Router 实验报告

黄泽文 2022013014 未央-软件21

1 简介

本项目实现了一个简单的路由程序中的数据包处理、arp 缓存处理和路由表读取功能。

2 代码结构

simple-router 中包含主要的数据包处理函数，即 `handlePacket` 与发送以下类型数据包的函数：

- `arp request` 与 `arp_reply`（询问路由器接口与非路由器接口）；
- `icmp_reply`，`icmp_ttl` 与 `icmp_unreachable`；
- 常规 `ipv4` 包的转发。

routing-table 中包含在路由表中搜寻 ip 对应接口的函数，只需要简单遍历路由表并记录最长匹配即可。

arp-cache 中包含一个定时触发的检查函数，重发队列中的请求并检查缓存的有效时间。

3 遇到的问题和解决

本项目的难度主要在于初上手阶段和基本实现代码后的调试，故遇到的问题除了理解项目的任务之外，也基本都是对一些开发细节的 debug。

转发时找不到 MAC 地址缓存的处理

当需要转发一个数据，但目标 IP 的 MAC 地址还没有被缓存时，应当先广播一个 ARP 请求并将要发送的包存入队列，待得到 MAC 地址回复后再发送。我起先没有注意到 arp-cache 文件中已经包括了等待的队列，故自己实现了处理，后改回了使用现有的代码。难点主要在于考虑 arp-cache 中定时函数和 handlePacket 函数的任务区分。我的实现思路是在 arp-cahce 中只调用 simple-router 中对应的函数以发送 arp-request，待 handlePacket 捕获特定的 reply 时，直接读取队列并发送等待的包。

IP 与 ICMP 头中的可选段的处理

在 ipv4 的报头末尾包含 options 和 padding 段，在 ICMP 的超时和不可达报头中包含一段 unused 部分。我在开发中尝试添加和略去了这些部分，发现略去 ipv4 的 options 和 padding 段（即只有 20 字节的报头）是可以识别的，但 ICMP 中的保留段应该正常置 0。

主机字节序与网络字节序的选择

这里主要是应当约定好储存的方式。对于一些需要转换的值，既可以转换作中间变量，也可以保留原状。开发时比较容易搞乱，在日志中多输出检查即可发现问题。

一些特定量的维护

这些问题在 debug 中基本都可以发现。报头中需要维护的量比较多且杂，有的问题比较隐蔽，例如 ip 头中的长度计算错误可能导致 ping 能通但返回的长度不对而提示 truncated。

4 总结

在本次项目中，我实际开发的时间并不长，但是阅读并理解作业文档花费了我大量时间。开发的重点在于理解 handlePacket 可能面对的情况分类，以及在各个情况下应该做出何种处理。文档中包含的内容量很大，但大多数只需在开发中供查阅用，我也是在完成后才意识到从 Helpful Hints 一节所给的思路出发是比较简单的。调试函数和流量监控的作用很大，有助于发现开发细节中的问题。我的代码中没有额外引入第三方库。