**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

Icon

Description automatically generated

**BÁO CÁO CUỐI KÌ**

**Môn: CÁC HỆ THỐNG PHÂN TÁN**

**Đề Tài:**

Mạng nội bộ của nhóm, mỗi máy tính có id từ 0 đến N-1, và được gắn nhán đồng hồ Lamport= id + 5. Xây dựng ứng dụng server và client loại trừ tương hỗ theo ứng dụng giải thuật bầu chọn theo thuật toán Ricart & Agrawala.

**Giảng viên hướng dẫn : LÊ NGỌC BẢO**

**Sinh viên thực hiện : THẨM NGỌC ÁNH**

**MSSV : N19DCCN012**

**Lớp : D19CQCNPM01-N**

*Tp. Hồ Chí Minh, 12/2022*

Mục lục

[CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ CẤU TRÚC CHƯƠNG TRÌNH 3](#_Toc122562209)

[1.1. Server 3](#_Toc122562210)

[1.2. Client 3](#_Toc122562211)

[1.3. Đồng hồ Lamport 3](#_Toc122562212)

[1.4. Ricart & Agrawala 3](#_Toc122562213)

[1.5. Loại trừ tương hỗ 4](#_Toc122562214)

[1.6. Giao thức kết nối 4](#_Toc122562215)

[CHƯƠNG 2: CÔNG NGHỆ ỨNG DỤNG VÀO ĐỒ ÁN 5](#_Toc122562216)

[2.1. Lập trình Socket trong Java 5](#_Toc122562217)

[2.2. TCP Socket 6](#_Toc122562218)

[2.3. Port Number 7](#_Toc122562219)

[CHƯƠNG 3: CHẠY CHƯƠNG TRÌNH 8](#_Toc122562220)

[3.1. Thành phần 8](#_Toc122562221)

[3.2. Mô tả quá trình chạy chương trình 9](#_Toc122562222)

# CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ CẤU TRÚC CHƯƠNG TRÌNH

* 1. Server

Máy chủ là bất kỳ quy trình (chương trình) độc lập nào chạy trên hệ thống có thể chấp nhận yêu cầu từ các quy trình khác, thực hiện các yêu cầu đó và sau đó trả lại kết quả (nếu có) cho người yêu cầu.

* 1. Client

Quy trình máy khách là bất kỳ quy trình (chương trình) độc lập nào yêu cầu các dịch vụ của quy trình khác.

* 1. Đồng hồ Lamport
* Thuật toán của dấu thời gian Lamport là một thuật toán đơn giản được sử dụng để xác định thứ tự của các sự kiện trong một hệ thống máy tính phân tán. Vì các nút hoặc quy trình khác nhau thường sẽ không được đồng bộ hóa hoàn hảo, thuật toán này được sử dụng để cung cấp thứ tự một phần của các sự kiện với chi phí tối thiểu và về mặt khái niệm cung cấp điểm bắt đầu cho phương pháp đồng hồ véc tơ nâng cao hơn.
* Để đồng bộ đồng hồ logic, Lamport dựa trên quan điểm nhân quả và đưa ra định nghĩa mối quan hệ gọi được gọi là xảy ra - trước khi (happens-before). Sự kiện a xảy ra trước sự kiện b, ký hiệu là a→b, được gọi là đúng nếu:

+ Nếu a và b là hai sự kiện xảy ra trong cùng một tiến trình và a xảy ra trước b.

+ Nếu a và b không thuộc cùng một tiến trình nhưng a là sự kiện gửi một thông điệp đi và b là sự kiện nhận thông điệp đó.

* 1. Ricart & Agrawala

Thuật toán Ricart–Agrawala là thuật toán loại trừ lẫn nhau trong hệ thống phân tán do Glenn Ricart và Ashok Agrawala đề xuất. Thuật toán này là phần mở rộng và tối ưu hóa của Thuật toán loại trừ lẫn nhau phân tán của Lamport. Giống như Thuật toán của Lamport, nó cũng tuân theo cách tiếp cận dựa trên sự cho phép để đảm bảo loại trừ lẫn nhau.

