***Project report***

Final Project-

DHCP Server & Client

Course Title: *Internet Applications*

Name: *TANG Muyang (2014212929)*

*OU Yihang (2014212948)*

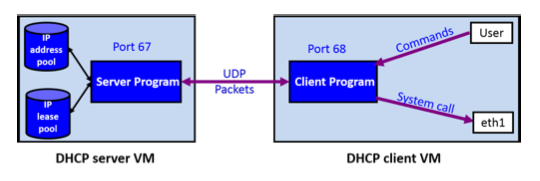
Date: *June 17th,2017*

1. Overview
2. Overview requirements

This project aim to implements a DHCP process which includes client and server. Through this project, we should understand the DHCP (Dynamic Host Configuration Protocol) comprehensively. All details of DHCP include protocol and different packet information and attain IP from Ubuntu virtual machine.

1. Goals

* Deeply understand the details of DHCP (Dynamic Host Configuration Protocol)
* Complete a DHCP server program and run it in one Ubuntu virtual machine
* Complete a DHCP client program and run it in another Ubuntu virtual machine



(DHCP Server & Client Program Model)

1. Requirements Analysis
2. Development environment

* IDE: Sublime
* Language: C
* Operation system: Ubuntu
* Compiler: gcc

1. Special function requirement
2. Client
3. Listen to UDP port 68, attain or set the IP address, MAC address and netmask for the client virtual machine.
4. Obtain IP address, netmask, gateway, dns server address, dhcp server ID and IP address lease time from DHCP server and configure IP address.
5. Support DHCP messages: **DISCOVERY/OFFER/REQUEST/ACK, RELEASE, REQUEST/NAK, REQUEST/ACK, INFORM/ACK**
6. Support DHCP options:

**1 (Subnet Mask Value)**

**3 (Router addresses)**

**6 (DNS Server addresses)**

**51 (IP Address Lease Time)**

**53 (DHCP Message Type)**

**54 (DHCP Server Identification)**

**55 (Parameter request list)**

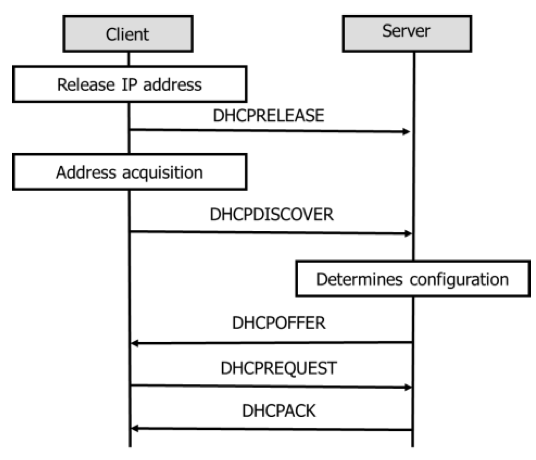
**58 (DHCP Renewal Time T1)**

**59 (DHCP Rebinding Time T2)**

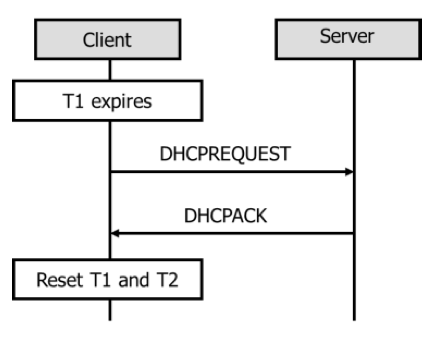
**60 (Class Identifier, set as our student number)**

**255 (END)**

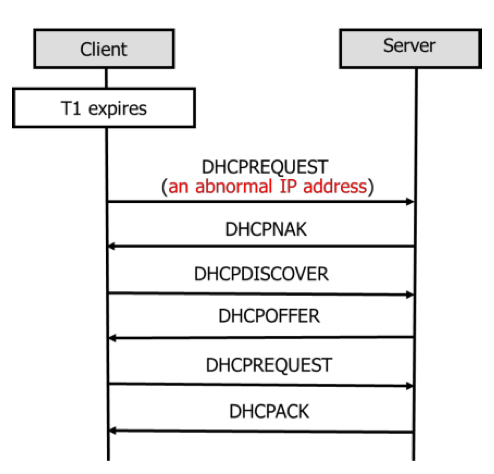
1. Four messages during address acquisition can be delivered on broadcast packets: DISCOVEY/OFFER/REQUEST/ACK.
2. Finish the following DHCP procedures



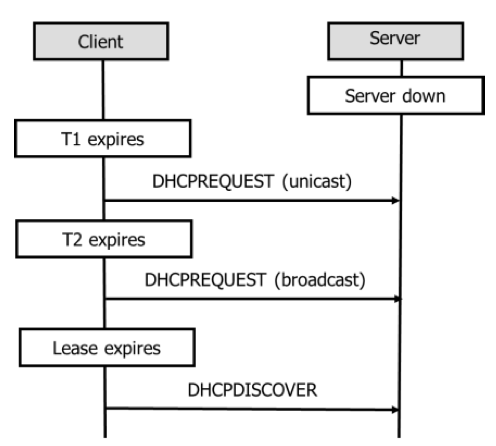
(release + address acquisition)



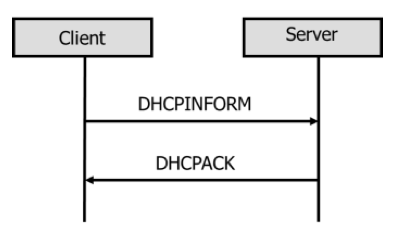
(successful lease renew)



(Fig.4 failed lease renew (with an abnormal IP address) + address acquisition again)



(failed lease renew (server down))



(inform)

1. Server
2. The main requirement of server is to response to the different requirements from the client. It mainly responses 3 kinds of packets: DHCP Offer, DHCP ACK and DHCP NAK, and some operation when it receives DHCP Release.
3. When the server receives DHCP Discover, it needs to broadcast or unicast DHCP Offer according to DHCP Discover’s broadcast flag, containing IP address offered from the dhcp.config file, server identifier (option 54), lease time, (option 51) from the dhcp.config file, subnet mask (option 1) from the dhcp.config file, router IP (option 3) from the dhcp.config file, domain name server IP (option 6) from the dhcp.config file, renewal time value (option 58), rebinding time value (option 59).
4. When the server receives DHCP Request for applying IP address, it needs to broadcast or unicast DHCP ACK according to DHCP Request’s broadcast flag, containing IP address offered, server identifier (option 54), lease time, (option 51), subnet mask (option 1), router IP (option 3), domain name server IP (option 6), renewal time value (option 58), rebinding time value (option 59).
5. When the server receives DHCP Request for applying IP address, it needs to broadcast or unicast DHCP ACK according to DHCP Request’s broadcast flag, containing IP address offered, server identifier (option 54), lease time, (option 51), subnet mask (option 1), router IP (option 3), domain name server IP (option 6), renewal time value (option 58), rebinding time value (option 59). And then modify the dhcp.config file to delete the offered IP and store the timestamp, mac address of client, and offered IP in the dhcp.lease.
6. When the server receives DHCP Request for renew a IP address, it needs to check in the dhcp.lease to see whether it is a legal IP. If it is a legal IP, it should unicast DHCP ACK, containing IP address offered, server identifier (option 54), lease time, (option 51), subnet mask (option 1), router IP (option 3), domain name server IP (option 6), renewal time value (option 58), rebinding time value (option 59). If it is not legal, the server will unicast DHCP NAK containing server identifier (option 54).
7. When the server receives DHCP Inform, it needs to unicast DHCP ACK containing IP address offered, server identifier (option 54), subnet mask (option 1), router IP (option 3), domain name server IP (option 6), renewal time value (option 58), rebinding time value (option 59).
8. When the server receives DHCP Release, it needs to delete the related part of the client in dhcp.lease and add the client IP to the IP address pool.
9. Preliminary Design
10. Client
11. Data structure design

* Struct

sockaddr\_in sendAddr //Client address struct

sockaddr\_in serverAddr //Server address struct

sockaddr\_in selfAddr //Client self-address struct

* Variable

char sendBuf[] //Send data buffer

char recvBuf[] //receive data buffer

int sendSize=312 //Send packet size=312

int recvSize //Receive packet size

int svrAddrLen //Server Address Length

int counter //Padding counter

unsigned int DHCPServerIPAddress =0xffffffff //Broadcast Server IP address

unsigned int LeaseTime //Lease time

unsigned int T1Time //T1 expires time

unsigned int T2Time //T2 expires time

int i=0 //timeclock

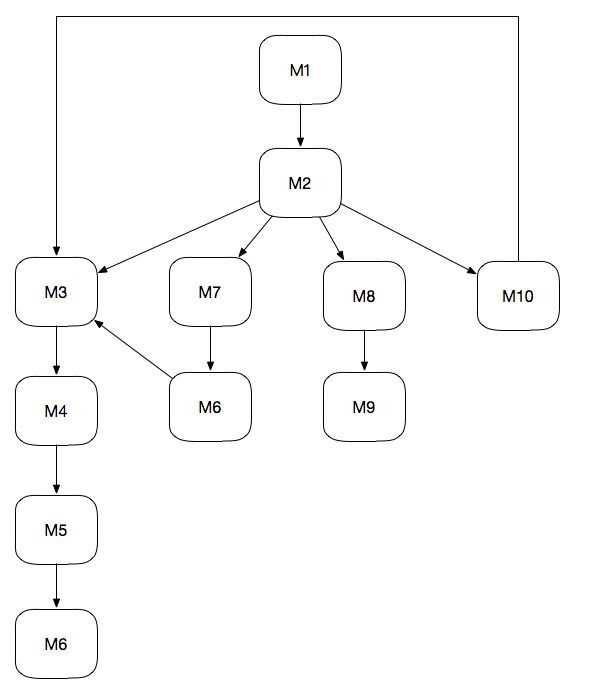
pthread\_t thd1 //thread id using for control T1, T2 and Lease time

int thd=0 //To judge thd1

int type =-1 //Set Message Type

int subtype =-1 //Set ACK message Type

1. Modules
2. C.M1: Set the Client Self Address and full-in DHCP packet
3. C.M2: Interaction command line
4. C.M3: Send Discover
5. C.M4: Receive Offer
6. C.M5: Send Request (Include T1 expire (Unicast) and T2 expire (Broadcast))
7. C.M6: Receive ACK and NAK (when receive Request)
8. C.M7: Send Release (Set abnormal IP)
9. C.M8: Send Inform
10. C.M9: Receive ACK (when receive Inform)
11. C.M10: T1, T2, Lease time interaction



(Client Module relationship)

1. Server
2. Data structure design

* Struct

struct sockaddr\_in svrAddr; /\* Local address \*/

struct sockaddr\_in cltAddr; /\* Client address \*/

struct sockaddr\_in broadcastAddr; /\* Broadcast address \*/

* Variable

unsigned int cliAddrLen; /\* Length of client address \*/

int sock; /\* Socket \*/

char recvBuf[MAX\_SIZE]; /\* Buffer for receive \*/

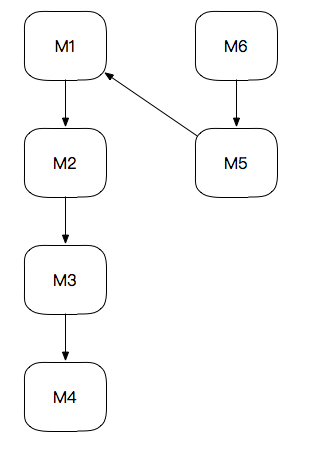
char sendBuf[MAX\_SIZE]; /\* Buffer for send \*/

int recvMsgSize; /\* Size of received message \*/

int i; /\* Counter \*/

int LEASETIME; /\* Lease Time\*/

1. Modules
2. S.M1: Receive Discover
3. S.M2: Send Offer
4. S.M3: Receive Request
5. S.M4: Send ACK
6. S.M5: Send ACK (renew IP)
7. S.M6: Receive Release (Release IP)
8. S.M7: Receive Inform



(Server Module relationship)

1. Detailed Design
2. Client
3. Main part

In main part, Client as the interaction role of this system, we should input command in command line. Using 7 different command to achieve different functions. Besides we should full in the content of DHCP packet information. When received packets and sending packets. Then modify the different parts and rewrite the options. When rewiring the different parts, the server will write down the content in the appropriate positions.

-------------------------------------------------------------------------

Sock()

Bind to eth1()

Bind()

Full in DHCP Packet Basic information

Interact()

For(;;)

receive()

sendBuf<-recBuf

get mask, router, DNS, lease time, prepared IP

if(option53=discover)/\*broadcast offer\*/

if(available IP=0)

warn

else

discoverflag<-1

fill in boot reply

fill in prepared IP

fill in option53

fill in option 54

fill in option 51

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

else if(option53=Request)

if(client IP=0.0.0.0)/\*broadcast ACK\*/

fill in boot reply

fill in prepared IP

fill in option53

fill in option 54

fill in option 51

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

else

discoverflag<-1

fill in boot reply

if(legal)/\*response ACK\*/

fill in client IP

fill in option53

fill in option 54

fill in option 51

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

else/\*response NAK\*/

fill in option53

fill in option 54

fill in option 255

padding

else if(option53=Release)

fill in option53

fill in option 54

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

else if(option53=Inform) /\*response ACK\*/

fill in option53

fill in option 54

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

if(receive DHCP packet)

if(flag=unicast)

send unicast

else if (flag=broadcast)

send broadcast

-------------------------------------------------------------------------

1. Get IP

Using ioctl to get IP from eth1.

1. T1, T2 and LeaseTime

-------------------------------------------------------------------

Set thread thd1

If (thd1)

Int t=0

t=t+1

if(t=T1)

send renew1

if (t=T2)

send renew2

if (t=LeaseTime)

send Discover

-------------------------------------------------------------------

1. Server
2. Main part

In main parts, firstly the server will copy all the content from the received packet to send packet since there are many parts are the same in the received packets and sending packets. Then modify the different parts and rewrite the options. When rewiring the different parts, the server will write down the content in the appropriate positions.

Pseudocode:

-------------------------------------------------------------------------

Sock()

Bind to eth1()

Bind()

For(;;)

receive()

sendBuf<-recBuf

get mask, router, DNS, lease time, prepared IP

if(option53=discover)/\*broadcast offer\*/

if(available IP=0)

warn

else

discoverflag<-1

fill in boot reply

fill in prepared IP

fill in option53

fill in option 54

fill in option 51

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

else if(option53=Request)

if(client IP=0.0.0.0)/\*broadcast ACK\*/

fill in boot reply

fill in prepared IP

fill in option53

fill in option 54

fill in option 51

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

else

discoverflag<-1

fill in boot reply

check legal

if(legal)/\*response ACK\*/

renew dhcp.lease

fill in client IP

fill in option53

fill in option 54

fill in option 51

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

else/\*response NAK\*/

fill in option53

fill in option 54

fill in option 255

padding

else if(option53=Release)

modify file

else if(option53=Inform) /\*response ACK\*/

fill in option53

fill in option 54

fill in option 1

fill in option 3

fill in option 6

fill in option 58

fill in option 59

fill in option 255

padding

if(receive DHCP packet)

if(flag=unicast)

send unicast

else if (flag=broadcast)

send broadcast

-------------------------------------------------------------------------

1. Get IP

Using ioctl to get IP from eth1.

1. Offer IP

Remove IP from dhcp.config and add the related information to the dhcp.lease. Get all the information except offered IP and write them in a new file. Then delete the former file and modify the file name.

Pseudocode:

---------------------------------------------------------------------

Input:MAC

Open dhcp.config

Create dhcp.config2

Get mask and write in dhcp.config2

Get router and write in dhcp.config2

Get DNS and write in dhcp.config2

Get lease time and write in dhcp.config2

Get prepared IP /\*Don’t write here\*/

While (!EOF)

Get free IP and write in dhcp.config2

Close files

dhcp.config<-dhcp.config2

get timestamp

open dhcp.lease to append

append timestamp MAC IP

close files

---------------------------------------------------------------------

1. Get IP address number

Get the number of lines of dhcp.config, and the number-5 is the available number of IPs.

Pseudocode:

---------------------------------------------------------------------

Input:file

Output:number of available IPs

Open dhcp.config

While (!EOF)

number +1

Close files

Return number -5

---------------------------------------------------------------------

1. Check IP

Since for each line in dhcp.lease, it will store the timestamp, MAC and client IP. Find the line contains IP and read line. Split it into 3 parts, check the IP and MAC, then compare the time now and lease time.

---------------------------------------------------------------------

Input:IP,MAC

Output:have or not

Open dhcp.lease

While (!EOF)

Get timestamp

Timestamp<-timestamp+leasetime

Get MAC

Get IP

Get now

If IP=IP, MAC=MAC, timestamp>now

Return 1

Init string

Close files

---------------------------------------------------------------------

1. Release IP

Since for each line in dhcp.lease, it will store the timestamp, MAC and client IP. Find the line contains IP and delete the line. Then append IP in the dhcp.config

---------------------------------------------------------------------

Input:IP

Flag<-0

Open dhcp.lease

Create dhcp.lease2

While (!EOF)

If (there is a IP in lease pool)

Get a line from dhcp.lease

If (the IP is not a part of the line)

Write it in the dhcp.lease2

Init string

Else

Flag<-1

Init string

Close files

Dhcp.lease<-dhcp.lease2

If (flag=1)

Open dhcp.config

Append IP to dhcp.config

Close files

---------------------------------------------------------------------

1. Renew IP

Similar as check. But write the time now in as new timestamp

---------------------------------------------------------------------

Input:IP,MAC

Output:have or not

Open dhcp.lease

Open dhcp.lease2

While (!EOF)

Get timestamp

Timestamp<-timestamp+leasetime

Get MAC

Get IP

Copy string

If IP=IP

Write timestamp MAC IP in the dhcp.lease2

Else

Write string in the dhcp.lease

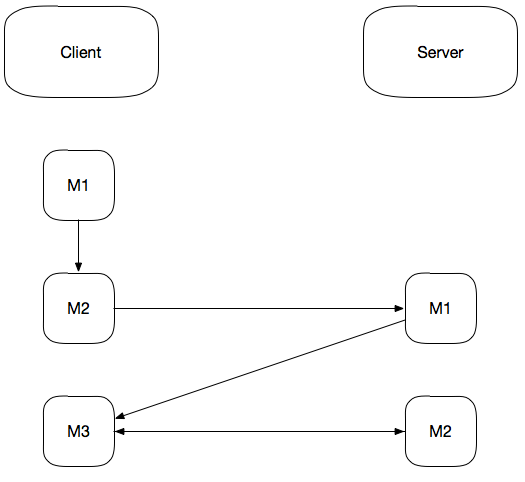
Init string

Close files

dhcp.lease<-dhcp.lease2

---------------------------------------------------------------------

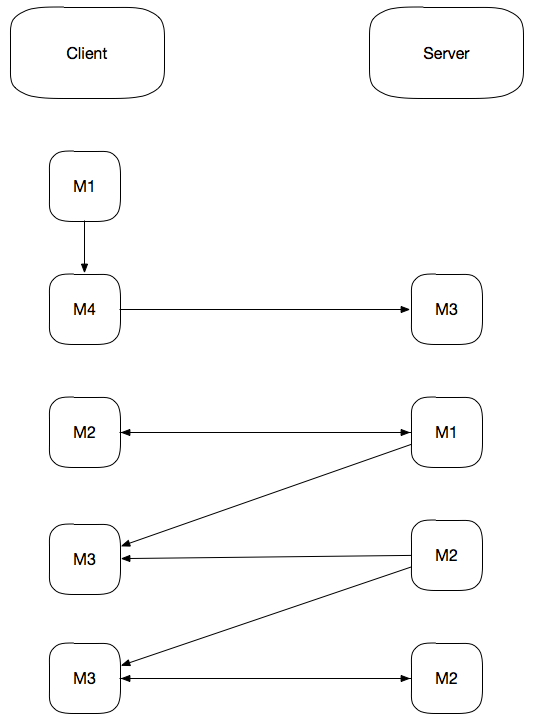
1. Interaction between Client and Server
2. Discover-Offer-Request-ACK



C.M1: main() C.M2: Send Discover

C.M3: Send Request S.M1: Send Offer S.M2: Send ACK

1. Release-Renew

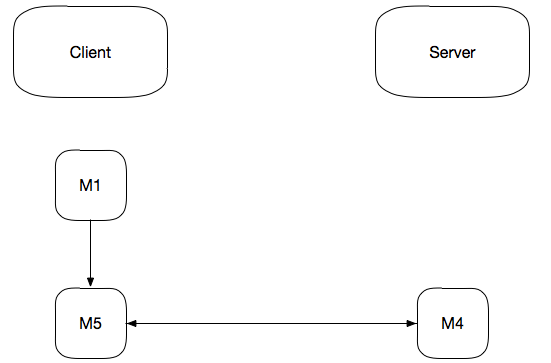


C.M1: main() C.M2: Send Discover C.M3: Send Request

C.M4: Send Release S.M1: Send Offer S.M2: Send ACK

S.M3: Release IP (Send NAK)

1. Inform

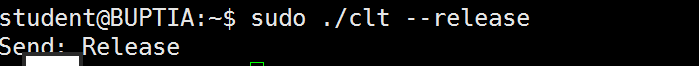


C.M1: main() C.M5: Send Inform S.M4: Send ACK

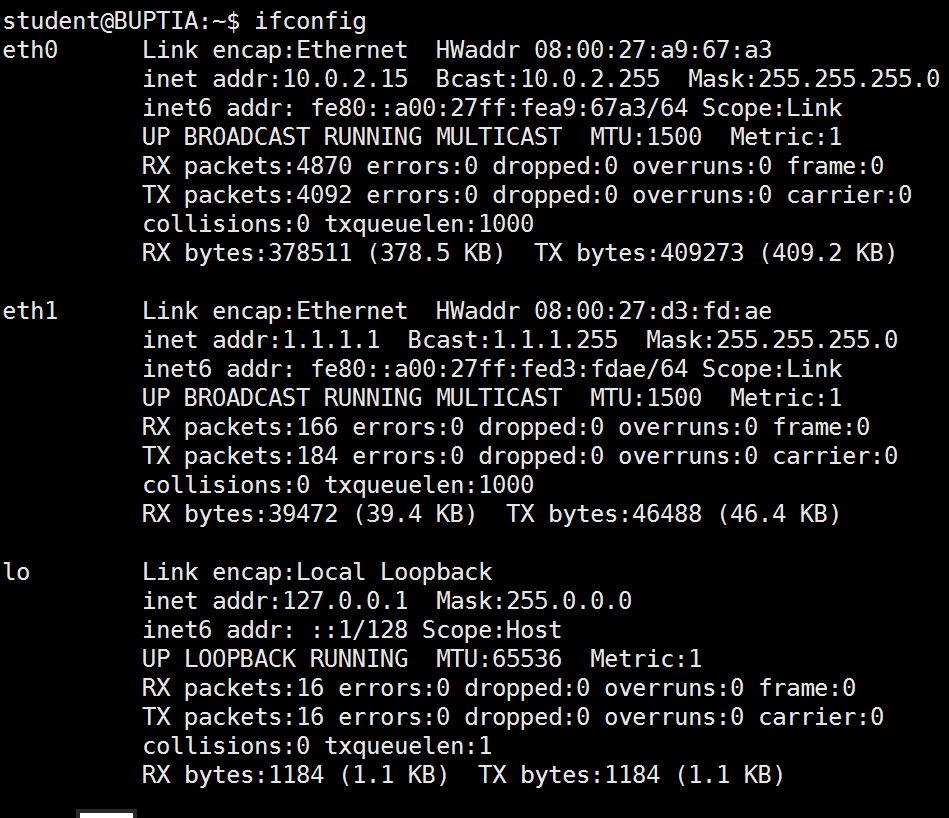
1. Results

At the beginning, the client releases the IP. Client sends release and the server operates the files.

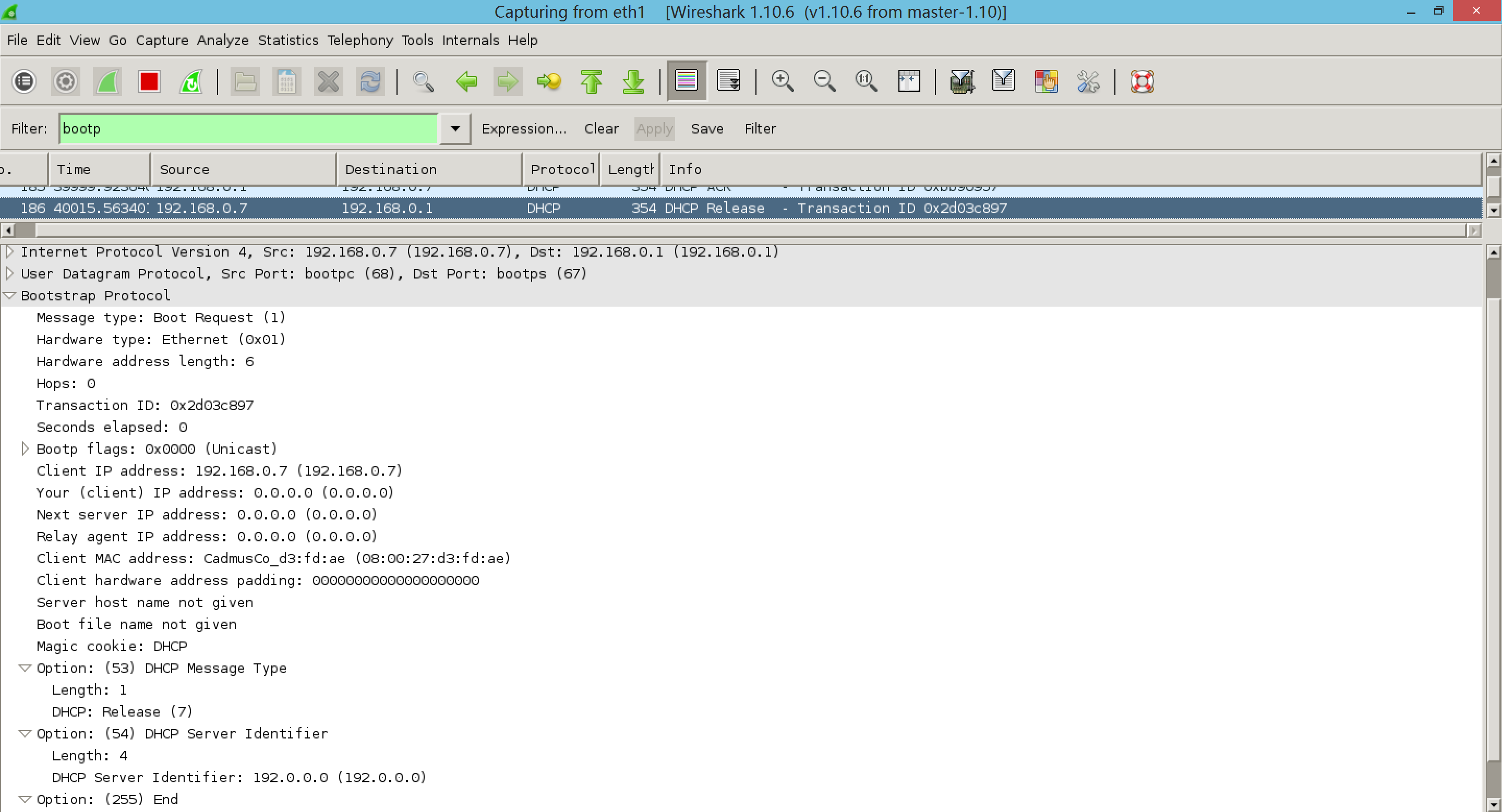
Client: Server:



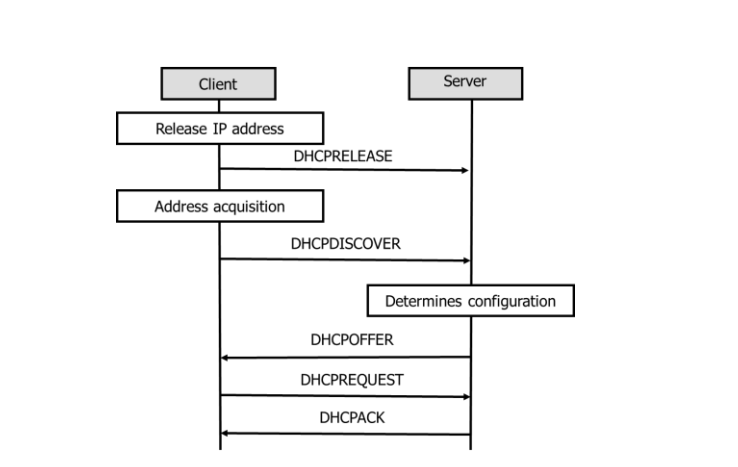
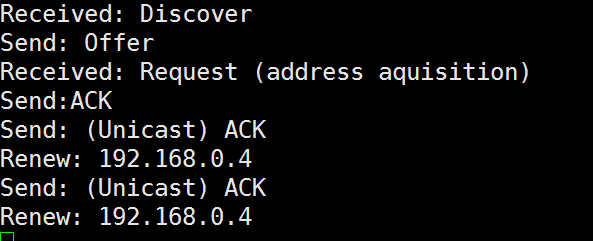
Use ifconfig to see whether it is released:



Release packet:

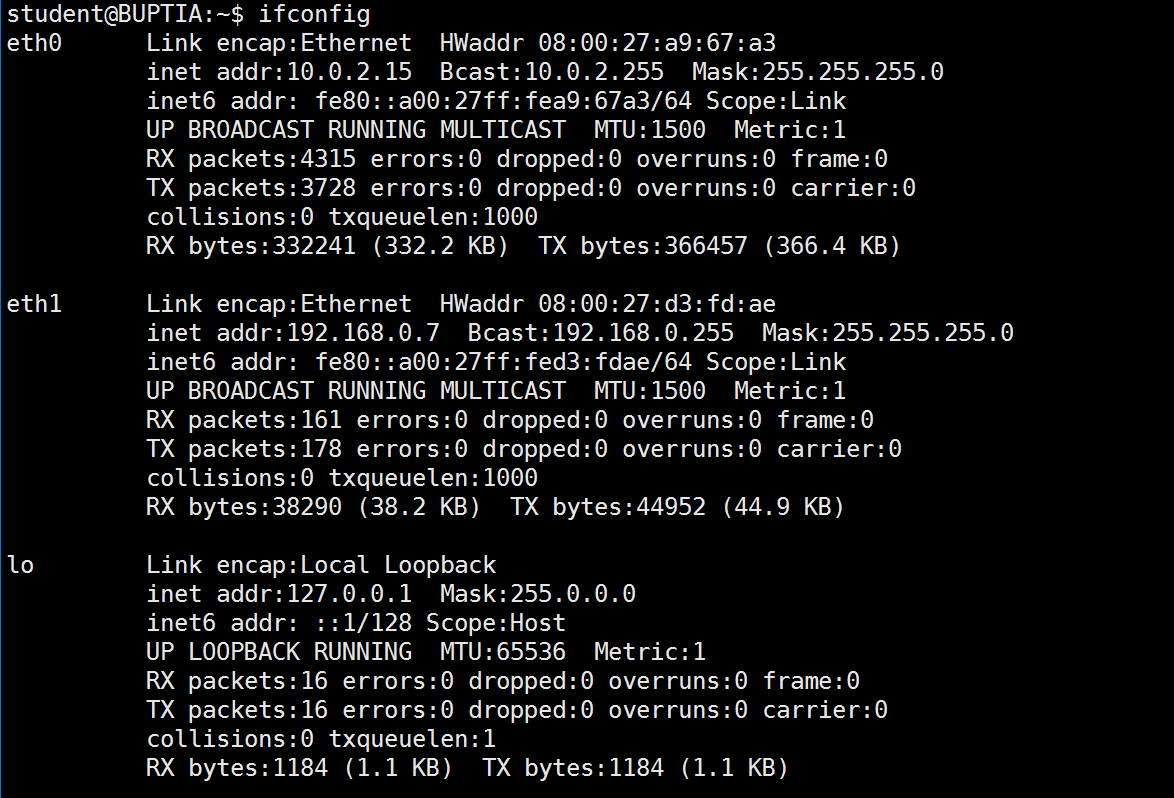
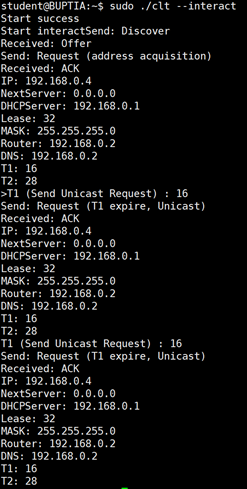


