Travaux pratiques Réseau (Routage) - Partie 1

Préambule

Lancez en préalable de toutes manipulations, les commandes suivantes :

Utilisez de préférence la commande sudo, plutôt que la commande sudo –i pour des commandes en mode privilégié.

Arrêt de NetworkManager : systemctl stop NetworkManager

Vous pouvez ensuite lancez la commande nmcli dev status pour vérifier
Suppression et visualisation des règles du pare-feu :
iptables -X
iptables -S
iptables -F
iptables -L

Identifiez l'adresse IP de la carte réseau reliée au LAN de la salle. Elle se présente sous la forme 10.192.10.Y.

Demandez à l'enseignant le numéro X de votre binôme (associé à deux machines côte à côte).

Sur les machines deux cartes sont installées : quelles sont leur nom ? Identifiez-les.

Prenez un switch dans l'armoire. Prenez en soin, comme de tout le matériel.

Branchez un câble réseau entre les deux premières entrées puis branchez l'alimentation pour le réinitialiser.

Remarque : pour passer un hôte en mode routeur, privilégiez les commandes
sysctl -w net.ipv4.ip_forward = 1 ou sysctl net.ipv4.conf.all.forwarding=1

1.- Réalisation du montage représenté sur la figure 1. En binôme.

En vous aidant du switch, réalisez le montage réseau présenté sur la figure 1. Vous utiliserez les interfaces réseaux situées en bas des tours pour les relier au switch.

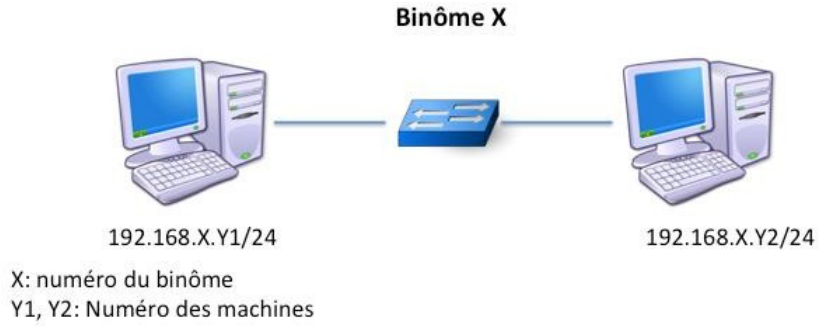


Figure 1. Deux PC connectés par l'intermédiaire d'un switch à leur carte réseau respective

La commande permettant de fixer les adresses IP aux interfaces réseaux est:
ifconfig [Interface réseau] [Adresse de la carte réseau]/[masque]

Vérifiez à l'aide de la commande ping la connexion entre les deux interfaces.

2.- Réalisation du montage représenté sur la figure 2. Deux binômes : (4 postes).

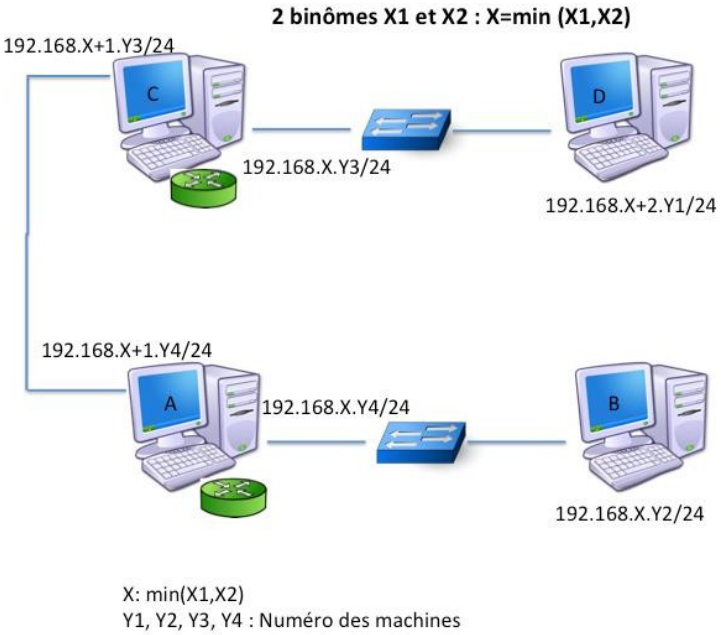


Figure 2. Montage de 4 PC en 3 sous-réseaux. Deux PC sont transformés en routeur.

