

# ĐÁNH GIÁ CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN TẮC NGHẼN TRONG DỊCH VỤ TRUYỀN TẢI ĐA ĐƯỜNG

Evaluating congestion control methods in Multipath TCP

Khẩu Văn Nhật<sup>1</sup>

## Tóm tắt

Multipath TCP là giao thức mở rộng thêm các đặc điểm từ giao thức TCP, cho phép một kết nối TCP phân chia thành nhiều luồng con và phân bổ lưu lượng thông qua những luồng con riêng biệt. Mục tiêu của giao thức này là sử dụng nhiều đường đồng thời giữa hai thiết bị đầu cuối nhằm cải thiện đáng kể hiệu suất đường truyền. Để kiểm soát nghẽn trong multipath TCP, đã có các đề xuất dùng giải thuật điều khiển nghẽn dựa vào tổn thất và cả các giải thuật điều khiển nghẽn dựa vào độ trễ. Tuy nhiên, loại giải thuật điều khiển nghẽn nào là tốt hơn cho multipath TCP vẫn còn là điều cần làm rõ. Ngoài ra, hiệu quả của mỗi loại giải thuật điều khiển nghẽn trên multipath TCP chịu ảnh hưởng của các loại lưu lượng khác nhau như thế nào, chẳng hạn như ảnh hưởng giữa lưu lượng thời gian thực và phi thời gian thực. Tất cả những điều này sẽ được làm sáng tỏ trong bài báo này. Căn cứ vào các kết quả mô phỏng bằng công cụ NS-2, các đánh giá và đề xuất nhằm cải thiện chất lượng của multipath TCP cũng được trình bày.

**Từ khóa:** Điều khiển tắc nghẽn, truyền tải đa đường, ứng dụng thời gian thực, ứng dụng phi thời gian thực, dựa vào tổn thất, dựa vào độ trễ.

## 1. Mở đầu

Ngày nay, nhu cầu sử dụng thông tin số ngày càng nhiều và đa dạng, nhu cầu kết nối thông tin diễn ra mọi lúc, mọi nơi. Thiết bị ngày nay phát triển mạnh về công nghệ kết nối không dây như Smartphone, tablet, laptop hỗ trợ kết nối như: Wifi, 3G. Các ứng dụng ngày nay đòi hỏi nhiều dung lượng lớn, cho nên yêu cầu băng thông cần được tăng lên.

Thực trạng đường truyền kết nối hiện nay không thoả mãn cho nhu cầu hiện tại và tương lai. Vì thế, mong muốn hiện nay của người dùng là kết nối thông tin nhanh và liên tục.

Các trung tâm dữ liệu như Amazon, Google hiện nay cũng đã kết nối với nhiều nhà cung cấp dịch vụ, xu hướng phát triển thiết bị di động đều trang bị nhiều đường kết nối như: wifi, 3G... Nếu

## Abstract

Multipath TCP is a set of extensions to regular TCP that allows one TCP connection to be spread across multiple paths. Multipath TCP distributes load through the creation of separate "subflows" across potentially disjoint paths. Multipath TCP is primarily concerned with utilizing multiple paths end-to-end to improve throughput. In terms of congestion control, loss-based algorithms and delay-based algorithms can be applied to multipath TCP. However, it needs to be clarified which kind of them be better than other in multipath TCP. Additionally, impacts of various traffic on performance of each ones in multipath TCP should be appraised, such as impacts of realtime traffic and non realtime traffic. These items are cleared up in this paper. Base on results of simulation with NS-2 tool, assessments and suggestions are also given for improving performance of multipath TCP.

**Key words:** Congestion control, multipath TCP, real-time applications, non-real-time applications, loss-base, delay-base.

thiết bị đầu cuối đồng thời sử dụng nhiều giao diện kết nối thì kỹ thuật truyền tải đa đường (Multipath TCP) sẽ đáp ứng được nhu cầu mong muốn hiện nay. Hình 1, minh họa cho việc sử dụng giao thức truyền tải đa đường cho thấy smartphone, tablet kết nối Internet với trung tâm dữ liệu đồng thời qua đường 3G và Wifi.



Hình 1. Minh họa sử dụng Multipath TCP

<sup>1</sup> Thạc sĩ, Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh

Đa số các thiết bị đầu cuối hiện nay được trang bị nhiều công cụ kết nối bằng nhiều đường, nhưng thông tin liên lạc thường được giới hạn một con đường duy nhất cho mỗi lần kết nối. Sử dụng tài nguyên trong hệ thống sẽ hiệu quả hơn nếu được sử dụng đa đường kết nối đồng thời. Giao thức truyền tải đa đường đã được IETF công nhận<sup>2</sup> cho việc nghiên cứu phát triển kỹ thuật truyền tải đa đường nhằm tăng hiệu suất cho nhu cầu truyền tải hiện nay.

Nhằm tăng hiệu quả hơn nữa trong kỹ thuật truyền tải đa đường, và trên cơ sở các tiêu chí được đặt ra<sup>3</sup>, các thuật toán điều khiển tắc nghẽn đa đường đã được đề xuất. Trong đó, một số tài liệu đã nói lên các thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất đạt hiệu quả trong việc truyền dữ liệu. Vậy đối với các ứng dụng theo thời gian thực thì sao? Tại sao không dùng điều khiển tắc nghẽn dựa vào tổn thất hay điều khiển tắc nghẽn dựa vào độ trễ? Để làm rõ những điều nói trên, bài viết sẽ tập trung nghiên cứu đánh giá hai dạng điều khiển tắc nghẽn dựa vào tổn thất và dựa vào độ trễ trong truyền tải đa đường. Qua đó xác định sự phù hợp hay không, ở mức độ nào khi triển khai các dạng ứng dụng sử dụng dịch vụ truyền tải đa đường theo từng phương pháp điều khiển tắc nghẽn nói trên.

