

# Introduction au routage dynamique avec OSPF

Philippe Latu

philippe.latu(at)linux-france.org

<http://www.linux-france.org/prj/inetdoc/>

Historique des versions		
\$Revision: 1360 \$	\$Date: 2008-12-08 12:56:31 +0100 (lun 08 déc 2008) \$	\$Author: latu \$
Année universitaire 2008-2009		
Résumé		
L'objectif de ce support de travaux pratiques est de mettre en évidence les caractéristiques de fonctionnement du protocole de routage OSPF. Cette illustration s'appuie sur des liens de type Ethernet et sur l'utilisation des VLANs. Les questions de ce support sont présentées comme une introduction pas à pas au protocole de routage OSPF. On débute avec la mise en place d'une topologie réseau type basée sur le routage inter-VLAN, puis on implante les instances de démons de routage.		

## Table des matières

1. Copyright et Licence .....	2
1.1. Méta-information .....	2
1.2. Conventions typographiques .....	2
2. Architecture réseau étudiée .....	3
2.1. Topologie type .....	3
2.2. Plan d'adressage .....	4
3. Préparation des routeurs .....	5
4. Communications entre routeurs .....	6
5. Configuration OSPF de base .....	8
6. Publication d'une route par défaut via OSPF .....	13
7. Ajout de routes fictives .....	15
8. Adaptation de la métrique de lien au débit .....	19
9. Manipulations sur machines virtuelles .....	22
9.1. Préparation des routeurs .....	22
9.2. Configuration du commutateur virtuel vde .....	23
9.3. Utilisation de l'interface virtio .....	25
10. Documents de référence .....	26

# 1. Copyright et Licence

Copyright (c) 2000,2008 Philippe Latu.  
 Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2008 Philippe Latu.  
 Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.2 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

## 1.1. Méta-information

Cet article est écrit avec *DocBook*<sup>1</sup> XML sur un système *Debian GNU/Linux*<sup>2</sup>. Il est disponible en version imprimable aux formats PDF et Postscript : [interco.ospf.pdf](#)<sup>3</sup> | [interco.ospf.ps.gz](#)<sup>4</sup>.

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. C'est la distribution *Debian GNU/Linux* qui est utilisée pour les tests présentés. Voici une liste des paquets contenant les commandes :

- procps - The /proc file system utilities
- net-tools - The NET-3 networking toolkit
- ifupdown - High level tools to configure network interfaces
- iputils-ping - Tools to test the reachability of network hosts
- quagga - BGP/OSPF/RIP routing daemon
- vlan - user mode programs to enable VLANs on your ethernet devices

## 1.2. Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou *prompt* spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite \$ ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.
- Toute commande précédée de l'invite # nécessite les privilèges du super-utilisateur.

<sup>1</sup> <http://www.docbook.org>

<sup>2</sup> <http://www.debian.org>

<sup>3</sup> <http://www.linux-france.org/prj/inetdoc/telechargement/interco.ospf.pdf>

<sup>4</sup> <http://www.linux-france.org/prj/inetdoc/telechargement/interco.ospf.ps.gz>

## 2. Architecture réseau étudiée

### 2.1. Topologie type

La topologie réseau étudiée peut être présentée sous deux formes distinctes : logique et physique.

Topologie logique

On retrouve un grand classique dans l'introduction aux protocoles de routage dynamiques : le triangle. Tous les liens sont de type LAN.

Topologie physique

On s'appuie sur le support *Introduction au routage inter-VLAN* pour constituer une topologie physique à base de réseaux locaux virtuels ou VLANs. On fait correspondre à chaque lien de la topologie logique en triangle un numéro de VLAN défini.

**Tableau 1. Topologie type étudiée**

Topologie logique	Topologie physique

Après avoir mis en œuvre la topologie physique en s'appuyant sur le support de la séance de travaux pratiques précédente : *Introduction au routage inter-VLAN*, on implante les démons de routage OSPF sur les trois routeurs R1, R2 et R3.

Cette séance se limite à l'étude du routage dynamique à l'intérieur d'une aire unique. La seule «frontière» de communication inter-aïres visible est constituée par le lien vers l'Internet. Cette route par défaut sera redistribuée via OSPF par le routeur R1 aux autres routeurs. On verra alors un exemple de route externe dans les bases de données OSPF.

On profite aussi de cette introduction pour employer une technique très répandue pour ajouter «artificiellement» des entrées de tables de routage en s'appuyant sur des interfaces virtuelles de type *dummy* équivalentes à des interfaces de boucle locale.

