

Le routage dynamique RIP

Objectifs

Mettre en œuvre le **routage IP** sur 3 réseaux reliés par deux routeurs (**R1** et **R2**) sous Linux (Debian 9). On utilisera d'abord le routage **statique**, puis le routage **dynamique** au moyen du protocole **RIP** et l'interface **Quagga** sous Linux.

On utilisera deux postes de travail sous Linux (Debian 9).

MISE EN PLACE DU RESEAU

Configuration initiale

On utilisera 3 réseaux :

- réseau **gauche** : 192.168.1.0/24
- **inter-réseau** : 172.16.0.0/16
- réseau **droite** : 192.168.2.0/24

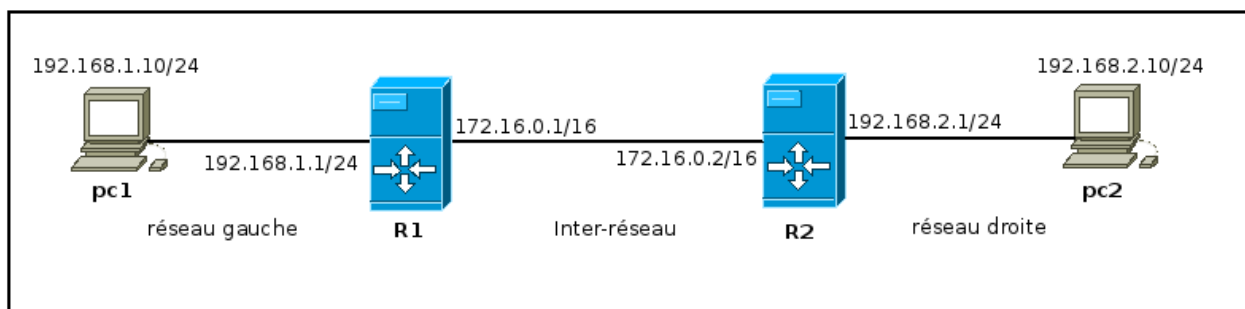
avec :

- poste de travail Linux (**pc1**) (192.168.1.10)
- serveur Linux (**R1**) (enp0s3 : 192.168.1.1 et enp0s8 : 172.16.0.1)
- serveur Linux (**R2**) (enp0s3 : 172.16.0.2 et enp0s8 : 192.168.2.1)
- poste de travail Linux (**pc2**) (192.168.2.10)

Au début, le routage n'est activé sur aucune des machines passerelles (**R1** et **R2**)

Les adresses de passerelles par défaut ne sont pas configurées.

Schéma



Virtualisation

- quatre machines virtuelles (4 Linux)
- les deux machines faisant office de routeur devront avoir deux cartes !
- nommer les machines (pc1, pc2, R1, R2) avec la commande :

```
sed -i 's/stretch64/pc1/g' /etc/host{s,name}
```

 pour le poste pc1
et reboot.
- au départ, on utilisera des cartes configurées en **mode NAT ou pont** de manière à installer tous les paquetages (**tcpdump, quagga, telnet, mc, net-tools ...**) sur **R1 et R2**.
- pour la mise en oeuvre du TP, une fois les paquets installés on utilisera les **réseaux internes VirtualBox** nommés précédemment.

Outils

- `ifconfig` ou `ip` (Linux) pour configurer/vérifier la configuration IP des machines
- `route` ou `ip` : pour afficher/modifier le contenu des tables de routage
- `traceroute` pour vérifier le chemin emprunté par les paquets
- `tcpdump`, `etherreal` pour observer les paquets transitant par une interface
- les paquets `quagga` (pour le protocole RIP sous Linux)
- `telnet` (pour se connecter aux serveurs zebra/ripd)
- et bien sûr la commande `ping`

PLAN DE L'ATELIER

1. Configuration des interfaces (en dynamique avec `ifconfig` ou `ip`, sans éditer les fichiers de configuration)
2. Vérification de la connectivité
3. Paramétrage des passerelles par défaut sur les postes de travail
4. Activation du routage sur les passerelles R1 et R2
5. Création des routes statiques sur les interfaces passerelles
6. Tests avec `ping` et `traceroute`
7. Routage dynamique avec RIP

1. Configuration des interfaces de chacune des machines

- sur la machine **pc1**, paramétrer l'adresse ip :
`ip addr add 192.168.1.10/24 dev enp0s3`
- vérifier : **ip a**

```
root@pc1:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:9a:b3:f0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.10/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9a:b3f0/64 scope link
        valid_lft forever preferred_lft forever
```

- Vérifier la table de routage sur la machine **pc1** avec la commande `ip r`

```
root@pc1:~# ip r
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.10
```

Ou `route -n` (si `net-tools` a été installé) :

```
root@pc1:~# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
0.0.0.0          10.0.2.2        0.0.0.0          UG    0     0         0 enp0s3
10.0.2.0         0.0.0.0         255.255.255.0   U     0     0         0 enp0s3
192.168.1.0     0.0.0.0         255.255.255.0   U     0     0         0 enp0s3
```

Configurer les interfaces des autres machines pc2, R1, R2 conformément au schéma de réseau ci-dessus. Vérifier avec ip a

```
root@R1:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:7d:e9:28 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7d:e928/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:79:be:6d brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.1/16 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe79:be6d/64 scope link
        valid_lft forever preferred_lft forever

root@R2:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:0a:33:9b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 172.16.0.2/16 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0a:339b/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:fa:2f:93 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.1/24 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fefa:2f93/64 scope link
        valid_lft forever preferred_lft forever
```

Remarque : vérifier que les interfaces ont le statut UP sinon taper : ip link set enp0sx up (x=3 ou 8)

2. Test de connectivité des machines entre elles avec ping

Connectivité pc1-R1

- ping pc1 ⇒ R1 sur l'interface réseau gauche (192.168.1.1)
- ping pc1 ⇒ R1 sur l'interface réseau inter-réseau (172.16.0.1)

Connectivité R1-R2

- ping R1 ⇒ R2 sur l'interface réseau inter-réseau (172.16.0.2)
- ping R1 ⇒ R2 sur l'interface réseau droite (192.168.2.1)

Connectivité R2-pc2

- ping R2 ⇒ pc2 (192.168.2.10)
- ping Pc2 ⇒ R2 sur l'interface réseau inter-réseau (172.16.0.2)

3. Paramétrage des passerelles par défaut sur les clients

Sur la machine pc1

- taper : `ip route add default via 192.168.1.1`

Remarque : s'il existe déjà une passerelle par défaut (voir la route précédente pour pc1), supprimer la : `ip route del default`

- vérifier les routes avec la commande `ip r`

```
root@pc1:~# ip r
default via 192.168.1.1 dev enp0s3
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.10
```

Ou route `-n` :

```
root@pc1:~# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref     Use Iface
0.0.0.0           192.168.1.1     0.0.0.0          UG    0      0       0 enp0s3
10.0.2.0         0.0.0.0         255.255.255.0   U     0      0       0 enp0s3
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0       0 enp0s3
```

Sur la machine pc2

- taper : `ip route add default via 192.168.2.1`
- vérifier les routes avec la commande `ip r`

```
root@pc2:~# ip route add default via 192.168.2.1
root@pc2:~# ip r
default via 192.168.2.1 dev enp0s3
192.168.2.0/24 dev enp0s3 proto kernel scope link src 192.168.2.10
```

4. Activation du routage sur les passerelles

Sur les machines R1 et R2

- taper la commande suivante :

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

ou bien encore : `sysctl net.ipv4.ip_forward=1`

5. Création des routes statiques sur les passerelles

Sur la machine R2

- taper la commande suivante : `ip route add 192.168.1.0/24 via 172.16.0.1`
- vérifier la table de routage avec : `ip r`

```
root@R2:~# ip r
172.16.0.0/16 dev enp0s3 proto kernel scope link src 172.16.0.2
192.168.1.0/24 via 172.16.0.1 dev enp0s3
192.168.2.0/24 dev enp0s8 proto kernel scope link src 192.168.2.1
```

Sur la machine R1

- taper la commande suivante :

```
ip route add 192.168.2.0/24 via 172.16.0.2
```

- vérifier la table de routage avec : `ip r`

```
root@R1:~# ip route add 192.168.2.0/24 via 172.16.0.2
root@R1:~# ip r
172.16.0.0/16 dev enp0s8 proto kernel scope link src 172.16.0.1
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.1
192.168.2.0/24 via 172.16.0.2 dev enp0s8
```

6. Tests avec ping et traceroute

Depuis la machine pc1

- taper `ping 192.168.2.10` et vérifier le fonctionnement effectif du routage
- taper `traceroute 192.168.2.10` et vérifier que les paquets passent effectivement par les deux passerelles :

```
root@pc1:~# traceroute 192.168.2.10
traceroute to 192.168.2.10 (192.168.2.10), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.120 ms  0.826 ms  0.671 ms
 2  172.16.0.2 (172.16.0.2)  4.201 ms  4.126 ms  3.993 ms
 3  192.168.2.10 (192.168.2.10)  5.301 ms  5.167 ms  5.027 ms
```

Depuis la machine pc2

- taper `ping 192.168.1.10` et vérifier le fonctionnement effectif du routage
- taper `traceroute 192.168.1.10` et vérifier que les paquets passent effectivement par les deux passerelles

```
root@pc2:~# traceroute 192.168.1.10
traceroute to 192.168.1.10 (192.168.1.10), 30 hops max, 60 byte packets
 1  192.168.2.1 (192.168.2.1)  1.119 ms  0.745 ms  1.781 ms
 2  172.16.0.1 (172.16.0.1)  1.647 ms  1.505 ms  1.121 ms
 3  192.168.1.10 (192.168.1.10)  6.577 ms  6.315 ms  6.160 ms
```

7. Routage dynamique avec RIP (Routing Information Protocol)

Dans ce cas de figure, les routes vers les réseaux distants ne seront plus créées par nous-même mais automatiquement par les deux passerelles qui s'enverront des informations de routage grâce au protocole RIP. Il faudra auparavant supprimer les deux routes statiques distantes (ip route del ...) sur chacune des deux passerelles ou bien redémarrer les machines.

- Soit en supprimant les routes distantes dans R1 et R2 :

ip route del 192.168.2.0/24 via 172.16.0.2 dev enp0s8 sur R1

```
root@R1:~# ip r
172.16.0.0/16 dev enp0s8 proto kernel scope link src 172.16.0.1
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.1
192.168.2.0/24 via 172.16.0.2 dev enp0s8
root@R1:~# ip route del 192.168.2.0/24 via 172.16.0.2 dev enp0s8
root@R1:~# ip r
172.16.0.0/16 dev enp0s8 proto kernel scope link src 172.16.0.1
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.1
```

ip route del 192.168.1.0/24 via 172.16.0.1 dev enp0s3 sur R2

```
root@R2:~# ip r
172.16.0.0/16 dev enp0s3 proto kernel scope link src 172.16.0.2
192.168.1.0/24 via 172.16.0.1 dev enp0s3
192.168.2.0/24 dev enp0s8 proto kernel scope link src 192.168.2.1
root@R2:~# ip route del 192.168.1.0/24 via 172.16.0.1 dev enp0s3
root@R2:~# ip r
172.16.0.0/16 dev enp0s3 proto kernel scope link src 172.16.0.2
192.168.2.0/24 dev enp0s8 proto kernel scope link src 192.168.2.1
```

- Soit en rebootant les machines, les tables de routage se vident. Par contre, il faut reconfigurer les interfaces à l'aide de l'history et activer le routage : echo « 1 » > ... sur R1 :

```
root@stretch64:~# ip addr add 172.16.0.1/16 dev enp0s8
root@stretch64:~# ip addr add 192.168.1.1/24 dev enp0s3
root@stretch64:~# ip -4 -o addr show
1: lo      inet 127.0.0.1/8 scope host lo\          valid_lft forever preferred_lft f
rever
2: enp0s3  inet 192.168.1.1/24 scope global enp0s3\      valid_lft forever pr
eferred_lft forever
3: enp0s8  inet 172.16.0.1/16 scope global enp0s8\      valid_lft forever pre
ferred_lft forever
root@stretch64:~# ip r
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.1
```

Sur chacune des machines R1 et R2 (aide avec [zebra.rip.pdf](#)) :

- créer le fichier **/etc/quagga/zebra.conf**

```
hostname Rx
password foo
```

où x est le numéro de routeur, foo étant le mot de passe

- créer le fichier **/etc/quagga/ripd.conf** selon le même principe :

```
hostname Rx
password foo
```

- créer le fichier **/etc/quagga/daemons** pour avoir :

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

- récupérer sur Store le fichier **majq** et copier le dans **/etc/quagga**
Contenu de majq :

```
chown quagga:quagga /etc/quagga/ripd.conf
chmod 640 /etc/quagga/ripd.conf
chown quagga:quagga /etc/quagga/zebra.conf
chmod 640 /etc/quagga/zebra.conf
unlink /etc/systemd/system/network.target.wants/ripd.service
unlink /etc/systemd/system/network.target.wants/zebra.service
#
ln -st /etc/systemd/system/network.target.wants /lib/systemd/system/ripd.service
ln -st /etc/systemd/system/network.target.wants /lib/systemd/system/zebra.service
```

- exécuter le fichier majq :

```
chmod 777 majq    ou chmod +x majq
bash majq
```

- relancer le service (daemon) :

```
systemctl restart zebra
```

- le service d'administration de **R1** écoute sur le port 2601, tester la connexion au serveur zebra avec :

```
telnet localhost 2601    (mdp : foo)    ctrl-D ou exit pour sortir
```

- Idem pour le serveur ripd :

```
systemctl restart ripd
telnet localhost 2602    (mdp : foo)
```

Faire les mêmes manipulations sur R2

Notes :

- Regarder le fichier zebra.rip.pdf pour activer le RIP sur R1 et R2 (page 6)
- Pour visualiser les paquets RIP, faire une capture de paquets avec tcdump. Taper :
tcpdump -i enp0sx -p 520

où enp0sx est l'interface qui se trouve sur l'inter-réseau et 520 le port utilisé par le protocole RIP.

- Activer RIP sur R1 et R2 :

Activation de RIP sur le routeur R1

Connectons-nous au routeur RIP sur R1 avec un telnet sur le port 2602. Dans le shell de R1 :

```
R1 # telnet localhost 2602
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello, this is zebra (version 0.91a).
Copyright 1996-2001 Kunihiro Ishiguro.
User Access Verification
```

Password:

R1 >

```
root@R1:/etc/quagga# telnet localhost 2602
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 1.1.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
R1> en
R1# conf t
R1(config)# router rip
R1(config-router)# _
```

Cette opération étant réalisée, il faut activer le mode privilégié, passer dans le terminal de configuration et enfin, entrer dans la configuration du routeur RIP :

```
R1> enable
R1# conf t
R1(config)# router rip
R1(config-router)# version 2
```

Nous choisissons de diffuser les adresses des réseaux directement connectés :

```
R1(config-router)# redistribute connected
```

Pour l'instant, rien ne se produit. Il faut indiquer à RIP sur quels réseaux nous souhaitons voir la diffusion des routes s'opérer :

```
R1(config-router)# network 172.16.0.0/16          (réseau inter-réseau)
```

À ce stade, R1 diffuse sur le réseau 172.16.0.0/16 la table RIP toutes les 30 secondes. Le résultat sur R2 doit ressembler à ceci :

```
root@R2:~# tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
19:52:17.018896 IP 172.16.0.1.route > 224.0.0.9.route: RIPv2, Response, length:
24
```

Les messages adressés par R1 se font via une adresse multicast convenue pour les routeurs RIP : 224.0.0.9

Observons sur R1 la table avec laquelle RIP travaille :

```
R1# show ip rip :
R1# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From          Tag Time
C(i) 172.16.0.0/16     0.0.0.0           1 self              0
C(r) 192.168.1.0/24   0.0.0.0           1 self              0
```

RIP a été activé sur le réseau 172.16.0.0/16

Activation de RIP sur le routeur R2

```
R2> en
R2# conf t
R2(config)# router rip
R2(config-router)# redistribute connected
R2(config-router)# network 172.16.0.0/16
R2(config-router)# end
R2# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network                Next Hop           Metric From        Tag Time
C(r) 10.0.2.0/24             0.0.0.0            1 self              0
C(i) 172.16.0.0/16           0.0.0.0            1 self              0
R(n) 192.168.1.0/24          172.16.0.1         2 172.16.0.1        0 02:52
C(r) 192.168.2.0/24          0.0.0.0            1 self              0
```

La table ci-dessus a été constituée par le processus RIP exécuté sur R2. Le routeur d'adresse 172.16.0.1 (R1) l'a informé de la présence d'une route vers le réseau 192.168.1.0/24

La distance (Metric) est égale à deux puisque ce réseau est directement connecté à R1.

Vous pouvez lancer la commande show ip rip sur R1 afin de constater qu'il a effectué un travail similaire :

- Avant l'activation sur R2 pour R1 :

```
R1# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network                Next Hop           Metric From        Tag Time
C(i) 172.16.0.0/16           0.0.0.0            1 self              0
C(r) 192.168.1.0/24          0.0.0.0            1 self              0
```

- Après l'activation sur R2 pour R1 :

```
R1# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network                Next Hop           Metric From        Tag Time
R(n) 10.0.2.0/24             172.16.0.2         2 172.16.0.2        0 02:44
C(i) 172.16.0.0/16           0.0.0.0            1 self              0
C(r) 192.168.1.0/24          0.0.0.0            1 self              0
R(n) 192.168.2.0/24          172.16.0.2         2 172.16.0.2        0 02:44
```

- Tester le routage depuis les postes **pc1** et **pc2** ; on procédera comme auparavant avec les commandes ping et traceroute.

Mise en évidence du fonctionnement du routage dynamique

- sur **R1**, désactiver l'interface coté réseau gauche (statut DOWN).
- sur **R2**, constater la disparition de la route menant au réseau gauche après quelques temps.

Extensions possibles

- Filtrage de la diffusion des routes
- Tolérance aux pannes
- sécurisation

Liens et références

- un tutoriel Quagga : <http://openmaniak.com/fr/quagga.php>
- les documents de Philippe Latu et P. Massol sur le site **inetdoc** :
<http://www.inetdoc.net/guides/>
 - ✓ zebra.statique.pdf
 - ✓ zebra.rip.pdf
 - ✓ zebra.ospf.pdf