THỰC HÀNH MẠNG MÁY TÍNH NHÓM 2 TỔ 2

HỌ TÊN: ĐẶNG QUỐC CƯỜNG

MÃ SINH VIÊN: B20DCAT020

**PHẦN 1: THỰC HÀNH CÁC TOOL**

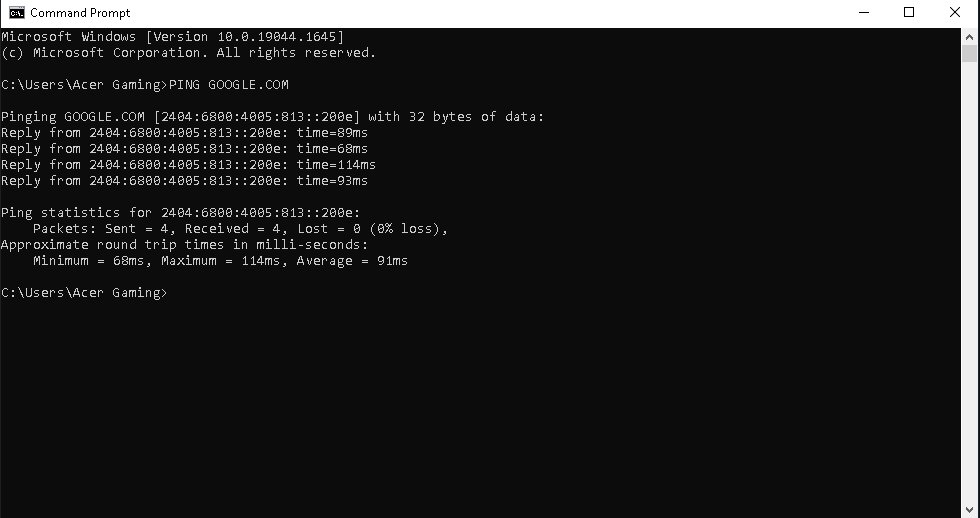
**1)PING**

- Dùng để kiểm tra kết nối tới máy chủ

- Hoạt động: Thiết bị A gửi 1 tín hiệu, 1 gói tin tới một địa chỉ IP của thiết bị B. Thiết bị B nhận được tín hiệu, trả về phản hồi cho A và hiển thị thành kết quả.

- Cách thực hiện: Chạy Command Prompt và nhập lệnh:

**Ping** \*địa chỉ IP/trang web\*-> Ấn enter và quan sát kết quả



- Giải thích tham số :

Reply from 2404 : 6800 : 4005 : 813 : 200e:Phản hồi từ IP địa chỉ của google

Bytes = 32 : gói tin ping đến server có kích thước là 32 byte.

Time = 89ms : thời gian 1 vòng gói tin được gửi đi và server gửi lại.

Sent = 4 : số lượng gói tin gửi đi.

Received = 4 : Số lượng gói tin server nhận được .

Lost = 0% : Phần trăm mất mát của gói tin.

Minimum=68ms: Thời gian gửi đi gói tin ngắn nhất

Maximum=114ms: Thời  gian đi gói tin  lâu nhất

Average =91ms: Thời gian gửi đi gói tin trung bình

**2)IPCONFIG**

- Xem thông tin cấu hình địa chỉ IP, xin cấp lại ip từ DHCP server, xóa cache DNS,…

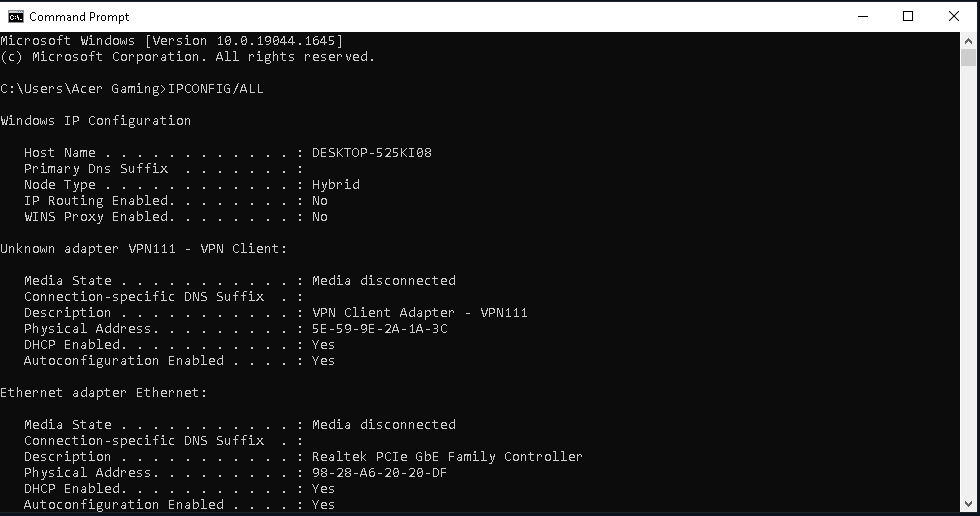
- Hoạt động: Nếu chạy lệnh ipconfig và không kèm theo bất kỳ tùy chọn dòng lệnh nào, nó sẽ liệt kê tất cả các giao diện mạng, bao gồm cả các card mạng ảo (virtual network adapter).

