

#### Exercise 4)

```
ubuntu@ubuntu:~/CWM-ProgNets/assignment4$ sudo python3 send.py 3 enx0c37965f8a24 192.168.10.1 192.168.10.2
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
Sent 3 packets in total
```

I sent 3 packets but wireshark showed that I had 6 packets. This is because the packets were reflected.

No.	Time	Source	Destination	Protocol	Length	Info
11	0.105664842	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
9	0.105115593	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
7	0.045575380	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
5	0.044968815	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
3	0.000895514	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
1	0.000000000	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface enx0c37965f8a24, id 0  
Ethernet II, Src: BizlinkT\_5f:8a:24 (0c:37:96:5f:8a:24), Dst: Raspberr\_84:ad:1a (e4:5f:01:84:ad:1a)  
Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.10.2  
User Datagram Protocol, Src Port: 50000, Dst Port: 1024  
Data (22 bytes)

No.	Time	Source	Destination	Protocol	Length	Info
11	0.105664842	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
9	0.105115593	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
7	0.045575380	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
5	0.044968815	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
3	0.000895514	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
1	0.000000000	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22

Frame 3: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface enx0c37965f8a24, id 0  
Ethernet II, Src: Raspberr\_84:ad:1a (e4:5f:01:84:ad:1a), Dst: BizlinkT\_5f:8a:24 (0c:37:96:5f:8a:24)  
Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.10.2  
User Datagram Protocol, Src Port: 50000, Dst Port: 1024  
Data (22 bytes)

If you look at the MAC addresses of packets that are consecutive to one another, you can see that the source and destination are swapped. This is as expected due to the swap\_mac\_addresses action being the default action.

```
RuntimeCmd: table_add MyIngress.src_mac_drop MyIngress.drop 0c:37:96:5f:8a:24 =>
Adding entry to exact match table MyIngress.src_mac_drop
match key:          EXACT-0c:37:96:5f:8a:24
action:             MyIngress.drop
runtime data:
Entry has been added with handle 0
```

Now when I send 3 packets, wireshark only picks up 3 as well since the drop command is used so there are no longer and packets reflected.

1	0.000000000	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024	Len=22
4	0.085042455	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024	Len=22
7	0.161287397	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024	Len=22

▶ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface enx0c37965f8a24, id 0

▶ Ethernet II, Src: BizlinkT\_5f:8a:24 (0c:37:96:5f:8a:24), Dst: Raspberr\_84:ad:1a (e4:5f:01:84:ad:1a)

▶ Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.10.2

▶ User Datagram Protocol, Src Port: 50000, Dst Port: 1024

▶ Data (22 bytes)

The MAC addresses of the 3 packets have the original source and destination addresses so you can tell that none of them have been swapped.

Link to directory: <https://github.com/VKing15/CWM-ProgNets.git>