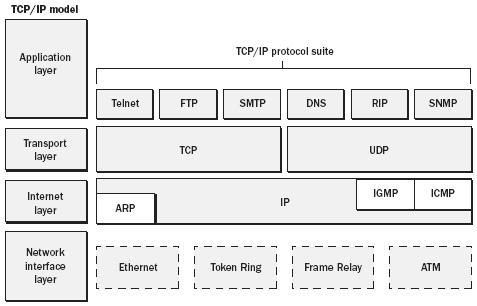
**II: Mô hình tổng thể hệ thống chat**

**1:Bộ giao thức TCP/IP**

* TCP/IP là một chuẩn phổ biến mà các mạng nội bộ và diện rộng có thể giao tiếp, cho phép các máy tính kết nối với nhau và cho các ứng dụng để gửi dữ liệu đi và về.
* Có 4 lớp trừu tượng khác nhau trong TCP/IP bao gồm:
  + Application
  + Transport
  + Internet
  + Network
* Các giao thức tầng Application
  + Trong TCP/IP, các giao thức tầng application hoạt động trong 3 tầng của mô hình OSI : session, presentation, application
  + Các giao thức : TELNET, DNS, HTTP, FTP, SMTP, …
* Các giao thức tầng Transport bao gồm : TCP, UDP
* Các giao thức tầng Internet (Network) bao gồm : IP, ARP, RARP, ICMP, IGMP
* Các giao thức tầng Network
  + TCP/IP không định nghĩa giao thức riêng ở tầng này.
  + Sử dụng những giao thức có sẵn: DSL, SONET, 802.11, Ethernet

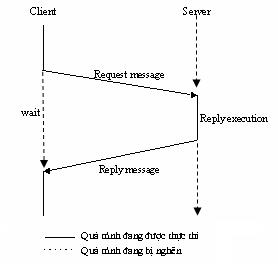
  
*Hình 1: Sơ đồ TCP/IP (http://hus.edu.vn)*

**2: Mô hình Client/Server**

-Trong mô hình này, chương trình ứng dụng được chia thành 2 thành phần:

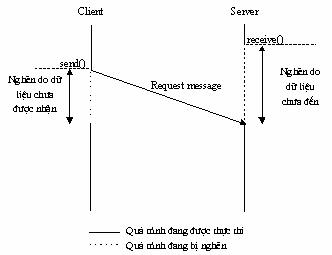
* Quá trình  chuyên cung cấp một số phục vụ nào đó, chẳng hạn: phục vụ tập tin, phục vụ máy in, phục vụ thư điện tử, phục vụ Web... Các quá trình này được gọi là các trình phục vụ hay **Server**.
* Một số quá trình khác có yêu cầu sử dụng các dịch vụ do các server cung cấp được gọi là các quá trình khách hàng hay **Client**.

-Việc giao tiếp giữa client và server được thực hiện dưới hình thức trao đổi các thông điệp (Message). Để được phục vụ, client sẽ gởi một thông điệp yêu cầu (Request Message) mô tả về công việc muốn server thực hiện. Khi nhận được thông điệp yêu cầu, server tiến hành phân tích để xác định công việc cần phải thực thi. Nếu việc thực hiện yêu cầu này có sinh ra kết quả trả về, server sẽ gởi nó cho client trong một thông điệp trả lời (Reply Message). Dạng thức (format) và ý nghĩa của các thông điệp trao đổi giữa client và server được qui định rõ bởi giao thức (protocol) của ứng dụng.



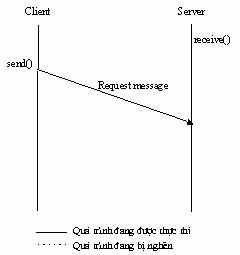
*Hình 2: Mô hình TCP/IP (https://voer.edu.vn )*

* Các chế độ trong giao tiếp: Quá trình giao tiếp giữa client và server có thể diễn ra theo hai chế độ là nghẽn (blocked) hay không nghẽn (Non blocked).
  + Chế độ nghẽn: Trong chế độ này, khi quá trình client hay server phát ra lệnh gởi dữ liệu,  (thông thường bằng lệnh send) , sự thực thi của nó sẽ bị tạm dừng cho đến khi quá trình nhận  phát ra lệnh nhận số dữ liệu đó (thường là lệnh receive). Tương tự cho trường hợp nhận dữ liệu, nếu  quá trình nào đó, client hay server, phát ra lệnh nhận dữ liệu, mà ở thời điểm đó chưa có dữ liệu gởi đến, sự thực thi của nó cũng tạm dừng cho đến khi có dữ liệu gởi đến.