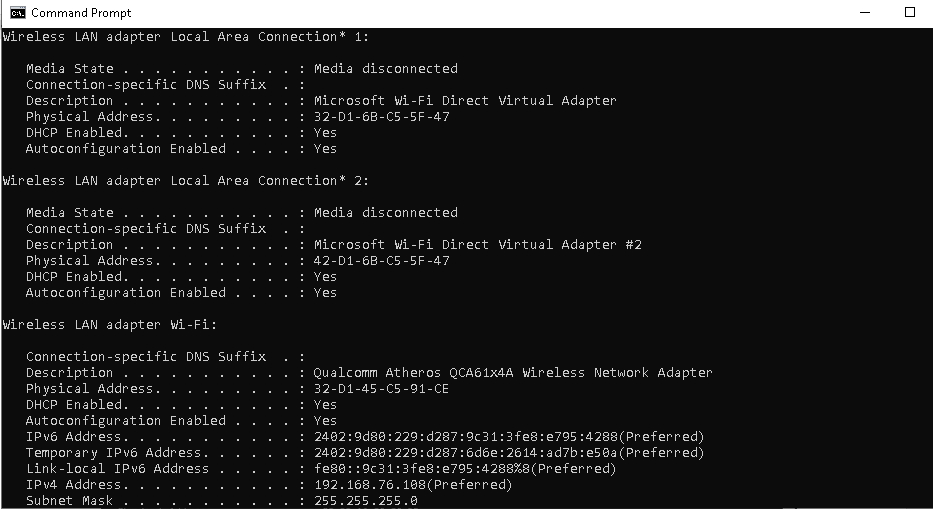
Đối với card mạng LAN và WiFi, ipconfig sẽ cung cấp địa chỉ IP cục bộ. Nếu bạn đang kết nối Internet thông qua WiFi, bạn sẽ thấy giá trị IPv6 và subnet mask (mặt nạ mạng con) bên dưới card mạng này. Nếu một card mạng (LAN hoặc WiFi) không kết nối mạng, bạn sẽ không nhận được bất kỳ thông tin gì khi chạy lệnh ipconfig. Nó đơn giản chỉ cho bạn biết một trong hai card này hiện không được kết nối.

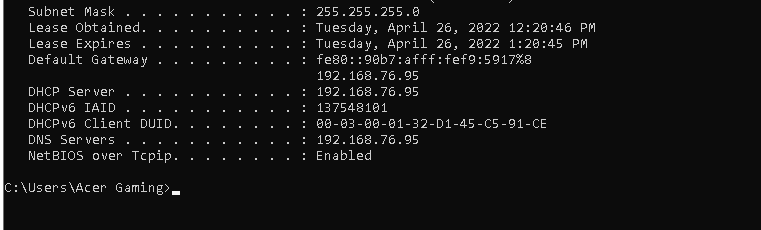
Đối với những card mạng ảo, cho dù có kết nối hay không, bạn đều sẽ thấy cả địa chỉ IPv6 và IPv4 cũng như giá trị subnet mask.

- Cách thực hiện:

Chạy Command Prompt và nhập lệnh: **ipconfig/all**=> ấn enter và quan sát kết quả







- Giải thích tham số

- Hệ thống có 6 giao diện mạng:

- Windows IP Configuration

- Unknown adapter VPN111 - VPN Client:

- Ethernet adapter Ethernet:

- Wireless LAN adapter Local Area Connection\* 1:

- Wireless LAN adapter Local Area Connection\* 2:

- Wireless LAN adapter Wi-Fi:

Default gateway: Cổng mặc định của mạng máy tính (Cho biết giao diện nào đang sử dụng mạng)

Link - local IPv6 Address: địa chỉ IPv6 trên máy của mình.

IPv4 Address: địa chỉ Ipv4 trên máy của mình.

Subnet Mask: Dải mạng hiện tại của lớp địa chỉ trên.

Giao diện có kết nối mạng là Wireless LAN adapter Wi-Fi:

+ Địa chỉ của Default Gateway: fe80::90b7:afff:fef9:5917%8

192.168.76.95

+ Địa chỉ của DNS server trỏ đến: 192.168.76.95

+ IP của giao diện mạng kết nối Internet: 192.168.76.108

+ Có sử dụng DHCP server, địa chỉ của DHCP: 192.168.76.95

**3)NETSTAT**

Netstat (network statistics) là 1 công cụ dòng lệnh được sử dụng để theo dõi các kết nối mạng vào và ra có sẵn trên toàn bộ các hệ điều hành, đặc biệt chúng có sẵn trên Windows.

Cách thức thực hiện : Chạy command prompt => gõ lệnh: “netstat –a –n -o”

Dùng để theo dõi các kết nối mạng ra và trên thiết bị

-a: Hiển thị toàn bộ các kết nối

-n: hiển thị các địa chỉ ip và port dưới dạng số

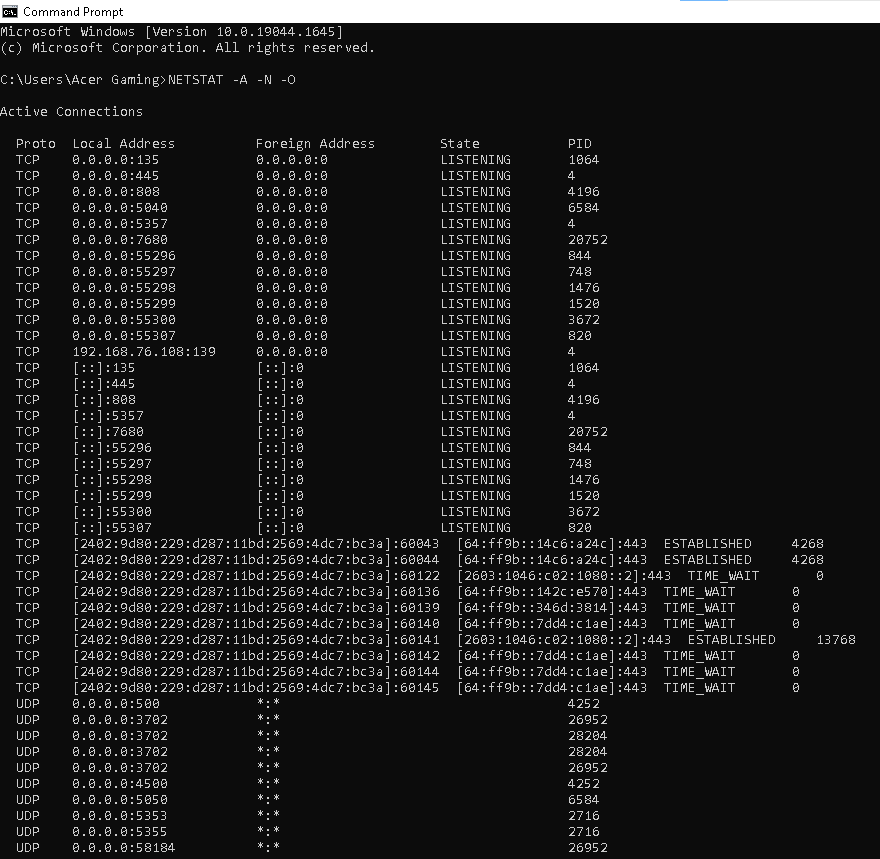
-o: sử dụng PID sau để tìm hiểu Process nào đang sử dụng

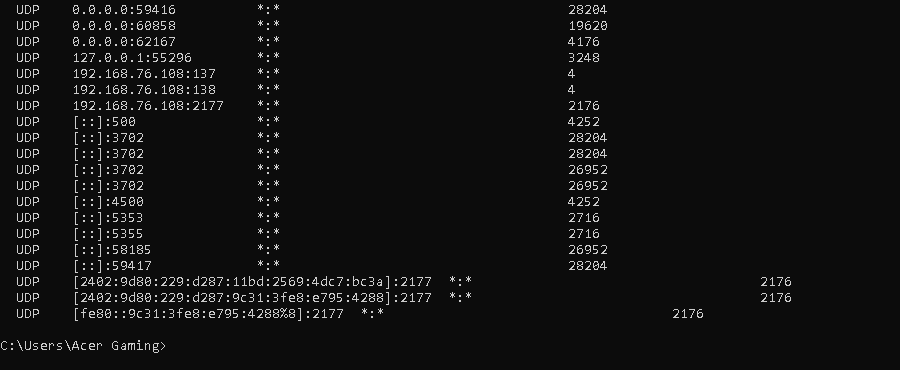
-Có hai loại giao thức trong Proto là TCP và UDP

-Cột Local Address là địa chỉ ip

-Foreign address: Địa chỉ IP và số cổng của máy tính từ xa mà ổ cắm được kết  nối

State: trạng thái kết nối





Các loại giao thức có trong cột Proto: TCP và UDP

**LISTENING:** đợi yêu cầu kết nối từ TCP.

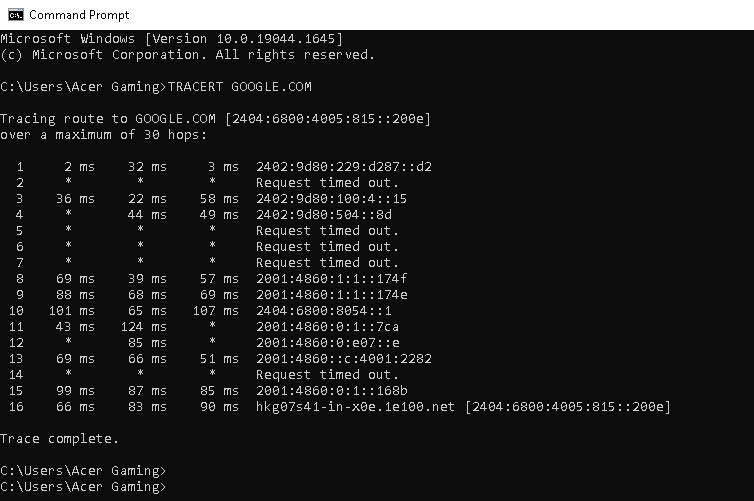
**ESTABLISHED:** cổng sẵn sàng nhận/gửi dữ liệu với TCP ở xa.

