

Build Your Own Router实验报告

桂阿臻 2017013597

程序介绍

```
RoutingTableEntry RoutingTable::lookup ( uint32_t ip ) const ;
```

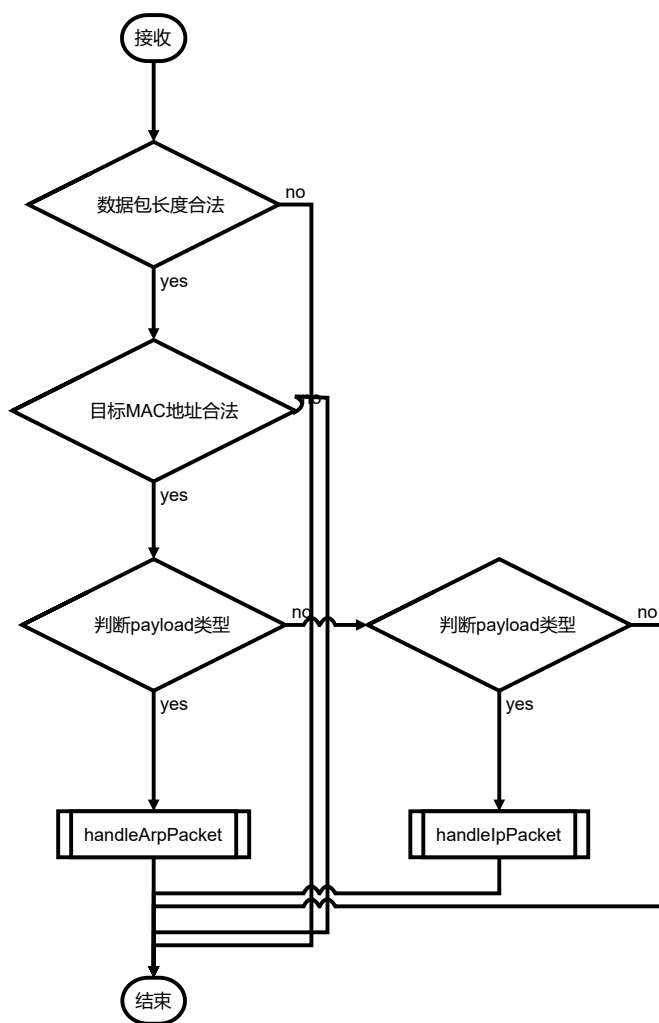
根据最长前缀匹配原则查找ip对应的路由表项

已经有匹配结果时，找到了子网掩码4字节对应的uint32_t值更大时，取新的匹配项

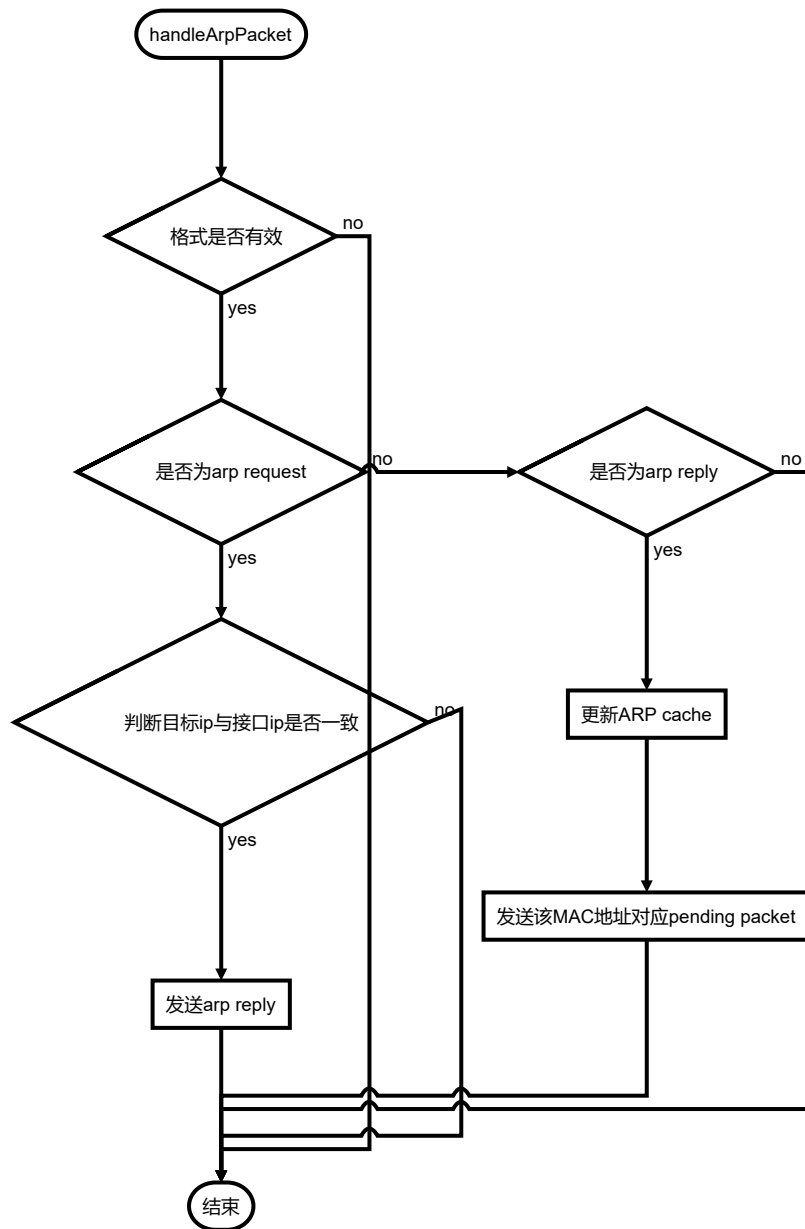
```
void SimpleRouter::handlePacket ( const Buffer& packet , const std::string& iface ) ;
```

