Les TP réseaux

- Netkit
- tcpdump

netkit

- **Émule** des réseaux d’ordinateurs
  - *Contrairement à la simulation, l'émulation ne reproduit que les fonctionnalités mais pas la performance d'un vrai réseau.*
- Logiciels open source (licence GPL)
- Utilise des logiciels libres
- Basé sur UML (User Mode Linux)

Emulation d'un réseau

- Fonctionne avec des machines virtuelles (VM)
- Chaque nœud du réseau émulé est une VM
- Les VMs sont basées sur User Mode Linux
- Plusieurs VMs peuvent être exécutées à un instant ’t’
- Les VMs peuvent communiquer entre-elle via des réseaux (domaine de collision).
  - Attention : par défaut les réseaux sont de « l'ethernet partagé » et non de « l'ethernet commuté ».

Les commandes

- Netkit dispose de 2 types de commandes :
- Les « vcommandes » qui commencent par « v » pour agir sur une seule VM
- Les « lcommandes » qui commencent par « l » pour un groupe de VM … Lab

Les vcommandes

- **vstart** : Pour démarrer une machine virtuelle équipée d'un certain nombre de cartes réseau reliées à des domaines de diffusion.

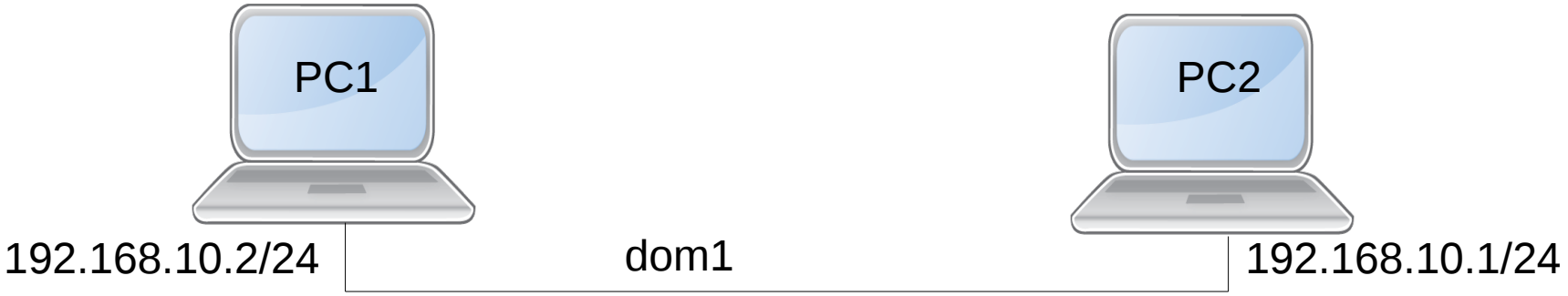
  Vstart --eth0=<domaine> <nom_machine>

- **vlist** : Pour prendre connaissance des machines virtuelles actuellement actives.
- **vhalt** : Pour arrêter proprement une machine virtuelle (sauvegarde du filsystem dans le fichier .disk).
- **vcrash** : Pour arrêter brutalement une machine virtuelle.
- **vclean** : commande "sous panique", pour supprimer tous les processus Netkit ainsi que les fichiers de la machine virtuelle.
- **vconfig** :Pour ajouter à la volée une carte réseau à une machine virtuelle.

Les lcommandes

- **lstart** : démarrage du Lab du répertoire courant.
- **lhalt** : arrêt ordonné des machines du Lab en cours d'exécution.
- **lcrash** : arrêt brutal de toutes les machines du Lab en cours d'exécution.
- **linfo** : présente le Lab du répertoire courant sans démarrer le Lab.
- **lclean** : supprime tous les fichiers temporaires crées par le Lab.

Exemple simple 1



- Création du dossier de travail :

  mkdir /tmp/lab1; cd /tmp/lab1

- Création des deux VM

  vstart --eth0=dom1 PC1

  vstart --eth0=dom1 PC2

- Vérification

  vlist

Exemple simple ...2

- Configuration de la première VM
  - Si on fait ifconfig → pas d'interface
    - (ip link pour valider la présence d'eth0 non configurée)
    - Sur PC1: ifconfig eth0 192.168.10.1 netmask 255.255.255.0
    - Sur PC2: ifconfig eth0 192.168.10.2 netmask 255.255.255.0
  - Le test : depuis PC1 : ping 192.168.10.2

Tcpdump : capture de trafic

La commande tcpdump est un outil de capture de paquets très puissant.

