**计算机网络课程设计：“DNS 中继服务器”的实现**

**实验报告**

**王延吉 梁鹏 蒋礼特**

**2021211785 2021211784 2021211899**

北京邮电大学 计算机学院

日期：2023 年 6 月 24 日

**1 概述**

**1.1任务目的**

设计一个 DNS 服务器程序，读入“IP 地址-域名”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：

• 检索到 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）

• 检索到普通 IP 地址，则向客户端返回该地址（服务器功能）

• 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 开发环境**

**Win10**

**Win11**

**Visual Studio 2022**

**1.3 成员分工**

**梁鹏**

**蒋礼特**

**王延吉**

### **2 设计要求分析**

### **基本功能**

①接受DNS请求与发送DNS请求和回应需要使用基于SOCKET的UDP通信。

②DNS服务器本地查询功能需要使用本地文件存储资源。

③对于请求的ID要进行重新分配以防重复。

④对域名查询不同的结果要进行不同的操作。

⑤考虑本机和网络不同的字节序，通信中对DNS报文进行适当地转换。

⑥收到上层DNS回应，发送回客户之外要将查询结果作为资源缓存下来以备再次请求。

⑦由于UDP的不可靠性，有可能造成等待上层DNS回应阻塞，设置超时重发。

⑧为避免长时间未处理的请求与缓存，设置有效时间并定时进行请求与缓存清理。

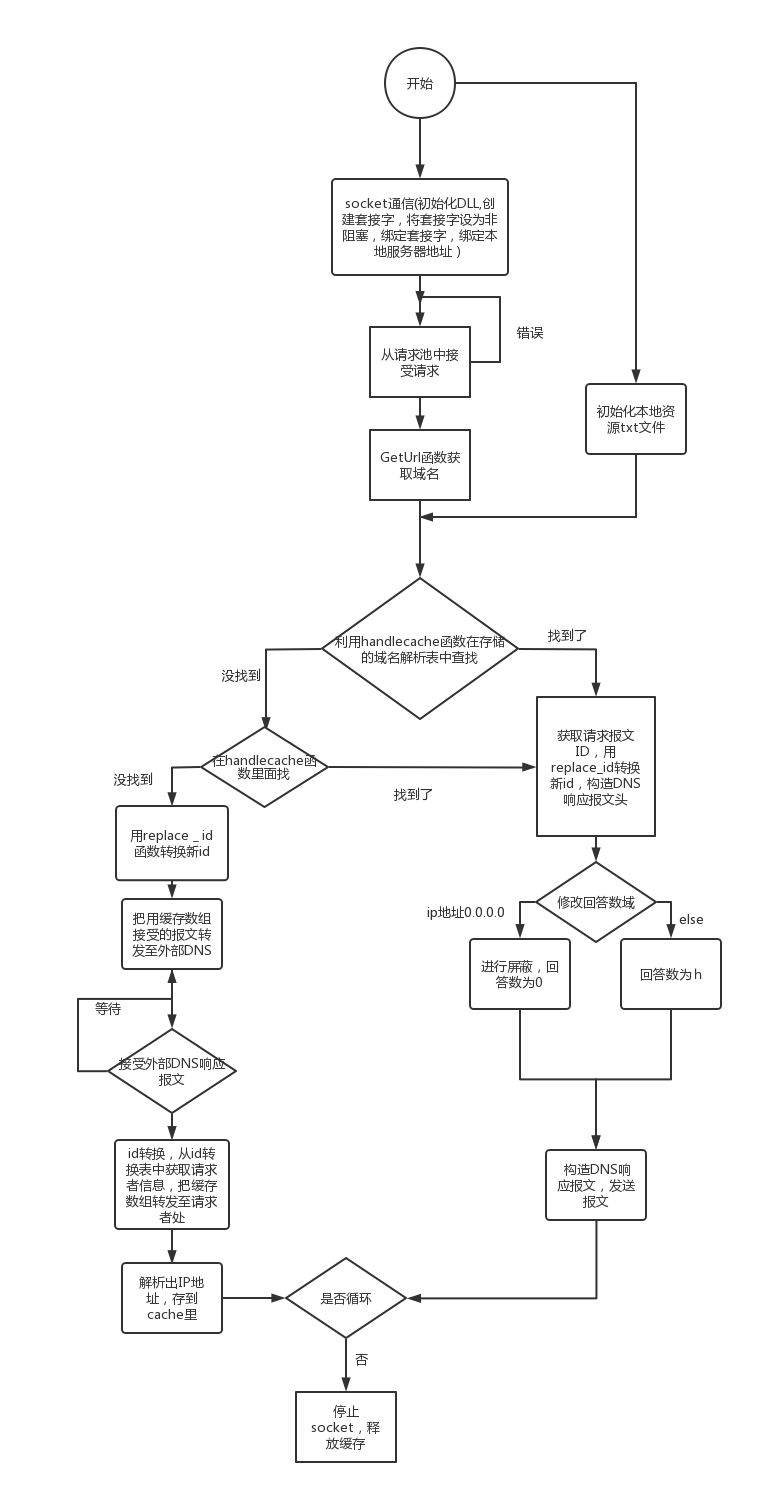
⑨将请求存储在请求池数据结构中，开设多个线程进行请求处理，防止长时间阻塞。

**3 模块划分**

**3.1总模块**

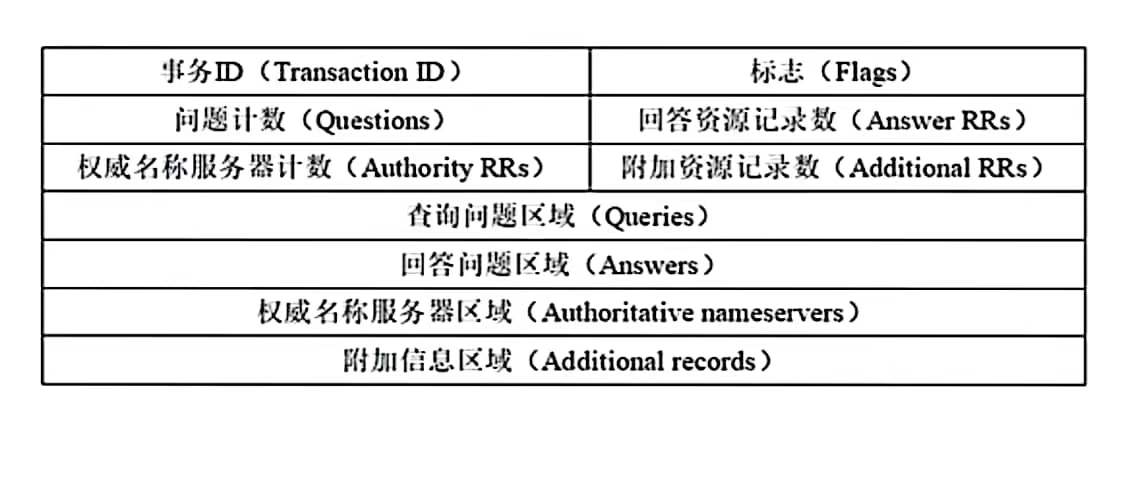
程序该包括以下模块，DNS报文解析，请求池，cache，socket，调试信息，获取域名，转换新id等。

**程序流程图及各模块关系如下图**

****

**3.2 DNS报文解析**

**3.2.1 DNS报文字段**

****

**程序中数据结构定义如下**

typedef struct DnsPacket

{

DNSHeader\* header;

QUESTION\* queries;//查询问题区域，动态分配的数组个数由qdCount决定

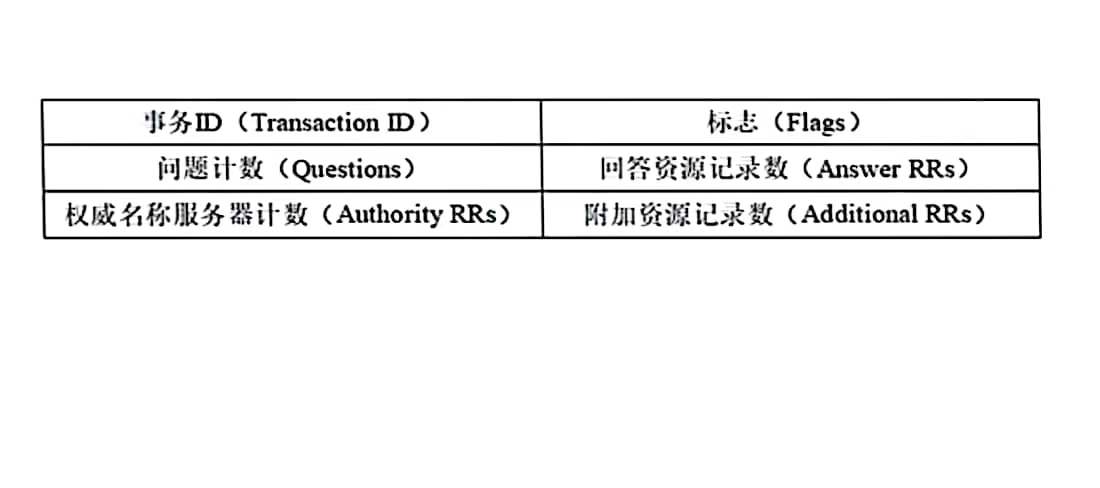
RR\* answers; //回答问题区域，动态分配的数组个数由anCount决定

RR\* authority; //权威名称服务器区域，动态分配的数组个数由nsCount决定

RR\* additional; //附加信息区域,动态分配的数组个数由arCount决定

}DNS\_Packet;

**3.2.2 Header部分格式**

****

**程序中数据结构定义如下**

typedef struct DNSheader

{

UINT16 id; //16位,该值由发出DNS请求的程序生成，DNS服务器在响应时会使用该ID，这样便于请求程序区分不同的DNS响应。

//位域排序类似于小段方式

unsigned char rd : 1; //1位，期望使用递归解析 (Recursion desired),如果该位为0，表示使用迭代查询方式

unsigned char tc : 1; //1位，被截断的报文(Truncated):当响应的总长度超512字节时，只返回前512个字节

unsigned char aa : 1; //1位，权威答案(Authoritative answer),表示响应的服务器是否是权威DNS服务器。只在响应消息中有效。

unsigned char opCode : 4; //4位，0:标准查询 ,指示请求的类型，有请求发起者设定，响应消息中复用该值。 1:反向查询 , 2:服务器状态请求

unsigned char qr : 1; //1位，0表示查询请求报文，1表示响应报文

unsigned char rCode : 4; //4位，响应码(Response coded)，仅用于响应报文

unsigned char z : 3; //3位，必须为0，保留字段

unsigned char ra : 1; //1位，递归可用(Recursion Available),该值在响应消息中被设置或被清除，以表明服务器是否支持递归查询。

UINT16 qdCount; //question section的问题个数，16位

UINT16 anCount; //answer section的RR(Resource Record)个数，16位

UINT16 nsCount; //authority records section的RR个数，16位

UINT16 arCount; //additional records section的RR个数，16位

}DNSHDR, \* pDNSHDR;

**事务ID：**用于由产生 DNS 查询的程序分配，用于标识一个请求；一对 DNS 查询和回复的 ID 相同。

**Flags：**基础结构部分中的标志字段又分为若干个字段，如下图所示

QR（Response）：查询请求/响应的标志信息。查询请求时，值为 0；响应时，值为 1。

**Opcode：**操作码。其中，0 表示标准查询；1 表示反向查询；2 表示服务器状态请求。

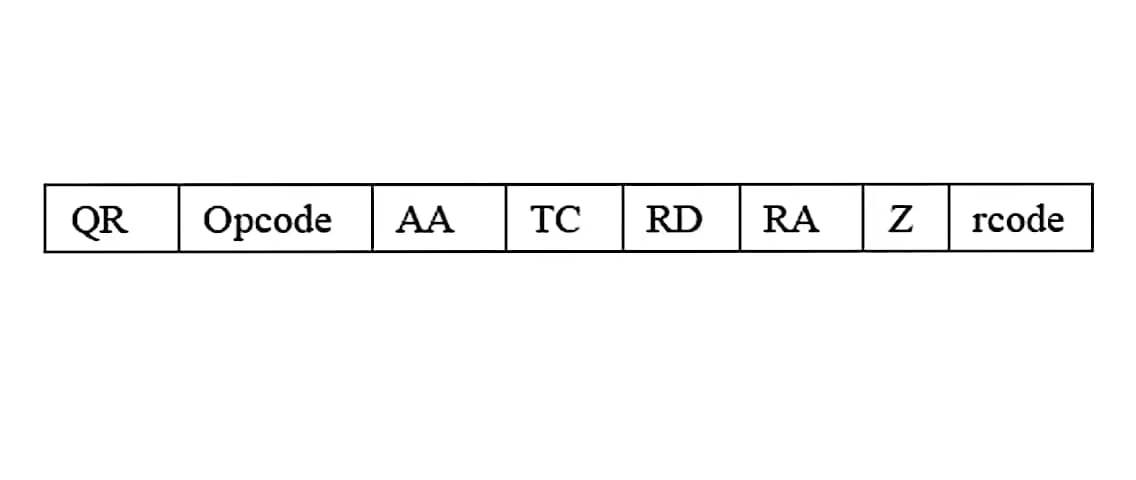
**AA（Authoritative）：**授权应答，该字段在响应报文中有效。值为 1 时，表示名称服务器是权威服务器；值为 0 时，表示不是权威服务器。

**TC（Truncated）：**表示是否被截断。值为 1 时，表示响应已超过 512 字节并已被截断，只返回前 512 个字节。

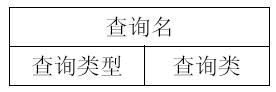
**RD（Recursion Desired）：**期望递归。该字段能在一个查询中设置，并在响应中返回。该标志告诉名称服务器必须处理这个查询，这种方式被称为一个递归查询。如果该位为 0，且被请求的名称服务器没有一个授权回答，它将返回一个能解答该查询的其他名称服务器列表。这种方式被称为迭代查询。

**RA（Recursion Available）：**可用递归。该字段只出现在响应报文中。当值为 1 时，表示服务器支持递归查询。

**Z：**保留字段，在所有的请求和应答报文中，它的值必须为 0。

**rcode（Reply code）：**返回码字段，表示响应的差错状态。当值为 0 时，表示没有错误；当值为 1 时，表示报文格式错误（Format error），服务器不能理解请求的报文；当值为 2 时，表示域名服务器失败（Server failure），因为服务器的原因导致没办法处理这个请求；当值为 3 时，表示名字错误（Name Error），只有对授权域名解析服务器有意义，指出解析的域名不存在；当值为 4 时，表示查询类型不支持（Not Implemented），即域名服务器不支持查询类型；当值为 5 时，表示拒绝（Refused），一般是服务器由于设置的策略拒绝给出应答，如服务器不希望对某些请求者给出应答。****

**3.2.3 Query部分格式**



数据结构定义如下

typedef struct

{

char\* qName; //字节数不定，以0x00作为结束符。表示查询的主机名。主机名被"."号分割成了多段标签。在QNAME中，每段标签前面加一个数字，表示接下来标签的长度。

UINT16 qType; //占2个字节。表示RR类型

UINT16 qClass; //占2个字节。表示RR分类

}QUESTION;

查询名：一般为要查询的域名，有时也会是 IP 地址，用于反向查询。

查询类型：DNS 查询请求的资源类型。通常查询类型为 A 类型，表示由域名获取对应的 IP 地址。

1. A，主机地址；

2，NS，权威域名服务器；

5，CNAME，域名引用；

6，SOA，授权机构起始；

12，PTR，域名指针，用于反向域名查找；

13，HINFO，主机信息；

14，MINFO，邮箱或邮件列表信息；

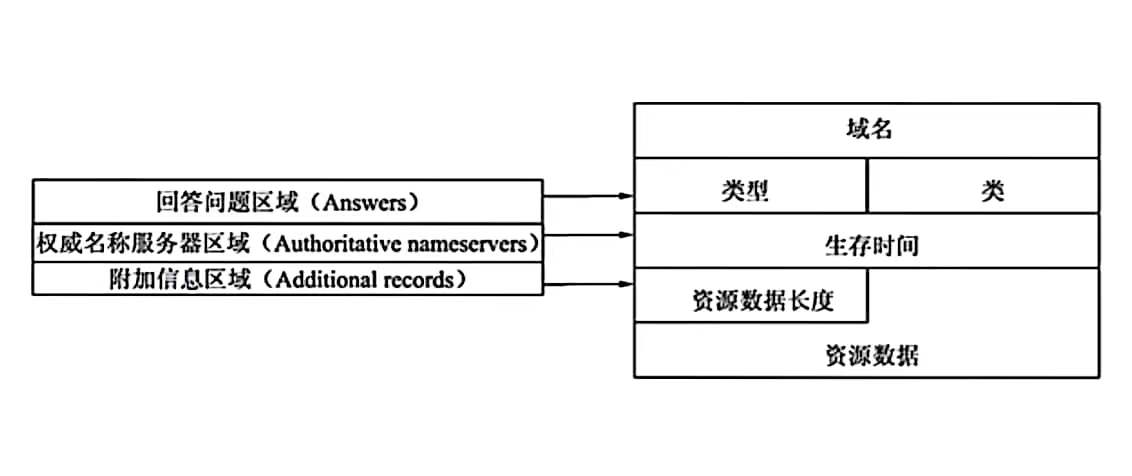
15，MX，邮件交换；

16，TXT，字符串。

28，AAAA，IPv6 地址。

查询类：地址类型，通常为互联网地址，值为 1。

**3.2.4 资源记录部分**

****

**数据结构定义如下**

typedef struct

{

char\* name; //域名

UINT16 type; //占2个字节。表示RR的类型，如A、CNAME、NS等

UINT16 rclass; //占2个字节。表示RR的分类

UINT32 ttl; //占4个字节。表示RR生命周期，即RR缓存时长，单位是秒

UINT16 rdLength;//占2个字节。指定RDATA字段的字节数

char\* rData; //资源数据

}RR;

**域名（name）：**DNS 请求的域名。

**类型(type)：**资源记录的类型，与问题部分中的查询类型值是一样的。

**类(class)：**地址类型，与问题部分中的查询类值是一样的。

**生存时间(TTL)：**以秒为单位，表示资源记录的生命周期，一般用于当地址解析程序取出资源记录后决定保存及使用缓存数据的时间。它同时也可以表明该资源记录的稳定程度，稳定的信息会被分配一个很大的值。

资源数据长度(RDLENGTH)：资源数据的长度。

资源数据(RDATA)：表示按查询段要求返回的相关资源记录的数据。

**RDATA 格式:**对于 DNS 中继服务器来说，RDATA 部分的内容并不重要，只是涉及到域名压缩， 以及调试输出的需求，所以需要对一部分类型的 RDATA 作处理。需要处理的 RDATA 如下。

**A 类型 RDATA 格式 :**ADDRESS，32 位 IPv4 地址。

**NS/CNAME 类型 RDATA 格式:** NAME，一个域名。

**MX 类型 RDATA 格式 :**

• PREFERENCE，16 位数字。

• EXCHANGE，一个域名。

**SOA 类型 RDATA 格式:**

• MNAME，一个域名。

• RNAME，一个域名。

• SERIAL，一个 32 位数字。

• REFRESH，一个 32 位数字。

• RETRY，一个 32 位数字。

• EXPIRE，一个 32 位数字。

• MINIMUM，一个 32 位数字。

**AAAA 类型 RDATA 格式:**ADDRESS，128 位 IPv6 地址。

**3.3 socket**

**3.4 cache**

**程序采用如下数据结构存储缓存**

map<char\*, cacheInfo> cache;

当程序访问完本地资源文件和cache后发现，没有找到对应IP地址，会从外部DNS响应报文的answer字段中解析出其IP地址，添加到cache中，这样在TTL时间之内，如果再查询该域名，会直接从cache获取IP地址，提高了程序效率。

具体解析原理如下：

**3.5 转换新id**

DNS服务器在发送和接收报文时，要转换其头部字段的ID，是为了区分不同的查询和响应。ID是一个16位的标识符，用于匹配查询和响应报文。如果不转换ID，那么DNS服务器可能无法正确地处理并发的查询和响应，导致混乱或错误。需要调取ReplaceNewID（）函数来重新获得新id，该函数通过维护ID转换表来对DNS报文字段的id部分进行转换。

**3.6 获取域名**

该程序多个部分需要实现这一功能，分别是从本地资源文件，缓存，外部DNS获取域名。

从本地txt获取域名是利用getUrl这一函数，这一函数通过数据结构map来存储，相较于顺序存储提高了查找效率

从缓存中获取域名也是采用map的数据结构来存储数据，相较于链表存储提高了查找效率

从外部DNS获取域名则是通过报文字段分析获取域名

**3.7 调试信息**

setParameter 函数，它接受两个参数 argc 和 argv[]，用于设置程序的参数。该函数的主要目的是解析命令行参数，并根据参数设置程序的配置选项。命令行格式为dnsrelay [-d | -dd] [dns-server-ipaddr] [filename]。命令行输入dnsrelay，参数默认为调试等级0，默认DNS服务器IP地址和默认文件路径。-d为调试等级一，-dd为调试等级二。[dns-server-ipaddr]为指定DNS服务器IP，[filename]为指定文件路径。

调试等级为全局变量，当配置好调试等级后，程序会输出不同的调试信息。等级0不输出，等级一首先会输出时间，然后输出ID序号，对应源地址和端口或者目的地址和端口，然后输出问题区域的域名。等级二会在等级一的基础上输出报文的具体字节，并且输出对报文头部的字段分析，假如是转发，则还会输出ID的转换。

**3.8 请求池**