

# Router-Lab实验报告

计73 林俊峰

2017011303

## 1. 实现过程

### 1.1 第一部分

`checksum` : 照着算法实现即可。

`forwarding` : 这里要对新的checksum进行特判, 若为0xffff则置0。这个实现是最简单的, 使用增量计算的方法也尝试过, 但是结果不正确, 应该是哪里理解错了。

`lookup` : 使用了线性查表。

`protocol` : 在判断字段是否满足约束条件时要注意端序转换。

### 1.2 第二部分

这里要处理的请求有两个, 分别是 `request` 和 `response`, 对不同请求的处理如下:

- request
  - 遍历本地的路由表, 构造出一个 `RipPacket` 结构体
  - 调用 `assemble` 函数, 另外再把 IP 和 UDP 头补充在前面
  - 通过 `HAL_SendIPPacket` 发回询问的网口
- response
  - 调用 `query` 和 `update` 函数进行查询和更新
  - 给 `RoutingTableEntry` 添加新的字段 (`metric`)
  - 如果有路由更新的情况, 需要构造出 `RipPacket` 结构体, 调用 `assemble` 函数
  - 再把 IP 和 UDP 头补充在前面, 通过 `HAL_SendIPPacket` 把它发到别的网口上

### 1.3 第三部分

组队将路由器连起来之后直接跑测试就可以了, 实现了三人成环的拓扑

## 2 实验问题以及解决思路

### 2.1. 端序问题

尽管已经考虑到了各种大小端序列的转换，但是在对ip地址做最长前缀匹配的时候还是搞错了。通过网络号长度获取mask的二进制表示，一开始用的是

```
uint32_t mask = ~(0xFFFFFFFF<< len)
```

这种表示方法在网络号长度为8的整数时没有问题，因此一开始没有发现。在实验第三阶段做压力测试时掩码长度出现了非8的倍数，发生错误。原因在于，大段序和小段序只是字节的排列顺序颠倒，而每个字节的排列是不变的，比如对于 `0xFFFE`，二进制表示为 `11111111 11111110`，端序变化后变成 `11111110 11111111` 而非 `01111111 11111111`，因而这种计算mask的方式是错误的，改用如下公式计算即可

```
uint32_t mask = htonl(~(1 << (32-len)));
```

## 2.2.Bird配置问题

起初我使用的Bird版本是2.0，在程序运行过程当中出现了开启Bird的路由器的路由表中存在直连路由但是路由器发送的 `arp` 包中不含其直连路由的问题，导致其它的路由器一直无法学习到该条路由。多人帮助调试无果，转而将Bird降级成1.6，问题解决。