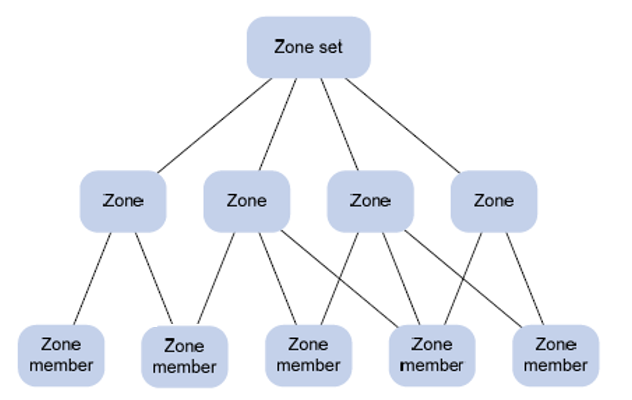
## 背景知识

## 用 zone 控制主机互访



* zone

zone是一组主机的集合。不在同一个zone内的主机不可以互相访问。

* zoneset

zoneset包含了一组zone，当zoneset active时，zoneset内的zone配置生效。

同一台设备上可以配置多个zoneset，但同时只能active一个zonset。

* 范例：

zone 1 //创建zone 1，进入zone 1的配置模式

add-member 10.0.0.1 //为zone 1 添加成员 10.0.0.1

add-member 20.0.0.1 //为zone 1 添加成员 20.0.0.1

add-member 30.0.0.1 //为zone 1 添加成员 30.0.0.1

zone 2 //创建zone 2，进入zone 2的配置模式

add-member 10.0.0.1 //为zone 2 添加成员 10.0.0.1

add-member 20.0.0.1 //为zone 2 添加成员 20.0.0.1

add-member 40.0.0.1 //为zone 2 添加成员 40.0.0.1

zone 3 //创建zone 3，进入zone 3的配置模式

add-member 30.0.0.1 //为zone 3 添加成员 30.0.0.1

add-member 40.0.0.1 //为zone 3 添加成员 40.0.0.1

zoneset 1 //创建zoneset 1，进入zoneset 1的配置模式

add-zone 1 //为zoneset 1 添加成员 zone 1

add-zone 2 //为zoneset 1 添加成员 zone 2

zoneset 2 //创建zoneset 2，进入zoneset 2的配置模式

add-zone 1 //为zoneset 2 添加成员 zone 1

add-zone 3 //为zoneset 2 添加成员 zone 3

active zoneset 1 //让zoneset 1 的配置生效。

在本例中：

交换设备配置了 zoneset 1 和 zoneset 2

其中 zoneset 1 被 active，因此设备使用zoneset 1添加的zone（zone1和zone2）来控制主机之间的互访。

如果交换设备收到3个报文：

input-frame 1

sip = 10.0.0.1

dip = 20.0.0.1

input-frame 2

sip = 10.0.0.1

dip = 40.0.0.1

input-frame 3

sip = 30.0.0.1

dip = 40.0.0.1

由于10.0.0.1 和 20.0.0.1 同属于zone 1，因此frame 1可以转发。

由于10.0.0.1 和 40.0.0.1 同属于zone 2，因此frame 2可以转发。

由于30.0.0.1 和 40.0.0.1 不属于同一个zone（既不是zone 1的共同成员，也不是zone 2的共同成员），因此frame 3不可以转发。

虽然 30.0.0.1和40.0.0.1都是zone3的成员，但是active的zoneset 1不包含zone 3，因此zone 3不生效。

## 出口是GRE隧道的IP转发表

* IP转发表

当交换设备收到一个IP报文后，需要根据报文的目的IP地址（下文简称DIP），查找同一个VRF内的IP转发表，选择一条最精确的转发表项，从转发表项的出口将报文发送出去。

**（出于简化考虑，本题目不考虑VRF。认为全局只有一个VRF。所有表项都没有VRF字段）**

* 出口是GRE隧道的IP转发表

IP转发表项的出口可能是一个GRE隧道，

例如：

add-route 20.0.0.0/24 tunnel 1 //添加一条IP转发表项，其中20.0.0.0是IP网段，24是掩码长度，tunnel 1表示出口的GRE隧道的名字。

* 转发过程举例：

输入一个报文：

input-frame 1

sip = 10.0.0.1

dip = 20.0.0.1

在设备中有2条IP转发表项：

add-route 20.0.0.1/32 tunnel 1

add-route 20.0.0.0/24 tunnel 2

设备根据输入报文的dip 20.0.0.1，查找IP转发表，发现精确匹配 add-route 20.0.0.1/32 tunnel 1，则将报文从GRE隧道tunnel 1 转发出去。

## GRE隧道

* 什么是GRE隧道

GRE隧道是一种虚拟接口，表示在三层IP网络中创建的一条VPN通道。GRE隧道的两头是2台交换设备。

* SIP和DIP

GRE隧道有两个关键属性：SIP和DIP。

SIP表示GRE隧道入口端交换设备的IP地址；

DIP表示GRE隧道入口端交换设备的IP地址；

每一个GRE隧道口的（sip + dip）的组合是唯一的，不允许2个隧道口配置相同的（sip + dip）组合。

例如：

int tunnel 1 //创建GRE隧道 tunnel 1，进入tunnel 1的配置模式

sip = 1.1.1.1 //配置tunnel 1的sip = 1.1.1.1

dip = 2.2.2.2 //配置tunnel 1的dip = 2.2.2.2

int tunnel 2

sip = 1.1.1.1

dip = 3.3.3.3

int tunnel 3

sip = 4.4.4.4

dip = 2.2.2.2

int tunnel 4 //创建GRE隧道 tunnel 4，进入tunnel 4的配置模式

sip = 4.4.4.4 //配置tunnel 4的sip = 4.4.4.4

dip = 2.2.2.2 //配置tunnel 4的dip = 2.2.2.2；但是这条配置失败，因为tunnel 4的（sip + dip）组合与tunnel 3冲突。这时tunnel 4只有sip 配置成功，dip配置失败（相当于没有配置dip）。

* 生效的GRE隧道

sip 和 dip 是隧道的必要属性。

只有当一个GRE隧道配置了sip和dip后，这个GRE隧道才才能用于IP转发。

上例中的tunnel 4就没有生效，不能用于转发。

## 根据GRE隧道拼接出IP转发表项的实际出口

* 为什么GRE隧道需要拼接实际出口

如1.2.所述，如果IP转发表项的出口是一个GRE隧道，交换设备需要将报文从GRE隧道转发出去。

但是GRE隧道是一个虚拟接口，报文转发时，需要找到一个实际的物理出口。

* 如何拼接实际出口

GRE隧道的DIP表示隧道出口端设备的ip，可以查找DIP对应的主机arp和mac，最终拼接得到（或通过拼接路由得到，本题只讨论arp）。

本题简化考虑，只需要通过DIP拼接ARP表项查找实际出口：

例如：

存在IP转发表项：

add-route 20.0.0.0/24 tunnel 1

存在GRE隧道tunnel 1：

int tunnel 1

sip = 1.1.1.1

dip = 2.2.2.2

存在IP 2.2.2.2的arp:

add-arp 2.2.2.2 0000.0000.0001 ten0/1

则通过拼接隧道tunnel 1 的 dip 2.2.2.2 和 arp表项2.2.2.2，得到实际生效的IP转发表项：

20.0.0.0/24 ten0/1 //实际出口是ten0/1

因此，在1.2中输入的那个报文：

input-frame 1

sip = 10.0.0.1

dip = 20.0.0.1

实际上是从ten0/1接口输出。

## 查找IP转发表项的注意点

查找IP转发表时，要注意：

1. **可能存在多条IP转发表项能匹配报文的dip，要匹配最精确的表项。**

例如：

存在两条IP转发表项：

20.0.0.0/24 tunnel 1， tunnel 1拼接arp出口是ten0/1

20.0.0.0/28 tunnel 2， tunnel 2拼接arp出口是ten0/2

则报文dip=20.0.0.1 匹配第二条IP转发表项，转发出口是ten0/2

1. **只匹配有实际出口的转发表项。**

例如：

存在两条IP转发表项：

20.0.0.0/24 tunnel 1， tunnel 1无效（隧道SIP或DIP配置失败）

20.0.0.0/28 tunnel 2， tunnel 2找不到隧道DIP对应的arp表项拼接。

则报文dip=20.0.0.1 找不到能匹配的IP转发表项，只能丢弃（DROP）。

1. **综合1）和2）的情况。**

例如：

存在两条IP转发表项：

20.0.0.0/24 tunnel 1， tunnel 1拼接arp出口是ten0/1

20.0.0.0/28 tunnel 2， tunnel 2找不到隧道DIP对应的arp表项拼接。

20.0.0.1/32 tunnel 3， tunnel 3无效（隧道SIP或DIP配置失败）

则报文dip=20.0.0.1 匹配第1条IP转发表项，转发出口是ten0/1.

虽然第3条和第2条IP转发表项更精确，但它们是无效表项，所以只能选择次精确的转发表项。

## 题目要求

## 概述

本题目要求编码实现基于zone控制的、出口是GRE隧道的转发模型。

* 要求设计数据结构保存zone、zoneset配置
* 要求设计数据结构保存IP转发表、ARP表项、GRE隧道，并拼接表项。
* 要求设计算法，查找拼接的表项，计算报文的转发行为。

## 输入和输出模型

1. 本题目会输入一串伪CLI配置，用来模拟zone和zoneset的配置。
2. 本题目会输入一串伪CLI配置，用来模拟GRE隧道（int tunnel）的配置（包括sip、dip）
3. 本题目会输入一串伪CLI配置，用来模拟ARP表项。
4. 本题目会输入一串伪CLI配置，用来模拟IP转发表项（出口是GRE隧道）。
5. 本题目会输入一串报文（包含报文ID、SIP、DIP），需要程序根据当前设备生效的配置拼接出转发表，算出每一个报文的转发出接口（实际物理口）。如果这个报文无法转发，则报文需要被DROP。

## 输入文件详解

* **每次执行程序的输入都包含在一个 .txt 后缀名命名的文档中，文档名通过参数传入。**

例如：输入文档名为input1.txt

./zone.elf input1.txt

**input1.txt 中包含以下配置信息：**

* **配置zone：**

配置zone相关的命令有以下这些：

* **zone 1**

//创建zone 1，并进入zone 1的配置模式； （参数1表示zone的ID，一定是一个数字）

* **add-member 30.0.0.1**

//为zone 1添加成员 30.0.0.1；（这条命令一定是在某个zone的配置模式下配置； 参数：添加的成员是一个IP地址；一个IP地址可以加入多个zone）

* **del-member 30.0.0.1**

//为zone 1 删除成员 30.0.0.1；（这条命令一定是在某个zone的配置模式下配置； 参数：删除的成员是一个IP地址）

* **no zone 1**

//删除zone 1

* **zoneset 1**

//创建zoneset 1，进入zoneset 1的配置模式；（参数1表示zoneset的ID，一定是一个数字）

* **add-zone 1**

//为zoneset 1 添加成员 zone 1；（这条命令一定是在某个zoneset的配置模式下配置； 参数1表示添加的成员zone的 ID； 只有已经创建的zone才能加入zoneset，否则配置失败）

* **del-zone 1**

//为zoneset 1 删除成员 zone 1；（这条命令一定是在某个zoneset的配置模式下配置； 参数1表示删除的成员zone的 ID）

* **no zoneset 1**

//删除zoneset 1， 注意删除zoneset不会删除zone。

* **active zoneset 1**

//让zoneset 1 的配置生效。 只有已经创建的zoneset才能active，否则配置失败。

如果多次配置active zoneset xx，后面配置覆盖早先的配置。

* **配置GRE隧道：**
* **int tunnel 1**

//创建GRE隧道 tunnel 1，进入tunnel 1的配置模式（参数1表示隧道的ID，一定是一个数字）

* **sip = 1.1.1.1**

//配置tunnel 1的sip = 1.1.1.1 （这条命令一定是在某个tunnel的配置模式下配置；）。如果重复进入同一个GRE隧道的配置模式，重复配置sip，则后面的配置会覆盖早先的配置。

* **dip = 2.2.2.2**

//配置tunnel 1的dip = 2.2.2.2 （这条命令一定是在某个tunnel的配置模式下配置；如果配置冲突，则配置失败，相当于没有这条配置。）如果重复进入同一个GRE隧道的配置模式，重复配置dip，则后面的配置会覆盖早先的配置。

* **no int tunnel 1**

//删除GRE隧道 tunnel 1

* **配置IP转发表：**
* **add-route 20.0.0.1/32 tunnel 1**

//创建一条静态路由，20.0.0.1是网段， 32是子网掩码， tunnel 1是出口对应的GRE隧道。 其中 网段/子网掩码 是表项的key，如果重复输入key相同的路由，表示更新出口信息。

* **del-route 20.0.0.1/32**

//删除一条静态路由，参数是表项的key 值：20.0.0.1是网段， 32是子网掩码。

* **配置ARP：**
* **add-arp 2.2.2.2 0000.0000.0001 ten0/1**

//创建一条arp，2.2.2.2是IP地址， 0000.0000.0001是MAC地址，ten0/1是物理出口（用字符串表示）。 其中IP地址是arp的key。 重复输入key相同的arp，表示更新arp的属性（mac地址和物理出口）信息。

* **del-arp 2.2.2.2**

//删除一条arp，参数是arp的key 值：2.2.2.2是IP地址

* **输入的报文：**
* **input-frame 1**

**sip = 10.0.0.1**

**dip = 20.0.0.1**

//输入的报文包括3行：

//第一行表示输入报文1， 1是报文的序号，一定是一个数字。

//第二行表示报文的sip

//第三行表示报文的dip

## 输出文件

* **每个输入文件对应一个输出文件；输出文件名= 输入文件名\_result**

例如：输入文档名为input1.txt

那么执行完 ./zone.elf input1.txt 后，应该生成一个**input1\_result.txt** 的输出文档

**input1\_ result.txt 中需要包含input1.txt中每一个报文的转发结果，按照报文序号排列，每个报文一行：**

* **输出报文转发行为，一共有2种：**
* **frame 1 DROP**

//frame 1的转发行为是丢弃（丢弃原因有几种： sip、dip不在同一个zone内，或者查不到IP转发表，或者IP转发表的出口GRE隧道不生效，或者IP转发表的出口GRE隧道没有对应的arp拼接找不到实际物理口）

* **frame 2 ten0/1**

//frame 2的转发出口（物理口）是ten0/1

## 考核验收方法

**1）代码提交要求：**

考生的所有源码放在一个文件夹内。

必须有makefile文件，生成可执行文件名称为 zone.elf

**2）功能检查：**

功能检查完全采用自动化程序，不允许人工修改、干预，一视同仁。所以请大家避免由于理解错误题目或者输出格式错误、读取输入文件错误等导致扣分。

自动检查程序会在考生文件夹下执行如下操作：

make

./zone.elf input1.txt

./zone.elf input2.txt

./zone.elf input3.txt

./zone.elf input4.txt

./zone.elf input5.txt

而后自动检查程序会读取以下5个输出文件，并判断答案正确性：

input1\_result.txt

input2\_result.txt

input3\_result.txt

input4\_result.txt

input5\_result.txt

**3）难度说明：**

5次输入的难度、复杂度递增.

**4）算法和代码质量检查：**

人工检查，评审代码。考察范围包括但不限于如下几点：

* 数据结构设计对性能和内存的考虑
* 查找算法对性能的考虑
* 低级bug
* 预防性编码
* 代码复用性
* 编码规范

……

**5）范例输入和输出：**

参见

example1.txt 和 example1\_result.txt

example2.txt 和 example2\_result.txt