

ISSN 0866 708X

VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

TẠP CHÍ

KHOA HỌC VÀ CÔNG NGHỆ

JOURNAL OF SCIENCE AND TECHNOLOGY



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

Tập 52 - Số 6C

2014

Chuyên san năm thứ 5

**CÁC CÔNG TRÌNH NGHIÊN CỨU VỀ ĐIỆN TỬ,
VIỄN THÔNG VÀ CÔNG NGHỆ THÔNG TIN**



TẠP CHÍ KHOA HỌC VÀ CÔNG NGHỆ

Tổng biên tập

Nguyễn Xuân Phúc
Viện Khoa học vật liệu, Hà Nội
phucnx@ims.vast.ac.vn

Phó Tổng biên tập

Nguyễn Dũng, Viện Cơ học và Tin học ứng dụng, Tp Hồ Chí Minh; nguyendung@vast-hcm.ac.vn
Thái Hoàng, Viện Kỹ thuật nhiệt đới, Hà Nội; thoang@vnd.vast.ac.vn
Nguyễn Ngọc San, Học viện Công nghệ Bưu chính Viễn thông, Hà Nội; khoaqtsh@hn.vnn.vn

Ủy viên Hội đồng biên tập

Lê Huy Bá, Trường Đại học Công nghiệp Tp Hồ Chí Minh, Tp Hồ Chí Minh; lehuyba@hui.edu.vn
Lê Thanh Bình, Viện Công nghệ sinh học, Hà Nội; ltbinh24@yahoo.com
Phạm Quang Bắc, Tổng Công ty Dầu khí, Hà Nội
Nguyễn Hoài Châu, Viện Công nghệ Môi trường, Hà Nội; nhchau.cet@gmail.com
Nguyễn Hữu Dũng, Trường Đại học Bách khoa Hà Nội, Hà Nội; dung-sdh@mail.hut.edu.vn
Nguyễn Công Định, Học viện Kỹ thuật Quân sự, Hà Nội; dinhnc@mta.edu.vn
Nguyễn Văn Gia, Viện Cơ học và Tin học ứng dụng, Tp Hồ Chí Minh; nvg40@yahoo.com
Nguyễn Công Hào, Viện Công nghệ Hóa học, Tp Hồ Chí Minh; nguyenconghao@yahoo.com
Phan Bùi Khôi, Trường Đại học Bách khoa HN, Hà Nội; phanbuikhoi@mail.hut.edu.vn
Phạm Hữu Lý, Viện Hóa học, Hà Nội; huuly@yahoo.com.vn
Phạm Quốc Long, Viện Hóa học Các hợp chất thiên nhiên, Hà Nội; mar.biochem@fpt.vn
Trần Đình Long, Trường Đại học Bách khoa Hà Nội; evnbk@hn.vnn.vn
Châu Văn Minh, Viện Hóa sinh Biển, Hà Nội; cvminh@vast.ac.vn
Hoàng Minh, Học viện Công nghệ Bưu chính Viễn thông, Hà Nội; hoangminh@ptit.edu.vn
Nguyễn Xuân Nguyên, Viện Hóa học Các hợp chất thiên nhiên, Hà Nội; nxnguyen_ctc@yahoo.com
Lê Văn Nhung, Trường Đại học Bách khoa Hà Nội, Hà Nội; levannhuongtc@yahoo.com
Nguyễn Hữu Phương, Trường ĐH Khoa học tự nhiên, Tp Hồ Chí Minh; nhphuong@hcmuns.edu.vn
Phạm Văn Thiêm, Trường Đại học Bách khoa Hà Nội, Hà Nội; thiempv-edc@mail.hut.edu.vn
Hồ Sĩ Thoảng, Tổng Công ty Dầu khí, Tp Hồ Chí Minh; hosithoang@gmail.com
Phan Minh Tân, Sở Khoa học Công nghệ Tp HCM, Tp Hồ Chí Minh; tanpm@tphcm.gov.vn
Phan Đình Tuấn, Trường Đại học Bách khoa Tp HCM, Tp Hồ Chí Minh; pdtuan@hcmut.edu.vn

Thư kí Tòa soạn: **Lê Thị Thiên Hương**; tc_khcn@yahoo.com

Địa chỉ: Tòa soạn và Trị sự: Tầng 1, Nhà A16, 18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội;

Điện thoại/Fax: (84)-4-9422825; **Email:** v-jst@vjs.ac.vn, **Website:** <http://vjs.ac.vn/index.php/jst>.

BAN BIÊN TẬP SỐ CHUYÊN SAN

Trưởng ban

Hoàng Minh, *Tạp chí Khoa học và Công nghệ*

Phó Trưởng ban

Vũ Văn San, *Học viện Công nghệ Bưu chính Viễn thông*

Nguyễn Ngọc San, *Tạp chí Khoa học và Công nghệ*

Michael E. Kress, *Trường Đại học CSI, Đại học Thành phố NewYork, Hoa Kỳ*

Alexeev Evgeniy B, *Trường Đại học Bưu điện Matxcova, Liên bang Nga*

Các thành viên

Vũ Tuấn Lâm, *Học viện Công nghệ Bưu chính Viễn thông*

Tân Hạnh, *Học viện Công nghệ Bưu chính Viễn thông*

Lê Hữu Lập, *Học viện Công nghệ Bưu chính Viễn thông*

Zhanyang Zhang, *Trường Đại học CSI, Đại học Thành phố NewYork, Hoa Kỳ*

Eun Park, *Trường Đại học CSI, Đại học Thành phố NewYork, Hoa Kỳ*

Ievlev Oleg P, *Trường Đại học Bưu điện Matxcova, Liên bang Nga*

Cao Tiến Huỳnh, *Viện Khoa học Kỹ thuật và Công nghệ, Bộ Quốc phòng*

Đào Khắc An, *Viện Hàn lâm Khoa học và Công nghệ Việt Nam*

Nguyễn Bình, *Học viện Công nghệ Bưu chính Viễn thông*

Từ Minh Phương, *Học viện Công nghệ Bưu chính Viễn thông*

Trần Đình Quế, *Học viện Công nghệ Bưu chính Viễn thông*

Bùi Trung Hiếu, *Học viện Công nghệ Bưu chính Viễn thông*

Hoàng Đăng Hải, *Trung tâm VNCERT Việt Nam, Bộ Thông tin và Truyền thông*

Trần Công Hùng, *Học viện Công nghệ Bưu chính Viễn thông*

Phạm Ngọc Anh, *Học viện Công nghệ Bưu chính Viễn thông*

Nguyễn Tiến Ban, *Học viện Công nghệ Bưu chính Viễn thông*

Đặng Hoài Bắc, *Học viện Công nghệ Bưu chính Viễn thông*

Lê Nhật Thăng, *Học viện Công nghệ Bưu chính Viễn thông (Thư ký)*

MỘT CÁCH TIẾP CẬN HEURISTIC CHO BÀI TOÁN ĐỊNH TUYẾN ĐẢM BẢO BĂNG THÔNG VÀ ĐỘ TRỄ

Cao Thái Phương Thanh^{1,*}, Hà Hải Nam², Trần Công Hùng³

¹Đại học Sài Gòn, 273 An Dương Vương, Q. 5, Tp. Hồ Chí Minh

²Học viện Công nghệ Bưu chính Viễn thông, 122 Hoàng Quốc Việt, Cầu Giấy, Hà Nội

³Học viện Công nghệ Bưu chính Viễn thông Cơ sở Tp. Hồ Chí Minh, 11 Nguyễn Đình Chiểu, Quận 1, Tp. Hồ Chí Minh

* Email: ctp THANH@sgu.edu.vn

Đến Tòa soạn: 19/8/2014; Chấp nhận đăng: 29/12/2014

TÓM TẮT

Bài báo này giới thiệu một thuật toán định tuyến đảm bảo hai điều kiện chất lượng dịch vụ: băng thông tối thiểu và độ trễ tối đa. Mục tiêu của thuật toán là đáp ứng càng nhiều yêu cầu định tuyến càng tốt. Nhằm lựa chọn đường đi linh hoạt theo yêu cầu, thuật toán tính giá trị trọng số liên kết dựa trên băng thông. Sau đó áp dụng một cách tiếp cận heuristic vào thuật toán Dijkstra để tìm đường đi thỏa yêu cầu độ trễ và có trọng số nhỏ. Kết quả thực nghiệm trên các sơ đồ mạng khác nhau với tham số và yêu cầu định tuyến khác nhau cho thấy thuật toán được đề xuất có tỉ lệ chấp nhận yêu cầu và thời gian tính toán trung bình tốt hơn các giải pháp đã công bố.

Từ khóa: định tuyến đảm bảo băng thông và độ trễ, kĩ thuật lưu lượng, thuật toán heuristic.

1. GIỚI THIỆU

Ứng dụng mạng, đặc biệt là các ứng dụng đa phương tiện như truyền thanh, truyền hình trực tuyến yêu cầu chất lượng dịch vụ (Quality of Service - QoS) ngày càng tốt hơn. Những yêu cầu này được định nghĩa thành các điều kiện chất lượng dịch vụ trong hợp đồng giữa nhà cung cấp mạng và khách hàng. Trong quá trình vận hành mạng, nhà cung cấp không chỉ cần đảm bảo điều kiện chất lượng dịch vụ mà còn muốn sử dụng tài nguyên mạng một cách hiệu quả nhất. Vì vậy, thuật toán định tuyến kĩ thuật lưu lượng với điều kiện đảm bảo chất lượng dịch vụ là một lĩnh vực quan trọng đã và đang được quan tâm nghiên cứu [1]. Mục tiêu chính của thuật toán là đáp ứng càng nhiều yêu cầu định tuyến càng tốt với điều kiện đường đi được thiết lập cho mỗi yêu cầu phải đảm bảo một hoặc một số điều kiện chất lượng dịch vụ.

Điều kiện chất lượng dịch vụ phổ biến nhất trong các công trình định tuyến đã công bố là điều kiện đảm bảo băng thông [2] của đường đi lớn hơn một giới hạn yêu cầu (băng thông tối thiểu). Một cách tổng quát, các thuật toán định tuyến tính giá trị trọng số liên kết, sau đó áp dụng thuật toán tìm đường có trọng số nhỏ nhất (ví dụ Dijkstra) để xác định đường đi cho mỗi yêu cầu định tuyến. Trong thuật toán tìm đường, điều kiện băng thông được đảm bảo bằng cách không

xét các liên kết có băng thông nhỏ hơn băng thông yêu cầu. Việc đảm bảo điều kiện băng thông đơn giản vì băng thông được xét độc lập trên từng liên kết. Một điều kiện chất lượng dịch vụ phổ biến khác là đảm bảo độ trễ của đường đi nhỏ hơn một giới hạn yêu cầu (độ trễ tối đa). Khác với băng thông, điều kiện độ trễ phải xét theo tổng độ trễ của tất cả liên kết trên đường đi. Vì vậy, các thuật toán định tuyến đảm bảo băng thông không thể áp dụng để đảm bảo yêu cầu độ trễ. Trong sự hiểu biết của tác giả, số lượng thuật toán định tuyến đảm bảo đồng thời hai yêu cầu chất lượng dịch vụ băng thông và độ trễ ít hơn hẳn số thuật toán đảm bảo một yêu cầu. Bài báo này giới thiệu một cách tiếp cận heuristic mới cho bài toán định tuyến đảm bảo băng thông và độ trễ với hai đóng góp: (1) tính trọng số liên kết một cách đơn giản nhưng hiệu quả dựa trên băng thông và (2) điều chỉnh thuật toán Dijkstra để tìm đường đi thỏa điều kiện độ trễ và có trọng số nhỏ.

Bài báo được trình bày gồm năm phần. Sau phần giới thiệu, phần 2 định nghĩa bài toán và trình bày các công trình liên quan. Phần 3 đề xuất thuật toán định tuyến đảm bảo băng thông và độ trễ với tên gọi enhanced-Heuristic Routing Algorithm with Bandwidth Delay Constraints (eHRABDC). Hiệu quả định tuyến của thuật toán eHRABDC được so sánh, đánh giá với các thuật toán khác trong phần 4 theo hai tiêu chí tỉ lệ chấp nhận yêu cầu và thời gian tính toán trung bình. Cuối cùng, phần 5 trình bày kết luận và hướng phát triển.

2. ĐỊNH NGHĨA BÀI TOÁN VÀ CÁC CÔNG TRÌNH LIÊN QUAN

2.1. Định nghĩa bài toán

Sơ đồ mạng được biểu diễn như một đồ thị $G(N, L)$. N là tập các nút mạng và L là tập các liên kết. Mỗi liên kết $l \in L$ có ba thông số: dung lượng (capacity - $c(l)$), băng thông còn lại (residual bandwidth - $b(l)$), và độ trễ (delay - $d(l)$). Tương tự như phần lớn công trình định tuyến đảm bảo băng thông và độ trễ đã công bố [3 – 5] bài báo này giới hạn độ trễ của liên kết là độ trễ lan truyền. Như vậy, dung lượng và độ trễ của liên kết có giá trị cố định, trong khi băng thông còn lại có giá trị thay đổi theo lượng băng thông đang được bảo đảm trên liên kết. Một yêu cầu định tuyến được kí hiệu $r(i, e, \beta, \delta)$ sẽ yêu cầu thiết lập một đường đi p_{ie} từ nút vào i đến nút ra e . Đường đi p_{ie} cần thỏa hai điều kiện: mỗi liên kết trên đường đi có băng thông còn lại lớn hơn hoặc bằng β và tổng độ trễ của các liên kết nhỏ hơn hoặc bằng δ . Hơn nữa, nếu p_{ie} được thiết lập, băng thông còn lại của các liên kết thuộc đường đi này sẽ giảm β đơn vị (đảm bảo băng thông).

Trong phạm vi nghiên cứu, yêu cầu định tuyến được giả thiết đến tuần tự và các cặp nút vào – ra trong sơ đồ mạng được biết trước. Ngoài ra, thuật toán định tuyến được thực hiện tập trung với giả thiết các thông số liên kết như dung lượng, băng thông còn lại, và độ trễ đã sẵn sàng cho quá trình xử lí. Mục tiêu của thuật toán định tuyến là đáp ứng tối đa số yêu cầu định tuyến. Vì vậy, tỉ lệ chấp nhận yêu cầu được sử dụng làm độ đo hiệu quả định tuyến.

$$\text{tỉ lệ chấp nhận} = \frac{\text{số yêu cầu được chấp nhận}}{\text{tổng số yêu cầu}} * 100\% \quad (1)$$

2.2. Công trình liên quan

Thuật toán định tuyến đảm bảo băng thông và độ trễ được giới thiệu đầu tiên là Least Delay Algorithm (LDA) [3]. Đối với mỗi yêu cầu, LDA đánh dấu bỏ qua các liên kết không thỏa điều kiện băng thông và áp dụng thuật toán Dijkstra với trọng số là độ trễ của các liên kết còn lại. Độ trễ (nhỏ nhất) của đường đi tìm được sẽ được so sánh với độ trễ cần đảm bảo δ để quyết định có chấp nhận yêu cầu hay không. Tuy đơn giản nhưng LDA luôn chọn cùng một đường đi (có độ

trễ nhỏ nhất) cho một cặp nút vào – ra. Kết quả là một phần sơ đồ mạng thuộc các đường đi có độ trễ nhỏ bị sử dụng nhiều, dễ tạo ra hiện tượng tắc cổ chai và cuối cùng làm giảm tỉ lệ chấp nhận yêu cầu định tuyến.

Thuật toán Minimum Interference Routing Algorithm (MIRA) [6] giới thiệu ý tưởng lựa chọn đường đi sao cho hạn chế ảnh hưởng đến các yêu cầu định tuyến trong tương lai. Sự ảnh hưởng này được đánh giá dựa trên tính chất maxflow – mincut và được thể hiện qua giá trị trọng số liên kết. Cụ thể, liên kết thuộc tập mincut được cho là có nhiều ảnh hưởng vì nếu được sử dụng sẽ làm giảm maxflow của một cặp vào – ra. Vì vậy, liên kết thuộc tập mincut được gọi là tới hạn và có trọng số cao. Ý tưởng hạn chế ảnh hưởng được kế thừa và phát triển với nhiều phương pháp tính trọng số heuristic khác nhau trong nhiều công trình định tuyến đảm bảo băng thông đã được công bố [2].

Các thuật toán đảm bảo băng thông có thể mở rộng thêm yêu cầu đảm bảo độ trễ bằng cách áp dụng Dijkstra mở rộng [7]. Cụ thể, thuật toán Dijkstra mở rộng tìm đường đi có trọng số nhỏ nhất và thỏa thêm yêu cầu tổng một thông số nguyên dương của các liên kết trên đường tìm được nhỏ hơn hoặc bằng một giá trị cho trước. Cụ thể với yêu cầu thỏa độ trễ δ , Dijkstra mở rộng duy trì ở mỗi nút mạng một danh sách gồm $(\delta + 1)$ mục $EN_n = \{en_n^0, \dots, en_n^\delta\}$. Mỗi mục $en_n^k = \{d_n^k, w_n^k, m_n^k\}$ ở nút n có thông tin về nút liền trước (m_n^k), trọng số (w_n^k), và độ trễ (d_n^k) của quá trình tìm đường đi từ nút vào i đến nút n . Ngoài ra, giá trị độ trễ của mỗi mục được gán trước $d_n^k = k$, $k = [0, 1, \dots, \delta]$ và được sử dụng như là chỉ số danh sách các mục. Dựa trên danh sách $(\delta + 1)$ mục này mà thuật toán Dijkstra mở rộng tìm được đường đi có trọng số nhỏ nhất thỏa điều kiện độ trễ nhỏ hơn hay bằng δ . Thuật toán 1 trình bày chi tiết thuật toán Dijkstra mở rộng tìm đường đi có trọng số nhỏ nhất thỏa yêu cầu độ trễ. Lưu ý, vì sử dụng độ trễ làm chỉ số của danh sách EN_n , thuật toán Dijkstra mở rộng chỉ có thể áp dụng với giá trị độ trễ nguyên dương.

Áp dụng Dijkstra mở rộng, các tác giả của [4] đề xuất hai thuật toán đảm bảo băng thông và độ trễ: Maximum Delay Weighted Capacity Routing Algorithm (MDWCRA) và Modified Maximum Delay Weighted Capacity Routing Algorithm (M-MDWCRA). Hai thuật toán này định nghĩa giá trị DWC (delay-weighted capacity) tỉ lệ thuận với khả năng đáp ứng yêu cầu của các cặp nút vào – ra theo công thức sau:

$$DWC_{ie} = \sum_{LP_{ie}^l \in LP_{ie}} \frac{B_{ie}^l}{D_{ie}^l} \quad (2)$$

trong đó, $LP_{ie} = \{LP_{ie}^1, \dots, LP_{ie}^l, \dots, LP_{ie}^k\}$ là tập các đường đi có độ trễ nhỏ nhất của cặp ie (được xác định bằng Dijkstra với trọng số là độ trễ). Hai thuật toán MDWCRA và M-MDWCRA khác nhau ở cách xác định đường đi trong tập LP_{ie} . Cụ thể, thuật toán MDWCRA áp dụng Dijkstra để tìm đường đi có độ trễ thấp nhất thứ l trong tập LP_{ie} sau khi bỏ tất cả liên kết của đường đi đã tìm được trước đó LP_{ie}^{l-1} . Trong khi thuật toán M-MDWCRA chỉ bỏ liên kết có băng thông còn lại nhỏ nhất trong đường LP_{ie}^{l-1} . Ngoài ra, trong công thức 2, B_{ie}^l là băng thông còn lại và D_{ie}^l là tổng độ trễ của đường LP_{ie}^l . Do DWC tỉ lệ thuận với khả năng đáp ứng yêu cầu nên thuật toán lựa chọn đường đi sao cho giá trị DWC ít giảm. Đối với mỗi đường LP_{ie}^l , liên kết có băng thông còn lại nhỏ nhất ($b(l) = B_{ie}^l$, $l \in LP_{ie}^l$) nếu được sử dụng sẽ làm giảm DWC nhiều nhất, vì vậy liên kết này được gọi là liên kết tới hạn và được tăng trọng số. Tập hợp các liên kết này được gọi là tập liên kết tới hạn theo độ trễ. Sau khi xử lý tập LP cho mọi cặp vào – ra (cập nhật trọng số liên kết trong mỗi bước tìm LP^l), MDWCRA và M-MDWCRA áp dụng Dijkstra mở rộng để xác định đường đi có trọng số nhỏ nhất thỏa yêu cầu độ trễ.

Gần đây, thuật toán đảm bảo băng thông và độ trễ Minimum Delay and Maximum Flow (MDMF) [8] kết hợp ý tưởng tập liên kết tới hạn theo độ trễ của M-MDWCRA [4] và tập liên

Thuật toán 1 Dijkstra mở rộng thỏa điều kiện độ trễ [7]

Input:
 Mạng $G(N, L)$
 Yêu cầu định tuyến $r(i, e, \beta, \delta)$

Output:
 Đường đi có trọng số nhỏ nhất thỏa yêu cầu độ trễ δ
 hoặc NULL // không có đường đi

```

1: function EXTENDED_DIJKSTRA( $G, r$ )
2:   INIT( $G$ )
3:    $Q = \{en_n^k | n \in N, k \in [0, \delta]\}$  //  $en_n^k$  là mục thứ  $k$  của nút  $n$ 
4:   while  $Q \neq \emptyset$  do
5:     tìm  $en_n^k \in Q : w_n^k = \min_{en \in Q} \{w\}$ 
6:      $Q = Q - en_n^k$ 
7:     if  $w_n^k = \infty$  then
8:       break
9:     end if
10:    for liên kết  $l$  từ nút  $n$  đến nút liền kề  $o$  do
11:      RELAX( $Q, en_n^k, l, o, \delta$ )
12:    end for
13:  end while
14:  GETPATH( $G$ )
15: end function

16: function INIT( $G$ )
17:   for nút  $n \in N$  do
18:     for mục  $en_n^k | k \in [0, \delta]$  do
19:        $d_n^k = k$  // độ trễ
20:        $w_n^k = \infty$  // trọng số
21:        $m_n^k = NULL$  // nút trước
22:     end for
23:   end for
24:   for mục  $en_i^k | k \in [0, \delta]$  do // nút vào  $i$ 
25:      $w_i^k = 0$ 
26:   end for
27: end function

28: function RELAX( $Q, en_n^k, l, o, \delta$ )
29:    $new\_d = d_n^k + d(l)$ 
30:   if  $en_o^{new\_d} \in Q$  và  $new\_d \leq \delta$  then
31:      $new\_w = w_n^k + w(l)$ 
32:     if  $new\_w < w_o^{new\_d}$  then
33:        $w_o^{new\_d} = new\_w$ 
34:        $m_o^{new\_d} = n$ 
35:     end if
36:   end if
37: end function

38: function GETPATH( $G$ )
39:   // xét các mục của nút ra  $e$ 
40:   tìm mục  $en_e^k : w_e^k \neq \infty$  và  $w_e^k = \min_{h \in [0, \delta]} \{w_e^h\}$ 
41:   if tìm được  $en_e^k$  then
42:     trả về đường đi dựa trên nút trước  $m_e^k$ 
43:   end if
44:   trả về NULL
45: end function
    
```

kết tới hạn theo maxflow - mincut của MIRA [6] để tính trọng số liên kết (công thức 3). Ngoài ra, MDMF áp dụng mô hình latency-rate server [9] để tính độ trễ theo băng thông. Từ đó, MDMF giới thiệu một phương pháp heuristic tăng lượng băng thông đảm bảo để làm giảm độ trễ cho đường đi nhằm cố gắng thỏa điều kiện độ trễ.

$$w(l) = \frac{1 + \mu_{flow}CM + v_{delay}CD}{b(l)} \quad (3)$$

$$CM = \sum_{(i,e) \in IE} v_l$$

$$v_l = \begin{cases} 1 & \text{nếu } l \text{ thuộc tập mincut của } (i, e) \\ 0 & \text{nếu } l \text{ không thuộc tập mincut của } (i, e) \end{cases}$$

$$CD = \sum_{(i,e) \in IE} w_l$$

$$w_l = \begin{cases} 1 & \text{nếu } l \text{ thuộc tập liên kết tới hạn theo độ trễ của } (i, e) \\ 0 & \text{nếu } l \text{ không thuộc tập liên kết tới hạn theo độ trễ của } (i, e) \end{cases}$$

IE là tập các cặp nút vào - ra

μ_{flow}, v_{delay} là các tham số điều khiển

3. THUẬT TOÁN ĐỊNH TUYẾN ĐẢM BẢO BẰNG THÔNG VÀ ĐỘ TRỄ

Đối với các thuật toán định tuyến kỹ thuật lưu lượng, mục đích của việc tính trọng số liên kết theo yêu cầu là thay đổi đường đi linh hoạt với thông số hệ thống mạng. Nhận xét rằng trong phạm vi bài toán, thông số quan trọng và thay đổi nhiều nhất là băng thông còn lại, nên trọng số liên kết được đề xuất tính đơn giản như sau:

$$w(l) = \frac{\text{băng thông đang đảm bảo}}{\text{băng thông còn lại}} = \frac{c(l) - b(l)}{b(l)} \quad (4)$$

Theo công thức 4, trong hai liên kết có cùng dung lượng, liên kết nào còn ít băng thông sẽ có giá trị trọng số lớn và được hạn chế sử dụng. Điều này giúp thuật toán cân bằng tải và hạn chế hiện tượng thất cổ chai. Với ý nghĩa tương tự, trong hai liên kết có cùng băng thông còn lại, liên kết nào có dung lượng lớn (dẫn đến giá trị trọng số lớn) cũng sẽ được hạn chế sử dụng vì liên kết này đang đảm bảo lượng băng thông lớn hơn liên kết còn lại.

Sau khi tính trọng số liên kết, có thể áp dụng Dijkstra mở rộng để tìm đường đi có trọng số nhỏ nhất, thỏa điều kiện độ trễ. Tuy nhiên, thuật toán Dijkstra mở rộng xét $(\delta + 1)$ mục cho mỗi nút mạng nên độ phức tạp cao và thời gian xử lý lâu. Trong bài toán định tuyến theo yêu cầu, độ trễ là điều kiện bắt buộc còn trọng số giúp lựa chọn đường đi linh hoạt và không bắt buộc phải nhỏ nhất. Vì vậy, nhằm giảm thời gian xử lý, thuật toán Dijkstra được điều chỉnh một cách heuristic để tìm đường đi thỏa điều kiện độ trễ và có trọng số càng nhỏ càng tốt. Đầu tiên, thuật toán tìm đường được đề xuất, gọi là heuristic-Dijkstra, tính giá trị độ trễ thấp nhất từ tất cả nút mạng đến nút ra. Mỗi nút mạng chỉ có một mục $en_n = \{d_n, w_n, m_n\}$ thay vì $(\delta + 1)$ như Dijkstra mở rộng. Xét hai nút mạng liền kề m và n với hai mục tương ứng $en_m = \{d_m, w_m, m_m\}$ và $en_n = \{d_n, w_n, m_n\}$. Liên kết từ m tới n được ký hiệu là l_{mn} . Trong quá trình tìm đường đi từ nút m đến nút ra e , mục en_n tại nút n được cập nhật (nghĩa là đường đi có qua nút n) khi thỏa hai điều kiện. Thứ nhất, trọng số tại nút n lớn hơn tổng trọng số của liên kết l_{mn} và trọng số lưu tại nút trước m ($w_n > w(l_{mn}) + w_m$). Thứ hai, độ trễ yêu cầu δ lớn hơn hoặc bằng tổng của ba giá trị: độ trễ lưu tại nút m , độ trễ của liên kết l_{mn} , và độ trễ thấp nhất từ nút n đến nút ra e ($\delta \geq d_m + d(l_{mn}) + l_{n,e}$). Điều kiện thứ hai này giúp đảm bảo đường đi được cập nhật qua nút n sẽ thỏa điều kiện độ trễ. Thuật toán 2 trình bày chi tiết thuật toán tìm đường heuristic-Dijkstra.

Heuristic Dijkstra không đảm bảo tìm được đường đi từ nút vào đến nút ra có trọng số nhỏ nhất vì ở mỗi nút chỉ có một mục giữ một đường đi trọng số nhỏ nhất tại thời điểm đang xét, trong khi các đường đi khác có thể có trọng số nhỏ hơn (khi đến nút ra) không được giữ lại. Nhằm tăng cường khả năng tìm đường đi có trọng số nhỏ, một vòng lặp được áp dụng cho thuật toán heuristic-Dijkstra. Sau mỗi lần lặp, liên kết có trọng số lớn nhất trong đường đi (nếu có) bị tạm thời loại bỏ khỏi sơ đồ mạng. Khi vòng lặp kết thúc, đường đi được lựa chọn là đường đi có tổng trọng số nhỏ nhất trong tập hợp các đường đi tìm được. Đề xuất này được gọi là enhanced-heuristic-Dijkstra và được trình bày trong thuật toán 3.

Thuật toán 2 Heuristic Dijkstra thỏa điều kiện độ trễ

Input:
 Mạng $G(N, L)$
 Yêu cầu định tuyến $r(i, e, \beta, \delta)$
 Độ trễ thấp nhất từ các nút mạng đến nút ra $ld_{n_e} | n \in N$

Output:
 Đường đi thỏa điều kiện độ trễ
 hoặc NULL // không có đường đi

```

1: function HEURISTIC_DIJKSTRA( $G, r$ )
2:   INIT( $G$ )
3:    $Q = \{en_n | n \in N\}$  //  $en_n$  là một mục của nút  $n$ 
4:   while  $Q \neq \emptyset$  do
5:     tìm  $en_n \in Q : w_n = \min_{en \in Q} \{w\}$ 
6:      $Q = Q - en_n$ 
7:     if  $w_n = \infty$  then
8:       break
9:     end if
10:    for liên kết  $l$  từ nút  $n$  đến nút liền kề  $o$  do
11:      RELAX( $Q, en_n, l, o, \delta, ld_{o_e}$ )
12:    end for
13:  end while
14:  GETPATH( $G$ )
15: end function

16: function INIT( $G$ )
17:   for nút  $n$  in  $N$  do
18:      $d_n = k$  // độ trễ
19:      $w_n = \infty$  // trọng số
20:      $m_n = NULL$  // nút trước
21:   end for
22:    $w_i = d_i = 0$  // nút vào  $i$ 
23: end function

24: function RELAX( $Q, en_n, l, o, \delta, ld_{o_e}$ )
25:    $new\_w = w_n + w(l)$ 
26:   if  $new\_w < w_o$  then
27:      $new\_d = d_n + d(l)$ 
28:     if  $en_o \in Q$  và  $(new\_d + ld_{o_e}) \leq \delta$  then
29:        $w_o = new\_w$ 
30:        $d_o = new\_d$ 
31:        $m_o = n$ 
32:     end if
33:   end if
34: end function

35: function GETPATH( $G$ )
36:   if  $m_e \neq NULL$  then
37:     trả về đường đi dựa trên nút trước  $m_e$ 
38:   end if
39:   trả về NULL
40: end function
    
```

Thuật toán 3 Enhanced-heuristic Dijkstra thỏa điều kiện độ trễ

Input:
 Mạng $G(N, L)$
 Yêu cầu định tuyến $r(i, e, \beta, \delta)$
 Độ trễ thấp nhất từ các nút mạng đến nút ra $ld_{n_e} | n \in N$

Output:
 Đường đi thỏa điều kiện độ trễ
 hoặc NULL // không có đường đi

```

1: function ENHANCED_HEURISTIC_DIJKSTRA( $G, r$ )
2:    $\Pi = \emptyset$ 
3:    $p = HEURISTIC_DIJKSTRA(G, r)$ 
4:   while  $p \neq NULL$  do
5:     Thêm  $p$  vào  $\Pi$ 
6:     Tạm thời bỏ liên kết  $max\_l$  khỏi  $G$  :
        $w(max\_l) = \max_{l \in p} \{w(l)\}$ 
7:      $p = HEURISTIC_DIJKSTRA(G, r)$ 
8:   end while
9:   if  $\Pi \neq \emptyset$  then
10:    trả về  $P : w(P) = \min_{p \in \Pi} \{w(p)\}$ 
11:   end if
12:   trả về NULL
13: end function
    
```

Cuối cùng, Bảng 1 trình bày các bước của thuật toán định tuyến bảo đảm băng thông và độ trễ được đề xuất với tên gọi enhanced-Heuristic Routing Algorithm with Bandwidth Delay Constraints (eHRABDC).

Bảng 1. Thuật toán đề xuất eHRABDC

Đầu vào	Mạng $G(N, L)$ Yêu cầu định tuyến $r(i, e, \beta, \delta)$
Đầu ra	Đường đi đảm bảo băng thông và độ trễ p_{ie} Hoặc không có đường đi
Thuật toán	Với mỗi yêu cầu $r(i, e, \beta, \delta)$: 1. Tính trọng số liên kết $w(l)$ theo công thức (4) 2. Bỏ qua các liên kết không đủ băng thông yêu cầu $b(l) < \beta$ 3. Tính độ trễ thấp nhất từ mọi nút mạng còn lại đến nút ra e để sử dụng trong enhanced-heuristic Dijkstra 4. Áp dụng enhanced-heuristic Dijkstra tìm đường đi từ nút i tới nút e thỏa yêu cầu độ trễ δ Nếu tìm thấy: trả về đường đi p_{ie} và cập nhật các thông tin cần thiết. Ngược lại: từ chối yêu cầu (không có đường đi).

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1. Thiết lập thực nghiệm

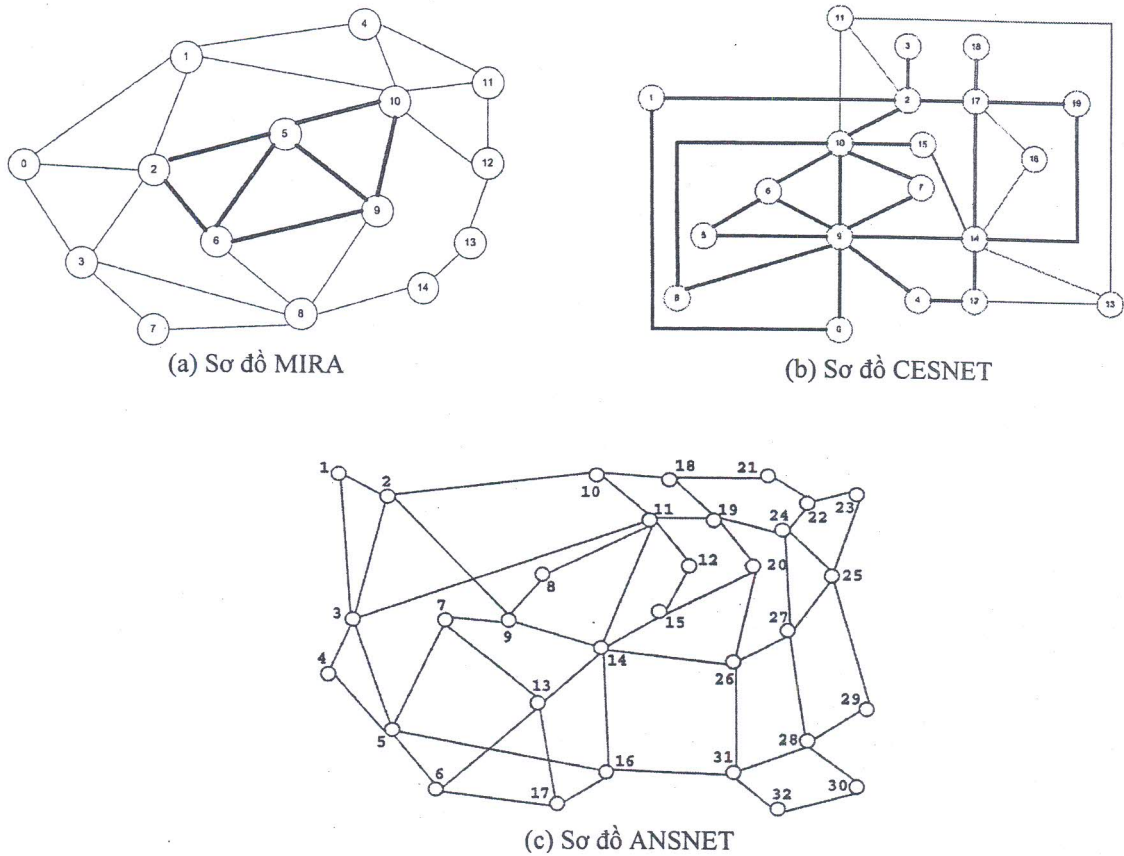
Thuật toán đề xuất eHRABDC được so sánh với các thuật toán đã công bố có cùng điều kiện đánh giá là Least Delay Algorithm (LDA) [3], và Modified Maximum Delay Weighted Capacity Routing Algorithm (M-MDWCRA) [4]. M-MDWCRA được thực nghiệm thay vì cả hai thuật toán MDWCRA và M-MDWCRA vì các tác giả của chúng đã khẳng định và chứng minh M-MDWCRA có hiệu quả tốt hơn MDWCRA [4]. Như đã trình bày ở phần 2.1 hiệu quả định tuyến được đánh giá theo tỉ lệ chấp nhận yêu cầu. Tỉ lệ chấp nhận cao chứng tỏ thuật toán định tuyến tốt. Tiêu chí đánh giá thứ hai là thời gian tính toán trung bình. Đây là giá trị thời gian trung bình từ khi nhận được yêu cầu đến khi thuật toán có kết quả đường đi hoặc từ chối yêu cầu. Như vậy, thuật toán định tuyến nên có thời gian tính toán trung bình thấp.

Hình 1 thể hiện ba sơ đồ mạng thực nghiệm với thông số cụ thể như sau:

- Sơ đồ MIRA (Hình 1(a)) là sơ đồ thực nghiệm được dùng trong hầu hết các công trình định tuyến đảm bảo chất lượng dịch vụ [2, 4, 6]. Sơ đồ này có hai loại dung lượng băng thông: 4800 (liên kết đậm) và 1200 (liên kết nhạt) đơn vị băng thông và bốn cặp nút vào - ra: (0, 12), (4, 8), (3, 1), (4, 14).
- Sơ đồ CESNET (Hình 1(b)) được xây dựng từ một mô hình mạng MPLS thực tế [10]. CESNET cũng có hai loại dung lượng 1000 và 10000 đơn vị băng thông lần lượt cho các liên kết nhạt và đậm. Các cặp nút vào - ra là: (0, 18), (1, 11), (3, 16), (4, 7), (5, 13), (6, 19), (15, 0), (19, 8).
- Sơ đồ ANSNET (Hình 1(c)) được xây dựng dựa trên địa lí nước Mỹ [4, 7] với các liên kết có dung lượng 1200 đơn vị. Năm cặp nút vào - ra được chọn là: (1, 29), (18, 6), (3, 23), (7, 31), (21, 17).

Tương tự như [4], độ trễ liên kết của cả ba sơ đồ mạng được gán giá trị ngẫu nhiên trong đoạn [5, 11] đơn vị thời gian.

Yêu cầu định tuyến được phát sinh ngẫu nhiên đều trong các cặp vào - ra của sơ đồ mạng. Băng thông và độ trễ cần đảm bảo cũng được phát sinh ngẫu nhiên trong một khoảng giá trị dựa



Hình 1. Sơ đồ mạng thực nghiệm.

trên thông số liên kết của từng sơ đồ. Ví dụ, băng thông yêu cầu trên sơ đồ CESNET có giá trị lớn hơn trên MIRA và ANSNET vì dung lượng của liên kết trên CESNET lớn. Tương tự, độ trễ yêu cầu trên MIRA nhỏ hơn hai sơ đồ còn lại vì sơ đồ MIRA có kích thước nhỏ nhất. Ngoài ra, thời gian đến của yêu cầu định tuyến tuân theo phân phối Poisson, trung bình có λ yêu cầu trong một đơn vị thời gian. Nếu được chấp nhận, yêu cầu định tuyến sẽ giữ băng thông của đường đi trong một khoảng thời gian được phát sinh theo phân phối Exponential với tham số μ đơn vị thời gian. Các bộ dữ liệu thử nghiệm được chia thành hai loại thể hiện hai điều kiện tải trong mạng. Điều kiện tải bình thường có 2000 yêu cầu trong mỗi bộ dữ liệu và giá trị các tham số thời gian đến λ , thời gian giữ μ nhỏ hơn so với điều kiện tải cao (có 5000 yêu cầu định tuyến / 1 bộ thử nghiệm).

4.2. Thử nghiệm thuật toán định tuyến

Hiệu quả định tuyến của các thuật toán được so sánh trên nhiều bộ dữ liệu với băng thông và độ trễ yêu cầu cũng như tham số thời gian đến (λ), thời gian giữ (μ) khác nhau. Kết quả thực nghiệm cho thấy thuật toán được đề xuất eHRABDC không chỉ đạt tỉ lệ chấp nhận yêu cầu cao mà còn có thời gian tính toán trung bình thấp.

Cụ thể, Bảng 2 trình bày kết quả thực nghiệm trên sơ đồ MIRA trong điều kiện tải bình thường (mỗi thực nghiệm có 2000 yêu cầu, với tốc độ đến được phát sinh ngẫu nhiên theo phân phối Poisson $\lambda = 40$ yêu cầu trong một đơn vị thời gian, thời gian giữ băng thông của mỗi yêu

cầu được phát sinh theo phân phối Exponential với $\mu = 10$ đơn vị thời gian) và điều kiện tải cao (có 5000 yêu cầu, $\lambda = 80$, $\mu = 30$). Bảng thông và độ trễ yêu cầu được phát sinh lần lượt trong tập $\{10, 15, 20, 25\}$ đơn vị băng thông và đoạn $[35, 60]$ đơn vị thời gian. Thuật toán eHRABDC có tỉ lệ chấp nhận cao nhất (89,9 % và 41,08 %) trong khi LDA có tỉ lệ thấp nhất (85,9 % và 36,74 %). Vì sơ đồ MIRA có ít liên kết và đường đi giữa các cặp vào - ra sử dụng chung một/một số liên kết nên việc lựa chọn đường đi độ trễ thấp nhất của LDA làm giảm đáng kể hiệu quả định tuyến. Ngược lại, việc tính trọng số theo băng thông của eHRABDC cho kết quả tốt vì giúp lựa chọn đường đi linh động. Thuật toán M-MDWCRA cũng có tỉ lệ chấp nhận cao hơn LDA đáng kể, tuy nhiên việc xử lý tập liên kết tới hạn theo độ trễ LP cho các cặp nút vào - ra và đặc biệt là phương pháp Dijkstra mở rộng làm thời gian tính toán trung bình của M-MDWCRA rất cao. Ví dụ trong thực nghiệm đã trình bày, thời gian trung bình của M-MDWCRA cao hơn eHRABDC 11 lần trong điều kiện tải bình thường và hơn 34 lần trong điều kiện tải cao. Hình 2 so sánh rõ hơn kết quả định tuyến của các thuật toán trên sơ đồ MIRA.

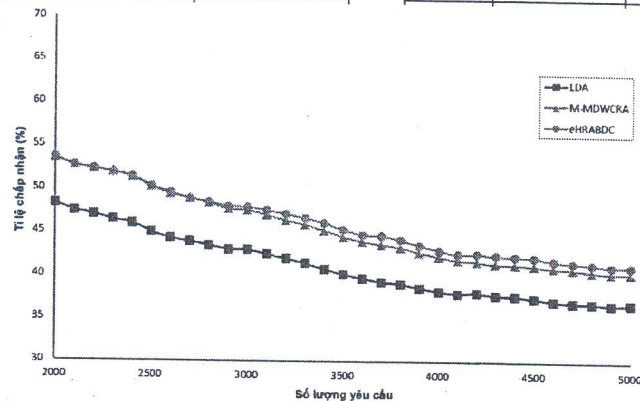
Bảng 2. Tỉ lệ chấp nhận (%) – thời gian tính toán trung bình (ms) theo số yêu cầu trên sơ đồ MIRA.

(a) Điều kiện tải bình thường

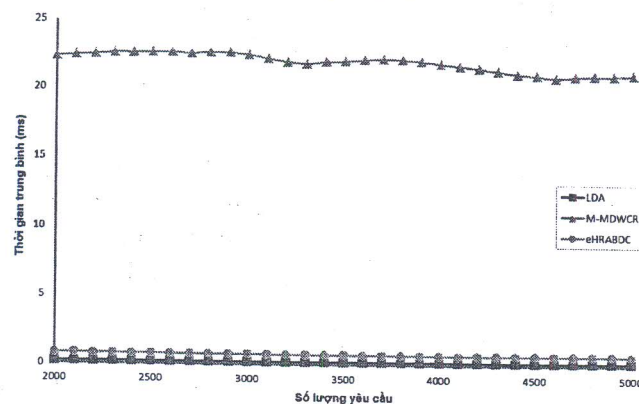
	LDA	M-MDWCRA	eHRABDC
500	100-0,48	100-15,36	100-1,55
1000	91,5-0,18	94,3-11,21	94,6-1,01
1500	88,8-0,15	91,5-12,59	92,1-1,01
2000	85,9-0,14		

(b) Điều kiện tải cao

	LDA	M-MDWCRA	eHRABDC
500	97,6-0,20	99,0-20,97	99,0-1,20
1500	54,0-0,18	59,3-22,70	59,3-0,92
3500	40,0-0,16	44,4-22,01	45,2-0,67
		3,3-21,18	41,0-0,62



(a) Tỉ lệ chấp nhận yêu cầu

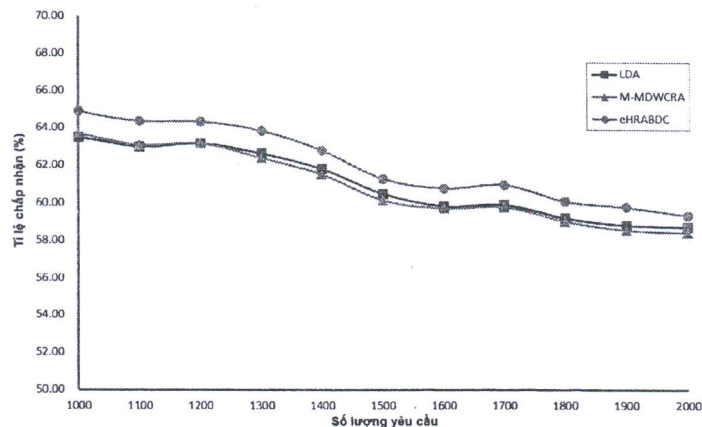


(b) Thời gian tính toán trung bình

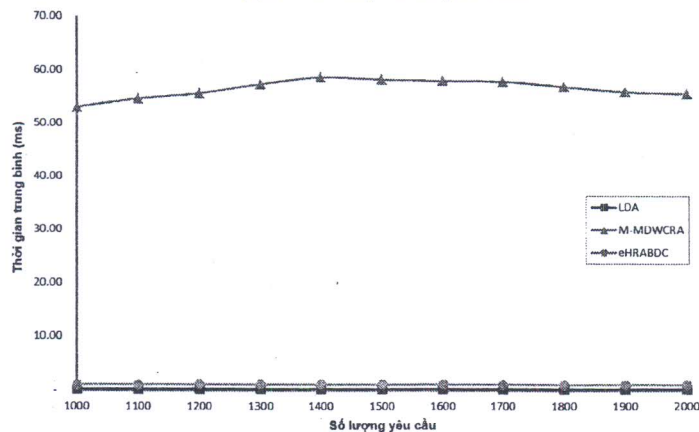
Chuyển sang sơ đồ CESNET, Bảng 3 trình bày kết quả tỉ lệ chấp nhận yêu cầu và thời gian tính toán trung bình trong hai điều kiện tải với giá trị băng thông yêu cầu trong $\{90, 110, 130, 150\}$ và độ trễ yêu cầu trong $[50, 80]$. Khác với sơ đồ MIRA, sơ đồ CESNET được xây dựng từ mô hình mạng thực tế đã được cấu hình tối ưu cho thuật toán tìm đường đi ngắn nhất (độ trễ nhỏ). Vì vậy, thuật toán LDA có kết quả tỉ lệ chấp nhận tương đối tốt. Mặt khác, trong một sơ đồ mạng thực tế, sự cải thiện tỉ lệ chấp nhận yêu cầu của eHRABDC so với LDA là có ý nghĩa dù giá trị tỉ lệ chênh lệch không lớn. Hình 3 so sánh rõ hơn kết quả thực nghiệm này, trong đó eHRABDC có tỉ lệ chấp nhận cao nhất (59,3%), kế đến là LDA và M-MDWCRA (lần lượt 58,7% và 58,4%). Mặt khác, thời gian tính của eHRABDC cũng so sánh được với thời gian tính của LDA (1 ms và 0,18 ms), trong khi kết quả của M-MDWCRA rất cao (55,6ms).

Bảng 3. Tỉ lệ chấp nhận (%) – thời gian tính toán trung bình (ms) theo số yêu cầu trên sơ đồ CESNET.

(a) Điều kiện tải bình thường				(b) Điều kiện tải cao			
	LDA	M-MDWCRA	eHRABDC		LDA	M-MDWCRA	eHRABDC
100	91,0-0,48	91,0-58,10	92,0-1,99	500	66,8-0,26	66,8-43,70	67,0-1,10
1000	63,5-0,21	63,7-53,11	64,9-1,05	1500	62,7-0,17	62,8-41,36	63,4-1,03
1500	60,4-0,19	60,1-58,22	61,2-1,07	3500	61,0-0,15	61,1-41,82	61,8-0,94
2000	58,7-0,18	58,4-55,61	59,3-1,01	5000	60,0-0,16	60,2-41,97	60,9-0,90



(a) Tỉ lệ chấp nhận yêu cầu



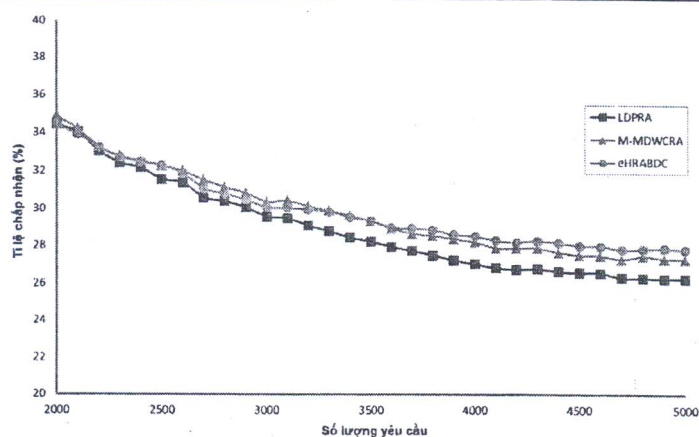
(b) Thời gian tính toán trung bình

Hình 3. So sánh kết quả thực nghiệm trên sơ đồ CESNET điều kiện tải bình thường.

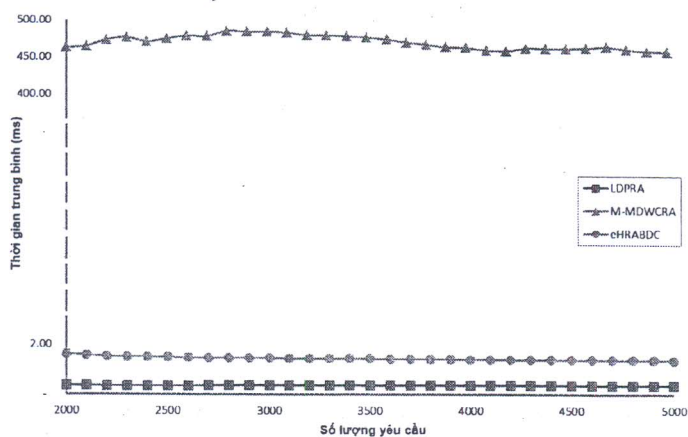
Thuật toán đề xuất eHRABDC cũng có kết quả thực nghiệm tốt trên sơ đồ ANSNET. Bảng 4 là một ví dụ kết quả với yêu cầu băng thông $\{90, 110, 130, 150\}$ và độ trễ $[100, 155]$. Trong điều kiện tải cao, tỉ lệ chấp nhận yêu cầu định tuyến của eHRABDC cao hơn M-MDWCRA 0,5% và hơn LDA 1,54% (Hình 4(a)). Thời gian tính toán trung bình cũng có xu hướng tương tự thực nghiệm trên các sơ đồ khác, trong đó LDA có kết quả thấp nhất (0,37ms), kế đến là eHRABDC (1,38ms) và thời gian của M-MDWCRA đặc biệt cao (457 ms) (Hình 4(b)). Lí do M-MDWCRA có thời gian tính toán rất cao trên sơ đồ ANSNET vì kích thước mạng lớn, độ trễ yêu cầu δ cao trong khi phương pháp tìm đường Dijkstra mở rộng mà M-MDWCRA áp dụng xét $(\delta + 1)$ mức cho mỗi nút mạng. Phần kế tiếp sẽ thực nghiệm và so sánh thêm về phương pháp tìm đường Dijkstra mở rộng và enhanced-heuristic-Dijkstra.

Bảng 4. Tỉ lệ chấp nhận (%) – thời gian tính toán trung bình (ms) theo số yêu cầu trên sơ đồ ANSNET.

(a) Điều kiện tải bình thường				(b) Điều kiện tải cao			
	LDA	M-MDWCRA	eHRABDC		LDA	M-MDWCRA	eHRABDC
100	100-0,66	100-481,89	100-4,50	500	74,2-0,41	74,2-431,06	74,2-3,56
1000	88,9-0,32	88,5-404,82	89,0-3,72	1500	39,8-0,36	40,5-486,49	40,1-1,96
1500	84,6-0,30	85,0-405,34	85,3-3,47	3500	28,2-0,36	29,3-476,67	29,4-1,43
2000	83,0-0,29	83,5-411,04	84,2-3,01	5000	26,2-0,37	27,3-457,38	27,9-1,38



(a) Tỉ lệ chấp nhận yêu cầu

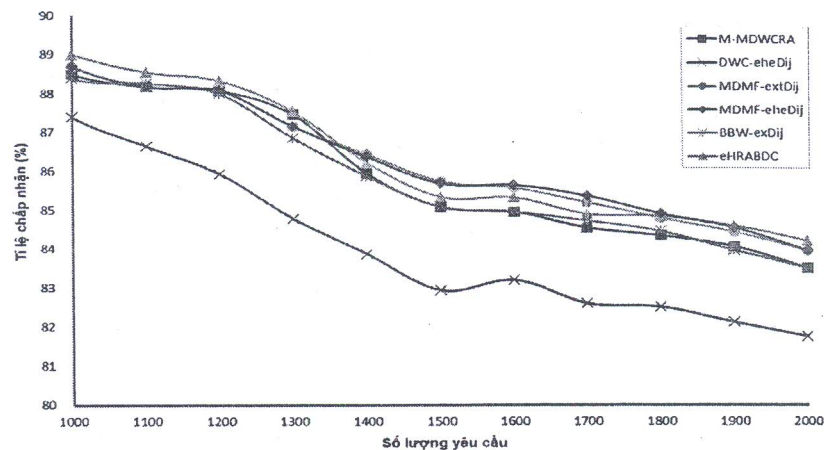


(b) Thời gian tính toán trung bình

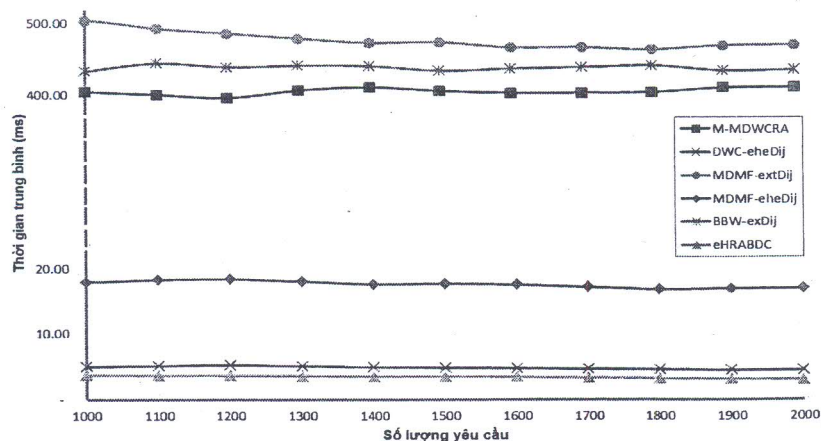
Hình 4. So sánh kết quả thực nghiệm trên sơ đồ ANSNET điều kiện tải cao.

4.3. Thử nghiệm phương pháp tính trọng số và tìm đường đi

Bên cạnh việc so sánh các thuật toán định tuyến như trên, phương pháp tính trọng số và phương pháp tìm đường heuristic thỏa điều kiện độ trễ (enhanced-heuristic-Dijkstra) của eHRABDC còn được đánh giá thêm bằng cách kết hợp với các phương pháp của thuật toán khác. Cụ thể, ba cách tính trọng số của thuật toán M-MDWCRA (tính dựa trên giá trị delay weighted capacity - DWC), MDMF (dựa trên minimum delay và maximum flow - MDMF), và eHRABDC (dựa trên băng thông bandwidth-based weight - BBW) lần lượt được áp dụng cho hai cách tìm đường Dijkstra mở rộng và enhanced-heuristic-Dijkstra tạo thành sáu phương pháp định tuyến đảm bảo băng thông và độ trễ: M-MDWCRA, DWC-eheDij, MDMF-exDij, MDMF-eheDij, BBW-exDij, và eHRABDC. Lưu ý, thuật toán MDMF hoạt động trên cơ sở độ trễ phụ thuộc vào băng thông còn lại của đường đi [8], trong khi thực nghiệm được giới hạn với thông số độ trễ cố định. Vì vậy, chưa thể so sánh thuật toán HRABDC và MDMF mà chỉ so sánh một phần phương pháp tính trọng số thông qua tỉ lệ chấp nhận yêu cầu. Hình 5 thể hiện kết quả tỉ lệ chấp nhận yêu cầu và thời gian tính toán trung bình đối với bộ dữ liệu yêu cầu định tuyến điều kiện tải bình thường trên sơ đồ ANSNET đã trình bày ở Bảng 4(a).



(a) Tỉ lệ chấp nhận yêu cầu



(b) Thời gian tính toán trung bình

Hình 5. So sánh kết quả các phương pháp kết hợp trên sơ đồ ANSNET điều kiện tải bình thường.

Kết quả thực nghiệm thể hiện rõ thời gian tìm đường của Dijkstra mở rộng cao gấp nhiều lần enhanced-heuristic-Dijkstra. Cụ thể, cùng cách tính trọng số DWC, thuật toán M-MDWCRA (sử dụng Dijkstra mở rộng) mất trung bình 411 ms trong khi DWC-eheDij (sử dụng enhanced-heuristic-Dijkstra) chỉ mất 4,3 ms. Tương tự, thời gian tính trung bình của MDMF-exDij cũng cao gấp 27 lần của MDMF-eheDij (Hình 5(b)). Nguyên nhân của sự khác biệt này là vì phương pháp Dijkstra mở rộng xét $(\delta + 1)$ mục cho mỗi nút mạng, còn enhanced-heuristic-Dijkstra chỉ xét một mục. Ngoài ra, trong phạm vi bài toán định tuyến đảm bảo độ trễ, enhanced-heuristic-Dijkstra không tìm được đường đi trọng số nhỏ nhất nhưng vẫn không làm giảm tỉ lệ chấp nhận yêu cầu. Hình 5(a) cho thấy tỉ lệ chấp nhận của eHRABDC và BBW-exDij (cùng cách tính trọng số BBW) cũng như của MDMF-eheDij và MDMF-exDij (cùng cách tính trọng số MDMF) tương đương nhau (khoảng 83 - 84%).

Mặt khác, công thức tính trọng số chỉ dựa trên băng thông liên kết của eHRABDC không chỉ đơn giản, mất ít thời gian tính toán mà còn cho tỉ lệ chấp nhận so sánh được với các phương pháp khác phức tạp hơn. Ví dụ, eHRABDC có tỉ lệ chấp nhận yêu cầu cao nhất (84,2 %) trong kết quả trình bày ở Hình 5(a); BBW-exDij cũng cho kết quả tốt (83,5 %) tương đương với các thuật toán còn lại. Trong khi đó, thời gian tính trọng số theo BBW thấp hơn hai phương pháp DWC và MDMF khi áp dụng với enhanced-heuristic-Dijkstra. Cụ thể, eHRABDC mất trung bình 3 ms cho một yêu cầu định tuyến so với 4,3 ms của DWC-eheDij và 16,9 ms của MDMF-eheDij. Ngoài ra, dựa trên kết quả thực nghiệm có thể nhận xét việc tính trọng số dựa trên hai tập liên kết tới hạn theo maxflow-mincut và theo độ trễ của thuật toán MDMF mất nhiều thời gian tính toán nhưng tỉ lệ chấp nhận yêu cầu không tốt nhất.

5. KẾT LUẬN

Bài báo này đề xuất một thuật toán định tuyến đảm bảo băng thông và độ trễ mới. Khi yêu cầu định tuyến đến, trọng số liên kết được tính một cách đơn giản nhưng hiệu quả dựa trên dung lượng và băng thông còn lại. Mục đích của việc tính toán trọng số là tìm đường đi một cách linh hoạt dựa trên trọng số thay đổi. Tuy nhiên, trong bài toán định tuyến đảm bảo độ trễ, không bắt buộc đường đi tìm được phải có trọng số nhỏ nhất, nên thuật toán Dijkstra đã được đề xuất thay đổi một cách heuristic nhằm tìm đường đi thỏa yêu cầu độ trễ và có trọng số nhỏ. Thực nghiệm trên ba sơ đồ mạng với các tham số, các bộ yêu cầu định tuyến khác nhau chứng tỏ thuật toán đề xuất eHRABDC cho kết quả tỉ lệ chấp nhận yêu cầu và thời gian tính toán trung bình tốt hơn các công trình đã công bố.

Trong tương lai, thuật toán eHRABDC cần được so sánh, đánh giá với công trình MDMF trong điều kiện độ trễ thay đổi. Ngoài ra, phương pháp enhanced-heuristic-Dijkstra cũng có thể được cải tiến nhằm tăng hiệu quả định tuyến hơn nữa.

TÀI LIỆU THAM KHẢO

1. Mark Yampolskiy, Wolfgang Fritz, Wolfgang Hommel - Managing Network Quality of Service in Current and Future Internet, Network and Traffic Engineering in Emerging Distributed Computing Applications, IGI Global, 2012, pp. 179–223.
2. Cao Thai Phuong Thanh, Tran Cong Hung - Bandwidth guaranteed routing algorithms for traffic engineering: An application case on MPLS networks, Journal of Science and Technology **50** (2012) 75–86.
3. Zheng Wang, Crowcroft, J. - Bandwidth-delay based routing algorithms, Global Telecommunications Conference, GLOBECOM, IEEE, 1995, pp. 2129–2133.

4. Yang, Y., Zhang, L., Muppala, J.K., Chanson, S.T. - Bandwidth-delay constrained routing algorithms, *Computer Networks* **42** (2003) 503–520.
5. Kotti, A., Hamza, R., Bouleimen, K. - Bandwidth Constrained Least Delay Least Cost Routing Algorithm, *The International Conference on “Computer as a Tool”, EUROCON*, IEEE, 2007, pp. 1005–1008.
6. Kar, K., Kodialam, M., Lakshman, T.V. - Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications, *IEEE Journal on Selected Areas in Communications* **18** (2000) 2566–2579.
7. Shigang Chen - Routing Support for Providing Guaranteed End-to-End Quality-of-Service, PhD thesis, University of Illinois at Urbana-Champaign, USA (1999).
8. Soorki, M.N., Rostami, H. - Label switched protocol routing with guaranteed bandwidth and end to end path delay in MPLS networks, *Journal of Network and Computer Applications* **42** (2014) 21 – 38.
9. Stiliadis, D., Varma, A. - Latency-rate servers: a general model for analysis of traffic scheduling algorithms, *IEEE/ACM Transactions on Networking* **6** (1998) 611–624.
10. Václav Novák, Petr Adamec, Pavel Šmrha, Josef Verich - CESNET2 IP/MPLS backbone network design and deployment in 2008, *Network Study III, Czech Education and Science Network (CESNET)* (2009).

ABSTRACT

A HEURISTIC APPROACH FOR BANDWIDTH-DELAY GUARANTEED ROUTING PROBLEM

Cao Thai Phuong Thanh^{1,*}, Ha Hai Nam², Tran Cong Hung³

¹*Saigon University, 273 An Duong Vuong Street, Dist. 5, Ho Chi Minh City*

²*Posts and Telecommunications Institute of Technology, 122 Hoang Quoc Viet Rd., Hanoi*

³*Posts and Telecommunications Institute of Technology Ho Chi Minh City Branch, 11 Nguyen Dinh Chieu Street, Dist. 1, Ho Chi Minh City*

*Email: ctphanh@sgu.edu.vn

This paper introduces a novel routing algorithm with two quality of service constraints: minimum bandwidth and maximum delay. The problem is complicated because the former constraint is a concave metric while the later is additive. In order to select routes dynamically, the proposed algorithm reactively and simply calculates link weights based on link bandwidths. Moreover, a heuristic idea is introduced to find routing paths satisfying delay constraints and having as little weight as possible. Simulated experiments on various networks demonstrate that the proposal have better performance than existing solutions with respect to acceptance ratio and average computing time.

Keywords: quality of service routing, guaranteed bandwidth-delay routing, heuristic algorithm.

TẠP CHÍ KHOA HỌC VÀ CÔNG NGHỆ

Journal of Science and Technology

Chuyên san Điện tử, Viễn thông và Công nghệ thông tin

Tập 52, số 6C, năm 2014

MỤC LỤC

CONTENTS

	Trang
1 Pham Thi Thuy Hien, Dang The Ngoc, Vu Tuan Lam - Ber performance of all-optical AF dual-hop FSO systems over gamma-gamma atmospheric turbulence channels. Hiệu năng BER của hệ thống FSO hai chặng toàn quang qua kênh nhiễu loạn không khí GAMMA-GAMMA.	1
2 Nguyen Anh Vinh, Nguyen Nam Tran, Nguyen Xuan Ha, and Nguyen Huu Phuong - Performance improvement for correlated multi-hop MIMO channels. Nâng cao chất lượng kênh truyền MIMO tương quan đa chặng.	12
3 Trung-Hien Nguyen, Tien Ban Nguyen, and Trung Hieu Bui - A study on spurious-free dynamic range (SFDR) enhancement of an analog-optical microwave link. Một nghiên cứu về cải thiện dải phổ động (SFDR) trong liên kết sóng vi ba quang-tương tự.	23
4 Nguyen Van Quan, Nguyen Vu Thang, Vo Le Cuong - Low cost FPGA implementation of adaptive rank order filter. Thực hiện bộ lọc theo bậc thích nghi trên FPGA chi phí thấp.	37
5 Le Duc Hung, Nguyen Dang Nhat Tam, Bui Trong Tu - Parameter extraction for EKV 2.6 MOSFET model based on genetic algorithm. Trích xuất tham số cho mô hình EKV 2.6 sử dụng thuật toán di truyền.	46
6 Do Thi Bích Ngọc - Numerical feature-tree for testing. Mở rộng cây đặc trưng cho kiểm thử.	57
7 Pham Quang Thai - Minimize standard deviation of main beam's error in optical beamformer utilizing chirped grating by factorial analysis. Tối thiểu hóa sai số quân phương của sai số hướng truyền cho hệ thống điều khiển búp sóng quang học sử dụng cách tử Chirp bằng phân tích Factorial.	67
8 Cao Thái Phương Thanh, Hà Hải Nam, Trần Công Hùng - Một cách tiếp cận Heuristic cho bài toán định tuyến đảm bảo băng thông và độ trễ. A Heuristic approach for Bandwidth-delay guaranteed routing problem.	80
9 Vũ Thị Thúy Hà, Lê Hữu Lập, Lê Nhật Thăng - Xây dựng mô hình Chord-DHT phân cấp tối ưu hỗ trợ dịch vụ trên nền P2P. Model construction of optimal Chord-DHT hierarchical overlay network supporting P2P service.	94

- 10 **Vĩnh Hào, Tống Văn Tuất, Lê Tấn Hùng** - Chế tạo máy thu tín hiệu tần số rất thấp để nghiên cứu nhiễu loạn tầng điện ly tức thời. 106
Manufacture of VLF receiver to study the sudden ionospheric disturbance.
- 11 **Hà Quốc Anh, Nguyễn Quốc Định** - Đề xuất một cấu trúc anten dạng vòng dùng cho thiết bị di động 3G. 121
Proposal of a structure of folded loop antenna for the 3G mobile devices.
- 12 **Hoàng Đức Thọ, Trần Đức Sự** - Thực hiện hộp thể kích thước lớn bằng cách ứng dụng đa thức nguyên thủy kết hợp. 129
Implementation of large-scale S-box by using composition primitive polynoms.
- 13 **Nguyễn Chiến Trinh** - Giải pháp đảm bảo QoS cho từng luồng lưu lượng. 142
An approach to guarantee flow QoS.