**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL 4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH**

**Đề tài 402: Xây dựng chương trình**

**định đường đi không thích nghi (Shortest Path Routing)**

**SINH VIÊN THỰC HIỆN:**

**Lê Phan Phú Việt Lớp: 21TCLC\_DT1 Nhóm: 10**

**Đoàn Văn Việt Lớp: 21TCLC\_DT1 Nhóm: 10**

**GIẢNG VIÊN HƯỚNG DẪN:**

**Th.S Nguyễn Văn Nguyên**

**Đà Nẵng, Tháng 11/2023**

MỤC LỤC

[MỤC LỤC 2](#_Toc152186399)

[DANH SÁCH HÌNH VẼ 3](#_Toc152186400)

[GIỚI THIỆU ĐỀ TÀI 4](#_Toc152186401)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 5](#_Toc152186402)

[1.1. Client 5](#_Toc152186403)

[1.2. Server 5](#_Toc152186404)

[1.3. Quan hệ giữa server và client 5](#_Toc152186405)

[1.4. Sử dụng socket với TCP 6](#_Toc152186406)

[1.4.1. Socket 6](#_Toc152186407)

[1.4.2. TCP (Transmission Control Protocol) 6](#_Toc152186408)

[1.4.3. Quan hệ giữa socket và TCP 6](#_Toc152186409)

[1.5. Thuật toán Dijkstra 7](#_Toc152186410)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc152186411)

[2.1. Client 9](#_Toc152186412)

[2.2. Server 9](#_Toc152186413)

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 10](#_Toc152186414)

[3.1. Triển khai 10](#_Toc152186415)

[3.1.1. Xác định mục tiêu 10](#_Toc152186416)

[3.1.2. Thiết lập kế hoạch dự án 10](#_Toc152186417)

[3.1.3. Xác định mô hình, thuật toán cần sử dụng 10](#_Toc152186418)

[3.1.4. Xây dựng chương trình 10](#_Toc152186419)

[3.1.5. Kiểm thử chương trình và sửa lỗi 10](#_Toc152186420)

[3.2. Kết quả chương trình 11](#_Toc152186421)

[3.2.1. Giao diện client 11](#_Toc152186422)

[3.2.2. Đường đi ngắn nhất 11](#_Toc152186423)

[3.2.3. Số đỉnh đường đi ngắn nhất tìm được đi qua 11](#_Toc152186424)

[3.2.4. Bảng chỉ đường đi ngắn nhất đến các đỉnh còn lại 12](#_Toc152186425)

[3.2.5. Khoảng cách ngắn nhất 12](#_Toc152186426)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 13](#_Toc152186427)

[TÀI LIỆU THAM KHẢO 15](#_Toc152186428)

DANH SÁCH HÌNH VẼ

[*Hình 1: Mô phỏng thuật toán Dijkstra* 7](#_Toc152099076)

[*Hình 2: Giao diện client* 10](#_Toc152099077)

[*Hình 3: Đường đi ngắn nhất* 10](#_Toc152099078)

[*Hình 4: Các đỉnh đi qua trên đường đi ngắn nhất* 11](#_Toc152099079)

[*Hình 5: Bảng chỉ đường ngắn nhất đến các đỉnh còn lại* 11](#_Toc152099080)

[*Hình 6: Khoảng cách ngắn nhất* 11](#_Toc152099081)

GIỚI THIỆU ĐỀ TÀI

Đề tài "tìm đường đi ngắn nhất của router không thích nghi" là một lĩnh vực nghiên cứu đầy thách thức trong hệ thống mạng và viễn thông. Trong ngữ cảnh này, "không thích nghi" ám chỉ rằng các thuật toán định tuyến không tự động thích nghi hoặc điều chỉnh dựa trên thay đổi trong môi trường mạng.

Nghiên cứu này sẽ tập trung vào việc phân tích và phát triển các thuật toán định tuyến cổ điển, không thích nghi, nhằm xác định và duy trì đường đi ngắn nhất giữa các thiết bị trong mạng. Thay vì dựa vào các cập nhật tự động, chúng ta sẽ xem xét cách thuật toán có thể duy trì sự ổn định và hiệu suất mà không phải dựa vào thông tin môi trường thay đổi liên tục.

Qua đó, chúng tôi dự kiến thử nghiệm và so sánh hiệu suất của các thuật toán này trong các tình huống mạng khác nhau. Nghiên cứu này mang lại cái nhìn mới về việc cân nhắc giữa tính linh hoạt và ổn định trong lựa chọn đường đi giữa các router, đặc biệt quan trọng trong các hệ thống yêu cầu sự kiểm soát chặt chẽ và không mong muốn tự động thích nghi.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## Client

* **Khái niệm**

Một client là một thiết bị hoặc phần mềm sử dụng dịch vụ hoặc tài nguyên từ server. Clients có thể là máy tính cá nhân, điện thoại di động, máy tính bảng, hoặc bất kỳ thiết bị kết nối mạng nào.

* **Chức năng**

Clients gửi yêu cầu đến server để nhận dữ liệu, tài nguyên hoặc thực hiện một loạt các thao tác khác. Trong môi trường mạng, các trình duyệt web, ứng dụng di động, và các ứng dụng desktop thường là các loại client phổ biến.

## Server

* **Khái niệm**

Một server là một thiết bị hoặc phần mềm cung cấp dịch vụ hoặc tài nguyên cho các máy tính khác trên mạng, được biết đến như là "clients." server có thể chia thành nhiều loại như web server (cung cấp trang web), file server (lưu trữ và chia sẻ tệp tin), database server (quản lý cơ sở dữ liệu), và nhiều loại khác.

* **Chức năng**

Server đáp ứng các yêu cầu từ clients bằng cách cung cấp dữ liệu, tài nguyên hoặc dịch vụ. Chúng có khả năng xử lý đồng thời nhiều kết nối từ các clients khác nhau.

## Quan hệ giữa server và client

* **Giao thức truyền thông**

Server và client giao tiếp thông qua các giao thức truyền thông như http (hypertext transfer protocol), ftp (file transfer protocol), và TCP/IP (transmission control protocol/internet protocol).

* **Tương tác**

Quan hệ giữa server và client thường dựa trên mô hình yêu cầu/phản hồi (request/response), trong đó client gửi yêu cầu và server đáp ứng bằng cách cung cấp dữ liệu hoặc dịch vụ.

* **Mô hình ứng dụng**

Mô hình client-server là cơ sở cho nhiều ứng dụng mạng, từ trang web đến trò chơi trực tuyến và hệ thống email.

* Tổng quan về server và client là quan trọng để hiểu cách các thiết bị và phần mềm tương tác trong môi trường mạng, đặc biệt trong ngữ cảnh của các ứng dụng và dịch vụ trực tuyến.

## Sử dụng socket với TCP

### Socket

* **Khái niệm**

Socket là một giao diện truyền thông giữa các chương trình chạy trên máy tính khác nhau thông qua mạng. Nó là một cổng kết nối mà qua đó dữ liệu có thể được truyền đi và nhận về.

* **Loại socket**

Có hai loại socket chính: socket UDP (User Datagram Protocol) và socket TCP. Socket UDP được sử dụng cho truyền thông không đảm bảo tính toàn vẹn của dữ liệu, trong khi socket TCP được sử dụng cho truyền thông đảm bảo tính toàn vẹn và theo thứ tự của dữ liệu.

### TCP (Transmission Control Protocol)

* **Khái niệm**

TCP là một trong các giao thức truyền thông cơ bản của internet. Nó đảm bảo việc truyền dữ liệu một cách đáng tin cậy giữa các thiết bị trên mạng.

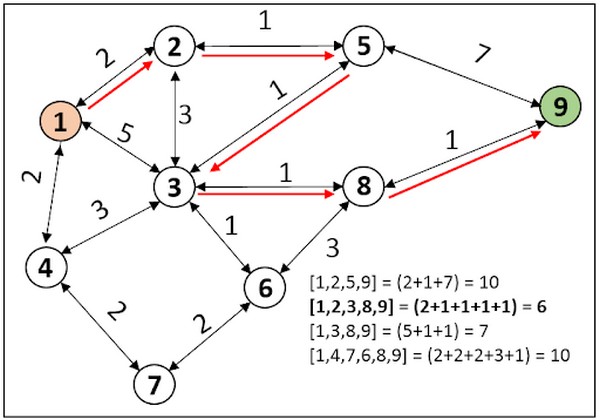
* **Đặc điểm**
  + Đảm bảo kết nối: TCP thiết lập một kết nối trước khi truyền dữ liệu và đảm bảo rằng cả hai bên đã sẵn sàng trước khi bắt đầu truyền thông tin.
  + Đảm bảo tính toàn vẹn: dữ liệu được chia thành các gói và mỗi gói đều được kiểm tra tính toàn vẹn để đảm bảo không có lỗi nào xảy ra trong quá trình truyền.
  + Đảm bảo theo thứ tự: dữ liệu được truyền và nhận theo thứ tự chính xác để đảm bảo tính đúng đắn của thông tin.

### Quan hệ giữa socket và TCP

* Sử dụng socket trong TCP: khi lập trình mạng với TCP, socket thường được sử dụng để tạo, kết nối và truyền thông giữa các thiết bị. Mỗi kết nối TCP sẽ được liên kết với một socket cụ thể để quản lý truyền thông.
* Trong lập trình mạng, việc sử dụng socket và TCP cùng nhau giúp tạo ra các ứng dụng mạng đáng tin cậy và linh hoạt. Socket cung cấp giao diện cho việc truyền thông, trong khi TCP đảm bảo tính toàn vẹn và theo thứ tự của dữ liệu.

## Thuật toán Dijkstra

* Thuật toán Dijkstra là một thuật toán quan trọng trong lĩnh vực đồ thị và tìm kiếm đường đi. Được đặt tên theo nhà toán học Edsger W. Dijkstra, thuật toán này được thiết kế để tìm đường đi ngắn nhất giữa một đỉnh bắt đầu và tất cả các đỉnh khác trong đồ thị có trọng số dương. Dijkstra thường được áp dụng trong các hệ thống mạng và định tuyến.
* Thuật toán hoạt động bằng cách duyệt qua các đỉnh trong đồ thị theo thứ tự tăng dần của khoảng cách từ đỉnh bắt đầu. Khi duyệt mỗi đỉnh, nó cập nhật các khoảng cách ngắn nhất từ đỉnh bắt đầu đến các đỉnh kề của nó. Thuật toán sử dụng một hàng đợi ưu tiên (priority queue) để lựa chọn đỉnh tiếp theo cần xem xét dựa trên khoảng cách hiện tại từ đỉnh bắt đầu.
* Thuật toán Dijkstra đảm bảo rằng khi nó hoàn tất, mỗi đỉnh sẽ có khoảng cách ngắn nhất từ đỉnh bắt đầu. Nó cũng tạo ra một cây đường đi ngắn nhất từ đỉnh bắt đầu đến tất cả các đỉnh khác trong đồ thị.
* Thuật toán Dijkstra là một công cụ mạnh mẽ để giải quyết vấn đề tìm kiếm đường đi ngắn nhất trong các ứng dụng thực tế như định tuyến mạng và lên lịch di chuyển trong hệ thống vận tải.



