

Chapitre 1

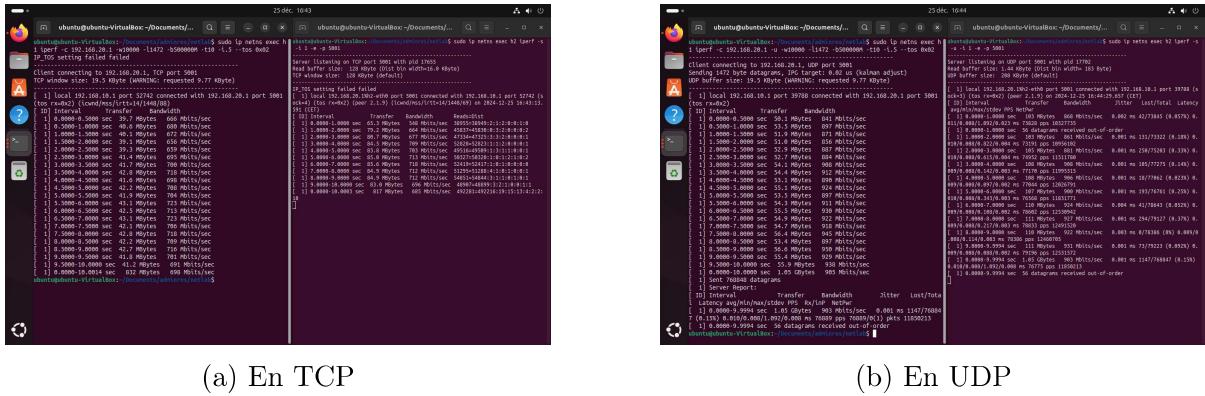
Qos et traffic entrant

1.1 Interface ifb

```
1 #!/bin/bash
2 # Supprime un module qui se nomme "ifb"
3 modprobe -r ifb
4
5 # Insere un module "ifb" et cree 1 interface ifb
6 modprobe ifb numifbs=1
7
8 # Accroche le lien au netns
9 ip link set ifb0 netns r2
10
11 # Active l'interface du namespace
12 ip netns exec r2 ip link set ifb0 up
13
14 # Creer un qdisc sur r2 eth1 qui permet d'intercepter les paquets
15 # entrant et de pouvoir
16 # appliquer du tc filter sur ces paquets.
17 # handle ffff: Permet d'identifier le qdisc (ffff est reserve a
18 # ingress)
19 ip netns exec r2 tc qdisc add dev r2-eth1 ingress handle ffff:
20
21 # Ajoute un filtre sur r2 eth1 qui doit s'appliquer a la qdisc
22 # ingress
23 # ce filtre prend tous les paquets entrant en faisant en sorte
24 # qu'a leur sortie,
25 # ils soient rediriger vers l'interface ifb0
26 ip netns exec r2 tc filter add dev r2-eth1 parent ffff: matchall
27 action mirred egress redirect dev ifb0
```

L'interface "ifb" permet donc de gérer le traffic entrant dans r2 en appliquant des règles aux paquets.

