用 **Python** 获取本机网卡 **IP** 数据包

正文

这几天用到了 raw socket，用 python 写了些 demo 程序，这里记录下，也方便我以后查阅。

首先我们看一个简单的 sniffer 程序：

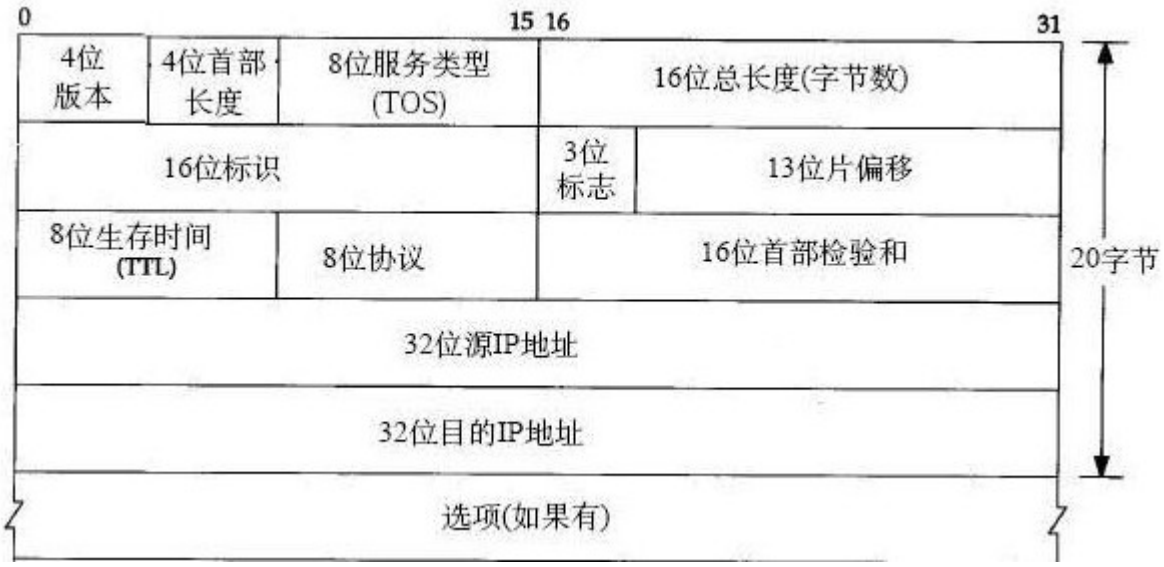
```
#! /usr/bin/python
# code for linux
import socket
#s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_UDP)
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
while True:
    print s.recvfrom(65535)
```

这里直接用 raw socket 接收数据，直接 print 操作。这个就几行代码，也没什么好解释的了，不懂的 google 下。

得到 IP 数据包后，接下来的工作就是对 IP 头进行解析，在这之前，我们先看看 RFC 中是怎么定义的（RFC791：http://www.ietf.org/rfc/rfc791.txt ）：

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

即对应的图：



从 RFC 和上图中可以看到 IP 数据包头各个字段所占的位数，我们可以根据这些定义去解析 IP 数据包头，然后根据相应的策略处理数据。

这里给出一段用 python 实现的解析 IP 头的代码（呵呵，是 demo 中的代码，只解析了前 20 个字节，这里贴出来，欢迎拍砖……）：

```
def decodeIpHeader(packet):

        mapRet = {}
        mapRet["version"] = (int(ord(packet[0])) & 0xF0)>>4
        mapRet["headerLen"] = (int(ord(packet[0])) & 0x0F)<<2
        mapRet["serviceType"] = hex(int(ord(packet[1])))
        mapRet["totalLen"] = (int(ord(packet[2]))<<8)+(int(ord(packet[3])))
        mapRet["identification"] = (int( ord(packet[4])>>8 )) + (int(
ord(packet[5])))
        mapRet["id"] = int(ord(packet[6]) & 0xE0)>>5
        mapRet["fragOff"] = int(ord(packet[6]) & 0x1F)<<8 +
int(ord(packet[7]))
        mapRet["ttl"] = int(ord(packet[8]))
        mapRet["protocol"] = int(ord(packet[9]))
        mapRet["checkSum"] = int(ord(packet[10])<<8)+int(ord(packet[11]))
        mapRet["srcaddr"] = "%d.%d.%d.%d" %
(int(ord(packet[12])),int(ord(packet[13])),int(ord(packet[14])),
int(ord(packet[15])))
        mapRet["dstaddr"] = "%d.%d.%d.%d" %
(int(ord(packet[16])),int(ord(packet[17])),int(ord(packet[18])),
int(ord(packet[19])))
        return mapRet
```

调用代码：

```
proto = socket.getprotobyname('tcp') # only tcp
sock = socket.socket(socket.AF_INET, socket.SOCK_RAW, proto)

while True:
        packet = sock.recvfrom(65535)[0]
```

```
        if len(packet) == 0:
                sck.close()
        else:
                #print str(packet)
                mapIpTmp = decodeIpHeader(packet)
                for k,v in mapIpTmp.items():
                        print k,"\t:\t",v

        print ""
```

附录

1、获取 TCP 数据

```
#!/usr/bin/python
# code for linux
'''
        File        : rawDataTcp.py
        Author      : Mike
        E-Mail      : Mike_Zhang@live.com
'''
import socket

def decodeIpHeader(packet):
        mapRet = {}
        mapRet["version"] = (int(ord(packet[0])) & 0xF0)>>4
        mapRet["headerLen"] = (int(ord(packet[0])) & 0x0F)<<2
        mapRet["serviceType"] = hex(int(ord(packet[1])))
        mapRet["totalLen"] = (int(ord(packet[2])<<8))+(int(ord(packet[3])))
        mapRet["identification"] = (int( ord(packet[4])>>8 )) + (int(
ord(packet[5])))
        mapRet["id"] = int(ord(packet[6]) & 0xE0)>>5
        mapRet["fragOff"] = int(ord(packet[6]) & 0x1F)<<8 +
int(ord(packet[7]))
        mapRet["ttl"] = int(ord(packet[8]))
        mapRet["protocol"] = int(ord(packet[9]))
        mapRet["checkSum"] = int(ord(packet[10])<<8)+int(ord(packet[11]))
        mapRet["srcaddr"] = "%d.%d.%d.%d" %
(int(ord(packet[12])),int(ord(packet[13])),int(ord(packet[14])),
int(ord(packet[15])))
        mapRet["dstaddr"] = "%d.%d.%d.%d" %
(int(ord(packet[16])),int(ord(packet[17])),int(ord(packet[18])),
int(ord(packet[19])))
        return mapRet

proto = socket.getprotobyname('tcp') # only tcp
sock = socket.socket(socket.AF_INET, socket.SOCK_RAW, proto)

while True:
        packet = sock.recvfrom(65535)[0]
```

```
        if len(packet) == 0:
                sck.close()
        else:
                #print str(packet)
                mapIpTmp = decodeIpHeader(packet)
                for k,v in mapIpTmp.items():
                        print k,"\t:\t",v

        print ""
```

2、获取所有数据

```
#!/usr/bin/python
# code for linux
'''
        File      : rawDataAll.py
        Author    : Mike
        E-Mail    : Mike_Zhang@live.com
'''
import socket

def decodeIpHeader(packet):
        mapRet = {}
        mapRet["version"] = (int(ord(packet[0])) & 0xF0)>>4
        mapRet["headerLen"] = (int(ord(packet[0])) & 0x0F)<<2
        mapRet["serviceType"] = hex(int(ord(packet[1])))
        mapRet["totalLen"] = (int(ord(packet[2])<<8))+(int(ord(packet[3])))
        mapRet["identification"] = (int( ord(packet[4])>>8 )) + (int(
ord(packet[5])))
        mapRet["id"] = int(ord(packet[6]) & 0xE0)>>5
        mapRet["fragOff"] = int(ord(packet[6]) & 0x1F)<<8 +
int(ord(packet[7]))
        mapRet["ttl"] = int(ord(packet[8]))
        mapRet["protocol"] = int(ord(packet[9]))
        mapRet["checkSum"] = int(ord(packet[10])<<8)+int(ord(packet[11]))
        mapRet["srcaddr"] = "%d.%d.%d.%d" %
(int(ord(packet[12])),int(ord(packet[13])),int(ord(packet[14])),
int(ord(packet[15])))
        mapRet["dstaddr"] = "%d.%d.%d.%d" %
(int(ord(packet[16])),int(ord(packet[17])),int(ord(packet[18])),
int(ord(packet[19])))
        return mapRet

proto = socket.ntohs(0x0003) # Every packet
sock = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, proto)

while True:
        packet = sock.recvfrom(65535)[0]
        if len(packet) == 0:
                sck.close()
```

```
        else:
                #print str(packet)
                mapIpTmp = decodeIpHeader(packet[14:]) # decode Ip data
                for k,v in mapIpTmp.items():
                        print k,"\t:\t",v
        print ""
```

3、Windows 下的 demo

```
import socket

# the public network interface
HOST = socket.gethostbyname(socket.gethostname())

# create a raw socket and bind it to the public interface
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
s.bind((HOST, 0))

# Include IP headers
s.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

while True:
        # receive all packages
        s.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)
        # receive a package
        print s.recvfrom(65565)

# disabled promiscuous mode
s.ioctl(socket.SIO_RCVALL, socket.RCVALL_OFF)
```