

Entrée [2]: **from scapy.all import ***

```
trames=rdpcap("Wireshark/ping6-display.pcapng")
print("Voici les trames capturées :\n")
trames.show()
```

Voici les trames capturées :

```
0000 Ether / IPv6 / ICMPv6ND_NS / ICMPv6 Neighbor Discovery Option
- Source Link-Layer Address 64:00:6a:6a:c4:01
0001 Ether / IPv6 / ICMPv6ND_NA / ICMPv6 Neighbor Discovery Option
- Destination Link-Layer Address 00:13:1a:a3:e1:18
0002 Ether / IPv6 / ICMPv6 Echo Request (id: 0x4 seq: 0x1)
0003 Ether / IPv6 / ICMPv6 Echo Reply (id: 0x4 seq: 0x1)
0004 Ether / IPv6 / ICMPv6 Echo Request (id: 0x4 seq: 0x2)
0005 Ether / IPv6 / ICMPv6 Echo Reply (id: 0x4 seq: 0x2)
0006 Ether / IPv6 / ICMPv6 Echo Request (id: 0x4 seq: 0x3)
0007 Ether / IPv6 / ICMPv6 Echo Reply (id: 0x4 seq: 0x3)
0008 Ether / IPv6 / ICMPv6 Echo Request (id: 0x4 seq: 0x4)
0009 Ether / IPv6 / ICMPv6 Echo Reply (id: 0x4 seq: 0x4)
0010 Ether / IPv6 / ICMPv6 Echo Request (id: 0x4 seq: 0x5)
0011 Ether / IPv6 / ICMPv6 Echo Reply (id: 0x4 seq: 0x5)
0012 Ether / IPv6 / ICMPv6 Echo Request (id: 0x4 seq: 0x6)
0013 Ether / IPv6 / ICMPv6 Echo Reply (id: 0x4 seq: 0x6)
```

Entrée [3]: trames[0]

```
Out[3]: <Ether  dst=33:33:ff:00:00:01 src=64:00:6a:6a:c4:01 type=IPv6 |<IPv
6  version=6 tc=0 fl=0 plen=32 nh=ICMPv6 hlim=255 src=2001:660:670
1:30cc:84fc:c335:133c:f204 dst=ff02::1:ff00:1 |<ICMPv6ND_NS  type=N
eighbor Solicitation code=0 cksum=0x1d60 res=0 tgt=2001:660:6701:30
cc::1 |<ICMPv6NDOptSrcLLAddr  type=1 len=1 lladdr=64:00:6a:6a:c4:01
|>>>>
```

Entrée [4]: trames[0][0]
trames[0][Ether]

```
Out[4]: <Ether  dst=33:33:ff:00:00:01 src=64:00:6a:6a:c4:01 type=IPv6 |<IPv
6  version=6 tc=0 fl=0 plen=32 nh=ICMPv6 hlim=255 src=2001:660:670
1:30cc:84fc:c335:133c:f204 dst=ff02::1:ff00:1 |<ICMPv6ND_NS  type=N
eighbor Solicitation code=0 cksum=0x1d60 res=0 tgt=2001:660:6701:30
cc::1 |<ICMPv6NDOptSrcLLAddr  type=1 len=1 lladdr=64:00:6a:6a:c4:01
|>>>>
```

Entrée [5]: trames[0][1]
trames[0][IPv6]
Affiche la même chose: le paquet IPv6

```
Out[5]: <IPv6  version=6 tc=0 fl=0 plen=32 nh=ICMPv6 hlim=255 src=2001:660:
6701:30cc:84fc:c335:133c:f204 dst=ff02::1:ff00:1 |<ICMPv6ND_NS  typ
e=Neighbor Solicitation code=0 cksum=0x1d60 res=0 tgt=2001:660:670
1:30cc::1 |<ICMPv6NDOptSrcLLAddr  type=1 len=1 lladdr=64:00:6a:6a:c
4:01 |>>>
```

```
Entrée [6]: trames[0][2]
            trames[0][ICMPv6ND_NS]
            # Affiche la même chose: le paquet ICMP
```

```
Out[6]: <ICMPv6ND_NS type=Neighbor Solicitation code=0 cksum=0x1d60 res=0
        tgt=2001:660:6701:30cc::1 |<ICMPv6ND0ptSrcLLAddr type=1 len=1 llad
        dr=64:00:6a:6a:c4:01 |>>
```

```
Entrée [7]: trames[0][3]
            trames[0][ICMPv6ND0ptSrcLLAddr]
            # Affiche la même chose: la sous-partie du paquet ICMP, contenant l'
```

```
Out[7]: <ICMPv6ND0ptSrcLLAddr type=1 len=1 lladdr=64:00:6a:6a:c4:01 |>
```

```
Entrée [8]: trames[0][ICMPv6ND0ptSrcLLAddr].lladdr
```

```
Out[8]: '64:00:6a:6a:c4:01'
```

```
Entrée [9]: trames[0][Ether].lladdr
            trames[0][IPv6].lladdr
            trames[0][ICMPv6ND0ptSrcLLAddr]
```

```
Out[9]: <ICMPv6ND0ptSrcLLAddr type=1 len=1 lladdr=64:00:6a:6a:c4:01 |>
```

```
Entrée [10]: trames[0].dst
             trames[0][Ether].dst
```

```
Out[10]: '33:33:ff:00:00:01'
```

```
Entrée [11]: trames[0][IPv6].dst
```

```
Out[11]: 'ff02::1:ff00:1'
```

```
Entrée [13]: #L'ensemble des trames capturées dans le fichier pcap sont chargées

            trames=rdpcap("Wireshark/ping6-display.pcapng") #<- variable trame

            for trame in trames: # on fait une boucle pour les traiter une par une
                if(trame[0][1].version)==6: # on ne prend que les paquets en IPv6
                    if (trame[0][1].nh)==58: # on ne prend que les paquets dont le
                                                protocole est l'IPv6

                        if (trame[0][2].type)==135: # on ne prend que les paquets
                                                        ICMPv6

                            print(f"Ethernet: MAC Source : {trame[0][0].src}")
                            print(f"Ethernet: MAC Destination : {trame[0][0].dst}")
                            print(f"IPv6 : IP Source : {trame[0][1].src}")
                            print(f"IPv6 : IP Destination : {trame[0][1].dst}")
                            print(f"ICMPv6 : IP Target : {trame[0][2].tgt}")
                            print(f"ICMPv6 : MAC Requested : {trame[0][3].lladdr}")
```

```
Ethernet: MAC Source : 64:00:6a:6a:c4:01
Ethernet: MAC Destination : 33:33:ff:00:00:01
IPv6 : IP Source : 2001:660:6701:30cc:84fc:c335:133c:f204
IPv6 : IP Destination : ff02::1:ff00:1
ICMPv6 : IP Target : 2001:660:6701:30cc::1
ICMPv6 : MAC Requested : 64:00:6a:6a:c4:01
```

```
Entrée [14]: from scapy.all import *

#L'ensemble des trames capturées dans le fichier pcap sont chargées

trames=rdpcap("Wireshark/ping6-total.pcapng")
for trame in trames: # on fait une boucle pour les traiter une par une
    try:
        trame[0][0].type # les trames 802.3 n'ont pas de champ type
    except:
        continue
    if(trame[0][0].type==34525): # on prend les trames contenant de l'IPv6
        if(trame[0][1].version)==6: # on ne prend que les paquets en IPv6
            if (trame[0][1].nh)==58: # on ne prend que les paquets en ICMPv6
                if (trame[0][2].type)==135: # on ne prend que les paquets en Echo Request
                    print(f"Ethernet: MAC Source : {trame[0][0].src}")
                    print(f"Ethernet: MAC Destination : {trame[0][0].dst}")
                    print(f"IPv6 : IP Source : {trame[0][1].src}")
                    print(f"IPv6 : IP Destination : {trame[0][1].dst}")
                    print(f"ICMPv6 : IP Target : {trame[0][2].tgt}")
                    print(f"ICMPv6 : MAC Requested : {trame[0][3].src}")

Ethernet: MAC Source : 64:00:6a:6a:c4:01
Ethernet: MAC Destination : 33:33:ff:00:00:01
IPv6 : IP Source : 2001:660:6701:30cc:84fc:c335:133c:f204
IPv6 : IP Destination : ff02::1:ff00:1
ICMPv6 : IP Target : 2001:660:6701:30cc::1
ICMPv6 : MAC Requested : 64:00:6a:6a:c4:01
```

Entrée [1]:

```

ICMPv6_types={ 128 : 'Echo-Request', 129 : 'Echo-Reply', 135 : 'Neigh

def print_icmpv6 (trame) :
    print(trame.summary())
    type=trame[2].type
    if (type==135 or type==136):
        print(f"TYPE PACKET ICMP : {ICMPv6_types[type]}")
        print(f"Ethernet: MAC Source : {trame[0].src}")
        print(f"Ethernet: MAC Destination : {trame[0].dst}")
        print(f"IPv6 : IP Source : {trame[1].src}")
        print(f"IPv6 : IP Destination : {trame[1].dst}")
        print(f"ICMPv6 : IP Target : {trame[2].tgt}")
        print(f"ICMPv6 : MAC Requested : {trame[3].lladdr}")
        print ("\n")
    else:
        print(f"TYPE PACKET ICMP : {ICMPv6_types[type]}")
        print(f"Ethernet: MAC Source : {trame[0].src}")
        print(f"Ethernet: MAC Destination : {trame[0].dst}")
        print(f"IPv6 : IP Source : {trame[1].src}")
        print(f"IPv6 : IP Destination : {trame[1].dst}")
        print ("\n")

carte=conf.iface

print(f"On commence le 'sniffing' sur la carte {carte}:")
print("\n")
sniff(filter="ip6 proto 58", prn=print_icmpv6, store=0, iface=carte,

-----
-----
NameError                                Traceback (most recent ca
ll last)
/tmp/ipykernel_4007/924984025.py in <module>
     21         print ("\n")
     22
--> 23 carte=conf.iface
     24
     25 print(f"On commence le 'sniffing' sur la carte {carte}:")

NameError: name 'conf' is not defined

```

Entrée []: