

# SỬ DỤNG GIẢI PHÁP TÍCH HỢP CÁC CƠ CHẾ QUẢN LÝ HÀNG ĐỢI VÀ BÁO NHẬN ĐỂ TRÁNH TẮC NGHẼN TRÊN MẠNG INTERNET

NGUYỄN THỨC HẢI<sup>1</sup>, VÕ THANH TÚ<sup>2</sup>

<sup>1</sup>*Khoa CNTT, Trường Đại học Bách khoa Hà Nội*

<sup>2</sup>*Khoa CNTT, Trường Đại học Khoa học Huế*

**Abstract.** Integrating model queue management schemes with acknowledgement (ACK) control have been proposed in [5]. Based on this model in this paper we focus on analyzing RED queue parameters and ACK delay time to avoid congestion in the Internet.

**Tóm tắt.** Mô hình tích hợp các cơ chế quản lý hàng đợi và điều khiển báo nhận (ACK) đã được đề xuất trong [5]. Dựa trên mô hình đó, chúng tôi tập trung phân tích các tham số hàng đợi RED với thời gian trễ ACK nhằm tránh tắc nghẽn trên mạng Internet.

## 1. MỞ ĐẦU

Trên hệ thống Internet ngày nay, sự bùng nổ thông tin đã dẫn đến lưu lượng thông tin trên đường truyền ngày càng lớn, làm cho sự xử lý quá tải tại các nút mạng trung tâm càng cao, không còn bộ đệm tự do, nó bỏ qua những gói tin mới đến khi mà khả năng xử lý của các nút yếu sẽ dẫn đến tắc nghẽn ở nút trung tâm.

Tổ chức IETF đã đưa ra cơ chế thông báo nhằm tránh tắc nghẽn (ECN: Explicit Congestion Notification) [3] để kết hợp dò tìm tắc nghẽn trên mạng. Khi có dấu hiệu lưu lượng gói tin vượt khả năng xử lý thì thiết lập cơ chế làm trễ các gói báo nhận (DA: Delay ACK) [2] tại trạm nhận, làm ảnh hưởng rất lớn đến hiệu năng của mạng. Một cơ chế tránh tắc nghẽn khác là dựa vào việc tổ chức lại hàng đợi phù hợp hơn cơ chế tổ chức hàng đợi DropTail (DT) là cơ chế huỷ bỏ sớm ngẫu nhiên (RED: Random Early Detection) [1, 4]. Trong bài báo này chúng tôi đề xuất giải pháp tích hợp các các tham số hàng đợi RED với thời gian trễ ACK dựa trên mô hình tích hợp các cơ chế quản lý hàng đợi và điều khiển báo nhận (ACK) đã được đề xuất trong [5]. Bằng phương pháp mô phỏng hoạt động của hệ thống mạng trên, chúng tôi thử nghiệm cài đặt DT hoặc ECN/RED tại hàng đợi và DA tại trạm nhận thông qua sự hiệu chỉnh các trọng số trong cơ chế quản lý hàng đợi tích cực, kết hợp với cơ chế ECN để tách pha làm chậm ACK trong khoảng thời gian bùng nổ thông tin tại điểm thắt nút, làm hạn chế tình trạng dẫn đến tắc nghẽn xảy ra tại nút mạng và ổn định lưu thông trên hệ thống.

## 2. CƠ CHẾ ĐIỀU KHIỂN TRÁNH TẮC NGHẼN

Cơ chế điều khiển được xét đến ở đây là quá trình điều khiển gói báo nhận hoặc điều

chính kích thước cửa sổ trượt. Việc sử dụng cửa sổ trượt có kích thước thay đổi là hỗ trợ điều khiển tốc độ truyền dữ liệu cũng như truyền đáng tin cậy. Để tránh việc nhận nhiều gói tin hơn khả năng lưu trữ, nơi nhận sẽ gửi đi thông báo cửa sổ nhỏ hơn và ngược lại. Trường hợp xấu nhất, nơi nhận sẽ gửi đi thông báo cửa sổ có kích thước là 0 để ngưng tất cả việc truyền. Nhưng việc nhiều lần ngưng truyền với những đợt ngắn do tràn hàng đợi tạm thời là không cần thiết, điều này làm tăng sự dao động thông lượng. Trong trường hợp này điều khiển gói báo nhận ACK hợp lý giữa luồng gói tin đến và cơ chế xử lý hàng đợi tích cực tại nút nhận trung tâm là hiệu quả hơn.

Thông thường các điểm đầu cuối không nhận biết sự tắc nghẽn và tại sao chúng xảy ra. Bởi vì tắc nghẽn là do độ trễ gia tăng, nên hầu hết các phần mềm giao thức sử dụng bộ đếm thời gian và truyền lại. Việc truyền lại có ảnh hưởng lớn đến hệ thống, vì nó sẽ làm tăng thêm sự nghẽn mạch và đến một lúc nào đó mạng sẽ trở nên vô dụng. Giải quyết toàn diện vấn đề này là vô cùng phức tạp, liên quan nhiều tầng giao thức khác nhau với nhiều dịch vụ khác nhau. Thông thường việc điều khiển sự tắc nghẽn được thực hiện qua 3 bước như sau:

*Bước 1:* Làm chủ hệ thống để phát hiện khi nào và xảy ra ở đâu. Khi xác định được tắc nghẽn ở đâu, lúc đó bước thứ 2 sẽ được thực hiện.

*Bước 2:* Chuyển thông tin đến những nút mạng (router) khác mà ở đó có thể tiến hành giải quyết được công việc đồng thời thông báo tắc nghẽn (ECN: Explicit Congestion Notification)[3] cho các router khác. Tất nhiên, các gói tin phụ sẽ làm tăng tải vào thời điểm nhiều tải không cần thiết.

Một phương pháp khác là máy chủ hay router gửi các gói tin thăm dò để biết rõ ràng về sự tắc nghẽn. Thông tin có thể được sử dụng chỉ lưu thông quanh khu vực có sự cố, cải tiến cơ chế xử lý gói tin tại hàng đợi của các nút mạng trung tâm theo một trật tự ưu tiên phù hợp.

*Bước 3:* Khi nhận được thông tin về sự tắc nghẽn, máy chủ có những hành động thích hợp để giảm sự tắc nghẽn như: Sắp xếp lại tuyến đường truyền tin, hạn chế không cho truyền gói tin vào những đường xảy ra tắc nghẽn. Đối với trạm đích cách để tác động sự tắc nghẽn là giảm tải nghĩa là phủ nhận dịch vụ với nơi sử dụng, giảm bớt dịch vụ từ các trạm gửi đến hoặc cải tiến giao thức điều khiển phù hợp.

Vì vậy cần thực hiện các giải pháp để hiệu chỉnh như sau:

- + Đo các tham số mạng liên quan để theo dõi chu kỳ của các quá trình thực hiện trên mạng. Dùng Timer biết được gói tin cần bao lâu sẽ báo nhận dựa trên RTT, từ khi bắt đầu yêu cầu gói tin đến cho đến lúc nhận được ACK. Đồng thời ghi lại biến cố xảy ra (dựa vào trình tự ACK) như là hiện tượng mất gói tin.
- + Theo dõi các tiến trình đang tiến hành dựa trên việc phát liên tục các gói tin và tích lũy các gói trả lời ACK để tăng hiệu quả trả lời trong trạng thái hàng đợi bình thường.
- + Xây dựng cơ chế điều chỉnh tăng, giảm thời gian đáp ACK để điều khiển lưu lượng gói tin đến.
- + Xử lý nhanh các gói tin dựa trên chiến lược quản lý hàng đợi tích cực tại các bộ đệm của nút mạng, xây dựng giao thức ở đầu cuối cho các mô hình mạng đáp ứng với hiệu suất cao.
- + Thiết kế hệ thống phải đảm bảo lưu lượng truyền tối đa nhưng không dẫn đến tắc nghẽn, không để xảy ra hiện tượng thắt nút cổ chai do sự chênh lệch tốc độ trong thiết kế mạng Internet, đặc biệt xử lý tốc độ cao tại vị trí các nút mạng trung tâm qua kết nối bộ định tuyến.

### 3. ẢNH HƯỞNG CỦA CÁC THAM SỐ TRONG QUẢN LÝ HÀNG ĐỢI VÀ ĐIỀU KHIỂN ACK

#### 3.1. Cơ chế hàng đợi RED

Khi các gói tin gửi đến nhanh hơn là chúng được chuyển đi thì hàng đợi sẽ dài ra hay các gói tin chuyển đến chậm hơn thì hàng đợi thu ngắn lại. Nhưng vì bộ nhớ là hữu hạn, hàng đợi không thể dài ra quá. Vì vậy để quản lý hàng đợi bị tràn phần mềm của bộ định tuyến sử dụng chiến lược cắt bớt phần đuôi (DropTail). Chiến lược này có ảnh hưởng đáng kể với TCP, làm cho TCP đi vào trạng thái khởi động chậm, nghĩa là giảm bớt tốc độ truyền cho tới khi TCP bắt đầu nhận các báo nhận và gia tăng kích thước cửa sổ nghẽn mạch. Ngoài ra việc huỷ các gói tin đến sau khi hàng đợi bị đầy có thể ảnh hưởng đến toàn bộ Internet, điều này đòi hỏi phải có một mô hình khác để thay thế đó là mô hình huỷ bỏ sớm ngẫu nhiên [6, 7], thường được gọi tắt là RED. Cơ chế này dựa vào việc tính toán xác suất gói tin rơi trong giới hạn của ngưỡng kích thước hàng đợi  $\max_{th}$ ,  $\min_{th}$  và tính toán kích thước hàng đợi trung bình  $\hat{k}$  tại nút mạng. Giá trị của  $\hat{k}$  được cập nhật mỗi khi có datagram gửi đến theo công thức sau:

$$\hat{k} := (1 - \omega) \times \hat{k} + \omega \times k \quad (1)$$

Với  $\omega$  là hệ số,  $0 \leq \omega \leq 1$ ,  $k$ : kích thước hàng đợi tức thì.