Pour les besoins de rédaction des questions et réponses de ce support, la topologie a été mise en œuvre sur machines virtuelles KVM avec le commutateur *Virtual Distributed Ethernet* fourni avec le paquet *vde2*. Les éléments de réponse

aux questions dépendent donc de cette mise en œuvre. Pour la séance de travaux pratiques «réelle», il convient donc de se conformer strictement au plan d'adressage fourni ci-après.

## 2.2. Plan d'adressage

Comme dans le support sur l'introduction au routage inter-VLAN, le seul point de configuration imposé est le raccordement au réseau d'interconnexion avec le routeur principal de la salle de travaux pratiques. Ce raccordement utilise le port `fa0/24` de chaque commutateur qui doit être configuré en mode *trunk* en utilisant le VLAN natif numéro 3. Le réseau IP correspondant au VLAN numéro 3 a l'adresse : `172.16.0.0/20`

Point important, la lecture de la section «Plan d'adressage» du document *Architecture réseau des travaux pratiques*<sup>5</sup> donne les adresses des deux routeurs ayant accès au réseau de Campus.

- Routeur `casper.infra.stri` : `172.16.0.2/20`
- Routeur `cooper.infra.stri` : `172.16.0.4/20`

**Tableau 2. Affectation des rôles, des numéros de VLANs et des adresses IP**

Groupe	Commutateur	Poste	Rôle	VLAN	Interface	Réseau
1	sw2.infra.stri	alderaan	R1	3	eth0	172.16.1.1/20
				312	eth0.312	10.1.12.1/26
				313	eth0.313	10.1.13.1/26
		bespin	R2	312	eth0.312	10.1.12.2/26
				323	eth0.323	10.1.23.2/26
				centares	R3	313
323	eth0.323	10.1.23.3/26				
2	sw3.infra.stri	coruscant	R1	3	eth0	172.16.2.1/20
				332	eth0.332	10.2.12.1/26
				333	eth0.333	10.2.13.1/26
		dagobah	R2	332	eth0.332	10.2.12.2/26
				343	eth0.343	10.2.23.2/26
		endor	R3	333	eth0.333	10.2.13.3/26
343	eth0.343			10.2.23.3/26		
3	sw4.infra.stri	felucia	R1	3	eth0	172.16.3.1/20
				352	eth0.352	10.3.12.1/26
				353	eth0.353	10.3.13.1/26
		geonosis	R2	352	eth0.352	10.3.12.2/26
				363	eth0.363	10.3.23.2/26
		hoth	R3	353	eth0.353	10.3.13.3/26
363	eth0.363			10.3.23.3/26		
4	sw5.infra.stri	kamino	R1	3	eth0	172.16.4.1/20

<sup>5</sup> <http://www.linux-france.org/prj/inetdoc/cours/archi.tp/>

Groupe	Commutateur	Poste	Rôle	VLAN	Interface	Réseau
				372	eth0.372	10.4.12.1/26
				373	eth0.373	10.4.13.1/26
		mustafar	R2	372	eth0.372	10.4.12.2/26
				383	eth0.383	10.4.23.2/26
		naboo	R3	373	eth0.373	10.4.13.3/26
				383	eth0.383	10.4.23.3/26

### 3. Préparation des routeurs

La première étape consiste à mettre en place la topologie physique.

- Vérifier l'installation des paquets quagga et vlan avant de rebrancher les postes sur les commutateurs non programmés.

```
$ dpkg -l quagga vlan | grep ^ii
ii quagga 0.99.10-1 BGP/OSPF/RIP routing daemon
ii vlan 1.9-3 user mode programs to enable VLANs on your ethernet devices
```

- Vérifier que la fonction de routage des paquets IPv4 est active au niveau noyau.

```
$ cat /proc/sys/net/ipv4/ip_forward
1
```

Si ce n'est pas le cas, il est possible d'éditer le fichier `/etc/sysctl.conf` pour fixer les valeurs des paramètres de configuration des protocoles de la pile TCP/IP dans le noyau Linux. Voir la section *Fonctions réseau d'une interface du support* [Configuration d'une interface réseau](#).

```
# sysctl -p
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_forward = 1
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.all.proxy_arp = 0
```

- Créer les VLANs sur chacun des routeurs R1, R2 et R3.

```
r1:~# vconfig add eth0 12
r1:~# vconfig add eth0 13

<snip/>
r2:~# vconfig add eth0 12
r2:~# vconfig add eth0 23

<snip/>
r3:~# vconfig add eth0 13
r3:~# vconfig add eth0 23
```

- Activer les démons zebra et ospfd sur chaque routeur en éditant le fichier `/etc/quagga/daemons` et en remplaçant `no` par `yes`.

```
r1:~# cat /etc/quagga/daemons | grep -v ^#
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
```

```
ripngd=no
isisd=no
```

- Créer les fichiers de configuration de base pour les deux démons `zebra` et `ospfd` sur chaque routeur en utilisant les patrons livrés avec le paquet `quagga`.

```
r1:/etc/quagga# cp /usr/share/doc/quagga/examples/zebra.conf.sample zebra.conf
r1:/etc/quagga# cp /usr/share/doc/quagga/examples/ospfd.conf.sample ospfd.conf
```

- Éditer le patron du fichier de configuration du démon `zebra` en fixant les paramètres de connexion à utiliser pour y accéder.

```
# cat zebra.conf
! *- zebra *-
!
hostname R1-zebra
password zebra
enable password zebra
!
log file /var/log/quagga/zebra.log
```

- Éditer le patron du fichier de configuration du démon `ospfd` en fixant les paramètres de connexion à utiliser pour y accéder.

```
# cat ospfd.conf
! *- ospf *-
!
hostname R1-ospfd
password zebra
enable password zebra
!
log file /var/log/quagga/ospfd.log
```

- Compléter la configuration des interfaces dans le démon `zebra` de façon à fixer la bande passante de chaque interface active.

Contrairement à un routeur «intégré» avec un système d'exploitation dédié, le démon de routage statique n'a pas directement accès aux interfaces matérielles. Or, sur un système GNU/Linux, le débit d'une interface nommée `eth0` peut aller de 10Mbps à 10Gbps. Sans information spécifique du noyau, l'application «service de routage» n'a aucun moyen de connaître le débit exact de l'interface `eth0`. C'est la raison pour laquelle il est nécessaire de paramétrer manuellement les débits de chaque interface dans la configuration du démon `zebra`.

```
r1:/etc/quagga# cat zebra.conf | grep -1 bandwidth
interface eth0
  bandwidth 100000
  ipv6 nd suppress-ra
--
interface eth0.12
  bandwidth 100000
  ipv6 nd suppress-ra
--
interface eth0.13
  bandwidth 100000
  ipv6 nd suppress-ra
```



### Avertissement

Ce dernier paramétrage est essentiel dans le calcul des métriques et le fonctionnement du protocole de routage OSPF. Si les calculs de métriques pour les liens actifs sont erronés, le choix des routes à emprunter pour faire transiter le trafic utilisateur entre deux routeurs peut lui aussi être erroné.

## 4. Communications entre routeurs

Avant d'aborder le déploiement du protocole de routage dynamique, il est nécessaire de valider les communications IP entre chaque routeur et de visualiser les tables de routage déjà connues.

1. Quelles sont les opérations à effectuer pour implanter les adresses IP des interfaces correspondant à chacun des VLANs routés ?

## Routeur R1

```
r1:~# vconfig add eth0 12
r1:~# vconfig add eth0 13
r1:~# ifconfig eth0.12 10.1.12.1/26
r1:~# ifconfig eth0.13 10.1.13.1/26
```

## Routeur R2

```
r2:~# vconfig add eth0 23
r2:~# vconfig add eth0 12
r2:~# ifconfig eth0.23 10.1.23.2/26
r2:~# ifconfig eth0.12 10.1.12.2/26
```

## Routeur R3

```
r3:~# vconfig add eth0 23
r3:~# vconfig add eth0 13
r3:~# ifconfig eth0.23 10.1.23.3/26
r3:~# ifconfig eth0.13 10.1.13.3/26
```

2. Quelles sont les opérations à effectuer pour valider les communications IP entre routeurs ?

## Exemple entre R1 et R2

```
r1:~$ ping -c 2 10.1.12.2
PING 10.1.12.2 (10.1.12.2) 56(84) bytes of data.
64 bytes from 10.1.12.2: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=64 time=0.420 ms

--- 10.1.12.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.420/0.768/1.117/0.349 ms
```

L'opération est à répéter sur chaque lien entre deux routeurs «brassés» sur le même VLAN.

3. Quelles sont les opérations à effectuer pour visualiser la table de routage existante d'un routeur au niveau système et au niveau du démon de routage statique zebra ?

## Routeur R1 - niveau système

```
r1:~$ /sbin/route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
192.200.0.0      0.0.0.0         255.255.255.224 U        0       0         0 eth0
10.1.12.0        0.0.0.0         255.255.255.192 U        0       0         0 eth0.12
10.1.13.0        0.0.0.0         255.255.255.192 U        0       0         0 eth0.13
0.0.0.0          192.200.0.1    0.0.0.0         UG       0       0         0 eth0
```

Toutes les métriques des routes affichées valent 0. Elles correspondent à des réseaux IP sur lesquels le routeur est directement connecté via une interface.

## Routeur R1 - démon zebra

```
R1-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.200.0.1, eth0
C>* 10.1.12.0/26 is directly connected, eth0.12
C>* 10.1.13.0/26 is directly connected, eth0.13
C>* 127.0.0.0/8 is directly connected, lo
```

```
C>* 192.200.0.0/27 is directly connected, eth0
```

On retrouve les mêmes informations qu'au niveau système. Une distinction apparaît entre la route par défaut héritée du niveau système qui est repérée avec l'indicateur **K** et les autres routes qui correspondent aux réseaux IP sur lesquels le routeur est directement connecté.

#### Routeur R2 - niveau système

```
r2:~$ /sbin/route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
10.1.12.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.12
10.1.23.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.23
```

#### Routeur R2 - démon zebra

```
R2-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.1.12.0/26 is directly connected, eth0.12
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo
```

#### Routeur R3 - niveau système

```
r3:~$ /sbin/route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
10.1.13.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.13
10.1.23.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.23
```

#### Routeur R3 - démon zebra

```
R3-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.1.13.0/26 is directly connected, eth0.13
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo
```

#### 4. Quelle est l'opération à effectuer pour activer la fonction routage du noyau Linux ?

Si cette fonction n'est pas active dans le noyau Linux, aucune décision d'acheminement d'un paquet d'une interface vers l'autre ne sera prise. Les paquets à router sont simplement jetés.

La commande suivante est à appliquer sur chacun des trois routeurs pour que le protocole de routage dynamique puisse fonctionner normalement.

##### Exemple

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

## 5. Configuration OSPF de base

Dans cette section, on introduit les premières commandes de configuration du protocole de routage dynamique OSPF qui permettent d'activer le protocole puis d'introduire des entrées de réseau dans la base de données de ce protocole.

#### 1. Comment peut-on contrôler si le protocole OSPF est actif ou non sur le routeur ?

Une fois connecté au démon `ospfd`, on lance la commande visualisation globale du protocole.

La commande suivante est utilisable sur chacun des trois routeurs.

```

etu@r1:~$ telnet localhost ospfd
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.10).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
R1-ospfd> en
Password:
R1-ospfd# sh ip ospf
  OSPF Routing Process not enabled

```

2. Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF et fixer manuellement l'identifiant du routeur ?

Toujours à partir de la connexion au démon `ospfd`, on exécute les commandes suivantes sur chacun des trois routeurs en prenant soin d'implanter le bon identifiant.

R1	0.0.0.1
R2	0.0.0.2
R3	0.0.0.3

```

R1-ospfd# conf t
R1-ospfd(config)# router ospf
R1-ospfd(config-router)# router-id 0.0.0.1
R1-ospfd(config-router)#
R1-ospfd# sh ip ospf
  OSPF Routing Process, Router ID: 0.0.0.1
  Supports only single TOS (TOS0) routes
  This implementation conforms to RFC2328
  RFC1583Compatibility flag is disabled
  OpaqueCapability flag is disabled
  Initial SPF scheduling delay 200 millise(c)s
  Minimum hold time between consecutive SPF(s) 1000 millise(c)s
  Maximum hold time between consecutive SPF(s) 10000 millise(c)s
  Hold time multiplier is currently 1
  SPF algorithm has not been run
  SPF timer is inactive
  Refresh timer 10 secs
  Number of external LSA 0. Checksum Sum 0x00000000
  Number of opaque AS LSA 0. Checksum Sum 0x00000000
  Number of areas attached to this router: 0

```

Le choix de codage des identifiants OSPF a pour but d'éviter une confusion avec les adresses des réseaux actifs sur chaque routeur.

Pour toute instance de routage OSPF, le choix de l'identifiant se fait dans l'ordre suivant :

1. Adresse IP définie à l'aide de l'instruction **router-id**.
2. Adresse IP la plus élevée parmi toutes les interfaces de boucle locale
3. Adresse IP la plus élevée parmi toutes les interfaces matérielles

Dans cet exemple, l'instruction **router-id** a été utilisée.

```
R1-ospfd# sh ip ospf database

    OSPF Router with ID (0.0.0.1)
```

3. Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF pour les réseaux connus de chaque routeur ?

Toujours à partir de la connexion au démon `ospfd`, on ajoute une entrée de réseau pour chaque lien connu.

#### Routeur R1

```
R1-ospfd# conf t
R1-ospfd(config)# router ospf
R1-ospfd(config-router)# network 10.1.12.0/26 area 0
R1-ospfd(config-router)# network 10.1.13.0/26 area 0
```

#### Routeur R2

```
R2-ospfd(config)# router ospf
R2-ospfd(config-router)# router-id 0.0.0.2
R2-ospfd(config-router)# network 10.1.12.0/26 area 0
R2-ospfd(config-router)# network 10.1.23.0/26 area 0
```

#### Routeur R3

```
R3-ospfd(config)# router ospf
R3-ospfd(config-router)# router-id 0.0.0.3
R3-ospfd(config-router)# network 10.1.13.0/26 area 0
R3-ospfd(config-router)# network 10.1.23.0/26 area 0
```

4. Quelle est la commande qui permet de visualiser l'état des interfaces actives du routeur vis-à-vis du protocole de routage OSPF ?

C'est la commande **sh ip ospf interface** qui affiche l'état des interfaces réseau. Les interfaces sont automatiquement activées dès qu'une entrée de réseau est saisie au niveau du démon `ospfd` et que l'adresse IP de l'interface correspond à ce réseau.

#### Exemple du routeur R1

```
R1-ospfd# sh ip ospf interface
eth0 is up
  ifindex 2, MTU 1500 bytes, BW 100000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  OSPF not enabled on this interface
eth0.12 is up
  ifindex 3, MTU 1500 bytes, BW 100000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.12.1/26, Broadcast 10.1.12.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.0.1, Interface Address 10.1.12.1
  Backup Designated Router (ID) 0.0.0.2, Interface Address 10.1.12.2
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
  Hello due in 3.665s
  Neighbor Count is 1, Adjacent neighbor count is 1
eth0.13 is up
  ifindex 4, MTU 1500 bytes, BW 100000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.13.1/26, Broadcast 10.1.13.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
```

```

Designated Router (ID) 0.0.0.1, Interface Address 10.1.13.1
Backup Designated Router (ID) 0.0.0.3, Interface Address 10.1.13.3
Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
Hello due in 6.435s
Neighbor Count is 1, Adjacent neighbor count is 1
lo is up
  ifindex 1, MTU 16436 bytes, BW 0 Kbit <UP,LOOPBACK,RUNNING>
  OSPF not enabled on this interface

```

5. À partir des informations affichées ci-dessus, retrouver l'identifiant de routeur et le type de réseau, repérer et identifier la présence d'un autre routeur sur le même réseau.

En reprenant l'exemple du routeur R1 et de son interface eth0.12, on relève les informations suivantes.

```
Router ID 0.0.0.1, Network Type BROADCAST, Cost: 1
```

L'identifiant OSPF du routeur est 0.0.0.1 ; soit le routeur R1. L'interface est connectée sur un réseau Ethernet de type *diffusion* dans lequel R1 est un «routeur désigné» (*Designated Router*) qui sert de référence aux autres routeurs du même périmètre de diffusion ; R2 dans cet exemple.

```
Neighbor Count is 1, Adjacent neighbor count is 1
```

Un routeur OSPF a été reconnu sur ce lien et une adjacence a été établie. Il s'agit du routeur R2 qui apparaît avec l'identifiant 0.0.0.2 sur une ligne au-dessus.

6. Comment peut-on vérifier que l'algorithme SPF du protocole OSPF a été correctement exécuté, que le protocole a convergé et que les entrées de table de routage ont été publiées ?

En reprenant l'exemple du routeur R1, on consulte la liste des voisins OSPF.

```
R1-ospfd# sh ip ospf neighbor
```

Neighbor	ID	Pri	State	Dead Time	Address	Interface	RXmtL	RqstL	DBsmL
0.0.0.2		1	Full/Backup	38.922s	10.1.12.2	eth0.12:10.1.12.1	0	0	0
0.0.0.3		1	Full/Backup	32.826s	10.1.13.3	eth0.13:10.1.13.1	0	0	0

Dans cette liste, on relève la présence des deux autres routeurs :

- R2 avec l'identifiant 0.0.0.2
- R3 avec l'identifiant 0.0.0.3

L'indicateur d'état Full/Backup montre que le dialogue entre les routeurs voisins a convergé vers l'état stable et que le routeur voisin de R1 joue le rôle de «routeur désigné de secours» (*Backup Designated Router*).

