

ỨNG DỤNG GIẢI THUẬT DI TRUYỀN MỜ TRONG BÀI TOÁN QUẢN LÝ HÀNG ĐỢI TÍCH CỰC AQM

NGUYỄN PHƯƠNG HUY, LÊ HOÀNG, LÊ BÁ DŨNG

1. GIỚI THIỆU CHUNG

Quản lý hàng đợi là một nhóm tổ hợp các phương pháp quản lý hàng đợi (queue management) và lập lịch trình (scheduling), đây là một trong những cơ chế cung cấp chất lượng dịch vụ (QoS). Quản lý bộ đệm quyết định việc phân phối bộ đệm và loại bỏ các gói đến theo một cách thức được quyết định trước khi cần thiết hoặc thích hợp. Trong khi đó lập lịch cho phép quản lý băng thông phân phối cho các luồng hay nói cách khác là nó quyết định xem gói nào sẽ được đưa ra từ hàng đợi nào. FIFO (First-In First-Out) là thuật toán lập lịch trình phổ biến nhất. Gói mất có thể được coi là một hình thức thông báo ẩn của tắc nghẽn.

Do có rất nhiều thuật toán rất hiệu quả được đưa ra trong kĩ thuật quản lý hàng đợi, với quản lý bộ đệm có các thuật toán: RED, Blue, PBS (chia sẻ bộ đệm từng phần), cắt-đuôi (DT-Drop Tail)... quá trình lập lịch gồm: RR, WFQ, EDF (Earliest Deadline First)... Quản lý hàng đợi dựa trên luồng gồm RED và xRED, dựa trên tốc độ gồm: Blue, PI, KT, Bat, Green, Purple [1, 2, 6 - 10]...

Khi tỉ lệ các gói tin đến cao hơn tỉ lệ gói tin đi của router, kích thước hàng đợi sẽ tăng lên, cuối cùng vượt quá không gian cho phép của bộ đệm. Một khi bộ đệm đầy, một số gói tin sẽ bị mất, cắt đuôi (DT) là nguyên tắc mất gói phổ biến nhất, nếu một gói tin đến và lúc đó hàng đợi đầy, gói tin sẽ bị loại bỏ. Vì vậy, cơ chế DT tương tác kém với các cơ chế điều khiển tắc nghẽn của TCP và dẫn đến hiệu suất thấp. Active Queue Management (AQM) là phương pháp chủ động thông báo với bên gửi khi mới bắt đầu tắc nghẽn trước khi xảy ra tràn bộ đệm. Bằng cách sử dụng cơ chế AQM, bên gửi được thông báo sớm về tắc nghẽn và có thể phản ứng phù hợp [6, 9, 10, 12].

Random Early Detection (RED) [7, 9, 12] là cơ chế AQM, được đề xuất để giải quyết các vấn đề gây ra bởi DT nêu trên. RED sử dụng sự ngẫu nhiên để giải quyết cả 2 vấn đề khoá đầu ra và hàng đợi đầy một cách hiệu quả, mà không đòi hỏi bất kỳ thay đổi nào tại các máy chủ kết cuối. Mục đích của RED là để tránh tràn hàng đợi bằng cách loại bỏ các gói ngẫu nhiên.

RED thiết lập ngưỡng mất gói cực tiểu min_{th} và cực đại max_{th} . Nếu kích thước hàng đợi trung bình (avg) vượt quá min_{th} , RED bắt đầu bỏ các gói tin dựa trên một xác suất tùy thuộc vào avg . Nếu avg vượt quá max_{th} , mọi gói tin sau đó đều bị loại bỏ [8, 9]. Do sử dụng kích thước hàng đợi như là chỉ thị tắc nghẽn cho nên không thể biểu thị hoàn toàn mức độ tắc nghẽn. Mặt khác, kích thước hàng đợi trung bình thay đổi theo mức độ tắc nghẽn và việc thiết lập các thông số dẫn đến trễ xếp hàng của RED là quá nhạy cảm với tải lưu lượng và việc thiết lập các thông số [11].

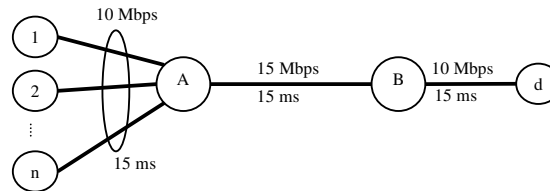
Để giải quyết các vấn đề của cơ chế RED truyền thống, các tác giả phát triển thuật toán GA-Fuzzy-AQM dựa trên nền RED đã được đề xuất. Trong khi Fuzzy-AQM đã được chứng minh là tốt hơn AQM đơn thuần [10]. Ý tưởng chính của GA-Fuzzy-AQM là đưa giải thuật di

truyền mờ vào thuật toán RED để tối ưu hoá các thông số của luật điều khiển mờ nhằm đạt được hiệu suất tốt hơn các biến thể Fuzzy-AQM đơn thuần [10, 12].

2. GIẢI THUẬT DI TRUYỀN MỜ TRONG AQM

2.1. Mô hình (Topo) hệ thống

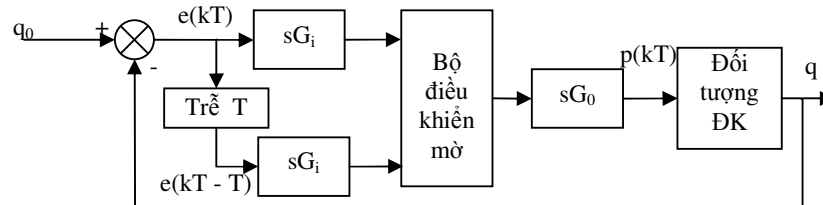
Sơ đồ tổng quát giả sử có một cấu hình mạng như hình 1:



Hình 1. Biểu diễn nút cổ chai từ A sang B

Hình 1 biểu diễn nút cổ chai thể hiện qua kết nối giữa các router A và B. Giữa A và B có tốc độ truyền dữ liệu là 15 Mbps (khoảng 15000 gói/s). Mỗi một gói tin chứa khoảng 125 bytes với thời gian trễ khoảng 15 ms. Trên tất cả các đường truyền đến A có tốc độ 10 Mbps và độ trễ là 15 ms với độ lớn của hàng đợi là 300 gói. Hàng đợi A được thực hiện theo các cơ chế đã đề cập ở trên. Việc xây dựng giải thuật di truyền mờ cho điều khiển AQM sẽ thực hiện theo hai bước, đó là xây dựng hệ điều khiển mờ và sau đó tinh chỉnh hệ điều khiển sử dụng giải thuật di truyền.

2.2. Hệ điều khiển mờ cho AQM



Hình 2. Mô hình hệ thống điều khiển mờ cho AQM

Xây dựng thuật toán mờ, thuật toán giải thuật di truyền mờ cho AQM có nhiều điểm khác biệt so với việc xây dựng các thuật toán PI hoặc PID. Nếu với thuật toán PI và I khó có thể dự báo dựa trên các sai số sẽ xảy ra trong tương lai thì thuật toán PID cho thấy đây là thuật toán truyền thống được dùng rất nhiều trong công nghiệp. Nhưng với thuật toán điều khiển mờ và thuật toán di truyền mờ dùng để điều khiển hệ AQM sẽ mang lại một hình ảnh mới với các đặc điểm nổi trội sau:

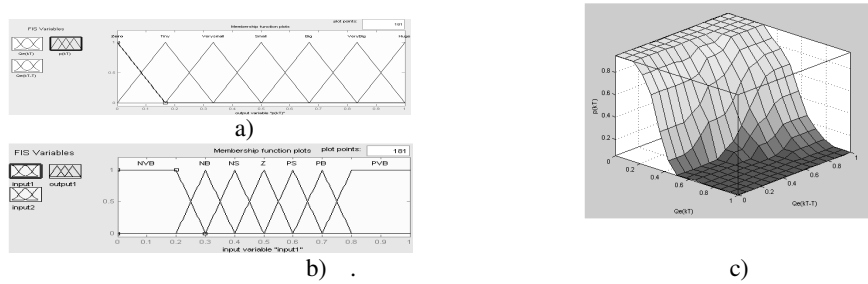
- + Có khả năng đưa các tri thức của các chuyên gia vào điều khiển hệ AQM.

- + Bộ điều khiển sử dụng thuật toán mờ, giải thuật di truyền mờ hết sức mềm dẻo.
- + Có khả năng tìm tối ưu toàn cục.
- + Không cần thiết phải xây dựng mô hình toán học cho hệ thống điều khiển
- + Không nhất thiết phải có một vùng nhớ đệm lớn...