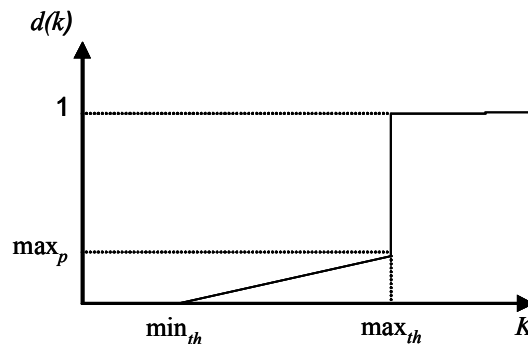
Cơ chế quản lý bộ đệm RED dựa vào gói tin bị đánh rơi với một xác suất tăng dần theo hàm  $p(\hat{k})$  của kích thước hàng đợi trung bình  $\hat{k}$ , có kích thước bộ đệm của  $K$  gói tin. Việc tính hàm xác suất  $p$  là một trong những giai đoạn phức tạp của RED. Thuật toán RED như sau:

Nếu  $\min_{th} < \hat{k} < \max_{th}$  thì hàm xác suất gói tin rơi  $p(\hat{k})$  là:

$$p(\hat{k}) = \frac{(\hat{k} - \min_{th}) \max_p}{\max_{th} - \min_{th}} = \frac{((1 - \omega) \times \hat{k} + \omega \times k - \min_{th}) \max_p}{\max_{th} - \min_{th}} \quad (2)$$

Hiệu số  $(\max_{th} - \min_{th})$  càng cao thì xác suất rơi gói tin thấp, khi hàng đợi rỗng thì xác suất huỷ gói tin ngẫu nhiên  $p(\hat{k}) = 0$ .

$p(\hat{k}) = 0$  nếu  $\hat{k} < \min_{th}$ ,  $p(\hat{k}) = 1$  nếu  $\hat{k} > \max_{th}$ .



Hình 1. Mô hình hàng đợi RED

Mặc dù mô hình tuyến tính hình thành nên cơ sở của phép tính xác suất cho RED, cần

phải có những hiệu chỉnh trọng số để tránh tình trạng phản ứng “quá vội”. Sở dĩ cần có những thay đổi là bởi vì giao thông trên mạng là theo từng “đợt”, và gây ra những dao động quá nhanh của hàng đợi trong bộ định tuyến. Nếu RED sử dụng một mô hình tuyến tính đơn giản, những gói tin đến sau trong mỗi đợt sẽ bị gán xác suất cao cho khả năng bị loại bỏ (vì chúng đến khi hàng đợi đã có nhiều gói tin). Nếu gặp một đợt (các gói tin) ngắn, ít có khả năng bị loại bớt gói tin khi hàng đợi chưa bị đầy thì bộ định tuyến không nên huỷ bỏ những gói tin này một cách không cần thiết, làm như thế sẽ tác động xấu đến hiệu suất của TCP. Dĩ nhiên, RED không thể tạm hoãn việc huỷ bỏ vô thời hạn, vì một đợt dài sẽ làm đầy hàng đợi, kết quả chẳng khác nào chiến lược “cắt bớt phần đuôi” (DT), và sẽ gây ra ảnh hưởng đến toàn bộ Internet.

### 3.2. Cơ chế điều khiển ACK

Để dung nạp được những độ trễ khác nhau trong môi trường Internet, TCP sử dụng giải thuật truyền lại có tính năng hiệu chỉnh để theo dõi những độ trễ trên môi trường kết nối và điều chỉnh giá trị bộ đếm thời gian của nó một cách tương ứng.

Vì vậy các giao thức Internet cần có sự kiểm soát tốc độ truyền end-to-end để điều chỉnh lượng dữ liệu, nếu không phần mềm giao thức sẽ bị quá tải với cơ chế cửa sổ trượt, đồng thời đối với các hệ thống trung gian (bộ định tuyến) phải kiểm soát tốc độ truyền, cho phép một nguồn có thể gửi nhiều hơn nhận của một máy, khi xảy ra quá tải tại máy trung tâm phải sử dụng cơ chế kiểm soát sự nghẽn mạch bằng nhiều giải pháp khác nhau như: tự huỷ rồi truyền lại khi quá thời gian sống hoặc truyền lại có tính năng hiệu chỉnh dựa vào mẫu thời gian khứ hồi (RTT) trung bình mới  $\hat{r}$ :

$$\hat{r} := (\alpha \times \hat{r}) + (1 - \alpha) \times r, \quad 0 \leq \alpha \leq 1, \quad (3)$$

$r$  là thời gian khứ hồi tức thì, nghĩa là, khi  $\alpha$  gần 1 thì giá trị trọng số trung bình ít ảnh hưởng bởi những thay đổi trong một thời gian ngắn,  $\alpha$  gần 0 thì thay đổi tức thì theo những thay đổi của sự trễ.

Trên thực tế, khi sử dụng một tập các giá trị được phát sinh ngẫu nhiên để minh họa việc tính thời gian hết hạn bị thay đổi khi thời gian RTT biến đổi, người ta thấy rằng RTT thay đổi (tăng, giảm), có thể làm tăng giá trị của bộ đếm thời gian tăng nhanh chóng khi độ trễ tăng nhưng lại không giảm đi nhanh chóng khi độ trễ giảm nên việc xây dựng giải thuật trở nên phức tạp và không phù hợp trong môi trường có thay đổi lớn độ trễ. Vì vậy, chúng ta cần tính khoảng cách thời gian giữa hai ACK phù hợp với trọng số hàng đợi  $\omega$  để điều khiển luồng tại trạm phát khi tại nút trung tâm có độ trễ lớn.

