**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 | 计算机学院 | 指导教师 | 吴起凡 |
| 班 级 | 班内序号 | 学 号 | 学生姓名 | 成绩 | |
| 310 | 02 | 2021211221 | 梁骋 |  | |
| 310 | 08 | 2021211229 | 朱宏明 |  | |
| 310 | 18 | 2021211240 | 郭政勋 |  | |
|  |  |  |  |  | |
| 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  （详见下方课程设计报告） | | | | |
| 学生  课程设计  报告  （附页） |  | | | | |
| 课  程  设  计  成  绩  评  定 | 遵照实践教学大纲并根据以下四方面综合评定成绩：  1、课程设计目的任务明确，选题符合教学要求，份量及难易程度  2、团队分工是否恰当与合理  3、综合运用所学知识，提高分析问题、解决问题及实践动手能力的效果  4、是否认真、独立完成属于自己的课程设计内容，课程设计报告是否思路清晰、文字通顺、书写规范  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。

目录

[1.系统功能设计 3](#_Toc138591728)

[2.模块划分 3](#_Toc138591729)

[3.软件流程图 4](#_Toc138591730)

[4.测试用例及运行结果 5](#_Toc138591731)

[5.调试中遇到并解决的问题 9](#_Toc138591732)

[6.心得体会 9](#_Toc138591733)

# 1.系统功能设计

DNS中继服务器可用于转发DNS查询请求和响应消息。我们所设计的DNS服务器主要功能包括：

* 接收来自DNS客户端的查询请求消息，解析并转发给其他DNS服务器；
* 接收来自其他DNS服务器的响应消息，解析并转发给DNS客户端；
* 在DNS查询过程中，自动缓存查询结果，加快后续查询速度；
* 支持DNS消息ID转换、过滤等高级功能，以提升网络安全性和可靠性；
* 支持并发，可同时为多个DNS客户端服务。

# 2.模块划分

基于上述功能设计，DNS中继服务器可以划分为以下几个模块：

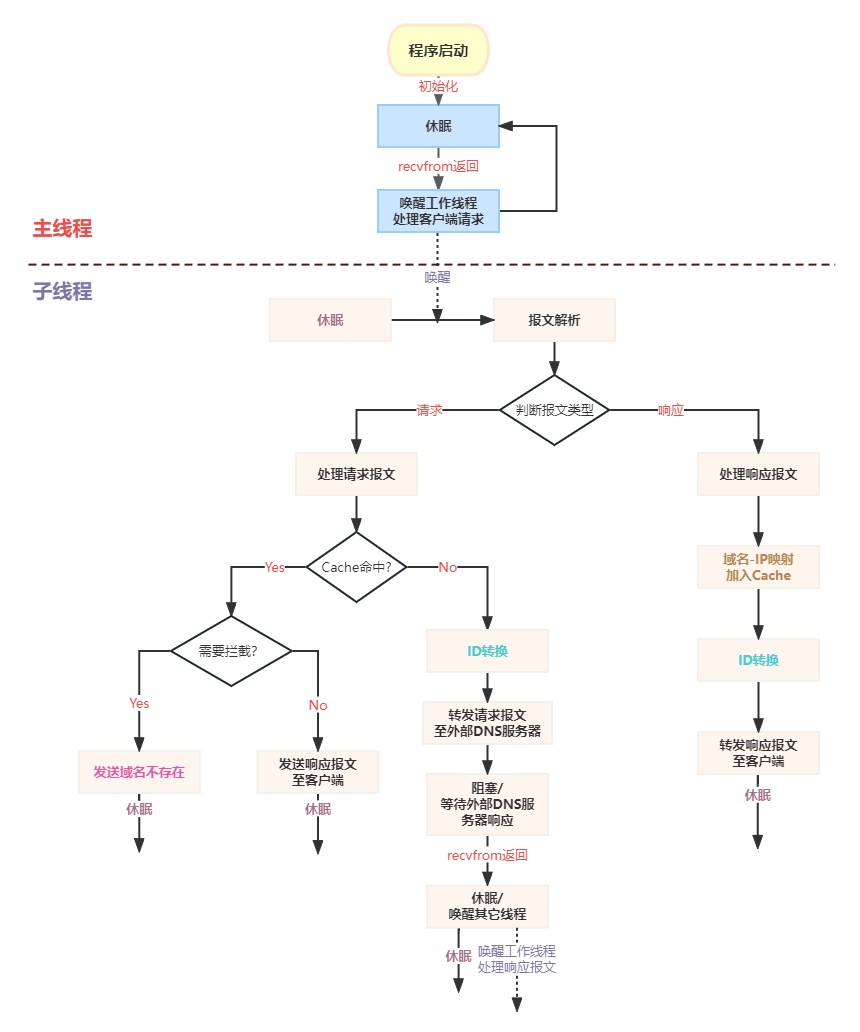
* DNS消息解析模块：用于解析收到的DNS消息，提取出查询或响应相关的信息，如查询类型、查询名称、响应结果等；
* DNS消息生成模块：用于生成需要发送的DNS消息，根据收到的查询请求和响应结果，生成相应的DNS消息并发送出去；
* DNS缓存管理模块：用于管理DNS查询结果的缓存，支持缓存的添加、更新、删除等操作，以加快后续查询速度；
* DNS消息过滤模块：用于过滤不安全、不合法或不必要的DNS消息，以提高网络安全性和可靠性；
* 网络接口模块：用于接收和发送DNS消息；
* 系统管理模块：用于管理DNS中继服务器的配置信息、运行状态和日志记录等，以便于运维管理。

在实际的工程中包含6个重要的源文件，用于实现上述模块的功能：

* dnsrelay.c：实现程序的初始化，以及程序主循环入口；
* socket.c：实现接收和发送DNS消息，并对DNS消息进行处理；
* cache.c：实现DNS缓存（使用字典树）；
* config.c：实现配置信息读取；
* thread.c：实现 “线程池”的初始化，提供工作线程的入口函数（使用Windows API），以及ID转换；
* log.c：实现DNS中继服务器的运行状态以及日志记录输出。

# 3.软件流程图

下面是DNS中继服务器的软件流程图：



关于线程：

* 主线程接收到DNS客户端的请求后，在线程池中唤醒一个工作子线程处理该请求；
* 工作子线程被唤醒后，可以根据传递的DNS报文类型进行相应的处理；
* 工作子线程也可以唤醒另一个子线程，在转发接收DNS报文时会出现这种情况；
* 基本思想是一个子线程只处理一个DNS报文，子线程接收到的DNS报文（而非其他线程所传递的DNS报文）必须唤醒线程池中的另一个线程进行处理（将接收到的DNS报文传递过去）。

# 4.测试用例及运行结果

以下是DNS中继服务器的测试用例，以及运行结果：

1. 首先清空DNS缓存，运行DNS中继服务器
2. 使用命令 ipconfig/flushdns清空缓存
3. 运行dnsrelay.exe
4. 查询 www.bing.com 的IP地址：
   * DNS中继服务器接收到查询请求；
   * 解析查询请求，并从缓存中查找相关的IP地址信息；
   * 如果缓存中存在相关信息，则返回缓存中的结果；
   * 如果缓存中不存在相关信息，则将查询请求转发给上游DNS服务器；
   * 接收到上游DNS服务器的响应消息，解析响应消息，并将结果添加到缓存中，然后将结果返回给DNS客户端。

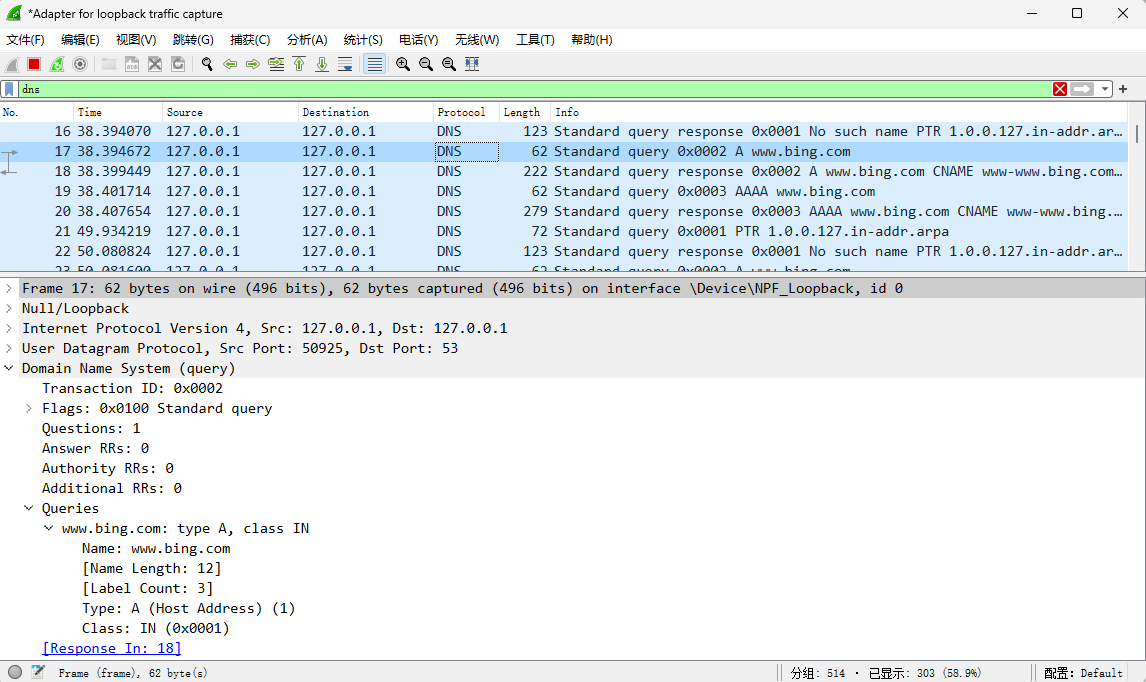


图2.1

1. 使用命令nslookup [www.bing.com](http://www.bing.com)

此时Cache中没有缓存[www.bing.com](http://www.bing.com)的IP地址，要向上游DNS服务器转发请求，并接收上游DNS服务器的响应，将该响应转发给DNS客户端。

图2.1显示了DNS客户端发出的DNS请求报文，用来查询[www.bing.com](http://www.bing.com)的IP地址。

图2.2显示了DNS中继服务器向上游DNS服务器转发的DNS请求报文。

图2.3显示了上游DNS服务器向DNS中继服务器发送的DNS响应报文。

图2.4显示了DNS中继服务器向DNS客户端转发的DNS响应报文。

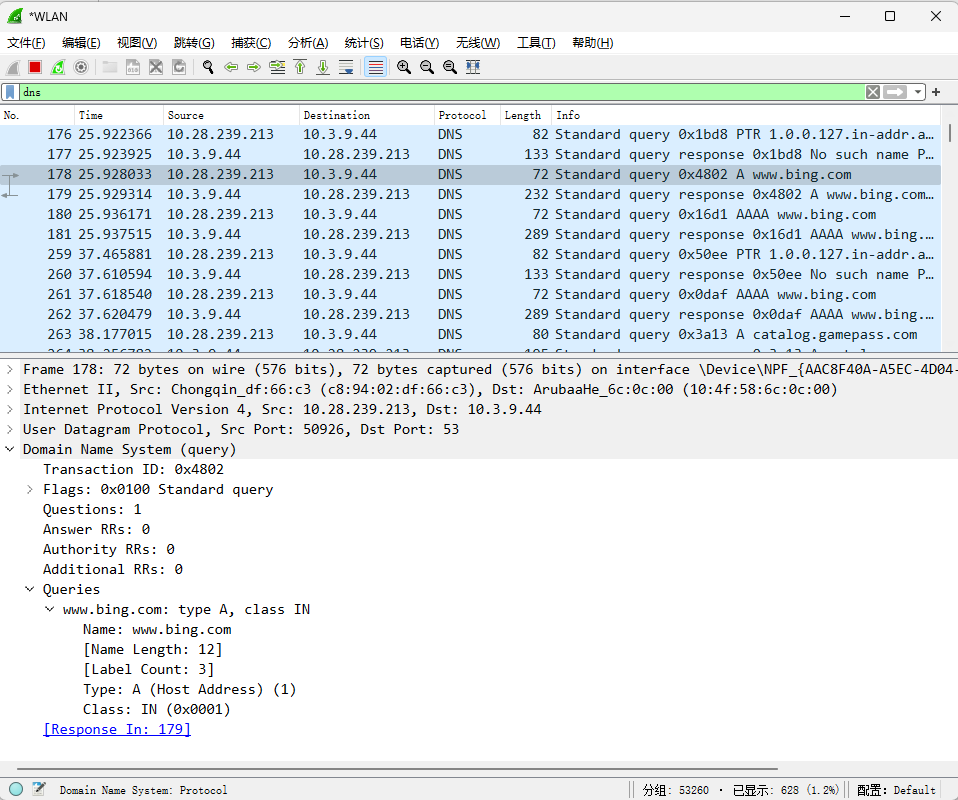


图 2.2

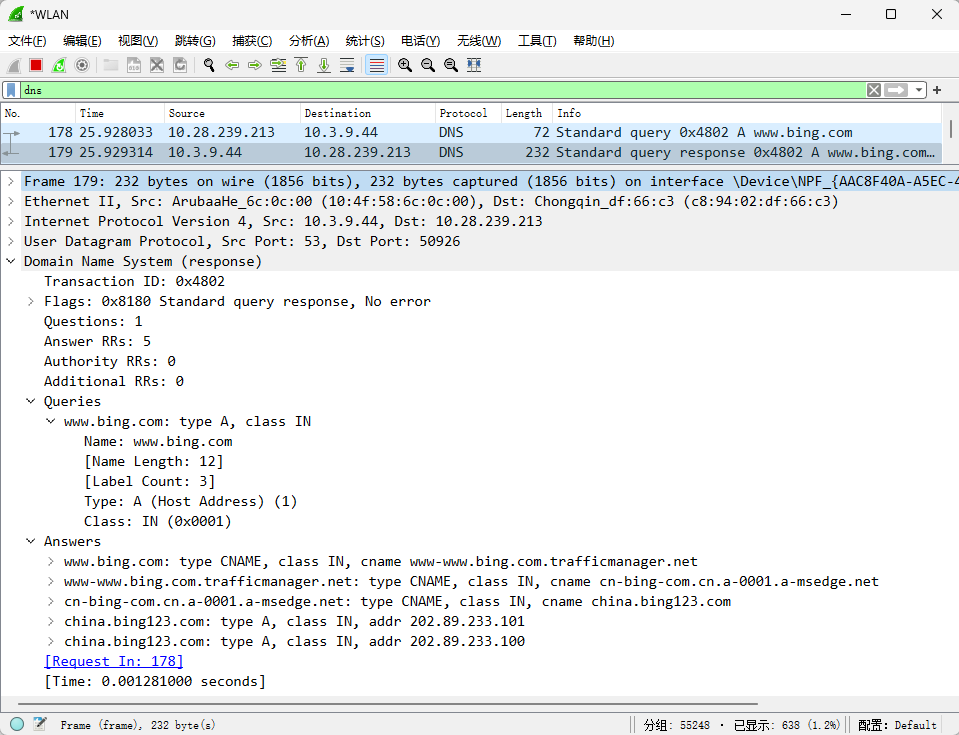


图 2.3

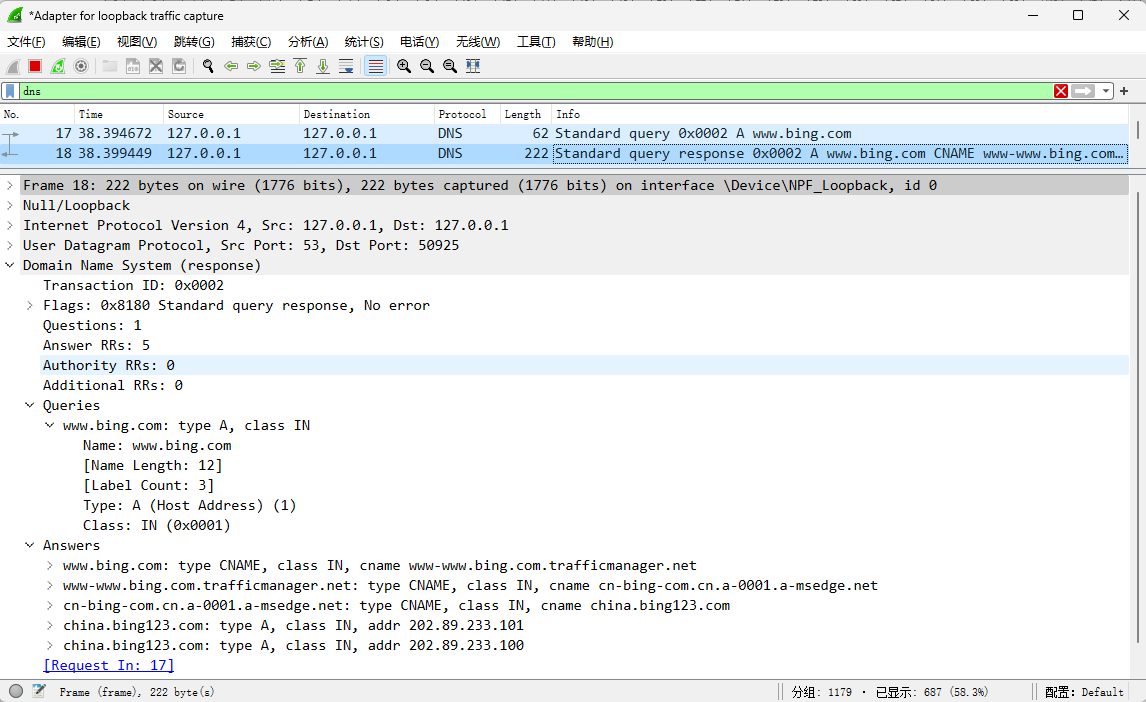


图 2.4

1. 使用命令nslookup [www.bing.com](http://www.bing.com)

此时Cache中有[www.bing.com](http://www.bing.com)的IP缓存，直接生成响应报文发送给DNS客户端。

图2.5显示了DNS中继服务器向DNS客户端发送的直接生成的响应报文，Cache中只缓存IP地址，而不缓存别名，所以生成的报文的Answer部分没有CNAME类型的记录。

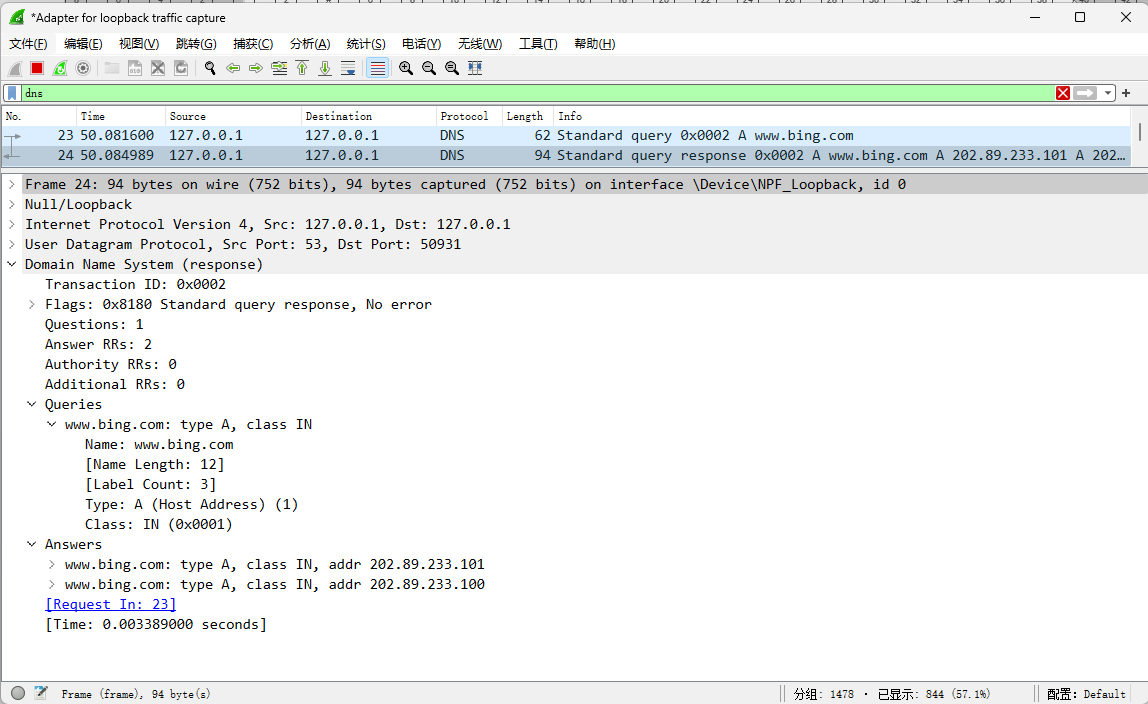


图 2.5

1. 等待2分钟，再次使用命令nslookup [www.bing.com](http://www.bing.com)

此时Cache中[www.bing.com](http://www.bing.com)的IP缓存已经过期，应向上游DNS服务器转发DNS请求，wireshark抓取的报文与1)相同。

1. 查询不存在或恶意的域名（008.cn）：
   * DNS中继服务器接收到查询请求；
   * 解析查询请求，并从缓存中查找相关信息；
   * 如果缓存中存在相关信息，并且IP地址为0.0.0.0，则返回不存在的响应结果给DNS客户端。

使用命令nslookup 008.cn

由于其在Cache中存储的IP地址为0.0.0.0，即表示其为不良网站，直接返回不存在的响应结果给DNS客户端。

图2.6显示了DNS中继服务器直接生成的向DNS客户端发送的响应报文，Reply code为No such name(3)。

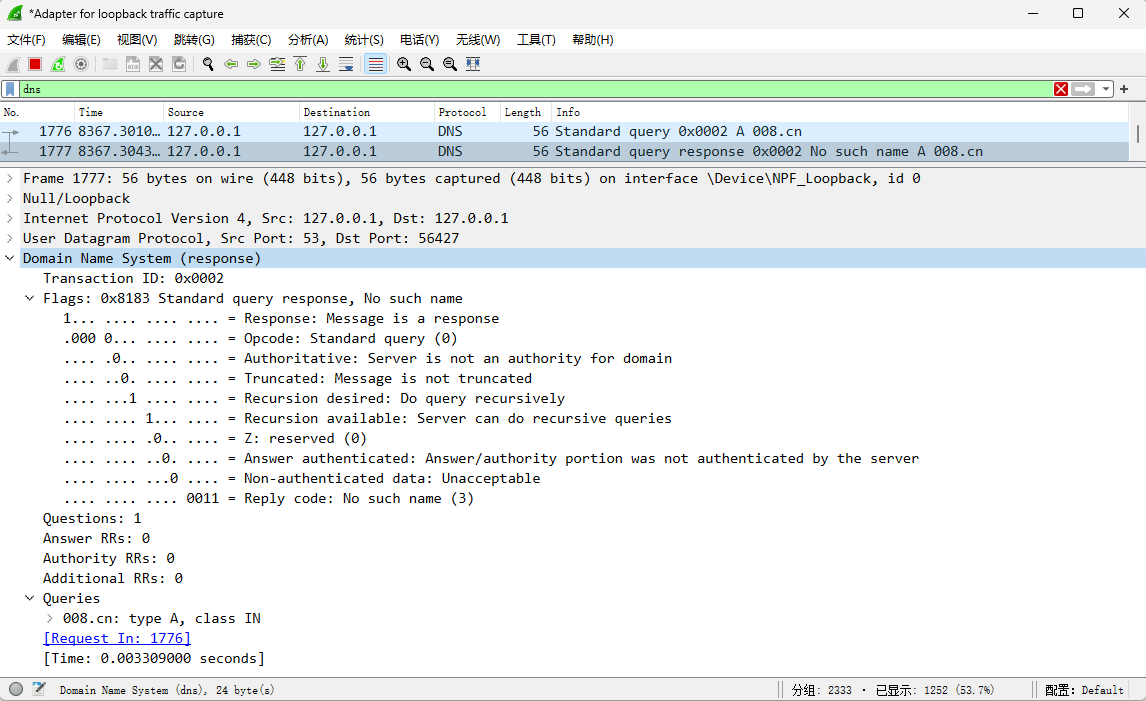


图2.6

以上测试用例主要是针对DNS中继服务器的基本功能进行测试，包括查询请求、缓存、转发和响应等方面。

# 5.调试中遇到并解决的问题

在设计和实现DNS中继服务器的过程中，遇到以下问题：

* 缓存管理不当：如果DNS中继服务器的缓存管理不当，会导致查询结果的不一致或不正确。需要在实现缓存管理模块时，设计合理的缓存更新策略和缓存清除策略，以保证查询结果的准确性和一致性。举例来说，对于不良域名，将一直存储在Cache中，设置一个长的生产期；对于一般域名，设置一个短的生存期，进行定时更新。
* 网络传输异常：DNS中继服务器需要通过网络接口与DNS客户端和上游DNS服务器进行通信，如果网络传输异常，会导致查询请求和响应消息的丢失或延迟。错误的DNS报文被递交给工作线程处理将导致严重的错误，对recvfrom的返回值以及GetLastError中的值进行检查，保证程序面对各种错误都有很好的健壮性。
* 安全漏洞问题：DNS中继服务器需要支持DNS消息的过滤和转换，以提高网络安全性和可靠性。如果过滤和转换不当，会导致安全漏洞问题。需要在实现DNS消息过滤模块时，设计合理的过滤规则和转换规则，以提高网络安全性和可靠性。我们通过使用ArrayList来实现ID转换表，通过ID转换提供了一定的安全性；对于不良域名，将其存储在Cache中，以特殊的标记标识(IP = 0.0.0.0)，对每个被标记的域名进行拦截。

# 6.心得体会

设计和实现DNS中继服务器是一项复杂的网络应用开发任务，需要掌握网络协议、分布式系统、并发编程等多方面知识。在开发过程中，需要深入理解DNS协议的工作原理和实现机制，同时考虑到网络传输的可靠性、安全性和性能等方面因素。

在实现DNS中继服务器时，需要注意以下几点：

* 设计合理的模块划分和接口设计：DNS中继服务器是一个复杂的系统，需要将其拆分为多个模块，每个模块负责不同的功能。在模块划分时，需要考虑到模块之间的接口设计，以保证模块之间的协作和数据交互的有效性和可靠性。
* 保证数据一致性和可靠性：DNS中继服务器需要管理缓存，对查询请求和响应消息进行处理，需要保证数据的一致性和可靠性。在实现时，需要考虑到数据的同步和更新机制，以保证数据的准确性和一致性。
* 考虑到性能和安全性：DNS中继服务器需要处理大量的查询请求和响应消息，同时需要保证网络传输的安全性和可靠性。在实现时，需要考虑到性能和安全性的平衡，采用合适的技术手段提高系统的性能和安全性。

综上所述，设计和实现DNS中继服务器是一项具有挑战性的任务，需要综合考虑多方面因素。在实现时，需要不断调整和优化系统架构、模块设计和接口设计，以提高系统的性能和可靠性。同时，需要关注网络安全性，采用合适的安全技术手段保证系统的安全性。