**TIME-WAIT:**đang đợi qua đủ thời gian để chắc chắn là TCP ở xa đã nhận được tin báo nhận về yêu cầu kết thúc kết nối của nó

**4)TRACERT:**

Tracert là công cụ để xác định đường đi từ nguồn tới tới đích của một gói giao thức mạng Internet

**Cách thức thực hiện** : Chạy Command Prompt gõ lệnh : tracert <địa chỉ server>=> enter => kiểm tra kết quả .



Đường đi của gói tin qua 16 router

Địa chỉ của server trỏ đến: 2404:6800:4005:813::200e

Biết được tốc độ phản hồi khi gói tin đi qua

Biết được số hop và số thiết bị mà gói tin đã đi qua

Nếu quan sát vào dòng cuối cùng trong đầu ra trên, khi đó sẽ thấy được nó bắt đầu bằng số 16. Con số này thể hiện cho 16 bước nhảy đã được thực hiện để đến được host đích.

Request timed out: kết nối bị vấn đề.

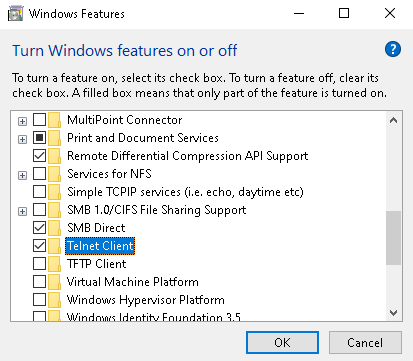
Dấu \* hiển thị timeout do đi quá lâu hoặc không có gói tin phản hồi lại.

**5)TELNET:**

Telnet là một giao thức dòng lệnh được sử dụng để quản lý các thiết bị khác nhau như máy chủ, PC, router, switch, camera, tường lửa từ xa. Telnet là một giao thức cung cấp kết nối từ xa đơn giản. Telnet chịu trách nhiệm gửi các lệnh hoặc dữ liệu đến kết nối mạng từ xa

Trước hết ta phải cài đặt Telnet:

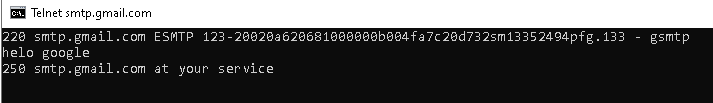
Mở cửa sổ Control Panel => Chọn Program : Trong Programs and Features: Chọn Turn Windows features on or off => Tích ô Telnet client => Nhấn OK => Hoàn thành



**\* Cách thức thực hiện** : Chạy Command Promptgõ telnet smtp.gmail.com 587 để kết nối với gmail



Ví dụ: Chào Server: HELO google



**6)WIRESHARK:**

+ Cách thực hiện: Mở wireshark, chọn mạng đang kết nối với máy tính và mở cmd và ping đến máy chủ bất kì nào đó => quan sát các gói tin trên wireshark

Wireshark sẽ bắt tất cả các gói dữ liệu trên từng địa chỉ IP được kết nối thông qua hệ thống mạng. Và thông tin sẽ liên tục được cập nhật mỗi khi có thiết bị sử dụng mạng.

Wireshark hỗ trợ nhiều tính năng nổi bật cho phép bạn xem chi tiết nội dung từng gói tinbằng việc click 2 lần vào gói tin muốn xem nội dung.

Để dừng tiến trình bạn hãy click vào ô vuông màu đỏ (Stop) trên thanh công cụ.

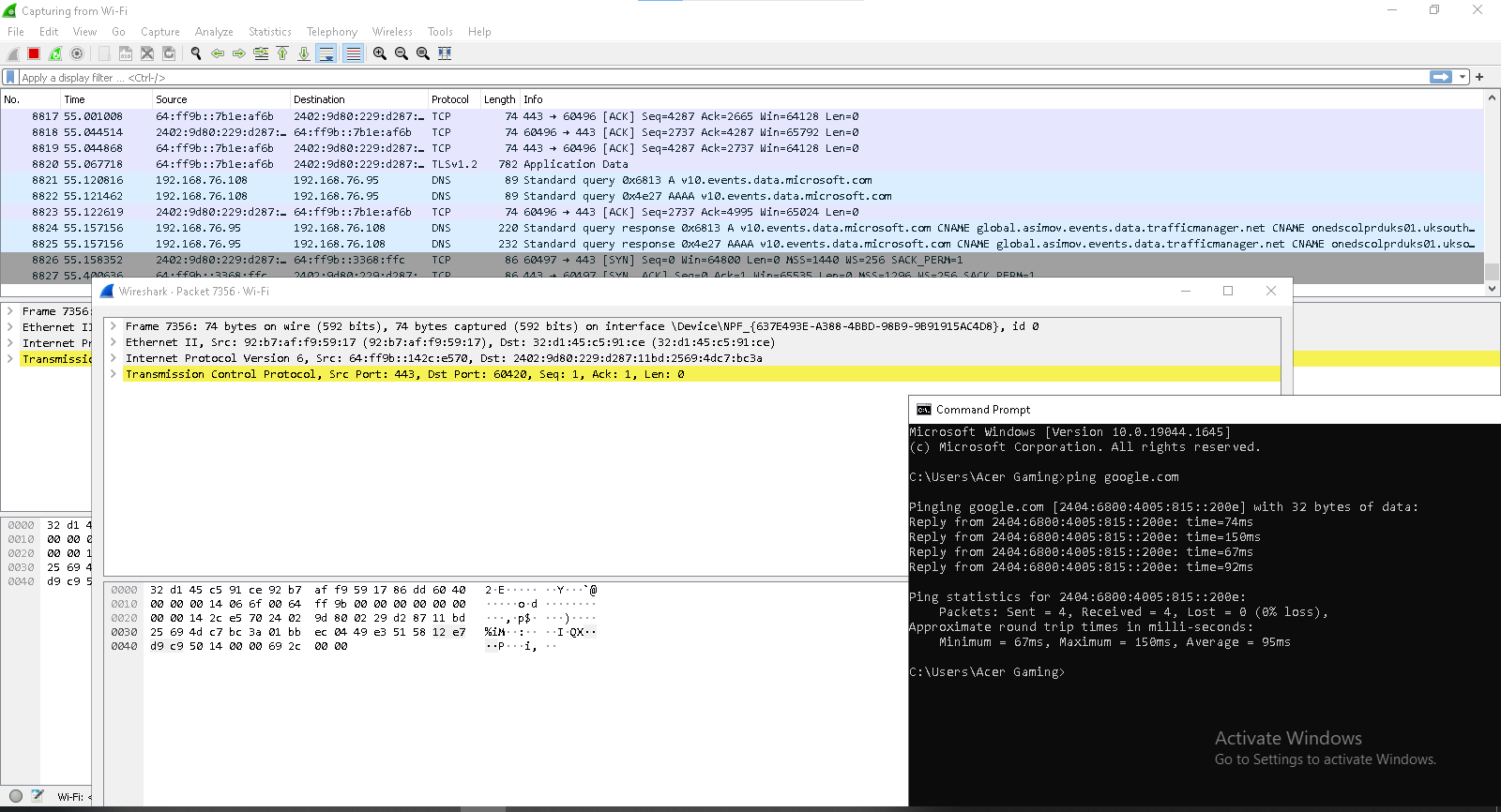
Với mỗi màu sắc được hiển thị khác nhau sẽ tương ứng với mỗi trạng thái khác nhau như:

Màu xanh lá cây là: Traffic TCP

Màu xanh da trời nhạt là: Traffic UDP

Màu hơi ngả tím là: Traffic DNS

Màu đen chữ cam là: Gói TCP gặp vấn đề

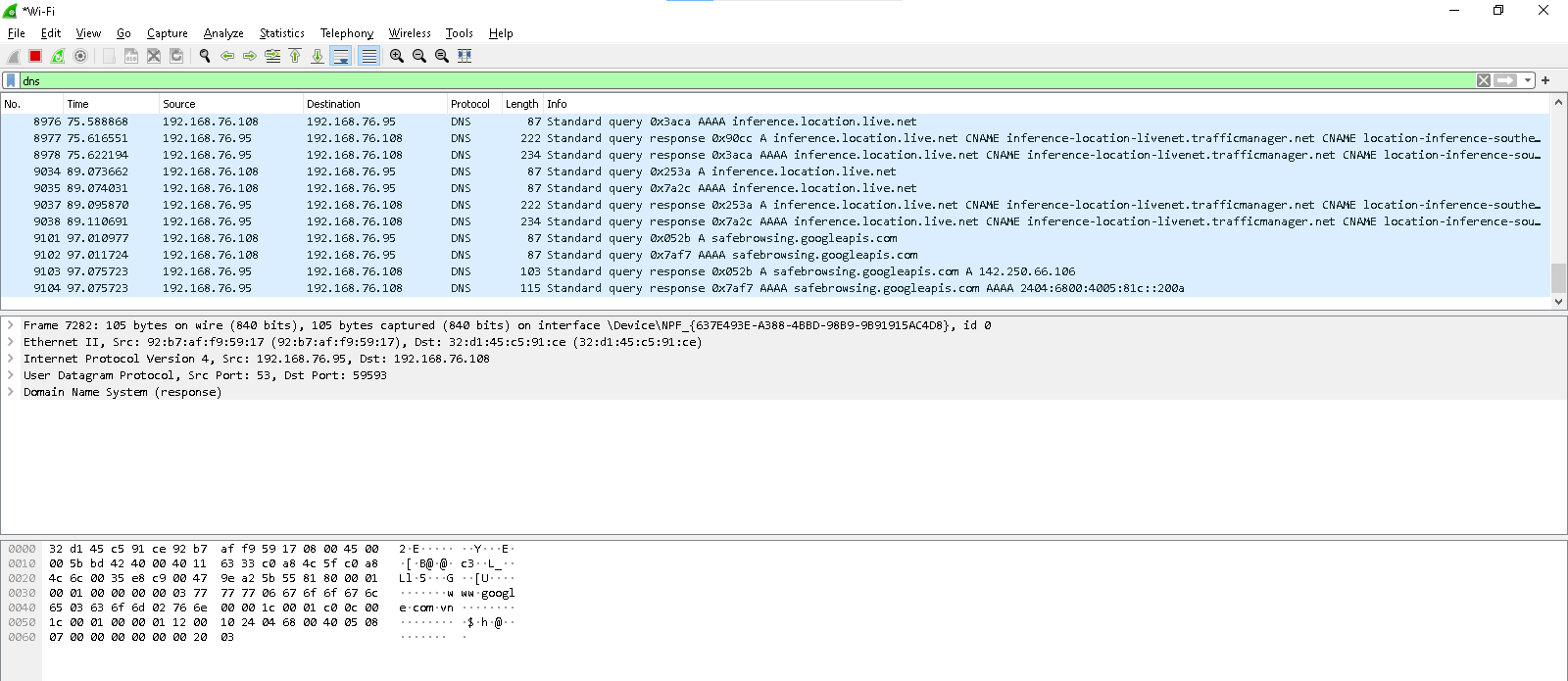


Destination là điểm đến

Source là tên mạng

Protocol là loại giao thức

-Filter thành công các gói tin DNS:



**PHẦN 2: LẬP TRÌNH SOCKET:**

**PYTHON:**

**TCPServer:**

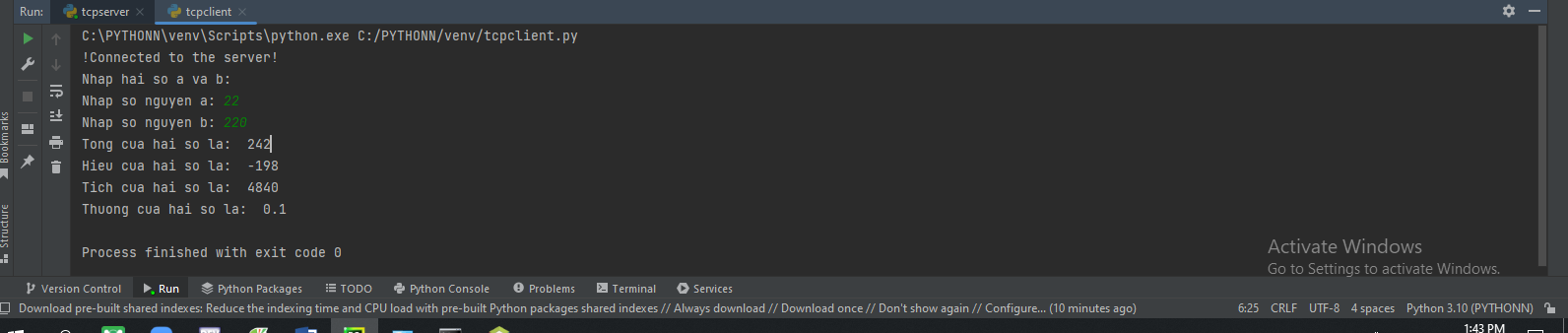
import socket  
import os  
SERVER\_PORT = 57123  
FORMAT = "utf8"  
s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
s.bind(('127.0.0.1', SERVER\_PORT))  
#Vì lập trình kết nối trên cùng 1 máy tính nên địa chỉ IP để là 127.0.0.1 và số hiệu cổng bất kì là 57123  
s.listen(1)  
#cho python biết socket là một server, tham số của phương thức này là số lượng các kết nối có thể có trong hàng đợi  
print("LISTENING....")  
# in 1 câu lệnh để hiển thị là phía server đang đợi câu lệnh từ phía Client  
while 1:# 1 vòng lặp liên tực  
 conn, addr = s.accept()  
 #os.system('cls')  
 #Client gửi yêu cầu, Server chấp nhận và tạo ra một server mới  
 #Client và server đã có thể nhận và truyền dữ liệu  
 print("Connected!")  
 print(conn.getsockname())  
 a = conn.recv(1024).decode(FORMAT)  
 b = conn.recv(1024).decode(FORMAT)  
 print("Da nhap 2 so a va b la :", a, " va ", b)  
 tong = int(a) + int(b)  
 hieu = int(a) - int(b)  
 tich = int(a) \* int(b)  
 thuong = int(a) / int(b)  
 conn.sendall(str(tong).encode(FORMAT))#gửi kết quả đã encode  
 conn.recv(1024)#nhập 1024 bit dữ liệu  
 conn.sendall(str(hieu).encode(FORMAT))#gửi kết quả đã encode  
 conn.recv(1024)#nhập 1024 bit dữ liệu  
 conn.sendall(str(tich).encode(FORMAT))#gửi kết quả đã encode  
 conn.recv(1024)#nhập 1024 bit dữ liệu  
 conn.sendall(str(thuong).encode(FORMAT))#gửi kết quả đã encode  
 conn.recv(1024)#nhập 1024 bit dữ liệu  
 print("Build Successfull!!!!")#in ra câu lệnh để hiển thị rằng kết nối giữa Client và Server đã thành công  
 conn.close()# đóng kết nối

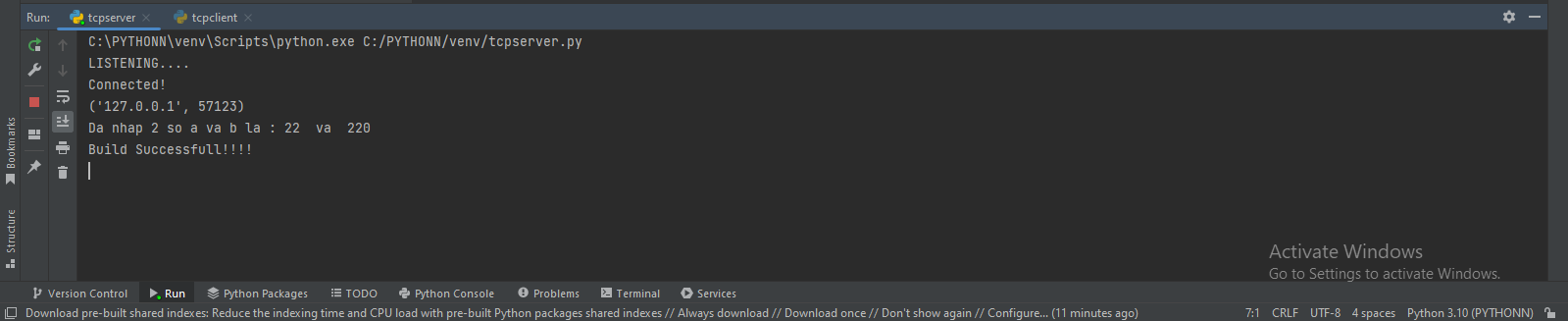
**TCPClient:**

import socket  
import os  
IP = "127.0.0.1"#Kết nối trên cùng 1 máy nên địa chỉ có thể để là 127.0.0.1  
SERVER\_PORT = 57123#số hiệu cổng  
FORMAT = "utf8"  
client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)#khởi tạo socket TCP  
client.connect((IP, SERVER\_PORT))#kết nối  
print("!Connected to the server!")#in ra câu lệnh để thông báo kết nối thành công tới server  
print("Nhap hai so a va b: ")  
a = int(input("Nhap so nguyen a: "))#nhập thông điệp  
client.sendall(str(a).encode(FORMAT))# gửi thông điệp  
b = int(input("Nhap so nguyen b: "))#nhập thông điệp  
client.sendall(str(b).encode(FORMAT))# gửi thông điệp  
tong = client.recv(1024).decode(FORMAT)#mỗi lần nhập 1024 bit dữ liệu  
print("Tong cua hai so la: ", tong)#in ra kết quả đã encode  
client.sendall(tong.encode(FORMAT))# gửi thông điệp  
hieu = client.recv(1024).decode(FORMAT)#mỗi lần nhập 1024 bit dữ liệu  
print("Hieu cua hai so la: ", hieu)#in ra kết quả đã encode  
client.sendall(hieu.encode(FORMAT))# gửi thông điệp  
tich = client.recv(1024).decode(FORMAT)#mỗi lần nhập 1024 bit dữ liệu  
print("Tich cua hai so la: ", tich)#in ra kết quả đã encode  
client.sendall(tich.encode(FORMAT))# gửi thông điệp  
thuong = client.recv(1024).decode(FORMAT)#mỗi lần nhập 1024 bit dữ liệu  
print("Thuong cua hai so la: ", thuong)#in ra kết quả đã encode  
client.sendall(thuong.encode(FORMAT))# gửi thông điệp  
client.close()#đóng kết nối

