổ về số lượng máy tính kết nối và sự đa dạng của các lớp ứng dụng triển khai trên nó. Quản lý hàng đợi tích cực là một trong các giải pháp cho điều khiển tránh tắc nghẽn trên Internet. Trong những năm gần đây, các nhà nghiên cứu về mạng đã đề xuất nhiều cơ chế quản lý hàng đợi tích cực. Bài báo này thực hiện đánh giá hiệu năng của các cơ chế quản lý hàng đợi tích cực dựa trên kích thước hàng đợi và tải nạp, để phân lớp và ứng dụng các cơ chế thích nghi với môi trường mạng khác nhau.

1. Đặt vấn đề

Cơ chế kiểm soát tránh tắc nghẽn là một trong những vấn đề đảm bảo truyền thông liên tục và hiệu quả trên mạng Internet. Để đáp ứng yêu cầu trên, các nhà khoa học đã không ngừng nghiên cứu cải tiến các giao thức điều khiển từ đầu cuối đến đầu cuối (end-to-end) nhằm nâng cao hiệu năng của giao thức TCP, như: TCP NewReno, Vegas, Vegas-A [4] và cải tiến các phương pháp quản lý hàng đợi tích cực, như: REM, FRED, BLUE [3,4,6] tại các nút mạng trung tâm. Thông qua các cơ chế đó, mỗi nút mạng đã kiểm soát được số lượng lớn các gói dữ liệu đến đồng thời trong hàng đợi của bộ định tuyến. Bài báo tập trung nghiên cứu một số giải pháp giải quyết vấn đề tránh tắc nghẽn trong môi trường mạng có mật độ gói tin dày đặc. Thêm vào đó, chúng tôi cũng đã xác định được các tham số điều khiển tối ưu để đáp ứng thích nghi những thay đổi tức thời của mạng truyền thông đa phương tiện. Qua đó, chúng tôi đánh giá và so sánh các cơ chế quản lý hàng đợi tích cực cho phù hợp với từng loại mạng và từng lớp ứng dụng dựa trên các kết quả mô phỏng trên NS-2 [11].

Phân tích hiệu quả một số cơ chế quản lý hàng đợi tích cực dựa theo kích thước hàng đợi

Cơ chế quản lý hàng đợi tích cực dựa theo kích thước hàng đợi đã được nghiên cứu nhiều trong thời gian qua, như: RED, FRED, ARED [7,14,16]. Các cơ chế này dựa trên xác suất hủy bỏ sớm ngẫu nhiên của các gói tin theo công thức:

$$p_b = \max_p \frac{\hat{k} - \min_{th}}{\max_{th} - \min_{th}} \quad (1)$$

Khi kích thước hàng đợi trung bình \hat{k} nhỏ hơn mức ngưỡng nhỏ nhất \min_{th} thì không có gói tin nào bị đánh dấu (hay gán xác suất đánh dấu bằng 0). Khi kích thước hàng đợi trung bình lớn hơn mức ngưỡng lớn nhất \max_{th} thì tất cả các gói đến đều bị đánh dấu và trên thực tế các gói có thể bị loại bỏ (hay gán xác suất đánh dấu bằng 1). Khi kích thước hàng đợi trung bình nằm trong khoảng giá trị ngưỡng nhỏ nhất \min_{th} và giá trị ngưỡng lớn nhất \max_{th} thì mỗi gói đến đều được đánh dấu bằng một xác suất p_b .

Kích thước hàng đợi trung bình được sử dụng với mục đích là để tránh sự dao động quá nhanh của hàng đợi khi có những đợt gửi gói tin với thời gian ngắn. RED tính toán \hat{k} với hệ số ω mỗi khi có gói tin đến theo công thức sau:

$$\hat{k} = (1 - \omega) * \hat{k} + \omega * k \quad (2)$$

Với ω ($0 \leq \omega \leq 1$) là trọng số hàng đợi và k là kích thước hàng đợi hiện thời. Nếu giá trị của ω đủ nhỏ, thì giá trị trung bình sẽ có khuynh hướng ít thay đổi, và sẽ ít bị ảnh hưởng đối với những đợt gửi gói tin ngắn.

Do vậy, RED tỏ ra kém hiệu quả khi có sự thay đổi nhanh chóng mật độ gói tin vào hàng đợi. Cụ thể là các tham số điều chỉnh kém thích nghi làm tăng số gói tin rơi, nên người ta đã cải tiến bằng ARED dựa vào việc thay đổi tham số \max_p phù hợp trong trường hợp số luồng đến đồng thời lớn [14]. Tuy nhiên những cải tiến này chỉ có ý nghĩa cho các loại gói tin có cùng dịch vụ [19]. Từ đó đã ra đời một số cơ chế quản lý hàng đợi dựa vào tải nạp như BLUE, SFB (Stochastic Fair Blue) [19,20].

3. Phân tích hiệu quả một số cơ chế quản lý hàng đợi tích cực dựa tải nạp

Ngày nay, môi trường mạng có nhiều luồng đến đồng thời khá phổ biến. Do vậy, BLUE là cơ chế quản lý hàng đợi theo tải nạp hiệu quả hơn phương pháp dựa vào kích thước hàng đợi trung bình [20]. Điều này tương phản một cách rõ ràng với tất cả các thuật toán điều khiển hàng đợi tích cực đã biết, bởi các thuật toán này sử dụng không gian của hàng đợi như là một tiêu chuẩn trong việc điều khiển tránh tắc nghẽn.

BLUE sử dụng một biến xác suất p_m gán cho các gói tin khi chúng vào hàng đợi. Xác suất này tăng/giảm một cách tuyến tính tùy thuộc vào tỉ lệ rơi gói tin hay mức độ sử dụng đường truyền. Nếu như hàng đợi liên tục hủy bỏ các gói tin vì nguyên nhân tràn hàng đợi, BLUE sẽ tăng p_m và như vậy tăng mức độ nghiêm trọng của thông báo tắc nghẽn mà nó gửi trở về nguồn. Ngược lại, nếu như hàng đợi trở nên trống hoặc đường truyền rỗi, BLUE giảm xác suất p_m . Điều này cho phép BLUE tự điều chỉnh tốc độ cần thiết để gửi thông báo tắc nghẽn trở lại nơi gửi hoặc cho rơi gói tin bằng cách:

dựa trên mức độ mất gói: if ((now – last_update) > freeze_time) then

$$p_m = p_m + \delta_1 \text{ và } \text{Last_update} = \text{now} \quad (3)$$

hoặc dựa trên kết nối rỗi: if ((now – last_update) > freeze_time) then

$$p_m = p_m - \delta_2 \text{ và } \text{Last_update} = \text{now} \quad (4)$$

trong đó:

p_m : Xác suất đánh dấu hoặc loại bỏ gói tin

δ_1 : Lượng tăng của p_m

δ_2 : Lượng giảm của p_m

now : Thời gian hiện hành

last_update : Thời gian cuối cùng p_m thay đổi

freeze_time : Lượng thời gian giữa các thay đổi thành công

Lượng tăng của p_m thể hiện bởi δ_1 và lượng giảm của p_m được thể hiện bằng δ_2 . Đại lượng này luôn được cập nhật khi có sự thay đổi của p_m , khi kích thước hàng đợi vượt quá giá trị ngưỡng hiện tại, tại tốc độ $1/\text{freeze_time}$. Tham số freeze_time thể hiện khoảng thời gian giữa các lần cập nhật thành công p_m .

4. Giải pháp tích hợp cơ chế quản lý hàng đợi dựa vào kích thước hàng đợi và tải nạp

Việc sử dụng các cơ chế quản lý hàng đợi dựa trên kích thước hàng đợi chỉ có hiệu quả đối với mạng ít có sự thay đổi lưu lượng vào, còn cơ chế quản lý hàng đợi dựa trên tải nạp thì đáp ứng nhanh trong mạng có nhiều luồng đến đồng thời. Vì vậy, việc kết hợp hai cơ chế này rất có ý nghĩa cho môi trường mạng phức tạp hiện nay, đó là cơ chế: REM (Random Exponential Marking), GREEN (Generalized Random Early Evasion Network).

4.1. Nguyên lý hoạt động của REM

Ý tưởng của REM là ổn định tải đầu vào và năng lực liên kết của hàng đợi, bất kể số lượng người sử dụng chia sẻ liên kết [13,17,18]. Mỗi hàng đợi ra của bộ định tuyến được cài đặt cơ chế REM và duy trì một biến gọi là 'pire'. Price như là một yếu tố đánh giá tắc nghẽn. Price được cập nhật, định kỳ hoặc không đồng bộ, dựa trên bất đối xứng của tải và bất đối xứng kích thước hàng đợi. Sự bất đối xứng của tải là sự khác biệt giữa tốc độ của các luồng dữ liệu vào và năng lực hiện có của liên kết tại bộ định tuyến. Sự bất đối xứng của kích thước hàng đợi là sự khác biệt giữa yêu cầu kích thước hàng đợi mong muốn với kích thước hàng đợi hiện có của bộ định tuyến.

Mức Price này tăng lên nếu tổng trọng lượng của các bất xứng này là dương và giảm đi trong trường hợp ngược lại. Tổng trọng lượng là dương khi một trong các đầu vào vượt quá khả năng liên kết hoặc có quá nhiều gói tồn đọng trong hàng đợi cần được xóa và âm trong các trường hợp ngược lại. Khi số lượng người sử dụng tăng, tức là sự không đồng bộ giữa các tải với kích thước hàng đợi tăng, Price sẽ tăng và do đó xác suất tắc nghẽn được thiết lập tăng. Điều này sẽ gửi một tín hiệu tắc nghẽn mạnh mẽ hơn đến các nguồn phát, yêu cầu các nguồn phát giảm tốc độ phát. Khi tải nguồn quá nhỏ, sự không đồng bộ sẽ âm, Price giảm, xác suất được thiết lập giảm và tải nguồn được yêu

cầu tăng lên cho đến khi sự bất xứng về không. Giả sử kích thước hàng đợi $p_i(t)$ trong giai đoạn t được cập nhật theo công thức sau:

$$p_i(t+1) = [p_i(t) + \gamma(\alpha_i(b_i(t) - b_i^*) + x_i(t) - c_i(t))]^+ \quad (5)$$

Trong đó, $\gamma > 0$ và $\alpha_i > 0$ là các hằng số nhỏ và $[z]^+ = \max\{z, 0\}$. Ở đây, $b_i(t)$ là kích thước hàng đợi trung bình của hàng đợi i ở thời điểm t và $b_i^* \geq 0$ là chiều dài hàng đợi mục tiêu, $x_i(t)$ là tốc độ tải trung bình của hàng đợi i ở thời điểm t , và $c_i(t)$ bằng thông cho phép của hàng đợi i tại thời điểm t . Sự khác biệt về tốc độ tải là $x_i(t) - c_i(t)$ và của kích thước hàng đợi $b_i(t) - b_i^*$. Các hằng số α_i có thể được thiết lập bởi mỗi hàng đợi mỗi riêng lẻ và được cập nhật theo hiệu suất và độ trễ ở mỗi hàng đợi. Các hằng γ kiểm soát đáp ứng của REM thay đổi tùy theo điều kiện mạng.

Mặc dù REM đã quản lý hàng đợi tích cực theo kích thước hàng đợi và tải nạp, nhưng REM còn tỏ ra thiếu tính cân bằng và sử dụng băng thông của đường truyền chưa cao. Vì vậy, cơ chế quản lý hàng đợi GREEN ra đời.

4.2. Nguyên lý hoạt động của GREEN

Băng thông của một kết nối phụ thuộc nhiều yếu tố, trong đó có RTT và xác suất mà các gói tin bị đánh rơi trong mạng. Băng thông một kết nối đáp ứng các phương trình (6):

$$BW = \frac{MSS \times c}{RTT \times \sqrt{p}} \quad (6)$$

Trong đó, BW là băng thông/thông lượng của kết nối, MSS là kích thước tối đa phân đoạn, RTT là thời gian đi vòng gói tin, p là xác suất mất gói tin và c là một hằng số phụ thuộc vào chiến lược báo nhận được sử dụng và trạng thái các gói tin bị mất định kỳ hoặc ngẫu nhiên.

Giả sử có N luồng đang hoạt động ở một bộ định tuyến vào và một liên kết ra có năng lực L . GREEN xem xét một luồng được hoạt động nếu nó đã có ít nhất một gói tin đi qua các bộ định tuyến trong một cửa sổ có thời gian nhất định. Chia sẻ công bằng băng thông cho mỗi luồng là L/N . Thay L/N cho BW trong phương trình trên, ta có xác suất mất gói tin p , là:

$$p = \left(\frac{N \times MSS \times c}{L \times RTT} \right)^2 \quad (7)$$

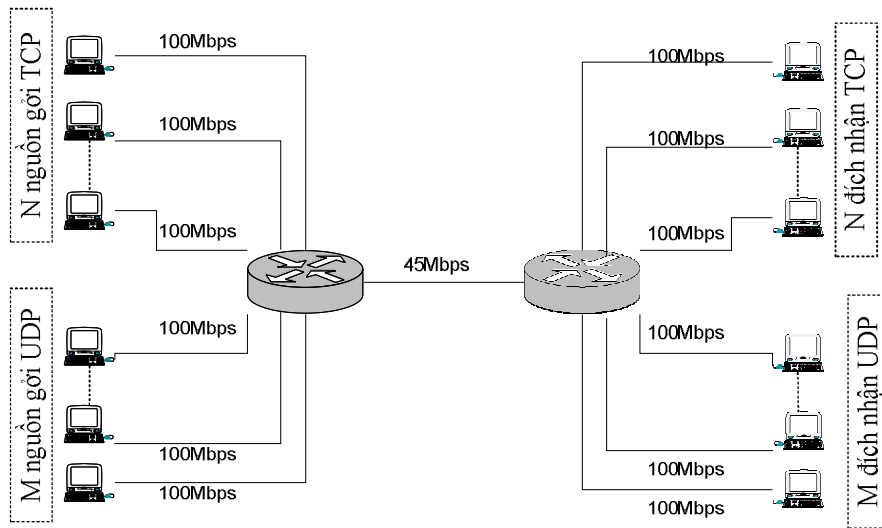
Sử dụng giá trị của p để thông báo tình trạng tắc nghẽn, GREEN buộc các luồng chia sẻ cân bằng tốc độ gửi. Vì p phụ thuộc vào số lượng luồng và RTT mỗi luồng, thông báo tắc nghẽn sẽ tích cực hơn khi cho N lớn và RTT nhỏ.

5. Đánh giá hiệu năng

Dựa trên các kết quả nghiên cứu của các mô hình quản lý hàng đợi phần trên, chúng tôi tiến hành cài đặt các mô hình trên phần mềm mô phỏng NS-2

5.1. Mô hình mạng mô phỏng

Mô hình mạng trong suốt quá trình thực nghiệm được mô tả theo mô hình sau:



Hình 1. Mô hình mô phỏng

Trong mô phỏng, chúng tôi sử dụng N luồng TCP và M luồng UDP (giá trị của N, M có thể thay đổi để tiện cho việc đánh giá). Các đường truyền từ nguồn TCP và UDP đến nút cổ chai và từ nút cổ chai đến các đích đều có băng thông là 100Mbps, còn độ trễ thay đổi từ 1 đến 20ms.

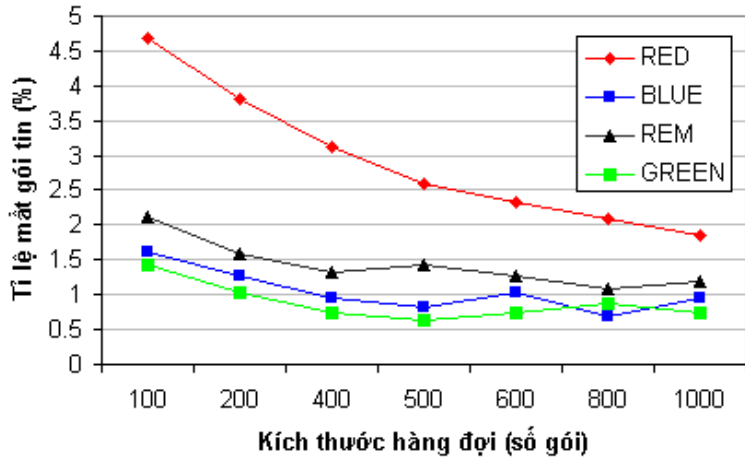
Đường truyền cổ chai trong kịch bản là liên kết giữa hai bộ định tuyến. Chúng tôi đặt băng thông tại đường truyền này là 45Mbps và độ trễ là 20ms. Bộ định tuyến tại nút thắt cổ chai được cài đặt các thuật toán để đánh giá. Kích thước hàng đợi tại nút thắt cổ chai thay đổi từng trường hợp để tiện đánh giá.

Ngoài ra, các tham số như kích thước gói tin của cả luồng TCP và UDP đều được thiết lập là 1000 Byte, kích thước cửa sổ TCP là 2000 gói và tốc độ truyền của luồng UDP thay đổi trong mô phỏng để làm cơ sở đánh giá. Thời gian chọn làm mô phỏng là 60 giây.

5.2. Đánh giá các cơ chế theo kích thước hàng đợi thay đổi

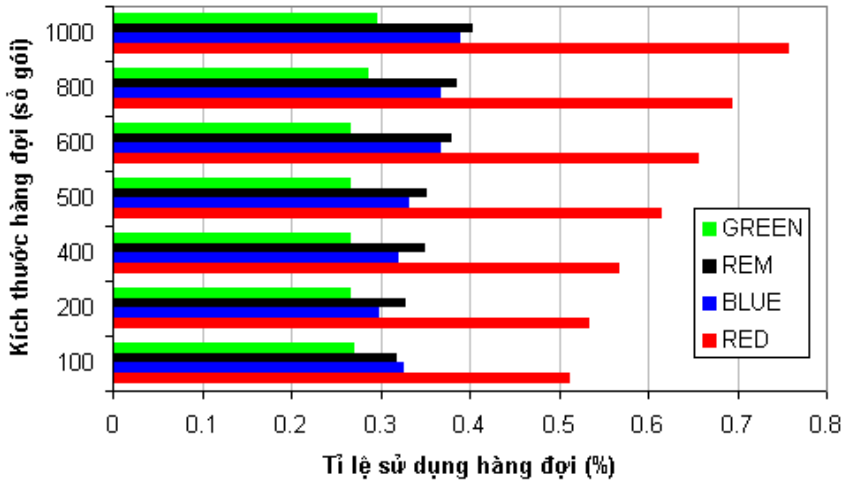
Thử nghiệm với 200 kết nối TCP qua đường truyền cổ chai, lần lượt cài đặt cơ chế quản lý hàng đợi tích cực RED, BLUE, REM và GREEN tại bộ định tuyến thắt cổ chai với giới hạn kích thước hàng đợi thay đổi từ 100 đến 1000 gói tin. Chúng tôi thu được kết quả như sau:

Xác suất mất gói tin: Từ đồ thị Hình 2, chúng tôi thấy rằng khi kích thước hàng đợi tăng thì xác suất mất gói tin tại hàng đợi của hầu hết các thuật toán đều giảm. Riêng RED có gia tốc giảm lớn nhất, giảm từ 4.7% xuống 2%, nhưng RED luôn có xác suất mất gói tin lớn nhất. GREEN và BLUE có tỉ lệ mất gói tin nhỏ nhất, kể cả khi kích thước hàng đợi nhỏ.



Hình 2. Tỷ lệ mất gói tin của các thuật toán theo kích thước hàng đợi

Kích thước hàng đợi trung bình: Dựa vào kết quả mô phỏng và đồ thị Hình 3 biểu diễn tỷ lệ của sử dụng kích thước hàng đợi của các thuật toán, chúng tôi nhận thấy rằng: RED là thuật toán sử dụng hàng đợi nhiều nhất và GREEN sử dụng hàng đợi ít nhất trong tất cả các trường hợp.



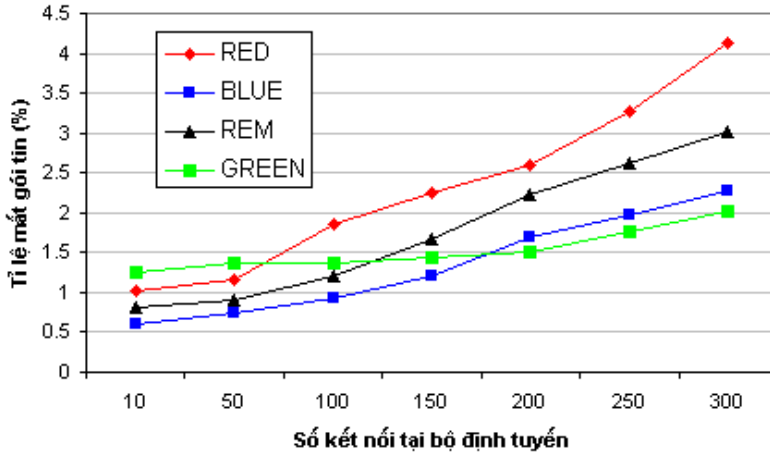
Hình 3. Hàng đợi trung bình của các thuật toán theo kích thước hàng đợi

5.3. Đánh giá các cơ chế dựa trên tải nạp thay đổi

Thử nghiệm với kích thước hàng đợi 200 tại cổ chai, cài đặt cơ chế quản lý hàng đợi tích cực RED, BLUE, REM và GREEN tại bộ định tuyến thắt cổ chai với số lượng kết nối TCP thay đổi từ 10 đến 300 để thay đổi tải ở nguồn phát. Chúng tôi thu được kết quả như sau:

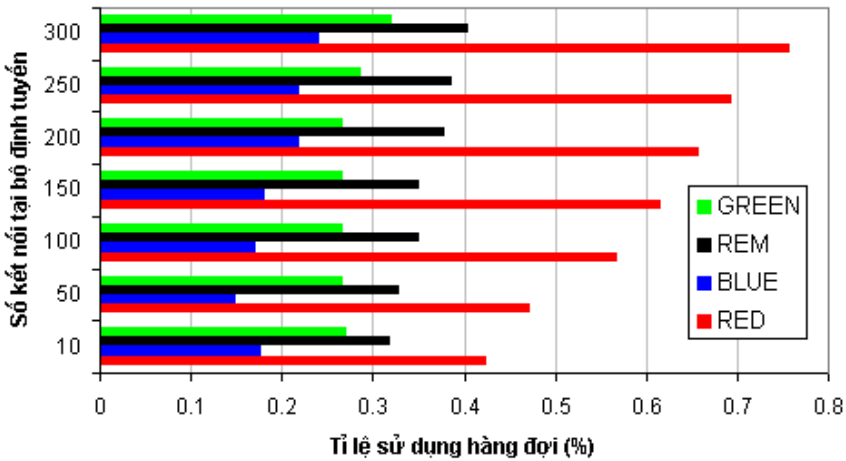
Xác suất mất gói tin: Đồ thị trên Hình 4 thể hiện xác suất mất gói tin của các cơ chế. Khi số lượng kết nối nhỏ (ít tải) thì tỷ lệ mất gói tin của các thuật toán rất thấp, hầu hết dưới 1%. Nhưng khi số lượng kết nối tăng (tăng tải) thì tỷ lệ mất gói tin của các thuật toán tăng dần. RED thể hiện rõ nhất về sự mất gói tin khi số lượng kết nối tăng. Ngược

lại, GREEN luôn thể hiện sự ổn định khi tăng tải, và GREEN có tỉ lệ mất gói tin thấp nhất khi số lượng kết nối lên đến 300.



Hình 4. Tỉ lệ mất gói tin của thuật toán theo số luồng

Kích thước hàng đợi: Hình 5 thể hiện kích thước hàng đợi trung bình của các thuật toán. RED có kích thước hàng đợi lớn nhất, đặc biệt khi số lượng kết nối lớn thì kích thước hàng đợi trung bình của RED gần xấp xỉ với ngưỡng max_{th} . Thuật toán BLUE có kích thước hàng đợi trung bình khá nhỏ (chiếm ít hơn 30% giới hạn không gian hàng đợi).



Hình 5. Kích thước hàng đợi trung bình các thuật toán khi số luồng

5.4. So sánh và phân lớp các cơ chế quản lý hàng đợi như sau

Chúng ta thấy rằng, hầu hết các cơ chế sử dụng thước đo về tắc nghẽn để phát hiện và giải quyết vấn đề này dựa trên kích thước hàng đợi, tải nạp. Ngoài ra, cũng có một số thuật toán bổ sung thêm thông tin về luồng để phát hiện tắc nghẽn. Để tổng hợp sự phân lớp này, chúng tôi đưa ra bảng tóm tắt đặc tính các cơ chế trong *Bảng 1* như sau:

Bảng 1. Phân lớp các thuật toán

Cơ chế		RED	BLUE	REM	GREEN	
Phân lớp	Dựa vào kích thước hàng đợi	✓	×	✓	✓	
	Dựa vào tải nạp	×	✓	✓	✓	
	Dựa vào hiệu suất sử dụng đường truyền	×	✓	✓	✓	
	Dựa vào thông tin luồng	×	×	×	×	
Điều khiển luồng	Thích nghi	✓	✓	✓	✓	
	Không thích nghi	×	×	×	✓	
		Mạnh	×	×	×	✓
		Yếu	×	×	×	✓

Trong các thuật toán đã đề xuất, RED chỉ dựa vào kích thước hàng đợi để phát hiện tắc nghẽn mà không sử dụng thông tin về luồng. RED không phát hiện và hạn chế được luồng không hồi đáp. Để khắc phục được vấn đề này, BLUE sử dụng sự kiện mất gói tin (dựa vào tải nạp) và lịch sử hiệu suất sử dụng đường truyền để điều khiển tắc nghẽn. Tuy vậy, BLUE cũng không phát hiện và hạn chế được luồng không hồi đáp. REM và GREEN dựa vào kích thước hàng đợi và tải nạp để điều khiển tắc nghẽn, nhưng GREEN tỏ ra thuật toán mạnh và phù hợp với nhiều môi trường mạng. Tất cả các thuật toán đều cung cấp hiệu quả sử dụng đường truyền cao.

Cùng với việc phân lớp các thuật toán, chúng tôi đưa bảng tóm tắt việc so sánh hiệu năng của các thuật toán để từ đó đưa ra giải pháp ứng dụng chúng vào môi trường mạng:

Bảng 2. Đánh giá hiệu năng các thuật toán

Cơ chế	RED	BLUE	REM	GREEN
Thông lượng	Vừa	Cao	Cao	Cao
Tỉ lệ mất gói tin	Cao	Thấp	Vừa	Thấp
Không gian bộ đệm	Lớn	Nhỏ	Vừa	Nhỏ

6. Kết luận

Qua bài báo chúng tôi đã phân tích được hoạt động của các cơ chế quản lý hàng đợi khác nhau. Từ đó đã đưa ra được các bảng so sánh và đánh giá mức độ hiệu quả của từng cơ chế quản lý hàng đợi khác nhau, như từ Bảng 1 và 2 chúng ta thấy rằng: Nếu kích thước bộ đệm lớn, chúng ta có thể dùng cơ chế RED, còn nếu kích thước bộ đệm nhỏ chúng ta có thể dùng BLUE. Tuy nhiên, GREEN là thuật toán đảm bảo sự công bằng giữa các luồng, có khả năng hạn chế luồng không thích nghi (bảo vệ luồng yếu, chống lại luồng mạnh và hạn chế luồng không hồi đáp). Do đó, với môi trường mạng có

hỗ trợ các luồng này, nếu có không gian bộ đệm nhỏ và tải lớn thì có thể sử dụng GREEN. Vì vậy, có thể nói GREEN là cơ chế nâng cao hiệu năng TCP tốt nhất. Trong thời gian tiếp theo chúng tôi sẽ nghiên cứu cải tiến sự hoạt động đồng bộ của các cơ chế quản lý hàng đợi tích cực trong môi trường mạng thế hệ mới.

TÀI LIỆU THAM KHẢO

- [1]. Apu Kapadia, Wu-chun Feng and Roy H. Campbell, *GREEN: A TCP Equation-Based Approach to Active Queue Management*, U.S Department of Energy through Los Alamos National Laboratory contract W-7045-ENG-36, 2011.
- [2]. Arash Dana and Ahmad Malekloo, *Performance Comparison between Active and Passive Queue Management*, IJCSI International Journal of Computer Science Issues, Vol. 7 Issue 3 No 5, (2010).
- [3]. Arkaitz Bitorika, Mathieu Robin, Meriel Huggard and Mc Goldrick, *A Comparative Study of Active Queue Management Schemes*, Department of Computer Science Trinity College Dublin, Ireland, 2011.
- [4]. Bartek Peter Wydrowski, *Techniques in Internet Congestion Control*, Electrical and Electronic Engineering Department The University of Melbourne, 2003.
- [5]. Bartek Wydrowski and Moshe Zukerman, *GREEN: An Active Queue Management Algorithm for a Self*, ARC Special Research Centre for Ultra-Broadband Information Networks, EEE Department, The University of Melbourne, Parkville, Vic. 3010, Australia, 2010.
- [6]. G.Thiruchelvi and J.Raja, *A Survey On Active Queue Management Mechanisms*, IJCSNS International Journal of Computer Science and Network Security, Vol.8 No.12, (2008).
- [7]. Go Hasegawa and Masayuki Murata, *Analysis of dynamic behaviors of many TCP connections sharing Tail-Drop - RED routers*, CybermediaCenter, Osaka University, Japan, 2003.
- [8]. Himanshu Chandra, Ajay Agarwal and T. Velmurugan, *Analysis of Active Queue Management Algorithms & Their Implementation for TCP/IP Networks Using OPNET Simulation Tool*, International Journal of Computer Applications (0975 – 8887), Vol.6 No.11, (2010).
- [9]. Jae Chung and Mark Claypool, *Analysis of Active Queue Management*, Computer Science Department Worcester Polytechnic Institute Worcester, MA 01609, USA, 2008.
- [10]. K.Chitra and G.Padamavathi, *Classification and Performance of AQM-Based*

Schemes for Congestion Avoidance, (IJCSIS) *International Journal of Computer Science and Information Security*, Vol.8 No.1, (2010).

- [11]. Kevin Fall and Kannan, *The ns Manual*, A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2010.
- [12]. Long Le, Kevin Jeffay, F. Donelson Smith, *A Loss and Queuing-Delay Controller for Router Buffer Management*, Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), 2006.
- [13]. Long Le, Jay Aikat, Kevin Jeffay and F. Donelson Smith, *The Effects of Active Queue Management and Explicit Congestion Notification on Web Performance*, IEEE/ACM Transactions On Networking, Vol.15 No.6, (2007).
- [14]. Minseok Kwon and Sonia Fahmy, *A Comparison of Load-based and Queue-based Active Queue Management Algorithms*, Dept. Of computer Science, Purdue University, West Lafayette, IN 47906-1398, USA, 2010.
- [15]. Mujdat Soyuturk, *Design and Analysis of TCP/IP Networks*, M.Soyuturk, Design and Analysis of TCP/IP Networks, G.Y.T.E., Spring, 2011.
- [16]. Sanjeeva Athuraliya, Steven H. Low, Victor H. Li and Qinghe Yin, *REM Active Queue Management*, IEEE Network, (2006).
- [17]. Shakeel Ahmad, Adli Mustafa, Bashir Ahmad, Arjamand Bano And Als Ammarraie Hosam, *Comparative Study Of Congestion Control Techniques In High Speed Networks*, IJCSIS, Vol.6 No.2, (2009).
- [18]. T.Bhaskar Reddy, Ali Ahammed and Reshma Banu, *Performance Comparison of Active Queue Management Techniques*, IJCSNS International Journal of Computer Science and Network Security, Vol.9, No.2, (2009).
- [19]. Wu-chang Feng, Kang G. Shin, Dilip D. Kandlur, Debanjan Saha, *The BLUE Active Queue Management Algorithms*, IEEE, (2006).
- [20]. Wu-chang, Fengy Dilip, D. Kandlurz Debanjan and Sahaz Kang G. Shin, *BLUE-A New Class of Active Queue Management Algorithms*, Department of EECS Network Systems Department University of Michigan, (2009).

**PERFORMANCE EVALUATION OF SOME ACTIVE QUEUE
MANAGEMENT MECHANISMS BASED ON QUEUE SIZE AND LOADING****Nguyen Kim Quoc¹, Vo Thanh Tu²***¹Nguyen Tat Thanh University**²College of Sciences, Hue University*

Abstract. The Internet is facing an explosion in the number of connected computers and the diversity of application layers deployed on it. Active queue management is one of the solutions to congestion control on the Internet. In recent years, network researchers have proposed various mechanisms for active queue management. This paper will evaluate the performance of the mechanisms of active queue management basing on queue size and traffic load, in order to classify and implement these mechanisms to adapt to different network environments.