



M3102 – Services et reseaux

RESPONSABLE : PATRICK MARTINET

MATÉRIEL : ONETE.NET/TEACHING.HTML

EMAIL : CRISTINA.ONETE@GMAIL.COM

Des vrais réseaux vs. la simulation

- ▶ Nos TPs l'année dernière :
 - ▶ Plusieurs machines, des équipements réseau (routeurs, switches, etc.)
 - ▶ Une machine virtuelle par machine
 - ▶ Les machines étaient connectées dans des sous-réseaux, éventuellement passant sur des passerelles et/ou via la machine prof.
- ▶ Quels sont les risques de cette approche ?

Dans ce module on va plutôt simuler des réseaux complexes

L'outil Netkit

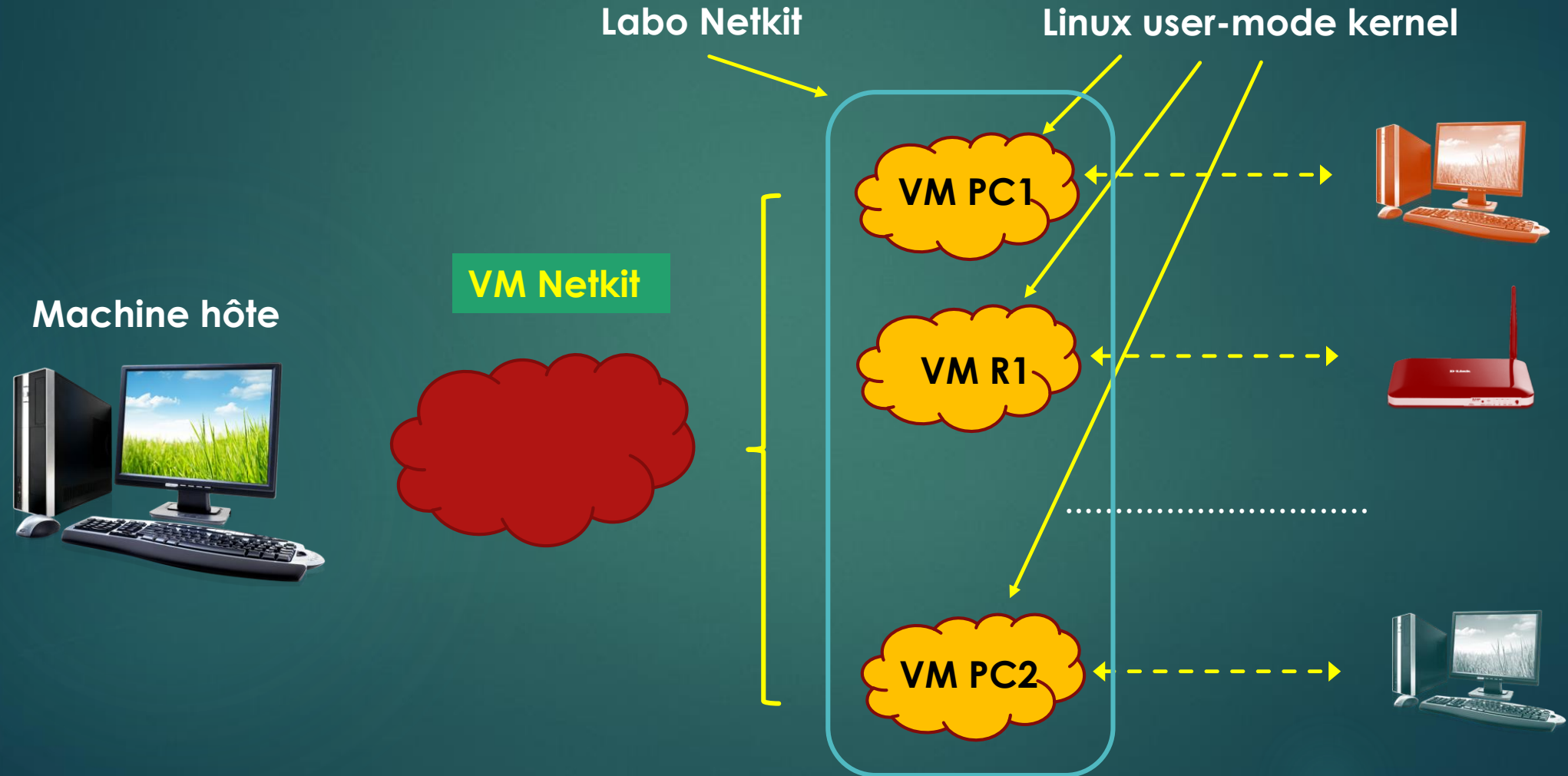
- ▶ Permet de simuler le fonctionnement de tout un reseau sur une machine
- ▶ Un logiciel qui permet de créer des VMs (Linux box) sur une **machine hôte**
 - ▶ Chaque VM bénéficie d'un noyau Linux
 - ▶ Chaque VM joue le rôle d'un élément d'un réseau : un PC, un routeur, un serveur...
 - ▶ Toutes ces VMs sont dans un seul VM dans lequel on exécute Netkit
 - ▶ On peut manipuler VM par VM, ou alternativement faire toute la simulation fonctionner à partir de Netkit
- ▶ Ceci nous permet de simuler le fonctionnement d'un réseau en toute sécurité