Trong thuật toán này:

* Hai loại thông báo (YÊU CẦU và TRẢ LỜI) được sử dụng và các kênh liên lạc được cho là tuân theo thứ tự FIFO.
* Một trang web gửi một thông báo YÊU CẦU tới tất cả các trang web khác để xin phép họ vào phần quan trọng.
* Một trang web gửi một tin nhắn TRẢ LỜI đến trang web khác để cho phép nó vào phần quan trọng.
* Dấu thời gian được cung cấp cho từng yêu cầu phần quan trọng bằng cách sử dụng đồng hồ logic của Lamport.
* Dấu thời gian được sử dụng để xác định mức độ ưu tiên của các yêu cầu phần quan trọng. Dấu thời gian nhỏ hơn được ưu tiên cao hơn dấu thời gian lớn hơn. Việc thực hiện yêu cầu phần quan trọng luôn theo thứ tự dấu thời gian của chúng.
  1. Loại trừ tương hỗ
* Loại trừ tương hỗ (Mutual Exclusion): là quá trình truy cập đồng thời của các tiến trình với một tài nguyên hoặc dữ liệu được chia sẻ, thực hiện theo cách loại trừ lẫn nhau - Trong hệ thống phân tán không có các biến chia sẻ có thể sử dụng để thực hiện loại trừ lẫn nhau.
* Sự tương tranh và cộng tác giữa các tiến trình. Có thể xảy ra trường hợp cùng một lúc có nhiều tiến trình cùng truy nhập đến một tài nguyên dẫn đến sự xung đột hoặc một thao tác nào đó không được thực thi trọn vẹn. Để tránh điều này giải pháp là cho phép xử lý truy cập theo kiểu lọai trừ lẫn nhau.
  1. Giao thức kết nối
* Giao thức mạng bao gồm trong đó có sự liên kết giữa máy tính, router, máy chủ và các thiết bị hỗ trợ mạng khác khi chúng muốn giao tiếp với nhau. Để đảm bảo quá trình giao tiếp dữ liệu/mạng diễn ra suôn sẻ thì các giao thức mạng luôn phải được xác nhận và cài đặt bởi người gửi và người nhận.
* [Transmission Control Protocol (TCP)](https://www.totolink.vn/article/149-mo-hinh-tcp-ip-la-gi-chuc-nang-cua-cac-tang-trong-mo-hinh-tcp-ip.html) là giao thức điều khiển truyền vận. Chúng là giao thức cốt lõi của Internet Protocol Suite (bộ giao thức liên mạng). Với nhiệm vụ thực thi mạng, bổ sung cho Internet Protocol. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự.

TCP còn cung cấp một phương thức phân phối đáng tin cậy một luồng octet. Là luồng khối dữ liệu có kích thước 8 bit, TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng khác nhau.

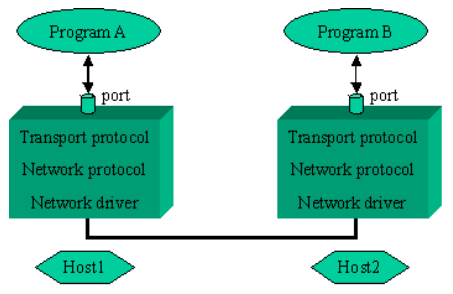
# CHƯƠNG 2: CÔNG NGHỆ ỨNG DỤNG VÀO ĐỒ ÁN

# 2.1. Lập trình Socket trong Java

* Lập trình socket đề cập đến giao tiếp giữa hai nút hoặc ứng dụng chạy trên JRE khác nhau (Java Runtime Environment). Chúng ta sử dụng lập trình socket trong Java để kết nối chương trình máy khách với chương trình máy chủ hoặc đơn giản hơn, kết nối máy khách và máy chủ. Lập trình socket trong Java có thể là hướng kết nối hoặc không kết nối. Gói java.net được sử dụng để lập trình socket trong Java. Có hai lớp trong gói này là lớp Socket và lớp ServerSocket.
* Lớp java.net.Socket trong Java: Sử dụng lớp Socket để tạo các socket cho máy khách và máy chủ. Lớp java.net.Socket đại diện cho socket. Cả chương trình máy khách và máy chủ đều sử dụng ổ cắm này để giao tiếp với nhau. Có một số phương thức trong lớp Socket.
* Lớp java.net.SeverSocket trong Java: Khi sử dụng lớp này tạo một socket máy chủ để lấy một cổng của máy chủ và nghe các yêu cầu của máy khách. Đối tượng của đối tượng lớp ServerSocket giúp thiết lập giao tiếp với các máy khách.

# 2.2. TCP Socket

* SocKet là một giao diện lập trình (API – Application Program Interface) ứng dụng mạng thông qua giao diện này có thể lập trình điều khiển việc truyền thông giữa 2 máy sử dụng các giao thức mức thấp như TCP, UDP. Có thể tưởng tượng nó như một thiết bị truyền thông 2 chiều tương tự như tệp tin, chúng ta gửi/nhận dữ liệu giữa 2 máy, tương tự như việc đọc ghi trên tệp tin.



Với mô hình khách chủ TCP

Diagram

Description automatically generated

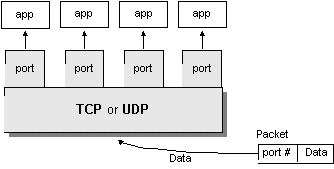
# 2.3. Port Number



Để có thể thực hiện các cuộc giao tiếp, một trong hai quá trình phải công bố số hiệu cổng (port number) của socket mà mình sử dụng. Mỗi cổng giao tiếp thể hiện một địa chỉ xác định trong hệ thống. Khi quá trình được gán một số hiệu cổng, nó có thể nhận dữ liệu gởi đến cổng này từ các quá trình khác. Quá trình còn lại cũng được yêu cầu tạo ra một socket.

Số cổng (port number) được sử dụng để xác định duy nhất các ứng dụng khác nhau. Nó hoạt động như một điểm kết cuối giao tiếp giữa các ứng dụng.

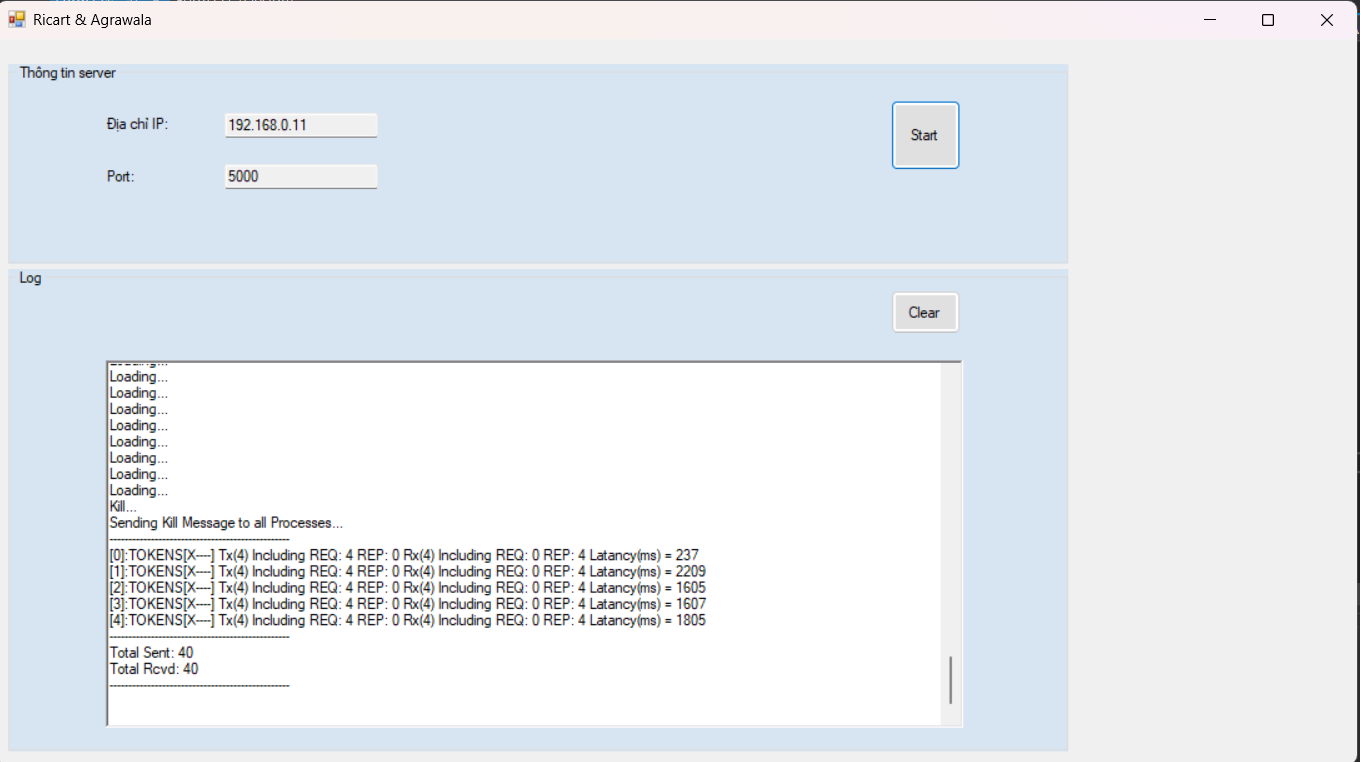
Số cổng được kết hợp với địa chỉ IP để giao tiếp giữa hai ứng dụng.



# CHƯƠNG 3: CHẠY CHƯƠNG TRÌNH

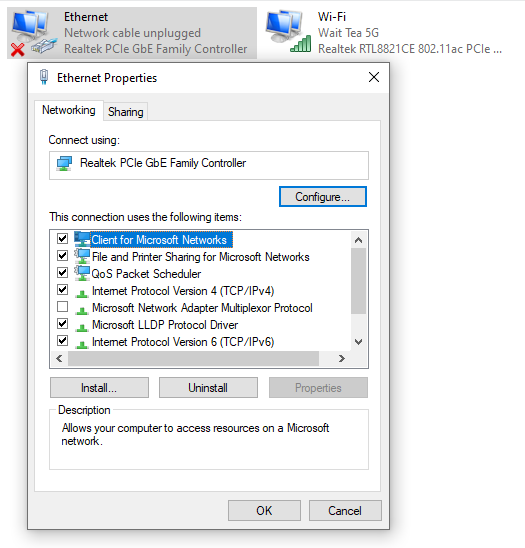
# 3.1. Thành phần

* Giao diện
* Địa chỉ IP: Gồm địa chỉ ip của những máy kết nối với nhau thông Switch.
* Port: Mỗi máy sẽ sử dụng mỗi port khác nhau.
* Nút Start: Dùng để bắt đầu khởi động kết nối giữa các máy với nhau.
* Nút Clear: Reset nội dung trong đã được chạy trong log.
* Log: Hiển thị output chương trình.



# 3.2. Mô tả quá trình chạy chương trình

* Đầu tiên ta cài đặt và tạo liên kết giữa các máy lại với nhau: Vào Control Panel 🡪 All Control Panel Items 🡪 Network and Sharing Center 🡪 Change adapter settings 🡪 Ethernet



Graphical user interface, application

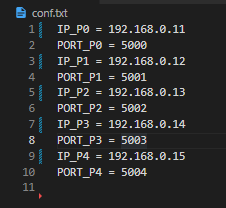
Description automatically generated

* Khi các máy đã kết nối được với nhau, ta có thể ping đến và kiểm tra xem đã thành công (Nếu không được thì ta mở tường lửa và kiểm tra lại đã thành công hay chưa)

Text

Description automatically generated

* Điền thông tin vào file conf.txt có chứa thông tin IP và Port đối với mỗi máy cần chạy bài.



* Mỗi khi một client bấm Connect kết nối đến cùng 1 tài nguyên thì những máy còn lại sẽ hiển thị thông tin máy đã vào cùng tài nguyên và ngược lại với những máy còn lại.

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

* Quá trình gửi và nhận, tính tổng cứ tiếp diễn cho đến khi nào các máy của chúng ta đều dừng lại. Khi đó chương trình chúng ta sẽ kết thúc.

Table

Description automatically generated with medium confidence