TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**VIỆN CNTT - TT**

====o0o====



**BÁO CÁO LẬP TRÌNH MẠNG (LAB)**

***ĐỀ TÀI:***

**LẬP TRÌNH SOCKET VỚI**

**GAME ĐUA MOTO**

***GVHD*: Phạm Huy Hoàng**

***DANH SÁCH SINH VIÊN:*  
1. Đào Duy Nhẫn Việt Nhật 6A (IS1) 20111939**

**2. Lương Thị Thắm Việt Nhật 6A (IS1) 20112218**

**Hà Nội, 11/2015**

CONTENTS

[**CHƯƠNG I: TỔNG QUAN ĐỀ TÀI 2**](#_Toc436153770)

[**1. Giới thiệu đề tài 2**](#_Toc436153771)

[**2. Phương pháp triển khai 2**](#_Toc436153772)

[**CHƯƠNG II: CƠ SỞ LÝ THUYẾT 3**](#_Toc436153773)

[**1. Giới thiệu mô hình, ứng dụng Client-Server 3**](#_Toc436153774)

[**a. Kiến trúc Client-Server 3**](#_Toc436153775)

[**b. Ứng dụng theo mô hình Client-Server 5**](#_Toc436153776)

[**2. Lập trình với giao thức TCP, Multi-threading 7**](#_Toc436153777)

[**a.** **IP, cổng và Socket** 7](#_Toc436153778)

[**b.** **Lập trình TCP Socket** 8](#_Toc436153779)

[**CHƯƠNG III: PHÂN TÍCH VÀ XÂY DỰNG** 11](#_Toc436153780)

[**CHƯƠNG TRÌNH** 11](#_Toc436153781)

[**1.** **Phân tích yêu cầu:** 11](#_Toc436153782)

[**2.** **Mô tả hoạt động:** 11](#_Toc436153783)

[**3.** **Xây dựng chức năng và demo:** 11](#_Toc436153784)

[ **ĐÁNH GIÁ KẾT QUẢ** 13](#_Toc436153785)

[**TÀI LIỆU THAM KHẢO** 14](#_Toc436153786)

# **CHƯƠNG I: TỔNG QUAN ĐỀ TÀI**

1. **Giới thiệu đề tài**

Xây dựng chương trình: Lập trình socket với game đua moto:

* Server:
* Server mở cổng và listening
* Authenicate
* Game có 2 lựa chọn: with computer, và multi-players
* Lưu kết quả game của users
* Client:
* Login hoặc sign in
* Chọn mode chơi và map
* Play
* Chọn lưu kết quả hay không

1. **Phương pháp triển khai**

Dùng phương pháp lập trình mạng TCP Socket threaded Server để tạo kết nối Client - Server, tạo và biểu diễn các luồng I/O.

Client sẽ kết nối tới máy tính server và thực hiên việc chơi game.

# **CHƯƠNG II: CƠ SỞ LÝ THUYẾT**

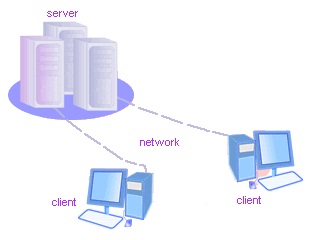
1. **Giới thiệu mô hình, ứng dụng Client-Server**
2. **Kiến trúc Client-Server**

* Tổng quan:

Kiến trúc client / server là một kiến trúc điện toán sản xuất-tiêu dùng mà các máy chủ hoạt động như các nhà sản xuất và khách hàng là người tiêu dùng. Các nhà máy chủ và cung cấp, dịch vụ điện toán chuyên sâu cao cấp cho khách hàng theo yêu cầu. Những dịch vụ này có thể bao gồm các ứng dụng truy cập, lưu trữ, chia sẻ tập tin, truy cập máy in và / hoặc truy cập trực tiếp đến khả năng tính toán nguyên của máy chủ. Kiến trúc client / server hoạt động khi máy tính của khách hàng sẽ gửi một yêu cầu tài nguyên hoặc quá trình để các máy chủ qua kết nối mạng, sau đó được xử lý và chuyển giao cho khách hàng. Một máy tính máy chủ có thể quản lý nhiều khách hàng cùng một lúc, trong khi một khách hàng có thể được kết nối với một số máy chủ tại một thời điểm, từng cung cấp một tập hợp khách nhau của các dịch vụ. Ở dạng đơn giản của nó, Internet cũng được dựa trên kiến trúc client / server mà các máy chủ web phục vụ nhiều người dùng đồng thời với các trang web và các dữ liệu hoặc trang web.

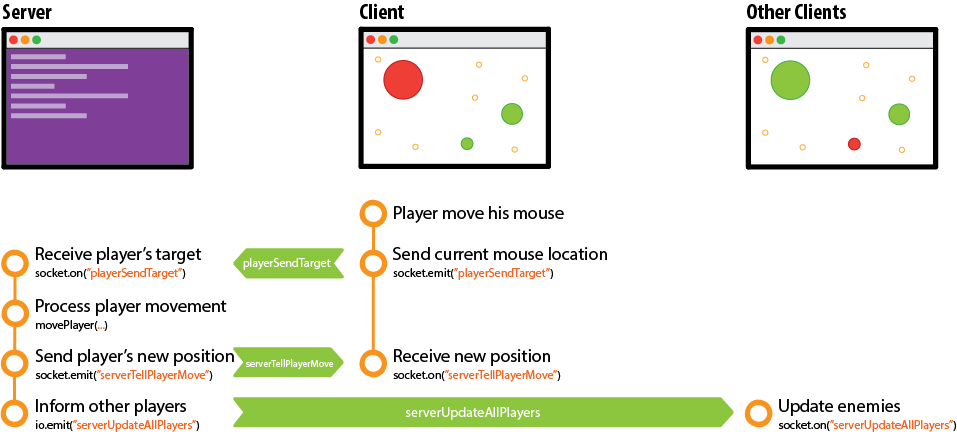
Kiến trúc Client-Server được sử dụng trong các hệ thống phân tán và bao gồm hai thành phần riêng biệt: Server đóng vai trò phục vụ cung cấp chức năng và Client đóng vai trò người tiêu thụ sử dụng các chức năng đó. Thông thường hai thành phần này kết nối với nhau qua mạng, với Client là bên chủ động tạo kết nối và gửi yêu cầu đến Server, trong khi Server thụ động lắng nghe và hồi đắp các yêu cầu đó.

Mô hình Client-Server đơn giản nhất gồm một Server phục vụ cho một hoặc nhiều Client đồng thời, còn gọi là kiến trúc hai lớp (2-Tier). Một ví dụ phổ biến nhất cho các ứng dung Client-Server là các chương trình chơi game desktop hoặc webgame đã quá thông dụng hiện nay.



Hình 1. Mô hình Client-Server

* Các ưu điểm của kiến trúc Client-Server:
* Quản lý tập trung: dữ liệu được lưu trữ tập trung trên Server thay vì nằm rải rác trênnhiều máy, giúp đơn giản hóa việc truy xuất và cập nhật dữ liệu.
* Dễ bảo trì: nhờ khả năng quản lý tập trung mà công việc bảo trì cũng trở nên nhẹ nhàng hơn vì phần lớn việc bảo trì chỉ cần thực hiện trên Server. Trong trường hợp hệ thống có nhiều Server với thiết bị dự phòng, quá trình bảo trì (như sửa chữa, thay thế Server) có thể diễn ra hoàn toàn trong suốt với phía Client.
* Bảo mật dữ liệu tập trung trên Server đồng nghĩa với việc kiểm soát dễ dàng hơn



Hình 2 . Mô hình hệ thống chơi game theo cấu trúc Client-Server

1. **Ứng dụng theo mô hình Client-Server**

* Hoạt động:

Các tiến trình Clients và Servers có thể chạy cùng một trạm (host) hoặc các trạm khác nhau và là các đối tượng logic tách biệt và liên lạc với nhau qua mạng để cùng thực hiện một công việc. Trong đó:

Server: quản lý nguồn tài nguyên, nhận request từ Clients để cung cấp và phân phối tài nguyên cho Clients.

Client: là chương trình giao tiếp với người sử dụng, có nhiệm vụ gửi request cho Server và thể hiện việc tiếp nhận tài nguyên đó.

* + Client:

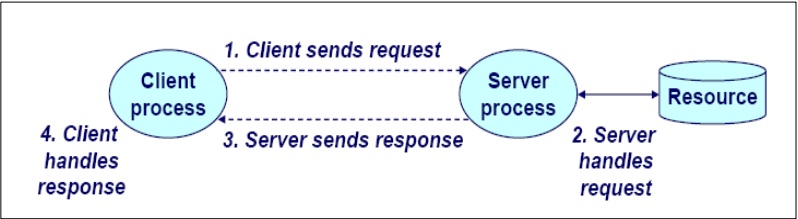
Khởi tạo liên lạc với Server trước.

