Lab report: vpn tunneling

Name: Guo Yuchen
Student ID: 1004885

Task 1: Network Setup

• Host U can communicate with VPN Server.

```
root@gyc4885-client-10:/# 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=0.182 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.093 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.095 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.054 ms
^C
--- 10.9.0.11 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3067ms
rtt min/avg/max/mdev = 0.054/0.106/0.182/0.046 ms
```

• VPN Server can communicate with Host V.

```
root@gyc4885-server-router:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.228 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.159 ms
^C
--- 10.9.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.062/0.149/0.228/0.068 ms
```

• Host U should not be able to communicate with Host V.

```
root@gyc4885-client-10:/# 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 ---
175 packets transmitted, 0 received, 100% packet loss, time 178323ms
```

• Run tcpdump on the router, and sniff the traffic on each of the network. Show that you can capture packets.

192.168.60.5 ping 192.168.60.6

```
root@gyc4885-server-router:/# tcpdump -i any -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol de
code
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size
262144 bytes
04:19:54.718284 ARP, Request who-has 192.168.60.6 tell 192.168.60.5, l
ength 28
```

192.168.60.6 ping 10.9.0.5

10.9.0.5 ping 10.9.0.11

```
04:21:59.037088 ARP, Request who-has 192.168.60.11 tell 192.168.60.6, length 28
04:21:59.037114 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
04:21:59.037149 IP 192.168.60.6 > 10.9.0.5: ICMP echo request, id 35, seq 1, length 64
04:21:59.037161 IP 192.168.60.6 > 10.9.0.5: ICMP echo request, id 35, seq 1, length 64
04:22:00.067819 IP 192.168.60.6 > 10.9.0.5: ICMP echo request, id 35, seq 2, length 64
04:22:00.067833 IP 192.168.60.6 > 10.9.0.5: ICMP echo request, id 35, seq 2, length 64
04:22:04.261187 ARP, Request who-has 10.9.0.5 tell 10.9.0.11, length 28
04:22:04.261333 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
04:22:39.742115 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 16, seq 1, length 64
04:22:39.742134 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 16, seq 1, length 64
04:22:40.743800 IP 10.9.0.5 > 10.9.0.11: ICMP echo request, id 16, seq 2, length 64
04:22:40.743823 IP 10.9.0.11 > 10.9.0.5: ICMP echo reply, id 16, seq 2, length 64
04:22:44.965638 ARP, Request who-has 10.9.0.11 tell 10.9.0.5, length 28
04:22:44.965649 ARP, Reply 10.9.0.11 is-at 02:42:0a:09:00:0b, length 28
```

Task 2: Create and Configure TUN Interface

Task 2.a: Name of the Interface

```
root@gyc4885-client-10:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
86: eth0@if87: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state 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
```

Change name:

ifr = struct.pack('16sH', b'guo%d', IFF_TUN | IFF_NO_PI)

```
root@gyc4885-client-10:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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: guo0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
86: eth0@if87: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state 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

After assigning ip address and bring up the interface:

```
root@gyc4885-client-10:/# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
4: guo0: <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 guo0
       valid_lft forever preferred_lft forever
86: eth0@if87: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state 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
```

The state id not DOWN, and an ip address is assigned.

Task 2.c: Read from the TUN Interface

Ping 192.168.53.1 from host U:

```
root@gyc4885-client-10:/volumes# tun.py
Interface Name: guo0
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
```

The message from Host U is sent to 192.168.53.1.

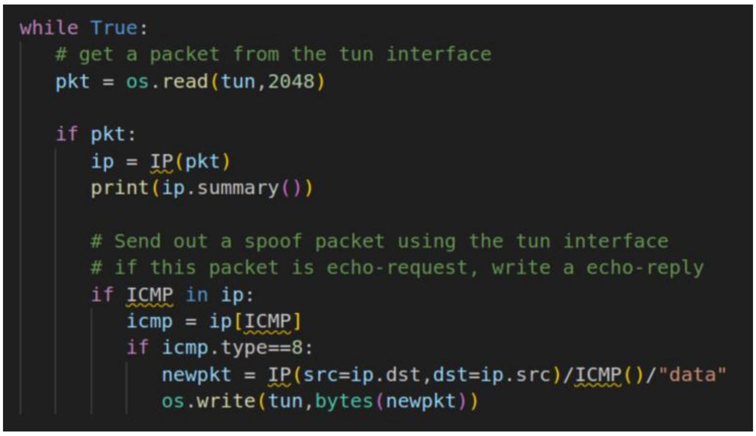
Ping 192.168.60.5:

```
root@gyc4885-client-10:/# 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
04:52:30.211714 IP 10.9.0.5 > 192.168.60.5: ICMP echo request, id 59, seq 123, length 64
04:52:31.243732 IP 10.9.0.5 > 192.168.60.5: ICMP echo request, id 59, seq 124, length 64
04:52:32.268777 IP 10.9.0.5 > 192.168.60.5: ICMP echo request, id 59, seq 125, length 64
```

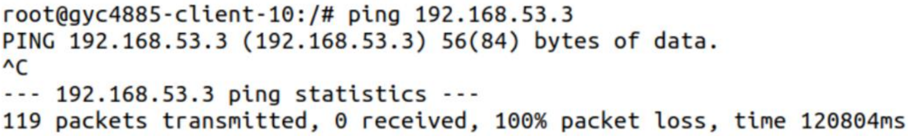
tun.py does not print out anything? Because any network outside of the host network is unreachable since the tunnel is not established on the other side.

Task 2.d: Write to the TUN Interface

• After getting a packet from the TUN interface, if this packet is an ICMP echo request packet, construct a corresponding echo reply packet and write it to the TUN interface.
The code is as follows:

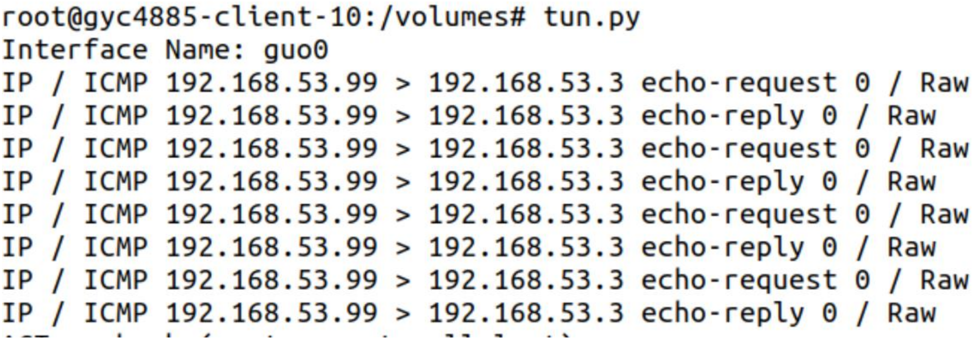


```
while True:
    # get a packet from the tun interface
    pkt = os.read(tun,2048)

    if pkt:
        ip = IP(pkt)
        print(ip.summary())

        # Send out a spoof packet using the tun interface
        # if this packet is echo-request, write a echo-reply
        if ICMP in ip:
            icmp = ip[ICMP]
            if icmp.type==8:
                newpkt = IP(src=ip.dst,dst=ip.src)/ICMP()/"data"
                os.write(tun,bytes(newpkt))
```



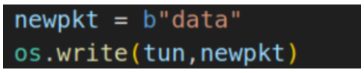
```
root@gyc4885-client-10:/# ping 192.168.53.3
PING 192.168.53.3 (192.168.53.3) 56(84) bytes of data.
^C
--- 192.168.53.3 ping statistics ---
119 packets transmitted, 0 received, 100% packet loss, time 120804ms
```

tun received the packet, and replied.



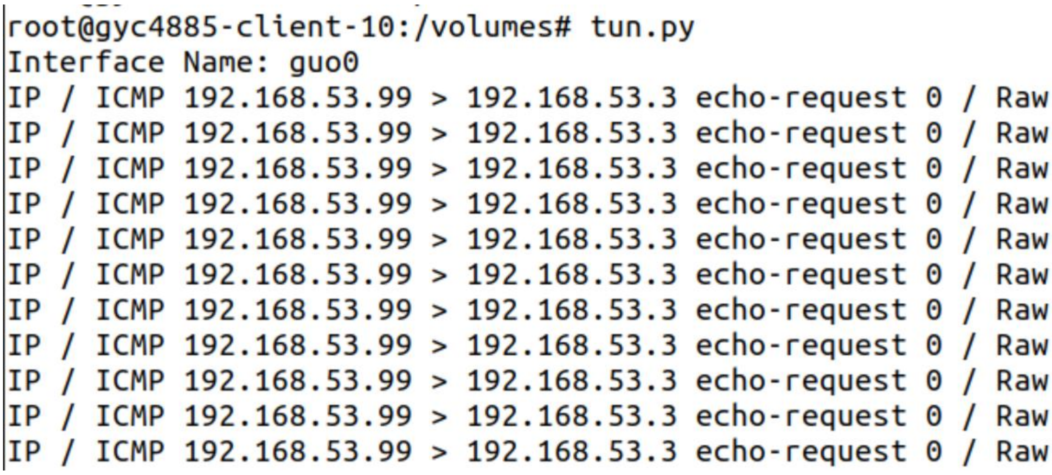
```
root@gyc4885-client-10:/volumes# tun.py
Interface Name: guo0
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-reply 0 / Raw
```

- Instead of writing an IP packet to the interface, write some arbitrary data to the interface:



```
newpkt = b"data"
os.write(tun,newpkt)
```

Tun received the packet, but does not reply correct content.