## 2. Nội dung

### 2.1. Điều khiển tắc nghẽn TCP đơn đường

#### 2.1.1. Khái niệm

Cơ chế điều khiển lưu lượng trong TCP gồm: cơ chế truyền lại, cơ chế cửa sổ trượt, quản lý cửa sổ, điều khiển lỗi.

Cơ chế truyền lại: để đảm bảo kiểm tra việc truyền lại và khắc phục lỗi trong việc truyền dữ liệu, TCP có cơ chế đồng hồ kiểm tra truyền lại (time-out) và cơ chế truyền lại (retransmission). Thời gian khứ hồi (Round Trip Time) được xác định từ thời điểm bắt đầu truyền dữ liệu của bên gửi cho đến khi nhận được trả lời (ACKnowledgment) của bên nhận là yếu tố quyết định giá trị đồng hồ kiểm tra truyền lại  $t_{out}$ . Vậy  $t_{out} \geq RTT$ .

Hiện tượng nghẽn mạng: xảy ra khi số lượng gói tin đến nút mạng vượt quá khả năng xử lý của

nó hoặc vượt quá khả năng vận tải của các đường truyền ra, điều đó dẫn đến việc thông lượng của mạng bị giảm đi khi lưu lượng đến mạng tăng lên. Hiện tượng tắc nghẽn có thể xảy ra ở một hoặc một số nút mạng, hay trên toàn mạng.

#### 2.1.2. Thuật toán điều khiển tắc nghẽn dựa vào tổn thất trong TCP

Để tránh hiện tượng tắc nghẽn, Jacobson và các cộng sự đã đề xuất các biện pháp để tránh tắc nghẽn. Giải pháp chính là kiểm soát tốc độ gửi dữ liệu còn gọi là “cửa sổ tắc nghẽn” ( $cwnd$ ), nhằm hạn chế số lượng dữ liệu gửi để tránh tắc nghẽn. Khi kích thước  $cwnd$  chưa vượt ngưỡng (*Slow Start threshold*), kích thước  $cwnd$  sẽ tăng theo hàm mũ. Khi kích thước  $cwnd$  vượt ngưỡng, kích thước  $cwnd$  sẽ tăng tuyến tính. Khi hết thời gian đợi (*timeout*), giá trị ngưỡng bằng một nửa giá trị kích thước  $cwnd$  hiện thời và kích thước  $cwnd$  nhận giá trị 1. Nhằm đạt hiệu quả hơn trong việc điều khiển tắc nghẽn cho giao thức truyền tải đơn đường dựa vào tổn thất, một số thuật toán được đề xuất cải tiến như: Reno, New Reno và SACK.

#### 2.1.3. Thuật toán điều khiển tắc nghẽn dựa vào độ trễ trong TCP.

Các thuật toán điều khiển tắc nghẽn đơn đường dựa vào độ trễ đã được đề xuất bởi Jain, Tri-S bởi Wang và Crowcroft, trong đó thuật toán Vegasdo Brakmo và cộng sự được phân tích kỹ lưỡng.

Thuật toán Vegas thực hiện:

$$ExpThroughput = \frac{cwnd}{BaseRTT}$$

(BaseRTT = min of all RTT)

$$ActThroughput = \frac{cwnd}{RTT}$$

(RTT = BaseRTT +  $\tau$ )

$$Diff = (ExpThroughput - ActThroughput) * BaseRTT$$

-  $ExpThroughput$ : thông lượng mong đợi khi truyền.

-  $ActThroughput$ : thông lượng thực tế khi truyền.

-  $Diff$ : thông lượng khác nhau giữa thông lượng mong đợi so với thông lượng thực tế.

Thuật toán điều chỉnh kích thước  $cwnd$  theo:

$$\begin{aligned} cwnd &= cwnd + 1 & Diff < \alpha \\ cwnd &= cwnd - 1 & Diff > \beta \\ cwnd &= cwnd & \alpha \leq Diff \leq \beta \end{aligned}$$

Với  $\alpha$ , và  $\beta$  là hằng số.

<sup>2</sup> A. Ford, C. Raiciu, M. Handley, S. Barre, J. Iyengar. 2011. “Architectural Guidelines for Multipath TCP Development”. Internet Engineering Task Force (IETF), RFC 6182, ISSN: 2070-1721

<sup>3</sup> C. Raiciu, M. Handly, D. Wischik. 2011. “Coupled Congestion Control for Multipath Transport Protocols”. Internet Engineering Task Force (IETF), RFC 6356

Nếu giá trị thấp nhất của RTT cho N gói tin (minRTT) là luôn cao hơn BaseRTT:

Cập nhật lại giá trị cho BaseRTT

Kích thước cửa sổ tăng theo tương ứng.

Nói cách khác, Vegas tăng cwnd khi giá trị gói tin tại hàng đợi nhỏ hơn  $\alpha$ , giảm cwnd khi giá trị gói tin tại hàng đợi lớn hơn  $\beta$ , ngược lại thì giữ nguyên cwnd.

