

Principe général de routage :

Lorsque qu'une trame arrive sur une interface :

- si l'adresse MAC destination portée par la trame ne correspond pas à l'adresse MAC de l'interface, la trame sera ignorée par cette interface,
- en revanche, si les 2 adresses MAC coïncident, alors la trame sera prise en charge par la machine en question ; il y a alors 2 cas possibles :
 - si l'adresse IP de l'interface et l'adresse IP de destination du paquet (porté par la trame) coïncident, alors le paquet est arrivé à destination,
 - sinon, le paquet n'est pas arrivé à destination, il doit poursuivre sa route. La machine qui a reçu ce paquet va donc faire suivre ce paquet (à condition que l'« ip-forwarding » soit activé sur cette machine), en déterminant le prochain nœud qui devra prendre en charge ce paquet et en lui envoyant (après avoir construit une nouvelle trame, comportant l'adresse MAC de ce prochain nœud).

Pour déterminer quel est le prochain nœud qui prendra en charge le paquet (y compris pour la machine qui envoie le paquet), un nœud consulte sa table de routage, il détermine ainsi l'adresse IP du prochain nœud. Il consulte alors sa table ARP pour connaître l'adresse MAC correspondante afin de pouvoir construire la trame qu'il enverra. Ces 2 tables doivent être mises à jour afin d'assurer ce routage :

- la table ARP est mise à jour grâce au protocole du même nom. Pour cela, quand c'est nécessaire un nœud envoie une requête ARP (broadcast) demandant « Qui a l'adresse IP ... » ; le nœud ayant cette adresse envoie une réponse ARP indiquant « l'adresse IP ... est associée à l'adresse MAC ... ». Il n'y a pas besoin de configurer ce service. Cette table peut être consultée avec la commande `arp`. La commande `arpspoof` permet d'envoyer des messages qui entraîneront une mise à jour de la table ARP d'une machine (en général dans le but de détourner certains paquets en indiquant un couple (@MAC, @IP) incorrect).
- Il existe 2 méthodes pour mettre à jour la table de routage, selon que le routage est statique ou dynamique. Dans le cas du routage statique, il faut fixer « manuellement » la table de routage de chaque nœud en ajoutant/supprimant des routes grâce à la commande `route`. Si cette méthode est pratique pour les petits réseaux, elle ne convient pas à des plus grands réseaux : dans ce cas, il faut activer (et paramétrer) un service pour le routage dynamique, sur chaque routeur. Ce service va déclencher l'échange de messages entre les routeurs, ces messages vont permettre aux routeurs de mettre à jour leur table de routage. Nous utiliserons par exemple RIP pour cela.

Routage statique

Le paramétrage manuel des tables de routages est important aussi bien au niveau des postes de travail que des routeurs.

Pour chaque machine, le simple fait d'attribuer une adresse IP et masque de sous-réseau à une interface entraîne l'ajout d'une ligne dans sa table de routage : cette route indique que pour toutes les adresses du même sous-réseau, le paquet doit être délivré directement. Si on affecte par exemple l'adresse 192.168.2.207/24 à l'interface d'une machine, alors depuis cette machine une route est connue pour toutes les adresses du même sous réseau (192.168.2.1 à 192.168.2.254). Sur une machine non routeur, en général la table sera complétée en spécifiant que pour toutes les autres adresses, le paquet sera pris en charge par un routeur (passerelle par défaut).

Pour les routeurs, la configuration est plus complexe car les routes vont en général dépendre des différents sous-réseaux mis en œuvre. Par ailleurs, pour qu'un routeur puisse traiter les paquets entrant et les rediriger vers l'interface appropriée, il faut activer l'ip-forwarding. Pour cela, il suffit de vérifier que le fichier `/proc/sys/net/ipv4/ip_forward`, contient simplement la valeur 1 ; on peut

fixer comportement avec la commande :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Notez que cette configuration est la configuration par défaut des machines virtuelles que vous utilisez, y compris pour les machines qui ne sont pas des routeurs. Ainsi, le principe général de routage énoncé ci-dessus comporte quelques exceptions, pouvez-vous en observer ?

Routage dynamique (RIP)

Attention, vous devez activer et configurer le routage dynamique sur chaque routeur !

Comme indiqué précédemment, le routage dynamique est mis en œuvre en activant des services (ou daemons). Pour activer le daemon quagga (qui prend en charge le routage dynamique) utilisant le protocole RIP, il faut modifier le fichier `/etc/quagga/daemons` afin qu'il contiennent les 2 lignes suivantes :

```
zebra=yes  
ripd=yes
```

Il faut alors paramétrer le protocole de routage RIP en modifiant le fichier `/etc/quagga/ripd.conf` : voici un exemple de fichier `ripd.conf` :

```
hostname ripd  
password zebra  
router rip  
redistribute connected  
network 192.168.190.0/28  
network ...  
...
```

La 4e ligne spécifie les adresses diffusées dans les vecteurs de distance seront celles des réseaux directement connectés. Les lignes suivantes indiquent sur quels réseaux ces informations seront diffusées (la ligne 5 compote un exemple d'adresse à adapter, vous devez spécifier ici les toutes les adresses de sous-réseaux auxquelles le routeur est connecté).

Le service peut alors être démarré par la commande :

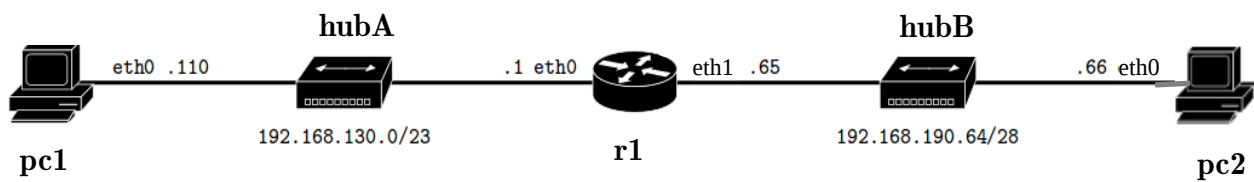
```
/etc/init.d/quagga start
```

Lab Netkit

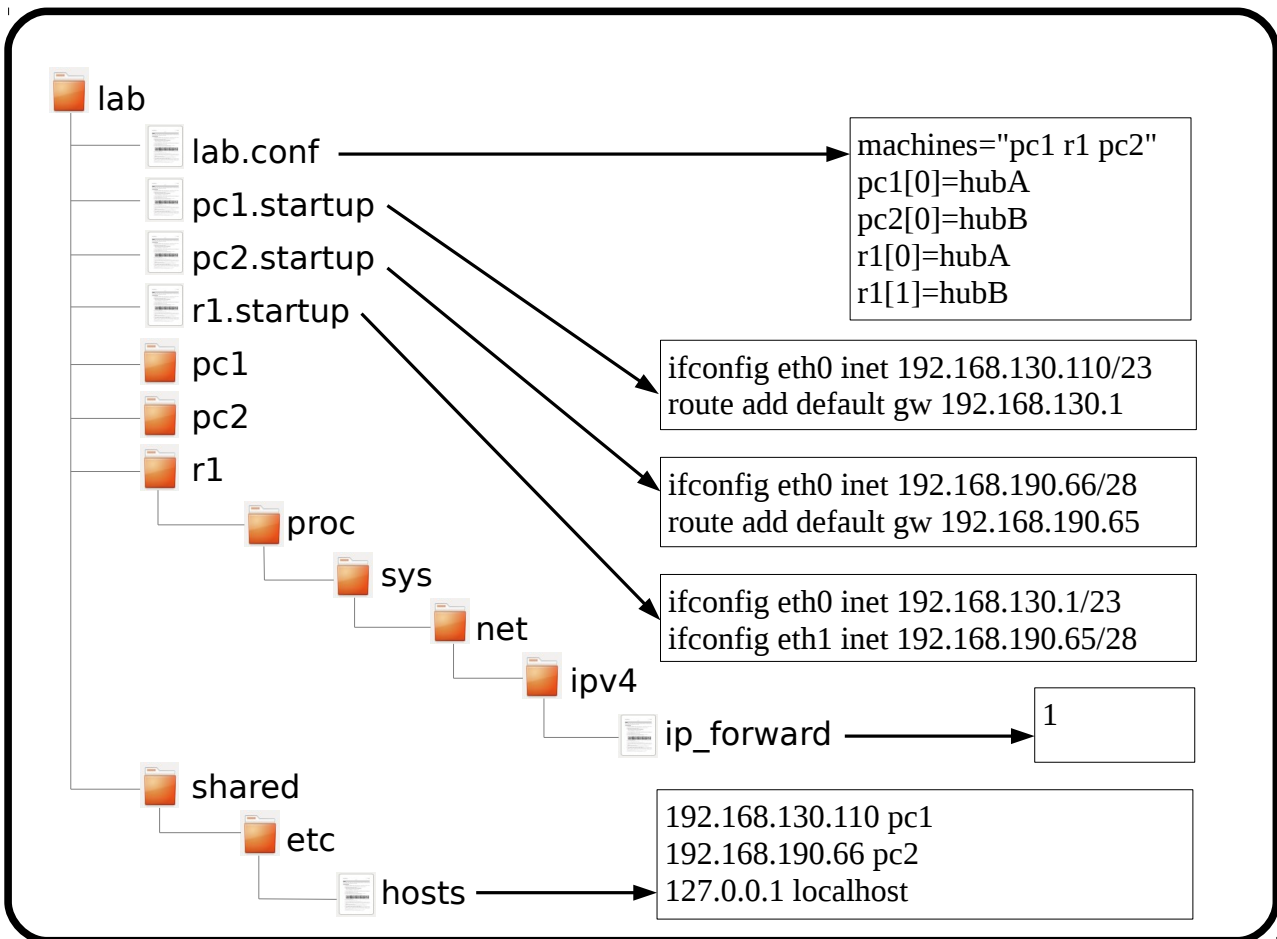
Avec Netkit, vous pouvez utiliser un *lab* pour faciliter la configuration. Un *lab* est un répertoire qui contient (plus de détail sur <http://keepin.org/pdf/keepin-netkit.pdf>) :

- un fichier `lab.conf` qui définit à quel domaine de collision est connectée chaque interface,
- un fichier `machine.startup` pour chaque machine contenant des commandes à exécuter au démarrage de la VM.
- éventuellement un dossier par machine (nom du dossier=nom de la machine) qui sera copié dans à la racine de la machine virtuelle (VM),
- éventuellement un dossier « shared » qui sera recopié à la racine de chaque VM,

Considérons par exemple ce réseau simple :



Nous pouvons créer le lab suivant :



Faut-il compléter la table de routage (fichier r1.startup) ? Pourquoi ?

Concernant le TP sur le routage RIP, vous pouvez créer des fichiers **/etc/quagga/daemons** et **/etc/quagga/ripd.conf** dans le dossier associé à chaque routeur, et insérer la commande d'activation du service dans les fichiers **.startup** des routeurs.

Ne pas oublier d'activer l'ip-forwarding.

Fichier /etc/hosts

Le fichiers **/etc/hosts** offre un moyen de résoudre localement des noms de machines. Les noms étant généralement plus simples à retenir que les adresses IP, il est en général plus pratique de pouvoir utiliser un nom plutôt qu'une adresse IP, par exemple quand on fait un ping. Des correspondances entre IP et noms sont stockées dans ce fichier. Pour notre TP, il peut être intéressant de créer une fichier **/shared/etc/hosts** dans le **lab** avec tous les noms utilisés, ils seront utilisables sur tous les postes.

DHCP (voir aussi http://www.premont.fr/tutos/dhcp_relais.pdf)

Serveur :

La configuration du serveur nécessite les étapes suivantes :

- Il faut spécifier les interfaces que dhcpd (démon de dhcp3-server) écoutera. Vérifier la présence de la ligne

```
INTERFACES="eth0"
```

dans le fichier **/etc/default/dhcp3-server** (ou une autre interface selon le cas)

- le fichier de configuration du serveur dhcp est : **/etc/dhcp3/dhcpd.conf** ; vous devrez modifier ce fichier au cours du TP.

