TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**GIẢI PHÁP GIA CỐ TOPOLOGY MẠNG CÁP QUANG ĐỂ ĐẢM BẢO TÍNH CHỊU LỖI CÓ XEM XÉT NHÓM NGUY CƠ CHUNG**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | : | **Hoàng Văn Phương** |
| Lớp | : | IS2 – K55 |
| Giáo viên hướng dẫn | : | **TS. Trương Thị Diệu Linh** |

HÀ NỘI 06-2015

# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: HOÀNG VĂN PHƯƠNG

Điện thoại liên lạc: 01673393821 Email: phuong1492.bka@gmail.com

Lớp: IS 2 – K55 Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại: Bộ môn Truyền thông mạng, Viện CNTT&TT, Đại học Bách Khoa Hà Nội.

Thời gian làm ĐATN: Từ ngày 21/02/2015 đến 29/05/2015

2. Mục đích nội dung của ĐATN

* Giải pháp gia cố topology mạng cáp quang để đảm bảo tính chịu lỗi có xem xét nguy cơ chung và áp dụng cho mạng Bách Khoa Hà Nội.

3. Các nhiệm vụ cụ thể của ĐATN

* Nghiên cứu bài toán thiết kế mạng chịu lỗi.
* Đề xuất giải thuật.
* Cài đặt giải thuật.
* Tiến hành chạy thực nghiệm và so sánh kết quả.

4. Lời cam đoan của sinh viên:

Tôi – *Hoàng Văn Phương* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS Trương Thị Diệu Linh*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Tác giả ĐATN  *Hoàng Văn Phương* |

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Giáo viên hướng dẫn  *TS Trương Thị Diệu Linh* |

# TÓM TẮT NỘI DUNG ĐỒ ÁN

# LỜI MỞ ĐẦU

Sự ra đời của hệ thống mạng, đặc biệt là Internet, đã đánh dấu một bước ngoặt quan trọng trong lĩnh vực công nghệ thông tin và truyền thông. Kể từ đó, mạng đã mang đến càng ngày càng nhiều lợi ích trong mọi lĩnh vực của cuộc sống như kinh tế, văn hóa, quân đội,… Do đó, đôi khi chỉ một sự cố mạng cũng có thể gây ra một loạt những hậu quả nghiêm trọng, đặc biệt là về kinh tế. Điều này đã đặt ra yêu cầu cho các nhà cung cấp mạng là làm thế nào để đảm bảo tính tin cậy cho sản phẩm của họ. Đó cũng là lý do cho sự ra đời của , bài toán về thiết kế mạng chịu lỗi (Survivable Network Design Problem - SNDP). Sau này, khi hệ thống mạng dần phát triển, SNDP ngày càng nhận được nhiều sự quan tâm của các nhà khoa học, các nhà quy hoạch, quản lý…

Hiện tại trên thế giới đã đưa ra rất nhiều mô hình cho bài toán thiết kế mạng chịu lỗi. Tùy thuộc vào thực tế, mỗi mô hình lại đại diện cho ứng dụng vào từng loại mạng. Đồ án này đã nghiên cứu lý thuyết tổng quan về bài toán thiết kế mạng chịu lỗi, đề xuất, cài đặt giải thuật, đánh giá kết quả chạy giải thuật. Qua đó đưa ra hướng phát triển tiếp theo của bài toán thiết kế mạng chịu lỗi.

# LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành tới cô Trương Thị Diệu Linh, người đã tận tình dạy dỗ và hướng dẫn em trong quá trình hoàn thành đồ án cũng như trong học tập.

Đồng thời, em xin bày tỏ lòng biết ơn đến các thầy cô giáo trong Viện Công nghệ Thông tin và Truyền thông – Đại học Bách Khoa Hà Nội, những người đã tận tình giảng dạy, truyền đạt cho chúng em những kiến thức cơ bản làm nền tảng cho việc thực hiện đồ án cũng như trong quá trình công tác sau này.

Em cũng xin gửi lời cảm ơn tới các anh chị tại Đại học Bách Khoa Hà Nội, các bạn, các bạn cùng lớp, những người luôn ở bên cạnh giúp đỡ, động viên em trong quá trình hoàn thành đồ án.

Cuối cùng, với tất cả sự kính trọng con xin bày tỏ lòng biết ơn sâu sắc tới bố mẹ và anh chị em trong gia đình đã luôn là chỗ dựa tinh thần vững chắc và tạo mọi điều kiện cho con ăn học nên người.

