

NGHIÊN CỨU ẢNH HƯỞNG CỦA TẬP HỢP CHÙM ĐẾN HIỆU QUẢ ĐIỀU KHIỂN CỦA SỔ LƯU LƯỢNG TCP TRONG MẠNG OBS

Lê Văn Hòa*, Phạm Trung Đức⁺

*Trường Du lịch – Đại học Huế

⁺ Công ty cổ phần Bến xe Huế

Tóm tắt: TCP là giao thức vận chuyển phổ biến nhất trên mạng Internet và có thể trong thời gian tới vẫn như vậy; trong khi, với việc triển khai rộng rãi các đường trục quang và sự trưởng thành của các mô hình mạng chuyển mạch chùm quang (mạng OBS) gần đây, một kết hợp của mạng Internet và cơ sở hạ tầng truyền thông OBS bên dưới có thể là xu hướng tiếp theo. Một nhu cầu cấp thiết được đặt ra là cần nghiên cứu và đánh giá hiệu quả của việc kết hợp TCP với mạng OBS (TCP-over-OBS). Trong bài báo này, chúng tôi tập trung vào nghiên cứu ảnh hưởng của tập hợp chùm đến hiệu quả điều khiển của sổ lưu lượng của một phiên bản khác nhau của TCP.

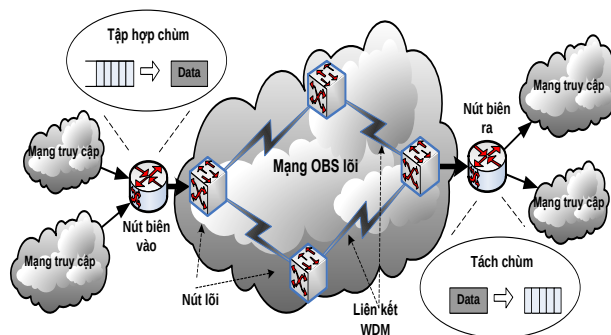
Từ khóa: Mạng chuyển mạch chùm quang, tập hợp chùm, giao thức TCP, kỹ thuật điều khiển của sổ TCP.

I. GIỚI THIỆU

Trong tiến trình phát triển của công nghệ mạng, có nhiều phương pháp khác nhau được sử dụng để truyền dữ liệu, như thông tin được vận chuyển bằng phương pháp thông thường là cáp hoặc phương pháp quang học mới. Sợi quang có thể vận chuyển dữ liệu xa hơn và băng thông cao hơn cáp điện. Phương pháp quang học sử dụng ánh sáng để mang thông tin bằng cách cung cấp các kết nối điểm-điểm. Với công nghệ ghép kênh phân chia kênh bước sóng quang (Wavelength Division Multiplexing, WDM), băng thông được chia thành sổ lượng không chồng lấp các kênh bước sóng. Các phương pháp quang học sử dụng công nghệ WDM bao gồm: chuyển mạch chùm quang (Optical Burst Switching, OBS), chuyển mạch kênh quang (Optical Circuit Switching, OCS) và chuyển mạch gói quang (Optical Packet Switching, OPS) [1].

Chuyển mạch chùm quang được xem là một thay thế hiệu quả của chuyển mạch gói quang, khi mà cần có một sự chuyển dịch từ chuyển mạch kênh quang sang chuyển mạch gói quang nhằm khai thác tốt hơn tiềm năng băng thông của các sợi dẫn quang. Hơn nữa, với những hạn chế của công nghệ quang hiện nay, như chưa thể sản xuất các bộ đệm quang (tương tự bộ nhớ RAM) và các chuyển mạch quang ở tốc độ nano giây, kỹ thuật OBS được xem như là một mô hình khả thi nhất đối với chuyển mạch gói quang trong một tương lai gần [2][3].

Một đặc trưng quan trọng của mạng OBS là gói điều khiển (Burst Control Packet, BCP) được gửi đi trước trên một kênh điều khiển dành riêng để đặt trước tài nguyên; sau một khoảng thời gian bù đắp (offset-time), chùm dữ liệu (burst) tương ứng mới được gửi theo sau trên một trong những kênh dữ liệu khả dụng (Hình 1). Bởi vì tài nguyên đã được đặt trước bởi gói điều khiển BCP, chùm dữ liệu sẽ không chịu một độ trễ nào tại mỗi nút trung gian, nên không cần bộ đệm quang. Mặt khác, kích thước của chùm là khá lớn so với các gói được mang bên trong, nên việc sử dụng các chuyển mạch có tốc độ micro giây sẽ không làm giảm hiệu quả sử dụng băng thông. Tuy nhiên, cách truyền thông này cũng đặt ra áp lực đối với việc làm thế nào để một gói điều khiển BCP đặt trước tài nguyên và cấu hình chuyển mạch thành công tại các nút lõi, đảm bảo cho việc truyền tải chùm đi sau đó.



Hình 1. Kiến trúc của mạng OBS

TCP là giao thức vận chuyển phổ biến nhất trên mạng Internet hiện nay. Bên cạnh đó, với việc triển khai rộng rãi các đường trục quang và sự trưởng thành của các mô hình mạng chuyển mạch chùm quang gần đây, một kết hợp giữa mạng Internet và cơ sở hạ tầng OBS bên dưới có thể là xu hướng phù hợp nhất. Do vậy, cần có các nghiên cứu và đánh giá hiệu quả của việc kết hợp TCP với mạng OBS (TCP-over-OBS) [4][5]. Bài báo này sẽ tập trung vào nghiên cứu ảnh hưởng của tập hợp chùm đến hiệu quả lưu lượng của một số phiên bản khác nhau của TCP (bao gồm TCP Vegas, TCP Sack, TCP Reno và TCP Tahoe). Các phần tiếp theo của bài báo này gồm: Mục 2 trình bày các nghiên cứu liên quan; Mục 3 mô tả kết quả mô phỏng về ảnh hưởng tập hợp chùm đến vấn đề điều khiển lưu lượng theo các phiên bản khác nhau của TCP và kết luận được trình bày ở Mục 4.

Tác giả liên hệ: Lê Văn Hòa,
Email: levanhoa@hueuni.edu.vn

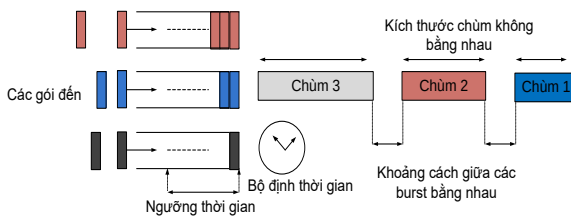
Đến tòa soạn: 10/8/2021, chỉnh sửa: 04/01/2022, chấp nhận đăng: 11/4/2022.

II. CÁC NGHIÊN CỨU LIÊN QUAN

2.1 Các kỹ thuật tập hợp chùm trong mạng OBS

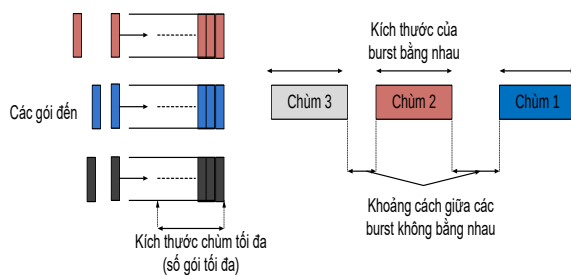
Tập hợp chùm truyền thống có ba cách tiếp cận chính là kỹ thuật tập hợp chùm dựa trên ngưỡng thời gian, ngưỡng độ dài và lai, tức là dựa trên đồng thời ngưỡng thời gian và ngưỡng độ dài [6].

Trong kỹ thuật tập hợp chùm dựa trên ngưỡng thời gian (được mô tả trong Hình 2), một bộ đếm thời gian (T_q) được khởi tạo khi có gói tin đầu tiên từ lớp mạng phía trên đến hàng đợi tập hợp chùm (q) của một nút biên vào mạng OBS. Khi bộ đếm tập hợp chùm của hàng đợi q (T_q) vượt quá một ngưỡng thời gian được xác định trước (T_{max}), các gói tin trong hàng đợi q được tập hợp lại thành một chùm. Với kỹ thuật tập hợp này, các chùm được tạo ra đều đặn sau từng khoảng thời gian bằng nhau nhưng kích thước chùm sẽ thay đổi lớn hay nhỏ tùy thuộc vào tốc độ luồng dữ liệu đến tại nút biên vào lớn hay bé.



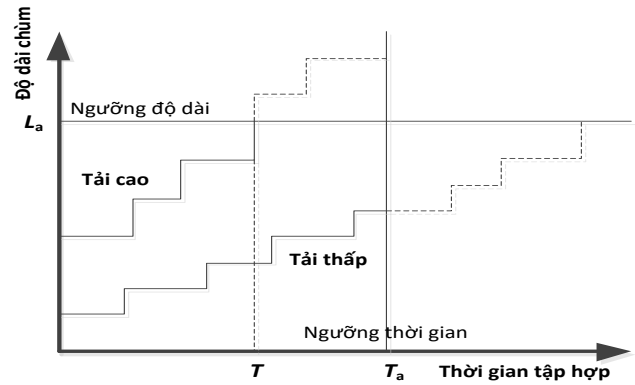
Hình 2. Tập hợp chùm theo ngưỡng thời gian

Giải thuật tập hợp chùm dựa trên ngưỡng độ dài chùm, như được mô tả trong Hình 3, thực hiện tập hợp chùm khi chiều dài của hàng đợi tập hợp chùm q (L_q) đạt đến một ngưỡng chiều dài cho trước (L_{max}). Nói một cách khác, giải thuật tập hợp chùm dựa trên ngưỡng chiều dài giới hạn số lượng gói tin chứa trong mỗi chùm, nên các chùm được tạo ra có kích thước bằng nhau, nhưng thời gian tập hợp chùm là thay đổi ngẫu nhiên. Với tập hợp chùm dựa trên ngưỡng thời gian, nếu tốc độ luồng đến nhanh thì thời gian tập hợp chùm sẽ nhỏ và ngược lại nếu tốc độ luồng đến chậm thì thời gian tập hợp chùm lớn.



Hình 3. Tập hợp chùm theo ngưỡng độ dài

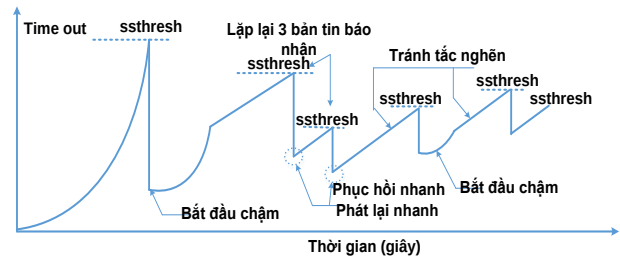
Giải thuật tập hợp chùm lai là sự kết hợp của hai giải thuật tập hợp chùm ở trên, tức là vừa dựa trên ngưỡng thời gian và ngưỡng chiều dài để thực hiện tập hợp chùm. Với ngưỡng thời gian và chiều dài cho trước, một chùm sẽ được hình thành khi một trong hai ngưỡng đạt đến trước. Như mô tả trong Hình 4, thời gian sinh ra chùm và kích thước của chùm được giới hạn trong hai ngưỡng thời gian (T_{max}) và ngưỡng độ dài (L_{max}) cho trước.



Hình 4. Tập hợp chùm theo ngưỡng lai

2.2 Điều khiển lưu lượng của TCP

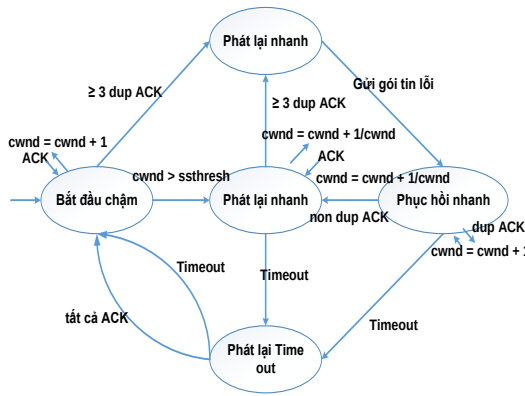
TCP là một giao thức hướng kết nối và tin cậy. Một trong những tính năng quan trọng của TCP là điều khiển cửa sổ tắc nghẽn, gồm 4 cơ chế chính: cơ chế bắt đầu chậm, cơ chế tránh tắc nghẽn, cơ chế phát lại nhanh và cơ chế phục hồi nhanh [7]. Theo đó, khi mới bắt đầu TCP sẽ chuyển sang chế độ bắt đầu chậm nghĩa là cửa sổ tắc nghẽn ($cwnd$) sẽ tăng theo cơ chế hàm mũ cho đến khi nào phát hiện mất dữ liệu (dựa trên time-out hoặc gói ACK trả về) thì cửa sổ được thiết lập lại bằng 0 và giá trị ngưỡng của cửa sổ được thiết lập lại còn một nửa, đồng thời sẽ tiến hành phát lại nhanh và phục hồi nhanh tùy theo các phiên bản khác nhau của TCP. Khi giá trị cửa sổ đạt đến ngưỡng thì sẽ chuyển sang chế độ tránh tắc nghẽn (chế độ tăng giá trị cửa sổ một cách tuyến tính), nếu nhận ba lần gói ACK thì giá trị ngưỡng tiếp tục giảm một nửa và giá trị cửa sổ tăng theo tuyến tính, như có thể nhìn thấy ở Hình 5.



Hình 5. Cơ chế điều khiển cửa sổ của giao thức TCP

Việc điều khiển cửa sổ TCP được thực hiện dựa vào 2 điều kiện: (1) mất gói (dựa vào gói ACK trả về) và (2) thời gian truyền gói tin (Round Trip Time - RTT). Cơ chế điều khiển cửa sổ TCP dựa vào mất gói có các phiên bản như TCP Tahoe, TCP Reno và TCP Sack. Với TCP Tahoe, 3 cơ chế điều khiển được sử dụng là bắt đầu chậm, tránh tắc nghẽn và phát lại nhanh như trong Hình 5. Trong cơ chế phát lại nhanh, TCP Tahoe không cần phải đợi cho đến hết thời gian time-out (mất dữ liệu) của gói tin mới phát lại gói tin bị mất, mà chỉ cần nhận 3 lần gói ACK là TCP Tahoe tiến hành phát lại gói tin này [7]. TCP Reno là một cải tiến từ TCP Tahoe trong đó ngoài cơ chế phát lại nhanh, TCP Reno còn sử dụng cơ chế phục hồi nhanh [8]. Như có thể nhìn thấy ở Hình 6.

Cơ chế phục hồi nhanh của TCP Reno dựa vào cơ chế phát lại nhanh, theo đó khi cơ chế phát lại nhanh được thực thi giá trị ngưỡng ($ssthresh$) được thiết lập bằng một nửa cửa sổ ($cwnd$) cộng với kích thước của 3 gói tin. $cwnd$ được thêm vào mỗi gói với một lần nhận được ACK. Khi ACK của gói phát lại được nhận, $cwnd$ thiết lập đến



Hình 6. Cơ chế hoạt động của TCP Reno

$ssthresh$ và trở về giai đoạn tránh tắc nghẽn. Khi khởi tạo lại giá trị, $cwnd$ được thiết lập lại bằng $\frac{1}{2}$ giá trị của $cwnd$ sau khi đã phục hồi nhanh [8].

Một cải tiến tiếp theo của TCP Reno là TCP Sack, trong đó ở cơ chế phục hồi nhanh TCP Sack còn cho phép các gói tin được truyền đồng thời theo cơ chế đường ống (pipeline) [9].

Vấn đề điều khiển cửa sổ tắc nghẽn dựa vào quan sát thời gian RTT của các gói tin tiêu biểu có giao thức TCP Vegas. Theo [10][11], RTT là khoảng thời gian được tính từ khi gói tin được gửi từ trạm phát đến trạm nhận, cho đến khi trạm phát nhận được ACK hồi đáp chứa thông tin về gói tin đã nhận được thành công. Nếu thời gian của RTT được theo dõi tăng, thì TCP Vegas nhận biết mạng sắp bị tắc nghẽn và thực hiện cơ chế tránh tắc nghẽn. Nếu thời gian của RTT giảm thì TCP Vegas nhận biết mạng đang được khai thông và TCP Vegas thực hiện cơ chế tăng kích thước cửa sổ để tận dụng thông lượng của đường truyền. Thuật toán TCP Vegas được thực hiện như sau:

Gọi D là thời gian RTT được theo dõi, d là giá trị nhỏ nhất của RTT được theo dõi, α và β là các giá trị hằng số trong TCP Vegas, t và $(t+1)$ là thời điểm thực hiện.

Đặt giá trị $diff = w(t)/d - w(t)/D$, cửa sổ TCP Vegas được điều chỉnh như sau:

$$w(t+1) = \begin{cases} w(t) + 1 & \text{nếu } diff < \frac{\alpha}{d} \\ w(t) & \text{nếu } \frac{\alpha}{d} \leq diff \leq \frac{\beta}{d} \\ w(t) - 1 & \text{nếu } diff > \frac{\beta}{d} \end{cases}$$

Có một số tính năng riêng của mạng OBS ảnh hưởng đáng kể đến hiệu năng của TCP, làm cho mạng OBS khác biệt với mạng chuyển mạch gói điện tử truyền thống. Đó là các gói điện tử TCP khi đến tại các nút biên vào mạng OBS, chúng sẽ được làm trễ (chờ đợi) để được tập hợp thành chùm và được gửi vào mạng OBS sau một khoảng thời gian chờ bù đắp. Với quy trình này, khả năng nhận đúp ACK ở tầng TCP là rất cao và cơ chế điều khiển TCP sẽ giới hạn thông lượng gửi vào mạng (bằng cách giảm cửa sổ lưu lượng), trong khi khả năng đáp ứng của mạng OBS bên dưới là chưa vượt quá khả năng băng thông.

2.3 Kết hợp TCP qua mạng OBS

Kostas Ramantas và các cộng sự trong [7] đã nghiên cứu kết hợp của giao thức TCP Sack và cơ chế tập hợp chùm

theo ngưỡng thời gian. Kostas Ramantas thiết lập các giá trị ngưỡng thời gian khác nhau để xem xét tỉ lệ mất chùm của chúng cũng như cơ chế điều khiển cửa sổ với giao thức TCP Sack. Shuping Peng và các cộng sự trong [8] đề xuất một giải thuật tập hợp chùm động có tên là DAP (Dynamic assembly period) nhằm tăng thông lượng đạt được với TCP Reno.

Đối với giao thức TCP Vegas đã có khá nhiều nghiên cứu gần đây kết hợp giữa TCP Vegas qua mạng OBS nhằm tăng thông lượng đạt được của mạng. Bassem Shihada và các cộng sự trong [12] nghiên cứu việc điều khiển linh động giá trị cửa sổ ($cwnd$) đối với giao thức TCP Vegas bằng cách sử dụng giải thuật dựa trên ngưỡng, nhằm xác định tình trạng tắc nghẽn trong mạng. Cụ thể thuật toán này dựa trên việc theo dõi giá trị RTT nhằm điều chỉnh kích thước cửa sổ $cwnd$. Một tham số T được sử dụng là số lượng các RTT theo dõi lớn hơn so với giá trị $MinRTT$ trong N vòng quan sát, nếu T lớn hơn giá trị ngưỡng thì tình trạng mạng đang tắc nghẽn và ngược lại mạng không tắc nghẽn nhằm tăng giảm $cwnd$.

Poorzare và các cộng sự [13] đề xuất một lược đồ cơ sở mờ nhằm phân biệt các nguyên nhân mất chùm trong mạng OBS. Tác giả sử dụng 2 tập mờ: tập đánh rơi do tắc nghẽn và tập đánh rơi do tranh chấp. Khi một chùm bị đánh rơi [14] sẽ tiến hành đo lường giá trị RTT của gói rơi so sánh với $BaseRTT$ của 2 tập đánh rơi do tắc nghẽn và tập đánh rơi do tranh chấp nhằm điều chỉnh $cwnd$. Một cải tiến tiếp theo của Poorzare và các cộng sự trong [15] đưa ra giải thuật FRPI nhằm nâng cao hiệu năng của mạng. Cụ thể FRPI dựa trên 2 giai đoạn: Giai đoạn 1 sử dụng giải thuật fuzzy logic để xác định tình trạng mạng và lưu lượng traffic trong mạng nhằm xác định tốc độ gửi gói tin. Giai đoạn 2 tác giả sử dụng một số giá trị ngưỡng và so sánh với RTT nhằm xác định tăng, giảm hoặc để nguyên cửa sổ $cwnd$.

Một kết hợp giữa điều chỉnh kích thước $cwnd$ với điều khiển tắc nghẽn được Abedidarabad và cộng sự [16] đưa ra, theo đó Abedidarabad phân biệt giữa chùm gây ra tắc nghẽn và chùm không gây ra tắc nghẽn nhằm điều khiển lưu lượng TCP Vegas tốt hơn.

Poorzare và các cộng sự [17] đề xuất giải thuật AVGR thực hiện theo 2 bước. Bước đầu tiên của quy trình này là đo lường giá trị của RTT lần lượt. Để xác định điểm kết thúc tính toán, một ngưỡng được sử dụng và khi đạt đến ngưỡng này, quy trình đo sẽ dừng lại và giá trị trung bình $AvgRTT$ của chúng được tính toán. Giai đoạn thứ hai thuật toán cố gắng điều chỉnh hành vi của mạng bằng cách sử dụng năm phương trình khác nhau. Trong giai đoạn này, thuật toán chia mạng thành năm khu vực làm việc khác nhau và dựa trên khu vực hiện tại, quyết định phản ứng thích hợp được thực hiện khi so sánh giá trị $AvgRTT$ với các giá trị ngưỡng nhằm tiến hành tăng giảm giá trị $cwnd$. Một bảng so sánh các đề xuất như được thể hiện trong Bảng 1.

Tuy nhiên, các đề xuất này chỉ sử dụng một phiên bản TCP riêng biệt chưa có các so sánh và đánh giá về tỉ lệ mất chùm, cũng như khả năng sử dụng băng thông của các phiên bản khác nhau TCP trong tập hợp chùm, và ảnh hưởng của tập hợp chùm đến vấn đề điều khiển lưu lượng cửa sổ với các phiên bản TCP.

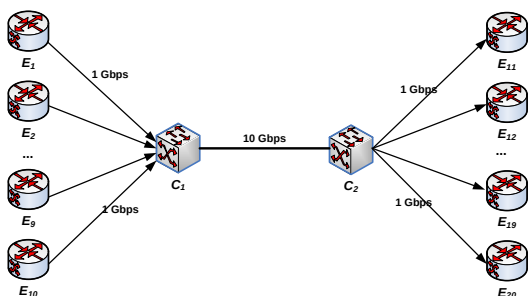
Do đó, bài báo này sẽ xem xét đến thông lượng đạt được của các phiên bản khác nhau của TCP trên mạng OBS khi tiến hành thay đổi thời gian tập hợp chùm.

Bảng 1. So sánh các đề xuất về kết hợp TCP qua mạng OBS

STT	Tác giả	Năm xuất bản	Ý tưởng chính	Xem xét ảnh hưởng OBS đến TCP	Điều khiển giá trị cửa sổ <i>cwnd</i>
1	K. Ramantas [7]	2008	Kết hợp giao thức TCP Sack và cơ chế tập hợp chùm theo ngưỡng thời gian nhằm điều khiển lại cửa sổ <i>cwnd</i> .	✓	✓
2	S. Peng [8]	2009	Đề xuất giải thuật tập hợp chùm động có tên là DAP nhằm tăng thông lượng đạt được với TCP Reno.	✓	
3	Shihada [12]	2010	Theo dõi giá trị <i>RTT</i> nhằm xác định tình trạng mạng để truyền lại kịp thời gói tin.	✓	
4	Poorzare [13]	2015	Sử dụng lược đồ cơ sở mở nhằm xác định nguyên nhân mất chùm trong OBS nhằm điều chỉnh <i>cwnd</i> .		✓
5	Poorzare [15]	2018	Sử dụng tập mở nhằm xác định tình trạng và lưu lượng mạng nhằm xác định tăng, giảm hoặc giữ nguyên <i>cwnd</i> .		✓
6	Abedidarabad [16]	2019	Phân biệt giữa chùm gây ra tắc nghẽn với chùm không gây ra tắc nghẽn nhằm điều chỉnh <i>cwnd</i> .		✓
7	Poorzare [17]	2021	Tiến hành đo lường giá trị độ trễ trung bình <i>AvgRTT</i> , đề tiến hành so sánh với các giá trị ngưỡng nhằm điều chỉnh linh động <i>cwnd</i> .		✓

