****

**BÁO CÁO ĐỒ ÁN 2:**

***Lập trình socket***

*Sinh viên thực hiện:*

*(Leader)Hồ Nguyễn Huy Hoàng-18127006*

*Nguyễn Trung Kiên-18127010*

*Lê Hồng Quang-18127190*

***Phần 1:***Cách xây dựng protocol

1. Kịch bản chương trình:

1.Nội dung và yêu cầu đồ án:

Sử dụng lập trình socket và multithread xây dựng protocol TCP gồm các chức năng cơ bản sau:

-Chức năng kết nối từ client đến server

-Chức năng đăng nhập và quản lý thông tin người dùng

-Chức năng upload,download dữ liệu

-Chức năng chat với người dùng khác

Và dùng chương trình wireshark bắt gói tin trao đổi giữa server và client theo mô hình mạng.

2.Phân tích các chức năng:

-Chức năng đăng nhập và quản lý thông tin người dùng:Client gửi username và password cho server.Server sau khi nhận sẽ kiểm tra cơ sở dữ liệu hiện tại.Nếu tài khoản tồn tại ,gửi mã đăng nhập thành công về cho client, nếu chưa tồn tại , gửi mã đăng nhập thất bại cho client. Khi đăng ký tài khoản mới, nếu tài khoản đã tồn tại gửi mã đăng ký thất bại cho client còn nếu thành công, cập nhật lại cơ sở dữ liệu cũng như danh sách client đang online và gửi mã đăng ký thành công về cho client.

-Chức năng Chat với 1 user khác:

+Client gửi username của đối tượng muốn chat lên server, server sẽ kiểm tra user có online hay không.Nếu có thì gửi mã lện thành công về cho client, client sẽ tạo phòng chat hai người.Ngược lại client sẽ nhận được thông báo thất bại

+Trong khi chat, client này gửi tin nhắn chat cho server, server sẽ gửi lại cho client kia tin nhắn đó.

+Khi một trong hai client muốn rời khỏi phòng chat, phải gửi yêu cầu cho server, server sẽ gửi thông báo kết thúc quá trình chat cho client còn lại.

-Chức năng gửi nhận dữ liệu:

Client gửi yêu cầu upload lên server,server nhận yêu cầu upload file từ Client.Sau đó server sẽ nhận tập tin và lưu vào thư mục hiện hành trên server. Với quá trình download, server nhận yêu cầu download file từ client sau đó kiểm tra file có tồn tại hay không. Nếu có ,cho phép download từ thư mục hiện hành trên server, nếu không gửi thông báo lỗi không tồn tại file cho client.

1. Cấu trúc chương trình:

+Giao thức thực hiện:TCP

+Các hàm chính và kiểu cấu trúc:

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

1. Client:

Class Socket\_Controller

->Các hàm xử lý:

+ public TcpListener Create\_TcpListener(int Port\_Number, string default\_IP):

.Ý nghĩa: khởi tạo listener cho server

.Tham số: Port\_Number và default\_IP , các tham số cơ bản để gửi và nhận dữ liệu giữa client và server

+ public void Client\_Connect\_Server(ref TcpClient client, int Port\_Number, string default\_IP):Hàm kết nối đến server bằng TCPClient- đối tượng để lắng nghe trên cổng cục bộ

+ public Socket Accept\_connection(ref TcpListener listener):Lắng nghe và khởi tạo socket để kết nối đến client

+ public bool Close\_connection(ref Socket sck, ref TcpListener listener), Close\_TcpClient(ref TcpClient client), Close\_Stream(ref Stream stream):Đóng kết nối đến client, tcpclient và luồng

+ public Stream Get\_Stream(ref TcpClient client): lấy luồng giao tiếp

+ public NetworkStream Create\_NetworkStream(ref Socket sck): tạo luồng network để sử dụng đối tượng streamreader và streamwriter

+ public bool Enable\_or\_Disable\_AutoFlush(ref StreamWriter writer, bool set): thuộc tính AutoFlush của streamwriter để tự động gửi dữ liệu mà không cần đợi bộ đệm đầy

+ public bool StreamWriter\_Send\_Message(string s, ref StreamWriter writer): thực hiện gửi tin nhắn bằng stream writer

+ public string StreamReader\_Receive\_Message(ref StreamReader reader): thực hiện nhận tin nhắn bằng stream writer

Class Client\_Controller:

+Các thuộc tính của lớp:

//Server Controller

private TcpClient client;

private Stream stream;

private StreamReader reader;

private StreamWriter writer;

//Account Info

private string UserName;

private string FullName;

private string DOB;

private string Note;

//

private string chat\_with;

+ public TcpClient Client\_Connection: nhận hoặc lưu kết nối client để control

+ public Stream Stream\_Connection: nhận hoặc lưu luồng (sending/receiving data) để control

+ public StreamReader StreamReader/Writer\_Connection: nhận hoặc lưu streamreader/writer để nhận/gửi dữ liệu

1. Server:

Class Socket\_Controller:

+public TcpListener Create\_TcpListener(int Port\_Number, string default\_IP):khởi tạo listener cho server với các tham số cơ bản

+ public void Client\_Connect\_Server(ref TcpClient client, int Port\_Number, string default\_IP):Hàm kết nối đến Server bằng TcpClient

+ public Socket Accept\_connection(ref TcpListener listener): lắng nghe và khởi tạo socket để kết nối đến client

+ public bool Close\_connection(ref Socket sck, ref TcpListener listener), Close\_TcpClient(ref TcpClient client), Close\_Stream(ref Stream stream):Đóng kết nối đến client, tcpclient và luồng

+ public Stream Get\_Stream(ref TcpClient client): lấy luồng giao tiếp

+ public NetworkStream Create\_NetworkStream(ref Socket sck): tạo luồng network để sử dụng đối tượng streamreader và streamwriter

+ public bool Enable\_or\_Disable\_AutoFlush(ref StreamWriter writer, bool set): thuộc tính AutoFlush của streamwriter để tự động gửi dữ liệu mà không cần đợi bộ đệm đầy

+ public bool StreamWriter\_Send\_Message(string s, ref StreamWriter writer): thực hiện gửi tin nhắn bằng stream writer

+ public string StreamReader\_Receive\_Message(ref StreamReader reader): thực hiện nhận tin nhắn bằng stream writer

Class Client:

+Các thuộc tính cơ bản:

private Socket socket;

private NetworkStream stream;

private StreamReader reader;

private StreamWriter writer;

private bool is\_connected;

+ public Socket socket\_connection

+ public NetworkStream stream\_connection

+public StreamReader reader\_connection

+ public StreamWriter writer\_connection

+public bool is\_connected\_to\_client

Class User:

+Các thuộc tính cơ bản:

//Info

private string User\_Name;

private string Password;

private string Full\_Name;

private string DOB;

private string Note;

private string IP\_Client;

//Online

private bool is\_Login;

private bool is\_Chat;

private string chatwith\_user;

Class Server\_Controller:

+Các thuộc tính cơ bản:

//Define

private string DATABASE\_PATH;

private const string DATABASE\_FILE\_NAME = "\\database.txt";

//

private TcpListener listener;

private List<Client> client\_controller;

private List<User> userList = new List<User>();

+ public void Load\_Database():Tải danh sách user

+public void Update\_Database():Cập nhật danh sách người dùng

+ public bool change\_Profile(string s, string IP\_Enpoint):thay đổi thông tin người dùng

+ public bool Check\_Login(string s, string IP\_Endpoint):đăng nhập tài khoản

+ private bool Check\_Reg(string user\_name):đăng ký tài khoản

+public bool Reg(string s):kiểm tra các điều kiện khi đăng ký

+public bool Logout(string IPEndpoint):đăng xuất tài khoản

+public void start\_chat(string userName):bắt đầu chat

+ public void stop\_chat(string userName):dừng chat

+public bool request\_chat(string IP\_sender, string receiver):gửi yêu cầu chat

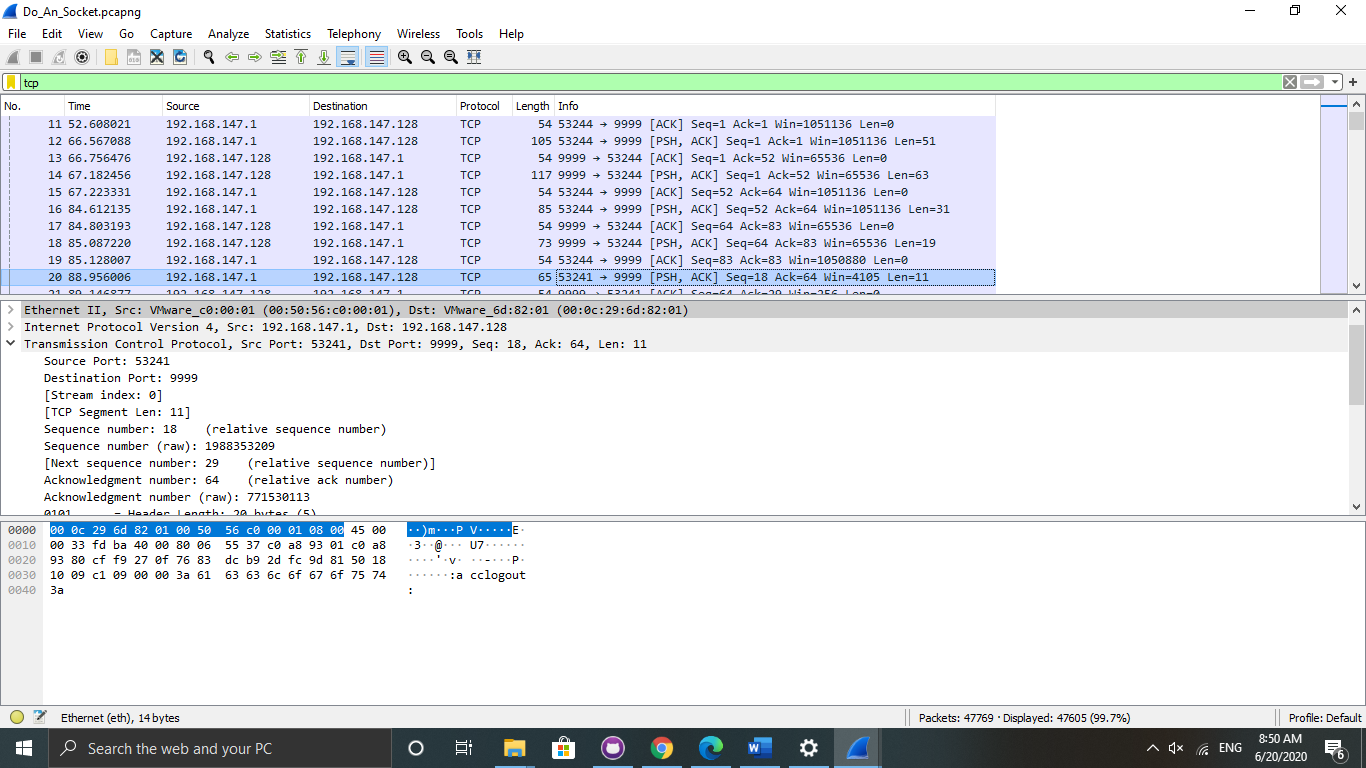
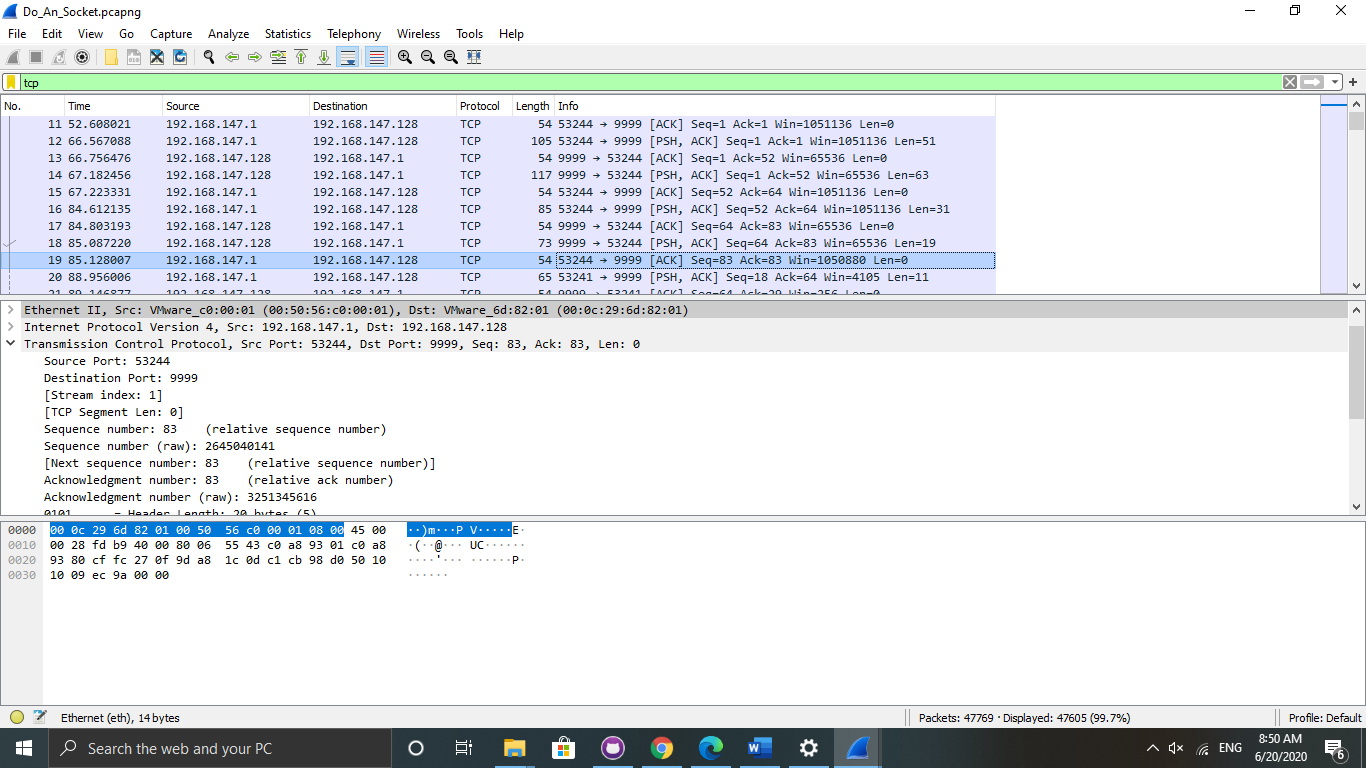
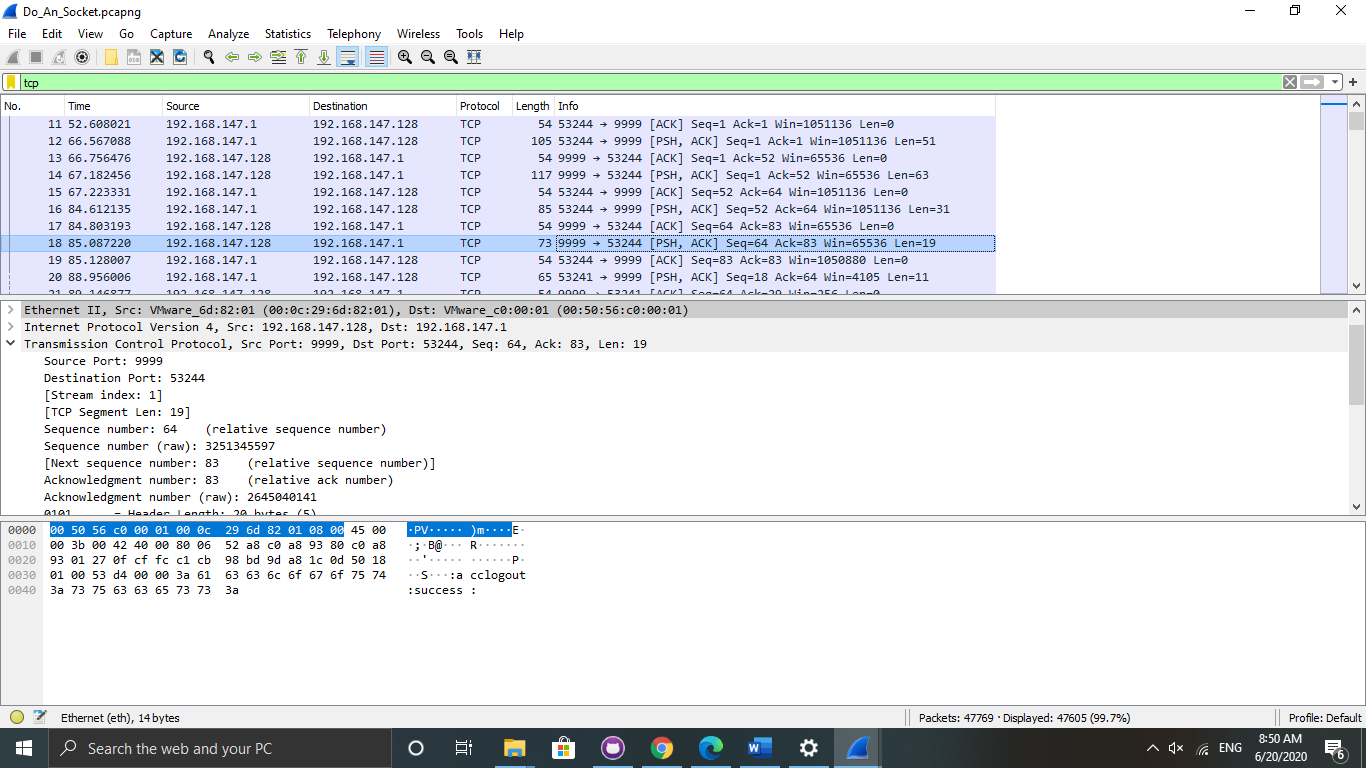
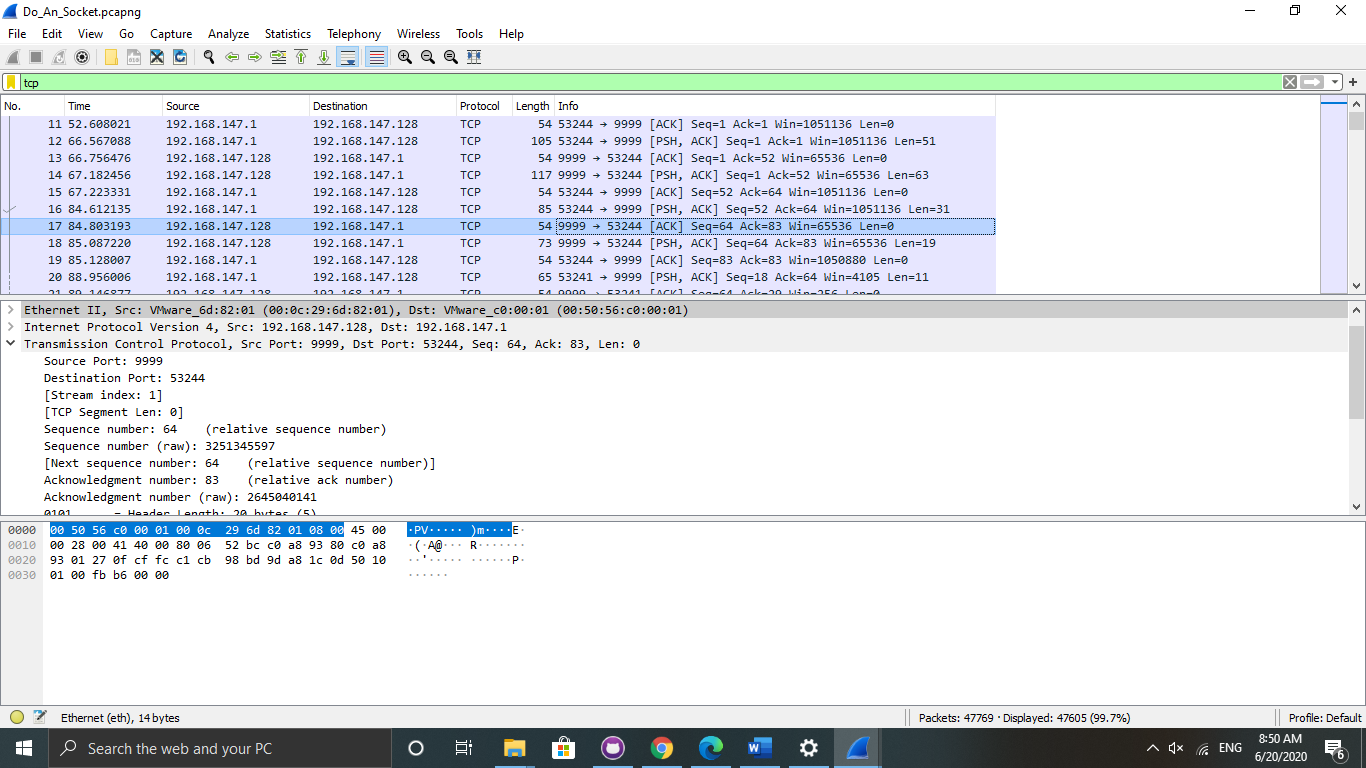
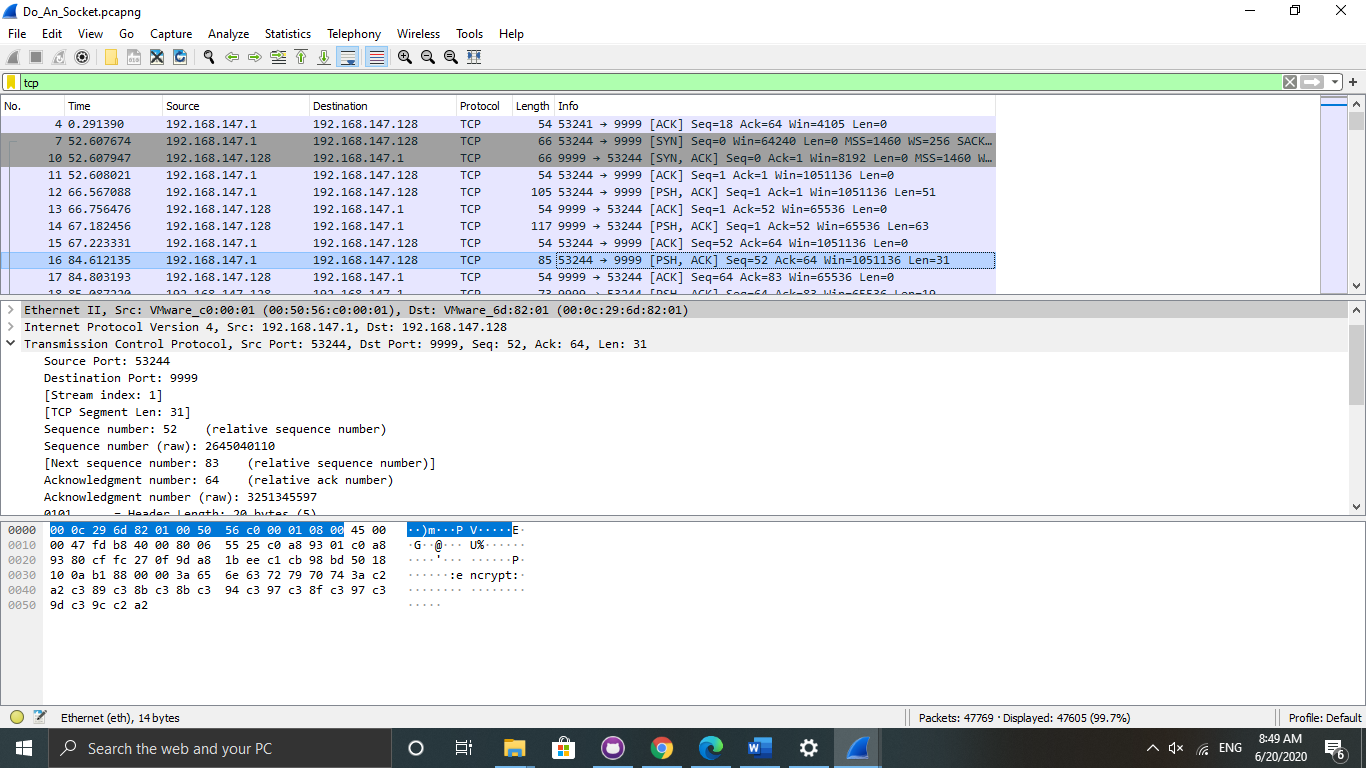
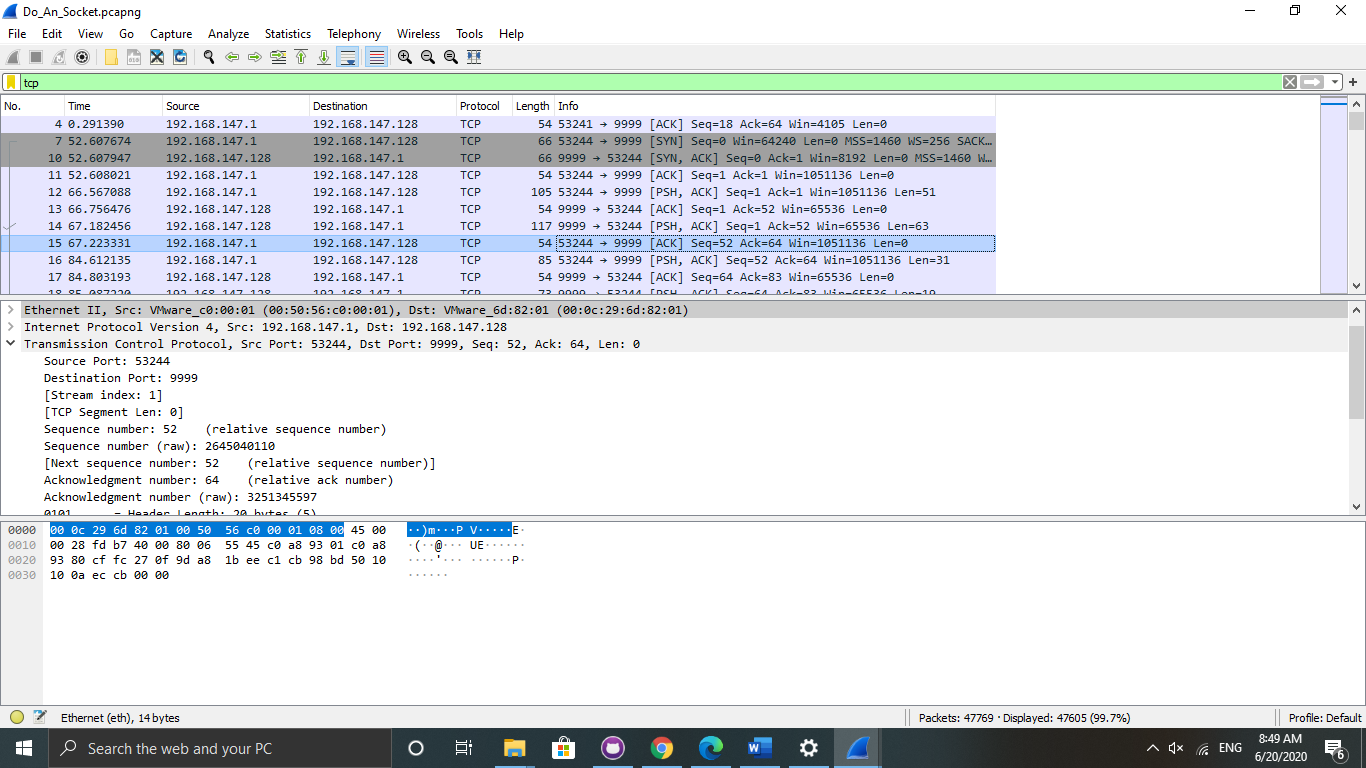
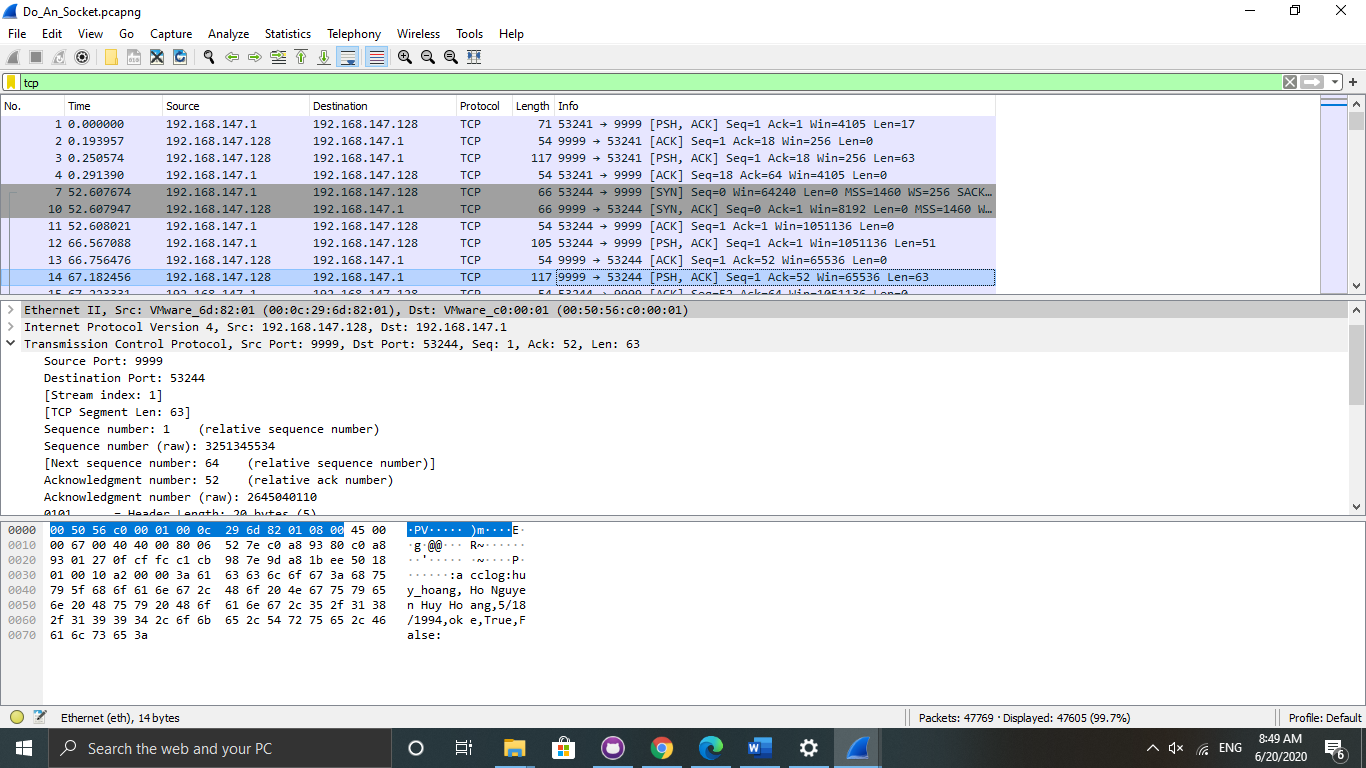
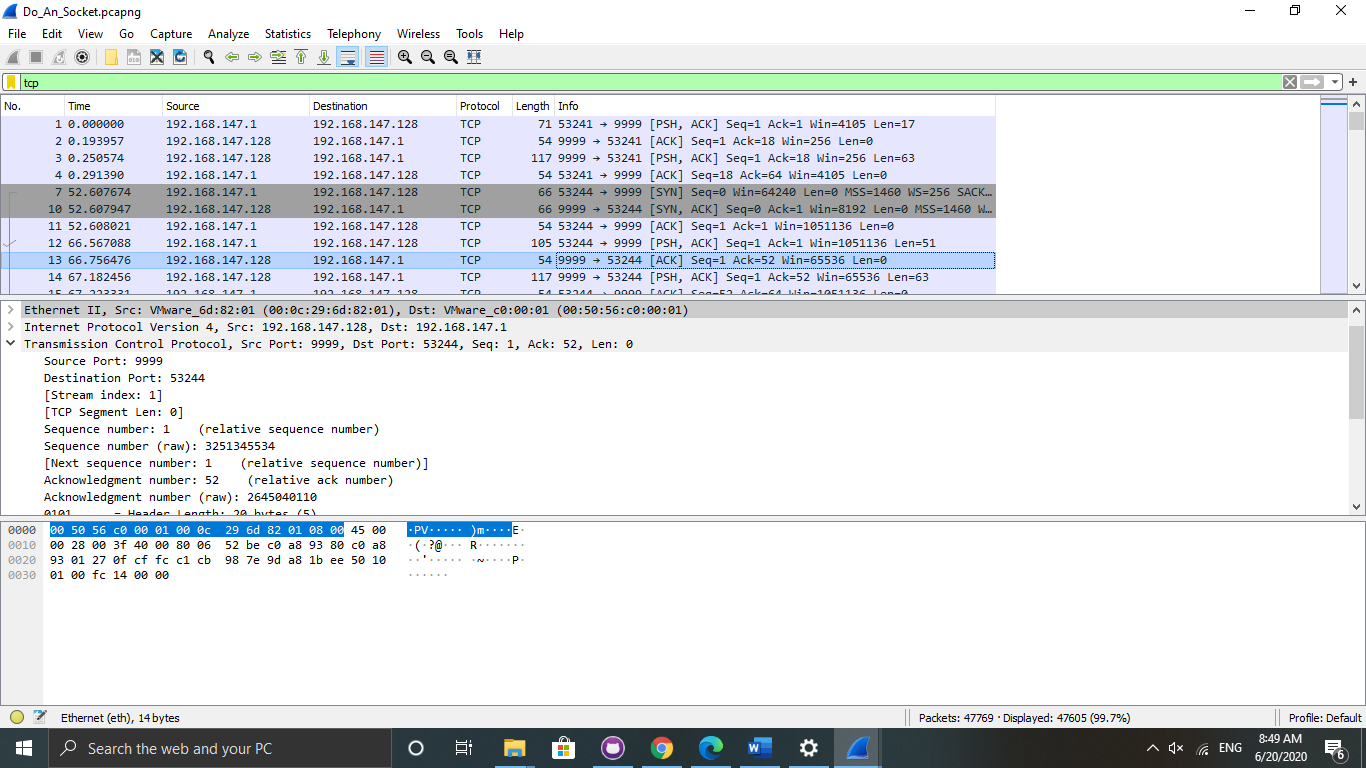