Hà Nội, ngày 19 tháng 05 năm 2015

# MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP i](#_Toc388905765)

[TÓM TẮT NỘI DUNG ĐỒ ÁN ii](#_Toc388905766)

[LỜI MỞ ĐẦU iii](#_Toc388905767)

[LỜI CẢM ƠN iv](#_Toc388905768)

[MỤC LỤC v](#_Toc388905769)

[DANH MỤC HÌNH VẼ viii](#_Toc388905770)

[DANH MỤC CÁC BẢNG x](#_Toc388905771)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ xi](#_Toc388905772)

[PHẦN 1: ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI QUYẾT](#_Toc388905773)

[CHƯƠNG 1.GIỚI THIỆU ĐỀ TÀI VÀ ĐỊNH HƯỚNG GIẢI QUYẾT](#_Toc388905774)

[CHƯƠNG 2.CƠ SỞ LÝ THUYẾT](#_Toc388905774)

[1.Lý thuyết về đồ thị](#_Toc388905774)

[1.1. Định nghĩa đồ thị](#_Toc388905775)

[1.2. Đường đi và tính liên thông](#_Toc388905776)

[2. Các giải thuật tìm đường đi ngắn nhất](#_Toc388905777)

[2.1. Giải thuật Dijkstra](#_Toc388905778)

[2.2. Giải thuật Bellman–Ford](#_Toc388905779)

[2.3. Giải thuật Floyd-Warshall](#_Toc388905780)

[3. Một số khái niệm về mạng cáp quang](#_Toc388905777)

[3.1. Mạng cáp quang](#_Toc388905775)

[3.2. Định tuyến và gán bước sóng](#_Toc388905775)

[3.3. Bài toán mạng chịu lỗi](#_Toc388905775)

[3.4. Nhóm nguy cơ chung](#_Toc388905775)

[PHẦN 2: KẾT QUẢ ĐẠT ĐƯỢC](#_Toc388905773)

[CHƯƠNG 3: ĐỀ XUẤT GIẢI THUẬT CHO BÀI TOÁN GIA CỐ TOPOLOGY MẠNG CHỊU LỖI CÓ XEM XÉT NGUY CƠ CHUNG](#_Toc388905804)

[1. Đề xuất giải thuật](#_Toc388905805)

[2. Cài đặt giải thuật](#_Toc388905813)

[CHƯƠNG 4: KẾT QUẢ THỬ NGHIỆM](#_Toc388905814)

[1. Dữ liệu thực nghiệm](#_Toc388905815)

[2. Thông số thực nghiệm](#_Toc388905816)

[3. Kết quả thực nghiệm](#_Toc388905817)

[KẾT LUẬN](#_Toc388905821)

[TÀI LIỆU THAM KHẢO](#_Toc388905822)

PHẦN I. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI QUYẾT

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI VÀ ĐỊNH HƯỚNG GIẢI QUYẾT

Nhờ có công nghệ Wavelength Division Multiplexing(WDM), một sợi quang có thể mang nhiều bước sóng khác nhau. Vì vậy mạng quang WDM có thể cung cấp băng thông rất lớn, và ngày càng được áp dụng rộng rãi. Mạng cáp quang được phát triển qua thế hệ thứ nhất được đặc trưng bởi tín hiệu quang truyền qua các liên kết và xử lí tín hiệu điện tại các nút mạng. Mạng SONET/SDH là một kiến trúc tiêu biểu của thế hệ này. Mạng quang ngày nay là mạng quang thế hệ thứ hai trong đó tín hiệu quang cả trên các liên kết và tại các nút mạng. Vì vậy mạng thế hệ thứ hai được gọi là mạng toàn quang. Mạng toàn quang loại bỏ quá trình xử lí tín hiệu điện và các thiết bị chuyển đổi quang điện khỏi mạng bởi vậy chi phí thiết bị giảm đi rất nhiều.Tuy nhiên, chi phí triển khai một mạng toàn quang lớn hơn rất nhiều so với mạng cáp đồng.

Hầu hết các topology mạng cáp quang hiện nay được triển khai chưa có khả năng chịu lỗi. Nói cách khác dữ liệu truyền tải trên các thiết kế trên sẽ bị mất nếu như một trong các kết nối trong mạng có sự cố. Chịu lỗi là khái niệm chỉ khả năng mà mạng có thể duy trì các dịch vụ khi có lỗi xảy ra. Mạng gặp vấn đề có thể là do đứt dây cáp hoặc các thiết bị kết nối có vấn đề, khi đó những yêu cầu sử dụng những kết nối này sẽ bị ảnh hưởng. Với mạng có tính chịu lổi, khi vấn đề trên xảy ra mạng sẽ chuyển những yêu cầu bị ảnh hưởng sang một kết nối khác. Vì vậy bài toán gia cố topology mạng cáp quang chịu lỗi rất cần thiết. Bài toán sẽ giữ lại các node, các liên kết cũ của topology mạng và sẽ thêm các liên kết mới để đảm bảo tính chịu lỗi của topology mạng. Đến nay có rất nhiều nghiên cứu về mạng chịu lỗi tuy nhiên các nghiên cứu này không cân nhắc đến khả năng trên thực tế là đường làm việc và đường dự phòng có một số kết nối thuộc nhóm nguy cơ chung(SRG), do các liên kết này được bó chung một bó cáp. Và khi xảy ra lỗi trên những nhóm nguy cơ chung này thì mạng cũng sẽ không thể đảm bảo khả năng chịu lỗi.

Bài toán gia cố topology mạng cáp quang chịu lỗi có xem xét các nhóm nguy cơ chung được phát biểu như sau:

**Đầu vào:**

* Cho trước các node mạng với tọa độ của chúng.
* Cho trước ma trận yêu cầu giữa các cặp nút trong mạng.
* Cho trước các nhóm nguy cơ chung.
* Cho trước các đường link trong mạng.

**Mục tiêu:**

* Thêm các đường link mới sao cho mạng có tính chịu lỗi.
* Chi phí đường đi cáp là nhỏ nhất.

Trong thực tế, trên mỗi đường truyền của mạng sẽ có một giới hạn về băng thông. Do đó bài toán thiết kế mạng quang còn cần phải quan tâm tới vấn đề định tuyến cho các đường kết nối sao cho lưu lượng truyền tải không vượt quá giới hạn của đường truyền. Tuy nhiên, mục tiêu của bài toán là gia cố topology mạng, do đó, ta có thể coi như khả năng thông qua của các đường truyền là vô hạn. Vì vậy, trong mô hình bài toán đưa ra chỉ tập trung xác định chi phí cho việc xây dựng mạng, các đường định tuyến cho các yêu cầu. Xét trên một cạnh đã được sử dụng trong một yêu cầu kết nối, khi có thêm một yêu cầu kết nối sử dụng tới cạnh này, thì chi phí để xây dựng mạng sẽ được tính bằng 0.

Trong đồ án này, tác giả đề xuất một giải pháp gia cố topology vật lý cho mạng quang với mục tiêu đáp ứng ma trận yêu cầu với chi phí tối thiểu về đường đi cáp và có khả năng sống sót khi mạng gặp lỗi. Giải thuật đề xuất sẽ được áp dụng vào topology mạng của trường Đại học Bách Khoa Hà Nội.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

1. **Lý thuyết về đồ thị**

Đồ thị biểu diễn được rất nhiều cấu trúc. Nhiều bài toán thực tế có thể được biểu diễn bằng đồ thị. Trong toán học, lý thuyết đồ thị là một lĩnh vực đã xuất hiện từ lâu và có rất nhiều ứng dụng trong thực tế. Vào những năm đầu của thế kỷ 18, những tư tưởng cơ bản của lý thuyết đồ thị được đề xuất bởi nhà toán học lỗi lạc người Thụy Sỹ Leonhard Euler. Đặc biệt trong khoảng vài chục năm trở lại đây, với sự ra đời máy tính điện tử và sự phát triển nhanh chóng của công nghệ thông tin, lý thuyết đồ thị ngày càng được quan tâm nhiều hơn.

Sau đây, xin được trình bày qua các khái niệm cơ bản của đồ thị làm nền tảng cho việc áp dụng các giải thuật trong các lĩnh vực ứng dụng đồ thị.

* 1. **Định nghĩa đồ thị**

Đồ thị là một cấu trúc rời rạc gồm các đỉnh và các cạnh nối các đỉnh này. Chúng ta phân biệt các loại đồ thị khác nhau bởi kiểu và số lượng cạnh nối các đỉnh này. Ta định nghĩa các loại đồ thị như sau.

**Định nghĩa 1.1** *Đơn đồ thị vô hướng bao gồm là tập không rỗng chứa các đỉnh, và là tập các cặp không có thứ tự gồm hai phần tử khác nhau của gọi là cạnh.*[Hình 1.a]

**Định nghĩa 1.2** *Đơn đồ thị có hướng bao gồm tập đỉnh và tập cạnh là các cặp có thứ tự gồm hai phần tử khác nhau của . Các cạnh của đồ thị có hướng còn được gọi là các cung.*[Hình 1.b]

(a) (b)

Hình 1: Đồ thị vô hướng và đồ thị có hướng

Để tiện sử dụng, chúng ta sẽ gọi đồ thị thay cho đơn đồ thị vô hướng mà không có chú thích gì thêm.

* 1. **Đường đi và tính liên thông**

**Định nghĩa 1.3** *Đường đi từ đỉnh đến đỉnh trên đồ thị là dãy , trong đó là số nguyên dương, , , , . Đỉnh được gọi là đỉnh đầu, đỉnh được gọi là đỉnh cuối. Đường đi mà đỉnh đầu trùng với đỉnh cuối được gọi là chu trình. Chu trình được gọi là đơn nếu không có đỉnh nào lặp lại, ngoại trừ đỉnh đầu trùng với đỉnh cuối.*

Để xác định xem có tồn tại đường đi giữa 2 cặp đỉnh trong đồ thị, chúng ta đưa ra khái niệm tính liên thông của đồ thị.

**Định nghĩa 1.4** *Đồ thị vô hướng được gọi là liên thông nếu luôn tìm được đường đi giữa hai đỉnh bất kỳ của nó.*

**Định nghĩa 1.5** *Đồ thị có hướng được gọi là liên thông mạnh nếu luôn tìm được đường đi giữa hai đỉnh bất kỳ của nó. Đồ thị có hướng được gọi là liên thông yếu nếu đồ thị vô hướng tương ứng với nó là liên thông.*

1. **Các giải thuật tìm đường đi ngắn nhất**

Trong lý thuyết đồ thị, bài toán đường đi ngắn nhất nguồn đơn là bài toán tìm một đường đi giữa hai đỉnh sao cho tổng các trọng số của các cạnh tạo nên đường đi đó là nhỏ nhất. Định nghĩa một cách hình thức, cho trước một đồ thị có trọng số (nghĩa là một tập đỉnh *V*, một tập cạnh *E*, và một hàm trong số có giá trị thực *f* : *E* → **R**), cho trước một đỉnh *v* thuộc *V*, tìm một đường đi *P* từ *v* tới mỗi đỉnh *v'* thuộc *V* sao cho \sum_{p\in P} f(p) là nhỏ nhất trong tất cả các đường nối từ *v* tới *v'* . Bài toán đường đi ngắn nhất giữa mọi cặp đỉnh là một bài toán tương tự, trong đó ta phải tìm các đường đi ngắn nhất cho mọi cặp đỉnh *v* và *v'* .

* 1. **Giải thuật Dijsktra**

Thuật toán Dijkstra cho phép tìm đường đi ngắn nhất từ một đỉnh đến các đỉnh còn lại của đồ thị có trọng số không âm.

Thuật toán được xây dựng dựa trên cơ sở gán cho mỗi đỉnh các nhãn tạm thời. Nhãn tạm thời của các đỉnh cho biết cận trên của chiều dài đường đi ngắn nhất từ s đến đỉnh đó. Nhãn của các đỉnh sẽ biến đổi trong các bước lặp, mà ở mỗi bước lặp sẽ có một nhãn tạm thời trở thành chính thức. Nếu nhãn của một đỉnh nào đó trở thành chính thức thì đó cũng chính là chiều dài ngắn nhất của đường đi từ đến đỉnh đó.

Đầu vào của giải thuật Dijkstra là đồ thị có đỉnh, , , là ma trận trọng số, .

Đầu ra của giải thuật Dijkstra: với mỗi :

* : độ dài đường đi ngắn nhất từ đến .
* : đỉnh đi trước trong đường đi ngắn nhất từ đến .

Dưới đây là mã giả của giải thuật Dijkstra.

|  |
| --- |
| 1 **procedure** Dijkstra;  2 **begin**  3 **for** vV **do**  4 **begin**  5 d[v] := w[s,v]; p[v]:=s;  6 **end**;  7 d[s] := 0;  8 T := V \ {s}; //T: tập các đỉnh có nhãn tạm thời  9 **while** T **do**  10 **begin**  11 Tìm đỉnh uT sao cho d[u]=min{d[z]:zT};  12 T := T \ {u};  13 **for** vT **do**  14 **if** d[v] > d[u] + w[u,v] **then**  15 **begin**  16 d[v] := d[u] + w[u,v] ; p[v] := u ;  17 **end**;  18 **end**;  19 **end**; |

Thuật toán có 2 vòng lặp lồng nhau, thời gian tính của các câu lệnh bên trong vòng lặp trong cùng chỉ là nên thuật toán có độ phức tạp là hay .

Như vậy, ở chương này đã trình bày cơ sở lý thuyết cơ bản về đồ thị và bài toán đường đi ngắn nhất. Chương 2 tiếp theo sẽ trình bày về bài toán thiết kế mạng chịu lỗi cũng như các ứng dụng của nó trong thực tế.

* 1. **Giải thuật Bellman–Ford**

Thuật toán Bellman–Ford cho phép tìm đường đi ngắn nhất từ một đỉnh đến các đỉnh còn lại trong đồ thị có trọng số.

Đầu vào của thuật toán Bellman-Ford là đồ thị với đỉnh xác định bởi ma trận trọng số , , đỉnh nguồn .

Đầu ra của thuật toán Bellman-Ford: với mỗi :

* : độ dài đường đi ngắn nhất từ đến .
* : đỉnh đi trước trong đường đi ngắn nhất từ đến .

Dưới đây là mã giả của thuật toán Bellman Ford.

|  |
| --- |
| 1 **procedure** Bellman\_ Ford;  2 **begin**  3 **for** vV **do** (\* Khởi tạo \*)  4 **begin**  5 d[v] := w[s,v]; p[v]:=s;  6 **end**;  7 d[s]:=0; p[s]:=s;  8 **for** k:=1 **to** n-1 **do**  9 **for** vV\{s} **do**  10 **for** uV **do**  11 **if** d[v]>d[u]+w[u,v] **then**  12 **begin**  13 d[v] := d[u] + w[u,v];  14 p[v] := u;  15 **end**;  16 **end**; |

Thuật toán có độ phức tạp hay vì thuật toán có 3 vòng lặp lồng nhau, thời gian tính của các câu lệnh bên trong vòng lặp trong cùng chỉ là .

* 1. **Giải thuật Floyd-Warshall**

Thuật toán Floyd–Warshall tìm đường đi ngắn nhất giữa mọi cặp đỉnh của đồ thị.Thuật toán này được thiết kế dựa trên tư tưởng quy hoạch động. Nguyên lý tối ưu được áp dụng cho thuật toán này: “Nếu k là đỉnh nằm trên đường đi ngắn nhất từ i đến j thì đoạn đường từ i đến k và từ k đến j cũng là ngắn nhất”.

Dưới đây là mã giả của thuật toán Floyd-Warshall.

|  |
| --- |
| 1 **procedure** FloydWarshall;  2 **begin**  3 int path[][]; /\* path[i][j]=độ dài đường đi ngắn nhất từ i đến j \*/  4 **for** k := 1 **to** n **do** /\* n là số đỉnh của đồ thị \*/  5 **for** i := 1 **to** n **do**  6 **for** j := 1 **to** n **do**  7 path[i][j]=min(path[i][j],path[i][k]+path[k][j]);  8 **end**; |

Thời gian tính của thuật toán Floyd–Warshall được xác định bởi ba vòng lặp for lồng nhau. Vì mỗi lần thực hiện dòng số 7cần thời gian là , nên thuật toán có độ phức tạp tính toán là hay .

1. **Một số khái niệm trong mạng cáp quang**
   1. **Mạng cáp quang**

Mạng cáp quang là mạng sử dụng môi trường ánh sáng để truyền tải thông tin. Việc sử dụng ánh sáng để truyền tải thông tin tạo ra khả năng cung cấp băng thông lớn, với tỷ lệ lỗi thấp và có thể truyền ở khoảng cách lớn. Thông tin trong mạng cáp quang được truyền tải trên các sợi quang (cáp quang).

Ngoài ra, môi trường quang còn cung cấp khả năng truyền tải dữ liệu trong suốt. Tính trong suốt cho phép các dạng dữ liệu khác nhau chia sẻ cùng một môi trường truyền và điều này đáp ứng nhu truyền tải các loại dữ liệu đa dạng như hiện nay: văn bản, âm thanh, hình ảnh….

Chính vì những ưu điểm đó, mạng cáp quang đã và đang được xem như là một công nghệ chính cho việc xây dựng các mạng lõi thế hệ mới.

* 1. **Định tuyến và gán bước sóng**
     1. **Định tuyến**

Định tuyến được coi là thành phần cốt yếu của kiến trúc mạng, thiết kế mạng và điều hành mạng của mọi mạng thông tin, là thành phần không thể thiếu trong mạng viễn thông. Các yếu tố thúc đẩy cho quá trình thay đổi và phát triển định tuyến chủ yếu do nhu cầu cải thiện hiệu năng mạng, các dịch vụ mới đưa vào khai thác và sự thay đổi công nghệ mạng và đây cũng là một trong những thách thức khi xây dựng và khai thác. Hầu hết các mạng viễn thông truyền thống được xậy dựng theo mô hình mạng phân cấp mô hình này cho phép sử dụng định tuyến tĩnh trên quy mô lớn.

Trong khi định tuyến tĩnh vẫn còn tồn tại tính chất độc lập giữa người sử dụng và mạng vẫn ở mức cao, định tuyến tĩnh chủ yếu dựa trên mong muốn của người sử dụng nhiều hơn là tình trạng hiện thời. Mạng hiện đại hiện nay có xu hướng hội tụ các dịch vụ mạng, yêu cầu đặt ra từ phía người sử dụng là rất đa dạng và phức tạp. Các phương pháp định tuyến động được sử dụng nhằm nâng cao hiệu năng mạng của mạng mới này, tăng thêm tính chủ động, mềm dẻo đáp ứng nhằm nâng cao hiệu năng mạng của mạng mới này, tăng thêm tính chủ động, mềm dẻo đáp ứng tốt hơn yêu cầu người sử dụng dịch vụ.

Định tuyến để chỉ sự lựa chọn đường đi trên một kết nối mạng để thực hiện việc gửi dữ liệu. Định tuyến chỉ ra hướng, sự dịch chuyển của các gói dữ liệu được đánh địa chỉ từ mạng nguồn đến đích thông qua các node trung gian, thiết bị chuyên dùng là bộ định tuyến. Tiến trình định tuyến thường chỉ hướng đi dựa vào bảng định tuyến, đó là bảng cho các lộ trình tốt nhất đến các đích khác nhau trên mạng. Vì vậy việc xây dựng bảng định tuyến được tổ chức trong bộ nhớ của router trở nên vô cùng quan trọng cho việc định tuyến hiệu quả.

Phân loại định tuyến:

+ Định tuyến tĩnh: với định tuyến tĩnh, các đường dẫn được chọn trước cho mỗi cặp nguồn – đích của node trong mạng. Các giải thuật định tuyến chi phí tối thiểu có thể được sử dụng. Kế hoạch định tuyến tĩnh được sử dụng hầu hết các mạng truyền thống, trong kế hoạch định tuyến này chủ yếu với mục đích làm giảm các hệ thống chuyển mạch phải đi qua với yêu cầu kết nối đường dài. Kĩ thuật định tuyến tĩnh bộc lộ một số nhược điểm như: quyết định định tuyến tĩnh không dựa trên sự đánh giá lưu lượng và topo mạng hiện thời. Các bộ định tuyến không phát hiện ra các bộ định tuyến mới, chúng chỉ có thể chuyển thông tin đến tới cá bộ định tuyến được chỉ trước của nhà quản lí mạng.

+ Định tuyến động: Định tuyến động lựa chọn tuyến dựa trên thông tin trạng thái hiện thời mạng. Thông tin trạng thái có thể đo hoặc dự đoán và tuyến đường có thể thay đổi khi topo mạng thay đổi khi topo mạng thay đổi hoặc lưu lượng mạng thay đổi. Định tuyến động thể hiện tính linh hoạt và dễ dàng mở rộng mạng.

* + 1. **Gán bước sóng**

Việc gán bước sóng là nhân tố chính ảnh hưởng đến xác suất tắc nghẽn và tính thực thi của mạng. Gán bước sóng thích hợp có thể làm giảm số bước sóng sử dụng hoặc không cần dùng đến bộ chuyển đổi bước sóng, nên ta có thể giảm được chi phí của mạng xuống rất nhiều. Gán bước sóng được chia làm hai loại cho lưu lượng mạng cố định và lưu lượng mạng thay đổi. Khi lưu lượng mạng cố định thì phép gán cố định, cùng một bước sóng được gán nếu(nếu có sẵn) cho mọi yêu cầu được tạo ra ở một nút, nếu không thì yêu cầu bị chặn. Khi lưu lượng mạng thay đổi, lúc có yêu cầu đến một nút mạng nào đó thì nút đó sẽ dùng một giải thuật để chọn một bước sóng riêng biệt còn rỗi ở nút đó và gán cho lightpath đó để định tuyến nó, nếu không thì yêu cầu không được giải quyết. Giải thuật cho phương pháp gán quản lí một danh sách các bước sóng được sử dụng, các bước sóng còn rỗi ở mỗi nút.

Các phương pháp gán bước sóng được chia làm các loại như sau:

* Kiểu gán Random: khi có yêu cầu đến một nút, nút đó sẽ xác định những bước sóng còn hiệu lực ( tức là còn rỗi) và chọn ngẫu nhiên một λ trong những bước sóng đó để gán cho yêu cầu đó. Các bước sóng còn rỗi ở mỗi nút được xác định bằng cách loại bỏ bước sóng λ đã sử dụng ra khỏi danh sách bước sóng còn rỗi; khi cuộc gọi kết thúc, λ được loại ra khỏi danh sách bước sóng bị bận và được thêm vào trở lại danh sách bước sóng rỗi ban đầu. Phương pháp này không cần đòi hỏi những thông tin về toàn bộ trạng thái của mạng khi thực hiện gán bước sóng. Phép gán này phân phối lưu lượng một cách tuỳ ý, do vậy sự tận dụng bước sóng được cân bằng và tranh chấp bước sóng thấp nên xác suất tắc nghẽn cũng thấp hơn.
* Kiểu gán First - Fit: phép gán này sẽ tìm và gán những bước sóng theo một trình tự cố định. Tất cả các bước sóng được đánh số từ thấp đến cao và các bước sóng được chọn để gán cũng theo chỉ số từ thấp đến cao, tức là bước sóng đầu tiên được chọn là bước sóng có chỉ số nhỏ nhất trong số bước sóng rỗi và gán cho yêu cầu. Cũng tương tự như phương pháp gán Random, phép gán này không cần bất kì thông tin nào về thông tin trạng thái mạng. Hạn chế của phương pháp này là các bước sóng có chỉ số nhỏ hơn được dùng nhiều, trong khi những bước sóng có chỉ số lớn hầu như không được sử dụng. Hơn nữa sự gia tăng số bước sóng trong sợi cũng không mang lại hiệu quả nào bởi vì những bước sóng có chỉ số cao rất ít khi được dùng. Do đó sự tranh chấp đối với những bước sóng có chỉ số nhỏ tăng lên, làm xác suất tắc nghẽn cũng tăng lên. Phép gán này cho chi phí thấp hơn so với phép gán Random bởi vì nó không cần phải kiểm tra tất cả các bước sóng trong mỗi tuyến, vì thế nó được ưa chuộng hơn.
* Phép gán Most - used: nó là phép gán chỉ là ngược với phép gán Least-used, nó tìm chọn những bước sóng được sử dụng nhiều nhất trong mạng. Phép gán này phải đòi hỏi những thông tin về trạng thái mạng để tìm ra bước sóng được sử dụng nhiều nhất. Nó cũng tốn nhưng chi phí tương tự như trong phép gán Least-used, tuy nhiên nó thực hiện tốt hơn so với phép so với phép gán Least-used.
* Phép gán Least - used: Phép gán này chọn những bước sóng mà những bước sóng này ít được sử dụng nhất trong mạng. Mục đích của phép gán này là cân bằng tải trên tất cả những bước sóng. Phép gán này đòi hỏi thông tin trạng thái về mạng để tìm ra bước sóng ít được sử dụng nhất. Tuy nhiên phương pháp này phải tốn kém cho chi phí lưu trữ và tính toán

Với các phép gán bước sóng kể trên, phương pháp Random và First - Fit là thực tế hơn vì dễ thực hiện. Không giống như hai phương pháp Least- used và Most- used đòi hỏi phải có các thông tin về mạng. Nó đơn giản chỉ dựa vào trạng thái nút lúc đó và chọn một bước sóng từ những bước sóng rỗi ở kết nối ngõ ra đó. Một cách tương đối, phương pháp ngẫu nhiên Random cho hiệu quả tốt hơn phương pháp First - Fit.

