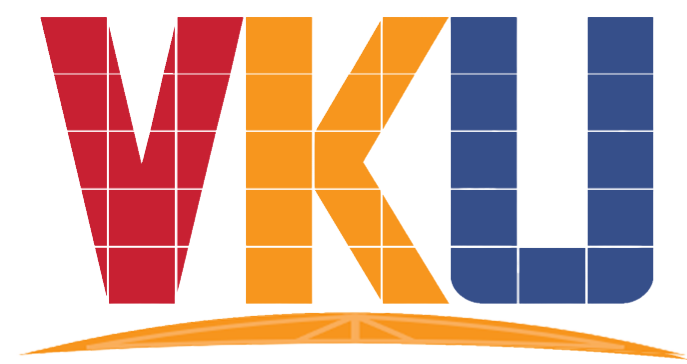


**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**VÀ TRUYỀN THÔNG VIỆT - HÀN**



**LẬP TRÌNH JAVA NÂNG CAO**

XÂY DỰNG CHƯƠNG TRÌNH CHAT SỬ DỤNG TCP SOCKET

**Giảng viên hướng dẫn : TS . Lê Văn Minh**

**Sinh viên thực hiện : Bùi Văn Vạn Quý**

**Lớp :** 18IT5

***Đà Nẵng, tháng 06 năm 2021***

# NHẬN XÉT

................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................ ................................................................................................................................

**Ngày 17 tháng 06 năm 2021**

**Giảng viên**

**MỤC LỤC**

[NHẬN XÉT 1](#_Toc74864978)

[LỜI CẢM ƠN 3](#_Toc74864979)

[LỜI MỞ ĐẦU 4](#_Toc74864980)

[CHƯƠNG 1: GIỚI THIỆU 5](#_Toc74864981)

[1.1 Lý do thực hiện đề tài 5](#_Toc74864982)

[1.2 Phương pháp triển khai đề tài 5](#_Toc74864983)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6](#_Toc74864984)

[2.1 Hiểu mô hình Client-Server 6](#_Toc74864985)

[2.1.1 Thế nào là mô hình Client-Server? 6](#_Toc74864986)

[2.1.2 Mô hình Client-Server hoạt động như thế nào? 6](#_Toc74864987)

[2.1.3 Mô hình Client-Server có những ưu, nhược điểm gì? 7](#_Toc74864988)

[2.2 Tìm hiểu về Socket 8](#_Toc74864989)

[2.2.1 Socket là gì? 8](#_Toc74864990)

[2.2.2 Tại sao người dùng lại cần đến Socket? 8](#_Toc74864991)

[2.2.3 Socket hoạt động như thế nào? 9](#_Toc74864992)

[2.2.4 Phân loại Socket 10](#_Toc74864993)

[2.3 Mô hình TCP/IP là gì? 13](#_Toc74864994)

[2.4 Sự phát triển và hình thành của mô hình TCP/IP 13](#_Toc74864995)

[2.5 Cách thức hoạt động của mô hình TCP/IP 14](#_Toc74864996)

[2.6 Chức năng của các tầng trong mô hình TCP/IP 14](#_Toc74864997)

[2.7 Lập trình TCP Socket 17](#_Toc74864998)

[2.3.1 Lập trình TCP Socket với Java 19](#_Toc74864999)

[CHƯƠNG 3: TRIỂN KHAI CHƯƠNG TRÌNH CHAT SỬ DỤNG TCP SOCKET 21](#_Toc74865000)

[3.1 Yêu cầu 21](#_Toc74865001)

[3.2 Môi trường triển khai 21](#_Toc74865002)

[3.3 Triển khai chương trình Chat 21](#_Toc74865003)

[3.3.1 Về phía Server 21](#_Toc74865004)

[3.3.2 Về phía Client 22](#_Toc74865005)

[KẾT LUẬN 24](#_Toc74865006)

[TÀI LIỆU THAM KHẢO 25](#_Toc74865007)

# LỜI CẢM ƠN

Sau khi được học môn lập trình mạng dưới sự giảng dạy nhiệt tình của thầy Lê Văn Minh em đã tiếp thu được đầy kiến thức để áp dụng thực hiện đề tài. Nhờ sự phân công của thầy, em đã hoàn thành xây dựng đề tài “Xây dựng chương trình Chat sử dụng TCP Socket”.

Và để hoàn thành tốt đề tài này, chúng em xin cảm ơn tới quý thầy cô Trường Đại học Công nghệ thông tin và truyền thông Việt – Hàn đã tạo điều kiện cho em nghiên cứu và hoàn thành đề tài. Tiếp đến, em xin chân trọng cảm ơn giảng viên hướng dẫn em thầy Lê Văn Minh đã tận tình giúp đỡ, chỉ bảo hướng dẫn chúng em trong suốt quá trình thực hiện đề trình này.

Vì thời gian và năng lực còn có hạn chế nên không thể tránh khỏi những sai sót trong khi thực hiện đề tài nghiên cứu của mình. Rất mong được sự góp ý bổ sung của thầy để đề tài của em ngày càng hoàn thiện hơn.

Xin chân thành cảm ơn !

# LỜI MỞ ĐẦU

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là được xem là một trong những vấn đề quan trong của các tổ chức, các công ty. Nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ. Cùng với sự phát triển mạnh mẽ của công nghệ thông tin và mạng Internet, nhu cầu sử dụng máy tính của con người cũng ngày càng cao hơn. Trao đổi thông tin hay trò chuyện với nhau bằng máy tính hay tham gia vào các phòng chat chính là những ví dụ điển hình cho hình thức này. Các hình thức chat là những nơi mà các tài khoản đã đăng ký đăng nhập vào bằng một hình thức nào đó, ví dụ như dùng tài khoản và mật khẩu, ví dụ như các ứng dụng mà chúng ta hay sử dụng hiện nay như Zalo, Facebook hoặc Yahoo. Hay đơn giản hơn là gia nhập vào một máy chủ hay còn gọi là một Server) bằng địa chỉ IP của chính máy chủ đó. Sau khi gia nhập vào máy chủ ấy thì các tài khoản (còn gọi là Client) sẽ tham gia chat.

Vì những lí do thiết yếu ở trên nên em đã thực hiện đề tài xây dựng một mô hình Chat giữa Sever và Client một cách đơn giản.

# CHƯƠNG 1: GIỚI THIỆU

## 1.1 Lý do thực hiện đề tài

Dưới sự phát triển bùng nổ của thời đại Internet, hiện nay nhu cầu sử dụng các thiết bị công nghệ thông tin của con người ở mức rất cao, việc trao đổi thông tin qua ứng dụng Chat trên mạng đã được phổ biến rộng rãi. Một số ứng dụng về Chat phổ biến hiện nay như Zalo, Messenger, Mocha,.... Đa phần mọi người dung ứng dụng Chat với mục đích trao đổi thông tin như trò chuyện, làm quen bạn bè, bàn bạc công việc gián tiếp qua môi trường Internet,..và ứng dụng Chat đơn giản là để kết nối mọi người lại với nhau. Khi xây dựng một chương trình Chat cần phải đảm bảo các yêu cầu về kết nối, dữ liệu trong quá trình truyền đi phải đảm bảo toàn vẹn, an toàn và bảo mật cao.

## 1.2 Phương pháp triển khai đề tài

Là một ứng dụng mạng nên các chương trình Chat 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, em đã xây dựng chương trình Chat hoạt động theo mô hình Client – Server và sử dụng giao thức kết nối TCP.

Ngôn ngữ lập trình sử dụng là ngôn ngữ Java (JDK 8) vì Java là ngôn ngữ có thể được sử dụng để viết Ứng dụng chạy trên hầu hết các thiết bị.

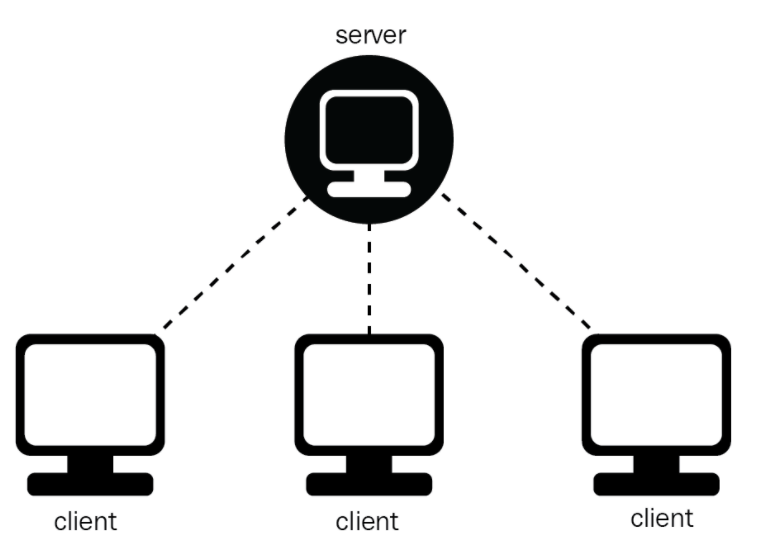
# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Hiểu mô hình Client-Server

### 2.1.1 Thế nào là mô hình Client-Server?

Mô hình Client-Server (hay còn gọi là mô hình máy khách - máy chủ) là mô hình mà ở đây, máy khách là các máy tính, các thiết bị điện tử như máy in, điện thoại bàn, máy fax,…

Ý tưởng của mô hình này là máy con (đóng vai trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý các yêu cầu đó và trả kết quả về cho máy khách.



*Hình 2.1: Mô hình Client-Server*

### 2.1.2 Mô hình Client-Server hoạt động như thế nào?

Chương trình server và client nói chuyện với nhau bằng các thông điệp (messages) thông qua một cổng truyền thông liên tác IPC (InterProcess Communication). Một số IPC phổ biến là Pipe, Socket, RPC.

Để một chương trình server và một chương trình client có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn để nói chuyện, chuẩn này được gọi là giao thức.

Nếu một chương trình client nào đó muốn yêu cầu lấy thông tin từ server thì nó phải tuân theo giao thức mà server đó đưa ra.

Các giao thức chuẩn (ở tầng mạng và vận chuyển) được sử dụng rộng rãi nhất hiện nay như: giao thức TCP/IP, giao thức SNA của IBM, OSI, ISDN, X.25 hoặc giao thức LAN-to-LAN NetBIOS.

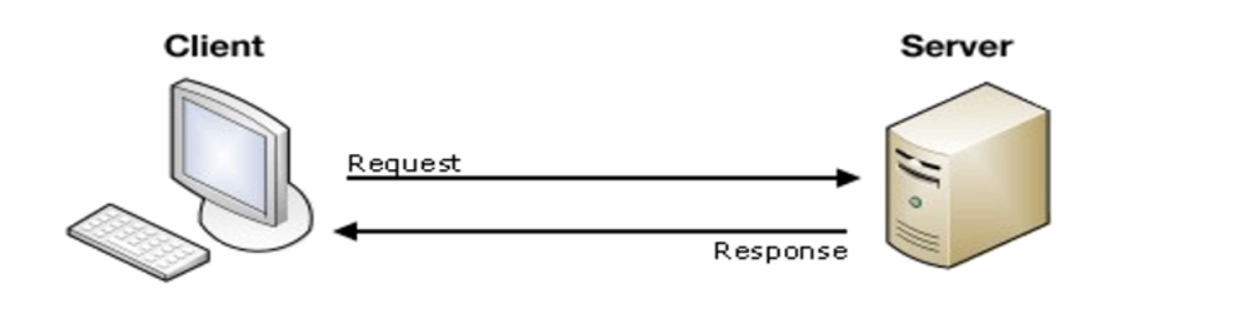
Quy trình hoạt động của mô hình này lặp đi lặp lại 2 quá trình như sau:

* Client: Gửi dữ liệu lên server, chờ nhận dữ liệu từ server trả về.

(Tạo ra 1 request -- gửi đến Server -- chờ Server xử lý – Nhận kết quả trả về)

* Server : Nhận thông tin, dữ liệu được gửi lên từ client, xử lý và gửi dữ liệu cho các client đang kết nối Server chấp nhận tất cả các yêu cầu hợp lệ từ mọi nơi khác nhau trên Internet, sau đó trả kết quả về máy tính đã gửi yêu cầu đó.

(Lắng nghe request từ Client -- nhận request -- xử lý request – Response cho Client)



*Hình 2.2: Nguyên tắc hoạt động mô hình Client-Server*

### 2.1.3 Mô hình Client-Server có những ưu, nhược điểm gì?

* **Ưu điểm:**

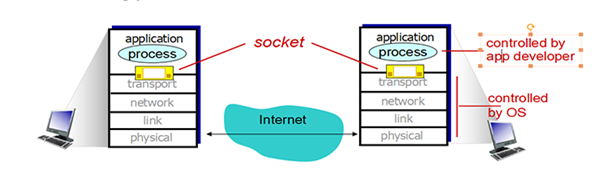
Mô hình client server là với mô hình client server thì mọi thứ dường như đều nằm trên bàn của người sử dụng, nó có thể truy cập dữ liệu từ xa (bao gồm các công việc như gửi và nhận file, tìm kiếm thông tin,…) với nhiều dịch vụ đa dạng mà mô hình cũ không thể làm được. Mô hình client/server cung cấp một nền tảng lý tưởng cho phép tích hợp các kỹ thuật hiện đại như mô hình thiết kế hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS)…

* **Nhược điểm:**

Một trong những vấn đề nảy sinh trong mô hình này đó là tính an toàn và bảo mật thông tin trên mạng. Do phải trao đổi dữ liệu giữa 2 máy ở 2 khu vực khác nhau cho nên dễ dàng xảy ra hiện tượng thông tin truyền trên mạng bị lộ.

## 2.2 Tìm hiểu về Socket

### 2.2.1 Socket là gì?



*Hình 2.3: Socket*

Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Nghĩa là một socket được sử dụng để cho phép 1 process nói chuyện với 1 process khác. Điểm cuối (end-point) của liên kết này được gọi là Socket.

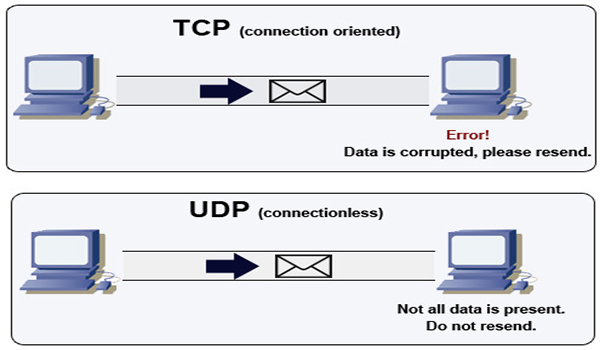
Một chức năng khác của socket là giúp các tầng TCP hoặc TCPLayer định danh ứng dụng từ đó truyền dữ liệu dữ liệu sẽ được gửi tới thông qua sự ràng buộc với một cổng Port [1024-9999], từ đó tiến hành kết nối giữa client và server.

### 2.2.2 Tại sao người dùng lại cần đến Socket?

Ưu điểm lớn nhất của socket là hỗ trợ hầu hết các hệ điều hành bao gồm MS Windows, Linux,… Ngoài ra, socket cũng được sử dụng với nhiều ngôn ngữ lập trình, gồm C, C++, Java, Visual Basic, Visual C++,… nên nó có thể tương thích với hầu hết mọi đối tượng người dùng với những cấu hình máy khác nhau.

Đặc biệt, người dùng cũng có thể chạy cùng một lúc nhiều socket liên tục, giúp nâng cao hiệu suất làm việc, cũng như tiết kiệm thêm nhiều thời gian và công sức hơn.

### 2.2.3 Socket hoạt động như thế nào?



*Hình 2.4: TCP và UDP*

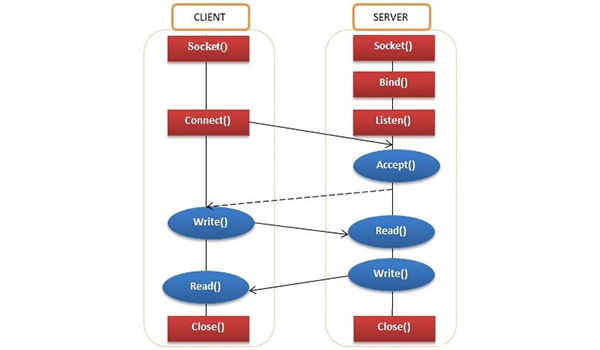
Chức năng của Socket là kết nối giữa client và server thông qua **TCP/IP** và**UDP** để truyền và nhận giữ liệu qua Internet. Giao diện lập trình ứng dụng mạng này chỉ có thể hoạt động khi đã có thông tin về thông số IP và số hiệu cổng của 2 ứng dụng cần trao đổi dữ liệu cho nhau.

Hai ứng dụng cần truyền thông tin phải đáp ứng điều kiện sau thì socket mới có thể hoạt động:

* 2 ứng dụng có thể nằm cùng trên một máy hoặc 2 máy khác nhau
* Trong trường hợp 2 ứng dụng cùng nằm trên một máy, số hiệu cổng không được trùng nhau.

### 2.2.4 Phân loại Socket

#### **a.** Stream Socket



*Hình 2.5: Stream Socket*

Stream Socket hay còn gọi là socket hướng kết nối, là socket hoạt động thông qua giao thức TCP (Transmission Control Protocol). Stream Socket chỉ hoạt động khi server và client đã kết nối với nhau.

**Ưu điểm:**

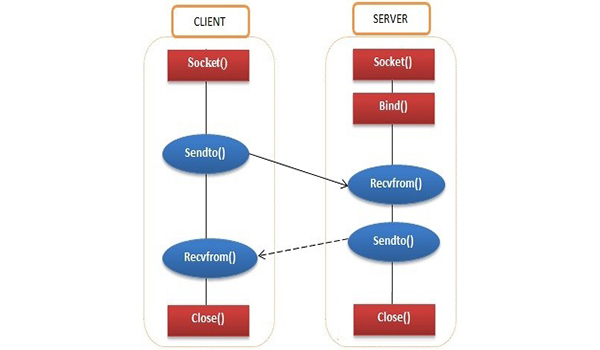
* Dữ liệu truyền đi được đảm bảo truyền đến đúng nơi nhận, đúng thứ tự với thời gian nhanh chóng.
* Mỗi thông điệp gửi đi đều có xác nhận trả về để thông báo cho người dùng thông tin về quá trình truyền tải.

**Nhược điểm:**

* Giữa máy chủ và máy nhận chỉ có 1 IP, nên khi kết nối, 1 máy phải chờ máy còn lại chấp nhận kết nối.

#### b. Datagram Socket

Datagram Socket hay còn gọi là socket không hướng kết nối, là socket hoạt động thông quagiao thức UDP ( User Datagram Protocol). Datagram Socket có thể hoạt động kể cả khi không có sự thiết lập kết nối giữa 2 máy với nhau.



*Hình 2.6: Datagram Socket*

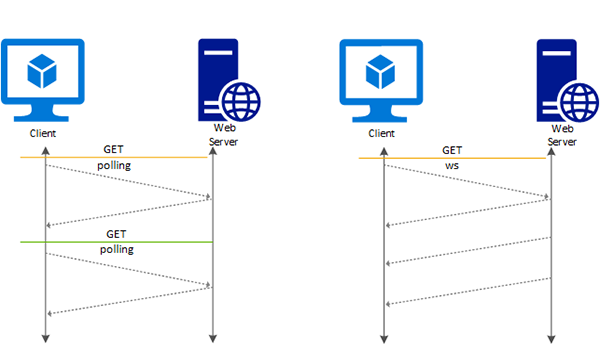
**Ưu điểm:**

* Quá trình kết nối và truyền tải thông tin đơn giản, không cần thực hiện nhiều thao tác.
* Thời gian truyền tải dữ liệu cực nhanh.

**Nhược điểm:**

* Quá trình truyền thông tin không đảm bảo tin cậy, thông tin có thể truyền sai thứ tự hoặc bị lặp.

#### c. Websocket



*Hình 2.7: WebSocket*

Websocket là công cụ hỗ trợ việc kết nối qua lại trên internet giữa client và server. Giúp diễn ra nhanh chóng và hiệu quả hơn thông qua việc sử dụng TCPsocket. Không chỉ sử dụng riêng cho ứng dụng web, Websocket có thể áp dụng cho bất kì ứng dụng nào khác cần có sự trao đổi thông tin trên Internet.

**Ưu điểm:**

Websocket mang lại nhiều ưu điểm trong việc kết nối giữa client và server. Cụ thể như sau:

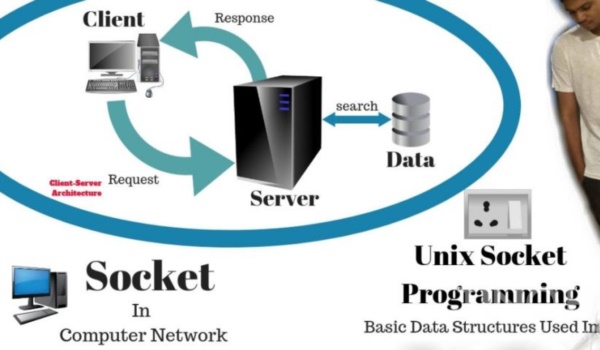
* Tăng tốc độ truyền tải thông tin giữa 2 chiều
* Dễ phát hiện và xử lý trong trường hợp có lỗi xảy ra
* Dễ dàng sử dụng, không cần cài đặt thêm các phần mềm bổ sung khác
* Không cần sử dụng nhiều phương pháp kết nối khác nhau

**Nhược điểm:**

Một số nhược điểm của Websocket mà bạn cần lưu ý khi sử dụng có thể kể đến như:

* Chưa hỗ trợ trên tất cả các trình duyệt
* Với các dịch vụ có phạm vi yêu cầu, Websocket chưa hỗ trợ hoàn toàn.

#### d. Unix socket



*Hình 2.8: Unix Socket*

Unix socket là điểm giao tiếp hỗ trợ trao đổi giữa các ứng dụng khác nhau ngay trên cùng máy tính. Mọi hoạt động Unix socket diễn ra ngay ở nhân của hệ điều hành. Nhờ vậy, tốc độ kết nối và truyền tải giữa các ứng dụng nhanh, nhẹ và hiệu quả hơn.

Unix socket có thể tránh được các bước kiểm tra hoặc routing. Vì thế mà quá trình truyền tin sẽ đảm bảo và dễ dàng hơn.

**Ưu điểm:**

Unix socket có nhiều ưu điểm vượt trội. Nó hỗ trợ cho việc truyền tải nhanh hơn và hiệu quả hơn. Trong đó, tiêu biểu phải kể đến các ưu điểm như:

* Tăng tốc độ truy cập MySQL lên đến 30-50%
* Giảm thời gian latency xuống, từ 60ms còn 5ms
* Tăng PostgreSQL lên  hơn 30%
* Tăng Redis lên 50%

**Nhược điểm:**

Unix socket cũng còn tồn tại một số nhược điểm như sau:

* Trong trường hợp các ứng dụng nằm trên những máy chủ khác nhau, sẽ không thể kết nối bằng Unix socket.
* Vấn đề phân quyền giữa các tệp tin trên Unix socket đôi khi vẫn xảy ra, ảnh hưởng đến việc sử dụng và thao tác.

## 2.3 Mô hình TCP/IP là gì?

TCP/ IP (Transmission Control Protocol/ Internet Protocol - Giao thức điều khiển truyền nhận/ Giao thức liên mạng), là một bộ giao thức trao đổi thông tin được sử dụng để truyền tải và kết nối các thiết bị trong mạng Internet. TCP/IP được phát triển để mạng được tin cậy hơn cùng với khả năng phục hồi tự động.

## 2.4 Sự phát triển và hình thành của mô hình TCP/IP

Ý tưởng hình thành mô hình TCP/IP được bắt nguồn từ Bộ giao thức liên mạng trong công trình DARPA vào năm 1970. Trải qua vô số năm nghiên cứu và phát triển của 2 kỹ sư [Robert E. Kahn](https://vi.wikipedia.org/w/index.php?title=Robert_E._Kahn&action=edit&redlink=1) và [Vinton Cerf](https://vi.wikipedia.org/wiki/Vinton_Cerf) cùng sự hỗ trợ của không ít các nhóm nghiên cứu. Đầu năm 1978, giao thức TCP/ IP được ổn định hóa với giao thức tiêu chuẩn được dùng hiện nay của Internet đó là mô hình TCP/IP Version 4.

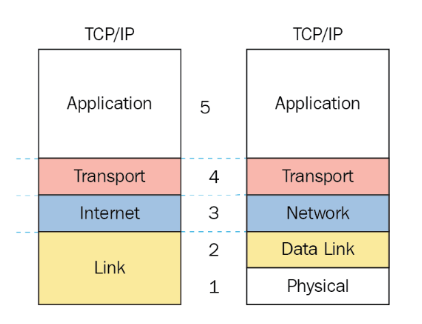
Vào năm 1975, cuộc thử nghiệm thông nối giữa 2 mô hình TCP/IP được diễn ra thành công. Cũng bắt đầu từ đây, cuộc thử nghiệm thông nối giữa các mô hình TCP/IP được diễn ra nhiều hơn và đều đạt được kết quả tốt. Cũng chính vì điều này, một cuộc hội thảo được Internet Architecture Broad mở ra, với sự tham dự của hơn 250 đại biểu của các công ty thương mại, từ đây giao thức và mô hình TCP/IP được phổ biến rộng rãi trên khắp thế giới.

## 2.5 Cách thức hoạt động của mô hình TCP/IP

Phân tích từ tên gọi, TCP/IP là sự kết hợp giữa 2 giao thức. Trong đó [IP](https://www.totolink.vn/article/74-ip-la-gi-cach-xac-dinh-dia-chi-ip-tren-may-tinh.html) (Giao thức liên mạng) cho phép các gói tin được gửi đến đích đã định sẵn, bằng cách thêm các thông tin dẫn đường vào các gói tin để các gói tin được đến đúng đích đã định sẵn ban đầu. Và giao thức TCP (Giao thức truyền vận) đóng vai trò kiểm tra và đảm bảo sự an toàn cho mỗi gói tin khi đi qua mỗi trạm. Trong quá trình này, nếu giao thức TCP nhận thấy gói tin bị lỗi, một tín hiệu sẽ được truyền đi và yêu cầu hệ thống gửi lại một gói tin khác. Quá trình hoạt động này sẽ được làm rõ hơn ở chức năng của mỗi tầng trong mô hình TCP/IP.

## 2.6 Chức năng của các tầng trong mô hình TCP/IP

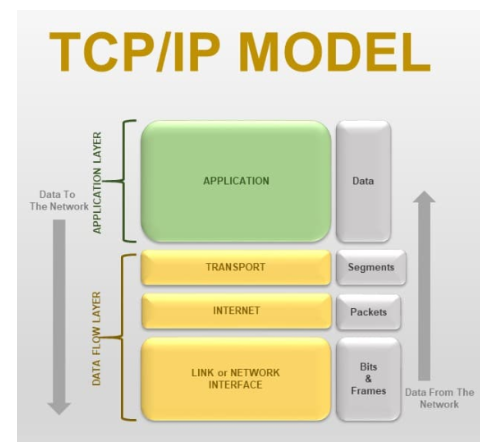
Một mô hình TCP/IP tiêu chuẩn bao gồm 4 lớp được chồng lên nhau, bắt đầu từ tầng thấp nhất là Tầng vật lý (Physical) → Tầng mạng (Network) → Tầng giao vận (Transport) và cuối cùng là Tầng ứng dụng (Application).

 Tuy nhiên, một số ý kiến lại cho rằng mô hình TCP/IP là 5 tầng, tức các tầng 4 đến 2 đều được giữ nguyên, nhưng tầng Datalink sẽ được tách riêng và là tầng nằm trên so với tầng vật lý.

*Hình 2.9: Mô hình TCP/IP*

***Tầng 4 - Tầng Ứng dụng (Application)***

Đây là lớp giao tiếp trên cùng của mô hình. Đúng với tên gọi, tầng Ứng dụng đảm nhận vai trò giao tiếp dữ liệu giữa 2 máy khác nhau thông qua các dịch vụ mạng khác nhau (duyệt web, chat, gửi email, một số giao thức trao đổi dữ liệu: SMTP, SSH, [FTP](https://www.totolink.vn/article/154-giao-thuc-ftp-la-gi-nhung-dieu-ban-chua-biet-ve-giao-thuc-ftp.html),...). Dữ liệu khi đến đây sẽ được định dạng theo kiểu Byte nối Byte, cùng với đó là các thông tin định tuyến giúp xác định đường đi đúng của một gói tin.

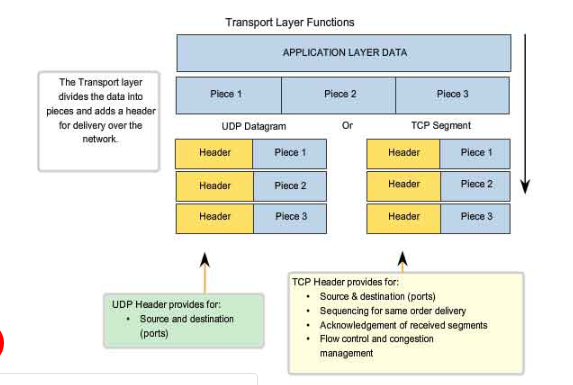


*Hình 2.10: Tầng ứng dụng*

***Tầng 3 - Tầng Giao vận (Transport)***

Chức năng chính của tầng 3 là xử lý vấn đề giao tiếp giữa các máy chủ trong cùng một mạng hoặc khác mạng được kết nối với nhau thông qua bộ định tuyến. Tại đây dữ liệu sẽ được phân đoạn, mỗi đoạn sẽ không bằng nhau nhưng kích thước phải nhỏ hơn 64KB. Cấu trúc đầy đủ của một Segment lúc này là Header chứa thông tin điều khiển và sau đó là dữ liệu.

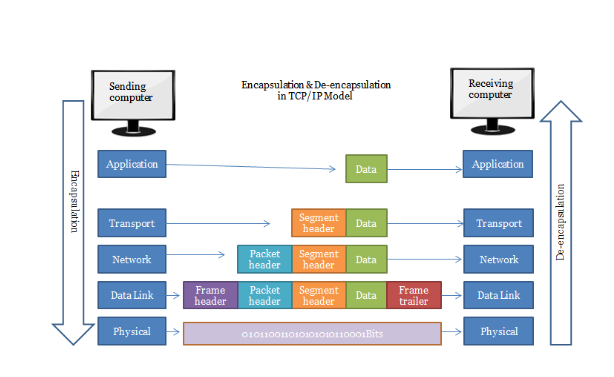
Trong tầng này còn bao gồm 2 giao thức cốt lõi là TCP và UDP. Trong đó, TCP đảm bảo chất lượng gói tin nhưng tiêu tốn thời gian khá lâu để kiểm tra đầy đủ thông tin từ thứ tự dữ liệu cho đến việc kiểm soát vấn đề tắc nghẽn lưu lượng dữ liệu. Trái với điều đó, UDP cho thấy tốc độ truyền tải nhanh hơn nhưng lại không đảm bảo được chất lượng dữ liệu được gửi đi.



*Hình 2.11: Tầng giao vận*

***Tầng 2 - Tầng mạng (Internet)***

Gần giống như tầng mạng của [mô hình OSI](http://www.totolink.vn/article/136-mo-hinh-osi-la-gi-chuc-nang-cua-cac-tang-giao-thuc-trong-mo-hinh-osi.html). Tại đây, nó cũng được định nghĩa là một giao thức chịu trách nhiệm truyền tải dữ liệu một cách logic trong mạng. Các phân đoạn dữ liệu sẽ được đóng gói (Packets) với kích thước mỗi gói phù hợp với mạng chuyển mạch mà nó dùng để truyền dữ liệu. Lúc này, các gói tin được chèn thêm phần Header chứa thông tin của tầng mạng và tiếp tục được chuyển đến tầng tiếp theo. Các giao thức chính trong tầng là IP, [ICMP](https://www.totolink.vn/article/73-icmp-la-gi-9-loai-icmp-thuong-thay.html) và ARP.



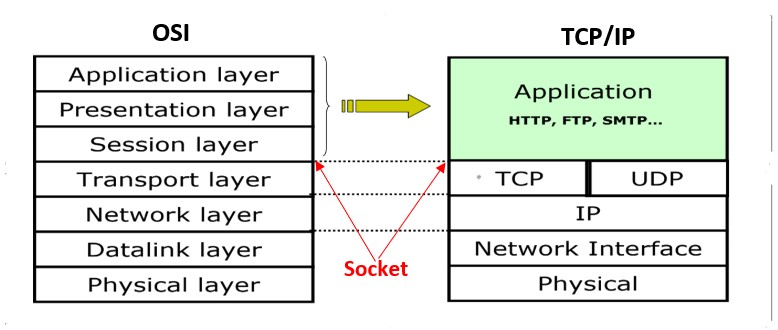
*Hình 2.12: Tầng mạng*

***Tầng 1 - Tầng Vật lý (Physical)***

Là sự kết hợp giữa tầng Vật lý và tầng liên kết dữ liệu của mô hình OSI. Chịu trách nhiệm truyền dữ liệu giữa hai thiết bị trong cùng một mạng. Tại đây, các gói dữ liệu được đóng vào khung (gọi là Frame) và được định tuyến đi đến đích đã được chỉ định ban đầu.

