|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  BÁO CÁO  **QUẢN LÝ VÀ QUY HOẠCH MẠNG VIỄN THÔNG**  **Đề tài:**  Xây dựng thiết kế topology sử dụng giải thuật MENTOR  GVHD: TS. Trần Thị Ngọc Lan  Nhóm sinh viên:   |  |  |  | | --- | --- | --- | | **Họ và tên** | **MSSV** | **Lớp** | | Nguyễn Hữu Hưng | 20182563 | ĐTVT. 08 – K63 | | Vũ Hiếu Trung | 20182839 | ĐTVT. 08 – K63 | | Nguyễn Minh Hiếu | 20182519 | ĐTVT. 08 – K63 | | Võ Minh Dũng | 20182452 | ĐTVT. 08 – K63 | | Giản Đình Toàn | 20182825 | ĐTVT. 05 – K63 |       Hà Nội, 6-2022 |

**LỜI NÓI ĐẦU**

Việc thiết kế cấu trúc liên kết (topology) cho các mạng hữu tuyến cố định là vô cùng quan trọng. Chỉ một sai sót nhỏ trong khâu thiết kế có thể tiêu tốn mỗi tháng hàng nghìn đô la. Trong khi đó, việc tính toán một cấu trúc liên kết tối ưu lại chẳng dễ dàng gì. Quá trình tối ưu cấu trúc liên kết bao gồm việc gán dung lượng cho các liên kết vật lý giữa các cặp nút với nhau để tối ưu chi phí, cùng lúc đó vẫn phải đáp ứng được lưu lượng đề ra.

Một phương pháp thường được sử dụng để giải quyết vấn đề trên là áp dụng giải thuật MENTOR (Mesh Network Topology assignment and Routing). Do chỉ phụ thuộc vào nguyên tắc thiết kế mạng, MENTOR có thể được ứng dụng cho nhiều loại mạng khác nhau, nhất là mạng ATM (Asynchronous Transfer Mode). Khuôn khổ đề tài của chúng em là tìm hiểu lý thuyết của giải thuật MENTOR và viết phần mềm mô phỏng cho giải thuật trên. Tuy rằng các giả thiết, điều kiện đưa ra có thể chưa đúng hoàn toàn với yêu cầu thực tế, nhưng chúng em hy vọng chương trình này có thể giúp mọi người nắm được quá trình xây dựng cấu trúc liên kết cho mạng.

Trong quá trình hoàn thiện đề tài, tuy chúng em đã cố gắng hết sức nhưng do những hạn chế về mặt kiến thức và thời gian nên đề tài còn nhiều thiếu sót. Chúng em kính mong nhận được những nhận xét quý giá của cô để có thể hoàn thiện đề tài này hơn.

**PHÂN CÔNG CÔNG VIỆC**

|  |  |
| --- | --- |
| **Họ và tên** | **Công việc** |
| Nguyễn Hữu Hưng | Code phần 4 |
| Võ Minh Dũng | Code phần 2 |
| Nguyễn Minh Hiếu | Code phần 3, giao diện |
| Vũ Hiếu Trung | Code phần 1 |
| Giản Đình Toàn | Làm báo cáo |

Mục Lục

[DANH MỤC HÌNH ẢNH i](#_Toc108035367)

[CHƯƠNG 1. LÝ THUYẾT CHUNG 1](#_Toc108035368)

[1.1 Tổng quan về hệ thống viễn thông 1](#_Toc108035369)

[1.2 Lý thuyết về thiết kế cấu trúc liên kết cho mạng 1](#_Toc108035370)

[1.3 Giới thiệu về giải thuật MENTOR 2](#_Toc108035371)

[1.3.1 Xác định các nút đường trục 2](#_Toc108035372)

[1.3.2 Xây dựng cây kết nối giữa các nút đường trục 4](#_Toc108035373)

[1.3.3 Thêm liên kết 5](#_Toc108035374)

[CHƯƠNG 2. TRIỂN KHAI GIẢI THUẬT 7](#_Toc108035375)

[2.1 Đề bài: 7](#_Toc108035376)

[2.2 Triển khai: 7](#_Toc108035377)

[2.3 Kết quả: 8](#_Toc108035378)

[KẾT LUẬN 14](#_Toc108035379)

[DANH MỤC TÀI LIỆU THAM KHẢO 15](#_Toc108035380)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: Chọn nút Home 5](#_Toc108035355)

[Hình 2.1 Phân loại các nút đường trục và nút truy nhập 8](#_Toc108035356)

[Hình 2.2 Xây dựng cây Prim-Dịjkstra 9](#_Toc108035357)

[Hình 2.3 Cây Prim-Dijkstra sau khi thêm liên kết sử dụng giải thuật MENTOR 1 9](#_Toc108035358)

[Hình 2.4 File lưu lượng thực tế đi qua các nút Backbone 10](#_Toc108035359)

[Hình 2.5 File số đường sử dụng trên từng liên kết và độ sử dụng trên liên kết đó 11](#_Toc108035360)

[Hình 2.6 Giá của mạng Backbone khi tăng thêm lưu lượng giữa các nút 11](#_Toc108035361)

[Hình 2.7 Mạng Backbone khi tăng thêm lưu lượng giũa các nút lên 20% 12](#_Toc108035362)

[Hình 2.8 Mạng backbone khi tăng thêm lưu lượng giữa các nút lên 30% 12](#_Toc108035363)

[Hình 2.9 Mạng Backbone khi tăng thêm lưu lượng giữa các nút lên 40% 13](#_Toc108035364)

# LÝ THUYẾT CHUNG

## Tổng quan về hệ thống viễn thông

Hệ thống mạng viễn thông là tập hợp các trang thiết bị kỹ thuật dùng để cung cấp dịch vụ viễn thông đến tay người sử dụng.

Khi nhìn dưới góc độ phần cứng, các bộ phận cấu thành nên một hệ thống mạng viễn thông bao gồm:

* Thiết bị đầu cuối thông tin: đưa thông tin của người sử dụng vào mạng và nhận thông tin của mạng cho người sử dụng.
* Thiết bị chuyển mạch: liên hệ giữa các đầu cuối theo yêu cầu.
* Thiết bị truyền dẫn: liên kết nhóm a với nhóm b và nhóm b với nhóm b
* (a - b): đường dây thuê bao.
* (b - b): đường dây trung kế.
* (a - b): mạng truy nhập (Access Network) và mạng lõi (Core Network).

Khi nhìn dưới góc độ phần mềm, các bộ phận cấu thành nên một hệ thống mạng viễn thông bao gồm:

* Cấu trúc liên kết (Topology) của mạng: cho biết các phần cứng liên hệ với nhau như thế nào.
* Giao thức mạng: cung cấp dịch vụ viễn thông cho người sử dụng.
* Giao thức để liên kết.
* Giao thức để trao đổi thông tin, giữa hai giao thức này có thể tách rời, có thể kết hợp với nhau.
* Quản lý và khai thác mạng.

## Lý thuyết về thiết kế cấu trúc liên kết cho mạng

Trong mạng viễn thông có rất nhiều nút mạng liên kết với nhau tạo thành một hệ thống mạng. Một hệ thống mạng lại chia ra thành các mạng truy nhập (Access Network) và các mạng đường trục (Backbone Network) có nhiệm vụ kết nối các mạng truy nhập với nhau.

Vì lý do kể trên, các nút trong một mạng viễn thông cũng chia làm 2 loại là nút truy nhập (Access Node) và nút đường trục (Backbone Node). Một mạng truy nhập chỉ có duy nhất một nút đường trục kết nối với nhiều nút truy nhập. Và như vậy, nút truy nhập ở một mạng truy nhập này khi muốn kết nối với nút truy nhập ở một mạng truy nhập khác thì bắt buộc phải thông qua các nút đường trục. Chính vì thế, mạng đường trục được sinh ra để làm cầu nối giúp các mạng truy nhập kết nối với nhau.

Bởi các chỉ tiêu, yêu cầu kỹ thuật và tính chất của hai mạng này là khác nhau, vậy nên bài toán được đặt ra ở đây là làm sao thiết kế được một hệ thống mạng trên các nút mạng cho trước thỏa mãn một số tiêu chuẩn, yêu cầu nhất định. Từ thực tiễn, ta có các thông số và giả thiết sau:

* Tổng số nút mạng trong toàn bộ hệ thống mạng được kí hiệu là N (Node) và được đánh số từ 1 ÷ N.
* Để biểu diễn sự liên hệ giữa các nút mạng với nhau, chúng ta có các ma trận sau:
  + Ma trận chi phí - Cost [NxN]: ma trận hai chiều đối xứng, chứa chi phí kết nối giữa các nút. Giá trị tại hàng i, cột j là chi phí đi từ nút i đến nút j.
  + Ma trận lưu lượng - Traffic [NxN]: ma trận hai chiều đối xứng, chứa lưu lượng giữa các nút với nhau.
  + Ma trận trọng số - Weight: ma trận một chiều chứa trọng số của một nút. Trọng số của một nút được tính bằng tổng lưu lượng đi vào và đi ra khỏi nút đó:

|  |  |
| --- | --- |
|  | (.) |

## Giới thiệu về giải thuật MENTOR

Giải thuật MENTOR gồm 3 bước chính như sau:

### Xác định các nút đường trục

Bước đầu tiên của giải thuật MENTOR là phân loại các nút trong mạng thành nút đường trục (Backbone Node) và nút đầu cuối (End Node). Để làm được điều đó, chúng ta phải áp dụng một phương pháp có tên hợp lại theo ngưỡng (Threshold Clustering). Từ trọng số các nút, ta xác định các nút thỏa mãn tiêu chuẩn trọng số để làm nút đường trục. Theo đó, trọng số chuẩn hóa (Normalized Weight) của một nút được tính như sau:

|  |  |
| --- | --- |
|  | (.) |

Trong đó Capacity (đôi khi được viết tắt là C) được gọi là dung lượng liên kết. Nếu Normalized Weight(i)>Threshold Weight (trọng số ngưỡng – đôi khi được viết tắt thành W) cho trước thì nút đó được chọn làm nút đường trục. Tất cả các nút không thỏa mãn điều kiện về trọng số và “gần” nút đường trục sẽ được chọn làm nút truy nhập (Access Node). Một nút được gọi là “gần” nút đường trục khi chi phí từ nó đến nút đường trục nhỏ hơn:

|  |  |
| --- | --- |
|  | (.) |
|  | (.) |
|  | (.) |

Điều đó có nghĩa là khi lấy một nút đường trục i bất kỳ làm tâm, quay một vòng tròn bán kính Radius thì tất cả các nút nằm trong vòng tròn mà không phải nút đường trục sẽ là nút truy nhập của nút đường trục i.

Có thể xảy ra trường hợp là sau phương pháp hợp lại theo ngưỡng kể trên vẫn còn một số nút chưa được phân loại là nút đường trục hay nút truy nhập. Khi đó ta phải sử dụng tiếp một phương pháp khác. Chúng ta sẽ tìm tâm của trọng lực (Center of Gravity - CG) theo công thức:

|  |  |
| --- | --- |
|  | (.) |
|  | (.) |

với (xi,yi) và Weighti là tọa độ và trọng số của nút i. Lưu ý là tọa độ của CG không cần tương ứng với bất kỳ một nút nào trong thực tế. Tiếp đó chúng ta sẽ tính khoảng cách từ nút i bất kỳ đến CG theo công thức:

|  |  |
| --- | --- |
|  | (.) |

Chúng ta tính các giá trị khoảng cách tối đa tới CG và trọng số tối đa như sau:

|  |  |
| --- | --- |
|  | (.) |
|  | (.) |

Sau đó, chúng ta sẽ tính giá trị thưởng (Merit) theo công thức:

|  |  |
| --- | --- |
|  | (.) |

Trong các nút chưa được xác định, chọn nút có giá trị thưởng lớn nhất trở thành nút đường trục. Ta lấy nó làm tâm, quay một vòng tròn bán kính Radius và xác định các nút truy nhập. Quá trình này sẽ được tiếp tục cho đến khi tất cả các nút đã được phân loại.

### Xây dựng cây kết nối giữa các nút đường trục

Sau khi đã phân loại tất cả các nút, chúng ta sẽ tiến hành xây dựng một cây kết nối tất cả các nút đường trục với nhau. Đầu tiên chúng ta chọn một nút đường trục có moment nhỏ nhất làm nút đường trục trung tâm.

|  |  |
| --- | --- |
|  | (.) |

trong đó N, N\* là các nút đường trục. Kế đó chúng ta sẽ xây dựng cây Prim – Dijkstra để kết nối nút đường trục trung tâm với các nút đường trục còn lại. Về các bước triển khai, cây Prim – Dijkstra khá giống với cây Prim và cây Dijkstra, duy chỉ có nhãn (Label) của nó có hơi khác hai cây trên. Cụ thể ta có:

|  |  |
| --- | --- |
|  | (.) |

Với α=0 ta có nhãn Prim, còn với với α=1 ta có nhãn Dijkstra.

Nhãn Prim =

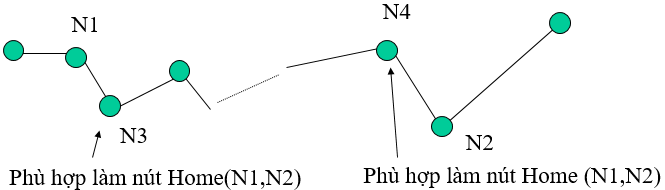
Nhãn Dijkstra =

### Thêm liên kết

Ở bước này chúng ta sẽ đưa ra các khái niệm mới như dãy (Sequence) và nút Home để thêm liên kết nhằm tối ưu thiết kế. Bằng cách sử dụng cây Prim – Dijkstra ở bước hai, chúng ta có thể xác định dãy các nút thỏa mãn những tiêu chí sau:

* Các nút được sắp xếp theo thứ tự từ ngoài vào trong.
* Không xếp cặp nút (N1, N2) cho đến khi tất cả các cặp nút (N1\*, N2\*) đã được xếp; trong đó N1, N2 nằm trên liên kết giữa N1\* và N2\*.
* Những liên kết dài nhất sẽ được xếp đầu tiên.

Với mỗi cặp nút N1, N2 không liền kề nhau, ta chọn nút Home là nút trung gian.



Hình .: Chọn nút Home

Trong trường hợp có nhiều nút có thể chọn làm nút Home, ví dụ như giữa N1 và N2 có hai nút trung gian N3 và N4 thì ta sẽ chọn nút Home là N3 nếu:

|  |  |
| --- | --- |
|  | (.) |

Ngược lại chúng ta sẽ chọn nút Home là N4. Xét từng cặp nút (N1,N2) một lần duy nhất, ta triển khai giải thuật sau:

* Từ tham số dung lượng liên kết C mà đề bài đã cho, tính:

|  |  |
| --- | --- |
|  | (.) |

* Tính độ sử dụng (Utilization) của liên kết:

|  |  |
| --- | --- |
|  | (.) |

* Thêm liên kết nếu U>U\_min, còn không thì di chuyển lưu lượng thông qua mạng (ví dụ như thêm lưu lượng cho cả và , làm tương tự với ). Sẽ xảy ra trường hợp liên kết (N1,N2) thuộc cây Prim – Dijkstra ban đầu, khi đó ta chỉ việc thêm liên kết trực tiếp.

# TRIỂN KHAI GIẢI THUẬT

## Đề bài:

Cho mạng gồm 100 nút. Các nút được đặt một cách ngẫu nhiên trên mặt phẳng kích thước 1000x1000. Giá của mỗi liên kết được tính bẳng round (0.4x khoảng cách đề các).

1. Sử dụng giải thuật MENTOR để tìm nút backbone và các nút truy nhập tương ứng với nút Backbone. Biết W=2, R=0,3. dung lượng liên kết C=12. Lưu lượng giữa nút i và i+2 là 1, Lưu lượng giữa nút i và nút i+54 là 2 và lưu lượng giữa nút i và nút i+88 là 3, lưu lượng giữa nút i và i+98 là 4, lưu lượng giữa nút 7 và nút 28 là 18, lưu lượng giữa nút 12 và nút 46 là 17 và lưu lượng giữa nút 27 và nút 48 là 4.

2. Hãy tính lưu lượng thực tế giữa các nút Backbone (ghi ra file)

3. Sử dụng giải thuật Mentor 1 để thiết kế toopology backbone biết umin = 85%. α =0.4. Đưa ra kết quả ra file số đường sử dụng trên từng liên kết và độ sử dụng trên liên kết đó.

4. Nếu tăng thêm lưu lượng giữa các nút lên 20%, 30%, 40% thì mạng backbone thì giá của mạng backbone vừa tạo ra ở mục 3 thay đổi như thế nào. Đưa ra đánh giá và nhận xét.

