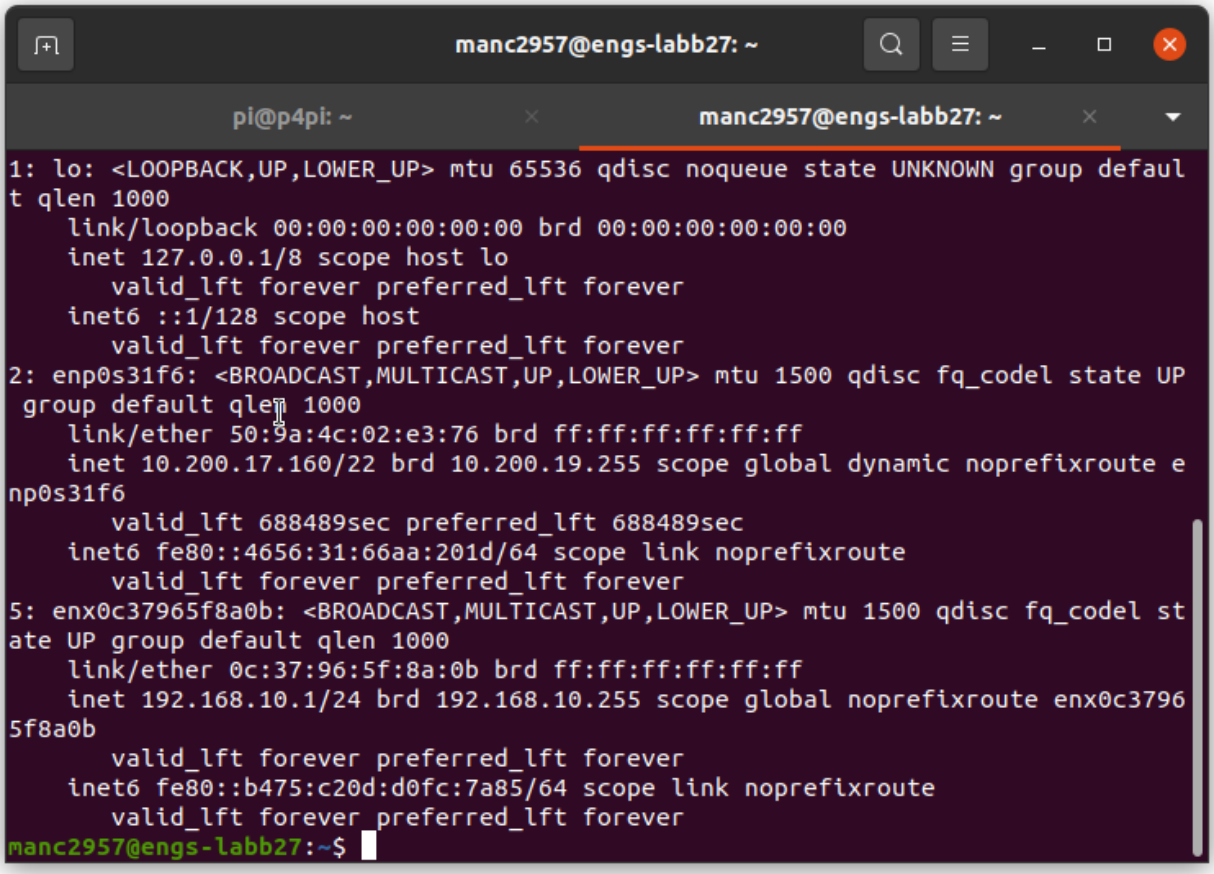
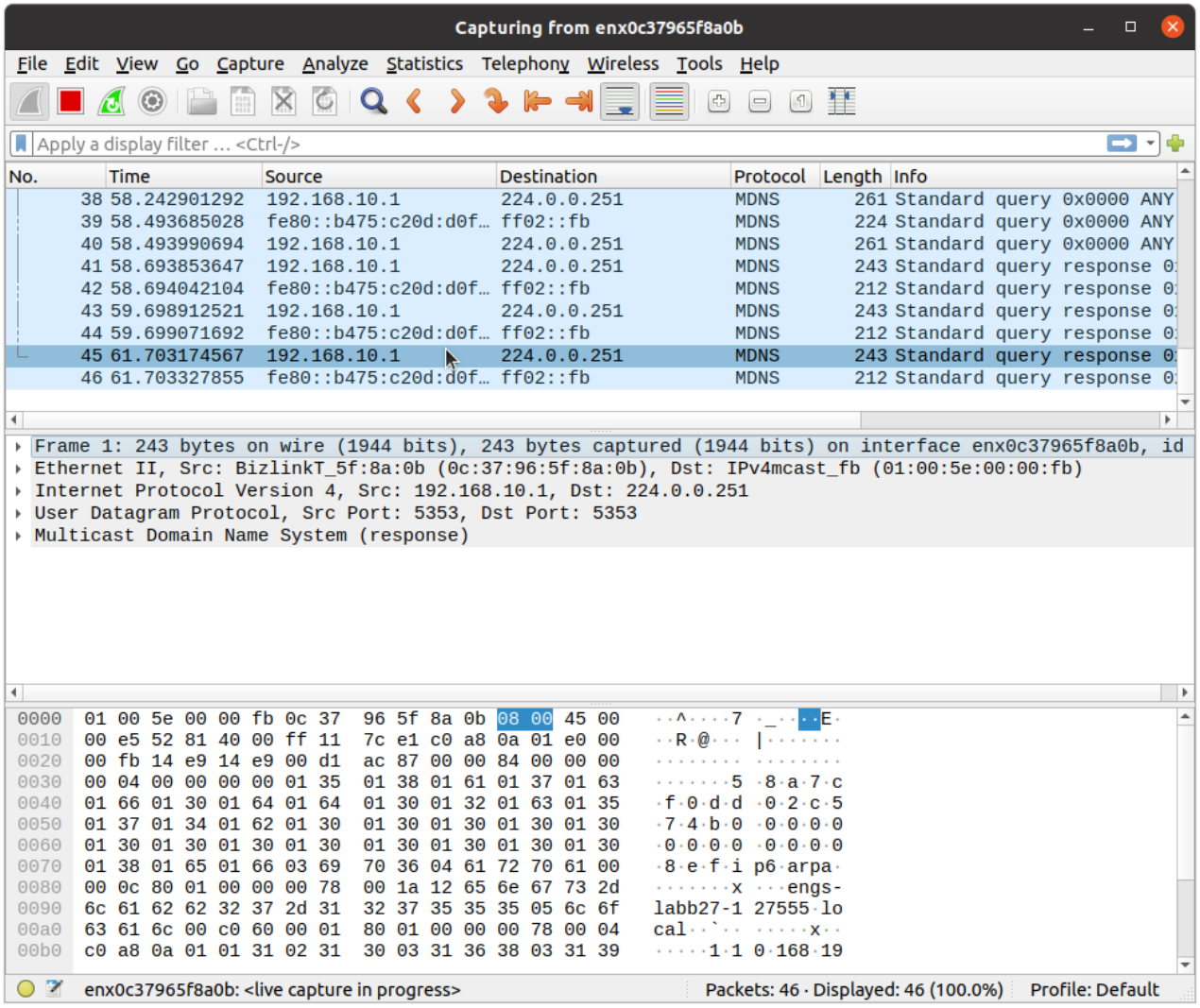