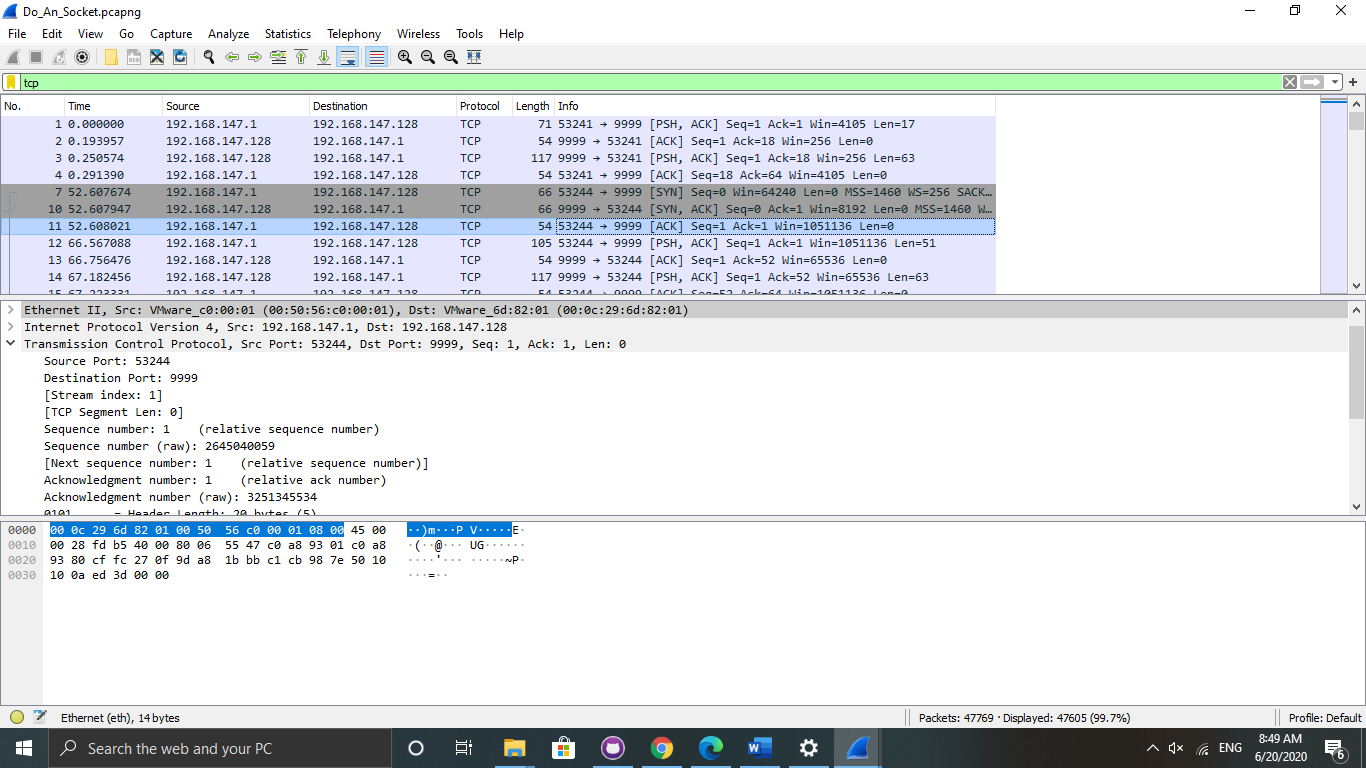
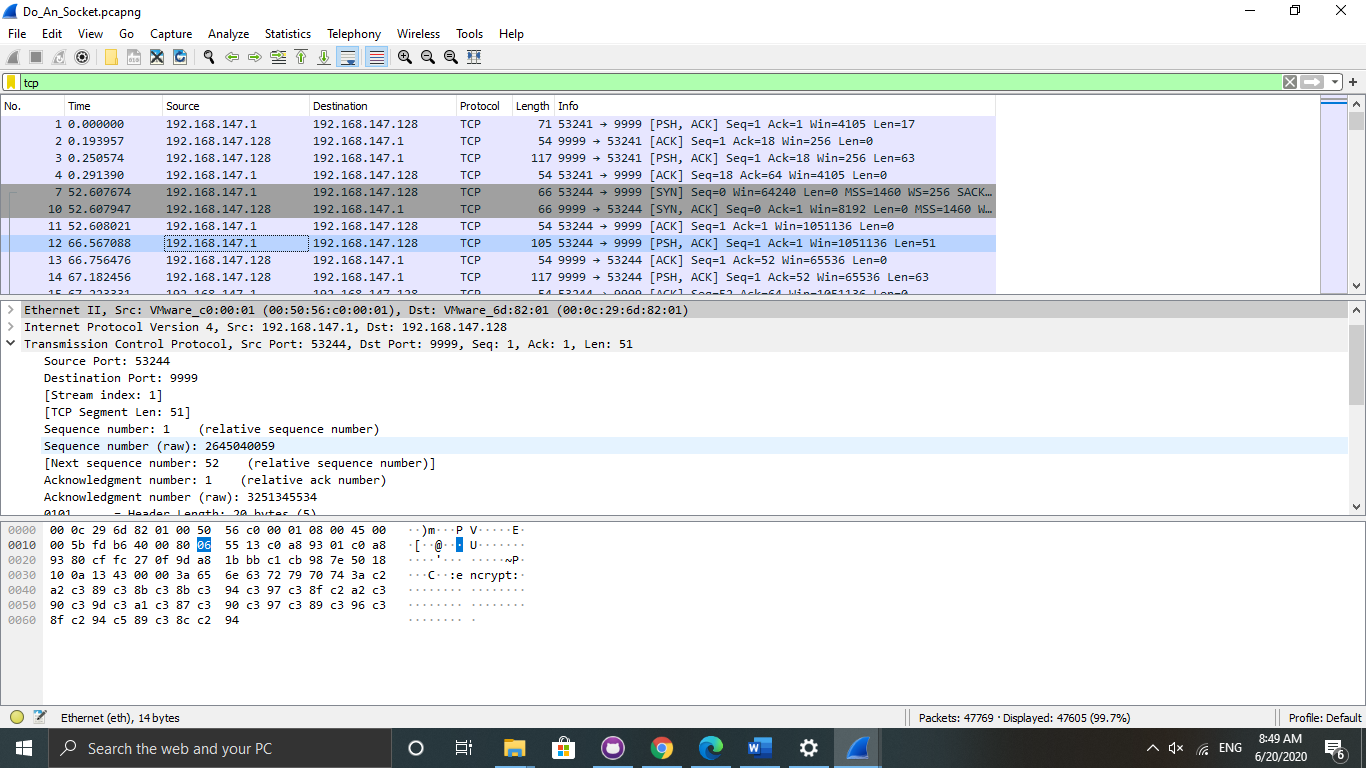
+ public void Add\_Client(ref Client clnt):thêm client

+public void Shutdown\_Connection():đóng kết nối

+ public void Disconnect\_All\_Client():đóng kết nối đến tất cả client

+public void Clear\_Client\_List():xóa danh sách client

***Phần 2:***Bắt và phân tích gói tin



***Phần 3:***Phân công cộng việc

|  |  |
| --- | --- |
| Thành viên | Công việc |
| Hồ Nguyễn Huy Hoàng | +Phân công công việc  +Xây dựng ứng dụng theo mô hình client-server để truyền nhận gói tin, chức năng chat  +Tổng hợp code và chỉnh sửa |
| Nguyễn Trung Kiên | +Xây dựng chức năng đăng nhập và quản lý thông tin người dùng |
| Lê Hồng Quang | +Xây dựng chức năng gửi và nhận gói tin  +Viết báo cáo |