Vous travaillerez en vous associant à vos collègues voisins autour de 4 machines connectées en 3 sous–réseaux comme il est précisé sur la figure 2. Deux machines seront transformées en routeur : elles seront connectées entre-elles deux via leur interface du haut. Les 4 machines seront déconnectées du réseau.

Construisez les routes de manière à ce que B communique avec D via A et C. Vous utiliserez les commandes

- ifconfig [Interface réseau] [Adresse de la carte réseau]/[masque]
- route add –net [Adresse réseau]/[Masque réseau] gw [adresse IP passerelle]
- route –n

Indication. Procédez par étapes successives. Vérifiez lors de la première étape, en utilisant la commande ping que B et A communiquent. A l'étape 2, que C et D communiquent, puis Etape 3 que A et C communiquent, (Etape 4) que A peut atteindre D, (Etape 5) C peut atteindre B, et enfin que D peut atteindre B.

Validation. Représentez sous la forme d'un schéma comme ceux présentés en cours et en TD les trois sous-réseaux créés. Montrez-le à l'enseignant, ainsi que les tableaux de route pour les postes et les deux routeurs. Demandez-lui de vérifier que votre réseau fonctionne. Aidez-vous du logiciel wireshark.

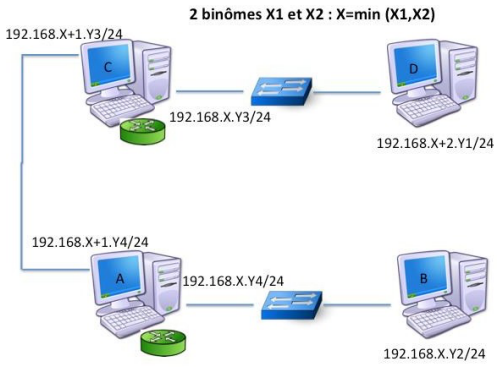
Dépannage : Déconnectez le câble réseau de C, et reconnectez-le. Quelles sont les lignes de commande que vous devez entrer pour que tout fonctionne à nouveau : D doit se connecter à B.

Demandez à un de vos collègues de débrancher un câble réseau et de supprimer une route (sans vous dire lesquels). Recherchez la panne et corrigez. Faites ceci pour chacun d'entre vous. Une fois terminé, reconnectez les machines au réseau et rangez switch et câbles. Vérifiez que vos machines accèdent de nouveau au LAN.

Travaux pratiques Réseau (Routage) Partie 2
Simulation réseau à l'aide de Kathara

L'objectif est ici d'étudier et d'utiliser le simulateur de réseau Kathara. Celui-ci est présenté en annexe en fin de document. Parcourez cette annexe avant de répondre aux questions.

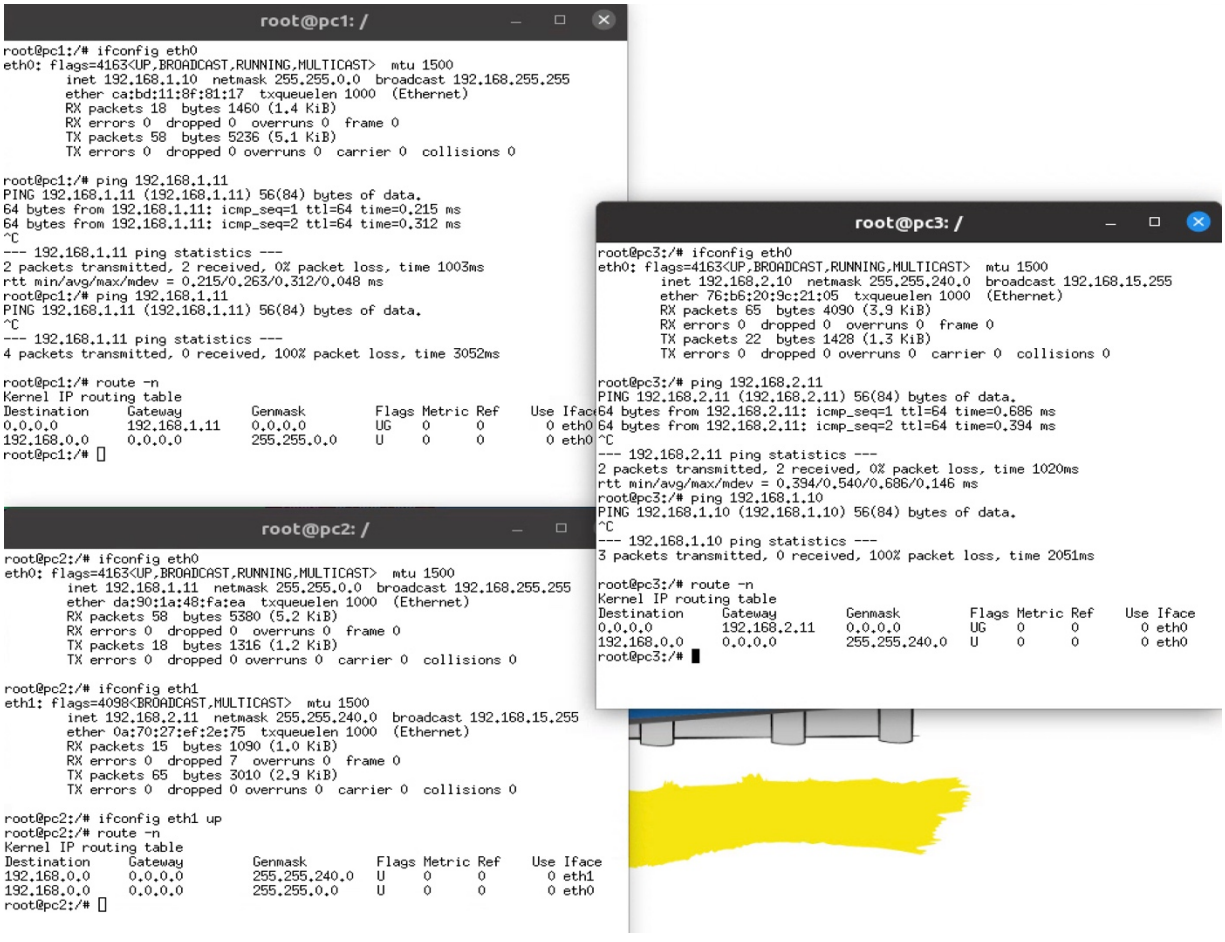
1.- Concevez et simulez l'architecture réseau que vous avez construite précédemment :



X: min(X1,X2)
Y1, Y2, Y3, Y4 : Numéro des machines

2.- Connectez la VM dont l'adresse IP est 192.168.X+1.Y4 au réseau de la salle. Apportez les modifications nécessaires pour que l'ensemble des machines se connectent au réseau de la salle.

3.- Simulez une nouvelle architecture réseau à partir des informations présentées ci-dessous. Elle est composée de trois machines. Comme le montre également la figure, cette architecture présente un disfonctionnement quand les deux interfaces de PC2 sont up. Analysez, puis corrigez le problème en justifiant votre démarche.



Annexe 1 : présentation, installation et premier pas avec Kathara

Pour installer kathara

https://github.com/KatharaFramework/Kathara/wiki/Linux

Commandes (à lancer):
sudo add-apt-repository ppa:katharaframework/kathara
sudo apt update
sudo apt install kathara
kathara check
docker images

Introduction à Kathara

Kathara permet de simuler des réseaux complexes à l'aide de machines virtuelles et de laboratoires virtuels appelés Labs. Il fournit deux types de commandes principales :

- v-commandes : Manipulent une seule machine virtuelle (VM).
- l-commandes : Gèrent des ensembles complexes de machines virtuelles organisées en Labs (laboratoires).

Travail avec une seule VM (v-commandes)

Les v-commandes sont utilisées lorsque vous souhaitez manipuler une seule machine virtuelle. Voici comment créer et gérer une VM.

1. Création d'une VM avec vstart

Pour créer une machine virtuelle, vous utiliserez la commande suivante :

```
kathara vstart -n sta1 --eth 0:HubA --bridged
```

Explication de la commande :

- vstart : Lance une VM.
- -n sta1 : Le nom de la VM est sta1.
- --eth 0:HubA : L'interface réseau eth0 de la VM est associée au domaine de collision HubA (un hub virtuel).
- --bridged : Crée une deuxième interface eth1 qui permet à la VM de se connecter en NAT à l'hôte via un bridge Docker.

2. Vérification de l'état de la VM

Une fois la VM créée, vous pouvez vérifier son état avec les commandes suivantes :

- Lister les VMs actives : kathara list
- Voir les conteneurs Docker : docker ps
- Lister les images Docker : docker images

Pour supprimer la VM sta1, utilisez la commande suivante :

```
kathara vclean -n sta1
```

Travail avec plusieurs VMs (l-commandes)

Lorsque vous avez besoin de gérer plusieurs machines virtuelles en réseau, il est plus pratique de créer un Lab (laboratoire) et d'utiliser les l-commandes.

1. Créer un réseau avec un fichier lab.conf

Le fichier lab.conf décrit la topologie de votre réseau. Voici un exemple de configuration simple :

```
secret[0]=A
public[0]=B
routeur[0]=A
routeur[1]=B
```

Explication :

- Chaque ligne décrit une machine virtuelle et ses interfaces réseau.
- secret[0]=A : la VM secret possède une interface eth0 connectée au switch A.
- routeur[1]=B : la VM routeur possède une deuxième interface eth1 connectée au switch B.

2. Création des répertoires des machines virtuelles

Créez des répertoires correspondant à chaque VM dans le dossier où se trouve votre fichier lab.conf. Dans l'exemple ci-dessus, vous devez créer trois répertoires nommés secret, public et routeur.

3. Démarrer le laboratoire

Une fois le fichier lab.conf et les répertoires créés, lancez les machines virtuelles avec la commande :

```
kathara lstart
```

Cette commande démarre toutes les VMs définies dans le fichier lab.conf.

4. Arrêter le laboratoire

Pour arrêter et nettoyer les machines virtuelles du laboratoire, utilisez :

```
kathara lclean
```

5. Exécution de commandes au démarrage des machines

Vous pouvez automatiser certaines configurations en créant un fichier nom_machine.startup. Par exemple, pour la machine secret, créez un fichier secret.startup avec le contenu suivant :

```
ifconfig eth1 172.30.0.1 netmask 255.255.0.0 up
route add default gw 172.30.0.254
```

Cela configurera l'adresse IP et la passerelle par défaut au démarrage de la VM secret.
Écoute des paquets réseaux

Chaque VM possède un répertoire /shared qui est synchronisé avec un répertoire sur la machine hôte. Cela permet de transférer des fichiers facilement entre les VMs et l'hôte.

Pour capturer des paquets réseau sur une VM et les analyser avec Wireshark, utilisez la commande suivante dans la VM :

```
tcpdump -w /shared/capture.cap
```

Vous pouvez ensuite ouvrir le fichier capture.cap dans Wireshark sur la machine hôte pour analyser le trafic.

Connexion des VMs à l'extérieur

Si vous souhaitez connecter vos machines virtuelles à Internet ou à d'autres réseaux externes, vous devrez

configurer un routeur. Ce routeur doit avoir une interface connectée à l'hôte Docker.

- Commande pour créer un routeur avec une interface bridged :

  kathara vstart routeur --bridged --eth0=A

  Cela crée une interface eth1 sur le routeur, qui est connectée au réseau Docker de l'hôte, et une interface eth0 qui est connectée au switch A.

- Configuration de la NAT sur le routeur

  Pour permettre aux autres machines de communiquer avec l'extérieur via le routeur, configurez la translation d'adresse (NAT) avec la commande suivante sur la VM routeur :

  iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

Cette commande permet de masquer l'adresse source des paquets sortants afin qu'ils puissent passer par l'interface eth1.

Conclusion

Les étapes principales pour utiliser Kathara :

- Travail avec une seule VM : utilisez les v-commandes comme vstart pour gérer une VM individuelle.
- Travail avec plusieurs VMs : créez un fichier lab.conf pour décrire la topologie du réseau et utilisez les l-commandes comme lstart pour démarrer toutes les VMs.
- Connexion à l'extérieur : configurez un routeur avec NAT pour permettre aux machines virtuelles de se connecter à Internet.

Annexe 2 : documentation détaillée de Kathara

Pour installer kathara

https://github.com/KatharaFramework/Kathara/wiki/Linux

Commandes (à lancer et à comprendre):
sudo add-apt-repository ppa:katharaframework/kathara
sudo apt update
sudo apt install kathara
kathara check
docker images

Pour lancer kathara

Toutes les commandes seront lancées à partir de l'exécutable kathara.
Netkit/Kathara fournit deux groupes de commandes :

- les vcommandes, préfixées par 'v', qui permettent de manipuler une seule VM à la fois ;
- les lcommandes, préfixées par 'l' qui servent à manipuler des ensembles complexes de machines virtuelles en réseau. Dans le langage de Kathara, il s'agit des Labs (laboratoires).
-

Si vous souhaitez travailler avec une seule VM, utilisez les vcommandes. Sinon, pour travailler avec plusieurs VMs, il est préférable et bien plus pratique de créer un laboratoire (Lab) et d'utiliser alors les lcommandes.

Les v-commandes

Commande	Action
kathara vstart	
kathara list	→ donner la liste des VMs actives
kathara vconfig	→ configurer à la volée une VM comme par exemple affecter une interface à la volée.
kathara vclean	→ arrêter une VM et nettoyer les processus, configurations et fichiers temporaires créés

Pour s'assurer du bon fonctionnement de Kathara, vous pouvez créer depuis un terminal une première machine virtuelle avec le nom sta1. Création d'une VM avec vstart

kathara vstart -n sta1 --eth 0:HubA –bridged
Explications :

- *La commande vstart permet de lancer en interactif une VM ;*
- *Le nom de la VM sta1 est le paramètre suivant précédé de -n*
- *eth permet de définir le numéro de l'interface réseau eth0 associée à au domaine de collision HubA (hub virtuel) ; Un domaine de collision correspond à un concentrateur pour Kathara. Il faudra définir manuellement une adresse IP avec ifconfig par exemple.*
- *bridged permet de créer une 2ème interface eth1 qui relie la VM en NAT à l'hôte sur le bridge (pont) Docker.*

Une fois la VM créée, testez les commandes : kathara list, docker ps ; docker image ; kathara vclean -n sta1

Construire un réseau à partir d'un fichier lab.conf

Il est possible de décrire la structure du réseau dans un fichier nommé lab.conf. C'est ce que vous ferez

d'une manière générale dans vos montages. La description d'une machine dans le lab.conf se fait de la manière suivante :

nom_machine[0]=A nom_machine[1]=B nom_machine[2]=C

Attention, le nom des machines ne doit pas contenir de majuscules.

Ces lignes créent une machine appelée nom_machine, qui comporte 3 interfaces, eth0, eth1 et eth2, qui sont respectivement connectées aux switchs A, B et C.

Il est nécessaire de créer dans le répertoire contenant le fichier lab.conf des dossiers dont le nom correspond à celui des machines virtuelles.
Par exemple, si votre fichier lab.conf contient :
secret[0]=A public[0]=B routeur[0]=A routeur[1]=B
Alors vous devez créer 3 répertoires nommés : secret, public et routeur. Attention : la casse et l'orthographe des noms des répertoires doivent être identiques aux noms des machines décrites dans le fichier lab.conf.

Une fois les répertoires et le fichier lab.conf créés, on démarre les machines en tapant simplement kathara lstart dans le répertoire contenant le lab.conf.

Pour arrêter le laboratoire, vous pouvez lancer la commande kathara lclean.

Attention ! Hormis les dossiers des noms des machines, le fichier lab.conf et les fichiers *.startup (voir juste après), vous ne devez placer dans le répertoire de travail aucun autre fichier ou répertoire.

Il est possible d'exécuter des commandes au démarrage des machines. Pour cela, il suffit de créer un fichier nom_machine.startup dans le même répertoire que le fichier lab.conf et d'y inscrire les commandes à exécuter. Il est notamment intéressant d'y placer les commandes de configuration des paramètres IP et de la table de routage.
Par exemple, le contenu du fichier secret.startup serait :
ifconfig eth1 172.30.0.1 netmask 255.255.0.0 up
route add default gw 172.30.0.254

Il faut toujours terminer le fichier par un retour à la ligne pour que la dernière commande soit exécutée.
Remarques :

- Par défaut, vous êtes connecté en root sur la machine virtuelle. Le mot de passe du compte root est root.
- Au redémarrage des machines, les modifications réalisées dans les fichiers ne sont pas perdues.

Écoute des paquets réseaux

Chaque machine virtuelle possède un répertoire /shared qui est lié au même répertoire de votre machine hôte. Ainsi, tous les fichiers placés dans ce dernier sur la machine hôte sont accessibles depuis les machines virtuelles par le répertoire /shared, et inversement. Ceci vous permet de transférer simplement des fichiers entre l'hôte et les machines virtuelles.
En particulier, vous pouvez enregistrer les captures tcpdump dans un fichier que vous placerez dans /shared et que vous ouvrirez dans wireshark lancé depuis votre machine hôte. Dans ce but, la ligne de commande à taper sur la machine virtuelle est donc :

tcpdump –w /shared/capture.cap.

Votre répertoire sur la machine hôte contiendra le fichier capture.cap qu'il suffit de l'ouvrir dans wireshark. Vous bénéficiez ainsi de l'interface graphique pour l'analyse de vos captures.

Connexion vers l'extérieur

Pour accéder à l'extérieur, nous créons généralement un routeur disposant d'une interface eth1 (interface qui sera reliée à l'interface docker du système hôte). De plus, grâce à ce routeur, nous pourrons ainsi faire communiquer l'ensemble des machines virtuelles vers l'extérieur et pourrons ainsi installer de nouveaux

paquets.

Exemple (ne pas réaliser) :
La commande suivante kathara vstart routeur --bridged --eth 0=A permet de créer :
• sur la machine virtuelle routeur une interface eth1 avec une adresse dans le même réseau que l'interface docker de l'hôte,
• une interface eth0 sur la machine virtuelle routeur qui pourra maintenant joindre l'extérieur, cette interface sera connectée au switch A

kathara a en fait effectué les opérations suivantes sur le système hôte:

- ajout d'une règle de translation d'adresse;
- ajout d'une règle autorisant les flux émis par notre interface bridged;
- activation du routage.

Il ne reste plus qu'à finir de configurer le DNS sur le routeur (fichier /etc/resolv.conf), son adresse interne connectée au switch A (ex : 172.30.0.250) et de s'assurer que le routage est activé (il l'est par défaut sur toutes les machines kathara).

Cela ne suffit pas pour que les machines autres que le routeur puissent joindre l'extérieur. Le système hôte sera en effet incapable de router les paquets retour. Il n'a en effet aucune connaissance du plan d'adressage internet des machines kathara connectées au switch A (dans le cas de l'exemple 172.30.0.0/16). La solution est de réaliser une translation d'adresse au niveau du routeur pour que les paquets qu'il émet son l'interface bridged (eth1) le soient avec l'adresse du réseau docker.

La commande est la suivante : iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
En résumé, pour qu'une machine puisse se connecter à l'extérieur, il faut créer une interface spéciale (ici eth2) sur une machine router, dont la syntaxe est la suivante : routeur[bridged]=true
Ajoutez sur le routeur (tout ce qui sort est masqué)
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE