

PHÂN TÍCH HIỆU QUẢ CÁC GIẢI THUẬT LẬP LỊCH TRÊN MẠNG CHUYỂN MẠCH CHÙM QUANG

Võ Viết Minh Nhật¹, Nguyễn Hồng Quốc²

¹Khoa Du lịch, Đại học Huế

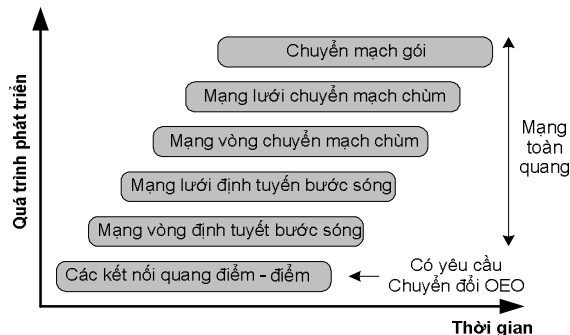
²Trường Đại học Sư phạm, Đại học Huế

Tóm tắt. Việc lập lịch là một trong những hoạt động có ảnh hưởng lớn đến hiệu năng của mạng chuyển mạch chùm quang. Thực tế, đã có khá nhiều giải thuật lập lịch khác nhau được đề nghị, nhưng chủ yếu dựa trên ý tưởng là có hay không lấp đầy khoảng trống. Bài viết này trình bày một cái nhìn khái quát về các giải thuật lập lịch và phân tích đánh giá hiệu quả dựa trên các kết quả mô phỏng trên phần mềm mô phỏng NS2-OBS0.9a.

Từ khóa: Mạng chuyển mạch chùm quang, giải thuật lập lịch, phần mềm mô phỏng NS2-OBS.

1. Giới thiệu

Mạng chuyển mạch chùm quang OBS (*Optical Burst Switched*) được biết đến như là một giải pháp trung gian của quá trình phát triển từ các mạng định tuyến bước sóng WR (*Wavelength-Routed*) đến các mạng chuyển mạch gói quang OPS (*Optical Packet Switched*) [8]. Thực tế, mạng OBS đã khắc phục được hạn chế về khả năng sử dụng và khai thác không hiệu quả băng thông của các mạng WR và bước đầu đưa mô hình chuyển mạch gói quang thành hiện thực khi mà công nghệ chế tạo vùng đệm quang chưa thực sự phát triển. Do đó, mạng OBS còn được gọi là mạng chuyển mạch gói quang không sử dụng vùng đệm.

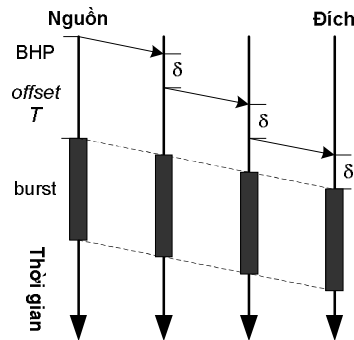


Hình 1. Sự phát triển mạng quang theo thời gian [8]

Để có thể truyền tải được dữ liệu, mạng OBS đã có những thiết kế đặc biệt. Tại một nút biên mạng OBS (nút biên OBS), các luồng dữ liệu đến (ví dụ các gói tin IP) được tập hợp trong các gói dữ liệu quang thô (còn được gọi là các *burst* dữ liệu) có kích thước lớn. Việc hình thành các *burst* này có thể dựa trên ngưỡng độ dài, tức là một *burst* sẽ được hình thành khi kích thước của *burst* (khối lượng luồng dữ liệu IP đến tập hợp

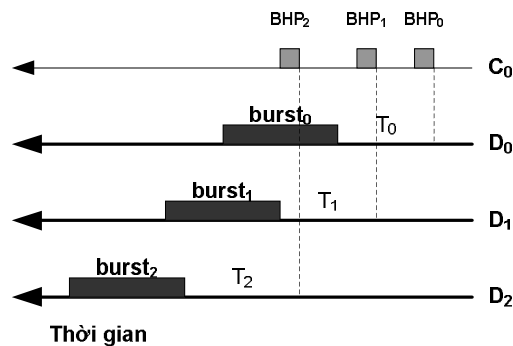
trong *burst*) đạt đến giá trị ngưỡng, hoặc dựa trên ngưỡng thời gian, tức là một *burst* sẽ được hình thành sau từng khoảng thời gian định kỳ.

Một đặc trưng tiêu biểu của mạng OBS là phần điều khiển của gói dữ liệu quang thô tách rời với phần *burst* dữ liệu (hình 2). Nói một cách khác, để truyền tải thành công một *burst* dữ liệu, một gói tin điều khiển BHP (Burst Header Packet) được hình thành và được gửi đi trước một khoảng thời gian offset đủ để cấu hình (đặt trước tài nguyên) tại các nút trung gian dọc theo hành trình mà *burst* dữ liệu sau đó sẽ di chuyển đi từ nút nguồn đến nút đích.



