**北京邮电大学课程设计报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **课程设计**  **名称** | **计算机网络课程设计** | | **学 院** | **计算机** | **指导教师** | **高占春** |
| **班 级** | **班内序号** | **学 号** | | **学生姓名** | **成绩** | |
| **2017211303** | **25** | **2017211195** | | **杨华林** |  | |
| **2017211303** | **27** | **2017211197** | | **赵永宁** |  | |
| **2017211303** | **19** | **2017211187** | | **张萧枫** |  | |
| **课**  **程**  **设**  **计**  **内**  **容** | 简要介绍课程设计的主要内容，包括课程设计教学目的、基本内容、实验方法和团队分工等  课程介绍：本课程以目前广泛使用的 TCP/IP 协议栈为基础，使用 UDP 或 TCP 通信平台，利用 SOCKET 编程接口，实现 DNS 中继器软件。通过本课程的学习，熟练掌握了 SOCKET 编程接口的使用、能够使用有限状态机对通信协议进行描述和设计、掌握计算机网络编程和测试的基本方法和相关技术，提高进行计算机网络与通信系统的设计与开发和实际环境中设计开发系统的能力。  实验方法：使用Python语言，在windows环境下编码，同时使用本机进行DNS查询，浏览网页、发送邮件测试。  团队分工：  杨华林：设计、编码、测试  张萧枫：设计、编码、测试  赵永宁：编码、测试 | | | | | |
| **学生**  **课程设计**  **报告**  （附页） |  | | | | | |
| **课**  **程**  **设**  **计**  **成**  **绩**  **评**  **定** | **评语**:  **成绩**:  指导教师签名：  年 月 日 | | | | | |

注：评语要体现每个学生的工作情况，可以加页。

# 一、系统的功能设计

**1.1 基本功能**

设计一个DNS服务器程序，读入“域名-IP地址”对照表，当客户端查询域名对应的IP地址时，用域名检索该对照表，三种检索结果：

1) 检索结果为ip地址0.0.0.0，则向客户端返回“域名不存在”的报错消息（不良网站拦截功能）

2) 检索结果为普通IP地址，则向客户返回这个地址（服务器功能）

3) 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（中继功能）

**1.2 服务配置**

我们提供了配置文件和命令行选项来配置服务的相关选项。可以配置ip-域名对照表文件所在的位置，上层DNS服务器配置，并提供了两个级别的调试选项。我们提供了一个示例json配置文件和-h（帮助命令）的命令选项来让用户快速配置服务。

**1.3 稳定运行**

1）快速处理请求。虽然所用语言为Pyhton，但查询ip-域名和ID-ip对照表时用Python的字典，时间复杂度接近O(n)，所以可以高速的处理一条DNS请求，即使ip-域名记录较多和DNS请求较多导致对照表比较大时，性能也不会下降的很快。

2）处理ID重复。当查询的ID有重复时，会生成一个不重复的随机ID替换然后提交给上层DNS服务器，来保证服务的稳定。

3）避免内存增长。对于上层DNS未回复的DNS请求，程序会定期清理，保证ID-ip对照表没有无用的项目，保证程序长时间运行占用的内存也不会不合理的增长，确保服务的快速响应。

**注：ID-ip对照表为转发给上层DNS服务器的报文，记录该报文的ID和应回复的IP和端口号的哈希表（字典）。**

# 二、模块划分

DNS中继服务器程序由python语言写成，在Windows环境下进行开发，共包括两个.py源代码文件（DNS\_Relay.py,DNS\_struct.py）和一个.json配置文件settings.json。

