**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH**

**Đề tài: Xây dựng chương trình giám sát việc sử dụng máy tính tại trường Đại học Bách Khoa trên môi trường mạng LAN**

**SINH VIÊN THỰC HIỆN:**

**1.Lê Hữu Hưng LỚP: 20TCLC-KHDL NHÓM: 20NH15A**

**2.Hoàng Nguyên Bách LỚP: 20TCLC-KHDL NHÓM: 20NH15A**

**GIẢNG VIÊN HƯỚNG DẪN:**

**ThS. Nguyễn Văn Nguyên**

**Đà Nẵng, 12/ 2022**

**MỤC LỤC**

[1](#_Toc122129647)

[LỜI MỞ ĐẦU 3](#_Toc122129648)

[DANH SÁCH HÌNH VẼ 4](#_Toc122129649)

[DANH SÁCH CÁC TỪ VIẾT TẮT 5](#_Toc122129650)

[GIỚI THIỆU ĐỀ TÀI 6](#_Toc122129651)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 8](#_Toc122129652)

[1.1. Tổng quan về lập trình mạng 8](#_Toc122129653)

[1.1.1. Họ giao thức TCP/IP 8](#_Toc122129654)

[1.1.2. So sánh 2 giao thức TCP và UDP 9](#_Toc122129655)

[1.1.3. Địa chỉ IP 10](#_Toc122129656)

[1.2. Mô hình truyền tin Socket 12](#_Toc122129657)

[1.3. Mô hình Client – Server 14](#_Toc122129658)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH 16](#_Toc122129659)

[2.1. Khảo sát thực trạng 16](#_Toc122129661)

[2.1.1. Phân tích nhu cầu thực tiễn 16](#_Toc122129662)

[2.1.2. Yêu cầu đề ra 16](#_Toc122129663)

[2.2. Thiết kế chương trình 16](#_Toc122129664)

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 21](#_Toc122129665)

[3.1. Kết quả chương trình 21](#_Toc122129667)

[3.1.1. Giao diện phía Client 21](#_Toc122129668)

[3.1.2. Giao diện phía Server: 21](#_Toc122129669)

[3.2. Đánh giá chương trình 23](#_Toc122129670)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24](#_Toc122129671)

[TÀI LIỆU THAM KHẢO 25](#_Toc122129672)

LỜI MỞ ĐẦU

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong cuộc sống, nó có thể tạo ra những bước đột phá mạnh mẽ. Điều này kéo theo sự phát triển của công nghệ mạng máy tính và sự phát triển của Internet.

Và bên cạnh đó, các hình thức phá hoại mạng cũng ngày càng tinh vi và phức tạp hơn. Do đó, yêu cầu cấp thiết đối với mỗi hệ thống là phải được bảo mật những thông tin của mình. Xuất phát từ những thực tế đó, đã có nhiều công nghệ liên quan đến bảo mật hệ thống mạng máy tính xuất hiện, việc nắm bắt các công nghệ này là rất cần thiết. Chính vì vậy, đó là lý do để chúng em thực hiện đề tài: “Xây dựng chương trình giám sát việc sử dụng máy tính tại trường Đại học Bách Khoa – Đại học Đà Nẵng trên môi trường mạng LAN”.

Chúng em xin chân thành cảm ơn thầy Nguyễn Văn Nguyên đã hỗ trợ cho chúng em những kiến thức cần thiết cũng như tinh thần học tập độc lập, tìm hiểu và sáng tạo trong đồ án môn học này. Chúng em cũng xin cảm ơn những thầy cô bộ môn Khoa Công nghệ thông tin, Đại học Bách Khoa – Đại học Đà Nẵng.

Trong quá trình thực hiện đồ án, mặc dù đã cố gắng hết sức song do thời gian và khả năng còn hạn chế nên chúng em không thể tránh khỏi những thiếu sót. Vì vậy, chúng em mong nhận được sự thông cảm, chỉ bảo và giúp đỡ của thầy cô và các bạn.

# 

DANH SÁCH HÌNH VẼ

Hình 1: Mô hình OSI và TCP/IP….….….….….….….….….….….…….……8

Hình 2: Các đặc trưng của 2 giao thức UDP và TCP…………………………10

Hình 3: Tổ chức địa chỉ IP……………………………………………………11

Hình 4: Các bước của một quá trình truyền tin……………………………….13

Hình 5: Mô hình Client-Server………………………………………………..14

Hình 6: Hoạt động mô hình Client-Server…………………………………….15

Hình 7: Giao diện Client khi chưa kết nối đến Server………………………...21

Hình 8: Giao diện Client khi kết nối thành công đến Server………………….21

Hình 9: Giao diện chính của Server…………………………………………...21

Hình 10: Giao diện của chức năng Scan Network…………………………….22

Hình 11: Giao diện của chức năng Print Screen………………………………22

Hình 12: Giao diện sau khi chụp ảnh màn hình máy Client.………………….22

DANH SÁCH CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Ký hiệu, viết tắt** | **Tên đầy đủ** |
| LAN | Local Network Area (Mạng cục bộ) |
| IP | Internet Protocol - giao thức Internet |
| WLAN | Wireless LAN |
| CPU | **Central Prossesing Unit** |
| SERVER | Máy chủ |
| CLIENT | Máy khách |
| NIC | Network Interface Card |
| WAN | Wide Area Network |
| UDP | User Datagram Protocol hay User Define Protocol |
| TCP | Transport Control Protocol - Giao thức điều khiển truyền vận |
| IDE | Intergrated Development Enviroment |

GIỚI THIỆU ĐỀ TÀI

* Mục đích của đề tài

Chương trình “Giám sát việc sử dụng máy tính tại trường Đại học Bách Khoa trên môi trường mạng LAN” được xây dựng nhằm hỗ trợ, giám sát và trao đổi dữ liệu trong cùng mạng LAN của trường Đại học Bách Khoa mà không cần sử dụng các phần mềm bên ngoài để đảm bảo dữ liệu không bị truyền ra bên ngoài.

* Lý do chọn đề tài

Việc nối máy tính thành mạng từ lâu đã trở thành một nhu cầu khách quan vì: Có rất nhiều công việc về bản chất là phân tán hoặc về thông tin, hoặc về xử lý hoặc cả hai đòi hỏi có sự kết hợp truyền thông với xử lý hoặc sử dụng phương tiện từ xa. Chia sẻ các tài nguyên trên mạng cho nhiều người sử dụng tại một thời điểm (ổ cứng, máy in, ổ CD ROM . . .) Nhu cầu liên lạc, trao đổi thông tin nhờ phương tiện máy tính. Các ứng dụng phần mềm đòi hòi tại một thời điểm cần có nhiều người sử dụng, truy cập vào cùng một cơ sở dữ liệu.

Hiện nay, mạng LAN đã có những tiến bộ vượt bậc và ngày càng phổ biến hơn trong đời sống sinh hoạt. Điều này làm cho nhu cầu liên lạc và trao đổi thông tin thông qua mạng LAN ngày càng lớn hơn, nhất là trong các trường học, dùng để hỗ trợ và trao đổi Database mà không cần sử dụng phần mềm bên ngoài nào khác. Chính vì vậy, chương trình “*Giám sát việc sử dụng máy tính tại trường Đại học Bách Khoa trên môi trường mạng LAN*” được xây dựng để đáp ứng phần nào những nhu cầu cấp thiết đó.

Trên thực tế có rất nhiều ứng dụng trên mạng đã được phổ biến rộng rãi cho việc hỗ trợ này, nhưng người dùng phải Download rất nhiều phần mềm khác nhau về. Người ta sử dụng các chương trình Chat, Teamview, Ultraview, Sky… với mục đích trao đổi thông tin như trò chuyện, bàn bạc công việc và gửi dữ liệu gián tiếp qua môi trường Internet - một cách đơn giản để kết nối mọi người lại với nhau.

Khi xây dựng một chương trình này cần phải đảm bảo các yêu cầu về kết nối, dữ liệu trong quá trình truyền đi phải đảm bảo toàn vẹn và an toàn. Đặc biệt chương trình này tích hợp các chức năng có thể chat, theo dõi và quản lý các máy tính trong mạng LAN đó. Là một ứng dụng mạng nên các chương trình hỗ

trợ, giám sát này hoạt động dựa trên mô hình Client – Server hoặc Point to Point và kết nối bằng một trong 2 giao thức TCP hoặc UDP. Trên cơ sở đó, chúng em đã xây dựng chương trình hoạt động theo mô hình Client – Server và sử dụng giao thức kết nối TCP.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## Tổng quan về lập trình mạng

### Họ giao thức TCP/IP

IP là một họ giao thức để cung cấp phương tiện truyền thông liên mạng và nó được cấu trúc theo kiểu phân cấp.

Khác với mô hình OSI tầng liên mạng sử dụng giao thức kết nối mạng "không liên kết" (connectionless) IP, tạo thành hạt nhân hoạt động của Internet. Cùng với các thuật toán định tuyến RIP, OSPF, BGP, tầng liên mạng IP cho phép kết nối một cách mềm dẻo và linh hoạt các loại mạng "vật lý" khác nhau như: Ethernet, Token Ring , X.25...

Giao thức trao đổi dữ liệu "có liên kết" (connection - oriented) TCP được sử dụng ở tầng vận chuyển để đảm bảo tính chính xác và tin cậy việc trao đổi dữ liệu dựa trên kiến trúc kết nối "không liên kết" ở tầng liên mạng IP.

Các giao thức hỗ trợ ứng dụng phổ biến như truy nhập từ xa (telnet), chuyển tệp (FTP), dịch vụ World Wide Web (HTTP), thư điện tử (SMTP), dịch vụ tên miền (DNS) ngày càng được cài đặt phổ biến như những bộ phận cấu thành của các hệ điều hành thông dụng như UNIX (và các hệ điều hành chuyên dụng cùng họ của các nhà cung cấp thiết bị tính toán như AIX của IBM, SINIX của Siemens, Digital UNIX của DEC), Windows9x/NT, NovellNetware,...



*Hình 1: Mô hình OSI và TCP/IP*

### So sánh 2 giao thức TCP và UDP

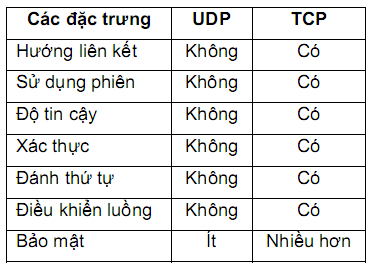
UDP (User Datagram Protocol) là một trong những giao thức cốt lõi của giao thức TCP/IP. Dùng UDP, không cần phải thiết lập liên kết, chương trình trên mạng máy tính có thể gởi những dữ liệu ngắn được gọi là datagram tới máy khác. UDP không cung cấp sự tin cậy và thứ tự truyền nhận mà TCP làm. Các gói dữ liệu có thể đến không đúng thứ tự hoặc bị mất mà không có thông báo. Tuy nhiên UDP nhanh và hiệu quả hơn đối với các mục tiêu như kích thước nhỏ và yêu cầu khắt khe về thời gian. Do bản chất không trạng thái của nó nên nó hữu dụng đối với việc trả lời các truy vấn nhỏ với số lượng lớn người yêu cầu.

TCP (Transmission Control Protocol) là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy, đúng thứ tự và nó còn cung cấp chức năng nhằm kiểm tra tính chính xác của dữ liệu khi đến và bao gồm cả việc gửi lại dữ liệu khi có lỗi xảy ra. TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ.

**Điểm khác nhau cơ bản của UDP và TCP:**

Các header của TCP và UDP khác nhau ở kích thước (20 và 8 byte) nguyên nhân chủ yếu là do TCP phải hộ trợ nhiều chức năng hữu ích hơn (như khả năng khôi phục lỗi). UDP dùng ít byte hơn cho phần header và yêu cầu xử lý từ host ít hơn.

|  |  |
| --- | --- |
| TCP | UDP |
| Đảm bảo việc cung cấp dữ liệu | Không đảm bảo việc chuyển dữ liệu |
| Theo dõi các gói tin | Các gói tin có thể mất |
| Tốc độ truyền thấp hơn UDP | Tốc độ truyền cao, VoIP truyền tốt qua UDP |

**

*Hình 2: Các đặc trưng giữa 2 giao thức TCP và UDP*

### Địa chỉ IP

Địa chỉ IP (IP là viết tắt của từ tiếng Anh: Internet Protocol - giao thức Internet) là một địa chỉ đơn nhất mà những thiết bị điện tử hiện nay đang sử dụng để nhận diện và liên lạc với nhau trên mạng máy tính bằng cách sử dụng giao thức Internet.

Nhiệm vụ chính của giao thức IP là cung cấp khả năng kết nối các mạng con thành liên kết mạng để truyền dữ liệu, vai trò của IP là vai trò của giao thức tầng mạng trong mô hình OSI. Giao thức IP là một giao thức kiểu không liên kết (connectionless) có nghĩa là không cần có giai đoạn thiết lập liên kết trước khi truyền dữ liệu.

**Tổng quan về địa chỉ IP**

- Là địa chỉ có cấu trúc, được chia làm hai hoặc ba phần là: Net\_id và Host\_id hoặc Net\_id và Subnet\_id và Host\_id

- Là một con số có kích thước 32 bit. Khi trình bày, người ta chia con số 32 bit thành bốn phần, mỗi phần có kích thước 8 bit, gọi là octet hoặc byte.

- Có các cách trình bày sau:

• Ký pháp thập phân có dấu chấm (dotted-decimal notation).

Ví dụ: 172.16.30.56

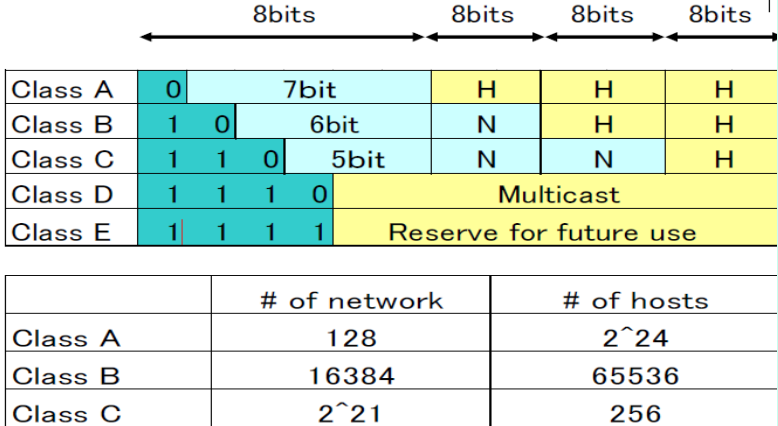
• Ký pháp nhị phân.

Ví dụ: 10101100 00010000 00011110 00111000

• Ký pháp thập lục phân.

Ví dụ: AC 10 1E 38

* Không gian địa chỉ IP (gồm 232 địa chỉ) được chia thành nhiều lớp (class) để dễ quản lý. Đó là các lớp: A, B, C, D và E; trong đó các lớp A, B và C được triển khai để đặt cho các host trên mạng Internet; lớp D dung cho các nhóm multicast; còn lớp E phục vụ chi mục đích nghiên cứu.



*Hình 3: Tổ chức địa chỉ IP*

* Địa chỉ IP còn được gọi là địa chỉ logical, trong khi địa chỉ MAC còn gọi là địa chỉ vật lý (hay địa chỉ physical).

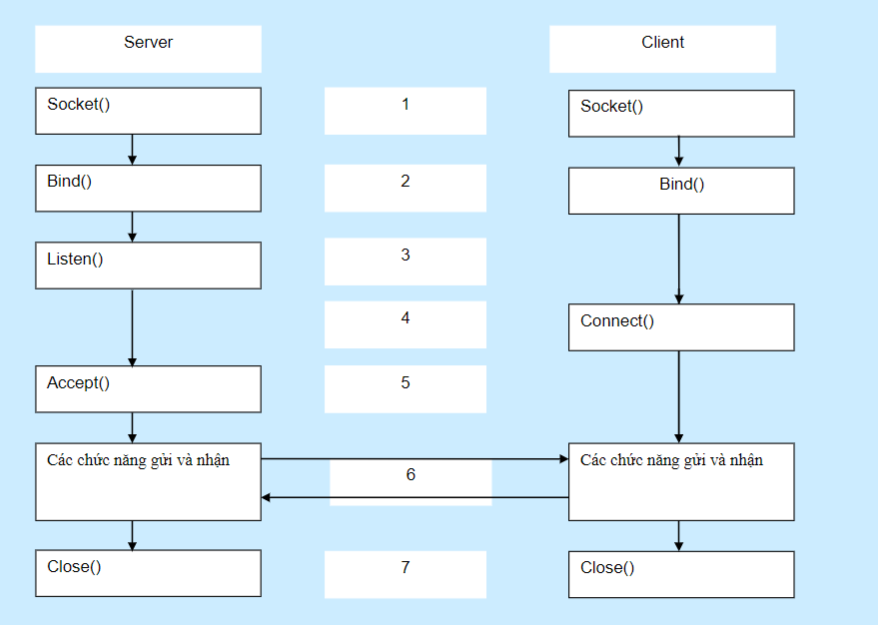
**Các bước hoạt động của giao thức IP**

* Đối với thực thể IP ở máy nguồn, khi nhận được một yêu cầu gửi từ tầng trên, nó thực hiện các bước sau đây:
* Tạo một IP datagram dựa trên tham số nhận được.
* Tính checksum và ghép vào header của gói tin.
* Ra quyết định chọn đường: hoặc là trạm đích nằm trên cùng mạng hoặc một gateway sẽ được chọn cho chặng tiếp theo.
* Chuyển gói tin xuống tầng dưới để truyền qua mạng.
* Đối với router, khi nhận được một gói tin đi qua, nó thực hiện các động tác sau:
* Tính chesksum, nếu sai thì loại bỏ gói tin.
* Giảm giá trị tham số Time - to Live, nếu thời gian đã hết thì loại bỏ gói tin.
* Ra quyết định chọn đường.
* Phân đoạn gói tin, nếu cần.
* Kiến tạo lại IP header, bao gồm giá trị mới của các vùng Time - to -Live, Fragmentation và Checksum.
* Chuyển datagram xuống tầng dưới để chuyển qua mạng.
* Cuối cùng khi một datagram nhận bởi một thực thể IP ở trạm đích, nó sẽ thực hiện bởi các công việc sau:
* Tính checksum. Nếu sai thì loại bỏ gói tin.
* Tập hợp các đoạn của gói tin (nếu có phân đoạn).
* Chuyển dữ liệu và các tham số điều khiển lên tầng trên.

## Mô hình truyền tin Socket

Sockets cung cấp một interface để lập trình mạng tại tầng Transport. Một socket là một end-point của một liên kết giữa hai ứng dụng. Ngày nay, Socket được hỗ trợ trong hầu hết các hệ điều hành như MS Windows (WinSock), Linux và được sử dụng trong nhiều ngôn ngữ lập trình khác nhau: như C, C++, Java, Visual Basic, C#, . . .Windows Socket Application Programming Interface (Winsock API) là một thư viện các hàm socket

* Một socket có thể thực hiện bảy thao tác cơ bản:
* Kết nối với một máy ở xa (để gửi và nhận dữ liệu)
* Gửi dữ liệu
* Nhận dữ liệu
* Ngắt liên kết
* Gán cổng
* Nghe dữ liệu đến
* Chấp nhận liên kết từ các máy ở xa trên cổng được gán
* Lớp Socket của Java được sử dụng cả client và server.
* Có các phương thức tương ứng với bốn thao tác đầu tiên.
* Ba thao tác cuối chỉ cần cho server để chờ các client liên kết với chúng.
* Các thao tác này được cài đặt bởi lớp ServerSocket.



*Hình 4: Các bước của một quá trình truyền tin*

* Các socket cho client thường được sử dụng theo mô hình sau :
* Sử dụng hàm Socket() để tạo một socket mới.
* Socket cố gắng liên kết với một host ở xa.
* Mỗi khi liên kết được thiết lập, các host ở xa nhận các luồng vào và ra từ socket, và sử dụng các luồng này để gửi dữ liệu cho nhau.
* Khi việc truyền dữ liệu hoàn thành, một hoặc cả hai phía ngắt liên kết.

**Cài đặt chương trình**

Cài đặt chương trình client:

- Bước 1: Tạo một đối tượng Socket

- Bước 2: Tạo một luồng xuất để có thể gửi thông tin tới Socket

- Bước 3: Tạo một luồng nhập để đọc thông tin đáp ứng từ Server

- Bước 4: Thực hiện cá thao tác vào ra với luồng nhập và luồng xuất

- Bước 5: Đóng Socket khi hoàn thành quá trình truyền tin

Cài đặt chương trình Server:

- Bước 1: tạo một đối tượng ServerSocket

- Bước 2: Tạo một đối tượng Socket bằng cách chấp nhận liên kết từ yêu cầu liên kết của client. Sau khi chấp nhận liên kết, phương thức accept() trả về đối tượng Socket thể hiện liên kết giữa Client và Server

- Bước 3: Tạo một luồng nhập để đọc dữ liệu từ Client

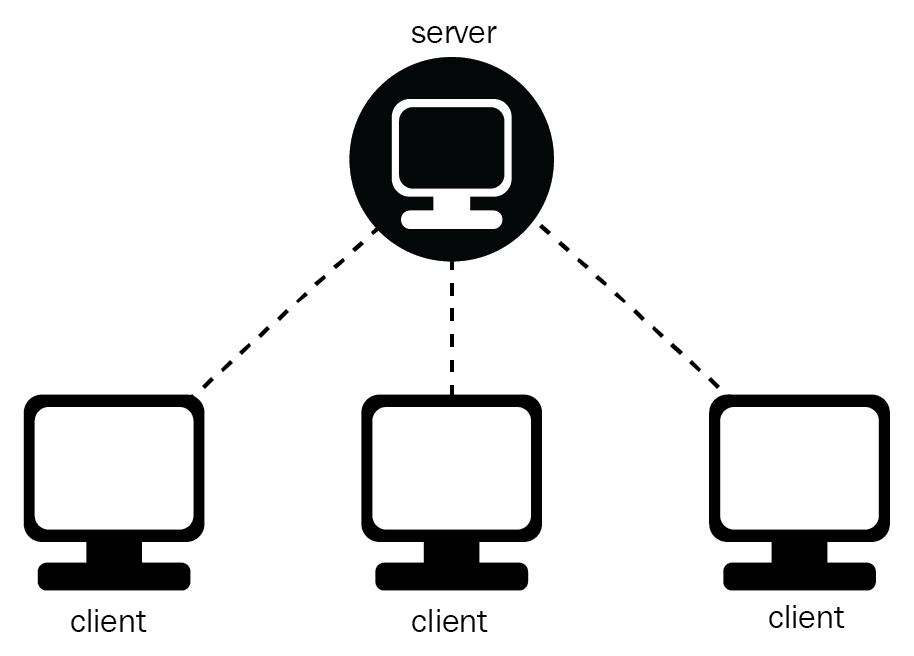
- Bước 4: Tạo một luồng xuất để trả dữ liệu lại cho Client

- Bước 5: Thực hiện cá thao tác vào ra với luồng nhập và luồng xuất

- Bước 6: Đóng Socket sau khi đã truyền tin xong. Việc đóng Socket cũng đồng nghĩa với việc đóng các luồng

## Mô hình Client – Server

Mô hình Client Server là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.



*Hình 5: Mô hình client-server*

+ Mô hình client/server cung cấp một cách tiếp cận tổng quát để chia sẻ tài nguyên trong các hệ thống phân tán.

+ Cả tiến trình client và tiến trình server đều có thể chạy trên cùng một máy tính.

+ Một tiến trình server có thể sử dụng dịch vụ của một server khác.

+ Mô hình truyền tin client/server hướng tới việc cung cấp dịch vụ.

Quá trình trao đổi dữ liệu bao gồm:

Bước 1: Truyền một yêu cầu từ tiến trình client tới tiến trình server

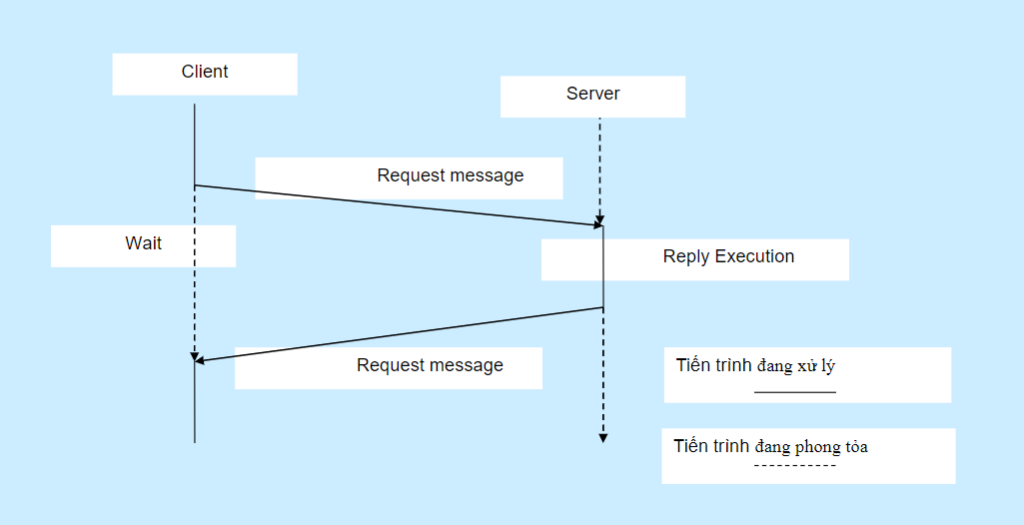
Bước 2: Yêu cầu được server xử lý

Bước 3: Truyền đáp ứng cho client

Mô hình client/server cung cấp một cách tiếp cận tổng quát để chia sẻ tài nguyên trong các hệ thống phân tán. Mô hình này có thể được cài đặt bằng rất nhiều môi trường phần cứng và phần mềm khác nhau. Các máy tính được sử dụng để chạy các tiến trình client/server có nhiều kiểu khác nhau và không cần thiết phải phân biệt giữa chúng; cả tiến trình client và tiến trình server đều có thể chạy trên cùng một máy tính. Một tiến trình server có thể sử dụng dịch vụ của một server khác.

Mô hình truyền tin này liên quan đến việc truyền hai thông điệp và một dạng đồng bộ hóa cụ thể giữa client và server. Tiến trình server phải nhận thức được thông điệp được yêu cầu ở bước một ngay khi nó đến và hành động phát ra yêu cầu trong client phải được tạm dừng (bị phong tỏa) và buộc tiến trình client ở trạng thái chờ cho tớ khi nó nhận được đáp ứng do server gửi về ở bước ba.

Mô hình client/server thường được cài đặt dựa trên các thao tác cơ bản là gửi (send) và nhận (receive).



*Hình 6: Hoạt động mô hình client-server*

Quá trình giao tiếp client và server có thể diễn ra theo một trong hai chế độ: bị phong tỏa (blocked) và không bị phong tỏa (non-blocked):

* Chế độ bị phong tỏa (blocked):

+ Trong chế độ bị phong tỏa, khi tiến trình client/server phát ra lệnh gửi dữ liệu (send), việc thực thi của tiến trình sẽ bị tạm ngừng cho tới khi tiến trình nhận phát ra lệnh nhận dữ liệu (receive).

+ Tương tự đối với tiến trình nhận dữ liệu, nếu tiến trình nào đó (client \*\*\* server) phát ra lệnh nhận dữ liệu, mà tại thời điểm đó chưa có dữ liệu gửi tới thì việc thực thi của tiến trình cũng sẽ bị tạm ngừng cho tới khi có dữ liệu gửi tới.

* Chế độ không bị phong tỏa (non-blocked):

+ Trong chế độ này, khi tiến trình client hay server phát ra lệnh gửi dữ liệu thực sự, việc thực thi của tiến trình vẫn được tiến hành mà không quan tâm đến việc có tiến trình nào phát ra lệnh nhận dữ liệu đó hay không.

+ Tương tự cho trường hợp nhận dữ liệu, khi tiến trình phát ra lệnh nhận dữ liệu, nó sẽ nhận dữ liệu hiện có, việc thực thi của tiến trình vẫn được tiến hành

mà không quan tâm đến việc có tiến trình nào phát ra lệnh gửi dữ liệu tiếp theo hay không.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ CHƯƠNG TRÌNH



## Khảo sát thực trạng

### Phân tích nhu cầu thực tiễn

Hiện nay, mạng LAN phát triển rất mạnh cả trong trường học, các cơ quan tổ chức và ở cả các hộ gia đình. Chính điều đó kéo theo nhu cầu liên lạc trao đổi thông tin trong mạng LAN cũng phát triển theo. Chính vì vậy, một chương trình phục vụ cho nhu cầu liên lạc, trao đổi thông tin trong mạng LAN là rất cần thiết nhất là lúc Offline.

### Yêu cầu đề ra

Yêu cầu đặt ra là xây dựng chương trình giám sát việc sử dụng máy tính tại trường Đại học Bách khoa – Đại học Đà Nẵng trên môi trường mạng LAN.

* Server có khả năng: chat, gửi thông điệp, quản lý, điều khiển, chụp ảnh đối với Client đã được kết nối.

## Thiết kế chương trình

Chương trình cài đặt gồm ba lớp cơ bản: Client, Server, Modle.

* Lớp Client: bao gồm các lớp về giao diện Client và lớp chụp ảnh màn hình ở máy Client.

+ **Hàm ảnh chụp màn hình của client đến Server**

public void Send\_image() {

try {

cScreenShot = new Capture\_IMG(1.0f);

oos = new ObjectOutputStream(socket.getOutputStream());

// Lấy màn hình mặc định của hệ thống

GraphicsEnvironment gEnv = GraphicsEnvironment.getLocalGraphicsEnvironment();

GraphicsDevice gDev = gEnv.getDefaultScreenDevice();

// Chuẩn bị robot thao tác màn hình

Robot robot = new Robot(gDev);

oos.writeObject(cScreenShot.execute(robot));

oos.flush();

}

catch (Exception ex) {

ex.printStackTrace();

}

}

* Lớp Server: bao gồm các lớp giao diện Server và lớp yêu cầu Scan\_IP máy Client có trong mạng.

+ **Hàm yêu cầu Scan\_IP**

public class Request\_ScanIP {

public static ArrayList<String> Scan(String firstIpInTheNetwork, int numOfIps) {

ExecutorService executorService = Executors.newFixedThreadPool(20);

final String networkId = firstIpInTheNetwork.substring(0, firstIpInTheNetwork.length() - 1);

ArrayList<String> ipsSet = new ArrayList<String>();

AtomicInteger ips = new AtomicInteger(0);

while (ips.get() <= numOfIps) {

String ip = networkId + ips.getAndIncrement();

executorService.submit(() -> {

try {

InetAddress inAddress = InetAddress.getByName(ip);

if (inAddress.isReachable(5000)) {

System.out.println("found ip: " + ip);

ipsSet.add(ip);

}

}

catch (IOException e) {

}

});

}

executorService.shutdown();

try {

executorService.awaitTermination(1, TimeUnit.MINUTES);

}

catch (InterruptedException e) {

System.out.println(e.getMessage());

}

return ipsSet;

}

}

* Lớp Modle: bao gồm các hàm xử lý ảnh chụp màn hình, thông điệp, IP của máy Client có trong mạng.

+ Lớp Handle\_IMG:

**Hàm xử lý ảnh chụp màn hình từ máy Client**

public Object execute(Robot robot) throws IOException {

// Tính thời gian gửi

long time = System.currentTimeMillis();

// Chụp màn hình

Toolkit defaultToolkit = Toolkit.getDefaultToolkit();

Rectangle shotArea = new Rectangle( defaultToolkit.getScreenSize() );

BufferedImage image = robot.createScreenCapture(shotArea);

// Thay đổi độ scale

if (scale != 1.0) {

image = getScaledInstance(image);

}

// Chuyển ảnh sang JPG và bytes để gửi cho server

byte[] bytes = convertToJPG(image);

time = System.currentTimeMillis() - time;

System.out.println("time = " + time);

// Chỉ gửi nếu số byte thực sự thay đổi

// Bằng cách so sánh với số byte trước đó

byte[] prev = previous.get();

if (prev != null && Arrays.equals(bytes, prev))

{

return null;

}

// Thêm số byte ở lần chụp này để so sánh với lần chụp tiếp theo

previous.set(bytes);

return bytes;

}

**Hàm gửi ảnh đến Server**

public void run(){

ObjectOutputStream oos = null;

try{

//Chuẩn bị gửi cho server

oos = new ObjectOutputStream(socket.getOutputStream());

if(rectangle != null){

// Nếu sử dụng cho remote desktop thì truyền rectangle

// để tính tọa độ chuột tương ứng

oos.writeObject(rectangle);

}

}catch(IOException ex){

ex.printStackTrace();

}

Capture\_IMG cScreenShot = new Capture\_IMG(1.0f);

while(true){

// Gửi màn hình đã chụp cho server

try {

System.out.println("Truoc khi goi man hinh");

oos.writeObject(cScreenShot.execute(robot));

oos.flush();

System.out.println("Man hinh da goi");

// Dùng 200ms để dữ liệu kịp chuyển đến

// Trung bình 1/2 RTM = 170~200

Thread.sleep(200);

} catch (Exception ex) {

break;

}

}

}

# CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ



## Kết quả chương trình

### Giao diện phía Client

Graphical user interface, text, application

Description automatically generated

*Hình 7: Giao diện Client khi chưa kết nối đến Server*

Graphical user interface, text, application

Description automatically generated

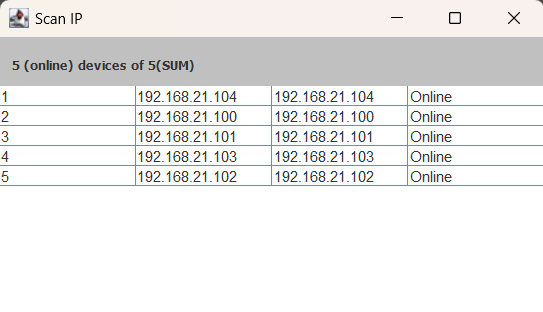
*Hình 8: Giao diện Client khi kết nối thành công đến Server*

### Giao diện phía Server:

Graphical user interface

Description automatically generated

*Hình 9: Giao diện chính của Server*



*Hình 10: Giao diện chức năng Scan Network*

Graphical user interface

Description automatically generated

*Hình 11: Giao diện chức năng Print Screen*

Graphical user interface, text

Description automatically generated

*Hình 12: Giao diện sau khi chụp ảnh màn hình máy Client*

## Đánh giá chương trình

* Chương trình có thể quản lý, hỗ trợ,… với các máy tính trong cùng mạng LAN, đáp ứng nhu cầu sử dụng nội bộ.
* Tốc độ xử lý của chương trình khá ổn.
* Tính năng chụp màn hình làm việc của máy Client khá tốt.
* Có thể sử dụng được cho cả Laptop, PC thông qua cả Wifi và LAN.
* Chương trình có giao diện trực quan thân thiện nên rất dễ sử dụng.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

* Kết quả đạt được

Đề tài “Giám sát việc sử dụng máy tính tại trường Đại học Bách Khoa trên môi trường mạng LAN” đã thực hiện được các nội dung sau:

* Tìm hiểu được cách thức lập trình Socket và lập trình Thread.
* Tìm hiểu được cách thức hoạt động quét IP của các máy User thông qua mạng LAN.
* Tìm hiểu được cách thức chụp màn hình của User trong cùng mạng LAN.

Ưu điểm và nhược điểm của chương trình:

* Ưu điểm:

+ Quản lý và giám sát máy tính khác đơn giản.

+ Dễ update và chỉnh sửa.

+ Giao diện đơn giản phù hợp với nhiều loại người dung và mục đích sử dụng.

* Nhược điểm:

+ Cần cải thiện giao diện nhiều hơn.

+ Tối ưu hóa source code.

* Hướng phát triển

Về cơ bản, đồ án đã đạt được những yêu cầu đặt ra. Tuy nhiên nếu có điều kiện, đồ án sẽ cố gắng phát triển thêm các chức năng sau:

* Hỗ trợ chức năng điều khiển máy tính khác trong cùng mạng LAN.
* Trao đổi dữ liệu và chat giữa các Client trong cùng mạng LAN
* Hỗ trợ chức năng Voice chat và Webcam
* Khóa mở bàn phím, chuột và các chức năng của hệ thống từ xa.
* Kết luận

Trong quá trình thực hiện đồ án môn học, chúng em đã có cơ hội làm quen và xây dựng một phần mềm hoàn chỉnh. Và chúng em đã tích lũy những kinh nghiệm về kiến thức trong công việc cũng như các kinh nghiệm về kỹ năng mềm.

**TÀI LIỆU THAM KHẢO**

**[1]** Eric F Crist & Jan Just Keijser, *“Mastering OpenVPN”,*Packt Publishing Ltd, 2015.

[2] Mai Văn Hà, “*Giáo trình bài giảng học phần Lập trình mạng”*, Tài liệu lưu hành nội bộ.