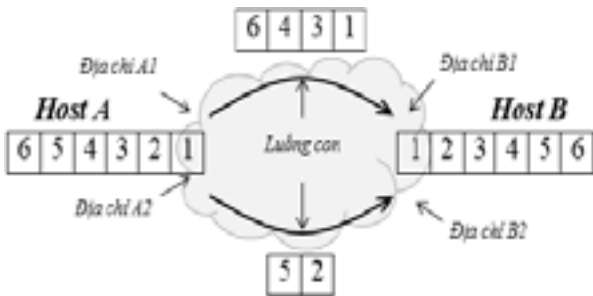
**2.2. Điều khiển tắc nghẽn TCP đa đường**

*2.2.1. Tổng quan về truyền tải đa đường*

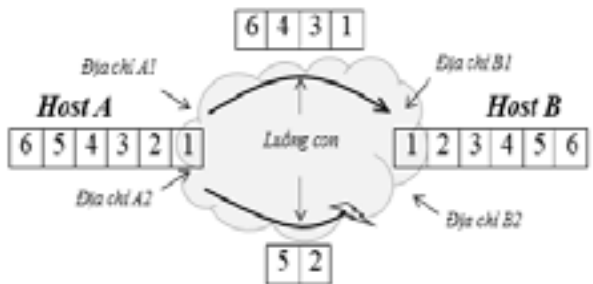
IETF khởi tạo nhóm nghiên cứu về giao thức truyền tải đa đường (MPTCP), nhằm phát triển kỹ thuật giao thức truyền tải đa đường cho các ứng dụng trên cơ sở tận dụng lợi thế sử dụng nhiều đường đồng thời để truyền dữ liệu.

*2.2.2. Mô hình cơ bản Multipath TCP*

Kết nối giữa các thiết bị đầu cuối trong giao thức truyền tải đa đường được hình thành từ một hoạt động nhiều luồng con. Các luồng con sẽ tạo ra các cặp địa chỉ khác nhau, và truyền dữ liệu cùng lúc trên các luồng con nhằm tăng thông lượng so với giao thức truyền tải đơn đường (Hình 2). Ngoài ra, một cơ chế cho giao thức truyền tải đa đường là khả năng phục hồi: khi một luồng con mất kết nối thì nó có cơ chế chuyển dữ liệu sang luồng con khác (Hình 3).



Hình 2. Minh họa kết nối Multipath TCP



Hình 3. Minh họa khả năng phục hồi Multipath TCP

*2.2.3. Chức năng giao thức truyền tải đa đường*

Giao thức truyền tải đa đường có các chức năng:

quản lý đường truyền thì tạo ra các luồng con, thiết lập kết nối cho các luồng con. Lập kế hoạch gói để phân chia dữ liệu, đánh số thứ tự phân đoạn dữ liệu trước khi gửi qua các luồng con. Cuối cùng, các thuật toán điều khiển tắc nghẽn sẽ thực hiện điều khiển các luồng dữ liệu.

Mục tiêu giao thức truyền tải đa đường: tăng thông lượng, cạnh tranh công bằng đường truyền, cân bằng cho đường truyền tải.

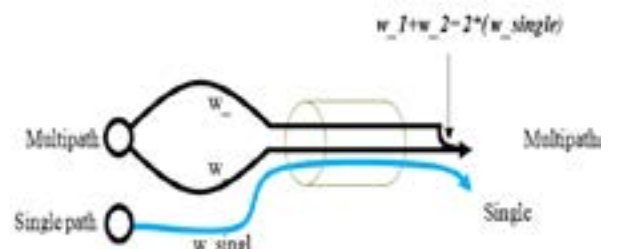
*2.2.4. Các thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất*

Thuật toán điều khiển tắc nghẽn đơn đường dựa vào tổn thất là trường hợp đặc biệt của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất:

Với mỗi thông báo xác nhận ACK trên luồng con thứ  $r$ , cửa sổ tắc nghẽn ( $w_r$ ) được tính:

$$w_r \leftarrow w_r + \frac{1}{w_r}$$

Thuật toán điều khiển tắc nghẽn đa đường với mỗi luồng con thực hiện điều khiển tắc nghẽn như là thuật toán điều khiển tắc nghẽn đơn đường cho luồng này, khi đó tổng thông lượng các luồng con sẽ tăng gấp đôi (giả sử lúc này thời gian khứ hồi của tất cả các đường là bằng nhau). Điều này dẫn đến cạnh tranh không công bằng đối với giao thức truyền tải đơn đường tại đường tắc nghẽn. Hình 4 minh họa cho việc cạnh tranh không công bằng khi hai luồng con của giao thức truyền tải đa đường cùng đi qua đường tắc nghẽn với đường truyền của giao thức truyền tải đơn đường.



Hình 4. Minh họa cho thấy cạnh tranh không công bằng

Một số thuật toán điều khiển tắc nghẽn đa đường đã đề xuất để giải quyết việc cạnh tranh công bằng với đường single path của giao thức truyền tải đơn đường hiện tại là thuật toán EWTCP; Couple

**Thuật toán EWTCP:** dựa trên TCP-New Reno trên mỗi đường  $r$  và điều chỉnh  $w_r$

+ Với mỗi thông báo xác nhận ACK trên luồng con thứ  $r$ ,  $w_r$  tăng:  $\frac{a}{w_r}$

**Thuật toán Couple:** thực hiện các bước khởi động chậm (*slow start*), truyền nhanh (*fast retransmit*) và phục hồi nhanh (*fastrecovery*) như thuật toán điều khiển tắc nghẽn đơn đường dựa vào tổn thất (*TCP Reno*). Với  $w_{total}$  là tổng kích thước cửa sổ tắc nghẽn của các luồng con kết nối. Thuật toán điều chỉnh  $w_r$ :

+ Với mỗi thông báo xác nhận ACK trên luồng con thứ  $r$ ,  $w_r$  tăng:

$$w_r \leftarrow \frac{w_r}{2}$$

*Tóm lại:* Các thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất đều có cơ chế cải tiến tăng kích thước cửa sổ tắc nghẽn ( $w_r$ ) trong trường hợp khi có thông báo xác nhận ACK trên luồng thứ  $r$ . Riêng trường hợp mất gói thì kích thước của sổ tắc nghẽn của các thuật toán giảm giống nhau theo công thức:  $w_r \leftarrow \frac{w_r}{2}$

### 2.2.5. Thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ

Được đề xuất trên cơ sở thuật toán điều khiển tắc nghẽn đơn đường dựa vào độ trễ Vegas<sup>4</sup>, có thể tóm tắt:

+ Trên mỗi luồng  $r$ , thực hiện giống như thuật toán điều khiển tắc nghẽn đơn đường dựa vào độ trễ.

+ Tổng các giá trị của các luồng là cố định, không phụ thuộc vào số lượng các luồng con.

+ Thích ứng tham số điều chỉnh  $\alpha$ ,  $\beta$  do ảnh hưởng đến tốc độ truyền tải của luồng con tương ứng với mục đích cân bằng mức độ tắc nghẽn mạng.

## 2.3. Kết quả và thảo luận

Ký hiệu trong phần mô phỏng này là:

- **MPTCP-loss**: thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất.

- **MPTCP-delay**: thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ.

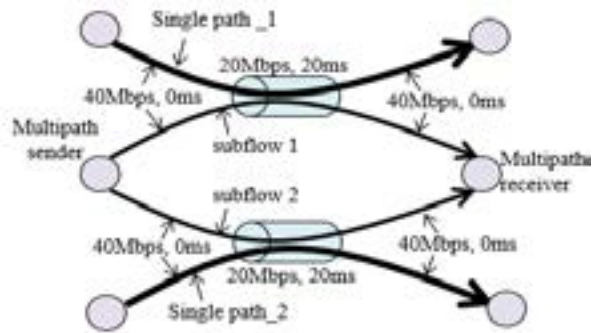
- **FTP**: loại ứng dụng phi thời gian thực.

- **CBR**: loại ứng dụng thời gian thực.

Bộ công cụ dùng để thực nghiệm mô phỏng là NS-2 (Network Simulator -2), phiên bản 2.34 và chạy trên môi trường là hệ điều hành Ubuntu với phiên bản 10.04. Thực nghiệm mô phỏng cho thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất trên cơ sở thuật toán *Couple* và thuật toán điều khiển tắc nghẽn đa đường dựa độ trễ là thuật toán *wVegas*.

### 2.3.1. Kết quả truyền tải của thuật toán điều khiển tắc nghẽn đa đường so với thuật toán điều khiển tắc nghẽn đơn đường

Nhằm làm rõ sự hiệu quả truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất và thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ đã được đề xuất. Trên cơ sở đó, chúng tôi xây dựng mô hình mạng như Hình 5:



Hình 5. Mô hình mạng Multipath với Single path

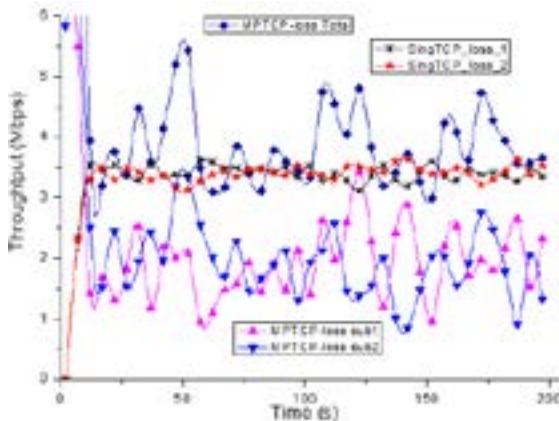
Với mô hình mạng (Hình 5), chúng tôi thiết lập cấu hình giống nhau cho hai loại thuật toán điều khiển tắc nghẽn “MPTCP-loss” và “MPTCP-delay”:

Multipath TCP bên gửi tạo ra hai luồng con subflow 1, subflow 2 được thiết lập thông lượng 40Mbps, thời gian trễ 0ms. Đường tắc nghẽn 1 và 2 được thiết lập thông lượng 20Mbps, thời gian trễ 20ms. Luồng Single path\_1 được thiết lập thông lượng 40Mbps, thời gian trễ 0ms và cùng đi qua đường tắc nghẽn 1 với luồng con subflow 1 của Multipath. Luồng Single path\_2 được thiết lập thông lượng 40Mbps, thời gian trễ 0ms và cùng đi qua đường tắc nghẽn 2 với luồng con subflow 2 của Multipath.

#### 2.3.1.1. Kết quả truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất so với thuật toán điều khiển tắc nghẽn đơn đường dựa vào tổn thất

Với thời gian là 200s, chúng tôi có được kết quả mô phỏng như Hình 6

<sup>4</sup> Yu Cao, Mingwei Xu, Xiaoming Fu. 2012. “Delay-based Congestion Control for Multipath TCP”. 2012 20th IEEE International Conference on Network Protocols (ICNP).



Hình 6. Thông lượng MPTCP-loss

Từ kết quả Hình 6, xét thấy thông lượng truyền của luồng con 1 và luồng con 2 của Multipath thấp hơn thông lượng truyền đường single path 1 và đường single path 2. Nhưng tổng thông lượng của hai luồng con ( $MPTCP-loss\ Total=4.25\ Mbps$ ) cao hơn thông lượng đường single path 1 và đường single path 2. ( $SingTCP\_loss\_1=SingTCP\_loss\_2=3.29Mbps$ )

Vậy, thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất đạt hiệu quả tăng thông lượng so với thuật toán điều khiển tắc nghẽn đơn đường dựa vào tổn thất.

2.3.1.2. Kết quả truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ so với thuật toán điều khiển tắc nghẽn đơn đường dựa vào độ trễ

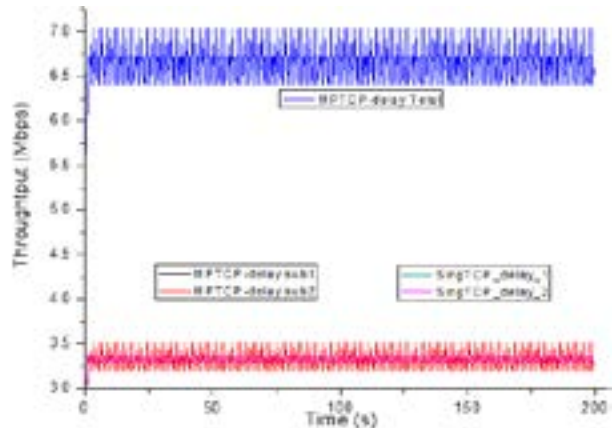
Với thời gian là 200s, chúng tôi có được kết quả mô phỏng như Hình 7.

Từ kết quả Hình 7, xét thấy thông lượng truyền của luồng con 1 và luồng con 2 của Multipath thấp hơn thông lượng truyền đường single path 1 và đường single path 2 ( $MPTCP\_delay\ sub1 = 3.331Mbps$ ;  $SingTCP\_delay\_1 = 3.332\ Mbps$ ). Nhưng tổng thông lượng trung bình của hai luồng con ( $MPTCP-delay\ Total = 6.66\ Mbps$ ) cao hơn thông lượng đường single path 1 và đường single path 2.

Như vậy, thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ đạt hiệu quả tăng thông lượng so với thuật toán điều khiển tắc nghẽn đơn đường dựa vào độ trễ.

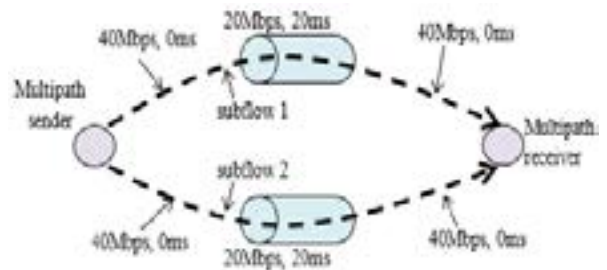
Tóm lại, thuật toán điều khiển tắc nghẽn đa đường truyền tải đạt hiệu quả hơn so với thuật toán điều khiển tắc nghẽn đơn đường.

2.3.2. Kết quả truyền tải của thuật toán điều khiển tắc nghẽn đa đường cho từng loại ứng dụng khác nhau



Hình 7. Thông lượng MPTCP-delay

Với mục tiêu làm rõ thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất và thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ, loại nào đạt hiệu quả hơn trong việc truyền tải cho các ứng dụng, chúng tôi tiến hành thực nghiệm mô phỏng qua 04 kịch bản với mô hình mạng như Hình 8.



Hình 8. Mô hình mạng Multipath TCP

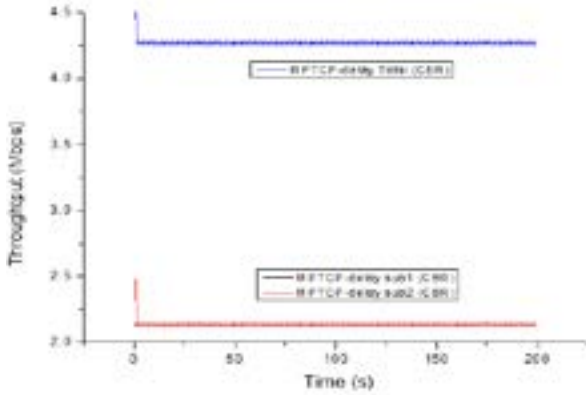
Với mô hình mạng Hình 8, chúng tôi thiết lập cấu hình:

Multipath TCP bên gửi tạo ra hai luồng con subflow 1, subflow 2 được thiết lập thông lượng 40Mbps, thời gian trễ 0ms. Tại nút mạng, thiết lập đường tắc nghẽn 1 và 2 thông lượng 20Mbps, thời gian trễ 20ms.

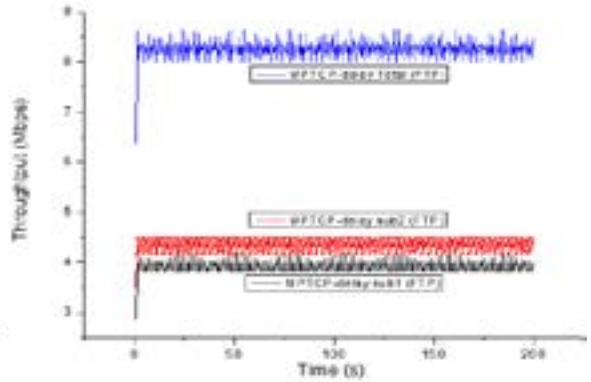
2.3.2.1. Mục tiêu mô phỏng kịch bản 1

Cùng một thuật toán **MPTCP-delay** truyền tải cho hai loại ứng dụng thời gian thực và phi thời gian thực. Vậy, loại ứng dụng thời gian thực có hiệu quả hay không so với ứng dụng phi thời gian thực. Với mục tiêu chúng tôi mô phỏng cho mô hình mạng (Hình 8).

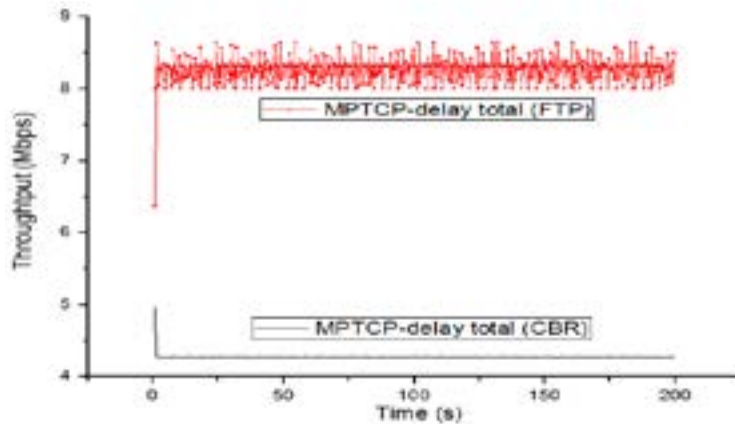
Với thời gian là 200s, Hình 9 là kết quả mô phỏng thuật toán **MPTCP-delay** cho ứng dụng thời gian thực, Hình 10 là kết quả mô phỏng **MPTCP-delay** cho ứng dụng phi thời gian thực. Hình 11 thông lượng truyền khác nhau cho hai loại ứng dụng thời gian thực và phi thời gian thực đối với thuật toán **MPTCP-delay**.



Hình 9. Thông lượng MPTCP-delay (CBR)



Hình 10. Thông lượng MPTCP-delay (FTP)



Hình 11. So sánh thông lượng MPTCP-delay total (CBR) và MPTCP-delay total (FTP)

Kết quả Hình 11 cho thấy thông lượng truyền của ứng dụng phi thời gian thực (*MPTCP-delay total (FTP)* là **8.3Mbps**) cao hơn thông lượng truyền của ứng dụng thời gian thực (*MPTCP-delay total (CBR)* là **4.3Mbps**).

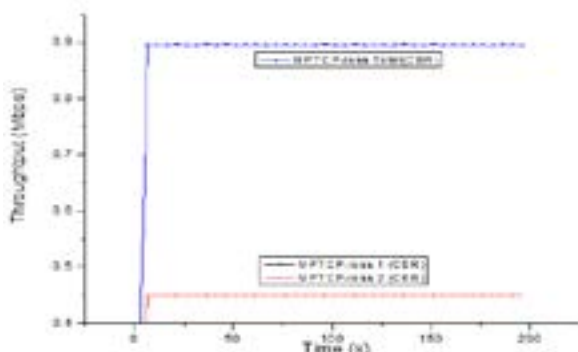
Với mục tiêu của kịch bản 1 đề ra, có thể thấy rằng đối với loại ứng dụng phi thời gian thực thì thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ hiệu quả hơn so với ứng dụng thời gian thực về tăng thông lượng.

### 2.3.2.2. Mục tiêu mô phỏng kịch bản 2

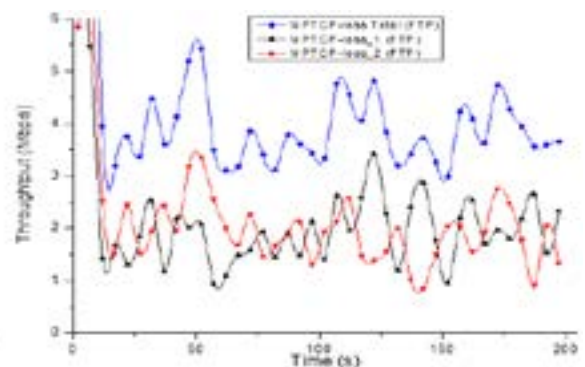
Qua kịch bản 1, chúng tôi nhận thấy với loại ứng

dụng phi thời gian thực thì thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ hiệu quả trong truyền tải. *Vậy đối với thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất thì loại ứng dụng nào đạt hiệu quả hơn.* Trên mục tiêu đề ra, chúng tôi thực nghiệm mô phỏng cho mô hình mạng (Hình 4). Với thời gian mô phỏng 200s, có kết quả:

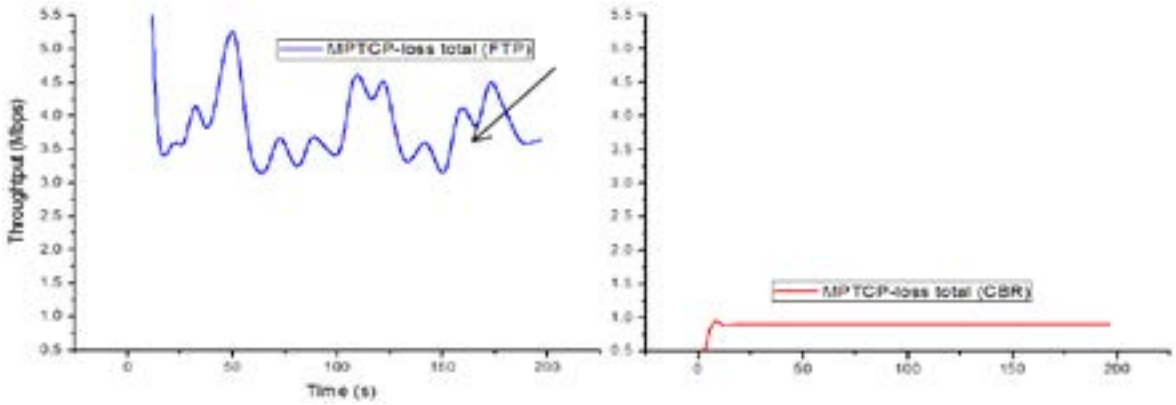
Hình 12 là kết quả mô phỏng cho *MPTCP-loss* với loại ứng dụng thời gian thực và Hình 13 là kết quả mô phỏng *MPTCP-loss* cho ứng dụng phi thời gian thực. Hình 14 thông lượng truyền khác nhau cho hai loại ứng dụng thời gian thực và phi thời gian thực đối với *MPTCP-loss*.



Hình 12. Thông lượng MPTCP-loss (CBR)



Hình 13. Thông lượng MPTCP-loss (FTP)



Hình 14. So sánh thông lượng MPTCP-loss total (CBR) và MPTCP-loss total (FTP)

Từ kết quả Hình 14, thông lượng truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất với loại ứng dụng thời gian thực (dao động 0.895Mbps-0.897Mbps) thấp hơn so với thông lượng truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất với loại ứng dụng phi thời gian thực (dao động 2.9Mbps-12.4Mbps).

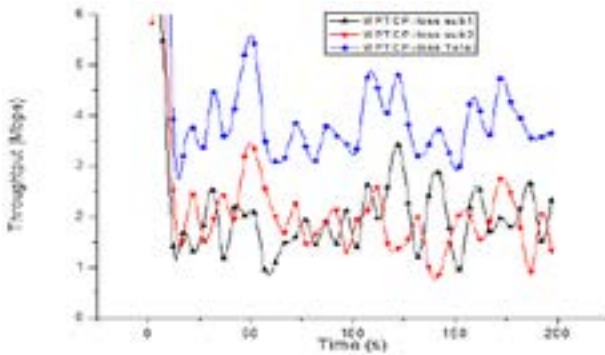
Với mục tiêu của kịch bản 2 đề ra, có thể thấy rằng đối với loại ứng dụng phi thời gian thực thì thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất truyền tải hiệu quả hơn so với ứng dụng thời gian thực.

2.3.2.3. Mục tiêu mô phỏng kịch bản 3

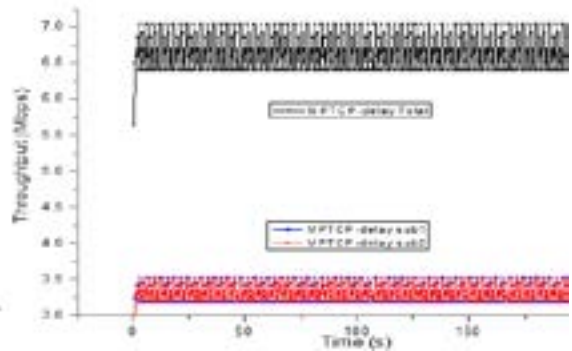
Cùng một loại ứng dụng phi thời gian thực, khi

truyền tải với thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất đạt hiệu quả như thế nào so với thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ. Với mục tiêu đề ra, chúng tôi thực nghiệm mô phỏng cho mô hình mạng (Hình 8) với thời gian mô phỏng 200s, có kết quả:

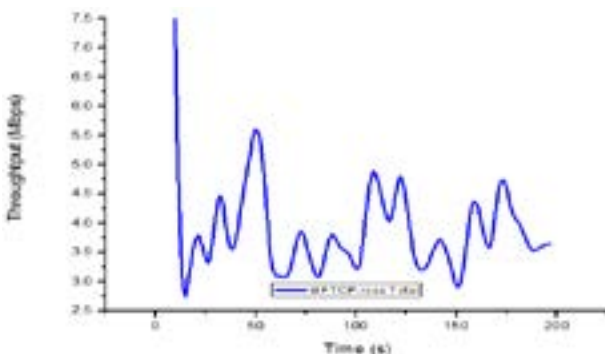
Cùng truyền tải loại ứng dụng phi thời gian thực. Hình 15 là kết quả mô phỏng thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất, Hình 16 là kết quả mô phỏng thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ. Hình 17 thông lượng truyền của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất so với thuật toán điều khiển tắc nghẽn đa đường dựa độ trễ cho một loại ứng dụng phi thời gian thực.



Hình 15. Thông lượng MPTCP-loss (FTP)



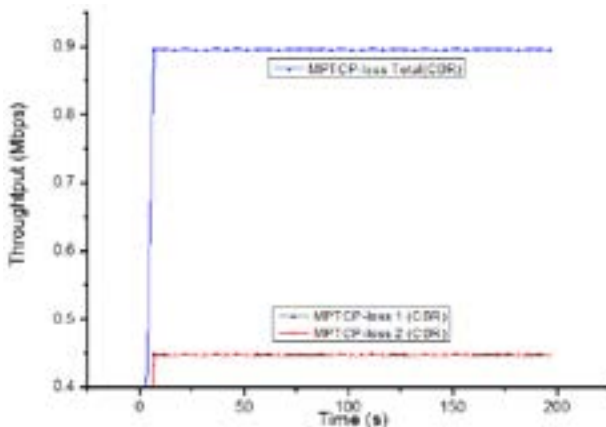
Hình 16. Thông lượng MPTCP-delay (FTP)



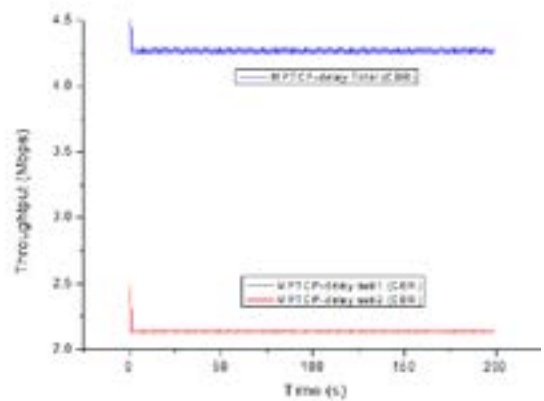
Hình 17. So sánh thông lượng MPTCP-loss total (FTP) và MPTCP-delay total (FTP)