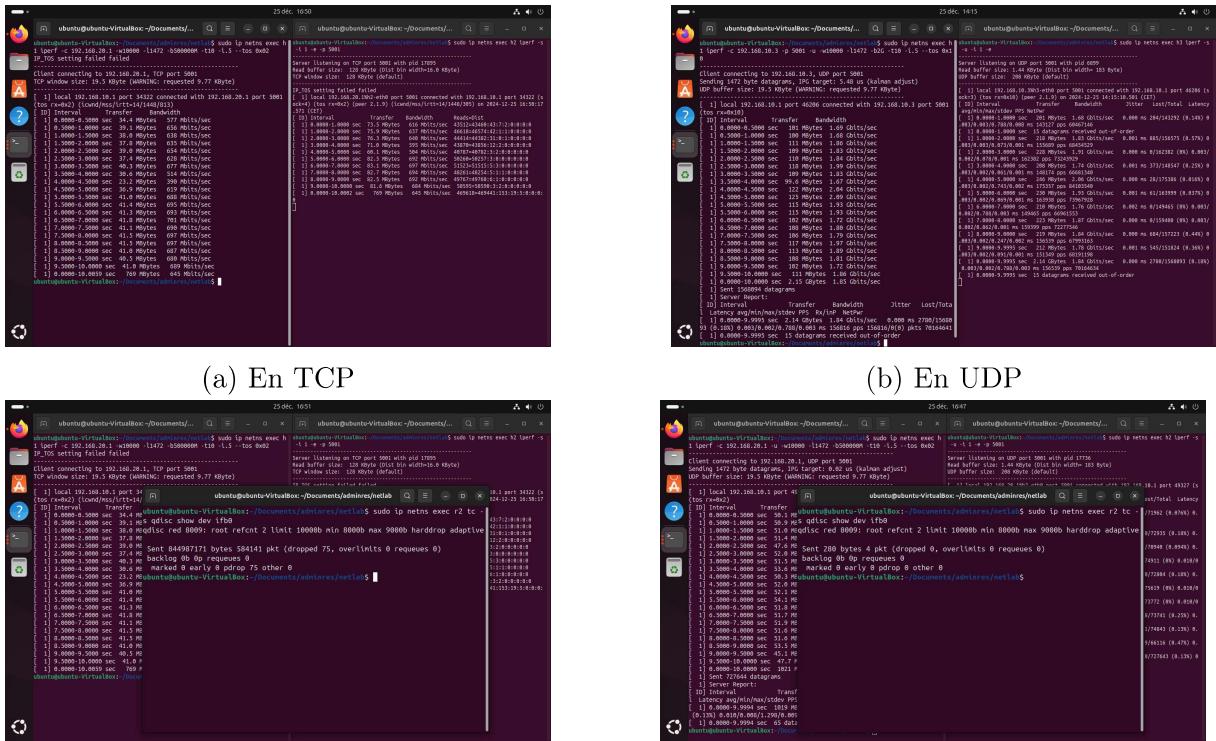
1.2 Mesure des débits sans qdisc



(a) En TCP (b) En UDP

On observe que le débit du protocole UDP est plus rapide que le protocole TCP. On peut noter cependant que le protocole UDP perd régulièrement des paquets

1.3 Mesure des débits avec qdisc



(c) Statistiques du qdisc en TCP

(d) Statistiques du qdisc en UDP

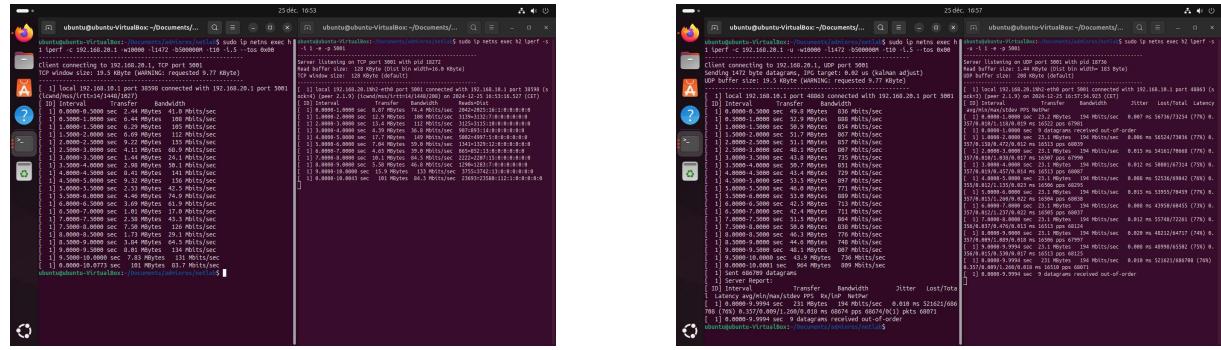
Avec l'utilisation des qdisc, on peut voir que le débit diminue pour protocole TCP mais augmente pour le protocole UDP.

Pour le protocole TCP, cela est du fait que lorsque des suppressions de paquets se produisent, le protocole TCP doit retransmettre ces paquets et peut réajuster la vitesse de transmission.

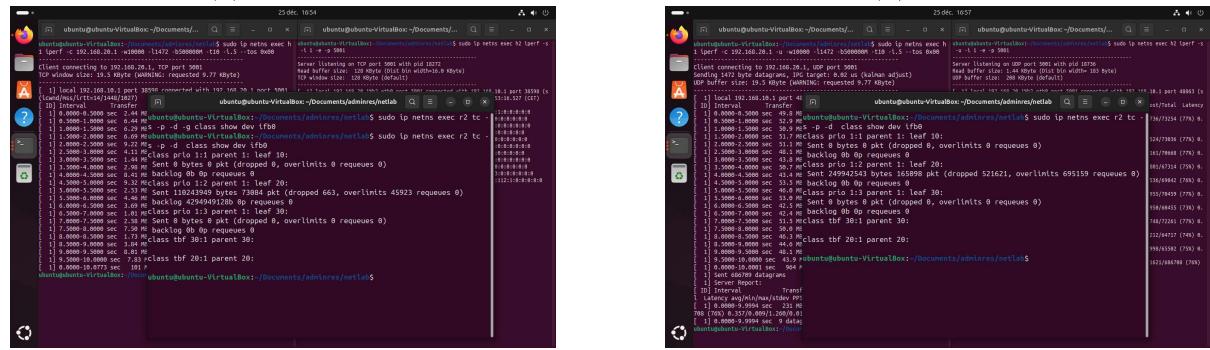
Pour le protocole UDP, étant donné qu'il ne possède pas de mécanisme permettant de modifier la vitesse de transmission ou de réémettre des paquets, il va simplement supprimer les paquets et envoyer ce qu'il peut, augmentant ainsi son débit (il envoie moins de paquets).

1.4 Mesure des débits avec classification par TOS

1.4.1 TOS 0x00

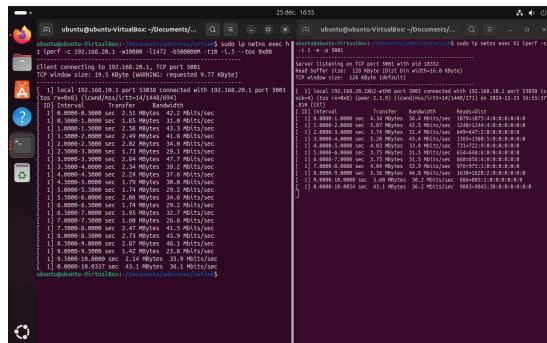


(a) En TCP

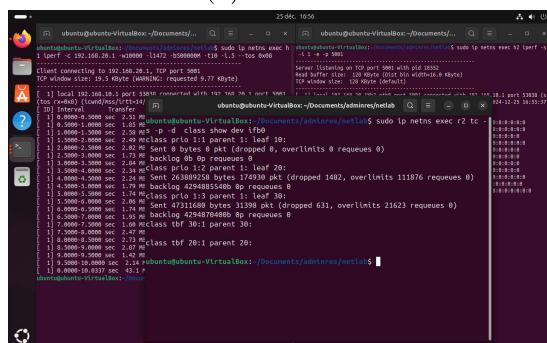


(c) Statistiques du qdisc en TCP

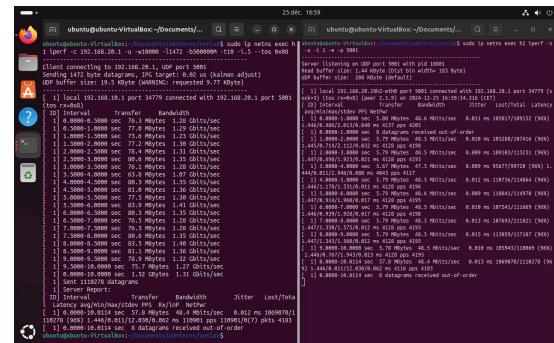
1.4.2 TOS 0x08



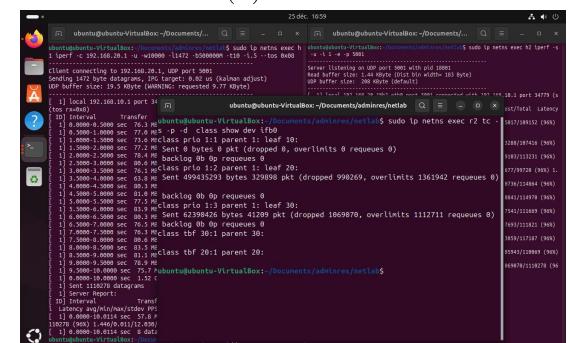
(a) En TCP



(c) Statistiques du qdisc en TCP

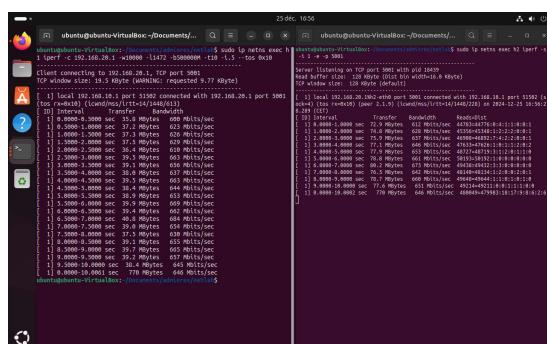


(b) En UDP

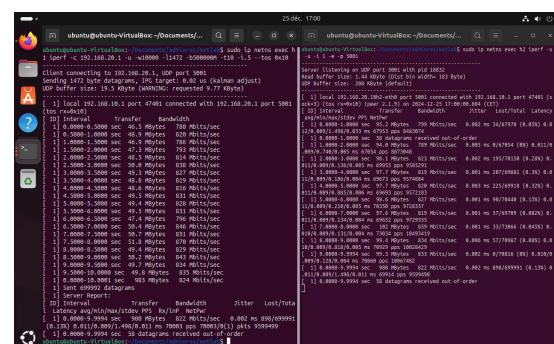


(d) Statistiques du qdisc en UDP

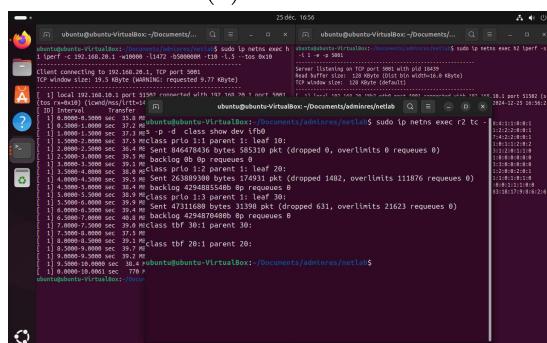
1.4.3 TOS 0x10



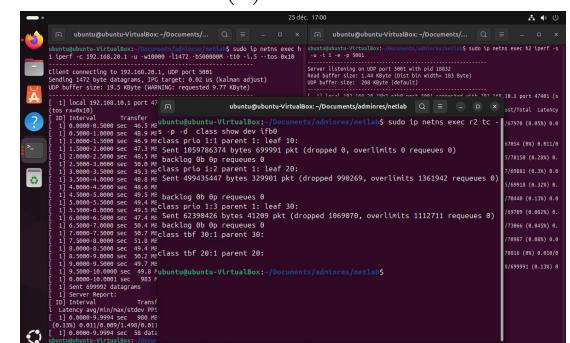
(a) En TCP



(b) En UDP



(c) Statistiques du qdisc en TCP



(d) Statistiques du qdisc en UDP

Cette fois-ci, on a 3 traffics. Un traffic prioritaire qui possède une bande passante "normale". Et deux traffics possédant une bande passante limitée. Comme on peut le voir sur les différentes captures, cette architecture permet de réduire à 0 le nombre de perte sur le traffic prioritaire. On peut donc aussi noter que les deux autres traffics subissent de grande pertes due à leur faible débit.

Chapitre 2

Qos et traffic sortant

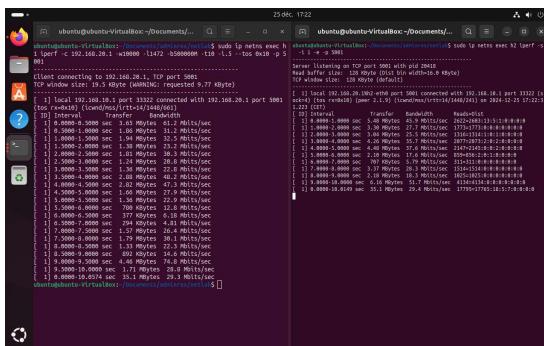
2.1 Qos et Traffic shaping

Voici les règles de firewall pour rediriger le traffic vers les différentes classes :

