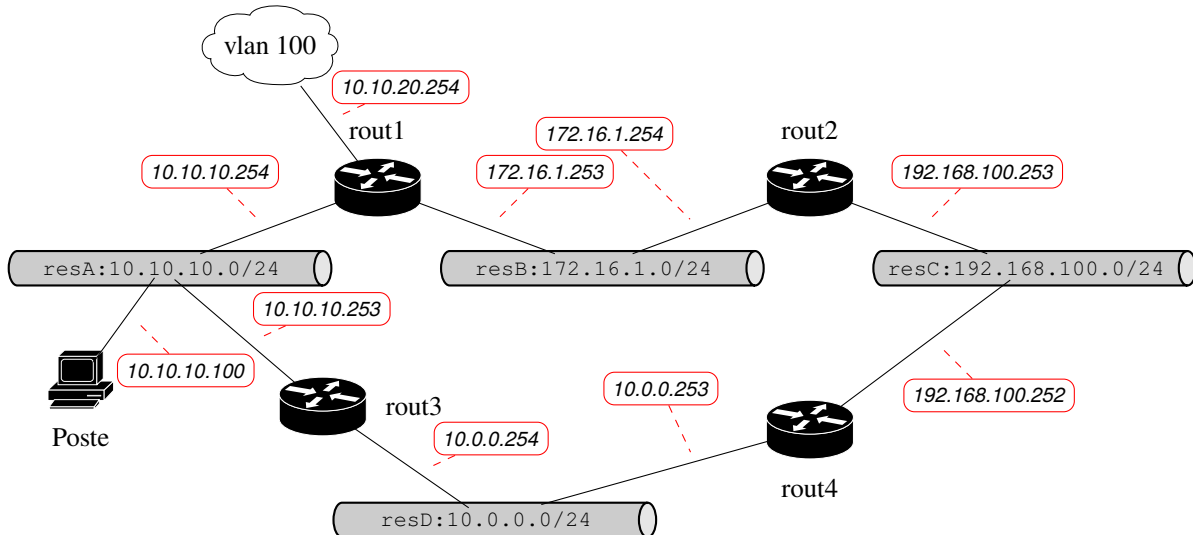


Routage dynamique avec RIP & OSPF

■ ■ ■ Routage à l'aide de RIP & OSPF — Extension du réseau d'interconnexion de la fiche de TP n°1



But de la simulation

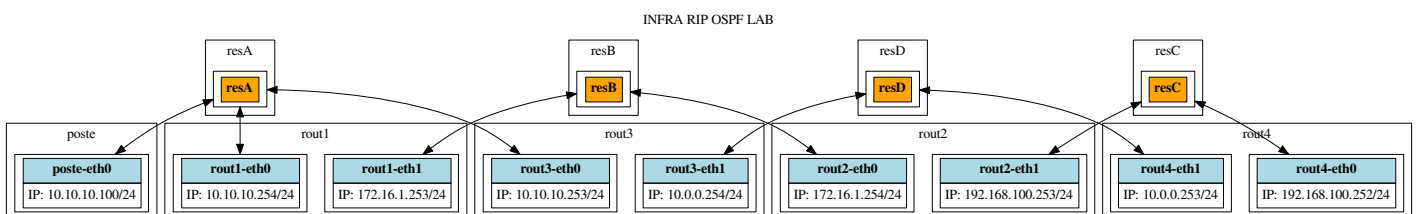
- ▷ étendre le réseau proposé dans la fiche de TP n°1 ;
- ▷ configurer et déployer le protocole RIP dans cette nouvelle configuration ;
- ▷ étudier les paquets échangés du protocole RIP dans cette nouvelle configuration ;
- ▷ générer une panne due à la perte d'une liaison entre routeurs ;
- ▷ analyser les paquets échangés et le fonctionnement du protocole suite à la détection de la panne ;

Extension de la simulation :

- * on ajoutera :
 - ◇ un netns : « rout3 » sur lequel tournera le service « quagga » ;
 - ◇ un switch que l'on nommera « resD » ;
- * vous récupérerez la nouvelle configuration :

```
xterm
$ git clone https://git.p-fb.net/pef/infrarés_tp2_lab.git
```

- * vous configurerez le répertoire « /etc/netns/quagga3 » comme décrit dans la fiche de TP n°1 ;



