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	Protoco 4	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
Г	1 0.000000000	192.168.10.1	192.168.10.2	UDP	64 50000 → 1024 Len=22
▶ Fra	me 1: 64 bytes o	n wire (512 bits)	, 64 bytes captured (51	L2 bits) on	interface enx0c37965f8a24, id 0
▶ Eth	ernet II, Src: B	izlinkT_5f:8a:24	(0c:37:96:5f:8a:24), Ds	st: Raspberr	_84:ad:1a (e4:5f:01:84:ad:1a)
▶ Int	ernet Protocol V	ersion 4, Src: 19	2.168.10.1, Dst: 192.16	8.10.2	
▶ Use	er Datagram Proto	col. Src Port: 50	000, Dst Port: 1024		
	a (22 bytes)		333, 232 13121 2321		
Dat	a (22 byccs)				
No.	Time	Source	Destination	Protoco 4	Length Info
No.	Time 11 0.105664842	Source 192.168.10.1	Destination 192.168.10.2	Protoco 4	Length Info 64 50000 → 1024 Len=22
No.	***************************************				3
No.	11 0.105664842	192.168.10.1	192.168.10.2	UDP	64 50000 → 1024 Len=22
No.	11 0.105664842 9 0.105115593	192.168.10.1 192.168.10.1	192.168.10.2 192.168.10.2	UDP UDP	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22
No.	11 0.105664842 9 0.105115593 7 0.045575380	192.168.10.1 192.168.10.1 192.168.10.1	192.168.10.2 192.168.10.2 192.168.10.2	UDP UDP UDP	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22 64 50000 → 1024 Len=22
No.	11 0.105664842 9 0.105115593 7 0.045575380 5 0.044968815	192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1	192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2	UDP UDP UDP UDP	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22 64 50000 → 1024 Len=22 64 50000 → 1024 Len=22
No.	11 0.105664842 9 0.105115593 7 0.045575380 5 0.044968815 3 0.000895514 1 0.000000000	192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1	192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2	UDP UDP UDP UDP UDP UDP	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22
▶ Fra	11 0.105664842 9 0.105115593 7 0.045575380 5 0.044968815 3 0.000895514 1 0.0000000000	192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 n wire (512 bits)	192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 , 64 bytes captured (51	UDP UDP UDP UDP UDP UDP UDP	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22 interface enx0c37965f8a24, id 0
→ Fra	11 0.105664842 9 0.105115593 7 0.045575380 5 0.044968815 3 0.000895514 1 0.0000000000 time 3: 64 bytes of	192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 n wire (512 bits) aspberr_84:ad:1a	192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 , 64 bytes captured (51 (e4:5f:01:84:ad:1a), DS	UDP UDP UDP UDP UDP UDP UDP UDP L2 bits) on	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22
→ Fra → Eth → Int	11 0.105664842 9 0.105115593 7 0.045575380 5 0.044968815 3 0.000895514 1 0.000000000 time 3: 64 bytes of the service of	192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 n wire (512 bits) aspberr_84:ad:1a ersion 4, Src: 19	192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 , 64 bytes captured (51 (e4:5f:01:84:ad:1a), Ds 2.168.10.1, Dst: 192.16	UDP UDP UDP UDP UDP UDP UDP UDP L2 bits) on	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22 interface enx0c37965f8a24, id 0
Fra Eth Int Use	11 0.105664842 9 0.105115593 7 0.045575380 5 0.044968815 3 0.000895514 1 0.000000000 time 3: 64 bytes of the service of	192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 192.168.10.1 n wire (512 bits) aspberr_84:ad:1a ersion 4, Src: 19	192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 192.168.10.2 , 64 bytes captured (51 (e4:5f:01:84:ad:1a), DS	UDP UDP UDP UDP UDP UDP UDP UDP L2 bits) on	64 50000 → 1024 Len=22 64 50000 → 1024 Len=22 interface enx0c37965f8a24, id 6

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.

Г	1 0.0000	00000 192.168	.10.1 192.1	68.10.2 U	OP 64 5000	00 → 1024 Len=22					
	4 0.0850	42455 192.168	.10.1 192.1	68.10.2 U	OP 64 5000	00 → 1024 Len=22					
	7 0.1612	87397 192.168	.10.1 192.1	68.10.2 U	OP 64 5000	00 → 1024 Len=22					
•	Frame 1: 64 b	ytes on wire (5	12 bits), 64 bytes	captured (512 bit	s) on interface e	nx0c37965f8a24, id 0					
•	Ethernet II,	Src: BizlinkT_5	f:8a:24 (0c:37:96:5	f:8a:24), Dst: Ra	spberr_84:ad:1a (e4:5f:01:84:ad:1a)					
+	Finternet 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