

计算机网络课程设计

DNS 中继服务器实验报告

班 级： 2014211306
学 号： 2014211281
姓 名： 黄智伟
教 师： 高占春

2017.5.25

目录

1	系统的功能设计	2
2	模块划分	2
3	软件流程图	3
4	测试用例以及运行结果	4
5	调试中遇到并解决的问题	7
6	课程设计工作总结	7

1 系统的功能设计

设计一个 DNS 服务器程序，DNS 中继服务器读入“IP 地址-域名”对照表，并以此响应客户端的 DNS 请求。当客户端查询域名对应的 IP 地址时，用域名检索该对照表，产生三种可能检索结果：

1. 当检索到的 IP 地址为 0.0.0.0，回复“域名不存在”的报错消息，实现不良网站拦截的功能；
2. 当检索为普通 IP 地址，则向客户端返回该地址，实现服务器功能；
3. 当表中未检到该域名，则向因特网 DNS 服务器发出查询，并将结果返给客户端，实现中继功能。

考虑多个计算机上的客户端会同时查询，需要进行消息 ID 的转换。同时 DNS 中继服务器还实现了可以指定远程 DNS 服务器 IP 地址，指定“IP 地址-域名”对照表的文件位置以及多种级别的调试信息的输出。

2 模块划分

1. 读取文件模块: 从用户指定的路径读入“IP 地址-域名”对照表
2. 运行模块: 接收客户端请求, 并放入线程池中运行
3. 处理请求模块: 解析请求, 并进行相应的处理
4. 构建数据包模块: 按照 RFC1035 协议构造出相应回复的数据包
5. 转发模块: 进行 ID 转换, 转发客户端请求给远端 DNS 服务器, 接受远程客户端的回复并转发给客户端