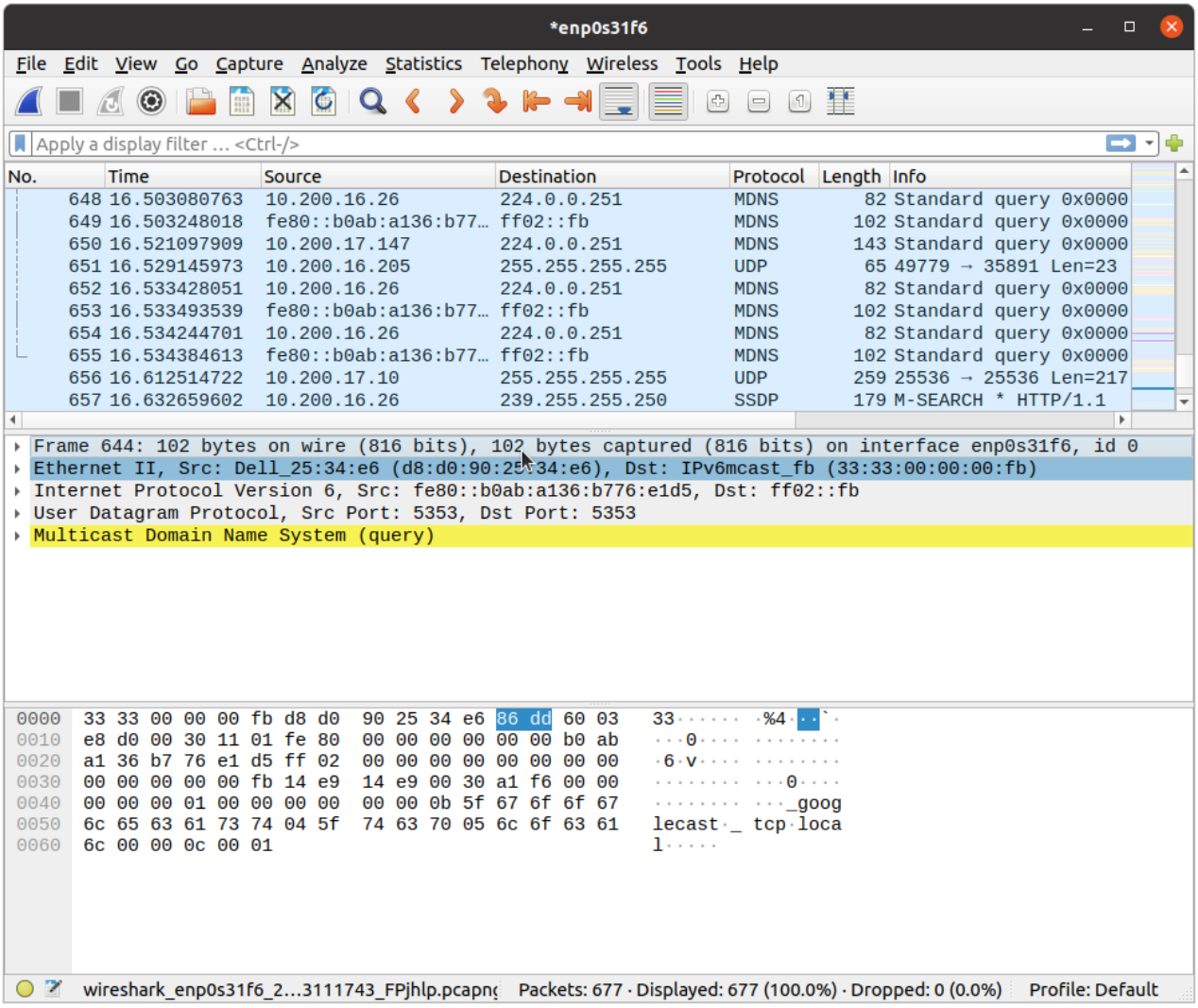


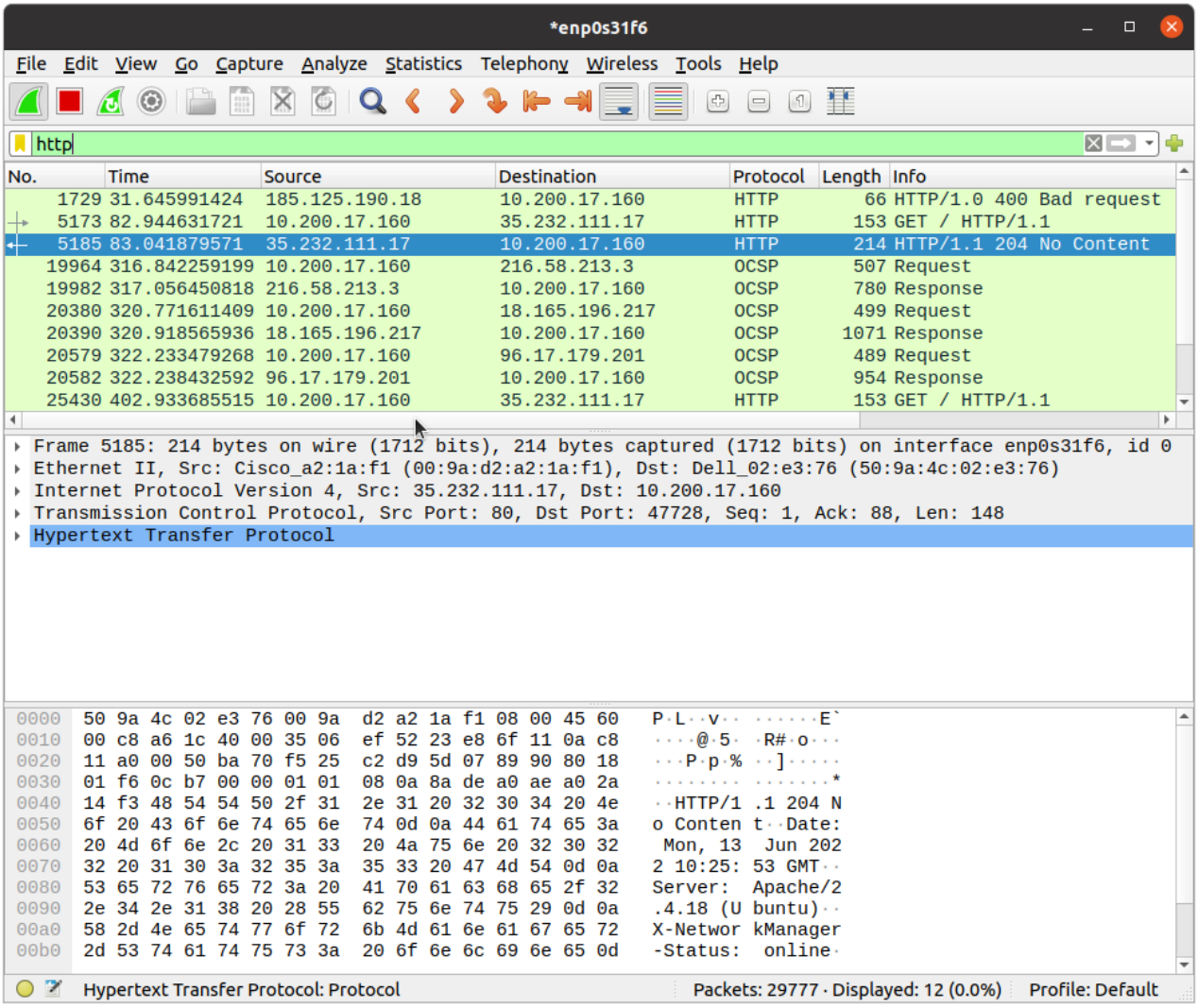
Local machine name is enx0c37965f8a0b.



