BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

MINISTRY OF INFORMATION AND COMMUNICATIONS
POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

ISSN 2525 - 2224

TẠP CHÍ KHOA HỌC CÔNG NGHỆ

THÔNG TIN VÀ TRUYỀN THÔNG

JOURNAL OF SCIENCE & TECHNOLOGY ON INFORMATION AND COMMUNICATIONS

ĐIỆN TỬ VIỄN THÔNG VÀ CÔNG NGHỆ THÔNG TIN

Số 02(CS.01)2020

TẠP CHÍ KHOA HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Journal of Science and Technology on Information and Communications

MỤC LỤC/CONTENTS

Lời nói đầu/Preface *2*

BÀI BÁO/REGULAR PAPERS

Dual-Band MIMO Antenna using Gradient Arcs for Construction Monitoring and Inspection Systems Based on IIoT
Anten MIMO hai băng sử dụng cấu trúc đường cong biến đổi đều cho hệ thống giám sát xây dựng trong kỷ nguyên IIoT
Dương Thị Thanh Tú, Dương Thị Thanh Hiền, Nguyễn Ngọc Thúy, Nguyễn Tuấn Ngọc *3*

Đề xuất hệ thống trong nhận dạng cử chỉ, hành động sử dụng trí tuệ nhân tạo cho các ứng dụng nhà thông minh
Proposing gesture algorithm using artificial inteligence for smart home applications
Nguyễn Hữu Phát, Nguyễn Thị Thu Hương *9*

A Boosting Classification Approach Based on SOM
Một phương pháp nâng cao khả năng phân loại dữ liệu của som sử dụng thuật toán boosting
Nguyễn Đình Hóa *15*

Một phương pháp xây dựng hệ mật pohlig-hellman trên vành đa thức
One method of implementing pohlig-hellman cryptosystem over polynomial rings
Ngô Đức Thiện *21*

Sinh dữ liệu kiểm thử cho mô hình hệ thống nhúng sử dụng kỹ thuật kiểm thử theo cặp
Test data generation for embedded model using pairwise testing technique
Đỗ Thị Bích Ngọc *27*

Giải pháp phân phối khóa lượng tử không dây lai ghép FSO và MMW
A solution of quantum key distribution over hybrid wireless system using FSO and MMW
Phạm Anh Thư, Đặng Thế Ngọc *33*

Cải thiện hiệu năng hệ thống QKD với kỹ thuật truyền lại khóa
Performance improvement of QKD systems using key retransmission technique
Phạm Thị Thúy Hiền *41*

Tìm kiếm sản phẩm theo ảnh bằng học sâu
Product image search using deep learning
Dương Trần Đức *47*

Nâng cao hiệu năng lập lịch của nút lõi OBS dựa trên phân tích dữ liệu và sử dụng đường trễ FDL
Improving the scheduling performance of core OBS nodes based on data analysis and using FDL
Phạm Trung Đức, Võ Viết Minh Nhật, Đặng Thanh Chương *53*

Đề xuất thuật toán khuyến nghị theo phân bố dựa trên mô hình hỗn hợp Gaussian
A proposal of robust content - based recommendation system using Gaussian mixture model
Nguyễn Văn Đạt, Tạ Minh Thanh *61*

Phát triển một dạng lược đồ chữ ký số mới dựa trên bài toán RSA
Developing a new type of digital signature scheme based on RSA problem
Pham Van Hiep, Luu Hong Dung *73*

Điều chỉnh tài nguyên tự động dựa vào tham số chất lượng dịch vụ trong Cloud Computing
Auto scaling based on quality of service parameters in Cloud Computing
Nguyễn Hồng Sơn *79*

NÂNG CAO HIỆU NĂNG LẬP LỊCH CỦA NÚT LÕI OBS DỰA TRÊN PHÂN TÍCH DỮ LIỆU VÀ SỬ DỤNG ĐƯỜNG TRỄ FDL

Phạm Trung Đức [1, 2], Võ Viết Minh Nhật [3], Đặng Thanh Chương [2]

[1] Công ty cổ phần Bến xe Huế

[2] Trường Đại học Khoa học, Đại học Huế

[3] Đại học Huế

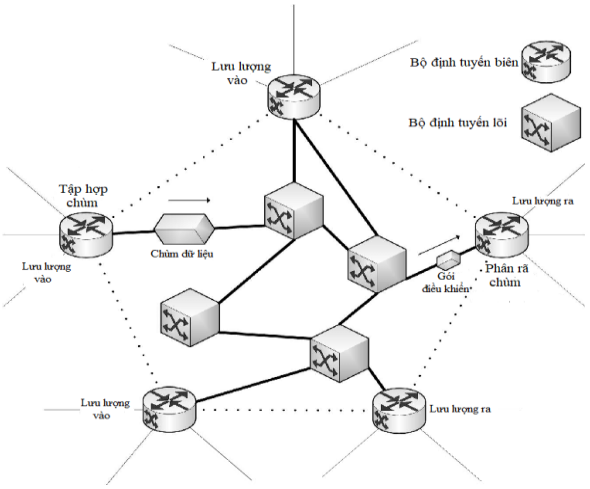
Tóm tắt: Hiệu năng lập lịch của nút lõi đóng một vai trò quan trọng đối với hiệu năng truyền thông chung của toàn mạng OBS. Nhằm nâng cao hiệu năng lập lịch của nút lõi, nhiều nghiên cứu tiến hành kết hợp các kỹ thuật lập lịch với chuyển đổi bước sóng, định tuyến lệch hướng hay sử dụng đường trễ FDL. Hiệu năng lập lịch tại một cổng ra của một nút lõi phụ thuộc vào nhiều yếu tố, trong đó thời gian đến của chùm, thời gian bắt đầu và kết thúc của các khoảng trống và thời điểm chưa được lập lịch khả dụng sau cùng nhất của mỗi kênh ra là các yếu tố có ảnh hưởng chính đến hiệu năng lập lịch. Bài báo này do đó đề xuất phương pháp phân tích dữ liệu trạng thái lập lịch nhằm tìm ra nguyên nhân lập lịch không thành công. Trên cơ sở kết quả phân tích, bài báo chỉ ra rằng việc sử dụng đường trễ FDL nhằm tránh chồng lấp đầu và chồng lấp LAUT là giải pháp phù hợp nhất nhằm tăng tỉ lệ lập lịch thành công, giảm mất mát chùm và do đó tăng hiệu quả sử dụng băng thông trong mạng OBS.

Từ khóa: mạng OBS, lập lịch, phân tích dữ liệu, xử lý tắc nghẽn, làm trễ dựa trên FDL.