其中，DNS\_struct.py文件中定义了三个类，分别用来分析报文，改变报文，存储所读取的指定的配置文件中的ip-域名对照表。

而DNS\_relay.py文件中则包含了main函数在内的几个重要函数，实现了不良网站拦截，服务器功能和中继功能。

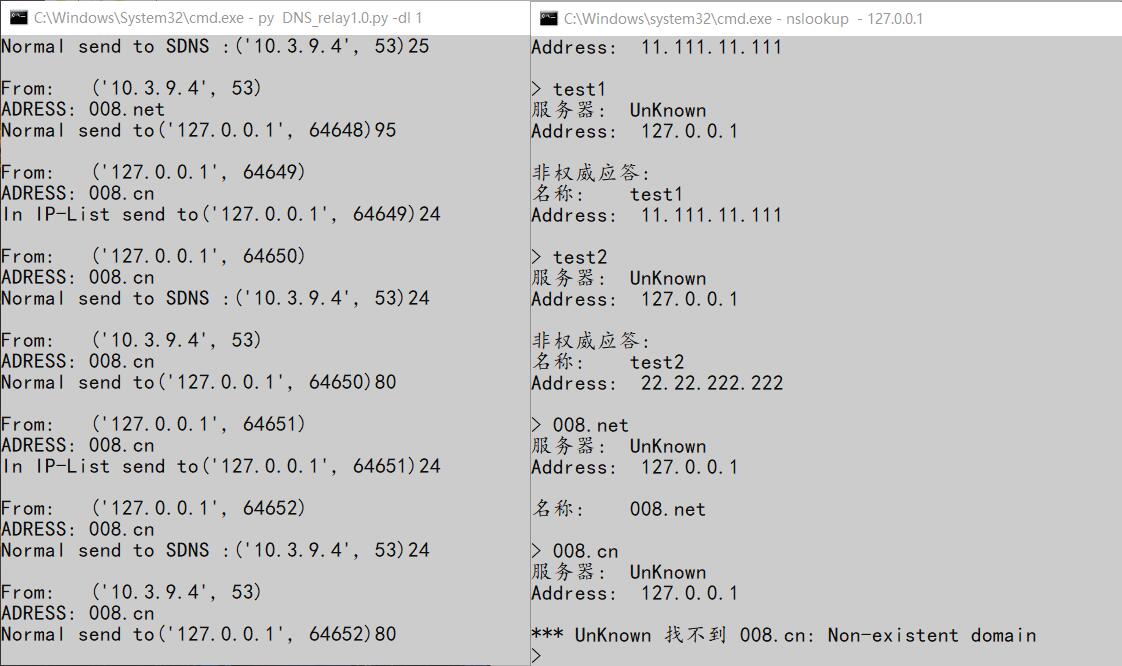


# 三、软件流程图

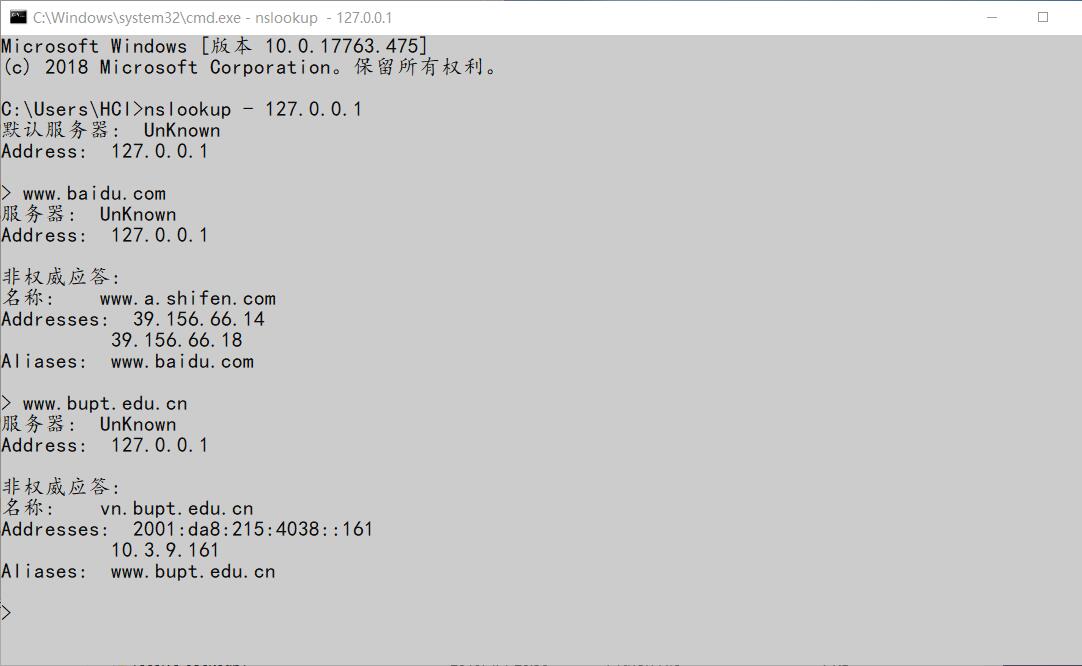


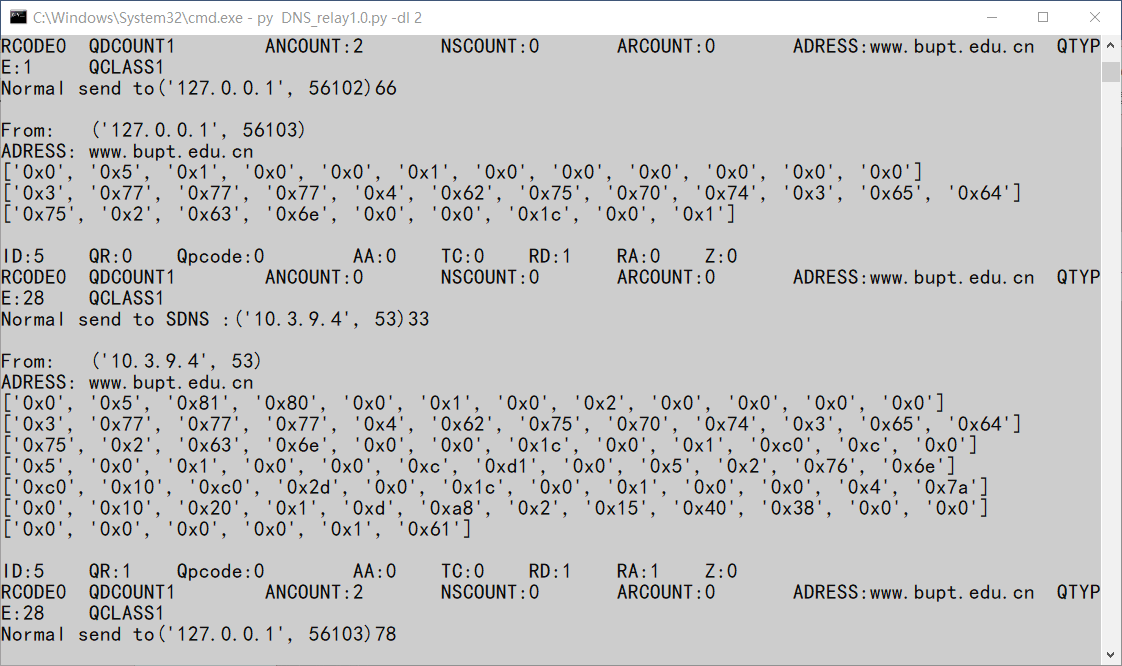
# 四、测试

**4.1** 调试等级1的情况下测试不良网站拦截和提供DNS解析功能，测试结果正确，符合预期。



**4.2** 调试等级为2时，测试中继功能，结果符合预期。





**4.3** 将本机DNS服务程序设为本地，进行ping，浏览网页，发邮件测试，均通过。

**4.4** 性能测试评估

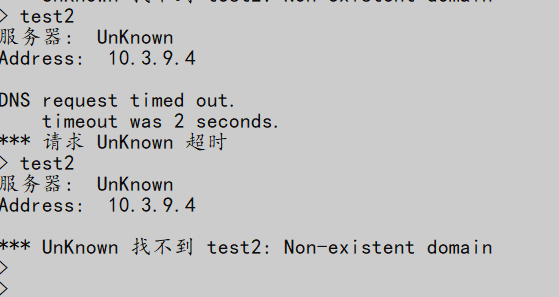
通过对处理每一条DNS请求所用时间的测试有。处理一条DNS请求所有的时间为10^-5秒，包括接收和发送的时间为1\*10^-4到2\*10^-4秒，故在测试机上一秒可以处理5000条不到的请求，且瓶颈在于网络，如果发送包的速度更快，每秒理论可处理上万条请求。

# 问题与解决方案

**1.** 程序刚开始遇到的一个BUG，在电脑绑定本地的DNS之后，浏览网页偶尔出现加载不出来的情况，抓包分析之后，发现是修改DNS请求包ID时，误将ANCOUNT数目置为1然后发给上层DNS服务器，上层DNS服务器会将该错误的包直接返回给我，然后程序就会产生一个错误的包给请求者。