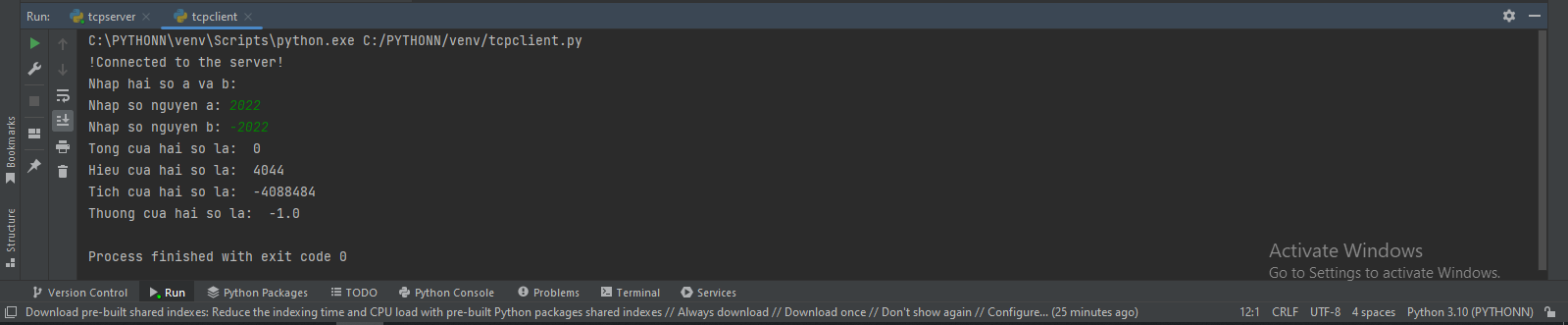
**Kết quả:**

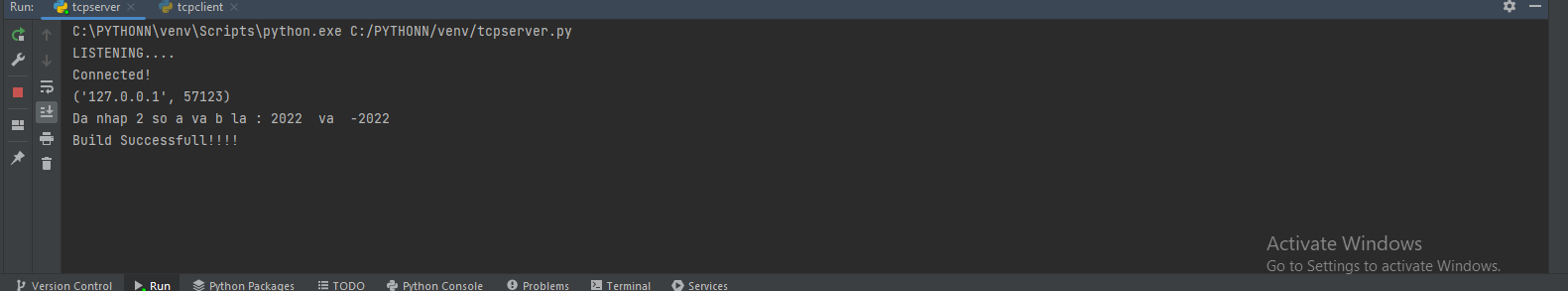
**- Test 1:**

****

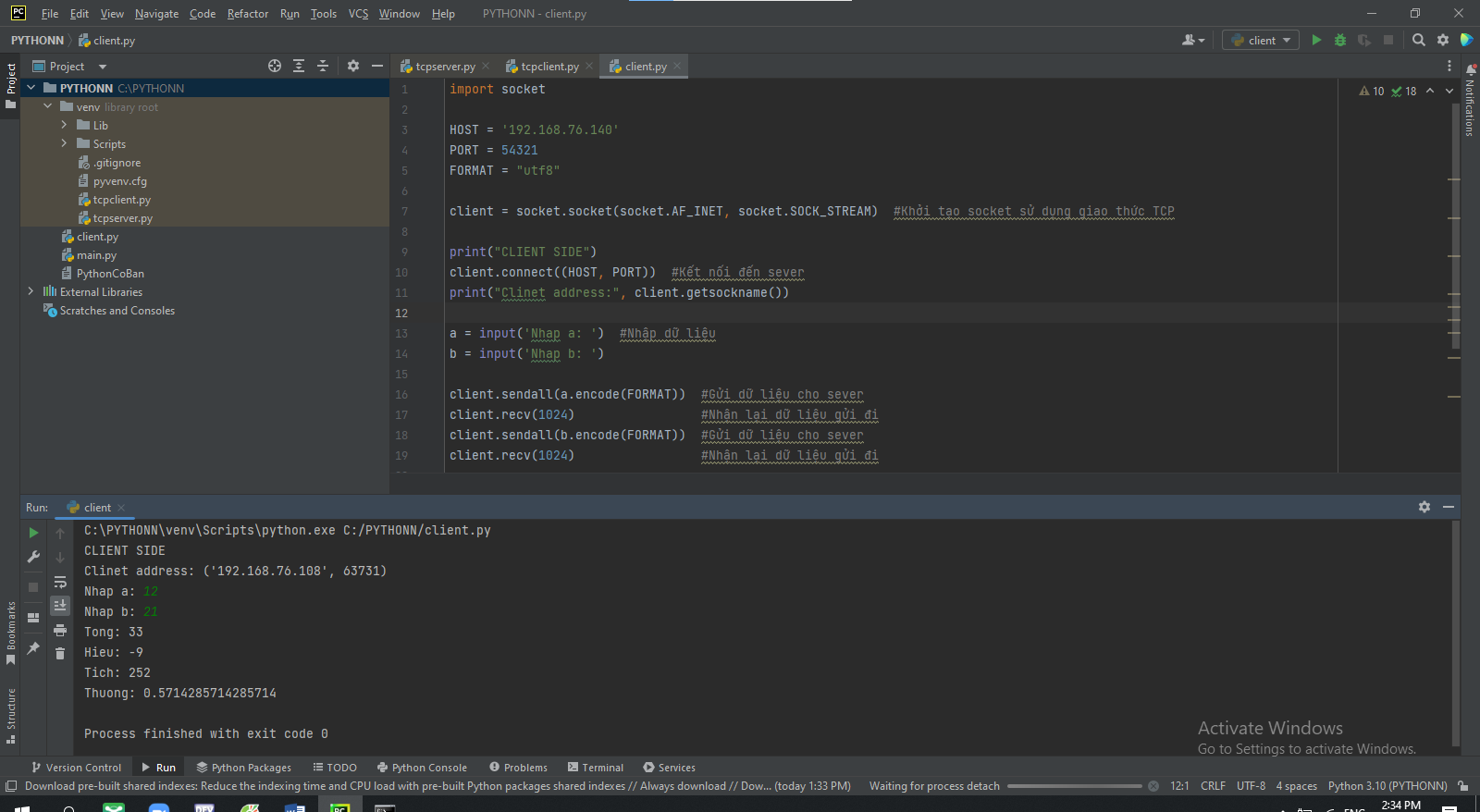
****

**- Test 2:**

****

****

**Test kết nối giữa 2 máy tính:**

****