Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XI về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 09-10/8/2018
DOI: 10.15625/vap.2018.00018

ĐIỀU KHIỂN CHẤP NHẬN LẬP LỊCH DỰA TRÊN DỰ BÁO TỐC ĐỘ CHÙM ĐẾN TRONG MẠNG CHUYỂN MẠCH CHÙM QUANG

Phạm Trung Đức[1], Võ Viết Minh Nhật[2], Đặng Thanh Chương[1]
[1]Trường Đại học Khoa học, Đại học Huế
[2] Đại học Huế

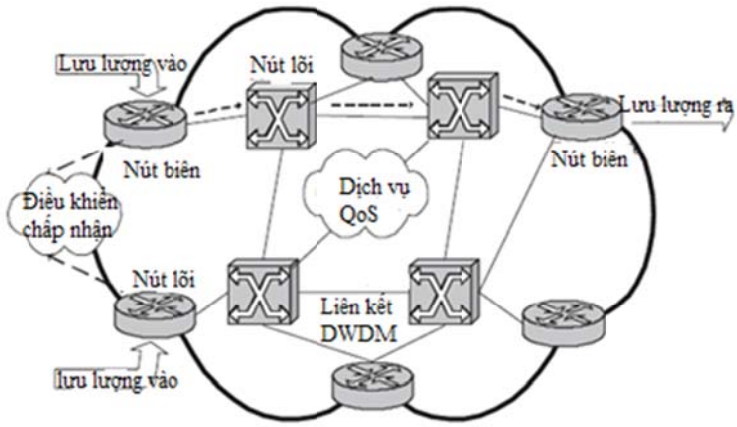
ptduc1008@gmail.com, vvmnhat@hueuni.edu.vn, dtchuong@hueuni.edu.vn

***TÓM TẮT*:** *Lập lịch là một trong những hoạt động quan trọng, có ảnh hưởng lớn đến hiệu năng của mạng OBS. Trong mạng OBS có hỗ trợ QoS, việc lập lịch cần phải xem xét với một số điều kiện ưu tiên cụ thể. Đã có một số công bố về điều khiển chấp nhận lập lịch, trong đó đa số xem xét trong các điều kiện tĩnh và lý tưởng. Bài viết sẽ nghiên cứu việc điều khiển chấp nhận lập lịch trong một môi trường động, trong đó việc dự đoán tốc độ chùm đến được sử dụng để điều khiển hiệu quả hoạt động lập lịch. Một số phương pháp dự đoán tốc độ chùm đến cũng được phân tích, so sánh và đánh giá.*

Từ khóa: *Mạng OBS, QoS, điều khiển chấp nhận lập lịch, dự đoán dựa trên tốc độ, hiệu quả lập lịch.*

I. GIỚI THIỆU

Chuyển mạch chùm quang (*Optical Burst Switching, OBS*) [1] là một giải pháp chuyển mạch gói quang khả thi cho mạng Internet thế hệ tiếp theo. Với những thành công gần đây của công nghệ ghép phân chia kênh bước sóng (*Wavelength Division Multiplexing, WDM*) [2], mỗi sợi quang đơn có thể ghép/tách từ 160 đến 320 kênh (bước sóng), trong đó tốc độ truyền trên mỗi bước sóng có thể đạt từ 10 Gb/s đến 40 Gb/s [3]; băng thông các sợi quang do đó có thể được khai thác một cách có hiệu quả.



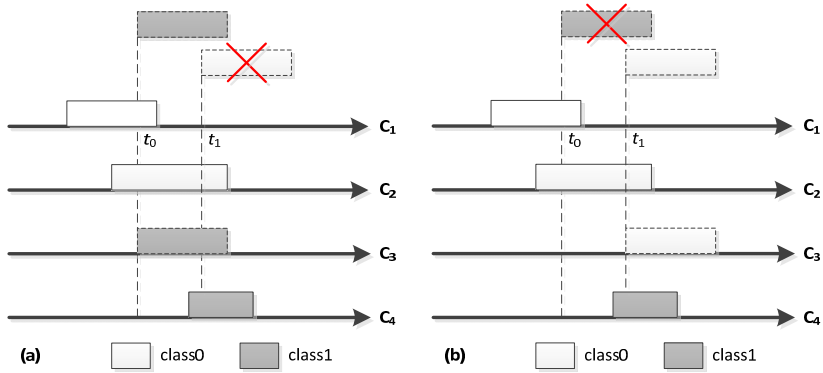
Hình 1. Kiến trúc tiêu biểu của mạng OBS

Kiến trúc một mạng chuyển mạch chùm quang (mạng OBS) có thể được mô tả như trong ***Hình 1***, trong đó có 2 loại nút mạng được phân biệt: nút biên và nút lõi. Nút biên, bao gồm nút biên vào và nút biên ra, được xem là giao diện giữa miền điện tử (các mạng truy cập) và miền quang (mạng OBS). Nút biên vào có nhiệm vụ nhận dữ liệu đến từ các mạng truy cập (chẳng hạn các gói tin IP), gộp chúng thành các chùm dữ liệu (*bursts*) và truyền các chùm này vào mạng lõi. Một gói điều khiển chùm (*Burst Control Packet, BCP*) được truyền đi trước một khoảng thời gian bù đắp (*offset-time*) nhằm đặt trước tài nguyên và cấu hình các chuyển mạch tại các nút trung gian (nút lõi); nhờ đó chùm dữ liệu của nó theo sau sẽ được chuyển mạch toàn quang tại tất cả các nút trung gian cho đến khi đạt đến nút biên ra mà không mất thời gian chờ đợi nào. Tại nút biên ra thực hiện một hoạt động ngược lại với nút biên vào được thực hiện nhằm khôi phục lại các dữ liệu (các gói IP) ban đầu.

