

CÁC YẾU TỐ ẢNH HƯỞNG ĐẾN HIỆU NĂNG THUẬT TOÁN LẬP LỊCH TRÊN MẠNG CHUYỂN MẠCH CHÙM QUANG OBS

THE FACTORS AFFECT TO PERFORMANCE OF SCHEDULING ALGORITHMS IN OBS NETWORK

Phạm Trung Đức

Công ty Quản lý bến xe Thừa Thiên Huế, Huế

TÓM TẮT

Trong mạng chuyển mạch chùm quang khi một gói điều khiển đến một nút lõi, một giải thuật lập lịch được thực hiện để ấn định chùm chưa được lập lịch lên một kênh dữ liệu trên liên kết ra. Mục đích của việc lập lịch giúp nâng cao hiệu năng sử dụng mạng. Bài báo này nhằm tập trung phân tích đến các yếu tố ảnh hưởng đến hiệu năng thuật toán lập lịch như độ phức tạp, số lượng thông tin trạng thái đã sử dụng, mức độ khai thác băng thông của thuật toán cũng như dựa trên các thiết bị hỗ trợ khác để đánh giá hiệu năng giải thuật lập lịch và mô phỏng trên gói OBS-NS để thể hiện sự ảnh hưởng đó thông qua tỷ lệ mất chùm.

Từ khóa: Mạng chuyển mạch chùm quang; thuật toán lập lịch; OBS-NS.

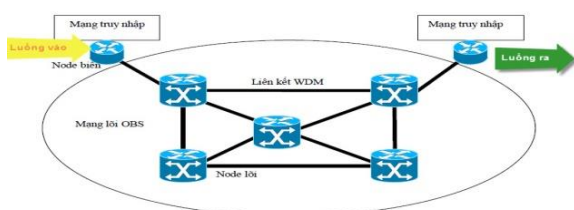
ABSTRACT

In optical burst switching network, when a control packet arrives a core node, a scheduling algorithms is carried to assign burst which has not been scheduled for a data channel on outgoing link. The purpose of scheduling is improving network utilization performance. This paper analyzes mainly the factors affecting to performance of scheduling algorithms in OBS network such as complexity measure, number of used state information, exploiting level of algorithms' bandwidth as well as basing on other supported devices to evaluate performance of scheduling algorithms and simulate on the NS-OBS package in order to show these influence through burst loss ratio.

Keywords: Optical Burst Switching; scheduling algorithms; NS-OBS.

1. Giới thiệu

Mạng truyền dẫn quang với những kỹ thuật truyền tin tiên tiến là giải pháp hữu hiệu nhằm đáp ứng nhu cầu truyền thông tăng cao. Kiến trúc của mạng gồm các nút biên, nút lõi, được kết nối với nhau bằng các sợi quang như Hình 1. Mạng chuyển mạch chùm quang là kỹ thuật giải quyết những vấn đề phát sinh trong truyền tải thông tin quang. Áp dụng mô hình chuyển mạch chùm quang tại các nút chuyển mạch làm nâng cao hiệu năng truyền tải thông tin đối với việc xử lý và định hướng thông tin từ nguồn đến đích một cách hiệu quả.

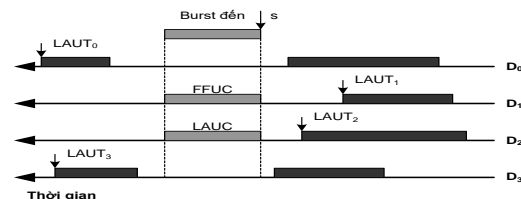


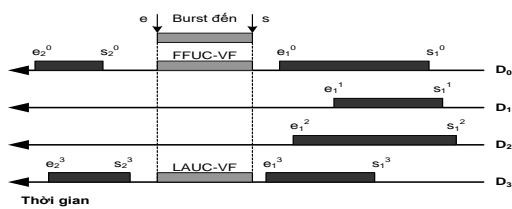
Hình 1. Kiến trúc mạng chuyển mạch chùm quang [2]

Hơn nữa mạng chuyển mạch chùm quang không yêu cầu các bộ đệm quang và chuyển mạch nhanh [4], do đó việc điều khiển lưu lượng, tránh tranh chấp xảy ra và nâng cao hiệu quả sử dụng tài nguyên mạng phụ thuộc vào các kỹ thuật lập lịch chùm tại các nút biên và nút lõi. Việc điều khiển lưu lượng, tránh tranh chấp xảy ra và nâng cao hiệu quả sử dụng mạng có thể được thực hiện nhờ các kỹ thuật lập lịch chùm.

Các thuật toán lập lịch cho kênh dữ liệu có thể được phân thành 2 loại, bao gồm:

+ Thuật toán không lấp đầy khoảng trống (without void filling) bao gồm thuật toán FFUC (First Fit Unscheduled Channel) và LAUC (Latest Available Unscheduled Channel)



Hình 2. Lập lịch không xét đến lấp đầy khoảng trống**Hình 3.** Lập lịch có xét đến lấp đầy khoảng trống

+ Thuật toán lấp đầy khoảng trống (with void filling) bao gồm thuật toán lấp đầy cả lỗ trống FFUC-VF (First Fit Unscheduled Channel with Void Filling), thuật toán lấp đầy một phần lỗ trống LAUC-VF (Latest Available Unscheduled Channel with Void Filling) và Min-EV (Minimum Ending Void), và thuật toán lấp đầy tối ưu lỗ trống BFUC-VF (Best Fit Unscheduled Channel – Void Filling).

Qua việc phân loại các giải thuật lập lịch, bài báo sẽ chỉ ra các yếu tố ảnh hưởng đến hiệu năng của thuật toán lập lịch trên mạng chuyển mạch chùm quang như sự khác nhau về độ phức tạp của thuật toán lập lịch không lấp đầy và lấp đầy khoảng trống. Được diễn giải qua việc cài đặt thuật toán trong ngôn ngữ lập trình C++, một thuật toán dài hơn hay sử dụng nhiều vòng lặp “for”, nhiều câu điều kiện “if ... then” sẽ ảnh hưởng đến độ phức tạp thuật toán, qua đó chỉ ra rằng sự phức tạp thời gian của các thuật toán không lấp đầy khoảng trống là ít hơn các thuật toán lấp đầy khoảng trống; hay việc chỉ ra số lượng thông tin trạng thái của thuật toán lấp đầy khoảng trống là nhiều hơn so với thuật toán không lấp đầy khoảng trống và cuối cùng là mô phỏng tỷ lệ mất chùm của giải thuật lập lịch trên gói OBS-NS để kết luận tỷ lệ nghịch với mức độ sử dụng băng thông của thuật toán, hiệu quả khi sử dụng các thiết bị hỗ trợ trong việc làm giảm tỷ lệ mất chùm và nêu một ví dụ đánh giá mức độ sử dụng băng thông thông qua việc sử dụng khoảng trống để so sánh và đối chiếu.

Phần 2 sẽ phân loại các giải thuật lập lịch cơ sở, 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), Min-EV, BFUC-VF và phân tích các yếu tố ảnh

hưởng đến hiệu năng của thuật toán lập lịch trên mạng chuyển mạch chùm quang.

Kết quả mô phỏng trên gói OBS-NS, được trình bày ở phần 3, sẽ chỉ ra hiệu năng của từng giải thuật và qua đó thấy được sự tác động qua lại của các yếu tố được phân tích và cần thiết phải có một mô hình chọn lựa các giải thuật tại mỗi nút khi thực hiện lập lịch.

2. Một số giải thuật lập lịch cơ sở và các yếu tố ảnh hưởng đến hiệu năng thuật toán lập lịch trên mạng chuyển mạch chùm quang

Phần này chỉ trình bày một số thuật toán lập lịch cơ sở của mạng chuyển mạch chùm quang bao gồm các giải thuật lấp đầy và không lấp đầy khoảng trống. Chi tiết đầy đủ về các giải thuật lập lịch trong mạng chuyển mạch chùm quang có thể xem trong [2], [3].

2.1. Tổng quan về kỹ thuật lập lịch trong mạng chuyển mạch chùm quang

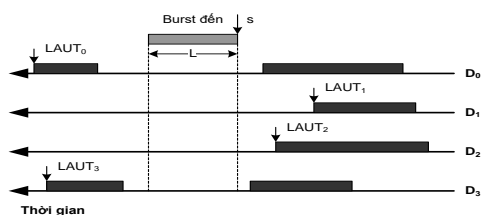
2.1.1. Lập lịch trên mạng chuyển mạch chùm quang

Khi một gói tin điều khiển tới tại một nút lõi, một thuật toán lập lịch kênh được gọi để gán chùm chưa được lập lịch với một kênh dữ liệu ở công ra. Bộ lập lịch kênh nhận được thời gian đến của chùm và khoảng thời gian chùm chưa được lập lịch từ gói tin điều khiển. Thuật toán có thể cần duy trì thời điểm chưa lập lịch khả dụng gần nhất (LAUT), các khoảng hở (gap) và các khoảng trống (voids) trên mọi kênh dữ liệu ra.

2.1.2. Phân loại các thuật toán lập lịch

Các thuật toán lập lịch cho kênh dữ liệu có thể được phân thành 2 loại, trong phạm vi bài báo sẽ nghiên cứu các thuật toán tương ứng sau: Thuật toán không lấp đầy khoảng trống (without void filling) bao gồm thuật toán FFUC và LAUC.

Thuật toán lấp đầy khoảng trống (with void filling) bao gồm thuật toán lấp đầy cả lỗ trống FFUC-VF, thuật toán lấp đầy một phần lỗ trống LAUC-VF và Min-EV, và thuật toán lấp đầy tối ưu lỗ trống BFUC-VF.



Hình 4. Lập lịch xét đến có hay không lấp đầy khoảng trống

- L: độ dài chùm đến chưa được lập lịch.
- D_i : kênh dữ liệu ra thứ i .
- $LAUT_i$: thời điểm chưa lập lịch khả dụng gần nhất của kênh dữ liệu thứ i ($i = 0, 1, 2, \dots, W-1$).
- s: thời điểm bắt đầu lập lịch.
- e: thời điểm kết thúc lập lịch.

2.2. Các yếu tố ảnh hưởng đến hiệu năng lập lịch trên mạng chuyển mạch chùm quang

2.2.1. Độ phức tạp thuật toán

Dựa trên bài báo [5] tác giả đã chỉ ra sự khác nhau về độ phức tạp của các thuật toán lập lịch không lấp đầy và lấp đầy khoảng trống. Để thấy rõ hơn điều này có thể được diễn giải qua việc cài đặt thuật toán trong ngôn ngữ lập trình C++, từ đó xác định độ phức tạp của từng thuật toán làm cơ sở để biết được thuật toán nào hiệu quả hơn trong việc làm giảm tỷ lệ mất chùm, qua đó rút ra kết luận độ phức tạp càng lớn thì tỷ lệ nghịch với tỷ lệ rơi chùm và chỉ ra rằng sự phức tạp thời gian của các thuật toán không lấp đầy khoảng trống là ít hơn các thuật toán lấp đầy khoảng trống, mà cụ thể được xác định như sau:

a) Độ phức tạp thuật toán của các thuật toán lập lịch không lấp đầy khoảng trống

Thuật toán FFUC ($O(\log W)$)

```
for( u_int i = ncc_; i < maxChannels_; i++ )
{
    if( schedTime >= unschTime_[i] )
    {
        result.channel() = i;
        result.startTime() = schedTime;
        break;
    }
}
```

```
}
```

Thuật toán LAUC ($O(W)$)

```
for( u_int i = ncc_; i < maxChannels_; i++ )
```

```
{
```

```
    if( schedTime >= unschTime_[i] )
```

```
    if( ( schedTime - unschTime_[i] ) < diffTime )
```

```
    { diffTime = schedTime - unschTime_[i];
```

```
      result.channel() = i;
```

```
      result.startTime() = schedTime;
```

```
    } }
```

b) Độ phức tạp thuật toán của các thuật toán lập lịch lấp đầy khoảng trống

Thuật toán FFUC-VF ($O(W \log N_b)$)

```
for( u_int i = ncc_; i < maxChannels_; i++ )
```

```
{
```

```
    if( schedTime >= startTime_[i] )
```

```
    if( ( endTime_[i] - schedTime >= schedDur )
```

```
    {
```

```
        result.channel() = i;
```

```
        result.startTime() = schedTime;
```

```
    }
```

```
    if( schedTime >= unschTime_[i] )
```

```
    {
```

```
        result.channel() = i;
```

```
        result.startTime() = schedTime;
```

```
        break;
```

```
    }
```

```
}
```

Thuật toán LAUC-VF ($O(W \log N_b)$)

```
for( u_int i = ncc_; i < maxChannels_; i++ )
```

```
{
```

```
    if( schedTime >= startTime_[i] )
```

```
    if( ( endTime_[i] - schedTime >= schedDur )
```

```
    if( ( schedTime - startTime_[i] ) < diffTime )
```

```
    { diffTime = schedTime - startTime_[i];
```

```
      result.channel() = i;
```

```
      result.startTime() = schedTime;
```

```
    }
```

```
    if( schedTime >= unschTime_[i] )
```

```

if( ( schedTime - unschTime_[i] ) < diffTime)
{diffTime = schedTime - unschTime_[i];
  result.channel() = i;
  result.startTime() = schedTime;
}
}

```

Thuật toán Min-EV ($O(\log_2 N_b)$)

```

for( u_int i = ncc_; i < maxChannels_; i++ )
{
  if( schedTime >= startTime_[i] )
  if( ( endTime_[i] - schedTime ) >= schedDur )
  if( ( schedTime - startTime_[i] ) < diffTime )
  {
    diffTime = schedTime - startTime_[i];
    result.channel() = i;
    result.startTime() = schedTime;
  }
}
if( result.channel() >= 0)
break;
for( u_int i = ncc_; i < maxChannels_; i++ )
{ if( schedTime >= unschTime_[i] )
  if( ( schedTime - unschTime_[i] ) < diffTime )
  { diffTime = schedTime - unschTime_[i];
    result.channel() = i;
    result.startTime() = schedTime;
  }
}
}

```

Thuật toán BFUC-VF ($O(\log_2 N_b)$)

```

for( u_int i = ncc_; i < maxChannels_; i++ )
{
  if( schedTime >= startTime_[i] )
  if( ( endTime_[i] - schedTime ) >= schedDur )
  if( ( endTime_[i] - startTime_[i] ) < diffTime )
  { diffTime = endTime_[i] - startTime_[i];
    result.channel() = i;
    result.startTime() = schedTime;
  }
}

```

```

}
if( result.channel() >= 0)
break;
for( u_int i = ncc_; i < maxChannels_; i++ )
{ if( schedTime >= unschTime_[i] )
  if( ( schedTime - unschTime_[i] ) < diffTime )
  { diffTime = schedTime - unschTime_[i];
    result.channel() = i;
    result.startTime() = schedTime;
  }
}
}

```

c) So sánh độ phức tạp

Bảng 1. So sánh độ phức tạp giữa các thuật toán lập lịch [1], [3]

Thuật toán	FFUC	LAUC	FFUC-VF	LAUC-VF	MIN-EV	BFUC-VF
Code tạo ra	Đơn giản	Đơn giản	Đơn giản	Phức tạp	Đơn giản	Đơn giản
Độ phức tạp	$O(\log W)$	$O(W)$	$O(W \log N_b)$	$O(W \log N_b)$	$O(\log_2 N_b)$	$O(\log_2 N_b)$

Trong đó, W: Số bước sóng tại mỗi công ra.

N_b : Số chùm đã lập lịch trên mỗi kênh dữ liệu.

2.2.2. Số lượng thông tin trạng thái

Thuật toán lập lịch lấp đầy khoảng trống sử dụng nhiều thông tin trạng thái được lưu để so sánh chùm đến. Thuật toán lập lịch không lấp đầy khoảng trống chỉ lưu giữ thông tin về Horizon và LAUT, còn thuật toán lập lịch lấp đầy khoảng trống lưu thời điểm kết thúc và bắt đầu của chùm đã lập lịch để so sánh chùm mới đến chưa lập lịch.

Bảng 2. So sánh số lượng thông tin trạng thái các thuật toán lập lịch [1]

Thuật toán	Không lấp đầy khoảng trống	Lấp đầy khoảng trống
Thông tin trạng thái	Horizon _i	$S_{i,j}E_{i,j}$

Trong đó: Horizon_i: Horizon của kênh dữ liệu thứ i.

$S_{i,j}$ và $E_{i,j}$: Thời gian bắt đầu và kết thúc của số lượng tối đa các chùm dữ liệu j trên kênh i.

2.2.3. Mức độ sử dụng băng thông

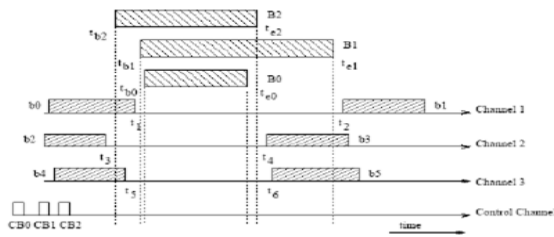
Thuật toán lập lịch được coi là đạt hiệu quả cao khi tỷ lệ mất chùm rơi ít trong quá trình lập lịch, điều đó được thể hiện ngược lại qua việc sử dụng băng thông (tức là băng thông sử dụng cao thì số lượng gói tin rơi sẽ thấp). Câu hỏi đặt ra là làm thế nào để nắm được việc sử dụng băng thông trong các thuật toán lập lịch, trong phần tiếp theo sẽ nêu ra 2 phương pháp để thực hiện điều này:

a) Đánh giá mức độ sử dụng băng thông thông qua mô phỏng

Bảng 3. So sánh mức độ sử dụng băng thông của các thuật toán lập lịch [1], [3]

Thuật toán	FFUC	LAUC	FFUC-VF	LAUC-VF	MIN-EV	BFUC-VF
Tỷ lệ mất chùm	Cao	Cao	Thấp	Thấp	Thấp	Thấp
Băng thông dùng	Thấp	Thấp	Cao	Cao	Cao	Cao

b) Đánh giá mức độ sử dụng băng thông qua việc sử dụng khoảng trống



Hình 5. Sự mất khoảng trống của thuật toán LAUC-VF và Min-EV [5]

Tính toán yếu tố việc sử dụng khoảng trống trong thuật toán lập lịch lấp đầy khoảng trống qua công thức sau:

$$\text{utilization} = (a \times 100) / x$$

với a là chiều dài chùm dữ liệu, x là chiều dài khoảng trống. Giá trị số thể hiện trong hình 5: $t_2 - t_1 = 12\mu s$, $t_4 - t_3 = 10\mu s$, $t_6 - t_5 = 8\mu s$, và chiều dài của chùm dữ liệu $B_0 = 5\mu s$. Khoảng trống sử dụng trong thuật toán:

- LAUC-VF, sử dụng = $(5 * 100) / 12 = 41,67\%$
- MIN-EV, sử dụng = $(5 * 100) / 10 = 50\%$
- BFUC-VF, sử dụng = $(5 * 100) / 8 = 62,5\%$.

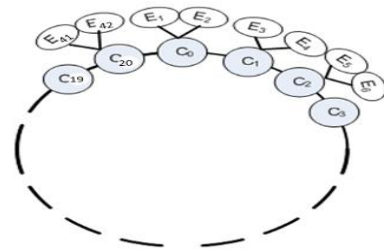
Điều này cho thấy, khoảng trống sử dụng trong thuật toán BFUC-VF được tốt hơn so với 2 thuật toán so sánh [5].

2.2.4. Dựa trên các thiết bị hỗ trợ [6], [7], [8]

Các thiết bị hỗ trợ đánh giá hiệu năng của các thuật toán lập lịch bao gồm bộ chuyển đổi bước sóng và đường trễ FDL [6], [7], [8]. Trong phạm vi bài báo, sử dụng đường trễ FDL để đánh giá sự ảnh hưởng đến hiệu năng các thuật toán lập lịch. Sử dụng thuật toán lập lịch không lấp đầy khoảng trống và lấp đầy khoảng trống kết hợp với đường trễ cố định để mô phỏng so sánh hiệu quả trong chương 3.

3. Mô phỏng và phân tích kết quả

Mô phỏng về các yếu tố ảnh hưởng đến hiệu năng lên việc lập lịch được thực hiện trên gói OBS-0.9a [10] của phần mềm mô phỏng NS [9]. Hình thái của mạng OBS thực hiện mô phỏng là một mạng hình vòng được tạo thành từ 21 nút lõi ($C_i, i=0..20$), mỗi nút lõi kết nối với hai nút biên ($E_i, i=1..42$) như mô tả ở hình 6. Các luồng dữ liệu được tạo ra liên tục (theo phân bố poisson) giữa các cặp nút E_i và E_j ($i,j=1..42$) với mật độ dày đặc. Các chùm do đó được sinh ra tại các thời điểm thay đổi và có kích thước thay đổi.



Hình 6. Hình thái mạng OBS mô phỏng tạo thành từ 21 nút lõi, 42 nút biên.

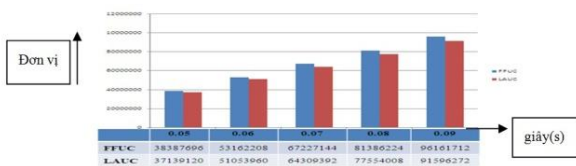
Mô phỏng được thực hiện trong các khoảng thời gian khác nhau (từ 0.05 đến 0.09 giây), kết quả mô phỏng (Hình 7) chỉ ra rằng các giải thuật lập lịch LAUC hiệu quả hơn FFUC vì tối ưu khoảng cách giữa các burst. So sánh mức độ sử dụng băng thông của thuật toán lập lịch lấp đầy và không lấp đầy khoảng trống qua việc phân tích số lượng gói tin rơi trong các trường hợp mô phỏng sau:

3.1. So sánh mức độ sử dụng băng thông thuật toán lập lịch

a) So sánh 2 thuật toán FFUC và LAUC

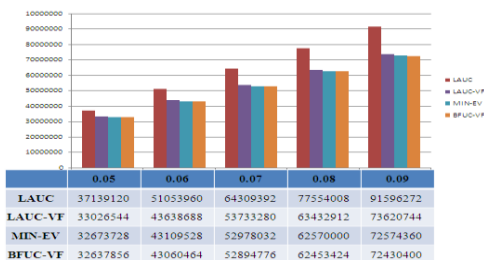
Kết quả mô phỏng hình 7 chỉ ra rằng các

thuật toán lập lịch LAUC hiệu quả hơn FFUC, thể hiện ở tỷ lệ mất chùm dữ liệu ít hơn.



Hình 7. Biểu đồ so sánh 2 thuật toán lập lịch FFUC và LAUC

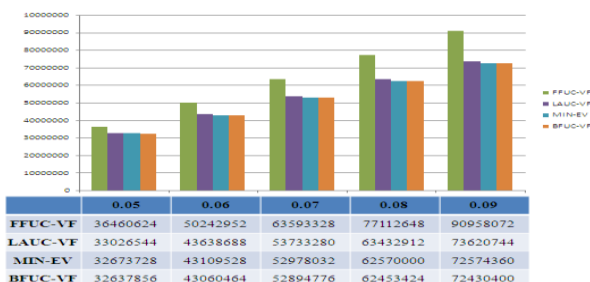
b) So sánh 2 nhóm thuật toán lập lịch không lấp đầy khoảng trống và có lấp đầy khoảng trống



Hình 8. Biểu đồ so sánh 2 nhóm thuật toán lập lịch có và không lấp đầy khoảng trống

Hình 8 cho thấy rằng nhóm thuật toán có lấp đầy khoảng trống (LAUC-VF, MIN-EV và BFUC-VF) là hiệu quả hơn rõ rệt so với nhóm thuật toán không lấp đầy khoảng trống (đại diện là thuật toán LAUC) thể hiện ở tỷ lệ mất chùm thấp hơn. Nguyên nhân là do nhóm thuật toán 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 thuật toán lập lịch không lấp đầy khoảng trống lại không xét đến.

c) So sánh các thuật toán lập lịch có lấp đầy khoảng trống với các tiêu chí tối ưu khác nhau

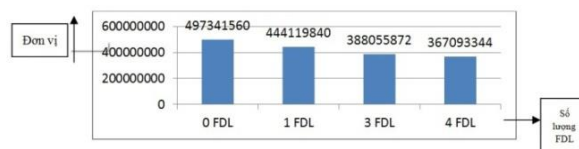


Hình 9. Biểu đồ so sánh 4 thuật toán lập lịch lấp đầy khoảng trống

Kết quả mô phỏng ở hình 9 cho thấy tỷ lệ mất chùm của các thuật toán LAUC-VF, MIN-EV và BFUC-VF chênh nhau không đáng kể. Nếu xét tỷ lệ mất chùm trên toàn mạng (trường hợp các nút mạng sử dụng cùng một thuật toán lập lịch), thuật toán BFUC-VF thể hiện hiệu quả cao nhất.

3.2. So sánh hiệu quả của việc sử dụng đường trễ FDL

Việc sử dụng thuật toán kết hợp đường trễ FDL làm cho số lượng chùm giảm đi đáng kể, thể hiện trong Hình 10.

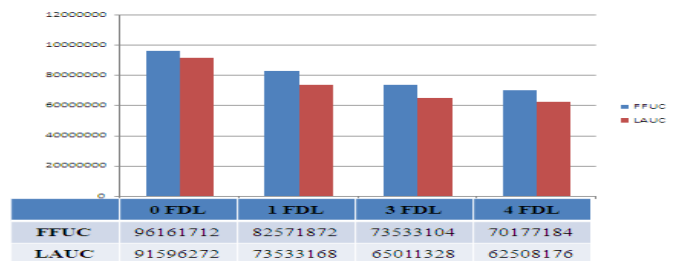


Hình 10. Hiệu quả thuật toán kết hợp đường trễ FDL

Kết quả mô phỏng áp dụng với tổng tỷ lệ mất chùm trên các thuật toán lập lịch FFUC, LAUC, FFUC-VF, LAUC-VF, MIN-EV và BFUC-VF chỉ ra rằng khi có sử dụng đường trễ FDL (với 1 FDL), số chùm rơi trên toàn mạng giảm đáng kể (tỉ lệ 10,70 %) so với khi không sử dụng đường trễ FDL (0 FDL). Khi số đường trễ tăng thêm (3 FDL), số chùm rơi càng giảm nhiều hơn (tỉ lệ 21,97 %) và (4 FDL) là (tỉ lệ 26,19 %).

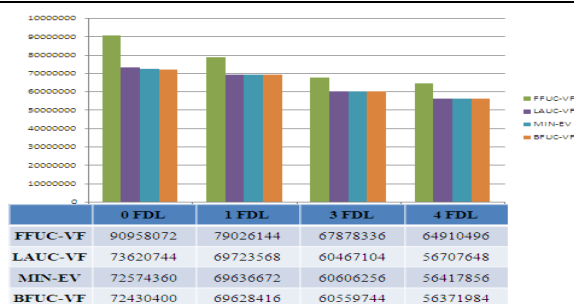
a) Nhóm thuật toán không lấp đầy khoảng trống

Kết quả mô phỏng (Hình 11) chỉ ra rằng thuật toán lập lịch không lấp đầy khoảng trống với đường trễ cố định LAUC-FDL hay FFUC-FDL là hiệu quả hơn nhiều so với thuật toán LAUC và FFUC.



Hình 11. Biểu đồ so sánh 4 thuật toán LAUC và FFUC có và không sử dụng FDL

b) Nhóm thuật toán có lấp đầy khoảng trống



Hình 12. Biểu đồ so sánh 2 thuật toán lấp đầy khoảng trống

Trong tự, thuật toán lập lịch có lấp đầy khoảng trống với đường trề cố định LAUC-VF-FDL cũng hiệu quả hơn so với thuật toán LAUC-VF, như mô tả trong Hình 12.

4. Kết luận

Việc nghiên cứu về mạng chuyển mạch chùm quang OBS và những thuật toán lập lịch chùm bước đầu đã thu được những kết quả quan trọng. Trong bài báo này, tôi đã tập trung trình bày về các yếu tố ảnh hưởng đến hiệu năng thuật toán lập lịch dựa trên độ phức tạp, số lượng thông tin trạng thái và việc sử dụng băng thông. Qua phân tích các yếu

tố và cài đặt thuật toán đó, đã thực hiện mô phỏng thành công trên một mô hình mạng để thấy được tỷ lệ mất chùm, đánh giá mức độ sử dụng băng thông

của giải thuật lập lịch. Qua kết quả phân tích, cho thấy hiệu năng của các thuật toán lập lịch phụ thuộc trực tiếp đến các yếu tố được đề xuất, với yếu tố độ phức tạp của thuật toán lập lịch và số lượng thông tin trạng thái khi tăng thì tương ứng hiệu năng của thuật toán được sử dụng tốt, làm giảm tỷ lệ mất chùm khi gửi gói tin. Ngược lại, với mức độ sử dụng băng thông càng cao thì tỷ lệ mất chùm càng lớn làm giảm hiệu năng sử dụng của các thuật toán lập lịch. Ngoài ra, qua mô phỏng các thuật toán kết hợp FDL còn nêu lên được vai trò đường dây trề tới các thuật toán. Từ đó, bài báo giúp nêu bật được mỗi ưu, nhược điểm các thuật toán lập lịch nhờ chỉ định được các yếu tố ảnh hưởng đến hiệu năng của các thuật toán lập lịch trong mạng chuyển mạch chùm quang OBS.

TÀI LIỆU THAM KHẢO

- [1] Adgaonkar R.P., Sharma S.N (2011), *A review of burst scheduling algorithm in WDM Optical Burst Switching Network*, Journal of Computer Science Issues, Vol. 8, 6(3).
- [2] Jason P. Jue, Vinod M., Vokkarane (2005), *Optical Burst Switched Networks*, Springer Science Business Media, Inc.
- [3] Lamba Rohit., Kumar Garg Dr.Amit (2012), *Performance Analysis of Scheduling Algorithms In Optical Burst Switching Networks*, International Journal of Advanced Research in Computer Science and Software Engineering, 2(1).
- [4] Mukherjee Biswanath (2006), *Optical WDM Networks*, Springer Science + Business Media, Inc.
- [5] Nandi M., Turuk A. K., Puthal D. K., Dutta S.(2009), *Best Fit Void Filling Algorithm in Optical Burst Switching Networks*, in Proceeding of IEEE ICETET, 609 – 614.
- [6] Sam S.M., Faisal N., Ariffin S.H.S.(2007), *“Quality-of-Service Provisioning using Hierarchical Scheduling in Optical Burst Switched Network”*, Networking and Services, 2007.
- [7] Vo Viet Minh Nhat, Nguyen Hong Quoc (2011), *“The role of fdl in scheduling in obs networks”*, Journal of science, Hue University, 69(6).
- [8] Vokkarane V.M., Thodime G.P.V., Challagulla V.U.B., Jue J.P. (2003), *“Channel Scheduling Algorithms using Burst Segmentation and FDLs for Optical Burst-Switched Networks”*, in Proceedings IEEE International Conference on Communications (ICC).
- [9] Network Simultor, <http://www.isi.edu/nsnam/ns/>
- [10] Optical Internet Research Center, <http://wine.icu.ac.kr/~obsns>, <http://nile.wpi.edu/NS/>.