*Hình 1: Mô phỏng thuật toán Dijkstra*

* Khi bắt đầu tìm hiểu thuật toán Dijkstra đa số các bạn đều sẽ thấy phức tạp bởi nó là tính toán của một chuỗi chu kỳ vòng lặp trông khá rắc rối, tuy nhiên, tóm tắt thuật toán có thể thực hiện 5 bước đơn giản sau:

**Bước 1**: Đánh dấu đỉnh nguồn (đỉnh mở đầu) là $0$ và các đỉnh còn lại là “vô cùng”.

**Bước 2**: Gọi đỉnh chưa xét với giá trị đánh dấu min là $C$ (current node).

**Bước 3**: Mỗi đỉnh kề $N$ với đỉnh $C$, ta cộng giá trị đang đánh dấu của đỉnh $C$ với trọng số của cạnh nối đỉnh current node cùng đỉnh kề, nếu kết quả nhỏ hơn giá trị đang đánh dấu ở $N$ thì ta cập nhật giá trị mới đó cho đỉnh.

**Bước 4:** Đánh dấu đỉnh $C$ đã xét.

**Bước 5**: Tiếp tục vòng lặp tại bước 2 cho đến khi không còn đỉnh chưa xét.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Client

* Trong quá trình khởi động, client thực hiện bước đọc dữ liệu từ tệp văn bản chứa đầu vào (input.txt) và sáng tạo một đồ thị dựa trên số lượng điểm và khoảng cách giữa các cặp điểm trong tệp đầu vào.
* Tiếp theo, client gửi chuỗi số thu được từ quá trình đọc tệp đầu vào tới server để thực hiện xử lý.
* Khi nhận được chuỗi số đã được xử lý từ server, client vẽ đường đi ngắn nhất trên đồ thị đã tạo trước đó. Đồng thời, client xuất chuỗi số đã được xử lý ra bảng chỉ đường.
* Người dùng có thể yêu cầu client gửi yêu cầu tìm đường đi ngắn nhất đến một điểm khác, và client tiếp tục gửi chuỗi số gồm hai điểm cần tìm đường đi ngắn nhất đến server.
* Client lặp lại công việc nhận và gửi điểm cần tìm đường đi ngắn nhất cùng với chuỗi ghi lại đường đi ngắn nhất từ server, tạo ra một trải nghiệm tương tác và linh hoạt cho người dùng.

## Server

* Server được khởi động trước client và duy trì sự chờ đợi để kết nối với client.
* Khi nhận được chuỗi đầu vào từ client, server tiến hành xử lý theo thuật toán dijkstra và trả về chuỗi số biểu diễn đường đi ngắn nhất cho client.
* Server tiếp tục đợi chuỗi đầu vào từ client và lặp lại quá trình xử lý và gửi trả kết quả về cho client. Sự duy trì liên tục này giúp server phản hồi nhanh chóng và hiệu quả đối với các yêu cầu của client, tạo ra một hệ thống linh hoạt và phản hồi tốt giữa client và server.

# CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

## Triển khai

### Xác định mục tiêu

* Mục tiêu của đồ án là xây dựng được chương trình tìm đường đi ngắn nhất không thích nghi bằng thuật toán Dijkstra sử dụng mô hình client – server và truyền nhận dữ liệu thông qua giao thức TCP.
* Chương trình cần có giao diện trực quan, thân thiện với người sử dụng; tốc độ xử lý đạt mức ổn

### Thiết lập kế hoạch dự án

* Lập kế hoạch chi tiết cho từng giai đoạn của dự án: nghiên cứu thuật toán, mô hình; xây dựng chương trình; kiểm thử và sửa lỗi cho chương trình; viết báo cáo.
* Phân chia công việc giữa các thành viên trong nhóm cho phù hợp với thế mạnh của từng thành viên.

### Xác định mô hình, thuật toán cần sử dụng

* Xác định mô hình mà ứng dụng được xây dựng theo là mô hình client – server.
* Sử dụng giao thức TCP để truyền và nhận dữ liệu.
* Thuật toán tìm đường đi ngắn nhất được sử dụng là thuật toán Dijkstra

### Xây dựng chương trình

* Xây dựng chương trình tìm đường đi ngắn nhất không thích nghi sử dụng thuật toán Dijkstra bằng ngôn ngữ lập trình Java.
* Xây dựng chương trình theo mô hình client – server sử dụng giao thức TCP để trao đổi dữ liệu dạng chuỗi các chữ cái giữa client và server.
* Kết hợp hai chương trình trên để tạo nên chương trình tìm đường đi ngắn nhất không thích nghi theo mô hình client – server hoàn chỉnh.

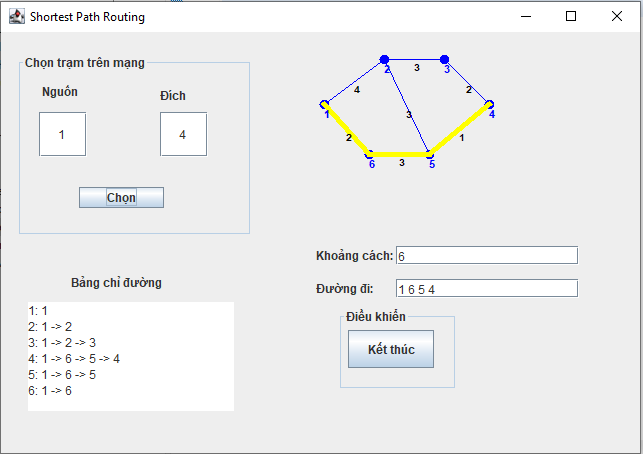
### Kiểm thử chương trình và sửa lỗi

* Kiểm thử chương trình xây dựng được với nhiều ma trận đồ thị khác nhau; đỉnh nguồn, đỉnh đích khác nhau để phát hiện lỗi.
* Sửa một số lỗi phát hiện được trong quá trình xây dựng và kiểm thử.

## Kết quả chương trình

### Giao diện client

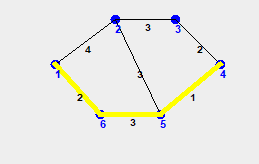
* Giao diện client được thiết kế đơn giản, dễ sử dụng, hiển thị những thông tin như đồ thị được vẽ bởi client và kết quả đường đi ngắn nhất mà client nhận được từ server.



*Hình 2: Giao diện client*

### Đường đi ngắn nhất

* Đồ thị hiển thị đường đi ngắn nhất 1-6-5-4 được vẽ bằng nét lớn màu vàng nổi bật để người dùng dễ dàng nắm bắt hơn



*Hình 3: Đường đi ngắn nhất*

### Số đỉnh đường đi ngắn nhất tìm được đi qua

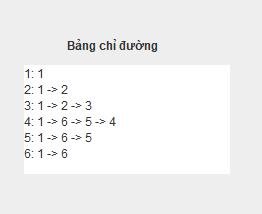
* Khung chữ thể hiện các đỉnh mà đường đi ngắn nhất đi qua từ đỉnh nguồn là 1 và đỉnh đích là 4.



*Hình 4: Các đỉnh đi qua trên đường đi ngắn nhất*

### Bảng chỉ đường đi ngắn nhất đến các đỉnh còn lại

* Bảng chỉ đường cho phép người dùng biết được đường đi ngắn nhất từ đỉnh nguồn đã chọn đến các đỉnh khác đỉnh đích.



*Hình 5: Bảng chỉ đường ngắn nhất đến các đỉnh còn lại*

### Khoảng cách ngắn nhất

* Khung chữ thể hiện khoảng cách phải đi từ đỉnh nguồn đến đỉnh đích khi đi đường đi ngắn nhất



*Hình 6: Khoảng cách ngắn nhất*

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **Kết luận**
   1. **Kết quả đã đạt được**

* Chúng em đã nắm vững được những nội dung về mô hình client – server cũng như những kiến thức về giao thức TCP, cách mạng máy tính và hệ điều hành hoạt động thông qua quá trình nghiên cứu.
* Chúng em đã hiểu và ứng dụng được thuật toán Dijkstra để viết chương trình định đường đi ngắn nhất không thích nghi.
* Từ đó, chúng em đã kết hợp thuật toán Dijkstra và mô hình client – server để xây dựng nên chương trình tìm đường đi ngắn nhất không thích nghi giúp trao đổi thông tin về đồ thị và đường đi ngắn nhất giữa hai máy.
* Chương trình xây dựng được có giao diện thân thiện người dùng, hoạt động nhanh, hiệu quả.
  1. **Kết quả chưa đạt được**
* Chương trình vẫn còn giới hạn về số lượng client cùng kết nối đến một server.
* Giao diện tuy đáp ứng được chức năng cơ bản tuy nhiên vẫn chưa có khả năng đáp ứng khi tăng số lượng đỉnh lên quá một số lượng nhất định. Khi đó đồ thị bị mất một phần do cửa sổ không đủ rộng và chưa có tính năng tự động dãn khi đồ thị quá lớn.
* Khả năng xử lý của chương trình tuy nhanh nhưng khi gặp những đồ thị rất nhiều đỉnh thì bị chậm lại đáng kể.
* Chưa áp dụng được chương trình để tìm đường đi cho các gói tin với đỉnh là các router.

1. **Hướng phát triển**
   1. **Ứng dụng thực tế vào router**

* Ứng dụng tìm đường đi ngắn nhất cho các gói tin giữa các router giúp việc truyền gói tin được nhanh chóng.
  1. **Nâng cấp giao diện**
* Hiện giao diện tuy đã đáp ứng được nhu cầu cơ bản tuy nhiên vẫn còn nhiều thiếu sót. Chúng em sẽ nâng cấp để giao diện có thể đáp ứng đồ thị lớn hơn, tăng tính thẩm mỹ của các thành phần giao diện như nút bấm, khung chữ,… để đem lại cái nhìn hiện đại cho chương trình.
  1. **Áp dụng đa luồng, đa tuyến để sử dụng trên nhiều máy hơn**
* Chương trình hiện còn hạn chế về số lượng client – server kết nối và sử dụng chương trình, trong tương lai chúng em sẽ mở rộng hơn, cho phép nhiều client cùng kết nối đến server để sử dụng ứng dụng hơn.
  1. **Tối ưu hóa thuật toán**
* Nghiên cứu và thử nghiệm các thuật toán tìm đường đi ngắn nhất khác nhau như Bellman-Ford, hoặc A\* để tối ưu hóa hiệu suất và độ chính xác của hệ thống.
  1. **Mở rộng ứng dụng vào các hệ thống mạng lớn hơn**
* Ứng dụng cho các hệ thống mạng lớn và phức tạp như mạng lưới (mesh network) hay mạng có số lượng lớn các nút.

TÀI LIỆU THAM KHẢO

**[1]** PGS.TS Nguyễn Đức Nghĩa, Cấu trúc dữ liệu và thuật toán, Nhà xuất bản Bách Khoa Hà Nội.

**[2]** Dijkstra - Ấn Phẩm Chuyên Đề Cho Kỹ Sư Phần Mềm Người Việt - Tập 2, Nhà xuất bản Đồng Nai.

**[3]** Trần Thông Quế, Cấu Trúc Dữ Liệu & Thuật Toán, Nhà xuất bản Thông Tin & Truyền Thông, Năm xuất bản 2014.

**[4]** Douglas E. Comer, Computer Networks and Internets with Internet Applications, Prentice-Hall,1993.

**[5]** W. Richard Stevens, Unix Network Programming - Networking APIs: Socket and XTI, Vol 1, Pearson Education Asia, 1999.