* 1. **Bài toán mạng chịu lỗi**
     1. **Định nghĩa**

Mạng chịu lỗi là khái niệm chỉ khả năng mà mạng có thể duy trì các dịch vụ khi có lỗi xảy ra. Mạng gặp vấn đề có thể là do đứt dây cáp hoắc các thiết bị kết nối có vấn đề, khi đó những yêu cầu sử dụng những kết nối này sẽ bị ảnh hưởng. Với mạng có tính chịu lỗi, khi vấn đề trên xảy ra mạng sẽ chuyển những yêu cầu bị ảnh hưởng sang một đường kết nối khác.Vấn đề thiết kế mạng chịu lỗi nhằm đưa ra được một thiết kế trên mạng vật lý có thể đáp ứng được tất cả các yêu cầu kết nối trên tầng logic, với chi phí thiết kế tối thiểu đồng thời thỏa mãn tính chịu lỗi.

* + 1. **Phát biểu bài toán**

**Đầu vào:**

* Cho đồ thị G1 = (V1, E1) biểu diễn mạng vật lý, là đơn đồ thị vô hướng, đầy đủ có trọng số. Trọng số cạnh là chi phí thiết lập cạnh vật lý.
* Cho đồ thị G2 = (V2,E2) biểu diễn mạng logic, là đơn đồ thị vô hướng với tập đỉnh V2 V1
* Trên G2, cho tập yêu cầu kết nối T.

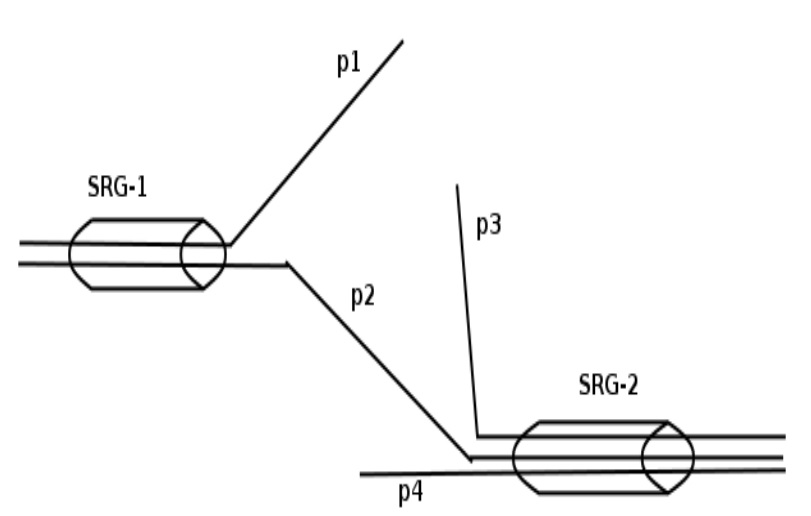
**Đầu ra**:

* Với mỗi yêu cầu kết nối cần tìm hai đường truyền vật lý phân biệt (không chung đỉnh, trừ đỉnh oi và di) tương ứng với hai lightpath.
  + Đường truyền vật lý tương ứng với lightpath là đường truyền chính.
  + Đường truyền vật lý tương ứng với lightpath là đường truyền dự phòng.
* Tổng chi phí xây dựng các đường truyền vật lý là nhỏ nhất:

Trong đó E’ là tập cạnh trên các đường truyền vật lý, c(e) là trọng số của cạnh e

* 1. **Nhóm nguy cơ chung**

Nhóm nguy cơ chung (Share Risk Group - SRG) là nhóm gồm các liên kết cùng chia sẻ nguy cơ chung bị mất đi liên kết, tức là khi một trong những liên kết trong nhóm nguy cơ này có vấn đề thì những liên kết khác thuộc nhóm nguy cơ này cũng chia sẻ chung vấn đề như vậy. Như vật khi đường làm việc và đường backup cùng thuộc một nhóm nguy cơ chung thì mạng sẽ không đảm bảo khả năng chịu lỗi nữa.



Hình 2: Ví dụ về nguy cơ chung

Hình 1 minh họa hai nhóm nguy cơ nhóm nguy cơ thứ nhất SRG-1 gồm các liên kết 𝑝1, 𝑝2, nhóm nguy cơ thứ hai SRG-2 gồm có các liên kết 𝑝2, 𝑝3, 𝑝4. Lúc này khi mà SRG-1 hay SRG-2 bị đứt thì các yêu cầu sử dụng các liên kêt trong nhóm này sẽ bị ảnh hưởng.

PHẦN 2: CÁC KẾT QUẢ ĐẠT ĐƯỢC

CHƯƠNG 3: ĐỀ XUẤT GIẢI THUẬT GIA CỐ TOPOLOGY MẠNG CHỊU LỖI CÓ XEM XÉT NGUY CƠ CHUNG

* 1. **Đề xuất giải thuật**

Để dễ dàng cho việc thực hiện thuật toán, chúng ta sẽ mô hình hóa topology mạng thành một đồ thị G(V, E), với V là tập các đỉnh trong đồ thị sẽ tương ứng với các node của topology mạng, V là tập các cạnh sẽ tương ứng với các đường liên kết giữa 2 điểm, trọng số các cạnh chính bằng khoảng cách giữa các đỉnh. Với mỗi 1 yêu cầu giữa (s,d) thì ta sẽ phải tìm đường đi làm việc và đường đi backup cho yêu cầu đó. Giải thuật đề xuất này sẽ thêm dần các cạnh mới vào đồ thị. Các bước thực hiện thuật toán sẽ thực hiện như sau:

Bước 1: Tạo 1 đồ thị G (V, E) với V là tập các đỉnh, E là tập cạnh. Mỗi cạnh (i ,j)(i, j thuộc V) thuộc E sẽ gán trọng số bằng khoảng cách của đỉnh i và đỉnh j.

Bước 2: Với mỗi yêu cầu (s,d) tìm đường đi làm việc là đường ngắn nhất trong G bằng thuật toán Dijkstra. Để tìm đường dự phòng cho yêu cầu, thuật toán xóa bỏ tất cả các cạnh thuộc đường đi làm việc và những cạnh trong G thuộc cùng 1 nhóm nguy cơ mà có mặt trong đường làm việc vừa tìm. Bước này nhằm mục đích tránh cho đường làm việc có bất kì cạnh chung nào với đường dự phòng và xem xét đến các nhóm nguy cơ chung. Tiếp theo, thuật toán Dijkstra được sử dụng để tìm đường dự phòng. Ở đây thuật toán Dijkstra sẽ ưu tiên dùng các đường còn lại trong tập E của đồ thị G(Ưu tiên bằng cách gắn trọng số của các cạnh này bằng 0). Nếu trong quá trình tìm đường phát sinh thêm cạnh mới không thuộc tập E, thì sẽ thêm cạnh đó vào tập E để sử dụng cho việc thực hiện tìm đường cho các yêu cầu sau.

Bước 3: Thêm vào G những cạnh vừa xóa bỏ khỏi đồ thị G. Lặp lại bước 2 cho đến khi không còn yêu cầu nào.

*Mã giả của thuật toán:*

INPUT:

+ Đồ thị G(V, E), với V là tập các điểm, E tập các cạnh.

+ Ma trận khoảng cách giữa các điểm: distance[][];

+ float weight[][]:ma trận trọng số (Ma trận trọng số dùng để tìm đường làm việc. Các điểm không có liên kết thì tương ứng trong weight[][] có trọng sô là vô cùng).

BEGIN:

+ Tính tổng chi phí của mạng cost = tổng độ dài các cạnh có đường đi vật lý trong G.

For loop: Yêu cầu 1 … Yêu cầu n:

Yêu cầu k (1<= k <= n):

+ Tìm đường làm việc:

/\* Sử dụng Dijktra để tìm working\_path trong G với ma trân trọng số là ma trận weight[][] .\*/

- working\_path = Dijkstra(src, dest, weight[][]);

- weight\_new[][] = distance[][].

- for i =0 ...số nút

for j = 0 … số nút

/\* gán trọng số ∞ với những cạnh thuộc working\_path \*/

if cạnh(i,j) thuộc working\_path

weight\_new[i][j] = ∞

end

/\* Xét nguy cơ chung \*/

if cạnh(m,n) thuộc nguy cơ chung với cạnh (i,j) &&

cạnh (i,j) thuộc working\_path

weight\_new[m][n] = ∞

end

/\* gán trọng số 0 với cạnh cũ \*/

if weight\_new[i][j] != ∞ && weight[i][j] != 0

weight\_new[i][j] = 0

end

end

end

+ Tìm đường đi back up:

/\* Sử dụng Dijkstra để tìm backup\_path trong G với ma trận trọng số weight\_new[][].\*/

- backup\_path = Dijkstra(src, dest, weight\_new[][])

- for i =0 ...số nút

for j = 0 … số nút

if cạnh(i,j) thuộc backup\_path &&weight [i][j] != 0

cost += weight\_new[i][j] /\*tăng chi phí\*/

weight [i][j] = distance[i][j] /\*thêm cạnh mới vào đồ thị\*/

end

end

end

k++

end

END

OUTPUT:

+ Topolopy mạng có tính chịu lỗi với chi phí thấp nhất (với chi phí bằng tổng độ dài các cạnh trong đồ thị).

* 1. **Cài đặt giải thuật**

Chương trình mô phỏng giải thuật được viết bằng ngôn ngữ lập trình Java, sử dụng trình dịch Eclipse IDE trên môi trường Ubuntu, được tổ chức thành các lớp và một số phương thức như sau:

* Lớp Node: mỗi đối tượng của lớp này mô tả một nút mạng gồm có id và tọa độ của nút. Tập nút được biểu diễn bằng một tập các Nodes, sử dụng kiểu dữ liệu ArrayList, nhờ sử dụng kiểu dữ kiệu này ta có thể dễ dàng thêm một nút hoặc truy xuất dữ liệu các nút.
* Lớp Edge: mỗi đối tượng của lớp này mô tả một cạnh của đồ thị gồm có nút nguồn và đích.
* Lớp Request: mỗi đối tượng của lớp này mô tả 1 yêu cầu gồm có nút nguồn và đích. Tập request được lưu dưới dạng một ArrayList.
* Lớp SRG: mỗi đối tượng của lớp này mô tả một nhóm nguy cơ chung gồm các cạnh thuộc nhóm nguy cơ chung.
* Lớp Graph: lớp này chứa các phương thức thực hiện giải thuật.
* Lớp GetData: lớp này chứa các phương thức thực hiện đọc dữ liệu từ file.

CHƯƠNG 4: CÁC KẾT QUẢ ĐẠT ĐƯỢC