

Router Report

李航 软13 2021011833

实现逻辑

- 收到`Packet`
 - 判断`mac`是否存在, 或是否为`broadcast_host`, 若均不满足则输出错误信息并返回。
 - 若存在, 则获得`ethernet_hdr`, 读取其`ether_type`, 区分为`Arp`和`IPv4`, 分别处理。

```
if(ethernetHdr->ether_type == ntohs(ethertype_ip)) {
    std::cout << "IPv4" << std::endl;
    handleIPv4Packet(packet, inIface, ethernetHdr);
}

if(ethernetHdr->ether_type == ntohs(ethertype_arp)) {
    std::cout << "ARP" << std::endl;
    handleARPPacket(packet, inIface, ethernetHdr);
}
```

- 处理`Arp`
 - 处理`Arp`请求:
 - 获取`arp_hdr->tip`是否与路由器接口地址相同, 不相同则返回错误。相同则进行处理, 构造返回`Packet`, 将`mac`填充到`Packet`中进行回复。
 - 处理`Arp`回复:
 - 更新映射表, 查看其是否存在未发送的请求, 若存在则依次发送`Packet`

```
const Interface* iface = findIfaceByName(inIface);
// request
if(arpHdr->arp_op == ntohs(arp_op_request)){
    ...
}
else if(arpHdr->arp_op == ntohs(arp_op_reply)){
    ...
}
```

- 处理`IPv4`
 - 首先计算`IPv4 check sum`, 如果不匹配则输出错误信息并返回。
 - 查询并检查`target_interface`是否存在

- 若存在，则判断`ipHdr->ip_p`，若为`ip_protocol_icmp`且为`echo request`，则构造`echo reply`进行返回。若`ipHdr->ip_p`为不可到达的`tcp udp`，则构造不可到达的`icmp t3`数据包进行返回。
- 若不存在，则更新其`ttl`。
 - 若`ttl`次数大于上限5次，则构造超时的`icmp t3`数据包进行返回。
 - 若未超时，尝试从`m_routingTable`中获得`route_entry`，若报错则直接输出错误信息并返回。成功则检查`ip`与`mac`映射表中是否存在有效的映射，不存在则放置于`Arp`的请求队列中，若存在则修改`mac`，发送数据包。

```
// check destination
const Interface *target_face = findInterfaceByIp(ipHdr->ip_dst);
if(target_face == nullptr){
    ...
}
else{
    ...
}
```

• 路由表最长匹配

- 对输入的`ip`，遍历路由表进行网络地址的匹配，匹配过程中记录最长匹配对应的条目并最终返回结果。

```
for (const auto& entry : m_entries) {
    uint32_t dst = ntohl(entry.dest);
    uint32_t mask = ntohl(entry.mask);
    // Apply the mask to the IP address to get the network part
    uint32_t networkPart = ip & mask;

    if(networkPart == (dst & mask)){
        // Calculate the prefix length by counting the number of
        // leading 1s in the mask
        int prefixLength = 0;
        while (mask & (1u << 31)) {
            prefixLength++;
            mask <<= 1;
        }
        // Update the longest match if the current prefix length is
        // longer
        if (prefixLength > longestPrefixLength) {
            longestPrefixLength = prefixLength;
            target = entry;
        }
    }
}
```

```
}  
}
```

- 周期更新 *Arp Cache*
 - 遍历缓存条目，对于不合法的项目直接删除
 - 遍历 *Arp* 请求队列，判断请求发送的次数，若大于五次则构造超时 *icmp t3* 数据包并发送，从队列中删除该请求。未大于五次则继续构造发送 *Arp* 数据包。

```
for(auto it = m_arpRequests.begin(); it != m_arpRequests.end(); ){  
    auto request = *it;  
    if(request->nTimesSent >= 5){  
        ...  
    }  
    else{  
        ...  
    }  
}
```

程序测试结果

```
*** Conducting File Downloading Tests.  
  
*** Shutting down stale SimpleHTTPServ  
*** Shutting down stale webservers  
*** Stopping 1 controllers  
c0  
*** Stopping 3 links  
...  
*** Stopping 1 switches  
sw0  
*** Stopping 3 hosts  
client server1 server2  
*** Done  
Your tests score is 45/45 pts  
THIS IS NOT YOUR FINAL SCORE !!!
```