Hệ luật cho bộ điều khiển mờ được mô tả trong bảng 1, và các giá trị ngôn ngữ cho các biến vào và các biến ra thể hiện trên hình 3a, 3b và bảng 1, [3].

Bảng 1. Biểu diễn hệ luật mờ cho bộ điều khiển mờ

$p(kT)$		$q_{c_err}(kT - T)$						
		NVB	NB	NS	Z	PS	PB	PVB
$q_{err}(kT)$	NVB	H	H	H	H	H	H	H
	NB	B	B	B	VB	VB	H	H
	NS	T	VS	S	S	B	VB	VB
	Z	Z	Z	Z	T	VS	S	B
	PS	Z	Z	Z	Z	T	T	VS
	PB	Z	Z	Z	Z	Z	Z	T
	PVB	Z	Z	Z	Z	Z	Z	Z



Hình 3. Các đầu vào a), đầu ra b) và mặt suy diễn c) của bộ điều khiển mờ

2.3 Giải thuật di truyền mờ cho AQM

Giải thuật di truyền sử dụng cho tìm kiếm tối ưu các dạng hàm thuộc được thực hiện theo các bước sau: Sinh sản, Chọn lọc, Lai ghép và Đột biến.

2.3.1 Mã hoá

Mã hoá là quá trình chuyển đổi một mô hình mờ vào các thông số trong không gian một chiều của cá thể. Nói một cách khác cá thể (chuỗi giá trị) chứa các thông số cho việc xây dựng mô hình mờ. Ở đây mô hình mờ với các luật “if..and...then ...” sẽ có phân điều kiện là các hàm thuộc dạng tam giác với độ rộng phải tâm và độ rộng trái của nó. Như vậy một cá thể chứa đựng các thông tin trong giải thuật di truyền bao gồm:

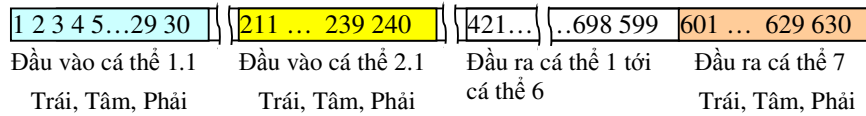
- Các giá trị vị trí cho xây dựng hàm thuộc.
- Độ rộng phải, trái, tâm của hàm thuộc.
- Các giá trị thực qua giải mờ trong phần kết quả của luật hình 3.

Quá trình mã hoá sử dụng phép ánh xạ tuyến tính có dạng:

$$C_{ij} = C_{min} + \frac{b}{2^L - 1} (C_{max} - C_{min}) \quad (1)$$

Với: b là giá trị dưới dạng thập phân được chuyển đổi sang dạng nhị phân; L là độ dài của chuỗi nhị phân; C_{max} , C_{min} là giá trị max, min của gen được định nghĩa bởi người sử dụng

Quá trình mã hoá một gen liên quan đến các cá thể. Giả sử mỗi một thông số có độ dài 10 bits thì một gen sẽ có tổng số $10 \times 3 \times 7 \times 3 = 630$ bits, như vậy hệ mờ với dạng luật if...and ...then sẽ có dạng:



Hình 4. Một nhiệm sắc thể cho chuỗi mã hoá

Theo hình 4 là quan hệ giữa các giá trị của chuỗi và cấu trúc của thông số vào của hàm thuộc đối với 1 nhiệm sắc thể. Đó chính là tọa độ của mỗi một hàm thuộc tương ứng liên quan đến tỉ lệ được chia ra trong tổng các giá trị, ở đây sử dụng ba lớp giá trị của một cá thể. Trong thực tế tồn tại nhiều cá thể, các cá thể nhận được một cách ngẫu nhiên và quá trình được thực hiện theo các phép toán di truyền.

Từ hình 4 ta thấy, điểm trái của hàm thuộc một thứ nhất là các bit (1, 2, ..., 10), tâm của hàm thuộc một là các bit (11, 12, ..., 20), điểm phải của hàm thuộc một thứ nhất là các bit (21, 22, ..., 30), có 7 hàm thuộc cho biến vào một, như vậy 30×7 bit đầu là gen của biến vào một. Điểm trái của hàm thuộc hai thứ nhất là các bit (211, 212, ..., 220), tâm của hàm thuộc hai là các bit (221, 222, ..., 230), điểm phải của hàm thuộc hai là các bit (231, 232, ..., 240), tương tự có 7 giá trị cho biến vào hai, như vậy 30×7 bit tiếp theo là gen của biến vào hai của phần điều kiện. Đối với các gen (421...630) sẽ cho ta các giá trị của 7 biến đầu ra. Từ đó xác định được các giá trị thực, sau đó sẽ được tính như phép giải mờ theo phương pháp trọng tâm. Như vậy vị trí của các gen trong không gian vào sẽ được thay đổi theo quá trình thực hiện giải thuật.

2.3.2. Lai tạo

Phép toán lai tạo được thực hiện thông qua thay đổi vị trí được sắp xếp cho một hàm thuộc. Hai cá thể cha mẹ trong quần thể được chọn lựa một cách ngẫu nhiên. Quá trình lai tạo cũng thể hiện thông qua việc chọn lựa giữa các giá trị cho các vị trí của hàm thuộc. Cấu trúc sẽ thay đổi giữa các cá thể thông qua các điểm lai tạo. Các thế hệ con cháu sẽ thừa hưởng từ cá thể A (cha) và cá thể B (mẹ). Như vậy các giá trị của hàm thuộc sẽ thay đổi liên quan đến thế hệ con cháu. Các con cháu sẽ kế thừa các đặc trưng tốt của bố mẹ thông qua quá trình lai tạo.

2.3.3 Đột biến

Các quá trình đột biến sẽ xảy ra với các cá thể thông qua quá trình lai tạo với xác suất P_m . Quá trình đột biến ở đây là chọn lựa chọn thông qua thay đổi các giá trị các bit lên 1 hay xuống 0. Như vậy sẽ cho phép tạo các cá thể mới thông qua lai tạo và đột biến.

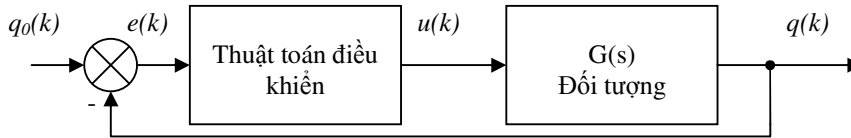
2.3.4 Hàm thích nghi

Hàm thích nghi dùng để đánh giá chất lượng các mô hình mờ, nó phản ánh quá trình chọn lọc tự nhiên theo một mức độ thích nghi nhất định. Ở đây các mô hình xây dựng phản ánh được các quá trình thay đổi thể hiện trên mức độ thích nghi của hệ thống, hay nói khác đi là mức độ thích nghi của các cá thể được tính theo:

$$f_m(k) = \exp(-[(\Delta\epsilon(k)/e(k)) - 1]^2) \quad (2)$$

trong đó: $\Delta\epsilon(k)$ là sai số giữa 2 thế hệ; $e(k)$ là sai lệch điều khiển.

Sơ đồ điều khiển AQM sử dụng thuật toán điều khiển tổng quát có thể thấy trên hình 5



Hình 5. Sơ đồ hệ thống điều khiển AQM tổng quát

3. THỰC NGHIỆM

3.1. Mô hình toán học của AQM

Hệ điều khiển sử dụng thuật toán điều khiển di truyền mờ sẽ được mô tả ở Hình 6 như sau:

Giả thiết là hệ thống được mô hình hóa với các thông số được tính theo [3],

$$G(s) = \frac{\frac{C^2}{2N} e^{-Rs}}{\left(s + \frac{2N}{R^2C}\right) \left(s + \frac{1}{R}\right)} \quad (3)$$

trong đó: C là tốc độ đường truyền (gói/s); q_0 là giá trị hàng đợi mong muốn; q là giá trị hàng đợi ở đầu ra; N tải (số phiên của TCP), R là RTT (round-trip time); $R = 2(q/C + T_p)$; T_p là giá trị xác định; P là xác suất mất gói/đánh dấu.

Để có thể thực hiện xem xét môi trường làm việc của mạng. Chúng ta lấy một ví dụ mô phỏng như sau: Hệ thống mạng máy tính hoạt động như TCP/IP với các thông số như dưới đây:

C_c là lưu lượng đường truyền với $C_c = 105$ gói/s = 100 Mbps, R_c là RTT = 0,03 s, $N_c = 30$ tải. Các thông số trên được xác định trong khoảng $C \in (0, C_c)$; $R \in (0, R_c)$; $N \in (N_c, +\infty)$;

Hàm truyền của hệ thống AQM dùng cho RED, được tính từ (3):

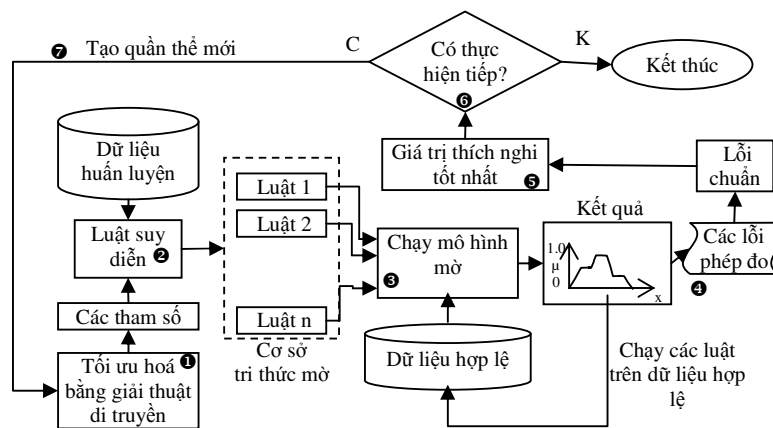
$$G_{dt}(s) = \frac{\frac{C^2}{2N} e^{-Rs}}{\left(s + \frac{2N}{R^2C}\right) \left(s + \frac{1}{R}\right)} = \frac{\frac{5}{3} \cdot 10^8 e^{-0,03s}}{\left(s + \frac{2}{3}\right) \left(s + \frac{100}{3}\right)} \quad (4)$$

$$W_{dt}(z) = \frac{5.10^{10}}{3.98} \left(\frac{\frac{3}{98}z}{z - e^{-\frac{2}{3}T}} - \frac{\frac{3}{98}z}{z - e^{-\frac{100}{3}T}} - \frac{\left(e^{-\frac{100}{3}T} z \right)}{\left(z - e^{-\frac{100}{3}T} \right)^2} \right). \quad (5)$$

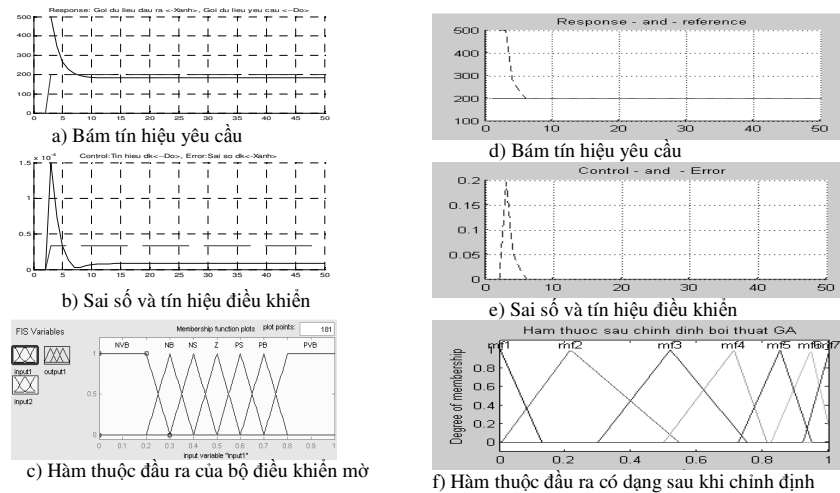
Biến đổi Z của đối tượng (4) sẽ cho (5). Thay số với cách chọn $T = 1$, ta có:

$$q(k+1) = 0,513417119q(k) - 3,42782 \cdot 10^{-15}q(k-1) + 5,72143 \cdot 10^{-30}q(k-2) + 2672933,733u(k) + 2,82558 \cdot 10^{-7}u(k-1) \quad (6)$$

3.2. Các kết quả mô phỏng



Hình 6. Mô hình chỉnh định mô hình mờ bằng GA [5]

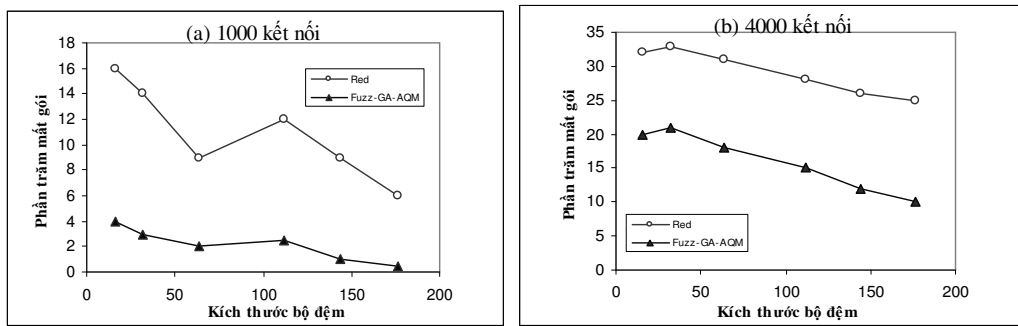


Hình 7. a), b), c) Các kết quả điều khiển chưa sử dụng giải thuật di truyền

Hình 7 biểu diễn gói dữ liệu đầu ra tiệm cận với gói dữ liệu yêu cầu $q_0 = 200$ với a) sử dụng thuật điều khiển mờ và với f) sử dụng giải thuật di truyền mờ. Cũng tương tự như vậy khi ta so sánh sai số giữa đầu ra với tín hiệu yêu cầu với thuật điều khiển mờ b) và giải thuật di truyền e). Sự thay đổi dạng hàm thuộc f) khi sử dụng giải thuật di truyền so với dạng hàm thuộc lúc ban đầu c).

3.3. Đánh giá

Để đánh giá hiệu suất của của Fuzz-GA-AQM so với RED, một thí nghiệm được thực hiện bằng cách sử dụng NS-2 cho mạng trong hình 1. Với mạng này, các kết nối được bật trong 2 giây và tắt trong 3 giây từ một trong các nút nguồn ($n_1, n_2, n_3, n_4, \dots, n_n$) đến nút đích (n_d). Ngoài ra, tất cả các nguồn cho phép hỗ trợ ECN và được bắt đầu ngẫu nhiên sau lần thứ hai của mô phỏng. Thống kê mất gói được đo sau 10 giây trong khoảng 100 giây mô phỏng. Thống kê mất gói cũng đo cho RED sử dụng cùng mạng và trong các điều kiện giống hệt nhau. Đối với các hàng đợi RED, min_{th} và max_{th} được thiết lập tương ứng 30% và 90% của kích thước hàng đợi. Cơ chế thông báo tắc nghẽn của RED đã được thực hiện tích cực nhất có thể bằng cách thiết lập max_p là 1. Với thí nghiệm này, đây là thiết lập lí tưởng của max_p vì nó giảm thiểu cả trễ xếp hàng và tỉ lệ mất gói tin cho RED.

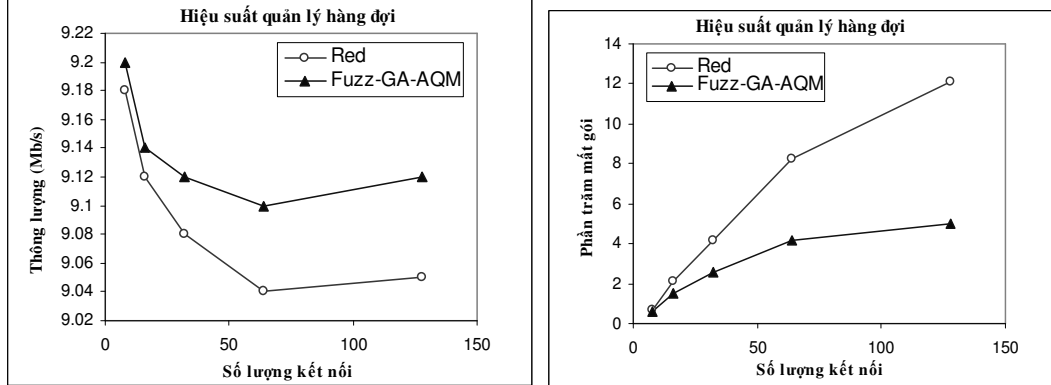


Hình 8. Tỉ lệ mất gói của RED, BLUE và Fuzz-GA-AQM

Hình 8 cho thấy tỉ lệ mất gói quan sát được trên các kích thước hàng đợi khác nhau đối với cả hai phương pháp Fuzz-GA-AQM và RED với 1.000 và 4.000 kết nối. Trong những thí nghiệm này, các hàng đợi tại kết nối cổ chai giữa A và B có kích thước từ 100 KB đến 1000 KB. Trễ hàng đợi 17,08 ms và 178ms như hình 8. Trong tất cả các thí nghiệm, kết nối vẫn duy trì trên 99,9% khả dụng. Theo Hình 7a cho thấy, với 1.000 kết nối. RED với tỉ lệ mất gói ở mức hai con số cũng như lượng giảm không gian đệm. Đối với Fuzz-GA-AQM cũng duy trì tỉ lệ mất gói khá nhỏ và tất nhiên tốt hơn so với RED.

Một điểm thú vị trong đồ thị mất gói RED thể hiện trong hình 8a là nó cho thấy lượng mất gói đáng kể với trễ bộ đệm khoảng 80 ms. Điều này xảy ra vì điểm hoạt động đặc biệt của RED khi chiều dài hàng đợi trung bình trên max_{th} tại tất cả các thời điểm. Tuy nhiên khi sử dụng GA cho thuật toán AQM (trong trường hợp này dùng GA cho RED) cho thấy Fuzz-GA-AQM cải thiện chất lượng đáng kể so với RED.

Theo hình 8b cho thấy, khi số lượng kết nối tăng lên 4.000, Fuzz-GA-AQM vẫn nhanh hơn đáng kể so với RED. Ngay cả bổ sung thêm không gian bộ đệm, RED vẫn không thể bằng với tỉ lệ mất gói của Fuzz-GA-AQM là 21,1 ms đối với bộ đệm tại kết nối cổ chai.



(a) Thông lượng so với số lượng kết nối; (b) Phần trăm mất gói so với số lượng kết nối

Hình 9. Hiệu suất quản lý hàng đợi của Red và Fuzz-GA-AQM

Hiệu suất quản lý hàng đợi được chỉ ra trên hình 9. Như hình 9a cho thấy, cả hai hàng đợi RED và Fuzz-GA-AQM đã được cấu hình tối ưu để duy trì mức thông lượng tương đối cao trên tất cả các tải. Tuy nhiên, vì RED định kỳ cho phép kết nối là khả dụng, thông lượng của nó vẫn thấp hơn Fuzz-GA-AQM. Thông lượng của Fuzz-GA-AQM cho thấy cải thiện đáng kể so với RED truyền thống. Như hình 8b cho thấy, RED duy trì mất gói tin ngày càng cao với số lượng kết nối tăng. Vì lưu lượng TCP tổng trở nên tích cực hơn giống như số lượng kết nối tăng lên, rất khó để RED duy trì tỉ lệ mất gói thấp. Biến động về độ dài hàng đợi xảy ra đột ngột làm cho thuật toán RED dao động giữa thời điểm duy trì đánh dấu và mất gói đến thời điểm đánh dấu tối thiểu và kết nối kém khả dụng. Như hình 9b) cho thấy, Fuzz-GA-AQM cũng duy trì tỉ lệ mất gói thấp hơn RED đáng kể.

Hệ quả quan trọng nhất của việc sử dụng Fuzz-GA-AQM là điều khiển tắc nghẽn có thể được thực hiện với lượng không gian đệm tối thiểu. Điều này làm giảm sự trễ kết cuối qua mạng, cải thiện hiệu quả của thuật toán điều khiển tắc nghẽn. Ngoài ra, bộ nhớ đệm yêu cầu nhỏ hơn cho phép thêm bộ nhớ để cấp cho các gói ưu tiên cao, và giải phóng bộ nhớ cho các chức năng router khác như lưu trữ các bảng định tuyến lớn. Cuối cùng, Fuzz-GA-AQM cho phép thiết bị định tuyến kế thừa để thực hiện tốt ngay cả với tài nguyên bộ nhớ hạn chế.

4. KẾT LUẬN

Bài báo, mới chỉ dừng lại ở việc sử dụng mô phỏng Fuzz-GA cho AQM (RED) nói chung. Fuzz-GA-AQM đã chứng tỏ đạt hiệu quả cao hơn so với AQM (RED) truyền thống, điều đó hứa hẹn có thể cải thiện hiệu suất hoạt động tối ưu bằng giải thuật di truyền cho một số cơ chế AQM mới như RED, BLUE, GREEN, PURPLE... Đặc biệt là với việc sử dụng mô hình kết hợp giữa hệ mờ và giải thuật di truyền có thể cho phép áp dụng tri thức chuyên gia có sẵn cho hệ thống, đồng thời có thể chỉnh định được các biến mờ tối ưu của hệ luật nhằm đạt kết quả tốt hơn.

TÀI LIỆU THAM KHẢO

1. D. Bauso, L. Giarré, G. Neglia - About the stability of active queue management mechanisms, American Control Conference 2004, Proceedings of the 2004, 02 May 2005, pp. 2-3.
2. C. Chrysostomou, A. Pitsillides, G. Hadjipollas, M. Polycarpou, A. Sekercioglu - Fuzzy logic control for active queue management in TCP/IP Networks, 12th. IEEE Mediterranean Conference on Control and Automation Kusadasi, Aydin, Turkey, (IEEE MED'04), 2004, pp. 2-3.
3. C. Chrysostomou, A. Pitsillides, G. Hadjipollas and others - Fuzzy logic congestion control in TCP/IP best-effort networks, University of Cyprus, Monash University Melbourne, Australia, 2007, pp. 2-5.
4. J. Chung, M. Claypool - Analysis of active queue management, Network Computing and Applications 2003, Second IEEE International Symposium on Issue Date: 16-18 April 2003, 28 May 2003.
5. Earl Cox - Fuzzy Model and Genetic Algorithms for Data Mining and Exploration, Morgan Kaufmann Publishers is an imprint of Elsevier, 2005, pp. 484.
6. G. D. Fatta, F. Hoffmann, G. L. Re, A. Urso - A Genetic Algorithm for the Design of a Fuzzy Controller for Active Queue Management, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions Aug 2003 **33** (2003) 3-8.
7. M. Jalili-Kharaajoo, F. Habibipour Roudsari, A. Dehestani, H. Hashemi Fesharaki - Adaptive Fuzzy Active Queue Management, Islamic Azad University, 2004, pp. 1-7.
8. T. Lehto, M. Laurikkala, T. Ekola, H. Koivisto - Behavior and performance of Fuzzy-RED AQM algorithm in best-effort networks, NEW2AN, Russia, 2004, pp. 2-6.
9. S. Özseskes - Evaluation of Active Queue Management Algorithms, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi Yıl: 4 Sayı: 7 Bahar, 2005, pp. 4-15.
10. M. H. Yaghmaee, H. AminToosi - A Fuzzy Based Active Queue Management Algorithm, In the proceeding of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS2003), 20-24 July, Montreal, Canada, 2003, pp. 458-462.
11. H. Zang - A Discrete time Model of TCP with AQM, Dong Hua University, China, Simon Fraser University, Aug 2004.
12. S. T. Zargar and M. H. Yaghmaee - Fuzzy Green: A modified TCP equation-based active queue management using fuzzy logic approach, International Journal of Computer Science and Network Security **6** (5A) (2006).

SUMMARY

AN APPLICATION OF FUZZY GENETIC ALGORITHM FOR ACTIVE QUEUE MANAGEMENT - AQM

Currently, all types of information services tend to integrate the Internet. To transport large amounts of data and better support for new Internet applications such as voice over IP and video on demand need to design algorithms for congestion control and efficient management of the queue. In this paper, we introduce a queue management algorithm - GA-Fuzzy-AQM, a fuzzy improvement for the well-known RED (Random Early Detection) AQM algorithm. Simulation results show that queue management algorithm GA-Fuzzy-AQM proposed has better performance than the traditional RED mechanism.

Địa chỉ:

Nguyễn Phương Huy, Lê Hoàng,
Đại học Kỹ thuật công nghiệp, Thái Nguyên.
Lê Bá Dũng,
Viện Công nghệ Thông tin.

Nhận bài ngày 15 tháng 9 năm 2009