Voici un exemple de fichier **/etc/dhcp3/dhcpd.conf** :

(<http://www.freebsd.org/doc/fr/books/handbook/network-dhcp.html>)

```
option domain-name "example.com";❶
option domain-name-servers 192.168.4.100;❷
option subnet-mask 255.255.255.0;❸

default-lease-time 3600;❹
max-lease-time 86400;❺
ddns-update-style none;❻

subnet 192.168.4.0 netmask 255.255.255.0 {❼
    range 192.168.4.129 192.168.4.254;❽
    option routers 192.168.4.1;❾
}

host mailhost {❿
    hardware ethernet 02:03:04:05:06:07;⓫
    fixed-address mailhost.example.com;⓫
}
```

Commentaires :

❶	Cette option spécifie le domaine qui sera donné aux clients comme domaine par défaut. Consultez la page de manuel de resolv.conf(5) pour plus d'information sur sa signification.
❷	Cette option donne une liste, séparée par des virgules, de serveurs DNS que le client devrait utiliser.
❸	Le masque de sous-réseau qui sera fourni aux clients.
❹	Un client peut demander un bail d'une durée bien précise. Sinon par défaut le serveur alloue un bail avec cette durée avant expiration (en secondes)
❺	C'est la durée maximale d'allocation autorisée par le serveur. Si un client demande un bail plus long, le bail sera accordé mais il ne sera valide que durant max-lease-time secondes.
❻	Cette option indique si le serveur DHCP doit tenter de mettre à jour le DNS quand un bail est accepté ou révoqué. Dans l'implémentation ISC, cette option est <i>obligatoire</i> .
❶❷❸ ❹❺❻	Si ces paramètres sont différents selon les sous réseaux, ils devront être spécifiés dans les rubriques subnet (❼)

⑦	Configuration d'une plage d'adresse pour un sous-réseau avec des options particulières :
⑧	Ceci indique quelles adresses IP devraient être utilisées dans l'ensemble des adresses réservées aux clients. Les adresses comprises dans l'intervalle spécifiée sont allouées aux clients.
⑧	Définit la passerelle par défaut fournie aux clients.
⑨	Rubrique permettant d'attribuer une adresse fixe.
⑩	L'adresse matérielle MAC d'une machine (de manière à ce que le serveur DHCP puisse reconnaître une machine quand elle envoie une requête).
⑩	Indique que la machine devrait se voir attribuer toujours la même adresse IP. Notez que l'utilisation d'un nom de machine ici est correct, puisque le serveur DHCP effectuera une résolution de nom sur le nom de la machine avant de renvoyer l'information sur le bail ; il aurait été bien sur possible de spécifier directement une adresse IP à la place du nom d'hôte.

Enfin, le service peut alors être démarré par la commande

```
/etc/init.d/dhcp3-server start
```

Vous pouvez consulter la liste des baux accordés dans **/var/lib/dhcp3/dhcpd.leases**.

Client

Pour configurer un poste afin qu'il obtienne dynamiquement une adresse IP, modifiez le fichier `/etc/network/interfaces` afin qu'il contienne par exemple (pour configurer `eth1`) :

```
auto eth1
iface eth1 inet dhcp
```

Il est possible de tester dynamiquement le service par la commande :

```
dhclient eth0
```

DHCP-relay

Les broadcasts (et par conséquent les demandes d'adresses DHCP) ne sont pas routés. C'est pourquoi il faut utiliser un relais dhcp pour que les clients puissent obtenir une IP alors qu'ils sont sur un réseau différent.

Commencez par installer **dhcp3-relay** dans une machine virtuelle(cf. procédure sur Célène). Lors de l'installation, vous devez spécifier :

- Nom du server : il s'agit du serveur dhcp vers lequel les requêtes DHCP seront redirigées ; mettre l'ip du serveur par exemple : 192.168.1.3
- Interfaces d'écoutes: mettre par exemple les interfaces : eth0 eth1
- Additional Options : laisser vide si pas d'options

Ces informations sont stockées dans le fichier **/etc/default/dhcp3-relay** , vous pourrez le modifier ultérieurement.

Le daemon peut alors être démarré par la commande :

```
/etc/init.d/isc-dhcp-relay start
```

Remarque : n'oubliez pas d'activer `ip_forwarding`.