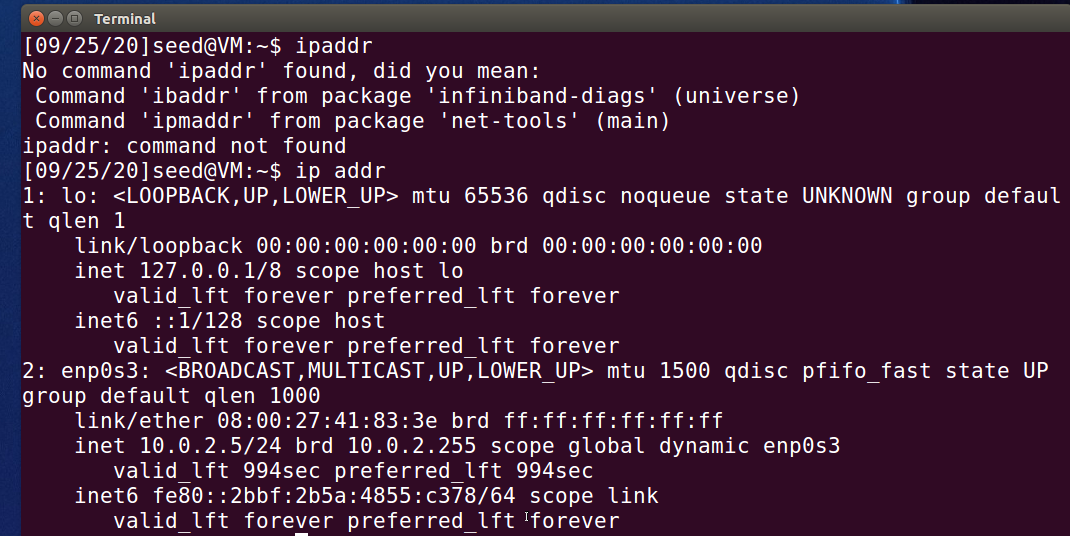
**lab7实验报告**

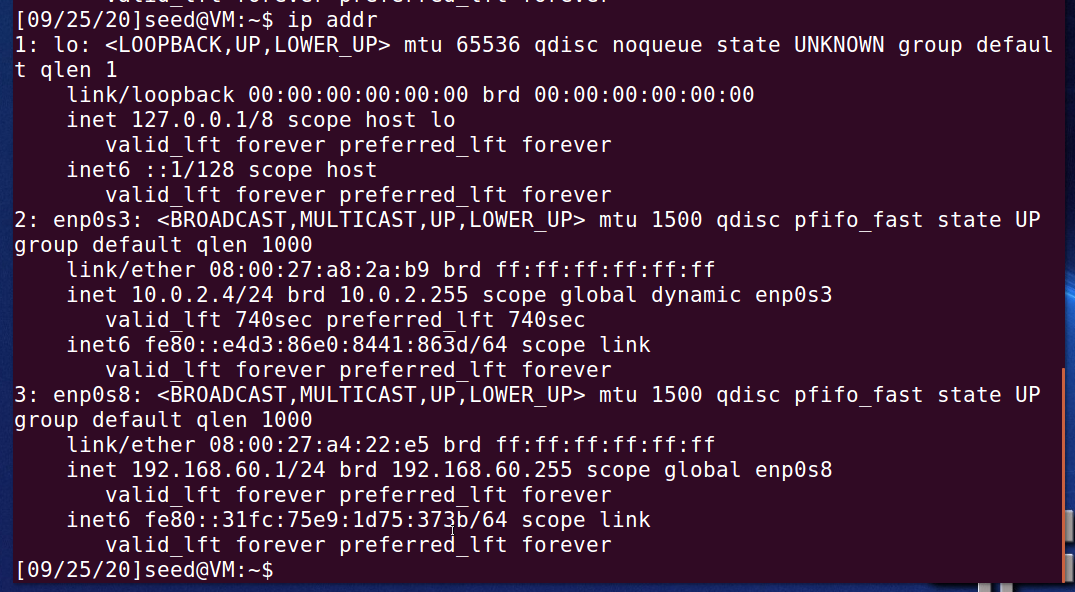
57117133 臧宇豪

**Task 1: Network Setup**

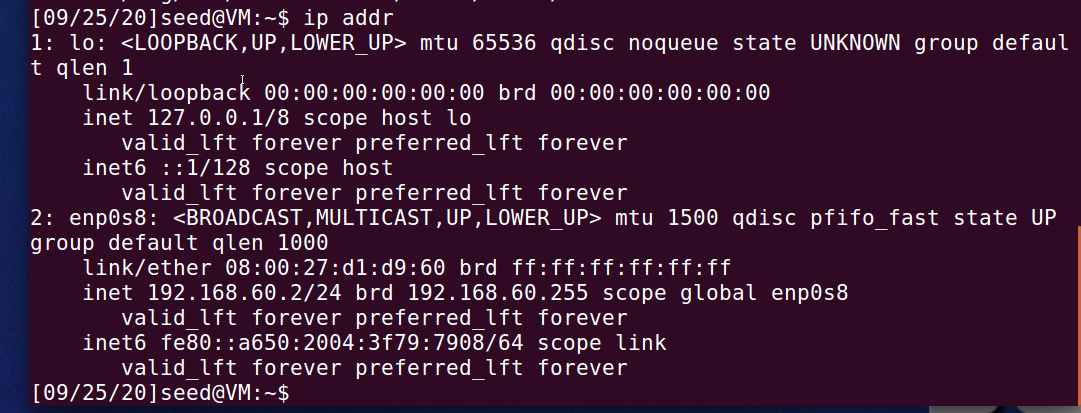
主机U：10.0.2.5



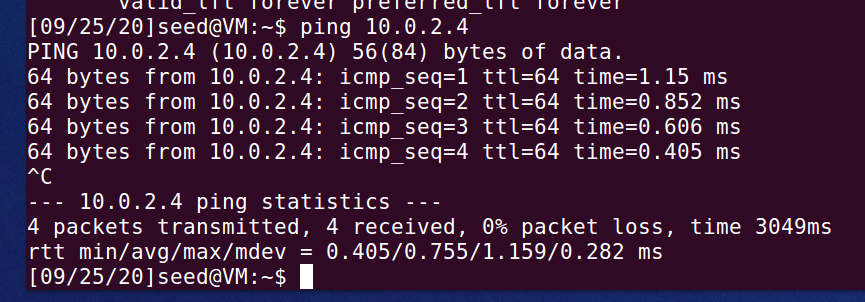