## Triển khai:

Giải thuật MENTOR được chúng em triển khai trên MATLAB, với kết quả thu được như sau:

- Đầu tiên chương trình sẽ tạo ra các nút mạng ở các vị trí ngẫu nhiên.

- Sau đó sử dụng giải thuật đã trình bày ở phần lý thuyết chung để tìm ra các nút đường trục (nút backbone) và các nút truy nhập tương ứng. Trong đó, nút đường trục trung tâm được biểu diễn dưới dạng hình vuông đặc màu xanh lá, các nút đường trục khác được biểu diễn dưới dạng hình vuông đặc màu đỏ, còn các nút truy nhập là hình tròn màu đen.

- Tiếp theo chương trình sẽ xây dựng cây Prim – Dijkstra để kết nối các nút đường trục với nhau (đường màu đỏ). Cuối cùng chương trình sẽ tính toán và thêm liên kết (đường màu xanh lá).

## Kết quả:

* Sau khi chạy chương trình tìm được các nút backbone bao gồm: 7, 1, 100, 8, 98, 19
* Nút backbone trung tâm là: 1
* Chương trình thực hiện phân loại nút đường trục và nút truy nhập như hình bên dưới:

Diagram

Description automatically generated

Hình . Phân loại các nút đường trục và nút truy nhập

* Xây dựng cây Prim – Dijkstra và thêm liên kết:

Chart, radar chart

Description automatically generated

Hình . Xây dựng cây Prim-Dịjkstra

Chart, radar chart

Description automatically generated

Hình . Cây Prim-Dijkstra sau khi thêm liên kết sử dụng giải thuật MENTOR 1

* Lưu lượng thực tế đi qua các nút backbone:

Graphical user interface, table

Description automatically generated

Hình . File lưu lượng thực tế đi qua các nút Backbone

* Đưa ra kết quả ra file số đường sử dụng trên từng liên kết và độ sử dụng trên liên kết đó

Graphical user interface, application, table, Excel

Description automatically generated

Hình . File số đường sử dụng trên từng liên kết và độ sử dụng trên liên kết đó

Giá của mạng backbone tính được ở yêu cầu 3 là 901 sau khi tăng thêm lưu lượng giữa các nút thêm 20% thì giá của mạng backbone đã tăng lên với giá là 4806, sau khi tăng thêm lưu lượng giữa các nút thêm 30% thì giá của mạng backbone đã tăng lên với giá là 4418.7, sau khi tăng thêm lưu lượng giữa các nút thêm 40% thì giá của mạng backbone đã tăng lên với giá là 8416.8.

Text

Description automatically generated

Hình . Giá của mạng Backbone khi tăng thêm lưu lượng giữa các nút

Chart

Description automatically generated

Hình . Mạng Backbone khi tăng thêm lưu lượng giũa các nút lên 20%

Chart

Description automatically generated

Hình . Mạng backbone khi tăng thêm lưu lượng giữa các nút lên 30%

Chart, scatter chart

Description automatically generated

Hình . Mạng Backbone khi tăng thêm lưu lượng giữa các nút lên 40%

KẾT LUẬN

Qua bài tập lớn này, chúng em đã phần nào hiểu rõ hơn về thuật toán MENTOR và cách hình thành cây theo thuật toán MENTOR cũng như xây dựng mạng đường trục. Đặc biệt, chúng em đã có thể mô phỏng thuật toán và giúp cho chúng em hiểu một cách sâu sắc hơn.

Tuy có nhiều khó khăn nhưng nhờ sự hướng dẫn tận tình của cô Trần Thị Ngọc Lan cũng như nỗ lực của bản thân mà chúng em đã hoàn thành bài tập lớn này. Do thời gian có hạn và kiến thức lập trình còn chưa tốt nên không thể tránh khỏi sai sót. Chúng em mong nhận được những lời khuyên, lời góp ý của cô để chúng em có thể hiểu thêm về bài tập lớn nói riêng và môn học nói chung.

Chúng em xin chân thành cảm ơn sự giúp đỡ của cô!

DANH MỤC TÀI LIỆU THAM KHẢO

1. Khoa điện tử viễn thông. “Bài giảng môn tổ chức mạng viễn thông”.
2. A. Kershenbaum, P. Kermani, G.A. Grover, "MENTOR: an algorithm for mesh network topological optimization and routing," *IEEE Transactions on Communications,* vol. 39, no. 4, pp. 503-513, 1991.