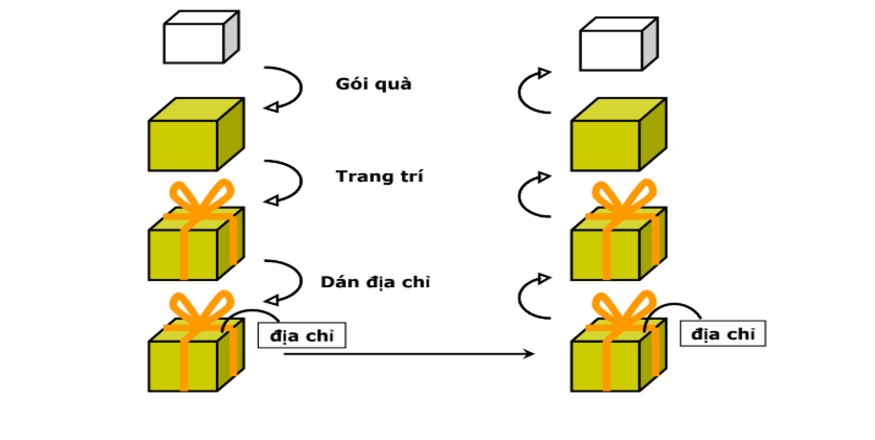
## 2.7 Lập trình TCP Socket

Trong hệ thống mạng máy tính tồn tại những mô hình tham chiếu có kiến trúc phần tầng (OSI, TCP/IP…) nhằm hỗ trợ chức năng trao đôi thông tin giữa các ứng dụng ở nhiều máy tính khác nhau.



*Hình 2.13: Kiến trúc phần tầng (OSI, TCP/IP…)*

Dữ liệu bên gửi sẽ được đóng gói (Encapsulation) từ tầng trên đến tầng cuối là tầng vật lí (Physical Layer), sau đó nhờ tầng vật lí này chuyển dữ liệu đến tầng vật lí máy bên nhận, bên nhận tiến hành giải mã (decapsulation) gói dữ kiện từ tầng dưới lên tầng trên cùng, là tầng ứng dụng (application layer).



*Hình 2.14: Truyền gói tin*

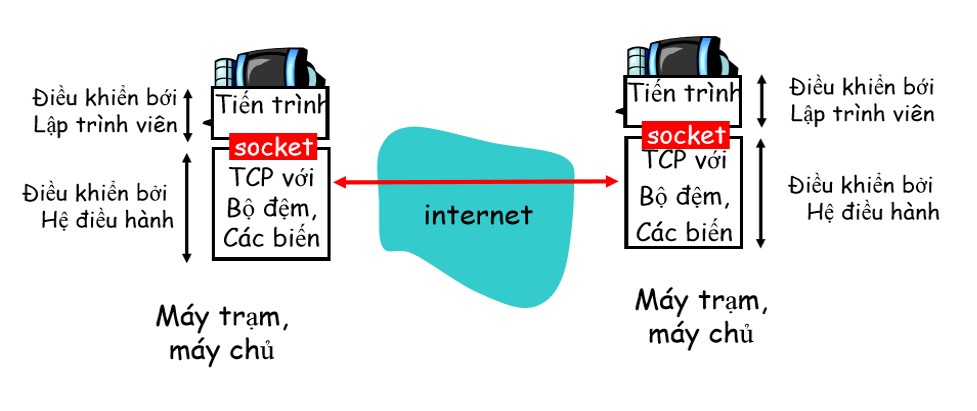
Ở đây, Socket chính là cửa giao tiếp giữa tầng ứng dụng và tầng giao vận (Transport layer). Nói cách khác, Socket là giao diện do ứng dụng tạo ra trên máy trạm, quản lí bởi hệ điều hành qua đó các ứng dụng có thể gửi/nhận thông điệp đến/từ các ứng  dụng khác. Ở đó, Socket sẽ được ràng buộc với một mã số cổng (Port Number) để giúp tầng giao vận định danh được ứng dụng nhận/gửi thông điệp.

Như ở hình 2.9, tầng giao vận có 2 phương thức là TCP (Ttransmission Control Protocol) và UDP (User Datagram Protocol), như vậy socket cơ bản là có 2 loại: Stream Socket sử dụng TCP truyền dòng bytes và Datagram Socket dử dụng UDP truyền gói tin. Với ngôn ngữ lập trình Java, chúng được cung cấp 3 loại khác nhau của Sockets:

* Stream Socket (TCP) : Tạo luồng dữ liệu hai chiều, đáng tin cậy, có trình tự và không trùng lặp, dữ liệu chỉ được gửi/nhận khi có đã có liên kết. Dùng với Socket Class của java.
* Datagram Socket (UDP): Có thể nhận dữ liệu không theo tình tự, trùng lặp. Dùng với DatagramSocket Class.
* Multicast Socket : cho phép dữ liệu được gửi đến nhiều bên nhận một lúc. Dùng với DatagramSocket Class.

Socket được hỗ trợ trên nhiều ngôn ngữ như C, Java, Pearl,  Python,….

### 2.3.1 Lập trình TCP Socket với Java



*Hình 2.15: Lập trình TCP Socket*

Đúng như tính chất của TCP chúng ta cần có liên kết 2 chiều trước khi server và client có thể trao đổi thông điệp với nhau.

Ban đầu, phía server tạo Socket được ràng buộc với một cổng (port number) để chờ nhận yêu cầu từ phía client.

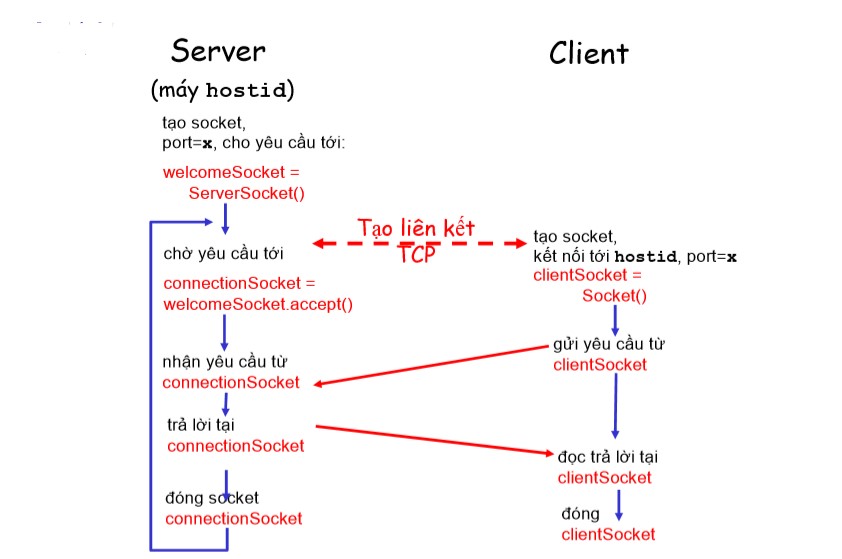
Tiếp đến phía client yêu cầu server bằng cách tạo một Socket TCP trên máy kèm với địa chỉ IP và port number của tiến tình tương ứng trên máy server. Khi client tạo Socket, client TCP tạo liên kết với server TCP và chờ chấp nhận kết nối từ server.

TCP cung cấp dịch vụ truyền dòng tin cậy và có thứ tự giữa client và server, giữa máy chủ và máy nhận chỉ có 1 địa chỉ IP duy nhất. Thêm vào đó, mỗi thông điệp truyền đi đều có xác nhận trả về.

Sau đây là một ví dụ ứng dụng đơn giản về lập trình TCP Socket với Java.

Miêu tả ứng dụng:

* Client đọc dòng văn bản nhập từ bàn phím người dùng , gửi tới server qua Socket
* Server đọc các dòng văn bản gửi từ Socket
* Server sẽ chuyển lại dòng văn bản kèm theo “Server accepted” tới phía client qua Socket
* Client đọc dòng văn bản từ socket và in ra dòng văn bản nhận được từ server



*Hình 2.16: Mô tả lập trình TCP Socket*

Chúng ta có thể thấy rằng mỗi phía server và client đều có 2 luồng dữ liệu, một luồng ra Socket để gửi thông điệp và một luồng vào từ Socket để nhận thông điệp, như vậy với mỗi bên mình có hai biến input và output (inFromServer, outToServer và inFromClient, outToClient).

# CHƯƠNG 3: TRIỂN KHAI CHƯƠNG TRÌNH CHAT SỬ DỤNG TCP SOCKET

## 3.1 Yêu cầu

* Client có giao diện đơn giản hay phức tạp tùy vào bạn.
* Server có giao diện đơn giản hay phức tạp tùy vào bạn

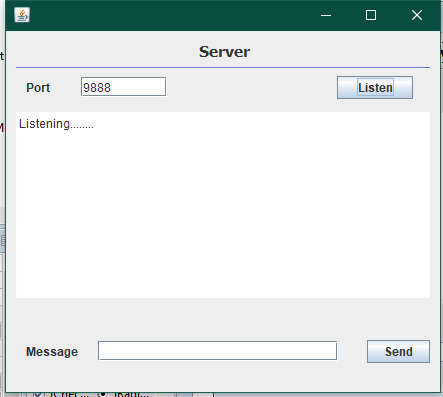
## 3.2 Môi trường triển khai

Ngôn ngữ lập trình: Java

Môi trường: Windows 10

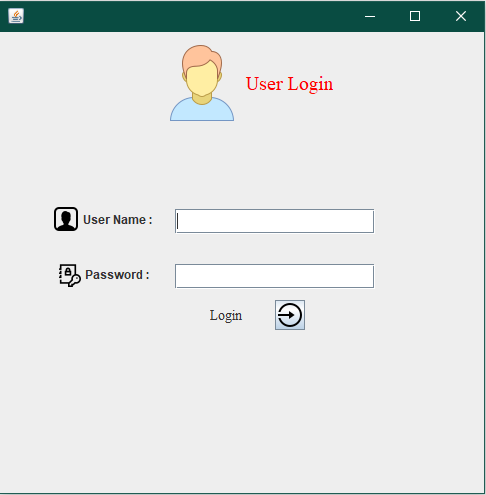
## 3.3 Triển khai chương trình Chat

### 3.3.1 Về phía Server

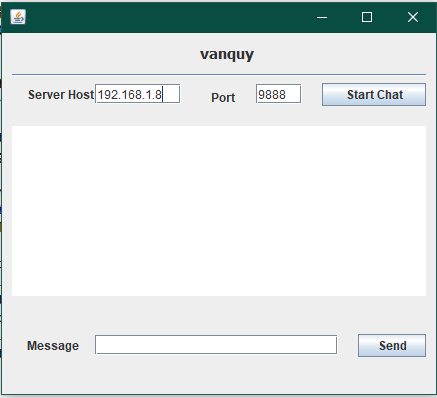


*Hình 3.1: Mở Server*

### 3.3.2 Về phía Client



*Hình 3.2 Minh hoạ Login*

**

*Hình 3.3 Minh hoạ chat Client*

# KẾT LUẬN

* Chương trình làm còn đơn giản, chưa đáp ứng được nhu cầu thiết thực
* Em sẽ cố gắng hoàn thiện và tiếp tục phát triển chương trình.
* Báo cáo môn Java Nâng Cao lần này, em đã trình bày các kiến thức cơ bản về TCP Socket và ứng dụng của nó. Từ kết quả thực nghiệm cho thấy, chương trình còn nhiều hạn chế. Để chương trình có thể được sử dụng vào trong thực tế, cần giải quyết những nhược điểm mà chương trình còn tồn tại, hoàn thành tiếp các chức năng chưa hoàn thiện, thực hiện các ý tưởng mới nhằm năng cao tốc độ, hiệu suất và độ chính xác của chương trình.

# TÀI LIỆU THAM KHẢO

1. <https://wiki.matbao.net/socket-la-gi-khai-niem-can-biet-ve-giao-thuc-tcp-ip-va-udp/>

2. <https://codelearn.io/sharing/lap-trinh-socket-co-voi-tcpip-ava>

3. <http://sontx.blogspot.com/2016/06/chatroom-su-dung-tcp-socket.html>