VPN服务器：10.0.2.4和192.168.60.1



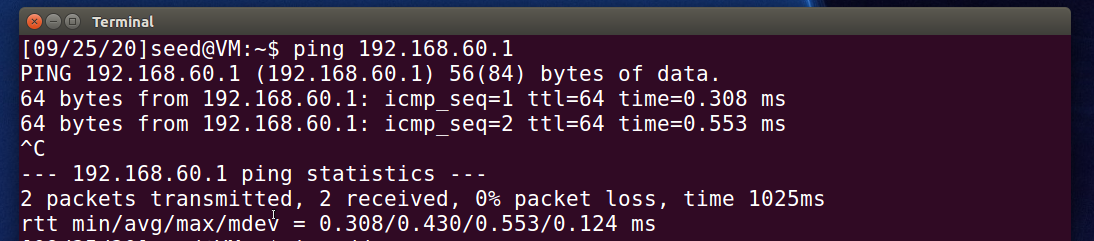
主机V：192.168.60.2



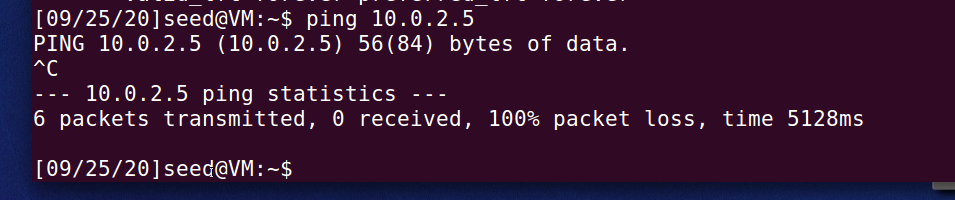
U主机ping VPN服务器：



V主机ping VPN服务器：



主机V ping主机U：

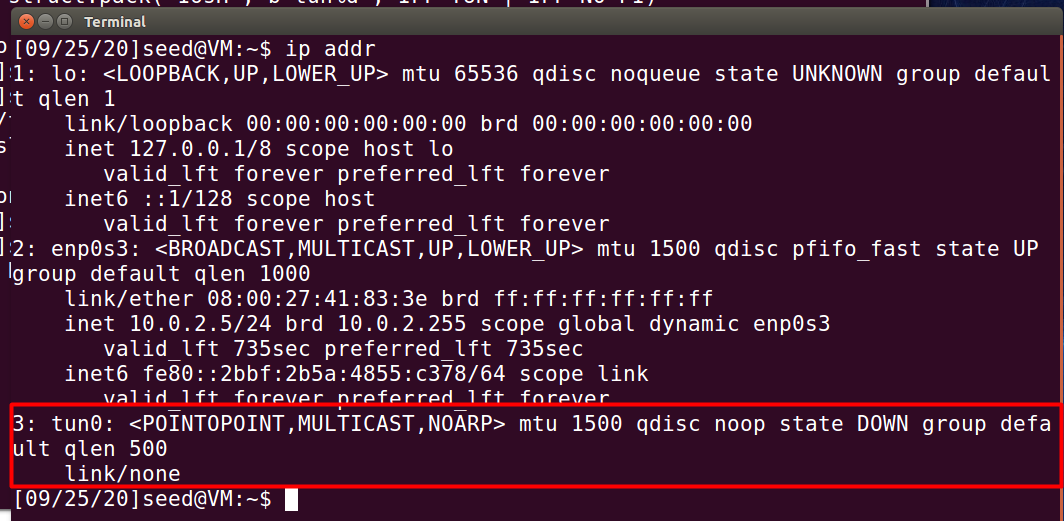


**Task 2: Create and Configure TUN Interface**

按题目要求在主机U上建立tun.py程序。

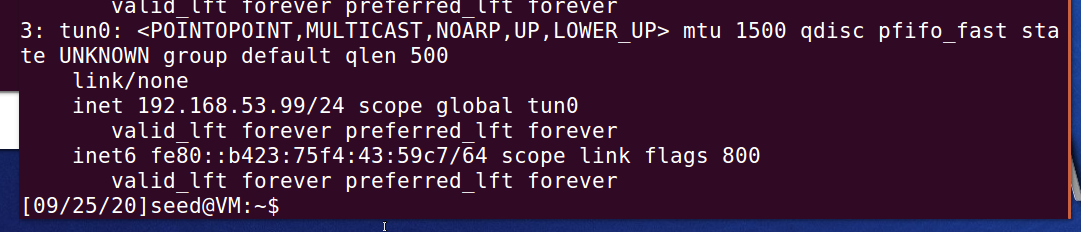
**a: Name of the Interface**

运行程序后发现多出了tun0接口：



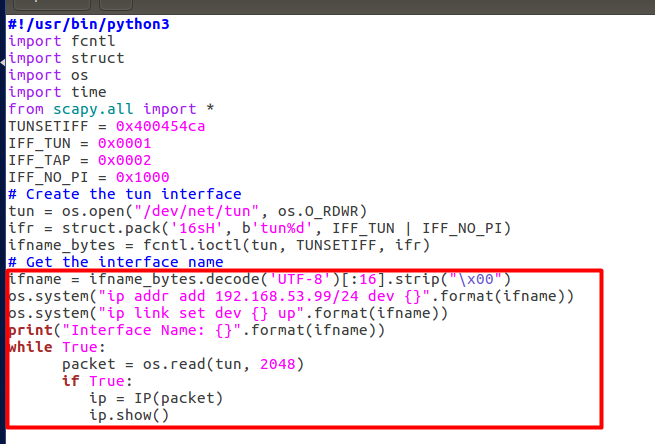
**b: Set up the TUN Interface**

配置后发现tun0接口有了ip地址：

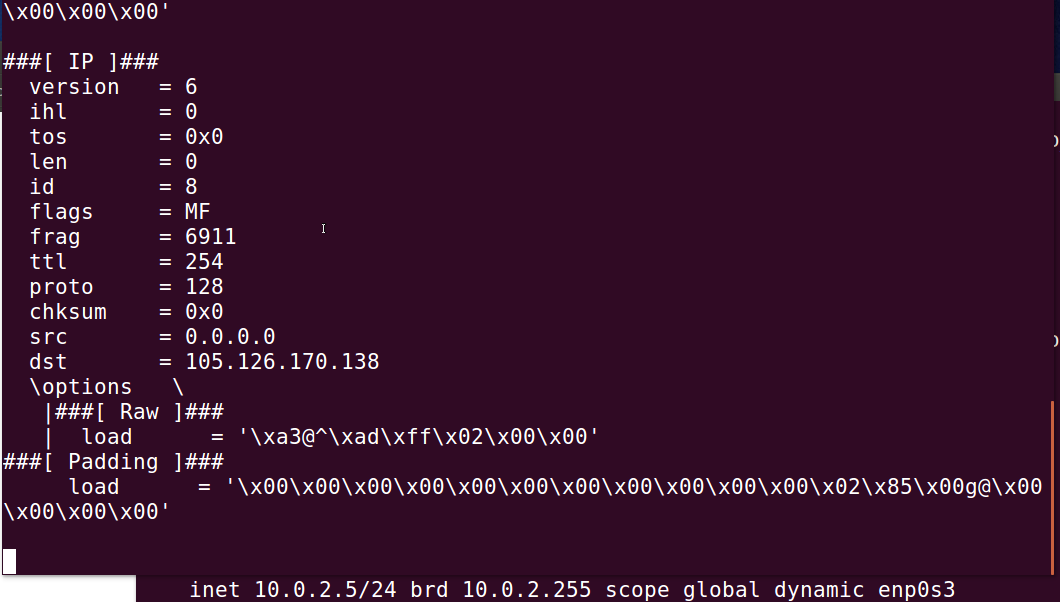


**c: Read from the TUN Interface**

代码更改：



打印报文：



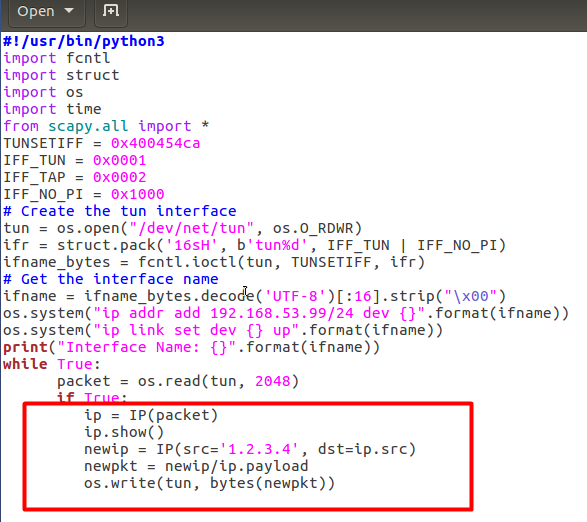
ping 192.168.53.1：



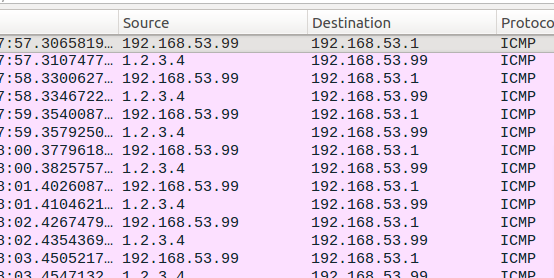
当ping 192.168.60.0/24中的主机时，tun.py没有输出。原因是这个ping命令没有从tun0端口发出。

**d: Write to the TUN Interface**

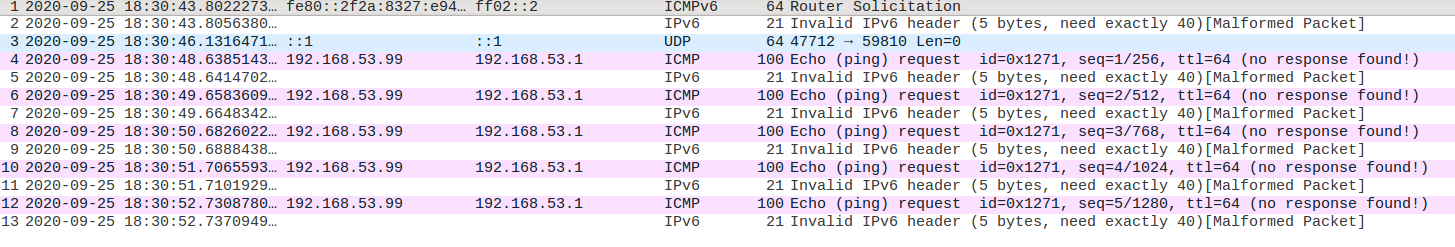
修改代码：



运行结果用wireshark查看：



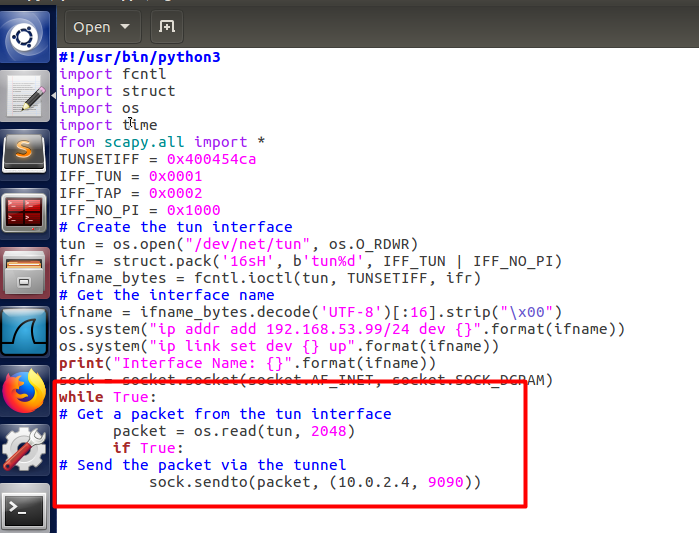
当把数据包改成数据时：



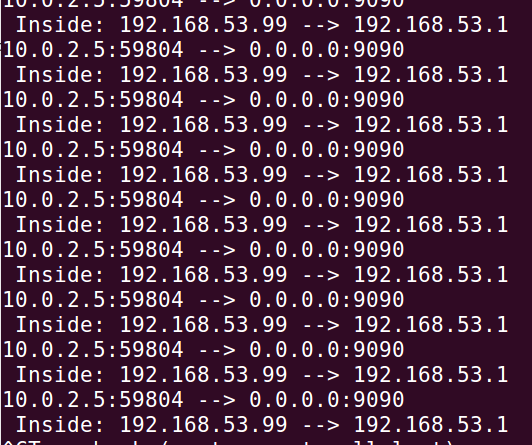
显示非法报文头。

**Task 3: Send the IP Packet to VPN Server Through a Tunnel**

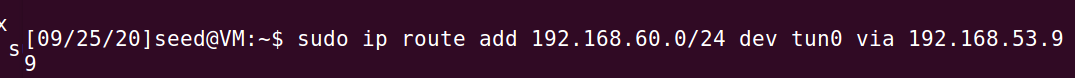
按要求运行服务器程序，并修改客户端程序：



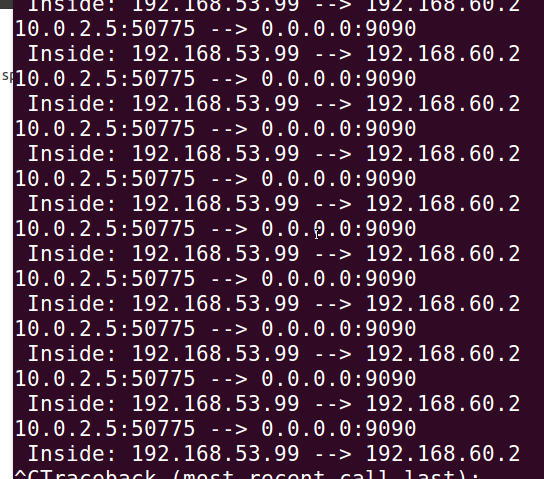
ping 192.168.53.0/24网段：



配置路由：

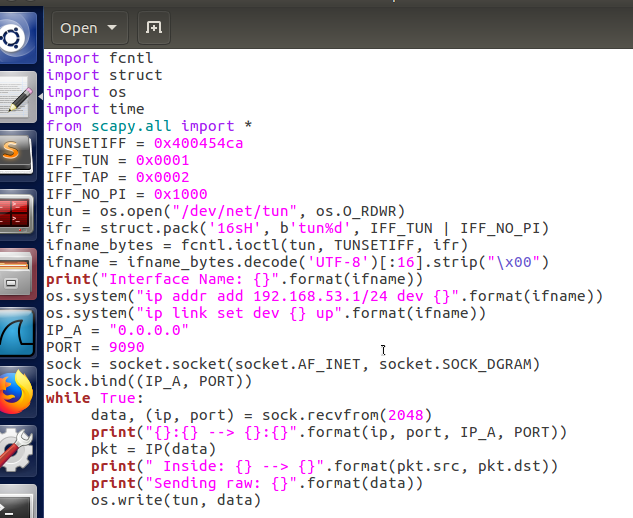


ping 192.168.60.2：

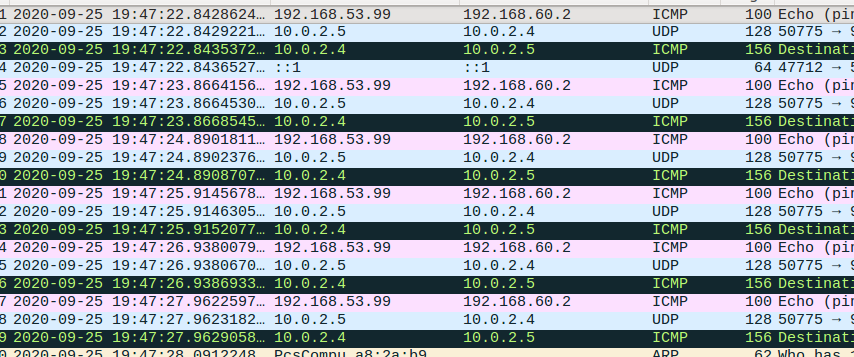


**Task 4: Set Up the VPN Server**

编辑VPN服务器的代码：

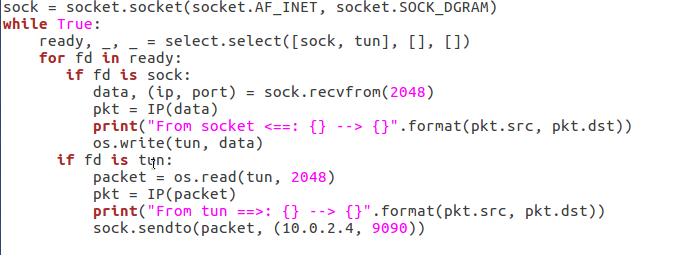


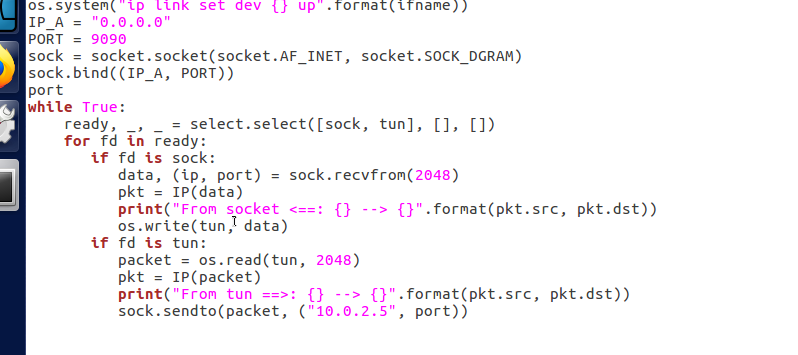
U主机ping主机V，报文记录如下：



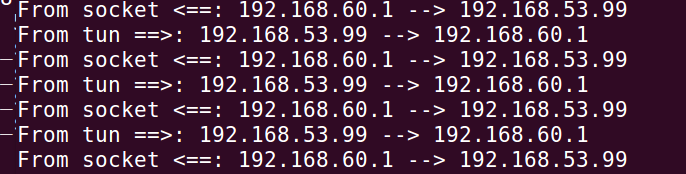
**Task 5: Handling Traffic in Both Directions**

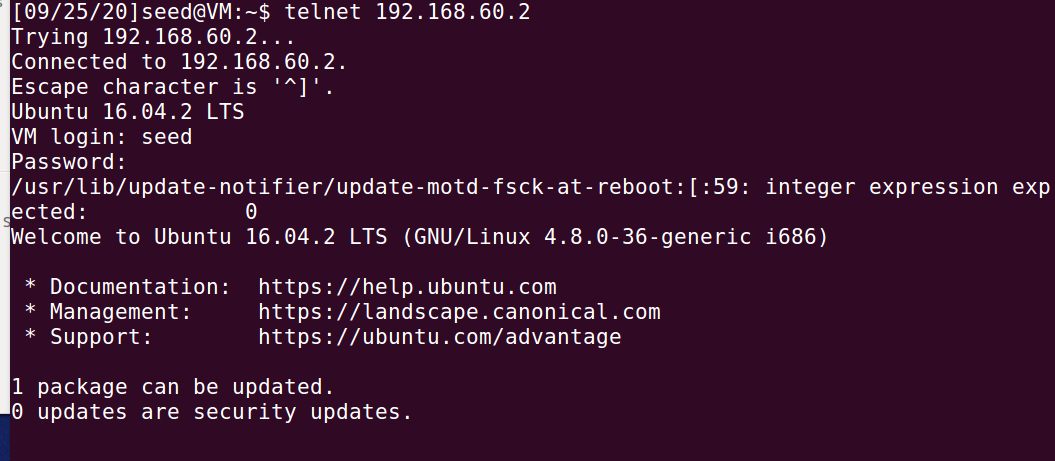
对服务器和客户端代码都进行修改：





可以发现能够成功通讯：



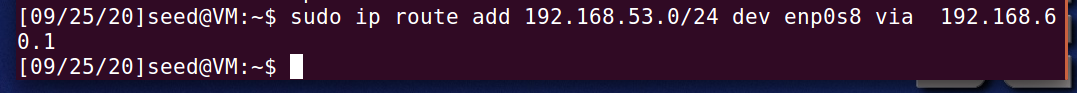


**Task 6: Tunnel-Breaking Experiment**

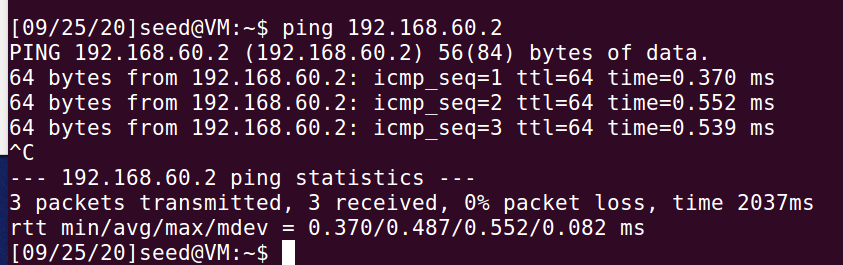
关闭其中一个tun.py后，发现无论输入什么，telnet页面不再有显示。重新打开程序后，又有了显示，可能的原因是报文缓存进入了缓冲区。

**Task 7: Routing Experiment on Host V**

更改路由：



U ping V:



**Task 8: Experiment with the TUN IP Address**

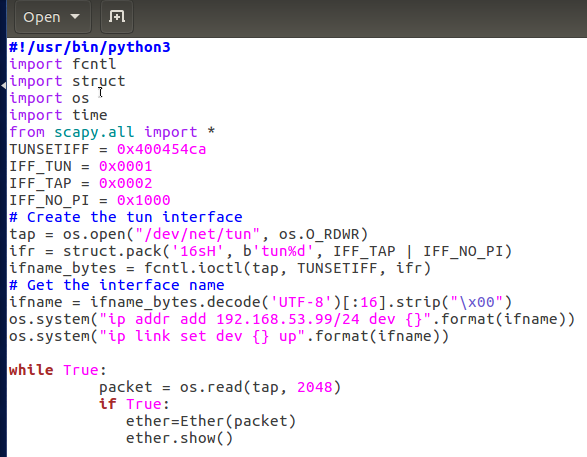
将U主机的tun端口的ip地址为192.168.30.99后，无法ping通。通过查看wireshark，主机U成功发出了UDP报文，服务器也成功收到了V的ICMP响应报文，但是服务器没有将其转发给主机U。

原因是在反向路由过滤中，ip地址和路由表中的不匹配，造成了丢包。

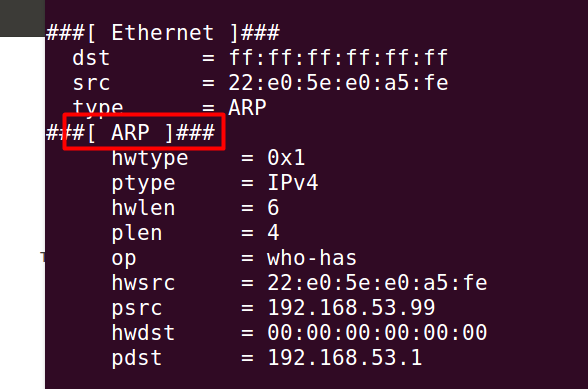
解决办法是添加路由表项，或者关闭反向过滤。

**Task 9: Experiment with the TAP Interface**

更改代码：



ping 192.168.53.1，发现有ARP报文：



这是因为ping时会发送ARP请求来查询IP，询问是否有该设备。