

# 计算机网络 Lab1

## DNS Relay

PB18111697 王章瀚

2020 年 12 月 25 日

## 目录

<b>1</b>	<b>实验目标</b>	<b>2</b>
1.1	任务描述	2
1.2	测试方法	2
<b>2</b>	<b>程序介绍</b>	<b>2</b>
2.1	文件说明	2
2.2	类与函数说明	3
2.2.1	dns.py	3
2.2.2	dns_relay.py	3
<b>3</b>	<b>程序设计特点</b>	<b>3</b>
3.1	代码运行逻辑	3
3.2	特殊功能设计	3
<b>4</b>	<b>程序运行过程与截图</b>	<b>3</b>
4.1	环境配置	3
4.1.1	实验环境	3
4.2	实验配置	4
4.2.1	设置 DNS 服务器	4
4.2.2	清空 DNS 缓存	4
4.2.3	清空浏览器缓存	4
4.3	nslookup	4
4.4	Web Browser	6
4.4.1	知乎网页广告测试	6
4.4.2	7K7K 网页图片测试	8
4.4.3	CSDN 网页广告测试	10
<b>5</b>	<b>实验总结</b>	<b>11</b>

# 1 实验目标

## 1.1 任务描述

给定一个 config 文件, 其每行都是 一个 IP 地址和一个域名, 要求实现一个 DNS 服务器以满足下列要求:

- Intercept: 如果查询的域名在列表中, 且对应为 0.0.0.0, 那么回复时应当表明无此条目, 设置 rcode 为 3
- Local Resolve: 如果查询的域名在列表中, 且对应有一个有意义的 IP 地址, 那么就正确地回复该 IP 地址
- Relay: 如果域名不在列表中, 就传递该域名给真正的 DNS 服务器, 然后将其返回结果发回客户端

此外, 还要求

- 指出每个 query 中被查询的 name
- query 处理的时长
- 能够处理并发的请求

## 1.2 测试方法

- nslookup: for basic functionality
- 浏览器: 过滤广告

# 2 程序介绍

## 2.1 文件说明

以下是代码文件结构:

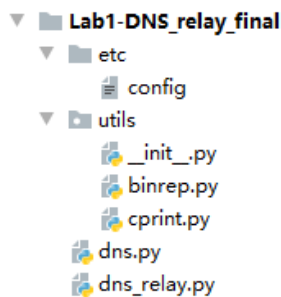


图 1: 代码文件结构

- utils.binrep: 用于 bytes, bin, binlist, int 等之间的相互转换, 主要是统一了一下函数名
- utils.cprint: colorful printer, 用来方便地进行带颜色风格等的文字输出
- dns: 用于解析 DNS 消息, 主要类包括 `DNSHeader`, `DNSQuestion`, `DNSAnswer`, 并提供了 `@property` 装饰器的参数访问方法, 提升了这些类的易用性.
- dns\_relay: 主程序. 主要包括 `receiver`, `relay`, `forward`, `backsender` 这么几个函数. 通过 `multiprocessing` 的进程调度来协调他们.

## 2.2 类与函数说明

### 2.2.1 dns.py

- `DNSHeader`: 提供完整的 DNS header 访问与修改接口
- `DNSQuestion`: 提供完整的 DNS question 访问与修改接口
- `DNSAnswer`: 提供完整的 DNS answer 访问与修改接口

### 2.2.2 dns\_relay.py

- `receiver`: 带锁机制的接收器函数, 用以接收客户端的请求
- `relay`: 进行请求解析, 根据其在 `config` 文件中的信息来决定如何返回
- `forward`: 用以转发请求给真正的 DNS server, 从而获取真正的信息
- `backsender`: 用以发回生成好的 answer

## 3 程序设计特点

### 3.1 代码运行逻辑

1. 从 `dns_relay` 中的 `main` 函数开始运行程序
2. `main` 函数会创建一个共享队列和锁, 并作为参数传给新建的两个进程 `receiver` 和 `backsender`
3. `receiver` 负责接收请求, 并 `start` 一个新的进程以调用 `relay` 来处理该请求
4. 当一个请求被处理完后, 它的相关结果被放入共享队列中
5. `backsender` 不时检查队列内容, 如有内容则发送它们, 以将结果发回客户端

### 3.2 特殊功能设计

- 彩色打印信息, 方便识别
- 自动加载更新了的 `config` 文件, 避免重新启动程序
- 相对完善的 `except` 处理
- 信号量控制进程数, 防止管道使用占满

## 4 程序运行过程与截图

### 4.1 环境配置

#### 4.1.1 实验环境

- Windows10
- Python 3.7
- Microsoft Edge 浏览器

## 4.2 实验配置

### 4.2.1 设置 DNS 服务器

为了让 DNS 请求都经过我的 DNS 服务器, 需要做如下配置



图 2: 代码文件结构

### 4.2.2 清空 DNS 缓存

只需要在命令行中运行 " `ipconfig /flushdns`" 即可

### 4.2.3 清空浏览器缓存

就一些平时常做的操作, 这里就不截图了.