**2.** 程序一直会抛出一个10054异常，调试了好久。因为这个关于这个错误的描述是“远程主机强行关闭了一个现有的连接”。这令我产生了许多疑惑，UDP是无连接的，为什么存在关闭一说。后来了解到如果你发送的地址未被监听就会报出该错误。就一直找了好久程序设计的问题，因为一直觉得是回复DNS报文地址搞错了。后来通过抓包分析得到，有时候有些程序的DNS请求报文会发送好几个ID相同的报文，而它收到一个后就停止监视这个端口了，然后引起了一个异常。然后修改程序，对于相同地址相同ID的报文只会回复一个，该异常停止抛出。

**3.** 有时候测试某一地址，会遇到超时情况，再次查询立刻回复。查了好久的BUG，后来发现是上层DNS服务器没有及时回复导致超时。有图为证。用nslookup直接测试学校DNS服务器发生超时情况。重复立刻回应，应该是第二次查询有缓存。



# 课程设计工作总结

**一）工作总结**

本次大作业为实现一个DNS中继服务器。通过完成本地域名解析、中继功能以及不良网站拦截等基本的三个功能，来实现用户访问外部服务器的需求。通过不断的测试和调试，完成了DNS中继器的基本功能以及提供了一些配置方式，提高程序运行的稳定性。

这次大作业的完成，需要我们阅读并了解RFC-1035（DSN协议）文档。一开始的时候对DNS协议完全没有一个清晰地概念，只知道这是一个规格说明，而且初读其中的内容也觉得一头雾水。但后来查阅相关资料后抓住了重点，然后一遍一遍反复的阅读，不懂的地方借助网络和图书资料，组员之间的积极交流和讨论渐渐明晰了我们需要做到的工作。最终我们对DNS协议和DNS服务器工作方式有了深刻的理解。

通过本次网络实验，我们掌握了小型网络应用程序的实现，通过抓包分析DNS数据报文，对传输层和应用层的数据包发送和接收等工作原理也有了更深的理解。同时掌握了网络编程的套接字的使用，学会了网络编程的调试方法。

**二）个人感悟**

1.赵永宁：本次课程设计实验让我感触最深的还是对两年来编程能力的反映。自身对编程略显生疏，编程能力亟待提高，而同组其他两位同学不仅熟练掌握多门课程内涉及到的语言，还对python有一定的掌握。实验开始后两位同学就语言方面对我进行了引导，包括相关书籍的推荐，IDE环境的安装配置，某些重点章节的帮助理解等等。在实验中遇到的许多问题，他们也都一一为我耐心解答。这次试验不仅让我学到了不少知识，激发了对网络工程的好奇，还让我意识到了与其他同学的差距，为我指引了学习的方向。我会在日后的学习和实践中不断提升自己的能力，尽快跟上大家的脚步。

2.张萧枫：本次课程设计实验，使我对python语言的使用更加熟练，熟悉了命令行，多线程和使用.json文件的使用，对于socket编程有了更加直观的认识，对于TCP\IP模型的理解和对于 OSI模型的了解更加深入，并再次体验了面向对象编程的思想和模块化编程的思想。

在两位同学的帮助下，本次实验进行的过程相对顺利，但是也遇到了不好解决的问题。在进行是否使用多线程的讨论中，认识到在I/O统一，处理时间占用非常少的情况下，多线程并不能提高系统的效率，反而会增加系统的负载。而在增加缓存功能的时候，在初步实现了缓存的模型时，认识到回复报文的格式可能遇到较为复杂的情况，不利于缓存，为保证回复的准确性，最终放弃了缓存功能。

本次的学习让收获了许多，希望在之后的学习中能够学习到更多的内容。

3.杨华林：这次实验让我学会了网络通信的基本知识，学会了利用SOCKET进行UDP的通信。让我对OSI模型有了更加深入的理解和掌握。这次实验也提升了我的编程技术，让我对Python的运用更加熟练，同时也让我学会了调试网络程序，在程序出错时认真分析原因，能够分析出BUG时从哪引起的会事半功倍。同时让我知道了网络抓包对于调试网络程序的关键。由于网络程序涉及多个程序，你不能限制别的程序给你的数据报是什么样子的，只能通过抓包来分析别人给你的信息然后看是否有遗漏的情况。