Program to implement packet capturing in Python:

```
#Packet sniffer in python    #For Linux - Sniffs all incoming and outgoing packets :)
import socket, sys
from struct import *

#Convert a string of 6 characters of ethernet address into a dash separated hex string
def eth_addr (a) :
   b = "%.2x:%.2x:%.2x:%.2x:%.2x:%.2x" % (ord(a[0]) , ord(a[1]) , ord(a[2]), ord(a[3]), ord(a[4]) , ord(a[5]))
   return b

#create a AF_PACKET type raw socket (thats basically packet level)
#define ETH_P_ALL       0x0003                  /* Every packet (be careful!!!) */
try:
      s = socket.socket( socket.AF_PACKET , socket.SOCK_RAW , socket.ntohs(0x0003))
except socket.error , msg:
      print 'Socket could not be created. Error Code : ' + str(msg[0]) + ' Message ' + msg[1]
      sys.exit()

x=input("Enter the number of packets you want: ")
count=tcp=icmp=udp=others=0
# receive a packet
while count<x:
      packet = s.recvfrom(65565)
      packet = packet[0]                              #packet string from tuple

      #parse ethernet header
      eth_length = 14
      eth_header = packet[:eth_length]
      eth = unpack('!6s6sH' , eth_header)
      eth_protocol = socket.ntohs(eth[2])
      print 'Packet Number: ' + str(count)
      count = count+1
      print 'Destination MAC : ' + eth_addr(packet[0:6]) + ' Source MAC : ' + eth_addr(packet[6:12]) + '
Protocol : ' + str(eth_protocol)
```

```
#Parse IP packets, IP Protocol number = 8
if eth_protocol == 8 :
    #Parse IP header
    #take first 20 characters for the ip header
    ip_header = packet[eth_length:20+eth_length]
    #now unpack them :)
    iph = unpack('!BBHHHBBH4s4s' , ip_header)
    version_ihl = iph[0]
    version = version_ihl >> 4
    ihl = version_ihl & 0xF
    iph_length = ihl * 4
    ttl = iph[5]
    protocol = iph[6]
    s_addr = socket.inet_ntoa(iph[8]);
    d_addr = socket.inet_ntoa(iph[9]);

    print 'Version : ' + str(version) + ' IP Header Length : ' + str(ihl) + ' TTL : ' + str(ttl) + ' Protocol : ' +
str(protocol) + ' Source Address : ' + str(s_addr) + ' Destination Address : ' + str(d_addr)
```

```
root@ankitha-pilli-ide50-5571437:/home/ubuntu/workspace/CNlab# python2 UDP.py
Enter the number of packets you want: 200
Packet Number: 1
Destination MAC : 02:42:ac:11:00:1c Source MAC : 42:01:0a:f0:01:5d Protocol : 8
Version : 4 IP Header Length : 5 TTL : 63 Protocol : 6 Source Address : 10.240.1.27 Destination Address : 172.17.0.28
Source Port : 48270 Dest Port : 22 Sequence Number : 2573348230 Acknowledgement : 1074619490 TCP header length : 8

Packet Number: 2
Destination MAC : 42:01:0a:f0:01:5d Source MAC : 02:42:ac:11:00:1c Protocol : 8
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 6 Source Address : 172.17.0.28 Destination Address : 10.240.1.27
Source Port : 22 Dest Port : 48270 Sequence Number : 1074619490 Acknowledgement : 2573348230 TCP header length : 8

Packet Number: 3
```

```
#TCP protocol
if protocol == 6 :
    t = iph_length + eth_length
    tcp_header = packet[t:t+20]

    #now unpack them :)
    tcph = unpack('!HHLLBBHHH' , tcp_header)

    source_port = tcph[0]
    dest_port = tcph[1]
    sequence = tcph[2]
    acknowledgement = tcph[3]
    doff_reserved = tcph[4]
    tcph_length = doff_reserved >> 4

    print 'Source Port : ' + str(source_port) + ' Dest Port : ' + str(dest_port) + ' Sequence Number : ' + str(sequence) + ' Acknowledgement : ' + str(acknowledgement) + ' TCP header length : ' + str(tcph_length)
    tcp=tcp+1
    """h_size = eth_length + iph_length + tcph_length * 4
    data_size = len(packet) - h_size
    #get data from the packet
    data = packet[h_size:]
    print 'Data : ' + data"""
```

```
ICMP:-> Type : 0 Code : 0 Checksum : 43946

Packet Number: 499
Destination MAC : 42:01:0a:f0:01:5d Source MAC : 02:42:ac:11:00:1c Protocol : 8
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 6 Source Address : 172.17.0.28 Destination Address : 10.240.1.27
Source Port : 22 Dest Port : 48270 Sequence Number : 1076084882 Acknowledgement : 2573794278 TCP header length : 8

Total: 500  TCP: 442 UDP: 4  ICMP: 54  Others: 0
root@ankitha-pilli-ide50-5571437:/home/ubuntu/workspace/CNlab#
```

```
#ICMP Packets
elif protocol == 1 :
    u = iph_length + eth_length
    icmph_length = 4
    icmp_header = packet[u:u+4]

    #now unpack them :)
    icmph = unpack('!BBH' , icmp_header)

    icmp_type = icmph[0]
    code = icmph[1]
    checksum = icmph[2]
    print 'ICMP:-> Type : ' + str(icmp_type) + ' Code : ' + str(code) + ' Checksum : ' + str(checksum)
    icmp=icmp+1
    """h_size = eth_length + iph_length + icmph_length
    data_size = len(packet) - h_size"""
```

```
CNlab/        workspace/

Destination MAC : 02:42:ac:11:00:1c Source MAC : 42:01:0a:f0:01:5d Protocol : 8
Version : 4 IP Header Length : 5 TTL : 63 Protocol : 6 Source Address : 10.240.1.27 Destination Address : 172.17.0.28
Source Port : 48270 Dest Port : 22 Sequence Number : 2573138790 Acknowledgement : 1074314018 TCP header length : 8

Packet Number: 154
Destination MAC : 42:01:0a:f0:01:5d Source MAC : 02:42:ac:11:00:1c Protocol : 8
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 6 Source Address : 172.17.0.28 Destination Address : 10.240.1.27
Source Port : 22 Dest Port : 48270 Sequence Number : 1074318242 Acknowledgement : 2573138790 TCP header length : 8

Packet Number: 155
Destination MAC : 02:42:ac:11:00:1c Source MAC : 42:01:0a:f0:01:5d Protocol : 8
Version : 4 IP Header Length : 5 TTL : 52 Protocol : 1 Source Address : 192.30.253.113 Destination Address : 172.17.0.28
ICMP:-> Type : 0 Code : 0 Checksum : 27139

Packet Number: 156
Destination MAC : 42:01:0a:f0:01:5d Source MAC : 02:42:ac:11:00:1c Protocol : 8
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 17 Source Address : 172.17.0.28 Destination Address : 172.17.0.1
Source Port : 51888 Dest Port : 53 Length : 53 Checksum : 22662
****UDP Packet it is ****

Packet Number: 157
Destination MAC : 42:01:0a:f0:01:5d Source MAC : 02:42:ac:11:00:1c Protocol : 8
Version : 4 IP Header Length : 5 TTL : 64 Protocol : 6 Source Address : 172.17.0.28 Destination Address : 10.240.1.27
Source Port : 22 Dest Port : 48270 Sequence Number : 1074319058 Acknowledgement : 2573138790 TCP header length : 8
```

```
        #UDP packets
        elif protocol == 17 :
            u = iph_length + eth_length
            udph_length = 8
            udp_header = packet[u:u+8]

            #now unpack them :)
            udph = unpack('!HHHH' , udp_header)
            source_port = udph[0]
            dest_port = udph[1]
            length = udph[2]
            checksum = udph[3]

            print 'Source Port : ' + str(source_port) + ' Dest Port : ' + str(dest_port) + ' Length : ' +
str(length) + ' Checksum : ' + str(checksum)
            udp=udp+1
            print '****UDP Packet it is ****'

        #some other IP packet like IGMP
        else :
            print 'Protocol other than TCP/UDP/ICMP'
            others=others+1
        print
print 'Total: '+str(count)+'  TCP: '+str(tcp)+' UDP: '+str(udp)+ '  ICMP: '+str(icmp)+'  Others: '+str(others)
```