**ĐỒ ÁN 1**

**Tên đề tài: Chương trình DNS Client**

Sinh viên: Đào Duy Thành

Giáo viên hướng dẫn: TS. Thân Quang Khát

# Mục lục

[Mục lục 2](#_Toc515797806)

[Phần mở đầu 3](#_Toc515797807)

[Phần 1: ĐẶC ĐIỂM CỦA DNS 4](#_Toc515797808)

[I. Tổng quan về DNS 4](#_Toc515797809)

[1. DNS là gì? 4](#_Toc515797810)

[2. Cấu trúc của hệ thống tên miền 4](#_Toc515797811)

[II. DNS SERVER 4](#_Toc515797812)

[1. DNS server là gì? 4](#_Toc515797813)

[2. Hệ thống máy chủ DNS. 4](#_Toc515797814)

[III. ZONE 6](#_Toc515797815)

[1. Zone là gì? 6](#_Toc515797816)

[2. Phân loại zone. 6](#_Toc515797817)

[IV. PHÂN GIẢI TÊN MIỀN 7](#_Toc515797818)

[PHẦN 2: KHUÔN DẠNG THÔNG ĐIỆP 8](#_Toc515797819)

[Phần 3: Chương trình DNS Client 13](#_Toc515797820)

[I. Tổng quan 13](#_Toc515797821)

[II. File query.py 13](#_Toc515797822)

[1. Gói tin truy vấn 14](#_Toc515797823)

[2. Gói tin trả lời 17](#_Toc515797824)

[III. File Client.py 20](#_Toc515797825)

[IV. Demo chương trình 21](#_Toc515797826)

[Tài liệu tham khảo 23](#_Toc515797827)

# Phần mở đầu

Ngày nay, mạng Internet và các trang web đã quá quen thuộc với chúng ta. Việc định địa chỉ trên Internet dựa vào địa chỉ IP - một chuỗi số rất khó nhớ và khô khan. Nhưng khi truy cập trang web, nếu phải nhớ chính xác địa chỉ IP của trang web sẽ rất khó khan. Thay vào đó, chúng ta chỉ cần biết tên trang web – một chuỗi kí tự dễ nhớ và có thể mang ý nghĩa riêng. Và dịch vụ DNS ra đời để giải quyết vấn đề đó: phân giải giữa tên miền giữa địa chỉ IP

Học kì này, tôi học môn mạng máy tính, học về các thức đóng gói các gói tin, truyền và nhận gói tin qua mạng Internet. Vì vậy, tôi chọn đề tài “Viết chương trình DNS Client”, để vận dụng những kiến thức được học về mạng máy tính vào việc lập trình.

# Phần 1: ĐẶC ĐIỂM CỦA DNS

## I. Tổng quan về DNS

### DNS là gì?

Hiện nay, trên thế giới có vô số các máy tính cần kết nối, liên lạc với nhau để trao đổi thông tin, dữ liệu. Để làm được việc này các máy tính phải biết rõ địa chỉ IP của nhau. Nhưng địa chỉ IP là một dãy số dài khó nhớ vì vậy để đơn giản hoá việc nhớ đoạn dãy số dài này người ta đã tạo ra một hệ thống có khả năng ánh xạ địa chỉ IP thành một chuỗi kí tự dễ nhớ (Domain name ): <http://www.hust.edu.vn>, <https://www.facebook.com>,...

Hệ thống đó là hệ thống phân giải tên miền DNS( Domain Name System). Hệ thống tên miền- Domain name sử dụng một file host.txt đặt trên máy làm máy chủ tên miền. File này lưu thông tin của tất cả các máy trong mạng. Khi đó các client trong mạng muốn truy xuất đến các client khác chỉ cần hỏi DNS.

Vậy mục đích của DNS là phân gải tên miền thành địa chỉ IP và ngược lại

Hệ thống tên miền- DNS hoạt động theo mô hình client-server.

### Cấu trúc của hệ thống tên miền

Hệ thống tên trong DNS được sắp xếp theo cấu trúc phân cấp vào kiến trúc hình cây logic được gọi là DNS namespace

* Gốc( Domain root): là đỉnh của nhánh cây tên miền
* Tên miền cấp một( top-level domain): gồm vài kí tự xá định một nước, khu vực hoặc tổ chức.
* Tên miền cấp hai( second-level domain): có thể là tên một công ty, tổ chức hay một cá nhân.
* Tên miền cấp nhỏ hơn(subdomain): có thể biểu diễn cho chi nhánh, phòng ban của một cơ quan hay chủ thể nào đó.

## II. DNS SERVER

### DNS server là gì?

DNS server là một cơ sở dữ liệu chứa các thông tin về vị trí của các DNS domain và phân giải các truy vấn xuất phát từ các Client.

DNS server có thể cung cấp các thông tin do Client yêu cầu và chuyển đến một DNS server khác để nhờ phân giải hộ trong trường hợp nó không thể trả lời các truy vấn về những tên miền không thuộc quyền quản lý và cũng luôn sẵn sàng trả lời các máy chủ khác về tên miền nó quản lý.

### Hệ thống máy chủ DNS.

* Có một máy chủ tên miền gốc( root server): Tên mạng internet có 13 hệ thống máy chủ gốc. Root server là server quản lý toàn bộ cấu trúc của hệ thống tên miền, nó không chứa thông tin về cấu trúc hệ thống DNS mà nó chỉ chuyển quyền quản lý xuống cho các server cấp thấp hơn. Do đó nó có khả năng định đường đến của một domain tại bất kì đâu trên mạng.
* Máy chủ tên miền cấp một( top-level domain): Quản lý các tên miền cấp một
* Máy chủ được uỷ quyền( Authoritative DNS servers): Quản lý các tên miền cấp dưới.
* Máy chủ của các tổ chức: của ISP

Không nằm trong phân cấp của DNS

* Máy chủ cục bộ: dành cho mạng nội bộ của cơ quan, tổ chức

Không nằm trong phân cấp của DNS.

#### Phân loại DNS server.

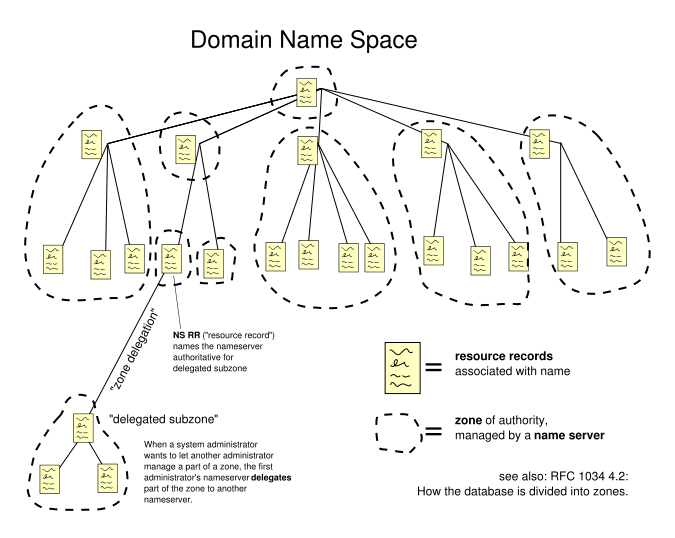
DNS server gồm có 2 loại:

* Primary server: được tạo ra khi ta add một primary zone mới thông qua New Zone Wizard. Là DNS server chính, cho phép thêm, sửa, xoá cơ sở dữ liệu của DNS. Mỗi Domain phải có một primary server.
* Secondary server: là DNS server phụ, backup lại CSDL của primary. Không được thay đổi CSDL của DNS. Trong trường hợp primary server bị lỗi, secondary server sẽ được sử dụng để phân giải tên. Sau 24h nếu secondary server không được chuyển lên primary server thì CSDL DNS của nó sẽ bị hết hạn sử dụng và lúc đó nó sẽ không phân giải tên được nữa.

Zone transfer: khi secondary server được khởi động nó sẽ tìm primary server nào mà nó được phép lấy dữ liệu về máy. Nó sẽ copy lại toàn bộ CSDL DNS của primary server mà nó được phép lấy dữ liệu đó, quá trình này đuộc gọi là quá trình zone transfer

## III. ZONE

### Zone là gì?



Trong hệ thống DNS, người ta chia nhỏ thành những phần để gán những quản lý riêng vì vậy vần phải chia ra các zone để đảm bảo việc quản lý DNS.

Một zone là tập hợp của những domain hoặc subdomain do DNS quản lý.

Dữ liệu DNS được chứa trên các máy chủ zone, dữ liệu của DNS là dữ liệu của các zone.

### Phân loại zone.

Zone có 3 loại:

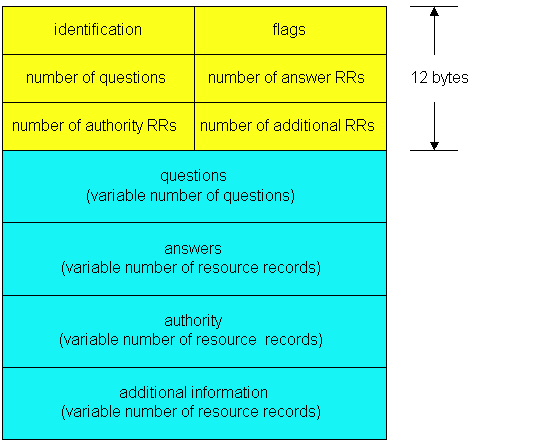
* Primary zone: Máy chủ chứa dữ liệu primary zone là máy chủ có thể toàn quyền trong việc update dữ liệu zone
* Secondary zone: Là một bản copy của primary zone. Do nó chứa dữ liệu zone nên cung cấp khả năng phân giải cho các máy có yêu cầu. Muốn cập nhật zone phải đồng bộ với máy chủ primary
* Stub zone: Dữ liệu của stub zone chỉ bao gồm dữ liệu NS Record trên máy chủ primary zone. Nó có vai trò chuyển các yêu cầu dữ liệu của một zone nào đó đến trực tếp máy chủ có thẩm quyền của zone đó

## IV. PHÂN GIẢI TÊN MIỀN

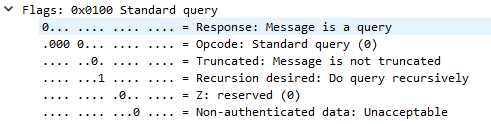
Có 2 cơ chế :

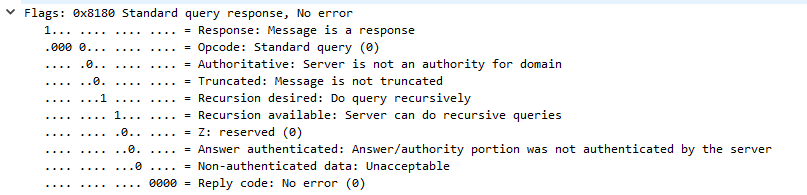
* Tự phân giải :
* File HOST
* Cache
* Dịch vụ phân giải tên miền DNS : Client/ server
* UDP, port 53
* Phân giải đệ quy( recursive query)
* Phân giải tương tác( interactive query)

# PHẦN 2: KHUÔN DẠNG THÔNG ĐIỆP



* 6 trường đầu tiên dài 16 bit, các trường khác có độ dài thay đổi.
* Identification: trường nhận dạng. dung để nhận biết gói tin có bị phân mảnh hay không.





* Trường cờ:

|  |  |
| --- | --- |
| Số thứ tự bit | Giá trị- đặc điểm |
| 1 | 0: gói tin truy vấn  1: gói tin trả lời |
| 2-5 | 0000:truy vấn đệ quy  0100: truy vấn nghịch  0010, 0001 không sử dụng |
| 6 | Câu trả lời có thẩn quyền, chỉ có hiệu lực trong phản hồi  0: câu trả lời không có thẩm quyền  1:câu trả lời có thẩm quyền |
| 7 | Kiểm tra dữ liệu gói tin có bị quá tải hay không. Nếu có sẽ bị cắt ngắn  0: dữ liệu gói tin không quá tải  1: dữ liệu gói tin bị quá tải |
| 8 | Đệ quy mong muốn- bit này có thể được đặt trong một truy vấn và được sao chép vào phản hồi nếu đệ quy được hỗ trợ bởi máy chủ định danh này. Hỗ trợ truy vấn đệ uy là tuỳ chọn  1: mong muốn đệ quy  0: không mong muốn đệ quy |
| 9 | Đệ quy có sẵn- bit này có trong gói tin trả lời biểu thị hỗ trợ truy vấn đệ quy có sẵn hay không.  1:có sẵn  0: không có sẵn |
| 10 | Dự trữ |
| 11 | Trả lời về thẩm quyền  0: phần được uỷ quyền không được máy chủ xác thực  1: phần được uỷ quyền được máy chủ xác thực |
| 12 | Dự trữ |
| 13-16 | Xác định loại phản hồi cho truy vấn  0000: không có lỗi  0100:lỗi định dạng- máy chủ định danh không thể diễn giải truy vấn  0010: lỗi máy chủ- máy chủ định danh không thể xủ lí truy vấn do sự cố  0001: lỗi tên- chỉ có ý nghĩa với các máy chủ có thẩm quyền. biểu thị tên được tham chiếu trong truy vấn không tồn tại |

* Number of question: số câu hỏi trong phần question
* Number of answer RRs: số câu trả lời trong phần answer
* Number of authority RRs: số lượng mục trong phần Authority
* Number of additionalRRs: số lượng mục trong phần additional information.
* Phần QUESTION

Phần question có cấu trúc:

|  |  |
| --- | --- |
| Tên trường | Ý nghĩa/ sử dụng |
| QNAME | Tên miền được truy vấn, nội dung của nó sẽ phụ thuộc vào QTYPE. |
| QTYPE | Các hồ sơ tài nguyên được yêu cầu |
| QCLASS | Lớp được yêu cầu |

* Phần ANSWER

Phần answer có cấu trúc :

|  |  |
| --- | --- |
| Tên trường | Ý nghĩa/ sử dụng |
| Name | Tên miền được truy vấn, có cùng định dạng với QNAME ở câu hỏi  16 bit |
| Type | Quy định liểu của dữ liệu trong bản ghi  16 bit |
| Class | Quy định lớp của bản ghi  16 bit |
| TTL | Số giây mà bản ghi có thể di chuyển  32 bit |
| Record Data Length | Dộ dài dữ liệu của bản ghi |
| Record Data | Dữ liệu của bản ghi,dữ liệu này được quyết định bởi trường Type. |

* Phần Authoritative: phần thẩm quyền

Chỉ có trong phần phản hồi. có cấu trúc tương tự phần ANSWER.

* Phần ADDITIONAL INFORMATION: phần bổ sung

Có cấu trúc tương tự phần answer, chỉ có trong phần phản hồi.

CÁC BẢN GHI DNS

Bản ghi DNS( resource record) là mẫu thông tin dùng để mô tả các thông tin về cơ sở dữ liệu DNS.

|  |
| --- |
| Khuôn dạng RR: ( name, value, type, TTL) |

TTL( time to live): thời gian sống của bản ghi. Giá trị này có thể được thay đổi bởi người quản trị trong việc khai báo TTL cho dữ liệu đó.

|  |  |  |
| --- | --- | --- |
| type | name | Value |
| A | Tên trạm | Địa chỉ IP của máy chủ hoặc tên miền |
| CNAME | Tên bí danh | Tên miền chính thống và bí danh. Cho phép 1 host có thể có nhiều tên |
| NS | Tên miền | Tên trạm của DNS server thẩm quyền của tên miền này |
| MX | Mail exchange | Tên mailserver gắn với tên |

# Phần 3: Chương trình DNS Client

## Tổng quan

Chương trình viết bằng ngôn ngữ python 3, và lập trình hướng đối tượng

Chương trình sử dụng module socket để gửi và nhận gói tin giữa Client và server (server ở đây mặc định là “8.8.8.8”)

Chương trình thiết lập các thông tin cho phần truy vấn DNS của gói tin được gửi đi và nhận về gói tin trả lời, sau đó giải mã

Chương trình chỉ phân giải tên miền sang địa chỉ IPv4, và phân giải ngược từ địa chỉ IPv4 sang tên miền

Chương trình gồm 2 file :

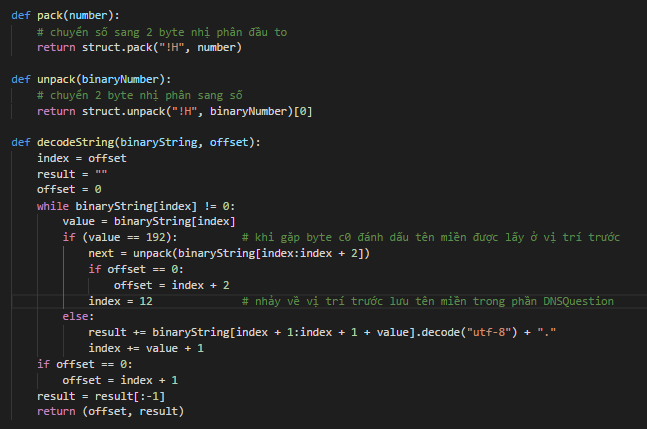
* File Message.py chứa các lớp về các gói tin truy vấn và gói tin trả lời, các lớp về các bản ghi A, CNAME, PTR
* File DNSClient.py chứa chương trình chính, thiết lập kết nối tới socket, chạy và đóng kết nối

## File Message.py

File này chứa các lớp về các gói tin truy vấn và gói tin trả lời, các lớp về bản ghi A, CNAME, PTR

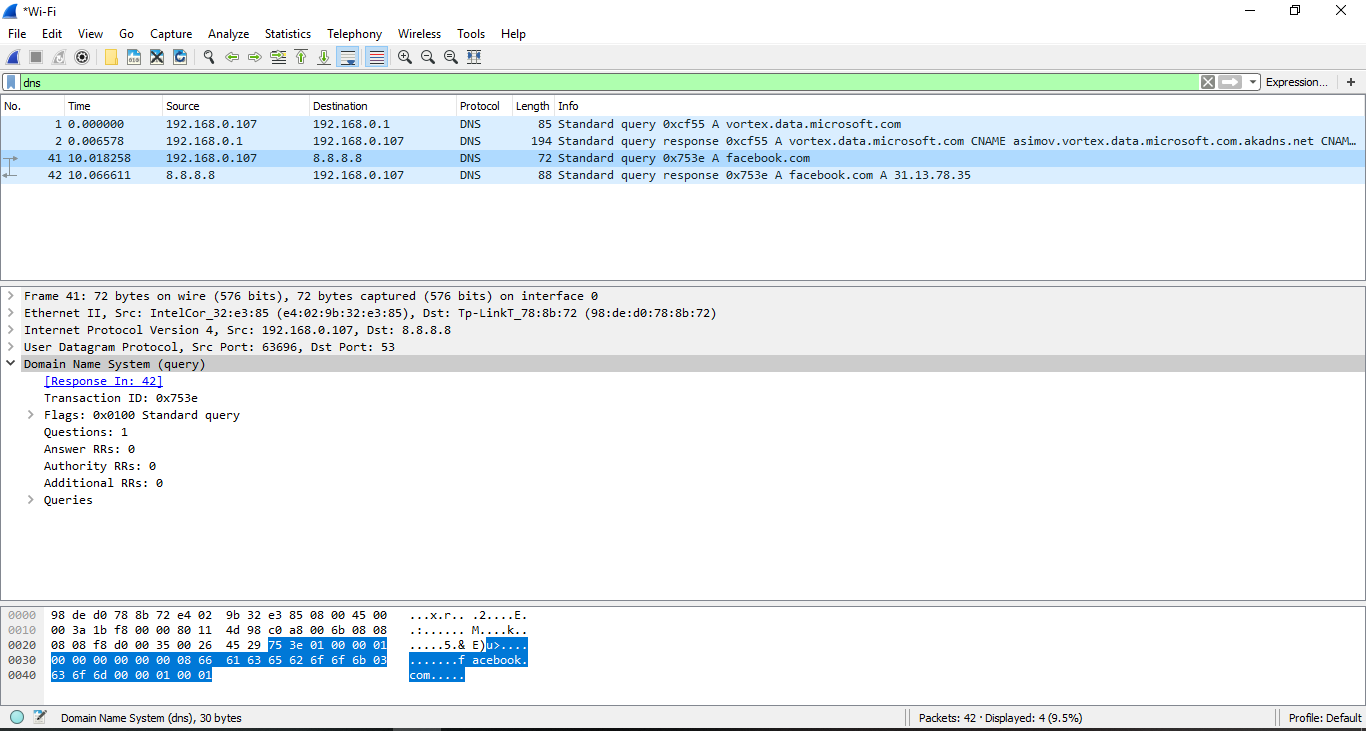
Các hàm bổ sung:

* Hàm pack(number) mã hóa số giá trị value bằng 2 byte nhị phân kiểu đầu to
* Hàm unpack(binaryNumber) giải mã chuỗi 2 byte nhị phân kiểu đầu to thành số nguyên giá trị tương ứng
* Hàm decodeString(binaryString, offset) giải mã các byte từ vị trí offset đến khi gặp byte 00 sang dạng chuỗi



### Gói tin truy vấn

Sử dụng phần mềm wireshark để bắt gói tin truy vấn DNS, ta được như sau:



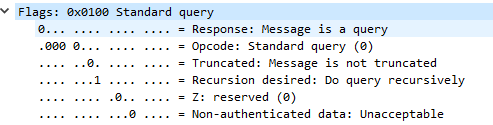
Các thông tin ở dòng thứ 2 (Ethernet), thứ 3 (Internet Protocol Version 4), thứ 4 (User Datagram Protocol) sẽ do module socket thiết lập. Chúng ta sẽ xây dựng các thông tin ở dòng Domain Name System (query)

Các phương thức encode() trong các lớp sẽ thiết lập thông tin cho gói tin dưới dạng các bit nhị phân

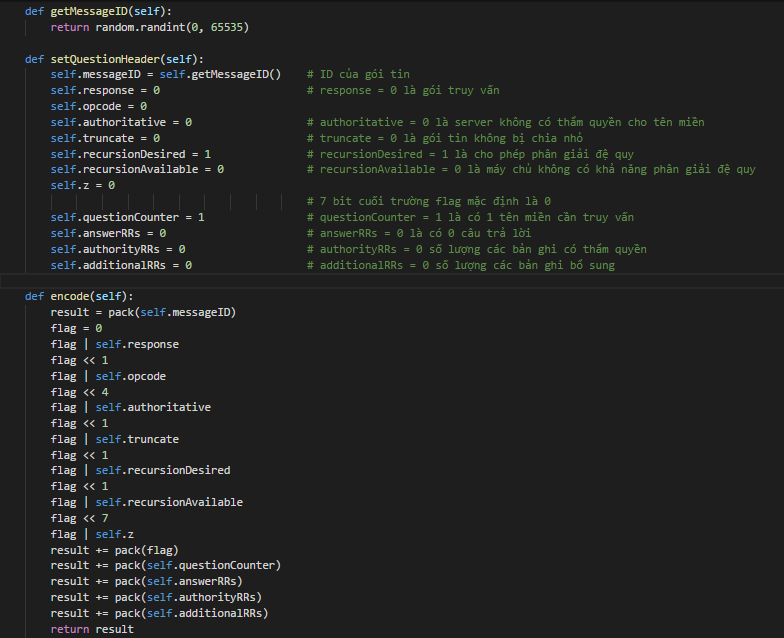
Lớp MessageHeader quy định thông tin cho 12 bytes đầu tiên, gồm có:

* ID của gói tin (random từ 0 đến 65535) : 2 bytes
* Trường cờ (mặc định của gói truy vấn có phân giải đệ quy là 0x0100):

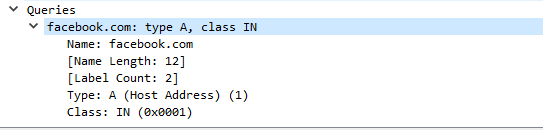
2 bytes



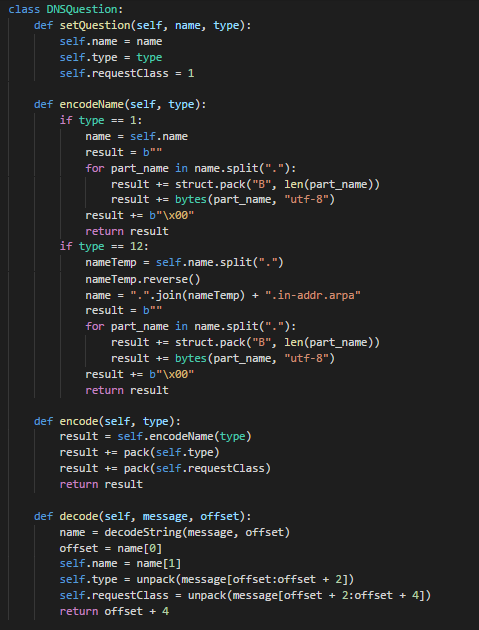
* Số lượng các tên miền cần truy vấn (mặc địng là 1): 2 bytes
* Số lượng các câu trả lời (mặc định là 0): 2 bytes
* Số lượng các bản ghi có thẩm quyền (mặc định là 0): 2 bytes
* Số lượng các bản ghi bổ sung (mặc định là 0) : 2 bytes



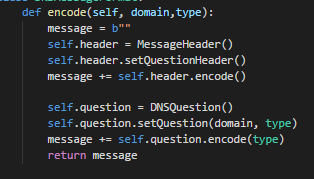
Lớp DNSQuestion quy định thông tin cho các thông tin tiếp theo:



* Chuyển tên miền hoặc địa chỉ IP sang dạng nhị phân
* Loại truy vấn: 1 là phân giải thuận, 12 là phân giải nghịch
* Mã lớp (mặc định là 1 - IN)

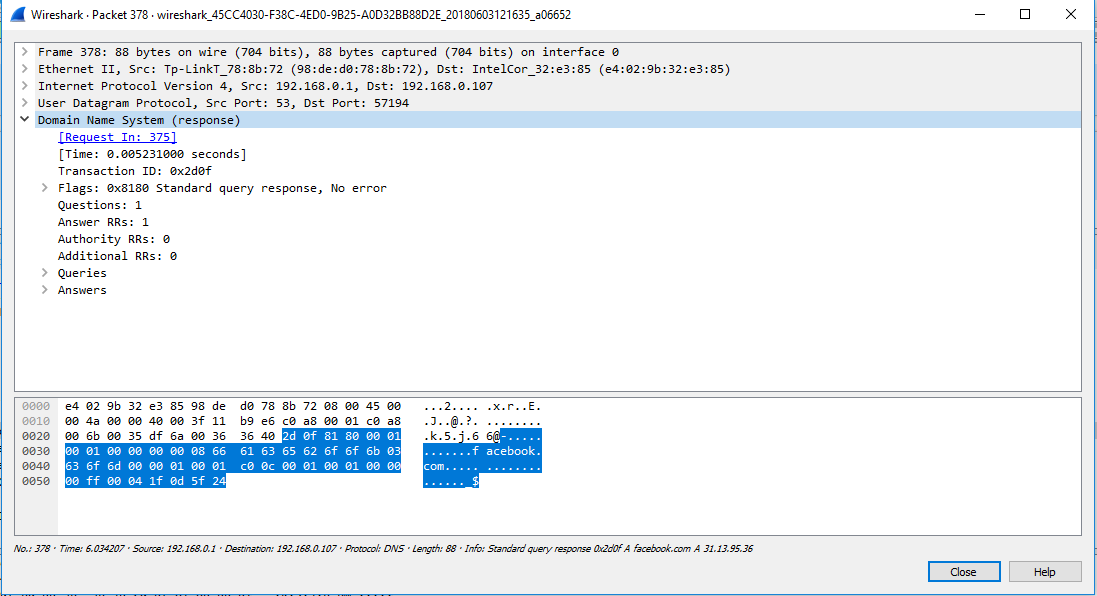


Lớp DNSMessageFormat sẽ nối 2 phần trên lại với nhau để tạo 1 gói truy vấn hoàn chỉnh, sau đó Client sẽ gửi gói tin đó đến server để yêu cầu truy vấn địa chỉ IP cho tên miền đã chọn



### Gói tin trả lời

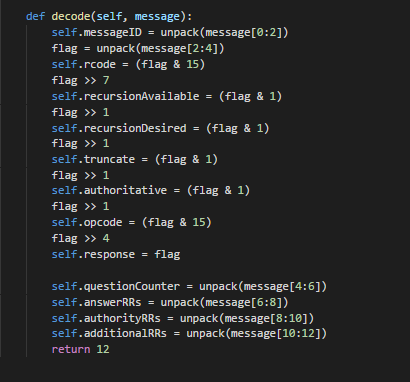
Gói tin trả lời cho truy vấn bên trên như sau:



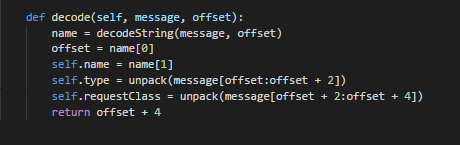
Chúng ta sẽ đọc và giải mã thông tin trong gói tin trả lời cho phần Domain Name System (response)

Các phương thức decode() trong các lớp sẽ giải mã từ các bit nhị phân sang các thông tin cần thiết, gồm các thông tin:

* Phần HeaderMessage
* Trường ID : 2 bytes
* Trường cờ : 2 bytes
* Số lượng các tên miền cần truy vấn : 2 bytes
* Số lượng các câu trả lời : 2 bytes
* Số lượng các bản ghi có thầm quyền: 2 bytes
* Số lượng các bản ghi bổ sung : 2 bytes



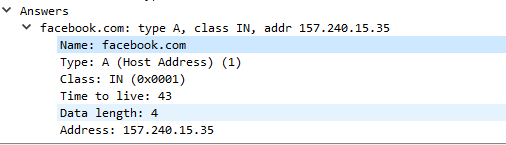
* Phần DNSQuestion



* Phần các câu trả lời

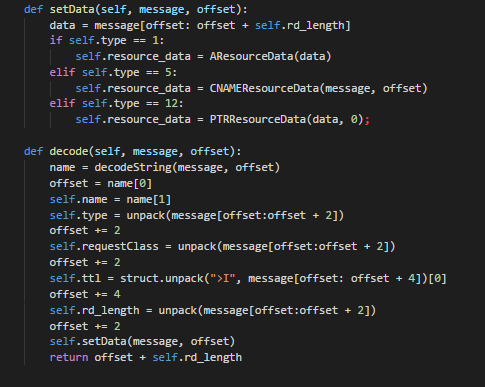
Thông tin về số lượng câu trả lời có thể lớn hơn 1, nghĩa là có thể có nhiều địa chỉ IP trả về, cần giải mã và đưa ra hết

Thông tin trường câu trả lời:



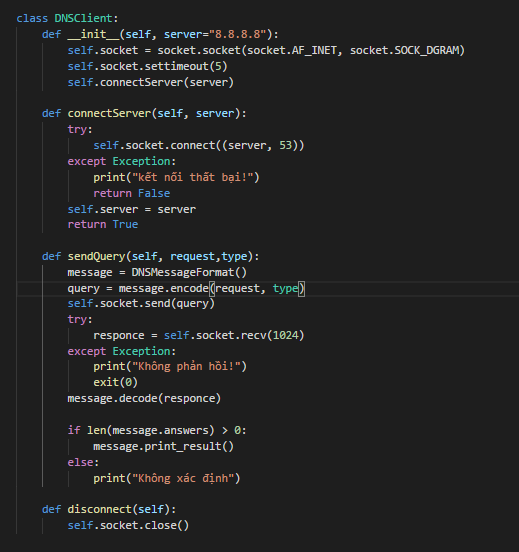
Trong đó:

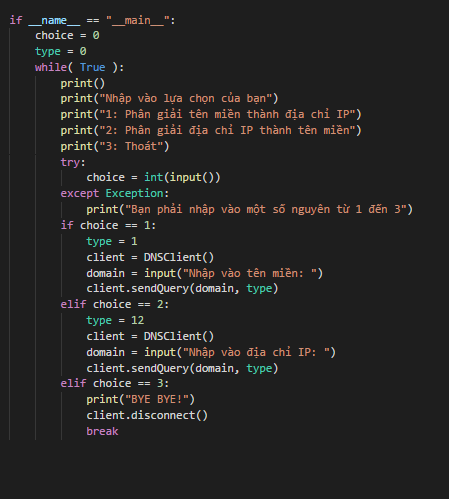
* Name: 2 bytes
* Type : 2 bytes
* Class : 2 bytes
* Time to live: 4 bytes
* Data length: 2 byte
* Address: 4 bytes (Là câu trả lời cần tìm)



## File Client.py

File DNSClient.py có 1 lớp DNSClient quy định việc tạo kết nối đến socket, truyền gói tin và đóng kết nối và chương trình chính để chạy.





## Demo chương trình



# Tài liệu tham khảo

1. COMPUTER NETWORKING A Top-Down Approach , 6th Edition, James F. Kurose, Keith W. Ross, Addison Wesley  2012
2. Data and Computer Communications, 8th Edition, William Stallings, Pearson Prentice Hall  2007
3. rfc1035
4. Giáo trình nhập môn mạng máy tính,Ths Hồ Đắc Phương, trường Đại Học Công Nghệ, Đại Học Quốc Gia Hà Nội,2009
5. <https://docs.python.org>