

```
In [1]: from scapy.all import *

print(f"La version de Scapy est {conf.version}.")
print(f"\nL'interface par défaut utilisée pour l'émission et la réception")
print(f"\nLa table de routage utilisée est : \n {conf.route}.")
print(f'\nLa passerelle par défaut est :', conf.route.route("0.0.0.0")[2])
```

La version de Scapy est 2.5.0.

L'interface par défaut utilisée pour l'émission et la réception des paquets est ens3.

La table de routage utilisée est :

Network	Netmask	Gateway	Iface	Output IP	Metric
0.0.0.0	0.0.0.0	192.168.40.1	ens3	192.168.40.2	20100
127.0.0.0	255.0.0.0	0.0.0.0	lo	127.0.0.1	1
169.254.0.0	255.255.0.0	0.0.0.0	ens3	192.168.40.2	1000
192.168.40.0	255.255.255.0	0.0.0.0	ens3	192.168.40.2	100

La passerelle par défaut est : 192.168.40.1

```
In [2]: ls(IP)
```

```
version      : BitField  (4 bits)          = ('4')
ihl          : BitField  (4 bits)          = ('None')
tos          : XByteField                = ('0')
len          : ShortField                 = ('None')
id           : ShortField                 = ('1')
flags        : FlagsField                 = ('<Flag 0 (>')
frag         : BitField  (13 bits)        = ('0')
ttl          : ByteField                  = ('64')
proto        : ByteEnumField              = ('0')
chksum       : XShortField                = ('None')
src          : SourceIPField              = ('None')
dst          : DestIPField                = ('None')
options      : PacketListField            = ('[]')
```

```
In [3]: paquet1=IP()/UDP()
paquet1.summary()
```

```
Out[3]: 'IP / UDP 127.0.0.1:domain > 127.0.0.1:domain'
```

```
In [4]: paquet1.show()
```

```
####[ IP ]####
version    = 4
ihl        = None
tos        = 0x0
len        = None
id         = 1
flags      =
frag       = 0
ttl        = 64
proto      = udp
chksum     = None
src        = 127.0.0.1
dst        = 127.0.0.1
\options   \
####[ UDP ]####
sport      = domain
dport      = domain
len        = None
chksum     = None
```

```
In [5]: IPs='192.168.1.1'
        IPd='192.168.1.254'
        paquet2=IP(src=IPs, dst=IPd)/UDP()
        paquet2.show()
        paquet2.show2()
```

```

####[ IP ]####
version    = 4
ihl        = None
tos        = 0x0
len        = None
id         = 1
flags      =
frag       = 0
ttl        = 64
proto      = udp
chksum     = None
src        = 192.168.1.1
dst        = 192.168.1.254
\options   \
####[ UDP ]####
sport      = domain
dport      = domain
len        = None
chksum     = None

####[ IP ]####
version    = 4
ihl        = 5
tos        = 0x0
len        = 28
id         = 1
flags      =
frag       = 0
ttl        = 64
proto      = udp
chksum     = 0xf680
src        = 192.168.1.1
dst        = 192.168.1.254
\options   \
####[ UDP ]####
sport      = domain
dport      = domain
len        = 8
chksum     = 0x7b24

```

In [6]:

```

MACs='11:22:33:44:55:66'
MACd='00:0A:1F:3B:4E:64'
IPs='192.168.1.1'
IPd='www.univ-pau.fr'
frm=Ether(src=MACs, dst=MACd)/IP(src=IPs, dst=IPd)/TCP(flags='SA')/"C es
frm.show()
frm.show2()

```

```

####[ Ethernet ]###
  dst      = 00:0A:1F:3B:4E:64
  src      = 11:22:33:44:55:66
  type     = IPv4
####[ IP ]###
  version  = 4
  ihl      = None
  tos      = 0x0
  len      = None
  id       = 1
  flags    =
  frag     = 0
  ttl      = 64
  proto    = tcp
  chksum   = None
  src      = 192.168.1.1
  dst      = Net("www.univ-pau.fr/32")
  \options \
####[ TCP ]###
  sport    = ftp_data
  dport    = http
  seq      = 0
  ack      = 0
  dataofs  = None
  reserved = 0
  flags    = SA
  window   = 8192
  chksum   = None
  urgptr   = 0
  options  = ''
####[ Raw ]###
  load     = 'C est vraiment bien Scapy'

####[ Ethernet ]###
  dst      = 00:0a:1f:3b:4e:64
  src      = 11:22:33:44:55:66
  type     = IPv4
####[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 65
  id       = 1
  flags    =
  frag     = 0
  ttl      = 64
  proto    = tcp
  chksum   = 0x59de
  src      = 192.168.1.1
  dst      = 194.167.156.135
  \options \
####[ TCP ]###
  sport    = ftp_data
  dport    = http
  seq      = 0
  ack      = 0
  dataofs  = 5
  reserved = 0
  flags    = SA
  window   = 8192
  chksum   = 0x7857
  urgptr   = 0
  options  = []

```

```
####[ Raw ]####
load = 'C est vraiment bien Scapy'
```

```
In [7]: trames=rdpcap('Wireshark/Ping_Google.pcapng')
print("La capture comprend les paquets suivants :\n")
trames.summary()
```

La capture comprend les paquets suivants :

```
Ether / IP / ICMP 192.168.1.48 > 8.8.8.8 echo-request 0 / Raw
Ether / IP / ICMP 8.8.8.8 > 192.168.1.48 echo-reply 0 / Raw
Ether / IP / ICMP 192.168.1.48 > 8.8.8.8 echo-request 0 / Raw
Ether / IP / ICMP 8.8.8.8 > 192.168.1.48 echo-reply 0 / Raw
```

```
In [9]: RT={'Siami' : 'chef de departement', 'Bascou' : 'enseignant', 'Rivenq' : 'secretaire'}
RT['Siami']
```

```
Out[9]: 'chef de departement'
```

```
In [10]: RT['Gallon']='enseignant'
[print(f"Mr {nom} est {RT[nom]} au département R&T de l'IUT de Mont de Marsan") for nom in RT]
```

```
Mr Siami est chef de departement au département R&T de l'IUT de Mont de Marsan
Mr Bascou est enseignant au département R&T de l'IUT de Mont de Marsan
Mr Rivenq est secretaire au département R&T de l'IUT de Mont de Marsan
Mr Gallon est enseignant au département R&T de l'IUT de Mont de Marsan
```

```
Out[10]: [None, None, None, None]
```

```
In [11]: RT={'Siami' : {'fonction' : 'chef de departement', 'année de recutement' : 'Enseignant en '},
'Bascou' : {'fonction' : 'DE 1ere annee de BUT', 'année de recutement' : 'Enseignant en '},
'Rivenq' : {'fonction' : 'secretaire', 'année de recutement' : 'secretaire'},
'Gallon' : {'fonction' : 'enseignant', 'année de recutement' : 'enseignant'}}
RT['Bascou']['fonction']
```

```
Out[11]: 'DE 1ere annee de BUT'
```

```
In [8]: print('La premiere trame est la suivante :')
trames[0]
```

La premiere trame est la suivante :

```
Out[8]: <Ether dst=f4:ca:e5:5d:b5:c8 src=8c:85:90:c5:71:56 type=IPv4 |<IP version=4 ihl=5 tos=0x0 len=84 id=3173 flags= frag=0 ttl=64 proto=icmp chksum=0x9c5c src=192.168.1.48 dst=8.8.8.8 |<ICMP type=echo-request code=0 checksum=0xf4cc id=0x5662 seq=0x0 unused='' |<Raw load='a\\xc0\\xbd\\xd2\\x00\\x05\\xa25\\x08\\t\\n\\x0b\\x0c\\r\\x0e\\x0f\\x10\\x11\\x12\\x13\\x14\\x15\\x16\\x17\\x18\\x19\\x1a\\x1b\\x1c\\x1d\\x1e\\x1f !"#%&\\'()*+,-./01234567' |>>>>
```

```
In [9]: from scapy.all import *

trames[2]['IP']
```

```
Out[9]: <IP version=4 ihl=5 tos=0x0 len=84 id=7714 flags= frag=0 ttl=64 proto=ic
mp chksum=0x8a9f src=192.168.1.48 dst=8.8.8.8 |<ICMP type=echo-request c
ode=0 chksum=0xe086 id=0x5662 seq=0x1 unused='' |<Raw load='a\\xc0\\xb
d\\xd3\\x00\\x05\\xb6y\\x08\\t\\n\\x0b\\x0c\\r\\x0e\\x0f\\x10\\x11\\x12\\x13\\x14\\x15\\x1
6\\x17\\x18\\x19\\x1a\\x1b\\x1c\\x1d\\x1e\\x1f !"#%&\\'()*+,-./01234567' |>>>
```

```
In [10]: trames[2][Raw].load
```

```
Out[10]: b'a\\xc0\\xbd\\xd3\\x00\\x05\\xb6y\\x08\\t\\n\\x0b\\x0c\\r\\x0e\\x0f\\x10\\x11\\x12\\x13\\x1
4\\x15\\x16\\x17\\x18\\x19\\x1a\\x1b\\x1c\\x1d\\x1e\\x1f !"#%&\\'()*+,-./01234567'
```

```
In [11]: trames[2][Raw].load.split(sep=None)
```

```
Out[11]: [b'a\\xc0\\xbd\\xd3\\x00\\x05\\xb6y\\x08',
b'\\x0e\\x0f\\x10\\x11\\x12\\x13\\x14\\x15\\x16\\x17\\x18\\x19\\x1a\\x1b\\x1c\\x1d\\x1e\\x
1f',
b'!"#%&\\'()*+,-./01234567']
```

```
In [12]: trames[2][Raw].load.split(sep=None)[2]
```

```
Out[12]: b'!"#%&\\'()*+,-./01234567'
```

```
In [13]: trames[2][Raw].load.split(sep=None)[2].decode('UTF8')
```

```
Out[13]: '!"#%&\\'()*+,-./01234567'
```

```
In [2]: ping=IP(dst='10.2.12.4')/ICMP()/"On peut ajouter ici des données à émett
sendp(ping)
```

Sent 1 packets..

```
In [12]: trameping=Ether()/IP(dst='10.2.12.4')/ICMP()/"On peut ajouter ici des do
sendp(trameping, iface="ens3")
```

.  
Sent 1 packets.

```
In [13]: ping=IP(dst='10.2.12.4')/ICMP()/"On peut ajouter ici des données à émett
sr(ping)
```

Begin emission:  
Finished sending 1 packets.  
\*

Received 1 packets, got 1 answers, remaining 0 packets

```
Out[13]: (<Results: TCP:0 UDP:0 ICMP:1 Other:0>,
<Unanswered: TCP:0 UDP:0 ICMP:0 Other:0>)
```

```
In [14]: ping=IP(dst='10.2.12.4')/ICMP()/"On peut ajouter ici des données à émett
ok, nonok = srloop(ping, count=10, inter=1)
```

```

RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw
RECV 1: IP / ICMP 10.2.12.4 > 192.168.40.2 echo-reply 0 / Raw

```

Sent 10 packets, received 10 packets. 100.0% hits.

In [15]: `ok[0]`

Out[15]: QueryAnswer(query=<IP frag=0 proto=icmp dst=10.2.12.4 |<ICMP |<Raw load='On peut ajouter ici des données à émettre avec le ping' |>>>, answer=<IP version=4 ihl=5 tos=0x0 len=85 id=12393 flags= frag=0 ttl=63 proto=icmp chksum=0x4c8f src=10.2.12.4 dst=192.168.40.2 |<ICMP type=echo-reply code=0 chksum=0x2929 id=0x0 seq=0x0 unused='' |<Raw load='On peut ajouter ici des données à émettre avec le ping' |>>>)

In [16]: `ok[0][0]`

Out[16]: <IP frag=0 proto=icmp dst=10.2.12.4 |<ICMP |<Raw load='On peut ajouter ici des données à émettre avec le ping' |>>>

In [17]: `ok[0][1]`

Out[17]: <IP version=4 ihl=5 tos=0x0 len=85 id=12393 flags= frag=0 ttl=63 proto=icmp chksum=0x4c8f src=10.2.12.4 dst=192.168.40.2 |<ICMP type=echo-reply code=0 chksum=0x2929 id=0x0 seq=0x0 unused='' |<Raw load='On peut ajouter ici des données à émettre avec le ping' |>>>

In [20]: `ICMP_types={ 0 : 'Echo-Reply', 3 : 'Destination Unreachable', 8 : 'Echo'}`

```

def print_icmp (packet) :
    type=packet[ICMP].type
    ips=packet[IP].src
    ipd=packet[IP].dst
    if ips==iface_ip :
        print(f"Emission d'un paquet ICMP {ICMP_types[type]} vers {ipd}")
    else :
        print(f"Réception d'un paquet ICMP {ICMP_types[type]} en provenant de {ips}")

iface_ip=get_if_addr(conf.iface)

sniff(filter="icmp", prn=print_icmp, store=0, iface='ens3', count=4)

```

```

Emission d'un paquet ICMP Echo vers 8.8.8.8
Emission d'un paquet ICMP Echo vers 8.8.8.8
Emission d'un paquet ICMP Echo vers 8.8.8.8
Emission d'un paquet ICMP Echo vers 8.8.8.8

```

Out[20]: <Sniffed: TCP:0 UDP:0 ICMP:0 Other:0>

In [ ]: