实验报告

姓名： 袁峥 学号： 2015K8009929008

一、实验题目：网络地址转换实验

二、实验内容

1、运行给定网络拓扑(nat\_topo.py)

2、在n1上执行NAT程序

首先，执行脚本(disable\_arp.sh, disable\_icmp.sh, disable\_ip\_forward.sh)，禁止协议栈的相应功能

在n1中运行nat，进行数据包的处理

3、在h2上运行HTTP服务

执行脚本(disable\_offloading.sh)，防止协议栈生成错误的checksum

执行python -m SimpleHTTPServer， 启动HTTP服务

4、在h1上访问h2的HTTP服务

执行脚本(disable\_offloading.sh)

h1 # wget http://159.226.39.123:8000

三、实验流程

1、NAT地址翻译

①如果为DIR\_OUT方向数据包且没有对应连接映射（SYN）

saddr = external\_iface->ip;

sport = assign\_external\_port();

建立连接映射关系

(internal\_ip, internal\_port) <-> (external\_ip, external\_port)

前者为私网地址和Port，后者为映射后的公网地址和Port

②其他合法数据包

查找映射关系，进行(internal\_ip, internal\_port) <-> (external\_ip, external\_port)之间的转换

更新IP/TCP数据包头部字段(包括校验和)

上述工作流程如下图：



2、根据数据包查找映射关系

(serv\_ip, serv\_port) 是地址翻译中的不变量，但不具有唯一性，可以先用(serv\_ip, serv\_port) 定位到一组映射结构（链表），再根据数据包方向，决定用(int\_ip, int\_port) 还是(ext\_ip, ext\_port) 来确定唯一的映射结构

3、区分数据包方向

①当源地址为内部地址，且目的地址为外部地址时，方向为DIR\_OUT

②当源地址为外部地址，且目的地址为external\_iface地址时，方向为DIR\_IN

4、NAT老化操作

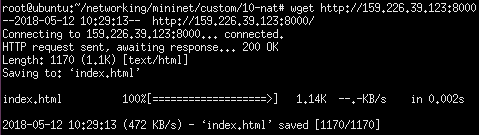
对认为已经结束的连接进行老化操作

①双方都已发送FIN且回复相应ACK的连接，或者一方发送RST包的连接，可以直接回收

②双方已经超过60秒未传输数据的连接，认为其已经传输结束，可以回收

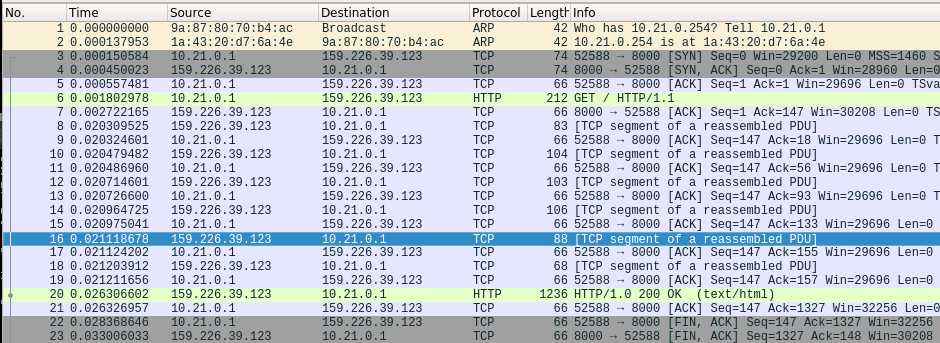
四、实验结果

1、节点h1结果

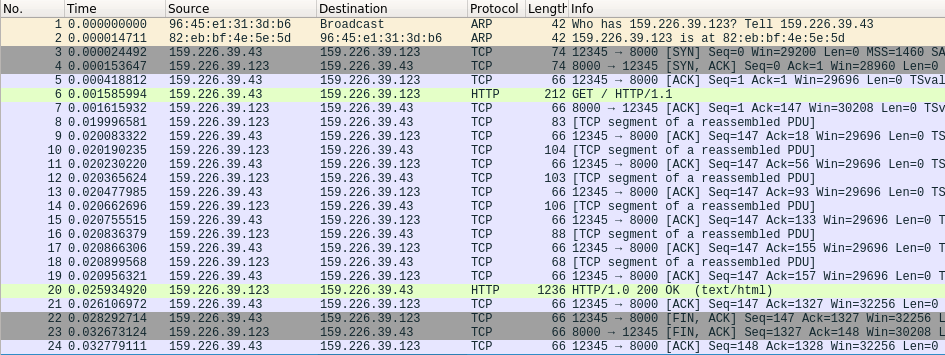


2、wireshark结果

①n1-eth0



②n1-eth1



五、结果分析

从结果可以看到节点h1成功与h2建立了连接，并获取到了服务器端的数据。从wireshark结果可以看到整个流程符合预期，先通过ARP包获取路由器转发端口的MAC地址，然后再通过NAT路由器更换源IP地址或目的IP地址，达到私网地址与公网地址相互访问的效果。

六、实验总结

此次实验完成起来整体较为顺利，要完成了代码也十分少。在整理流程的过程中碰到的一个问题是，一开始没有想明白是用什么内容来完成到存储映射关系的链表数组的Hash映射的，在反复阅读PPT并思考后，明白了应该用要访问的服务器端的IP地址及其端口来进行Hash映射，因为服务器端的IP地址及其端口号无论在DIR\_IN还是DIR\_OUT时都不会改变，因此可以先用此映射到一个链表，然后再在里面逐个查找相应需要的映射转换条目。

此外在编写代码时由于需要在双方都已发送FIN且回复相应ACK的连接时进行端口的回收，因此利用struct nat\_connection conn中的internal\_fin和external\_fin来完成此项工作，当TCP为FIN包时，将相应的internal\_fin或external\_fin置1，如果之后又收到对端口发来的ACK包，则将相应fin标志置2，表示此方向已经断开连接。在老化操作中检查如果internal\_fin或external\_fin都为2，则回收该端口并删除该映射转换条目。

在完成代码编写后，我又将libipstack.a替换为08-router实验中自己完成的代码，在进行适当修改后，同样实现了NAT地址转化的工作并成功建立连接进行访问，说明前面实验中完成的代码可靠性还是较高的。