Hình 2. Khoảng thời gian offset được tính ít nhất bằng tổng thời gian xử lý tại các nút trung gian và thời gian lang truyền từ nguồn đến đích

Thêm vào đó, mạng OBS còn dành riêng một số kênh (bước sóng) cho việc truyền tải loại gói tin điều khiển BHP. Do vậy, chúng ta có thể nói, mỗi gói điều khiển BHP là tách rời với *burst* dữ liệu của nó về mặt không gian (trên kênh truyền khác) và cũng như về mặt thời gian (gửi đi trước một khoảng thời gian offset).



Hình 3. Gói điều khiển BHP tách rời với *burst* dữ liệu về mặt không gian và thời gian

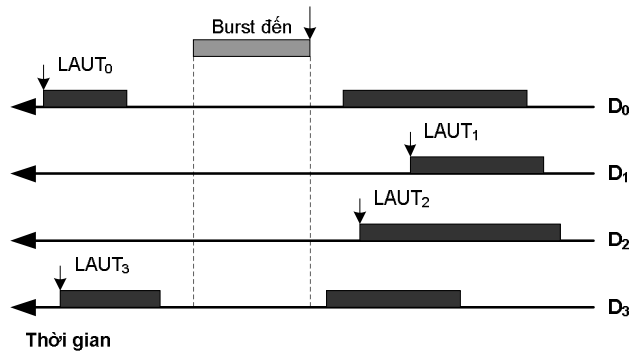
Với cách truyền tải dữ liệu như mô tả, rõ ràng mạng OBS không cần đến các vùng đệm quang để lưu tạm thời các *burst* dữ liệu trong khi chờ đợi việc xử lý chuyển mạch tại các nút trung gian. Tuy nhiên, điều này cũng đặt ra áp lực đối với việc làm thế nào để một gói điều khiển BHP cấu hình (đặt trước tài nguyên) thành công tại các nút lõi bên trong mạng OBS cho việc truyền tải *burst* dữ liệu của nó. Đó chính là vai trò của các giải thuật lập lịch mà gói điều khiển BHP phải thực hiện. Trong bài viết này chúng

tôi sẽ đề cập đến và phân tích hiệu quả các giải thuật lập lịch thông qua các kết quả mô phỏng trên gói obs-0.9a [10] của phần mềm mô phỏng NS (*Network Simulator*) [11].

Cấu trúc tiếp theo của bài viết như sau: Mục 2 tóm tắt các giải thuật lập lịch, bao gồm các giải thuật không xét đến lấp đầy khoảng trống như FFUC và LAUC, và các giải thuật có xét đến lấp đầy khoảng trống như FFUC-VF và LAUC-VF (hay Min-SV). Ngoài ra, các mở rộng của chúng, Min-EV, Max-EV, BFUC-VF hay kết hợp (Min-SV & Min-EV), là tương tự với LAUC-VF, chỉ thay đổi đối tượng xem xét tối ưu. Kết quả mô phỏng trên NS2-obs0.9a, được trình bày ở mục 3, sẽ chỉ ra hiệu quả của từng giải thuật và qua đó cần thiết phải có một mô hình chọn lựa các giải thuật tại mỗi nút khi thực hiện lập lịch.

2. Tóm tắt các giải thuật lập lịch

Các giải thuật lập lịch được phân loại dựa trên ý tưởng chủ đạo là có hay không lấp đầy khoảng trống (*void filling*). Một khoảng trống là đoạn băng thông khả dụng trên một kênh, giữa hai *burst* đã được lập lịch liên tiếp như mô tả ở hình 4, nếu chúng ta chỉ xem xét việc lập lịch *burst* đến đối với các kênh D_1 và D_2 , giải thuật lập lịch được xem xét là không xét đến việc lấp đầy khoảng trống. Ngược lại, nếu có xét đến cả kênh D_0 và D_3 thì giải thuật lập lịch được xem xét là có xét đến việc lấp đầy khoảng trống. Thực tế, các khoảng trống này được sinh ra khi có những biến thiên quan trọng về mật độ luồng dữ liệu IP đến tại một nút biên vào OBS, cũng như là mật độ các *burst* đến tại các nút lõi.

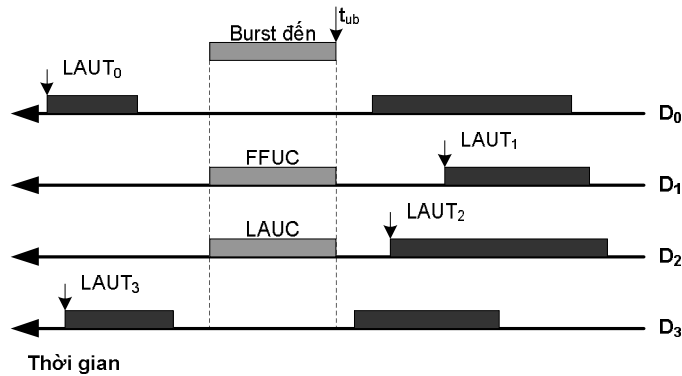


Hình 4. Việc lập lịch có thể xét đến có hay không lấp đầy khoảng trống

2.1. Các giải thuật lập lịch không xét đến lấp đầy khoảng trống

Có hai giải thuật lập lịch không xét đến lấp đầy khoảng trống: FFUC (*First Fit Unscheduled Channel*) [1,2,6,8] và LAUC (*Latest Available Unused Channel*) [2,4].

Đối với loại giải thuật này, chúng ta cần lưu ý đến 2 tham số: thời điểm đến t_{ub} của *burst* so với thời điểm kết thúc của *burst* sau cùng nhất LAUT_{*i*} (*Latest Available Unscheduled Time*) trên kênh dữ liệu khả dụng thứ *i*. Nếu LAUT_{*i*} $\leq t_{ub}$, kênh thứ *i* mới được xem xét cho việc lập lịch *burst* đến. Như mô tả ở hình 5, rõ ràng chỉ có kênh D_1 và D_2 là được xem xét vì thỏa mãn điều kiện LAUT₁ $\leq t_{ub}$ và LAUT₂ $\leq t_{ub}$.



Hình 5. *Lập lịch không xét đến lấp đầy khoảng trống*

2.1.1. Giải thuật FFUC

Giải thuật FFUC [1,3,5,6] chọn kênh đầu tiên được tìm thấy có thể lập lịch được. Như mô tả ở hình 5, kênh D_1 được chọn vì đó là kênh đầu tiên được tìm thấy thỏa mãn điều kiện $LAUT_1 \leq t_{ub}$. Giải thuật FFUC có thể được mô tả như sau:

Tham số vào:

- t_{ub} : thời điểm đến của *burst* đến chưa được lập lịch;
- W : số kênh dữ liệu ra tối đa;
- i : kênh dữ liệu thứ i , $i = 0 \dots W-1$;
- $LAUT_i$, $i = 0 \dots W-1$: thời điểm kết thúc của *burst* sau cùng nhất trên kênh thứ i ;

Giải thuật:

Bước 1: Khởi tạo $i = 0$;

Bước 2: Nếu $i \geq W$, thông báo không thể lập lịch được và kết thúc;

Bước 3: Nếu $LAUT_i \leq t_{ub}$: Lập lịch cho *burst* trên kênh thứ i và kết thúc; Nếu không, quay lại **bước 2** với kênh $i = i+1$;

2.1.2. Giải thuật LAUC

Giải thuật LAUC [2,4] chọn kênh mà khoảng cách giữa *burst* đến và *burst* đã được lập lịch sau cùng nhất là tối thiểu. Như mô tả ở hình 5, kênh D_2 được chọn vì thỏa mãn điều kiện $LAUT_2 \leq t_{ub}$ và $gap = t_{ub} - LAUT_2$ là nhỏ nhất. Giải thuật LAUC được mô tả như sau:

Tham số vào:

- t_{ub} : thời điểm đến của *burst* đến chưa được lập lịch;
- W : số kênh dữ liệu ra tối đa;
- $LAUT_i$, $i = 0 \dots W-1$: thời điểm kết thúc của *burst* sau cùng nhất trên kênh thứ i ;

sc: chỉ số kênh được chọn (sc = -1 khi chưa có kênh nào được chọn);

i: kênh dữ liệu thứ $i = 0 \dots W-1$;

gap_{min}: khoảng cách tối thiểu giữa *burst* đến và *burst* đã được lập lịch sau cùng nhất;

Giải thuật:

Bước 1: Khởi tạo $i = 0$; $sc = -1$; gap_{min} = <giá trị lớn>;

Bước 2: Nếu $i \geq W$: Chuyển sang **bước 4**;

Bước 3: Nếu $LAUT_i \leq t_{ub}$ thì:

Tính $gap_i = t_{ub} - LAUT_i$

- Nếu $gap_i < gap_{min}$ thì:

gap_{min} = gap_i; sc = i;

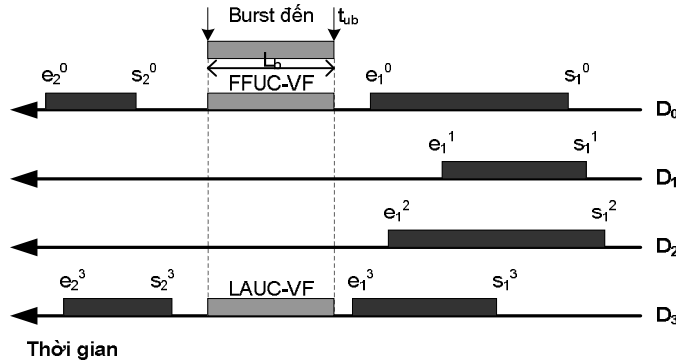
$i = i+1$; quay lại **bước 2**

Bước 4: Nếu $sc \neq -1$ thì: - Lập lịch cho *burst* đến trên kênh sc;

Bước 5: Kết thúc;

2.2. Các giải thuật lập lịch có xét đến lấp đầy khoảng trống

Trên cơ sở 2 giải thuật không xét đến lấp đầy khoảng trống, 2 giải thuật tương tự có xét đến lấp đầy khoảng trống (*Void-Filling*) đã được đề nghị [1,5,6,9]: FFUC-VF và LAUC-VF.