Yêu cầu dịch vụ nào đó từ Server.

* + Server:

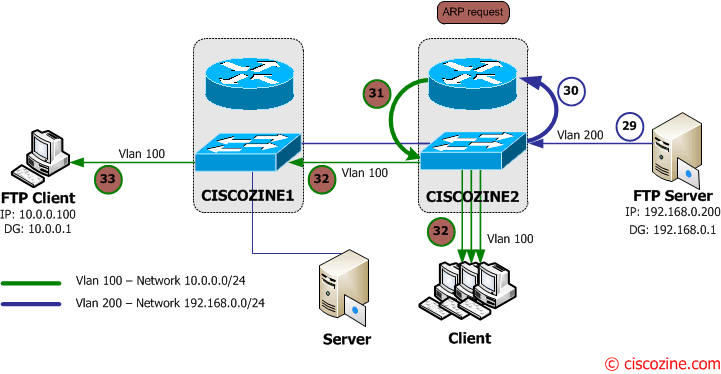
Chấp nhận yêu cầu và tạo kết nối với Client.

Tính toán rồi cung cấp dịch vụ theo yêu cầu từ Client



*Hình 3. Cách hoạt động của mô hình Client-Server.*

* Đặc trưng của mô hình ứng dụng Client-Server
* Hoạt động theo kiểu giao thức bất đối xứng (các client request một service và các server đáp ứng cho request này)
* Thể hiện quan hệ hai chiều giữa các Client và các Server.
* Client bắt đầu phiên hội thoại bằng cách yêu cầu dịch vụ
* Server sẵn sàng chờ các yêu cầu từ Client.



*Hình 4. Giao thức hoạt động bất đối xứng của các tiến trình Client-Server*

* Chia sẻ tài nguyên:

Một Server có thể chia sẻ tài nguyên cho nhiều Client cùng một lúc.

Server cũng có khả năng điều phối truy nhập các Client đến các tài nguyên dùng chung.

1. **Lập trình với giao thức TCP, Multi-threading**
2. **IP, cổng và Socket**

* IP: là một [địa chỉ](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%8Ba_ch%E1%BB%89_m%E1%BA%A1ng&action=edit&redlink=1) 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](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh) bằng cách sử dụng [giao thức](https://vi.wikipedia.org/wiki/Giao_th%E1%BB%A9c_Internet) TCP/IP, có thể là IPv4 hoặc IPv6.

Mỗi địa chỉ IP là duy nhất trong cùng một cấp mạng.

Một cách đơn giản hơn: IP là một [địa chỉ](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%8Ba_ch%E1%BB%89_lu%E1%BA%ADn_l%C3%BD&action=edit&redlink=1) của một máy tính khi tham gia vào mạng nhằm giúp cho các máy tính có thể chuyển thông tin cho nhau một cách chính xác, tránh thất lạc. Có thể coi địa chỉ IP trong mạng máy tính giống như địa chỉ nhà của bạn để nhân viên bưu điện có thể đưa thư đúng cho bạn chứ không phải một người nào khác.