III. MÔ PHỎNG VÀ PHÂN TÍCH KẾT QUẢ

Mô phỏng được thực hiện trên một PC với 2.4 GHz Intel Core 2 CPU, 2G RAM. Các gói tin đến tại hàng đợi của một nút biên mạng OBS có phân bố Poisson. Mô hình mạng được xem xét là Dumbell (Hình 7) bằng thông tin mỗi liên kết là 10GB/s, số kênh được xem xét là 8 trong đó có 2 kênh điều khiển và 6 kênh dữ liệu. Mô phỏng được thực hiện trong vòng 10s trong NS2 [18] với gói hỗ trợ mạng OBS obs-0.9a. Với mục tiêu mô phỏng là xem xét thông lượng đạt được của TCP Vegas, TCP Sack, TCP Reno và TCP Tahoe khi tiến hành thay đổi thời gian tập hợp chùm.



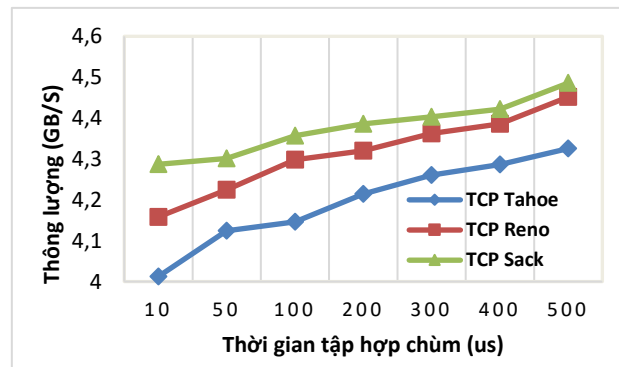
Hình 7. Mô hình mạng Dumbell dành cho mô phỏng

Như đã được phân tích ở trong Mục 2, vấn đề điều khiển cửa sổ tắc nghẽn TCP được phân thành 2 loại chính là dựa vào tỉ lệ mất gói tin và dựa vào thời gian quan sát *RTT* của các gói tin. Với vấn đề điều khiển cửa sổ tắc nghẽn

TCP dựa vào việc mất gói tin chúng tôi tiến hành mô phỏng với 3 phiên bản của TCP là TCP Tahoe, TCP Reno và TCP Sack. Trong khi điều khiển cửa sổ tắc nghẽn dựa vào quan sát *RTT* của các gói tin chúng tôi tiến hành mô phỏng với giao thức TCP Vegas.

3.1 Điều khiển cửa sổ tắc nghẽn dựa vào việc mất gói tin

Chúng tôi tiến hành xem xét thông lượng đạt được của 3 phiên bản TCP khi tải đến tại nút biên vào là 0,5 và thời gian tập hợp chùm thay đổi từ 10μs đến 500μs; kết quả thu được được mô tả ở Hình 8.

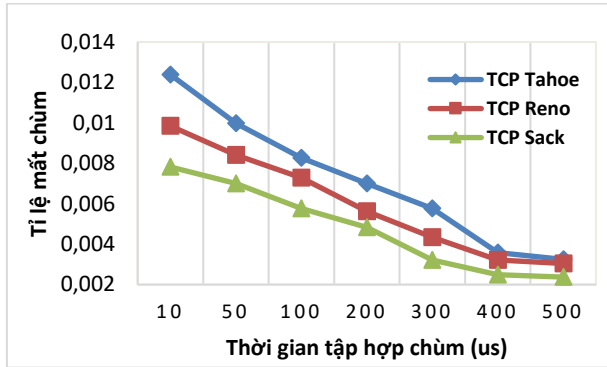


Hình 8. Thông lượng đạt được của TCP Tahoe, TCP Reno, và TCP Sack.

Như được chỉ ra ở Hình 8, thông lượng của các phiên bản TCP có xu hướng tăng khi thời gian tập hợp chùm tăng. Trong tất cả thời gian tập hợp chùm xem xét thì TCP Sack cho thông lượng đạt được tốt nhất hơn TCP Reno khoảng 4% và gần 10% so với TCP Tahoe. Để hiểu rõ hơn về vấn

đề này chúng tôi tiến hành xem xét tỉ lệ mất chùm của 3 phiên bản TCP này.

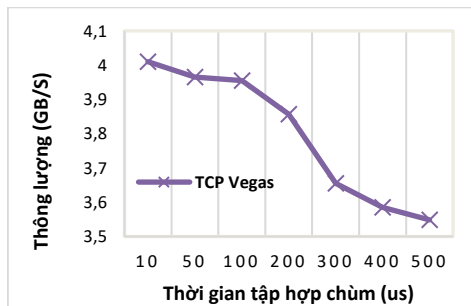
Như được chỉ ra ở Hình 9 nguyên nhân mà TCP Sack cho thông lượng tốt nhất ở 3 phiên bản là do tỉ lệ mất chùm của TCP Sack thấp nhất ở 3 phiên bản, điều này sẽ hạn chế đóng cửa sổ tắc nghẽn hơn so với 2 phiên bản TCP Tahoe và TCP Reno. Khi thời gian tập hợp chùm tăng số lượng chùm sinh ra sẽ ít điều này dẫn đến tỉ lệ mất chùm giảm nên thông lượng đạt được ở 3 phiên bản có xu hướng tăng khi thời gian tập hợp chùm tăng.



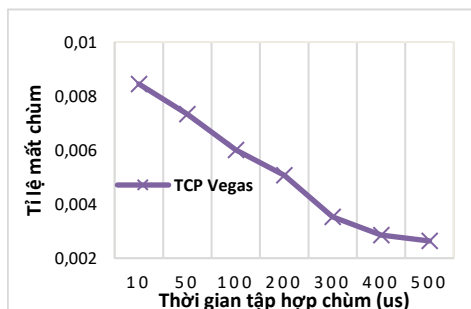
Hình 9. Tỉ lệ mất chùm của TCP Tahoe, TCP Reno và TCP Sack khi thời gian tập hợp chùm thay đổi.

3.2 Điều khiển cửa sổ tắc nghẽn dựa vào việc quan sát RTT của các gói tin

Chúng tôi tiến hành mô phỏng với giao thức TCP Vegas với tải đến 0,5 thời gian tập hợp chùm thay đổi từ 10μs đến 500μs để xem xét thông lượng và tỉ lệ mất chùm đạt được, kết quả như được chỉ ra ở Hình 10.



(a) Thông lượng đạt được



(b) Tỉ lệ mất chùm

Hình 10. So sánh TCP Vegas khi thời gian tập hợp chùm thay đổi

Như có thể thấy ở Hình 10(a) thông lượng của TCP Vegas có xu hướng giảm khi thời gian tập hợp chùm tăng, điều này trái ngược với 3 phiên bản TCP Tahoe, TCP Reno và TCP Sack (Như Hình 8). Trong khi ở Hình 10(b) tỉ lệ mất chùm của TCP Vegas có xu hướng giảm như 3 phiên bản TCP Tahoe, TCP Reno và TCP Sack.

Điều này là do TCP Vegas điều khiển cửa sổ tắc nghẽn dựa vào thời gian quan sát RTT của các gói tin. Nên khi tăng thời gian tập hợp chùm thì thời gian quan sát RTT cũng tăng, chính điều này làm cho giao thức TCP Vegas tiến hành giảm cửa sổ tắc nghẽn do nghĩ rằng mạng có khả năng tắc nghẽn. Trong khi 3 phiên bản TCP Tahoe, TCP Reno và TCP Sack điều khiển cửa sổ dựa vào việc mất gói tin nên khi tăng thời gian tập hợp chùm tỉ lệ mất chùm giảm làm cho 3 phiên bản TCP Tahoe, TCP Reno và TCP Sack có xu hướng tăng kích thước cửa sổ.

IV. KẾT LUẬN

Bài viết này đã xem xét ảnh hưởng của thời gian tập hợp chùm trong mạng OBS đến hiệu quả hoạt động của các phiên bản khác nhau của TCP. Với các phiên bản TCP điều khiển kích thước cửa sổ tắc nghẽn dựa vào việc mất gói tin như TCP Tahoe, TCP Reno và TCP Sack thì việc tăng thời gian tập hợp chùm làm cho thông lượng đạt được cao hơn do tỉ lệ mất chùm thấp hơn như được chỉ ra ở Hình 8 và 9. Tuy nhiên, với việc sử dụng giao thức TCP Vegas của sổ tắc nghẽn được điều khiển dựa vào thời gian quan sát RTT của gói tin thì việc tăng thời gian tập hợp chùm sẽ làm thông lượng đạt được của TCP. Rõ ràng, không có một giá trị thời gian tập hợp chùm nào là phù hợp với tất cả các phiên bản khác nhau của TCP, mà tùy thuộc vào từng phiên bản TCP khác nhau sẽ cho ta một ngưỡng thời gian tập hợp chùm phù hợp trong mạng OBS.

TÀI LIỆU THAM KHẢO

- [1] T. H. Dao, "Leveraging the Survivable All-Optical WDM Network Design With Network Coding Assignment," IEEE Communications Letters, vol. 21, no. 10, pp. 2190-2193, 2017.
- [2] Q. Chunming, and Y. Myungsik, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet," Journal of High Speed Networks, vol. 8, pp. 69-84, 1999.
- [3] C. Yang, Q. Chunming, and Y. Xiang, "Optical burst switching: A new area in optical networking research," IEEE Network, vol. 18, pp. 16-23, 2004.
- [4] J. Suliman, N. Charbonneau, and V.M. Vokkarane, "Performance Evaluation of TCP over Optical Burst Switched (OBS) Networks using Coordinated Burst Cloning and Forward Segment Redundancy," in: IEEE Global Telecommunications Conference GLOBECOM 2010, Miami, FL, USA.
- [5] N. Charbonneau, D. Chandra, and V. Vokkarane, "Improving TCP Performance over Optical Burst-Switched Networks Using Forward Segment Redundancy," Photonic Network Communications, vol. 23, no.1, pp. 1-15, 2012.
- [6] X. Yu, Y. Chen, and C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks," Proceeding of Opticomm, pp. 149-159, 2002.
- [7] Kostas Ramantas, Kyriakos Vlachos, Óscar González de Dios, and Carla Raffaelli, "Window-based burst assembly scheme for TCP traffic over OBS," Journal of Optical Networking, Vol. 7, Issue 5, pp. 487-495, 2008.
- [8] Shuping Peng et al, "TCP Window-Based Flow-Oriented Dynamic Assembly Algorithm for OBS Networks," Journal of Lightwave Technology, vol. 27, no. 6, March 15, 2009.

- [9] Maurizio Casoni, "TCP window estimation for burst assembly in OBS networks," The IEEE symposium on Computers and Communications, 22-25 June 2010, Italy.
- [10] L. S. Brakmo, et al, "TCP Vegas: new techniques for congestion detection and avoidance," SIGCOMM '94 Proceedings of the conference on Communications architectures, protocols and applications, Pages 24-35, London, United Kingdom, 1994.
- [11] O. Leon, and S. Sallent, "Issues in TCP Vegas over Optical Burst Switching Networks," in: 32nd IEEE Conference on Local Computer Networks (LCN 2007), vol.1, pp. 242 – 243, 2007.
- [12] B. Shihada, Q. Zhang, P.H. Ho, and J. P. Jue, " A novel implementation of TCP Vegas for optical burst switched networks," Optical Switching and Networking , vol. 7, pp. 115–126, 2010.
- [13] R. Poorzare, and S. Abedidarabad, "Improving performance of optical networks by using FRPI algorithm," J Opt Commun, 2018. <https://doi.org/10.1016/j.osn.2010.04.002>.
- [14] Kostas Ramantas, Kyriakos Vlachos, Óscar González de Dios, and Carla Raffaelli, "Window-based burst assembly scheme for TCP traffic over OBS," Journal of Optical Networking, Vol. 7, Issue 5, pp. 487-495, 2008.
- [15] R. Poorzare, and S. Jamali, "Optimizing TCP Vegas for optical networks: a fuzzy logic approach," (IJCSIS) Int J Comput Sci Inf Secur, vol. 13, pp. 33–45, 2015.
- [16] S. Abedidarabad, and R. Poorzare, "Improving Performance of Optical Networks by a Probable Approach ," J Opt Commun, 2019.
- [17] R. Poorzare, A. Calveras, and S. Abedidarabad, "An improvement over TCP Vegas to enhance its performance in optical burst switching networks," Optical Review, vol. 28, pp. 215–226, 2021. <https://doi.org/10.1007/s10043-021-00652-w>
- [18] <https://www.isi.edu/nsnam/ns/>

RESEARCH THE EFFECT OF BURST ASSEMBLY ON THE EFFECTIVENESS OF WINDOW CONTROL OF TCP PROTOCOL IN OBS NETWORKS

Abstract: TCP is the most popular transport protocol on the Internet and will likely remain so in the future; while, with the widespread deployment of optical backbones and the recent maturation of optical burst switched network (OBS networks) models, a mix of the Internet and the underlying OBS communication infrastructure has could be the next trend. There is an urgent need to study and evaluate the effectiveness of combining TCP with OBS (TCP-over-OBS) network. In this paper, we focus on studying the influence of burst assembly at the ingress nodes of the OBS networks on the traffic window control efficiency of several different versions of TCP protocol.

Keywords: OBS networks, burst assembly, TCP protocol, TCP window control.



Lê Văn Hòa, Nhận bằng Tiến sĩ Khoa học máy tính năm 2020 tại Trường Đại học Khoa học, Đại học Huế. Hiện công tác tại Trường Du lịch – Đại học Huế. Lĩnh vực nghiên cứu: Mạng máy tính, IoT, *Mạng chuyển mạch chùm quang*, Đa dạng chất lượng dịch vụ (QoS).
Email: levanhhoa@hueuni.edu.vn



Phạm Trung Đức Nhận bằng Tiến sĩ Khoa học Máy tính năm 2021 tại Trường Đại học Khoa học, Đại học Huế. Các lĩnh vực nghiên cứu: *Mạng OBS, QoS trong mạng OBS, điều khiển chấp nhận lập lịch, TCP*.
Email: phamtrungduc@hueuni.edu.vn