**Trường Đại Học Bách Khoa Hà Nội**

**Viện Điện Tử Viễn Thông**

====o0o====



Báo Cáo Bài Tập Lớn

Môn: Mạng Máy Tính

Đề tài: Lập Trình Socket Xây Dựng Chương Trình

Truyền File Qua Mạng LAN

Giảng viên hướng dẫn: TS. Nguyễn Tài Hưng

Sinh viên thực hiện:

Nguyễn Xuân Sang 20102071 ĐTVT09-K55

Nguyễn Ngọc Hưng 20101673 ĐTVT02-K55

Đỗ Tuấn Anh 20101076 ĐTVT01-K55

Hà Nội - 12/2014

Mục Lục

[Lời Nói Đầu 3](#_Toc388889965)

[Chương 1. Tìm Hiểu Chung 4](#_Toc388889966)

[1.1 Cơ bản mạng máy tính 4](#_Toc388889967)

[1.2 Tổng quan về C# 7](#_Toc388889968)

[1.3 Lập trình socket 8](#_Toc388889969)

[Chương 2: Xây Dựng Chương Trình Ứng Dụng 12](#_Toc388889970)

[2.1. Giới thiệu 12](#_Toc388889971)

[2.3 Cơ chế hoạt động của chương trình 12](#_Toc388889972)

[2.4 Giao diện chương trình 13](#_Toc388889973)

[Chương 3: Kết Luận 14](#_Toc388889974)

# LỜI NÓI ĐẦU

Thế kỷ 21 được mệnh danh là thế kỷ của công nghệ thông tin, với sự bùng nổ mạnh mẻ về khoa học công nghệ. Đây là kỷ nguyên của nền văn minh dựa trên cơ sở công nghiệp trí tuệ . Ngày nay, tin học đã trở thành một môn khoa học quan trọng trên thế giới.

Sự phát triền mạnh mẽ như vậy thì công việc lập trình các ứng dụng nhằm phục vụ nhu câù, lợi ích của con người trở nên cấp thiết. Mảy tỉnh đã trở thành công cụ đăc lực và không thê thiếu của con người. Các tổ chức, công ty hay các cơ quan cần phải xây dựng luận văn - bảo cáo – tiểu luận chuyên ngành xây dựng hệ thống mạng máy tính cho riêng mình để trao đổi dữ liệu giữa các bộ phận. Dữ liệu được truyền đi trên mạng phải đảm bảo: dữ liệu được chuyên tới đích nhanh chóng và chính xác. Hâù hết dữ liệu được truyền qua mạng là truyền dưới dạng file.

Nhằm tìm hiểu thấu đáo một trong số các phương pháp truyền file qua mạng em chọn để tài "Lập Trình Socket Xây Dựng Chương Trình Truyền File Qua Mạng Lan” Với lập trình socket sẽ bắt buộc các máy đó phải được nối mạng với nhau. Ta đã thấy các máy muốn trao đổi dữ liệu qua mạng, chúng sẽ tạo ra ở mỗi phía một socket và trao đổi dữ liệu bằng cách đọc/ghi từ socket. Khi một chương trình tạo ra một socket, một định danh dạng số (định danh dạng số này còn được gọi là số hiệu cổng) sẽ được gán cho socket. Việc gán số hiệu cổng này cho socket có thể được thực hiện bởi chương trình hoặc hệ điều hành. Trong mỗi gói tin mà socket gửi đi có chứa hai thông tin để xác định đích đến của gói tin:

Một địa chỉ mạng đê xác định hệ thống sẽ nhận gói tin.

Một sổ định danh cổng để nói cho hệ thống đích biết socket nào trên nó sẽ nhận dữ liệu.

Mục tiêu của bài tập lớn là tìm hiểu về mạng mảy tính và kĩ thật lập trình socket với ngôn ngữ lập trình là c#, trên cơ sở đủ xây dựng ứng dụng truyền file trên mạng cục bộ. Bài tập lớn sẽ xây dựng thử nghiêm một hệ thống cho phép trao đổi dữ liệu dưới dạng truyền file, tương tác điểm – điểm trên mạng LAN

# CHƯƠNG 1. TÌM HIỂU CHUNG

## 1.1. Cơ bản mạng máy tính

1.1.1. Định nghĩa Mạng máy tính

Mạng máy tính là một tập hợp các máy tính được nối với nhau bởi đường truyền theo một cấu trúc nào đó và thông qua đó các máy tính trao đổi thông tin qua lại cho nhau.

Đường truyền là hệ thống các thiết bị truyền dẫn có dây hay không dây dùng để chuyển các tín hiệu điện tử từ máy tính này đến máy tính khác. Các tín hiệu điện tử đó biểu thị các giá trị dữ liệu dưới dạng các xung nhị phân (on - off). Tất cả các tín hiệu được truyền giữa các máy tính đều thuộc một dạng sóng điện từ. Tùy theo tần số của sóng điện từ có thể dùng các đường truyền vật lý khác nhau để truyền các tín hiệu. Ở đây đường truyền được kết nối có thể là dây cáp đồng trục, cáp xoắn, cáp quang, dây điện thoại, sóng vô tuyến ... Các đường truyền dữ liệu tạo nên cấu trúc của mạng. Hai khái niệm đường truyền và cấu trúc là những đặc trưng cơ bản của mạng máy tính.

1.1.2. Nhu cầu phát triển mạng máy tính

Ngày nay, khi máy tính được sử dụng một cách rộng rãi và số lượng máy tính trong một văn phòng hay cơ quan được tăng lên nhanh chóng thì việc kết nối chúng trở nên vô cùng cần thiết và sẽ mang lại nhiều hiệu quả cho người sử dụng.

Với một lượng lớn về thông tin, nhu cầu xử lý thông tin ngày càng cao, mạng máy tính đã trở nên quá quen thuộc đối với chúng ta trong mọi lĩnh vực như: khoa học, quân sự, quốc phòng, thương mại, dịch vụ, giáo dục...

Người ta thấy được việc kết nối các máy tính thành mạng cho chúng ta những khả năng mới to lớn như:

*- Sử dụng chung tài nguyên:* những tài nguyên (như thiết bị, chương trình, dữ liệu) khi được trở thành các tài nguyên chung thì mọi thành viên của mạng đều có thể tiếp cận được mà không quan tâm tới những tài nguyên đó ở đâu.

*- Tăng độ tin cậy của hệ thống:* người ta có thể dễ dàng bảo trì máy móc, lưu trữ (backup) các dữ liệu chung và khi có trục trặc trong hệ thống thì chúng có thể được khôi phục nhanh chóng. Trong trường hợp có trục trặc trên một trạm làm việc thì người ta cũng có thể sử dụng những trạm khác thay thế.

*- Nâng cao chất lượng và hiệu quả khai thác thông tin:* khi thông tin có thể được sử dụng chung thì nó mang lại cho người sử dụng khả năng tổ chức lại các công việc với những thay đổi về chất như:

+ Ðáp ứng những nhu cầu của hệ thống ứng dụng kinh doanh hiện đại.

+ Cung cấp sự thống nhất giữa các dữ liệu.

Với nhu cầu đòi hỏi ngày càng cao của xã hội nên vấn đề kỹ thuật trong mạng là mối quan tâm hàng đầu của các nhà tin học. Ví dụ như làm thế nào để truy xuất thông tin một cách nhanh chóng và tối ưu, trong khi việc xử lý thông tin trên mạng quá nhiều, đôi khi có thể làm tắc nghẽn và gây ra mất thông tin một cách đáng tiếc. Hiện nay, việc làm sao có được một hệ thống mạng chạy thật tốt, thật an toàn với lợi ích kinh tế cao đang rất được quan tâm.

1.1.3. Giao thức mạng

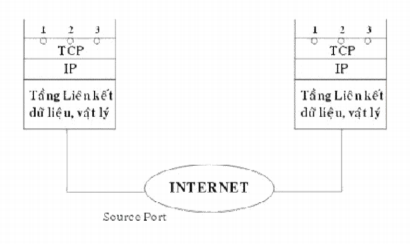
Giao thức mạng là một tập các quy tắc, quy ước để trao đổi thông tin giữa hai hệ thống máy tính hoặc hai thiết bị máy tính với nhau. Nói một cách hình thức thì giao thức mạng là một ngôn ngữ được các máy tính trong mạng sử dụng để trao đổi dữ liệu với nhau. Có nhiều loại giao thức được sử dụng trong mạng máy tính như: Apple Talk, DLC, NetBEUI,… nhưng hiện nay giao thức được sử dụng phổ biến nhất trong mạng máy tính là giao thức TCP/IP.

1.1.3.1 Giao thức TCP

Định nghĩa: TCP(Transmission Control Protocol) là giao thức hướng kết nối, nó cung cấp một đường truyền dữ liệu tin cậy giữa hai máy tính. Tính tin cậy thể hiện ở việc nó đảm bảo dữ liệu được gửi sẽ đến được đích và theo đúng thứ tự như khi nó được gửi.

Tính tin cậy của đường truyền được thể hiện ở hai điểm sau:

* Mọi gói tin cần gửi sẽ đến được đích. Để làm được điều này thì mỗi lần phía gửi gửi xong một gói tin nó sẽ chờ nhận một xác nhận từ bên nhận rằng đã nhận được gói tin. Nếu sau một khoảng thời gian mà phía gửi không nhận được thông tin xác nhận phản hồi thì nó sẽ phát lại gói tin. Việc phát lại sẽ được tiến hành cho đến khi việc truyền tin thành công, tuy nhiên sau một số lần phát lại max nào đó mà vẫn chưa thành công thì phía gửi có thể suy ra là không thể truyền tin được và sẽ dừng việc phát tin.
* Các gói tin sẽ được trình ứng dụng nhận được theo đúng thứ tự như chúng được gửi. Bởi các gói tin có thể được dẫn đi trên mạng theo nhiều đường khác nhau trước khi tới đích nên thứ tự khi tới đích của chúng có thể không giống như khi chúng được phát. Do đó để đảm bảo có thể sắp xếp lại gói tin ở phía nhận theo đúng thứ tự như khi chúng được gửi, giao thức TCP sẽ gắn vào mỗi gói tin một thông tin cho biết thứ tự của chúng trong cả khối tin chung được phát nhờ vậy bên nhận có thể sắp xếp lại các gói tin theo đúng thứ tự của chúng.



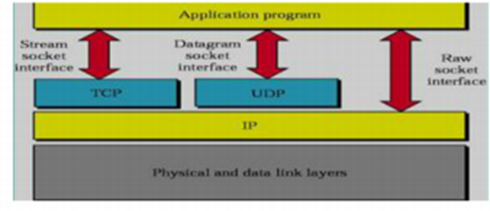
Hình 1: Kết nối TCP

1.1.3.2 Giao thức UDP

Định nghĩa: UDP (User Datagram Protocol) là giao thức không hướng kết nối, nó gửi các gói dữ liệu độc lập gọi là datagram từ máy tính này đến máy tính khác mà không đảm bảo việc dữ liệu sẽ tới đích.

Bảng sau so sánh sự khác biệt giữa hai chế độ giao tiếp hướng kết nối và không hướng kết nối.

|  |  |
| --- | --- |
| **Chế độ có hướng kết nối(TCP)** | **Chế độ không hướng kết nối(UDP)** |
| Tồn tại kênh giao tiếp ảo giữa hai bên giao tiếp | Không tồn tại kênh giao tiếp ảo giữa hai bên giao tiếp |
| Dữ liệu được gửi đi theo chế độ bảo đảm: Có kiểm tra lỗi truyền lại gói tin lỗi hay mất bảo đảm thứ tự đến của các gói tin… | Dữ liệu được gửi đi theo chế độ không bảo đảm: Không kiểm tra lỗi, không phát hiện không truyền lại gói tin bị lỗi hay mất, không bảo đảm thứ tự đến của các gói tin… |
| Dữ liệu chính xác, tốc độ truyền chậm | Dữ liệu không chính xác, tốc độ truyền nhanh. |
|  | Thích hợp cho các ứng dụng cần tốc độ, không cần chính xác cao: Truyền âm thanh, hình ảnh |



Hình 2: Các giao thức trong lập trình Socket

## 1.2. Tổng quan về C#

1.2.1. Các tính năng cơ bản của ngôn ngữ lập trình C#

* C# là một ngôn ngừ lập trình hướng đối tượng thuần túy (pure object oriented programming)
* Thu gom rác tự động: giảm bớt gánh nặng cho người lập trình viên trong việc phải viết các đoạn code thực hiện cấp phát và giải phóng bộ nhớ.
* Hồ trợ các chuẩn hóa được ra bởi tổ chức ECMA (European Computer Manufactures Association).
* Hỗ trợ các phương thức và các kiểu phổ quát (chung).

1.2.2. Các ứng dụng của C#

* C# có thể sử dụng để viết các kiểu ứng dụng khác nhau:
* Các ứng dụng game.
* Các ứng dụng cho doanh nghiệp.
* Các ứng dụng cho thiết bị di động: PC Pocket, PDA , cell phone.
* Các ứng dụng quản lý đơn giản: ứng dụng quản lý thư viện, quản lý thông tin cá nhân...
* Các ứng dụng phân tán phức tạp trải rộng qua nhiều thành phổ, đất nước.

1.2.3. Các lợi ích của C#

* Cross Language Support: hỗ trợ khả năng chuyến đổi dễ dàng giữa các ngôn ngữ.
* Hồ trợ các giao thức Internet chung.
* Triển khai đơn giản.
* Hỗ trợ tài liệu XML: các chú thích XML có thể được thêm vào các đoạn code

và sau đó có thể được chiết xuất để làm tài liệu cho các đoạn code để cho phép các lập trình viên khi sử dụng biết được ý nghĩa của các đoạn code đã viết.

## 1.3 Lập trình socket

1.3.1. Khái niệm Socket

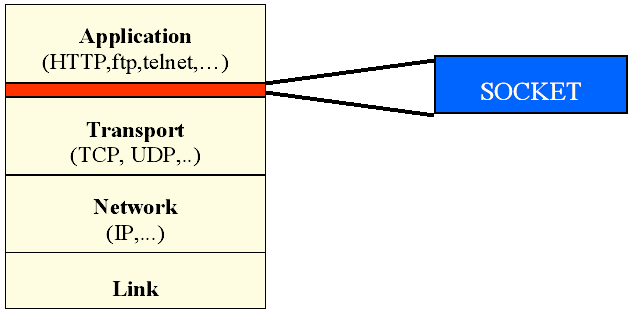
1.3.1.1. Lịch sử hình thành

Khái niệm Socket xuất hiện lần đầu tiên vào khoảng năm 1980 tại trường đại học Berkeley Mỹ. Đó là một chương trình được thiết kế để giúp máy tính nối mạng ở khắp mọi nơi có thể trao đổi thông tin với nhau. Lúc đầu có được sử dụng trên các máy Unix và có tên gọi là Berkeley Socket Interface.

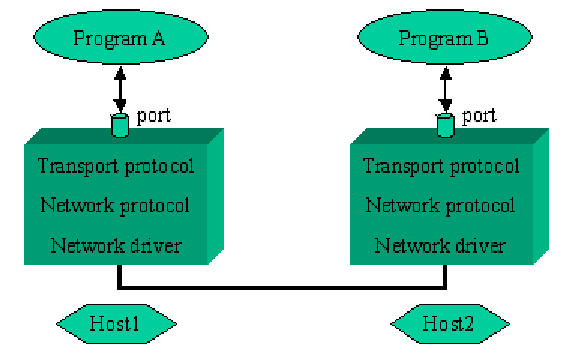
Tiếp đó cùng với sự phát triển của các ứng dụng mạng, socket được hỗ trợ trong nhiều ngôn ngữ lập trình và chạy trên nhiều nền tảng hệ điều hành khác nhau. Ví dụ như WinSock dùng cho các ứng dụng của Microsoft, Socket++ dùng cho các lập trình viên sử dụng Unix…

Nhìn trên quan điểm của người phát triển ứng dụng người ta có thể định nghĩa Socket là một phương pháp để thiết lập kết nối truyền thông giữa một chương trình yêu cầu dịch vụ (được gán nhãn là Client) và một chương trình cung cấp dịch vụ (được gán nhãn là server) trên mạng hoặc trên cùng một máy tính.

Đối với người lập trình, họ nhìn nhận *Socket như một giao diện nằm giữa tầng ứng dụng và tầng khác* trong mô hình mạng OSI có nhiệm vụ thực hiện việc giao tiếp giữa chương trình ứng dụng với các tầng bên dưới của mạng.



Hình 3. Mô hình OSI rút gọn



Hình 4. Mô hình Socket

Tuy nhiên, các lập trình viên hiện nay gần như luôn luôn bị ngăn cản tạo socket riêng bằng cách thủ công bởi dù bạn dùng Java, serlet hay CGI, PHP… có thể bạn sẽ không bao giờ mở được cổng một cách tường minh. Thay vào đó các lập trình viên sử dụng thư viện socket được hỗ trợ sẵn bởi các ngôn ngữ lập trình. Như vậy các socket vẫn tồn tại để kết nối các ứng dụng của người dùng, nhưng các chi tiết của socket được ẩn trong những lớp sâu hơn để mọi người không phải động chạm đến.

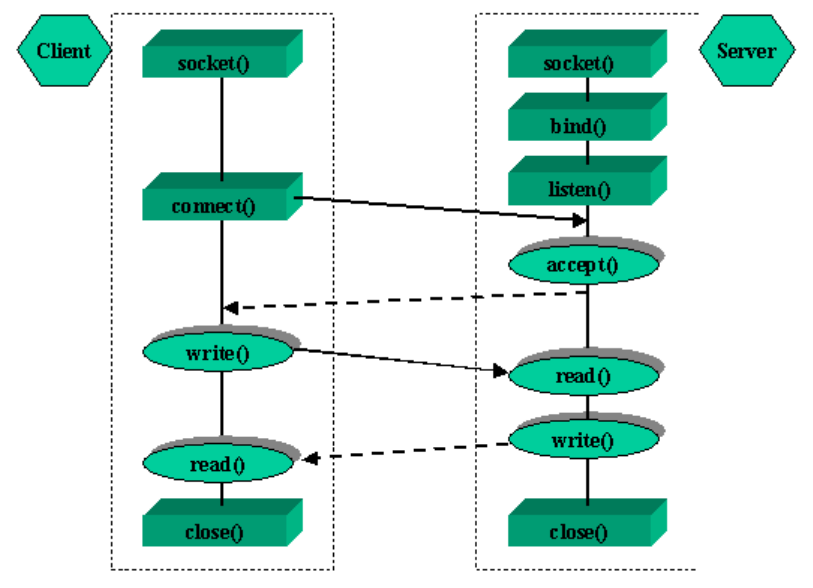
Số hiệu cổng của Socket

Để có thể thực hiện các cuộc giao tiếp, một trong hai quá trình phải công bố số hiệu cổng 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 gửi đến cổng này từ các quá trình khác. Quá trình còn lại cũng được yêu cầu tạo ra một socket.

Ngoài số hiệu cổng, hai bên giao tiếp còn phải biết địa chỉ IP của nhau. Địa chỉ IP giúp phân biệt máy tính này với máy tính kia trên mạng TCP/IP. Trong khi số hiệu cổng dùng để phân biệt các quá trình khác nhau trên cùng một máy tính.

Số hiệu cổng gán cho Socket phải duy nhất trên phạm vi máy tính đó, có giá trị trong khoảng từ 0 đến 65535 (16 bit). Trong thực tế thì các số hiệu cổng từ 0 đến 1023 (gồm có 1024 cổng) đã dành cho các dịch vụ nổi tiếng như: http: 80, telnet:21, ftp:23,…. Nếu chúng ta không phải là người quản trị thì nên dùng từ cổng 1024 trở lên.

Toàn bộ tiến trình diễn ra như sau:



Hình 5. Nguyên lý truyền socket

1.3.1.2. Nguyên lý hoạt động

Ta đã thấy khi hai ứng dụng muốn trao đổi dữ liệu qua mạng, chúng sẽ tạo ra ở mỗi phía một socket và trao đổi dữ liệu bằng cách đọc/ghi từ socket. Để hiểu rõ cách thức socket trao đổi dữ liệu chúng ta hãy xem xét nguyên lý hoạt động của chúng.

Trước hết chúng ta hãy xem xét làm thế nào các socket có thể xác định được nhau. Khi một chương trình tạo ra một socket, một định danh dạng số (định danh dạng số này còn được gọi là số hiệu cổng) sẽ được gán cho socket. Việc gán số hiệu cổng này cho socket có thể được thực hiện bởi chương trình hoặc hệ điều hành tùy theo cách socket được sử dụng như thế nào. Trong mỗi gói tin mà socket gửi đi có chứa hai thông tin để xác định đích đến của gói tin:

+ Một địa chỉ mạng để xác định hệ thống sẽ nhận gói tin.

+ Một số định danh cổng để nói cho hệ thống đích biết socket nào trên nó sẽ nhận dữ liệu.

Nhờ hai thông tin này mà gói tin có thể đến được đúng máy tính chứa socket mà nó cần đến (nhờ địa chỉ mạng) và được phân phối đến đúng socket đích (nhờ địa chỉ cổng của socket đích).

Chúng ta có thể khái quát quá trình trao đổi dữ liệu thông qua các socket như sau:

+ Chương trình phía server tạo ra một socket, socket này được chương trình gắn với một cổng trên server. Sau khi được tạo ra socket này (ta gọi là socket phía server) sẽ chờ nghe yêu cầu từ phía clients.

+ Khi chương trình phía clients cần kết nối với một server, nó cũng tạo ra một socket, socket này cũng được hệ điều hành gắn với một cổng. Chương trình client sẽ cung cấp cho socket của nó (ta gọi là socket phía client) địa chỉ mạng và cổng của socket phía server và yêu cầu thực hiện kết nối (nếu chương trình định sử dụng giao thức hướng kết nối) hoặc truyền dữ liệu (nếu chương trình sử dụng giao thức không hướng kết nối)

+ Chương trình phía server và chương trình phía clients trao đổi dữ liệu với nhau bằng cách đọc từ socket hoặc ghi vào socket của mình. Các socket ở hai phía nhận dữ liệu từ ứng dụng và đóng gói để gửi đi hoặc nhận các dữ liệu được gửi đến và chuyển cho chương trình ứng dụng bởi socket ở cả hai phía đều biết được địa chỉ mạng và địa chỉ cổng của nhau.

Ở bước thứ hai chúng ta thấy chương trình ứng dụng phải lựa chọn giao thức mà nó định sử dụng để trao đổi dữ liệu. Tuỳ theo việc chúng ta sử dụng giao thức nào (TCP hay UDP) mà cách thức xử lý trước yêu cầu của clients có thể khác.

Sau đây chúng ta sẽ xem xét chi tiết cách thức trao đổi dữ liệu của socket với từng loại giao thức.

*Socket hỗ trợ TCP*

a. Ở phía Server: Khi một ứng dụng trên server hoạt động nó sẽ tạo ra một socket và đăng ký với server một cổng ứng dụng và chờ đợi yêu cầu kết nối từ phía clients qua cổng này.

b. Ở phía clients: Nó biết địa chỉ của máy trên đó server đang chạy vào cổng và server đang chờ nghe yêu cầu. Do đó khi muốn kết nối đến server, nó cũng tạo một socket chứa địa chỉ máy client và cổng của ứng dụng trên máy clients đồng thời clients sẽ cung cấp cho socket của nó địa chỉ và cổng của server mà nó cần kết nối và yêu cầu socket thực hiện kết nối.

Khi server nhận được yêu cầu kết nối từ clients, nếu nó chấp nhận thì server sẽ sinh ra một socket mới được gắn với một cổng khác với cổng mà nó đang nghe yêu cầu. Sở dĩ server làm như vậy bởi nó cần cổng cũ để tiếp tục nghe yêu cầu từ phía clients trong khi vẫn cần một kết nối với clients.

Sau đó chương trình ứng dụng phía server sẽ gửi thông báo chấp nhận kết nối cho clients cùng thông tin về địa chỉ cổng mới của socket mà nó dành cho clients.

c. Quay lại phía clients, nếu kết nối được chấp nhận nghĩa là socket của nó đã được tạo ra thành công và nó có thể sử dụng socket để giao tiếp với server bằng cách viết và ghi tới socket theo cách giao tiếp với một tài nguyên trên máy tính thông thường.

*Socket hỗ trợ UDP*

a. Ở phía Server: Khi một ứng dụng trên server hoạt động nó sẽ tạo ra một socket và đăng ký với server một cổng ứng dụng và chờ đợi yêu cầu kết nối từ phía clients qua cổng này.

b. Ở phía Clients: Nó biết địa chỉ của máy trên đó server đang chạy vào cổng và server đang chờ nghe yêu cầu. Do đó khi muốn giao tiếp với server, nó cũng tạo ra một socket chứa địa chỉ máy clients và cổng của ứng dụng trên máy clients đồng thời clients sẽ cung cấp cho socket của nó địa chỉ và cổng của server mà nó cần kết nối. Khi clients muốn gửi tin đế server nó sẽ chuyển dữ liệu cho socket của mình, socket này sẽ chuyển thẳng gói tin mà client muốn gửi tới server dưới dạng một datagram có chứa địa chỉ máy server và cổng mà server đang chờ nghe yêu cầu. Như vậy không hề có một kết nối nào được thực hiện giữa client với server và server cũng không cần tạo ra một socket khác để kết nối với clients thay vào đó server dùng ngay cổng ban đầu để trao đổi dữ liệu.

# CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH ỨNG DỤNG

**2.1. Giới thiệu**

Trao đổi dữ liệu giữa hai máy tính trong mạng thực chất là sự trao đổi dữ liệu giữa hai chương trình ứng dụng chạy trên hai máy tính đó. Trong đó, một chương trình được gán nhãn là server và một chương trình được gán nhãn là client, có nhiều phương pháp để xây dựng chương trình ứng dụng mạng nhưng phương pháp sử dụng phổ biến là lập trình ứng mạng dựa trên cơ chế socket. Trong chương này sẽ trình bầy một ứng dụng của lập trình socket TCP là xây dựng chương trình truyền file qua mạng giữa hai máy tính bằng Java Socket TCP.

## 2.2. Phân tích chương trình

Chương trình ứng dụng được xây dựng theo mô hình clients/server. Chương trình bao gồm hai mô đun server và client. Người sử dụng có thể truyền file từ phía client cho server hoặc ngược lại.

**2.3. Cơ chế hoạt động của chương trình**

Chương trình gồm hai mô đun. Phía server là file chương trình có tên là FileTransferServer.java, phía client là file chương trình có tên là FileTransferClient.java. Sau khi biên dịch file .java này ta nhận được các file .class tương ứng.

*Chạy chương trình ở server mode:*

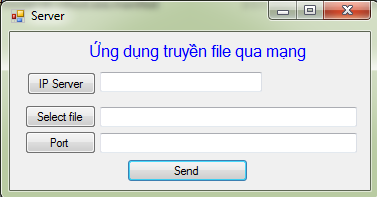
Chương trình chạy phía server đã được chỉ định chạy server mode, sau đó máy tính này sẽ chờ đợi các kết nối từ phía clients đến nó. Ta phải nhập tiếp port number, ứng với server mode này thì ta có thể chọn bất cứ port number nào lớn hơn 1024, vì những port number dưới 1024 đã bị giữ trước và sử dụng bởi hệ thống.

*Chạy chương trình ở client mode:*

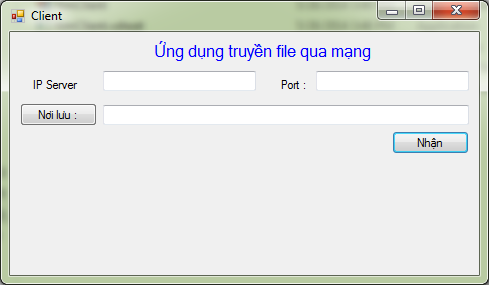
Tương tự cho phía bên client, chương trình sẽ yêu cầu nhập vào địa chỉ của server (host address), ta có thể nhập địa chỉ IP hay nhập vào tên của máy chạy server mode đều được (trong trường hợp trên hình bên dưới thì tên máy chạy server mode có địa chỉ IP trong mạng LAN là 192.168.1.43). Tiếp tục ta sẽ nhập port number (số hiệu cổng) của server socket (đã biết) cần kết nối đến.

## 2.4. Giao diện chương trình

2.4.1 Giao diện Server



2.4.2 Giao diện client



# CHƯƠNG 3: KẾT LUẬN

Truyền file qua mạng là một trong những ứng dụng phổ biến trên mạng LAN và Internet như: tải xuống các file từ một máy chủ file ở xa, gửi/nhận thư điện tử, ... Truyền file qua mạng dựa trên Socket TCP là một phương pháp truyền file có độ tin cậy cao bởi vì trước khi truyền nó cần thiết lập thành công kênh truyền dữ liệu

Không phải là phương pháp thay thế hoàn toàn những phương pháp truyền file khác mà ta đã từng sử dụng. Bản chất của phương pháp truyền file dựa vào Socket TCP là nhằm tăng thêm hiệu xuất làm việc. Đề tài “Tìm hiểu lập trình socket TCP trong java và ứng dụng truyền file qua mạng”đã đạt được kết quả nhất định.

Về cơ sở lý thuyết, đồ án đã trình bầy được các nội dung về mạng máy tính, sơ lược về ngôn ngữ Java, lập trình Socket TCP nói chung và lập trình Socket TCP trong Java nói riêng; các nội dung liên quan đến truyền file quua mạng.

Về ứng dụng đồ án đã phân tích một cách khá chi tiết cơ chế hoạt động của chương trình ở phía clients, phía server và đã cài đặt thành công chương trình. Java là một ngôn ngữ mạnh mẽ, tính bảo mật cao và độc lập với nền, do đó chương trình ứng dụng của đồ án có thể dễ dàng chạy trên các hệ thống khác nhau mà không phải lập trình lại.