实验报告

姓名： 袁峥 学号： 2015K8009929008

一、实验题目：路由器转发实验

二、实验内容

1、实验一

①在主机上安装arptables, iptables，用于禁止每个节点的相应功能

sudo apt install arptables iptables

②运行给定网络拓扑(router\_topo.py)

路由器节点r1上执行脚本(disable\_arp.sh, disable\_icmp.sh, disable\_ip\_forward.sh)，禁止协议栈的相应功能

终端节点h1-h3上执行脚本disable\_offloading.sh, disable\_ipv6.sh

③在r1上执行路由器程序

在r1中运行./router，进行数据包的处理

④在h1上进行ping实验

Ping 10.0.1.1 (r1)，能够ping通

Ping 10.0.2.22 (h2)，能够ping通

Ping 10.0.3.33 (h3)，能够ping通

Ping 10.0.3.11，返回ICMP Destination Host Unreachable

Ping 10.0.4.1，返回ICMP Destination Net Unreachable

2、实验二

①构造一个包含多个路由器节点组成的网络

手动配置每个路由器节点的路由表

有两个终端节点，通过路由器节点相连，两节点之间的跳数不少于3跳，手动配置其默认路由表

②连通性测试

终端节点ping每个路由器节点的入端口IP地址，能够ping通

③路径测试

在一个终端节点上traceroute另一节点，能够正确输出路径上每个节点的IP信息

三、实验流程

1、路由器转发流程

①给定数据包，提取该数据包的目的IP地址（u32 dst\_ip）

注意进行字节序转换：数据包中的都是网络字节序，本地存储的数据结构都是本地字节序

②遍历路由表（链表），查找满足条件的条目

rt\_entry\_t \*entry = NULL;

list\_for\_each\_entry(entry, &rtable, list) {}

最长前缀匹配：(entry->dest & entry->mask) == (dst\_ip & entry->mask)，且掩码长度最长

③如果查找到相应条目，则将数据包从该条目对应端口转出，否则回复目的网络不可达(ICMP Dest Network Unreachable)

④路由器在转发数据包时

对IP头部的TTL值进行减一操作，如果该值<= 0，则将该数据包丢弃，并回复ICMP信息

IP头部数据已经发生变化，需要重新设置checksum

将以太网头部的源MAC地址设置为转发端口的MAC地址

将目的MAC地址设置为对应的MAC地址

2、查询IP地址对应的MAC地址

MAC地址在局域网内转发数据包时起作用

应查询下一跳IP地址对应的MAC地址

entry->gw (路由器端口与目的地址不在同一网段）

dst\_ip (路由器端口与目的地址在同一网段）

3、ARP查询

当所查找IP对应MAC地址不在ARP缓存中时，使用ARP机制查询下一跳IP对应的MAC地址

路由器维护一个缓存ARP相关内容的数据结构：arpcache

arpcache缓存两类数据：IP->MAC映射条目和查找不到相应条目而等待ARP应答的数据包

4、ARP缓存操作

①查找IP->MAC映射

如果在arp缓存中找到相应映射，则填充数据包的目的MAC地址，并转发该数据包

否则，将该数据包缓存在arpcache->req\_list中，并发送ARP请求

②收到新的IP->MAC映射

将该映射写入arp缓存中 ，如果缓存已满（最多32条），则随机替换掉其中一个

将在缓存中等待该映射的数据包，依次填写目的MAC地址，转发出去，并删除掉相应缓存数据包

③每1秒钟，运行arpcache\_sweep操作

如果一个缓存条目在缓存中已存在超过了15秒，将该条目清除

如果一个IP对应的ARP请求发出去已经超过了1秒，重新发送ARP请求

如果发送超过5次仍未收到ARP应答，则对该队列下的数据包依次回复ICMP（Destination Host Unreachable）消息，并删除等待的数据包

综上，本次实验需要实现的内容有：

①处理ARP请求和应答

收到ARP请求时，如果Target Proto Addr为本端口地址，则ARP应答

转发数据包时，如果ARP缓存中没有相应条目，则发送ARP请求

②ARP缓存管理

进行ARP查询、更新等操作

③IP地址查找和IP数据包转发

收到数据包后，查找对应的转发端口；更新IP头部，转发数据包

④发送ICMP数据包

TTL值为0；查找不到路由表条目；ARP查询失败；收到ping本端口的包

四、实验结果

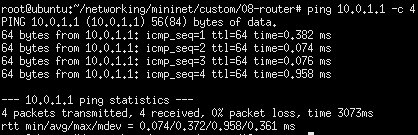
1、实验一

网络拓扑结构：

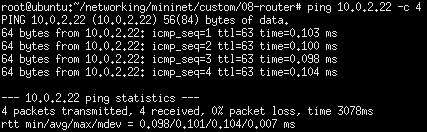


运行结果：

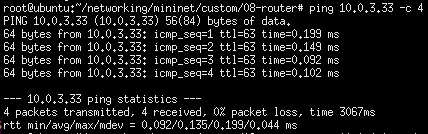
①Ping 10.0.1.1(r1)，能够ping通



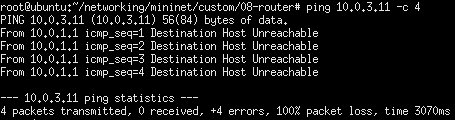
②Ping 10.0.2.22 (h2)，能够ping通



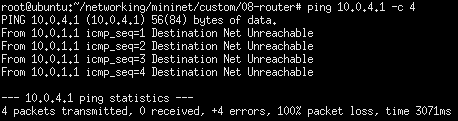
③Ping 10.0.3.33 (h3)，能够ping通



④Ping 10.0.3.11，返回ICMP Destination Host Unreachable



⑤Ping 10.0.4.1，返回ICMP Destination Net Unreachable



2、实验二

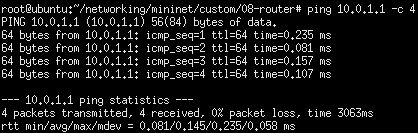
网络拓扑结构：



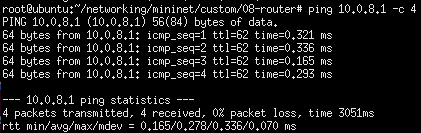
运行结果：

①连通性测试（从h1节点）

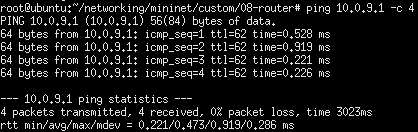
Ping 10.0.1.1(r1)



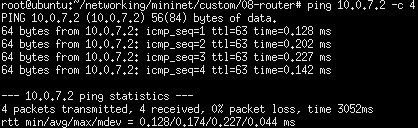
Ping 10.0.8.1(r2)



Ping 10.0.9.1(r3)



Ping 10.0.7.2(r4)

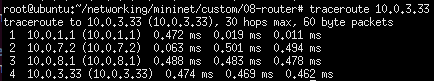


②路径测试（从h1节点）

traceroute 10.0.2.22

C:\Users\YZ\AppData\Roaming\Tencent\Users\593538317\TIM\WinTemp\RichOle\6J$1S1KWOH_QKTW$THWK8UT.png

traceroute 10.0.3.33



五、结果分析

1、实验一

实验一中从h1节点进行的5个ping测试结果符合预期。

10.0.1.1为从h1出发的第一跳路由器，可以ping通。

10.0.2.22和10.0.3.33分别为另两个终端，均可以ping通。

10.0.3.11是一个不存在的终端节点。ICMP数据包从10.0.1.11发出，根据h1路由表发送到默认网关10.0.1.1，然后根据路由器r1的路由表，判断10.0.3.11的下一跳直接为该IP地址，再在ARP缓存中查找10.0.3.11所对应MAC地址，无法查找到广播发送ARP请求。由于10.0.3.11地址不存在，因此不会得到ARP应答。在等待超时后从10.0.1.1发送ICMP的ARP查询失败数据包。

10.0.4.1是一个不存在的IP网段。ICMP数据包从10.0.1.11发出，首先根据h1的路由表查找到下一跳地址为10.0.1.1，到达路由器r1后，查找路由器的路由表，中间无法找到10.0.4.0网段的条目，因此返回ICMP\_NET\_UNREACH的ICMP数据包。

2、实验二

实验二中从h1节点向4个路由器节点的入端口IP地址，均可以ping通，结果符合预期。

路径测试选取了两种情况。一种是两个终端节点中间只有一个路由器，从h1出发经过r1到达h2，另一种是经过较多的路由器，从h1出发陆续经过r1、r4、r2最终到达h3。从路径测试的输出可以看到情况与网络拓扑中一致。

六、实验总结

本次实验从代码量来看与之前的实验相比增加了不少，所需要实现的函数也比较多，但是好在基本每个函数的注释中都把具体要实现的内容写清楚了。在编写代码之前根据课件和课本把IP包、ICMP包、ARP包、路由表等概念以及路由器转发数据包的流程弄明白，然后熟悉了本次实验的其他代码，再开始代码的编写。在调试过程中碰到了如下几个问题：

1、在熟悉实验源代码时不够仔细，误以为checksum会在函数中将相关字段完成更新，但实际上需要通过函数返回值在函数外进行校验和的更新。

2、在处理icmp\_send\_packet的时候，不能偷懒将源数据包的源和目的IP交换作为icmp包的源和目的IP，icmp包的源IP必须是发送iface端口的IP，不然在traceroute中输出的信息会有错误，每个中间节点的IP都会显示为数据包的目的IP。

3、iface中的内容是本地的，使用时不需要进行网络序到本地序的转换。

4、IP包TTL超时的判断条件是TTL<=1，而不是TTL<1，否则在traceroute时中间的节点信息不正确。

5、在源代码中对于路由表中网关的解析不正确，导致转发数据包查找路由表时使用网关信息出现问题。

此外，在本次实验的调试时，采用的方法基本是大量的在程序中加入调试输出，这个方法即增加了许多无效代码，根据大量调试输出找到问题根源也比较费力，以后在实验中可以好好利用wireshark这个工具，可以大大减少调试的难度。