

UNIVERSITY OF SCIENCE – HO CHI MINH



CITY FACULTY OF INFORMATION  
TECHNOLOGY

# PROJECT: SOCKET

TEACHERS

LÊ GIANG THANH

NGUYỄN THỊ THANH HUYỀN

STUDENTS

-THIRD TERM-  
COURSE: 2020 - 2021

## PHẦN 1: CÁCH XÂY DỰNG PROTOCOL

## 1. Kịch bản giao tiếp của chương trình

### a. Giao thức Register:

Bước 1: Client → Server

- Client gửi Server 1 gói tin flag có chứa flag REGISTER
- Sau đó, gửi tiếp liên tiếp 2 gói tin Data lần lượt chứa các dữ liệu:
  - Encrypt là kí tự Y/N;
  - Password – là một chuỗi kí tự. + Username – là một chuỗi kí tự

Bước 2: Server → Client

- Server nhận 1 gói tin Flag và 2 gói tin Data từ Client.
- Gửi lại Client gói tin Flag chứa kết quả register:
  - REGISTER\_SUCCESS – nếu thành công.
  - REGISTER\_FAIL – nếu thất bại vì username đã tồn tại.

### b. Giao thức Login:

Bước 1: Client → Server

- Client gửi Server 1 gói tin Flag có chứa flag LOGIN
- Sau đó, gửi tiếp 2 gói tin Data lần lượt chứa các dữ liệu:
  - Encrypt là kí tự Y/N;
  - Password – là một chuỗi kí tự. + Username – là một chuỗi kí tự

Bước 2: Server → Client

- Server nhận 1 gói tin Flag và 2 gói tin Data từ Client.
- Gửi lại Client gói tin Flag chứa kết quả login:
  - LOGIN\_SUCCESS – nếu thành công.
  - LOGIN\_FAIL – nếu thất bại vì username không tồn tại hoặc password không trùng khớp.
  - USER\_FAIL\_ONLINE – nếu thất bại vì username đó đang được Client khác sử dụng để Online.

### c. Giao thức Đổi mật khẩu

Bước 1: Client → Server

- Client gửi thông điệp cho Server với Flag là CHANGE\_PASSWORD. Rồi gửi tiếp 2 gói tin là
  - -Current password là chuỗi kí tự
  - - Encrypt là kí tự Y/N;

Bước 2: Server → Client

- Server nhận gói tin và kiểm tra Current password và trả về các kết quả
  - - Current password đúng qua bước 3
  - - Current password gửi gói có flag CHANGE\_FAIL quay lại bước 1

Bước 3: Client → Server

- Client gửi gói tin chứa mật khẩu mới cho Server
  - New\_password là chuỗi kí tự

#### d. Giao thức Check User:

Bước 1: Client → Server

- Client gửi Server 1 gói tin Flag có chứa flag CHECK\_USER
  - -option – là một chuỗi kí tự chứa lệnh –show\_
  - -Username – là một chuỗi kí tự

Bước 2: Server → Client

- Server nhận 1 gói tin Flag từ Client.
- Gửi lại Client gói tin Flag chứa kết quả Check\_User:
  - CHECK\_USER option– hiển thị thông tin user cần tìm kiếm.
  - FORMAT ERROR – không tồn tại hoặc cú pháp tìm kiếm không đúng định dạng
  - NOT EXIST – thông tin user không tồn tại hoặc tài khoản không tồn tại

#### e. Giao thức Setup Info:

Bước 1: Client → Server

- Client gửi Server 1 gói tin Flag có chứa flag SETUP\_INFO
  - -option – là một chuỗi kí tự lệnh cho việc setup info tương ứng
  - -Username – là một chuỗi kí tự

Bước 2: Server → Client

- Server nhận 1 gói tin Flag từ Client.
- Gửi lại Client gói tin Flag chứa kết quả Check\_User:
  - SETUP\_INFO – nếu setup thành công

## f. Giao thức Logout của Client

Bước 1: Client → Server

- Client gửi thông điệp cho Server với Flag là LOGOUT,”. Client gọi hàm shutdown đóng socket (closesocket) luôn., tức Client không gửi dữ liệu cho Server nữa.

Bước 2: Server → Client

- Server nhận thông điệp và xử lý, nhận thấy Flag là LOGOUT nên bỏ Client ra khỏi danh sách các Client online, sau đó loại client này ra khỏi danh sách user online .
- Server gửi tiếp cho Client muốn logout một thông điệp với Flag là LOGOUT\_CLIENT.
- Server đóng socket và thread của client logout này luôn với hàm closesocket, tức không nhận dữ liệu từ Client muốn logout này nữa.

## 2. Cấu trúc chương trình:

- Giao thức: TCP

```
ZeroMemory(&hints, sizeof(hints));
hints.ai_family = AF_UNSPEC;
hints.ai_socktype = SOCK_STREAM;
hints.ai_protocol = IPPROTO_TCP;
```

- Chức năng kết nối từ client đến server:

```
while (1)
{
    SOCKET incoming = INVALID_SOCKET;
    incoming = accept(server_socket, NULL, NULL);

    if (incoming == INVALID_SOCKET) continue;

    //Reset the number of clients
    num_clients = -1;
```

- Nhận yêu cầu và chấp nhận tài khoản client vừa đăng ký lên server.

```
* int process_client (client_type&new_client, std::vector<client_type>& client_array, std::thread
```

&thread)

- Khi server chấp nhận kết nối đến client thì hàm này có chức năng tạo thread để client và server có thể giao tiếp với nhau.

```
int ReceivedMessage(client_type& new_client)
{
    vector<string> user_online;
    bool Open = true;
    bool encrypt = true;
    // Game play
    vector<vector<int>> Ships;
    vector<vector<int>> pos;
```

- Giúp nhận tất cả các gói tin từ server đến client

```
void Running() {
    WSADATA wsa_data;
    struct addrinfo* result = NULL, * ptr = NULL, hints;
    string sent_message = "";
    client_type client = { INVALID_SOCKET, -1, "" };
    int iResult = 0;
```

- Tạo ra 1 socket để gửi gói tin đó đi.

- Các hàm chính còn lại như:

+void Login(SOCKET& client,bool encrypt)

+void Register( SOCKET& client, bool encrypt)

## b. Chức năng đăng nhập và quản lý người dung.

- Khi thực hiện mở client và server lên thì client lúc đó sẽ đăng ký 1 tài khoản, lúc đó server sẽ chấp nhận tài khoản đó và lưu lại thông tin người dung cho các tính năng sau:

Ví dụ: check\_user[-option][username]: khi đó chương trình sẽ gửi đến server 1 Flag như sau:

```
}
else if (s.find("CHECK_USER") != -1) { // Fix by D
    return 4;
}
```

kèm theo option, sau đó client sẽ gửi đến server, khi server nhận gói tin thì sẽ kiểm tra xem flag này thuộc nhóm nào, đủ thông tin username và option thì sẽ tiến hành truy xuất database từ bước đăng ký tài khoản. Tương tự đối với chức năng setup\_info, khi nhận được flag sẽ gửi yêu cầu đến server để truy xuất database và cập nhật database mới.

- Các hàm còn lại bao gồm:

+ void Change\_Password(SOCKET &client, bool encrypt)

+ void Check\_User(SOCKET &client, bool encrypt)

+ void SetUp\_Info(SOCKET &client, bool encrypt)

## c. Chức năng tham gia trò chơi:

\* Tạo room:

```
void CREATE_ROOM(SOCKET client, string sent_message, bool encrypt,int& P2_ID) {
    Clean(1);
    Choose_user_play_with(client, sent_message, encrypt);
    P2_ID = Get_P2_ID(sent_message);
}
```

-Chọn ID và gán flag CREATE\_ROOM vào và gửi cho server. Sau đó server sẽ tháo flag và lấy ID, dựa vào trên ID người gửi và gửi flag INVITE đến để hỏi xem người gửi có muốn tham gia không. Nếu chấp nhận thì server sẽ được gửi flag ACCEPT cho server từ lúc này trở về sau thì server sẽ tạo liên kết giữa 2 client và server.

- Server sẽ gửi thông báo tạo phòng thành công và bắt đầu game bằng flag COMPETITOR\_ACCEPT, còn nếu từ chối thì sẽ gửi flag REJECT, client sẽ nhận được flag này và trả về màn hình.

\* Upload map:

```

void Upload_Map() {
    cout << setw(72) << "====UPLOAD_MAP====" << endl;
    cout << endl;
    cout << setw(72) << "(Namefile.txt)      " << endl;
    cout << endl;
    cout << setw(72) << "=>>";
}
// USER MENU

```

- Server sẽ nhận gói tin flag UPLOAD\_MAP+ tên của file.txt, server sẽ tháo flag UPLOAD\_MAP và thay bằng flag REV\_SEND\_MAP và gửi cho client P2, tiếp đó P2 sẽ nhận diện gói tin chứa flag REV\_SEND\_MAP nên nó sẽ gửi cho server gói tin flag UPLOAD\_MAP+ tên của file.txt

- Server nhận được gói tin có flag UPLOAD\_MAP 2 lần nên biết rằng cả 2 người chơi đều đã upload map tàu của mình lên, nên nó gửi map tiếp P1 với flag UPLOAD\_MAP. Sau khi client P1 nhận được gói tin này nó sẽ gỡ flag và lấy file.txt về, tương tự với P2.

\* Cách tấn công:

```

void battle(int** ships_ptr, char** pseudo_gui_ptr, int n, struct coord x, int* count, int* miss)

```

- Hàm có chức năng cho biết người chơi có bắn trúng tàu của đối phương không với các tham số truyền vào size của map, tọa độ, các biến đếm số lần bắn trúng và số lần bắn hụt. Mỗi lần bắn sẽ nhập 2 thông số bao gồm tọa độ x và tọa độ y theo



# Bảng so sánh gói tin

## Dữ liệu mã hóa

## Dữ liệu chưa mã hóa

### Chức năng Login

759 583.549242	172.16.20.100	172.16.20.1	TCP	76 53863 + 3504 [PSH, ACK] Seq=52 Ack=18 Win=131328 Len=22
<div>&gt; Frame 759: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 53863, Dst Port: 3504, Seq: 52, Ack: 18, Len: 22</div> <div>▼ Data (22 bytes)</div> <div>Data: 3334363437353633363133313332333333333433353336</div> <div>[Length: 22]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 3e 1b cb 40 00 80 06	5e 69 ac 10 14 64 ac 10	-> @... ^i...d..	
0020	14 01 d2 67 0d b0 4a f0	ed ca 31 e5 76 00 50 18	--g-J-...l-v-P-	
0030	02 01 2c 37 00 00 33 34	36 34 37 35 36 33 36 31	--7-34 64756361	
0040	33 31 33 32 33 33 33 34	33 35 33 36	31323334 3536	

797 721.183281	172.16.20.100	172.16.20.1	TCP	65 53865 + 3504 [PSH, ACK] Seq=7 Ack=2 Win=1051136 Len=11
<div>&gt; Frame 797: 65 bytes on wire (520 bits), 65 bytes captured (520 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 53865, Dst Port: 3504, Seq: 7, Ack: 2, Len: 11</div> <div>▼ Data (11 bytes)</div> <div>Data: 3464756361313233343536</div> <div>[Length: 11]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 33 1b d9 40 00 80 06	5e 66 ac 10 14 64 ac 10	-> @... ^f...d..	
0020	14 01 d2 69 0d b0 8f 9d	e5 9e 96 b6 7e 52 50 18	--i.....RP-	
0030	10 0a 0d 71 00 00 34 64	75 63 61 31 32 33 34 35	--q-4d uca12345	
0040	36		6	

### Đổi mật khẩu

#### Mật khẩu hiện tại

848 885.033606	172.16.20.100	172.16.20.1	TCP	66 53866 + 3504 [PSH, ACK] Seq=45 Ack=15 Win=131328 Len=12
<div>&gt; Frame 848: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 53866, Dst Port: 3504, Seq: 45, Ack: 15, Len: 12</div> <div>▼ Data (12 bytes)</div> <div>Data: 333133323333333433353336</div> <div>[Length: 12]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 34 1b e7 40 00 80 06	5e 57 ac 10 14 64 ac 10	-4-@... ^M...d..	
0020	14 01 d2 6a 0d b0 da 9d	4e c1 69 b5 5c 9e 50 18	--j.....N i \P-	
0030	02 01 2a 36 00 00 33 31	33 32 33 33 33 34 33 35	--*6-31 32333435	
0040	33 36	36		

#### Mật khẩu đã đổi

855 961.042738	172.16.20.100	172.16.20.1	TCP	70 53866 + 3504 [PSH, ACK] Seq=57 Ack=15 Win=131328 Len=16
<div>&gt; Frame 855: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 53866, Dst Port: 3504, Seq: 57, Ack: 15, Len: 16</div> <div>▼ Data (16 bytes)</div> <div>Data: 36343735363336313665333133323333</div> <div>[Length: 16]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 30 1b e8 40 00 80 06	5e 52 ac 10 14 64 ac 10	-8-@... ^R...d..	
0020	14 01 d2 6a 0d b0 da 9d	4e cd 69 b5 5c 9e 50 18	--j.....N i \P-	
0030	02 01 b3 92 00 36 34	37 35 36 33 36 31 36 65	.....64 7563616e	
0040	33 31 33 32 33 33	313233		

885 1099.167529	172.16.20.100	172.16.20.1	TCP	60 53866 + 3504 [PSH, ACK] Seq=133 Ack=43 Win=131328 Len=6
<div>&gt; Frame 885: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 53866, Dst Port: 3504, Seq: 133, Ack: 43, Len: 6</div> <div>▼ Data (6 bytes)</div> <div>Data: 313233343536</div> <div>[Length: 6]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 2e 1b f1 40 00 80 06	5e 53 ac 10 14 64 ac 10	--@... ^S...d..	
0020	14 01 d2 6a 0d b0 da 9d	4f 19 69 b5 5c ba 50 18	--j.....O i \P-	
0030	02 01 c3 61 00 31 32	33 34 35 36	...a-12 3456	

898 1187.266339	172.16.20.100	172.16.20.1	TCP	62 53866 + 3504 [PSH, ACK] Seq=139 Ack=43 Win=131328 Len=8
<div>&gt; Frame 898: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 53866, Dst Port: 3504, Seq: 139, Ack: 43, Len: 8</div> <div>▼ Data (8 bytes)</div> <div>Data: 647563616e313233</div> <div>[Length: 8]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 30 1b f2 40 00 80 06	5e 50 ac 10 14 64 ac 10	-0-@... ^P...d..	
0020	14 01 d2 6a 0d b0 da 9d	4f 1f 69 b5 5c ba 50 18	--j.....O i \P-	
0030	02 01 f4 ba 00 00 64 75	63 61 6e 31 32 33	.....du can123	

### Chức năng Register

1784 2576.244626	172.16.20.100	172.16.20.1	TCP	80 51023 + 3504 [PSH, ACK] Seq=10 Ack=2 Win=131328 Len=26
<div>&gt; Frame 1784: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 51023, Dst Port: 3504, Seq: 10, Ack: 2, Len: 26</div> <div>▼ Data (26 bytes)</div> <div>Data: 3336363136653631366533313332333333333433353336</div> <div>[Length: 26]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 42 1c 54 40 00 80 06	5d dc ac 10 14 64 ac 10	-8-T@... ]....d..	
0020	14 01 c7 4f 0d b0 56 40	94 50 41 10 50 0d 50 18	--O-V@-PA-P-P-	
0030	02 01 2f ce 00 00 33 36	36 31 36 65 36 31 36 65	--/-36 616e616e	
0040	36 31 36 65 33 31 33 32	33 33 33 34 33 35 33 36	616e3132 33343536	

1810 2673.563002	172.16.20.100	172.16.20.1	TCP	67 51030 + 3504 [PSH, ACK] Seq=10 Ack=2 Win=1051136 Len=13
<div>&gt; Frame 1810: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface \Device\NPF_{C867E341-B08C-4106-814E-8AE3FFD287DA}, id 0</div> <div>&gt; Ethernet II, Src: VMware_c0:00:01 (00:50:56:c0:00:01), Dst: VMware_c1:15:1e (00:0c:29:c1:15:1e)</div> <div>&gt; Internet Protocol Version 4, Src: 172.16.20.100, Dst: 172.16.20.1</div> <div>&gt; Transmission Control Protocol, Src Port: 51030, Dst Port: 3504, Seq: 10, Ack: 2, Len: 13</div> <div>▼ Data (13 bytes)</div> <div>Data: 36616e616e616e313233343536</div> <div>[Length: 13]</div>				
0000	00 0c 29 c1 15 1e 00 50	56 c0 00 01 08 00 45 00	-->....P V....E-	
0010	00 35 1c 5e 40 00 80 06	5d df ac 10 14 64 ac 10	-5-@... ]....d..	
0020	14 01 c7 56 0d b0 03 74	94 46 72 fe 1d 4c 50 18	--V...t-Fr-LP-	
0030	10 0a 04 66 00 00 36 61	6e 61 6e 61 6e 31 32 33	...f-6a nanan123	
0040	34 35 36	456		

So sánh và cho biết khác biệt giữa hai dữ liệu :

-Độ dài dữ liệu mã hóa nhiều hơn dữ liệu chưa mã hóa.

-Dữ liệu chưa mã hóa có thể đọc và hiểu được thông tin , dữ liệu mã hóa là dạng thông tin không hiểu được

-Việc mã hóa dữ liệu có thể bảo vệ toàn thông tin, dữ liệu của mình tốt hơn, an toàn trong việc truyền dữ liệu còn chưa mã hóa dữ liệu thì ngược lại.

- Việc mã hóa dữ liệu sẽ không thể nào ngăn việc dữ liệu có thể bị đánh cắp, nhưng nó sẽ ngăn việc người khác có thể đọc được nội dung của tập tin đó, vì nó đã bị biến sang thành một dạng ký tự khác, hay nội dung khác.