

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Đào Văn Toán

**TÌM KIẾM NGẪU NHIÊN TRÊN CÁC MẠNG
NGANG HÀNG PHI CẤU TRÚC**

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

HÀ NỘI - 2010

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Đào Văn Toán

**TÌM KIẾM NGẪU NHIÊN TRÊN CÁC MẠNG
NGANG HÀNG PHI CẤU TRÚC**

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS. Nguyễn Đại Thọ

HÀ NỘI - 2010

LỜI CẢM ƠN

Để có thể hoàn thành được khóa luận có kết quả như ngày hôm nay, ngoài sự nỗ lực của chính bản thân, tôi còn nhận được sự giúp đỡ từ Nhà trường, thầy cô, gia đình và bạn bè, đó là điều may mắn đối với tôi, và cũng là niềm hạnh phúc.

Đầu tiên, em chân thành cảm ơn giảng viên, tiến sĩ Nguyễn Đại Thọ, người đã hướng dẫn trực tiếp cho em làm khóa luận này. Thầy đã giành cho em nhiều thời gian để thảo luận về vấn đề nghiên cứu, nhiệt tình hỗ trợ em trong việc nhìn nhận, đánh giá vấn đề gặp phải và phát triển ý tưởng. Hỗ trợ em trong việc kiểm nghiệm, mô phỏng chương trình để có kết quả đánh giá và góp ý kiến cho em thực hiện khóa luận này.

Em xin cảm ơn trường Đại học Công Nghệ- ĐHQG Hà Nội đã tạo điều kiện cho em tham gia học tập, rèn luyện và sinh hoạt trong môi trường tốt, hiện đại. Đặc biệt là tạo điều kiện cho em tham gia thực hiện khóa luận, cho em cơ hội phát huy vốn kiến thức, kỹ năng đã tiếp thu được, cũng như phát huy khả năng nhìn nhận vấn đề khoa học-công nghệ-cuộc sống trong lĩnh vực học tập của mình sau khóa học.

Và lời cảm ơn sâu sắc tôi muốn giành cho gia đình tôi, đặc biệt là bố mẹ tôi, những người vất vả ngày đêm lao động để lo cho tôi có thể hoàn thành tốt khóa học, luôn động viên tôi học tập cho tốt, tạo điều kiện cho tôi về mặt vật chất trong quá trình theo học tại trường.

Cuối cùng, tôi xin gửi lời cảm ơn tới những người bạn của tôi, cảm ơn các bạn đã giúp đỡ tôi khi tôi gặp khó khăn trong học tập, cũng như trong cuộc sống. Đặc biệt để hoàn thành khóa luận này, các bạn còn giành thời gian để thảo luận cùng tôi, giúp tôi thu thập kết quả mô phỏng.

Hà Nội, tháng 5 năm 2010.

Đào Văn Toán

TÓM TẮT NỘI DUNG

Trong các mô hình client-server, mô hình mạng ngang hàng tập trung hay mô hình mạng ngang hàng lai ghép, nếu một người dùng ở trong mạng sử dụng máy tính để tìm kiếm tài nguyên thì việc tìm kiếm là đơn giản bởi sự hỗ trợ của server hoặc siêu điểm nút. Tuy nhiên, với mô hình mạng ngang hàng thuần túy việc tìm kiếm lại không đơn giản, đó là bởi vì điểm nút tìm kiếm không có thông tin vị trí tài nguyên, không có thông tin định tuyến, cũng như thông tin về các điểm nút khác trong mạng, trừ các điểm hàng xóm với nó. Chính bởi những đặc trưng này, đã có nhiều bài báo, công trình nghiên cứu trước đây đề xuất ra giải pháp cải tiến phương pháp tìm kiếm đơn lẻ hay đề xuất phương pháp tìm kiếm kết hợp như là: phương pháp tìm kiếm động [20], phương pháp tìm kiếm lai [14],... Ngoài ra còn có những đề xuất để cải tiến hiệu suất tìm kiếm của các phương pháp tìm kiếm đơn lẻ như trong các tài liệu [16], [17], [23].

Tuy nhiên chưa có bài báo nào đề cập đến việc kết hợp 2 phương pháp tìm đơn lẻ theo trình tự: phương pháp di chuyển ngẫu nhiên trước và phương pháp phát tràn sau. Khóa luận của chúng tôi đề xuất phương pháp tìm kiếm lai ghép mới từ ý tưởng này, sau đó thực hiện mô phỏng các phương pháp trên một số dạng đồ thị chung của mạng ngang hàng thuần túy. Chúng tôi cũng đưa ra các phân tích, đánh giá về các phương pháp tìm kiếm.

Phương pháp của chúng tôi cho kết quả tốt trên đồ thị luật hàm mũ trong một số trường hợp, còn với tô pô phân cụm thì cho kết quả kém hơn nhưng tốt hơn so với phương pháp phát tràn trên đồ thị này.

MỤC LỤC

Bảng ký hiệu viết tắt.....	1
MỞ ĐẦU	1
CHƯƠNG 1. TỔNG QUAN VỀ MẠNG NGANG HÀNG	6
1.1. Thành phần cấu tạo mạng ngang hàng.....	6
1.1.1. Khái niệm điểm nút	6
1.1.2. Cách phân loại peer trong mạng ngang hàng.....	7
1.2. Mạng ngang hàng.....	8
1.2.1. Định nghĩa mạng ngang hàng	8
1.2.2. Phân loại các mô hình mạng ngang hàng.....	11
1.3. Mạng xếp chồng.....	18
CHƯƠNG 2. LÝ THUYẾT ĐỒ THỊ VÀ CÁC DẠNG ĐỒ THỊ MẠNG.....	19
2.1. Khái niệm đồ thị.....	19
2.1.1. Đồ thị có hướng.....	19
2.1.2. Đồ thị vô hướng	19
2.1.3. Các khái niệm khác	20
2.2. Các dạng đồ thị trong mạng ngang hàng	20
2.2.1. Đồ thị ngẫu nhiên	21
2.2.2. Đồ thị luật hàm mũ.....	21
2.2.3. Tô pô phân cụm.....	22
CHƯƠNG 3. CÁC PHƯƠNG PHÁP TÌM KIẾM ĐÃ ĐỀ XUẤT TRƯỚC ĐÂY	24
3.1. Các phương pháp tìm kiếm đơn lẻ.....	24
3.1.1. Phương pháp tìm kiếm phát tràn thông thường	24
3.1.2. Phương pháp tìm kiếm di chuyển ngẫu nhiên.....	25
3.2. Các phương pháp tìm kiếm kết hợp.....	26
3.2.1. Phương pháp tìm kiếm động	27
3.2.2. Phương pháp tìm kiếm lai	27
CHƯƠNG 4. CÁC PHƯƠNG PHÁP TÌM KIẾM LAI GHÉP CỦA CHÚNG TÔI	30

4.1. Phương pháp tìm kiếm lai ghép sử dụng phát tràn thông thường.....	30
4.1.1. Phương pháp tìm kiếm lai ghép biến thể thứ nhất	30
4.1.2. Phương pháp tìm kiếm lai ghép biến thể thứ hai	34
4.2. Phương pháp tìm kiếm lai ghép sử dụng phát tràn cải tiến	37
4.2.1. Phương pháp tìm kiếm lai ghép biến thể thứ ba	38
4.2.2. Phương pháp tìm kiếm lai ghép biến thể thứ tư.....	41
CHƯƠNG 5. MÔ PHÒNG VÀ ĐÁNH GIÁ HIỆU NĂNG	46
5.1. Các đơn vị đo hiệu năng trong mô phỏng.....	46
5.1.1. Mức độ bao phủ.....	46
5.1.2. Tỷ lệ thành công.....	47
5.1.3. Số lượng truy vấn thành công	47
5.1.4. Hiệu quả truy vấn	48
5.1.5. Số lượng nút nhận truy vấn dư thừa.....	48
5.2. Kết quả mô phỏng trên đồ thị luật hàm mũ	49
5.2.1. Đồ thị luật hàm mũ với 5^5 thông báo truy vấn	49
5.2.2. Đồ thị luật hàm mũ với N thông báo truy vấn	51
5.3. Kết quả mô phỏng trên tô pô phân cụm.....	53
5.3.1. Mô phỏng trên tô pô phân cụm với 5^5 thông báo truy vấn	53
5.3.2. Mô phỏng trên tô pô phân cụm với N thông báo truy vấn.....	55
5.4. Đánh giá về phân bố thông báo truy vấn.....	61
CHƯƠNG 6. KẾT LUẬN VÀ ĐỊNH HƯỚNG	65
TÀI LIỆU THAM KHẢO	67

Bảng ký hiệu viết tắt

ARPANET	Advanced Research Projects Agency Network
BFS	Breadth-First Search
CPU	Control Processing Unit
DFS	Depth-First Search
GUID	General Unique ID
FTP	File Transfer Protocol
Telnet	Telecommunication Network
TTL	Time-to-Live

MỞ ĐẦU

Thế hệ mạng Internet đầu tiên có tên là mạng ARPANET, mạng này được phát triển từ dự án của Bộ quốc phòng Mỹ vào những năm cuối của thập niên 1960. Mục đích của mạng ARPANET là dùng để chia sẻ các tài nguyên tính toán và các tài liệu giữa các trung tâm nghiên cứu khác nhau trên nước Mỹ. Mô hình đầu tiên của mạng chỉ có 4 máy, những máy này được đặt tại các địa điểm khác nhau là: Trường Đại học California, trung tâm nghiên cứu phát triển của Học viện nghiên cứu Stanford, trường Đại học California tại Santa Barbara và Đại học Utah. Các máy trong mạng ARPANET đầu tiên không có đặc trưng gì giống như client hay server, chúng được xem là ngang hàng nhau vì vậy mạng này còn được gọi là mạng ngang hàng đầu tiên.

Các ứng dụng đầu tiên và vượt trội trên mạng Internet là: FTP và Telnet...vv..nhưng bản thân chúng lại là các ứng dụng client-server, sau khi mạng Internet xuất hiện thì các ứng dụng phát triển cho mạng chủ yếu là ứng dụng cho mô hình mạng client-server. Ngày nay, các ứng dụng mạng ngang hàng cũng trở nên phổ biến hơn và ngày càng đa dạng như là: BitTorrent, Skype, FlashGet, Gnutella, Sopcast, Napster...vv. Sự trở lại và phát triển của các ứng dụng mạng ngang hàng là vì sự tồn tại của mô hình mạng client-server có nhiều hạn chế. Điều đó có thể thấy rõ ràng, server không thể lưu tất cả các thông tin mà client yêu cầu được bởi vì vấn đề lưu trữ có hạn và khi số lượng client tăng đến mức độ nào đó thì nhu cầu về tải, băng thông tăng lên dẫn đến việc các server không có khả năng cung cấp dịch vụ cho các client tham gia vào, chi phí để mở rộng mạng là tốn kém. Tuy nhiên, với mô hình mạng ngang hàng có thể giải quyết được những vấn đề này, ngoài ra còn tận dụng được sức mạnh tập thể của các máy tham gia trong việc tính toán, dễ dàng mở rộng và chi phí thấp.

Mạng ngang hàng có nhiều tiêu chí để phân loại nhưng phân loại một cách tương đối dựa trên đặc điểm cấu trúc của mạng thì phân chia thành 2 loại : loại có cấu trúc, và loại không có cấu trúc. Những mạng ngang hàng không có cấu trúc còn được phân chia tiếp thành 3 loại: mạng ngang hàng tập trung, mạng ngang hàng thuần túy, mạng ngang hàng lai. Trong khóa luận của chúng tôi, chúng tôi tập trung vào các mô hình mạng ngang hàng thuần túy.

Hiện tại, để tìm kiếm thông tin hay tài nguyên trên Internet, hầu hết người sử dụng thường thông qua các trình duyệt để truy cập tới các server cung cấp dịch vụ tìm kiếm

như Google, Bing..vv..sau đó người sử dụng sẽ gửi yêu cầu tìm kiếm của mình lên đó. Khi tìm kiếm với Google, người dùng sẽ nhận được hàng nghìn kết quả, có cả những kết quả chẳng liên quan gì đến thông tin mà người dùng cần, thậm chí có cả những kết quả đã quá cũ và không còn tồn tại, hay cả những kết quả không có giá trị. Điều này làm cho người dùng có quá nhiều thông tin lựa chọn không cần thiết và dễ gây lẫn lộn. khó chịu. Tuy việc tìm kiếm cho kết quả nhanh nhưng những máy tìm kiếm này vẫn còn nhiều nhược điểm khác như là: vấn đề yêu cầu nhiều phần cứng để hỗ trợ lưu trữ thông tin và tài nguyên bổ sung, vấn đề khi máy chủ tìm kiếm đột nhiên tạm ngưng hoạt động, vấn đề khi mà kích thước mạng tăng lên trong khi số lượng máy hỗ trợ cho dịch vụ tìm kiếm là có hạn, vấn đề các tài nguyên chỉ được phép lưu hành trong nội bộ..vv.. Nhưng một dịch vụ tìm kiếm tương tự mà được cài đặt trên mạng ngang hàng thì có thể giải quyết được các vấn đề với kết quả tìm kiếm trả về, ngoài ra còn có nhiều lợi thế khác như là: hạn chế kết quả không cần thiết, không lo hiện tượng máy chủ bị ngưng hoạt động, không lo vấn đề kích thước mạng tăng...vv., thông tin có thể tham khảo thêm trong tài liệu [3].

Các ứng dụng chia sẻ tài nguyên phổ biến của mạng ngang hàng vào thời điểm hiện tại như là: BitTorrent, Napster,...vv. các ứng dụng này thuộc mô hình mạng ngang hàng tập trung và mạng ngang hàng lai. Việc tìm kiếm tài nguyên với các mô hình này là đơn giản và việc tìm kiếm giống như tìm kiếm trong mô hình client-server bởi vì được hỗ trợ bởi máy chủ tìm kiếm trung tâm hay siêu điểm nút (SuperPeer hay SuperNode) do đó tìm kiếm không phải là vấn đề đối với các mô hình mạng ngang hàng này. Nhưng mô hình mạng ngang hàng thuần túy không tồn tại máy chủ tìm kiếm trung tâm hay các siêu điểm nút để lưu trữ thông tin về các tài nguyên được các điểm nút khác trong mạng chia sẻ. Do đó mạng ngang hàng thuần túy là một mô hình mạng đặc biệt và việc tìm kiếm là vấn đề quan trọng với mạng này.

Nếu một công ty hay tổ chức xây dựng mô hình mạng theo kiểu mô hình mạng ngang hàng thuần túy thì cần thiết có một ứng dụng để hỗ trợ những người dùng máy trong hệ thống mạng có thể tìm kiếm các tài nguyên chia sẻ trong tổ chức, công ty. Các tài nguyên chia sẻ này có thể là: âm nhạc, phim, ảnh, tác phẩm văn học, không gian lưu trữ, thiết bị đắt tiền hay thông tin du lịch, thông tin hội họp..vv.. của các thành viên trong công ty, tổ chức chia sẻ.

Để đáp ứng việc tìm kiếm tài nguyên trên mô hình mạng này có một số phương pháp được đề xuất như là phương pháp phát tràn (hay lan tỏa) và bước dịch chuyển

ngẫu nhiên, những phương pháp này chúng tôi gọi là nhóm phương pháp đơn lẻ phổ biến. Ngoài ra có một vài công trình nghiên cứu đề xuất về tìm kiếm trước đây, các công trình này đề xuất các phương pháp tìm kiếm kết hợp 2 phương pháp đơn lẻ, đó là: phương pháp tìm kiếm động [19], phương pháp tìm kiếm lai [5], phương pháp tìm kiếm lai [14], ...vv. Phương pháp lai trong tài liệu [14] thực hiện như sau: phát tràn trước, rồi sau đó thực hiện di chuyển ngẫu nhiên trên các nút phát tràn tìm được. Tất cả các phương pháp kết hợp được đề xuất trước đây là có sự kết hợp của cả phương pháp phát tràn và phương pháp di chuyển ngẫu nhiên nhưng đều được xây dựng theo tiêu chí phạm vi tìm kiếm, tùy theo phạm vi và cách thức mà có sự kết hợp thỏa mãn.

Đối với mô hình mạng thuần túy do đặc trưng cấu trúc của mạng và bởi vì việc lưu trữ tài nguyên là ngẫu nhiên, bất kỳ trên các nút trong mạng khi đó các phương pháp tìm kiếm được sử dụng dựa trên phạm vi chỉ mang tính ước lượng và rất khó để chọn lựa giá trị chính xác phạm vi là bao nhiêu cho hợp lý. Nói chung việc tìm kiếm các tài nguyên trong mô hình mạng thuần túy vẫn là tìm kiếm ngẫu nhiên bởi các thông tin tìm kiếm không được biết trước. Cách thức tìm kiếm có thể là sử dụng phương pháp tìm kiếm mù đơn thuần hoặc là có sự kết hợp của nhiều phương pháp tìm kiếm mù.

Giả sử trong trường hợp chúng ta phát tràn toàn bộ phạm vi có thể của phát tràn nhưng chưa có tài nguyên cần tìm, sau đó lại phải mất vài lần di chuyển ngẫu nhiên mới thấy, nếu như làm ngược lại thì sẽ có hiệu quả thế nào, như vậy đây cũng là một trong những trường hợp cần xem xét. Việc thực hiện thứ tự ngược lại sẽ có trình tự tìm kiếm là: thực hiện di chuyển ngẫu nhiên số chẳng bằng với lượng phát tràn trên, sau đó thực hiện phát tràn vài bước tiếp theo thì sẽ không làm tăng tải cho các nút khác và không gây tổn bằng thông chung toàn mạng. Dĩ nhiên giả thiết chung cho các phương pháp tìm kiếm vẫn là không thể biết vị trí nào có tài nguyên, không có thông tin định tuyến tới các nút khác trong mạng trừ nút hàng xóm.

Đồng thời trong các phương thức đề xuất trước đây chưa có phương thức nào sử dụng cho phương pháp di chuyển ngẫu nhiên trước, rồi sau đó sử dụng phương pháp phát tràn. Vì vậy chúng tôi đề xuất xây dựng phương pháp tìm kiếm lai ghép của mình dựa trên ý tưởng đó, không chỉ đề xuất phương pháp tìm kiếm chúng tôi còn phân tích, đánh giá cùng với các phương pháp tìm kiếm khác dựa trên các tiêu chí đánh giá để có thể thấy được phương pháp tìm kiếm nào cho hiệu quả tốt, phương pháp nào không hiệu quả.

Kết quả sau mô phỏng cho thấy phương pháp tìm kiếm lai ghép biến thể thứ ba và thứ nhất của chúng tôi cũng cho kết quả tốt không kém gì phương pháp phát tràn trên đồ thị luật hàm mũ với lượng thông báo 5^5 , sinh ra số lượng nút trùng lặp thông báo là thấp hơn. Còn với lượng thông báo truy vấn N (N là số rất lớn) thì phương pháp lai ghép biến thể thứ ba và phương pháp lai đề xuất trong [14] và phương pháp lai ghép biến thể thứ tư cho kết quả tốt, tuy nhiên các phương pháp lai cho kết quả số lượng nút bị trùng lặp thông báo truy vấn cao.

Trên tô pô phân cụm thì phương pháp di chuyển ngẫu nhiên vẫn là tốt nhất đối với lượng thông báo truy vấn nhỏ và lớn, phương pháp lai trong tài liệu [14] cũng là phương pháp tốt, còn các phương pháp đề xuất của chúng tôi cho giá trị tốt hơn phương pháp tràn nhưng vẫn là phương pháp tồi. Nhưng số nút nhận thông báo truy vấn dư thừa trong phương pháp của chúng tôi là ít hơn. Ngoài ra chúng tôi còn đánh giá mẫu 2 phương pháp về mức độ phân bố tải, đánh giá để nhìn nhận tổng quan về kết quả của các phương pháp.

Phần tiếp theo của khóa luận được tổ chức như sau:

Chương 1: Tổng quan về mạng ngang hàng. Trong chương này, chúng tôi giới thiệu một cách tổng quan các kiến thức liên quan đến mạng ngang hàng như là khái niệm về điểm nút (peer), khái niệm mạng ngang hàng, các mô hình mạng ngang hàng hiện tại.

Chương 2: Lý thuyết đồ thị và các dạng đồ thị mạng. Nội dung chúng tôi trình bày ở trong chương này, tóm lược lý thuyết đồ thị như : khái niệm đồ thị, khái niệm bậc của một đỉnh trong đồ thị, các dạng đồ thị. Và tập trung vào các dạng đồ thị mạng phục vụ cho mô phỏng của chúng tôi như : đồ thị ngẫu nhiên, đồ thị luật hàm mũ, tô pô phân cụm.

Chương 3: Các phương pháp tìm kiếm đã đề xuất trước đây. Trong chương này, chúng tôi trình bày các phương pháp tìm kiếm đã được đề xuất: Các phương pháp đơn lẻ : phương pháp phát tràn, phương pháp di chuyển ngẫu nhiên. Các phương pháp kết hợp của 2 phương pháp đơn lẻ: phương pháp tìm kiếm động, phương pháp lai ghép.

Chương 4: Các phương pháp tìm kiếm lai của chúng tôi. Những đề xuất, hướng giải quyết trong việc kết hợp 2 phương pháp đơn lẻ theo cách của chúng tôi, theo vấn đề chúng tôi đặt ra, được trình bày chi tiết trong chương này.

Chương 5: Mô phỏng và đánh giá hiệu năng. Trong chương này, chúng tôi giới thiệu một vài độ đo để làm cơ sở đánh giá một phương pháp tìm kiếm tốt hay không tốt so với

phương pháp khác. Đó là : mức độ bao phủ, số lượng nút nhận truy vấn dư thừa, tỉ lệ thành công, các truy vấn thành công, và hiệu quả truy vấn.

Kết quả của các mô phỏng cũng được trình bày trong chương này. Và các nhận xét, đánh giá về giá trị của các phương pháp tìm kiếm đã đạt được.

Chương 6: Kết luận và định hướng. Từ những kết quả thu được, ở chương 6, trong chương này chúng tôi đi đến kết luận, đánh giá về đề xuất của chúng tôi so với những đề xuất đã nêu, tìm ra ưu điểm, nhược điểm của phương pháp và từ đó đề xuất hướng phát triển mới, cũng như công việc dự định tiến hành trong tương lai.

Như vậy trong phần này, chúng tôi giới thiệu tổng quan về khóa luận của chúng tôi, cũng như trình tự nội dung vấn đề chúng tôi sẽ trình bày.

CHƯƠNG 1. TỔNG QUAN VỀ MẠNG NGANG HÀNG

Hiện nay, tên gọi “ mạng ngang hàng ” hay “ mạng đồng đẳng ” và các ứng dụng của kiểu mạng này như là: Napster, Skype, BitTorrent, FlashGet, Sopcast, ICQ...vv.. không còn xa lạ gì với người dùng Internet, đặc biệt là những người dùng có nhu cầu về tìm kiếm tài nguyên, chia sẻ tài nguyên như: phần mềm, phim ảnh, âm nhạc, file văn bản. Mạng máy tính gia đình cũng là mạng ngang hàng khi người dùng cấu hình máy tính theo nhóm (WorkGroup), cho phép các máy trong nhóm có thể chia sẻ file, máy in và các tài nguyên, thiết bị khác. Như vậy, sự phổ biến của mạng ngang hàng là rất rộng nhưng hiểu biết về mạng ngang hàng, cũng như mạng ngang hàng bao gồm những thành phần gì? Thế nào được gọi là mạng ngang hàng? Cấu trúc ra sao? Hoạt động ra sao? Có bao nhiêu loại mô hình mạng được gọi là mạng ngang hàng?...vv..thì đa phần người dùng chưa có cái nhìn tổng quan và chi tiết về chúng. Trong chương này, chúng tôi sẽ trình bày về những vấn đề đó, không chỉ để hiểu rõ về mạng ngang hàng mà còn có giúp cho việc tìm hiểu các vấn đề liên quan đến nội dung tiếp theo của chúng tôi.

Đầu tiên, chúng tôi sẽ trình bày về các thành phần trong mạng ngang hàng và khái niệm mạng ngang hàng. Sau đó, chúng tôi sẽ giới thiệu về các loại mô hình mạng ngang hàng, trong đó chúng tôi sẽ tập trung trình bày chi tiết mô hình mạng ngang hàng liên quan đến vấn đề khóa luận của chúng tôi, đó là mạng ngang hàng thuần túy. Một mô hình mạng ngang hàng đặc biệt.

1.1. Thành phần cấu tạo mạng ngang hàng

Trong các mô hình mạng cấu trúc kiểu client-server thì thông thường, cấu trúc của mô hình này bao gồm 2 thành phần chính là: client (máy khách) và server (máy chủ). Tuy nhiên trong mô hình mạng cấu trúc kiểu mạng ngang hàng thì thành phần lại là các điểm nút .

1.1.1. Khái niệm điểm nút

Điểm nút tên tiếng Anh là “peer”. Một điểm nút là một thành phần trong mạng ngang hàng. Nếu đánh giá về mặt trực quan thì có thể thấy điểm nút thường được hiểu là một máy tính, thiết bị di động, máy in, hay thiết bị hỗ trợ cá nhân..vv..Tuy nhiên, không thể định nghĩa điểm nút là một thiết bị như thế vì sẽ không thể hiện được đúng khái niệm về điểm nút. Nếu nhìn nhận điểm nút như là một ứng dụng đang chạy trên một máy tính

đơn và máy tính này được kết nối vào một mạng như mạng Internet, định nghĩa như thế này cũng không thể hiện được chức năng và tất cả những thể hiện có thể của điểm nút được. Như vậy, nếu nhìn nhận phiên diện về điểm nút thì đã làm giảm khả năng của một điểm nút và chỉ thấy nó giống như một thiết bị có thể đáp ứng các điểm nút khác đang chạy trong mạng.

Trong tài liệu [3] đã phát biểu đầy đủ về điểm nút như sau:

“Một thực thể nào đó có khả năng thực hiện chức năng có ích nào đó và truyền đạt các kết quả của chức năng đó tới các thực thể khác trên một mạng, một cách trực tiếp hoặc gián tiếp, được gọi là một điểm nút”.

Như vậy, từ định nghĩa ta có thể thấy một điểm nút nó là một thành phần thiết bị nào đó trong mạng, có thể là máy tính cá nhân, có thể là thiết bị hỗ trợ cá nhân, hay thiết bị di động, router, modem, máy in...vv. Khi các thành phần này tham gia vào mạng thì mỗi thành phần sẽ chạy các ứng dụng nào đó để thực hiện chức năng của mình như: hoặc chia sẻ tài nguyên file, chia sẻ tài nguyên không gian lưu trữ, chia sẻ tài nguyên phần cứng, chia sẻ tiến trình nhân rồi, cung cấp nội dung...vv.hoặc hỗ trợ định tuyến hoặc nhận chia sẻ từ các thiết bị khác trong mạng hoặc tìm kiếm tài nguyên...vv.

Trong một số tài liệu thì khái niệm điểm nút còn được hiểu là nút hoặc nốt .

1.1.2. Cách phân loại peer trong mạng ngang hàng

Theo định nghĩa ở trên “chức năng có ích” là phụ thuộc vào từng loại điểm nút, công việc mà điểm nút đó đảm nhiệm. Do đó có thể phân làm 3 loại điểm nút khác nhau như trong tài liệu [3] đã phân chia:

- Điểm nút thông thường : Tên tiếng Anh là “Simple Peer”, là một điểm nút bình thường trong mạng ngang hàng, nó được thiết kế cho người dùng cuối. Một điểm nút thông thường sẽ cung cấp dịch vụ cho các điểm nút khác và cũng nhận sự cung cấp từ các điểm nút khác trong mạng. Tuy nhiên, trong thực tế một điểm nút thông thường sẽ nằm trong một mạng riêng biệt, hoặc là đằng sau một tường lửa nên các điểm nút khác nằm phía bên ngoài sẽ khó có thể giao tiếp trực tiếp với điểm nút thông thường nằm đằng sau tường lửa.

- Điểm nút gặp gỡ: Tên tiếng Anh là “Rendezvous Peer”, cũng là một loại điểm nút, tuy nhiên nó nằm phía ngoài các mạng riêng biệt, nhiệm vụ của nó là cung cấp các thông tin để tìm kiếm các điểm nút khác và các tài nguyên của điểm nút khác cho các điểm nút thông thường gửi yêu cầu tìm kiếm.

- Điểm nút định tuyến: Tên tiếng Anh là “Router Peer”, là điểm nút cung cấp các cơ chế để chuyển thông tin giao tiếp giữa các điểm nút khác với nhau, cho phép các thông tin này vượt qua các tường lửa hoặc các thiết bị chuyển đổi địa chỉ mạng.

Nhóm điểm nút là: một nhóm các điểm nút có cùng một mục đích hay dịch vụ, thường gọi là Peer Group. Các điểm nút trong một nhóm sẽ chia sẻ thông tin cho các điểm nút khác trong nhóm và các điểm nút ở ngoài nhóm không thể tiếp cận được.

Phần trình bày trên, đã nêu rõ khái niệm về điểm nút, nhóm điểm nút, cũng như cách phân loại điểm nút. Ngoài cách phân loại như trong tài liệu [3] thì phân loại điểm nút như trong các tài liệu khoa học, kỹ thuật khác lại có sự khác biệt, chẳng hạn như trong tài liệu [2], [12] phân chia thành 2 loại điểm nút là: điểm nút thông thường và siêu điểm nút.

- Điểm nút thông thường : là một điểm nút bình thường, một thành phần trong mạng, tham gia vào mạng, khi tham gia vào mạng thì các điểm nút loại này chia sẻ tài nguyên hoặc nhận chia sẻ từ các điểm nút khác trong mạng.

- Siêu điểm nút: Tên tiếng Anh là “Super Peer” hoặc “Super Node”, là một điểm nút trong mạng, chúng có vai trò quản lý điểm nút trong vùng (quản lý về các peer, thông tin tài nguyên mà các peer chia sẻ, thông tin định tuyến...vv..), định tuyến cho các điểm nút trong mạng sang điểm nút ở vùng khác. Nói chung có đầy đủ tính năng của ba loại điểm nút đã mô tả ở trên, tức là nó không lưu trữ tài nguyên mà chỉ cung cấp, quản lý thông tin về tài nguyên, quản lý các điểm nút trong vùng của nó, định tuyến cho các điểm nút sang vùng khác nếu cần thiết và giúp các điểm nút trong vùng trao đổi với các điểm nút ở vùng khác.

1.2. Mạng ngang hàng

1.2.1. Định nghĩa mạng ngang hàng

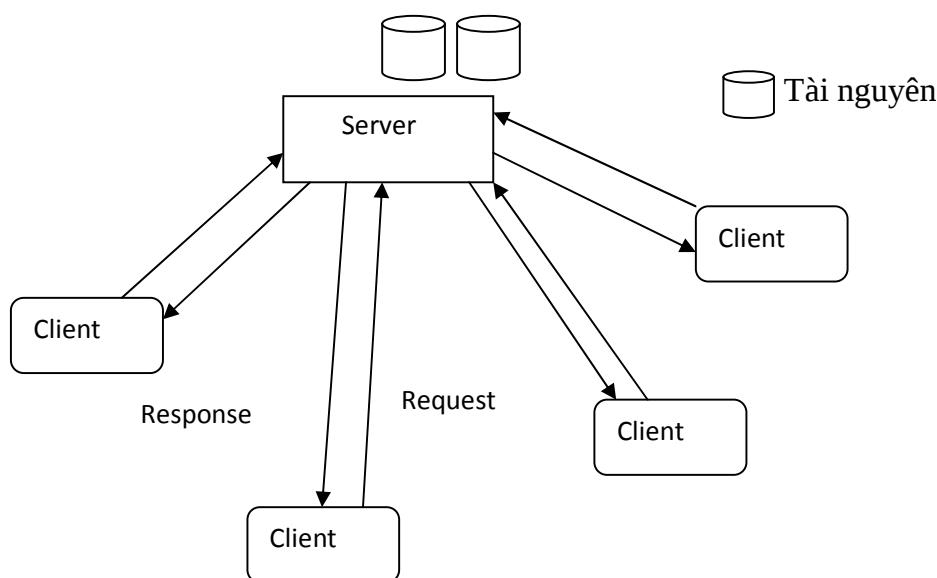
Trong tài liệu tham khảo [2], Oram đã định nghĩa về mạng ngang hàng như sau:

“Mạng ngang hàng là 1 lớp ứng dụng tận dụng ưu điểm của lưu trữ các tài nguyên, các chu trình, nội dung, giá trị hiện diện của con người ở phía rìa của mạng Internet. Bởi vì việc truy cập tới các tài nguyên phi tập trung này giống như đang hoạt động trong một môi trường kết nối không ổn định và các địa chỉ IP không thể đoán trước được, các nút mạng ngang hàng phải hoạt động bên ngoài hệ thống DNS và có quyền tự trị đáng kể hoặc hoàn toàn độc lập với các máy chủ trung tâm”.

Theo định nghĩa này, mạng ngang hàng là một hệ thống phân tán đặc biệt trong tầng ứng dụng, ở đó mỗi cặp điểm nút có thể giao tiếp với nhau thông qua giao thức định tuyến trong các tầng mạng ngang hàng. Mỗi điểm nút giữ 1 đối tượng dữ liệu nào đó có thể là nhạc, ảnh, tài liệu,...vv... Mỗi điểm nút có thể truy vấn tới đối tượng nó cần từ các điểm nút khác thông qua kết nối logic trong tầng mạng ngang hàng.

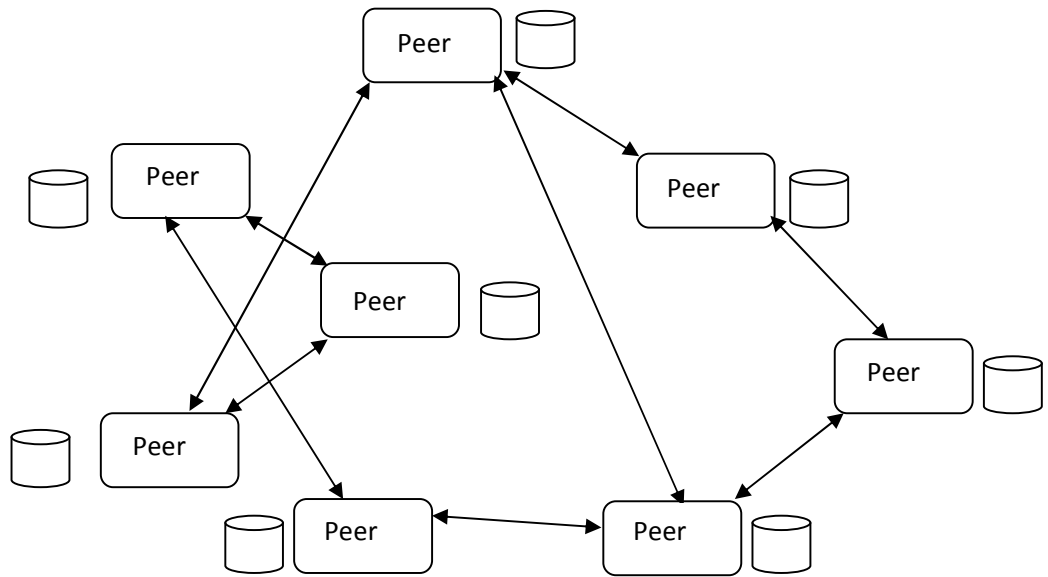
Sau này, Oram và các đồng nghiệp đã đưa ra định nghĩa cơ bản và hoàn chỉnh hơn:

“[a Peer-to-Peer system is] *a self-organizing system of equal, autonomous entities (peers) [which] aims for the shared usage of distributed resources in a networked environment avoiding central services.*” [12].



Hình 1. Mô hình mạng client-server

Trong mô hình mạng client-server như ở Hình 1 bao gồm: máy client và máy server, có nhiều máy client truy cập tới máy server để tìm kiếm tài nguyên. Nhưng trong mạng ngang hàng không có khái niệm máy trạm (client) hay máy chủ (server) mà chỉ có khái niệm các nút (peer) đóng vai trò như cả client và server. Hay còn được gọi là “Servent” được ghép từ “**Server**” và “**Client**”.



Hình 2. Mô hình mạng ngang hàng

Các định nghĩa về mạng ngang hàng ở trên là định nghĩa trừu tượng và tổng quát cho mạng ngang hàng nói chung. Còn đối với mạng máy tính ngang hàng thì định nghĩa theo tài liệu[12] như sau:

Một mạng ngang hàng bao gồm các phần tử máy tính:

- (1) được kết nối bởi 1 mạng máy tính,
- (2) địa chỉ có thể trong 1 phạm vi duy nhất, và
- (3) chia sẻ 1 giao thức truyền thông chung.

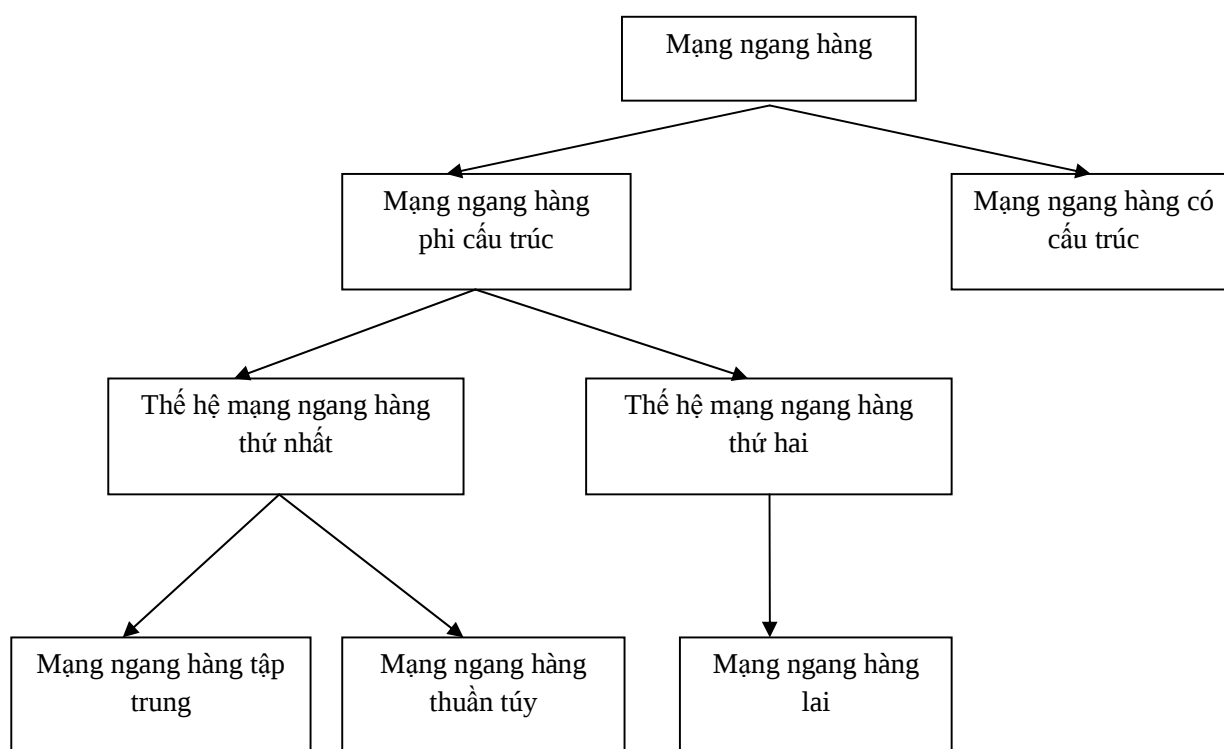
Tất cả các thành phần máy tính đồng nghĩa với các nút hay các điểm nút, có vai trò tương đương nhau và có trách nhiệm chia sẻ và được dùng các tài nguyên.

Do đó đối với các mạng máy tính ngang hàng, khi xem xét một điểm nút ta có thể hiểu đó là máy tính. Tuy nhiên trong phạm vi của khóa luận này, chúng tôi không đề cập đến vấn đề về các giao thức chia sẻ, giao thức truyền thông giữa các điểm nút, cũng như cách thức hoạt động của từng điểm nút mà chỉ nêu định nghĩa tổng quát về mạng ngang hàng máy tính, thông tin chi tiết của vấn đề có thể tham khảo trong tài liệu [2], [3], [12].

1.2.2. Phân loại các mô hình mạng ngang hàng

Việc phân loại mạng ngang hàng có nhiều cách phân loại, trong khóa luận này, chúng tôi phân loại dựa theo phân chia trong tài liệu [12], đó là dựa theo đặc điểm cấu trúc của mạng ngang hàng.

Cụ thể về cách phân loại mạng ngang hàng có thể tham khảo trong Hình 3 dưới đây. Trong Hình 3 thì mạng ngang hàng phi cấu trúc được chia làm 3 mô hình mạng ngang hàng khác. Tuy nhiên trong phạm vi nghiên cứu của khóa luận, chúng tôi tập trung vào mô hình mạng phi cấu trúc, phi tập trung, hay còn gọi là mô hình mạng ngang hàng thuần túy. Một mô hình mạng ngang hàng đặc biệt. Để tìm hiểu đầy đủ về các loại mô hình mạng ngang hàng, có thể tham khảo thêm trong tài liệu [12].



Hình 3. Phân loại mạng ngang hàng.

1.2.2.1. Mạng ngang hàng tập trung

Mạng ngang hàng tập trung là một trong những thể hệ mạng ngang hàng đầu tiên, đặc trưng của mạng này vẫn dựa vào một máy chủ tìm kiếm trung tâm. Tô pô xếp chồng của một mạng ngang hàng tập trung do đó có thể được miêu tả như là một mạng hình sao.

Trong mô hình mạng này, mỗi điểm nút kết nối tới máy chủ tìm kiếm trung tâm để có thể gửi truy vấn tìm kiếm tài nguyên, sau khi gửi yêu cầu tới máy chủ tìm kiếm trung tâm, máy chủ tìm kiếm trung tâm trả về thông tin phản hồi tương ứng với từ khóa được quy định trong truy vấn. Tức là tại máy chủ tìm kiếm trung tâm, từ khóa trong thông báo truy vấn sẽ được ánh xạ với bảng danh sách tài nguyên mà máy chủ có. Nếu máy chủ tìm kiếm trung tâm có thông tin mà điểm nút đó yêu cầu thì nó sẽ trả về thông tin vị trí truy cập tới các điểm nút chia sẻ (đa phần là trả về các địa chỉ IP và các cổng). Sau khi điểm nút đã nhận được thông tin từ máy chủ tìm kiếm trung tâm thì lúc này quá trình trao đổi thông tin cần tìm được thực hiện theo đúng cơ chế của mạng ngang hàng, tức là trao đổi trực tiếp giữa các nút mạng với nhau mà không cần qua máy chủ tìm kiếm trung tâm.

Như vậy, trong mô hình này bao gồm :

- Một máy chủ tìm kiếm trung tâm, máy chủ này chứa danh sách thông tin về các điểm nút trong mạng do nó quản lý (bao gồm : địa chỉ IP, cổng, bảng thông kết nối..vv.) và danh sách thông tin về các tài nguyên (tên tài nguyên, dung lượng tài nguyên, kiểu tài nguyên...vv.) mà mỗi điểm nút trong mạng chia sẻ.
- Các điểm nút, các điểm nút này lưu trữ các tài nguyên cần chia sẻ với các điểm nút khác trong mạng.

Cơ chế hoạt động của mô hình này bao gồm 2 hoạt động : hoạt động giữa các điểm nút với máy chủ tìm kiếm trung tâm, hoạt động giữa các điểm nút với nhau.

- Hoạt động giữa điểm nút ↔ máy chủ tìm kiếm trung tâm:
 - Tìm kiếm tài nguyên
 - Đăng nhập vào mạng xếp chồng
 - Để đăng ký
 - Cập nhật thông tin các bảng định tuyến
 - Cập nhật thông tin tài nguyên được chia sẻ

- Hoạt động giữa điểm nút ↔ điểm nút:

- Trao đổi dữ liệu.

Ứng dụng điển hình cho mô hình mạng kiểu này là: Napster. Ứng dụng Napster hỗ trợ việc chia sẻ file và nhạc (miễn phí) giữa những người dùng mạng Internet và được xem như là điểm bắt đầu của mạng ngang hàng. Do các vấn đề luật pháp về bản quyền và sở hữu trí tuệ đối với các tác phẩm âm nhạc nên Napster đã thay đổi dịch vụ của nó thành một dịch vụ chia sẻ file hợp pháp trên Internet.

Với Napster, việc tìm kiếm một file có thể bị thất bại nếu như bảng tìm kiếm tại máy chủ tìm kiếm trung tâm không có thông tin cần tìm kiếm hay thông tin cần tìm kiếm là không có giá trị được chia sẻ. Trong mô hình mạng kiểu này, chỉ có tiến trình tìm kiếm file đã lưu trữ trong một mạng và vấn đề lưu trữ là phân tán.

Để biết thông tin về tài nguyên các điểm nút phải gửi truy vấn tìm kiếm tới máy chủ, nếu như số lượng lớn các điểm nút trong mạng đồng thời gửi truy vấn tìm kiếm tới máy chủ thì khi đó máy chủ đóng vai trò là một nút cổ chai và điểm duy nhất chịu lỗi. Để tránh tình trạng đó thì sức mạnh tính toán và khả năng lưu trữ của máy chủ tìm kiếm tập trung phải tăng lên tương xứng với số lượng điểm nút trong mạng. Vì vấn đề là “tương xứng” nên khả năng mở rộng mạng bị hạn chế.

Ngoài ra, còn có các ứng dụng khác như là: BitTorrent, WinMX, Audiogalaxy...vv.

❖ Ưu điểm :

- Tìm kiếm nhanh và hiệu quả.
- Quản lý tập trung/ quản trị tin cậy.
- Dễ xây dựng.

❖ Nhược điểm:

- Dễ dàng bị tấn công.
- Nút cổ chai.
- Khả năng tắc nghẽn.
- Khó mở rộng.
- Cần quản trị.

1.2.2.2. Mạng ngang hàng thuần túy

Mạng ngang hàng thuần túy là một kiểu mạng của thể hệ mạng ngang hàng thứ nhất, đặc trưng nổi bật của mô hình này là không có máy chủ tìm kiếm tập trung như trong mô hình mạng ngang hàng tập trung, do đó nó không gặp phải vấn đề nút cổ chai. Các điểm nút giao tiếp trực tiếp với điểm nút khác trong mạng mà không cần các máy chủ trung tâm riêng biệt nào, các điểm nút thiết lập kết nối với nhau ngẫu nhiên.

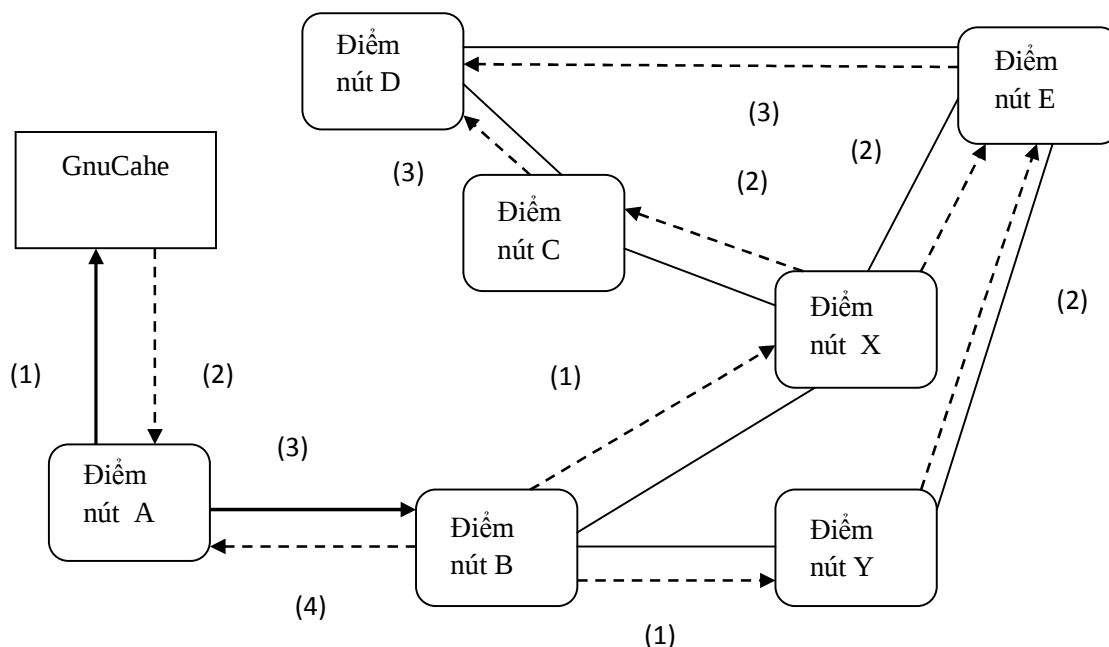
Trong mô hình mạng ngang hàng này, việc tìm kiếm file sử dụng phương pháp phát tràn (phương pháp này có sử dụng giá trị giới hạn phạm vi tìm kiếm là TTL và sử dụng GUID để trao đổi). Khi muốn tìm kiếm một file nào đó thì yêu cầu tìm kiếm được gửi từ điểm nút nguồn tới tất cả các điểm nút mạng là hàng xóm của nó. Đây vừa là đặc trưng hấp dẫn của mạng ngang hàng thuần túy và cũng là một điểm yếu của các mạng ngang hàng này bởi vì phương pháp tìm kiếm làm tăng đáng kể lưu lượng trong mạng và gây ra dư thừa hay trùng lặp thông báo truy vấn. Nếu tài nguyên được tìm thấy là tồn tại thì khi đó điểm nút có tài nguyên chia sẻ sẽ trao đổi với điểm nút yêu cầu dựa vào GUID của điểm nút yêu cầu.

Vì đặc trưng của mô hình mạng này là không có máy chủ tìm kiếm trung tâm nên việc một điểm nút mới gia nhập vào một mạng sẽ thế nào, thực hiện tìm kiếm ra sao là vấn đề cần được làm rõ. Dựa vào ứng dụng phần mềm điển hình cho mô hình mạng này là phần mềm Gnutella 0.4, chúng tôi sẽ mô tả sơ lược cách thức một điểm nút tham gia vào mạng và cách thức thực hiện tìm kiếm.

❖ Điểm nút A tham gia vào mạng Gnutella (Hình 4):

- (1) Điểm nút A kết nối tới GnuCache (GnuCache được sử dụng để cache các host Gnutella và được tích hợp bên trong phần mềm Gnut cho unix) để lấy danh sách các điểm nút có giá trị và đã được kết nối trong mạng.
- (2) GnuCache gửi trả danh sách mà mình có cho điểm nút A.
- (3) Điểm nút A sau khi nhận danh sách, trong danh sách của nó có duy nhất điểm nút B, nó gửi thông báo GNUTELLA CONNECT (thông báo dùng để kết nối tới các điểm nút đã tồn tại trong mạng.) tới điểm nút B.
- (4) Nếu điểm nút B chấp nhận thông báo của điểm nút A thì nó sẽ gửi thông báo GNUTELLA OK, và hỗ trợ điểm nút A tham gia vào mạng. Bây giờ điểm nút A đã là

một phần của mạng Gnutella và đã được kết nối tới một điểm nút Gnutella khác là điểm nút B.



Hình 4. Mô tả một nút tham gia vào mạng Gnutella và tìm kiếm file.

❖ Điểm nút A tìm kiếm tài nguyên (Hình 4):

Để biểu diễn cách thức tìm kiếm được sử dụng trong mạng Gnutella, các ký hiệu (1), (2), (3) ở bên phải điểm nút B trên Hình 4, thể hiện chặng thứ i, còn mũi tên biểu thị các hàng xóm có thể lan tỏa được của điểm nút tại chặng đó.

- Điểm nút A sau khi tham gia vào mạng, khi nó cần file nào đó nhưng nó lại không có file đó, nó sẽ tiến hành tìm kiếm file đó trên mạng. Vì điểm nút A lại không có thông tin gì ngoài thông tin ai là hàng xóm của nó nên nó gửi thông báo truy vấn (điểm nút A trong trường hợp này chỉ gửi 1 thông báo vì nó chỉ có một hàng xóm) tới hàng xóm của nó là điểm nút B để hỏi B có tài nguyên nó cần hay không.
- Điểm nút B sau khi nhận được thông báo yêu cầu của điểm nút A, đầu tiên nó sẽ kiểm tra thông báo có phải là một thông báo cũ hay không. Nếu là một thông báo mới thì nó sẽ kiểm tra thông tin mà điểm nút A yêu cầu bằng cách ánh xạ từ khóa yêu cầu trong thông báo truy vấn tới cơ sở dữ liệu của nó. Nếu nó có dữ liệu mà điểm nút A cần thì nó sẽ gửi lại thông báo queryHit cho điểm nút A. Quá trình truy vấn dừng lại, và thực

hiện trao đổi dữ liệu giữa điểm nút A và điểm nút B, việc trao đổi là trực tiếp giữa 2 điểm nút. Nếu điểm nút B không có tài liệu mà điểm nút A cần thì nó sẽ giảm giá trị TTL trong thông báo truy vấn đi 1 và chuyển tiếp thông báo truy vấn tới các hàng xóm của nó là X, Y.

- Điểm nút X và điểm nút Y lúc này thực hiện quá trình tương tự như điểm nút B. Tại điểm nút E nhận được nhiều thông báo truy vấn do từ điểm nút X và điểm nút Y gửi đến.
- Điểm nút C và E lại chuyển tiếp tới điểm nút D, và tại điểm nút D có tài liệu mà điểm nút A cần. Giả sử điểm nút C gửi thông báo đầu tiên tới điểm nút D khi đó thông báo do điểm nút E gửi thông báo truy vấn tới điểm nút D bị loại bỏ (đây là cách tối ưu cho tìm kiếm và hạn chế truy vấn lặp tại các điểm nút có tài nguyên). Điểm nút D gửi cho điểm nút A thông báo queryHit theo đường từ D-C-X-B-A. Nhận được queryHit của D, bây giờ điểm nút A đã có địa chỉ, số cổng mà D cung cấp, khi đó việc trao đổi file diễn ra trực tiếp giữa D và A, không theo tuyến đường đã tìm thấy D.

Ví dụ về việc tham gia vào mạng của một điểm nút, cũng như cách thức tìm kiếm của một điểm nút trong mạng Gnutlla có thể tham khảo thêm thông tin trong các tài liệu [3], [12], [13].

Ngoài ra ứng dụng điển hình còn có các ứng dụng khác như là: FreeNet, Gnu-Net...vv.

❖ Ưu điểm:

- Không có điểm duy nhất chịu lỗi → khó bị tấn công.
- Có thể thích nghi với mạng vật lý.
- Cho phép ẩn danh.
- Dễ xây dựng.
- Các điểm nút tham gia và rời khỏi mạng một cách tùy mà không ảnh hưởng đến cấu trúc của toàn mạng.

❖ Nhược điểm:

- Tốn tài nguyên băng thông.

- Tìm kiếm phức tạp
- Các điểm nút có khả năng khác nhau (sức mạnh xử lý của CPU, băng thông, không gian lưu trữ...vv.) đều có thể phải chịu tải như nhau.

Như vậy, qua ví dụ với ứng dụng Gnutella chúng tôi đã trình bày về cách thức một điểm nút mới tham gia vào mạng ngang hàng thuần túy như thế nào và cách thức một điểm nút tìm kiếm tài liệu trong mạng đó. Đồng thời cũng nêu ra những ưu điểm, nhược điểm của mạng này khi đặc trưng của nó là không có máy chủ tìm kiếm trung tâm.

1.2.2.3. Mạng ngang hàng lai

Mạng ngang hàng lai là mạng ngang hàng thuộc thể hệ thứ 2. Chúng được phát triển để khắc phục nhược điểm của các mô hình mạng ngang hàng trước đó. Mô hình mạng ngang hàng lai bao gồm các: các siêu điểm nút, các điểm nút thông thường (hay còn gọi là điểm nút client). Trong đó các siêu điểm nút tạo thành một mạng không có cấu trúc, và mỗi siêu điểm nút kết nối đến nhiều điểm nút thông thường, mỗi siêu điểm nút quản lý vùng của nó, vai trò của siêu điểm nút giống như một máy chủ trong mô hình mạng ngang hàng tập trung.

Ứng dụng điển hình cho mô hình mạng này là: Gnutella 0.6, Kazaa/FastTrack. Ngoài ra còn có: Edonkey, Emule, OpenNap, JXTA, Skype...vv.

❖ Ưu điểm:

- Không có điểm duy nhất chịu lỗi vì có nhiều siêu điểm nút.
- Cho phép ẩn danh.
- Phù hợp với các nhóm lợi ích đặc biệt.
- Khả năng mở rộng quy mô tốt.
- Hiệu quả thỏa mãn các truy vấn.
- Hạn chế phát tràn các truy vấn và tránh được hiện tượng nút cổ chai.

❖ Nhược điểm:

- Phân chịu tải không cân bằng: các siêu điểm nút chịu tải cao hơn.

1.3. Mạng xếp chồng

Là mạng máy tính được xây dựng trên nền của một mạng khác. Các nút trong mạng xếp chồng được xem là nối với nhau bằng liên kết ảo (liên kết logic), mỗi liên kết ảo có thể bao gồm nhiều liên kết vật lý của mạng nền.

Có rất nhiều mạng ngang hàng được gọi là mạng xếp chồng vì nó được xây dựng và hoạt động trên nền của mạng Internet. Ví dụ như: Gnutella, FreeNet...vv.

CHƯƠNG 2. LÝ THUYẾT ĐỒ THỊ VÀ CÁC DẠNG ĐỒ THỊ MẠNG

Đồ thị là một mô hình toán học được sử dụng để biểu diễn một tập đối tượng có quan hệ với nhau theo một cách nào đó. Chẳng hạn trong rất nhiều vấn đề, lĩnh vực khác nhau như công nghệ điện, hoá học, chính trị, kinh tế,...vv, cũng có thể biểu diễn bởi đồ thị. Trong lĩnh vực tin học, đồ thị được sử dụng để mô hình hoá một mạng truyền thông, kiến trúc của các máy tính song song,...vv. Vì khóa luận của chúng tôi có liên quan đến đồ thị nên trong chương này chúng tôi sẽ trình bày những nội dung của Lý thuyết đồ thị phù hợp với vấn đề của chúng tôi. Tiếp đó chúng tôi sẽ giới thiệu các dạng đồ thị mạng truyền thông được sử dụng cho mô phỏng trong khóa luận của chúng tôi. Tìm hiểu về các dạng đồ thị này sẽ giúp cho việc tìm hiểu hoạt động của các phương pháp tìm kiếm mà chúng tôi trình bày ở phần nội dung tiếp theo của bài báo, cũng như phục vụ cho mô phỏng của chúng tôi.

2.1. Khái niệm đồ thị

2.1.1. Đồ thị có hướng

Đồ thị có hướng là đồ thị bao gồm tập các đỉnh $V=\{1,2,...,n\}$, và tập các cung E . Mỗi cung là 1 cặp có thứ tự của các đỉnh (u,v) tương ứng với một liên kết có hướng từ u tới v .

Bậc ra của một đỉnh u là số lượng các các cung (liên kết) xuất phát từ u , và bậc vào là số lượng các cung tới u .

Một đường đi từ u tới v là một dãy các cung $(u,u_1), (u_1,u_2), (u_2,u_3), \dots, (u_k,v)$. Có đường đi từ u tới v không bao hàm nghĩa đây là đường đi từ v tới u .

Đồ thị có hướng được gọi là liên thông mạnh nếu với mọi cặp đỉnh u và v khác nhau trong tập nút bao giờ cũng có một đường đi từ u tới v . Đồ thị có hướng có nhiều hơn một thành phần liên thông mạnh.

2.1.2. Đồ thị vô hướng

Một đồ thị vô hướng là một đồ thị bao gồm 1 tập các đỉnh và tập các cạnh, mỗi một cạnh trong đó là một cặp đỉnh không có thứ tự $\{u,v\}$.

Bậc của một đỉnh u là số lượng các cạnh liên quan tới u (hàng xóm của u).

Đường đi trong đồ thị vô hướng khác với đồ thị có hướng là đường đi từ u tới v thì có hàm ý là có đường đi từ v đến u .

Một thành phần liên thông của đồ thị vô hướng là một tập các đỉnh sao cho mọi cặp đỉnh u và v bất kỳ bao giờ cũng có đường đi từ u tới v . Thành phần liên thông của đồ thị vô hướng thu được từ thành phần liên thông của đồ thị có hướng sau khi loại bỏ hướng của các cung.

2.1.3. Các khái niệm khác

- Đồ thị đơn là đồ thị không có khuyên và bất kỳ 2 đỉnh phân biệt nào cũng được nối với nhau bởi không quá một cạnh.
- Đồ thị đa là đồ thị không có khuyên và tồn tại một cặp đỉnh phân biệt được nối với nhau bởi nhiều hơn một cạnh.
- Đồ thị đầy đủ, ký hiệu K_n là một đơn đồ thị bao gồm n đỉnh mà mọi đỉnh đều có bậc $n-1$.

Để tìm hiểu rõ hơn về lý thuyết đồ thị, bạn đọc có thể đọc trong tài liệu [1], [8], [18]. Trong phần tiếp theo chúng tôi sẽ trình bày ứng dụng đồ thị trong mạng truyền thông, cụ thể là mô hình hóa các mô hình mạng ngang hàng thành các dạng đồ thị tương ứng.

2.2. Các dạng đồ thị trong mạng ngang hàng

Khi xem xét mô hình hóa một mạng ngang hàng bởi một đồ thị thì khi đó khái niệm tập $V=\{1,2,3,...,n\}$ của đồ thị $G=(V,E)$ gọi là tập n điểm nút hay nút trong mạng, mỗi điểm nút được cung cấp một định danh ID và địa chỉ mạng. Và E là tập các kết nối giữa các điểm nút hay tập các cạnh giữa các đỉnh trong tập V . Với $u, v \in V$; $\{u,v\} \in E$ biểu thị rằng các điểm nút u và v biết địa chỉ IP của nhau, giữa chúng tồn tại kết nối, u và v còn được gọi là hàng xóm của nhau, u là nút nguồn, v là nút đích. Tất cả các cạnh là *liên kết có hướng* với nhau.

Trong mô hình hóa mạng dưới dạng đồ thị có thể là có 1 hay 2 hướng liên kết, còn trong mô hình vật lý thực, thì chỉ có 1 liên kết vật lý giữa các peer với nhau hoặc 1 liên kết trong mô hình hóa lại có nhiều liên kết vật lý qua nhiều máy trung gian ở tầng mạng. Vì vậy, đồ thị G đôi khi còn được gọi là “*mạng xếp chồng*” của một hệ thống mạng ngang hàng.

Có nhiều dạng đồ thị thỏa mãn mô hình mạng ngang hàng thuần túy này như là : đồ thị ngẫu nhiên, đồ thị luật hàm mũ, tô pô phân cụm. Đây cũng là những dạng đồ thị chung cho mạng ngang hàng. Vì vậy chúng tôi sẽ mô phỏng những phương pháp tìm kiếm trên những dạng đồ thị này để đánh giá xem phương pháp nào tốt, phương pháp nào tồi.

2.2.1. Đồ thị ngẫu nhiên

Một đồ thị ngẫu nhiên là một đồ thị được sinh ra bởi nhiều thủ tục ngẫu nhiên. Có nhiều cách để định nghĩa các đồ thị ngẫu nhiên. Cách đơn giản nhất, được biểu thị bằng $G_{n,m}$, trong đó n là số đỉnh của đồ thị, các đỉnh được lựa chọn ngẫu nhiên để xây dựng đồ thị, và m là số cạnh có thể của đồ thị, với $0 \leq m \leq \binom{n}{2}$. Định nghĩa này là định nghĩa đồ thị theo mô hình của Erdős and Renyi, chi tiết thông tin tham khảo trong các tài liệu [10], [12], [18].

Một biểu diễn khác về đồ thị ngẫu nhiên được đưa ra trong tài liệu [12], [18] là $G_{n,p}$ trong đó $0 \leq p \leq 1$, các đỉnh là giống nhau nhưng lựa chọn cạnh có thể với xác suất p , độc lập với tất cả các cạnh khác, đây là định nghĩa của mô hình Gilbert. Gọi z là bậc trung bình của 1 đỉnh thì:

$$z = \frac{n(n-1)p}{n} = (n-1)p \approx np \quad (1)$$

Độ xấp xỉ càng tốt khi n càng lớn.

Như vậy, đồ thị ngẫu nhiên được xây dựng theo cách thức hoặc giá trị ngẫu nhiên tại mỗi bước xây dựng đồ thị. Dạng đồ thị này làm cơ sở cho xây dựng các dạng đồ thị khác, và sử dụng trong mô phỏng.

2.2.2. Đồ thị luật hàm mũ

Đồ thị luật hàm mũ có tên tiếng Anh là “power-law” hay “scale-free networks”. Đồ thị này được xây dựng bởi 3 anh em: Michael, Petros và Christos Faloutsos, thông tin về lịch sử ra đời của đồ thị có thể tham khảo thêm trong tài liệu [12].

Đồ thị luật hàm mũ là đồ thị trong đó tất cả các trang web (web tĩnh) đã thăm được biểu diễn như là các nút, và 2 trang được kết nối bởi 1 cạnh (i,j) có hướng nếu trang i có 1 liên kết điểm tới trang j . Trong đồ thị, số lượng các nút cùng với bậc quy định đã được tính toán bằng việc phân chia bậc bởi số lượng các nút trong đồ thị, xác suất $P(k)$ để vẽ 1

nút ngẫu nhiên với bậc k đã được tính toán là như nhau. Xác suất $P(k)$ là tỉ lệ với hàm mũ của k với 1 hằng số γ .

$$P(k) \propto k^{-\gamma} \quad (2)$$

Trong đó γ là hằng số không phụ thuộc vào kích thước của mạng, và trong mô phỏng của chúng tôi thì $\gamma = 2.1$ cho các bậc vào, $\gamma = 2.7$ cho các bậc ra, để hiểu rõ hơn có thể tham khảo theo tài liệu [11], [14], [19].

Có nhiều mô hình cho đồ thị web, nhưng trong chương trình mô phỏng của chúng tôi, chúng tôi lựa chọn mô hình “Protean Model”. Mô hình này yêu cầu tính toán một tham số tự nhiên thêm vào của một đỉnh, đó là *age* và dự đoán nó sẽ ảnh hưởng thế nào đến bậc của một đỉnh. Với đồ thị G có tập n đỉnh và tại bước nào đó chọn ngẫu nhiên một trong những đỉnh v của tập đỉnh để *thay đổi mới* (*renewed*). Chính xác hơn, xóa bỏ từ đồ thị G tất cả các cạnh phụ thuộc vào đỉnh v , việc này tương ứng với việc xóa bỏ một nút ngẫu nhiên từ mạng. Sau đó tạo ra d cạnh mới từ đỉnh v với các đỉnh đã tồn tại, các đỉnh này được lựa chọn ngẫu nhiên với các xác suất có trọng số (các đỉnh ‘cũ’ có xác suất cao hơn để chọn lựa). Lưu ý rằng đỉnh v có thể được coi như là một nút mới, nút mà thiết lập kết nối với một vài nút trong mạng. Khi tất cả các đỉnh là thay mới tối thiểu 1 lần, đồ thị ngẫu nhiên là một đồ thị protean $P_n(d, \eta)$, thông tin tham khảo thêm trong tài liệu [11], [14], [19].

Trong tài liệu [11] của tác giả và đồng nghiệp đưa ra rằng các bậc của $P_n(d, \eta)$ là được phân phối phù hợp với một luật hàm mũ. Chính xác hơn, số lượng các đỉnh bậc k giảm mạnh $k^{-1-1/\eta}$.

Trong mô phỏng chúng tôi thực hiện kiểm tra với bậc trung bình là 5.

2.2.3. Tô pô phân cụm

Một đồ thị $G=(V,E)$ được gọi là một tô pô phân cụm (cluster topo) nếu mọi thành phần liên thông của G là một đồ thị đầy đủ. Đồ thị G cũng được gọi là một p -phân cụm (p -cluster) nếu nó là 1 tô pô phân cụm với p thành phần liên thông hoặc tương đương nếu nó là phép toán hợp các đỉnh rời khỏi mạng của p phân cụm, tham khảo thêm trong [15].

Trong chương trình mô phỏng, với dạng đồ thị này sử dụng 3 mô hình phân cụm đó là :

- K_l để phân kích cỡ cụm.
- Đồ thị ngẫu nhiên $G_{l,1/2}$.
- Đồ thị ngẫu nhiên $G_{l,1/5}$.

Kích thước của các phân cụm ở trong mô hình này là $l=100$.

Như vậy, trong chương này chúng tôi đã trình bày khái niệm cơ bản về lý thuyết đồ thị và các dạng đồ thị mạng được sử dụng chung cho mạng ngang hàng cũng như cho mạng ngang hàng thuần túy.

CHƯƠNG 3. CÁC PHƯƠNG PHÁP TÌM KIẾM ĐÃ ĐỀ XUẤT TRƯỚC ĐÂY

Tìm kiếm là một chức năng chính của mạng ngang hàng chia sẻ tài nguyên. Một trong những vấn đề phổ biến nhất với những người dùng mạng ngang hàng đó là sử dụng lượng thời gian đáng kể cho việc tìm kiếm tài nguyên, nguyên nhân là do người dùng thường không tìm thấy tài nguyên mà họ cần. Thậm chí khi họ tìm thấy những tài nguyên họ cần, họ phải gửi truy vấn trở lại tới các tài nguyên giống thế trong các nguồn khác khi các nút từ xa ngắt kết nối khỏi mạng. Như vậy cần một phương pháp tìm kiếm mà tiết kiệm thời gian, tìm kiếm được tài nguyên cần thiết, và hoạt động tốt trong các môi trường mạng không ổn định, phức tạp. Mô hình mạng mà khóa luận chúng tôi nghiên cứu là mô hình mạng ngang hàng thuần túy và phương pháp tìm kiếm như đã trình bày ở trên là sử dụng phương pháp phát tràn (flooding). Ngoài phương pháp phát tràn ra, cũng có phương pháp tìm kiếm khác được sử dụng cho mô hình mạng loại này, đó là phương pháp dịch chuyển ngẫu nhiên (random walk). Vì những phương pháp tìm kiếm đơn lẻ này có những ưu điểm, nhược điểm riêng và với nhu cầu để có được phương pháp tốt hơn nên đã có nhiều đề xuất cải tiến hoặc kết hợp giữa các giải thuật này. Trong chương này, chúng tôi trình bày chi tiết về từng phương pháp.

3.1. Các phương pháp tìm kiếm đơn lẻ

3.1.1. Phương pháp tìm kiếm phát tràn thông thường

Đây là một trong những phương pháp được sử dụng phổ biến, đơn giản nhất trong mạng ngang hàng, đồng thời cũng là phương pháp dễ cài đặt, đối với các mạng ngang hàng thuần túy (như trong mục 1.2.2.2 đã trình bày) phương pháp này áp dụng trong ứng dụng Gnutella. Phương pháp này thuộc nhóm phương pháp tìm kiếm mù (blind search), và trong tìm kiếm đồ thị thì phương pháp này giống với phương pháp duyệt ưu tiên theo chiều rộng (BFS).

Cơ chế hoạt động của phương pháp này như đúng tên gọi của nó, tức là : Một nút muốn tìm kiếm tài nguyên trong mạng mà nó lại không có thông tin về tài nguyên của các nút lưu trữ trong mạng và cũng không biết đường đi đến những nút có tài nguyên, không có siêu nút hỗ trợ nó trong việc tìm kiếm thông tin cần, mà chỉ biết thông tin định tuyến tới các nút hàng xóm của nó. Khi đó nút nguồn này sẽ gửi thông báo truy vấn tới các hàng

xóm của nó. Nếu hàng xóm của nó có tài nguyên mà nó cần thì hàng xóm sẽ gửi cho nó tài nguyên đó và truy vấn kết thúc. Còn nếu như tất cả các hàng xóm của nó đều không có tài nguyên mà nó cần thì các hàng xóm này lại tiếp tục chuyển tiếp tin truy vấn này tới các hàng xóm của chúng. Quá trình tìm kiếm cứ tiếp tục như thế. Để hiểu rõ và trực quan hơn, có thể xem xét lại mục 1.2.2.2 của tài liệu này, hoặc tham khảo thêm trong tài liệu [12].

Tài nguyên sẽ được tìm thấy nếu như nó tồn tại trên mạng và quá trình tìm kiếm sẽ dừng lại khi tìm thấy tài nguyên hoặc khi giá trị quy định thời gian tìm kiếm là bị sử dụng hết, giá trị này gọi là TTL.

Khi mà nút chứa tài nguyên cần tìm không phải là nút hàng xóm của nó thì quá trình trao đổi tài nguyên sẽ diễn ra trực tiếp giữa hai nút với nhau, không đi theo tuyến đường đã tìm kiếm vì lúc này 2 nút đã có thông tin về nhau, trong đó có cả địa chỉ IP.

❖ Ưu điểm của phương pháp :

- Là dễ dàng cài đặt, và dễ dàng sử dụng.
- Hiệu quả trong một phạm vi hẹp nhất định

❖ Tuy nhiên phương pháp này còn có những hạn chế là:

- Số lượng thông báo truy vấn tăng theo hàm mũ khi TTL tăng.
- Để chọn một giá trị TTL hợp lý rất khó, thông thường thì giá trị TTL được chọn trong một số bài báo là $TTL = 7$, như trong tài liệu [5], [20].
- Một nút có thể nhận nhiều hơn 1 thông báo $\rightarrow$ hiện tượng trùng lặp thông báo, làm tăng tải, và ảnh hưởng tới giao tiếp của toàn mạng.

Phương pháp này gây tốn băng thông và làm cho các điểm nút khác phải chịu tải, mặc dù không chứa tài nguyên. Với mô hình mạng phức tạp, số lượng nút trong mạng là khá lớn như tô pô phân cụm chẳng hạn và số lượng thông báo truy vấn lớn thì hiệu quả tìm kiếm của phương pháp này là thấp.

3.1.2. Phương pháp tìm kiếm di chuyển ngẫu nhiên

Ngoài phương pháp phổ biến phát tràn thì việc tìm kiếm một tài nguyên ngẫu nhiên lưu trong mạng có thể sử dụng phương pháp dịch chuyển ngẫu nhiên. Phương pháp này hạn chế nhược điểm sinh ra số lượng lớn các thông báo dư thừa, làm giảm hiệu năng sử

dụng bằng thông chung của mạng bởi các phương pháp tìm kiếm phát tràn. Phương pháp này cũng đòi hỏi kỹ thuật phức tạp và cao hơn so với các phương pháp tìm kiếm mù, cơ chế của phương pháp này cũng giống như phương pháp tìm kiếm DFS nhưng có sự khác biệt về việc lựa chọn nút tại mỗi chặng phát triển để chuyển tiếp truy vấn tìm kiếm.

Phương pháp phát tràn và phương pháp này có sự khác biệt về cơ chế tìm kiếm. Thay vì phát tràn sang tất cả các hàng xóm như trong phát tràn thì phương pháp này lại gửi sang một hàng xóm được lựa chọn ngẫu nhiên. Việc chọn lựa là ngẫu nhiên và bình đẳng với tất cả các hàng xóm. Nếu như hàng xóm nhận được thông báo, nó lại chứa tài liệu mà điểm nút nguồn cần, trong khi các giá trị giới hạn tìm kiếm vẫn chưa đạt đến ngưỡng dừng thì điểm nút này sẽ gửi lại queryHit cho điểm nút nguồn theo con đường mà thông báo truy vấn đã gửi tới nó. Sau đó việc trao đổi tài liệu diễn ra trực tiếp giữa điểm nút này và điểm nút nguồn. Nếu như điểm nút này không có tài liệu thì nó sẽ thực hiện tương tự, chọn lựa điểm nút hàng xóm ngẫu nhiên của nó và chuyển tiếp truy vấn yêu cầu đến đó. Thông báo truy vấn sử dụng trong phương pháp này được gọi là “walker”.

Cũng như phương pháp phát tràn, nếu như tài nguyên cần tìm là tồn tại và sau một lượng walker nhất định nào đó thì tài nguyên sẽ được tìm thấy. Sau khi xác định được vị trí của tài nguyên cần tìm thì việc trao đổi để có tài nguyên là diễn ra trực tiếp giữa các điểm nút.

Ưu điểm của phương pháp này hạn chế lượng thông báo dư thừa sinh ra do đó giảm trùng lặp vì tại mỗi bước chỉ chọn lựa có một hàng xóm.

Tuy nhiên phương pháp có hạn chế đó là độ trễ thời gian tìm kiếm thấy tài nguyên cao.

3.2. Các phương pháp tìm kiếm kết hợp

Xuất phát từ nhu cầu có một phương pháp tìm kiếm cho hiệu quả tốt trong các mạng ngang hàng phi cấu trúc và dựa trên những ưu điểm, nhược điểm của hai phương pháp tìm kiếm đơn lẻ trên, đã có nhiều công trình nghiên cứu, báo cáo khoa học đề xuất phương pháp kết hợp. Các phương pháp này cho kết quả tốt theo hướng nghiên cứu của tác giả. Trong phần nội dung này, chúng tôi trình bày sơ lược ý tưởng của các phương pháp đó.

3.2.1. Phương pháp tìm kiếm động

Phương pháp tìm kiếm động được thiết kế là một phương pháp tổng quát cho phát tràn và di chuyển ngẫu nhiên, thiết kế phương pháp này dựa trên phạm vi mà phương pháp có ưu thế, tức là phương pháp phát tràn tốt trong phạm vi hẹp và sử dụng ở kích thước mạng lớn thì phương pháp di chuyển ngẫu nhiên tốt. Chính vì thế phương pháp này bao gồm 2 giai đoạn, mỗi giai đoạn có một chiến lược tìm kiếm khác nhau. Lựa chọn chiến lược tìm kiếm nào tại mỗi giai đoạn phụ thuộc vào mối quan hệ giữa số lượng chạng đếm được h của các thông báo truy vấn và ngưỡng quyết định n của phương pháp tìm kiếm động [20].

Nếu như $h \leq n$ thì phương pháp tìm kiếm động thực hiện như là phát tràn để tìm kiếm hoặc là phương pháp phát tràn cải tiến.

Nếu $h > n$ thì phương pháp tìm kiếm động sử dụng di chuyển ngẫu nhiên để tìm kiếm

Để tìm hiểu rõ hơn về hoạt động của phương pháp, bạn đọc có thể tham khảo trong bài báo chi tiết [20].

Như vậy phương pháp này, sử dụng điểm mạnh, điểm yếu trong phạm vi tìm kiếm, kích thước của mạng để kết hợp 2 phương pháp đơn lẻ, tùy theo phạm vi, kích thước mà sử dụng phương pháp hợp lý.

3.2.2. Phương pháp tìm kiếm lai

Phương pháp này được đề xuất bởi Reza, Alejandro, và Pawel [14]. Ý tưởng của phương pháp này là : Đầu tiên sử dụng phương pháp phát tràn từ nút nguồn tìm kiếm để tìm kiếm các nút có khả năng được tìm thấy; phương pháp phát tràn cho đến khi chính xác k nút mới cuối cùng (vòng ngoài của phạm vi tìm kiếm) được khám phá, giá trị k được xác định trước và sau đó bắt đầu thực hiện phương pháp di chuyển ngẫu nhiên từ các nút này. Tức là khi đó k nút này, mỗi node sẽ đóng vai trò một nút nguồn thực hiện di chuyển ngẫu nhiên.

Các tác giả cũng sử dụng điểm mạnh của các 2 phương pháp tìm kiếm theo phạm vi tìm kiếm, kích thước mạng, tức là : Nếu như tài nguyên được xác định vị trí gần với nút nguồn tìm kiếm thì phương pháp phát tràn vùng lân cận sẽ đủ để xác định được tài nguyên nhanh hơn, và chỉ với vài thông báo thông báo trao đổi. Nếu như tài nguyên được

xác định ở xa mạng, có nghĩa là trọng phạm vi lân cận, tài nguyên cần tìm không có thì phương pháp tìm kiếm lai cho rằng nó sẽ xác định vị trí bởi phương pháp di chuyển ngẫu nhiên và phương pháp thực hiện từ các nút vòng ngoài của phương pháp phát tràn trước đó (những nút cuối cùng trong phương pháp phát tràn). Lúc này, việc thực hiện phương pháp tìm kiếm di chuyển ngẫu nhiên từ những nút vòng ngoài giống như thực hiện nhiều phương pháp di chuyển ngẫu nhiên một lúc. Một cách trực quan có thể thấy phương pháp tạo ra một vùng phủ, sau đó từ vùng phủ lại tạo ra nhiều tua lan tỏa để thực hiện phương pháp di chuyển ngẫu nhiên, nhưng điều quan trọng hơn, từ phương pháp phát tràn chỉ xảy ra trong vùng lân cận nhỏ, nên độ phức tạp thông báo thấp, mặc dù vẫn xuất hiện nhiều thông báo trùng lặp trong phạm vi hẹp.

Có thể thấy cũng sử dụng điểm mạnh của cả 2 phương pháp tìm kiếm đơn lẻ phổ biến nhưng có sự khác biệt rõ ràng về “*chất*” trong 2 phương pháp kết hợp này. Phương pháp tìm kiếm động chỉ sử dụng thuần túy 2 phương pháp cổ điển dựa theo số chặng tìm kiếm tương ứng với lợi thế của phương pháp. Còn phương pháp tìm kiếm lai ghép, cũng sử dụng ưu điểm của phương pháp ứng với phạm vi tìm kiếm nhưng không đơn thuần là thực hiện phương pháp cổ điển nào trước, nào sau, mà có sự kết hợp về mặt kỹ thuật trong 2 phương pháp này.

Cũng có một phương pháp là lai ghép khác được đề xuất trong tài liệu [5], nhưng phương pháp đề xuất không liên quan tới mô hình mà chúng tôi nghiên cứu, tuy nhiên cũng là bài báo hữu ích để tham khảo việc kết hợp 2 phương pháp tìm kiếm cổ điển.

Như vậy, trong chương này chúng tôi đã giới thiệu về phương pháp tìm kiếm trước đây đã được sử dụng và các phương pháp đã được đề xuất cho mô hình mạng ngang hàng thuần túy. Tuy nhiên, sự kết hợp mà các phương pháp đề xuất này là dựa trên phạm vi tìm kiếm, có phương pháp thì chỉ sử dụng thông thường, có phương pháp có sự can thiệp về mặt kỹ thuật. Giả sử xét trường hợp như sau: phạm vi mà sử dụng phát tràn là tốt, nhưng mà phương pháp này không tìm thấy tài nguyên, và phải thực hiện thêm vài chặng tìm kiếm tiếp bằng phương pháp di chuyển ngẫu nhiên để tìm kiếm thì cũng là một trường hợp lãng phí tài nguyên bằng thông, và hiệu quả tìm kiếm cũng chưa phải cao. Tuy nhiên, nếu thay đổi lại trình tự tìm kiếm: phương pháp di chuyển ngẫu nhiên trước (số chặng tương xứng với số chặng của phương pháp phát tràn ở trên), sau đó thực hiện phương pháp di chuyển ngẫu nhiên thì sẽ giảm lãng phí tài nguyên bằng thông của mạng. Việc sử dụng ngược lại trình tự tìm kiếm, chúng tôi không xét đơn thuần như với phương pháp

tìm kiếm động, mà thực hiện như với phương pháp tìm kiếm lai ghép, tức là có sự thay đổi về kỹ thuật trong tìm kiếm. Đây cũng chính là vấn đề và là ý tưởng để chúng tôi đề xuất ra phương pháp của mình, và chúng tôi sẽ trình bày chi tiết trong chương tiếp theo

CHƯƠNG 4. CÁC PHƯƠNG PHÁP TÌM KIẾM LAI GHÉP CỦA CHÚNG TÔI

Dựa vào những kết quả phân tích trong [14], chúng tôi thấy phương pháp phát tràn hầu như tốt trên tất cả các dạng đồ thị mà các tác giả đề xuất, nhưng phương pháp di chuyển ngẫu nhiên lại tốt trên đồ thị phân cụm. Đồng thời dựa trên những phân tích đã trình bày trong chương 3, chúng tôi thấy nếu trong trường hợp nào đó thì sử dụng phương pháp di chuyển ngẫu nhiên trước, sau đó sử dụng phương pháp phát tràn thì đó cũng là một giải pháp và cũng chưa có tác giả nào đề cập đến vấn đề này. Phương pháp tìm kiếm trong mạng ngang hàng thuần túy vẫn là “ngẫu nhiên” bởi vì thông tin về vị trí tài nguyên, thông tin định tuyến vẫn là chưa xác định với các điểm nút cần tài tìm tài nguyên và các phương pháp thường được sử dụng là các phương pháp tìm kiếm mù. Trong phần này, chúng tôi sẽ trình bày chi tiết những phương pháp kết hợp mới của chúng tôi: về mặt ý tưởng, mã giải thuật. Và chúng tôi gọi các phương pháp lai ghép mới này là các biến thể tìm kiếm.

4.1. Phương pháp tìm kiếm lai ghép sử dụng phát tràn thông thường

Trong nhóm phương pháp này, chúng tôi sử dụng 2 phương pháp tìm kiếm của nhóm phương pháp tìm kiếm đơn lẻ là: phương pháp tìm kiếm di chuyển ngẫu nhiên và phương pháp phát tràn thông thường.

4.1.1. Phương pháp tìm kiếm lai ghép biến thể thứ nhất

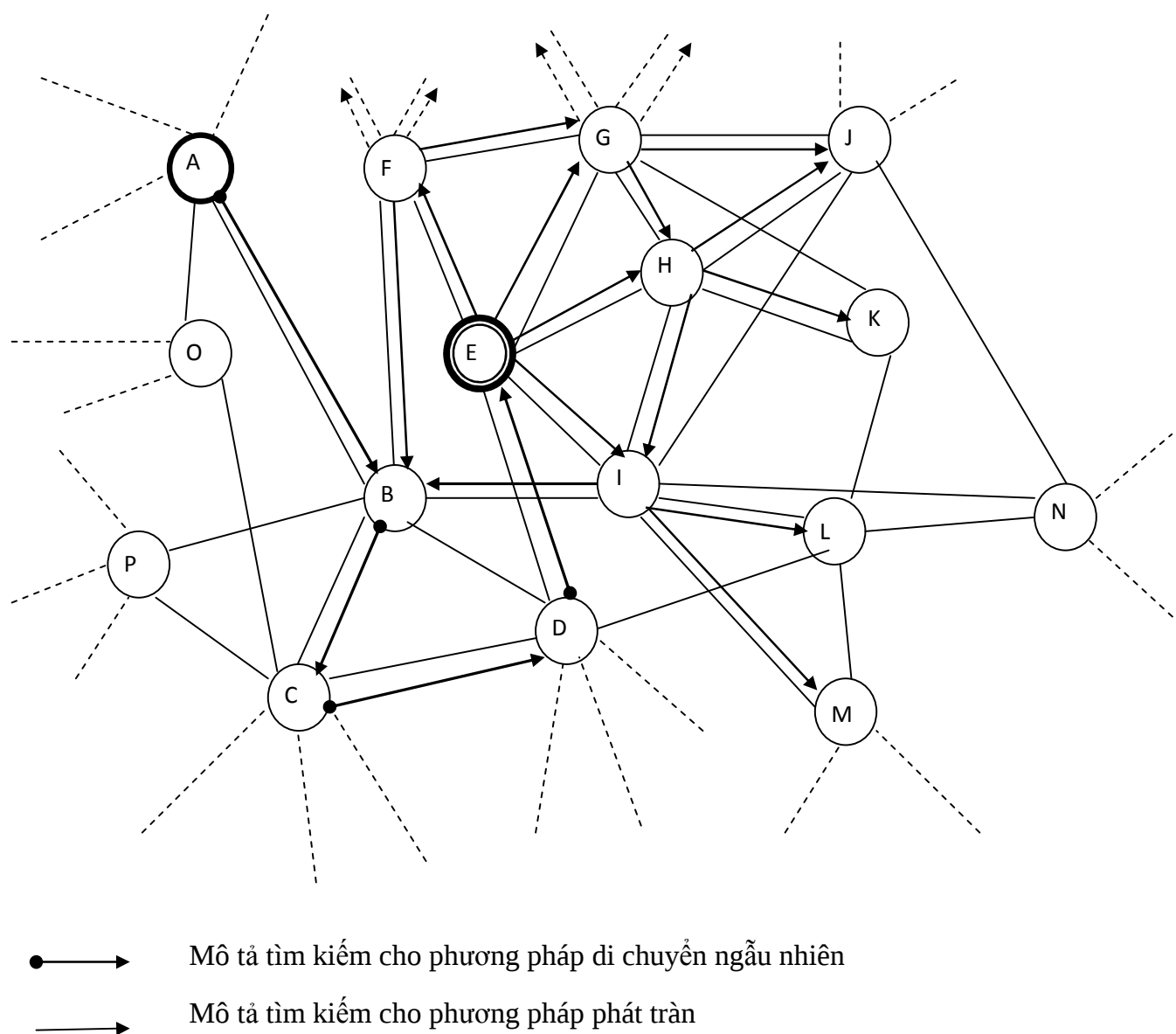
4.1.1.1. Ý tưởng của phương pháp

Ý tưởng của biến thể này là: Đầu tiên sử dụng phương pháp di chuyển ngẫu nhiên, thực hiện di chuyển ngẫu nhiên tìm kiếm tới một mức nào đó, sau khi kết thúc phương pháp di chuyển ngẫu nhiên, thực hiện tiếp phương pháp phát tràn từ nút cuối cùng tìm được của phương pháp di chuyển ngẫu nhiên.

Về mặt trực quan thì có thể hình dung cách thực hiện của phương pháp như hình vẽ, và mô phỏng sau đây:

Giả sử nguồn phát động tìm kiếm file là nút A nào đó trong mạng và đồ thị trong hình vẽ này chỉ mang tính chất minh họa, không xét đến đồ thị có hướng. Nút A có 2 hàng xóm là nút B và nút O, vì sử dụng phương pháp di chuyển ngẫu nhiên nên tại chạng

thứ nhất này, nút A chọn ngẫu nhiên nút để gửi thông báo truy vấn là nút B. Nếu nút B có file mà nút A cần thì nút B gửi cho nút A thông báo queryHit và tìm kiếm kết thúc. Nếu như nút B không có file nút A cần thì nút B sẽ giảm giá trị TTL đi 1 trong thông báo truy vấn của nút A và chuyển tiếp ngẫu nhiên thông báo này tới hàng xóm của mình. Trong trường hợp này, nút B chuyển tiếp sang nút C. Quá trình cứ tiếp tục như thế



Hình 5. Phương pháp tìm kiếm lai ghép biến thể thứ nhất.

Quá trình tìm kiếm bằng phương pháp di chuyển ngẫu nhiên sẽ từ nút A, qua nút B, nút C, nút D và kết thúc tại nút E.

Sau các chặng, các nút truy vấn bằng phương pháp di chuyển ngẫu nhiên chưa tìm kiếm thấy file, ta thực hiện tiếp phương pháp phát tràn từ nút E.

Từ nút E bắt đầu thực hiện phương pháp phát tràn, phát tràn sang các nút hàng xóm của E là: F, G, H, I. Và từ các nút mới này lại thực hiện phát tràn tiếp. Quá trình cứ tiếp tục như thế.

Quá trình tìm kiếm sẽ kết thúc khi thấy file, hoặc giá trị TTL đã hết, hoặc như trong mô phỏng chúng tôi sử dụng số lượng tin nhắn truy vấn, để đi qua tất cả các chặng có thể tìm kiếm, khi số lượng thông báo truy vấn đã hết, thì quá trình tìm kiếm sẽ kết thúc.

Như vậy, phương pháp này có ý tưởng thực hiện khá đơn giản và để xem ảnh hưởng của phương pháp với các mô hình mạng đã nêu ra sao, và kết quả so với các đề xuất trước đó tốt hơn hay xấu hơn thì trong chương mô phỏng chúng tôi sẽ làm rõ vấn đề này.

4.1.1.2. Mã giả của phương pháp

Việc mô tả hoạt động của phương pháp trong phần nội dung trên để hiểu về cách thức tìm kiếm của một nút khi sử dụng phương pháp, còn trong phần nội dung này, chúng tôi không đề cập đến việc tìm thấy file hay không mà chỉ đưa ra cách thức tìm kiếm, và kết quả là tìm được vùng bao phủ có số lượng các nút càng lớn càng tốt. Do đó mã giả cho phương pháp như sau:

VariantSearch1()

{

*/*Thực hiện phương pháp di chuyển ngẫu nhiên trước*/*

Khởi tạo hàng đợi Q rỗng;

Đánh dấu đã thăm đỉnh nguồn truy vấn;

Xen đỉnh nguồn truy vấn vào hàng đợi Q;

for (mỗi một lần thực hiện walker)

```

{
    Lấy đỉnh ở cuối hàng đợi mà khác rỗng, ra khỏi Q;
    Chuyển thông báo truy vấn tới 1 đỉnh hàng xóm được lựa chọn ngẫu nhiên;
    Đánh dấu đã thăm đỉnh hàng xóm;
    Xen đỉnh hàng xóm vào Q;
}

```

*/*Thực hiện phương pháp phát tràn với lượng thông báo truy vấn còn lại*/*

```

while( số lượng thông báo truy vấn vẫn còn)
{
    Lấy tuần tự các đỉnh ra khỏi hàng đợi, bắt đầu từ đỉnh cuối cùng, khác rỗng
    của phương pháp tìm kiếm di chuyển ngẫu nhiên ở trên;

    while(mỗi đỉnh hàng xóm của đỉnh vừa lấy ra khỏi hàng đợi và số lượng
    thông báo truy vấn vẫn còn)
    {
        /*Không cho phép đỉnh được lấy ra gửi truy vấn ngược lại đỉnh đã
        gửi truy vấn tới nó trước đó */

        if ( đỉnh hàng xóm!= đỉnh thứ tự trước của đỉnh được lấy ra)
        {
            Chuyển thông báo truy vấn tới đỉnh hàng xóm;
            Đánh dấu đã thăm đỉnh hàng xóm;
            Xen đỉnh hàng xóm vào hàng đợi Q;
        }
    }
}

```

```

    }
}
}

```

4.1.2. Phương pháp tìm kiếm lai ghép biến thể thứ hai

4.1.2.1. Ý tưởng của phương pháp

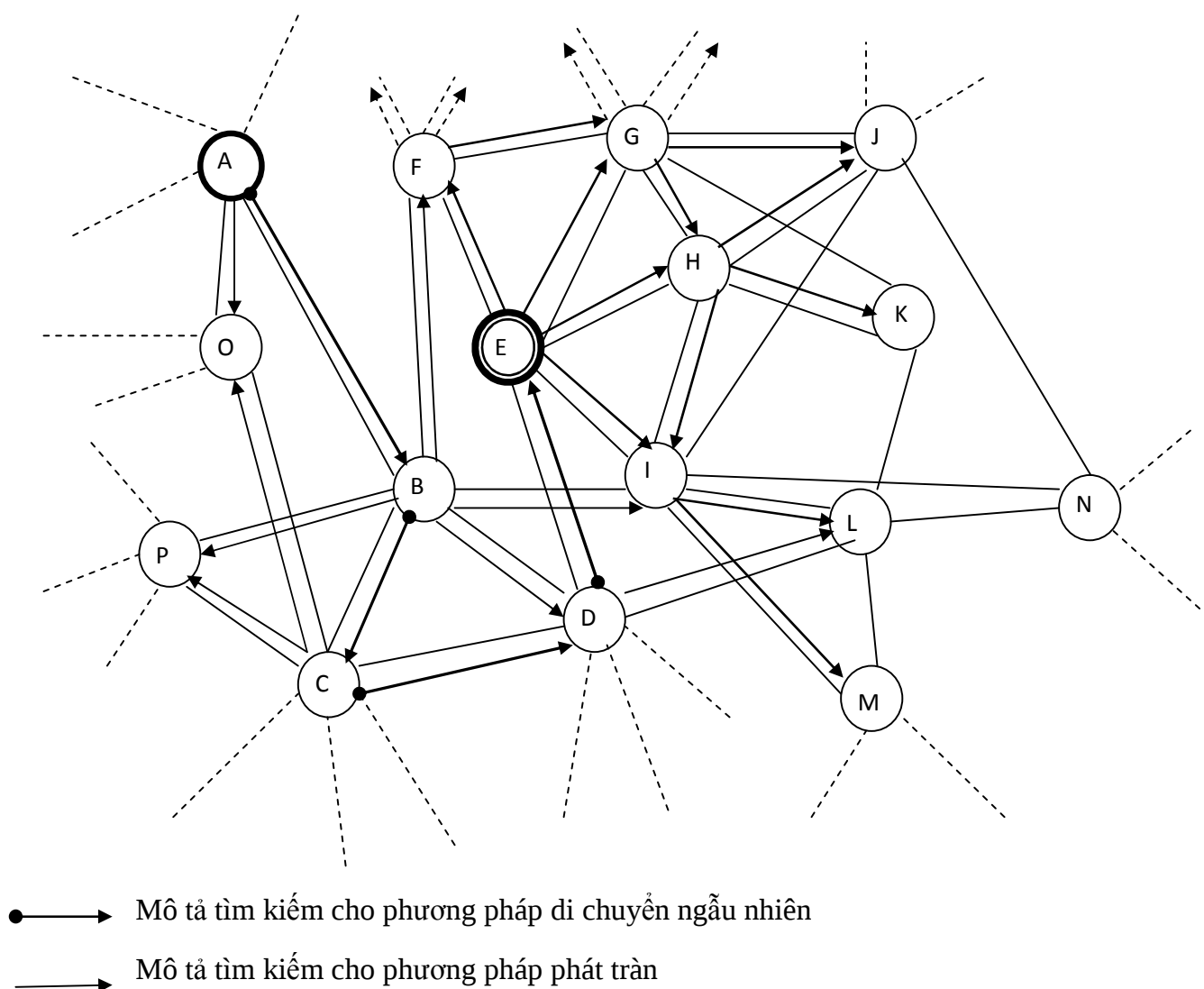
Về mặt ý tưởng trong phương pháp lai ghép biến thể thứ hai này khác so với biến thể thứ nhất. Với biến thể lai ghép thứ hai này, cảm nhận trực quan có thể thấy giống như tạo ra một trục, rồi sau đó thực hiện lan tỏa từ trục. Tức là, trong giai đoạn đầu của phương pháp, chúng tôi sử dụng phương pháp di chuyển ngẫu nhiên, để tìm kiếm tất cả các peer có thể tìm thấy được bằng phương pháp này, sau đó đánh dấu lại tất cả các peer mà phương pháp tìm được, sau khi kết thúc phương pháp, từ những peer đánh dấu được chúng tôi thực hiện phương pháp phát tràn tiếp. Việc này giống như, tạo ra một lúc nhiều nguồn lan tỏa hơn là một nguồn lan tỏa như đã làm, mức độ bao trùm sẽ rộng hơn, tuy nhiên lượng thông báo trùng lặp sẽ tăng hơn so với biến thể đầu tiên.

Về mặt trực quan và hoạt động tìm kiếm của phương pháp được biểu diễn như hình vẽ và mô tả dưới đây:

Vấn giả sử là nguồn truy vấn tìm kiếm là nút A và vấn giả sử sau khi thực hiện phương pháp di chuyển ngẫu nhiên qua các peer có thể mà vẫn chưa tìm được file. Sau khi thực hiện phương pháp di chuyển ngẫu nhiên, ta có danh sách các peer được phát hiện thấy là : A, B, C, D, E.

Thay vì thực hiện như biến thể 1 thì tại đây các nút A-B-C-D-E sẽ là nguồn phát tràn, lan tỏa tìm kiếm tiếp. Khi đó ở chặng tiếp theo này, có thêm được các nút mới chưa được tìm bao giờ là : O, P, F, G, H, I, L.

Giả sử là tại vị trí nút P là có file cần tìm, thì khi đó nút B gửi trước nút C, nên nút P sẽ trả về queryHit cho nút A và quá trình tìm kiếm kết thúc, còn thông báo nút C gửi đến nút P, bị nút P đối chiếu và loại bỏ. Còn nếu, trong trường hợp nút P không phải là nút có chứa file cần tìm, thì khi đó nút P sẽ có thêm thông báo truy vấn của nút C, gọi là hiện tượng nhận dư thừa lượng thông báo truy vấn.



Hình 6. Phương pháp tìm kiếm lai ghép biến thể thứ hai.

Tại chặng tiếp theo thì các nút vừa được xác định sẽ thực hiện tiếp. Quá trình tìm kiếm cứ tiếp tục cho đến khi có file được tìm thấy, hoặc giá trị TTL hết, hoặc số lượng thông báo truy vấn đã hết (đây là giá trị để dừng chương trình trong mô phỏng của chúng tôi).

Vấn đề trong phương pháp lai ghép này là chúng tôi không loại bỏ trùng lặp trong trường hợp nút đã được tìm thấy trong phương pháp di chuyển ngẫu nhiên mà vẫn nhận thêm thông báo truy vấn từ nút nguồn lần nữa khi thực hiện phương pháp phát tràn.

4.1.2.2. Mã giả của phương pháp

So với phương pháp lai ghép biến thể 1, sự khác biệt chính trong phương pháp này là ở bước phát tràn. Tức là sẽ thực hiện đồng thời phát tràn từ các đỉnh tìm được từ phương pháp di chuyển ngẫu nhiên trước đó. Như vậy trong mã giả của phương pháp này sẽ thực hiện phát tràn từ đỉnh đầu tiên trong hàng đợi.

VariantSearch2()

```
{  
    /*Thực hiện phương pháp di chuyển ngẫu nhiên trước*/  
  
    Khởi tạo hàng đợi Q rỗng;  
    Đánh dấu đã thăm đỉnh nguồn truy vấn;  
    Xen đỉnh nguồn truy vấn vào hàng đợi Q;  
  
    for (mỗi một lần thực hiện walker)  
    {  
        Lấy đỉnh ở cuối hàng đợi mà khác rỗng, ra khỏi Q;  
        Chuyển thông báo truy vấn tới 1 đỉnh hàng xóm được lựa chọn ngẫu nhiên;  
        Đánh dấu đã thăm đỉnh hàng xóm;  
        Xen đỉnh hàng xóm vào Q;  
    }  
  
    /*Thực hiện phương pháp phát tràn với lượng thông báo truy vấn còn lại*/  
  
    while( số lượng thông báo truy vấn vẫn còn)  
    {
```

Lấy tuần tự các đỉnh ra khỏi hàng đợi, bắt đầu từ đỉnh đầu tiên của phương pháp tìm kiếm di chuyển ngẫu nhiên ở trên;

while(mỗi đỉnh hàng xóm của đỉnh vừa lấy ra khỏi hàng đợi và số lượng thông báo truy vấn vẫn còn)

{

*/*Không cho phép đỉnh được lấy ra gửi truy vấn ngược lại đỉnh đã gửi truy vấn tới nó trước đó*/*

if (đỉnh hàng xóm!= đỉnh thứ tự trước của đỉnh được lấy ra)

{

Chuyển thông báo truy vấn tới đỉnh hàng xóm;

Đánh dấu đã thăm đỉnh hàng xóm;

Xen đỉnh hàng xóm vào hàng đợi Q;

}

}

}

}

4.2. Phương pháp tìm kiếm lai ghép sử dụng phát tràn cải tiến

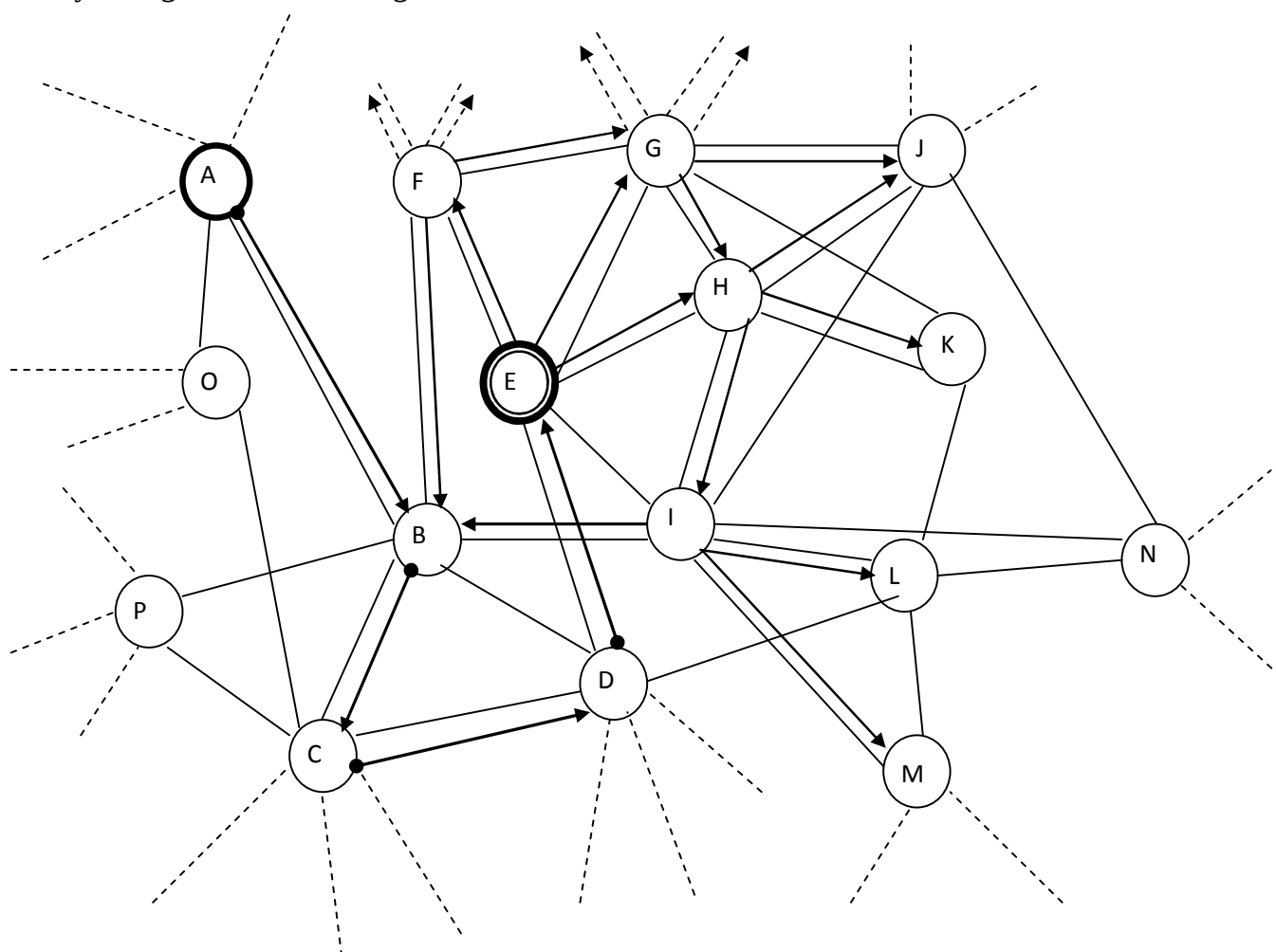
Do sử dụng phương pháp phát tràn cổ điển để tìm kiếm thì việc phát tràn sẽ gây ra lượng dư thừa thông báo truy vấn không cần thiết. Và đã có phương pháp được đề xuất hạn chế nhược điểm này, mà vẫn sử dụng đặc trưng của phương pháp phát tràn đó là phương pháp phát tràn cải tiến. Cách thức tìm kiếm của phương pháp này vẫn giống như phương pháp phát tràn thông thường nhưng tại mỗi bước gửi thông báo truy vấn thay vì gửi tới toàn bộ hàng xóm thì chỉ gửi tới một tập con các hàng xóm của nó và việc lựa chọn hàng xóm là ngẫu nhiên, bất kỳ, thông tin tham khảo thêm trong [14] và [20].

Dựa trên những phân tích và kết quả trả về trong tài liệu [14], [20], thấy rằng phương pháp phát tràn cải tiến là một phương pháp tốt và tốt trên hình trạng mạng mà chúng tôi dùng mô phỏng. Do đó chúng tôi đã đề xuất thêm 2 phương pháp tìm kiếm lai ghép nữa từ việc kết hợp phương pháp này bằng cách thay thế phương pháp phát tràn cổ điển và chúng tôi gọi là biến thể thứ ba và biến thể thứ tư.

4.2.1. Phương pháp tìm kiếm lai ghép biến thể thứ ba

4.2.1.1. Ý tưởng của phương pháp

Về mặt ý tưởng thì giống như ý tưởng của phương pháp tìm kiếm lai ghép biến thể thứ nhất nhưng chỉ khác cách thức thực hiện của phương pháp phát tràn. Và sự khác biệt này chúng tôi sẽ mô tả trong ví dụ sau:



Hình 7. Phương pháp tìm kiếm lai ghép biến thể thứ ba.

Sau khi đã thực hiện phương pháp di chuyển ngẫu nhiên như trong ví dụ của phương pháp tìm kiếm biến thể thứ nhất, điểm nút E là điểm nút cuối cùng. Từ E thực hiện tiếp phương pháp phát tràn cải tiến, giả sử tại E chọn 3 hàng xóm để chuyển tiếp thông báo truy vấn là: nút F, G, H. Và các nút này lại thực hiện tương tự như E đã làm, xét với trường hợp nút H, lúc này chỉ có 3 hàng xóm, và số hàng xóm bằng số lượng nút lựa chọn mỗi lần gửi truy vấn nên H gửi thông báo tới J, K, I. Quá trình cứ thế tiếp tục.

Trong trường hợp này, G nhận thêm thông báo từ F, và H nhận thêm thông báo từ G.

4.2.1.2. Mã giả của phương pháp

Do chọn lựa hàng xóm để phát tràn là khác so với phương pháp phát tràn thông thường nên mã giả về cơ bản là như biến thể thứ nhất nhưng ở giai đoạn phát tràn thì phải thêm đoạn mã thực hiện việc lựa chọn tập hàng xóm ngẫu nhiên để truyền truy vấn.

```
VariantSearch3()
```

```
{
```

```
    /*Thực hiện phương pháp di chuyển ngẫu nhiên trước*/
```

```
    Khởi tạo hàng đợi Q rỗng;
```

```
    Đánh dấu đã thăm đỉnh nguồn truy vấn;
```

```
    Xen đỉnh nguồn truy vấn vào hàng đợi Q;
```

```
    for (mỗi một lần thực hiện walker)
```

```
    {
```

```
        Lấy đỉnh ở cuối hàng đợi mà khác rỗng, ra khỏi Q;
```

```
        Chuyển thông báo truy vấn tới 1 đỉnh hàng xóm được lựa chọn ngẫu nhiên;
```

```
        Đánh dấu đã thăm đỉnh hàng xóm;
```

```
        Xen đỉnh hàng xóm vào Q;
```

```
    }
```

*/*Thực hiện phương pháp phát tràn với lượng thông báo truy vấn vẫn còn lại*/*

while (số lượng thông báo truy vấn vẫn còn)

{

Lấy tuần tự các đỉnh ra khỏi hàng đợi, bắt đầu từ đỉnh cuối cùng, khác rỗng của phương pháp tìm kiếm di chuyển ngẫu nhiên ở trên;

*/*Trường hợp bậc nhỏ hơn, thì giống như là biến thể thứ nhất*/*

if(bậc của nút đang xét $\leq$ số lượng hàng xóm yêu cầu được chọn)

{

while (mỗi đỉnh hàng xóm của đỉnh vừa lấy ra khỏi hàng đợi và số lượng thông báo truy vấn vẫn còn)

{

*/*Không cho phép đỉnh được lấy ra gửi truy vấn ngược lại đỉnh đã gửi truy vấn tới nó trước đó */*

if (đỉnh hàng xóm!= đỉnh thứ tự trước của đỉnh được lấy ra)

{

Chuyển thông báo truy vấn tới đỉnh hàng xóm;

Đánh dấu đã thăm đỉnh hàng xóm;

Xen đỉnh hàng xóm vào hàng đợi Q;

}

}

}

```

/*Trường hợp bậc lớn hơn, thì thực hiện chọn tập con hàng xóm*/
else
{
    for ( mỗi lần chọn lựa hàng xóm từ số lượng hàng xóm yêu cầu)
    if (số lượng thông báo truy vấn vẫn còn)
    {
        Lựa chọn đỉnh hàng xóm ngẫu nhiên trong tập hàng xóm;
        Đánh dấu đã thăm đỉnh hàng xóm;
        Xen đỉnh hàng xóm vào hàng đợi Q;
    }
}
}
}

```

4.2.2. Phương pháp tìm kiếm lai ghép biến thể thứ tư

4.2.2.1. Ý tưởng của phương pháp

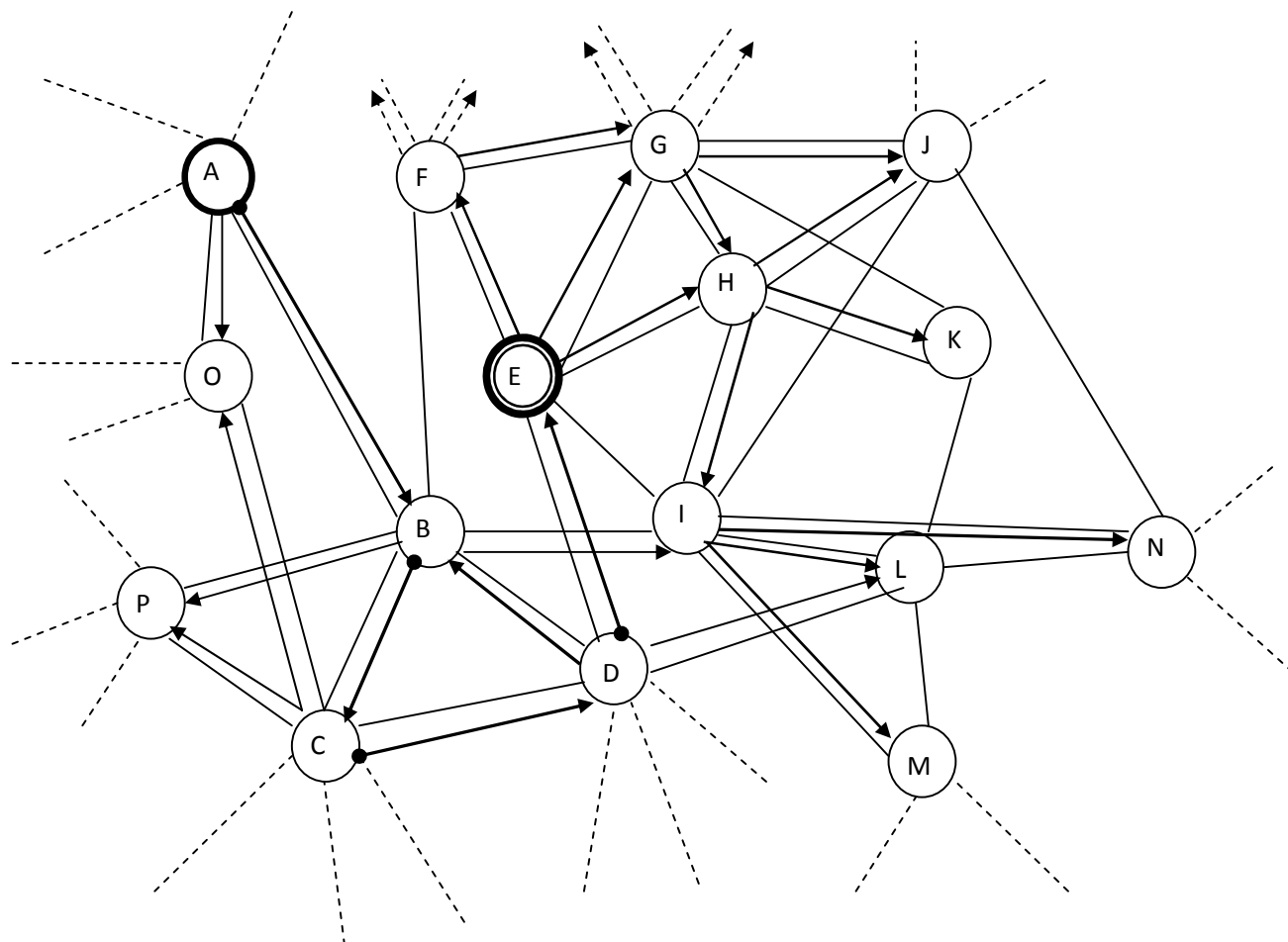
Ý tưởng của phương pháp này giống như ý tưởng của phương pháp tìm kiếm lai ghép biến thể thứ hai nhưng khác ở bước thực hiện phát tràn vì phương pháp phát tràn lúc này được thay bằng phương pháp phát tràn cải tiến.

Có thể mô tả hoạt động của phương pháp tìm kiếm lai ghép biến thể thứ tư như sau:

Trước khi thực hiện phương pháp phát tràn cải tiến thì việc thực hiện di chuyển ngẫu nhiên tương tự như trong mô tả về phương pháp tìm kiếm lai ghép biến thể thứ hai. Tức là thu được danh sách các điểm nút : A-B-C-D-E.

Tiến hành phương pháp phát tràn cải tiến từ những nút này, giả sử số lượng hàng xóm của tập con là 3 hàng xóm khi đó thì A và C thực hiện như phương pháp phát tràn thông thường, và vẫn giống như trong biến thể tìm kiếm thứ hai. Do nút B có nhiều hơn 3 hàng xóm nên khi đó giả sử nút B gửi thông báo truy vấn tới: P, C, I. Tương tự với nút D,

E thông báo truy vấn gửi tương ứng với nút nguồn tới: B, E, L và gửi tới: F, G, H. Quá trình thực hiện phát tràn cứ thế tiếp tục từ các nút vừa tìm được.



● → Mô tả tìm kiếm cho phương pháp di chuyển ngẫu nhiên

→ Mô tả tìm kiếm cho phương pháp phát tràn

Hình 8. Phương pháp tìm kiếm lai ghép biến thể thứ tư.

4.2.2.2. Mã giả của phương pháp

Mã giả trong phương pháp này chỉ thêm một chút so với phương pháp tìm kiếm lai ghép biến thể thứ hai trong phần thực hiện phát tràn.

VariantSearch4()

{

*/*Thực hiện phương pháp di chuyển ngẫu nhiên trước*/*

Khởi tạo hàng đợi Q rỗng;

Đánh dấu đã thăm đỉnh nguồn truy vấn;

Xen đỉnh nguồn truy vấn vào hàng đợi Q;

for (mỗi một lần thực hiện walker)

{

Lấy đỉnh ở cuối hàng đợi mà khác rỗng, ra khỏi Q;

Chuyển thông báo truy vấn tới 1 đỉnh hàng xóm được lựa chọn ngẫu nhiên;

Đánh dấu đã thăm đỉnh hàng xóm;

Xen đỉnh hàng xóm vào Q;

}

*/*Thực hiện phương pháp phát tràn với lượng thông báo truy vấn còn lại*/*

while(số lượng thông báo truy vấn vẫn còn)

{

Lấy tuần tự các đỉnh ra khỏi hàng đợi, bắt đầu từ đỉnh đầu tiên của phương pháp tìm kiếm di chuyển ngẫu nhiên ở trên;

*/*Trường hợp bậc nhỏ hơn, thì giống như là biến thể thứ nhất*/*

```

if(bậc của nút đang xét  $\leq$  số lượng hàng xóm yêu cầu được chọn)
{
    while (mỗi đỉnh hàng xóm của đỉnh vừa lấy ra khỏi hàng đợi và số
    lượng thông báo truy vấn vẫn còn)
    {
        /*Không cho phép đỉnh được lấy ra gửi truy vấn ngược lại
        đỉnh đã gửi truy vấn tới nó trước đó */

        if ( đỉnh hàng xóm!= đỉnh thứ tự trước của đỉnh được lấy ra)
        {
            Chuyển thông báo truy vấn tới đỉnh hàng xóm;
            Đánh dấu đã thăm đỉnh hàng xóm;
            Xen đỉnh hàng xóm vào hàng đợi Q;
        }
    }
}

/*Trường hợp bậc lớn hơn, thì thực hiện chọn tập con hàng xóm*/
else
{
    for ( mỗi lần chọn lựa hàng xóm từ số lượng hàng xóm yêu cầu)
    if (số lượng thông báo truy vấn vẫn còn)
    {
        Lựa chọn đỉnh hàng xóm ngẫu nhiên trong tập hàng xóm;
        Đánh dấu đã thăm đỉnh hàng xóm;
        Xen đỉnh hàng xóm vào hàng đợi Q;
    }
}

```

```

    }
  }
}

```

Như vậy, trong chương này chúng tôi đã trình bày về các phương pháp lai mới mà chúng tôi đề xuất, bao gồm ý tưởng thực hiện và mã giả của chúng.

Từ chương 3 và chương 4, chúng tôi đã không chỉ giới thiệu về các phương pháp được đề xuất trước đây và các phương pháp mới của chúng tôi mà còn phân tích từng phương pháp, phân tích tổng thể về vấn đề tìm kiếm trong mạng ngang hàng thuần túy. Việc khảo sát, đánh giá phương pháp nào là tốt so với phương pháp nào và trên dạng đồ thị nào, dựa trên tiêu chí nào để đánh giá thì trong các chương tiếp theo của chúng tôi sẽ trình bày chi tiết.

CHƯƠNG 5. MÔ PHÒNG VÀ ĐÁNH GIÁ HIỆU NĂNG

Trong chương này, chúng tôi tiến hành mô phỏng các phương pháp tìm kiếm đã nêu trên từng dạng đồ thị đã giới thiệu. Trong từng dạng đồ thị, chúng tôi sẽ tiến hành mô phỏng với mỗi dạng đồ thị là 60 lần mô phỏng, tương đương với thử nghiệm 60 mẫu đồ thị ngẫu nhiên của cùng 1 dạng đồ thị cho một phương pháp tìm kiếm. Số lượng thông báo truy vấn tại mỗi lần mô phỏng là như nhau đối với các phương pháp, trong mô phỏng chúng tôi sử dụng 5^5 thông báo và N thông báo (N là số rất lớn, trong mô phỏng chúng tôi đặt $N = 50000$), trong quá trình tìm kiếm nếu số lượng thông báo này đã sử dụng hết thì phương pháp sẽ dừng tìm kiếm. Số lượng điểm nút hình thành lên một đồ thị là 100000 điểm nút đối với tất cả các dạng đồ thị. Mỗi một lần thực hiện mô phỏng, giá trị điểm nút nguồn yêu cầu tài nguyên được chọn lựa khác nhau, điểm nút bất kỳ nào đó. Và mức độ phân bố tài liệu trong mạng là 0.001, tức là với 100000 điểm nút thì có 100 điểm nút chứa tài nguyên cần tìm. Sự phân bố này là đồng đều và ngẫu nhiên trên các điểm nút, giả sử không có điểm nút nào chứa nhiều hơn 1 tài nguyên. Các tham số này là tham số chung cho tất cả các lần mô phỏng, trên tất cả các đồ thị.

Nếu đánh giá phương pháp tìm kiếm dựa trên tiêu chí tìm thấy tài nguyên và bao nhiêu lượng tài nguyên được tìm thấy (các tài nguyên là phân bố đồng đều ngẫu nhiên) thì chúng tôi đã tiến hành thu thập kết quả. Nhưng vì đánh giá theo tiêu chí tìm thấy tài nguyên và lượng tài nguyên tìm thấy là không bao quát tất cả các trường hợp, nên chúng tôi không nêu trong phần kết quả để đánh giá, cũng không dựa vào đây để đánh giá. Việc đánh giá hiệu năng của phương pháp phải mang tính tổng quát cho tất cả các trường hợp có thể, do đó chúng tôi xét trên tiêu chí : mức độ bao phủ, số lượng nút nhận truy vấn dư thừa,..vv. Nếu giá trị mức độ bao phủ càng lớn thì khả năng tìm kiếm thấy tài nguyên càng lớn.

5.1. Các đơn vị đo hiệu năng trong mô phỏng

5.1.1. Mức độ bao phủ

Mức độ bao phủ của phương pháp tìm kiếm là tổng số các điểm nút mà phương pháp có thể tìm được (bao gồm cả các nút có chứa file cần tìm và nút không chứa).

Tham số này thể hiện khả năng mở rộng vùng tìm kiếm của phương pháp, giá trị tham số này càng lớn thì hiệu quả của phương pháp tìm kiếm càng tốt, trong chương trình

mô phỏng của mình, chúng tôi gắn với tập $V=\{v_0, v_1,...,v_n\}$ trong đó v_i là số lượng đỉnh được thăm i lần khi sử dụng phương pháp.

Rõ ràng giá trị của v_0 càng nhỏ thì phương pháp tìm kiếm càng tốt

Ký hiệu cho tham số này là : C – Coverage.

5.1.2. Tỷ lệ thành công

Tỷ lệ thành công là xác suất để một truy vấn là thành công, tức là có tối thiểu một lần truy vấn thành công. Giả sử rằng, các tài nguyên tìm kiếm là được phân bố đều trên mạng với tỉ lệ phân bố bản sao R , hay mức độ phổ biến của tài liệu . Khi đó tỷ lệ thành công ký hiệu là SR – Success Rate, được tính như sau:

$$SR = 1 - (1-R)^C \quad (3)$$

Giả sử trong một mạng có n nút, và có k nút được phân tài liệu, và khả năng phân bố các nút là như nhau, khi đó tỉ lệ phân bố bản sao là $R = k / n$

Sau khi kết thúc phương pháp tìm kiếm, tìm được C nút

Như vậy tỉ lệ thành công với vùng bao phủ C là :

$$SR = 1 - (1 - k/n)^C \quad (4)$$

5.1.3. Số lượng truy vấn thành công

Giả sử rằng các tài nguyên tìm kiếm được phân bố đều trên mạng với tỉ lệ phân bố bản sao là R , và vùng bao phủ C . Khi đó số lượng các truy vấn thành công, được ký hiệu là QH - Query Hits, được tính theo tài liệu [20] như sau:

$$QH = R * C \quad (5)$$

Như vậy QH phụ thuộc lớn vào C

Từ các công thức (3), (5), nếu ta cho R là một giá trị cho trước thì giải thuật nào có C càng lớn thì giải thuật đó có tỉ lệ tìm thấy số file thành công càng cao và truy vấn thành công cũng càng cao $\rightarrow$ Đó là phương pháp tốt.

Như vậy, trong mô phỏng chúng tôi chỉ thực hiện đo đặc giá trị C, và sẽ tính toán các giá trị còn lại sau khi có C. Các kết quả của mô phỏng sẽ được trình bày trong chương tiếp theo của khóa luận.

5.1.4. Hiệu quả truy vấn

Là một đơn vị đo đánh giá hiệu suất của phương pháp tìm kiếm, được nêu ra trong tài liệu [20][22]. Theo tiêu chí đánh giá này, phương pháp tìm kiếm có hiệu quả tốt là phương pháp sử dụng lượng thông báo truy vấn trong suốt quá trình tìm kiếm phải có tỉ lệ thành công cao.

Ký hiệu là QE - Query Efficiency

Công thức tính :

$$\begin{aligned} QE &= QH / (\text{Số lượng thông báo truy vấn} / \text{Kích thước mạng}) \\ &= QH / \text{Trung bình số lượng thông báo trên 1 nút.} \end{aligned} \quad (6)$$

Trong công thức này giá trị QE được hiểu là QE %

5.1.5. Số lượng nút nhận truy vấn dư thừa

Trong mỗi một lần tìm kiếm, phương pháp tìm kiếm có tính chất ngẫu nhiên, các dạng đồ thị sử dụng cũng là các đồ thị có tính chất ngẫu nhiên và thông tin định tuyến toàn mạng là không biết trước do đó có hiện tượng trùng lặp thông báo truy vấn.

Tham số độ đo này là kết quả chúng tôi tính toán được sau khi mô phỏng. Tham số này đánh giá tổng số nút được thăm nhiều hơn một lần, tức là số lần thông báo truy vấn tới một nút nhiều hơn một lần. Tham số này phản ánh mức độ dư thừa thông báo không kiểm soát được và gây lãng phí băng thông một cách không cần thiết. Tham số này có kết quả càng lớn thì số lượng nút sẽ chịu tải xử lý càng cao và ảnh hưởng đến giao tiếp chung toàn mạng. Tham số phản ánh về khía cạnh nhược điểm của phương pháp tìm kiếm.

Ký hiệu: SQ - Sum of Redundant Query

$$SQ = C - \text{số lượng nút chỉ được truy vấn 1 lần.} \quad (7)$$

5.2. Kết quả mô phỏng trên đồ thị luật hàm mũ

Trong bảng kết quả của mô phỏng, cột đầu tiên là các tham số độ đo chẳng hạn như: C_{mean} : mức độ bao phủ trung bình, C_{min} : mức độ bao phủ nhỏ nhất, C_{max} : mức độ bao phủ lớn nhất trong 60 lần thử nghiệm, σ : độ lệch chuẩn, các ký hiệu khác đã được giải thích ở mục trên. Các cột tiếp theo là các giá trị tương ứng với tham số độ đo của 1 phương pháp tìm kiếm. Kết quả mức độ bao phủ trung bình tốt nhất được bôi đậm.

Ký hiệu mà chúng tôi quy ước cho phương pháp phát tràn là F (Flooding search), phương pháp di chuyển ngẫu nhiên là R (Random walk search), phương pháp lai ghép được đề xuất trong [14] là H (Hybrid search). Còn các phương pháp của chúng tôi: phương pháp lai ghép biến thể thứ nhất là V1 (1st variant of Hybrid search), phương pháp lai ghép biến thể thứ hai là V2 (2nd variant of Hybrid search), phương pháp lai ghép biến thể thứ ba là V3 (3rd variant of Hybrid search), và phương pháp lai ghép biến thể thứ tư là V4 (4th variant of Hybrid search).

Vì các phương pháp tìm kiếm lai ghép này đều sử dụng đến phương pháp di chuyển ngẫu nhiên cho nên chúng tôi chọn giá trị số lượng walker cho các phương pháp thử nghiệm này là 10 trong từng lần mô phỏng.

Trong mô phỏng, chúng tôi chỉ đề cập đến một phương pháp tìm kiếm lai đã được đề xuất là phương pháp tìm kiếm trong tài liệu [14] vì phương pháp này đề cập đến sự kết hợp các phương pháp đơn lẻ nhưng có mang tính chất kết hợp về kỹ thuật.

Các mô hình đồ thị được chúng tôi đề cập để đánh giá trong mô phỏng của mình là đồ thị luật hàm mũ với $\gamma = 2.1$ và $\gamma = 2.7$, tôi phân cụm.

Trên mỗi một đồ thị được mô phỏng, mỗi phương pháp tìm kiếm chúng tôi tiến hành tìm kiếm tự động 15 lần và lấy kết quả trung bình của 15 lần thực hiện. Việc thực hiện mô phỏng các phương pháp tìm kiếm trên 60 đồ thị của một dạng đồ thị cũng được thực hiện tự động, các kết quả thu được không chỉ biểu diễn bằng bảng kết quả số mà còn tổng kết biểu diễn dưới dạng biểu đồ cột để dễ dàng so sánh bằng trực quan.

5.2.1. Đồ thị luật hàm mũ với 5^5 thông báo truy vấn

5.2.1.1. Đồ thị luật hàm mũ với $\gamma = 2.1$

Bảng 1 Kết quả mô phỏng các phương pháp tìm kiếm trên đồ thị luật hàm mũ với 5^5 thông báo và $\gamma = 2.1$

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	2980.58	2572.61	2573.11	2980.45	2847.57	2670.59	2651.888
C_{min}	2889.67	2540.07	2538.07	2791.93	2580	2620.13	2551.867
C_{max}	3083.87	2611.73	2608.27	3063.93	3081.27	2725.87	2671.333
σ	43.932	15.387	15.53	50.236	117.872	25.791	25.752
SR	0.949	0.924	0.924	0.949	0.942	0.931	0.93
QH	2.981	2.573	2.573	2.98	2.848	2.671	2.652
QE	95.378	82.323	82.34	95.374	91.122	85.459	84.86
SQ	63.51	232.939	232.003	64.423	236.701	158.187	197.978

Từ bảng kết quả, chúng tôi thấy phương pháp phát tràn là tốt nhất, sau đó đến phương pháp lai ghép biến thể thứ nhất của chúng tôi (nếu như xét với số lượng peer lớn, thì coi như xấp xỉ bằng nhau) và phương pháp biến thể thứ hai cũng cho kết quả tốt, biến thể thứ ba và thứ 4 cho kết quả trung gian. Tuy nhiên, thì phương pháp biến thể thứ nhất và thứ hai có độ lệch chuẩn lớn hơn so với các phương pháp còn lại. Giá trị độ lệch chuẩn lớn phản ánh độ biến thiên của các kết quả mô phỏng thu được so với giá trị trung bình của tất cả các phương là cao.

5.2.1.2. Đồ thị luật hàm mũ với $\gamma = 2.7$

Kết quả thu được trong mô phỏng này, phương pháp phát tràn vẫn là phương pháp tốt nhất, sau đó đến biến thể thứ ba và thứ nhất của chúng tôi, còn biến thể thứ hai và thứ tư lại cho kết quả tồi tệ nhất. Tuy phương pháp phát tràn cho giá trị mức độ bao phủ trung bình lớn nhất nhưng tổng số nút nhận truy vấn dư thừa nhiều hơn so với biến thể thứ ba của chúng tôi.

Bảng 2 Kết quả mô phỏng các phương pháp tìm kiếm trên đồ thị luật hàm mũ với 5^5 thông báo và $\gamma = 2.7$

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	3014.866	2981.014	2980.461	3013.051	2574.232	3013.421	2850.119
C_{min}	2949.600	2970.467	2967.133	2875.800	2406.467	3004.600	2802.933
C_{max}	3066.333	2994.600	2996.000	3066.267	2712.600	3023.667	2968.800
σ	32.365	4.260	4.780	36.919	61.239	4.556	21.907
SR	0.951	0.949	0.949	0.951	0.924	0.951	0.942245
QH	3.015	2.981	2.980	3.013	2.574	3.013	2.850
QE	96.476	95.392	95.375	96.418	82.375	96.429	91.204
SQ	103.632	122.020	122.372	104.892	469.808	91.397	220.380

5.2.2. Đồ thị luật hàm mũ với N thông báo truy vấn

5.2.2.1. Đồ thị luật hàm mũ với $\gamma = 2.1$

Với N thông báo truy vấn thì phương pháp tìm kiếm lai ghép biến thể thứ ba của chúng tôi cho kết quả tốt nhất, sau đó là biến thể thứ tư, còn biến thể thứ hai cho kết quả tồi nhất, phương pháp phát tràn, di chuyển ngẫu nhiên và lai cho kết quả trung gian.

Phương pháp phát tràn có tổng số nút nhận truy vấn dư thừa là thấp nhất.

Bảng 3 Kết quả mô phỏng các phương pháp tìm kiếm trên đồ thị luật hàm mũ với N thông báo và $\gamma = 2.1$

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	28540.1	29909.5	29907.8	28451	26424.2	30907.6	30795.439

$C_{\min}$	25646.3	29730.3	29731.3	25097.5	23597.3	30587.2	30483.4
$C_{\max}$	31139.7	30090.3	30065.2	31713.3	30013.1	31506.4	31211.067
σ	1407.03	81.3279	75.0908	1460.73	1263.92	177.187	151.6254
SR	1	1	1	1	1	1	1
QH	28.4501	29.9095	29.9078	28.451	26.4242	30.9076	30.795439
QE	56.9002	59.8191	59.8157	56.902	52.8484	61.8152	61.590878
SQ	11850	7680.57	7680.46	11875.9	12361.9	7355.85	7386.036

5.2.2.2. Đồ thị luật hàm mũ với $\gamma = 2.7$

Bảng 4 Kết quả mô phỏng các phương pháp tìm kiếm trên đồ thị luật hàm mũ với N thông báo và $\gamma = 2.7$.

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	30648.496	35608.714	35611.398	30559.425	27458.621	35930.992	35426.392
$C_{\min}$	28986.733	35482.533	35460.667	28752.467	26020.667	35780.333	35248.200
$C_{\max}$	32564.133	35715.133	35702.333	32690.667	29655.333	36051.067	35589.667
σ	750.709	49.177	50.906	876.404	716.013	59.892	83.295
SR	1.000	1.000	1.000	1.000	1.000	1.000	1.000
QH	30.648	35.609	35.611	30.559	27.459	35.931	35.426
QE	61.297	71.217	71.223	61.119	54.917	71.862	70.853
SQ	10345.080	9335.130	9331.882	10365.120	11609.580	9541.426	9404.772

Đối với mô phỏng này, thì kết quả của phương pháp tìm kiếm lai đề xuất trong tài liệu [14] lại cho kết quả tốt nhất, sau đó là phương pháp tìm kiếm di chuyển ngẫu nhiên.

Kết quả với phương pháp của chúng tôi và phương pháp phát tràn cho kết quả thấp hơn.

5.3. Kết quả mô phỏng trên tô pô phân cụm

5.3.1. Mô phỏng trên tô pô phân cụm với 5^5 thông báo truy vấn

5.3.1.1. Tô pô phân cụm với K kích thước $l=100$

Bảng 5 Kết quả mô phỏng các phương pháp tìm kiếm trên tô pô với phân cụm K_{100}

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	100.200	179.377	160.039	100.039	100.813	103.494	103.758
C_{min}	100.200	139.400	130.600	100.000	100.200	100.867	100.267
C_{max}	100.200	218.200	200.933	100.667	107.533	109.533	111.600
σ	0.000	17.027	16.165	0.106	1.499	2.028	2.696
SR	0.095	0.164	0.148	0.095	0.096	0.098	0.099
QH	0.100	0.179	0.160	0.100	0.101	0.103	0.104
QE	3.206	5.740	5.121	3.212	3.226	3.312	3.320
SQ	97.483	172.979	139.088	99.092	100.363	100.588	101.016

5.3.1.2. Tô pô phân cụm với đồ thị ngẫu nhiên $G_{100,1/2}$

Bảng 6 Kết quả mô phỏng các phương pháp tìm kiếm trên tô pô phân cụm $G_{100,1/2}$

Độ đo	F	R	H	V1	V2	V3	V4

C_{mean}	111.906	236.518	204.928	103.937	108.863	107.694	107.577
C_{min}	107.800	175.533	163.400	1002.000	100.800	101.600	102.400
C_{max}	113.600	281.667	243.667	122.533	118.400	126.400	124.800
σ	0.855	25.026	18.052	4.895	4.419	4.546	4.316
SR	0.106	0.211	0.185	0.099	0.104	0.102	0.102
QH	0.112	0.237	0.205	0.104	0.109	0.108	0.108
QE	3.581	7.569	6.558	3.326	3.484	3.446	3.442
SQ	99.977	222.036	168.688	99.993	100.696	102.149	102.128

5.3.1.3. Tô pô phân cụm với đồ thị ngẫu nhiên $G_{100,1/5}$

Bảng 7 Kết quả mô phỏng các phương pháp tìm kiếm trên tô pô phân cụm $G_{100,1/5}$

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	107.032	371.354	298.993	103.179	112.836	116.697	119.023
C_{min}	105.400	288.067	228.000	101.200	105.200	104.333	105.800
C_{max}	108.867	472.933	356.000	111.667	137.667	136.533	154.400
σ	0.617	31.685	26.817	1.889	7.193	7.073	9.700
SR	0.102	0.310	0.259	0.098	0.107	0.110	0.112
QH	0.107	0.371	0.299	0.103	0.113	0.117	0.119
QE	3.425	11.883	9.568	3.302	3.611	3.734	3.809

SQ	101.483	329.026	235.036	101.282	103.118	105.000	106.339
----	---------	---------	---------	---------	---------	---------	---------

Từ kết quả các Bảng 5, 6, 7 thì kết quả phương pháp tìm kiếm theo phương pháp di chuyển ngẫu nhiên là tốt nhất, phương pháp biến thể lai thứ hai và phương pháp phát tràn cho kết quả thấp, còn phương pháp lai ghép biến thể thứ nhất cho kết quả tồi nhất. Phương pháp tìm kiếm lai ghép đề xuất trong tài liệu [14] cũng là một phương pháp tìm kiếm tốt.

5.3.2. Mô phỏng trên tô pô phân cụm với N thông báo truy vấn

Các kết quả mô phỏng trên tô pô phân cụm với N thông báo truy vấn như sau:

5.3.2.1. Tô pô phân cụm với K kích thước $l=100$

Bảng 8 Kết quả mô phỏng các phương pháp tìm kiếm trên cluster với phân cụm K_{100}

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	123.011	990.677	117.229	105.646	135.431	148.430	150.573
C_{min}	122.800	831.733	107.867	102.067	122.733	127.333	128.800
C_{max}	123.200	1152.600	189.933	143.200	183.000	174.733	167.267
σ	0.094	76.684	10.267	7.714	15.450	10.067	9.639
SR	0.116	0.629	0.111	0.100	0.127	0.138	0.140
QH	0.123	0.991	0.117	0.106	0.135	0.148	0.151
QE	0.246	1.981	0.234	0.211	0.271	0.297	0.301
SQ	102.802	970.919	741.972	102.976	103.888	116.133	116.722

5.3.2.2. Tô pô phân cụm với đồ thị ngẫu nhiên $G_{100,1/2}$

Bảng 9 Kết quả mô phỏng các phương pháp tìm kiếm trên cluster với phân cụm $G_{100,1/2}$

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	179.843	3733.171	2827.121	156.888	194.015	187.749	187.829
C_{min}	147.400	3410.600	2271.867	115.000	156.000	158.800	159.667
C_{max}	208.333	4107.467	3251.067	212.467	278.400	226.267	222.067
σ	11.331	145.733	203.436	17.486	23.332	13.413	11.887
SR	0.165	0.976	0.941	0.145	0.176	0.171	0.171
QH	0.180	3.733	2.827	0.157	0.194	0.188	0.188
QE	0.360	7.466	5.654	0.314	0.388	0.375	0.376
SQ	102.802	1669.491	1215.564	102.953	105.366	132.901	134.527

5.3.2.3. Tô pô phân cụm với đồ thị ngẫu nhiên $G_{100,1/5}$

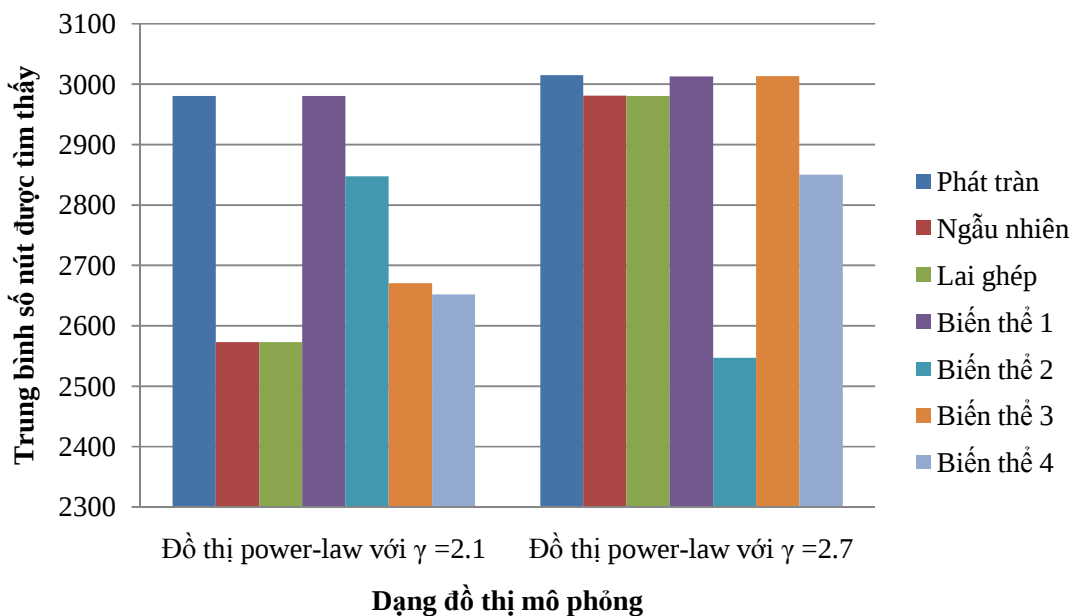
Bảng 10 Kết quả mô phỏng các phương pháp tìm kiếm trên tô pô phân cụm $G_{100,1/5}$

Độ đo	F	R	H	V1	V2	V3	V4
C_{mean}	113.424	1726.614	1265.070	104.876	128.688	273.071	276.028
C_{min}	112.333	1518.800	1090.800	102.000	112.733	235.133	227.267
C_{max}	114.400	1975.800	1449.800	114.600	155.600	328.133	335.000
σ	0.550	106.327	82.386	3.933	13.370	19.039	19.823
SR	0.107	0.822	0.718	0.100	0.121	0.239	0.241
QH	0.113	1.727	1.265	0.105	0.129	0.273	0.276

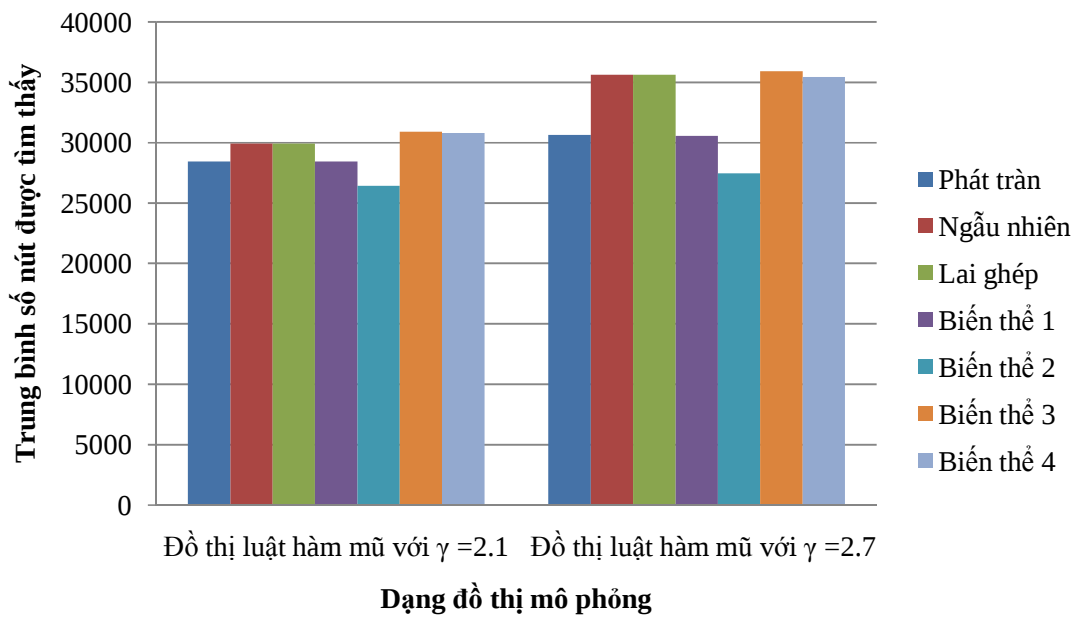
QE	0.227	3.453	2.530	0.210	0.257	0.546	0.552
SQ	146.698	3444.922	2595.674	126.747	155.504	186.929	188.454

Từ kết quả bảng 8, bảng 9 và bảng 10 cho thấy kết quả trên tô pô phân cụm thì phương pháp di chuyển ngẫu nhiên luôn cho kết quả tốt nhất, phương pháp lai đề xuất trong tài liệu [14] cũng cho kết quả tốt, còn các phương pháp còn lại cho kết quả tồi, tuy nhiên so với phương pháp phát tràn thì biến thể thứ hai, thứ ba và thứ tư của chúng tôi lại cho kết quả tốt hơn. Như vậy đối với tô pô phân cụm thì phương pháp tìm kiếm ngẫu nhiên đạt hiệu quả tìm kiếm rất cao, nhưng tổng số nút nhận truy vấn dư thừa cũng là cao nhất.

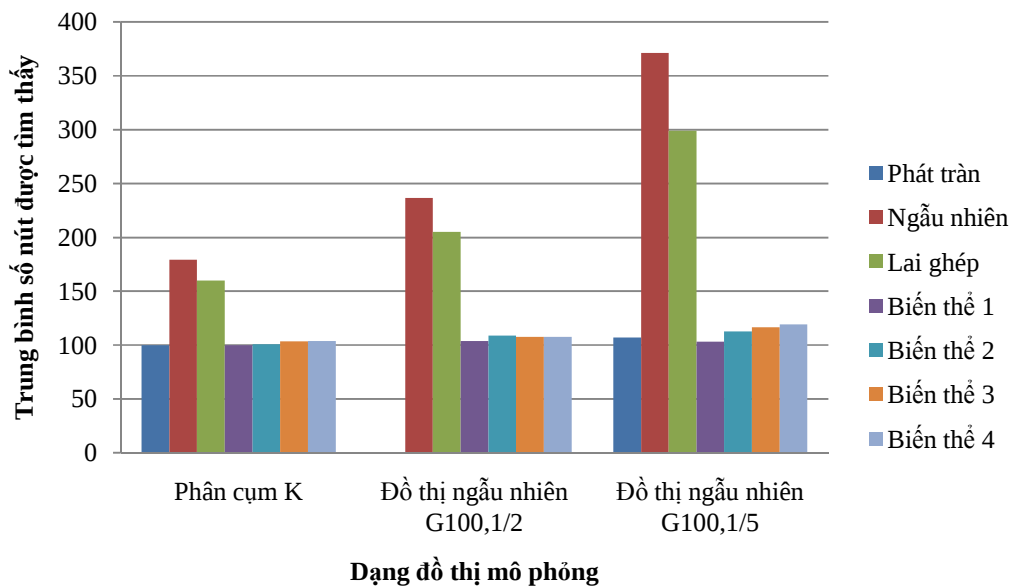
Để nhìn nhận về kết quả mô phỏng trực quan hơn, chúng tôi biểu diễn dưới dạng biểu đồ các kết quả này, trong đó trục tung miêu tả giá trị C_{mean} mà mỗi phương pháp tìm kiếm đạt được, trục hoành biểu diễn các phương pháp tìm kiếm của từng dạng đồ thị. Bởi vì các tiêu chí đánh giá đều phụ thuộc tỉ lệ vào C_{mean} theo công thức đã đưa ra nên chỉ cần biểu diễn C_{mean} , trừ tiêu chí về số nút nhận thông báo truy vấn dư thừa không phụ thuộc vào C_{mean} .



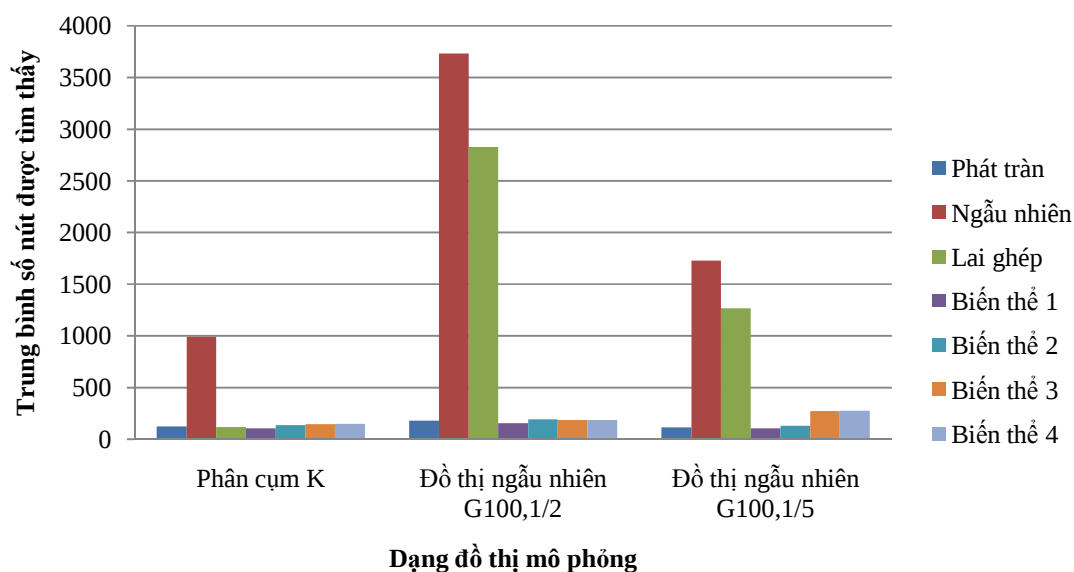
Biểu đồ 1. Kết quả mức độ bao phủ của các phương pháp tìm kiếm trên đồ thị luật hàm mũ với 5^5 thông báo.



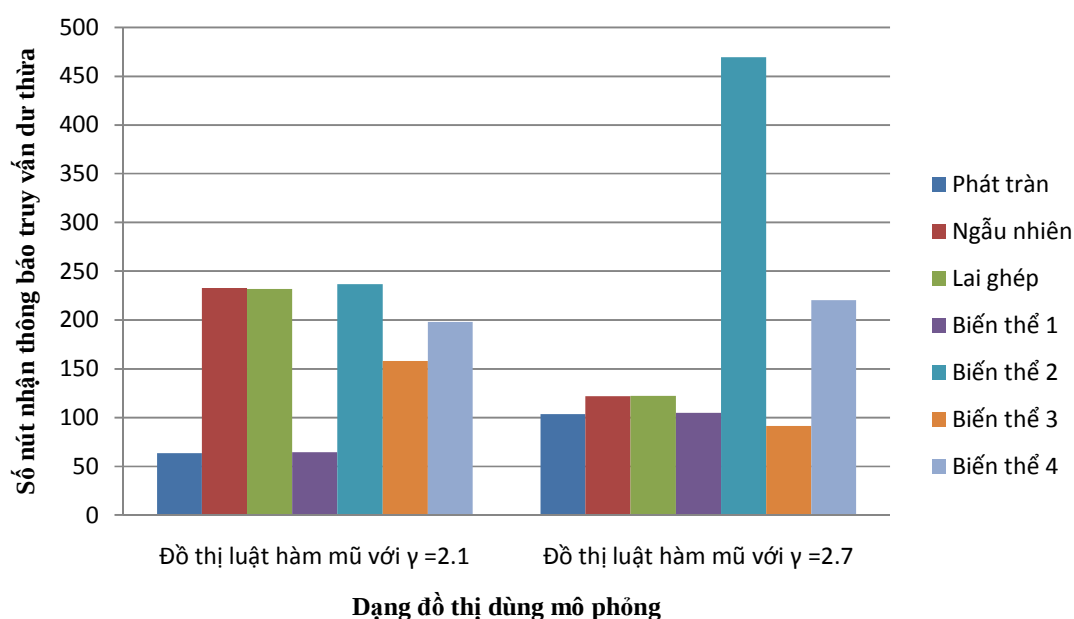
Biểu đồ 2. Kết quả mức độ bao phủ của các phương pháp tìm kiếm trên đồ thị luật hàm mũ với N thông báo.



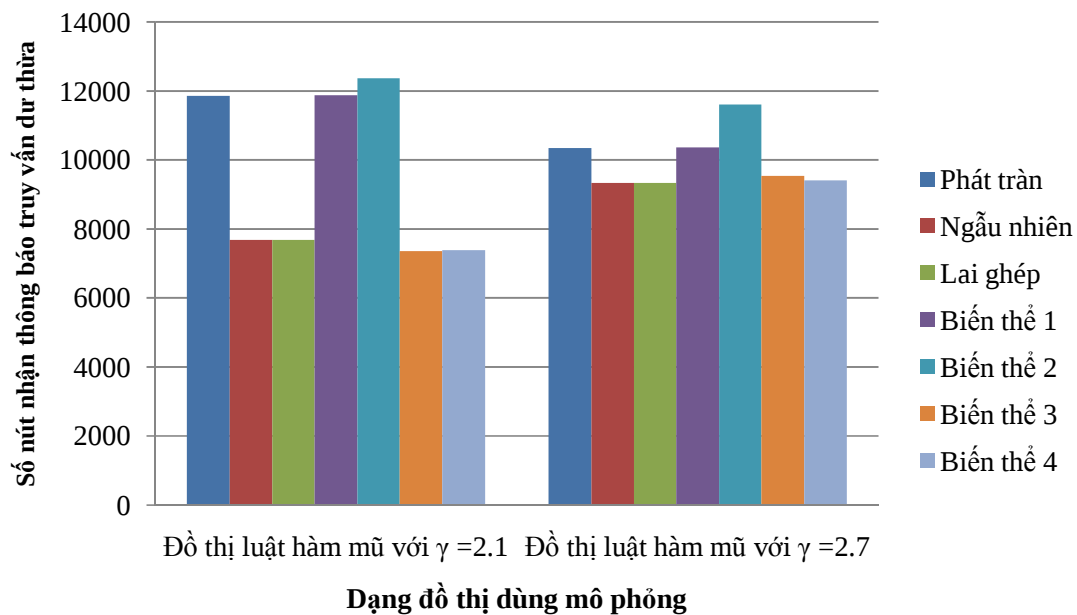
Biểu đồ 3. Kết quả mức độ bao phủ của các phương pháp tìm kiếm trên tô pô phân cụm với 5^5 thông báo.



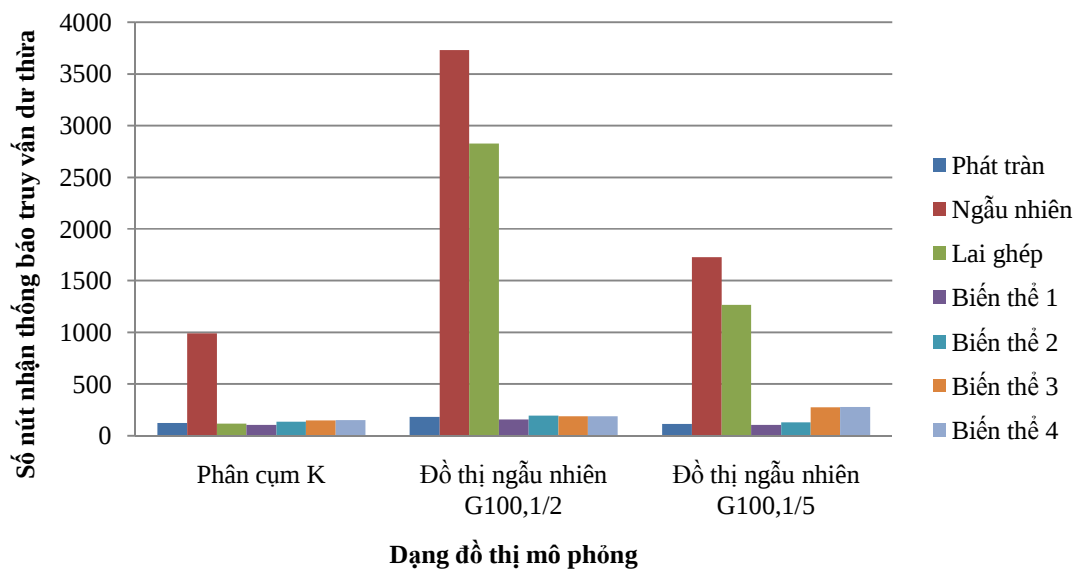
Biểu đồ 4. Kết quả mức độ bao phủ của các phương pháp tìm kiếm trên tô pô phân cụm với N thông báo.



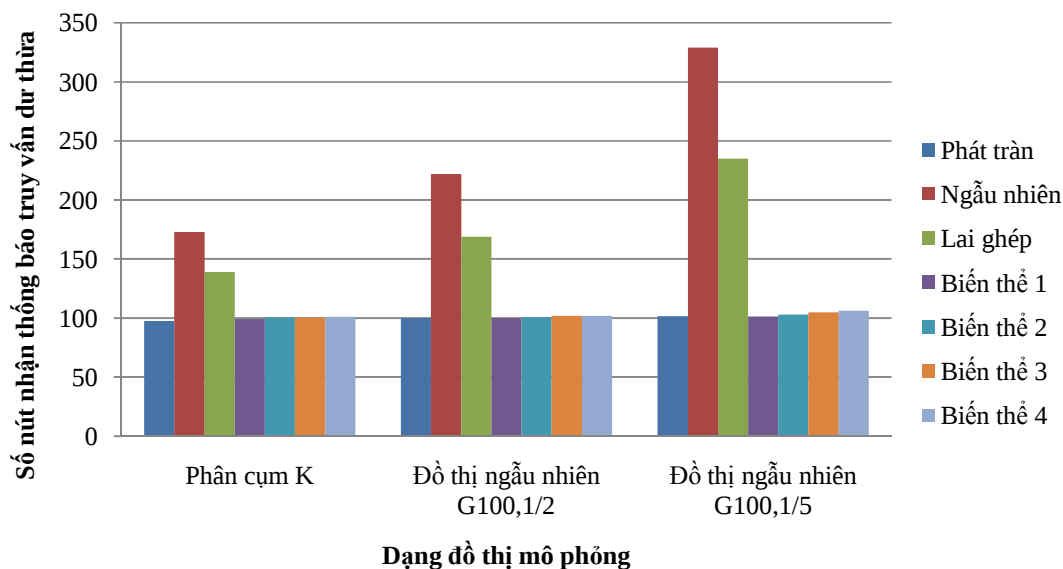
Biểu đồ 5. Kết quả số nút nhận được thông báo truy vấn dư thừa của các phương pháp tìm kiếm trên đồ thị luật hàm mũ với 5^5 thông báo.



Biểu đồ 6. Kết quả số nút nhận được thông báo truy vấn dư thừa của các phương pháp tìm kiếm trên đồ thị luật hàm mũ với N thông báo.



Biểu đồ 7. Kết quả số nút nhận được thông báo truy vấn dư thừa của các phương pháp tìm kiếm trên tô pô phân cụm với 5^5 thông báo.



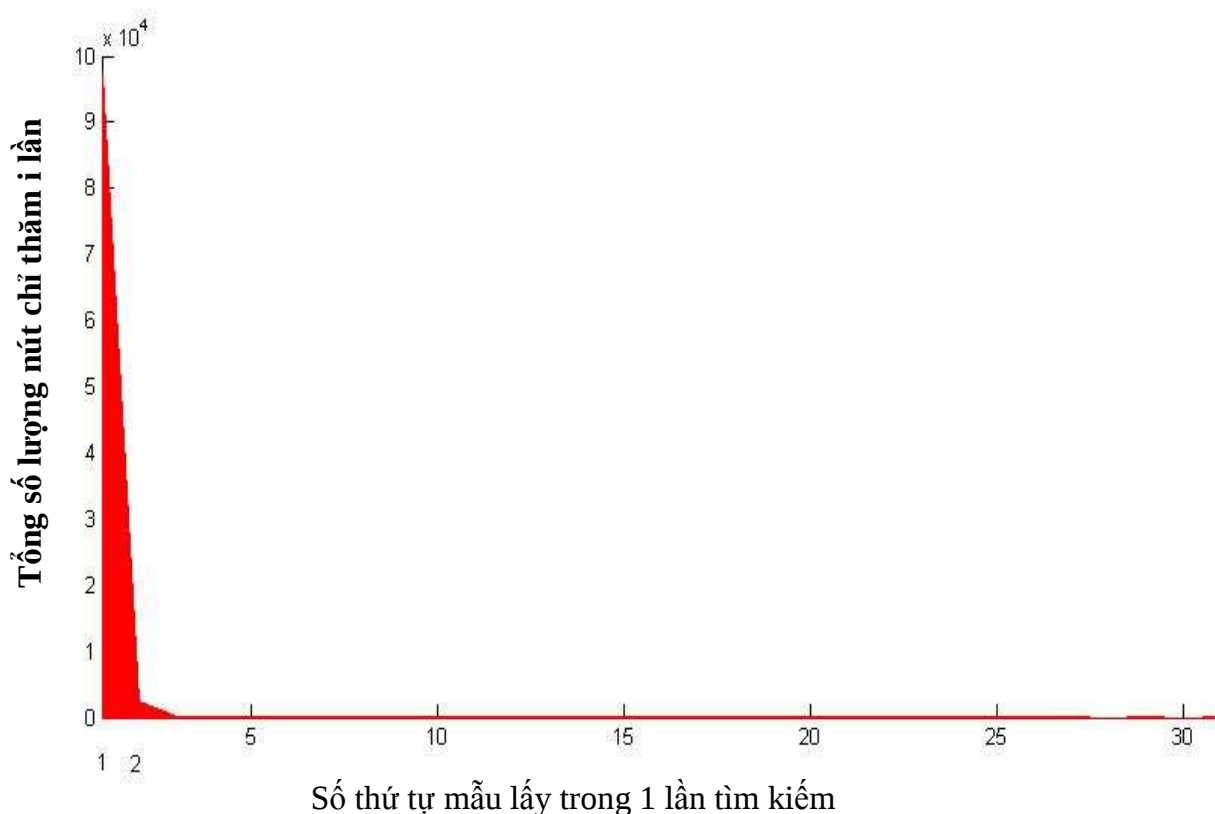
Biểu đồ 8. Kết quả số nút nhận được thông báo truy vấn dư thừa của các phương pháp tìm kiếm trên tô pô phân cụm với N thông báo.

5.4. Đánh giá về phân bố thông báo truy vấn

Ngoài những tiêu chí đã nêu để đánh giá hiệu năng của một phương pháp tìm kiếm thì chúng tôi bổ sung thêm một tiêu chí nữa trong phần nội dung này, đó là tiêu chí về phân bố lượng thông báo truy vấn trên mạng hay còn gọi là phân bố tải. Tiêu chí này được chúng tôi xây dựng sau khi có được kết quả mô phỏng. Biểu diễn tiêu chí này là dạng biểu đồ vùng bởi vì mỗi lần thử nghiệm một phương pháp trên một đồ thị, thì phương pháp đó được thực hiện 15 lần với 15 nút nguồn truy vấn khác nhau và giá trị các nút tìm thấy trả về là giá trị trung bình của 15 lần. Tuy nhiên trong đánh giá chúng tôi chỉ đưa ra biểu diễn mẫu một lần thử nghiệm của một phương pháp trên 1 dạng đồ thị thay vì 60 lần bởi vì vùng biểu diễn chung 1 dạng, và sự thay đổi về giá trị các mẫu không lớn. Kết quả chúng tôi thu được sau khi dữ liệu mô phỏng được xử lý bằng phần mềm MatLab

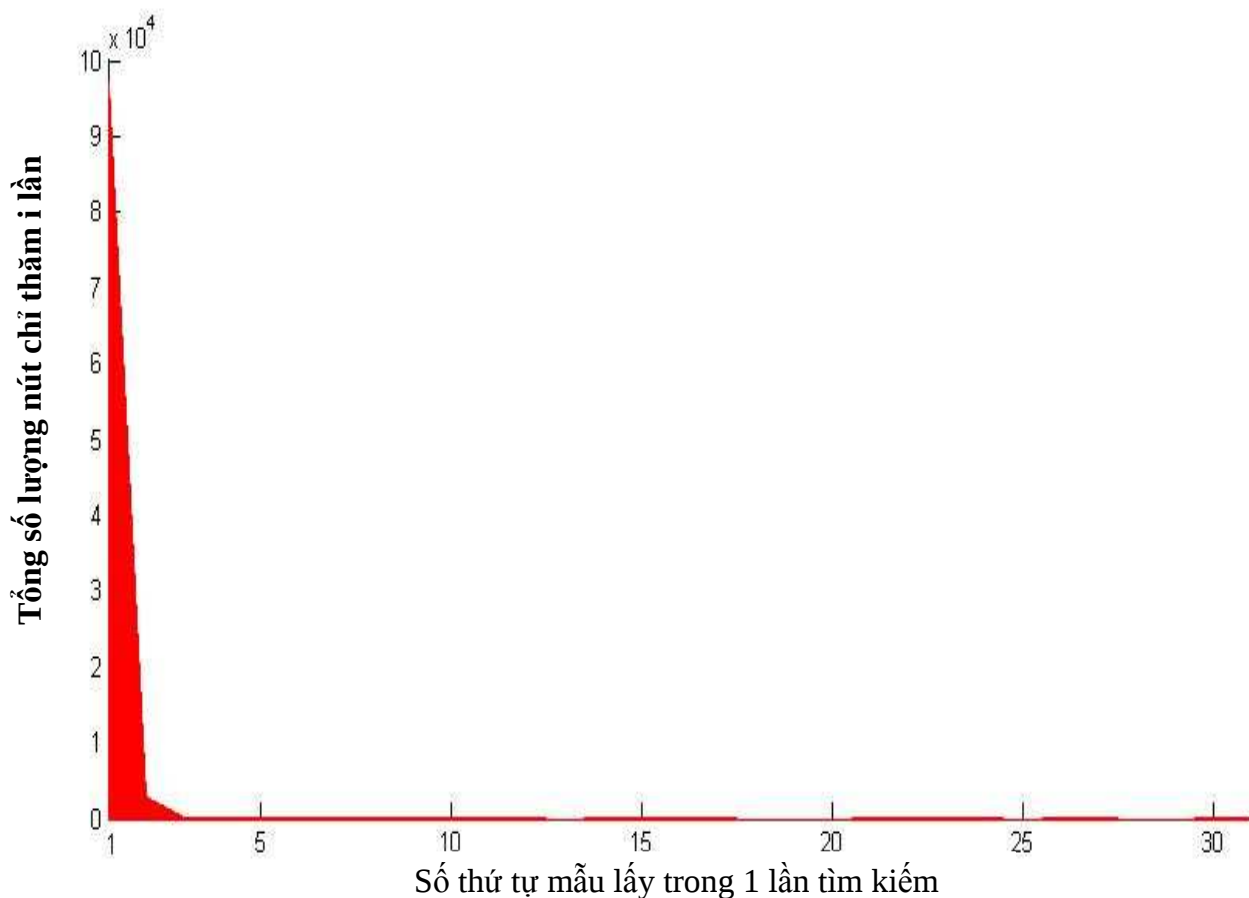
Trong biểu diễn của chúng tôi, trục tung là số lượng nút được thăm i lần, trục hoành là số lượng mẫu của một phương pháp tìm kiếm trên 1 đồ thị. Số lượng mẫu này chúng tôi chọn là 31 mẫu cho tất cả các phương pháp, mặc dù kết quả có nhiều hơn 31 mẫu nhưng chúng tôi chỉ chọn 31 là vì sau 31 mẫu thì số lượng nút được thăm nhiều hơn 31 lần là rất ít. Mẫu đầu tiên là mẫu cho các nút mà không nhận thông báo truy vấn, tiếp

là mẫu thứ hai là mẫu mà các nút chỉ chặn một thông báo truy vấn duy nhất, các mẫu khác thì tăng lên 1 truy vấn tương ứng (có nghĩa là mẫu thứ 3 thì bao gồm các nút chỉ nhận 2 thông báo truy vấn)



Biểu đồ 9. Biểu diễn sự phân bố thông báo truy vấn trên đồ thị luật hàm mũ với $\gamma = 2.1$ và 5^5 truy vấn, phương pháp tìm kiếm di chuyển ngẫu nhiên.

Vùng bao lý tưởng là vùng mà chỉ có 2 mẫu 1 và mẫu 2, trong đó đỉnh của mẫu 2 càng cao, đỉnh của mẫu 1 càng thấp càng tốt, tức cùng là dạng hình thang. Tuy nhiên đó là vùng bao lý tưởng vì trên thực tế có hiện tượng nút chịu tải nhiều và nút không chịu tải do đó vùng bao nào có đáy càng rộng và càng trải dài trên trục hoành thì phương pháp tìm kiếm có hiệu quả càng thấp. Tức là đường bao tiến tới sát trục hoành hơn là sát trục tung.



Biểu đồ 10. Biểu diễn sự phân bố thông báo truy vấn trên đồ thị luật hàm mũ với $\gamma = 2.1$ và 5^5 truy vấn, phương pháp tìm kiếm phát tràn.

Chúng tôi chỉ đưa ra 2 minh họa cho tiêu chí này, các giá trị khác 0 nhưng lại nhỏ hơn nhiều so với giá trị mẫu ban đầu thì biểu diễn bởi các vùng nằm bên trên, sát trùng hoành. Còn các mẫu có giá trị bằng 0 thì biểu diễn bởi 1 điểm nằm trên trục hoành. Mức độ vùng thưa thớt của biểu đồ 10 nhiều hơn biểu đồ 9 và đường bao trong biểu đồ 10 có độ dốc cao hơn so với biểu đồ 9 vì mẫu 2 (mẫu mà các nút chỉ được thăm duy nhất 1 lần) có giá trị cao hơn so với mẫu tương ứng trong biểu đồ 2 nên phương pháp ở biểu đồ 1 cho hiệu quả tốt hơn.

Như vậy mức độ phân bố tải lên các nút trong biểu đồ 9 dày đặc trên các nút mạng và dày hơn trên các nút so với biểu đồ 8, điều đó chứng tỏ có nhiều nút chịu tải quá nhiều, lượng thông báo truy vấn sử dụng không hiệu quả. Đây chỉ là kết quả trên 1 lần thử

nghiệm của 60 lần thử nghiệm, tuy nhiên thì dạng biểu đồ vùng là không khác nhau mấy, các giá trị cũng tương đương nhau.

Mức độ phân bố tải biểu diễn cho các dạng tô pô phân cụm hoặc các thử nghiệm với N thông báo truy vấn sẽ cho đánh giá tốt hơn về việc phân bố thông báo truy vấn đối với các nút. Với một số phương pháp tìm kiếm có thể số lượng nút nhận thông báo truy vấn dư thừa là ít nhưng tần suất nhận thông báo truy vấn trên một nút lại rất cao trong khi số lượng nút nhận thông báo truy vấn dư thừa là nhiều nhưng tần suất trên các nút lại ít trên một số phương pháp khác. Nhưng phương pháp nào có mẫu thử thứ 2 cao và mức độ phân bố đều thì hiệu năng vẫn tốt hơn là phương pháp có mẫu thử thứ 2 thấp, số lượng nhận thông báo truy vấn ít nhưng mức độ phân bố trên mỗi nút này lại cao. Vì ở đây chúng tôi đánh giá bổ sung thêm và chỉ làm với 2 phương pháp, trong tương lai chúng tôi sẽ làm rõ mối liên hệ này.

CHƯƠNG 6. KẾT LUẬN VÀ ĐỊNH HƯỚNG

Trong khóa luận của chúng tôi, chúng tôi đề xuất 4 phương pháp tìm kiếm và đã tiến hành phân tích, mô phỏng, đánh giá hiệu suất tìm kiếm cùng các phương pháp tìm kiếm khác đã được đề xuất trước đó trên các dạng đồ thị của mô hình mạng ngang hàng thuần túy. Các biến thể của chúng tôi là sự kết hợp của phương pháp tìm kiếm: phương pháp di chuyển ngẫu nhiên (bao gồm cả phương pháp thông thường và cải tiến) trước và phương pháp phát tràn sau. Mỗi biến thể có một kỹ thuật khác nhau, với biến thể thứ nhất: đầu tiên sử dụng phương pháp phát tìm kiếm ngẫu nhiên, tìm kiếm k nút, sau đó từ nút cuối cùng tìm được trong k nút này, tiến hành phương pháp phát tràn thông thường. Với biến thể thứ hai: đầu tiên cũng sử dụng phương pháp tìm kiếm ngẫu nhiên, tìm kiếm k nút, sau đó thực hiện phương pháp phát tràn thông thường từ k nút này. Biến thể thứ ba: tương tự như biến thể thứ nhất nhưng sử dụng phương pháp phát tràn cải tiến thay cho phát tràn thông thường. Biến thể thứ tư: tương tự như biến thể thứ hai, nhưng đã thay thế phương pháp phát tràn thông thường bằng phương pháp phát tràn cải tiến.

Sau khi thực hiện mô phỏng, chúng tôi đã đưa ra kết quả và nhận xét về hiệu suất của từng phương pháp trên từng dạng đồ thị. Do đó chúng tôi đi đến kết luận:

- Phương pháp phát tràn là phương pháp tìm kiếm tốt nhất với lượng thông báo nhỏ trên đồ thị luật hàm mũ và các phương pháp tìm kiếm lai ghép biến thể của chúng tôi cũng là phương pháp tốt trong trường hợp $\gamma = 2.1$. Tuy nhiên trong trường hợp $\gamma = 2.7$ thì phương pháp tìm kiếm lai ghép biến thể thứ nhất, thứ ba là tốt và phương pháp tìm kiếm lai ghép biến thể thứ hai là tồi tệ nhất. Với lượng thông báo truy vấn N là lớn thì phương pháp tìm kiếm lai ghép biến thể thứ ba của chúng tôi cho kết quả tốt nhất, phương pháp di chuyển ngẫu nhiên và phương pháp lai ghép trong tài liệu [14] cũng cho kết quả tốt. Tuy nhiên tổng số nút nhận truy vấn dư thừa lại cao với những phương pháp này.
- Đối với tô pô phân cụm thì phương pháp di chuyển ngẫu nhiên cho kết quả tốt nhất cả đối với lượng thông báo truy vấn 5^5 hay N thông báo, sau đó là phương pháp lai ghép trong tài liệu [14], còn đối với các phương pháp còn lại cho kết quả tồi nhưng tổng số nút nhận truy vấn dư thừa lại thấp.

- Đánh giá về mức độ phân bố thông báo truy vấn thì phương pháp di chuyển ngẫu nhiên phân bố tồi hơn so với phương pháp phát tràn trong trường hợp chúng tôi minh họa là với đồ thị luật hàm mũ $\gamma = 2.1$ và sử dụng 5^5 thông báo truy vấn.

Như vậy, những biến thể của chúng tôi cũng là một trong giải pháp tốt cho việc tìm kiếm các tài nguyên trong một mạng ngang hàng thuần túy. Trong tương lai, chúng tôi sẽ tiến hành thử nghiệm trên nhiều dạng đồ thị khác để có đánh giá tốt hơn. Ngoài ra sẽ thử nghiệm kết hợp với trường hợp cải tiến của phương pháp tìm kiếm phát tràn, hoặc phương pháp cải tiến của phương pháp di chuyển ngẫu nhiên là phương pháp k-random walk. Vì các kết quả mà các phương pháp cải tiến này mang lại là tốt, như kết quả trong tài liệu [14] đã đưa ra nên sẽ đánh giá các biến thể của chúng tôi so với phương pháp tìm kiếm phát tràn cải tiến.

Chúng tôi sẽ biểu diễn chi tiết hơn với tiêu chí đánh giá về mức độ phân bố chịu tải với những mẫu có giá trị nhỏ và rất nhỏ và tiến hành trên nhiều phương pháp, nhiều dạng đồ thị để có phân tích, đánh giá tốt hơn. Đồng thời làm rõ mối liên hệ giữa 2 tiêu chí mức độ phân bố chịu tải (phân tán thông báo truy vấn) và số lượng nút nhận thông báo truy vấn dư thừa.

TÀI LIỆU THAM KHẢO

- [1]. Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, Janet Wiener. “Graph structure in the web”, October 6, 2004.
- [2]. Andy Oram. “Peer to Peer: Harnessing the Power of Disruptive Technologies”. OReilly Publishing, first edition March 2001. Page 9, page 19. Chapter 8.
- [3]. Brendon J. Wilson. “JXTA”. New Riders Publishing. First Edition: June, 2002. Chapter 1 and 2.
- [4]. Chonggang Wang, Bo Li. “Peer-to-Peer overlay networks: A survey”, April 20, 2003.
- [5]. Christos Gkantsidis, Milena Mihail, Amin Saberi. “Hybrid search schemes for unstructured Peer-to-Peer networks”, IEEE Infocom 2005.
- [6]. Christos Gkantsidis, Milena Mihail, and Amin Saberi. “Random walks in Peer-to-Peer networks”, IEEE Infocom 2004.
- [7]. Dimitrios Tsoumakos, Nick Roussopoulos. “Analysis and comparison of P2P search methods”, CS-TR-4539, UMIACS-TR-2003-107.
- [8]. Đỗ Đức Giáo. “Hướng dẫn giải bài tập toán rời rạc”. Nhà xuất bản Giáo dục, 2007. Tr 95-97, tr 122-125.
- [9]. Kai-Hsiang YANG, *Member*, Chi-Jen WU, and Jan-Ming HO, *Nonmembers*. “Ant-Search: An ant search algorithm in unstructured Peer-to-Peer networks”. IEICE TRANS. COMMUN., Vol.E89-B, No.9 September 2006.
- [10]. M.E.J. Newman. “Random graphs as models of networks”, 2005.
- [11]. Pawel Pralat and Nicholas Wormald. “Growing protean graphs”, May 15, 2006.
- [12]. Ralf Steinmetz, Klaus Wehrle (Eds). “Peer-to-Peer systems and applications”, LNCS 3485, pp. 1-5, 2005. Springer Publishing-Verlag Berlin Heidelberg 2005. Page 10-12, page 17-24, page 35-79.

- [13]. Ramesh Subramanian, Brian D. Goodman. "Peer-to-Peer Computing: The evolution of a disruptive technology". Idea group publishing, 2005. Chapter II.
- [14]. Reza Dorrigiv, Alejandro López-Ortiz, Pawel Pralat. "Search algorithms for unstructured Peer-to-Peer networks", 2007.
- [15]. Ron Shamir, Roded Sharan, Dekel Tsur. "Cluster graph modification problems", December 2002.
- [16]. Ronaldo A. Ferreira, Murali Krishna Ramanathan, Ananth Grama, Suresh Jagannathan. "Efficient randomized search algorithms in unstructured Peer-to-Peer networks". July 2004.
- [17]. S. Anbu and K.P. Thooyamani. "Improved search efficiency in unstructured Peer to Peer networks using search result path caching". Int. J. Soft Comput., 4 (6): 243-249, 2009.
- [18]. Svante Janson, Tomasz Luczak, Andrzej Rucinski. "Random graphs". Wiley Publishing, New York, 2000.
- [19]. Tomasz Luczak and Pawel Pralat. "Protean graphs". April 15, 2006.
- [20]. Tsungnan Lin, *Senior Member, IEEE*, Pochiang Lin, *Student Member, IEEE*, Hsinping Wang, and Chiahung Chen. "Dynamic search algorithm in unstructured Peer-to-Peer networks". IEEE. Infocom 2009.
- [21]. Tsungnan Lin, Hsinping Wang. "On efficiency in searching networks". IEEE 2005.
- [22]. Tsungnan Lin, Hsinping Wang. "Search performance analysis in Peer-to-Peer networks". IEEE 2003.
- [23]. Vicent Cholvi, Pascal Felber, Ernst Biersack. "Efficient search in unstructured Peer-to-Peer networks", June 27–30, 2004, Barcelona, Spain.
- [24]. William Aiello, Fan Chung, Linyuan Lu. "A random graph model for power law graphs", 2001.