Điều khiển chấp nhận lập lịch có thể được thực hiện tại nút biên vào và nút lõi; tuy nhiên nhờ có sử dụng các bộ đệm điện tử mà điều khiển lập lịch tại các nút biên vào thường hiệu quả hơn [4]. Trong đa số các kỹ thuật điều khiển lập lịch, những chùm dữ liệu có lớp QoS (ưu tiên) thấp sẽ có xác suất bị đánh rơi nhiều hơn để dành tài nguyên cho lớp QoS cao hơn khi có tranh chấp xảy ra.Việc lập lịch các chùm đến thông thường được thực hiện tuần tự, theo kiểu đến trước, phục vụ trước (*first come, first served*). Tuy nhiên khi có xét đến QoS, việc lập lịch thành công một chùm trước (thuộc lớp QoS thấp) có thể gây tắc nghẽn cho một chùm đến sau (thuộc lớp QoS cao hơn). Như được chỉ ra trong ví dụ ở ***Hình 2a***, nếu chùm QoS thấp đến tại thời điểm t_0 được lập lịch thì chùm QoS cao đến sau đó (tại thời điểm t_1) sẽ không thể lập lịch được do hết tài nguyên. Để ưu tiên tài nguyên cho các chùm QoS cao, chùm QoS thấp đến tại t_0

thường được chủ động đánh rơi. Do đó, lập lịch với điều khiển chấp nhận là cần thiết nhằm để dành nhiều tài nguyên hơn cho chùm QoS cao, trong khi hạn chế tài nguyên đối với các chùm QoS thấp.

Đã có một số kỹ thuật điều khiển lập lịch được đề xuất trong mạng OBS có hỗ trợ QoS, như kỹ thuật điều khiển tĩnh (*Static Admission Control, SAC*) [5], điều khiển động (*Dynamic Admission Control, DAC*) [5] và điều khiển dựa trên mức tải (*Load-Level Admission Control, LLAC*) [6].

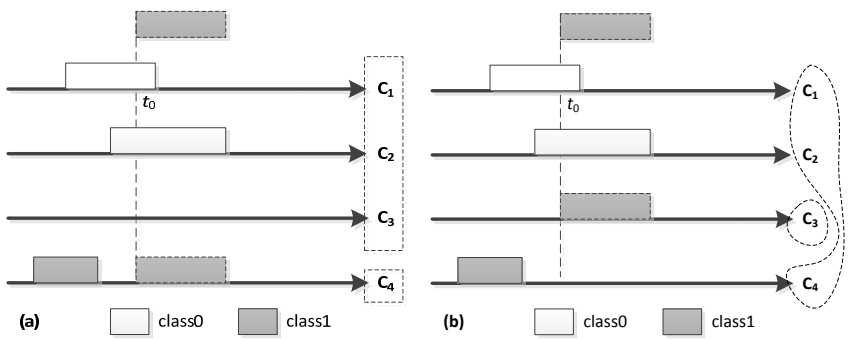


Hình 2. Một ví dụ về việc chủ động đánh rơi chùm QoS thấp để dành tài nguyên cho chùm QoS cao đến sau (b), so với kiểu lập lịch đến trước, phục vụ trước (a)

Tuy nhiên, các kỹ thuật này vẫn bộc lộ một số nhược điểm sau: đối với SAC và DAC, (1) tỉ lệ phân bổ các kênh (bước sóng) ra đối với các chùm QoS cao và QoS thấp là cố định; do đó (2) hiệu quả sử dụng băng thông thấp, khi mà lưu lượng chùm QoS cao đến thấp trong khi lưu lượng chùm QoS thấp đến cao; và (3) thông tin (về các chùm đang được lập lịch trên mỗi kênh) cần phải lưu trữ tại mỗi nút là lớn và cần nhiều không gian nhớ. Đối với LLAC, mặc dù thông tin cần lưu trữ đã được cải thiện hơn so với SAC và DAC nhưng (4) tỉ lệ tải được thiết lập cho các luồng chùm QoS cao và QoS thấp vẫn là cố định; hơn nữa (5) LLAC cho tỉ lệ chùm QoS thấp bị đánh rơi cao hơn so với DAC. Rõ ràng có một nhu cầu cần phân bổ tài nguyên linh hoạt hơn đối với các luồng chùm đến nhằm sử dụng hiệu quả băng thông. Bài viết này sẽ phân tích vấn đề điều khiển chấp nhận lập lịch dựa trên một số kỹ thuật dự đoán tốc độ chùm đến; từ đó chỉ ra rằng một giải pháp thoả hiệp trong điều khiển lập lịch sẽ mang lại hiệu quả sử dụng băng thông tốt hơn.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Đã có một số đề xuất về các kỹ thuật điều khiển lập lịch tại một cổng ra của một nút mạng OBS. Cụ thể, Moraes và CS trong [5] đã đề xuất hai cơ chế điều khiển chấp nhận lập lịch tĩnh và động. Xét tại một cổng ra có W kênh bước sóng khả dụng, kỹ thuật điều khiển tĩnh (kỹ thuật SAC) sẽ phân bổ W_0 ($W_0 < W$) bước sóng cho các chùm QoS cao và W_1 ($W_1 < W$) bước sóng cho các chùm QoS thấp, trong đó $W_0 > W_1$ và $W_0 + W_1 = W$. Như ví dụ được chỉ ra trong ***Hình 3a***, với $W = 4$, kỹ thuật SAC phân bổ $W_0 = 3$ kênh bước sóng (C_1, C_2, và C_3) cho các chùm QoS cao và $W_1 = 1$ bước sóng còn lại (C_4) cho các chùm QoS thấp; nên một chùm QoS thấp đến tại thời điểm t_0 sẽ được lập lịch vì kênh C_4 rỗi.

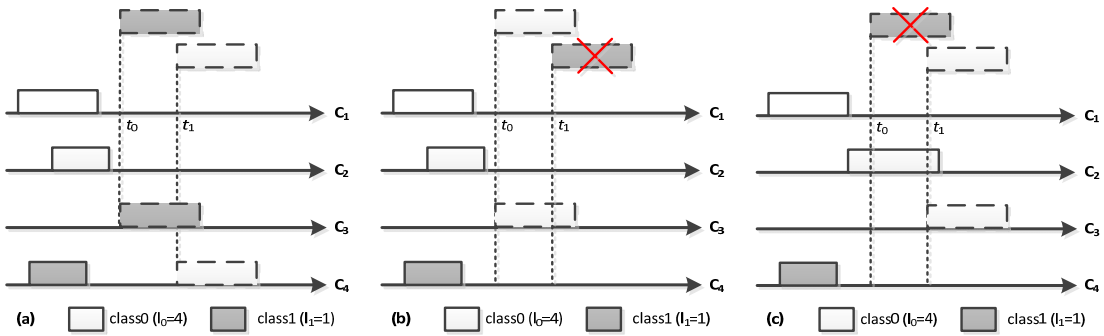


Hình 3. Ví dụ về điều khiển lập lịch của SAC (a) và DAC (b)

Ưu điểm của kỹ thuật SAC là đơn giản, đảm bảo dành riêng các kênh bước sóng cho các chùm QoS cao nên đảm bảo chất lượng dịch vụ. Tuy nhiên, sẽ xuất hiện một sự phân bố tải không đồng đều trên các kênh bước sóng nếu các chùm đến chỉ thuộc về một phía. Cụ thể, nhóm các kênh W_0 (hoặc W_1) sẽ quá tải, trong khi nhóm các kênh W_1 (hoặc W_0) là nhàn rỗi nếu các chùm đến chỉ thuộc về QoS cao (hoặc QoS thấp). Để linh hoạt hơn trong sử dụng các kênh bước sóng, Moraes và CS trong [5] đã đề xuất tiếp kỹ thuật phân phối động (kỹ thuật DAC), trong đó các kênh không được chỉ định cụ thể cho các lớp QoS mà chỉ kiểm soát số lượng kênh được phép sử dụng tối đa cho mỗi loại chùm QoS cao và QoS thấp. Như ví dụ được chỉ ra trong ***Hình 3b***, xét tại thời điểm đến (t_0) của một chùm đến, nếu số chồng lấp của chùm này với các chùm cùng loại (ω_0 và ω_1) mà nhỏ hơn lượng bước sóng được phân bổ ($\omega_0 < W_0$ và $\omega_1 < W_1$), việc lập lịch là được chấp nhận. Do đó một chùm QoS thấp đến tại thời điểm t_0 vẫn có thể được lập lịch lên kênh C_3 hoặc C_4 vì số chồng lấp đối với loại chùm này là $\omega_1 = 0$, trong khi $W_1 = 1$, nên $\omega_1 < W_1$.

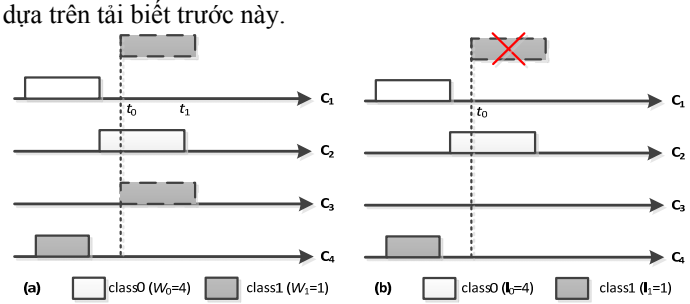
Ưu điểm của kỹ thuật DAC là phân bố tải đồng đều trên các kênh hơn so với SAC nhưng cũng như SAC, kỹ thuật DAC cũng gây lãng phí tài nguyên đối với các kênh dành cho các chùm QoS cao nếu mật độ các chùm này đến thấp, trong khi các chùm QoS thấp bị đánh rơi vì thiếu tài nguyên. Cả hai kỹ thuật SAC và DAC đều yêu cầu lưu lại thông tin trạng thái của tất cả chùm đang được lập lịch trên các kênh nên cần nhiều bộ nhớ cho việc lưu trữ. Một hạn chế khác của hai kỹ thuật SAC và DAC là chúng chỉ hiệu quả khi biết trước lưu lượng các luồng chùm (ưu tiên thấp và cao) đến và các lưu lượng này không biến đổi đáng kể trong một thời gian dài; nhưng điều này ngược với thực tế vì lưu lượng trên mạng luôn thay đổi.

Để tăng hiệu quả việc phân phối các kênh bước sóng ra và giảm không gian lưu trữ, các tác giả trong [6] đã đề xuất một kỹ thuật điều khiển dựa trên tải (kỹ thuật LLAC), trong đó chỉ có thông tin về mức tải của mỗi lớp và tổng số bước sóng bị chiếm là được lưu trữ. Các mức tải (l_0 và l_1) sẽ được quy định trước cho mỗi lớp QoS cao và thấp; nếu số chồng lấp của một chùm đến với bất kỳ loại chùm nào đã được lập lịch trên các kênh ra (ω) là nhỏ hơn mức tải ($\omega < l_0$ và $\omega < l_1$) chùm đến sẽ được lập lịch. Như ví dụ được chỉ ra trong ***Hình 4a***, với hai mức tải $l_0 = 4$ và $l_1 = 1$ được thiết lập trước đối với 2 loại chùm QoS cao và thấp, khi có một chùm QoS thấp đến tại thời điểm t_0, nó được lập lịch vì $\omega < l_1$ ($\omega = 0$). Tại thời điểm t_1, một chùm QoS cao đến và nó được lập lịch vì $\omega < l_0$ (lúc này $\omega = 1$). Tuy nhiên trong ***Hình 4b***, với một chùm QoS cao đến tại thời điểm t_0, nó được lập lịch vì $\omega < l_0$ ($\omega = 0$). Tại thời điểm t_1, một chùm QoS thấp đến và nó không được lập lịch vì $\omega = l_1$ (lúc này $\omega = 1$). Tương tự với ***Hình 4c***, một chùm QoS thấp đến tại thời điểm t_0 sẽ không được lập lịch vì $\omega = l_1$ ($\omega = 1$ và $l_1 = 1$), trong khi một chùm QoS cao đến tại thời điểm t_1 vẫn được lập lịch vì $\omega < l_0$ ($\omega = 1$ và $l_0 = 4$).



Hình 4. Các ví dụ về cách thức hoạt động của kỹ thuật LLAC

Một nhược điểm của kỹ thuật LLAC là nó có xu hướng đánh rơi các chùm QoS thấp nhiều hơn so với kỹ thuật DAC. Như ví dụ trong ***Hình 5***, kỹ thuật DAC sẽ không đánh rơi chùm ưu tiên thấp khi nó đến tại thời điểm t_0 (***Hình 5a***); trong khi kỹ thuật LLAC sẽ đánh rơi nó vì $\omega = l_1$ ($\omega = 1$ và $l_1 = 1$). Ngoài ra, một nhược điểm khác của LLAC là không đưa ra bất cứ cách nhận biết nào về tải đến trong khi khi ý tưởng của đề xuất là giả định biết trước được tải lưu lượng sau đó phân bổ kênh dựa trên tải biết trước này.



Hình 5. Một so sánh về hiệu quả lập lịch giữa kỹ thuật DAC (a) và LLAC (b)

Tóm lại, cả 3 kỹ thuật SAC, DAC và LLAC đều không xét việc phân phối băng thông trong môi trường mạng mà ở đó lưu lượng chùm đến cổng ra thay đổi liên tục. Hơn nữa, việc cần lưu nhiều thông tin về trạng thái các kênh ra cũng đòi hỏi cần nhiều vùng nhớ được triển khai tại các nút để có thể áp dụng các kỹ thuật này. Trong phần tiếp theo, chúng tôi sẽ phân tích ảnh hưởng của việc phân phối kênh ra đối với tốc độ chùm đến và đề xuất áp dụng một số phương pháp ước tính tốc độ đến nhằm phân phối các kênh ra một cách hiệu quả hơn.

III. ĐIỀU KHIỂN LẬP LỊCH DỰA TRÊN TỐC ĐỘ CHÙM ĐẾN

Xét một cổng ra của một nút mạng OBS mà tại đó có các luồng chùm đến được giả thiết thuộc về một trong hai lớp QoS cao và QoS thấp. Giả sử cổng ra có W kênh (bước sóng) khả dụng và chùm QoS cao đến có thể được lập lịch trên bất kỳ W kênh bước sóng, vấn đề là cần tính toán lưu lượng các chùm (QoS cao và QoS thấp) đến để thực hiện phân phối băng thông một cách hiệu quả nhất cho các chùm QoS thấp.

Gọi λ_0 là tốc độ đến của các chùm QoS cao và λ_1 là tốc độ đến của các chùm QoS thấp, số kênh cấp phát cho các chùm ưu tiên thấp (W_1, $W_1 < W$) có thể được tính tỉ lệ với tốc độ đến của hai loại chùm này như Công thức 1. Lưu ý rằng, các chùm ưu tiên cao được sử dụng tất cả các kênh, có nghĩa là $W_0 = W$.

$$W_1 = \left\rceil W_0 \times \frac{\lambda_1}{\lambda_0 + \lambda_1} \right\lceil \quad (1)$$

Có nhiều cách tiếp cận khác nhau để dự đoán tốc độ chùm đến. Cách đơn giản nhất là đếm số chùm đến (n_i, trong đó $i = 0$ với chùm QoS cao và $i = 1$ với chùm QoS thấp) trong một khung cửa sổ quan sát (T_w) sau từng khoảng thời gian định kỳ (ΔT, $\Delta T \geq T_w$). Tốc độ chùm đến trong tương lai khi đó được dự đoán bởi Công thức 2.

$$\lambda_i = \delta \times \frac{n_i}{T_w} \quad (2)$$

trong đó δ là một tham số điều khiển và thường được chọn gần bằng 1 với một mức lỗi ước tính chấp nhận được nào đó; trong điều kiện lý tưởng, $\delta = 1$.

Tuy nhiên, cách tiếp dựa trên cửa sổ quan sát cần phải xác định kích thước T_w đủ lớn sao cho có ít nhất có một vài chùm đến trong khoảng thời gian này. Tuy nhiên trong trường hợp lưu lượng chùm đến thay đổi với biên độ lớn thì việc chọn kích thước cửa sổ quan sát phù hợp trở nên khó khăn hơn. Một cách tiếp cận khác được đề xuất là đếm một lượng được cho các chùm đến rồi chia cho khoảng thời gian đến của chúng (T'_w). Tốc độ chùm đến do đó được ước tính bằng Công thức 3:

$$\lambda_i = \delta \times \frac{n_i^{'}}{T_w^{'}} \quad (3)$$

Một trường hợp đặc biệt của mô hình đựa trên đếm chùm là việc cấp phát lại bước sóng được thực hiện mỗi khi có một chùm QoS thấp đến theo Công thức 4:

$$\lambda_i = \delta \times \frac{1}{T_w^{"}} \quad (4)$$

trong đó T''_w là khoảng thời gian 2 chùm QoS thấp đến liên tiếp.

Các mô hình dự đoán nêu trên về bản chất hoặc dựa trên tốc độ trung bình các chùm đến trong một cửa sổ thời gian (Công thức 2 và 3) hoặc dựa trên tốc độ tức thời của chùm QoS thấp (Công thức 4). Tuy nhiên, các cách tiếp cận này không phản ánh được quá trình đến trung bình trong quá khứ và sự tăng giảm đột biến gần đây nhất của luồng chùm đến. Để kết hợp cả hai yếu tố này, mô hình dự đoán dựa trên phương pháp TW-EWMA [7] xác định tốc độ đến của chùm ưu tiên cao ($i = 0$) và ưu tiên thấp ($i = 1$) bằng Công thức 5

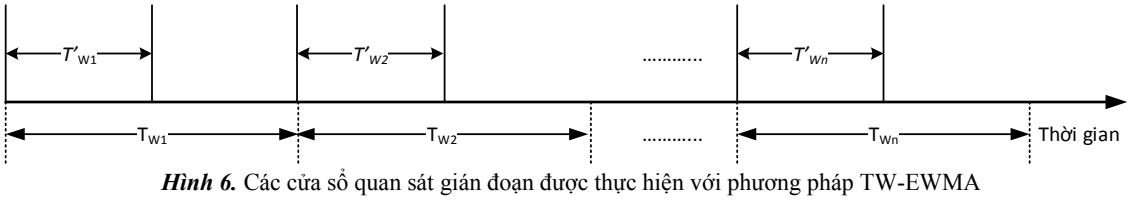
$$\lambda_i = (1 - \alpha_i) \times \lambda_i^{avg} + \alpha_i \times \lambda_i^{cur} \quad (5)$$

trong đó λ_i^{avg} là tốc độ đến trung bình trong quá khứ, λ_i^{cur} là tốc độ đến hiện thời; $(1 - \alpha_i)$ và α_i là trọng số của λ_i^{avg} và λ_i^{cur}. Trong [7], trọng số này được chọn là $\alpha = 0.3$. Tuy nhiên, theo đề xuất trong [8], hệ số này có thể được điều chỉnh linh hoạt dựa trên tốc độ trung bình trong quá khứ và tốc độ hiện thời như Công thức 6.

$$\frac{1 - \alpha_i}{\alpha_i} = \frac{\lambda^{avg}}{\lambda^{cur}} \Rightarrow \alpha_i = \frac{\lambda^{cur}}{\lambda^{cur} + \lambda^{avg}} \quad (6)$$

Thuật toán điều khiển chấp nhận lập lịch dựa trên dự đoán tốc độ đến một cách thích nghi do đó có tên gọi là ARP-SAC (*Adaptive Rate Predition SAC*).

Các cửa sổ quan sát có thể là liên tục; tuy nhiên, để giảm chi phí tính toán, việc quan sát có thể được thực hiện với các cửa sổ gián đoạn như được mô tả trong Hình 6. Một lưu ý rằng kích thước cửa sổ quan sát có một tác động đáng kể đến tính chính xác của việc dự đoán và chi phí tính toán. Nếu cửa sổ lớn thì việc dự đoán sẽ chính xác hơn, nhưng chi phí tính toán sẽ tăng cao; trong khi nếu cửa sổ quan sát bé thì tính chính xác của dự đoán sẽ giảm, nhưng chi phí tính toán sẽ thấp [7]. Một thoả hiệp giữa tính chính xác và khối lượng tính toán do đó cần được tính đến. Trong bài báo này, chúng tôi chọn cửa sổ quan sát bằng một phần hai chu kỳ thực hiện dự đoán: $T_W^{'} = \frac{T_W}{2}$.



Hình 6. Các cửa sổ quan sát gián đoạn được thực hiện với phương pháp TW-EWMA

Việc điều khiển chấp nhận lập lịch một chùm đến được thực hiện dựa trên một tham số ω được định nghĩa là mức độ chồng lấp của chùm đến đối với các chùm đã được lập lịch trên các kênh ra. Cụ thể, một chùm đến (*ub*, *unscheduled burst*) sẽ được xem xét lập lịch (bằng cách gọi một giải thuật lập lịch, chẳng hạn BFVF [9] là giải thuật lập lịch lấp đầy khoảng trống tốt nhất hiện nay) nếu mức độ chồng lấp (ω) nhỏ hơn số kênh (băng thông) được cấp phát (W_i).

Sau đây là mô tả chi tiết về thuật toán ARP-SAC và hai hàm ***omega***(ub_i, W) và ***BFVF***(ub,W).

Thuật toán ARP-SAC

Đầu vào: - Tập các chùm đến $B_{ub} = \{ub_i | i = 0, 1\}$, với $ub_i = \{s_{ub}, e_{ub}, prio_{ub}\}$, trong đó s_{ub} và e_{ub} là thời gian đến và kết thúc, $prio_{ub} = \{0, 1\}$ xác định ub là QoS cao (0) hay thấp (1).

Đầu ra: - Tập các chùm QoS cao được lập lịch, S_0, và bị đánh rơi, D_0;
- Tập các chùm QoS thấp được lập lịch, S_1, và bị đánh rơi, D_1;

Phương pháp

1 $t^{start} := 0$; // khởi gán điểm bắt đầu của cửa sổ quan sát
2 $T_w := 5000$; // khởi gán kích thước cửa sổ quan sát (bằng ½ của 10.000 μs)
3 $\lambda_0^{avg} := 0$; $\lambda_1^{avg} := 0$; // khởi gán tốc độ trung bình của các lớp ưu tiên
4 $pause := 0$; // chỉ đếm khi $pause = 0$ và không đếm khi $pause = 1$
5 $ch := -1$; // kênh được chọn để lập lịch, $ch = -1$ khi không chọn được kênh nào
6 **while** (B_{ub} != ∅) **do**
7 ub := chùm đầu tiên từ tập B_{ub};
8 $B_{ub} := B_{ub} \setminus \{ub\}$; // loại chùm đến đã được xem xét lập lịch
9 **if** ($s_{ub} - t^{start} < T_w$) **then**
10 **if** ($prio_{ub} = 0$) **then**
11 $n_0 := n_0 + 1$; // đếm số chùm QoS cao.
12 **else**
13 $n_1 := n_1 + 1$; // đếm số chùm QoS thấp
14 **end if**
15 **else**
16 **if** ($pause$ = false) **then** // trường hợp trong vùng cửa sổ quan sát
17 $\lambda_0^{cur} := \frac{n_0}{T_w}$; $\lambda_1^{cur} := \frac{n_1}{T_w}$; // tính tốc độ đến hiện thời
18 $\alpha_0 := \frac{\lambda_0^{cur}}{\lambda_0^{cur} + \lambda_0^{avg}}$; $\alpha_1 := \frac{\lambda_1^{cur}}{\lambda_1^{cur} + \lambda_1^{avg}}$; // điều chỉnh trọng số theo tốc độ đến
19 $\lambda_0^{avg} := (1 - \alpha_0) \times \lambda_0^{avg} + \alpha_0 \times \lambda_0^{cur}$; // tính lại tốc độ đến trung bình
20 $\lambda_1^{avg} := (1 - \alpha_1) \times \lambda_1^{avg} + \alpha_1 \times \lambda_1^{cur}$; // tính lại tốc độ đến trung bình
21 $W_1 := \left\lceil W_0 \times \frac{\lambda_1^{avg}}{\lambda_0^{avg} + \lambda_1^{avg}} \right\rceil$; // điều chỉnh lượng băng thông cấp phát W_1
22 $n_0 := 0$; $n_1 := 0$;
23 $pause := 1$;
24 **end if**
25 **end if**
26 **if** ($s_{ub} - t^{start} > 2 \times w$) **then** // trường hợp ngoài vùng cửa sổ quan sát
27 $pause := 0$;
28 $t^{start} := s_{ub}$;
29 **end if**
30 **if** (($prio_{ub} = 0$) ∧ (***omega***(ub, W) < W_0)) **then** // lập lịch chùm QoS cao đến
31 ch:= ***BFVF***(ub, W);
32 **if** (ch != -1) **then**
33 $S_0 := S_0 \cup \{ub\}$;
34 **else**
35 $D_0 := D_0 \cup \{ub\}$;
36 **end if**
37 **else if** ($prio_{ub} = 1$) ∧ (***omega***(ub, W) < W_1) **then** // lập lịch chùm QoS thấp đến
38 $ch := $ ***BFVF***(ub, W);
39 **end if**

```
40                  if (ch != -1) then
41                      S1 := S1 ∪ {ub};
42                  else
43                      D1 := D1 ∪ {ub};
44                  end if
45              end if
46      end while
```

Hàm ***omega***(*ub*, *W*) là thuật toán xác định số chồng lấp (*count*) của chùm đến (*ub*) với các chùm cùng lớp QoS đã được lập lịch. Do thực tế tốc độ đến và tốc độ chuyển tiếp các chùm tại cổng ra là rất nhanh, nên chỉ tối đa hai chùm được lập lịch sau cùng trên mỗi kênh ra là được xem xét ($|SB_k| \le 2$). Độ phức tạp của hàm ***omega***(*ub*, *W*) do đó là O(*W*).

Hàm *omega*(*ub*, *W*)
Đầu vào: - *ub*; // chùm đến $ub = \{s_{ub}, e_{ub}, prio_{ub}\}$
- *W*; // các kênh bước sóng
Đầu ra: - *count.* // số chồng lấp
Phương pháp

```
1   count := 0;
2   for each k ∈ W do
3       e0,k := 0;
4       SBk := tập các chùm đã lập lịch trên kênh k;
5       for each sbi,k ∈ SBk do
6           if (prioub = prioi,k) ∧
              (((sub ≥ si,k) ∧ (sub ≤ ei,k)) ∨ ((eub ≥ si,k) ∧ (eub ≤ ei,k)) ∨ ((sub ≤ si,k) ∧ (eub ≥ ei,k))) then
7               count++;
8           end if
9       end for
10  end for
11  return count;
```

Hàm ***BFVF***(*ub*,*W*) là một thuật toán lập lịch lấp đầy khoảng trống tốt nhất hiện nay được tham khảo từ [9], có nhiệm vụ lập lịch một chùm đến *ub* lên một trong *W* kênh khả dụng tại cổng ra: nếu lập lịch thành công, ***BFVF*** trả về kênh được chọn để lập lịch cho *ub*; nếu không thành công, ***BFVF*** trả về -1. Độ phức tạp của hàm ***BFVF*** phụ thuộc vào 2 vòng lặp for each (từ dòng 3 đến 12 và từ dòng 6 đến 11) nên có độ phức tạp bằng O($W \times |SB_k|$), trong đó $|SB_k|$ là số chùm đã lập lịch trên kênh *k*. Như đã giải thích ở trên, $|SB_k| \le 2$, nên là độ phức tạp của hàm ***BFVF***(*ub*,*W*) là O(*W*).

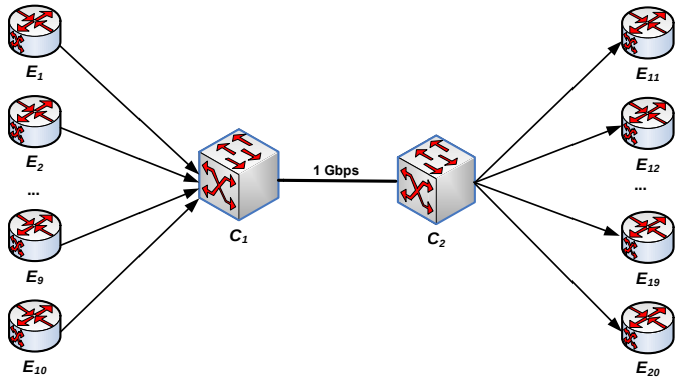
Hàm *BFVF*(*ub*, *W*)
Đầu vào: - $ub(s_{ub}, e_{ub})$ chùm đến;
- *W* kênh bước sóng
Đầu ra: - Kênh được lựa chọn *best_channel.*
Phương pháp

```
1   best_utilisation := ∞;
2   best_channel := -1;
3   for each k ∈ W do
4       e0,k := 0;
5       SBk :=tập các chùm đã lập lịch trên kênh k;
6       for each sbi,k ∈ SBk do
7           if (((sub ≥ ei,k) ∧ (si+1,k ≥ eub)) ∧ ((si+1,k − ei,k) < best_utilisation))) then
8               best_utilisation := si+1,k − ei,k;
9               best_channel := k;
10          end if
11      end for
12  end for
13  return best_channel;
```

Độ phức tạp của ARP-SAC phụ thuộc vào hàm ***omega***(*ub*, *W*) và hàm ***BFVF***(*ub*, *W*). Do hai hàm này thực hiện rời rạc (không lồng nhau) nên độ phức tạp của thuật toán ARP-SAC là O(*N*×*W*), trong đó *N* là số chùm đến trong tập các chùm chưa được lập lịch B_{ub} và *W* là tổng số kênh ra. Xem xét việc lập lịch của từng chùm đến (*N* = 1), độ phức tạp của ARP-SAC giảm còn O(*W*).

IV. MÔ PHỎNG VÀ PHÂN TÍCH KẾT QUẢ

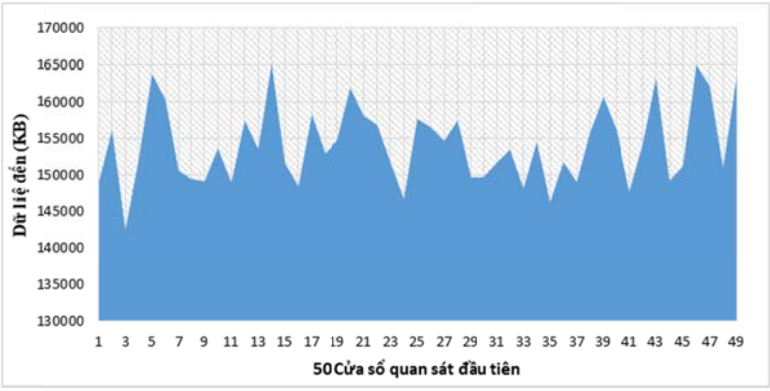
Các mô hình điều khiển lập lịch trước đây (SAC, DAC và LLAC) và mô hình được đề xuất dựa trên dự đoán tốc độ chùm đến thích nghi ARP-SAC được cài đặt máy tính có CPU 2,4 GHz Intel Core 2,2G RAM.



Hình 7. Mô hình mạng mô phỏng Dumbbell

Mạng mô phỏng có hình thái Dumbbell (Hình 7), trong đó các chùm đến tại nút C_1 có phân phối Poisson, được sinh ra từ phần mềm mô phỏng NS2 [10] với gói hỗ trợ obs 0.9a. Xét 2 loại luồng chùm theo tải chuẩn hóa 0.9 (QoS cao và QoS thấp) đến với tốc độ lần lượt r_0 và r_1, với ba trường hợp tỷ lệ r_0:r_1 được xem xét là 3:7, 5:5 và 7:3. Thời gian mô phỏng 1 s (giây); số bước sóng trên liên kết ở cổng ra (*W*) là 12, cửa sổ quan sát trong ARP-SAC là 10 ms (mili giây) được lấy như trong [7], với giá trị cửa sổ ước tính này chúng tôi có thể quan sát được sự thay đổi lưu lượng đến (Như thể hiện trong Hình 8). Tải chuẩn hóa (β) được tính theo công thức:

β = (tốc độ chùm đến × độ dài chùm trung bình) / băng thông kênh ra.



Hình 8. Sự thay đổi dữ liệu đến trong 50 cửa sổ ước tính đầu tiên

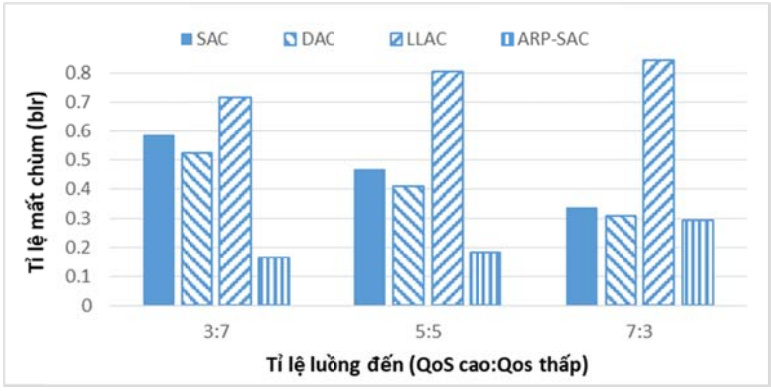
Mục tiêu mô phỏng là so sánh hiệu quả của mô hình điều khiển lập lịch dựa trên dự đoán ARP-SAC với các mô hình đã được đề xuất trước đây (SAC, DAC và LLAC) dựa trên tỷ lệ mất chùm ưu tiên, không ưu tiên và toàn mạng (cả ưu tiên và không ưu tiên), trong đó tỷ lệ mất chùm được tính như sau:

Tỉ lệ mất chùm (*burst loss rate, blr*) = số chùm bị đánh rơi / số chùm đến.

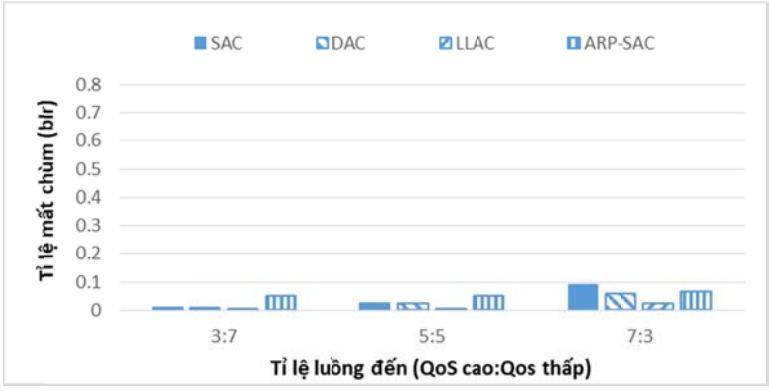
A. So sánh tỷ lệ mất chùm của các lớp QoS

Như được chỉ ra trong ***Hình 9, 10*** và ***11***, mô hình điều khiển lập lịch dựa trên dự đoán ARP-SAC có những cải thiện đáng kể về tỉ lệ mất chùm. Cụ thể, trong ***Hình 9***, DAC luôn có tỉ lệ mất chùm thấp hơn so với SAC và LLAC luôn có tỉ lệ mất chùm lớn hơn DAC đối với cả 3 trường hợp tỉ lệ luồng QoS cao và thấp đến: 3:7, 5:5 và 7:3. Điều này là phù hợp với kết quả mô phỏng đã được thực hiện trong [6]. Riêng đối với ARP-SAC, kết quả mô phỏng trong ***Hình 9*** chỉ ra rằng tỉ lệ mất chùm của ARP-SAC luôn cho kết quả tốt nhất nhờ chính sách cấp phát thêm băng thông cho luồng chùm QoS thấp. Tỉ lệ cải thiện về tỉ lệ mất chùm đối với luồng chùm QoS thấp trong trường hợp này là 44%, 38% và 20% với các trường hợp tỉ lệ luồng QoS cao và thấp đến 3:7, 5:5 và 7:3 Tuy nhiên việc cải thiện hiệu quả của luồng chùm QoS thấp cũng có tác động tiêu cực đến hiệu quả của luồng chùm QoS cao như được chỉ ra trong ***Hình 10***. Tuy nhiên tỉ lệ tăng về tỉ lệ mất chùm đối với luồng chùm QoS cao trong trường hợp này chỉ chiếm 4,5%, 3,4% và 0,8% đối với các trường hợp tỉ lệ mất chùm luồng QoS thấp đến 3:7, 5:5 và 7:3. Ngoài ra, khi xét về tỉ lệ mất chùm đối

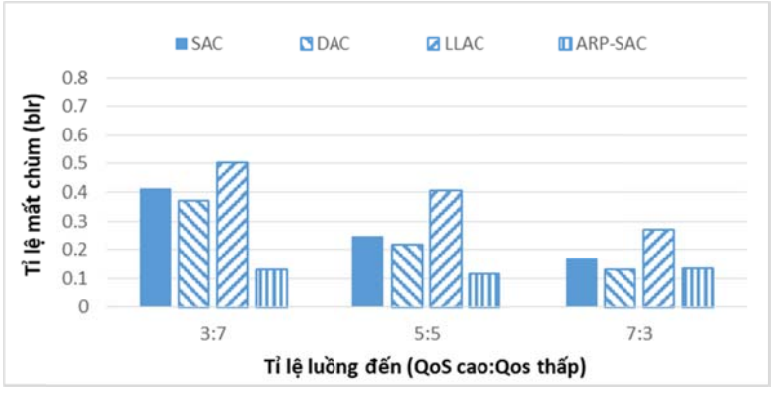
với cả hai luồng chùm QoS cao và thấp, ARP-SAC luôn cho kết quả tốt nhất như trong ***Hình 11***. Cụ thể hơn, tỉ lệ cải thiện về tỉ lệ mất chùm đối với luồng chùm QoS cao và QoS thấp trong trường hợp này là 30%, 17% và 6% đối với các trường hợp tỉ lệ luồng QoS cao và thấp đến 3:7, 5:5 và 7:3. Kết quả này khẳng định hiệu quả của mô hình điều khiển lập lịch dựa trên dự đoán ARP-SAC.



Hình 9. So sánh tỷ lệ mất chùm lớp QoS thấp giữa SAC, DAC, LLAC và ARP-SAC



Hình 10. So sánh tỷ lệ mất chùm lớp QoS cao giữa SAC, DAC, LLAC và ARP-SAC

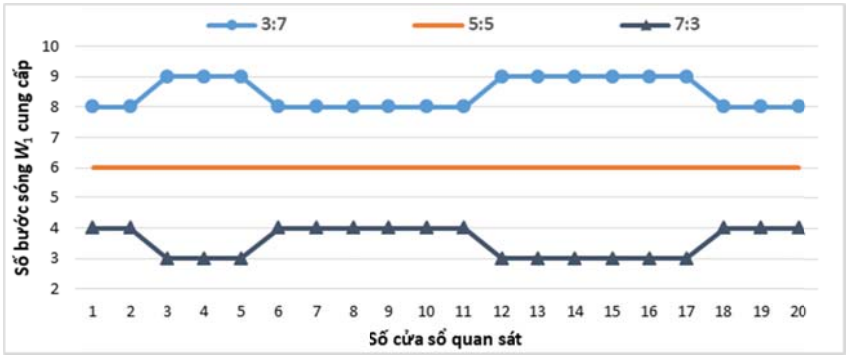


Hình 11. So sánh tỷ lệ mất chùm của lớp QoS cao và thấp giữa SAC, DAC, LLAC và ARP-SAC

B. So sánh số bước sóng được phẩn bổ cho lớp QoS thấp

Như được phân tích ở trong Mục IV.A mô hình ARP-SAC đã cải thiện hiệu quả lập lịch đáng kể cho các chùm QoS thấp. Điều này có được là do ARP-SAC cung cấp một cách linh động số bước sóng W_1 cho luồng chùm QoS thấp khi mật độ đến của chúng có sự thay đổi so với luồng chùm QoS cao. Như được chỉ ra trong ***Hình 12***, tuỳ thuộc vào tốc độ luồng chùm QoS cao và thấp đến mà số bước sóng W_1 được cấp phát cho luồng chùm QoS thấp có thay đổi: ví dụ với tỷ lệ 3:7, số bước sóng được cung cấp thêm cho luồng chùm QoS thấp được thực hiện tại của sổ quan sát thứ 3 hay 12, trong khi số sóng được cung cấp lại giảm xuống tại của sổ quan sát thứ 6 hay 18. Trong trường hợp tỷ lệ luồng chùm QoS cao và thấp đến bằng nhau (5:5), số bước sóng W_1 được cấp phát không thay đổi và bằng ½ W_0 (theo công thức 1) nên không có thay đổi về số bước sóng được cấp phát.

Tóm lại, việc cấp phát linh động số bước sóng W_1 cho luồng chùm QoS thấp đến theo công thức 5 và 6 chỉ có hiệu quả khi có một sự chênh lệch đáng kể về tốc độ đến của các luồng chùm QoS cao và thấp. Trong trường hợp tỷ lệ luồng chùm QoS cao và thấp đến xấp xỉ nhau, số bước sóng W_1 được cấp phát cho luồng chùm QoS thấp là không thay đổi.



Hình 12. So sánh sự phân bổ bước sóng cho luồng chùm QoS thấp trong 20 cửa sổ ước tính đầu tiên

V. KẾT LUẬN

Bài báo đã đề xuất một thuật toán điều khiển chấp nhận lập lịch thích nghi dựa trên dự đoán tốc độ đến ARP-SAC, trong đó việc cấp phát số lượng bước sóng cho luồng chùm QoS thấp được thực hiện linh hoạt tuỳ thuộc vào tỉ lệ tốc độ đến của các chùm QoS thấp và cao. Dựa trên các phân tích và kết quả mô phỏng, ARP-SAC đã thể hiện được các ưu điểm thông qua tỉ lệ mất chùm tổng thể thấp, tỉ lệ mất chùm đối với luồng QoS thấp và lượng thông tin cần lưu trữ ít hơn so với thuật toán đã công bố trước đây. Tuy nhiên, thuật toán điều khiển chấp nhận lập lịch thích nghi ARP-SAC chỉ hiệu quả trong mô trường mạng mà ở đó tốc độ đến của các luồng chùm này có nhiều thay đổi đột ngột. Cũng như các phương pháp dựa trên dự đoán khác, ARP-SAC cũng chịu chi phí tính toán đáng kể, mặt dù của sổ quan sát đã được giảm ½ so với chu kỳ dự đoán.

VI. TÀI LIỆU THAM KHẢO

[1] Costa, L. H. M. K., Fdida, S., Duarte "O. C. M. B.: Incremental servicedeployment using the Hop By Hop multicast routing protocol", IEEE/ACM Trans, Netw, 14(3), 543–556, 2006.

[2] C. Qiao et al, "Optical burst switching (OBS) - A new paradigm for an optical Internet, Journal of High Speed Networks", vol.8, pp.69-84, 1999.

[3] Qiao Zhang, "Absolute QoS Differentiation in OpticalBurst-Switched Network", IEEE, Vol 22, No 9, 2004.

[4] Pedro Revirego, J. A. Hernandez, J. Aracil, "Assembly admission control based on random packet selection at border nodes in Optical Burst-Switched networks", Springer Science+Business Media, LLC, 2008.

[5] Igor M. Moraes et al, "An efficient admission control mechanism for optical burst-switched networks", Springer Science+Business Media, LLC, 2008.

[6] Igor M. Moraes and O. C. M. B. Duarte, "Using the Network Load for AdmissionControl in OBS Networks: A Multilink Approach", J. Opt. Commun. Netw, Vol 2, No. 3, 2010.

[7] Salad K, Haidari F, "On the performan ceo fas imple packet rate estimator", IEEE/ACS2008 International Conference on Computer Systems and Applications, Doha, Qatar. NewYork, NY, USA: IEEE.pp.392-395, 31 March - 4 April 2008.

[8] Vo Viet Minh Nhat, Le Van Hoa, Nguyen Hoang Son, "A model of optimal burst assembly for delay reduction at ingress OBS nodes", Turkish Journal of Electrical Engineering & Computer Sciences, 3970-3982, 2017.

[9] M. Nandi, A. K. Turuk, D. K. Puthal and S. Dutta, "Best Fit Void Filling Algorithm in Optical Burst Switching Networks", IEEE Second International Conference on Emerging Trends in Engineering and Technology, 609-614, 2009.

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

SCHEDULING ADMISSION CONTROL BASED ON ARRIVING BURST RATE PREDICTION IN OBS NETWORK

Pham Trung Duc, Vo Viet Minh Nhat, Dang Thanh Chuong

ABSTRACT: *Scheduling is one of the important operations that has a significant influence on the performance of OBS networks. In the QoS supporting OBS network, scheduling needs to be considered with extra priority conditions. There have been several proposals about scheduling admission control, which most of them is considered in static and ideal conditions. The article will examine the scheduling admission control in a dynamic environment in which the rate of arriving bursts is predicted to control the scheduling operation effectivety. Some methods of arriving burst rate prediction is also analyzed, compared and evaluated.*