Une description de Netkit

LES V-COMMANDES, LES L-COMMANDES, ETC.

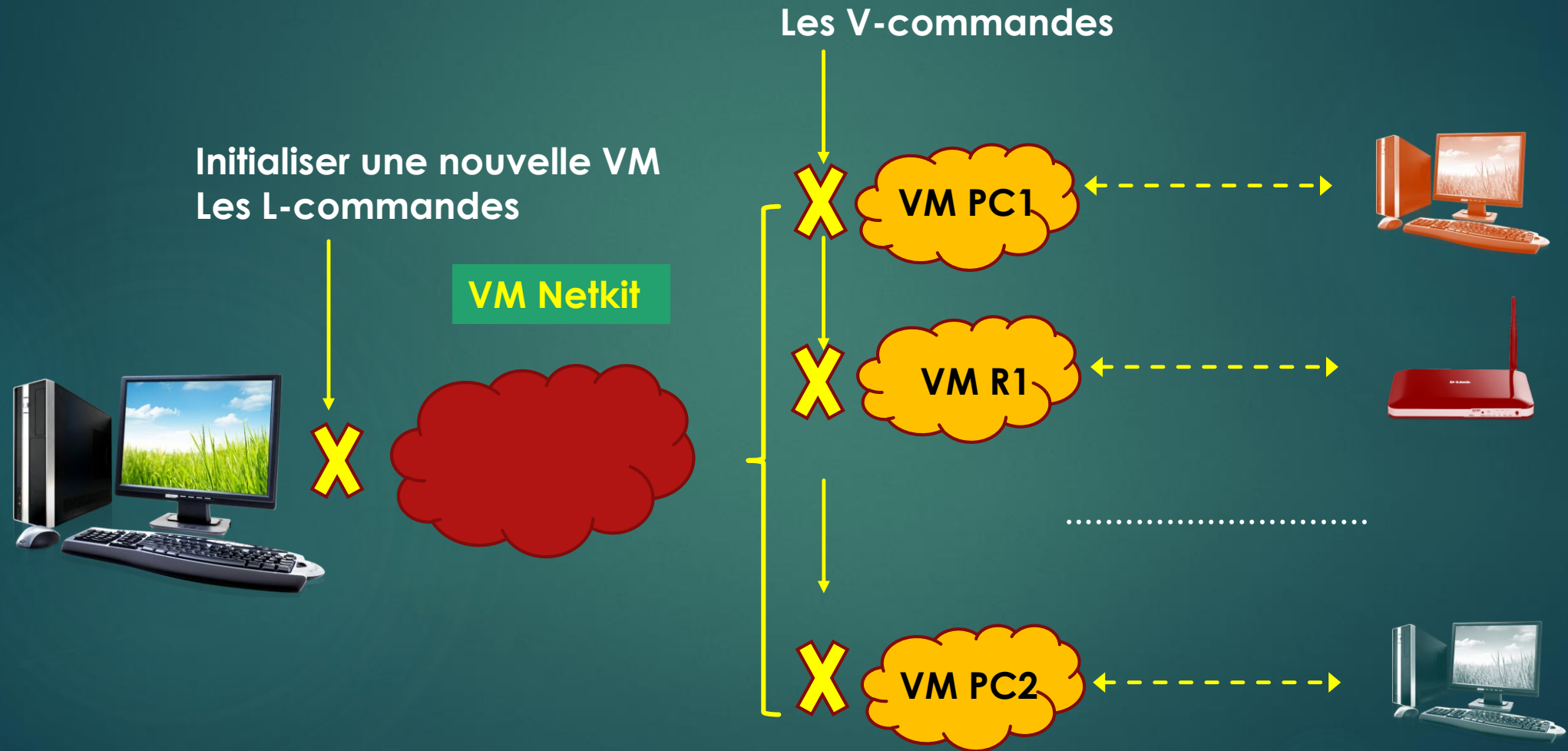
L'infrastructure avec Netkit



Deux types de commandes

- ▶ Netkit nous donne la possibilité de "programmer" un réseau
 - ▶ Spécifier quels rôles elles jouent
 - ▶ Détailler comment elles sont connectées
 - ▶ Simuler la communication entre elles
- ▶ Essentiellement deux types de commandes :
 - ▶ Les V-commandes (vcommands) : des commandes pour chaque machine virtuelle (d'où le v)
 - ▶ Les L-commandes (lcommands) : des commandes pour le labo en entier (d'où le L)

Faire entrer les commandes



Quelques V-commandes

- ▶ **vstart** : fait démarrer une nouvelle machine virtuelle
- ▶ **vlist** : liste des VMs actuelles
- ▶ **vconfig** : permet d'ajouter dynamiquement des nouvelles interfaces à une VM déjà existente
- ▶ **vhalt** : permet d'arrêter une VM sans causer des crashes
- ▶ **vcrash** : provoque un crash
- ▶ **vclean** : permet de nettoyer tout Netkit et la configuration de la machine hôte

Quelques L-commandes

- ▶ **Istart** : démarre un labo
- ▶ **Ihalt** : arrêt
- ▶ **Icrash** : provoque un crash sur toutes les VMs du labo
- ▶ **Iclean** : enlève les fichiers temporaires du repertoire d'un labo