Travail

- vous finaliserez la configuration de la simulation étendue :
 - vous réaliserez la configuration de RIP sur « rout3 » et « rout4 » ;
 - vous vérifierez que les tables de routage de « rout3 » et « rout4 » sont bien configurées ;
- Par quel chemin est accessible le réseau 10.0.0.0/24 depuis l'hôte ? Est-ce normal ?
- Questions sur le fonctionnement de RIP :
 - quel est la table de routage de « rout3 » ?
 - vous snifferez les paquets RIP reçu sur « rout1 » en provenance de « rout3 » :

```
rezo@ishtar:~/$ sudo ip netns exec rout1 bash
rezo@ishtar:~/# tcpdump -lnvv -i rout1-eth0 '(udp and port 520) and host 10.10.10.253'
```

- Existents-ils des différences entre la table de « rout3 » et la table qu'il diffuse vers « rout1 » ? Pourquoi ?
- Vous installerez la commande traceroute :

```
rezo@ishtar:~/$ sudo apt-get install traceroute
```

Vous exécuterez depuis rout2, la commande :

```
rezo@ishtar:~/$ sudo ip netns exec rout2 bash
rezo@ishtar:~/$ traceroute 10.0.0.254
```

Le résultat est-il correct ?

- Identifiez le nom des différentes interfaces et leur connexion pour « rout1 », « rout3 » et « rout4 » ;
- Installez les éléments de surveillance suivants :
 - sur rout2, vous surveillerez sa configuration de routage :

```
rezo@ishtar:~/$ sudo ip netns exec rout2 bash
rezo@ishtar:~/# watch ip route
```

- sur « rout1 », vous continuerez votre surveillance avec tcpdump ;
- sur « rout3 », où vous substituerez le nom de l'interface identifiée au paramètre INTERFACE :

```
rezo@ishtar:~/$ sudo ip netns exec rout3 bash
rezo@ishtar:~/# tcpdump -lnvX -i INTERFACE '(udp and port 520) and host 10.0.0.253'
```

- vous créerez un « incident » de routage sur « rout4 » :

```
rezo@ishtar:~/$ sudo ip netns exec rout4 bash
rezo@ishtar:~/# ip link set dev rout4-eth1 down
```

Vous exécuterez depuis rout2, la commande :

```
rezo@ishtar:~/$ traceroute 10.0.0.254
```

Le résultat est-il correct ?

- Que pouvez vous observer sur la capture par tcpdump sur « rout3 » lorsque l'interface sur « rout4 » tombe ?
 - Et sur « rout1 » ?
 - Combien de temps faut-il pour que rout2 se rende compte du problème ?
- En réactivant l'interface sur « rout4 » :

```
rezo@ishtar:~/# ip l set dev rout4-eth1 up
```

Est-ce que l'hôte mets beaucoup de temps à se mettre à jour ? Pourquoi ?

- Vous lancerez un « ping » depuis rout2 :

```
rezo@ishtar:~/$ ping 10.0.0.254
```

Vous ferez de nouveau tomber l'interface sur « rout4 ».

Combien de paquets icmp sont perdus avant que la route ne soit rétablie ?

■ ■ ■ Configuration et étude d'OSPF

Vous arrêterez la simulation à l'aide du script « clean » :

```
rez@ishtar:~/$ sudo bash clean
```

Vous configurerez pour chaque netns le fichier associé « /etc/quagga/daemons » avec le contenu suivant :

```
1 zebra=yes
2 bgpd=no
3 ospfd=yes
4 ripd=no
```

Vous copierez pour chaque netns le fichier « ospfd.conf.sample » dans le répertoire /etc/quagga associé au netns.

Il peut être obtenu suivant la version du Linux de votre VM ou de votre machine dans le répertoire « /usr/share/doc/quagga-core/examples/ » ou « /usr/share/doc/quagga/examples/ » :

```
rez@ishtar:~$ sudo cp /usr/share/doc/quagga-core/examples/ospfd.conf.sample
/etc/netns/rout1/quagga/ospfd.conf
```

ou

```
rez@ishtar:~$ sudo cp /usr/share/doc/quagga-core/examples/ospfd.conf.sample
/etc/netns/rout1/quagga/ospfd.conf
```

Vous n'oublierez pas de donner les droits à quagga : quagga sur ce fichier.

Vous ajouterez le lancement du démon « ospfd » dans le fichier « build_architecture » pour chaque netns correspondant à un routeur. Par exemple pour rout3 :

```
ip netns exec rout3 bash -c "mount --bind /tmp/quagga3 /run/quagga; chown -R quag
ga:quagga /run/quagga; zebra -d; ospfd -d"
```

Vous vérifierez que sur chaque netns ospfd fonctionne en attente sur le port 2604 :

```
$ netstat -ln
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat
tcp 0 0 127.0.0.1:2601 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:2604 0.0.0.0:* LISTEN
```

Ensuite, vous pourrez vous connecter pour configurer OSPF :

```
$ telnet localhost 2604
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.20.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
ospfd>
```

La procédure de configuration pour OSPF est la suivante :

- a. l'ensemble des routeurs va faire partie de l'« area 0 » ou le « backbone » ;
- b. chaque routeur va être identifié par un ID, exprimé sous la forme d'une adresse IP (si on laisse le routeur s'auto-configurer, il prend comme ID l'adresse de son interface interprétable numériquement sur 32 bits comme étant la plus grande) :

- ◇ rout1 1.1.1.1;
- ◇ rout2 2.2.2.2;
- ◇ rout3 3.3.3.3;
- ◇ rout4 4.4.4.4;

Pour configurer l'ID :

```
ospfd> enable
ospfd# configure terminal
ospfd(config)# router ospf
ospfd(config-router)# router-id 1.1.1.1
ospfd(config-router)#
```

- c. Pour configurer les réseaux à prendre en charge, par exemple sur rout2 :

```
ospfd(config-router)# network 172.16.1.0/24 area 0
ospfd(config-router)# network 192.168.100.0/24 area 0
ospfd(config-router)# default-information originate
ospfd(config-router)# write file
Configuration saved to /etc/quagga/ospfd.conf
```

Étude du fonctionnement d'OSPF :

- ★ Pour vérifier le fonctionnement d'OSPF, par exemple sur rout1 :

```
ospfd# show ip ospf
OSPF Routing Process, Router ID: 1.1.1.1
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 200 millisec(s)
Minimum hold time between consecutive SPF's 1000 millisec(s)
Maximum hold time between consecutive SPF's 10000 millisec(s)
Hold time multiplier is currently 1
SPF algorithm last executed 1m39s ago
SPF timer is inactive
Refresh timer 10 secs
This router is an ASBR (injecting external routing information)
Number of external LSA 1. Checksum Sum 0x000001bd
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 1

Area ID: 0.0.0.0 (Backbone)
  Number of interfaces in this area: Total: 2, Active: 2
  Number of fully adjacent neighbors in this area: 3
  Area has no authentication
  SPF algorithm executed 13 times
  Number of LSA 9
    Number of router LSA 5. Checksum Sum 0x0002c236
    Number of network LSA 4. Checksum Sum 0x0002c374
    Number of summary LSA 0. Checksum Sum 0x00000000
    Number of ASBR summary LSA 0. Checksum Sum 0x00000000
    Number of NSSA LSA 0. Checksum Sum 0x00000000
    Number of opaque link LSA 0. Checksum Sum 0x00000000
    Number of opaque area LSA 0. Checksum Sum 0x00000000
```

★ Pour voir la gestion des interfaces, par exemple sur « rout3 » :

```
ospfd# show ip ospf interface
rout3-eth0 is up
  ifindex 16, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.10.10.253/24, Broadcast 10.10.10.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 3.3.3.3, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 1.1.1.1, Interface Address 10.10.10.254
  Backup Designated Router (ID) 3.3.3.3, Interface Address 10.10.10.253
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 2.488s
  Neighbor Count is 1, Adjacent neighbor count is 1
rout3-eth1 is up
  ifindex 18, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.0.0.254/24, Broadcast 10.0.0.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 3.3.3.3, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 4.4.4.4, Interface Address 10.0.0.253
  Backup Designated Router (ID) 3.3.3.3, Interface Address 10.0.0.254
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 3.894s
  Neighbor Count is 1, Adjacent neighbor count is 1
lo is up
  ifindex 20, MTU 16436 bytes, BW 0 Kbit <UP,LOOPBACK,RUNNING>
  OSPF not enabled on this interface
```

★ Pour voir les relations d'adjacence :

```
ospfd# show ip ospf neighbor
```

	Neighbor ID	Pri	State	Dead Time	Address	Interface	RXmtL	RqstL	DBsmL
1.1.1.1		1	Full/DR	37.622s	10.10.10.254	eth0:10.10.10.253	0	0	0
4.4.4.4		1	Full/DR	37.097s	10.0.0.253	eth1:10.0.0.254	0	0	0

★ Pour voir les routes :

```
ospfd# show ip ospf route
===== OSPF network routing table =====
N    10.0.0.0/24          [10] area: 0.0.0.0
      directly attached to eth0
N    10.10.10.0/24       [20] area: 0.0.0.0
      via 10.0.0.254, eth0
N    10.10.20.0/24       [30] area: 0.0.0.0
      via 10.0.0.254, eth0
      via 192.168.100.253, eth1
N    172.16.1.0/24       [20] area: 0.0.0.0
      via 192.168.100.253, eth1
N    192.168.100.0/24    [10] area: 0.0.0.0
      directly attached to eth1

===== OSPF external routing table =====
N E2 0.0.0.0/0          [10/10] tag: 0
      via 192.168.100.253, eth1
```

- ★ Pour voir la base de donnée d'un routeur :

```
ospfd# show ip ospf database

OSPF Router with ID (4.4.4.4)

Router Link States (Area 0.0.0.0)

Link ID      ADV Router    Age Seq#          CkSum Link count
1.1.1.1      1.1.1.1       851 0x800000008 0xb27b 3
2.2.2.2      2.2.2.2       803 0x800000004 0x1588 1
3.3.3.3      3.3.3.3       793 0x800000007 0x9434 2
4.4.4.4      4.4.4.4       790 0x800000007 0x7be7 2

Net Link States (Area 0.0.0.0)

Link ID      ADV Router    Age Seq#          CkSum
10.0.0.253   4.4.4.4       790 0x800000001 0xb96c
10.10.10.254 1.1.1.1       851 0x800000001 0x9c8c
172.16.1.253 1.1.1.1       888 0x800000001 0x8b01
192.168.100.254 2.2.2.2     803 0x800000002 0xe27b

AS External Link States

Link ID      ADV Router    Age Seq#          CkSum Route
0.0.0.0      1.2.3.4       594 0x800000001 0x01bd E2 0.0.0.0/0 [0x0]
```

Travail

1. Quel est le coût d'une liaison par défaut ?
2. Vous snifferez les paquets échangés par OSPF à l'aide de la commande suivante (OSPF est associé au protocole 89 dans le datagramme IP) :

```
rezo@ishtar:~/$ sudo ip netns exec rout3 bash
rezo@ishtar:~/# tcpdump -lnvv -i rout3-eth0 'ip[9] == 89'
```

3. Sur rout2 vous essaieriez la commande `traceroute` :

```
root@ishtar:~# traceroute 10.0.0.254
```

Quel est le chemin emprunté ?

Consultez la table de routage, est-ce conforme ?

4. Vous modifierez le coût associé au lien de « rout4 » sur son interface `rout4-eth1` :

```
ospfd> enable
ospfd# configure terminal
ospfd(config)# interface eth1
ospfd(config-if)# ospf cost 30
ospfd(config-if)# exit
ospfd(config)# exit
```

Vous ferez de même pour l'interface « resC » sur rout2.

Pourquoi doit-on faire la modification sur « rout4 » et rout2 ?

Est-ce que le `traceroute` donne le même résultat ?

La table de routage a-t-elle été modifiée ?

Est-ce conforme à la théorie ?

5. Vous rétablirez les coûts sur « rout4 » et sur rout2.

En recommençant une surveillance d'affichage de la table de routage de rout2 :

```
$ watch ip route
```

Vous ferez tomber l'interface `rout4-eth1` sur « rout4 ».

```
rezo@ishtar:~# ip link set dev rout4-eth1 down
```

Est-ce que la modification est rapide ?

Vous réactiverez l'interface et vous lancerez un ping depuis rout2 :

```
root@ishtar:~# ping 10.0.0.254
```

Puis de nouveau de désactiver l'interface : combien de paquets `icmp` sont perdus ?