

Kathara

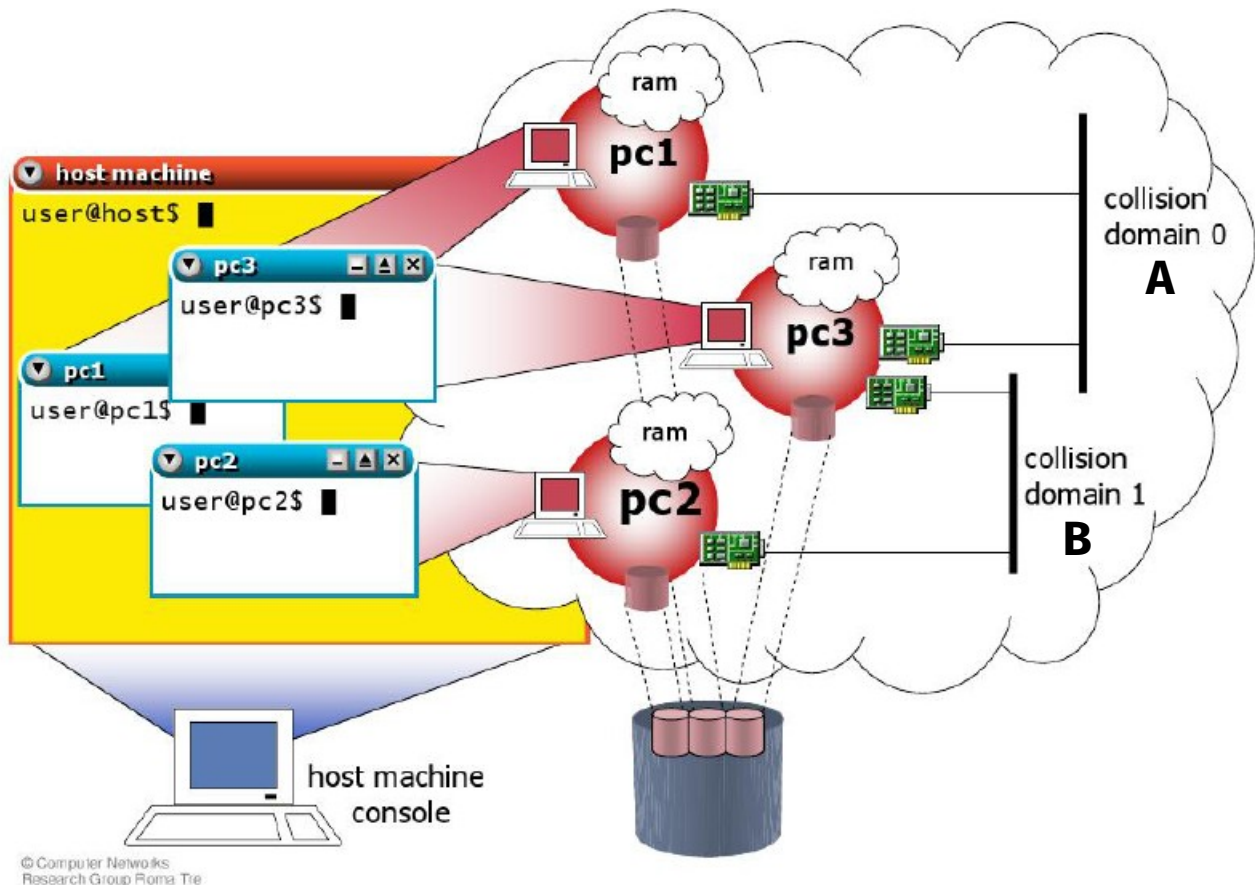
Kathará est une infrastructure basée sur des conteneurs pour déployer des réseaux virtuels. Kathará est une implémentation de **Netkit** en utilisant Python et **Docker**.

www.kathara.org

Netkit est un ensemble de scripts permettant de démarrer et configurer un ensemble de machines et de créer un réseau virtuel entre ces machines.

<http://wiki.netkit.org/>

En résumé : **un système pour émuler des réseaux informatiques.**



Les **machines virtuelles** (ici pc1, pc2 et pc3) sont reliées à des domaines de collisions virtuels (un **hub**) et elles pourront donc communiquer entre-elles.

Chaque machine virtuelle peut jouer le rôle de PC (client et/ou serveur), routeur ou **switch**.

Un **domaine de collision** virtuel peut être connecté à plusieurs interfaces. Chaque interface peut être connectée à un ou plusieurs domaines de collision

kathará fournit trois ensembles de commandes avec les préfixes **v** (commandes v), **l** (commandes l) et des commandes globales. cf. **man kathara**

- Les commandes **v** agissent comme des outils de bas niveau Netkit pour configurer et démarrer un seul appareil.
- Les commandes **l** fournissent un environnement plus facile à utiliser pour configurer des laboratoires Netkit composés de plusieurs appareils.
- Les commandes globales sont principalement des commandes de gestion

LAB

Un **lab** est un ensemble de fichier permettant de créer puis de configurer un scénario réseau netkit automatiquement.

Le lab est décrit dans un fichier **lab.conf** contenant le nom des machines (**pc1**, **pc2** et **pc3**) et la configuration réseau : un domaine de collision A partagé entre l'interface 0 (ou **eth0**) de la machine **pc1** et l'interface 0 (ou **eth0**) de la machine **pc3**.

```
host> cat lab.conf
pc1[0]="A" # eth0 de pc1 connecté au domaine A
pc2[0]="B"
pc3[0]="A"
pc3[1]="B" # eth1 de pc3 connecté au domaine B

host> sudo kathara lstart # démarre le lab

host> sudo kathara lclean # arrête le lab
```



Des fichiers **pcx.startup** peuvent contenir des commandes qui seront exécutées au démarrage de la machine.

Dans Netkit, le concentrateur (**hub**) est transparent. Il suffit de relier des machines sur un même domaine de collision (virtuel) pour qu'elle puisse échanger des trames. Il sera possible par la suite de **capturer** (**sniffer**) les trames échangées sur un domaine de collision en utilisant **tcpdump** directement sur les machines virtuelles ou **wireshark** à partir de la machine hôte.

```
host> sudo docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
9f7f3f546efb	bridge	bridge	local
55f6e6d10d7f	host	host	local
4946dae3ed83	kathara_vagrant_A	kathara/katharanp:stretch	local

```
host> sudo docker network inspect kathara_vagrant_A
```



Attention : si vous avez plusieurs domaines de collision (A, B, C, ...), il vous faudra exécuter plusieurs **wireshark** pour **sniffer** les différents trafics ou utiliser **any**.

Travaux pratiques

- Installer si nécessaire **VirtualBox** et **Vagrant**

- Ouvrir un terminal :

```
$ mkdir tp-reseaux-kathara; cd tp-reseaux-kathara
$ vagrant init "tvaira/lubuntu-kathara-bts"
$ vim Vagrantfile
    Vagrant.configure("2") do |config|
      config.vm.box = "tvaira/lubuntu-kathara-bts"
      config.vm.provider "virtualbox" do |v|
        v.gui = true
      end
    end
```

- Démarrer la machine virtuelle

```
$ vagrant up
```

- Arrêter la machine virtuelle

```
$ vagrant halt
```