Giả sử có 2 gói xác nhận  $Ack1, Ack2$  và  $\Delta a = Ack2 - Ack1 > 1$ , tương ứng ở thời điểm  $t1, t2$ . Như vậy khi  $Ack2$  được truyền đến thiết bị gửi thì có ít nhất  $\Delta a$  gói dữ liệu được truyền. Cửa sổ tắc nghẽn tăng lên nhiều nhất là 1, độc lập với  $\Delta a$ . Bộ điều chỉnh tốc độ tại trạm nhận (AR Agent) giải quyết vấn đề này bằng cách tạo ra các gói ACK giả, cung cấp cho thiết bị gửi theo một tốc độ không đổi làm giá trị cửa sổ phát tăng lên phù hợp trên mỗi kênh phát. Nghĩa là mỗi AR có một tham số ngưỡng  $ack\_thresh$ , quyết định số ACK được thêm vào. Như vậy với  $\Delta a$  gói dữ liệu tương ứng gói ACK là  $Ack1$  và  $Ack2$ , ta thêm ACK giả bằng  $\Delta a / ack\_thresh - 2$  gói ACK.

Gọi  $\Delta t$  là tốc độ trung bình các gói ACK trong bộ điều chỉnh tốc độ AR Agent, nó phụ thuộc vào  $\Delta a, t1, t2$  và số liệu thống kê. Gọi  $ack\_interval$  là khoảng cách 2 ACK liên tiếp và được tính theo công thức:

$$ack\_interval = (ack\_thresh/\Delta a) \times \Delta t$$

Bộ AR Agent đo tần suất xuất hiện các gói ACK đầu vào, mỗi AR dùng một bộ điều chỉnh tốc độ để tính giá trị trung bình của tần suất này. Giả sử đầu ra của bộ điều chỉnh tốc độ là  $\Delta t$  (phụ thuộc  $\alpha$ ), khi đó nếu  $ack\_interval = \Delta t$ , tức là thực hiện phép chuyển 1-1, không giữ chậm, khi đó hiệu suất mạng hoàn toàn phụ thuộc vào gói ACK của thiết bị nhận. Với khoảng cách trung bình được tính toán  $\Delta t$ , ứng với 2 gói ACK được xử lý tại thiết bị phát nhằm mềm hoá quá trình xử lý tắc nghẽn cũng như điều chỉnh khung cửa sổ phát một cách tuyến tính, liên tục, thoả mãn biểu thức:  $\Delta a/ack\_thresh = \Delta t/ack\_interval$ .

Trong trường hợp hàng đợi bị tràn cần phải tăng khoảng cách  $Ack(ack\_interval)$  để làm chậm việc truyền báo nhận đến trạm phát, từ đó giảm được luồng gói tin đến nút mạng, hạn chế xảy ra tắc nghẽn.

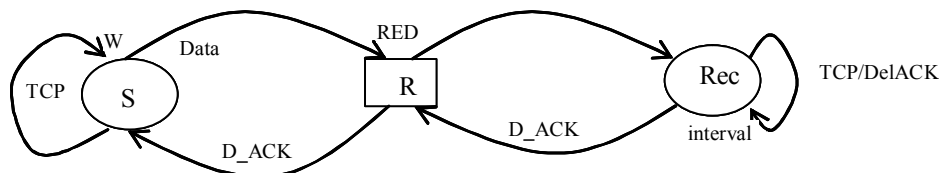
#### 4. SỰ TÍCH HỢP GIỮA HÀNG ĐỢI RED VÀ CƠ CHẾ CHẬM BÁO NHẬN ACK

##### 4.1. Phương pháp luận

Để mở rộng cơ chế điều khiển tắc nghẽn của các gói xác nhận ACK đối với giao thức TCP tại bộ đệm hàng đợi máy nhận, chúng ta tích hợp thuật toán hàng đợi RED để phát hiện sớm tắc nghẽn tiềm ẩn (khả năng vượt quá ngưỡng) có thể xảy ra bằng cách tính toán kích thước trung bình của hàng đợi  $k$  trong khoảng thời gian ngay trước đó, khi đó một gói ACK hay gói dữ liệu sẽ được đánh dấu bit  $ECN^+$  trong trường Options của TCP. Như vậy khi nhận gói có  $ECN^+$ , bên nhận làm chậm phát ACK, bên gửi sẽ giảm tần suất phát gói dữ liệu.

Nghĩa là, tại mỗi thiết bị nhận người ta duy trì một hệ số động **DelAck** gọi là *hệ số giữ chậm ACK* nhằm xác định số gói dữ liệu tương ứng một gói xác nhận ACK. Khi một gói ACK có đánh dấu  $ECN^+$ , giá trị DelAck được nhân lên nhiều lần nên làm giảm gói ACK trên đường truyền làm giảm tần suất phát gói dữ liệu. Ngược lại nếu trong khoảng thời gian khứ hồi RTT không có gói ACK bị đánh dấu ECN hay đánh dấu  $ECN^-$  thì giá trị DelAck sẽ giảm xuống một cách tuyến tính, tức là giảm giá trị  $ack\_interval$ . Thiết bị nhận truyền ít nhất 1 gói ACK cho mỗi khung cửa sổ phát của thiết bị gửi. Nếu không thiết bị phát sẽ rơi vào trạng thái chờ cho đến khi gói ACK được phát miễn cưỡng theo nhịp đồng hồ phát của thiết bị nhận (200 ms).

Vì vậy, tích hợp thuật toán RED tại bộ định tuyến với việc điều khiển luồng dữ liệu đến nhờ sự điều khiển của luồng báo nhận ACK có ảnh hưởng rất lớn đến quá trình truyền từ đầu cuối đến đầu cuối. Thông qua cơ chế báo hiệu tắc nghẽn ECN khi hàng đợi có xác suất huỷ bỏ cao hay bộ đệm gần bị đầy thì chiến lược làm chậm ACK tại trạm nhận đầu cuối làm giảm luồng đến từ trạm gửi, dẫn đến cân bằng được hàng đợi RED.



Hình 2. Mô hình trạng thái điều chỉnh cân bằng lưu thông

Dựa vào công thức (1) và (3), chúng ta thấy vị trí thời gian khứ hồi trung bình cũ tương ứng với kích thước hàng đợi trung bình mới và thời gian khứ hồi trung bình mới tương ứng với kích thước hàng đợi trung bình cũ. Vì vậy, thời gian khứ hồi trung bình có tác động mạnh đến tốc độ đến các gói tin, làm thay đổi (tăng, giảm) kích thước hàng đợi trung bình, nghĩa là tỷ lệ giữa RTT và  $\hat{k}$  phải được tính toán cân bằng để điều chỉnh hệ thống.

Ở đây 2 hệ số  $\omega, \alpha$  có ảnh hưởng mạnh đến tỷ lệ này ta gọi là trọng số  $\omega$ , trọng số  $\alpha$  trong đoạn  $[0,1]$ .

*Nhận xét 1:* Tỷ lệ tích hợp của  $\hat{k}/\hat{r}$  là cân bằng hợp lý khi và chỉ khi  $\alpha, \omega$  thay đổi tương ứng tỷ lệ nghịch với nhau trong đoạn  $[0,1]$ .

*Nhận xét 2:* Để hàng đợi RED có xác suất huỷ bỏ thấp khi hàng đợi bị đầy trong thời gian ngắn thì trọng số  $\omega \rightarrow 0$  của phương trình  $\hat{k}$ , dẫn đến sự thay đổi RTT, nghĩa là trọng số  $\alpha$  của  $\hat{r}, \alpha \rightarrow 1$  để thời gian khứ hồi ít thay đổi thì sẽ đáp ứng với sự trễ ngắn của hàng đợi, hạn chế sự khởi động chậm của TCP nên hạn chế việc làm giảm thông lượng truyền tin.

Ngược lại RED có xác suất huỷ bỏ cao khi hàng đợi bị đầy trong thời gian dài thì trọng số  $\omega$  của  $\hat{k} : \omega \rightarrow 1$  dẫn đến RTT thay đổi theo, nghĩa là trọng số  $\alpha$  của  $\hat{r} : \alpha \rightarrow 0$  để thời gian khứ hồi thay đổi tức thì, đáp ứng với sự trễ kéo dài của hàng đợi.

Trong mạng truyền thông hai chiều thì thời gian khứ hồi (RTT) chính là thời gian gói tin đi từ nguồn đến đích  $t_{data}$  cộng với thời gian báo nhận từ đích tới nguồn nên để thay đổi RTT ta chỉ cần thay đổi  $\theta$ , từ đó điều khiển được luồng dữ liệu đến nút mạng, hạn chế số lượng các gói tin đến xếp hàng khi gần bị đầy trong cơ chế RED. Từ đây ta có phương trình tính giá trị thời gian báo nhận trung bình được thiết lập như sau:

$$\hat{t}_{ACK} := (\theta \times \hat{t}_{ACK}) + (1 - \theta) \times t_{ACK}, \quad 0 \leq \theta \leq 1 \quad (4)$$

$\theta$  là hệ số, nghĩa là, khi  $\theta$  gần 1 thì giá trị  $\hat{t}_{ACK}$  trung bình ít ảnh hưởng,  $\theta$  gần 0 thì  $\hat{t}_{ACK}$  thay đổi tức thì theo những thay đổi của sự trễ.

Vì vậy, khi có tín hiệu báo nhận tắc nghẽn ECN đến bên nhận thì giao thức TCP/DelAck sẽ giữ chậm trả lời ACK tương ứng với sự thay đổi của độ trễ  $\hat{t}_{ACK}/\hat{k}$  là tỷ lệ giữ chậm hợp lý cân bằng với độ trễ tại hàng đợi RED.

Khi kích thước hàng đợi tức thời  $k < \min_{th}$  thì không huỷ bỏ gói tin theo xác suất  $p$  đồng thời gửi tín hiệu  $ECN^-$  đến bên nhận để sử dụng chiến lược phục hồi nhanh ACK (AR), tăng quá trình khởi động nhanh.

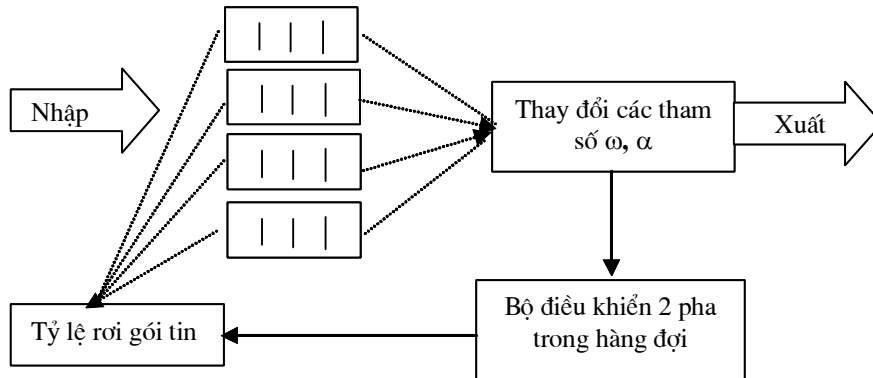
Khi kích thước hàng đợi tức thời  $k > \max_{th}$  thì huỷ bỏ gói tin theo xác suất  $p$  đồng thời gửi tín hiệu  $ECN^+$  đến bên nhận để sử dụng chiến lược điều khiển tắc nghẽn ACK (ACC) và làm trễ ACK để giảm tiến trình các gói tin đến Router.

## 4.2. Mô hình cơ chế điều chỉnh tại nút mạng trung tâm

Mô hình điều khiển: Xét mô hình có nhiều dòng xếp hàng đến tại một bộ định tuyến trung tâm được thiết lập cơ chế hàng đợi RED. Khi một gói tin phải bị rút thì modul điều khiển sẽ chọn ra một trật tự ưu tiên tắc theo RED+DelACK.

Vì vậy, chúng tôi đề xuất cơ chế điều chỉnh tự động 2 pha cho phù hợp với sự thay đổi lưu thông trên mạng ở các thời điểm nhạy cảm với sự tắc nghẽn, nghĩa là thời điểm xảy ra tràn hàng đợi trong khoảng thời gian dài vượt quá khả năng xử lý tại nút mạng thông qua cơ chế báo hiệu tắc nghẽn ECN.

Các lớp hàng đợi logic tại bộ định tuyến



Hình 3. Mô hình điều khiển 2 pha

Phương pháp tự động điều chỉnh 2 pha:

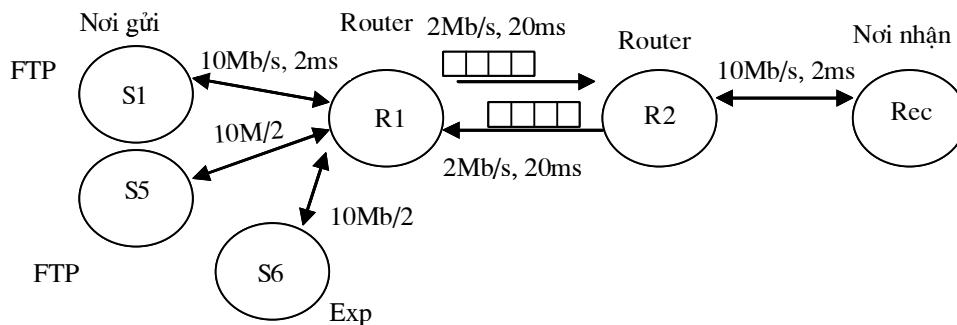
Việc đưa ra tín hiệu  $ECN^+$  và  $ECN^-$  nhằm cải tiến lưu lượng của giao thức. Các biện pháp đề xuất nhằm mục đích thông qua việc phát hiện tắc nghẽn làm giảm vấn đề mất mát gói tin. Các bộ định tuyến sẽ gửi các ECN để chỉ rõ sự tắc nghẽn hoặc hết nguy cơ tắc nghẽn, hình thành nên 2 pha: 1 pha điều khiển luồng ACK làm hạn chế luồng đến bộ định tuyến và 1 pha thông báo việc xếp hàng tại hàng đợi đang ngăn lại, để tăng phát ACK, nhanh chóng tăng luồng dữ liệu đến như sau:

*Pha 1:* Gán  $\omega$  lớn ( $\omega \rightarrow 1$ ) khi gặp đợt dài, thông báo tín hiệu thông báo rõ tắc nghẽn  $ECN^+$  đến bên nhận để thực hiện cơ chế giảm ACK bằng cách giảm  $\alpha$  ( $\alpha \rightarrow 0$ ), từ đó làm giảm luồng đến hàng đợi RED. Cài đặt phần mềm trên Router để thay đổi cơ chế hàng đợi RED động (thay đổi trọng số  $\omega$ ).

*Pha 2:* Gán  $\omega$  nhỏ ( $\omega \rightarrow 0$ ) khi gặp đợt ngắn, thông báo cho bên gửi  $ECN^-$  đến bên nhận để thực hiện phát tăng ACK bằng cách tăng  $\alpha$  ( $\alpha \rightarrow 1$ ), theo cơ chế tái tạo gói ACK (AR), làm tăng nhanh luồng đến RED sau quá trình khởi động chậm.

#### 4.3. Mô phỏng và đánh giá

Sử dụng NS-2 là công cụ mô phỏng mạng cài đặt trên hệ điều hành Linux. Giả thiết mô hình thiết kế mạng mô phỏng được xây dựng [5] như sau:



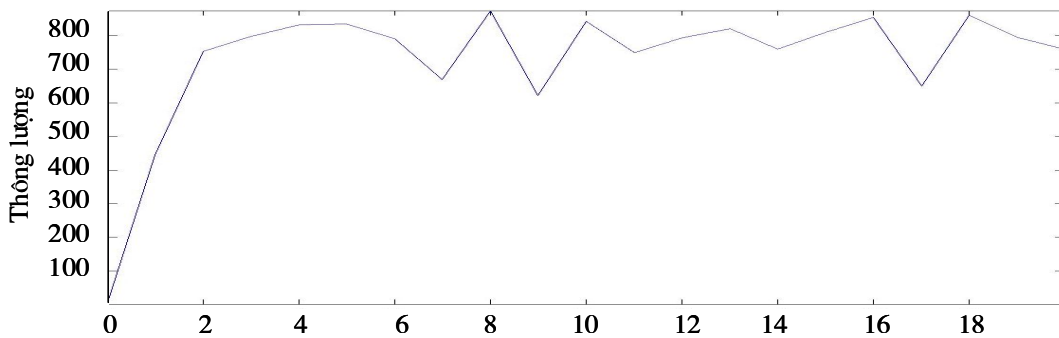
Hình 4. Mô hình hệ thống mô phỏng

Hệ thống có 5 nguồn gửi tin S1 đến S5 theo dịch vụ FTP cài đặt TCP và 1 nguồn gửi

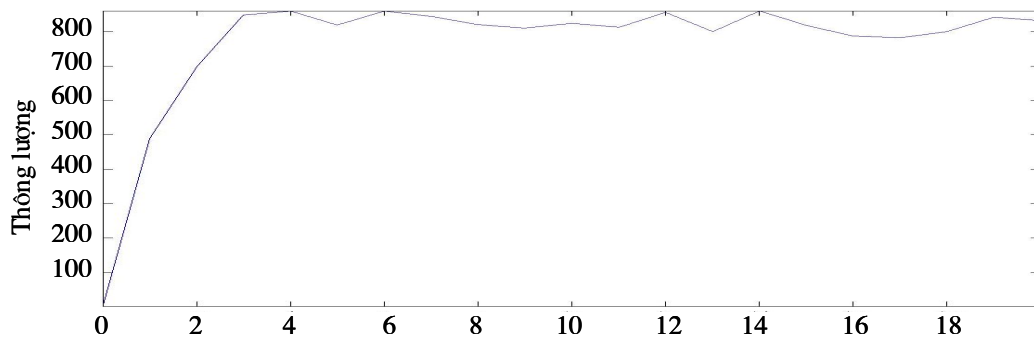
tin S6 theo dịch vụ có cấu hình thay đổi tỷ suất truyền thích nghi (on/off) trên giao thức không yêu cầu báo nhận UDP. Bên nhận (Rec) được cài đặt giao thức giao vận TCP. Cơ chế điều khiển hàng đợi được tổ chức tại hàng đợi ra của Router (R1). Mô phỏng ứng dụng trong mạng có băng thông không đồng bộ giữa các cung truyền, đoạn từ R1(Router) đến R2 (Router) có băng thông 2 Mb/s, độ trễ đường truyền là 20 ms (tương ứng với tốc độ mạng WAN), băng thông các cung còn lại là 10Mb/s và độ trễ là 2 ms (tương ứng với tốc độ của mạng LAN hiện tại).

Giới hạn hàng đợi tại các nút là 20 gói. Thứ tự mô phỏng tổ chức hàng đợi tràn ở các bộ định tuyến trung tâm. Đồng hồ TCP là 100 ms. Quy luật dùng hàng đợi Drop Tail cho Router R2 và các trạm từ đầu cuối đến đầu cuối. Sử dụng hàng đợi RED tại đầu ra R1 (ngưỡng tối thiểu  $min_{th} = 5$ , ngưỡng tối đa  $max_{th} = 15$ ,  $\omega = 0.002$ ).

Qua quá trình theo dõi mô phỏng NS của mô hình trên, chúng tôi đã ghi lại toàn bộ hoạt động trong thời gian mô phỏng 20 s, thời gian bắt đầu gửi của S1 đến S5 lần lượt cách nhau 0.1 ms, kết quả ghi lại được phân tích bằng chương trình TRGRAPH. Tại các thời gian mô phỏng 7s, 9s, 17s (hình 5) trên mô hình cài đặt cơ chế DropTail+TCP/DelAck thông lượng toàn bộ hệ thống giảm mạnh do có sự tràn hàng đợi liên tục tại nút mạng trung tâm. Vấn đề này đã được giải quyết thể hiện qua (hình 6) mô hình có sự tích hợp giữa cơ chế RED và TCP/DelAck, đồng thời tỷ lệ gói tin rơi giảm 46,2%, tỷ lệ gói tin mất giảm 36,8%. Với phương pháp tích hợp này tỏ ra có hiệu quả cho cơ chế tránh tắc nghẽn đồng thời tăng được thông lượng ở những thời điểm hàng đợi bị đầy vượt quá khả năng lưu trữ, xử lý tại nút mạng trung tâm và không cần thiết cài đặt ở các nút có lưu lượng gói tin thấp, số kết nối đến không làm vượt khả năng xử lý tại nút mạng.



Hình 5. Biểu đồ thông lượng của hệ thống có cài đặt DropTail+TCP/DelAck



Hình 6. Biểu đồ thông lượng của hệ thống có cài đặt RED+TCP/DelAck



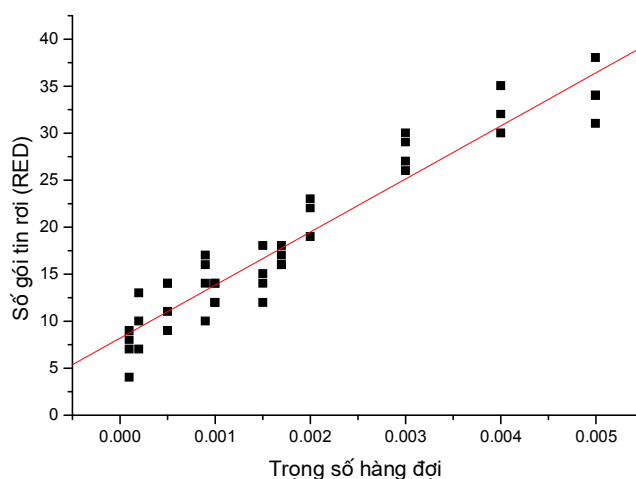
#### 4.4. Tính tương quan và hồi qui giữa các tham số tích hợp

Xét mô hình (hình 4) có sự phối hợp giữa giao thức TCP/DelAck tại đầu cuối Server(Rec) có cơ chế điều chỉnh gói báo nhận với cơ chế hàng đợi RED tại R1, có kết hợp tín hiệu báo tắc nghẽn ECN để làm thay đổi tín hiệu báo nhận ACK, điều chỉnh luồng dữ liệu đến tại nút mạng trung tâm theo mô hình mạng hàng đợi đóng. Trạm nhận (Rec) được cài đặt giao thức TCP/DelAck cải tiến tránh tắc nghẽn bằng phương pháp điều chỉnh độ trễ gói biên nhận ACK theo tham số  $\alpha$  để điều chỉnh tăng hoặc giảm luồng gói tin đến R1 từ trạm phát khi có báo hiệu ECN.

Qua kết quả phân tích bằng mô phỏng với sự thay đổi của các tham số  $\omega$ ,  $\alpha$  và dựa vào cơ sở lý thuyết xác suất thống kê, chúng tôi tính được hệ số tương quan giữa trọng số hàng đợi  $\omega$  của RED và số gói tin rơi trung bình là 0,959 và hệ số tương quan giữa khoảng cách báo nhận ACK với tổng số gói tin được phát đi là (-0,753).

Theo lý thuyết tương quan và hồi qui đây là các hệ số tương quan (correlation) mạnh. Từ đó ta xác định được phương trình hồi qui (regression) tuyến tính đa biến của số gói tin rơi trung bình phụ thuộc theo hai tham số độc lập  $\omega$  và  $\alpha$  là:

$$\hat{d} = 9,45 + 5645,12\omega - 33,05\alpha$$



Hình 7. Biểu đồ hồi qui tuyến tính

Dựa vào hình 7, chứng minh được mối tương quan tuyến tính giữa các tham số trong mô hình tích hợp để điều chỉnh lưu thông trên mạng, đảm bảo lưu thông trên mạng được trong suốt, tăng hiệu năng chung toàn mạng. Ta nhận thấy trọng số hàng đợi  $\omega$  càng gần về 0 thì số gói tin rơi giảm và hệ số của hai tham số  $\omega$  và  $\alpha$  là ngược dấu nhau phù hợp với lý thuyết đã phân tích ở mục 4. Do đó việc đề xuất tích hợp giữa cơ chế RED và DelAck là hợp lý và có hiệu quả.

## 5. KẾT LUẬN

Qua bài báo này chúng tôi đã nghiên cứu được các mô hình điều khiển lưu thông trên mạng, đặc biệt chú trọng đến các cơ chế phòng tránh tắc nghẽn, tăng hiệu năng chung toàn

mạng. Trên cơ sở phân tích bộ giao thức TCP/IP và cơ chế quản lý hàng đợi của mạng Internet. Chúng tôi đã đề xuất được mô hình điều khiển lưu thông trên mạng một cách hợp lý dựa vào tích hợp cơ chế hàng đợi RED và thời gian trễ gói báo nhận ACK với sự điều chỉnh các tham số hợp lý. Thông qua việc sử dụng phương pháp mô phỏng để thử nghiệm và những kết quả thu được, tìm ra được mối tương quan và ước lượng quan hệ (hồi qui) giữa các tham số làm thay đổi luồng lưu thông trên mạng, điều chỉnh hợp lý để tránh tắc nghẽn xảy ra ở những thời điểm khác nhau. Trong thời gian tới chúng tôi sẽ tiếp tục nghiên cứu ảnh hưởng của các dịch vụ khác biệt trên mô hình để tăng thêm hiệu năng chung toàn mạng.

### TÀI LIỆU THAM KHẢO

- [1] Agharebparast, V.C.M. Leung, Improving the performance of RED deployment on a class based queue with shared buffers, Global Telecommunications Conference, 2001. GLOBECOM '01. *IEEE* (4) (2001) 2363–2367.
- [2] D.Towsley, M. Yamamoto, J.F. Kurose, A Delay Analysis of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols, *Proceedings of IEEE INFORCOM' 97* (1997) 480–488.
- [3] K. Ramakrishnan, S. Floyd, A Proposal to Add Explicit Congestion Notification (ECN) to IP, RFC 2481, *IETF*, Jan. (1999).
- [4] M. Borden, V. Firoiu, Queue Management for Congestion Control, *IEEE INFOCOM*, (2000).
- [5] Nguyễn Thức Hải, Võ Thanh Tú, Tích hợp cơ chế điều khiển gói báo nhận và quản lý hàng đợi trong điều khiển lưu thông mạng, Chuyên san “Các công trình nghiên cứu triển khai Viễn thông và Công nghệ thông tin”, *Tạp chí Bưu chính Viễn thông* (11) (2004) 41–48.
- [6] S. Floyd, V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Trans. Net.* 1 (4) (1993) 397–413.
- [7] Sally Floyd, A Report on Some Recent Developments in TCP Congestion Control, *IEEE Magazine*, (2001).

Nhận bài ngày 10 - 4 - 2004