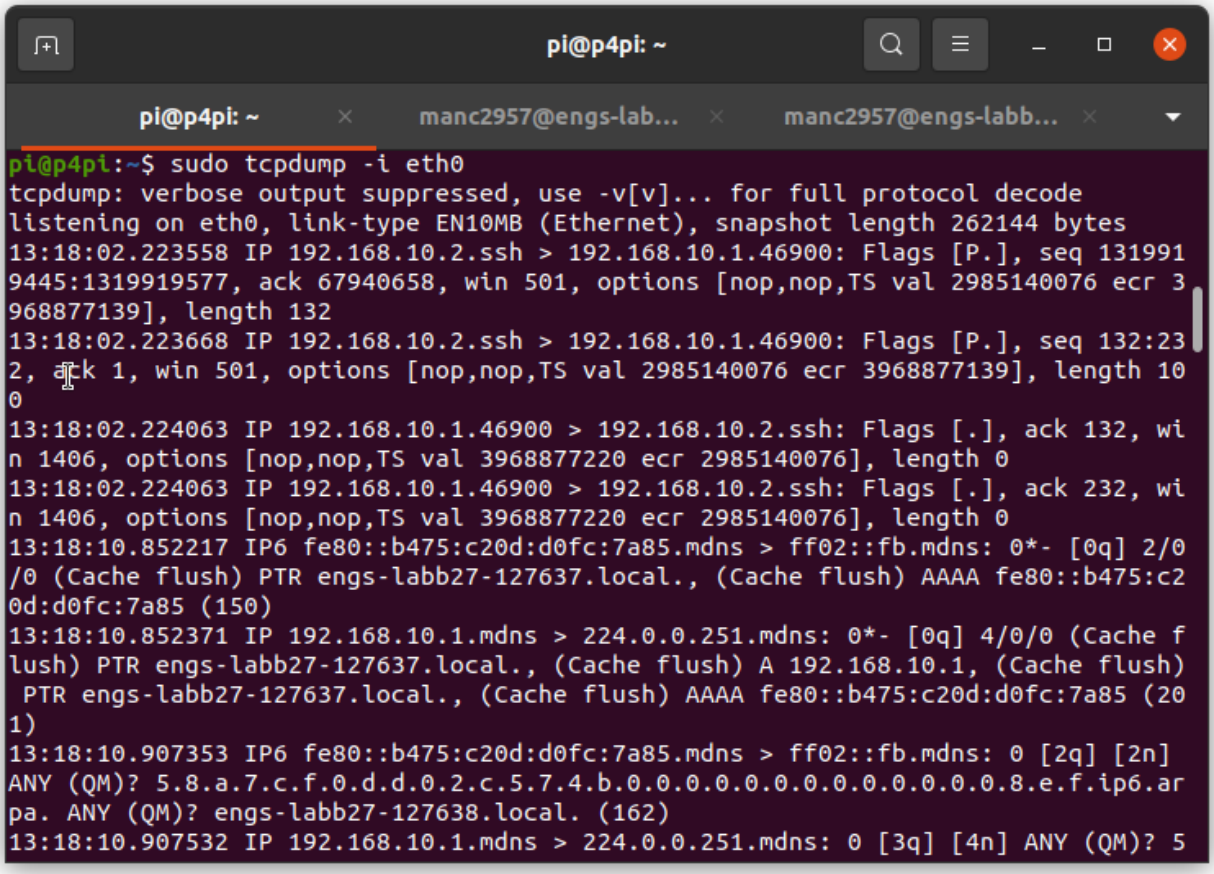
Packets are being captured.
Not quite sure what they are for. It seems to be some form of standard query for small stand alone networks under the multicast DNS protocol.
Packets are quite small (around 2000 bits). Contains the name of my local machine (engs-labb27)



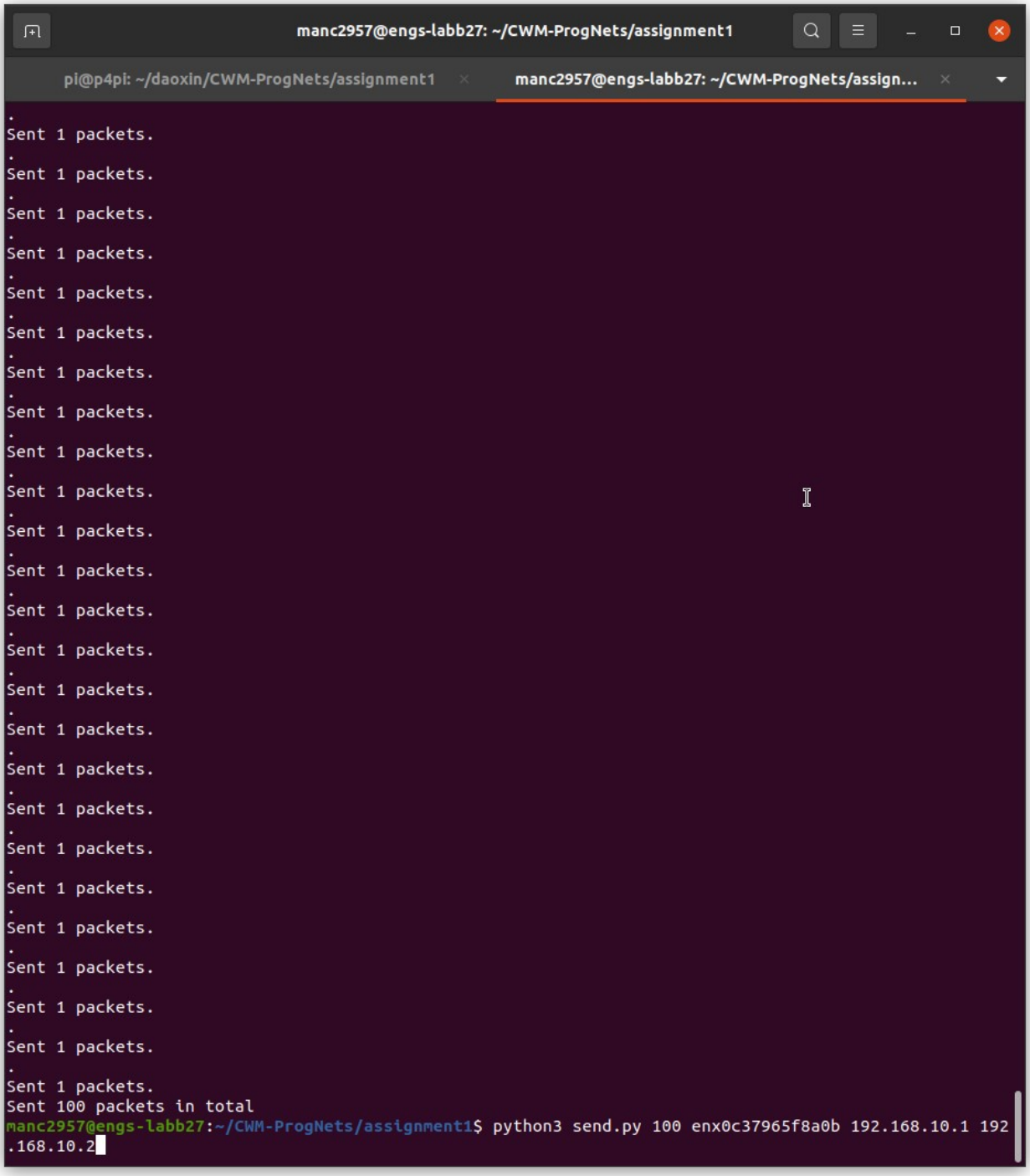
We see a lot more packets being sent within the network.



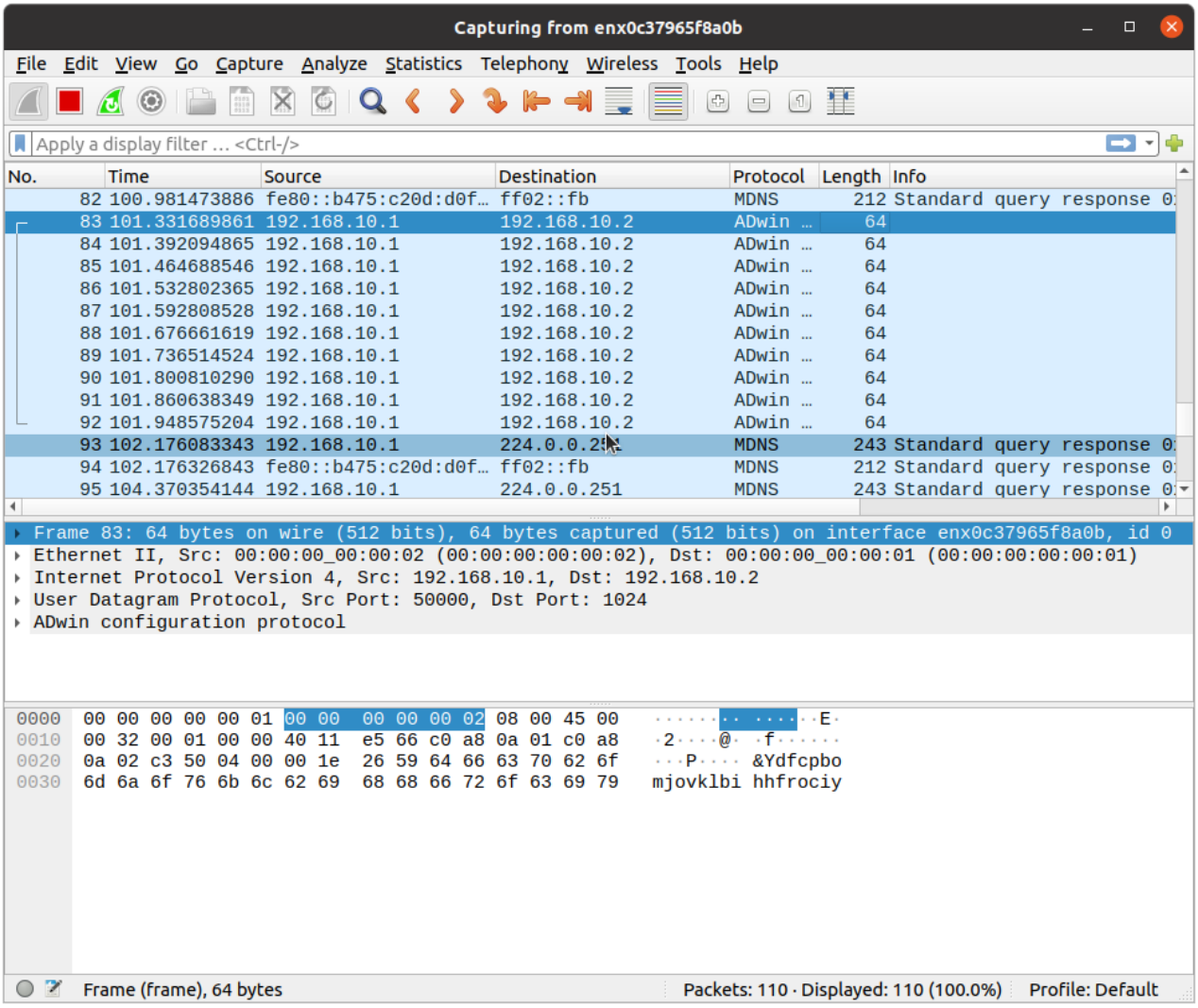
We can apply a display filter to only view the http packages. This will give us the packets sent by HTTP.



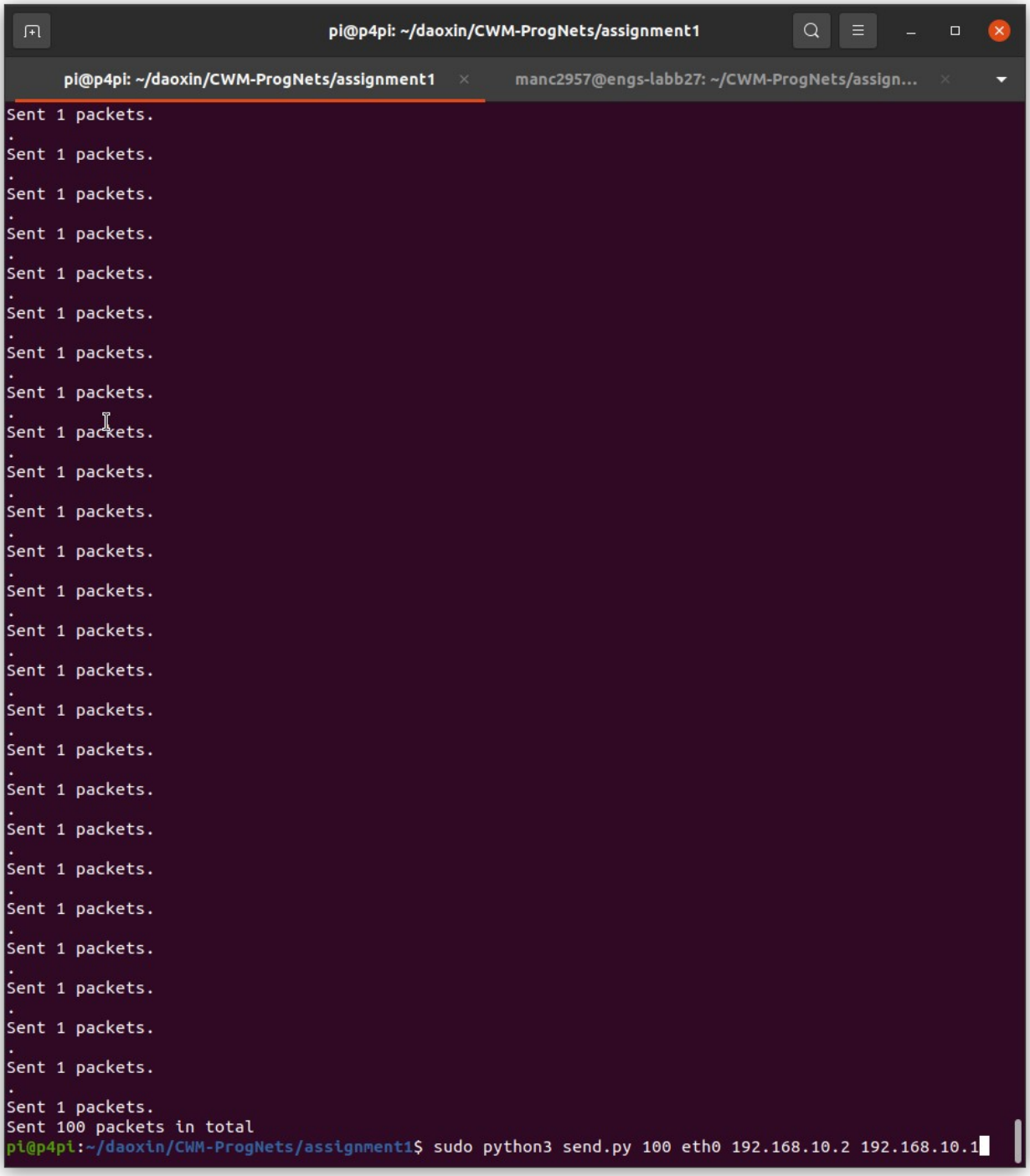
When we run sudo tcpdump -i eth0, we get to see some of the packets.



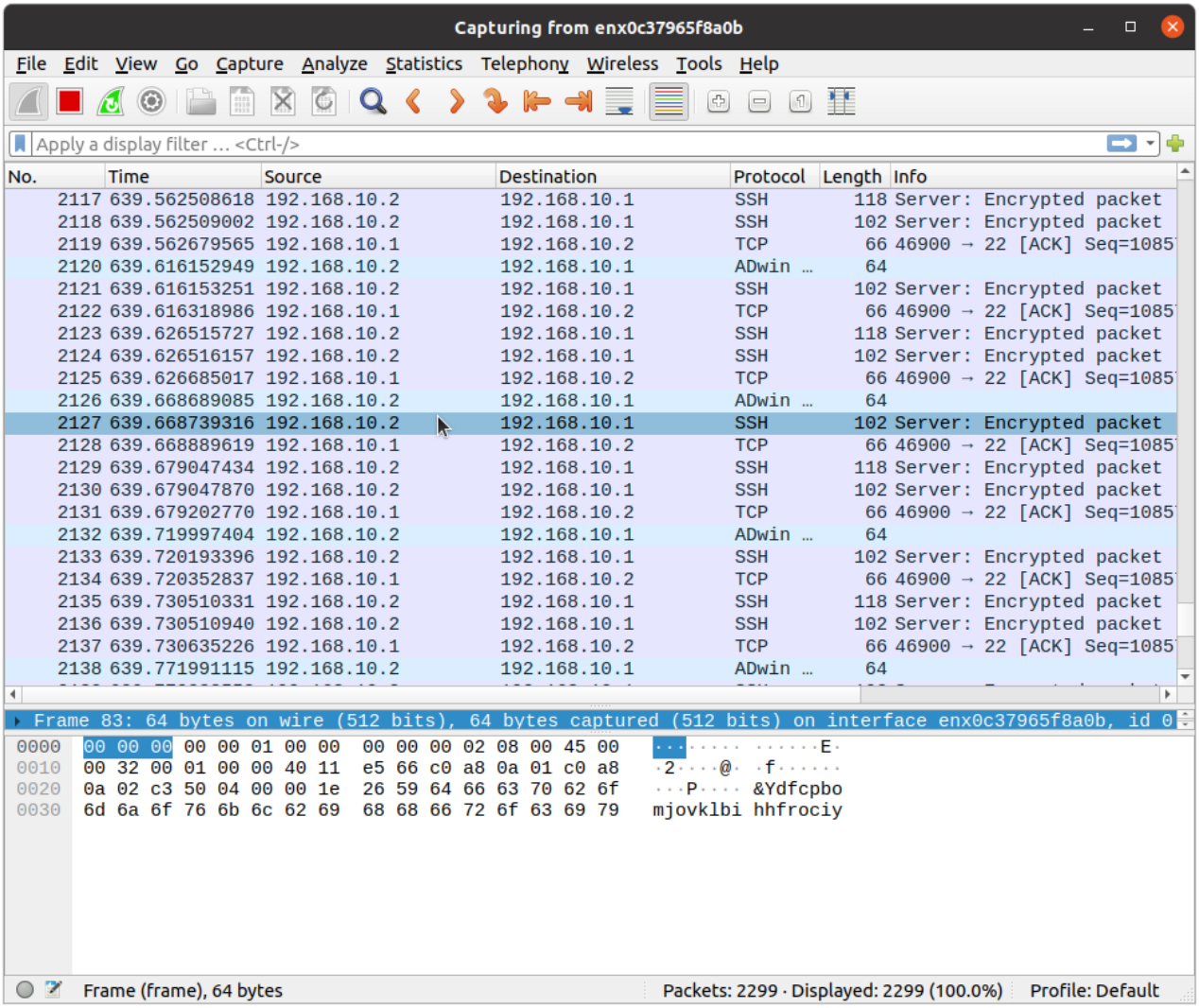
We can run python3 send.py 100 enx0c37965f8a0b 192.168.10.1 192.168.10.2 to our custom packets to the Raspberry PI.



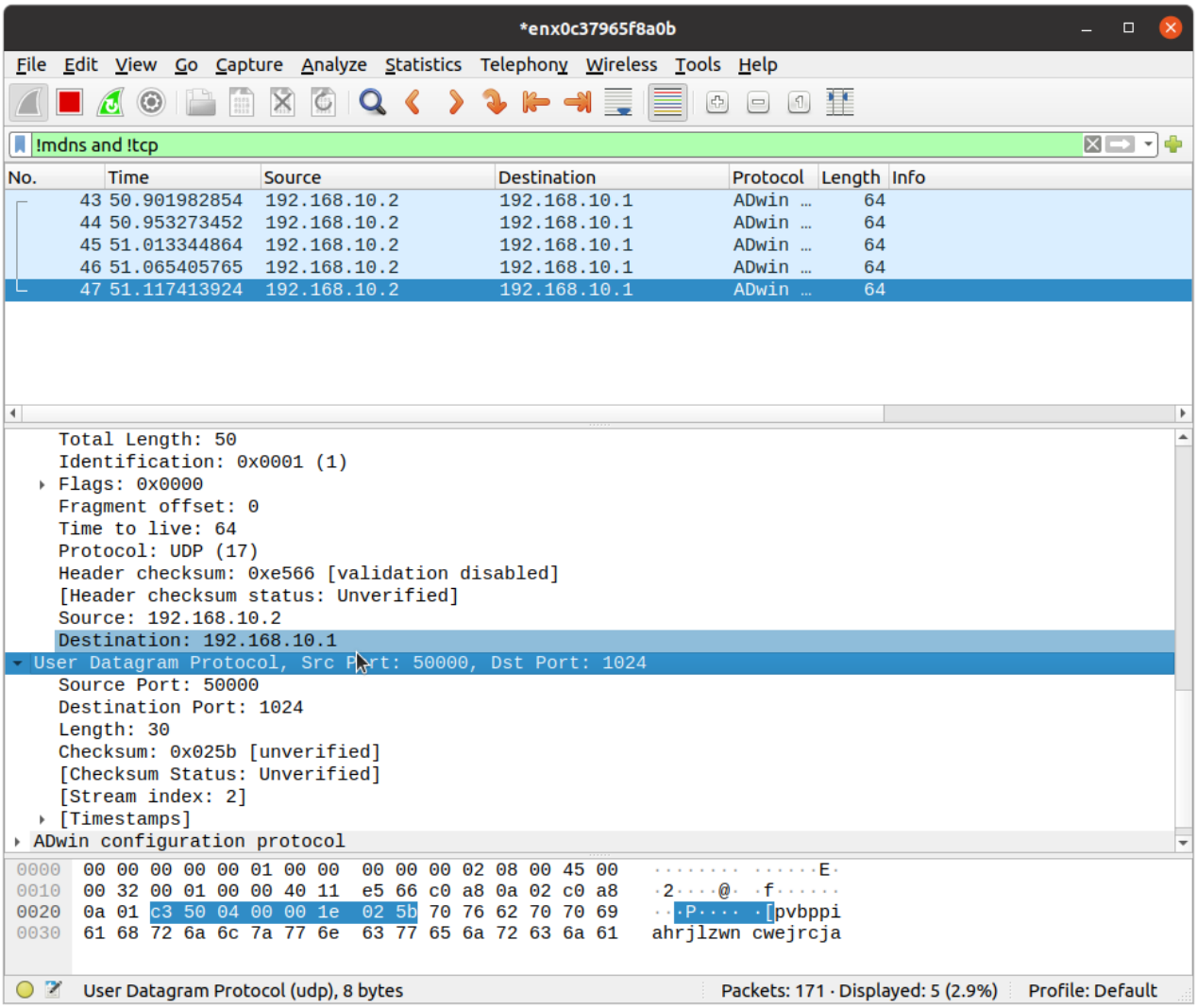
We do see these packages in Wireshark.



If we were to run the same python script in the Raspberry Pi, the packets are now sent to the local machine.



The packets that are being sent across in a format known as Adwin. Each packet has a size of 64 bits.



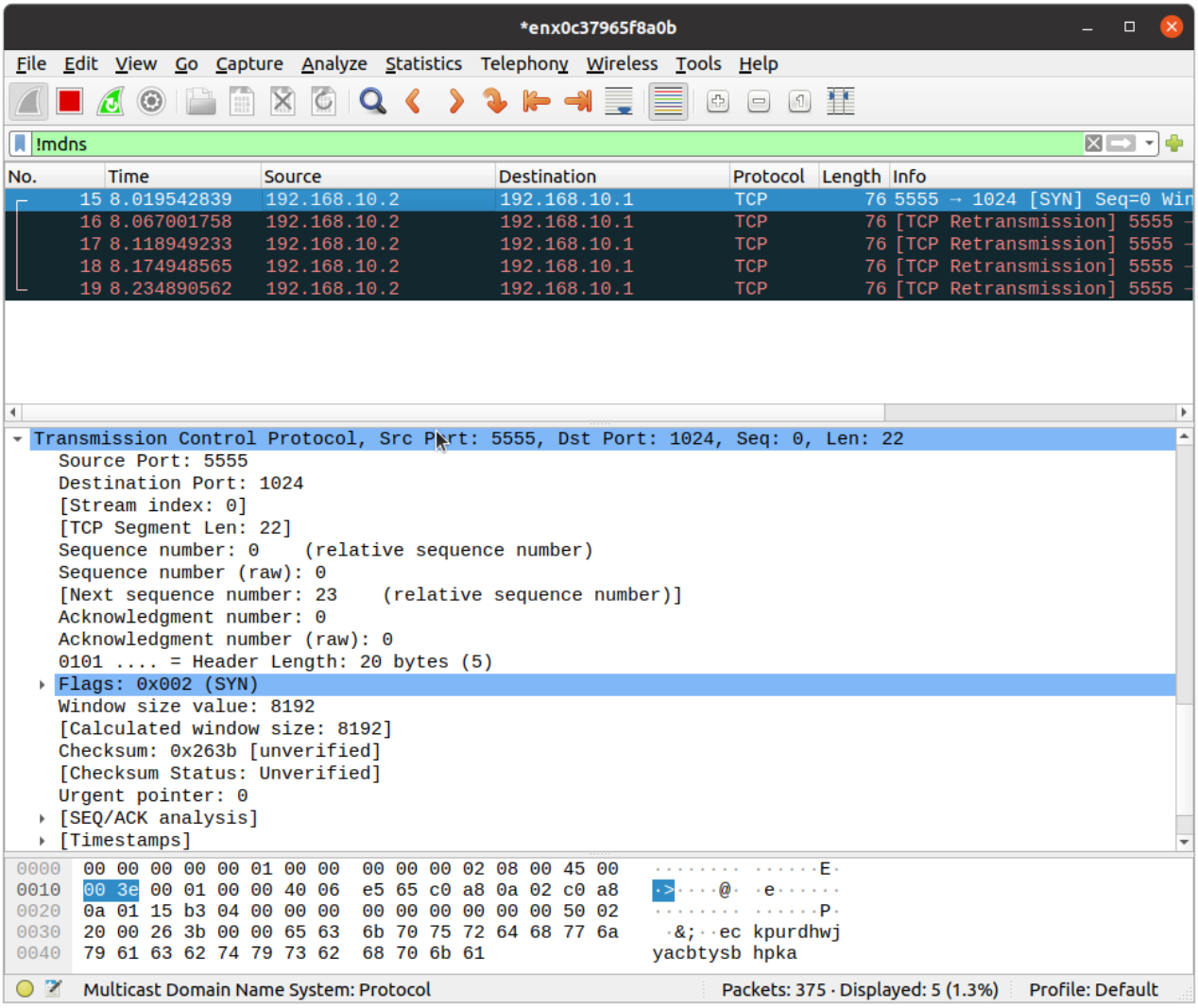
Wee see that the Package is sent by UDP protocol on port 50000. We can filter out the MDNS packets with a display filter.

send.py
~/CWM-ProgNets/assignment1

```python
#!/usr/bin/python

from scapy.all import Ether, IP, sendp, get_if_hwaddr, get_if_list, TCP, Raw, UDP
import sys
import random, string


def randomword(max_length):
    length = random.randint(22, max_length)
    return ''.join(random.choice(string.ascii_lowercase) for i in range(int(length)))

def send_random_traffic(num_packets, interface, src_ip, dst_ip):
    dst_mac = "00:00:00:00:00:01"
    src_mac= "00:00:00:00:00:02"
    total_pkts = 0
    port = 1024
    for i in range(num_packets):
            data = randomword(22)
            p = Ether(dst=dst_mac,src=src_mac)/IP(dst=dst_ip,src=src_ip)
            p = p/TCP(sport= 5555, dport=port)/Raw(load=data)
            sendp(p, iface = interface, inter = 0.01)
            total_pkts += 1
    print ("Sent %s packets in total" % total_pkts)

if __name__ == '__main__':
    if len(sys.argv) < 5:
        print("Usage: python send.py packet_num interface src_ip dst_ip")
        sys.exit(1)
    else:
        num_packets = sys.argv[1]
        interface = sys.argv[2]
        src_ip = sys.argv[3]
        dst_ip = sys.argv[4]
        send_random_traffic(int(num_packets), interface, src_ip, dst_ip)
```

Python Tab Width: 8 Ln 20, Col 22 INS

We now send the packets via TCP protocol on port 5555.



We do see the packets on Wireshark.