After we run the client with command, we can see the whole process of getting an IP address. After the discover-offer-request-ack process, the client get the information of IP, next server IP, DHCP server IP, lease time, mask, router, DNS and T1, T2.

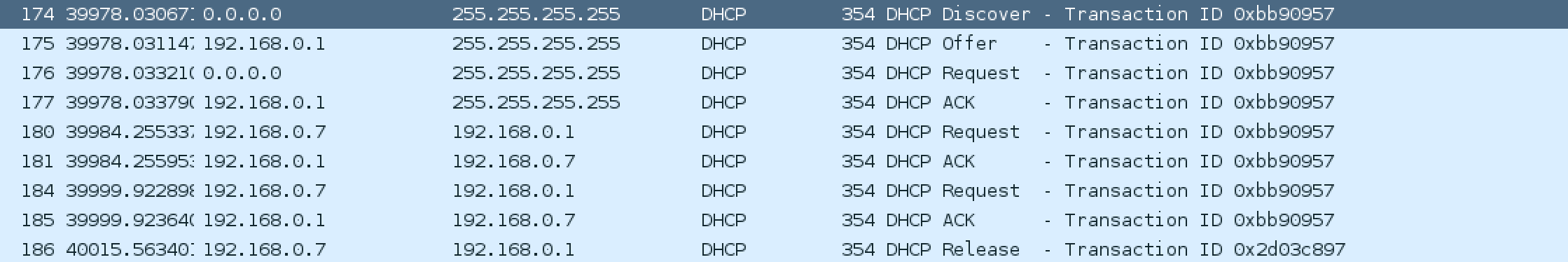


Server:

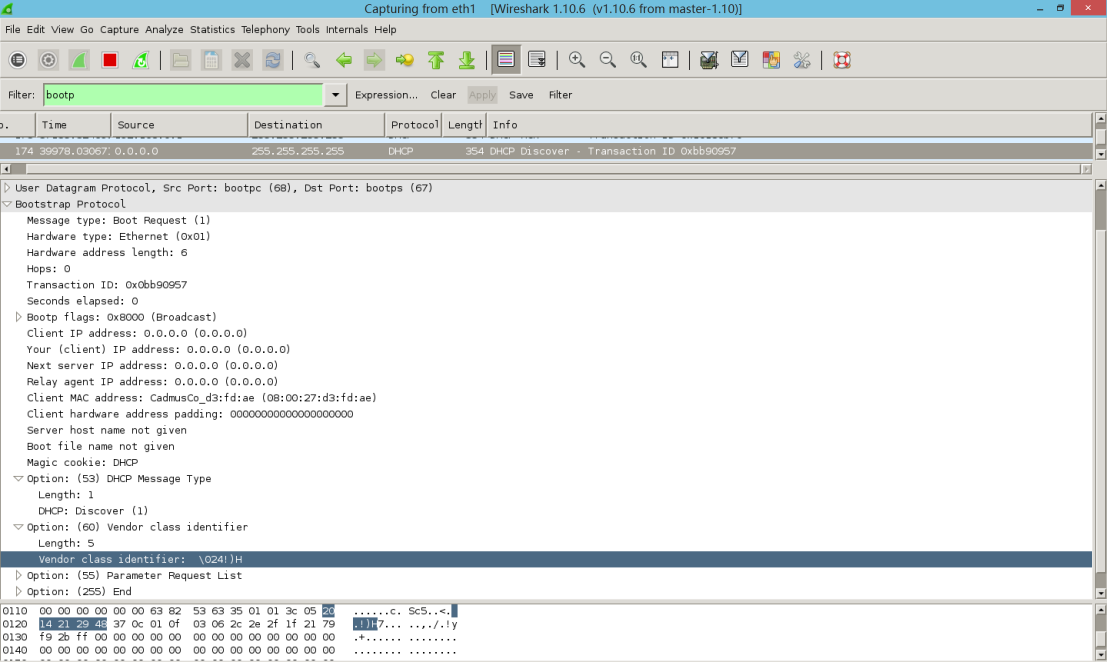
Client:



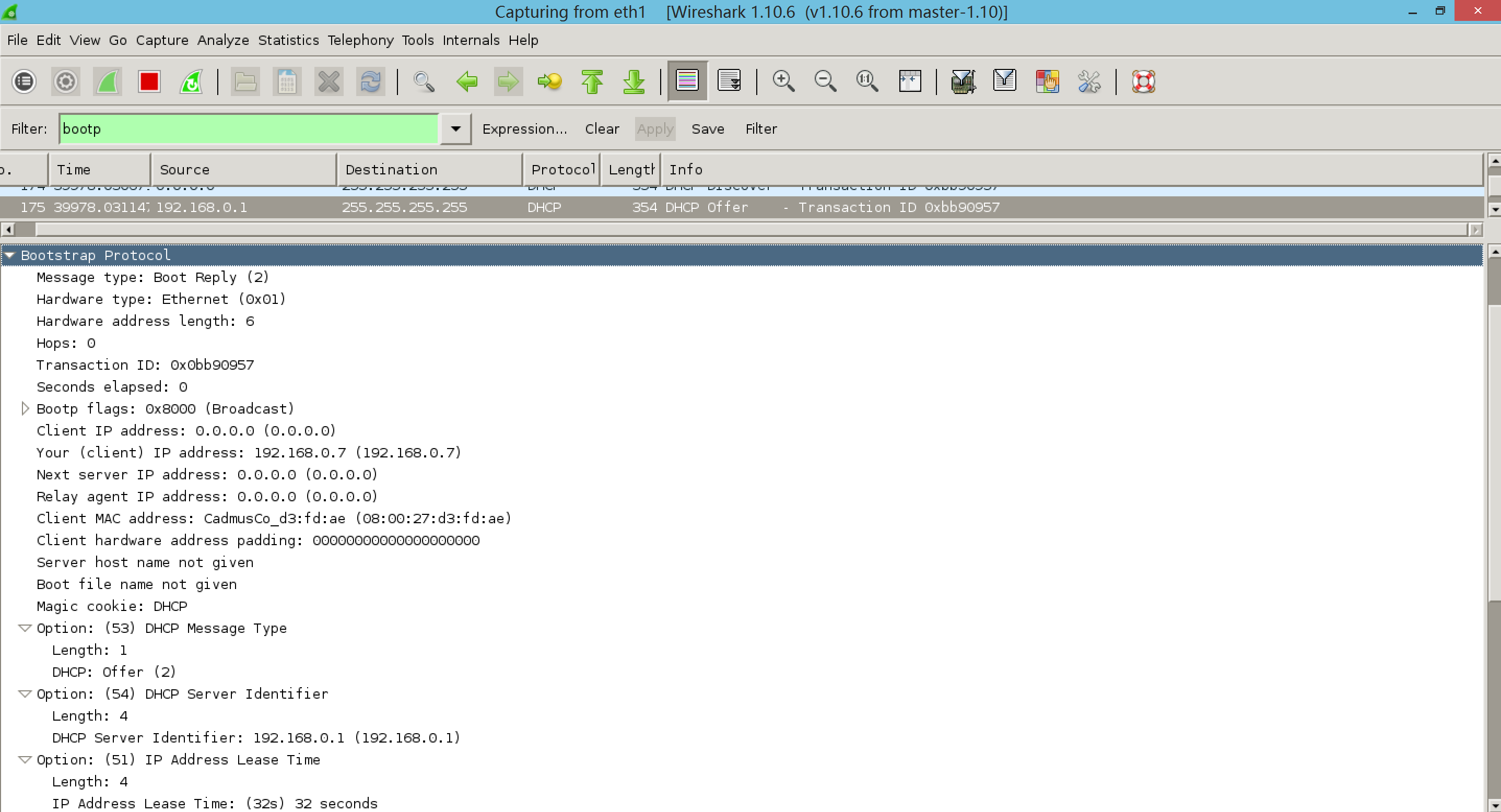
Then after the T1, the client will ask for renew, unicast a request and get an ACK from the server. It will reset the timer and wait for the next T1.