## 4.3 nslookup

根据要求, 考虑以下配置:



```
PS C:\Users\89565> ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
PS C:\Users\89565> nslookup
Default Server: localhost
Address: 127.0.0.1

> picb.zhiming.com
Server: localhost
Address: 127.0.0.1

*** localhost can't find picb.zhiming.com: Non-existent domain
>

[recv query 4]: b'\x00\x04\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x04picb\x07zhiming\x03com\x00\x00\x01\x00\x01' from ('127.0.0.1', 58452)
[header] id: 4, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: picb.zhiming.com, qtype: 1, qclass: 1, next_offset: 34
[intercept 4]: b'\x00\x04\x05\x03\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x04picb\x07zhiming\x03com\x00\x00\x01\x00\x01'
[header] id: 4, rcode: 3, qdcount: 1, ancourt: 1
[question] qname: picb.zhiming.com, qtype: 1, qclass: 1, next_offset: 34
[time cost 4]: 0:00:00.204954
```

图 6: nslookup 查询 picb.zhiming.com 的结果

- 查询 `www.4399.com` 的结果: 此时它不在 `config` 文件中, 因此就进行 relay.

```
> www.4399.com
Server: localhost
Address: 127.0.0.1

Non-authoritative answer:
Name: www.4399.com.lxdns.com
Addresses: 112.30.238.103
112.30.236.25
Aliases: www.4399.com
>

[recv query 2]: b'\x00\x02\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x03www\x044399\x03com\x00\x00\x01\x00\x01' from ('127.0.0.1', 58431)
[header] id: 2, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: www.4399.com, qtype: 1, qclass: 1, next_offset: 38
[relay msg 2]: b'\x00\x02\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x03www\x044399\x03com\x00\x00\x01\x00\x01\x00\x0c\x00\x05\x00\x00\x00\x0f\x00\x15\x03www\x044399\x03com\x05'
[header] id: 2, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: www.4399.com, qtype: 1, qclass: 1, next_offset: 38
[answer] name: www.4399.com, type: 5, class: 1, ttl: 15, rdlength: 21, rdata: www.4399.com.lxdns.com
[answer] name: www.4399.com.lxdns.com, type: 1, class: 1, ttl: 15, rdlength: 4, rdata: 112.30.238.103
[answer] name: www.4399.com.lxdns.com, type: 1, class: 1, ttl: 15, rdlength: 4, rdata: 112.30.236.25
[time cost 2]: 0:00:00.203950
[recv query 3]: b'\x00\x03\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x03www\x044399\x03com\x00\x00\x1c\x00\x01' from ('127.0.0.1', 58433)
[header] id: 3, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: www.4399.com, qtype: 28, qclass: 1, next_offset: 38
[relay msg 3]: b'\x00\x03\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x00\x03www\x044399\x03com\x00\x1c\x00\x01\x00\x0c\x00\x05\x00\x01\x00\x00\x00\x15\x03www\x044399\x03com\x05'
[header] id: 3, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: www.4399.com, qtype: 28, qclass: 1, next_offset: 38
[answer] name: www.4399.com, type: 5, class: 1, ttl: 49, rdlength: 21, rdata: www.4399.com.lxdns.com
[time cost 3]: 0:00:00.204904
```

图 7: nslookup 查询 www.4399.com 的结果

## 4.4 Web Browser

### 4.4.1 知乎网页广告测试

为了测试知乎的广告, 使用以下配置:



图 8: Web Browser 测试的 config 文件

然后打开知乎, 会发现广告的图片的网址被 intercept:

```
[recv query 46105]: b'\xb4\x19\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00\x04pic1\x05zhimg\x00'
[header] id: 46105, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: pic1.zhimg.com, qtype: 1, qclass: 1, next_offset: 32
[intercept 46105]: b'\xb4\x19\x05\x03\x00\x01\x00\x00\x00\x00\x00\x00\x04pic1\x05zhimg\x00'
[header] id: 46105, rcode: 3, qdcount: 1, ancourt: 1
[question] qname: pic1.zhimg.com, qtype: 1, qclass: 1, next_offset: 32
```

图 9: 访问知乎首页的输出效果

从而使浏览器无法收到广告的图片无法接受, 最终使得广告无法显示:



图 10: 去除广告前的知乎(红线框出)



图 11: 去除广告后的知乎

#### 4.4.2 7K7K 网页图片测试

同理,



```
112.28.235.150 www.7k7k.com
0.0.0.0 i.7k7king.cn
0.0.0.0 i1.7k7king.cn
0.0.0.0 i2.7k7king.cn
0.0.0.0 i3.7k7king.cn
0.0.0.0 i4.7k7king.cn
0.0.0.0 i5.7k7king.cn
0.0.0.0 i6.7k7king.cn
0.0.0.0 i7.7k7king.cn
0.0.0.0 i8.7k7king.cn
0.0.0.0 i9.7k7king.cn
```

图 12: Web Browser 测试的 7k7k config 文件



图 13: 去除图片前的 7k7k



图 14: 去除图片后的 7k7k

#### 4.4.3 CSDN 网页广告测试

同理,

```
47.95.164.112 www.csdn.net
0.0.0.0 img-bss.csdn.net
```

图 15: Web Browser 测试的 csdn config 文件

```
[2020-12-25 11:40:24.711597][recv query 31667]: b'\xb3\x05\x00\x01\x00\x00\x00\x00\x00\x00\x00\x07img-bss\x04csdn\x03net\x00\x00\x01\x00\x01' from ('127.0.0.1', 54543)
[header] id: 31667, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: img-bss.csdn.net, qtype: 1, qclass: 1, next_offset: 34
[2020-12-25 11:40:24.715596][recv query 38224]: b'\x95P\x05\x00\x00\x01\x00\x00\x00\x00\x00\x00\x07img-bss\x04csdn\x03net\x00\x00\x1c\x00\x01' from ('127.0.0.1', 63102)
[header] id: 38224, rcode: 0, qdcount: 1, ancourt: 1
[question] qname: img-bss.csdn.net, qtype: 28, qclass: 1, next_offset: 34
[2020-12-25 11:40:25.131493][intercept 31667]: b'\xb3\x05\x03\x00\x01\x00\x00\x00\x00\x00\x00\x07img-bss\x04csdn\x03net\x00\x00\x01\x00\x01'
[header] id: 31667, rcode: 3, qdcount: 1, ancourt: 1
[question] qname: img-bss.csdn.net, qtype: 1, qclass: 1, next_offset: 34
[2020-12-25 11:40:25.131493][time cost 31667]: 0:00:00.419896
[2020-12-25 11:40:25.131493][intercept 38224]: b'\x95P\x05\x03\x00\x01\x00\x00\x00\x00\x00\x00\x07img-bss\x04csdn\x03net\x00\x00\x1c\x00\x01'
[header] id: 38224, rcode: 3, qdcount: 1, ancourt: 1
[question] qname: img-bss.csdn.net, qtype: 28, qclass: 1, next_offset: 34
[2020-12-25 11:40:25.131493][time cost 38224]: 0:00:00.415897
```

图 16: 访问知乎首页的输出效果

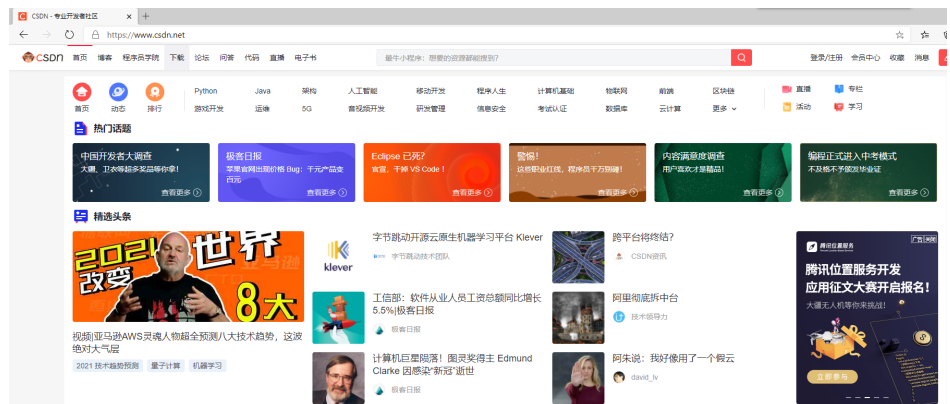


图 17: 去除广告前的 csdn

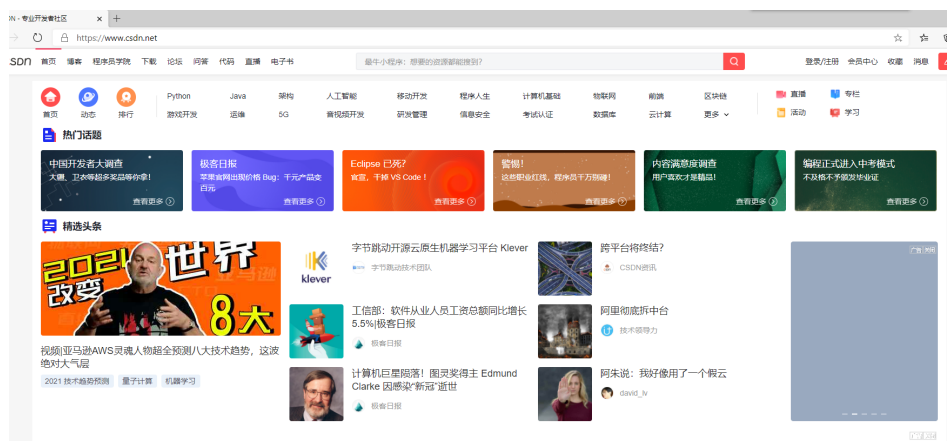


图 18: 去除广告(屏幕右侧)后的 csdn

## 5 实验总结

本次实验中, 不仅更深刻地了解了 DNS 请求的相关协议, 且也更熟悉了 Python 的代码编写, 包括但不限于其 socket, multiprocessing 的使用等, 收获颇丰.