

ẢNH HƯỞNG CỦA TÍNH CHẤT LUỒNG DỮ LIỆU ĐẾN HIỆU QUẢ TẬP HỢP CHÙM GIẢM ĐỘ TRỄ TẠI NÚT BIÊN MẠNG OBS

Lê Văn Hòa¹, Võ Viết Minh Nhật¹, Nguyễn Hoàng Sơn²

¹ Đại học Huế

² Trường Đại học Khoa học, Đại học Huế

levanhhoa@hueuni.edu.vn, vvmnhat@hueuni.edu.vn, nhson@hueuni.edu.vn

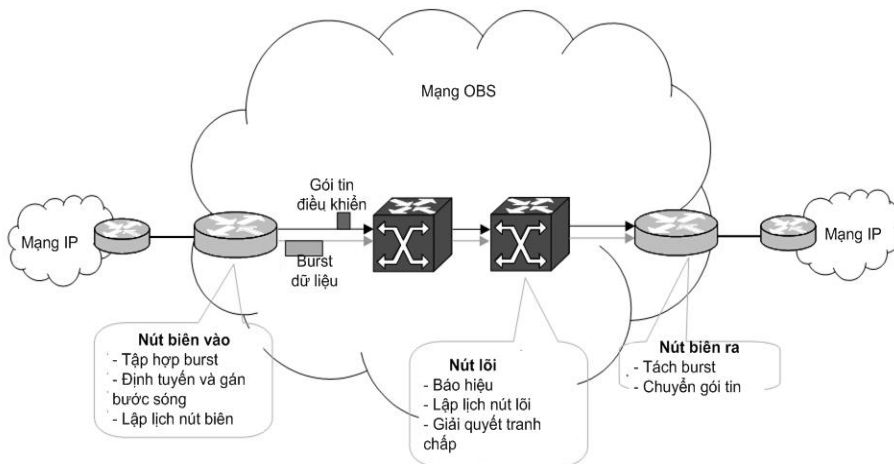
TÓM TẮT: Độ trễ đầu cuối của một chùm quang (burst) khi được truyền qua mạng chuyển mạch chùm quang chủ yếu là do 4 thành phần: (1) độ trễ tập hợp chùm tại nút biên vào, (2) thời gian bù đắp cho việc đặt trước tài nguyên của gói điều khiển, (3) độ trễ chuyển tiếp chùm tại các nút lõi và (4) độ trễ truyền bá bên trong mạng lõi. Hai độ trễ (3) và (4) thường phụ thuộc vào đường đi đã lựa chọn và băng thông khả dụng trên đường đi này, nên không thể giảm được với một giao thức đã được cài đặt. Chỉ có hai độ trễ (1) và (2), độ trễ tập hợp và thời gian bù đắp, là có thể giảm được. Kết hợp của hai độ trễ này có tên gọi chung là độ trễ đệm chùm. Đã có một số đề xuất nhằm giảm độ trễ đệm chùm này bằng cách gửi sớm gói điều khiển trước khi chùm được hoàn thành và ước đoán thông tin kích thước chùm được mang trong gói điều khiển. Bài báo này sẽ phân tích tác động của tính chất luồng (gói tin) dữ liệu đến đối với việc ước tính độ dài chùm hoàn thành trong tập hợp chùm giảm độ trễ tại nút biên mạng OBS.

Từ khóa: Mạng chuyển mạch chùm quang, tập hợp chùm, giảm độ trễ, dự báo dựa trên tốc độ gói tin, lỗi ước tính.

I. GIỚI THIỆU

Chuyển mạch chùm quang (Optical Burst Switching, OBS) 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 (như 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 tương lai gần [1, 2].

Một đặc trưng quan trọng của mạng OBS là gói điều khiển (Burst Header Packet, BHP) 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 BHP, 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 BHP đặt trước tài nguyên và cấu hình chuyển mạch thành công tại các nút lõi, đảm bảo cho việc truyền tải chùm quang đi sau đó.



Hình 1. Kiến trúc của mạng OBS

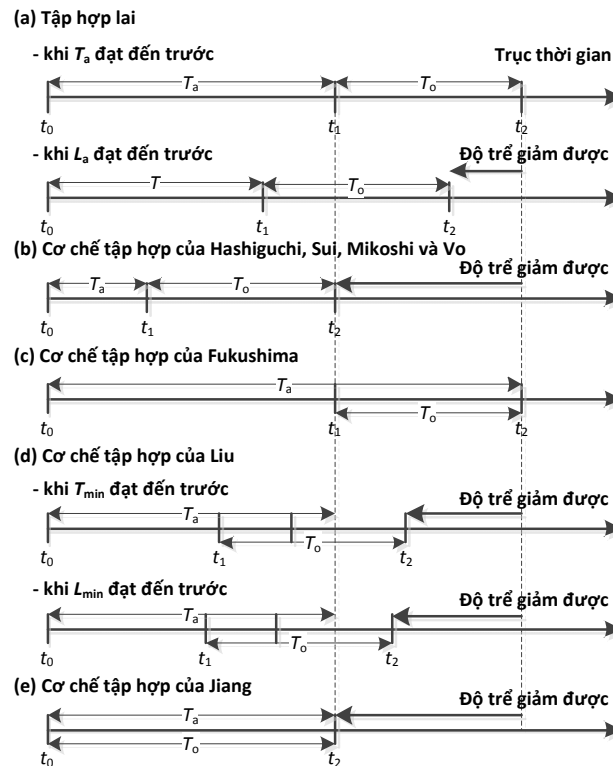
Độ trễ đầu cuối của một chùm dữ liệu được truyền qua mạng OBS bị chịu tác động chủ yếu từ 4 thành phần: (1) độ trễ tập hợp chùm tại nút biên vào, (2) thời gian bù đắp để đặt trước tài nguyên của gói điều khiển, (3) độ trễ chuyển tiếp chùm tại các nút lõi và (4) độ trễ truyền bá bên trong mạng lõi. Hai độ trễ (3) và (4) thường phụ thuộc vào đường đi đã được lựa chọn và băng thông khả dụng trên đường đi này, nên không thể giảm được với một giao thức đã được cài đặt trước. Chỉ có 2 độ trễ (1), độ trễ tập hợp, và (2), thời gian bù đắp, là có thể giảm được. Kết hợp của hai độ trễ này được đặt tên gọi chung là độ trễ đệm chùm (buffering delay).

Đã có một số giải pháp khác nhau được công bố nhằm làm giảm độ trễ đầu cuối dựa trên hoạt động tập hợp chùm tại nút biên, trong đó ý tưởng chính là gửi gói điều khiển đi sớm trước khi chùm dữ liệu tương ứng được hoàn thành. Cách làm này làm giảm đáng kể độ trễ đệm chùm, nhưng cần phải ước tính độ dài của chùm sẽ được hoàn thành bởi vì thông tin này phải được mang trong gói điều khiển. Tuy nhiên, cách tiếp cận này sẽ gây ra lỗi ước tính và có ảnh hưởng đáng kể đến độ trễ của các giải thuật. Bài báo này sẽ đánh giá các giải pháp về vấn đề tập hợp chùm giảm độ trễ bằng cách gửi sớm gói điều khiển và phân tích tác động của tính chất luồng (gói tin) dữ liệu đến đối với việc ước tính độ dài chùm hoàn thành trong tập hợp chùm giảm độ trễ tại nút biên mạng OBS.

Các phần tiếp theo của bài báo này gồm: Mục II tóm lược và đánh giá các phương pháp tập hợp chùm giảm độ trễ đã công bố, trong đó tập trung vào vấn đề ảnh hưởng của tính chất luồng đến hiệu quả ước tính độ dài chùm hoàn thành. Mục III xem xét chi tiết đề xuất của các tác giả trong [10] liên quan đến việc điều chỉnh động trong số α trong ước tính độ dài chùm dựa trên phương pháp TW-EWMA [12]. Mục IV mô tả kết quả mô phỏng về ảnh hưởng của tính chất luồng đến hiệu quả ước tính độ dài chùm hoàn thành và kết luận ở Mục V.

II. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Tập hợp chùm (burst assembly) truyền thống có ba cách tiếp cận chính là tập hợp chùm dựa trên ngưỡng thời gian [3], dựa trên ngưỡng độ dài [3] và lai, tức là dựa trên cả ngưỡng thời gian và ngưỡng độ dài. Một ví dụ được mô tả trong Hình 2a chỉ ra rằng tập hợp chùm lai có thể giảm được độ trễ nếu mật độ dữ liệu cao đến tại hàng đợi tập hợp chùm và ngưỡng độ dài đạt đến trước. Tuy nhiên, nếu ngưỡng thời gian T_a đạt đến trước, không có độ trễ nào được giảm.



Hình 2. Tập hợp chùm lai và các giải pháp tập hợp chùm giảm độ trễ đã công bố

Giải pháp tập hợp chùm giảm độ trễ đầu tiên là được đề xuất bởi Hashiguchi và cộng sự (CS) [4], trong đó gói điều khiển được gửi đi tại thời điểm t_1 (xem Hình 2b) trước khi chùm được hoàn thành. Chùm sẽ được gửi đi ngay tại thời điểm chùm được hoàn thành t_2 và thời gian đệm chùm lúc này chỉ còn là T_a , giảm so với tập hợp chùm truyền thống một khoảng thời gian T_o . Để ước tính độ dài chùm được hoàn thành, Hashiguchi đề xuất công thức sau:

$$L_e = \delta \times L_w \times \frac{T_a}{T_a - T_o}, \quad (1)$$

trong đó L_w là độ dài chùm thực tế đo được trong khoảng thời gian ước tính ($T_a - T_o$) và δ là một tham số điều khiển.

Các giải pháp tập hợp chùm trong [5] và [6] cũng tương tự với ý tưởng của Hashiguchi, nhưng khác ở phương pháp ước tính độ dài chùm. Cụ thể, Sui và CS [5] sử dụng một hàm lọc tuyến tính hồi qui tự động thích nghi (Auto Adaptive Regression Linear Filter) của M lần tập hợp chùm trước đó kết hợp với tốc độ gói tin đến trong khoảng thời gian $T_a - T_o$ để ước tính độ dài chùm, được mô tả bởi công thức sau:

$$L_e = \sum_{i=1}^{M-1} w(i) \times L(i) + \delta \times L_w \times \frac{T_a}{T_a - T_o}, \quad (2)$$

trong đó $L(i)$ là độ dài chùm đo được ở lần tập hợp thứ i , $1 \leq i \leq M$, và $w(i)$ là trọng số tác động của lần tập hợp này. Lưu ý rằng $\delta = w(M)$ và $\sum_{i=1}^M w(i) = 1$.

Với đề xuất của Mikoshi và CS [6], việc ước tính độ dài chùm được dựa trên thuật toán Jacobson/Karels [11]:

$$\begin{aligned} E(n) &= L(n) - L_e(n) \\ L_e(n+1) &= L_e(n) + \delta \times E(n) \\ D(n+1) &= D(n) + \delta \times (|E(n)| - D(n)) \\ L_e(n+1) &= \mu \times L_e(n+1) + \varphi \times D(n+1) \end{aligned} \quad (3)$$

trong đó $E(n)$ là lỗi ước tính của lần tập hợp thứ n , được tính là khoảng cách giữa độ dài chùm thực tế đo được $L(n)$ so với độ dài chùm ước tính $L_e(n)$; $D(n)$ là độ lệch của lần tập hợp chùm thứ n ; và δ, μ, φ là các tham số điều khiển.

Với một cách tiếp cận khác, Fukushima và CS [7] cho phép tiếp tục tập hợp các gói tin đến trong khoảng thời gian bù đắp T_o vào chùm hiện thời (Hình 2c) và đề xuất cách ước tính độ dài chùm dựa trên tốc độ đến trung bình λ_{avg} của M gói tin đến sau cùng nhất (công thức 4), trong đó L là độ dài chùm đo được trong khoảng thời gian T_a .

$$L_e = L + \lambda_{avg} \times T_o \quad (4)$$

Thực tế, giải thuật tập hợp chùm của Fukushima cũng tương tự như các đề xuất của Hashiguchi, Sui và Mikoshi nếu xem xét thời gian tập hợp chùm lúc này là $T_a + T_o$, thay vì chỉ là T_a .

Khác với các cách tiếp cận trên đều sử dụng phương pháp tập hợp chùm dựa trên ngưỡng thời gian, Liu [8] đề xuất một giải thuật tập hợp chùm lai, trong đó gói điều khiển sẽ được gửi đi nếu thời gian tập hợp chùm đạt đến một ngưỡng thời gian tối thiểu xác định trước T_{min} hoặc độ dài chùm đạt đến một ngưỡng độ dài tối thiểu xác định trước L_{min} ; chùm dữ liệu tương ứng chỉ được hoàn thành khi ngưỡng thời gian T_a đạt đến. Độ dài chùm được hoàn thành trong đề xuất của Liu được ước tính dựa trên khác biệt giữa tốc độ gói tin đến trong lần tập hợp chùm hiện thời (λ_{cur}) so với lần tập hợp chùm trước đó (λ_{pre}) như sau:

$$L_e = L + \left(\lambda_{pre} + (\lambda_{cur} - \lambda_{pre}) \times \frac{T_o}{T + T_o} \right) \times T_o, \quad (5)$$

trong đó cửa sổ thời gian $T + T_o$ là khoảng thời gian giữa 2 thời điểm hoàn thành chùm liên tiếp. Nếu ngưỡng thời gian T_{min} đạt đến trước, thì $T = T_a$ và giải thuật tập hợp của Liu tương đương với giải thuật tập hợp dựa trên ngưỡng thời gian truyền thống (Hình 2a); trong khi nếu ngưỡng độ dài L_{min} đạt đến trước (xem Hình 2d) thì các gói tin được tập hợp trong chùm hiện thời giảm được một độ trễ là $T_a + T_o - t_1$.

Jiang [9] cũng đề xuất một giải thuật tập hợp lai, nhưng gói điều khiển được gửi ngay khi có gói tin đầu tiên đến tại hàng đợi rỗng. Một ưu điểm của đề xuất của Jiang là ngưỡng thời gian và ngưỡng độ dài được điều chỉnh một cách linh động. Cụ thể, ngưỡng thời gian của lần tập hợp tiếp theo (T_a) là trung bình ngưỡng thời gian của M lần tập hợp trước đó ($T(i)$, $1 \leq i \leq M$), được tính bởi công thức

$$T_a = \frac{\sum_{i=1}^M T(i)}{M} \quad (6)$$

và ngưỡng độ dài (chính là độ dài chùm ước tính L_e) của lần tập hợp tiếp theo được tính toán dựa trên một cặp ngưỡng độ dài chùm tối thiểu và tối đa (L_{min}, L_{max}) như sau:

$$L_e = L_{min} + step \times \frac{(L_{max} - L_{min})}{N} \quad (7)$$

trong đó $step$ là bước điều chỉnh, $1 \leq step \leq N$, và N là tổng bước điều chỉnh.

Như vậy, lỗi ước tính trong giải thuật tập hợp của Jiang có thể được triệt tiêu ($E = 0$) nếu ngưỡng độ dài (L_e) đạt đến trước nhưng chỉ trong trường hợp kích thước các gói tin đến là bằng nhau hoặc L_e là bội số của kích thước các gói tin đến. Trong trường hợp ngưỡng thời gian T_a đạt đến trước, luôn tồn tại một lỗi ước tính nhất định, $E = L - L_e$. Hơn nữa, đề xuất của Jiang luôn luôn đòi hỏi phải tìm được giá trị tối ưu của cặp ngưỡng [L_{min}, L_{max}] và N vì các giá trị này ảnh hưởng trực tiếp đến quá trình ước tính của Jiang như trong công thức 7.

Vo và CS trong [10] đề xuất giải pháp tập hợp chùm giảm độ trễ OBADR (Optimal Burst Assembly for Delay Reduction) cũng dựa trên ý tưởng gửi sớm gói tin điều khiển (Hình 2b). Điểm khác biệt của OBADR là một mô hình tập hợp chùm hai giai đoạn: tập hợp chùm dựa trên ngưỡng thời gian $T_a - T_o$ trong giai đoạn 1 và tập hợp chùm dựa

trên ngưỡng độ dài ước tính L_e trong giai đoạn 2. Một lưu ý của tập hợp chùm trong giai đoạn 1 là Vo và CS đã sử dụng phương pháp TW-EWMA [12] để ước tính kích thước chùm hoàn thành (Công thức 8):

$$\begin{aligned}\lambda_e &= (1 - \alpha) \times \lambda_{avg} + \alpha \times \lambda_{cur} \\ L_e &= L_w + \lambda_e \times T_o\end{aligned}\quad (8)$$

trong đó L_w là độ dài chùm thực tế đo được trong cửa sổ ước tính ($T_a - T_o$); λ_{avg} và λ_{cur} là tốc độ trung bình của các gói tin đến trước đó và hiện thời; α là một trọng số thể hiện mối tương quan (tỉ lệ) giữa tốc độ trung bình các gói tin đến trước đó và tốc độ trung bình các gói tin đến trong cửa sổ ước tính hiện thời. Tốc độ trung bình của các gói tin đến hiện thời (trong cửa sổ ước tính $T_a - T_o$) được tính như sau:

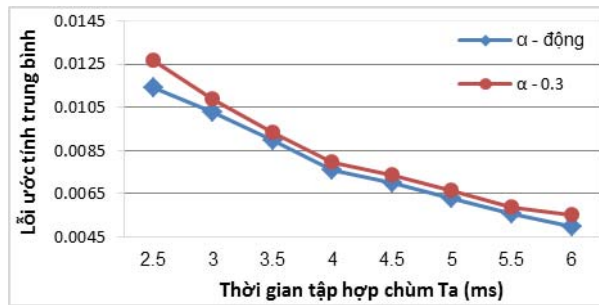
$$\lambda_{cur} = \frac{N}{T_a - T_o}, \quad (9)$$

với N là số gói tin đến trong cửa sổ ước tính.

Khác với các tác giả trong [12] là gán trọng số α bằng một hằng số cố định 0.3, Vo và CS [10] đã đề xuất một cách tính trọng số α một cách linh hoạt hơn và chuyển biến thích nghi theo tình trạng lưu lượng đến. Cụ thể, giá trị α được tính bởi công thức 10:

$$\frac{\alpha}{1 - \alpha} = \frac{\lambda_{cur}}{\lambda_{avg}} \Rightarrow \alpha = \frac{\lambda_{cur}}{\lambda_{avg} + \lambda_{cur}} \quad (10)$$

Thông qua mô phỏng (xem thêm Mục IV), kết quả mô phỏng (Hình 3) đã chứng minh được rằng đề xuất của Vo và cs. trong [10] với việc thay đổi α theo công thức 10 sẽ cho lỗi ước tính thấp hơn so với trường hợp α cố định bằng 0.3. Tuy nhiên Vo và cs. chưa xem xét đến những trường hợp nào áp dụng α động sẽ có hiệu quả và miền giá trị tốt nhất nào cho α . Các phần tiếp theo sẽ phân tích hai vấn đề này.



Hình 3. So sánh hiệu quả giữa $\alpha = 0.3$ và α động

III. PHÂN TÍCH TÁC ĐỘNG CỦA TÍNH CHẤT LUỒNG DỮ LIỆU ĐẾN ĐỐI VỚI VIỆC ƯỚC TÍNH ĐỘ DÀI CHÙM HOÀN THÀNH

Giải thuật tập hợp chùm giảm độ trễ OBADR được đề xuất trong [10] tiếp tục được sử dụng để phân tích tác động của tính chất luồng dữ liệu đến đối với việc ước tính độ dài chùm hoàn thành. Mô tả chi tiết của giải thuật OBADR được trình bày như sau:

Giải thuật OBADR

Input:

T_a ; // ngưỡng thời gian tập hợp
 T_o ; // giá trị thời gian bù đắp
 S_q ; // danh sách các gói tin đến trong hàng đợi
 max_p ; // kích thước lớn nhất của gói tin đến trong hàng đợi S_q

Output: \bar{R}_E ;

Begin

```

1   $\lambda_{avg} := 0$ ; // tốc độ đến trung bình các gói tin
2   $R_E := 0$ ; // khởi tạo lỗi ước tính
3   $t_1 := T_a - T_o$ ; // thời điểm gửi gói điều khiển
4   $M := 0$ ; // số chùm sinh ra
5   $b := \emptyset$ ; // khởi tạo bộ đệm chùm
6   $KT := false$ ; // kiểm tra thời điểm gửi gói điều khiển
7  While ( $S_q \neq \emptyset$ ) do
8      $p :=$  gói tin đến hàng đợi;  $S_q := S_q \setminus \{p\}$ ;
9      $T_q := s_p$ ; //  $s_p$  là thời điểm đến gói tin p

```

```

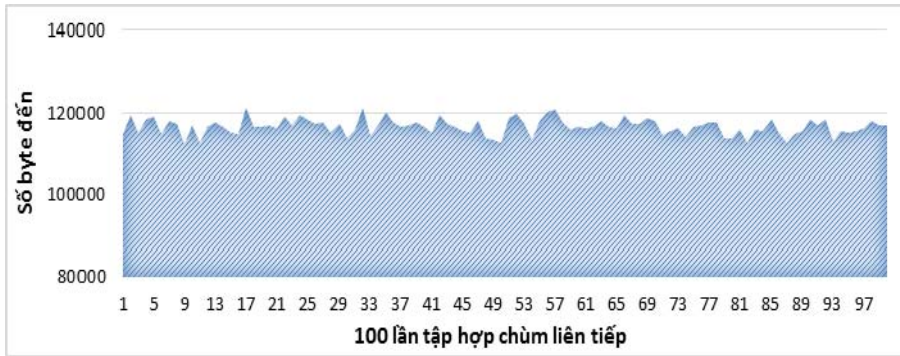
10   $b := b \cup \{p\};$  // tập hợp gói p vào bộ đệm chòm b
11  If ( $T_q \geq t_1$ ) and ( $KT = \text{false}$ ) then // giai đoạn 1: gửi gói điều khiển
12     $L := |b|;$  // độ dài chòm hiện thời
13     $\lambda_{cur} := L / (T_a - T_0);$  // tốc độ đến các gói tin trong cửa sổ ước tính
14     $\alpha := \lambda_{cur} / (\lambda_{cur} + \lambda_{avg});$  // điều chỉnh trọng số  $\alpha$ 
15     $\lambda_{avg} := (1 - \alpha) \times \lambda_{avg} + \alpha \times \lambda_{cur};$ 
16     $L_e := L + T_0 \times \lambda_{avg};$  // độ dài chòm ước tính được
17     $KT := \text{true};$  // kiểm tra gói điều khiển đã gửi
18  End if
19  If ( $L_e - \max_p \leq |b| \leq L_e$ ) then // giai đoạn 2: gửi chòm
20     $L := |b|;$  // độ dài chòm hoàn thành
21     $R_E := R_E + |L - L_e| / L;$  // cập nhật tổng lỗi ước tính
22     $b := \emptyset;$ 
23     $t_1 := t_1 + T_q;$ 
24     $KT := \text{false};$ 
25     $M := M + 1;$  // cập nhật số chòm sinh ra
26  End if
27 End while
28  $\overline{R_E} := R_E / M;$ 
29 Return  $\overline{R_E};$ 
End

```

Độ phức tạp của giải thuật chủ yếu thực hiện ở vòng lặp **While** (từ dòng 7 đến dòng 27), do độ phức tạp của các lệnh trong vòng lặp **While** là $O(1)$, nên độ phức tạp của giải thuật là $O(N)$, ở đây N là số gói tin đến trong hàng đợi S_q .

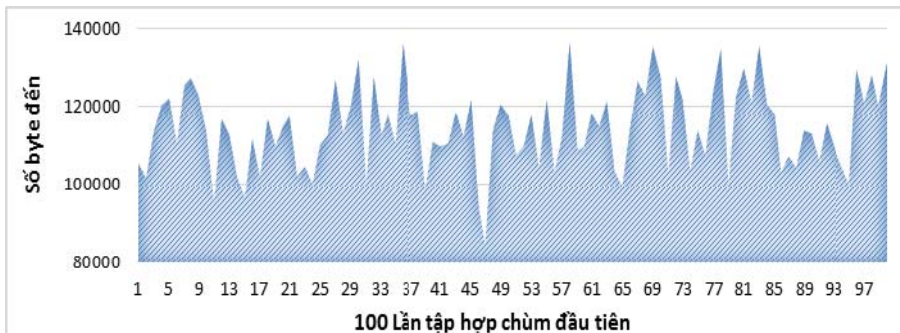
Theo công thức 8 (tương ứng dòng 15 trong giải thuật OBADR), độ dài ước tính phụ thuộc vào hai đại lượng: tốc độ trung bình của các gói tin đến trước đó λ_{avg} và tốc độ trung bình của các gói tin đến hiện thời λ_{cur} . Xét trong điều kiện luồng (gói tin) dữ liệu đến ít có biến động (Hình 4), tốc độ trung bình của các gói tin đến hiện thời λ_{cur} gần như không có biến đổi nhiều so với tốc độ trung bình của các gói tin đến trước đó λ_{avg} , tức là $\lambda_{cur} \approx \lambda_{avg}$, khi đó tốc độ ước tính (λ_e) trong công thức 8 được viết lại là:

$$\lambda_e = (1 - \alpha) \times \lambda_{avg} + \alpha \times \lambda_{avg} \Rightarrow \lambda_e = \lambda_{avg} \quad (11)$$



Hình 4. Trường hợp tốc độ luồng dữ liệu đến không có nhiều biến đổi

Rõ ràng α đã biến mất trong công thức tính độ dài ước tính, hay nói cách khác α không có tác động đến việc ước tính độ dài chòm. Tuy nhiên, trong trường hợp luồng (gói tin) dữ liệu đến nhiều biến động (Hình 5) với các trường hợp tăng/giảm đột biến của tốc độ đến, tức là $\lambda_{cur} \neq \lambda_{avg}$, nên α sẽ có tác động đáng kể đến việc ước tính độ dài chòm. Một đánh giá dựa trên mô phỏng trong Mục V sẽ chỉ rõ trường hợp này.



Hình 5. Trường hợp tốc độ luồng dữ liệu đến có nhiều biến đổi với các trường hợp tăng/giảm đột biến.

IV. MÔ PHỎNG VÀ PHÂN TÍCH KẾT QUẢ

Giải thuật OBADR sẽ được cài đặt lại với môi trường mô phỏng tương tự trong [10]. 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 với kích thước của chúng thay đổi ngẫu nhiên từ 500 đến 1000 bytes. Lưu lượng tải chuẩn hóa (tải chuẩn hóa được định nghĩa là bằng tốc độ đến trên khả năng đáp ứng của băng thông) được xem xét thay đổi từ 0.1 đến 0.9. Tốc độ đến của luồng chùm có nhiều biến động đột ngột như được chỉ ra trong Hình 5. Mô phỏng được thực hiện trong vòng 1s, dữ liệu được trích xuất từ NS2. Các tham số tập hợp chùm bao gồm: $T_a = 6\text{ms}$, $T_o = 2\text{ms}$.

Các mục tiêu mô phỏng bao gồm:

- Khảo sát giá trị α tương ứng với tải đến thay đổi nhằm đạt được lỗi ước tính thấp nhất;
- So sánh hiệu quả giữa α cố định và α động dựa trên lỗi ước tính trung bình.

Lỗi ước tính trung bình được tính bởi Công thức 12:

$$\overline{R_E} = \frac{\sum_{i=1}^M (|L(i) - L_e(i)| / L(i))}{M}, \quad (12)$$

trong đó $L(i)$ là kích thước thật của chùm và $L_e(i)$ là kích thước chùm dự đoán ở lần tập hợp thứ i .

A. Khảo sát giá trị α tương ứng với tải đến thay đổi nhằm đạt được lỗi ước tính thấp nhất

Với tải chuẩn hóa thay đổi từ 0.1 đến 0.9 và khảo sát các giá trị α từ 0.1 đến 0.9, kết quả thu được trong Bảng 1 cho thấy lỗi ước tính tối thiểu (các ô tô vàng trong Bảng 1) có phân bố tương ứng với α từ 0.4 đến 0.6. Như vậy, việc thiết lập α cố định rõ ràng không hiệu quả đối với tất cả mọi tải. Dựa trên Bảng 1, có một miền giá trị α (trong trường hợp này là từ 0.4 đến 0.6) dao động xung quanh giá trị 0.5 cho lỗi ước tính thấp nhất.

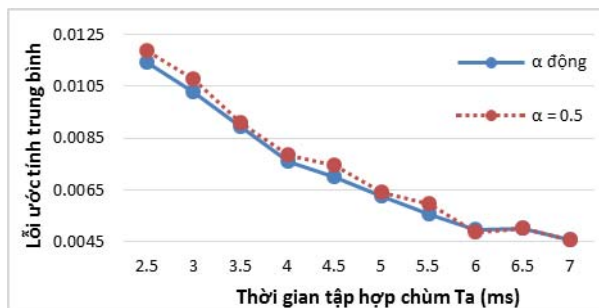
Bảng 1. Lỗi ước tính R_E với tải chuẩn hóa thay đổi và các giá trị α từ 0.1 đến 0.9.

Tải \ α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.02740	0.02656	0.02546	0.02651	0.02457	0.02612	0.02705	0.02667	0.02651
0.2	0.01361	0.01359	0.01341	0.01304	0.01348	0.01370	0.01445	0.01350	0.01343
0.3	0.00905	0.00861	0.00912	0.00842	0.00867	0.00897	0.00883	0.00896	0.00909
0.4	0.00686	0.00675	0.00695	0.00672	0.00705	0.00651	0.00665	0.00666	0.00673
0.5	0.00551	0.00519	0.00549	0.00526	0.00489	0.00545	0.00521	0.00518	0.00518
0.6	0.00447	0.00456	0.00445	0.00426	0.00443	0.00452	0.00437	0.00462	0.00439
0.7	0.00414	0.00377	0.00377	0.00373	0.00370	0.00382	0.00384	0.00399	0.00385
0.8	0.00327	0.00354	0.00332	0.00333	0.00342	0.00326	0.00332	0.00343	0.00337
0.9	0.00300	0.00300	0.00308	0.00299	0.00289	0.00298	0.00316	0.00299	0.00311

Một kết quả khác cũng được rút ra từ Bảng 1 là khi tăng tải đến thì lỗi ước tính tương ứng giảm với mọi giá trị α được khảo sát. Nguyên nhân là do giai đoạn 2 (dòng 19 đến 26) của giải thuật OBADR sử dụng phương pháp tập hợp chùm dựa trên độ dài chùm ước tính L_e , nên khi tải tăng, các gói tin đến nhiều giúp ngưỡng độ dài (L_e) luôn đạt đến trước, đẩy kích thước thật của chùm hoàn thành L về gần với L_e . Điều này đã làm giảm đáng kể lỗi ước tính.

B. So sánh hiệu quả giữa α cố định và α động dựa trên lỗi ước tính trung bình

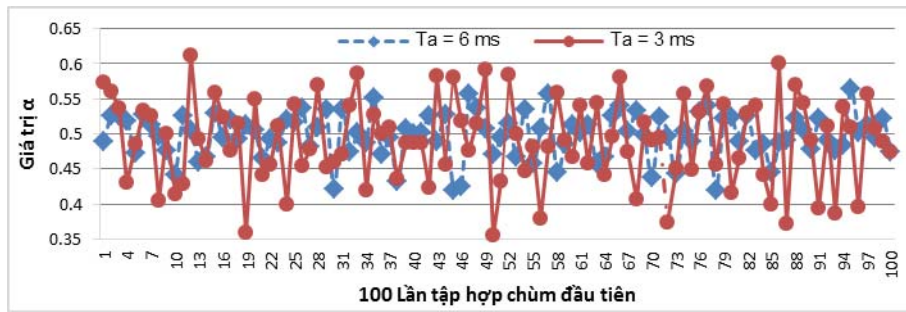
Như chỉ ra trong Bảng 1, giá trị α tốt nhất có phân bố xung quanh là 0.5. Với tải chuẩn hóa 0.5, một so sánh giữa α cố định và α động dựa trên lỗi ước tính trung bình được chỉ ra trong Hình 6.



Hình 6. So sánh hiệu quả giữa α động và $\alpha = 0.5$ dựa trên lỗi ước tính trung bình khi thay đổi thời gian tập hợp chùm (T_a) từ 2.5ms đến 7ms

Hình 6 cho thấy rằng khi thời gian tập hợp chùm bé (từ $T_a = 2.5\text{ms}$ đến $T_a = 5.5\text{ms}$), α động cho kết quả lỗi ước tính trung bình tốt hơn so với α cố định bằng 0.5. Điều này là do khi thiết lập thời gian tập hợp bé thì cửa sổ ước tính $T_w = T_a - T_o$ sẽ nhỏ, α động lúc này sẽ góp phần phản ánh sự thay đổi đột ngột luồng chùm đến tốt hơn, mà minh chứng

là giá trị α sẽ có sự thay đổi (dao động) lớn hơn như được chỉ ra ở Hình 7. Nếu thời gian tập hợp lớn, do việc tính trung bình trong khoảng thời gian lớn sẽ không phản ánh được các thay đổi đột ngột của luồng chùm đến mà kết quả là làm cho giá trị α ít thích nghi hơn, nên lỗi ước tính không có sự thay đổi đáng kể so với giá trị cố định $\alpha = 0.5$ (từ $T_a = 6\text{ms}$ đến $T_a = 7\text{ms}$).



Hình 7. Sự biến thiên giá trị α động trong 100 lần tập hợp chùm đầu tiên với $T_a = 6\text{ms}$ và $T_a = 3\text{ms}$

Hình 6 cũng phản ánh một điều là lỗi ước tính có xu hướng giảm khi tăng thời gian tập hợp chùm. Thực tế độ chính xác của việc ước tính phụ thuộc vào kích thước của cửa sổ ước tính, trong đó cửa sổ càng lớn thì việc ước tính càng chính xác hơn. Tuy nhiên, việc thiết lập thời gian tập hợp lớn cũng sẽ làm gia tăng kích thước các chùm sinh ra, gây thêm khó khăn trong quá trình lập lịch chùm ở mạng lõi do có ít khoảng trống phù hợp với kích thước chùm.

V. KẾT LUẬN

Bài viết này đã đánh giá các giải pháp tập hợp chùm giảm độ trễ bằng cách gửi sớm gói điều khiển và phân tích tác động của tính chất luồng (gói tin) dữ liệu đến đối với việc ước tính độ dài chùm hoàn thành trong tập hợp chùm giảm độ trễ tại nút biên mạng OBS. Một đại lượng được tập trung xem xét trong bài viết này là trọng số α trong đó việc điều chỉnh giá trị α linh hoạt theo tốc độ của luồng dữ liệu đến đã làm tăng hiệu quả của việc ước tính độ dài chùm hoàn thành trong trường hợp luồng dữ liệu đến có sự thay đổi đáng kể (Như được chỉ ra ở Hình 5), trong trường hợp tốc độ đến không có sự thay đổi đáng kể (Như được chỉ ra trong Hình 4) việc sử dụng giá trị α (cả α động và tĩnh) sẽ không có ý nghĩa, vì việc ước tính tốc độ đến bây giờ không phụ thuộc vào giá trị α (Như được chỉ ra ở Công thức 11). Như được phân tích và được kiểm chứng thông qua mô phỏng, rõ ràng tính chất của luồng đến có tác động đáng kể đến độ chính xác của việc ước tính độ dài chùm hoàn thành, được thể hiện thông qua tỉ lệ lỗi ước tính như trong Bảng 1 và Hình 6. Kết quả này cũng khẳng định hiệu quả của việc điều chỉnh linh hoạt α theo tốc độ của luồng dữ liệu khi tốc độ này có sự thay đổi đáng kể.

VI. TÀI LIỆU THAM KHẢO

- [1] Q. Chunming, Y. Myungsik, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet", Journal of High Speed Networks, vol. 8, pp. 69-84, 1999.
- [2] C. Yang, Q. Chunming, Y. Xiang, "Optical burst switching: A new area in optical networking research", IEEE Network, vol. 18, pp. 16-23, 2004.
- [3] X. Yu, Y. Chen, C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks", Proceeding of Opticomm, pp. 149-159, 2002.
- [4] T. Hashiguchi et al, "Burst assembly mechanism with delay reduction for OBS networks", Proc. of Conf. on the Optical Internet., pp. 664-666. 2003.
- [5] Z. Sui, Q. Zeng, S. Xiao, "An offset differential assembly method at the edge of OBS network", Proc. of SPIE Optical Transmission, Switching and Subsystems III., vol. 6021, pp. 1-6, 2005.
- [6] T. Mikoshi, T. Takenaka, "Improvement of burst transmission delay using offset time for burst assembly in optical burst switching", Proc. of IEICE., pp. 13-18, 2008.
- [7] Y. Fukushima et al., "A burst assembly method to reduce end-to-end delay in optical burst switching networks", WSEAS Transactions on Communications, vol.8, no 8, pp. 894-903, 2009.
- [8] H. Liu, S. Jiang, "A mixed-length and time threshold burst assembly algorithm based on traffic prediction in OBS network", Int. Journal of Sensing, Computing & Control., vol.2, no.2, pp. 87-93, 2012.
- [9] X. Jiang, N. Zhu, L. Yuan, "A novel burst assembly algorithm for OBS networks based on burst size and assembly time prediction", Journal of Computational Information Systems., vol.9, no.2, pp. 463-475, 2013.

- [10] Viet Minh Nhat Vo, Van Hoa Le, Hoang Son Nguyen, "A model of optimal burst assembly for delay reduction at ingress OBS nodes", Turkish Journal of Electrical Engineering & Computer Sciences, vol.25, no.5, pp. 3970-3982, 2017.
- [11] Peterson, L. Larry, "Computer networks: a system approach, Morgan Kaufmann, pp. 552, 1996.
- [12] K. Salad, F. Haidari, "On the Performance of a Simple Packet Rate Estimator", Proc. of IEEE/ACS Inter. Conf. on Computer Systems and Applications, pp. 392-395, 2008.

EFFECTS OF DATA FLOW PROPERTIES TO THE EFFICIENCY OF BURST ASSEMBLY FOR DELAY REDUCTION AT INGRESS OBS NODES

Le Van Hoa, Vo Viet Minh Nhat, Nguyen Hoang Son

ABSTRACT: *The end-to-end delay of a burst transmitted over optical burst switching networks is mainly due to four components: (1) the assembly delay at ingress node, (2) the offset time for resource reservation, (3) the forwarding delay at core nodes and (4) the propagation delay in core network. Two later delays usually depend on the selected path and the available bandwidth of the path, so it can not be reduced with an already implemented protocol. Only two first delays, the assembly delay and the offset time, can be reduced. The combination of these two delays is commonly referred to as the buffering delay. There are several proposals to reduce this buffering delay by sending the packet early before the burst completion and estimating the burst size which is carried in the control packet. This paper will analyze the impact of incoming (burst) flows on the completed burst length estimation in the burst assembly for delay reduction at ingress OBS nodes.*