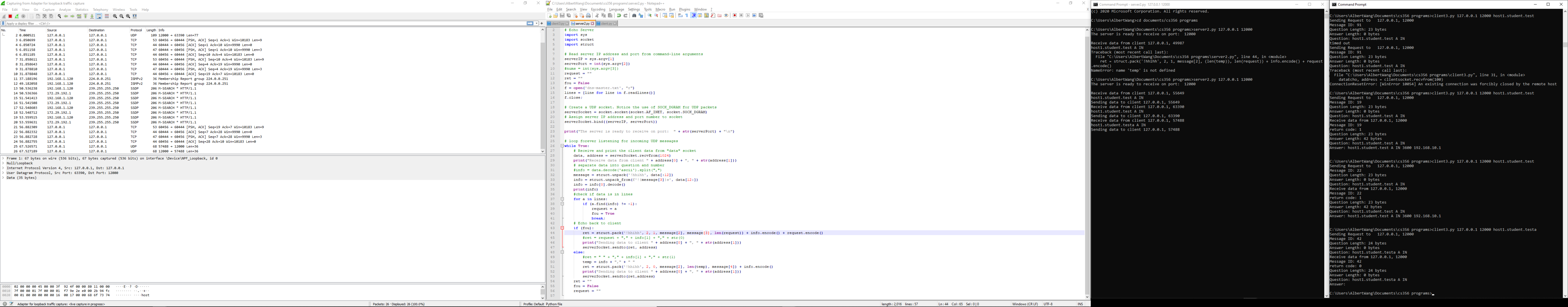

Capturing from Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time       Source          Destination      Protocol Length Info
    2 0.000521   127.0.0.1       127.0.0.1        UDP       109 12000 → 63390 Len=77
    3 6.850699   127.0.0.1       127.0.0.1        TCP        53 60456 → 60444 [PSH, ACK] Seq=1 Ack=1 Win=10183 Len=9
    4 6.850724   127.0.0.1       127.0.0.1        TCP        44 60444 → 60456 [ACK] Seq=1 Ack=10 Win=9990 Len=0
    5 6.851158   127.0.0.1       127.0.0.1        TCP        47 60444 → 60456 [PSH, ACK] Seq=1 Ack=10 Win=9990 Len=3
    6 6.851185   127.0.0.1       127.0.0.1        TCP        44 60456 → 60444 [ACK] Seq=10 Ack=4 Win=10183 Len=0
    7 31.858611  127.0.0.1       127.0.0.1        TCP        53 60456 → 60444 [PSH, ACK] Seq=10 Ack=4 Win=10183 Len=9
    8 31.858643  127.0.0.1       127.0.0.1        TCP        44 60444 → 60456 [ACK] Seq=4 Ack=19 Win=9990 Len=0
    9 31.878810  127.0.0.1       127.0.0.1        TCP        47 60444 → 60456 [PSH, ACK] Seq=4 Ack=19 Win=9990 Len=3
   10 31.878848  127.0.0.1       127.0.0.1        TCP        44 60456 → 60444 [ACK] Seq=19 Ack=7 Win=10183 Len=0
   11 37.188196  192.168.1.120   224.0.0.251      IGMPv2     36 Membership Report group 224.0.0.251
   12 49.182058  192.168.1.120   224.0.0.251      IGMPv2     36 Membership Report group 224.0.0.251
   13 50.536238  192.168.1.120   239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   14 50.536366  172.29.192.1    239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   15 51.541413  192.168.1.120   239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   16 51.541508  172.29.192.1    239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   17 52.548603  192.168.1.120   239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   18 52.548712  172.29.192.1    239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   19 53.559515  192.168.1.120   239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   20 53.559631  172.29.192.1    239.255.255.250  SSDP      206 M-SEARCH * HTTP/1.1
   21 56.882309  127.0.0.1       127.0.0.1        TCP        53 60456 → 60444 [PSH, ACK] Seq=19 Ack=7 Win=10183 Len=9
   22 56.882332  127.0.0.1       127.0.0.1        TCP        44 60444 → 60456 [ACK] Seq=7 Ack=28 Win=9990 Len=0
   23 56.882728  127.0.0.1       127.0.0.1        TCP        47 60444 → 60456 [PSH, ACK] Seq=7 Ack=28 Win=9990 Len=3
   24 56.882755  127.0.0.1       127.0.0.1        TCP        44 60456 → 60444 [ACK] Seq=28 Ack=10 Win=10183 Len=0
   25 67.526571  127.0.0.1       127.0.0.1        UDP        68 57488 → 12000 Len=36
   26 67.527189  127.0.0.1       127.0.0.1        UDP        68 12000 → 57488 Len=36
```

```
> Frame 1: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 63390, Dst Port: 12000
> Data (35 bytes)
```

```
0000  02 00 00 00 45 00 00 3f  92 4f 00 00 80 11 00 00   ····E··? ·O······
0010  7f 00 00 01 7f 00 00 01  f7 9e 2e e0 00 2b 94 fc   ········ ··.··+··
0020  00 01 00 00 00 00 00 16  00 17 00 00 68 6f 73 74   ········ ····host
```

Adapter for loopback traffic capture: <live capture in progress>
Packets: 26 · Displayed: 26 (100.0%)
Profile: Default
C:\Users\AlbertWang\Documents\cs356 programs\server2.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
client3.py
server2.py
client.py

```
 2  # Echo Server
 3  import sys
 4  import socket
 5  import struct
 6
 7  # Read server IP address and port from command-line arguments
 8  serverIP = sys.argv[1]
 9  serverPort = int(sys.argv[2])
10  #nums = int(sys.argv[3])
11  request = ""
12  ret = ""
13  fou = False
14  f = open('dns-master.txt', "r")
15  lines = [line for line in f.readlines()]
16  f.close;
17
18  # Create a UDP socket. Notice the use of SOCK_DGRAM for UDP packets
19  serverSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
20  # Assign server IP address and port number to socket
21  serverSocket.bind((serverIP, serverPort))
22
23  print("The server is ready to receive on port:  " + str(serverPort) + "\n")
24
25  # loop forever listening for incoming UDP messages
26  while True:
27      # Receive and print the client data from "data" socket
28      data, address = serverSocket.recvfrom(1024)
29      print("Receive data from client " + address[0] + ", " + str(address[1]))
30      # separate data into question and number
31      #info = data.decode('ascii').split(",")
32      message = struct.unpack('!hhihh', data[:12])
33      info = struct.unpack_from(f'!{message[3]}s', data[12:])
34      info = info[0].decode()
35      print(info)
36      #check if data is in lines
37      for a in lines:
38          if (a.find(info) != -1):
39              request = a
40              fou = True
41              break;
42      # Echo back to client
43      if (fou):
44          ret = struct.pack('!hhihh', 2, 1, message[2], message[3], len(request)) + info.encode() + request.encode()
45          #ret = request + "," + info[1] + "," + str(0)
46          print("Sending data to client " + address[0] + ", " + str(address[1]))
47          serverSocket.sendto(ret, address)
48      else:
49          #ret = " " + "," + info[1] + "," + str(1)
50          temp = info + "," + " "
51          ret = struct.pack('!hhihh', 2, 0, message[2], len(temp), message[4]) + info.encode()
52          print("Sending data to client " + address[0] + ", " + str(address[1]))
53          serverSocket.sendto(ret,address)
54      ret = ""
55      fou = False
56      request = ""
57
```

Profile: Default Python file
length : 2,016 lines : 57
Ln : 44 Col : 65 Sel : 0 | 0
Windows (CR LF) UTF-8 INS
Command Prompt - server2.py 127.0.0.1 12000

```
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\AlbertWang>cd documents/cs356 programs

C:\Users\AlbertWang\Documents\cs356 programs>server2.py 127.0.0.1 12000
The server is ready to receive on port:  12000

Receive data from client 127.0.0.1, 49987
host1.student.test A IN
Traceback (most recent call last):
  File "C:\Users\AlbertWang\Documents\cs356 programs\server2.py", line 44, in <module>
    ret = struct.pack('!hhihh', 2, 1, message[2], (len(temp)), len(request)) + info.encode() + request
.encode()
NameError: name 'temp' is not defined

C:\Users\AlbertWang\Documents\cs356 programs>server2.py 127.0.0.1 12000
The server is ready to receive on port:  12000

Receive data from client 127.0.0.1, 55649
host1.student.test A IN
Sending data to client 127.0.0.1, 55649
Receive data from client 127.0.0.1, 63390
host1.student.test A IN
Sending data to client 127.0.0.1, 63390
Receive data from client 127.0.0.1, 57488
host1.student.testa A IN
Sending data to client 127.0.0.1, 57488
```

```
C:\Users\AlbertWang\Documents\cs356 programs>client3.py 127.0.0.1 12000 host1.student.test
Sending Request to   127.0.0.1, 12000
Message ID: 91
Question Length: 23 bytes
Answer Length: 0 bytes
Question: host1.student.test A IN
timed out
Sending Request to   127.0.0.1, 12000
Message ID: 91
Question Length: 23 bytes
Answer Length: 0 bytes
Question: host1.student.test A IN
Traceback (most recent call last):
  File "C:\Users\AlbertWang\Documents\cs356 programs\client3.py", line 31, in <module>
    dataEcho, address = clientsocket.recvfrom(100)
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host

C:\Users\AlbertWang\Documents\cs356 programs>client3.py 127.0.0.1 12000 host1.student.test
Sending Request to   127.0.0.1, 12000
Message ID: 19
Question Length: 23 bytes
Answer Length: 0 bytes
Question: host1.student.test A IN
Receive data from 127.0.0.1, 12000
Message ID: 19
return code: 1
Question Length: 23 bytes
Answer Length: 42 bytes
Question: host1.student.test A IN
Answer: host1.student.test A IN 3600 192.168.10.1


C:\Users\AlbertWang\Documents\cs356 programs>client3.py 127.0.0.1 12000 host1.student.test
Sending Request to   127.0.0.1, 12000
Message ID: 22
Question Length: 23 bytes
Answer Length: 0 bytes
Question: host1.student.test A IN
Receive data from 127.0.0.1, 12000
Message ID: 22
return code: 1
Question Length: 23 bytes
Answer Length: 42 bytes
Question: host1.student.test A IN
Answer: host1.student.test A IN 3600 192.168.10.1


C:\Users\AlbertWang\Documents\cs356 programs>client3.py 127.0.0.1 12000 host1.student.testa
Sending Request to   127.0.0.1, 12000
Message ID: 42
Question Length: 24 bytes
Answer Length: 0 bytes
Question: host1.student.testa A IN
Receive data from 127.0.0.1, 12000
Message ID: 42
return code: 0
Question Length: 24 bytes
Answer Length: 0 bytes
Question: host1.student.testa A IN
Answer:

C:\Users\AlbertWang\Documents\cs356 programs>
```