- Pour diminuer le "bruit", Tcpdump utilise des expressions pour filtrer (conserver) les paquets :
- **dst** adresse IP destination, **src** adresse IP source, **host** IP source ou destination
- **tcp** pour capturer seulement les trames TCP et **udp** pour les trames UDP
- Enfin, la commande tcpdump comporte des options pouvant s'avérer utiles pour afficher plus d'informations, citons :
- -XX pour l'affichage du contenu de la trame
- -i pour indiquer l'interface où doit se faire l'écoute
- -n affichage des adresses (à la place du nom)
- -A pour l'affichage du contenu des paquets (en ASCII)
- -e pour l'affichage des adresses MAC de la trame
- -t sans l'affichage de l'horodatage des traces
- -vv pour une interprétation détaillée des champs des en-têtes.

Tcpdump : exemples

- Par exemple, la capture des paquets vers le port 80 et la machine 10.10.10.1

  tcpdump -i eth0 -n -t dst 10.10.10.1 ***and*** *port http*

- *Capture avec information niveau Ethernet*

  tcpdump -i eth0 ***-e -t -q***

  - *-e info niveau trame*
  - *-t sans timestamp*
  - *-q « version courte »*

Pour en savoir plus, consulter le manuel en ligne de tcpdump et de nombreux tutos.

Exercice 2

lab.conf

```
r1[0]="A"
r1[1]="B"

r2[0]="C"
r2[1]="B"

pc1[0]="A"

pc2[0]="C"

pc3[0]="C"
```

pc1.startup

```
ifconfig eth0 195.11.14.5 up
route add default gw 195.11.14.1
```

pc2.startup

```
ifconfig eth0 200.1.1.7 up
route add default gw 200.1.1.1
```

pc3.startup

```
ifconfig eth0 200.1.1.3 up
route add default gw 200.1.1.1
```

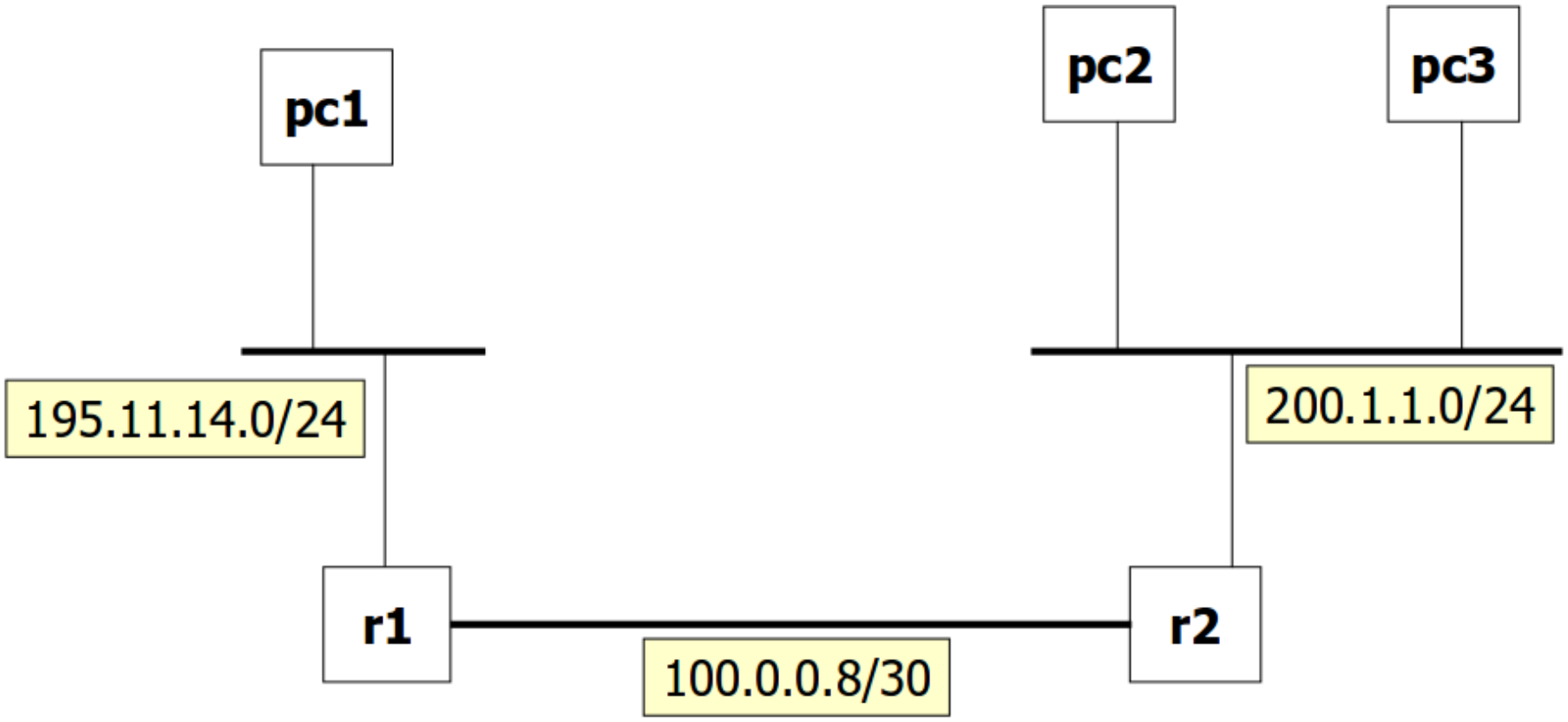
r1.startup

```
ifconfig eth0 195.11.14.1 up
ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
```

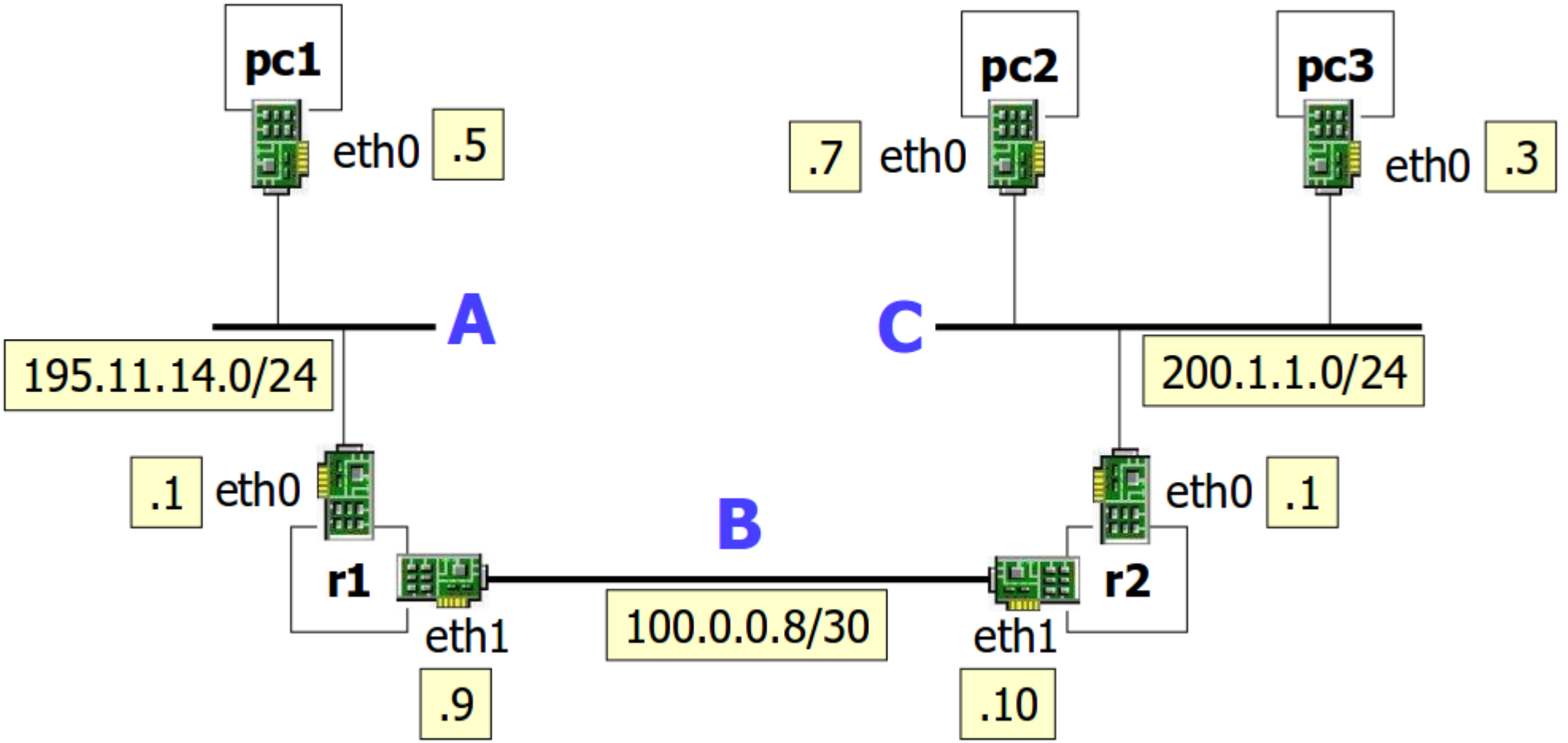
r2.startup

```
ifconfig eth0 200.1.1.1 up
ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
```

Exercice 2



Exercice 2


pc1
eth0
.5
pc2
.7
eth0
pc3
eth0
.3
A
C
195.11.14.0/24
200.1.1.0/24
.1
eth0
B
eth0
.1
r1
r2
eth1
100.0.0.8/30
eth1
.9
.10