Compte rendu TP2 Mode Non Connecté

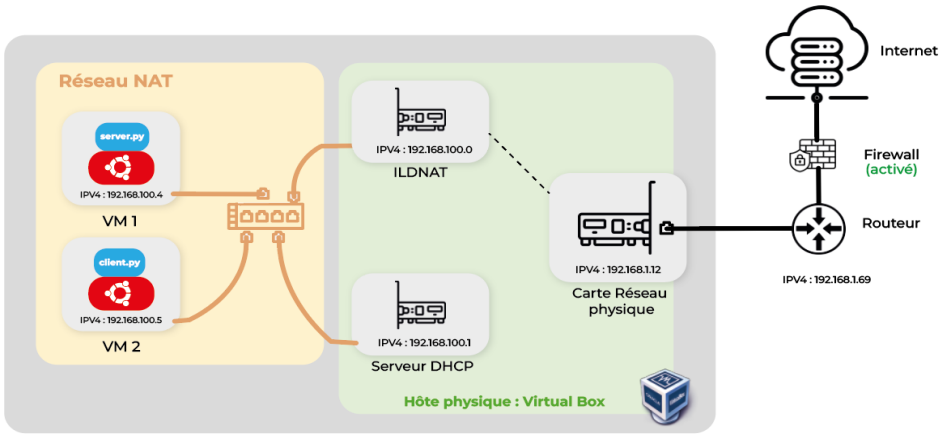
Nom : Ibrahim Lahlou

0. Programmation de Socket

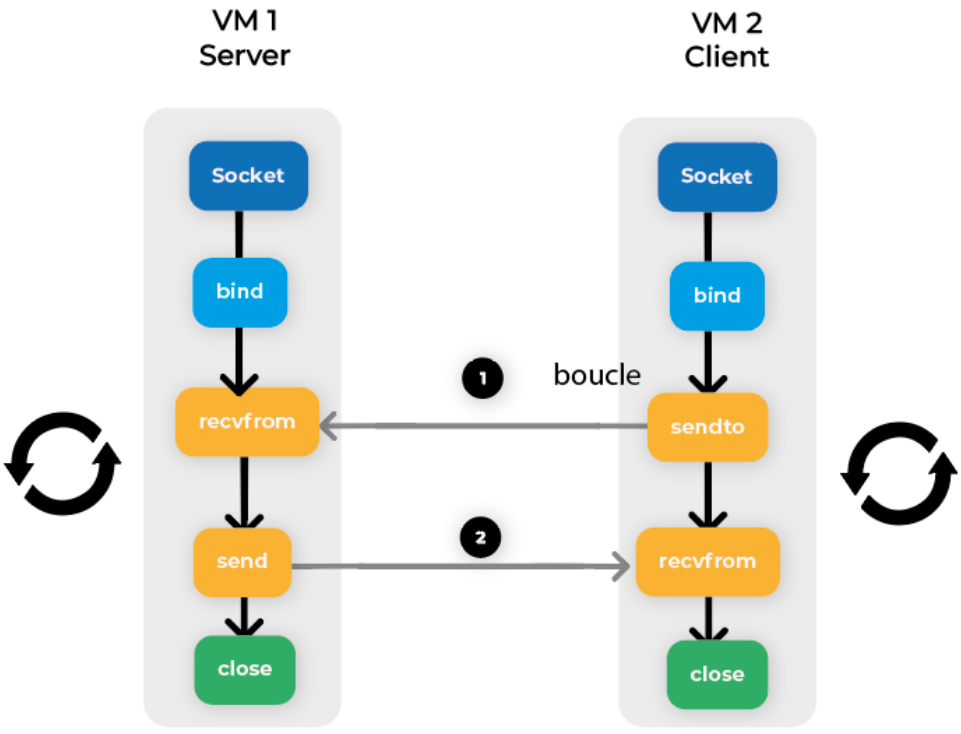
Création d'un programme communicant des sockets avec python entre les deux machine VM1 et VM2 des instances d'ubuntu server en **mode non connecté**

1 . Architecture réseau de cette manipulation

- **Réseau NAT**



2. Architecture Client-Server



1 et 2 Transfère de donnée entre client et server

Au mode non Connecté, on spécifie un socket de datagramme .les données sont envoyées sous forme de paquets indépendants sans qu'une connexion persistante soit établie au préalable.

3. Programmation

(Voir le compte rendu du TP1)

3.1. Programmation coté client

```
  GNU nano 6.2                                    client2.py
# encoding : utf-8
# author : Ibrahim Lahlou
# code : client2.py
import socket as skt
import time

host='192.168.100.4'
port=6999

receiver=(host,port)

print("",end="\n\n")
print("                  TP1 CLI")
print("_____________________________________________")
with skt.socket(skt.AF_INET,skt.SOCK_DGRAM) as s:
        while True :
                # Sending Data
                for i in range(3):
                        msg=input("client@vm2:6999$ ")
                        s.sendto(msg.encode(),receiver)

                # Receiving Data
                data,addr = s.recvfrom(1024)
                print("Server : ",data.decode())

                if msg == 'exit':
                        break
        s.close()
```

3.2 Programmation coté serveur

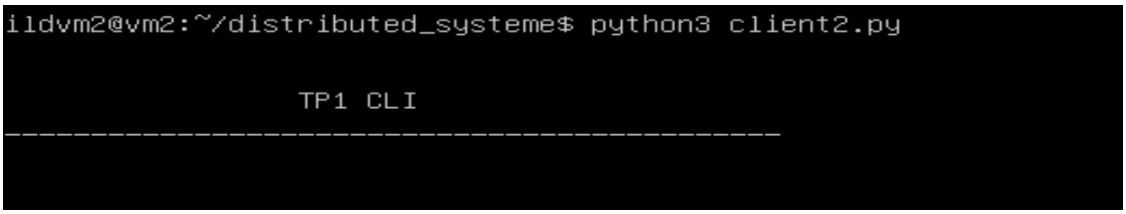
```
  GNU nano 6.2                                    server2.py
# encoding : utf-8
# author : Ibrahim Lahlou
# code : server.py
import socket as skt

port= 6999
print("",end="\n\n")
print("                    TP1 Server ")
print("_____________________________________________________")
with skt.socket(skt.AF_INET,skt.SOCK_DGRAM) as s:
        s.bind(('',port))
        while True  :
                for i in range(3):
                        # Receiving data
                        data,addr=s.recvfrom(1024)
                        print("Client : ",data.decode())

                # Sending data
                msg=input(f'server@vm1:{port}$ ')
                s.sendto(msg.encode(),addr)
                r=input("")
                if msg == "exit" :
                        break
        s.close()
```

SOCK_DGRAM Fournit des datagrammes, qui sont des messages sans connexion d'une longueur maximale fixe.

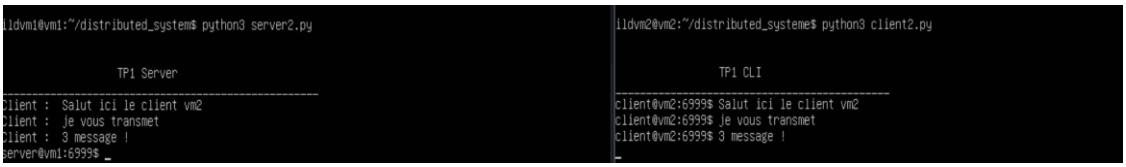
Ce type de socket est généralement utilisé pour des messages courts, tels qu'un serveur de noms ou un serveur de temps, car l'ordre et la fiabilité de la livraison des messages ne sont pas garantis.

5. Éxecution du Programme

1. Lancement des deux programme le serveur puis le client



2. Transmission des donnée depuis le client vm2



3. retoure du serveur

4. Le client stop l'envoie de message alors que le serveur n'est pas consient de cette fermetture

Cela prouve la non connectivité de mode de communication