Router report

2021011807 张泽卿 软件学院

实验要求：

本次实验要求实现一个具有静态路由表的简单路由器，该路由器能够接收原始以太网帧并正确处理，例如转发到适当的接口或创建新的帧。

实验环境：

Ubuntu 16.04.7 系统上虚拟机，使用 Mininet 模拟网络拓扑。

函数功能：

Simple-router.cpp：

为了实现处理网络流量，响应 ARP 请求，以及根据 IP 和 ICMP 规则正确转发或回应数据包，构造了以下函数：

handlePacket：
参数：原始数据包，输入接口名称。
功能：检查接口有效性和数据包大小。解析以太网头部，并根据包的类型（ARP 或 IP）调用对应的处理函数。

handle_arp_packet：
参数：ARP 数据，输入接口，源 MAC 地址。
功能：处理 ARP 请求和回复。对 ARP 请求，生成并发送 ARP 回复；对 ARP 回复，更新 ARP 缓存。

handle_ip_packet：
参数：IP 数据包，输入接口，源 MAC 地址。
功能：执行 IP 包头检查，更新 ARP 缓存。根据目的地址决定回应 ICMP 请求或转发数据包。对目的地址是路由器接口的包，发送 ICMP 不可达消息。

send_icmp_packet：
参数：原始数据包，输入接口，ICMP 类型，ICMP 代码。
功能：创建新的 ICMP 数据包（如时间超时或端口不可达消息），并发送。

judgeMac：
参数：MAC 地址，输入接口。
功能：判断 MAC 地址是否是广播地址或特定接口地址。

Arp-cache.cpp:

为了定期检查和维护 ARP 请求列表和缓存条目来确保网络通信的连贯性和有效性，完善了 periodicCheckArpRequestsAndCacheEntries 函数，先初始化广播地址，创建一个表示广播 MAC 地址的数组，用于后续 ARP 请求。然后遍历 ARP 请求：循环检查每个未解决的 ARP 请求。如果某个 ARP 请求的发送次数达到最大限制（MAX_SENT_TIME），则从列表中移除它。

同时为每个未超时的 ARP 请求构建以太网头和 ARP 头。以太网头包括源 MAC 地址、目的广播地址和以太网类型。ARP 头包括发送者和接收者的 MAC 和 IP 地址，以及 ARP 操作码（请求）。将构建的 ARP 请求发送到网络，记录 ARP 请求的最新发送时间和累计发送次数。最后遍历 ARP 缓存，移除标记为无效的缓存条目。

Routing-table.cpp:

为了通过掩码运算和地址比较来确定最适合给定 IP 地址的路由条目，从而实现 IP 分组的有效路由，构造了 lookup 函数，检查表是否为空：如果路由表不为空，则开始遍历。
对于表中的每个条目，计算输入 IP 地址和表条目的目的地址，使用各自的掩码。
比较计算后的地址，如果匹配，则检查当前匹配是否是最佳匹配（即具有最长掩码）。
如果是最佳匹配，更新 result 为当前条目。

返回结果：如果找到匹配的条目，返回 result；否则抛出异常

实验结果：

对于 ping 测试，traceroute 测试、文件下载测试均可通过，并且 autograde 跑出结果 45 分。

```
*** Stopping 1 switches
sw0
*** Stopping 3 hosts
client server1 server2
*** Done
Your tests score is 45/45 pts
THIS IS NOT YOUR FINAL SCORE !!!
```

```
mininet> pingall
*** Ping: testing ping reachability
client -> server1 server2
server1 -> client server2
server2 -> client server1
*** Results: 0% dropped (6/6 received)
mininet>
```

```
traceroute to 192.168.2.1 (192.168.2.1), 30 hops max, 60 byte packets
 1  10.0.1.1 (10.0.1.1)  22.126 ms  24.102 ms  23.942 ms
 2  10.0.1.1 (10.0.1.1)  23.829 ms  24.584 ms  69.251 ms
mininet>
```

```
mininet> client wget http://192.168.2.2/index.html
--2023-12-26 02:02:50--  http://192.168.2.2/index.html
Connecting to 192.168.2.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 161 [text/html]
Saving to: 'index.html'

index.html          100%[===================>]     161  --.-KB/s    in 0s

2023-12-26 02:02:51 (13.5 MB/s) - 'index.html' saved [161/161]
```