```
root@gyc4885-client-10:/volumes# tun.py
Interface Name: guo0
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.53.3 echo-request 0 / Raw
```

```
root@gyc4885-client-10:/volumes# tcpdump -i guo0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on guo0, link-type RAW (Raw IP), capture size 262144 bytes
12:25:09.512493 IP 192.168.53.99 > 192.168.53.3: ICMP echo request, id 645, seq 236, length 64
12:25:09.513175 [|ip6]
12:25:10.537720 IP 192.168.53.99 > 192.168.53.3: ICMP echo request, id 645, seq 237, length 64
12:25:10.539235 [|ip6]
12:25:11.561018 IP 192.168.53.99 > 192.168.53.3: ICMP echo request, id 645, seq 238, length 64
12:25:11.561509 [|ip6]
12:25:12.585266 IP 192.168.53.99 > 192.168.53.3: ICMP echo request, id 645, seq 239, length 64
12:25:12.586205 [|ip6]
```

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

Ping 192.168.53.3 from host U:

```
root@gyc4885-server-router:/volumes# tun_server.py
10.9.0.5:46870 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:46870 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:46870 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
```

Original ip source address is U's ip address, but the new ip packet's src ip is 192.168.53.99

Ping host V:

ICMP packet is NOT sent to VPN Server through the tunnel. Add ip route as follows:

```
root@gyc4885-client-10:/volumes# ip route add 192.168.60.0/24 via 192.168.53.3 dev guo0
```

Then when ping an IP address in the 192.168.60.0/24 network, the ICMP packets are received by tun_server.py through the tunnel.

```
root@gyc4885-client-10:/volumes# tun_client.py
Interface Name: guo0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
```

```
root@gyc4885-server-router:/volumes# tun_server.py
10.9.0.5:43541 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:43541 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:43541 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:43541 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:43541 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
```

Task 4: Set Up the VPN Server

tun_client.py:

```
# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'guo%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 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

os.system("ip route add 192.168.60.0/24 dev guo0 via 192.168.53.3")

# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while True:
    # Get a packet from the tun interface
    packet = os.read(tun, 2048)
    ip = IP(packet)
    print(ip.summary())
    if packet:
        # Send the packet via the tunnel
        sock.sendto(packet, ("10.9.0.11", 9090))
```

tun_server.py:

```python
# Create a TUN interface and configure it.
TUNSETIFF = 0x400454ca
IFF_TUN   = 0x0001
IFF_TAP   = 0x0002
IFF_NO_PI = 0x1000

tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'guo%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 192.168.66.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))

while True:
    data, (ip, port) = sock.recvfrom(2048)
    pkt = IP(data)
    print(pkt.summary())

    # Write the packet to the TUN interface.
    os.write(tun,bytes(pkt))
```

Ping Host V from Host U:

```
root@gyc4885-client-10:/# 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 ---
3 packets transmitted, 0 received, 100% packet
 loss, time 2036ms
```

```
root@gyc4885-client-10:/volumes# tun_client.py
Interface Name: guo0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
```

```
root@gyc4885-server-router:/volumes# tun_server.py
Interface Name: guo0
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
IP / ICMP 192.168.53.99 > 192.168.60.5 echo-request 0 / Raw
```

The ICMP echo request packets arrive at Host V through the tunnel.

```
root@gyc4885-host-5:/# tcpdump -i any -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
11:35:23.567001 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 906, seq 5, length 64
11:35:23.567044 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 906, seq 5, length 64
11:35:24.590585 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
11:35:24.590648 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
11:35:24.590656 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
11:35:24.590659 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
11:35:24.591884 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 906, seq 6, length 64
11:35:24.591894 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 906, seq 6, length 64
11:35:25.616675 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 906, seq 7, length 64
11:35:25.616697 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 906, seq 7, length 64
```

Task 5: Handling Traffic in Both Directions

tun_server.py:

```python
while True:
    # this will block until at least one interface is ready
    ready, _, _ = select.select([sock, tun], [], [])

    for fd in ready:
        if fd is sock:
            data, (ip, port) = sock.recvfrom(2048)
            pkt = IP(data)
            print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))

            # Write the packet to the TUN interface.
            os.write(tun,bytes(pkt))

        if fd is tun:
            packet = os.read(tun, 2048)
            pkt = IP(packet)
            print("From tun    ==>: {} --> {}".format(pkt.src, pkt.dst))

            # Send the packet via the tunnel
            sock.sendto(packet, ("10.9.0.5",1234)) #port
```

tun_client.py:

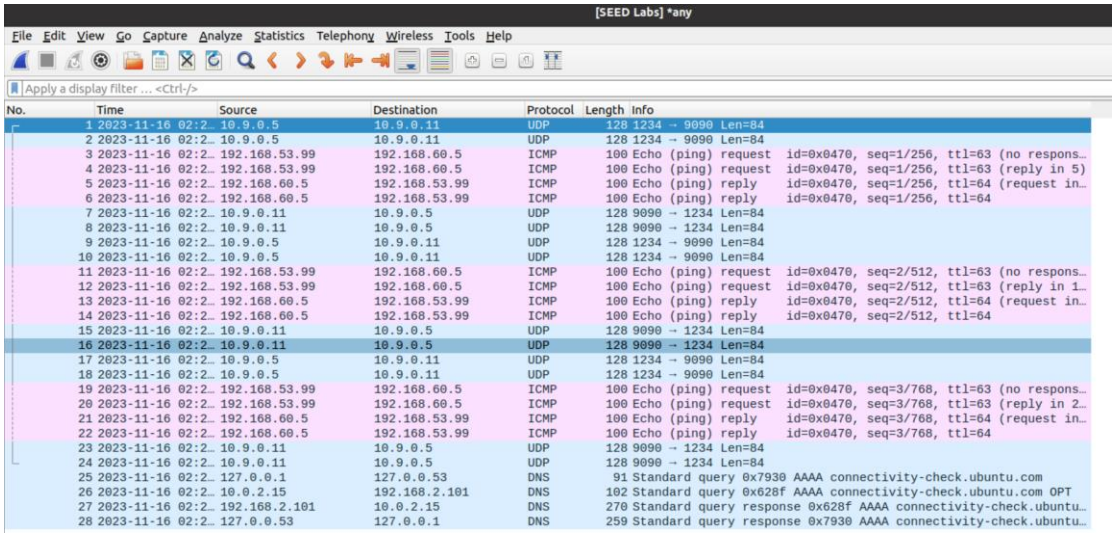
```python
# Create UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(('0.0.0.0',1234))

while True:
    # this will block until at least one interface is ready
    ready, _, _ = select.select([sock, tun], [], [])

    for fd in ready:
        if fd is sock:
            data, (ip, port) = sock.recvfrom(2048)
            pkt = IP(data)
            print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))

            # Write the packet to the TUN interface.
            os.write(tun,bytes(pkt))

        if fd is tun:
            packet = os.read(tun, 2048)
            pkt = IP(packet)
            print("From tun    ==>: {} --> {}".format(pkt.src, pkt.dst))

            # Send the packet via the tunnel
            sock.sendto(packet, ("10.9.0.11", 9090))
```

Ping Host V from Host U: both sides can send and receive packets.

```
root@gyc4885-client-10:/# 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=11.4 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=4.91 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=4.34 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=5.15 ms
^C
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 4.336/6.437/11.355/2.854 ms
```

```
root@gyc4885-server-router:/volumes# tun_server.py
Interface Name: guo0
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun    ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun    ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun    ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun    ==>: 192.168.60.5 --> 192.168.53.99
```

```
root@gyc4885-client-10:/volumes# tun_client.py
Interface Name: guo0
From tun    ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun    ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun    ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
From tun    ==>: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.60.5 --> 192.168.53.99
```



No.	Time	Source	Destination	Protocol	Length	Info
1	2023-11-16 02:2…	10.9.0.5	10.9.0.11	UDP	128	1234 → 9090 Len=84
2	2023-11-16 02:2…	10.9.0.5	10.9.0.11	UDP	128	1234 → 9090 Len=84
3	2023-11-16 02:2…	192.168.53.99	192.168.60.5	ICMP	100	Echo (ping) request id=0x0470, seq=1/256, ttl=63 (no respons…
4	2023-11-16 02:2…	192.168.53.99	192.168.60.5	ICMP	100	Echo (ping) request id=0x0470, seq=1/256, ttl=63 (reply in 5)
5	2023-11-16 02:2…	192.168.60.5	192.168.53.99	ICMP	100	Echo (ping) reply id=0x0470, seq=1/256, ttl=64 (request in…
6	2023-11-16 02:2…	192.168.60.5	192.168.53.99	ICMP	100	Echo (ping) reply id=0x0470, seq=1/256, ttl=64
7	2023-11-16 02:2…	10.9.0.11	10.9.0.5	UDP	128	9090 → 1234 Len=84
8	2023-11-16 02:2…	10.9.0.11	10.9.0.5	UDP	128	9090 → 1234 Len=84
9	2023-11-16 02:2…	10.9.0.5	10.9.0.11	UDP	128	1234 → 9090 Len=84
10	2023-11-16 02:2…	10.9.0.5	10.9.0.11	UDP	128	1234 → 9090 Len=84
11	2023-11-16 02:2…	192.168.53.99	192.168.60.5	ICMP	100	Echo (ping) request id=0x0470, seq=2/512, ttl=63 (no respons…
12	2023-11-16 02:2…	192.168.53.99	192.168.60.5	ICMP	100	Echo (ping) request id=0x0470, seq=2/512, ttl=63 (reply in 1…
13	2023-11-16 02:2…	192.168.60.5	192.168.53.99	ICMP	100	Echo (ping) reply id=0x0470, seq=2/512, ttl=64 (request in…
14	2023-11-16 02:2…	192.168.60.5	192.168.53.99	ICMP	100	Echo (ping) reply id=0x0470, seq=2/512, ttl=64
15	2023-11-16 02:2…	10.9.0.11	10.9.0.5	UDP	128	9090 → 1234 Len=84
16	2023-11-16 02:2…	10.9.0.11	10.9.0.5	UDP	128	9090 → 1234 Len=84
17	2023-11-16 02:2…	10.9.0.5	10.9.0.11	UDP	128	1234 → 9090 Len=84
18	2023-11-16 02:2…	10.9.0.5	10.9.0.11	UDP	128	1234 → 9090 Len=84
19	2023-11-16 02:2…	192.168.53.99	192.168.60.5	ICMP	100	Echo (ping) request id=0x0470, seq=3/768, ttl=63 (no respons…
20	2023-11-16 02:2…	192.168.53.99	192.168.60.5	ICMP	100	Echo (ping) request id=0x0470, seq=3/768, ttl=63 (reply in 2…
21	2023-11-16 02:2…	192.168.60.5	192.168.53.99	ICMP	100	Echo (ping) reply id=0x0470, seq=3/768, ttl=64 (request in…
22	2023-11-16 02:2…	192.168.60.5	192.168.53.99	ICMP	100	Echo (ping) reply id=0x0470, seq=3/768, ttl=64
23	2023-11-16 02:2…	10.9.0.11	10.9.0.5	UDP	128	9090 → 1234 Len=84
24	2023-11-16 02:2…	10.9.0.11	10.9.0.5	UDP	128	9090 → 1234 Len=84
25	2023-11-16 02:2…	127.0.0.1	127.0.0.53	DNS	91	Standard query 0x7930 AAAA connectivity-check.ubuntu.com
26	2023-11-16 02:2…	10.0.2.15	192.168.2.101	DNS	102	Standard query 0x628f AAAA connectivity-check.ubuntu.com OPT
27	2023-11-16 02:2…	192.168.2.101	10.0.2.15	DNS	270	Standard query response 0x628f AAAA connectivity-check.ubuntu…
28	2023-11-16 02:2…	127.0.0.53	127.0.0.1	DNS	259	Standard query response 0x7930 AAAA connectivity-check.ubuntu…

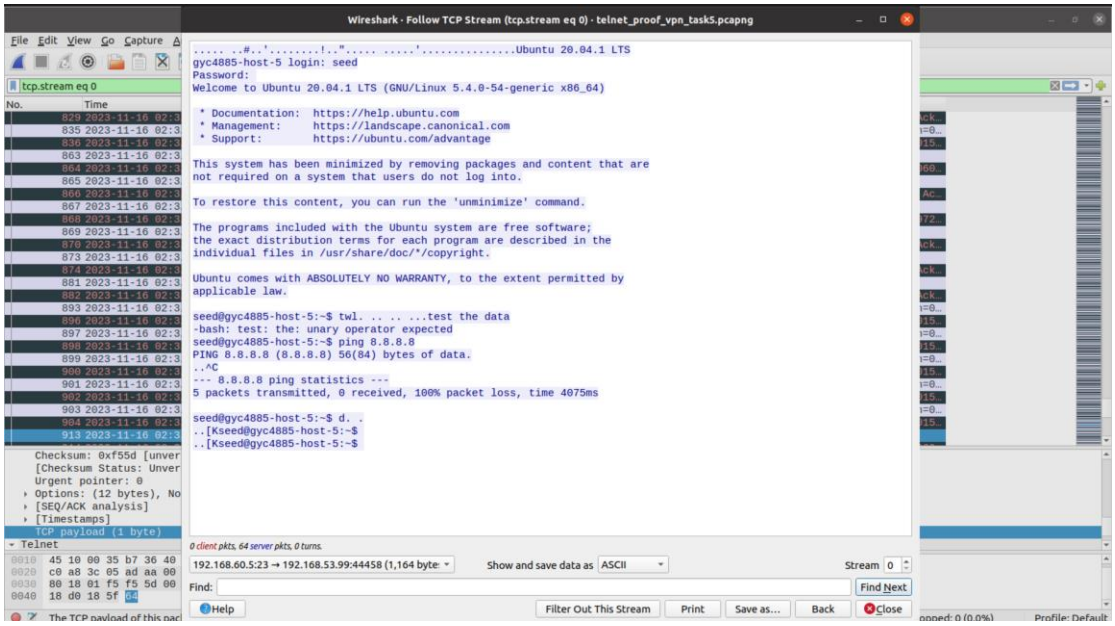
- Host U sends ping request to VPN server by tun through tunnel
- VPN server sends ping request to Host V
- Host V sends echo reply to VPN server
- VPN server sends echo reply to Host U by tun through tunnel

telnet:

```
root@gyc4885-client-10:/# telnet 192.168.60.5
Trying 192.168.60.5...
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
gyc4885-host-5 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@gyc4885-host-5:~$ test the data
-bash: test: the: unary operator expected
seed@gyc4885-host-5:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4075ms
```

on client:

```
07:35:29.875403 IP 192.168.53.99.44458 > 192.168.60.5.23: Flags [.], ack 975, win 501, options [nop,nop,TS val 1349541776 ecr 416286411], len
gth 0
07:35:30.868996 IP 192.168.60.5 > 8.8.8.8: ICMP echo request, id 67, seq 2, length 64
07:35:31.892956 IP 192.168.60.5 > 8.8.8.8: ICMP echo request, id 67, seq 3, length 64
07:35:32.916731 IP 192.168.60.5 > 8.8.8.8: ICMP echo request, id 67, seq 4, length 64
07:35:33.941126 IP 192.168.60.5 > 8.8.8.8: ICMP echo request, id 67, seq 5, length 64
07:35:34.358154 IP 192.168.53.99.44458 > 192.168.60.5.23: Flags [P.], seq 135:136, ack 975, win 501, options [nop,nop,TS val 1349546259 ecr 4
16286411], length 1
07:35:34.359035 IP 192.168.60.5.23 > 192.168.53.99.44458: Flags [P.U], seq 975:976, ack 136, win 509, urg 1, options [nop,nop,TS val 41629090
5 ecr 1349546259], length 1
```

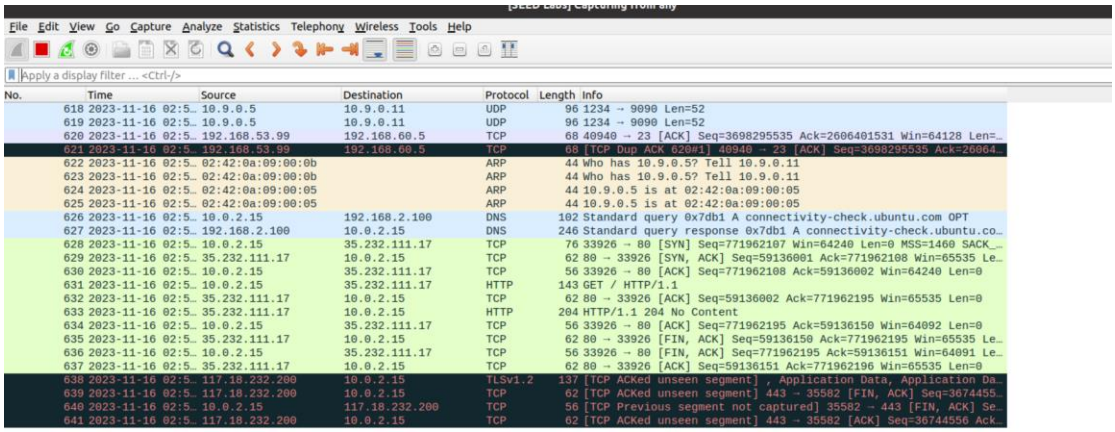


Data flow: Host U -> tun_client -> tun_server -> VPN server -> Host V

Vice versa for packets from host v to host u.

Task 6: Tunnel-Breaking Experiment

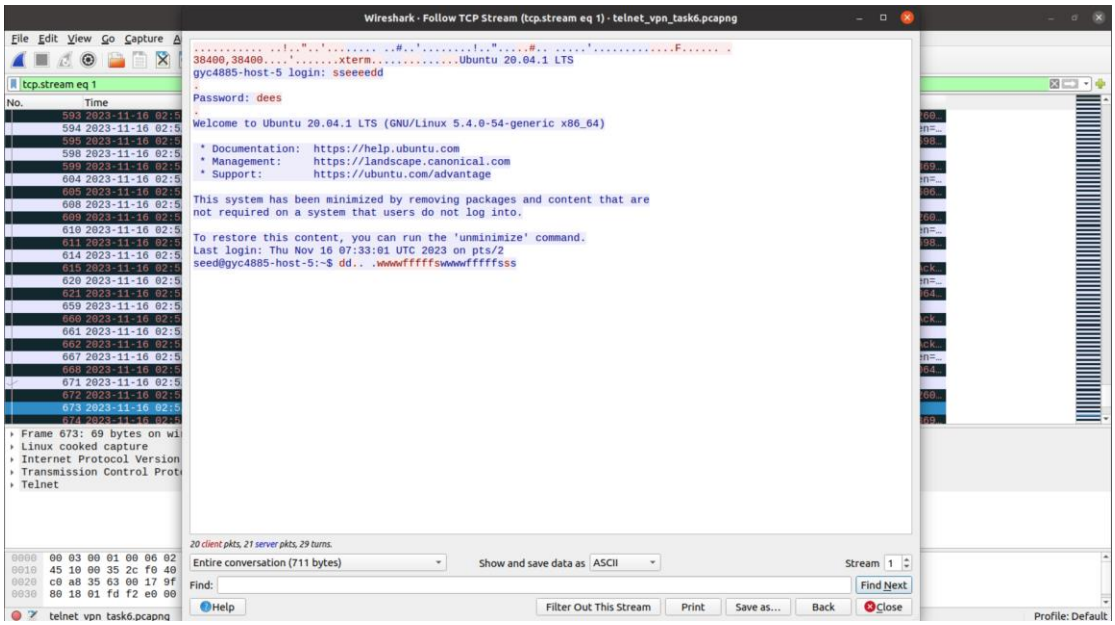
Break the VPN tunnel by terminating tun client and server programmes.. We are not able to type anything because the connection was broken.

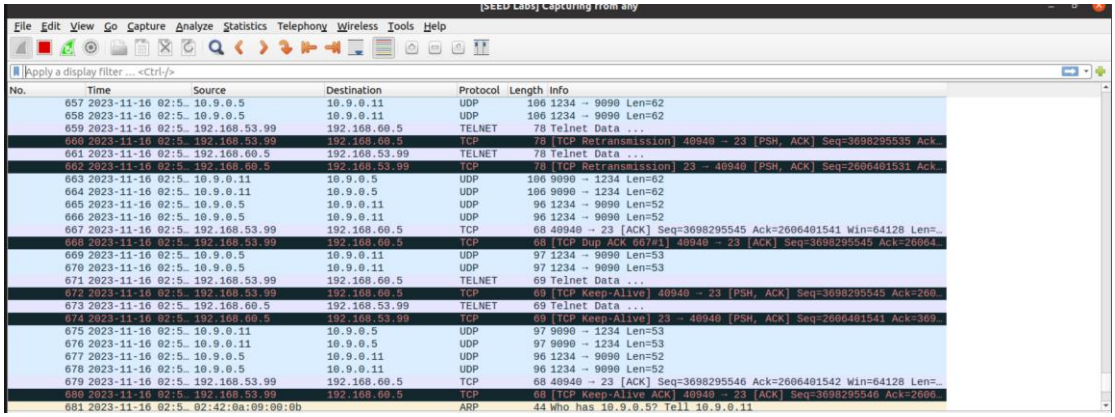


No.	Time	Source	Destination	Protocol	Length	Info
618	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	96	1234 → 9090 Len=52
619	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	96	1234 → 9090 Len=52
620	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	68	40940 → 23 [ACK] Seq=3698295535 Ack=2606401531 Win=64128 Len=…
621	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	68	[TCP Dup ACK 620#1] 40940 → 23 [ACK] Seq=3698295535 Ack=26064…
622	2023-11-16 02:5…	02:42:0a:09:00:0b		ARP	44	Who has 10.9.0.5? Tell 10.9.0.11
623	2023-11-16 02:5…	02:42:0a:09:00:0b		ARP	44	Who has 10.9.0.5? Tell 10.9.0.11
624	2023-11-16 02:5…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
625	2023-11-16 02:5…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
626	2023-11-16 02:5…	10.0.2.15	192.168.2.100	DNS	102	Standard query 0x7db1 A connectivity-check.ubuntu.com OPT
627	2023-11-16 02:5…	192.168.2.100	10.0.2.15	DNS	246	Standard query response 0x7db1 A connectivity-check.ubuntu.co…
628	2023-11-16 02:5…	10.0.2.15	35.232.111.17	TCP	76	33926 → 80 [SYN] Seq=771962107 Win=64240 Len=0 MSS=1460 SACK_…
629	2023-11-16 02:5…	35.232.111.17	10.0.2.15	TCP	62	80 → 33926 [SYN, ACK] Seq=59136001 Ack=771962108 Win=65535 Le…
630	2023-11-16 02:5…	10.0.2.15	35.232.111.17	TCP	56	33926 → 80 [ACK] Seq=771962108 Ack=59136002 Win=64240 Len=0
631	2023-11-16 02:5…	10.0.2.15	35.232.111.17	HTTP	143	GET / HTTP/1.1
632	2023-11-16 02:5…	35.232.111.17	10.0.2.15	TCP	62	80 → 33926 [ACK] Seq=59136002 Ack=771962195 Win=65535 Len=0
633	2023-11-16 02:5…	35.232.111.17	10.0.2.15	HTTP	204	HTTP/1.1 204 No Content
634	2023-11-16 02:5…	10.0.2.15	35.232.111.17	TCP	56	33926 → 80 [ACK] Seq=771962195 Ack=59136150 Win=64092 Len=0
635	2023-11-16 02:5…	35.232.111.17	10.0.2.15	TCP	62	80 → 33926 [FIN, ACK] Seq=59136150 Ack=771962195 Win=65535 Le…
636	2023-11-16 02:5…	10.0.2.15	35.232.111.17	TCP	56	33926 → 80 [FIN, ACK] Seq=771962195 Ack=59136151 Win=64091 Le…
637	2023-11-16 02:5…	35.232.111.17	10.0.2.15	TCP	62	80 → 33926 [ACK] Seq=59136151 Ack=771962196 Win=65535 Len=0
638	2023-11-16 02:5…	117.18.232.200	10.0.2.15	TLSv1.2	137	[TCP ACKed unseen segment] , Application Data, Application Da…
639	2023-11-16 02:5…	117.18.232.200	10.0.2.15	TCP	62	[TCP ACKed unseen segment] 443 → 35582 [FIN, ACK] Seq=3674455…
640	2023-11-16 02:5…	10.0.2.15	117.18.232.200	TCP	56	[TCP Previous segment not captured] 35582 → 443 [FIN, ACK] Se…
641	2023-11-16 02:5…	117.18.232.200	10.0.2.15	TCP	62	[TCP ACKed unseen segment] 443 → 35582 [ACK] Seq=36744556 Ack…

Reconnect the VPN tunnel, everything typed just now (which was not able to be seen) suddenly shows up, and the connection is re-established.

