

# Boucles de routage

## Introduction

Une boucle de routage est le résultat d'une mauvaise configuration des routes où deux routeurs se pointent mutuellement pour la même destination. Ceci peut découler d'une mauvaise configuration de routes statiques ou d'une erreur de configuration dans un protocole de routage à vecteur de distance tel que RIP ou EIGRP.

Une boucle de routage peut être aussi simple que deux routeurs qui se pointent mutuellement ce qui est généralement rapide à diagnostiquer mais peut aussi impliquer plusieurs routeurs. Lorsqu'un protocole de routage dynamique à vecteur de distance est configuré, plus il y aura de routeurs au sein d'un réseau, plus il y aura de chances de créer une boucle de routage advenant le cas où deux routeurs annonceraient le même sous-réseau IP et plus il sera difficile de diagnostiquer la situation. Il est donc impératif de bien planifier et bien documenter les sous-réseaux utilisés pour éviter ce genre de situation.

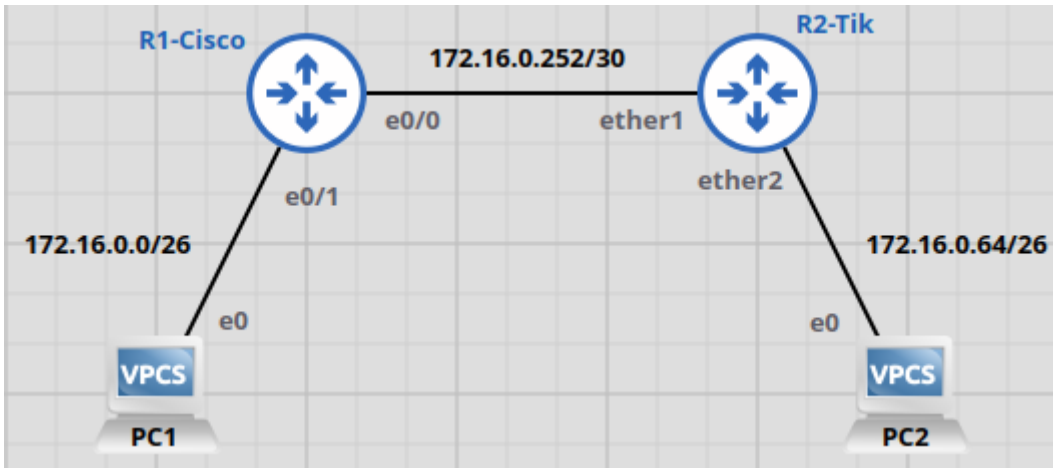
Heureusement, contrairement à une boucle de commutation, un paquet IP a un délai maximal d'existence et sera abandonné après un certain nombre de sauts évitant ainsi de complètement saturer les liens entre les routeurs.

## Exemples

### Routage statique a deux routeurs

Puisqu'une route directement connectée a toujours une distance de 0 et sera toujours prioritaire à tout autre type de route, ces réseaux ne seront généralement pas victimes d'une boucle de routage. On peut observer ceci en configurant deux réseaux derrière deux routeurs qui se pointeront mutuellement pour leur route par défaut (*0.0.0.0/0*). Les clients des réseaux situés derrière ces routeurs pourront communiquer ensemble mais toute tentative de contact envers d'autres destination causera une boucle de routage.

Imaginons le scénario suivant. Le routeur R1 est directement connecté aux réseaux 172.16.0.0/26 et 172.16.0.252/30. Ceci implique que la distance de ces routes est 0 et ces routes auront priorité. Idem pour R2 qui est directement connecté à 172.16.0.64/26 et 172.16.0.252/30. Ils pourront donc correctement acheminer le trafic destiné à ces deux réseaux.



Toutefois, la route par défaut de R1 pointe vers R2 et la route par défaut de R2 pointe vers R1. Les deux PC seront en mesure de communiquer ensemble mais toute tentative de communication à l'extérieur de ces trois sous-réseaux causera une boucle de routage. On peut observer à l'aide d'une capture de paquets entre les deux routeurs que la requête ainsi que la réponse sont correctement acheminées.

L'extrait de terminal suivant est le résultat d'un ping de PC1 vers PC2.

```
PC1> ping 172.16.0.66

84 bytes from 172.16.0.66 icmp_seq=1 ttl=62 time=15.451 ms
```

Source	Destination	Protocol	Info
172.16.0.2	172.16.0.66	ICMP	Echo (ping) request
172.16.0.66	172.16.0.2	ICMP	Echo (ping) reply

Toutefois, si un ping est envoyé à une destination qui n'est pas directement connectée à un de ces routeurs, on pourra observer que les deux routeurs se relanceront continuellement la requête jusqu'à son délai d'expiration. On observe ici que le "Time To Live" a expiré en chemin.

```
PC1> ping 8.8.8.8

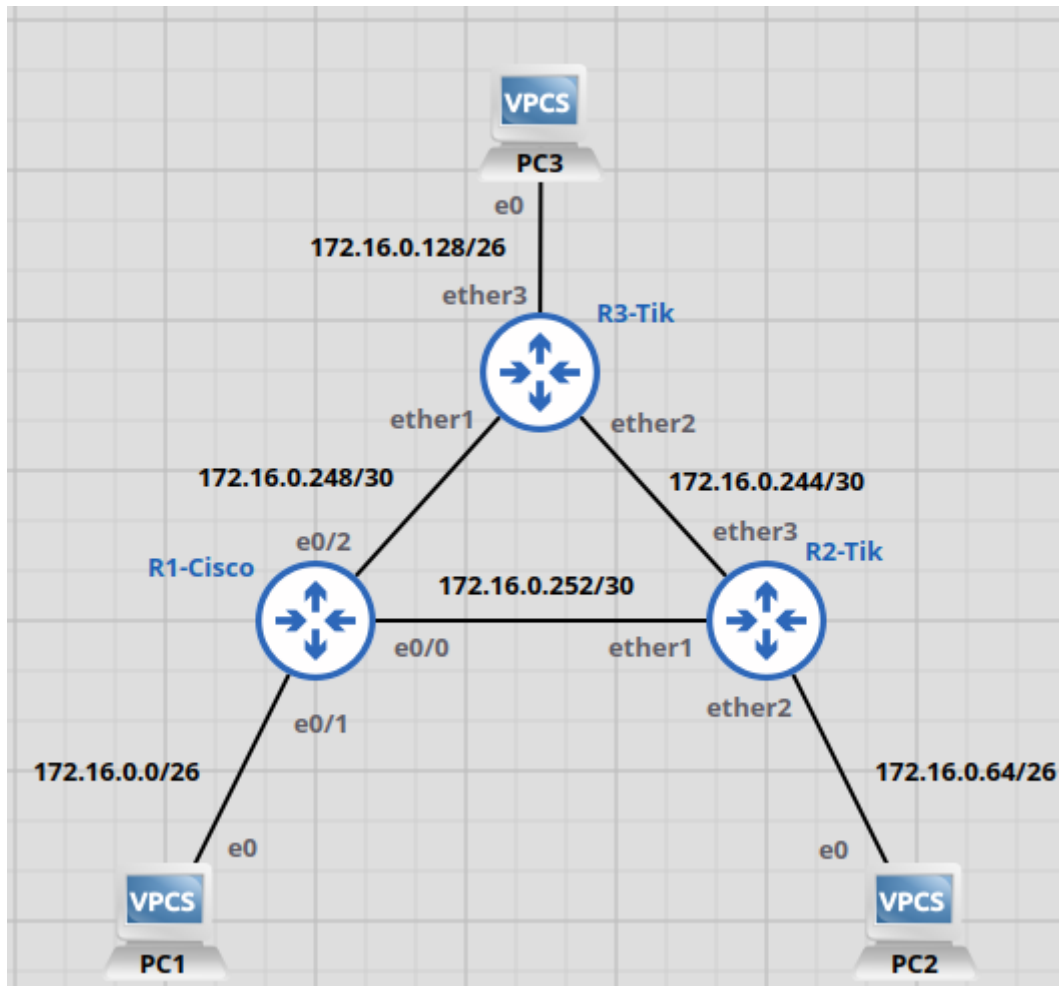
*172.16.0.254 icmp_seq=1 ttl=63 time=31.106 ms (ICMP type:11, code:0, TTL expired in transit)
```

Dans la capture d'écran suivante, on peut observer que les adresses MAC de source et de destination s'alternent à chaque saut. Il s'agit de chaque routeur qui se renvoie la requête jusqu'à son expiration.

Source	Destination	Source	Destination	Info
172.16.0.2	8.8.8.8	0c:e5:0c:3a:00:00	aa:bb:cc:00:01:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=10 (no response found!)
172.16.0.2	8.8.8.8	aa:bb:cc:00:01:00	0c:e5:0c:3a:00:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=9 (no response found!)
172.16.0.2	8.8.8.8	0c:e5:0c:3a:00:00	aa:bb:cc:00:01:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=8 (no response found!)
172.16.0.2	8.8.8.8	aa:bb:cc:00:01:00	0c:e5:0c:3a:00:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=7 (no response found!)
172.16.0.2	8.8.8.8	0c:e5:0c:3a:00:00	aa:bb:cc:00:01:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=6 (no response found!)
172.16.0.2	8.8.8.8	aa:bb:cc:00:01:00	0c:e5:0c:3a:00:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=5 (no response found!)
172.16.0.2	8.8.8.8	0c:e5:0c:3a:00:00	aa:bb:cc:00:01:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=4 (no response found!)
172.16.0.2	8.8.8.8	aa:bb:cc:00:01:00	0c:e5:0c:3a:00:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=3 (no response found!)
172.16.0.2	8.8.8.8	0c:e5:0c:3a:00:00	aa:bb:cc:00:01:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=2 (no response found!)
172.16.0.2	8.8.8.8	aa:bb:cc:00:01:00	0c:e5:0c:3a:00:00	Echo (ping) request id=0xa6ae, seq=1/256, ttl=1 (no response found!)
172.16.0.254	172.16.0.2	0c:e5:0c:3a:00:00	aa:bb:cc:00:01:00	Time-to-live exceeded (Time to live exceeded in transit)

## Routage statique à trois routeurs

Dans la situation suivante, R1 pointe vers R3 pour sa route par défaut, R3 vers R2 et R2 vers R1. Lorsque le PC2 tente de joindre le PC3, on voit sa requête passer entre R2 et R1 puis entre R1 et R3. La réponse passe ensuite de R3 à R2. Quoi que fonctionnel, cette situation se nomme du routage asymétrique.



```
PC2> ping 172.16.0.130 -c 1
```

```
84 bytes from 172.16.0.130 icmp_seq=1 ttl=62 time=12.107 ms
```

\*- [R1-Cisco Ethernet0/0 to R2-Tik ether1]

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

icmp

Source	Destination	Source	Destination	Info
172.16.0.66	172.16.0.1...	0c:e5:0c:3a:00:00	aa:bb:cc:00:01:00	Echo (ping) request

\*- [R1-Cisco Ethernet0/2 to R3-Tik ether1]

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

icmp

Source	Destination	Source	Destination	Info
172.16.0.66	172.16.0.1...	aa:bb:cc:00:01:20	0c:5a:63:62:00:00	Echo (ping) request

\*- [R2-Tik ether3 to R3-Tik ether2]

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

icmp

Source	Destination	Source	Destination	Info
172.16.0.130	172.16.0.66	0c:5a:63:62:00:01	0c:e5:0c:3a:00:02	Echo (ping) reply

En cas de tentative de contact vers une adresse qui n'est pas directement connectée à un de ces trois routeurs, on pourra encore une fois observer que la requête ne fait que circuler entre les trois routeurs. Notez ici le TTL qui diminue de 3 à chaque fois que la requête repasse entre les routeurs R3 et R2. Ceci est causé par le fait que la requête a effectué deux autres sauts avant de repasser par ce lien.

Source	Destination	Source	Destination	Info
172.16.0.66	8.8.8.8	0c:5a:63:62:00:01	0c:e5:0c:3a:00:02	Echo (ping) request id=0x22b7, seq=5/1280, ttl=10 (no response found!)
172.16.0.66	8.8.8.8	0c:5a:63:62:00:01	0c:e5:0c:3a:00:02	Echo (ping) request id=0x22b7, seq=5/1280, ttl=7 (no response found!)
172.16.0.66	8.8.8.8	0c:5a:63:62:00:01	0c:e5:0c:3a:00:02	Echo (ping) request id=0x22b7, seq=5/1280, ttl=4 (no response found!)
172.16.0.66	8.8.8.8	0c:5a:63:62:00:01	0c:e5:0c:3a:00:02	Echo (ping) request id=0x22b7, seq=5/1280, ttl=1 (no response found!)

Le résultat de la commande indiquera encore que le délai d'expiration a été atteint.

```
PC2> ping 8.8.8.8

*172.16.0.65 icmp_seq=1 ttl=64 time=61.277 ms (ICMP type:11, code:0, TTL expired in transit)
```

[Projet GNS3 adapté à cette machine virtuelle incluant GNS3](#)