- ▶ **Iinfo** : donne des informations concernant un labo, sans le démarrer
- ▶ **Itest** : permet l'exécution de tests, pour savoir si un labo fonctionne correctement

Les labos Netkit

- ▶ Décrivent une certaine topologie, spécifiée dans un fichier lab.conf
 - ▶ La topologie spécifie des machines avec des interfaces réseau
 - ▶ Les reseaux forment des "domaines de collision" (collision domain)
 - ▶ On a la possibilité de donner des specifications pour chaque machine (par exemple combien de RAM on a, etc.)
- ▶ Chaque machine virtuelle sera associée à un sous-répertoire
- ▶ Des fichiers pour décrire le comportement des machines au start et au halt

```
~/TP_netkit/LP/Harmo $ tree .  
.  
├── lab.conf  
├── pc1  
├── pc2  
└── pc3
```

La configuration d'un labo : lab.conf

10

On déclare 3 machines,
nommées pc1, pc2, pc3

```
3 directories, 1 file
~/TP_netkit/LP/Harmo $ cat lab.conf
machines="pc1 pc2 pc3"
pc1[0]=domain0
pc2[0]=domain1
pc3[0]=domain0
pc3[1]=domain1
pc3[mem]=64
```

- <nom_machine>[<numéro>]** : se réfère à l'interface <numéro> de la machine <nom_machine>
- domain0, domain1** : deux domaines de collision distincts, domain0, domain1
- <nom_machine>[mem]** : se réfère à la capacité de mémoire de la machine <nom_machine>

Dessinez la topologie indiquée par ce script

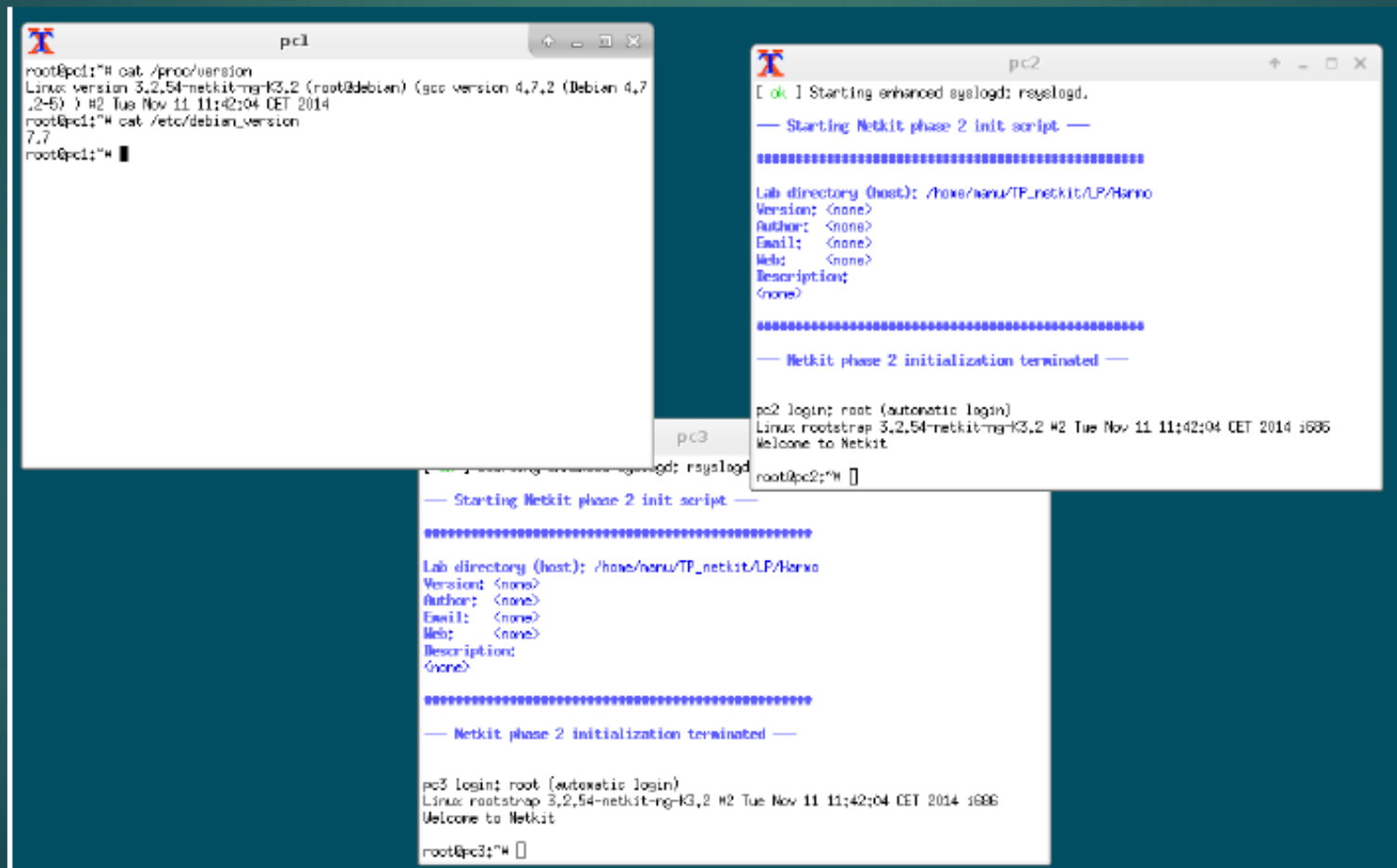
Configurer les interfaces des VMs

11

- ▶ Se fait en Netkit via les scripts de start et de shutdown
- ▶ Deux possibilités :
 - ▶ Chaque VM peut avoir un fichier <nom machine>.startup
 - ▶ Une instruction typique dans un tel fichier pourrait être
ifconfig eth1 192.168.1.2/24 up
 - ▶ Les configurations qui affectent toutes les VMs sont dans deux fichiers
shared.startup et shared.shutdown
- ▶ Au démarrage, la VM exécute shared.startup, puis <nom de la VM>.startup
- ▶ Le fichier de shutdown est utilisé au cas des opérations de reboot ou de halt
 - ▶ On ne l'utilise pas dans le cas d'un crash