处理接收到的每一个以太网帧

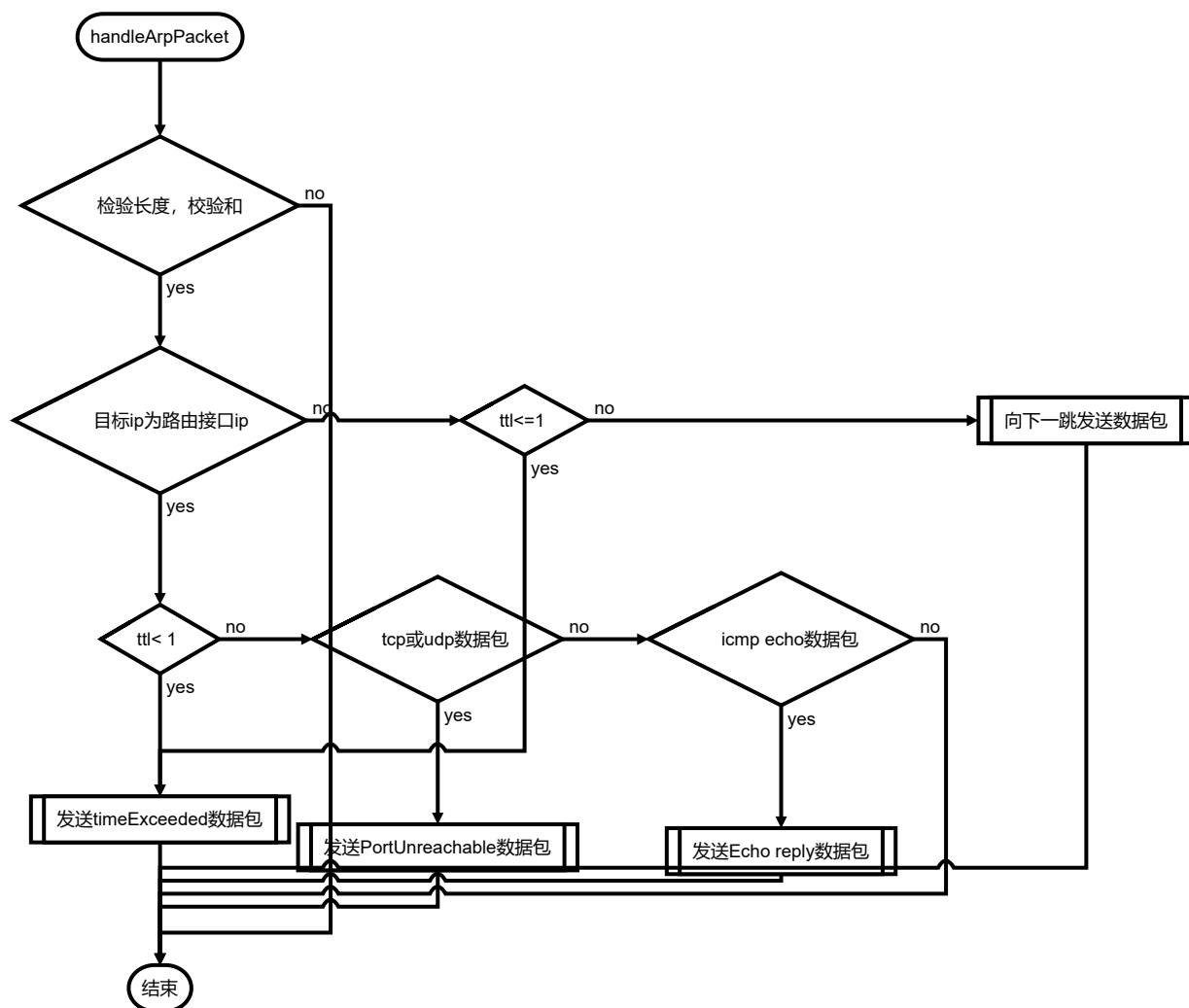
逻辑如下



handleArpPacket逻辑如下



handleIpPacket逻辑如下



```
void ArpCache::periodicCheckArpRequestsAndCacheEntries();
```

每秒更新依次ARP cache

对于每一个ip对应的ArpRequest, 若发送不超过5次, 则再发送一次, 若发送超过5次, 对于保存在路由中对应ip的pending packets, 向每个发送方返回一个portUnreachable数据包

对于每一个ip对应的ArpEntry, 持续存在30s后将被清除

实验结果

经测试, 作业要求中的grading guidelines中的pint tests, traceroute tests, filedownloading tests均能通过

限于报告篇幅, 此处不展示实验结果

遇到的问题

在periodicCheckArpRequestsAndCacheEntries中, 由于上层函数ticker()中锁的存在, 不能调用ArpCache的部分成员函数更新ARP cache

最终解决方式是直接将需要使用的成员函数的相关代码改写后在periodicCheckArpRequestsAndCacheEntries中使用

作业建议

希望在给未来同学布置该作业时能将文档进一步完善, 说明各文件的功能, 如本次作业代码包中提供的 show-arp.py 可以帮助调试判断ARP cache, 但在作业说明文档的pdf中似乎并没有说明 (只在布置作业的ppt中有不明显的提示), 完善的说明文档能帮助同学们更快地上手作业