Hình 6. Lập lịch có xét đến lấp đầy khoảng trống

Đối với loại giải thuật này, bộ lập lịch duy trì thời điểm bắt đầu và kết thúc (s_j^i , e_j^i) của các *burst* đã lập lịch trên tất cả các kênh (trong đó $i = 0 \dots W-1$, W là tổng số kênh, và $j = 1 \dots N_b$, N_b là tổng số *burst* đang được lập lịch trên một kênh). Như mô tả ở hình 6, kênh D_0 và D_3 là được xem xét vì thỏa mãn điều kiện $e_1^i \leq t_{ub}$ và $s_1^i \geq t_{ub} + L_b$, $i = 0 \dots 3$.

2.2.1. Giải thuật FFUC-VF

Giải thuật FFUC-VF [1,5] chọn khoảng trống khả dụng được tìm thấy đầu tiên. Như mô tả ở hình 6, kênh D_0 được chọn vì thỏa mãn điều kiện $e_1^0 \leq t_{ub}$ và $s_2^0 \geq t_{ub} + L_b$. Giải thuật FFUC-VF được mô tả như sau:

Tham số vào:

L_b : độ dài *burst* đến chưa được lập lịch;

t_{ub} : thời điểm đến của *burst* chưa được lập lịch;

W : số kênh dữ liệu ra tối đa cho việc lập lịch;

i : kênh dữ liệu thứ $i = 0 \dots W-1$;

s_j^i thời điểm bắt đầu và e_j^i kết thúc của *burst* j trên kênh dữ liệu thứ i , $i = 0 \dots W-1$;

Giải thuật:

Bước 1: Khởi tạo $i = 0$;

Bước 2: Nếu $i \geq W$: Chuyển sang **bước 5**;

Bước 3: Tìm khoảng trống khả dụng trên kênh i sao cho $t_{ub} \geq e_j^i$ và $s_{j+1}^i \geq t_{ub} + L_b$;
- Nếu tìm thấy: Lập lịch cho *burst* đó trên kênh i và chuyển sang **bước 5**

Bước 4: Gán $i = i+1$ và quay lại **bước 2**;

Bước 5: Kết thúc;

2.2.2. Giải thuật LAUC-VF (Min-SV)

Giải thuật LAUC [5,6] chọn kênh có khoảng cách từ thời điểm đến của *burst* đến và thời điểm kết thúc của *burst* liền kề đã được lập lịch trước đó là nhỏ nhất. Như mô tả ở hình 6, dựa vào thời gian đến và thời gian kết thúc của *burst* thì tất cả các kênh D_0, D_1, D_2, D_3 đều khả dụng để lập lịch cho *burst* đến. Ngoài ra ta thấy kênh D_3 có $t_{ub} - e_1^3$ là nhỏ nhất vì vậy kênh D_3 sẽ được chọn để lập lịch cho *burst* đến. Giải thuật LAUC-VF còn có một tên khác là giải thuật Min-SV (*Minimum Starting Void*) [9]. Giải thuật LAUC-VF (Min-SV) được mô tả như sau.

Tham số vào:

L_b : độ dài *burst* đến chưa được lập lịch;

t_{ub} : thời điểm đến của *burst* chưa được lập lịch;

W : số kênh dữ liệu ra tối đa cho việc lập lịch;

sc : chỉ số kênh được chọn;

i : kênh dữ liệu thứ $i = 0 \dots W-1$;

s_gap_{min} : khoảng cách tối thiểu giữa *burst* đến và *burst* đã được lập lịch trước đó;

Giải thuật:

Bước 1: Khởi tạo $i = 0$; $sc = -1$; $s_gap_{min} = \text{<giá trị lớn>}$;

Bước 2: Nếu $i \geq W$ chuyển sang **bước 5**;

Bước 3: Tìm khoảng trống trên kênh i sao cho: $t_{ub} \geq e_j^i$ và $s_{j+1}^i \geq t_{ub} + L_b$;

Nếu tìm thấy:

- Nếu: $t_{ub} - e_j^i < s_gap_{min}$ thì $s_gap_{min} = t_{ub} - e_j^i$ và $sc = i$;

Bước 4: Gán $i = i+1$ và quay lại **bước 2**;

Bước 5: Nếu $sc \neq -1$ thì lập lịch cho *burst* trên kênh sc ;

Bước 6: Kết thúc;

2.2.3. Giải thuật Min-EV

Khác với giải thuật LAUC-VF, giải thuật Min-EV (Minimum Ending Void) [9] chú ý đến việc tối thiểu là khoảng cách từ thời điểm kết thúc của *burst* đến và thời điểm bắt đầu của *burst* đã được lập lịch trước đó trên một kênh. Như vậy, mô tả của giải thuật Min-EV tương tự với giải thuật LAUC-VF, chỉ khác giải thuật Min-EV tìm các kênh có khoảng trống khả dụng để lập lịch cho *burst* trước, nếu không có thì sử dụng các giải thuật không lấp đầy khoảng trống để chọn kênh và đối với giải thuật Min-EV điều kiện chọn kênh để lập lịch cho *burst* đó sao cho $e_gap_{min} = \text{Min}(s_j^i - (t_{ub} + L_b))$, $\forall i, i = 0 \dots W-1$.

2.2.4. Giải thuật Max-EV

Trên cơ sở ý tưởng của các giải thuật LAUC-VF (Min-SV), Min-EV, J.P.Jue và V.M.Vokkarane [8] đã đề xuất một biến đổi khác là giải thuật Max-EV, trong đó mục tiêu là tối thiểu khoảng cách từ thời điểm kết thúc của *burst* đến và thời điểm bắt đầu của *burst* đã được lập lịch trước đó với kỳ vọng là khoảng trống được sinh ra sau khi lập lịch sẽ có cơ hội được sử dụng cho lập lịch các *burst* đến sau. Giải thuật Max-EV do đó cũng tương tự với giải thuật LAUC-VF, chỉ khác điều kiện chọn kênh để lập lịch cho *burst* đến sao cho $e_gap_{max} = \text{Max}(s_j^i - (t_{ub} + L_b))$, $\forall i, i = 0 \dots W-1$. (thay thế $s_gap_{min} = \text{Min}(t_{ub} - e_j^i)$, $\forall i, i = 0 \dots W-1$).

2.2.5. Giải thuật BFUC-VF

Giải thuật BFUC-VF (Best Fit Unscheduled Channel – Void Filling) đề xuất một hướng tiếp cận khác khi chọn kênh tối ưu nhất cho một *burst* đến. Theo Nandi M., và nhóm nghiên cứu (2009) [7] đã định nghĩa một tham số “**hiệu quả (tỉ lệ) sử dụng băng thông khoảng trống**” (**utilization**) khi một *burst* được lập lịch vào một khoảng trống trên một kênh:

$$\text{utilization} = L_b * 100 / (s_{j+1}^i - e_j^i), \forall i, i = 0 \dots W-1.$$

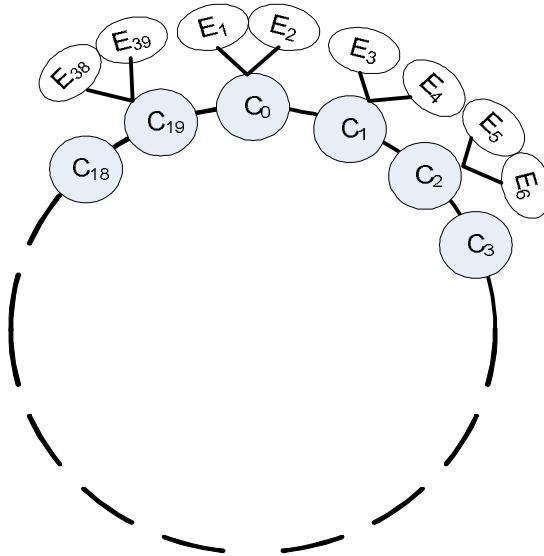
Kênh nào có giá trị hiệu quả sử dụng băng thông khoảng trống lớn nhất sẽ được

chọn.

Xét về bản chất, giải thuật BFUC-VF tương tự với giải thuật kết hợp Min-SV và Min-EV, bởi vì tối đa hóa đại lượng $L_b * 100 / (s_{j+1}^i - e_j^i)$, $\forall i, i = 0 \dots W-1$, cũng tương đương với tối thiểu hóa đại lượng $gap_{\min} = \text{Min}((t_{ub} - e_j^i) + (s_{j+1}^i - (t_{ub} + L_b)))$, $\forall i, i = 0 \dots W-1$. Mô tả của giải thuật **BFUC-VF** do đó là hoàn toàn tương tự giải thuật LAUC-VF, chỉ khác ở điều kiện chọn kênh sc sao cho **utilization** = $\text{Max}(L_b * 100 / (s_{j+1}^i - e_j^i))$, $\forall i, i = 0 \dots W-1$ (thay thế $s_{\text{gap}_{\min}} = \text{Min}(t_{ub} - e_j^i)$, $\forall i, i = 0 \dots W-1$).

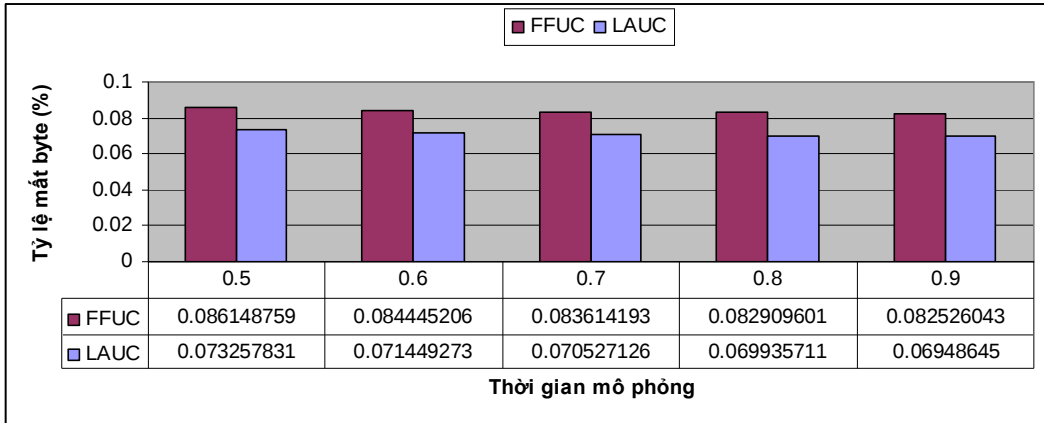
3. Mô phỏng và phân tích hiệu quả các giải thuật lập lịch