Discover packet:

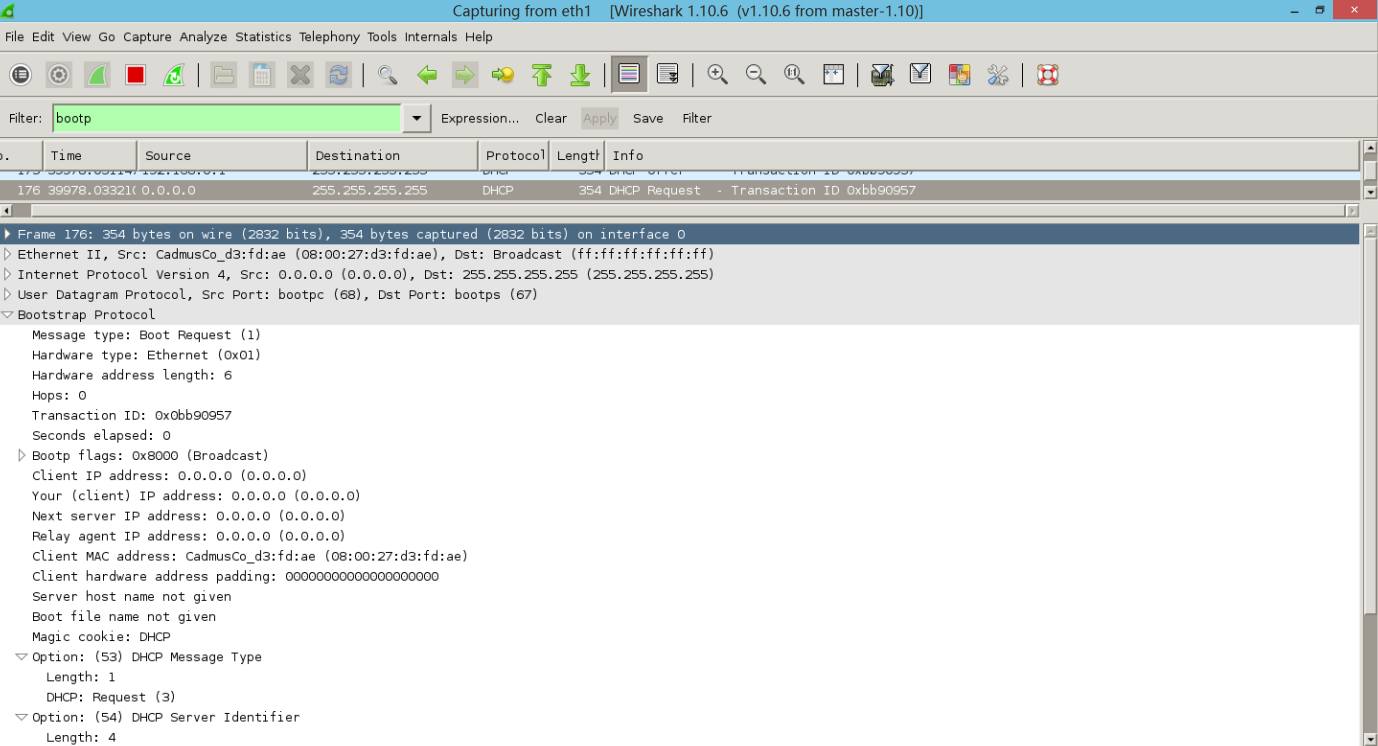


Offer:



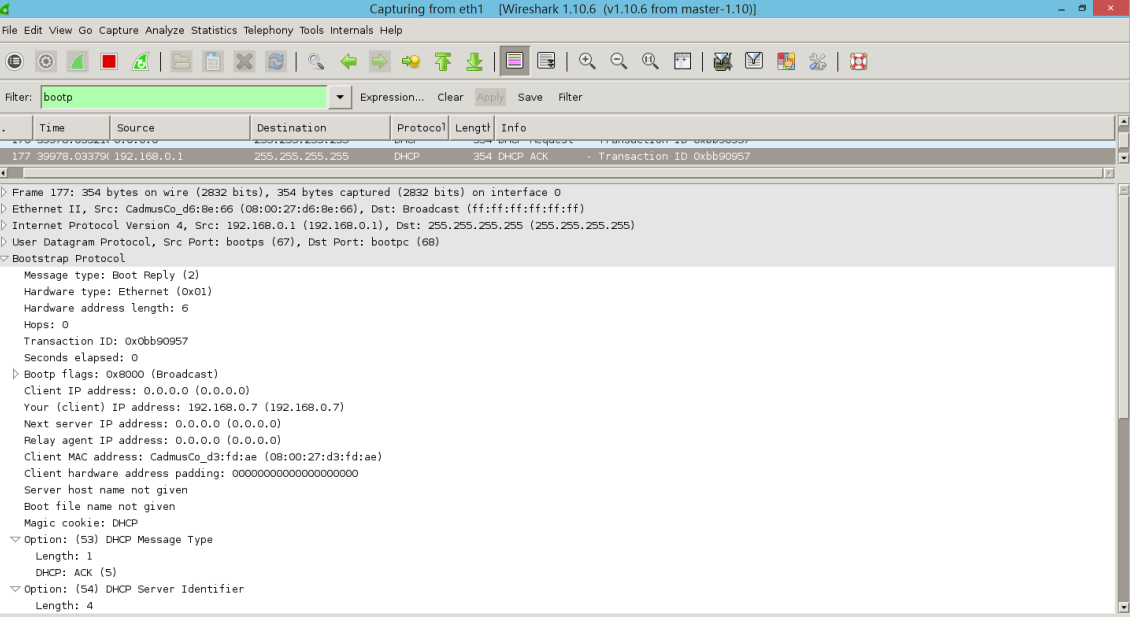


Request(d-o-r-a):



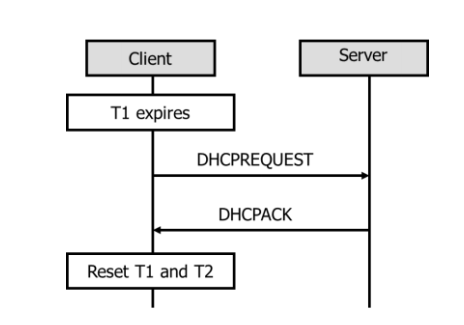


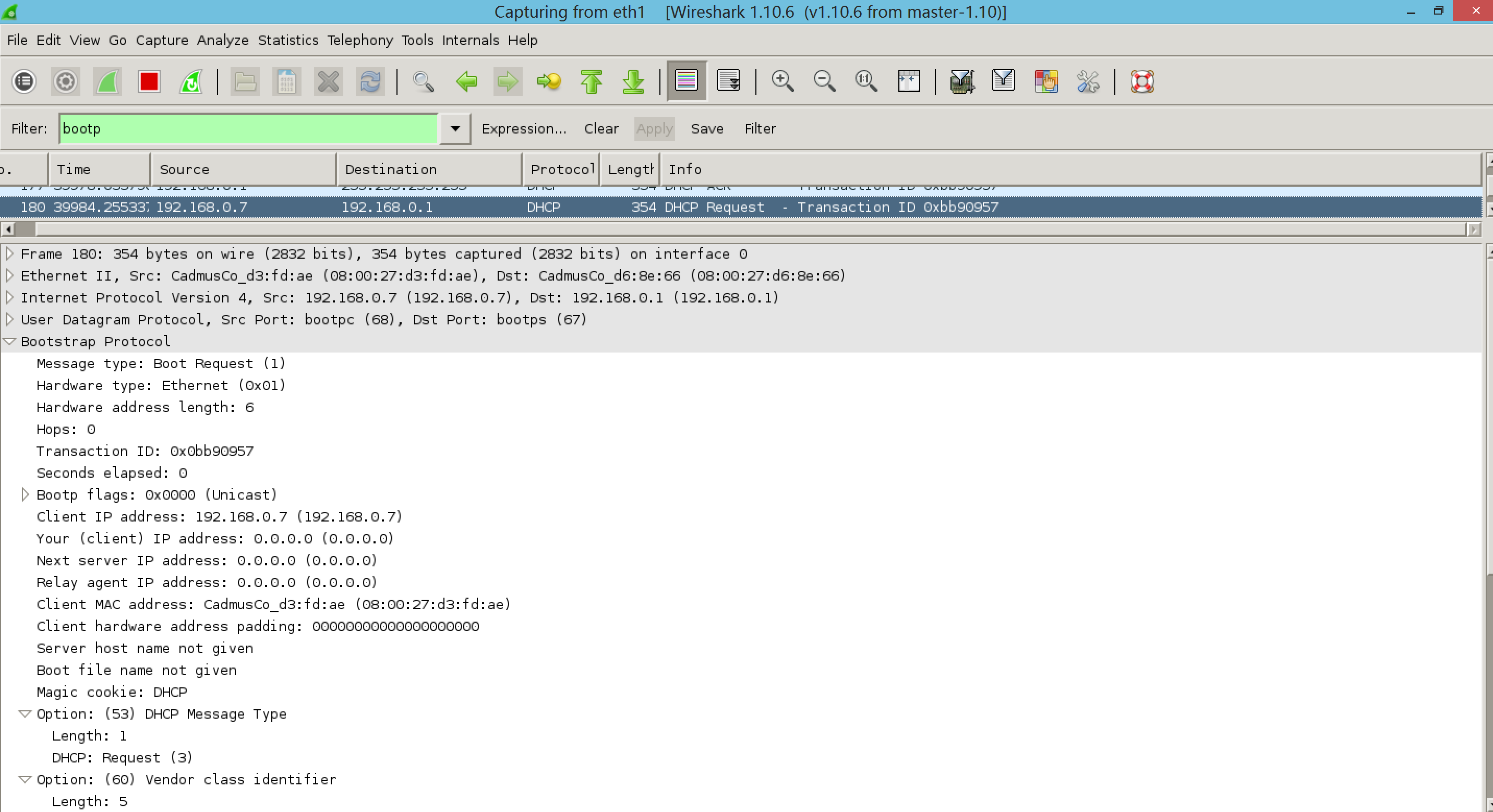
ACK:





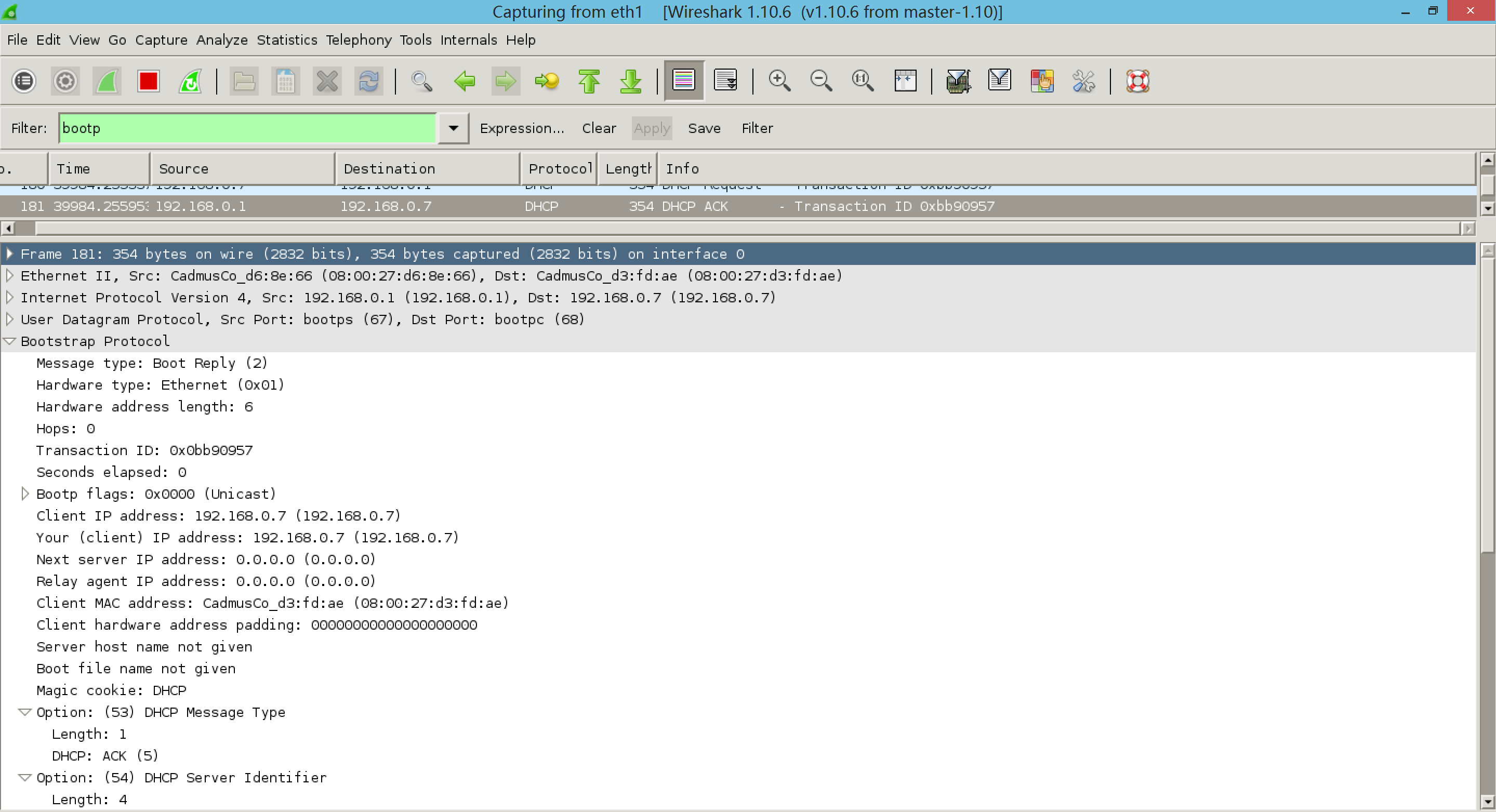
Request(T1):

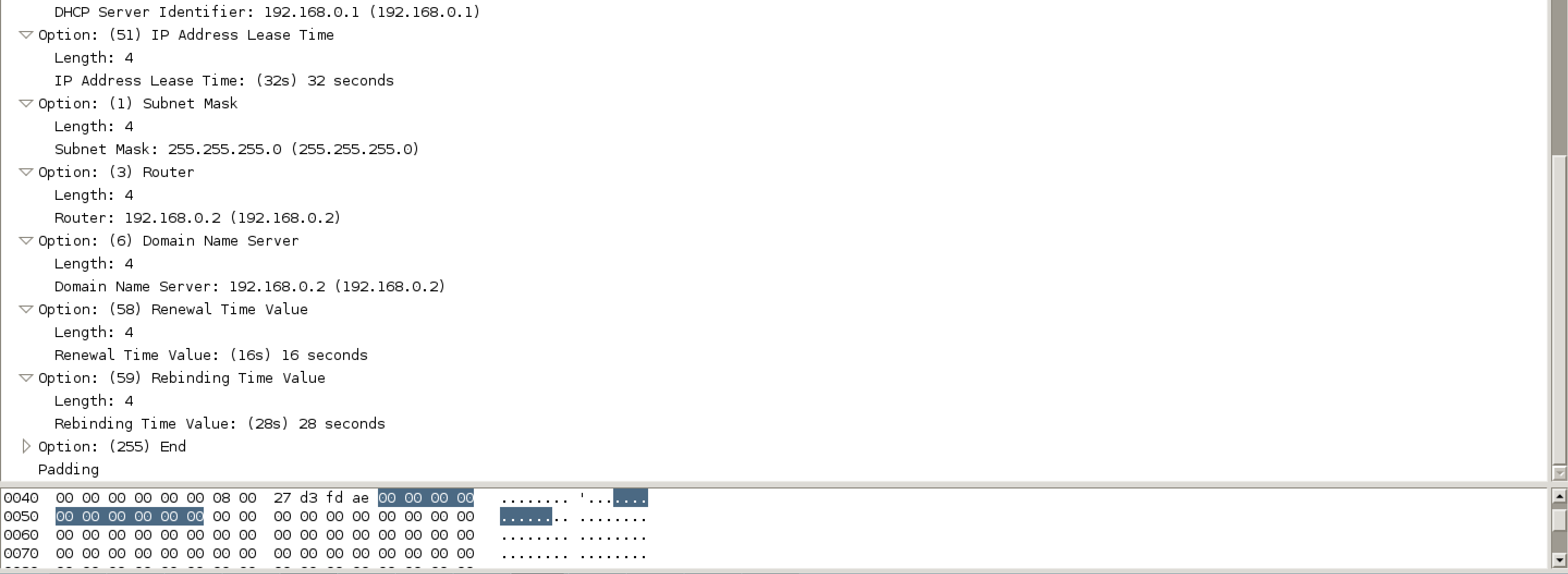




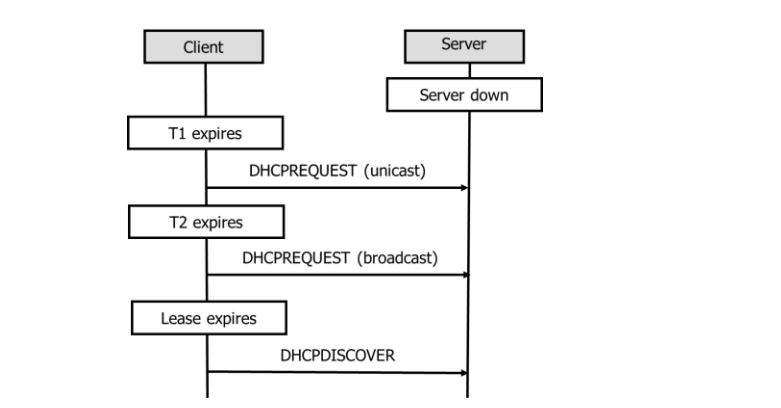


ACK(Unicast):

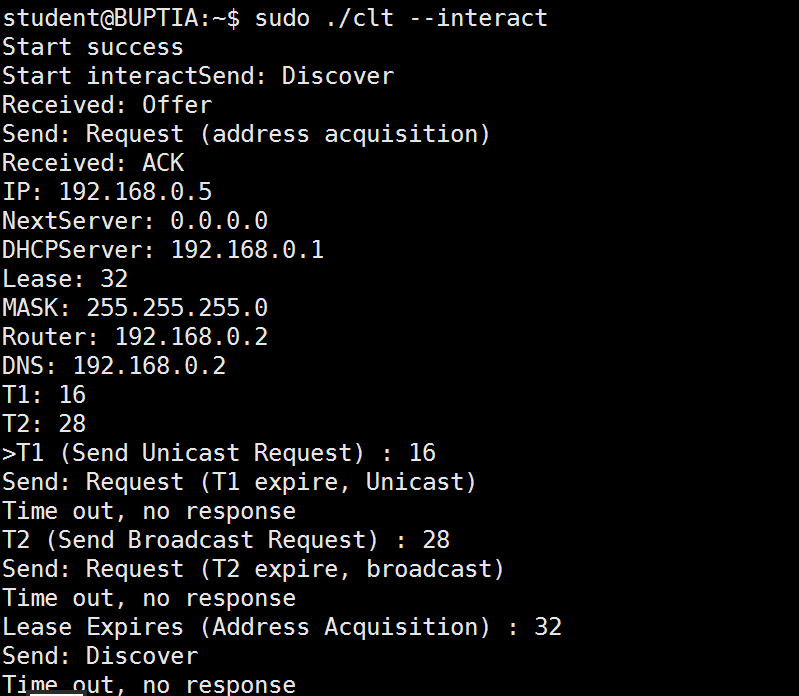




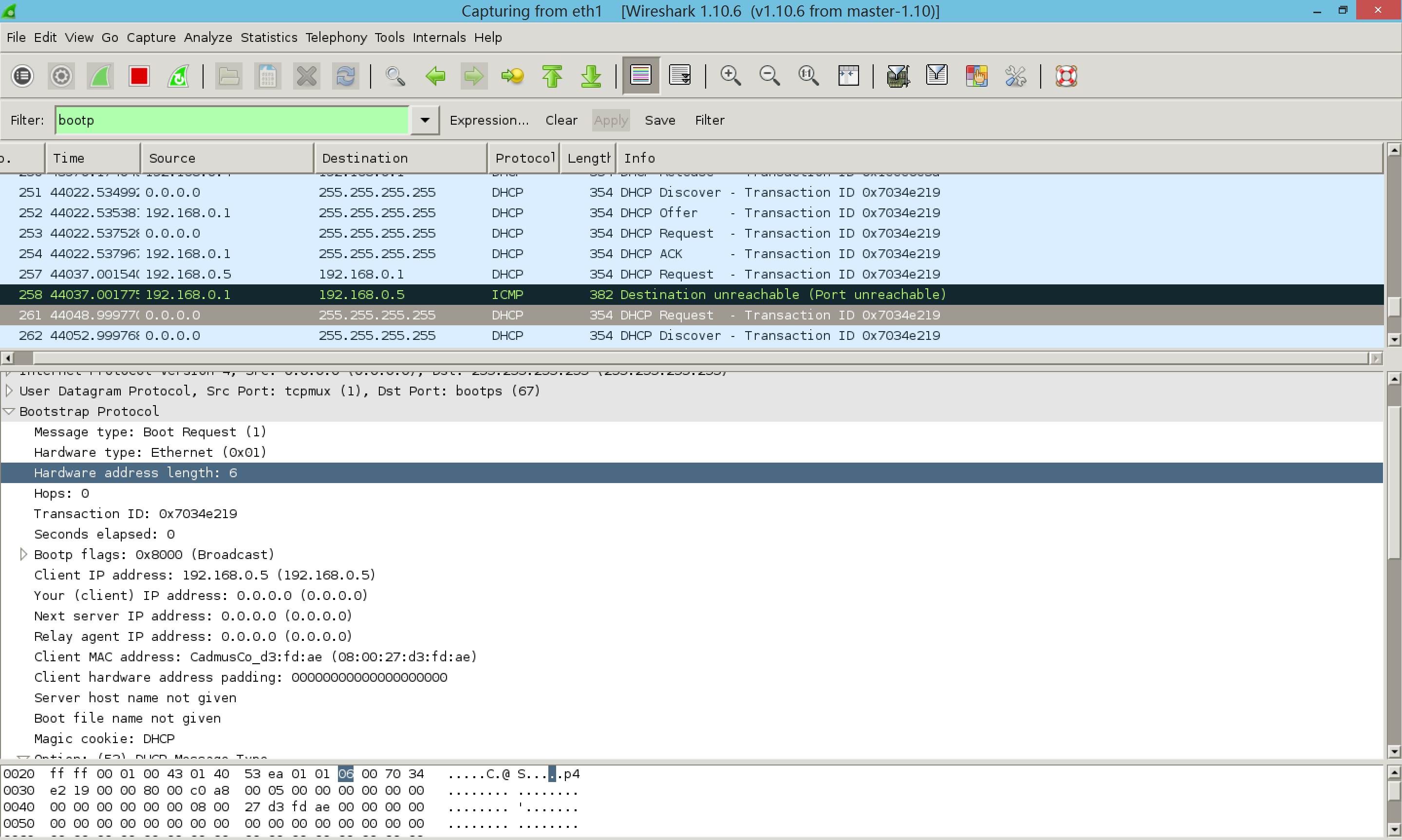
If the client get the IP but server shuts down, then in the T1 expire it will unicast request, then T2 expire it will broadcast request, after lease time it will broadcast discover.



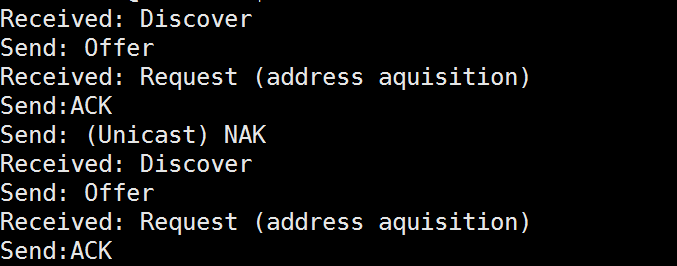
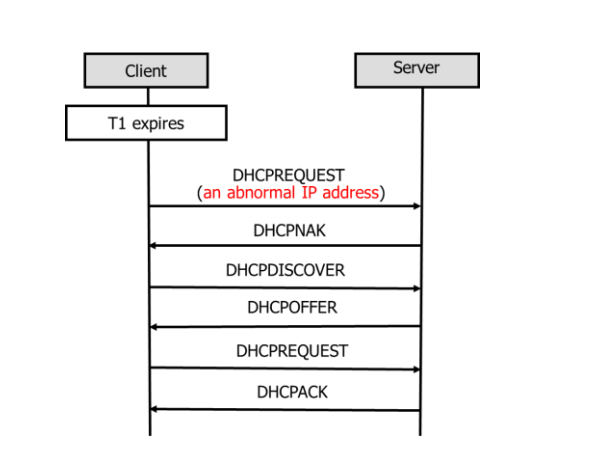
Client:



Packet process: Discover->Offer->Request->ACK->Request(unicast)-> Request(broadcast)->discover.

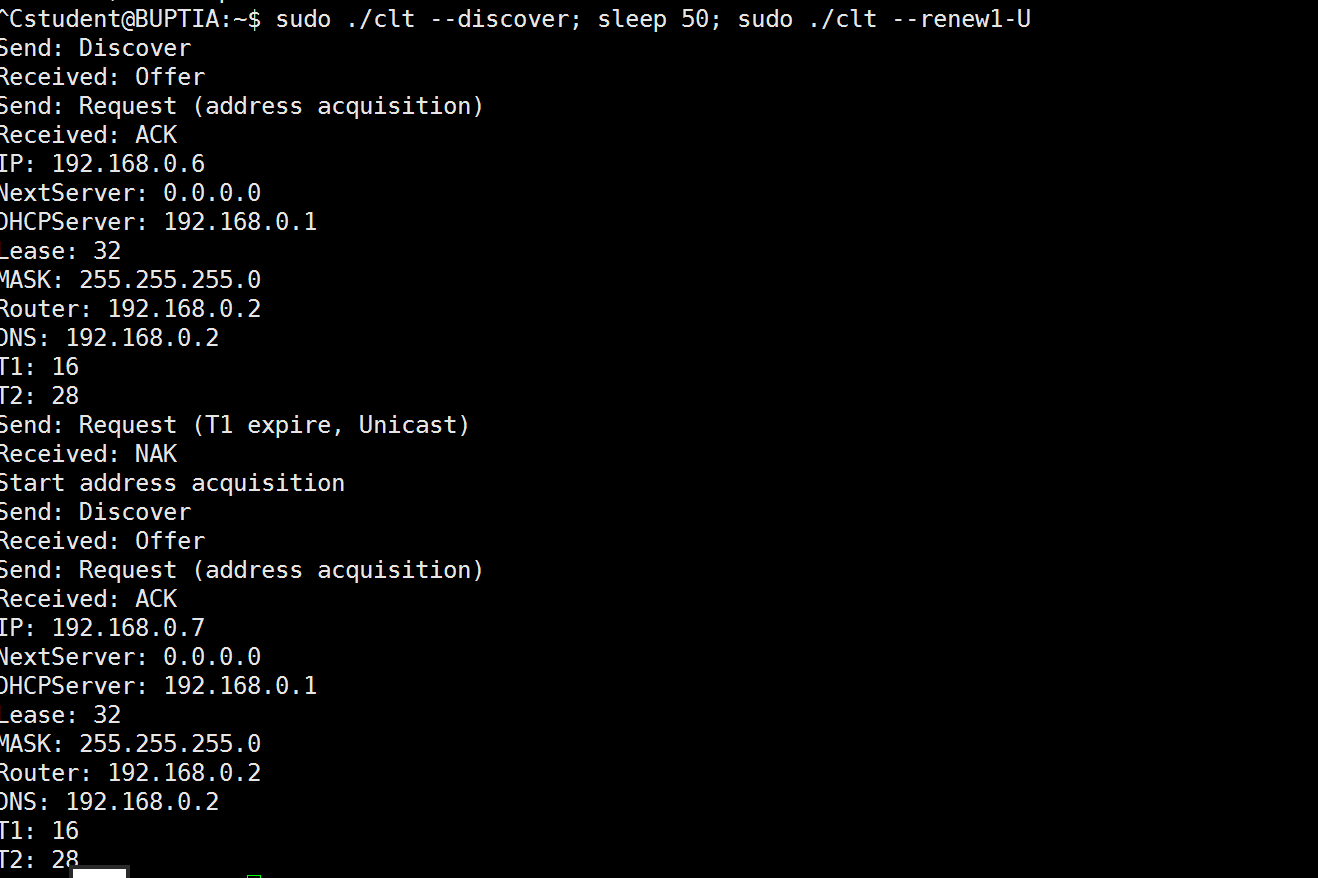


When lease time passed, the client wants to renew for an illegal IP, it will receive NAK, and the client will go to the discover process.

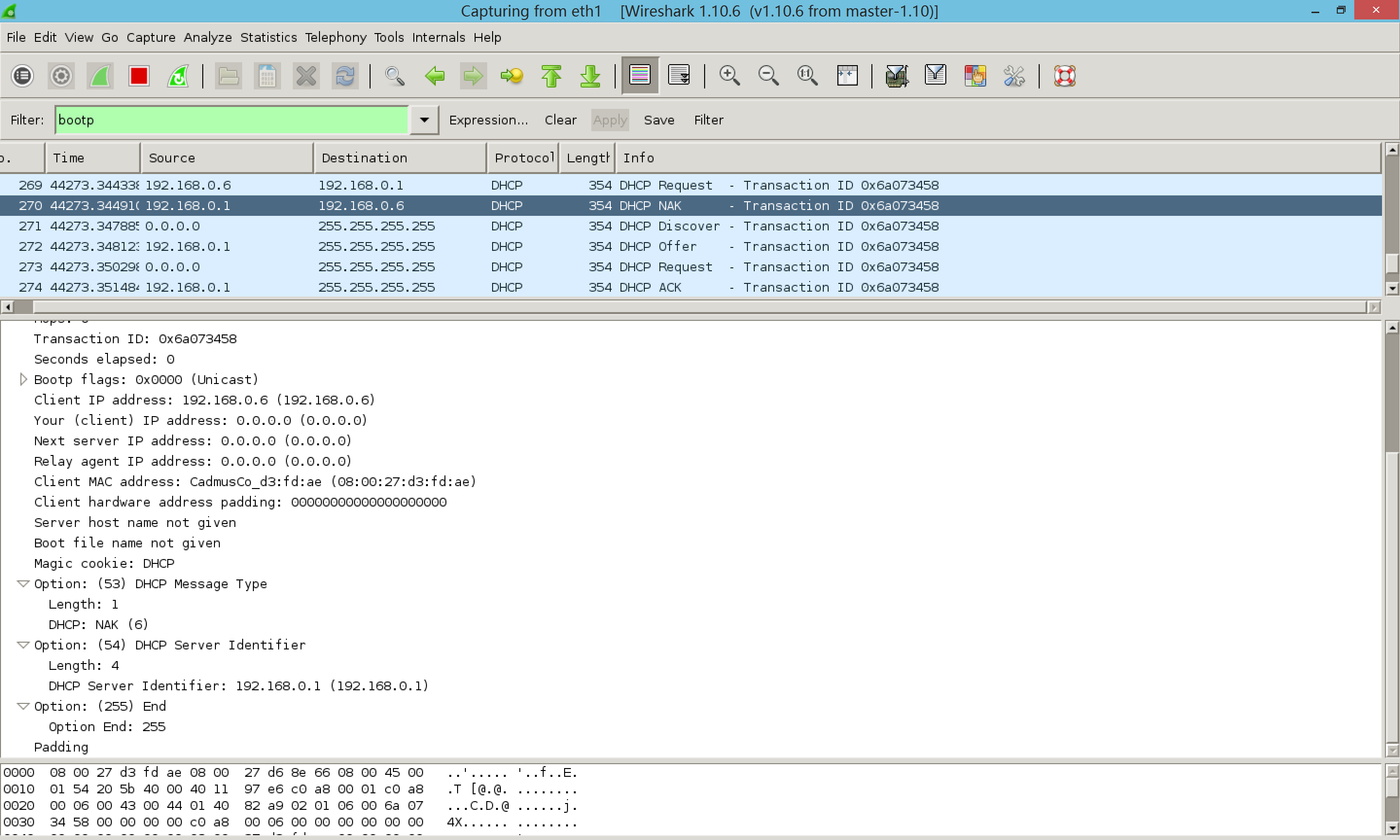


Server:

Client:



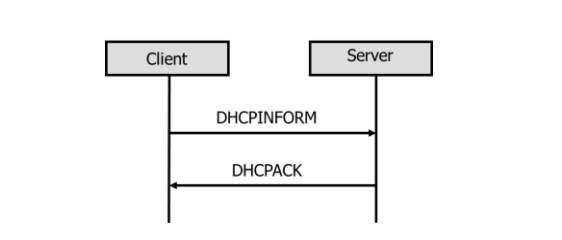
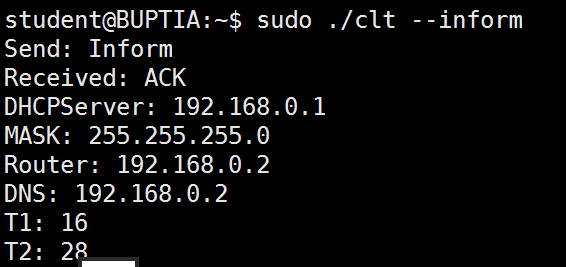
Packet process: Request(unicast)->NAK->Discover->Offer->Request->ACK. NAK packet:



Inform:

After get an IP, client unicast inform and server sends ACK without option 51.

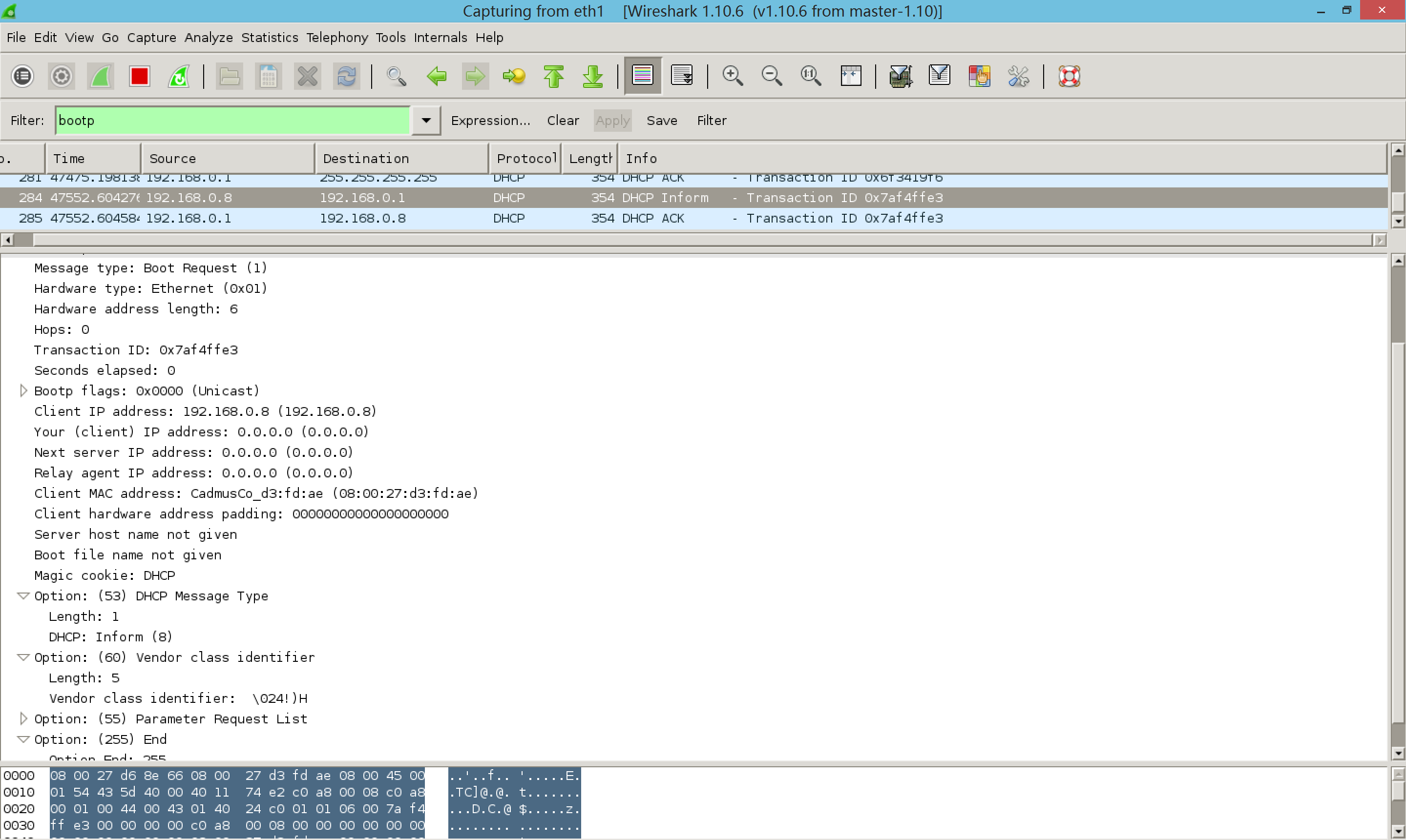
Client:



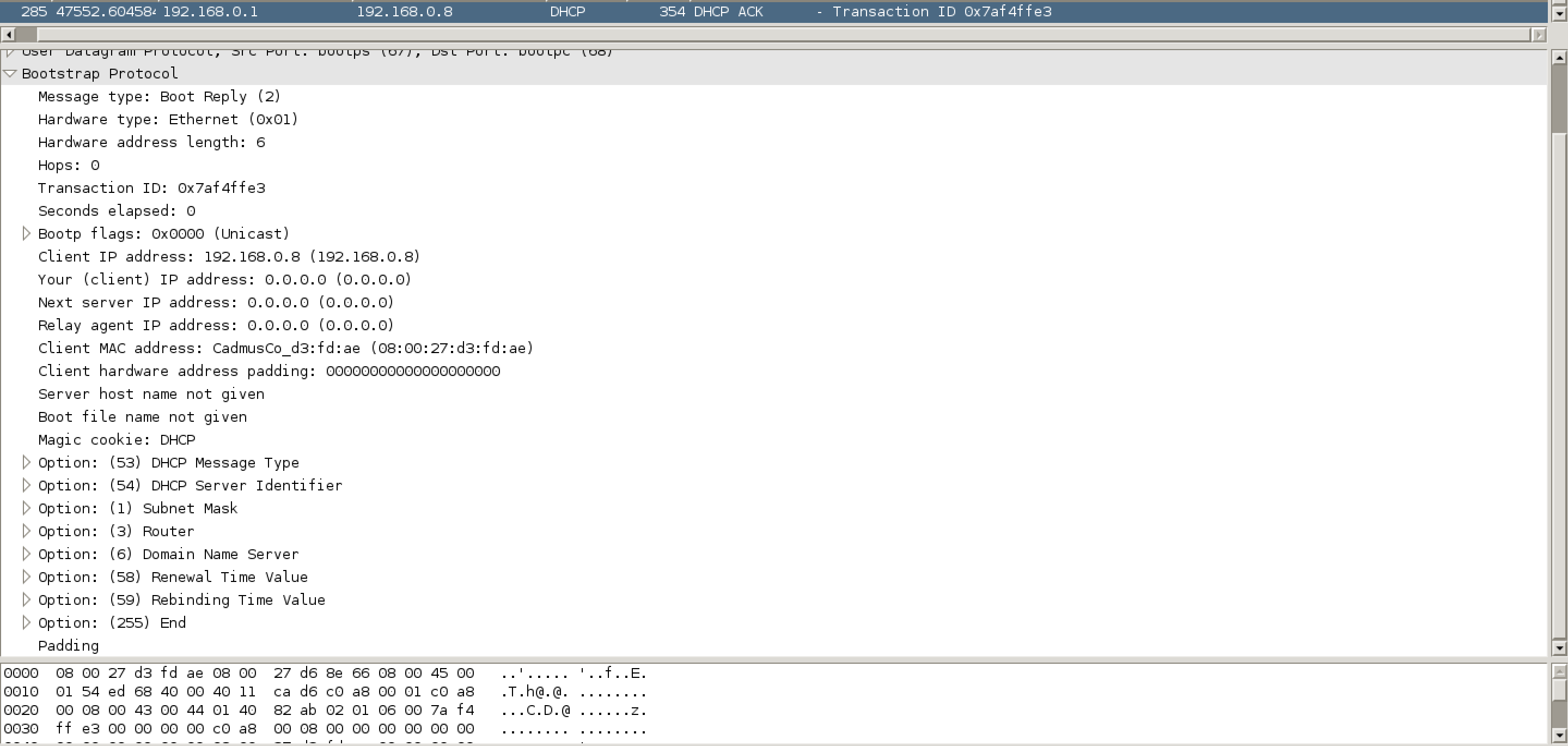
Server:



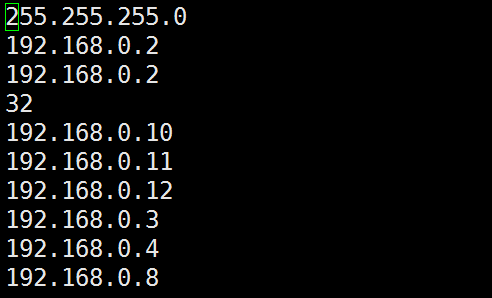
Inform:



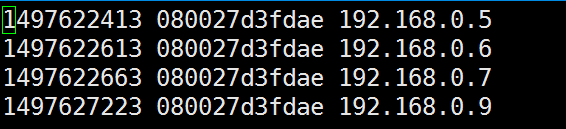
ACK(no option 51):



dhcp.config



dhcp.lease



1. User Manual

Please configure the dhcp.config file in the format of:

Submask

Router IP

DNS IP

Lease time

IPs

Use “sudo ./client --interact” to run the whole process and input renew to manually renew.

Use “sudo ./client --discover” to get new IP.

Use “sudo ./client --inform” to inform.

Use “sudo ./client –renew1-U” to manually renew.

Use “sudo ./client --release” to release.