

Introduction aux Réseaux TP3 :

Interconnexions de réseaux : le problème du routage

Line POUVARET, Mickaël TURNEL

2015-2016

2.1 Manipulation de la table de routage

- Host A a pour adresse IP 192.168.1.1 sur son interface em0 => A1
Routeur B a pour adresse IP 192.168.2.2 sur son interface em0 => B2, 192.168.1.2 sur son interface bge0 => B1
Host C a pour adresse IP 192.168.2.1 sur son interface em0 => C2
- FAIRE TABLEAU! (voir screen1)
- C2 ne peut envoyer un ping vers A1.
A1 ne peut envoyer un ping vers C2.
B peut envoyer un ping vers A1 et C2.
C2 peut envoyer un ping vers B.
A1 peut envoyer un ping vers B.
Donc A1 et C2 ne peuvent communiquer entre eux.
- route add net2 B1
add net net2 : gateway B1
FAIRE TABLEAU (voir screen2 + screen3)
- On remarque bien que quand C2 exécute pong et que A1 exécute un ping sur C2, le ping de A1 bloque car le routeur ne fait pas suivre la requête.
- Après avoir modifié le comportement du routeur, on constate que les paquets envoyés par A1 sur C2 sont bien envoyés et reçus mais C2 ne répond pas à A1.
- Cela est dû au fait que nous n'avons pas mis dans la table de routage de C2 la route vers B puis A1.
Donc on fait :
route add net1 B2
add net net1 : gateway B2
Ceci résout le problème et permet un ping de A vers C (avec réponse de C vers A).
- On lance un ping de A1 vers C2 :
 - La machine A1 envoie un paquet de type ARP request pour récupérer l'adresse MAC de B1.
B1, en recevant ce paquet, lui répond et ajoute lui même A1 dans sa table ARP.
Puis, B2, voulant transmettre les paquets ICMP du ping à C2, envoie un paquet de type ARP request pour récupérer l'adresse MAC de C2.
C2 lui répond et ajoute lui même B2 dans sa table ARP.

Dans la table de A on a les correspondances IP/MAC de B1 et de A1.

Dans la table de B, on a les correspondances IP/MAC de A1, B1, B2 et C2.

Dans la table de C, on a les correspondances IP/MAC de B2 et C2.

- Sur Wireshark, lors d'un ping de A1 vers C2 on observe deux paquets ICMP request et deux paquets ICMP reply :

A1 envoie un paquet à B1 (Adresse MAC source = A1, adresse MAC cible = B1).

B2 transmet ce paquet au destinataire C2 (Adresse MAC source = B2, adresse MAC cible = C2).

C2 répond en envoyant un paquet à B2 (Adresse MAC source = C2, adresse MAC cible = B2).

B1 transmet ce paquet à A1 (Adresse MAC source = B1, adresse MAC cible = A1).

Les adresses Internet pour source et destinataire ne changent pas pour un paquet car ce paquet doit être envoyé d'une certaine machine à une autre.

Alors que pour Ethernet, la source et le destinataire sont forcément respectivement l'adresse de celui qui envoie le paquet et l'adresse de celui qui le reçoit.

- Le travail IP dans le routeur B consiste à transiter le paquet IP reçu sur un réseau vers un autre réseau (ici net1 vers net2).

A1 envoie un paquet IP C2, le paquet arrive à B1. B analyse le destinataire du paquet IP et l'envoie à l'adresse MAC correspondante.

- Il faut ajouter une route par défaut dans la table de routage d'une station si l'on veut que celle-ci puisse communiquer avec le "reste du monde".

- Sur la station A on exécute :

route add default B1

Sur la station C on exécute :

route add default B2

- L'intérêt réside dans le fait que A et C vont envoyer les paquets dont les destinataires sont sur un autre réseau (adresses MAC inconnues). L'inconvénient est que B n'est pas forcément la bonne route à prendre pour ces paquets.
- A1 envoie son paquet à B mais celui-ci lui répond qu'il ne connaît pas la destination de ce paquet (l'adresse MAC).

La différence avec la configuration sans défaut est que si le routeur ne connaît pas la destination d'un paquet envoyé par une station, il lui indiquera que la destination est inatteignable. Avec une configuration sans défaut, la station ne peut pas envoyer de paquet au routeur.

2.2 Observation du routage automatique

- Les tables de routage se sont correctement remplies (même contenu que manuellement). Les stations A et C peuvent communiquer normalement.

- On constate que périodiquement, 2 paquets utilisant le protocole RIPv1 transitent (un de B1 vers le masque du réseau net1, un de B2 vers le masque du réseau net2)

Dans la trame RIPv1 de chaque paquet, on voit qu'il communique l'adresse IP du réseau net1 et net2.

Les paquets RIPv1 sont émis toutes les 30 secondes automatiquement.

- Quand on déconnecte la station A lors de son check-route, au bout d'environ 8 minutes, la route vers le deuxième réseau via B1 est effacé de la table de routage.

La durée du timer d'effacement de RIPv1 est donc de 8 minutes environ pour notre réseau. (un peu plus de 490 secondes pour nous)

- Quand on reconnecte la station A, 30 secondes après, la table affiche une nouvelle entrée : celle de la route vers le deuxième réseau via B1.
- Le routeur envoie sur les stations des réseaux des paquets RIPv1 toutes les 30 secondes.
Si une station ne reçoit pas ce paquet, alors après un certain temps (le timer d'effacement de RIP), celle-ci supprime la/les routes transitant par ce routeur de sa table de routage.
- Au moment où nous avons tué le démon de routage sur B, on capture 4 paquets RIP (envoyés 16 secondes après les derniers paquets RIP envoyés).
Le champ Metric est à 16 pour ceux-ci contrairement à tous les autres paquets RIP où Metric : 1.
- Au moment où nous relançons le démo de routage sur B, on capture 4 paquets RIP.
Dans la trame RIP, on a un champ Address not specified. Ce qui signifie que les machines connectées au routeur sont informées qu'il existe une route passant par B.

2.3 Troisième manipulation

2.3.1 Routage statique

- On a attribué l'adresse 192.168.3.1 (D3) pour la station D du réseau net3
On a attribué l'adresse 192.168.3.2 (C3) pour la station C du réseau net3 (sur bge0).
- Sur la station A :
route add net2 B1
route add net3 B1

Sur la station B :
route add net3 C2

Sur la station C :
route add net1 B2

Sur la station D :
route add net2 C3
route add net1 C3
- Lors d'un ping de A vers D :
 - Ping (ICMP) Request de A1 vers B1
 - Ping (ICMP) Request de B2 vers C2
 - Ping (ICMP) Request de C3 vers D3
 - Ping (ICMP) Reply de D3 vers C3
 - Ping (ICMP) Reply de C2 vers B2
 - Ping (ICMP) Reply de B1 vers A1

2.3.2 Routage automatique

- Les paquets émis lors du lancement des démons servent à ...
- Le paquet qui circule sur net2 possède les adresses IP de net1 et net2. Metric : 16
Il informe donc les machines de net2 d'enlever les routages vers net1 et net2 de la table de routage.
Le paquet qui circule sur net3 possède les adresses IP de net1. Metric : 16
Il informe donc les machines de net3 d'enlever les routages vers net1 de la table de routage.

2.4 Quatrième manipulation

- On a attribué l'adresse 192.168.4.1 (D4) pour la station D du réseau net4 (sur bge0).
On a attribué l'adresse 192.168.4.2 (A4) pour la station A du réseau net4 (sur bge0).
- La table de routage de A (voir screen5) TABLEAU
Pour accéder au réseau 2, le chemin que doit emprunter un paquet envoyé par A passe par D4.
- Le réseau 1 n'est pas connu de C et D car il se trouve entre deux hosts, il est donc isolé.
- Pour accéder au réseau 2, le chemin que doit emprunter un paquet envoyé par A passe maintenant par B1.
- Ce chemin a été choisi car le Metric est inférieur à celui du chemin passant par D4.
- Principe de fonctionnement...

2.5 Cinquième manipulation

- Nous avons configuré nos interfaces comme suit :

A1 : 192.168.1.1

A2 : 192.168.2.1

B2 : 192.168.2.2

B3 : 192.168.3.1

C3 : 192.168.3.2

D3 : 192.168.3.3

D1 : 192.168.1.2

- Lorsqu'on exécute la commande traceroute C3 (à partir de A), on obtient le chemin passant par B2 (Routeur) et arrivant à C3.
Le ping de A vers C a bien été effectué.
Pourquoi la route ?

- Après déconnexion de A du réseau net2, on relance un ping de A vers C qui, pendant quelques minutes, affiche en boucle "No route to host".

Puis, il arrive à effectuer son ping de nouveau en empruntant un nouveau chemin qui passe par D1.

On l'observe bien dans la table de routage → Destination : 192.168.3.0, Gateway : D1

Le ping ne fonctionnait pas tant que le timer d'effacement de RIP n'avait pas expiré. Au moment où celui-ci a expiré, la station A a du retrouver un autre chemin qui ne passait plus par net2 mais par net1.

1.
 - Les paquets RIP émis par des routeurs contiennent les adresses des réseaux auquel ils ont accès + leur Metric (la distance à laquelle ils sont de ces réseaux)
 - Le paquet RIP envoyé sur une interface du routeur ne sera pas le même sur une autre interface du routeur puisqu'il contiendra les réseaux auquel l'interface a accès.
 - RIP mémorise la route la plus courte. Si celle-ci n'existe plus, les timers d'effacement de RIP vont faire qu'il va chercher une nouvelle route (avec le Metric le plus petit).
 - Un routeur calcule la partie réseau de l'adresse des réseaux auxquels il appartient grâce aux masques de réseaux associés aux adresses IP des réseaux.
 - Les paquets RIP ne contiennent pas les masques de réseaux associés aux adresses qu'ils transportent. Les paquets contiennent uniquement les adresses IP des réseaux (Par exemple 192.168.1.0 pour net1, 192.168.2.0 pour net2, etc..).
 - RIP calcule les masques de réseaux grâce au système de classes et les place dans la table de routage.
 - Le traitement d'une ligne contenue dans un paquet RIP ne sera pas le même quelle que soit l'adresse du routeur émetteur du paquet RIP car ce ne sont pas forcément les mêmes Metric et pas non plus les mêmes adresses IP.
 - RIP va simplement mettre à jour le Metric de la route déjà connue (qui a un coût supérieur à celui précédemment annoncé). Si la route a disparu, elle aura donc un Metric de 16 et RIP mettra à jour ce Metric dans les paquets avant de l'enlever de la table de routage.
 - A la réception d'un paquet RIP Request, les routeurs vont envoyer un paquet RIP Response contenant les informations de leur table de routage. L'émetteur du RIP Request va donc pouvoir mettre à jour sa propre table de routage et informer les autres adresses de son réseau de ces changements.
 - Lorsque le démon est arrêté sur une machine, RIP envoie 4 paquets RIP avec un Metric de 16 à toutes les adresses du réseau sur lequel le démon était lancé.

2. Algorithme Protocole RIP (non routeur)

```

tant que vrai faire
    attendre (evenement);
    si évènement == "réception d'un paquet RIP Response" alors
        pour chaque adresse contenue dans le paquet RIP faire
            si La route (Adresse, Routeur) est déjà présente dans la table de routage alors
                /* On met à jour éventuellement notre table de routage (le champ Use par
                   exemple) */
                maj(tableRoutage);
                /* On réinitialise le timer d'effacement de RIP */
                reset(timerRIP);
            fin
            si L'adresse est déjà dans la table de routage alors
                si new.Metric < old.Metric alors
                    /* Si le Metric (new.Metric) est inférieur à celui connu (old.Metric):
                       on met à jour la table de routage avec le routeur qui a new.Metric
                       */
                    maj(tableRoutage);
                fin
                si new.Metric == 16 alors
                    /* Le Metric 16 signifie que la route a disparu */
                    wait(timerRIP);
                    /* L'entrée est supprimée de la table de routage après expiration du
                       timer d'effacement de RIP */
                    enleverTableRoutage(adresse, routeur);
                fin
            fin
            sinon
                si new.Metric != 16 alors
                    /* L'adresse n'est pas dans la table de routage */
                    ajouterTableRoutage(adresse, routeur); init(timerRIP);
                fin
            fin
        fin
    fin
    si évènement == "réception d'un paquet RIP request" alors
        | envoiRIPResponse(tableDeRoutage);
    fin
    si évènement == expiration du timer associé à une entrée dans la table de routage alors
        | new.Metric = 16;
    fin
    si évènement == expiration du timer d'émission de paquet RIP alors
        | envoiRIPBroadcast(tableDeRoutage);
    fin
    si évènement == "Démarrage routeur" alors
        | envoiRIPRequest();
    fin
    si évènement == "Arrêt routeur" alors
        /* Tous les Metric des réseaux dans le paquet RIP Response seront à 16 */
        all.Metric = 16;
        envoiRIPResponse(tableDeRoutage);
    fin
fin

```

Algorithme Protocole RIP (routeur)

```

tant que vrai faire
    attendre (evenement);
    si évènement == "réception d'un paquet RIP Response" alors
        pour chaque adresse contenue dans le paquet RIP faire
            si La route (Adresse, Routeur) est déjà présente dans la table de routage alors
                /* On met à jour éventuellement notre table de routage (le champ Use par
                   exemple) */
                maj(tableRoutage);
                /* On réinitialise le timer d'effacement de RIP */
                reset(timerRIP);
            fin
            si L'adresse est déjà dans la table de routage alors
                si new.Metric < old.Metric alors
                    /* Si le Metric (new.Metric) est inférieur à celui connu (old.Metric):
                       on met à jour la table de routage avec le routeur qui a new.Metric
                       */
                    maj(tableRoutage);
                fin
                si new.Metric == 16 alors
                    /* Le Metric 16 signifie que la route a disparu */
                    wait(timerRIP);
                    /* L'entrée est supprimée de la table de routage après expiration du
                       timer d'effacement de RIP */
                    enleverTableRoutage(adresse, routeur);
                fin
            fin
            sinon
                si new.Metric != 16 alors
                    /* L'adresse n'est pas dans la table de routage */
                    ajouterTableRoutage(adresse, routeur); init(timerRIP);
                fin
            fin
        fin
    fin
    si évènement == "réception d'un paquet RIP request" alors
        | envoiRIPResponse(tableDeRoutage);
    fin
    si évènement == expiration du timer associé à une entrée dans la table de routage alors
        | new.Metric = 16;
    fin
fin

```

3 Pour ceux qui veulent aller un peu plus loin

- h
- e
- a