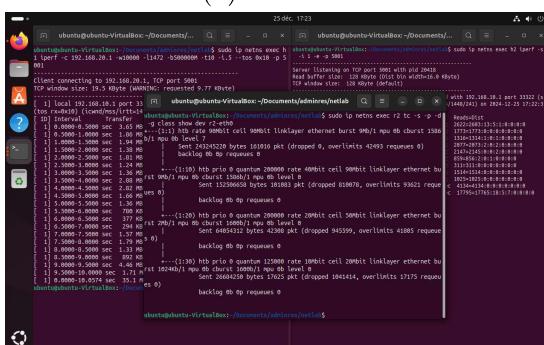
```
1  iptables -t mangle -A POSTROUTING -p tcp --dport 5001 -j CLASSIFY  
   --set-class 1:10  
2  iptables -t mangle -A POSTROUTING -p tcp --dport 5002 -j CLASSIFY  
   --set-class 1:20  
3  iptables -t mangle -A POSTROUTING -p tcp --dport 5003 -j CLASSIFY  
   --set-class 1:30  
4  
5  iptables -t mangle -A POSTROUTING -p udp --dport 5001 -j CLASSIFY  
   --set-class 1:10  
6  iptables -t mangle -A POSTROUTING -p udp --dport 5002 -j CLASSIFY  
   --set-class 1:20  
7  iptables -t mangle -A POSTROUTING -p udp --dport 5003 -j CLASSIFY  
   --set-class 1:30
```

On utilise le numéro du port pour savoir quel traffic nous allons utilisé.

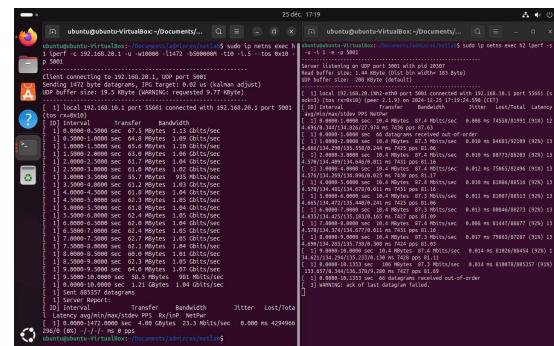
2.2 Classe 1 :10



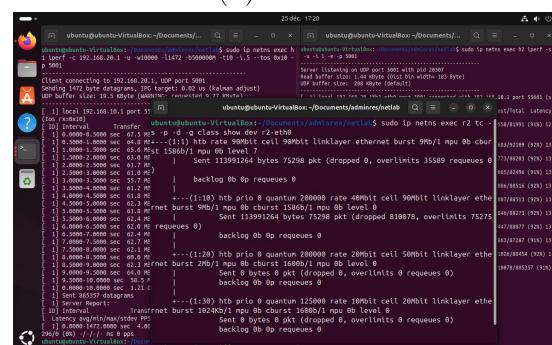
(a) En TCP



(c) Statistiques du qdisc en TCP

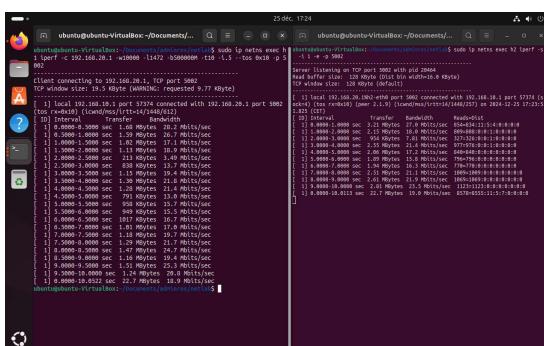


(b) En UDP

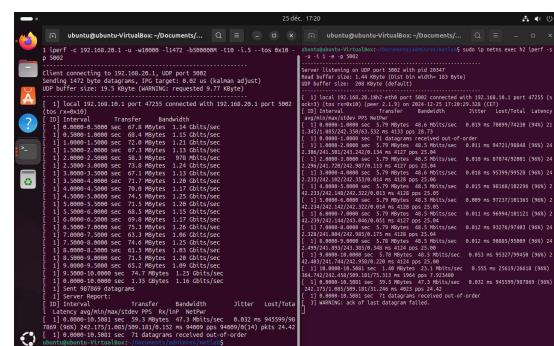


(d) Statistiques du qdisc en UDP

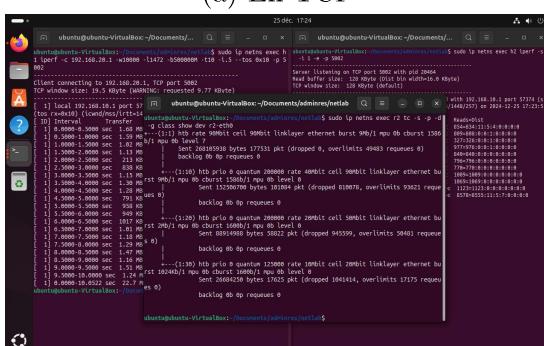
2.3 Classe 1 :20



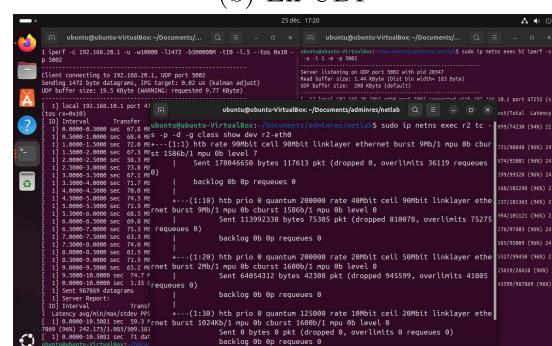
(a) En TCP



(b) En UDP

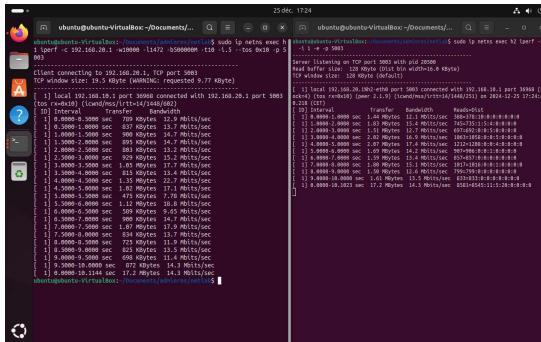


(c) Statistiques du qdisc en TCE

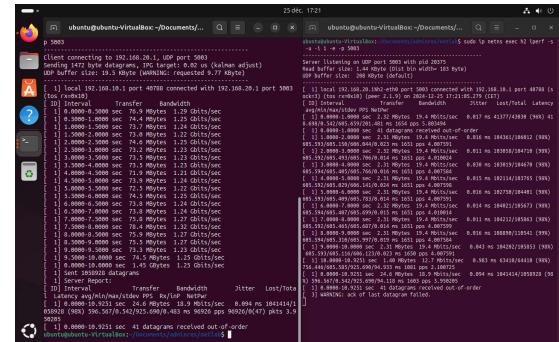


(d) Statistiques du qdisc en UDP

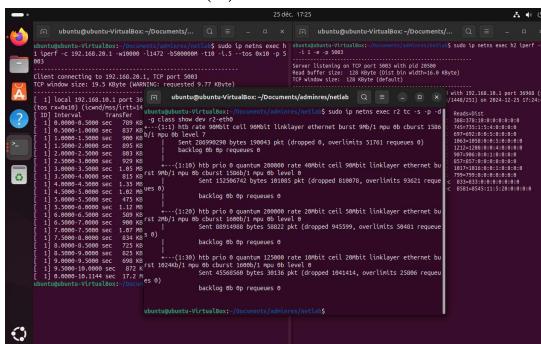
2.4 Classe 1 :30



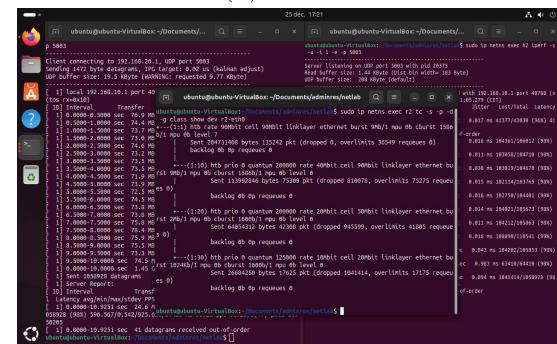
(a) En TCP



(b) En UDP



(c) Statistiques du qdisc en TCP



(d) Statistiques du qdisc en UDP

On peut voir qu'en sortie, les débits sont bien limités et que la QoS est donc bien appliquée.

Chapitre 3

QoS à la demande

3.1 Paquet forgé

Pour forger les paquets, nous allons utilisé hping3. Cette commande sera lancé depuis H1 vers H2. Pour faire en sorte que le ttl du paquet forgé fasse disparaître le paquet sur R2, il doit être égal à 2 car il fait 2 sauts, le premier saut vers R1 et le deuxième vers R2. Cependant pour pouvoir tester la QoS, nous allons mettre le ttl à 3 pour que le paquet puisse exister suffisamment longtemps pour atteindre H2.

3.2 Interception du paquet

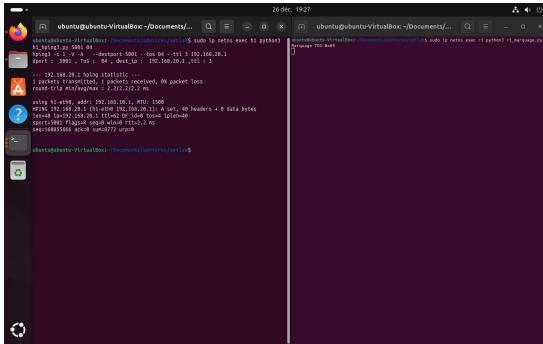
L'outil permettant d'intercepter les paquets présente quelques problèmes.

Le premier problème est que le programme ne s'arrête pas même si le paquet a été intercepté. Cela est du au fait que nous avons utilisé des threads pour intercepter les paquets. En effet, l'utilisation de tcpdump bloque le programme. Pour remédier à ce problème, nous avons donc utilisé les threads, où chaque thread lance la commande tcpdump.

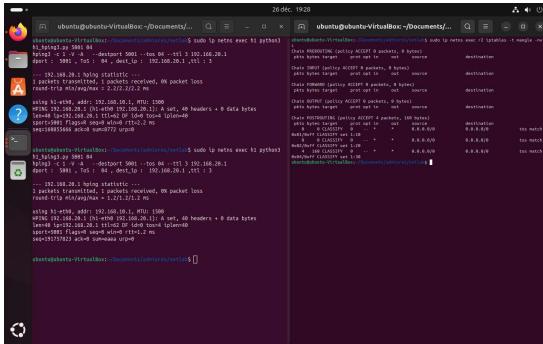
Le deuxième problème est que le paquet qui déclenche la création d'une règle de firewall ne va pas passer dans cette règle. Seuls les paquets arrivant après ce premier paquet vont être marqué par le firewall.

3.3 Test des outils

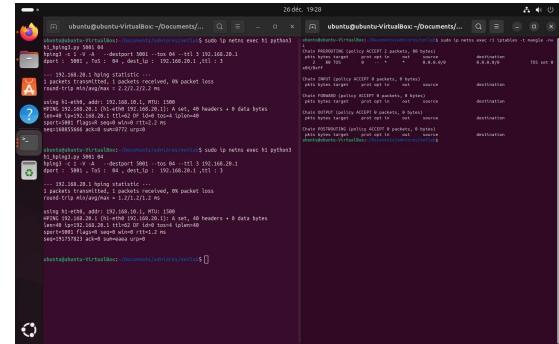
Tout d'abord, nous allons mettre en place les règles de firewall de R2 permettant d'appliquer la QoS (voir figure (c)). R2 appliquera la QoS en fonction du TOS des paquets. Ensuite, R1 attend pour recevoir des paquets et depuis H1, on envoie les paquets forgés. R1 reçoit le paquet, crée la règle de firewall en fonction du TOS du paquet, le paquet est marqué, il arrive sur R2 et entre dans le bon traffic.



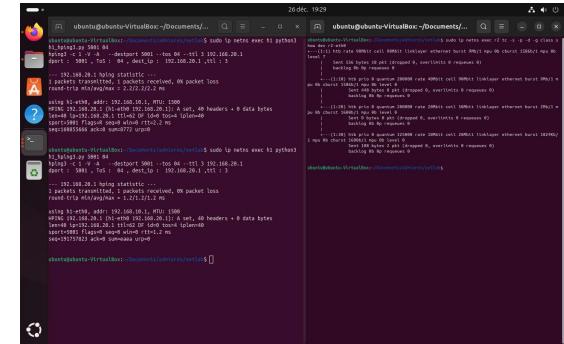
(a) Interception du paquet



(c) Les paquets passent par le firewall de R2



(b) Création d'une règle de firewall



(d) La QoS de R2 a été appliquée