I. GIỚI THIỆU

Sợi quang được xem là phương tiện truyền dẫn dữ liệu hiệu quả nhất hiện nay, hay ít nhất là trong một tương lai gần, nhờ vào tiềm năng băng thông rất cao và khả năng truyền được dữ liệu đi xa với tỉ lệ suy giảm tín hiệu thấp. Cùng với những tiến bộ về công nghệ ghép kênh phân chia kênh bước sóng (*Wavelength Division Multiplexing, WDM*), các mô hình chuyển mạch quang cũng đã được phát triển, ban đầu là các mô hình chuyển mạch kênh quang (*Optical Circuit Switching, OCS*) với các kênh bước sóng dành riêng, đến các mô hình chuyển mạch gói quang (*Optical Packet Switching, OPS*) được lấy cảm hứng từ các mô hình chuyển mạch gói điện tử (chẳng hạn, mô hình chuyển mạch gói IP). Tuy nhiên, mô hình chuyển mạch chùm quang (*Optical Burst Switching, OBS*) mới thực sự là mô hình chuyển mạch gói quang khả dụng do nó có tính linh hoạt và độ mịn chuyển mạch ở mức "gói" dữ liệu, nhưng lại không yêu cầu các bộ chuyển mạch phức tạp và bộ đệm quang như OPS. Nói một cách khác, mạng OBS đã tích hợp được các ưu điểm của OCS và OPS, trong khi loại bỏ các hạn chế của mỗi loại chuyển mạch này [1][2].

Như được mô tả trong Hình 1, mạng OBS bao gồm các nút biên và nút lõi. Nút biên vào có nhiệm vụ tiếp nhận dữ liệu (ví dụ, các gói IP) đến từ các mạng truy cập, tập hợp chúng thành các chùm dữ liệu (*data burst*) và truyền chúng vào mạng lõi. Một gói điều khiển chùm (*Burst Control Packet - BCP*) được truyền đi trước một khoảng thời gian *offset* nhằm đặt trước tài nguyên và cấu hình các chuyển mạch tại các nút (lõi) trung gian; nhờ đó chùm dữ liệu theo sau sẽ được chuyển mạch toàn quang khi đến tại mỗi nút trung gian này trên hành trình của từ nguồn đến đích. Tại nút biên ra, một hoạt động ngược lại được thực hiện nhằm tách chùm và khôi phục lại dữ liệu ban đầu.



Hình 1. Kiến trúc tiêu biểu của mạng OBS.

Hiệu năng lập lịch của nút lõi đóng một vai trò quan trọng đối với hiệu năng truyền thông chung trong toàn mạng OBS. Có 2 loại kỹ thuật lập lịch đã được đề xuất: lập lịch không lấp đầy khoảng trống (*without void filling*) và lập lịch với lấp đầy khoảng trống (*with void filling*). Trong lập lịch không lấp đầy khoảng trống, một chùm đến sẽ được lập lịch sau thời điểm chưa được lập lịch khả dụng sau cũng nhất (*Lastest Available Unscheduled Time*,

Tác giả liên hệ: Phạm Trung Đức
Email: phamtrungduc@hueuni.edu.vn
Đến tòa soạn: 5/2020; chỉnh sửa: 7/2020; chấp nhận đăng: 7/2020

LAUT) như được chỉ ra ở Hình 2(a) và Hình 2(b), ngược lại với lập lịch với lấp đầy khoảng trống, chùm đến sẽ được lập lịch vào khoảng băng thông nhàn rỗi được tạo ra giữa hai chùm đã được lập lịch trước đó trên một kênh ra như từ Hình 2(c) đến Hình 2(f). Lập lịch lấp đầy khoảng trống được đánh giá là hiệu quả hơn về mặt sử dụng băng thông [3]. Tuy nhiên, việc lập lịch thành công phụ thuộc vào nhiều yếu tố như thời gian đến và chiều dài của chùm, LAUT, thời gian bắt đầu và kết thúc của khoảng trống trên các kênh ra. Bài báo sẽ trình bày một phương pháp phân tích dữ liệu trạng thái lập lịch nhằm xác định nguyên nhân gây ra việc lập lịch không thành công, từ đó đề xuất giải pháp sử dụng đường trễ trong việc tăng tỉ lệ lập lịch thành công, giảm mất mát dữ liệu và do đó tăng tỉ lệ sử dụng băng thông.

Các đóng góp chính của bài báo bao gồm:

- Đề xuất ý tưởng phân tích dữ liệu trạng thái lập lịch nhằm xác định các yếu tố chính ảnh hưởng đến việc lập lịch không thành công. Để thực hiện điều đó, một mô hình thu thập dữ liệu được đề xuất, ở đó mô hình mạng mô phỏng và cấu trúc dữ liệu lập lịch cần trích xuất được phân tích;

- Dữ liệu thu thập được sau đó được phân tích trên công cụ Weka [4], ở đó một số thuật toán đánh giá thuộc tính dữ liệu được thực hiện để tìm ra các yếu tố tác động đến việc lập lịch không thành công;

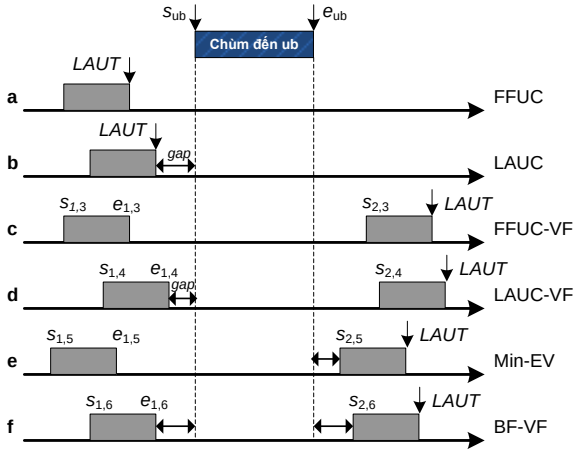
- Trên cơ sở các yếu tố tác động đến việc lập lịch không thành công được xác định, một mô hình lập lịch kết hợp sử dụng đường trễ FDL tại nút lõi được đề xuất nhằm tăng tỉ lệ lập lịch thành công và giảm mất mát dữ liệu.

Các phần tiếp theo trong bài báo này bao gồm: Phần II trình bày các giải thuật lập lịch trong đó các điều kiện lập lịch của các giải thuật khác nhau được phân tích. Trên cơ sở phân tích các trường hợp không thể lập lịch được tại Phần III, bài báo đã đề xuất mô hình trích xuất dữ liệu lập lịch, cấu trúc dữ liệu cần thu thập và cách tiếp cận phân tích nguyên nhân lập lịch không thành công. Một mô hình điều khiển lập lịch tiếp đó được đề xuất trong Phần IV, trong đó các đường trễ FDLs được sử dụng để tránh chồng lấp đầu và chồng lấp LAUT. Mô phỏng và phân tích kết quả mô phỏng được thực hiện trong Phần V để chỉ ra hiệu quả của các đề xuất. Cuối cùng, kết luận của bài báo được mô tả trong Phần VI.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Lập lịch trong mạng OBS là một hoạt động có tác động đáng kể đến hiệu năng truyền thông chung trong toàn mạng OBS. Lập lịch được thực hiện tại cả nút biên vào và nút lõi. Do nút biên vào được trang bị các bộ đệm cho quá trình tập hợp chùm nên việc lập lịch có thể được điều khiển dễ dàng bởi các kỹ thuật dựa trên bộ đệm, nếu có xung đột xảy ra. Tuy nhiên tại nút lõi, do không có bộ đệm, một chùm đến sẽ không được lập lịch nếu không tìm thấy tài nguyên tại cổng ra. Bài báo sẽ tập trung vào vấn đề lập lịch tại nút lõi.

Việc lập lịch một chùm đến lên một trong các kênh ra tại một cổng ra được mô tả như Hình 2. Một chùm đến thường được đặc trưng bởi cặp đại lượng: thời gian đến (s_{ub}) và độ dài chùm (len_{ub}) trong đó *ub* là ký hiệu của chùm chưa được lập lịch (*unscheduled burst*). Khi một chùm *ub* đến tại một cổng ra, một giải thuật lập lịch sẽ được gọi để tìm kiếm băng thông nhàn rỗi cho chùm *ub*. Băng thông nhàn rỗi là phần băng thông của một kênh ra mà chưa được đặt trước cho một chùm nào. Giả sử tại cổng ra có trang bị các bộ chuyển đổi đầy đủ, một chùm đến trên một bước sóng có thể được chuyển đổi thành bất kỳ bước sóng nào tương ứng với bước sóng của kênh ra có băng thông nhàn rỗi. Trong trường hợp cổng ra chỉ được trang bị các bộ chuyển đổi bước sóng giới hạn, miền chuyển đổi bước sóng đối với chùm đến sẽ bị thu hẹp; đặc biệt khi không có bộ chuyển đổi nào được trang bị, chùm đến chỉ được xem xét lập lịch trên kênh có cùng bước sóng với nó. Bài báo giả thiết cổng ra được trang bị các bộ chuyển đổi bước sóng đầy đủ, nên một chùm đến có thể được lập lịch lên một kênh bước sóng bất kỳ.



Hình 2. Ví dụ mô tả các kỹ thuật lập lịch khác nhau.

Mỗi kênh ra duy trì một giá trị LAUT, là thời điểm sau cùng nhất của băng thông khả dụng chưa được sử dụng. Tuy nhiên, như mô tả trong hình vẽ, các khoảng trống (*void*), khoảng băng thông nhàn rỗi được tạo ra giữa 2 chùm được lập lịch liên tiếp, thực tế là khá phổ biến trên mỗi kênh ra như trong các Hình 2(c), 2(d), 2(e) và 2(f), nên một chùm đến có thể được xem xét để lập lịch vào một trong các khoảng trống này. Mô hình lập lịch này được gọi là lập lịch lấp đầy khoảng trống. Như vậy, mỗi kênh ra, ngoài việc duy trì giá trị LAUT, thông tin về các khoảng trống: thời điểm bắt đầu (s_{void}) và kết thúc (e_{void}), cũng phải được duy trì.

Dựa vào cách thức khai thác băng thông nêu trên, các

kỹ thuật lập lịch được phân thành 2 loại: lập lịch không lấp đầy khoảng trống, như FFUC, LAUC [5] và lập lịch với lấp đầy khoảng trống, FFUC-VF, LAUC-VF [6], Min-EV [7], BF-VF [8].

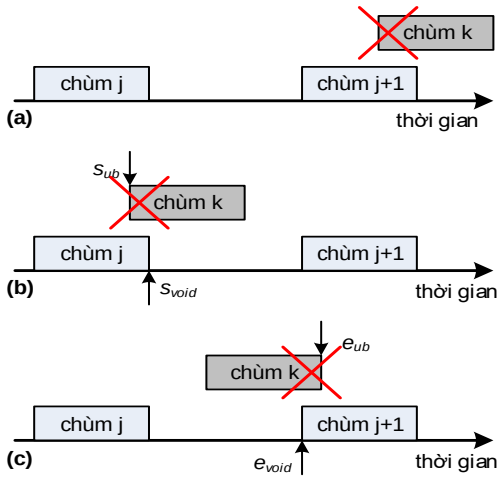
Như mô tả trong Hình 2, các kỹ thuật lập lịch không lấp đầy khoảng trống (FFUC và LAUC) bỏ qua thông tin khoảng trống và chỉ so sánh thời điểm đến của chùm đến (s_{ub}) và LAUT của mỗi kênh ra. Nếu $s_{ub} > LAUT_i$, chùm đến sẽ được lập lịch lên kênh thứ i. Sự khác biệt của FFUC và LAUC là trong khi FFUC chọn kênh khả dụng đầu tiên đáp ứng điều kiện lập lịch trên (Hình 2a), thì LAUC sẽ chọn kênh có khoảng hở (*gap*) nhỏ nhất (Hình 2b). Khoảng hở là khoảng cách từ thời điểm đến của chùm đến LAUT. Kỹ thuật LAUC có độ phức tạp lớn hơn FFUC vì phải duyệt qua tất cả các kênh để chọn kênh có khoảng hở bé nhất.

Với các kỹ thuật lập lịch lấp đầy khoảng trống, nếu $s_{ub} > LAUT_i$, tiến trình lấp đầy khoảng trống sẽ được kích hoạt. FFUC-VF sẽ tìm khoảng trống phù hợp đầu tiên ($s_{ub} > s_{void}$ và $len_{ub} < len_{void}$, trong đó $len_{void} = e_{void} - s_{void}$ là độ dài khoảng trống), như Hình 2c, trong khi LAUC-VF sẽ tìm khoảng trống có khoảng hở đầu, là khoảng cách từ thời điểm đến của chùm *ub* đến thời điểm bắt đầu của khoảng trống ($gap = s_{ub} - s_{void}$), bé nhất, như Hình 2d. Min-EV [7] sẽ tìm khoảng trống có khoảng hở đuôi, là khoảng cách từ thời điểm kết thúc của chùm đến đến thời điểm kết thúc của khoảng trống ($gap = e_{void} - (s_{ub} + len_{ub})$), bé nhất, như Hình 2e. BF-VF được đánh giá là kỹ thuật lấp đầy khoảng trống tốt nhất, trong đó một đại lượng về mức độ sử dụng băng thông (*utilization*) được định nghĩa:

$$utilization = (len_{ub} \times 100) / len_{void};$$

BF-VF sẽ chọn kênh với khoảng trống có *utilization* lớn nhất.

Một so sánh dựa trên mô phỏng về hiệu quả lập lịch giữa các giải thuật có lấp đầy khoảng trống đã được đánh giá trong [3], trong đó khi tải đến càng lớn thì tỉ lệ (xác suất) mất chùm càng cao, nhưng BF-VF luôn cho tỉ lệ mất mát thấp nhất. Tuy nhiên, một chùm đến vẫn có thể không được lập lịch mặc dù băng thông của kênh ra chưa bị khai thác hết. Chẳng hạn như ví dụ trong Hình 3, một chùm đến vẫn sẽ bị loại bỏ nếu nó chồng lấp LAUT (Hình 3(a)), chồng lấp đầu (Hình 3(b)) và chồng lấp đuôi (Hình 3(c)) với các chùm đã được lập lịch trước đó. Nếu chùm đến có chiều dài lớn hơn kích thước khoảng trống, nó cũng sẽ bị loại bỏ.



Hình 3. Các trường hợp không lập lịch được do chồng lấp: (a) LAUT, (b) đầu hay (c) đuôi.

III. PHÂN TÍCH DỮ LIỆU TRẠNG THÁI LẬP LỊCH

III.1 Đặt vấn đề

Xét một cổng ra có W kênh ra khả dụng và được trang bị các bộ chuyển đổi bước sóng đầy đủ, một chùm đến do đó có thể lập lịch lên một trong W kênh. Xét đối với một kênh i cụ thể, một chùm đến chỉ lập lịch được nếu thời điểm đến của nó sau $LAUT_i$ ($s_{ub} > LAUT_i$), hoặc nó lấp được vào khoảng trống ($s_{ub} > s_{void}$ và $s_{ub} + len_{ub} < e_{void}$) của kênh i này. Một chùm đến sẽ không lập lịch được nếu nó:

(1) chồng lấp LAUT, tức chồng lấp đuôi với chùm đã được lập lịch sau cùng nhất ($e_{void} < s_{ub} < LAUT$),

(2) chồng lấp đầu với khoảng trống ($s_{ub} < s_{void}$),

(3) chồng lấp đuôi với khoảng trống ($s_{ub} < e_{void}$ và $s_{ub} + len_{ub} > e_{void}$).

Như vậy, nếu có thể điều chỉnh được thời gian đến hay/và chiều dài thì một chùm đến sẽ có cơ hội được lập lịch cao hơn. Việc điều chỉnh này rõ ràng cần các thông tin như: thời gian đến và độ dài chùm, LAUT, thời gian bắt đầu và kết thúc của khoảng trống.

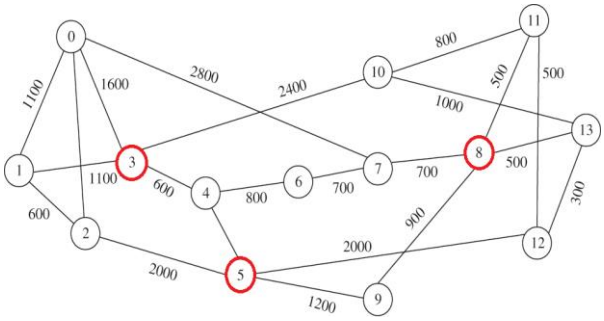
Một số yếu tố khác như: tải các chùm đến, băng thông khả dụng của cổng ra cũng có tác động đến xác suất lập lịch thành công. Tuy nhiên, 2 yếu tố này thực tế đã được bao hàm trong tần suất các chùm đến (thời gian đến giữa các chùm liên tiếp gần hay xa nhau) và kích thước bị thu nhỏ của các khoảng trống được tạo ra. Việc dựa vào tần suất (và độ dài) các chùm đến và kích thước các khoảng trống được sinh ra sẽ dự báo được tải chuẩn hoá (thương của tải đến với băng thông) tại cổng ra.

Tóm lại, các thuộc tính cần thiết cho dữ liệu lập lịch cần trích xuất là: thời gian đến và độ dài chùm, LAUT, thời gian bắt đầu và kết thúc của khoảng trống.

III.2 Trích xuất dữ liệu trạng thái lập lịch

Mô hình mạng được mô phỏng để trích xuất dữ liệu lập lịch là một mạng NFSNET bao gồm 14 nút, trong đó độ dài liên kết (km) giữa mỗi cặp nút được chuyển thành độ trễ truyền thông (s). Các luồng dữ liệu được thiết lập giữa các nút lõi có kết nối đến tại các nút biên là như mô tả trong Hình 4. Kết quả mô phỏng cho thấy tỷ lệ mất dữ liệu tại các nút 3, 5 và 8 là đáng kể và dữ liệu lập lịch được trích xuất tại đây có thể đại diện cho dữ liệu lập lịch chung tại các nút lõi trên toàn mạng.

Mô phỏng được triển khai trên một máy tính PC với cấu hình 2.4 GHz Intel Core 2 CPU, 2G RAM. Giả thiết rằng các gói tin đến tại các hàng đợi của 14 nút lõi OBS có phân phối Poisson; mỗi nút biên vào có ngưỡng tập hợp chùm lai với cặp giá trị: ngưỡng độ dài là 100Kb và ngưỡng thời gian 100μs; giá trị thời gian bù đắp tại mỗi nút được thiết lập là 10μs; số bước sóng trên mỗi liên kết ra là W = 8; băng thông của mỗi liên kết là 1Gbps. Mô phỏng được thực hiện trong NS2 [9] với gói mô phỏng mạng OBS là obs-0.9a.



Hình 4. Mạng NFSNET được sử dụng cho việc trích xuất dữ liệu lập lịch

Với thời gian mô phỏng là 1s, có 44959 mẫu (bản ghi) được trích xuất tại các nút lõi số 3, 5, 8. Ví dụ các mẫu dữ liệu lập lịch được mô tả như trong Bảng 1, trong đó trạng thái lập lịch được thể hiện dưới dạng nhị phân: *status* = 1 nếu chùm được lập lịch thành công và *status* = 0 nếu chùm không lập lịch được. Lưu ý, các thuộc tính (*attributes*) khác trong Bảng 1 được đo bằng đơn vị thời gian giây (s).

Bảng 1. Mô tả dữ liệu trạng thái lập lịch được trích xuất từ các nút lõi.

Thuộc tính (*attribute*) / Mẫu (*pattern*)	Thời điểm chùm đến (s_{ub})	Độ dài chùm đến (len_{ub})	LAUT (*LAUT*)	Bắt đầu khoảng trống (s_{void})	Kết thúc khoảng trống (e_{void})	Trạng thái lập lịch (*status*)
5	0.073854	0.000004608	0.073897	0.073852	0.073872	1
…	…	…	…	…	..	…
25	0.073953	0.000008704	0.073957	0.073917	0.073932	1
26	0.073917	0.00003328	0.07363	0.073317	0.073614	0
…	…	…	…	…	..	…

Bảng 2. Mô tả dữ liệu trạng thái lập lịch không thành công được trích xuất từ Bảng 1, trong đó các trạng thái nguyên nhân lập lịch không thành công bao gồm: LAUT_overlap, head_overlap và tail_overlap.

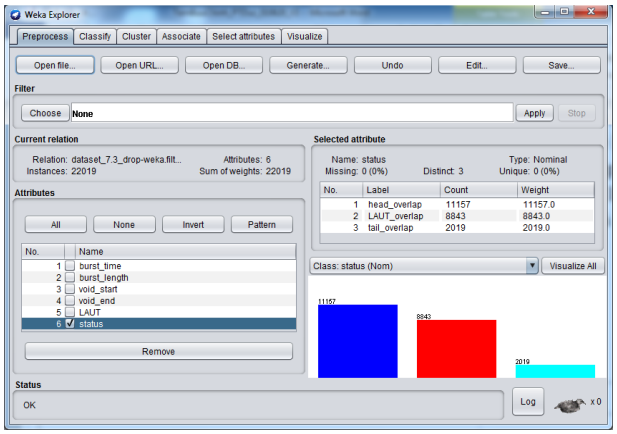
Thuộc tính (*attribute*) / Mẫu (*pattern*)	Thời điểm chùm đến (s_{ub})	Độ dài chùm đến (len_{ub})	LAUT (*LAUT*)	Bắt đầu khoảng trống (s_{void})	Kết thúc khoảng trống (e_{void})	Nguyên nhân lập lịch không thành công (*status*)
26	0.073917	0.00003328	0.073957	0.073917	0.073932	LAUT_overlap
27	0.073917	0.00003328	0.073962	0.073952	0.073953	head_overlap
28	0.073881	0.000016896	0.073917	0.073823	0.073896	tail_overlap
…	…	…	…	…	..	…

III.3 Xác định thuộc tính ảnh hưởng đến mất chùm

Trong 44959 mẫu thu thập được, có 22187 trường hợp lập lịch thành công với không lấp đầy khoảng trống, 752 trường hợp lập lịch thành công với lấp đầy khoảng trống và 22019 mẫu trường hợp lập lịch không thành công.

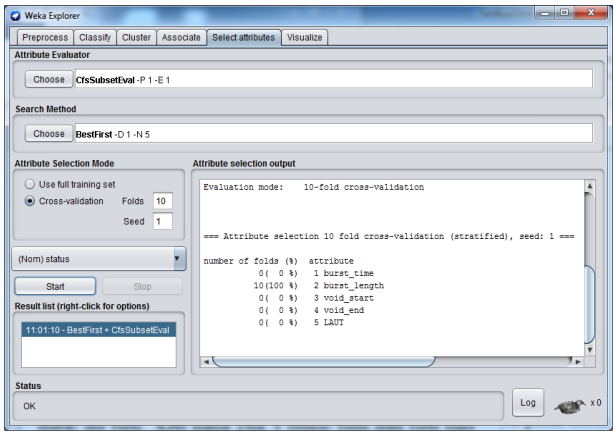
Điều chúng ta cần quan tâm là những thuộc tính nào có ảnh hưởng đến việc lập lịch không thành công này. Do đó, 22019 mẫu trường hợp lập lịch không thành công sẽ được phân tích. Như đã được mô tả trong mục III.1, nguyên nhân của lập lịch không thành công là do: (1) chồng lấp LAUT (*LAUT_overlap*), (2) chồng lấp đầu với khoảng trống (*head_overlap*), (3) chồng lấp đuôi với khoảng trống (*tail_overlap*). Dữ liệu lập lịch với trường hợp không thành công được thể hiện như Bảng 2.

Với phân bố dữ liệu lập lịch không thành công thuộc về 3 lớp: *LAUT_overlap*, *head_overlap* và *tail_overlap*, như Hình 5, chúng tôi sử dụng phương pháp chọn thuộc tính CfsSubsetEval (CSE) và Correlation Attribute Eval (CA). CSE đo lường tầm quan trọng của các thuộc tính trên cơ sở khả năng dự đoán của các thuộc tính và mức độ dư thừa của nó. Các tập con (*subset*) có ít giao thoa nhưng tương quan cao với lớp mục tiêu là được ưu tiên. Xếp hạng của 3 thuộc tính đầu tiên bao gồm: 1. *burst_time*, 2. *burst_length*, 3. *void_start*. Hình 6 chỉ ra thứ hạng các thuộc tính được tìm thấy bởi phương pháp CSE.

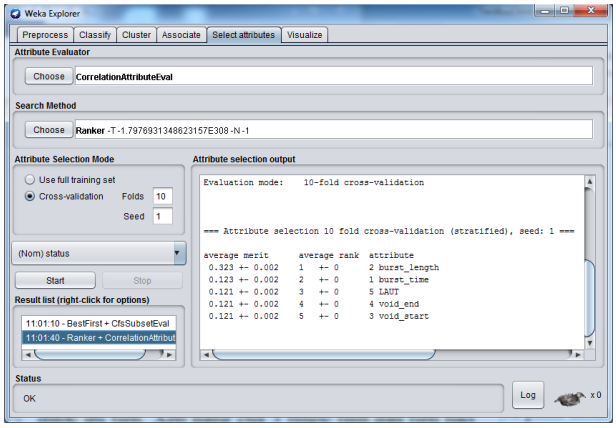


Hình 5. Phân bố các lớp *head_overlap*, *LAUT_overlap* và *tail_overlap* trong dữ liệu lập lịch không thành công

CA đánh giá các thuộc tính liên quan đến lớp đích. Phương pháp tương quan Pearson được sử dụng để đo mối tương quan giữa từng thuộc tính và thuộc tính lớp đích. Nó xem xét các thuộc tính trên cơ sở giá trị và mỗi giá trị đóng vai trò như một chỉ báo. Bằng cách kết hợp cách đánh giá thuộc tính này với phương pháp xếp hạng (Ranker) tìm kiếm, xếp hạng của ba thuộc tính đầu tiên bao gồm: 1. *burst_length*, 2. *burst_time*, 3. *LAUT*. Hình 7 chỉ ra thứ hạng các thuộc tính được tìm thấy bởi phương pháp CA.



Hình 6. Kết quả xếp hạng các thuộc tính với CSE



Hình 7. Kết quả xếp hạng các thuộc tính với CA

Kết quả phân tích dữ liệu lập lịch trong Hình 5 cho thấy rằng chồng lấp đầu (*head_overlap*) và chồng lấp LAUT (*LAUT_overlap*) là nguyên nhân chính gây nên mất chùm (chiếm hơn 90%), trong đó hai thuộc tính thời gian đến (*burst_time*) và độ dài chùm (*burst_length*) có tác động chính đến việc lập lịch không thành công (Hình 6 và 7). Đây chính là cơ sở của đề xuất sau đây nhằm nâng cao hiệu năng lập lịch theo nghĩa tăng tỉ lệ lập lịch thành công, giảm mất mát dữ liệu và tăng mức sử dụng băng thông tại nút lõi.

IV NÂNG CAO HIỆU QUẢ LẬP LỊCH DỰA TRÊN FDL

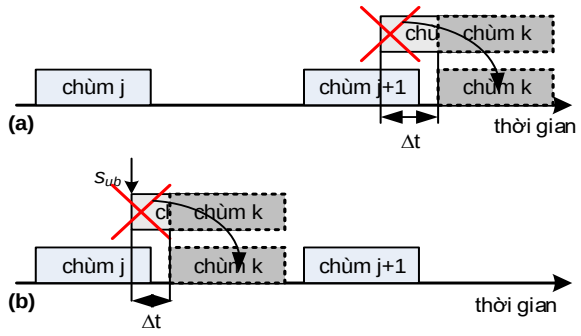
IV.1 Xử lý tranh chấp tài nguyên tại nút lõi

Khi một chùm đến không thể lập lịch được, các giải pháp xử lý có thể gồm: chuyển đổi bước sóng [10] sử dụng đường trễ FDL (*fiber delay link*) [11] hay định tuyến lệch hướng đối với chùm tranh chấp [12]. Chuyển đổi bước sóng là hoạt động chuyển đổi bước sóng của chùm đến sang một bước sóng mới của một kênh ra khả dụng có thể lập lịch được chùm này. Trong mô hình mạng OBS được xem xét trong bài báo này, chuyển đổi bước sóng được giả thuyết là đầy đủ, nên phương pháp này không cần xem xét nữa. Định tuyến lệch hướng là định tuyến lại chùm tranh chấp ra một cổng ra mới (so với cổng ra dự kiến ban đầu) và di chuyển trên một hành trình mới để

đến đích. Phương pháp định tuyến lệch hướng là ít tốn kém (vì không cần trang cấp thêm thiết bị) nhưng đòi hỏi các tính toán phức tạp và làm tăng xác suất tắc nghẽn do phát sinh các tuyến mới ngoài dự kiến, đặc biệt khi khi tải mạng cao. Giải pháp đường trễ FDL được chúng tôi lựa chọn bởi vì dễ cài đặt và phù hợp với mục tiêu là làm trễ thời gian đến của chùm nhằm tăng cơ hội lập lịch thành công cho nó.

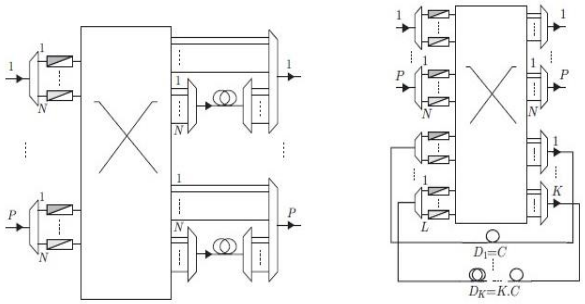
IV.2 Mô hình điều khiển lập lịch dựa trên FDL

Dựa trên kết quả phân tích ở Mục III.3, đa số các trường hợp lập lịch không thành công là do chồng lấp đầu hay chồng lấp LAUT. Thêm vào đó, thuộc tính có tác động chính đến việc lập lịch không thành công này là độ dài chùm (*burst_length*) và thời gian đến của chùm (*burst_time*). Tại nút lõi, việc thay đổi độ dài chùm chỉ có thể được thực hiện bằng cách phân đoạn chùm [13], theo đó phần chồng lấp của chùm tranh chấp sẽ bị loại bỏ và một yêu cầu truyền lại phần này sẽ được gửi trở lại nút nguồn. Tuy nhiên phương pháp phân đoạn chùm yêu cầu nhiều xử lý phức tạp hơn và gây lãng phí băng thông do gói điều khiển đặt trước tài nguyên luôn nhiều hơn so với nhu cầu băng thông cần sử dụng; thứ tự dữ liệu đến nút biên ra là không được đảm bảo. Bài viết này không xem xét đến việc thay đổi độ dài chùm mà chỉ tập trung vào làm trễ thời gian đến của chùm tại cổng ra. Như mô tả ở Hình 8, nếu thời gian đến được làm trễ một khoảng vừa đủ Δt, việc lập lịch sẽ thành công.



Hình 8. Chồng lấp LAUT (a) hay chồng lấp đầu (b) đều có thể được khắc phục nhờ thay đổi thời gian chùm đến

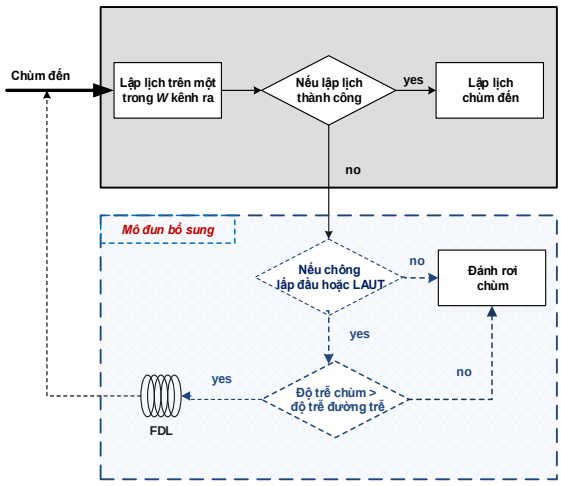
Có 2 kiến trúc đường trễ FDL đã được đề xuất trong mạng OBS [14]: truyền thẳng (*feed-forward*) hay hồi quy (*feed-back*). Cụ thể, như mô tả trong Hình 9a các FDL được đặt tại các cổng ra và chỉ được sử dụng bởi các chùm đến cổng ra đó, trong khi các FDL hồi quy được đặt tại nút lõi và các chùm đến trên cổng vào bất kỳ đều có thể sử dụng nếu chúng khả dụng (Hình 9b). Trong bài báo này, chúng tôi sử dụng kiểu đường trễ truyền thẳng do tính đơn giản về cài đặt và điều khiển của chúng [14].



(a) FDL truyền thẳng (b) FDL hồi quy

Hình 9. Hai mô hình đường trễ chính trong mạng OBS

Việc điều khiển lập lịch dựa trên đường trễ FDL được đề xuất như trong Hình 10, trong đó khi một chùm đến không thể lập lịch được, việc kiểm tra chồng lấp đầu và chồng lấp LAUT sẽ được xem xét (phần hình chữ nhật nét đứt). Nếu có chồng lấp đầu hay chồng lấp LAUT, chùm sẽ được đưa vào đường trễ để làm thay đổi (làm trễ) thời gian đến của nó. Trong trường hợp chồng lấp đuôi, chùm sẽ bị loại bỏ (*dropped*).



Hình 10. Mô hình điều khiển lập lịch tại nút lõi OBS

Độ dài đường trễ FDL cũng có những tác động đáng kể đến kích thước của nút chuyển mạch và tăng thêm độ trễ cho các chùm. Độ dài đường trễ cần được trang bị phụ thuộc vào thời gian cần làm trễ cho các chùm, mà cụ thể ở đây là Δt của khoảng chồng lấp đầu hay chồng lấp LAUT. Tuy nhiên, đường trễ cũng cần phải đủ dài để chứa trọn vẹn một chùm có kích thước lớn nhất; đó cũng chính là ngưỡng độ dài tập hợp chùm trong mạng OBS được xem xét. Các đường trễ được sử dụng trong bài viết này có độ dài bằng ngưỡng độ dài tập hợp chùm.

V. MÔ PHỎNG VÀ PHÂN TÍCH KẾT QUẢ

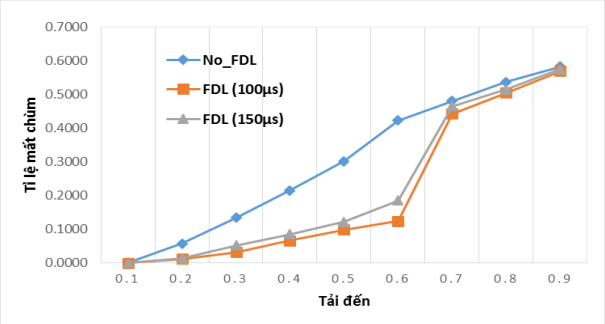
Với các tham số cài đặt như trong Mục III.2, chúng tôi tiến hành mô phỏng để so sánh tỉ lệ mất chùm (được tính bằng *số chùm bị loại bỏ / tổng số chùm đến tại cổng ra*) của 3 trường hợp: không sử dụng đường trễ và có sử dụng đường trễ với giá trị đường trễ lần lượt là 1 và 1,5 lần

ngưỡng độ dài tập hợp chùm. Chúng tôi chỉ cài đặt trường hợp làm trễ 1 lần với FDL đối với các chùm không thành công ở lần lập lịch đầu tiên.

V.1 Phân tích tỉ lệ mất chùm khi sử dụng FDL

Chúng tôi tiến hành so sánh tỉ lệ mất chùm trong 3 trường hợp không sử dụng đường trễ (No_FDL), sử dụng đường trễ với giá trị đường trễ là ngưỡng độ dài tập hợp chùm (FDL (100 µs)) và bằng 1,5 lần ngưỡng độ dài tập hợp chùm (FDL (150 µs)).

Hình 11 cho thấy rằng việc sử dụng đường trễ FDL cho tỉ lệ mất chùm trong khoảng tải chuẩn hóa từ 0.1 đến 0.6 giảm nhiều nhất khoảng 60% so với không sử dụng đường trễ. Nhưng khi tải tăng từ 0.7 đến 0.9 thì tỉ lệ mất giảm rất ít (khoảng 4%). Điều này là do khi tải cao, với số lượng chùm đến nhiều thì các chùm bị làm trễ tiếp tục bị chồng lấp với các chùm đã được lập lịch trước đó, nên bị loại bỏ.

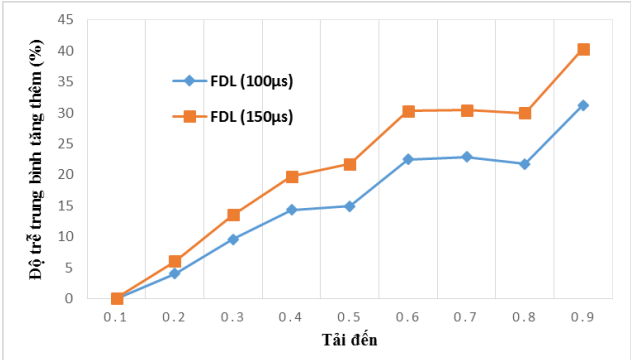


Hình 11. So sánh tỉ lệ mất chùm khi có và không sử dụng đường trễ FDL

FDL với độ trễ 100 µs cho tỉ lệ mất chùm thấp hơn so FDL với độ trễ 150 µs. Nguyên nhân là do khi tăng độ dài đường trễ thì các trường hợp chồng lấp LAUT (như được chỉ ra ở Hình 8a) sẽ không bị tác động, nhưng các trường hợp chồng lấp đầu (như được chỉ ra ở Hình 8b) sẽ có có khả năng trở thành chồng lấp đuôi với chùm đã được lập lịch trước đó. Trong trường hợp này, chùm tiếp tục bị loại bỏ. Do đó việc chọn độ dài đường trễ phù hợp sẽ có ảnh hưởng đáng kể đến hiệu quả lập lịch.

V.2 Phân tích độ trễ khi sử dụng FDL

Như thể hiện trong Hình 12 tỉ lệ độ trễ trung bình của các chùm tăng thêm khi sử dụng đường trễ FDL. Chúng ta nhận thấy rằng khi độ dài FDL càng lớn thì độ trễ trung bình càng cao hơn. Cụ thể ở đây là khi sử dụng FDL với độ trễ 150µs thì độ trễ chùm tăng thêm trung bình khoảng 21% so với gần 16% khi sử dụng FDL với độ trễ 100 µs. Hình 12 cũng cho chúng ta thấy rằng khi tải cao thì độ tăng thêm do sử dụng đường trễ cũng tăng cao. Điều này là do khi tải cao thì sẽ có nhiều chùm không thành công trong lần lập lịch đầu nên phải sử dụng đường trễ để thử lại lập lịch ở lần tiếp theo; kết quả là tỉ lệ độ trễ trung bình của các chùm tăng lên.



Hình 12. Tỉ lệ phần trăm độ trễ trung bình tăng thêm khi sử dụng đường trễ FDL

Tóm lại, việc sử dụng đường trễ đối với trường hợp chồng lấp đầu và chồng lấp LAUT rõ ràng đã mang lại hiệu quả về giảm tỉ lệ mất chùm như được chỉ ra ở Hình 11. Tuy nhiên việc sử dụng đường trễ FDL cũng làm gia tăng độ trễ tăng thêm cho các chùm không thành công ở lần lập lịch đầu tiên. Việc sử dụng đường trễ FDL trong trường hợp tải cao thì sẽ làm tăng thêm độ trễ lớn nhưng không làm giảm đáng kể tỉ lệ mất chùm. Do đó, việc sử dụng đường trễ chỉ phù hợp ở các mức tải thấp, mà cụ thể nên dưới mức tải chuẩn hóa 0.6.

VI KẾT LUẬN

Bài báo đã đề xuất một hướng tiếp cận phân tích nguyên nhân mất chùm dựa trên dữ liệu trạng thái lập lịch. Để làm được điều này, chúng tôi đã mô phỏng và trích xuất dữ liệu trạng thái lập lịch tại các nút lõi của một mạng OBS với hình thái mạng NSFNET. Dữ liệu trạng thái lập lịch sau đó được phân tích bằng ứng dụng Weka. Kết quả phân tích cho thấy rằng chồng lấp đầu và chồng lấp LAUT là các nguyên nhân chính gây nên mất chùm. Từ đó, chúng tôi đề xuất giải pháp sử dụng đường trễ FDL để thay đổi thời gian đến của chùm không thể lập lịch được trong lần đầu tiên. Kết quả mô phỏng đã chỉ ra rằng đường trễ FDL đã giúp giảm tỉ lệ mất chùm, nhưng làm tăng độ trễ trung bình của chùm. Kết quả mô phỏng cũng khuyến cáo rằng việc làm trễ bằng FDL chỉ phù hợp với tải mạng chuẩn hóa trong khoảng [0.1, 0.6]. Khi tải mạng cao, việc làm trễ bằng FDL chỉ làm tăng thêm tắc nghẽn.

VII. TÀI LIỆU THAM KHẢO

[1] S. J. B. Yoo, "Optical packet and burst switching technologies for the future photonic internet," *J. Light. Technol.*, vol. 24, no. 12, pp. 4468–4492, 2006, doi: 10.1109/JLT.2006.886060.

[2] A. Zalesky, "To burst or circuit switch?," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 305–318, 2009, doi: 10.1109/TNET.2008.923718.

[3] N. H. Q. Vo Viet Minh Nhat, "Phân tích hiệu quả các giải thuật lập lịch trên mạng chuyển mạch chùm quang," *Tạp chí khoa học, Đại học Huế*, vol. 74A, no. 5, pp. 85–97, 2012.

[4] https://www.cs.waikato.ac.nz/ml/weka/

[5] J. S. Turner, "Terabit burst switching," *J. High Speed Networks*, vol. 8, no. 1, pp. 3–16, 1999, doi: 10.7936/K7FJ2F2T.

[6] J. Xu, C. Qiao, J. Li, and G. Xu, "Efficient channel

scheduling algorithms in optical burst switched networks," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, 2003, vol. 3, pp. 2268–2278, doi: 10.1109/INFCOM.2003.1209247.

[7] M. Iizuka, M. Sakuta, Y. Nishino, and I. Sasase, "A scheduling algorithm minimizing voids generated by arriving bursts in optical burst switched WDM network," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, 2002, vol. 3, no. May, pp. 2736–2740, doi: 10.1109/GLOCOM.2002.1189127.

[8] M. Nandi, A. K. Turuk, D. K. Puthal, and S. Dutta, "Best Fit Void Filling Algorithm in Optical Burst Switching Networks," in *2009 Second International Conference on Emerging Trends in Engineering & Technology*, 2009, pp. 609–614, doi: 10.1109/ICETET.2009.46.

[9] https://www.isi.edu/nsnam/ns/.

[10] F. Z. Khan, M. F. Hayat, T. Holynski, and M. J. Khan, "Towards dynamic wavelength grouping for QoS in optical burst-switched networks," in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2017, pp. 79–85, doi: 10.1109/TSP.2017.8075941.

[11] Y. Lee, "Dynamic burst length controlling algorithm-based loss differentiation in OBS networks through shared FDL buffers," *Photonic Netw. Commun.*, vol. 31, no. 1, pp. 36–47, Feb. 2016, doi: 10.1007/s11107-015-0527-x.

[12] P. Khumalo, B. Nleya, and A. Mutsvangwa, "A controllable deflection routing and wavelength assignment algorithm in OBS networks," *J. Opt.*, vol. 48, no. 4, pp. 539–548, 2019, doi: 10.1007/s12596-019-00578-2.

[13] V. M. Vokkarane and J. P. Jue, "Segmentation-based nonpreemptive channel scheduling algorithms for optical burst-switched networks," *J. Light. Technol.*, vol. 23, no. 10, pp. 3125–3137, Oct. 2005, doi: 10.1109/JLT.2005.856265.

[14] C. McArdle, D. Tafani, and L. P. Barry, "Analysis of a buffered optical switch with general interarrival times," *J. Networks*, vol. 6, no. 4, pp. 536–548, 2011, doi: 10.4304/jnw.6.4.536-548.



Phạm Trung Đức nhận học vị Thạc sĩ Khoa học Máy tính tại Đại học Khoa học, Đại học Huế năm 2012; Hiện đang là nghiên cứu sinh tại Đại học Khoa học, Đại học Huế.

Các lĩnh vực nghiên cứu: *Mạng OBS, QoS trong mạng OBS, điều khiển chấp nhận lập lịch*.

Email: phamtrungduc@hueuni.edu.vn



Võ Viết Minh Nhật nhận học hàm Phó Giáo sư năm 2016, học vị Tiến sỹ năm 2007 tại đại học UQAM, Canada; Hiện công tác tại ĐH Huế. Lĩnh vực nghiên cứu gồm: *Mạng OBS, mạng cảm biến tích hợp RFID, tính toán mềm (mạng nơ ron nhân tạo, giải thuật tiến hóa, di truyền).*

Email: vvmnhat@hueuni.edu.vn



Đặng Thanh Chương nhận học vị tiến sĩ về Đảm bảo toán học cho máy tính năm 2014 tại Viện Công nghệ thông tin, Viện HL KH&CN VN.

Lĩnh vực nghiên cứu gồm: *mạng OBS, lý thuyết hàng đợi*

Email: dtchuong@hueuni.edu.vn

IMPROVING THE SCHEDULING PERFORMANCE OF CORE OBS NODES BASED ON DATA ANALYSIS AND USING FDL

Abstract: The scheduling performance of the core node plays an important role in the overall communication performance of OBS networks. In order to improve the scheduling performance of the core node, many studies have been carried out combining scheduling techniques with wavelength conversion, deflection routing or using FDL. The scheduling performance at an output of a core node depends on many factors, including the arrival time of burst, the start and end time of voids and the latest available unscheduled time (LAUT) of each channel. It is the factors that have a major influence on scheduling performance. This paper therefore proposes a method for analyzing scheduling data to find the cause of scheduling failure. Based on the results of the analysis, the article shows that using FDL to avoid head- and LAUT overlaps is the most appropriate solution to increase the success scheduling rate, reduce the burst loss and thus increase the efficient use of bandwidth in OBS networks.