Les trois VMs initialisées

12



```
pc1
root@pc1:~# cat /proc/version
Linux version 3.2.54-netkit-ng-K3.2 (root@debian) (gcc version 4.7.2 (Debian 4.7
.2-5) ) #2 Tue Nov 11 11:42:04 CET 2014
root@pc1:~# cat /etc/debian_version
7.7
root@pc1:~# █

pc2
[ ok ] Starting enhanced syslogd: rsyslogd.
--- Starting Netkit phase 2 init script ---
=====
Lab directory (host): /home/naru/TP_netkit/LP/Hanno
Version: <none>
Author: <none>
Email: <none>
Web: <none>
Description:
<none>
=====
--- Netkit phase 2 initialization terminated ---
pc2 login: root (automatic login)
Linux rootstrap 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 :685
Welcome to Netkit
root@pc2:~# █

pc3
[ ok ] Starting enhanced syslogd: rsyslogd.
--- Starting Netkit phase 2 init script ---
=====
Lab directory (host): /home/naru/TP_netkit/LP/Hanno
Version: <none>
Author: <none>
Email: <none>
Web: <none>
Description:
<none>
=====
--- Netkit phase 2 initialization terminated ---
pc3 login: root (automatic login)
Linux rootstrap 3.2.54-netkit-ng-K3.2 #2 Tue Nov 11 11:42:04 CET 2014 :686
Welcome to Netkit
root@pc3:~# █
```

La commande vstart

13

- ▶ Elle peut être exécutée dans un script ou comme ligne de commande

```
vstart <options> <nom_machine>
```

- ▶ Les options :

-M <valeur> ou -mem=<valeur> : associe la valeur de Mo indiquée pour la VM

-eth<numéro> = <nom_domaine> : associe l'interface eth<numéro> (par ex. eth0 ou eth1) avec le domaine de collision

-eth<numéro> = tap, <adresse TAP><adresse guest> : fonction avancée, permet d'obtenir accès à l'Internet via l'interface ethN

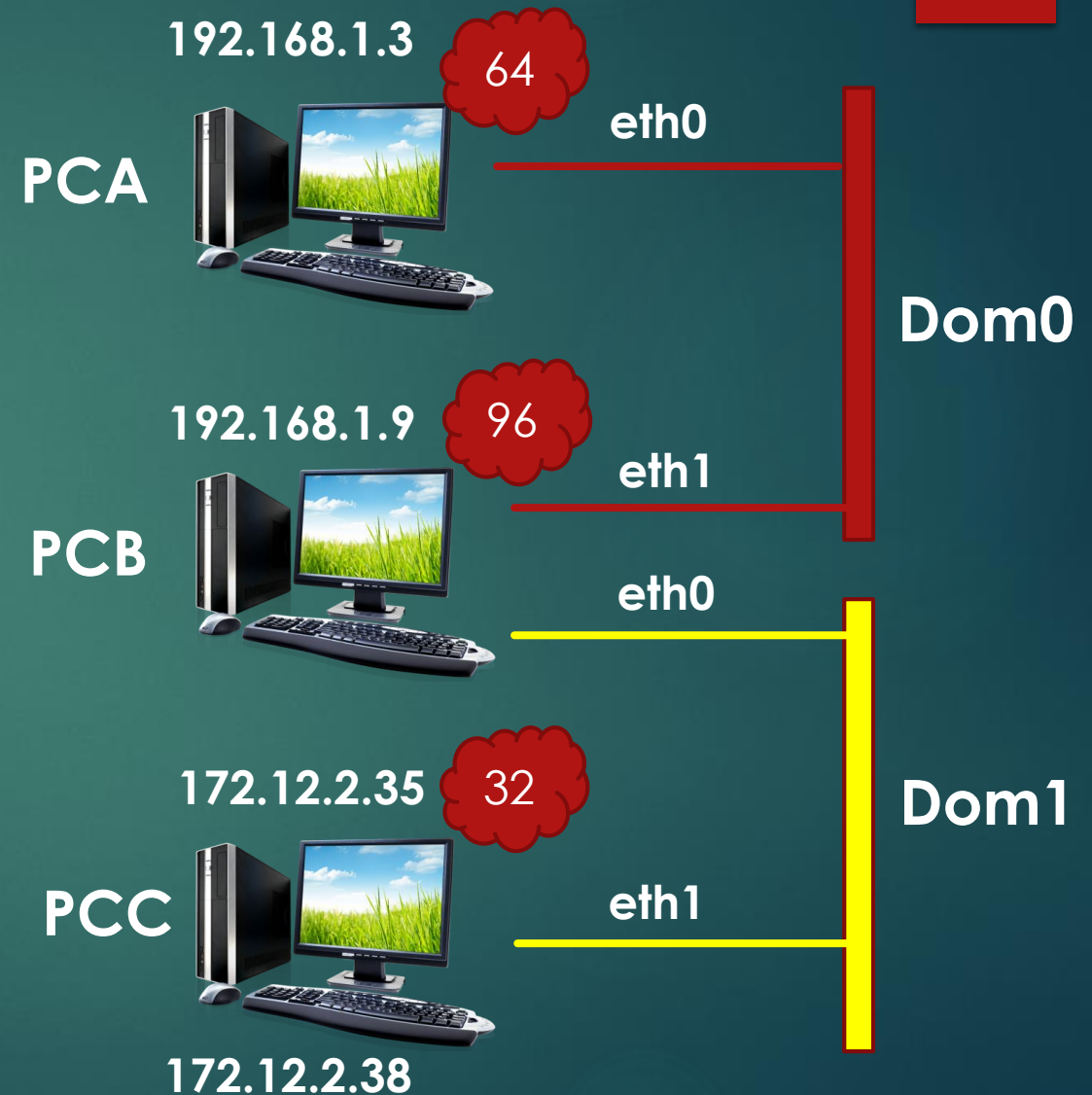
- ▶ Exemple :

```
vstart -M 64 --eth0=DOM0 --eth1=tap,192.168.1.3,192.168.1.11
```

Exercices

14

- ▶ Ecrivez un fichier lab.conf qui décrit la configuration à droite
- ▶ Quels seraient les contenus des fichiers PCB.startup et PCB.shutdown ?





Des fonctions avancées

L'INTERFACE TAP ET L'ACCES INTERNET

Une connexion Internet

16

- ▶ Le logiciel Netkit sert à simuler des configurations de réseau avant de les mettre en pratique
 - ▶ Par exemple pour anticiper des crashes, des difficultés concernant le routage...
- ▶ Parfois, même pour une simulation il nous faut une connexion Internet

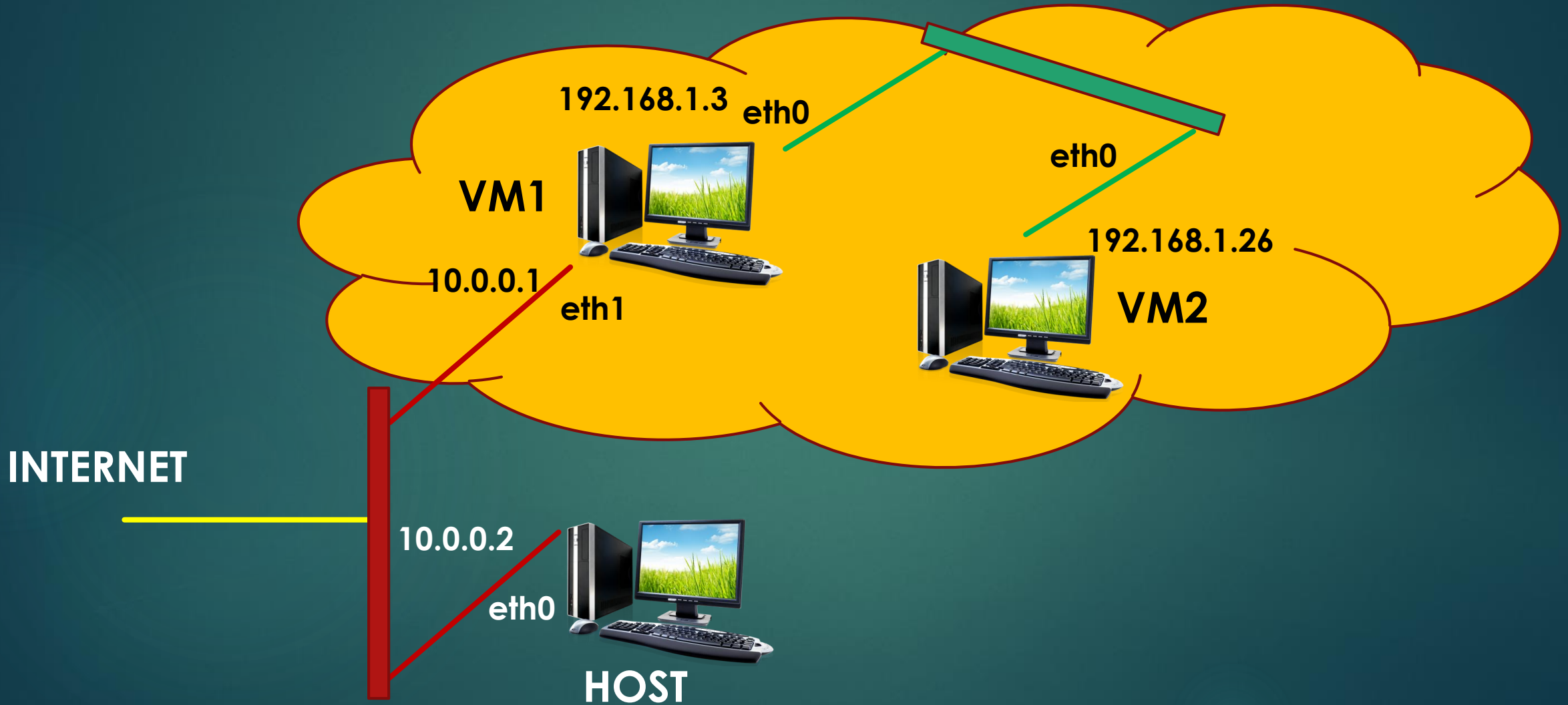
Pourquoi ?

L'interface TAP

17

- ▶ Réaliser un tunnel entre une VM et l'Internet en utilisant la machine hôte
 - ▶ Si la VM est connectée à d'autres machines, la VM jouera le rôle d'un routeur
 - ▶ La VM sera juste connectée à l'Internet sinon
- ▶ Typiquement, la VM qui forme un bout du tunnel a plusieurs interfaces :
 - ▶ Une dans le même domaine de collision que la machine hôte DOM_hote
 - ▶ Les autres dans d'autres domaines de collision DOM_i
 - ▶ L'accès Internet de cette machine est direct via l'interface sur DOM_hote
 - ▶ Les autres VMs peuvent accéder l'Internet via cette VM

Un exemple



Configurer un tunnel TAP

19

- ▶ En ligne de commande :

```
vstart -eth<numéro> = tap, <adresse TAP>, <adresse GUEST> <nom VM>
```

- ▶ Par exemple :

```
vstart -eth1 = tap, 10.0.0.2, 10.0.0.1 VM1
```

- ▶ En lab.conf :

```
<nom VM>[<numéro interface>] = tap, <adresse TAP>, <adresse GUEST>
```

- ▶ Par exemple :

```
VM1[1] = tap, 10.0.0.2, 10.0.0.1
```

Exercices

20

- ▶ Ecrivez le fichier lab.conf qui correspond à cette figure
- ▶ Quelles instructions doit-on fournir pour que la machine VM2 ait accès à l'Internet ?
- ▶ Où va-t-on écrire ces instructions ?