*Hình 3: Chế độ nghẽn (voer.edu.vn)*

* + Chế độ không nghẽn: Trong chế độ này, khi quá trình client hay server phát ra lệnh gởi dữ liệu, sự thực thi của nó vẫn được tiếp tục mà không quan tâm đến việc có quá trình nào phát ra lệnh nhận số dữ liệu đó hay không. Tượng tự cho trường hợp nhận dữ liệu, khi quá trình phát ra lệnh nhận dữ liệu, nó sẽ nhận được số lượng dữ liệu hiện có (bằng 0 nếu chưa có quá trình nào gởi dữ liệu đến). Sự thực thi của quá trình vẫn được tiếp tục. Trong thực tế cần chú ý đến chế độ giao tiếp nghẽn khi lập trình, vì nó có thể dẫn đến trường hợp chương trình bị "treo" do số lần gởi và nhận giữ liệu không bằng nhau giữa hai bên giao tiếp.



*Hình 4: Chế độ giao tiếp không nghẽn (voer.edu.vn)*

**3: Cơ chế Socket trong Java**

Một socket là một điểm cuối của thông tin hai chiều liên kết giữa hai chương trình đang chạy trên mạng. Những lớp socket được dùng để đại diện cho kết nối giữa một chương trình client và một chương trình server. Trong Java gói Java.net cung cấp hai lớp Socket và ServerSocket để thực hiện kết nối giữa client và server.

Thông thường thì server sẽ chạy trên một máy đặc biệt và có một socket giới hạn trong 1 Portnumber đặc biệt.

Phía client: client được biết hostname của máy mà server đang chạy và port number mà server đang lắng nghe. Để tạo một yêu cầu kết nối client sẽ thử hẹn gặp server ở trên máy của server thông qua port number. Client cũng cần xác định chính nó với server thông qua local port number.

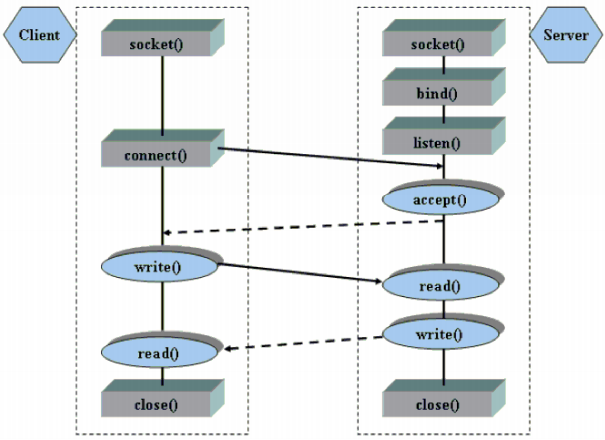


*Hình 5: Client gửi yêu cầu kết nối tới Server*

Nếu mọi thứ tốt đẹp thì server sẽ đồng ý kết nối. khi đồng ý kết nối thì server sẽ tạo ra một socket mới để nói chuyện với client và cũng tạo ra một socket khác để tiếp tục lắng nghe.



*Hình 6 : Server đồng ý kết nối và tiếp tục lắng nghe.*

****

*Hình 7: Mô hình truyền tin Socket (http://www.cit.ctu.edu.vn)*

* Một socket có thể thực hiện bảy thao tác cơ bản:
  + Kết nối với một máy ở xa
  + Gửi dữ liệu Nhận dữ liệu
  + Ngắt liên kêt
  + Gán cổng
  + Nghe dữ liệu đến
  + Chấp nhận liên kết từ các máy ở xa trên cổng đã được gán

[**http://www.toancau.vn/vietnam/4/5/32/giao-thuc-tcpip-la-gi**](http://www.toancau.vn/vietnam/4/5/32/giao-thuc-tcpip-la-gi)

[**http://hus.edu.vn/thcs/bai10.html**](http://hus.edu.vn/thcs/bai10.html)

[**https://voer.edu.vn/c/mo-hinh-client-server/761b0302/eedbe7c9**](https://voer.edu.vn/c/mo-hinh-client-server/761b0302/eedbe7c9)

[**http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf**](http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf)

[**http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf**](http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf)