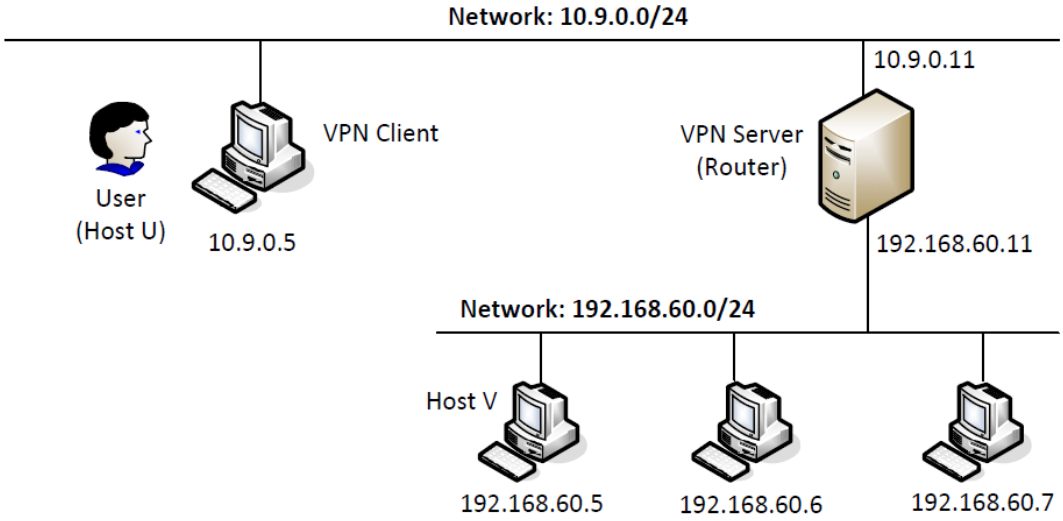
lab7-report

571118106 强珂阳



task 1 Network Setup

验证 U 可以和 VPN Server 相通，并在路由器上捕获报文。

```
root@70a884552c63:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.123 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.213 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.072 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.057 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.076 ms
^C
--- 10.9.0.11 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5116ms
rtt min/avg/max/mdev = 0.057/0.100/0.213/0.054 ms
```

```
root@c24e54cb21f4:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:33:08.371873 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 1, l
ength 64
12:33:08.371885 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 1, len
gth 64
12:33:09.405455 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 2, l
ength 64
12:33:09.405493 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 2, len
gth 64
12:33:10.415621 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 3, l
ength 64
12:33:10.415713 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 3, len
gth 64
12:33:11.439057 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 4, l
ength 64
12:33:11.439079 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 4, len
gth 64
12:33:12.463027 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 14, seq 5, l
ength 64
12:33:12.463043 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 14, seq 5, len
gth 64
```

验证 VPN Server 可以和 V 相通，并在路由器上捕获报文。

```
root@ff649d991563:/# ping 192.168.60.11
PING 192.168.60.11 (192.168.60.11) 56(84) bytes of data.
64 bytes from 192.168.60.11: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 192.168.60.11: icmp_seq=2 ttl=64 time=0.113 ms
64 bytes from 192.168.60.11: icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from 192.168.60.11: icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from 192.168.60.11: icmp_seq=5 ttl=64 time=0.068 ms
^C
--- 192.168.60.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 0.058/0.076/0.113/0.019 ms
```

```
root@c24e54cb21f4:/# tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
12:36:55.269731 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 1, length 64
12:36:55.269749 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 1, length 64
12:36:56.271637 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 2, length 64
12:36:56.271666 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 2, length 64
12:36:57.294817 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 3, length 64
12:36:57.294838 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 3, length 64
12:36:58.318859 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 4, length 64
12:36:58.318875 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 4, length 64
12:36:59.343899 IP 192.168.60.5 > 192.168.60.11: ICMP echo request, id 37,
seq 5, length 64
12:36:59.343919 IP 192.168.60.11 > 192.168.60.5: ICMP echo reply, id 37, se
q 5, length 64
12:37:00.498824 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, lengt
```

验证 U 和 V 不通。

```
root@70a884552c63:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6148ms
```

task 2.a Name of the Interface

修改代码：

```
# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'qky%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)
```

运行程序，发现成功修改了接口名称：

```
root@70a884552c63:/volumes# chmod a+x tun.py
root@70a884552c63:/volumes# tun.py
Interface Name: qky0
```

```
root@70a884552c63:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d
efault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: qky0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group
 default qlen 500
    link/none
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue sta
te UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
       valid_lft forever preferred_lft forever
```

task 2.b Set up the TUN Interface

增加两行代码:

```
23 os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
24 os.system("ip link set dev {} up".format(ifname))
```

```
root@70a884552c63:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d
efault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: qky0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel
state UNKNOWN group default qlen 500
    link/none
    inet 192.168.53.99/24 scope global qky0
       valid_lft forever preferred_lft forever
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue sta
te UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
```

再次运行代码，发现接口有了 IP 地址。

task 2.c Read from the TUN Interface

替换 while 循环:

```
while True:
        # Get a packet from the tun interface
        packet = os.read(tun, 2048)
        if packet:
                ip = IP(packet)
                print(ip.summary())
```

U 上运行代码并 ping 192.168.53.1，请求无响应，因为该主机不存在。

```
root@70a884552c63:/# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
^C
--- 192.168.53.1 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11293ms
```

```
root@70a884552c63:/volumes# tun.py
Interface Name: qky0
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
```

U 上运行代码并 ping 192.168.60.1，可以 ping 通，但程序没有打印任何结果，因为没有添加路由。

```
root@70a884552c63:/# ping 192.168.60.1
PING 192.168.60.1 (192.168.60.1) 56(84) bytes of data.
64 bytes from 192.168.60.1: icmp_seq=1 ttl=64 time=0.058 ms
64 bytes from 192.168.60.1: icmp_seq=2 ttl=64 time=0.074 ms
64 bytes from 192.168.60.1: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 192.168.60.1: icmp_seq=4 ttl=64 time=0.095 ms
64 bytes from 192.168.60.1: icmp_seq=5 ttl=64 time=0.062 ms
64 bytes from 192.168.60.1: icmp_seq=6 ttl=64 time=0.056 ms
64 bytes from 192.168.60.1: icmp_seq=7 ttl=64 time=0.058 ms
64 bytes from 192.168.60.1: icmp_seq=8 ttl=64 time=0.086 ms
64 bytes from 192.168.60.1: icmp_seq=9 ttl=64 time=0.078 ms
64 bytes from 192.168.60.1: icmp_seq=10 ttl=64 time=0.079 ms
64 bytes from 192.168.60.1: icmp_seq=11 ttl=64 time=0.044 ms
64 bytes from 192.168.60.1: icmp_seq=12 ttl=64 time=0.046 ms
^C
--- 192.168.60.1 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11263ms
rtt min/avg/max/mdev = 0.044/0.067/0.095/0.015 ms
```

```
root@70a884552c63:/volumes# tun.py
Interface Name: qky0
^CTraceback (most recent call last):
  File "./tun.py", line 28, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt
```

task 2.d Write to the YTUN Interface

修改 while 循环:

```python
while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if True:
        pkt = IP(packet)
        print(pkt.summary())

        if ICMP in pkt:
            newip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
            newip.ttl = 99
            newicmp = ICMP(type = 0, id = pkt[ICMP].id, seq = pkt[ICMP].seq)

            if pkt.haslayer(Raw):
                data = pkt[Raw].load
                newpkt = newip/newicmp/data
            else:
                newpkt = newip/newicmp
        os.write(tun, bytes(newpkt))
```

```
root@70a884552c63:/# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
64 bytes from 192.168.53.1: icmp_seq=1 ttl=99 time=2.17 ms
64 bytes from 192.168.53.1: icmp_seq=2 ttl=99 time=2.15 ms
64 bytes from 192.168.53.1: icmp_seq=3 ttl=99 time=2.03 ms
64 bytes from 192.168.53.1: icmp_seq=4 ttl=99 time=1.96 ms
64 bytes from 192.168.53.1: icmp_seq=5 ttl=99 time=3.04 ms
64 bytes from 192.168.53.1: icmp_seq=6 ttl=99 time=2.03 ms
64 bytes from 192.168.53.1: icmp_seq=7 ttl=99 time=2.07 ms
64 bytes from 192.168.53.1: icmp_seq=8 ttl=99 time=1.97 ms
64 bytes from 192.168.53.1: icmp_seq=9 ttl=99 time=1.80 ms
64 bytes from 192.168.53.1: icmp_seq=10 ttl=99 time=3.48 ms
64 bytes from 192.168.53.1: icmp_seq=11 ttl=99 time=2.12 ms
^C
--- 192.168.53.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10050ms
rtt min/avg/max/mdev = 1.797/2.254/3.484/0.493 ms
```

```
root@70a884552c63:/volumes# tun.py
Interface Name: qky0
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
^CTraceback (most recent call last):
  File "./tun.py", line 28, in <module>
    packet = os.read(tun, 2048)
KeyboardInterrupt
```

task 3 Send the IP Packet to VPN Server Through a Tunnel

修改代码 tun_client.py:

```
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))

# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP="10.9.0.11"
SERVER_PORT=9090

while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    if packet:
          pkt = IP(packet)
          print(pkt.summary())
          sock.sendto(packet,(SERVER_IP,SERVER_PORT))
```

修改代码 tun_server.py:

```
# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))

os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))

server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP = "0.0.0.0"
SERVER_PORT = 9090
server.bind((SERVER_IP, SERVER_PORT))

while True:
        data,(ip, port) = server.recvfrom(2048)
        print("{}:{} --> {}:{}".format(ip, port, SERVER_IP, SERVER_PORT))
        pkt = IP(data)
        print("Inside: {} --> {}".format(pkt.src, pkt.dst))
```

在 U 上 ping 192.168.53.1，查看 server 和 client 的运行结果:

```
root@c24e54cb21f4:/volumes# tun_server.py
Interface Name: qky0
RTNETLINK answers: File exists
10.9.0.5:51780 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:51780 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:51780 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:51780 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:51780 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:51780 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.1
10.9.0.5:51780 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.53.1
```

```
root@70a884552c63:/volumes# tun_client.py
Interface Name: qky0
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.1 echo-request 0 / Raw
```

Server 上外部是 10.9.0.5 到 0.0.0.0，内部是 192.168.53.88 到 192.168.53.1.

task 4 Set Up the VPN Server

修改代码 tun_server.py：

```
# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'qky%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))

server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP = "0.0.0.0"
SERVER_PORT = 9090
server.bind((SERVER_IP, SERVER_PORT))

while True:
        data,(ip, port) = server.recvfrom(2048)
        print("{}:{} --> {}:{}".format(ip, port, SERVER_IP, SERVER_PORT))
        pkt = IP(data)
        print("Inside: {} --> {}".format(pkt.src, pkt.dst))
        os.write(tun, data)
        print("write")
```

server 上看到：

```
root@c24e54cb21f4:/# tcpdump -nni eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
16:49:17.775089 IP6 fe80::503e:6ff:fe79:c85b > ff02::2: ICMP6, router solicitation, length 16
16:49:40.955678 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 99, seq 1, length 64
16:49:40.955706 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 99, seq 1, length 64
16:49:41.968736 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 99, seq 2, length 64
16:49:41.968799 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 99, seq 2, length 64
16:49:42.993644 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 99, seq 3, length 64
16:49:42.993694 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 99, seq 3, length 64
16:49:44.016216 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 99, seq 4, length 64
16:49:44.016285 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 99, seq 4, length 64
16:49:45.040931 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 99, seq 5, length 64
16:49:45.040970 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 99, seq 5, length 64
16:49:46.065003 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 99, seq 6, length 64
16:49:46.065028 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 99, seq 6, length 64
16:49:46.196190 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
16:49:46.196222 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
16:49:46.196231 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
16:49:46.196237 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
16:49:47.088305 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 99, seq 7, length 64
16:49:47.088330 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 99, seq 7, length 64
```

task 5 Handling Traffic in Both Directions

修改代码 client:

```
# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'qky%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.1/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP = "0.0.0.0"
SERVER_PORT = 9090
ip = '10.9.0.5'
port = 10000
sock.bind((SERVER_IP, SERVER_PORT))
fds = [sock,tun]
while True:
        ready,_,_=select.select(fds,[],[])
        for fd in ready:
                if fd is sock:
                        print("sock...")
                        data,(ip, port) = sock.recvfrom(2048)
                        print("{}:{} --> {}:{}".format(ip, port, SERVER_IP, SERVER_PORT))
                        pkt = IP(data)
                        print("Inside: {} --> {}".format(pkt.src, pkt.dst))
                        os.write(tun, data)
                if fd is tun:
                        print("tun...")
                        packet = os.read(tun,2048)
                        pkt = IP(packet)
                        print("Return: {}--{}".format(pkt.src,pkt.dst))
                        sock.sendto(packet,(ip,port))
```

server:

```
# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'qky%d', IFF_TUN | IFF_NO_PI)
ifname_bytes  = fcntl.ioctl(tun, TUNSETIFF, ifr)

# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))

server = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
SERVER_IP = "0.0.0.0"
SERVER_PORT = 9090
server.bind((SERVER_IP, SERVER_PORT))

while True:
        data,(ip, port) = server.recvfrom(2048)
        print("{}:{} --> {}:{}".format(ip, port, SERVER_IP, SERVER_PORT))
        pkt = IP(data)
        print("Inside: {} --> {}".format(pkt.src, pkt.dst))
        os.write(tun, data)
        print("write")
```

U 可以 ping 通 V:

```
root@70a884552c63:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=3.91 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=2.20 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=2.71 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=2.29 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=3.59 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=2.56 ms
^C
--- 192.168.60.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5104ms
rtt min/avg/max/mdev = 2.201/2.876/3.912/0.645 ms
```

server:

```
root@c24e54cb21f4:/volumes# tun_server1.py
Interface Name: qky0
RTNETLINK answers: File exists
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
```

client:

```
root@70a884552c63:/volumes# tun_client1.py
Interface Name: qky0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
From socket: 192.168.60.5 --> 192.168.53.99
```

U 可以 telnet 登录 V:

```
root@70a884552c63:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
ff649d991563 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@ff649d991563:~$
```

server：

```
tun...
Return: 192.168.60.5--192.168.53.99
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
tun...
Return: 192.168.60.5--192.168.53.99
sock...
10.9.0.5:46415 --> 0.0.0.0:9090
Inside: 192.168.53.99 --> 192.168.60.5
```

client：

```
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet A
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet A
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet PA / Raw
From socket: 192.168.60.5 --> 192.168.53.99
IP / TCP 192.168.53.99:36652 > 192.168.60.5:telnet PA / Raw
```

task 6 Tunnel-BreakingExperiment

停止 client 后，敲击键盘并没有输入，重新连接之后，会显示出来：

```
root@70a884552c63:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
ff649d991563 login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

seed@ff649d991563:~$ qwertqiangkeyang
```