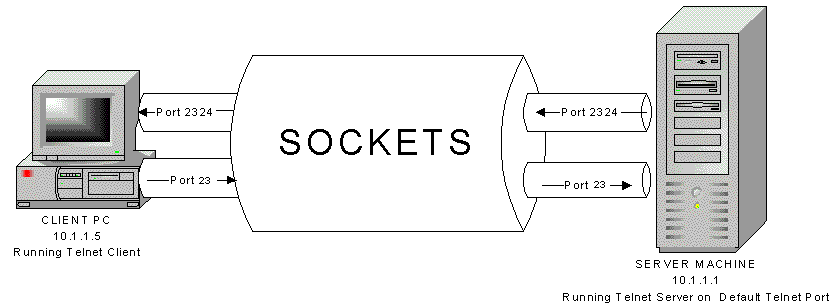
Bất kỳ thiết bị mạng nào—bao gồm [bộ định tuyến](https://vi.wikipedia.org/wiki/B%E1%BB%99_%C4%91%E1%BB%8Bnh_tuy%E1%BA%BFn), [bộ chuyển mạch mạng](https://vi.wikipedia.org/w/index.php?title=B%E1%BB%99_chuy%E1%BB%83n_m%E1%BA%A1ch_m%E1%BA%A1ng&action=edit&redlink=1), [máy vi tính](https://vi.wikipedia.org/wiki/M%C3%A1y_vi_t%C3%ADnh), máy chủ hạ tầng (như [NTP](https://vi.wikipedia.org/w/index.php?title=Giao_th%E1%BB%A9c_Th%E1%BB%9Di_gian_M%E1%BA%A1ng&action=edit&redlink=1), [DNS](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_t%C3%AAn_mi%E1%BB%81n), [DHCP](https://vi.wikipedia.org/wiki/DHCP), [SNMP](https://vi.wikipedia.org/w/index.php?title=Giao_th%E1%BB%A9c_Qu%E1%BA%A3n_tr%E1%BB%8B_M%E1%BA%A1ng_%C4%90%C6%A1n_gi%E1%BA%A3n&action=edit&redlink=1), v.v.), [máy in](https://vi.wikipedia.org/wiki/M%C3%A1y_in), máy [fax](https://vi.wikipedia.org/wiki/Fax) qua [Internet](https://vi.wikipedia.org/wiki/Internet), …

* Cổng (port): với một địa chỉ IP, chúng ta chỉ có thể xác định được một máy tính duy nhất trên mạng, tuy nhiên khi một máy tính chạy nhiều dịch vụ khác nhau thì chúng phải được phân biệt bởi khái niệm port. Ví dụ, Server A có dịch vụ web, DNS và FTP server, có địa chỉ IP là 210.245.126.14. Các máy tính khác khi muốn đến Server A thì cần đến địa chỉ IP (hay tên miền), nhưng để phân biệt dịch vụ là web, DNS hay FTP, cần xác định thêm port.
* Một socket là một đầu cuối của một sự truyền thông hai chiều, liên kết giữa hai chương trình chạy trên mạng.

Một socket được gắn với một số hiệu cổng (port), vì thế tầng giao vận có thể nhận biết ứng dụng mà dữ liệu được chuyển đến.

Socket cho phép thực hiện các hoạt động sau:

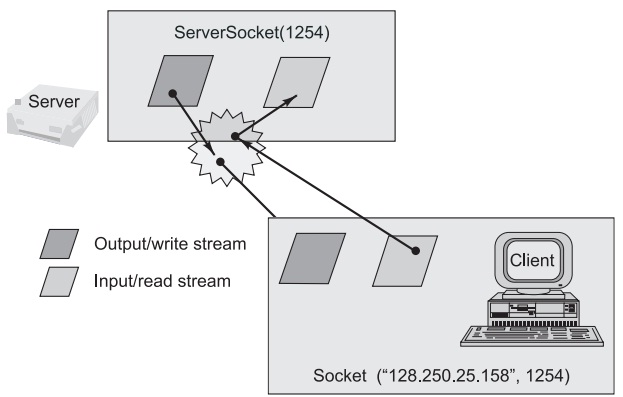
* Kết nối đến máy ở xa
* Gửi dữ liệu
* Nhận dữ liệu
* Đóng kết nối
* Gắn với một cổng
* Lắng nghe dữ liệu đến
* Chấp nhận kết nối từ máy ở xa trên cổng đã được gắn.



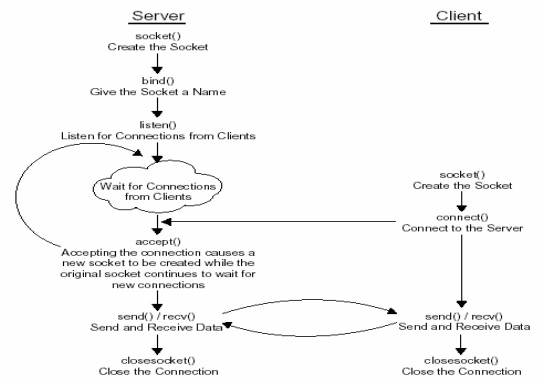
*Hình 5: Thiết lập đường dẫn cho kết nối hai chiều*

*giữa client và server.*

1. **Lập trình TCP Socket**



*Hình 6: Chương trình client-server có kết nối socket.*



*Hình 7: Các bước tạo một ứng dựng TCP.*

* Lập trình với chương trình server đa tuyến (ThreadServer)

Để cho phép nhiều client có thể kết nối đến server thì server phải là chương trình đa tuyến. Mối tuyến (thread) đảm nhận việc liên lạc với client. Nghĩa là khi có một client kết nối đến, chương trình server sinh ra một tuyến (thread) để điều khiển việc truyền thông với client.

