Build Your Router 说明文档

软件 92 诸葛向文 2018010115

2021 年 11 月 26 日

1 代码逻辑

1.1 SimpleRouter::handlePacket 函数

在文档的 Hint 部分，给出了代码的部分逻辑，下面给出更细化的版本：

(1) 检测端口（代码已经给出）

(2) 对 frame 的头部进行检测

(a) 检测数据包的长度是否符合要求，不符合要求不处理

(b) 检测数据包的类型是否为 ARP 或 IPV4，不符合要求不处理

(c) 检测数据包的目的地是否为本 router，不是不进行处理

(3) 根据数据报的类型进行分类：ARP 或 IPV4

ARP 类型：

(a) 对 ARP 的头部进行检测，大小不正确不处理

(b) 根据分类分别处理：arp-request 以及 arp-reply

i. arp-request, 判断是否请求的本机端口，如是回复，否则忽略。

ii. arp-reply，检测 arp-cache 中是否已经存在 IP-MAC 映射。如存在，忽略；不存在，在 cache 中加入，然后发送等待该 MAC 地址的包。

IPV4 类型：

(a) 对 IPV4 数据报的头部进行检验，判断长度和 checksum 是否合适，忽略错误的。

(b) 判断该数据报是否目的地为本 router

如是

i. 传输层数据为 ICMP 类型，发送 ICMP echo reply

ii. TCP/UDP 负载，发送 ICMP port unreachable

iii. 忽略其他类型

如否：数据包继续向前传递

i. 检测 TTL，TTL=0，发送 ICMP time Exceed

ii. 继续向前传递。如果此时 ARP-CACHE 中没有相应的 MAC 地址，则加入等待的队列，否则直接发送。

1.2 ArpCache::periodicCheckArpRequestsAndCacheEntries 函数

代码的注释中已经给出了 periodicCheckArpRequestsAndCacheEntries 的函数框架，并不需要我们自己动脑，根据需求填完代码框架即可。

```
for each request in queued requests:
    if now - req->timeSent > seconds(1)
        if req->nTimesSent >= 5:
            send icmp host unreachable to source addr of all pkts waiting
              on this request
            cache.removeRequest(req)
        else:
            send arp request
            req->timeSent = now
            req->nTimesSent++

for each cache entry in entries:
    if not entry->isValid
        record entry for removal
remove all entries marked for removal
```

1.3 RoutingTable::lookup 函数

即为路由表的最长匹配算法，比较简单，不做赘述。

2 存在的问题与解决

2.1 ping 时前两个包总是丢失

这个问题看着有点像意外，起先我也以为是某种不可控的因素导致的。但是反复实验之后发现每次 ping 一个新的地址时，总是前两个包会丢失，ping 之前刚 ping 过的位置，则所有包都能正确地被我的路由器转发。由此确定了问题必然出现在处理 arpReply 消息上，在得到目的地 ip-mac 地址之前的两个消息被我忽略了。

经过仔细的 debug，我发现在处理指针时有一个逻辑写反了。在获取到 insertArpEntry 返回的指针之后，应该当指针不为空时，将积压的消息处理。但是我手抖多打了一个!。得到的经验就是虽然写 if(ptr) 这样的程序可以少打几个字，但是还是写成 if(ptr != nullptr) 这样更加清晰和直观，debug 的时候也方便些。

2.2 对 icmp ipv4 协议理解不正确

由于对文档的阅读的疏忽，想当然的认为 icmp echo reply, icmp hostunreachable, icmp port unreachable，icmp time exceed 几个包的长度都应该一样长，自认为聪明地将其封装进一个函数，但实际上 icmp echo reply 的长度与其余几个并不相同。

另外一个困扰我比较久的地方是路由器发送 icmp 包时 ipv4 数据包时，包头 id 和 off 字段的确定。事实上 RFC791 和课本对 id 字段都有些含糊其词，RFC791 认为 id 字段应当是辅助数据报拼接，返回给同

一主机不同数据报的 id 不同，相同数据报 id 一致，而课本上则说主机每发一个数据报则 id+1。最后折中考虑了两者，直接采取了发送过来的 ip 数据报的 id 值，相当于借用发送主机的 id 来保证 id 的正确。off 字段同理。

3 额外的库

没有额外使用的库。

4 建议和感受

感觉这个作业的质量比 FTP 高了不知道几档。一方面，这个作业的工作量明显小一点；另一方面这个作业让我对路由器的工作机制，网络的链路层、网络层、传输层三层的协调工作有了更深刻的理解，而我觉得 ftp 实验和计算机网络这门课程关系甚微。希望之后几届能够换掉 ftp 这个作业。我看 cs118 课程还有 2 个 project，不妨考虑借鉴下那个。