Mô phỏng các giải thuật lập lịch được thực hiện trên gói OBS-0.9a [10] của phần mềm mô phỏng NS [11]. Hình thái của mạng OBS thực hiện mô phỏng là một mạng hình vòng được tạo thành từ 20 nút lõi ($C_i, i=0..19$), mỗi nút lõi kết nối với hai nút biên ($E_i, i=0..39$) như mô tả ở hình 7. Các luồng dữ liệu được tạo ra liên tục (theo phân bố poisson) giữa các cặp nút E_i và E_j ($i, j=0..39$) với mật độ dày đặc. Các *burst* do đó được sinh ra tại các thời điểm thay đổi và cũng như có kích thước thay đổi. Số kênh dữ liệu 8 và kênh điều khiển 6 trên mỗi liên kết. Băng thông trên mỗi kênh là 10Gb/s.

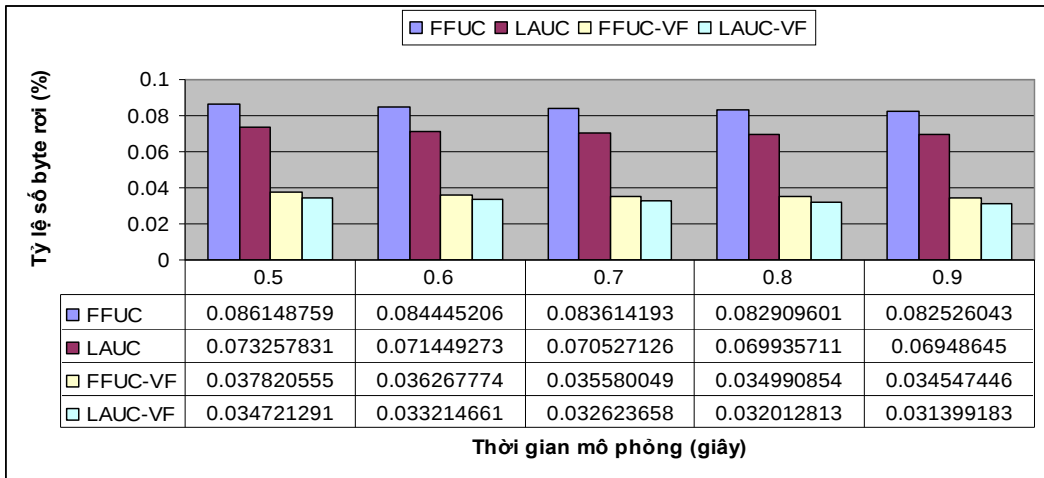


Hình 7. Hình thái mạng mô phỏng được tạo thành từ 20 nút lõi C_i , mỗi nút lõi kết nối với 2 nút biên

Kết quả mô phỏng hình 8 chỉ ra rằng các giải thuật lập lịch LAUC hiệu quả hơn FFUC, thể hiện ở tỷ lệ mất byte dữ liệu ít hơn, vì LAUC tối thiểu được khoảng cách giữa các *burst* được lập lịch trên một kênh, nên nó tối đa được số *burst* có thể lập lịch trên mỗi kênh và do đó tạo được nhiều cơ hội cho các *burst* đến sau tìm thấy được kênh khả dụng để lập lịch (kết quả là làm giảm số *burst* rơi).



Hình 8. So sánh tỷ lệ mất byte trên toàn mạng đối với 2 giải thuật FFUC và LAUC

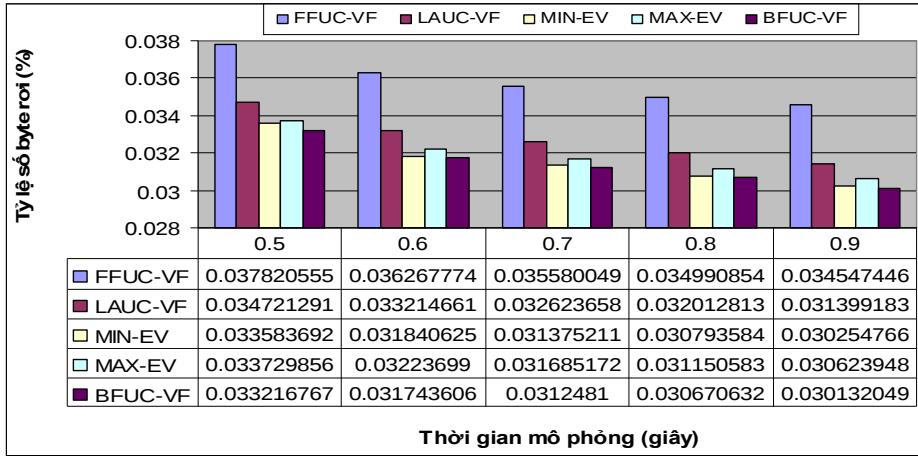


Hình 9. So sánh tỷ lệ mất byte trên toàn mạng đối với 2 nhóm giải thuật không lấp đầy khoảng trống và có lấp đầy khoảng trống

Tuy nhiên, 2 giải thuật FFUC và LAUC đã không tận dụng được khoảng trống không dùng đến giữa 2 *burst* đã được lập lịch trước. Hình 9 cho thấy rằng 2 giải thuật có lấp đầy khoảng trống (FFUC-VF và LAUC-VF) hiệu quả hơn rõ rệt so với nhóm giải thuật không lấp đầy khoảng trống (thể hiện ở tỷ lệ mất byte thấp hơn). Nguyên nhân là do nhóm giải thuật có lấp đầy khoảng trống đã tận dụng được băng thông nhàn rỗi trong các khoảng trống được sinh ra khi lập lịch, trong khi các giải thuật lập lịch không lấp đầy khoảng trống lại không xét đến. Điều này đã giúp FFUC-VF và LAUC-VF tối đa được số *burst* có thể lập lịch trên mỗi kênh và do đó tạo được nhiều cơ hội cho các *burst* đến sau tìm thấy được kênh khả dụng để lập lịch. Kết quả số byte rơi trên toàn mạng giảm.

Như đã trình bày trong mục 2.2, các giải thuật lập lịch có lấp đầy khoảng trống khác nhau chủ yếu ở điều kiện chọn kênh: tối thiểu khoảng trống (được sinh ra sau khi lập lịch) trước, sau hay cả trước và sau. Kết quả mô phỏng ở hình 10 cho thấy tỷ lệ mất byte của các giải thuật LAUC-VF, Min-EV, Max-EV và BFUC-VF chênh nhau không

đáng kể. Nếu xét tỷ lệ mất byte trên toàn mạng (trường hợp các nút mạng sử dụng cùng một giải thuật lập lịch), giải thuật BFUC-VF thể hiện hiệu quả cao nhất. Tuy nhiên, nếu xét trên từng nút riêng biệt, thứ tự hiệu quả của mỗi giải thuật có những thay đổi.



Hình 10. So sánh tổng số *burst* rơi trên toàn mạng đối với các giải thuật

Thực tế, hiệu quả của các giải thuật lập lịch phụ thuộc nhiều vào tốc độ đến và độ dài *burst* của đến, vào mật độ luồng mà ở đó các khoảng trống sinh ra có độ dài ngắn thay đổi, và quan trọng nhất là sự đồng nhịp giữa thời điểm đến của *burst* ngay sau thời điểm bắt đầu của khoảng trống. Như mô tả trong bảng 1, số byte mất tối thiểu tại các nút khác nhau đối với từng giải thuật lập lịch khác nhau là khác nhau. Ví dụ, BFUC-VF thể hiện được vị thế của mình tại các nút 13,16,18 và 19, trong khi Min-EV lại khẳng định được mình ở nút 12 và 17. Lý do của tính tương đương nhau về hiệu quả giữa các giải thuật là: (1) nếu một khoảng trống là khả dụng cho một *burst* đến cần lập lịch và các khoảng trống mới được sinh ra sau khi lập lịch không đủ (và không có cơ hội) cho lập lịch một *burst* khác (tiếp sau đó) thì việc tối ưu của các giải thuật là không có giá trị. (2) Khi mật độ luồng tăng cao, khoảng trống được sinh ra là nhỏ không đủ để lập lịch cho một *burst* đến nào, nên các giải thuật lập lịch có khoảng trống không thể thực hiện được. (3) Việc lấp đầy khoảng trống còn phụ thuộc vào sự đồng nhịp giữa *burst* đến là khoảng trống. Có những khoảng trống có độ rộng đủ để lập lịch một *burst* đến nhưng *burst* đến lại quá sớm, hay quá trễ so với khoảng trống nên bị chồng lấp với khoảng trống và do đó không thể lập lịch được.

Bảng 1. So sánh số *burst* rơi tại các nút đối với các giải thuật

Nút	BFUC	MINEV	MAXEV	LAUCVF	FFUCVF
1	2128	2128	2128	38384	3192
2	92240	95368	91176	89176	106432
3	16448	16448	12384	12384	23768

4	1064	1064	1064	2128	1064
5	8256	8256	5192	7192	8256
6	8256	8256	8256	8256	6192
7	5128	5128	2064	2064	3064
8	5192	5192	5129	5192	3128
9	116088	120152	100960	111024	117088
10	4128	4128	4128	4128	6256
11	0	0	0	0	0
12	14384	14384	20512	17448	28640
13	164176	169304	178304	184432	169176
14	65664	65664	70728	75856	47280
15	39216	41344	47408	32152	36024
16	1554864	1523608	1588120	1648760	1775400
17	111392	109328	126776	113520	148224
18	10478264	10634968	10615904	10865672	12298072
19	21010592	21048976	21482640	22121200	24745336
20	326432	316496	327496	401904	480160

Tóm lại, không có giải thuật lập lịch nào tối ưu cho tất cả các nút mạng. Mỗi giải thuật chỉ phù hợp với một số tình trạng mật độ luồng và tình trạng *burst* đến cần lập lịch. Một mô hình lập lịch thích nghi (chọn lựa giải thuật phù hợp nhất trong các giải thuật có thể) sẽ tìm kiếm được giải pháp tối ưu cho việc lập lịch tại mỗi nút. Đây là hướng mở cho việc cải tiến các mô hình lập lịch trong tương lai.

Việc lập lịch có thể thực hiện theo nhóm, khi một tập các *burst* cần lập lịch đến đồng thời. Giả sử tại thời điểm đó tại một nút đang xem xét có một tập các khoảng trống khả dụng. Mục tiêu được đặt ra là, hoặc tối đa số *burst* có thể lập lịch được vào các khoảng trống (nếu số *burst* đến nhiều hơn số khoảng trống có thể lập lịch), hoặc là tối ưu hóa việc chọn các khoảng trống vừa khít nhất cho các *burst* đến (nếu số *burst* đến ít hơn số khoảng trống có thể lập lịch). Đây là loại bài toán tối ưu với độ phức tạp NP-khó và giải pháp thông thường được chọn là các giải thuật *heuristic*. Kết quả lập lịch có thể đạt được với hướng tiếp cận này thường là xấp xỉ tối ưu, thay vì là tối ưu. Đây cũng là một hướng mở khác cho việc cải tiến các mô hình lập lịch.

4. Kết luận

Việc lập lịch là một trong những hoạt động có ảnh hưởng lớn đến hiệu năng mạng chuyển mạch chùm quang. Thực tế, đã có khá nhiều giải thuật lập lịch khác nhau được đề nghị, nhưng chủ yếu dựa trên ý tưởng là có hay không lấp đầy khoảng trống. Bài viết đã trình bày chi tiết các giải thuật lập lịch này, phân tích và đánh giá hiệu năng dựa trên kết quả mô phỏng trên gói NS2-obs0.9a. Các kết quả mô phỏng đã chỉ ra rằng các giải thuật có xét đến lấp đầy khoảng trống là hiệu quả hơn (dựa trên số *burst* rơi) các giải thuật không lấp đầy khoảng trống. Đối với các giải thuật có xét đến lấp đầy khoảng trống, hiệu quả của chúng chênh lệch nhau không đáng kể và giải thuật lập lịch tốt nhất là BFUC. Như đã phân tích, hiệu quả của các giải thuật lấp đầy khoảng trống phụ thuộc rất nhiều vào tình trạng luồng bên trong mạng. Do đó, việc chọn một giải thuật lập lịch có xét đến lấp đầy khoảng trống cho toàn mạng hay từng giải thuật khác nhau cho từng nút khác nhau phụ thuộc vào khả năng phân tích mật độ luồng đi qua tại từng nút khi xem xét một mạng cụ thể.

TÀI LIỆU THAM KHẢO

- [1]. Y. Xiong, M. Vandenhouste, and H.C. Cankaya, *Control Architecture in Optical Burst Switched WDM Networks*, IEEE Jsc, Vol. 18, No. 10, (2000), 1838 - 1851.
- [2]. J. Xu, C. Qiao, J. Li, and G. Xu, *Efficient Channel Scheduling Algorithms in Optical Burst Switching Networks*, in Proceeding of IEEE Infocom, Vol. 3, (2003), 2268 - 2278.
- [3]. J.P. Jue and V.M. Vokkarane, *Optical Burst Switched networks*, Springer, 2005.
- [4]. K.Dozer, C.Gauger, J.Spath, and S.Bodamer, *Evaluation of Reservation Mechanisms for Optical Burst Switching*, AEU International Journal of Electronics and Communications, Vol. 55, No. 1, (2001).
- [5]. M. Nandi, A.K. Turuk, D.K. Puthal and S. Dutta, *Best Fit Void Filling Algorithm in Optical Burst Switching Networks*, in Proceeding of IEEE Ictet, (2009), 609 - 614.
- [6]. M. Yang, S.Q. Zheng, and D.Verchere, *A QoS Supporting Scheduling Algorithm for Optical Burst Switching DWDM Networks*, in Proceeding of Globecom 01, (2001), 86 - 91.
- [7]. M. Yoo, C. Qiao, and S. Dixit, *QoS Performance of Optical Burst Switching in IP-Over-WDM Networks*, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 10, (2000).
- [8]. M. Ljolje, R. Inkret, and B. Mikac, *A Comparative Analysis of Data Scheduling Algorithms in Optical Burst Switching Networks*, in Proceeding of Optical Network Design and Modeling, (2005), 493 - 500.

- [9]. W.M. Golab and R. Boutaba, *Resource Allocation in User-Controlled Circuit-Switched Optical Networks*, LNCS Springer -Verlag, Vol. 16, No. 12, (1998), 2081 - 2094.
- [10]. Obs-ns Simulator. <http://www.wine.icu.ac.kr/obsns/>.
- [11]. Ns2 Simulator. <http://www.isi.edu/nsnam/ns>.

PERFORMANCE ANALYSES OF SCHEDULING ALGORITHMES IN OBS NETWORKS

Vo Viet Minh Nhat¹, Nguyen Hong Quoc²

¹*Faculty of Hospitality and Tourism, Hue University*

²*College of Education, Hue University*

Abstract. Scheduling is one of the operations that have a considerable influence on the performance of OBS networks. There are many scheduling algorithms proposed which are based on the idea of void filling or not. The main objective of these algorithms is to maximize the number of bursts scheduled and also to optimize the performance using network resources. This paper presents an analysis of the performance of scheduling algorithms, basing on simulation results in the NS-OBS package.

Keywords: Optical Burst Switching, Scheduling algorithms, NS-OBS .