Réseaux
Bridge

Sommaire

Un pont est un dispositif matériel permettant de relier des réseaux travaillant avec le même protocole.

Contrairement au répéteur, qui travaille au niveau physique, le pont travaille également au niveau logique (niveau 2 du modèle OSI).

Il est capable de filtrer les trames en ne laissant passer que celles dont l'adresse correspond à une machine située sur un autre port.

De la sorte, le pont permet de segmenter un réseau en conservant au niveau du réseau local les trames destinées au niveau local et en transmettant les trames destinées aux autres réseaux.

Cela permet de réduire le trafic et notamment les collisions sur chacun des réseaux.

Cela permet également d'augmenter le niveau de confidentialité car les informations destinées à un réseau ne peuvent pas être écoutées sur un autre brin.

Cependant, l'opération de filtrage réalisée par le pont peut conduit à un léger ralentissement lors du passage d'un réseau à l'autre.

C'est pourquoi les ponts doivent être **judicieusement** placés dans un réseau.

Un pont possède deux connexions à deux réseaux distincts.

Lorsque le pont reçoit une trame sur l'une de ses interfaces, il analyse l'adresse MAC du destinataire et de l'émetteur.

Si jamais le pont ne connaît pas l'émetteur, il stocke son adresse dans une table afin de se "souvenir" de quel côté du réseau se trouve l'émetteur.

Si l'émetteur et le destinataire sont situés du même côté, le pont ignore le message, sinon il transmet la trame sur l'autre réseau.

Un pont possède deux connexions à deux réseaux distincts.

Lorsque le pont reçoit une trame sur l'une de ses interfaces, il analyse l'adresse MAC du destinataire et de l'émetteur.

Si jamais le pont ne connaît pas l'émetteur, il stocke son adresse dans une table afin de se "souvenir" de quel côté du réseau se trouve l'émetteur.

Si l'émetteur et le destinataire sont situés du même côté, le pont ignore le message, sinon il transmet la trame sur l'autre réseau.

Un pont fonctionne selon la couche Liaison de données du modèle OSI, et opére au niveau des adresses physiques des machines.

Le pont, relié à plusieurs réseaux locaux appelés segments, élabore une table de correspondance entre les adresses des machines et le segment auquel elles appartiennent.

Lors d'une transmission de données, le pont vérifie sur sa table de correspondance le segment auquel appartiennent les ordinateurs émetteurs et récepteurs grâce à leur adresse physique, appelée adresse MAC, et non leur adresse IP.

Si ceux-ci appartiennent au même segment, le pont ne fait rien, dans le cas contraire il va faire basculer les données vers le segment auquel appartient le destinataire.

Un pont maintien l'heure d'arrivée (avec mise à jour continue) des trames dans les tables de correspondance.

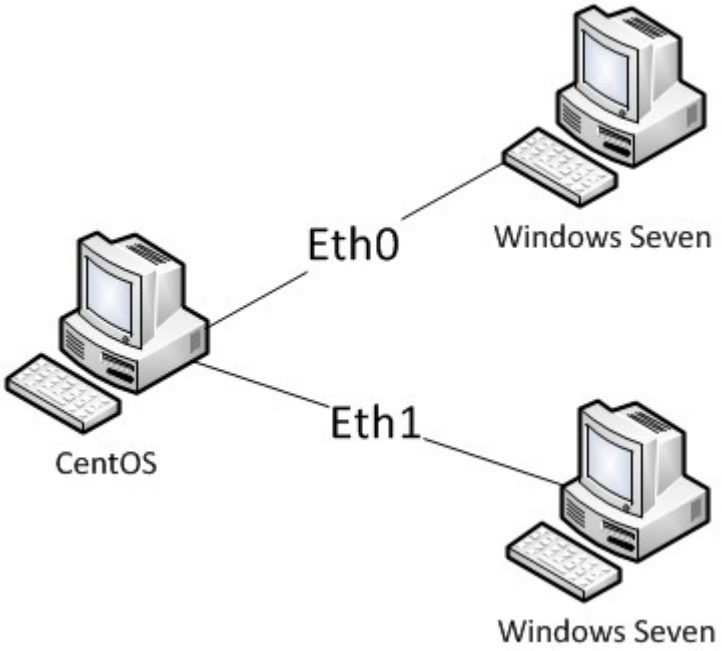
De la sorte il peut invalider certaines entrées périmées et ainsi gérer l'arrêt ou le déplacement de stations dans le réseau.

Il doit laisser passer les messages de diffusion ou multicast.

L'algorithme de fonctionnement extrait l'adresse destination de la trame et :

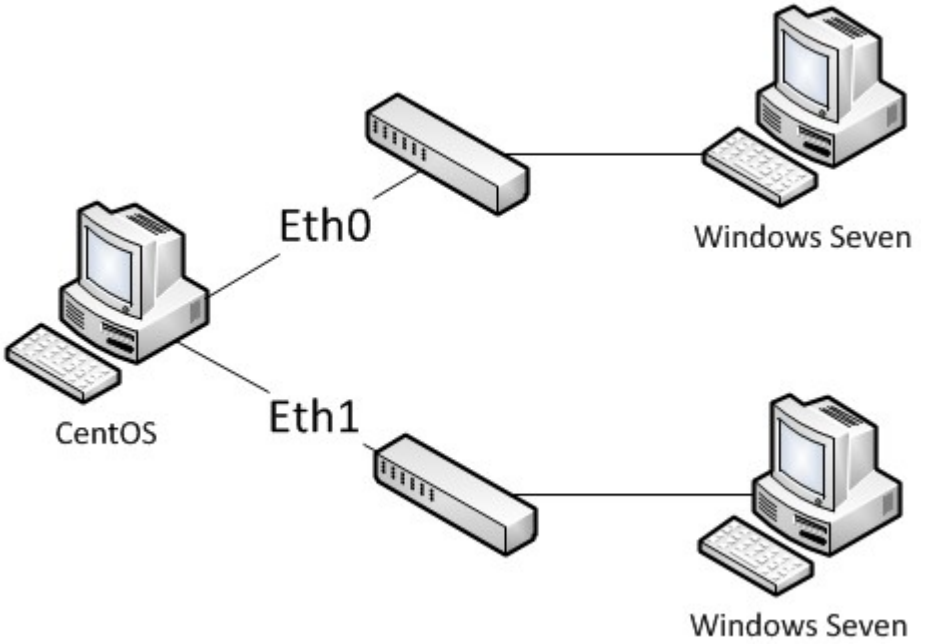
- si aucune entrée n'est trouvée dans la table de correspondance, il réémet la trame sur tous les segments sauf le segment émetteur ;
- sinon il achemine la trame vers le segment correcte.

Le pont ne fonctionnera plus s'il existe des boucles car le nombre de retransmissions est alors infini lorsque le destinataire est inconnu : Il faut utiliser ***Spanning Tree***.



Le montage le plus simple est celui avec des câbles croisés car il ne nécessite aucun autre équipement.

S'il n'y a que des câbles droits, il faudra obligatoirement introduire un équipement qui va croiser les paires d'émission et de réception



Tout d'abord, il faut installer le paquetage 'bridge-utils':

```
# yum install bridge-utils
```

Une fois ce paquetage présent, lancer la commande suivante pour vous assurer de la bonne configuration du noyau:

```
# sysctl -p
    net.ipv4.ip_forward = 0
    ...
    net.bridge.bridge-nf-call-ip6tables = 0
    net.bridge.bridge-nf-call-iptables = 0
    net.bridge.bridge-nf-call-arptables = 0
```

Le premier paramètre signifie que le forwarding ip n'est pas actif et les autres paramètres permettent de 'débrayer' le passage des trames dans NetFilter (iptables)

Il est possible d'appliquer une configuration temporaire:

```
# brctl addbr br0
```

Cette ligne indique que nous créons (*addbr*) un bridge nommé 'br0'.

Ensuite il ne reste plus qu'à ajouter des interfaces au bridge:

```
# brctl addif br0 eth0;
# brctl addif br0 eth1;
# …
```

Pour être sûr que cela fonctionne, il faut absoluement déconfigurer le niveau 3 OSI des interface en appliquant:

```
# ifconfig ethX 0.0.0.0
```

Il est possible d'appliquer une configuration permanente en utilisant les fichiers de configuration 'ifcfg-ethX' présent dans le répertoire '/etc/sysconfig/network-scripts'. Ces fichiers doivent contenir les lignes suivantes :

```
DEVICE=ethX
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
BRIDGE=br0
```

La dernière ligne indique que l'interface est attachée au bridge 'br0'.

Il ne reste plus qu'à créer le fichier de configuration du brdige 'br0':

```
# touch ifcfg-br0
```

Et y placer les lignes suivantes:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.50.254
NETMASK=255.255.255.0
```

Ne pas oublier d'appliquer la nouvelle configuration :

```
# service network restart
```

Vérifions la bonne configuration du bridge:

 # brctl show

bridge name	bridge id	STP enabled	interfaces
br0	8000.000c291db74d	no	eth0
			eth2
			eth3

Contrôlons les adresse MAC apprisent :

 # brctl showmacs br0

port no	mac addr	is local?	ageing timer
3	**00:0c:29:1a:3e:2e**	**no**	**13.10**
1	00:0c:29:1d:b7:4d	yes	0.00
3	00:0c:29:1d:b7:57	yes	0.00
2	00:0c:29:1d:b7:61	yes	0.00
2	**00:0c:29:b2:43:b7**	**no**	**0.09**
1	**08:60:6e:6f:8b:46**	**no**	**44.78**
1	**4c:72:b9:e2:4d:d2**	**no**	**0.00**

Pour ajouter du filtrage, il suffit de modifier le fichier '/etc/sysctl.conf' et notament la ligne suivante:

net.bridge.bridge-nf-call-iptables = **1**

On peut vérifier en utilisant la commande suivante:

```
# iptables -nvL FORWARD
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  4 240 REJECT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           reject-with icmp-
host-prohibited
```

On voit clairement le nombre de paquets '***droper***' de la chaîne ***FORWARD*** qui augmente !

Comme on pouvait s'y attendre, lorsque l'on fait un 'ping' sur une machine connecté à un port de ce bridge, on reçoit un ***ICMP Reject*** de la part du bridge:

```
C:\Users\user>ping 192.168.50.253

Envoi d'une requête 'Ping'  192.168.50.253 avec 32 octets de données :
Réponse de 192.168.50.254 : Impossible de joindre l'hôte de destination.
Réponse de 192.168.50.254 : Impossible de joindre l'hôte de destination.
Réponse de 192.168.50.254 : Impossible de joindre l'hôte de destination.
Réponse de 192.168.50.254 : Impossible de joindre l'hôte de destination.

Statistiques Ping pour 192.168.50.253:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),

C:\Users\user>_
```

Il faudra obligatoirement modifier la chaîne '***FORWARD***' de la table ***filter*** pour rétablir le passage du traffic.

Ce montage réseau s'appelle une ***passerelle transparente*** !

La commande ***brctl*** controle finement le fonctionnement du bridge et les options suivantes peuvent être utiles :

- ***setageing***: permet de fixer le temps en seconde qu'une adresse MAC sera conservée dans la base de données de 'forwarding' après avoir reçu un paquet de cette adresse MAC. Les entrées de cette table sont périodiquement effécées pour assurer qu'elle ne vont pas restée eternellement.

- ***setbridgeprio***: permet de définir la priorité relative d'un bridge. Le bridge avec la priorité la plus faible sera élu 'root' et aura une position centrale lors de l'utilisation du **Spanning Tree**.

- **sethello**: permet de fixer l'intervalle en seconde d'envois des paquets 'Hello'. Ces paquets sont utilisés par le protocole **Spanning Tree** pour communiquer les informations de topologie.

- ***setmaxage***: permet de fixer le temps maximal de reception du message hello. Si ce temps et dépassé, le bridge en question va commencer la procédure d'élection pour devir le bridge 'root'

- ***setpathcost***: permet de définir le coût d'envois d'un paquet sur une interface. Plus l'interface est rapide, plus sont coût doit être faible. Les chemins avec des coût faible seront retenu dans l'élaboration de l'arborescence ***Spanning Tree***.

- **stp**: permet d'activer / désactiver le ***Spanning Tree Protocol***.