

## Phương pháp dựa trên thuật toán tăng cường độ dốc nhẹ cho dự đoán đáp ứng động lực học của tấm cơ tính biến thiên

Đỗ Thị Thanh Diệu\*, Nguyễn Hoàng Yên  
Khoa Công nghệ Thông tin, Trường Đại học Nguyễn Tất Thành  
dtttdieu@ntt.edu.vn

**Tóm tắt** Mục tiêu chính của bài viết này là để dự đoán một cách hiệu quả đáp ứng động lực học của các tấm cơ tính biến thiên bằng cách sử dụng thuật toán tăng cường độ dốc nhẹ (LightGBM) mà không phụ thuộc vào bất kỳ công cụ phân tích nào. Để thu được mô hình LightGBM tối ưu, một tập dữ liệu bao gồm 1 000 cặp dữ liệu đầu vào và đầu ra được tạo ra thông qua các lần lặp bằng cách sử dụng kết hợp phân tích đẳng hình học (IGA) và lý thuyết tấm biến dạng cắt bậc ba (TSDT). Trong mô hình này, đầu vào là chỉ số mũ, nó chi phối sự phân bố vật liệu của tấm, và đầu ra bao gồm 200 giá trị minh họa chuyển vị theo thời gian. Để chứng minh tính hiệu quả của mô hình LightGBM về độ chính xác và thời gian tính toán, kết quả thu được từ mô hình đề xuất được so sánh với các kết quả đạt được bởi các mô hình tối ưu ANN, XGBoost và IGA.

**Từ khóa** tấm cơ tính biến thiên; đáp ứng động lực học; thuật toán tăng cường độ dốc nhẹ; phân tích đẳng hình học; mạng thần kinh nhân tạo; thuật toán tăng cường độ dốc cực cao.

# Khảo sát các thuật toán kiểm soát tắc nghẽn và lập lịch vận chuyển gói tin trong giao thức MPTCP

Nguyễn Văn Thành\*, Nguyễn Kim Quốc\*\*, Dương Minh Tuấn

Khoa Công nghệ Thông tin - Trường Đại học Nguyễn Tất Thành

\*thanhnv@ntt.edu.vn, \*\*nkquoc@ntt.edu.vn

## Tóm tắt

Giao thức MPTCP là một đề xuất cải tiến từ giao thức kiểm soát đường truyền truyền thống. MPTCP hỗ trợ cho các ứng dụng nâng cao hiệu suất truyền dữ liệu trên mạng bằng cách sử dụng nhiều đường truyền vật lý. Cho đến nay, MPTCP vẫn chưa được phổ biến rộng rãi cho dù đã có nhiều giải pháp kiểm soát tắc nghẽn và lập lịch vận chuyển gói tin đã được nghiên cứu. Bài báo này thực hiện khảo sát các nghiên cứu đã công bố liên quan đến kiểm soát tắc nghẽn và lập lịch các luồng dữ liệu trong giao thức MPTCP. Khảo sát này phân loại kiểm soát tắc nghẽn và lập lịch gói tin theo hướng tiếp cận truyền thống và hướng tiếp cận học máy. Với mỗi nghiên cứu đã khảo sát sẽ có đánh giá ngắn gọn, nhằm hỗ trợ cho những nghiên cứu áp dụng học máy vào cải thiện hiệu năng của giao thức mạng MPTCP trong tương lai.

Nhận 10/03/2024  
Được duyệt 26/04/2024  
Công bố 20/06/2024

## Từ khóa

giao thức TCP đa đường, kiểm soát tắc nghẽn, lập lịch gói tin, MPTCP dựa trên học máy

© 2024 Journal of Science and Technology - NTTU

## 1 Giới thiệu

Kể từ khi mạng máy tính ra đời và lớn mạnh thành hệ thống Internet như ngày nay, giao thức IP đã chiếm vị trí độc tôn trong truyền thông dữ liệu giữa các máy tính và thiết bị trên mạng. Hỗ trợ cho các ứng dụng vận chuyển dữ liệu cho nhau trên mạng có 2 giao thức gồm: “giao thức gói dữ liệu người dùng” (User Datagram Protocol - UDP) và “giao thức kiểm soát đường truyền” (Transmission Control Protocol - TCP). Giao thức UDP [1] truyền dữ liệu theo kiểu phi kết nối: bên gửi sẽ đẩy dữ liệu đi mà không cần biết bên nhận có sẵn sàng hay không. Điều này giúp cho giao thức UDP xử lý nhanh nhưng là giao thức không tin cậy. Giao thức còn lại là TCP [2] truyền dữ liệu kiểu hướng kết nối: bên gửi liên lạc với bên nhận trước khi truyền dữ liệu, đảm bảo độ tin cậy, tính chính xác cho dữ liệu truyền. Đây cũng là lý do mà TCP được sử dụng phổ biến trên các ứng dụng có trao đổi thông tin trên mạng hiện nay. Ngày nay, với sự bùng nổ của công nghệ thông tin, các ứng dụng có nhu cầu gia tăng tốc độ truyền thông tin

trên mạng. Trong khi đó, giao thức TCP chỉ hỗ trợ vận chuyển dữ liệu từ các ứng dụng đến một thiết bị giao tiếp mạng duy nhất tại một thời điểm. Do vậy, cải tiến giao thức TCP là nhu cầu tất yếu.

Giải quyết cho vấn đề này, một giao thức mới với tên gọi “giao thức TCP đa đường” (Multipath TCP - MPTCP) [3] đã được đề xuất. MPTCP hỗ trợ cho ứng dụng vận chuyển dữ liệu lên mạng bằng nhiều luồng con khác nhau, với mỗi luồng con được thiết lập trên một giao tiếp mạng của thiết bị. Điều này giúp tăng băng thông mạng, cải thiện hiệu năng truyền dữ liệu giữa các ứng dụng trên mạng một cách đáng kể.

Tuy vậy, những vấn đề mà giao thức TCP truyền thống đã từng gặp phải như: mất gói, trễ gói, thời gian hồi đáp trở nên phức tạp hơn đối với giao thức MPTCP do việc truyền dữ liệu trên nhiều luồng con của nó. Đặc biệt là những thách thức trong kiểm soát tắc nghẽn trên nhiều luồng con. Kiểm soát tắc nghẽn đa đường là cơ sở cho việc lập lịch vận chuyển gói tin đến các luồng con sao cho đạt hiệu quả cao nhất có thể.



Trước tiên, xin giới thiệu vắn tắt một số kiến thức nền tảng về giao thức TCP và MPTCP và những bài toán cần giải quyết của 2 giao thức này. Sau đó, dựa trên những nghiên cứu đã thực hiện khảo sát, bài báo sẽ tóm tắt lại những thuật toán kiểm soát tắc nghẽn và các thuật toán lập lịch gói tin của giao thức MPTCP. Cả hai vấn đề này, bài báo sẽ trình bày theo 2 dạng: theo hướng tiếp cận truyền thống và theo hướng tiếp cận học máy. Với mỗi nội dung khảo sát sẽ có một số đánh giá nhằm cung cấp góc nhìn tổng quan về tình trạng của giao thức MPTCP hiện nay.

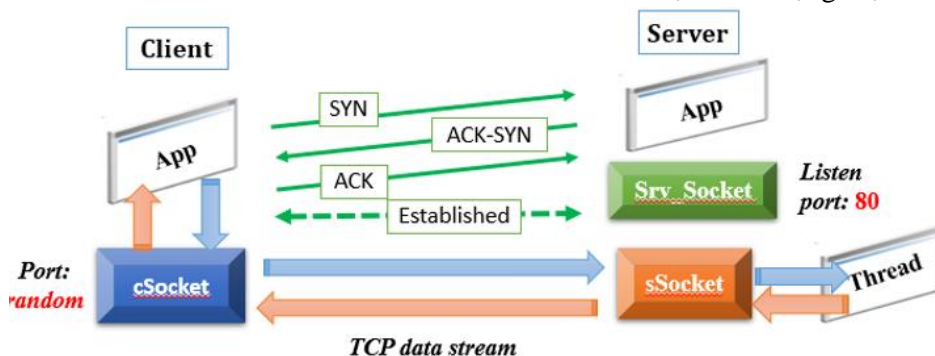
## 2 Nền tảng về TCP và MPTCP

Phần này, sẽ trình bày ngắn gọn một số kiến thức liên quan đến giao thức TCP và MPTCP. Những kiến thức nền này sẽ giúp nhìn rõ hơn những thách thức mà giao

thức TCP và MPTCP gặp phải. Ngày nay, các giải pháp cho TCP đã đạt được sự tin cậy, còn MPTCP thì chưa.

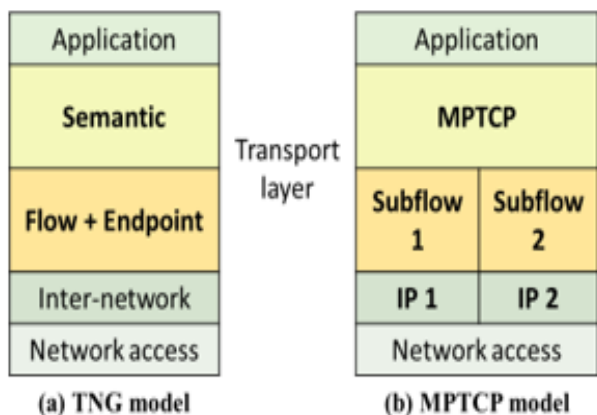
### 2.1 Tổng quan về giao thức TCP

TCP là giao thức dạng hướng kết nối: các thủ tục, qui trình của TCP hướng đến việc đảm bảo độ tin cậy, tính chính xác, tính đầy đủ cho dữ liệu truyền và nhận. Giao thức này yêu cầu bên gửi phải thực hiện qui trình thỏa hiệp “bắt tay 3 bước” với bên nhận trước khi truyền dữ liệu đi. Dữ liệu chỉ được truyền đi khi qui trình thỏa hiệp thành công, đảm bảo bên nhận sẽ nhận được dữ liệu đó. Trong quá trình truyền, bên gửi sẽ chia dữ liệu thành các gói tin, đánh số thứ tự cho các gói tin, lưu giữ bản sao của gói tin cho đến khi nhận được thông báo xác nhận (Acknowledgment - ACK) của bên nhận. Hình 1 cho thấy quy trình “bắt tay 3 bước” (three-way handshake) do phía máy khách (client) - nơi có nhu cầu trao đổi dữ liệu - chủ động thực hiện.



Hình 1 Qui trình “bắt tay 3 bước” của TCP

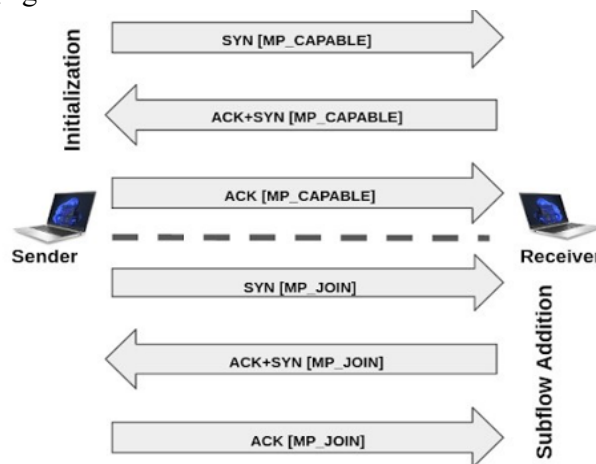
### 2.2 Tổng quan về giao thức MPTCP



Hình 2 Kiến trúc MPTCP

Giao thức MPTCP được phát triển từ mô hình “Transport thế hệ mới” (TNG) [4]. Hình 2 mô tả kiến trúc của MPTCP, theo đó, mỗi giao tiếp mạng vật lý trên thiết bị được MPTCP xây dựng thành một luồng con (subflow). Dữ liệu từ ứng dụng sẽ được MPTCP chia thành các gói tin và lập lịch vận chuyển vào các

luồng con để truyền ra ngoài. Tương tự, các gói tin từ ngoài vào các luồng con sẽ được MPTCP đưa vào ứng dụng.



Hình 3 Qui trình “bắt tay 3 bước” của MPTCP

Không phải thiết bị nào cũng sử dụng MPTCP thay thế cho TCP. Chính vì vậy, qui trình “bắt tay 3 bước” của MPTCP được thực hiện qua 2 giai đoạn được mô tả tại

Hình 3. Theo đó: giai đoạn (1): bên gửi (Sender) có dùng MPTCP thực hiện “bắt tay 3 bước” với bên nhận (Receiver) như TCP truyền thống. Giai đoạn (2): bên gửi sẽ gửi gói SYN (MP\_Join) gợi ý mời kết nối MPTCP. Nếu bên nhận có dùng MPTCP thì các luồng con sẽ hình thành [5].

### 2.3 Giải pháp của TCP và MPTCP đối với các thách thức trong truyền thông mạng

Truyền thông mạng sẽ gặp phải nhiều vấn đề. Bài báo này trình bày một số vấn đề lớn đối mà giao thức TCP và MPTCP cần phải giải quyết.

#### 2.3.1 Vấn đề mất gói tin.

Mất gói tin (packet loss) là tình trạng gói tin không đến được bên nhận. Với TCP, khi xảy ra mất gói, TCP sẽ xem đó là dấu hiệu tắc nghẽn mạng và phản ứng bằng các giảm giá trị của cửa sổ tắc nghẽn (Congestion Window - CWND) và ngưỡng bắt đầu chậm (Slow Start Threshold - SST) cho những gói tin tiếp theo. Do đó, băng thông truyền đi sẽ giảm ngay lập tức. Với MPTCP, nó sẽ giảm CWND và SST của luồng con bị tắc nghẽn, các luồng con khác vẫn bình thường. Vấn đề này đã được thực nghiệm trên điện thoại thông minh có wifi và LTE [6]. Kết quả cho thấy với dữ liệu nhỏ thì TCP ít bị mất gói hơn, nhưng với dữ liệu lớn thì MPTCP ít mất gói hơn, cho hiệu suất tốt hơn.

#### 2.3.2 Vấn đề về độ trễ gói tin.

Độ trễ gói tin (packet delay) là khoảng thời gian gói tin đi từ nơi phát đến nơi nhận. Độ trễ bị ảnh hưởng bởi các bộ định tuyến trên đường truyền, chất lượng đường truyền và nhất là tình trạng tắc nghẽn mạng. Đường truyền không dây thường có độ trễ cao hơn với đường có dây. Cơ chế hoạt động của giao thức MPTCP cho phép nó linh động trong việc lựa chọn đường truyền. Trong thực nghiệm đo hiệu năng của TCP và MPTCP dựa trên độ trễ của gói tin trong môi trường mạng có liên kết hỗn hợp, kết quả cho thấy MPTCP có độ trễ thấp hơn và sử dụng băng thông tốt hơn so với TCP [7].

#### 2.3.3 Vấn đề về mất thứ tự gói tin.

Trong giao thức TCP, dữ liệu cần truyền sẽ được chia thành nhiều gói tin (Segment - SEG) và đánh số thứ tự cho gói tin trước khi gửi đi. Bên nhận sẽ dùng số thứ tự này để sắp xếp lại các SEG cho đúng với dữ liệu gốc. Với TCP, các SEG lần lượt gửi đi trên một luồng duy nhất theo đúng thứ tự của nó. Còn với MPTCP, các SEG gửi đi trên nhiều luồng con nên có thể xảy ra tình trạng sai lệch thứ tự (Out-of-Order Packets). Sự sai lệch thứ tự này cũng là nguyên nhân gây ra hiện tượng các

gói tin truyền có mật độ không đồng đều, lúc quá nhiều, lúc quá ít, gọi là hiện tượng giật gói tin (jitter).

Do vậy, lập lịch vận chuyển gói tin là một trong những nhiệm vụ khó khăn nhất trong MPTCP. Một giải pháp lập lịch đa đường có tên là “kỹ thuật lập lịch gói tin theo độ trễ” đã được đề xuất nhằm giải quyết hiện tượng giật gói tin trong MPTCP [8]. Nhằm hạn chế sai lệch thứ tự trong mạng di động, một giải pháp khác sử dụng hàng đợi để lập lịch đa đường cũng đã được đề xuất [9].

#### 2.3.4 Thời gian quay về.

Thời gian quay về (Round-Trip Time - RTT) là khoảng thời gian tính từ lúc gói tin được gửi đi cho đến khi bên gửi nhận được ACK gói tin đó. RTT càng ngắn chứng tỏ hiệu suất truyền thông mạng càng cao. Giảm RTT cũng là mục tiêu quan trọng của MPTCP.

## 3 Kiểm soát tắc nghẽn trong MPTCP

Trong thực tế, tốc độ phát ra luồng dữ liệu truyền của bên gửi và tốc độ nhận luồng dữ liệu đó của bên nhận thường không đồng bộ nhau. Tắc nghẽn xảy ra khi tốc độ nhận chậm hơn tốc độ phát hoặc tốc độ vận chuyển dữ liệu của đường truyền chậm hơn tốc độ phát.

Kiểm soát tắc nghẽn (Congestion Control - CC) là khả năng mà bên phát dữ liệu nhận biết hoặc dự đoán được trạng thái của bên nhận hoặc của đường truyền để điều chỉnh lại tốc độ phát, kích thước gói tin sao cho sử dụng hiệu quả hạ tầng mạng dùng chung và cải thiện tình trạng tắc nghẽn cho bên nhận.

CC trong giao thức TCP gồm 3 pha:

- Pha “khởi đầu chậm” (Slow Start - SS): pha này được bên gửi sử dụng để xác định ngưỡng tối đa mà bên nhận có thể tiếp nhận luồng dữ liệu. Để xác định giá trị ngưỡng, bên gửi sẽ truyền gói tin có kích thước nhỏ để bên nhận phản hồi lại thông tin “kích thước tối đa” của gói mà nó có thể nhận được; nếu bên gửi nhận được ACK, nó sẽ gia tăng kích thước của gói tin và tiếp tục gửi cho bên nhận; quá trình này tiếp diễn cho tới khi bên gửi không còn nhận được ACK; giá trị kích thước của gói tin của gói cuối cùng chính là ngưỡng SST.

- Pha “tránh tắc nghẽn” (Congestion Avoidance - CA): đây là giai đoạn bên gửi truyền dữ liệu cho bên nhận. Bên gửi sẽ gia tăng kích thước gói tin truyền theo cấp số cộng cho đến khi đạt đến ngưỡng đã xác định ở pha SS.

- Pha “nhận dạng tắc nghẽn” (Congestion Detection - CD): trong quá trình truyền dữ liệu, nếu xảy ra tắc



nghe, bên gửi sẽ giảm kích thước gói tin truyền theo cấp số nhân.

Quá trình kiểm soát tắc nghẽn ở 2 pha CA và CD theo quy luật tăng kích thước gói tin truyền theo cấp số cộng và giảm theo cấp số nhân được gọi là AIMD.

Với giao thức TCP, đã có nhiều thuật toán kiểm soát tắc nghẽn được đề xuất và áp dụng như: TCP Reno, TCP Vegas [10] và TCP Cubic [11]. Với MPTCP, những thuật toán kiểm soát tắc nghẽn của TCP chỉ có thể áp dụng khi MPTCP chạy trên một luồng con, còn khi chạy đa đường thì các thuật toán trên không mang lại hiệu quả. Vì vậy, đã có nhiều nghiên cứu kiểm soát tắc nghẽn khi vận chuyển dữ liệu đồng thời trên nhiều luồng con. Bài báo sẽ chia các nghiên cứu kiểm soát tắc nghẽn cho MPTCP thành 2 nhóm, gồm nhóm dùng hướng tiếp cận truyền thống và nhóm sử dụng kỹ thuật học máy.

### 3.1 Các phương pháp kiểm soát tắc nghẽn cho MPTCP theo hướng tiếp cận truyền thống

Một phương pháp kiểm soát tắc nghẽn có tên là “kết hợp điều chỉnh kích thước gói linh động” (Dynamic Window Coupling - DWC) đã được đề xuất [12]. DWC dùng sơ đồ theo dõi các thông số mất gói và độ trễ để phát hiện tắc nghẽn. Phương pháp này xây dựng một “tập luồng con” là tập hợp các đường truyền mạng có trên thiết bị để chia sẻ băng thông hoặc hoạt động độc lập theo sự điều khiển của sơ đồ kiểm soát tắc nghẽn ở trên. Trong nghiên cứu này, các tác giả đã triển khai thực nghiệm trên phần mềm mô phỏng NS-2. Kết quả cho thấy thông lượng đạt cao hơn trên mỗi luồng con so với thuật toán “điều khiển tắc nghẽn kết hợp” (Coupled Congestion Control - CCC) đã được công bố tại RFC-6356 [13]. Nhóm tác giả trong nghiên cứu này còn đề xuất sử dụng bộ nhớ cho phương pháp của họ để nhận dạng và sử dụng lại các “tập luồng con” đã thành lập trước đó.

Một nhóm nghiên cứu khác đã phát triển thuật toán có tên gọi “điều khiển tắc nghẽn cho TCP đa đường dựa trên mức tiêu hao năng lượng” (energy-aware congestion control for multipath TCP - ecMTCP) [14]. ecMTCP sử dụng phương pháp phân bổ lưu lượng giữa các luồng con dựa vào các tiêu chí như: mật độ lưu lượng truyền tải, chi phí năng lượng của đường truyền để đạt tính cân bằng tải và tiết kiệm năng lượng. Phương pháp này được thực nghiệm trên phần mềm mô phỏng NS-2, tham số mật độ lưu lượng, chi phí năng lượng được cài đặt trên phần mềm. Kết quả là thông

lượng mạng đạt khá cao trên cả luồng TCP và MPTCP. Nghiên cứu này chưa thực nghiệm trên môi trường mạng thật, nhiều biến động.

Thuật toán “VEGAS có trọng số” (weighted VEGAS - wVEGAS) cải tiến từ thuật toán VEGAS dùng cho TCP để kiểm soát tắc nghẽn cho MPTCP [15]. wVEGAS kiểm soát thông số độ trễ của các gói tin trên các luồng con, từ đó làm cơ sở để chuyển gói tin trong hàng đợi của luồng con có độ trễ cao sang hàng đợi của luồng con có độ trễ thấp. Việc chuyển đổi lưu lượng của wVEGAS giúp giảm thiểu sự cố mất gói và mang lại cân bằng lưu lượng tốt hơn cho các luồng con. Nhóm tác giả đã thực nghiệm trên phần mềm mô phỏng NS-3, tối ưu hóa các tham số để đạt được mục tiêu kiểm soát tắc nghẽn.

Một phương pháp kiểm soát tắc nghẽn khác được đề xuất vào năm 2013 [16] bằng cách kết hợp thuật toán DWC và “thuật toán tăng liên kết cơ hội” (OLIA) [17]. Phương pháp này dùng kiểm soát độ trễ để sắp xếp gói tin trong hàng đợi, kết hợp với việc sử dụng “kích thước bộ nhớ đệm” ở phía bên nhận để kiểm soát tắc nghẽn đa đường. Phương pháp này đạt hiệu suất tốt trong các môi trường mạng mà tắc nghẽn luôn biến động.

Trong MPTCP, nếu tăng băng thông của các liên kết mạng để đạt được thông lượng cao hơn sẽ không thể hiện thực được nếu các đường mạng không chia sẻ sự cố thất cổ chai cho nhau [18]. Họ đã triển khai thuật toán “phát hiện sự cố thất cổ chai chia sẻ” (Shared Bottleneck Detection - SBD) cho MPTCP. Thuật toán này hướng tới sự cân bằng giữa tắc nghẽn và thông lượng mạng. Thực nghiệm của họ cũng đã cho thấy có một số trường hợp xảy ra tình trạng tắc nghẽn nhưng không chia sẻ. Tuy vậy, thông lượng mạng tối đa có thể đạt được tới 40 % khi sử dụng 2 luồng con. Mức thông lượng đạt đến 100 % khi sử dụng 5 luồng con. Thực nghiệm của họ được triển khai trên Linux kết hợp với phần mềm mô phỏng mạng CORE.

Khảo sát qua một số thuật toán trên, có thể thấy các thuật toán kiểm soát tắc nghẽn trong MPTCP đa phần tập trung vào pha CA và pha CD. Kiểm soát tắc nghẽn ở các pha này giúp mạng không xảy ra sự cố tràn dữ liệu dẫn đến hủy các gói có độ ưu tiên thấp. Điều này lý giải được nguyên nhân mà MPTCP thường chỉ có hiệu quả với việc truyền dữ liệu lớn. Kiểm soát tắc nghẽn ở giai đoạn này dễ gặp sự cố tràn bộ đệm (buffer overflow).

Thuật toán “kiểm soát tắc nghẽn nhất quán thông lượng” (Throughput Consistency Congestion Control - TCCC) là một giải pháp dùng hướng tiếp cận tại pha SS [19], TCCC đề xuất sử dụng phương pháp “khởi đầu chậm kết hợp” (Coupled SlowStart - CSS) và phương pháp “tránh tắc nghẽn tích cực” (Aggressive Congestion Avoidance - ACA). Nhóm tác giả sử dụng CSS để ngăn ngừa sự cố mất gói do sự dư thừa gia tăng và dùng ACA để kiểm soát băng thông chia sẻ hợp lý hơn. Trong thực nghiệm, TCCC nâng cao hiệu suất truyền tải dữ liệu. Tuy nhiên, trong MPTCP, các luồng con thường hoạt động ở các giai đoạn khác nhau trong khi CSS chỉ có vai trò trong pha SS. Thuật toán CSS cần được xem xét nghiên cứu cải thiện thêm.

Thực nghiệm cho thấy: các thuật toán kiểm soát tắc nghẽn đa đường hiện tại không thể nhanh chóng điều chỉnh lưu lượng mạng cho các hệ thống sử dụng chính sách “chất lượng dịch vụ” (Quality of Service - QoS) không đồng nhất. Điều đó dẫn đến hiệu suất kém trong một số môi trường mạng biến động [20]. Nhóm tác giả đề xuất thuật toán tên gọi ACCeSS. Thuật toán này sử dụng phương pháp “Hồi quy rừng ngẫu nhiên” (Random Forest Regressing) để thực hiện tối ưu hóa hiệu suất QoS theo các chỉ số đã chọn. Qua thực nghiệm cho thấy thuật toán này thích ứng nhanh với những thay đổi của mạng, đáp ứng tốt các chính sách QoS đặt trước. Trong nghiên cứu này, nhóm tác giả đã triển khai giải pháp trên Linux và cho kết quả vượt trội hơn hầu hết các phương pháp kiểm soát tắc nghẽn đa đường như wVegas, OLIA, BALIA [21].

### 3.2 Các phương pháp kiểm soát tắc nghẽn cho MPTCP theo hướng tiếp cận học máy

Có khá nhiều phương pháp và thuật toán kiểm soát tắc nghẽn cho MPTCP, nhưng các hướng tiếp cận kiểu truyền thống chú trọng vào các chỉ báo để nhận dạng tắc nghẽn như: mất gói, độ trễ, ACK và RTT... Trong thực tế, những chỉ báo này thường có giá trị bất thường, không ổn định. Chính vì thế mà các phương pháp kiểm soát tắc nghẽn truyền thống đạt hiệu suất cao trong mô phỏng nhưng khi thực nghiệm thực tế lại cho hiệu suất thấp hơn kỳ vọng. Trong khi đó, các phương pháp kiểm soát tắc nghẽn với hướng tiếp cận dựa trên học máy (ML) đưa ra quyết định dựa trên kinh nghiệm từ dữ liệu huấn luyện. Vì thế, chúng có thể thích ứng tốt với mọi tình huống mạng.

MPCC là một phương pháp kiểm soát tắc nghẽn đa đường dựa vào học máy trực tuyến [22]. Phương pháp

này sử dụng cơ chế AIMD truyền thống cho từng luồng con nhưng đáp ứng nhanh với những biến động mạng nhờ vào học máy trực tuyến. Nó cho phản ứng nhanh hơn nhằm giúp kiểm soát tắc nghẽn phản ứng nhanh với những biến động trong môi trường mạng. Nhóm tác giả đã thực nghiệm MPCC trên Linux kernel với các môi trường mạng có nhiều liên kết mạng khác nhau, Kết quả cho hiệu suất truyền cao, đạt tính công bằng cho các luồng con trong môi trường mạng nhiều biến động. Tuy nhiên, khi nghiên cứu đến việc tăng hiệu suất cho các luồng ngắn và giải quyết các vấn đề sai lệch băng thông trên đường truyền mạng thì phương pháp MPCC cho hiệu suất kém trong các tình huống trạng thái mạng thay đổi.

Như đã bàn luận, hoạt động của mạng máy tính trong thực tế luôn biến động, MPTCP cũng cần phải thích ứng nhanh với trạng thái mới. Một đề xuất sử dụng công nghệ “Học tăng cường” (Reinforcement Learning - RL) ra quyết định lựa chọn đường định tuyến để gửi các gói TCP sao cho thông lượng mạng được tối đa hóa [23]. Với vấn đề kiểm soát tắc nghẽn đa đường, nhóm tác giả áp dụng thuật toán trong ML gọi là “học trực tuyến dựa trên đa trang bị” [24]. Thuật toán điều khiển tắc nghẽn này cho phép MPTCP điều chỉnh tốc độ truyền linh hoạt và thích ứng cho từng luồng con với hiệu suất tốt.

SmartCC là một thuật toán sử dụng RL vào kiểm soát tắc nghẽn đa đường trong môi trường mạng không đồng nhất [25]. Nhóm nghiên cứu đã huấn luyện mô hình từ bộ quy tắc tắc nghẽn để giám sát môi trường mạng và thực hiện các hành động để điều chỉnh CWND của mỗi luồng con. Nhóm tác giả đã triển khai phương pháp của họ trên phần mềm mô phỏng NS-3.

Với mô hình “Kiểm soát tắc nghẽn sử dụng học tăng cường sâu” (Deep Reinforcement Learning Congestion Control - DRL-CC) [26], Nhóm tác giả sử dụng một DRL Agent để kiểm soát tắc nghẽn cho tất cả các luồng con của MPTCP nhằm tối đa hóa tổng tiện ích. Họ sử dụng mạng nơ ron dựa trên LSTM để biểu diễn lại hoạt động của các luồng con, huấn luyện cho DRL framework nhận dạng nhanh tắc nghẽn trên các luồng con. Nhóm đã triển khai thực nghiệm giải pháp trên Linux.

Một mô hình kiểm soát tắc nghẽn khác cũng dựa trên DRL có tên gọi là DeepCC [27]. Khác với DRL-CC ở trên, DeepCC sử dụng nhiều DRL agent. Mỗi DRL agent sẽ điều khiển CWND của một luồng con. Mô



hình của họ sử dụng một tiến trình có tên “tự chú ý” (self-attention) mang nhiệm vụ kiểm tra sự phụ thuộc của một luồng con với tổng trọng số của các luồng con khác. Nhóm đã so sánh giải pháp của họ với DRL-CC và cho thấy phương pháp của họ vượt trội hơn DRL-CC trong việc tăng/giảm tốc độ gửi gói để ứng phó với tình trạng tắc nghẽn biến động của mạng.

Một framework kiểm soát tắc nghẽn cho MPTCP dựa trên Q-learning và Deep Q-Network được đề xuất như là một giải pháp cải thiện thông lượng cho “mạng vạn vật trong không gian” (Internet of Deep Space Things - IoDST) nhằm cấp dịch vụ liên lạc cho tàu vũ trụ về trái đất. Giải pháp chính của họ là tính toán tìm ra một giá trị lý tưởng nhất cho CWND để truyền dữ liệu các cuộc hội thoại trên IoDST. Giải pháp của họ đạt kết quả tốt khi truyền video và âm thanh từ tàu vũ trụ về trái đất [28]. Hạn chế của của nghiên cứu này là giải pháp chỉ thực hiện trong môi trường IoDST.

#### 4 Lập lịch vận chuyển gói tin trong MPTCP

Lập lịch MPTCP là lựa chọn lưu lượng dữ liệu cần được lên lịch cho các luồng con khác nhau vận chuyển lên mạng. Mục tiêu của lập lịch MPTCP là sao cho đạt được hiệu suất tốt nhất như: thông lượng cao, độ trễ thấp, giảm tỉ lệ mất gói,... Trong phần này, bài báo này sẽ trình bày các phương pháp lập lịch MPTCP truyền thống và các phương pháp dựa trên học máy.

##### 4.1 Các phương pháp lập lịch MPTCP truyền thống

Phương pháp lập lịch MPTCP đầu tiên được đề cập trong RFC-6356, đề xuất 3 phương pháp lập lịch khác nhau:

(1) Phương pháp “mặc định” (default): đầu tiên, các gói tin sẽ được đưa vào luồng con có RTT thấp nhất (đường truyền nhanh) cho đến khi CWND của luồng đạt đến ngưỡng cho phép. Các gói tin tiếp theo sẽ được đưa vào luồng con có RTT cao hơn (đường truyền chậm). Đến luồng con cuối cùng thì xét lại RTT của các luồng con. Phương pháp này đảm bảo tất cả luồng con đều tham gia vận chuyển dữ liệu nhưng không hiệu quả với nhiều trường hợp như: kích thước dữ liệu không đồng nhất, vấn đề độ trễ, sự có mất gói tin, sự có thất cổ chai...

(2) Phương pháp “xoay vòng” (round-robin): các gói tin lần lượt gửi vào các luồng con theo hình thức xoay vòng. Phương pháp này tạo sự bất công bằng cho những luồng con có tốc độ truyền tải yếu; dễ xảy ra tình trạng tắc nghẽn luồng dữ liệu.

(3) Phương pháp “dư thừa” (redundant): phương pháp này chọn một luồng con dùng làm luồng dự phòng. Những gói tin cần độ trễ thấp sẽ được ưu tiên chuyển đến luồng dự phòng này. Phương pháp này hữu ích khi muốn đạt được độ trễ thấp nhưng phải chấp nhận giảm băng thông mạng.

Khảo sát thực nghiệm trên các công trình nghiên cứu cho thấy, so với TCP, giao thức MPTCP thường chiếm ưu thế trên các luồng có thời gian tồn tại lâu và hoạt động kém hơn khi kích thước luồng dữ liệu nhỏ. Với những trường hợp kích thước luồng dữ liệu nhỏ thì độ trễ cần được ưu tiên hơn so với băng thông mạng, vì thế mà những phương pháp nhanh sẽ được ưu tiên lựa chọn hơn. Nếu CWND của “đường truyền nhanh” không khả dụng thì bộ lập lịch MPTCP thường chọn “đường truyền chậm”, điều này dẫn đến thời gian hoàn thành luồng bị trễ. [29]

Khắc phục vấn đề trên, một bộ lập lịch MPTCP mới đã đề xuất, có khả năng chặn tạm thời “đường truyền chậm” khi chênh lệch độ trễ giữa nó với “đường truyền nhanh” là đáng kể [30]. Giải pháp lập lịch gói tin này cho phép lưu lượng dữ liệu nhỏ được phân phối nhanh chóng qua “đường truyền nhanh”. Các tác giả đã sử dụng phương pháp này để tìm luồng con có RTT thấp nhất bất kể tính khả dụng của CWND, dùng thuật toán “lập lịch theo độ ưu tiên của RTT” (Lowest-RTT-First) để chọn luồng con tối ưu. Sau đó, họ trả về giá trị tốt nhất nếu chênh lệch giữa luồng con có RTT tốt nhất và luồng con có RTT nhỏ hơn một điểm tới hạn nhất định. Với trên mạng 3G và wifi họ đã chọn điểm tới hạn độ trễ là 100 mili giây [30]. Với sự phát triển của mạng di động và Wifi ngày nay, bộ lập lịch này cần được đánh giá lại.

Mạng trong thực tế đa phần là mạng không đồng nhất, đa dạng băng thông, đa dạng độ trễ. Trong MPTCP, mạng không đồng nhất dễ gây ra hiện tượng mất thứ tự gói tin ở nơi nhận.. Khắc phục vấn đề này thuật toán “lập lịch gói tin cho MPTCP dựa trên độ trễ chuyển tiếp” (FDPS) được đề xuất. Thuật toán này gồm hai phần: dự đoán sự khác biệt về độ trễ chuyển tiếp trên các đường dẫn và chọn dữ liệu để gửi qua đường dẫn khi có sẵn CWND. Bài báo kết luận thuật toán của họ duy trì đáng kể thứ tự các gói đến tại máy nhận so với một số thuật toán trước đó [31].

Năm 2019, trong một nghiên cứu đánh giá các thuật toán lập lịch cho giao thức MPTCP hiện hữu, nhóm nghiên cứu đã chọn các thuật toán lập lịch như: thuật

toán lập lịch theo độ ưu tiên của RTT (Lowest-RTT-First), thuật toán lập lịch dựa trên độ trễ (Delay-aware packet scheduler - DAPS) [32], thuật toán lập lịch đặt lại thứ tự gói tin ở điểm đến (Out-of-order transmission for in-order arrival scheduler - OTIAS) [33] và thuật toán lập lịch dựa trên ước tính (Blocking estimation-based MPTCP scheduler - BLEST) [34] để đánh giá. Nhóm tác giả đã thực nghiệm ảnh hưởng của một số thông số mạng như: RTT, kích thước vùng nhớ đệm, kích thước tập tin đến hiệu suất của các thuật toán lập lịch. Ngoài ra, nhóm tác giả cũng đánh giá hiệu năng của các thuật toán lập lịch qua các yếu tố ảnh hưởng như: thời gian chờ và thời gian hoàn thành luồng và mạng có các đường truyền không đồng nhất. Kết quả cho thấy “Lowest-RTT-First” có thời gian chờ nhiều nhất, trong khi BLEST có thời gian chờ ít nhất. Trong thử nghiệm các kích thước vùng nhớ đệm khác nhau: BLEST cho hiệu năng hơn hẳn các thuật toán khác, tiếp theo là OTIAS và DAPS. Kết quả này cũng dễ hiểu vì thuật toán BLEST dự đoán sự cố “chặn đầu dòng” linh hoạt hơn, do đó giảm thiểu số lượng gói không đúng thứ tự. Trong các thử nghiệm các kích thước tập tin khác nhau, BLEST và LowRTT hoạt động tốt hơn DAPS và OTIAS [35].

Trong một nghiên cứu khác đã phân tích các bộ lập lịch MPTCP hiện có trong thời điểm nghiên cứu và xác định một số khoảng trống còn tồn tại, như: sự cố “chặn đầu dòng”, phân phối gói không theo thứ tự [36]. Sự cố “chặn đầu dòng” có thể xảy ra khi một hàng đợi gói tin chờ để được truyền và gói ở đầu đường truyền có thể không thể di chuyển về phía trước do tắc nghẽn. Những vấn đề này làm giảm hiệu suất MPTCP. Nhóm tác giả đã trình bày một “bộ lập lịch thích ứng và hiệu quả” (Adaptive and Efficient Packet Scheduler - AEPS). Bộ lập lịch AEPS này không chỉ giải quyết những mối lo ngại về sự cố “chặn đầu dòng” mà còn cung cấp thông lượng cao với thời gian hoàn thành ngắn bằng cách sử dụng tối đa băng thông của các đường truyền mạng có sẵn. AEPS gửi các gói tin đến bên nhận theo đúng thứ tự mà hiệu suất không bị ảnh hưởng bởi kích thước vùng nhớ đệm bên nhận hoặc kích thước của dữ liệu được truyền. AEPS đã được phát triển với ba mục tiêu: (1) đảm bảo tất cả gói sẽ đến bộ nhớ đệm của bên nhận; (2) sử dụng tối đa băng thông của các đường truyền mạng; (3) thời gian hoàn thành phải càng ngắn càng tốt [36].

4.2 Các phương pháp MPTCP schedule dựa vào Machine learning (ML)

Thời gian gần đây, nhiều phương pháp lập lịch dựa trên ML đã được đề xuất để cải thiện cơ chế lập lịch của MPTCP. Mặc dù các phương pháp tiếp cận cổ điển đạt được hiệu suất tốt, các phương pháp tiếp cận dựa trên ML cũng cho thấy kết quả đầy hứa hẹn và trở nên phổ biến trong việc đạt được thông lượng cao hơn với độ trễ thấp hơn so với các phương pháp không dùng ML. (F. Silva và cộng sự 2020) đã sử dụng giải thuật “hồi quy tuyến tính” (linear regression) để dự đoán thông lượng và độ trễ trong các luồng con và đề xuất bộ phân loại tuyến tính dựa trên “mạng thần kinh nhân tạo” (Artificial Neural Network) để chọn luồng con tốt nhất có thể mang lại hiệu suất tốt hơn trong bộ lập lịch MPTCP [37]. Nhóm tác giả đã triển khai nghiên cứu của mình trên phần mềm mô phỏng NS-3.

FALCON là tên gọi của một bộ lập lịch đa đường dùng ML [38]. Thuật toán này sử dụng kỹ thuật học máy “meta-learning” để thích ứng với việc thay đổi trạng thái mạng một cách nhanh chóng và chính xác. Thuật toán “meta-learning” bao gồm hai mô đun: “mô đun huấn luyện ngoại tuyến” (offline training module - FTM) và “mô đun “huấn luyện trực tuyến” (online training module - OTM). Sử dụng OTM để nắm bắt các điều kiện thay đổi mạng trong khi FTM lấy kinh nghiệm từ OTM và phân loại những kinh nghiệm này thành các nhóm khác nhau tùy thuộc vào điều kiện mạng. Kết quả cho thấy, bộ lập lịch FALCON có tốc độ nhanh hơn 8 lần so với bộ lập lịch tốt nhất ở thời điểm đó, FALCON còn cho thấy sự thích nghi nhanh với môi trường mạng biến động.

Thuật toán lập lịch gói tin cho MPTCP có tên “Bộ lập lịch gói dự phòng thích ứng theo kinh nghiệm” (Experience driven Adaptive Redundant packet scheduler - EdAR) được công bố năm 2023. EdAR cho phép lập lịch linh hoạt các gói dự phòng bằng cách sử dụng chiến lược học dựa trên kinh nghiệm để nâng cao hiệu suất truyền đa đường. Để cho phép học và dự đoán chính xác, EdAR dùng DRL để thu thập, giám sát môi trường mạng và quá trình hành động tối ưu. EdAR có hai chế độ truyền: truyền tiêu chuẩn và truyền dự phòng. Chế độ truyền tiêu chuẩn tuân theo quá trình chuyển đổi dữ liệu thường xuyên. Chế độ truyền dự phòng sử dụng một bộ đệm dự phòng chứa các gói đã được truyền đi nhưng chưa được xác nhận. Nếu một gói mới được truyền từ bộ đệm đưa vào luồng con, gói đó

sẽ được sao chép vào bộ đệm dự phòng. Nếu một gói trong bộ đệm dự phòng không được gửi đi hoặc không được xác nhận, nó sẽ bị xóa khỏi bộ đệm dự phòng. Thử nghiệm đã cho thấy thuật toán này đáp ứng nhanh chóng trong môi trường mạng biến động mạnh, [39]

## 5 Kết luận và đề xuất

Bài báo này thực hiện khảo sát hai vấn đề quan trọng của giao thức MPTCP, đó là kiểm soát tắc nghẽn và lập lịch gói tin. Các thuật toán kiểm soát tắc nghẽn và lập lịch gói tin theo hướng tiếp cận truyền thông có điểm chung là không thích ứng tốt với môi trường mạng không đồng nhất, nhiều biến động. Còn những hướng tiếp cận bằng kỹ thuật học máy hay học tăng cường,

học tăng cường sâu đã cải thiện những điểm yếu trên, nhưng bù lại, chúng phản ứng chậm hơn so với hướng tiếp cận truyền thông. Bài báo này giúp cho các nhà nghiên cứu có cái nhìn tổng quan về hiện trạng của giao thức MPTCP, tiếp tục nghiên cứu, đưa ra các thuật toán mới hoặc cải tiến các nghiên cứu đã có để giao thức MPTCP đạt ổn định và tin cậy cao. Hướng tiếp cận học máy được khuyến nghị vận dụng vào kiểm soát tắc nghẽn và lập lịch gói tin cho giao thức MPTCP.

## Lời cảm ơn

Nghiên cứu này được tài trợ bởi Quỹ phát triển Khoa học và Công nghệ - Trường Đại học Nguyễn Tất Thành, mã đề tài 2023.01.146/HĐ-KHCN.

## Tài liệu tham khảo

1. J. Postel. (1980). *RFC-768: User Datagram Protocol*. Internet Engineering Task Force (IETF),
2. J. Postel. (1981). *RFC-793: Transmission control protocol*. Internet Engineering Task Force (IETF),
3. A. Ford, C. Raiciu, M. Handley, O. Bonaventure. (2011). *Architectural Guidelines for Multipath TCP Development [RFC-6182]*. Internet Engineering Task Force (IETF). [Online]. Available: <https://www.rfc-editor.org/rfc/rfc6182>.
4. J. Iyengar, B. Ford. (2009). *A Next Generation Transport Services Architecture*,
5. L. Chao, C. Wu. (2021). *A Brief Review of Multipath TCP for Vehicular Networks*. Sensors.
6. S. Deng, R. Netravali, A. Sivaraman, H. Balakrishnan. (2014). *Wifi, LTE or both? Measuring multi-homed wireless Internet performance*. Proceedings of the (2014). Conference on Internet Measurement Conference.
7. C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik. (2011). *Improving datacenter performance, robustness with multipath TCP*. ACM SIGCOMM Computer Communication Review.
8. F. Yang, Q. Wang, P. D. Amer. (2014). *Out-of-order transmission for inorder arrival scheduling for multipath TCP*. 28th International Conference on Advanced Information Networking and Applications Workshops.
9. J. Han, K. Xue, J. Li, R. Zhuang, R. Li, R. Yu, G. Xue, Q. Sun. (2023). *EdAR: An experience-driven multipath scheduler for seamless handoff in mobile networks*. IEEE Transactions on Wireless Communication.
10. O. Ait-Hellal, E. Altman. (2000). *Analysis of TCP Vegas and TCP reno*. Telecommunication Systems.
11. S. Ha, I. Rhee, L. Xu. (2008). *CUBIC: a new TCP-friendly high-speed TCP variant*. ACM SIGOPS Operating Systems Review.
12. S. Hassayoun, J. Iyengar, D. Ros. (2011). *Dynamic window coupling for multipath congestion control*. 19<sup>th</sup> IEEE International Conference on Network Protocols.
13. C. Raiciu, M. Handley, D. Wischik. (2011). *Coupled Congestion Control for Multipath Transport Protocols Internet*. Engineering Task Force (IETF) RFC-6356.
14. T. A. Le, C. S. Hong, M. A. Razzaque, S. Lee, H. Jung. (2011). *ecMTCP: An energy-aware congestion control algorithm for multipath TCP*. IEEE Communications Letters.
15. Y. Cao, M. Xu, X. Fu. (2012). *Delay-based Congestion Control for Multipath TCP*. IEEE International Conference on Network Protocols (ICNP). IEEE Access.

16. A. Singh, M. Xiang, A. Konsgen, C. Goerg, Y. Zaki. (2013). *Enhancing fairness and congestion control in multipath TCP*. 6th joint IFIP Wireless and Mobile Networking Conference (WMNC). pp. 1-8.
17. N. Gast, R. Khalili, J.-Y. Boudec, M. Popovic. (2013). *Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP*. Internet Engineering Task Force (IETF).
18. S. Ferlin, O. Alay, T. Dreibholz, D. A. Hayes, M. Welzl. (2016). *Revisiting congestion control for multipath TCP with shared bottleneck detection*. IEEE INFOCOM, The 35th Annual IEEE International Conference on Computer Communications.
19. J. Yang, J. Han, K. Xue, Y. Wang, J. Li, Y. Xing, H. Yue, D. S. Wei. (2023). *TCCC: a throughput consistency congestion control algorithm for MPTCP in mixed transmission of long and short flows*. IEEE Transactions on Network and Service Management.
20. X. Ji, B. Han, R. Li, C. Xu, Y. Li, J. Su. (2022). *ACCeSS: adaptive QoS-aware congestion control for multipath TCP*. IEEE/ACM 30th International Symposium on Quality of Service (IWQoS).
21. A. Walid, Q. Peng, J. Hwang, S. Low. (2016). *Balanced Linked Adaptation Congestion Control Algorithm for MPTCP*. Internet Engineering Task Force (IETF).
22. T. Gilad, N. Rozen-Schiff, P. B. Godfrey, C. Raiciu, M. Schapira. (2020). *MPCC: Online learning multipath transport*. 16th International Conference on Emerging Networking Experiments and Technologies.
23. T. Zhang, S. Mao. (2020). *Machine learning for end-to-end congestion control*. IEEE Communications Magazine.
24. A. Slivkins. (2019). *Introduction to Multi-Armed Bandits*. Foundations and Trends(R) in Machine Learning. [Online]. Available: <https://arxiv.org/abs/1904.07272>.
25. W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, S. Lu. (2019). *SmartCC: A reinforcement learning approach for multipath TCP congestion control in heterogeneous networks*. IEEE Journal on Selected Areas in Communications.
26. Z. Xu, J. Tang, C. Yin, Y. Wang, G. Xue. (2019). *Experience-driven congestion control: When multi-path TCP meets deep Reinforcement learning*. IEEE Journal on Selected Areas in Communications.
27. He, J. Wang, Q. Qi, H. Sun, J. Liao, C. Du, X. Yang, Z. Han. (2021). *DeepCC: Multi-agent deep reinforcement learning congestion control for multi-path TCP based on self-attention*. IEEE Transactions on Network and Service Management.
28. S. R. Pokhrel, L. Pan, N. Kumar, R. Doss, H. L. Vu. (2021). *Multipath TCP meets transfer learning: A novel edge-based learning for industrial IoT*. IEEE Internet of Things Journal.
29. C. Paasch, S. Ferlin, O. Alay, O. Bonaventur. (2014). *Experimental evaluation of multipath TCP schedulers*. Proceedings of the ACM SIGCOMM Workshop on Capacity Sharing Workshop.
30. J. Hwang, J. Yoo. (2015). *Packet scheduling for multipath TCP*. Seventh International Conference on Ubiquitous and Future Networks.
31. T. Le, L. X. Bui. (2018). *Forward delay-based packet scheduling algorithm for multipath TCP*. Mobile Networks and Applications.
32. N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, R. Boreli. (2014). *APS: Intelligent delay-aware packet scheduling for multipath transport*. IEEE International Conference on Communications (ICC).
33. F. Yang, Q. Wang, P. D. Amer. (2014). *Out-of-order transmission for in-order arrival scheduling for multipath TCP*. 28th International Conference on Advanced Information Networking and Applications Workshops.
34. S. Ferlin, O. Alay, O. Mehani, R. Boreli. (2016). *BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks*. IFIP Networking Conference (IFIP Networking) and Workshops.
35. P. Dong, J. Xie, W. Tang, N. Xiong, H. Zhong, A. V. Vasilakos. (2019). *Performance evaluation of multipath TCP scheduling algorithms*. IEEE Access.
36. R. K. Chaturvedi, S. Chand. (2021). *An adaptive and efficient packet scheduler for multipath TCP*. Iranian Journal of Science and Technology; Transactions of Electrical Engineering.
37. F. Silva, M. Togou, G. Muntean. (2020). *An innovative machine learning approach to improve MPTCP performance*. International Conference on High Performance Computing and Simulation.