Các bước tạo chương trình đa server đa tuyến:

* Tạo một lớp extends đến lớp Thread và ghi đè lên phương thức run():

class MyThread extends Thread {

public void run() {

// Nơi thực thi

}

}

* Tạo một đối tượng thread:

MyThread thr1 = new MyThread();

* Bắt đầu thực thi lớp Thread đã được tạo:

thr1.start();

* Một chương trình ví dụ minh họa việc tạo và gọi một đối tượng thread:

extending the standard Thread class. \*/

class MyThread extends Thread {

public void run() {

System.out.println(“ this thread is running ... ”);

}

}

class ThreadEx1 {

public static void main(String [] args ) {

MyThread t = new MyThread();

t.start();

}

}

# **CHƯƠNG III: PHÂN TÍCH VÀ XÂY DỰNG** **CHƯƠNG TRÌNH**

1. **Phân tích yêu cầu:**
   * Client: truy cập đến máy tính Server, login và play game
   * Server: đáp ứng yêu cầu:

* Gửi message cho client để thông báo khi client logged in, và sau khi kết thúc chơi.
* Lưu kết quả và hiển thị khi có yêu cầu
* Server có thể đáp ứng được multi-client cùng truy cập

1. **Mô tả hoạt động:**

* Client gửi yêu cầu tới Server.
* Server tiếp nhận yêu cầu kết nối từ Client, tất cả các dữ liệu giữa client và server sẽ được kết nối thông với nhau.
* Trao đổi dữ liệu giữa client tới server, từ đây server và client gởi dữ liệu xuyên suốt với nhau.

1. **Xây dựng chức năng và demo:**

* Mô tả chương trình Client:

Client gửi yêu cầu tới Server.

Client nhận kết nối từ Server và login hoặc sign in.

Client play game

**//cho code client vào đây**

* Mô tả chương trình Server:

Tiếp nhận yêu cầu kết nối từ Client.

Tạo Thread cho mỗi Client kết nối.

Thread kết nối để lấy dữ liệu theo yêu cầu Client.

ThreadClient nhận dữ liệu từ ThreadServer.

Gửi trả về cho Client.

Tiếp nhận dữ liệu từ Client.

**//cho code server vào đây**

* Chương trình game moto

**//Cho mấy hình ảnh chạy chương trình tại đây**

* **ĐÁNH GIÁ KẾT QUẢ**

Trong khuôn khổ đề tài chúng em đã cơ bản làm rõ một số vấn đề cơ bản liên quan đến môn học Lập Trình Mạng, và đã xây dựng thành công một chương trình demo nhằm làm rõ các vấn đề lí thuyết. Đã đạt được một số kết quả như sau :

* Tìm hiểu được cách xây dựng một chương trình với giao thức TCP/IP theo mô hình client – server.
* Tìm hiểu được các khái niệm cơ bản về lập trình mạng, phân tầng mạng, cách lấy dữ liệu từ máy client….
* Ứng dụng Socket và đa luồng trong ngôn ngữ Java để xây dựng một chương trình theo mô hình client – server.

Tuy nhiên về mặt ứng dụng đồ án vẫn còn rất nhiều điểm hạn chế như chương trình thiếu xót nhiều để có thể đạt đến được sản phẩm thương mại. Bên cạnh đó việc quản lí nhóm, quản lí mã nguồn vẫn còn diễn ra rất thủ công chưa thực sự áp dụng đúng quy trình của công nghệ phần mềm. Tuy vậy chúng em xin kết thúc đồ án lập trình mạng với đề tài “Lập trình Socket với game đua moto” tại đây. Chúng em xin được cảm ơn sự hướng dẫn tận tình của thầy “Phạm Huy Hoàng” trong suốt quá trình

**TÀI LIỆU THAM KHẢO**

1. Wikipedia, “Client-server model”, <https://en.wikipedia.org/wiki/Client%E2%80%93server_model>
2. Oracle documentation, “Java/Swing”, “Java DB - Reference ”, “Java API”
3. TS.Huỳnh Công Pháp, Giáo trình lập trình mạng(2004-2006), Đại Học Đà Nẵng