

TP Réseau : Routage Statique

© 2011-2018 tv <tvaira@free.fr> - v.1.2

Travail préparatoire	2
Installation du TP	2
Démarrage des machines virtuelles	2
La maquette	2
Rappels	3
Routage	3
Table de routage	4
Travail demandé	4
Routage	4
Tracer une route	5
Route par défaut	6

Les TP d'acquisition des fondamentaux visent à construire un socle de connaissances de base, à appréhender des concepts, des notions et des modèles qui sont fondamentaux. Ce sont des étapes indispensables pour aborder d'autres apprentissages. Les TP sont conduits de manière fortement guidée pour vous placer le plus souvent dans une situation de découverte et d'apprentissage.

Objectifs

Les objectifs de ce TP sont de découvrir le routage réseau IP :

- étudier le principe du routage réseau
- configurer les tables de routage d'une machine et d'un routeur sous Linux
- étudier le fonctionnement des protocoles de la couche réseau IP
- utiliser les commandes de configuration et de test

TP Réseau : Routage Statique

Travail préparatoire

Installation du TP

Le TP2 est disponible dans l'archive `/home/user/sujets-tp/tp2a.tgz` :

```
host> tar zxvf sujets-tp/tp2a.tgz
```



Le prompt `host>` indiquera que la commande doit être tapée dans le terminal de votre machine réelle (par opposition aux terminaux ouverts par les machines virtuelles). Par exemple, votre terminal distant `ssh`. Le prompt `name:~#` représentera le terminal de la machine virtuelle `name`.

Démarrage des machines virtuelles

Démarrer le premier tp en lançant la commande `lstart` (`-s` pour le mode séquentiel) dans le répertoire du *lab* (ici `tp2a`) :

```
host> cd /home/user/tp2a
host> lstart -s
```

Remarque : il est conseillé de lire la FAQ Netkit fournie.

La maquette

Dans ce TP, on dispose de deux machines (`pc1` et `pc2`) reliées entre elles par deux routeurs :

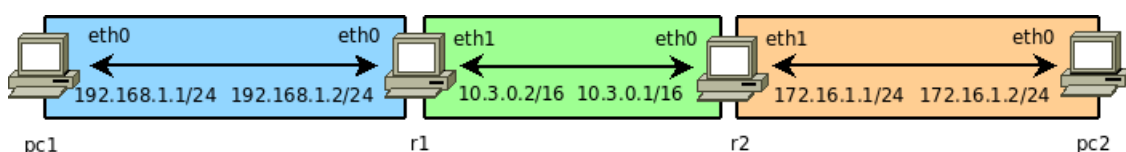
```
pc1[0]="A"
pc2[0]="C"
r1[0]="A"
r1[1]="B"
r2[0]="B"
r2[1]="C"
```

Le fichier `lab.conf`



Le numéro précise l'interface utilisé : `0` → `eth0`, `1` → `eth1`, ... La lettre identifie le domaine de collision (virtuel) sur lequel les machines échangeront des trames.

Ce qui donne le schéma suivant :



Il sera possible par la suite de capturer (*sniffer*) les trames échangées sur un domaine de collision en précisant son nom.

```
host> vdump A | wireshark -i - -k &  
host> vdump B | wireshark -i - -k &  
host> vdump C | wireshark -i - -k &
```

Rappels

Sous GNU/Linux, la commande `ifconfig` permet la configuration des interfaces réseaux.

```
pc1:~# ifconfig eth0 up  
pc1:~# ifconfig eth0
```

Activer et visualiser les paramètres de `eth0`

Pour configurer l'interface `eth0` de chaque machine, on utilisera donc la commande `ifconfig` en précisant l'adresse de diffusion générale (*broadcast*) et le masque réseau (*netmask*) :

```
pc1:~# ifconfig eth0 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
```

// Ou :

```
pc1:~# ifconfig eth0 192.168.1.1/24
```

Pour réaliser un test de base d'une liaison réseau, on utilise souvent la commande `ping` qui permettra, en cas de succès, de valider la pile de protocoles jusqu'au niveau IP.

```
pc1:~# ping 192.168.1.2 -c 1
```

pc1 envoie un message ICMP de type *echo request* auquel la machine **pc2** va répondre par un message ICMP *echo reply*. Pour connaître les options de la commande `ping`, consulter les pages `man`.

ICMP est un protocole de couche Réseau, qui permet le contrôle des erreurs de transmission. En effet, comme le protocole IP ne gère que le transport des paquets et ne permet pas l'envoi de messages d'erreur, c'est grâce à ce protocole qu'une machine émettrice peut savoir qu'il y a eu un incident de réseau (par exemple lorsqu'un service ou un hôte est inaccessible). Il est détaillé dans la **RFC 792**.

Routage

Le **routage** consiste à déterminer la route qu'un paquet doit prendre pour atteindre une destination. Cette tâche est réalisée au niveau de la couche RESEAU du modèle à couches. Dans cette couche, on utilise un adressage qui permet de spécifier à quel réseau appartient un équipement (hôte ou routeur) : dans le modèle TCP/IP, on utilise l'adressage IP.

Les fonctions au niveau de la couche RESEAU sont :

- **acheminer (hôte ou routeur)** : envoyer un paquet vers une destination (hôte ou routeur)
- **relayer (routeur)** : acheminer un paquet d'un réseau vers un autre réseau (*forwarding*)

Pour déterminer la route à prendre, le pilote IP utilise sa **table de routage** qui indique pour chaque destination (hôte, réseau ou sous-réseau), la route (interface ou passerelle) à prendre : routage de proche en proche.

Table de routage

Une **table de routage** indique pour chaque destination (hôte, réseau ou sous-réseau) la route (interface et passerelle) qu'il faut prendre.

Pour visualiser la table de routage, on utilisera la commande `route` ou `netstat -r`. L'option `-n` désactive pour ces deux commandes la résolution de noms.

Un équipement (hôte ou routeur) aura plusieurs possibilités pour construire sa table de routage :

- manuellement : routes statiques
- automatiquement par échange de routes (protocoles RIP, OSPF, ...) avec ses voisins ou par connaissance directe : routes dynamiques

On distinguera aussi :

- les routes directes : délivrance d'un paquet à un hôte qui appartient au même réseau physique. La commande `ifconfig` permet la configuration du routage direct en associant une adresse IP à une carte réseau.
- les routes indirectes : délivrance d'un paquet à un hôte qui appartient à un réseau physique différent. La commande `route` permet la configuration du routage indirect en permettant l'ajout et la suppression de route vers un hôte, un réseau ou une route par défaut.

On peut indiquer trois types de route indirecte dans une table de routage :

- route vers un poste :

```
pc1:~# route add 192.168.131.2 gw 192.168.3.131 dev eth0
```

- route vers un réseau :

```
pc1:~# route add -net 192.168.3.0 gw 192.168.3.131 netmask 255.255.255.0 dev eth0
```

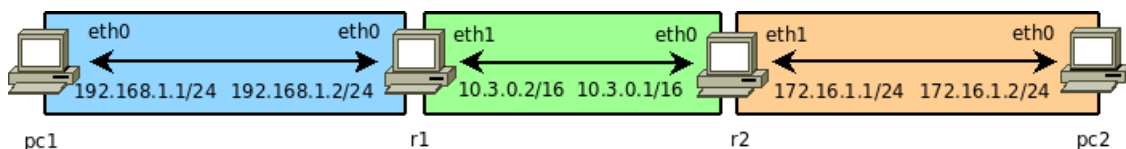
- route par défaut :

```
pc1:~# route add default gw 192.168.3.131 dev eth0
```

Travail demandé

Routage

L'architecture réseau est la suivante :



Vérifier les paramètres IP des interfaces de l'ensemble des machines.

Question 1. Tester une communication depuis **pc1** vers **pc2**. Est-ce-que **pc2** est « joignable » ?

Question 2. Visualiser la table de routage de **pc1**. Existe-t-il une route pour joindre le réseau de **pc2** ?



Parfois l'étoile '*' de la colonne *Passerelle* est remplacée par 0.0.0.0. Cela correspond à une route directe ne nécessitant pas de passerelle.

Question 3. Ajouter sur **pc1** une route pour joindre le réseau 172.16.1.0/24 en passant par la passerelle (*gateway*) 192.168.1.2.

Question 4. Tester une communication depuis **pc1** vers **pc2**. Est-ce-que **pc2** est maintenant « joignable » ? Si non, pourquoi ?

Question 5. Visualiser la table de routage de **r1**. Existe-t-il une route pour joindre le réseau de **pc2** ?

Question 6. Configurer les tables de routage de tous les machines afin que chaque **pcx** puisse communiquer avec n'importe quel autre. Fournir les tables de routages des 4 machines.



On n'utilisera pas de routes par défaut pour ces machines.

Question 7. Quel est le contenu de la table de routage lorsqu'on désactive l'interface **eth0** de **pc1** (`ifconfig eth0 down`) ? Quel est le contenu de la table de routage lorsqu'on réactive l'interface **eth0** de **pc1** (`ifconfig eth0 up`) ?

Tracer une route

L'en-tête d'un paquet IP contient un champ sur 8 bits nommé TTL pour *Time To Live*. Ce champ permet de gérer la durée d'un paquet sur un réseau. En effet, avec une topologie de réseau maillé, il y a un risque qu'un paquet boucle indéfiniment sur le réseau.

Si la valeur du champ TTL devient nulle, son paquet IP ne sera plus relayé. Le routeur devra le détruire et envoyé à l'émetteur du paquet un message ICMP.

Une utilisation détournée de ce champ permet de « tracer la route empruntée » par un paquet.

Question 8. Tester une communication depuis **pc1** vers **pc2** avec un TTL de 1. Quel est le type de message ICMP reçu ? Relever l'adresse IP de l'émetteur de ce message ICMP.

Question 9. Recommencer en incrémentant au fur et à mesure la valeur du TTL. Relever alors la route empruntée. Combien de routeurs ont été traversés ? À partir de **pc1**, est-il possible d'obtenir les adresse IP des interfaces **eth1** de **r1** et **r2** ?

Capturer avec wireshark l'échange lors d'un ping.

Question 10. Exécuter une commande `traceroute` depuis **pc1** vers **pc2**. Commenter la valeur du TTL dans cet échange.

Question 11. D'après la capture, quels sont les protocoles encapsulant les requêtes de la commande `traceroute` ?

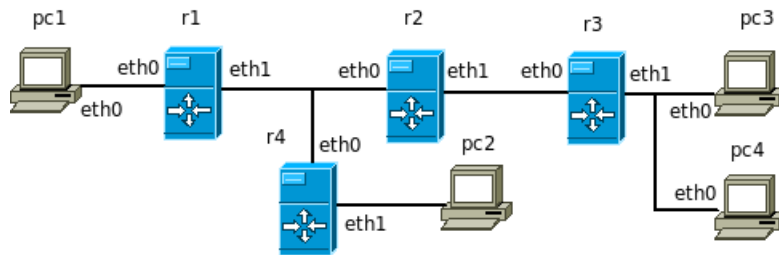
Question 12. Quelle est le nom de la commande équivalente sous *Windows* ? Sachant que la commande sous *Windows* utilise le protocole ICMP, pourquoi la commande `traceroute` de *GNU/Linux* a-t-elle plus de chance de recevoir une réponse que la commande de *Windows* ?

Question 13. En vous aidant d'un `man traceroute`,

- que permet de faire l'option `-I` ?
- Avec l'option `-I 1`, à quoi correspond la valeur 1 ?
- Dans quel fichier a-t-on obtenu cette valeur ?
- Quelle valeur faudrait-il mettre pour utiliser le protocole TCP dans la commande `traceroute` ?

Route par défaut

L'architecture réseau est la suivante :



Ce TP est disponible dans l'archive `tp2b.tgz`.

Le plan d'adressage est le suivant :

- entre r1 et pc1 : 172.16.32.0/24
- entre r1, r2 et r4 : 192.168.10.0/28
- entre r2 et r3 : 192.168.10.16/28
- entre r4 et pc2 : 10.30.0.0/16
- entre r3, pc3 et pc4 : 172.16.45.0/24

Le but est de mettre en place des **routes par défaut** pour chaque **pcx** afin qu'il puisse se « ping ».

Question 14. Mettre les tables de routage de chaque **pcx** avec une seule règle *default* pour communiquer avec son routeur.

Question 15. Fournir les tables de routage de **pc1**, **pc2**, **pc3** et **pc4**.

Question 16. Vérifier avec la commande `tracpath` que chaque machine puisse communiquer avec chaque autre.

Question 17. Ajouter une liaison entre **r1** et **r3** : 10.100.0.0/16.



La commande `vconfig` permet d'affecter une interface à la volée à une machine virtuelle. Cette commande affecte `eth2` à la VM **r1** dans le domaine de collision F : `host> vconfig --eth2=F r1`

Question 18. Comment devez-vous reconfigurer les tables si le routeur **r2** est saturé ou en panne ?

Question 19. Tester en désactivant les deux interfaces de **r2** (`ifconfig ethx down`).

Question 20. La reconfiguration manuelle des tables de routage en cas de défaillance d'un routeur peut vite devenir fastidieuse. Proposer au moins un protocole de routage qui permettrait de mettre à jour dynamiquement les tables.