```
Connected to 192.168.60.5.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
gyc4885-host-5 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 a
re
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Nov 16 07:33:01 UTC 2023 on pts/2
seed@gyc4885-host-5:~$ wwwwfffffss
```





No.	Time	Source	Destination	Protocol	Length	Info
657	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	106	1234 → 9090 Len=62
658	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	106	1234 → 9090 Len=62
659	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TELNET	78	Telnet Data ...
660	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	78	[TCP Retransmission] 40940 → 23 [PSH, ACK] Seq=3698295535 Ack…
661	2023-11-16 02:5…	192.168.60.5	192.168.53.99	TELNET	78	Telnet Data ...
662	2023-11-16 02:5…	192.168.60.5	192.168.53.99	TCP	78	[TCP Retransmission] 23 → 40940 [PSH, ACK] Seq=2606401531 Ack…
663	2023-11-16 02:5…	10.9.0.11	10.9.0.5	UDP	106	9090 → 1234 Len=62
664	2023-11-16 02:5…	10.9.0.11	10.9.0.5	UDP	106	9090 → 1234 Len=62
665	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	96	1234 → 9090 Len=52
666	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	96	1234 → 9090 Len=52
667	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	68	40940 → 23 [ACK] Seq=3698295545 Ack=2606401541 Win=64128 Len=…
668	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	68	[TCP Dup ACK 667#1] 40940 → 23 [ACK] Seq=3698295545 Ack=26064…
669	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	97	1234 → 9090 Len=53
670	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	97	1234 → 9090 Len=53
671	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TELNET	69	Telnet Data ...
672	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	69	[TCP Keep-Alive] 40940 → 23 [PSH, ACK] Seq=3698295545 Ack=260…
673	2023-11-16 02:5…	192.168.60.5	192.168.53.99	TELNET	69	Telnet Data ...
674	2023-11-16 02:5…	192.168.60.5	192.168.53.99	TCP	69	[TCP Keep-Alive] 23 → 40940 [PSH, ACK] Seq=2606401541 Ack=369…
675	2023-11-16 02:5…	10.9.0.11	10.9.0.5	UDP	97	9090 → 1234 Len=53
676	2023-11-16 02:5…	10.9.0.11	10.9.0.5	UDP	97	9090 → 1234 Len=53
677	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	96	1234 → 9090 Len=52
678	2023-11-16 02:5…	10.9.0.5	10.9.0.11	UDP	96	1234 → 9090 Len=52
679	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	68	40940 → 23 [ACK] Seq=3698295546 Ack=2606401542 Win=64128 Len=…
680	2023-11-16 02:5…	192.168.53.99	192.168.60.5	TCP	68	[TCP Keep-Alive ACK] 40940 → 23 [ACK] Seq=3698295546 Ack=2606…
681	2023-11-16 02:5…	02:42:0a:09:00:0b		ARP	44	Who has 10.9.0.5? Tell 10.9.0.11