Hình 17 cho thấy, thông lượng truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất cao hơn so với thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ. Nhưng thông lượng trung bình của thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ (*MPTCP-delay* là  $6.66Mbps$ ) cao hơn thông lượng trung bình của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất (*MPTCP-loss* là  $4.25Mbps$ )

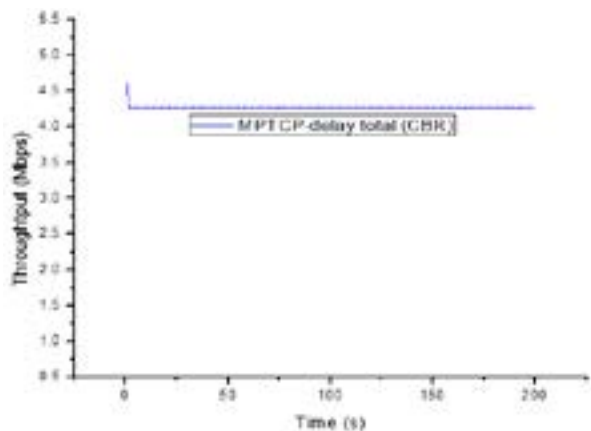
Từ đó thấy rằng thuật toán điều khiển tắc nghẽn đa đường dựa độ trễ đạt hiệu quả hơn về tăng thông lượng so với thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất khi truyền tải với loại ứng dụng phi thời gian thực.



Hình 18. Thông lượng MPTCP-loss (CBR)



Hình 19. Thông lượng MPTCP-delay (CBR)



Hình 20. So sánh thông lượng MPTCP-delay total (CBR) và MPTCP-loss total (CBR)

Kết quả Hình 20 cho thấy thông lượng truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ (dao động  $4.2Mbps-4.6Mbps$ ) cao hơn so với thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất (dao động  $0.895Mbps - 0.897Mbps$ ).

Từ đó thấy rằng thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ hiệu quả hơn về tăng

#### 2.3.2.4. Mục tiêu mô phỏng kịch bản 4

Đối với loại ứng dụng thời gian thực thì loại thuật toán nào đạt hiệu quả hơn. Trên mục tiêu đề ra, chúng tôi thực nghiệm mô phỏng cho mô hình mạng (Hình 8), với thời gian mô phỏng 200s, có kết quả:

Cùng truyền tải loại ứng dụng thời gian thực, Hình 18 là kết quả mô phỏng thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất, Hình 19 là kết quả mô phỏng thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ. Hình 20 thông lượng truyền tải của thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất so với thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ cho một loại ứng dụng thời gian thực.

thông lượng so với thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất khi truyền tải với loại ứng dụng thời gian thực.

### 3. Kết luận và đề xuất

Qua nghiên cứu cơ sở lý thuyết, sau đó tiến hành thực nghiệm mô phỏng thuật toán điều khiển tắc nghẽn đa đường so với thuật toán khiển tắc



nghe đơn đường hiện tại và mô phỏng hai loại thuật toán điều khiển tắc nghẽn đa đường cho từng loại ứng dụng khác nhau, từ mô phỏng chúng tôi nhận xét các kết quả như sau:

**Với kết quả mô phỏng, chứng tỏ rằng:**

- Thứ nhất: cả hai thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất và thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ đều đạt hiệu quả tăng thông lượng so với thuật toán điều khiển tắc nghẽn đơn đường.

- Thứ hai: thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất và thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ đạt hiệu quả khi truyền tải với loại ứng dụng phi thời gian thực

về tiêu chí tăng thông lượng so với loại ứng dụng thời gian thực.

- Thứ ba: đối với loại ứng dụng thời gian thực thì thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ đạt hiệu quả hơn về tăng thông lượng so với thuật toán điều khiển tắc nghẽn đa đường dựa vào tổn thất.

**Với kết quả đạt được, kiến nghị đề xuất:**

- Nghiên cứu phát triển và cải tiến các thuật toán điều khiển tắc nghẽn đa đường vì sự hiệu quả của nó so với thuật toán điều khiển tắc nghẽn đơn đường.

- Cần nghiên cứu cải thiện thuật toán điều khiển tắc nghẽn đa đường dựa vào độ trễ do đạt hiệu quả khi truyền tải cho loại ứng dụng thời gian thực.

### Tài liệu tham khảo

A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. 2013. “TCP Extensions for Multipath Operation with Multiple Adresse”. *Internet Engineering Task Force (IETF)*, RFC6824. A.Ford, C.Raiciu, M.Handley.

L.S Brakmo, and L.L. Peterson. 1995. “TCP Vegas: End to end congestion avoidance on a global Internet”. *Selected Areas in Communications, IEEE Journal on 13.8 (1995)*: 1465-1480.

C. Raiciu, M. Handley, D. Wischik. 2011. “Coupled Congestion Control for Multipath Transport Protocols”. *Internet Engineering Task Force (IETF)*, RFC 6356.

Damon Wischik, Costin Raiciu, Adam Greenhalgh, Mark Handley. 2011. “Design, implementation and evaluation of congestion control”. *Usenix NSDI*.

Qiuyu Peng, Anwar Walid, Steven H. Low. 2013. “Multipath TCP Algorithms: Theory and Design”. *SIGMETRICS'13*, June 17-21, 2013.

Jain Raj.1989.“A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks”. *ACM Computer Communication Review*, 19(5):56–71, Oct. 1989.

Cao Yu, Xu Mingwei, Fu Xiaoming. 2012. “Delay-based Congestion Control for Multipath TCP”. *2012 20th IEEE International Conference on Network Protocols (ICNP)*.