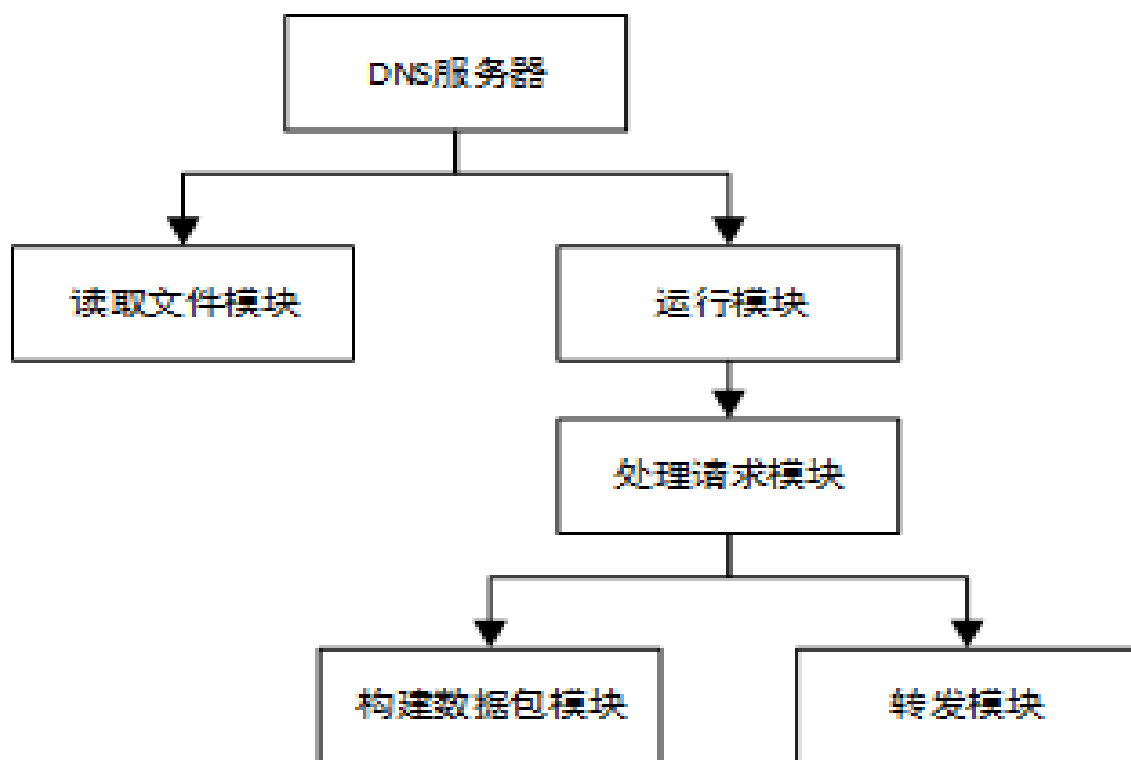


图 1: 模块调用图

3 软件流程图

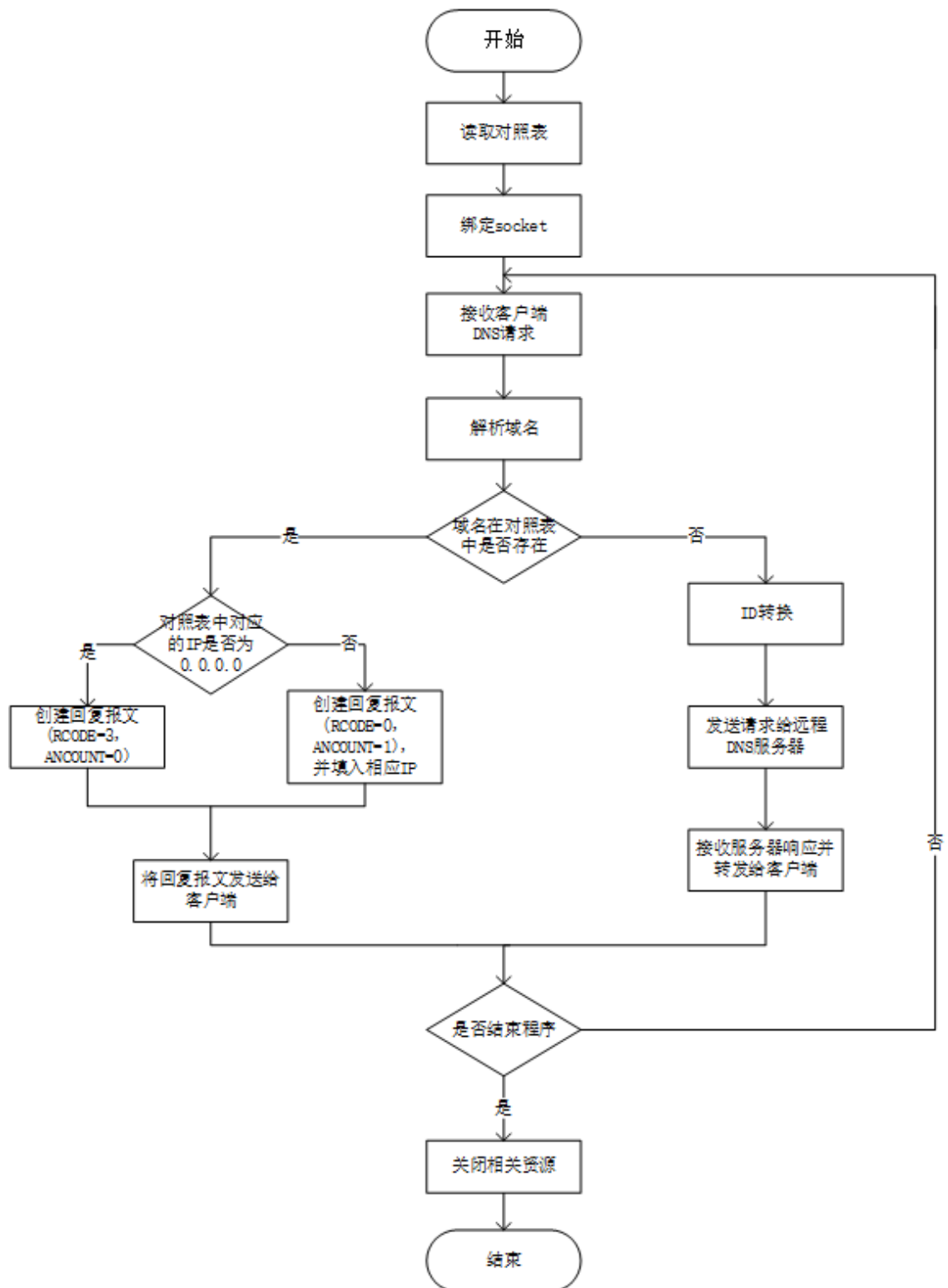


图 2: 软件流程图

4 测试用例以及运行结果

1. 命令行提示:

```
~/Document/Program/DNS-Reply > P master± sudo python3 DNS-Reply.py -h
usage: DNS-Reply.py [-h] [-d] [-dd] [dnsServerIp] [dbFile]

This is a DNS relay.

positional arguments:
  dnsServerIp  DNS server ipaddr
  dbFile       DB filename

optional arguments:
  -h, --help  show this help message and exit
  -d          Debug level 1
  -dd         Debug level 2
```

2. 使用默认参数:

```
~/Document/Program/DNS-Reply > P master± sudo python3 DNS-Reply.py
NameServer: 10.3.9.5
DB file: ./dnsrelay.txt
Debug level: 0
```

运行结果:

```
~ > nslookup sina. 127.0.0.1
Server:                127.0.0.1
Address:               127.0.0.1#53

Non-authoritative answer:
Name:   sina
Address: 202.108.33.89

~ >
```

3. 调试级别 1, 指定远程 DNS 服务器地址, 指定对照表文件位置:

```
~/Document/Program/DNS-Reply > P master± sudo python3 DNS-Reply.py -d 10.3.9
5 ./dnsrelay.txt
NameServer: 10.3.9.5
DB file: ./dnsrelay.txt
Debug level: 1
2017-05-29 13:24:08 try to load table...OK

2017-05-29 13:25:48 No:1 QName:sina
```

运行结果:

```
~ ➔ nslookup sina. 127.0.0.1
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
Name:   sina
Address: 202.108.33.89

~ ➔
```

4. 调试级别 2, 指定远程 DNS 服务器地址, 指定对照表文件位置:

```
~/Document/Program/DNS-Reply ➔ master ➔ sudo python3 DNS-Reply.py -dd 10.3.9
.5 ./dnsrelay.txt
NameServer: 10.3.9.5
DB file: ./dnsrelay.txt
Debug level: 2
2017-05-29 13:29:03 try to load table...OK

2017-05-29 13:29:09 reveive from:('127.0.0.1', 49647) request:b'\xbd\xac\x01\x00
\x00\x01\x00\x00\x00\x00\x00\x00\x04sina\x00\x00\x01\x00\x01'
2017-05-29 13:29:09 No:1 QName:sina
2017-05-29 13:29:09 ID:1011110110101100 QR:0 OPCODE:0000 QTYPE:0000000000000001
QCLASS:000000000000000001
2017-05-29 13:29:09 find IP in local server
2017-05-29 13:29:09 creating response
2017-05-29 13:29:09 sending response...OK
```

运行结果 (查找对照表里存在且不为"0.0.0.0" 的域名, 即服务器功能):

```
~ ➔ nslookup sina. 127.0.0.1
Server:          127.0.0.1
Address:         127.0.0.1#53

Non-authoritative answer:
Name:   sina
Address: 202.108.33.89

~ ➔
```


5 调试中遇到并解决的问题

1. 问题: 刚开始发现域名字段是不定长, 不会解析域名字段
解决: 查阅资料后发现域名是先由数字表明分段的字符个数, 再接着字符 (省略小数点?), 字符个数为 0 时域名字段才结束, 之后开始编码解析域名字段
2. 问题: 查询对照表中存在且不值为"0.0.0.0" 的域名时, 返回的 IP 地址与对照表中相应域名对应的 IP 不一致
解决: 查阅资料得, 网络字节流是大端序, 与 Intel x86 字节流的小端序正好相反, 故错误. 之后修改代码, 问题解决
3. 问题: 由于使用了多线程, 访问和修改一些线程间的共享资源存在问题, 导致出现错误
解决: 采用线程锁, 解决线程间共享资源的访问与修改
4. 问题: 由于使用了多线程, 同时处理多个客户端请求, 当多个客户端请求均为对照表中不存在的域名时, 向远程 DNS 服务器转发存在 ID 一致的问题
解决: 进行 ID 转换, 转换规则为: 若存在与自己 ID 一致的请求, 则 ID 加 1
5. 问题: 当客户端查询对照表中不存在的域名时, 程序向远程 DNS 服务器转发, 有时远程 DNS 服务器响应时间过长, 导致线程阻塞
解决: 使用定时器, 设置远程服务器 3.5 秒未响应, 则停止等待, 释放资源, 且不回复客户端, 使客户端超时重发

6 课程设计工作总结

此次课程设计为实现一个 DNS 中继服务器。具有不良网站拦截功能,DNS 服务器功能及中继功能这三个基本功能, 同时具有指定远程 DNS 服务器 IP 地址及本地"IP-域名" 对照表文件位置的功能。

由于此次课程设计依赖于 RFC 1035 协议, 所以在进行实际编码实现之前, 我首先阅读了 RFC 1035 文档. 由于文档是英文编写的, 刚开始学习时, 遇到了很多问题. 但是在询问老师及同学之后, 对文档有了一定的了解, 之后开始编码实现.

此次的课程设计, 使用到了 socket 编程以及 wireshark 等抓包软件, 我也从中学到了很多. 通过此次课程设计, 我掌握了一定的 socket 编程知识, 也对 DNS 数据报的格式有了一定的了解, 同时也明白了 DNS 服务器的工作原理及流程.

在编程实现过程中, 由于使用了多线程, 因此进行了 ID 转换, 同时也对线程间的共享资源进行了加锁操作, 以便共享资源能正确的被访问. 虽然在这期间遇到了很多问题, 但是也让我更加了解了多线程编程以及锁机制,. 同时为了防止线程阻塞, 对 socket 进行了定时操作, 超过一定时间未回复, 则放弃等待, 释放资源, 处理下一请求. 这是为了服务器的性能做出的考虑, 这也让我明白了有时候为了性能, 不得不放弃一些服务质量.