Ensuite, on trouve de gauche à droite l'adresse IP de l'interface réseau du routeur voisin et l'interface du routeur local ainsi que son adresse IP.

Toujours sur le routeur R1, la liste des routes publiées par le démon `ospfd` est donnée par la commande **sh ip ospf route**.

```

R1-ospfd# sh ip ospf route
===== OSPF network routing table =====
N   10.1.12.0/26          [1] area: 0.0.0.0
    directly attached to eth0.12
N   10.1.13.0/26          [1] area: 0.0.0.0
    directly attached to eth0.13
N   10.1.23.0/26          [2] area: 0.0.0.0
    via 10.1.12.2, eth0.12
    via 10.1.13.3, eth0.13

===== OSPF router routing table =====

===== OSPF external routing table =====

```

Les valeurs notées entre crochets correspondent à la métrique du lien pour joindre le réseau à gauche. Pour le protocole OSPF, le calcul de métrique se fait à partir de l'expression :  $10^8 / \text{Bande\_Passante\_du\_lien}$ .

Les deux premiers réseaux de la table sont joignable via un lien Ethernet à 100Mbps ; soit une métrique de 1. Le troisième réseau est joignable via deux liens Ethernet à 100Mbps ; d'où la métrique de 2.

## 7. Quel est le mode d'affichage de la table de routage du système qui offre le plus d'informations ?

C'est le démon zebra qui synthétise les entrées de table de routage de l'ensemble du système. Il faut donc se connecter à ce démon pour visualiser la table de routage la plus complète.

```
R1-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

K>* 0.0.0.0/0 via 192.200.0.1, eth0
O 10.1.12.0/26 [110/1] is directly connected, eth0.12, 04:52:04
C>* 10.1.12.0/26 is directly connected, eth0.12
O 10.1.13.0/26 [110/1] is directly connected, eth0.13, 04:54:50
C>* 10.1.13.0/26 is directly connected, eth0.13
O>* 10.1.23.0/26 [110/2] via 10.1.12.2, eth0.12, 04:38:58
   *                   via 10.1.13.3, eth0.13, 04:38:58
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.200.0.0/27 is directly connected, eth0
```

- Les entrées avec le caractère \* correspondent aux routes actives mémorisées dans la base de commutation des paquets IP ou *Forward Information Base* (FIB).
- L'entrée notée K correspond à une route apprise via la configuration système.
- Les entrées notées C correspondent à des routes pour lesquelles il existe une interface sur le routeur. Les métriques de ses routes ont la valeur 0. Ce sont les routes les plus prioritaires.
- Les entrées notées O correspondent aux routes apprises via le protocole OSPF. La métrique de ces routes se décompose en deux parties. La valeur figée à 110 définit le niveau de priorité du protocole OSPF (*Administrative Distance*) relativement aux autres protocoles de routage. Les valeurs notées après le / sont les métriques de liens calculées comme indiqué ci-dessus.

## 8. Comment visualiser la table de routage au niveau système ?

C'est la commande historique **route** ou la commande **ip** qui permettent de visualiser la synthèse du travail du ou des protocoles de routage.

```
$ /sbin/route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref       Use Iface
192.200.0.0      0.0.0.0         255.255.255.224 U        0      0        0 eth0
10.1.12.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.12
10.1.13.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.13
10.1.23.0        10.1.12.2       255.255.255.192 UG       2      0        0 eth0.12
0.0.0.0          192.200.0.1     0.0.0.0          UG       0      0        0 eth0
```

Avec cet affichage, on ne retrouve que les entrées actives de la table affichée via le démon zebra.

```
$ ip route ls
192.200.0.0/27 dev eth0 proto kernel scope link src 192.200.0.2
10.1.12.0/26 dev eth0.12 proto kernel scope link src 10.1.12.1
10.1.13.0/26 dev eth0.13 proto kernel scope link src 10.1.13.1
10.1.23.0/26 proto zebra metric 2
    nexthop via 10.1.12.2 dev eth0.12 weight 1
    nexthop via 10.1.13.3 dev eth0.13 weight 1
default via 192.200.0.1 dev eth0
```

Avec ce dernier affichage, on voit apparaître les «sources» d'alimentation de la table de routage finale du système.

## 6. Publication d'une route par défaut via OSPF

Dans la topologie logique étudiée, le routeur R1 dispose d'un lien montant vers l'Internet. On peut donc considérer que ce lien est la route par défaut vers tous les réseaux non connus de l'aire OSPF contenant les trois routeurs.

Il est possible de publier une route par défaut via le protocole OSPF depuis le routeur R1 vers les routeurs R2 et R3.

Voici, pour mémoire, une copie de la base de données OSPF avant la mise en place de la publication de route par défaut.

```
R1-ospfd# sh ip ospf database

      OSPF Router with ID (0.0.0.1)

          Router Link States (Area 0.0.0.0)

Link ID      ADV Router    Age  Seq#           CkSum  Link count
0.0.0.1      0.0.0.1       470  0x80000015    0x4c7e  2
0.0.0.2      0.0.0.2       466  0x80000016    0x466a  2
0.0.0.3      0.0.0.3       1423 0x8000000e    0x842e  2

          Net Link States (Area 0.0.0.0)

Link ID      ADV Router    Age  Seq#           CkSum
10.1.12.1    0.0.0.1       470  0x8000000b    0xf47e
10.1.13.1    0.0.0.1       1417 0x8000000c    0xf57a
10.1.23.2    0.0.0.2       1428 0x8000000b    0x83e0
```

### 1. Quelle est l'instruction à utiliser pour publier une route par défaut via le protocole de routage OSPF ?

C'est la commande **default-information originate** que l'on doit placer dans la configuration de l'instance OSPF.

```
R1-ospfd# conf t
R1-ospfd(config)# router ospf
R1-ospfd(config-router)# default-information originate
```

Une fois cette instruction exécutée, la base de données OSPF devient :

```
R1-ospfd# sh ip ospf database

      OSPF Router with ID (0.0.0.1)

          Router Link States (Area 0.0.0.0)

Link ID      ADV Router    Age  Seq#           CkSum  Link count
0.0.0.1      0.0.0.1       2    0x80000016    0x5077  2
0.0.0.2      0.0.0.2       488  0x80000016    0x466a  2
0.0.0.3      0.0.0.3       1444 0x8000000e    0x842e  2

          Net Link States (Area 0.0.0.0)

Link ID      ADV Router    Age  Seq#           CkSum
10.1.12.1    0.0.0.1       492  0x8000000b    0xf47e
10.1.13.1    0.0.0.1       1438 0x8000000c    0xf57a
10.1.23.2    0.0.0.2       1449 0x8000000b    0x83e0

          AS External Link States

Link ID      ADV Router    Age  Seq#           CkSum  Route
0.0.0.0      0.0.0.1       4    0x80000001    0x4186  E2 0.0.0.0/0 [0x0]
```

On voit apparaître une nouvelle rubrique baptisée *AS External Link States*. Ce nouveau rôle pour le routeur R1 apparaît aussi lorsque l'on affiche l'état de l'instance de routage OSPF.

```
R1-ospfd# sh ip ospf
```

```

OSPF Routing Process, Router ID: 0.0.0.1
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 200 millisecond(s)
Minimum hold time between consecutive SPF's 1000 millisecond(s)
Maximum hold time between consecutive SPF's 10000 millisecond(s)
Hold time multiplier is currently 3
SPF algorithm last executed 43m21s 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 0x00003f87
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: 2
  Area has no authentication
  SPF algorithm executed 9 times
  Number of LSA 6
  Number of router LSA 3. Checksum Sum 0x0001e267
  Number of network LSA 3. Checksum Sum 0x000176e6
  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

```

Le routeur R1, est maintenant à la frontière entre deux systèmes autonomes : *This router is an ASBR (injecting external routing information)*.

Le principe de la redistribution de route consiste à collecter les routes apprises via un protocole de routage et à les injecter dans un autre domaine de routage. Les routes statiques ou connectées peuvent aussi être redistribuées. Lorsqu'un routeur OSPF utilise des routes apprises via un autre protocole de routage et les rend disponibles pour les autres routeurs OSPF avec lesquels il est en communication, ce routeur devient un *Autonomous System Border Router* ou ASBR.

Ici, R1 utilise OSPF et possède une route statique définie au niveau système vers l'Internet. L'instruction donnée ci-dessus assure la redistribution de la route statique vers R2 et R3. Cette route apparaît comme une entrée de type E2 dans les tables de routage de ces routeurs.

L'indicateur E2 correspond au type par défaut des routes apprises par le biais de la redistribution. La métrique est un point important à considérer avec les routes de type E2. Ces routes ne présentent que le coût du chemin allant du routeur ASBR vers le réseau de destination ; ce qui ne correspond pas au coût réel du chemin.

## 2. Comment la publication de route par défaut apparaît-elle sur les différents routeurs OSPF ?

En prenant l'exemple du routeur R2, on retrouve les informations suivantes :

- Vu du démon ospfd

```

R2-ospfd# sh ip ospf route
===== OSPF network routing table =====
N    10.1.12.0/26          [1] area: 0.0.0.0
      directly attached to eth0.12
N    10.1.13.0/26          [2] area: 0.0.0.0
      via 10.1.12.1, eth0.12
      via 10.1.23.3, eth0.23
N    10.1.23.0/26          [1] area: 0.0.0.0

```

```

                                directly attached to eth0.23

===== OSPF router routing table =====
R   0.0.0.1                      [1] area: 0.0.0.0, ASBR
                                via 10.1.12.1, eth0.12

===== OSPF external routing table =====
N E2 0.0.0.0/0                   [1/10] tag: 0
                                via 10.1.12.1, eth0.12

```

- vu du démon zebra

```

R2-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/10] via 10.1.12.1, eth0.12, 00:19:15
O   10.1.12.0/26 [110/1] is directly connected, eth0.12, 05:27:21
C>* 10.1.12.0/26 is directly connected, eth0.12
O>* 10.1.13.0/26 [110/2] via 10.1.12.1, eth0.12, 05:27:21
   *
   *   via 10.1.23.3, eth0.23, 05:27:21
O   10.1.23.0/26 [110/1] is directly connected, eth0.23, 05:44:12
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo

```

- vu du niveau système

```

$ /sbin/route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
10.1.12.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.12
10.1.13.0        10.1.12.1       255.255.255.192 UG       2      0        0 eth0.12
10.1.23.0        0.0.0.0         255.255.255.192 U        0      0        0 eth0.23
0.0.0.0          10.1.12.1       0.0.0.0          UG      10     0        0 eth0.12

```

La métrique relevée pour la route par défaut vaut 10. Il s'agit donc d'un lien à 10Mbps. Comme indiqué précédemment, cette métrique ne comprend que le coût du lien allant du routeur R1 ASBR vers l'Internet.

## 7. Ajout de routes fictives

L'introduction de nouvelles entrées fictives dans les tables de routage est une pratique très répandue. Elle permet de qualifier le bon fonctionnement du filtrage réseau ou d'un service Internet sans ajouter de matériel. C'est d'ailleurs au service Web que cette section s'intéresse.

1. Quelles sont les opérations à effectuer pour pouvoir utiliser des interfaces réseau virtuelles de type boucle locale sur un système GNU/Linux ?

Avec un noyau Linux, il est conseillé d'utiliser les interfaces baptisées *dummy* pour ce genre d'usage. Les opérations à effectuer consistent à charger le module du même nom en mémoire et à appliquer une nouvelle configuration IP.

```

# modprobe -v dummy numdummies=2
insmod /lib/modules/2.6.26-1-686/kernel/drivers/net/dummy.ko numdummies=2
# ifconfig -a |grep dummy
dummy0      Link encap:Ethernet  HWaddr de:17:c4:8a:04:27
dummy1      Link encap:Ethernet  HWaddr b2:37:fb:84:5d:97

```

En prenant l'exemple du routeur R3, on crée une nouvelle interface avec l'adresse IP : 10.1.3.3/30. Bien sûr, cette adresse ne correspond à aucun réseau déjà connu des trois routeurs.

```

# ifconfig dummy0 10.1.3.3/29
# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
10.1.3.0         0.0.0.0         255.255.255.248 U        0      0        0 dummy0

```

10.1.12.0	10.1.13.1	255.255.255.192	UG	2	0	0	eth0.13
10.1.13.0	0.0.0.0	255.255.255.192	U	0	0	0	eth0.13
10.1.23.0	0.0.0.0	255.255.255.192	U	0	0	0	eth0.23
0.0.0.0	10.1.13.1	0.0.0.0	UG	10	0	0	eth0.13

Cette nouvelle interface se retrouve automatiquement disponible dans la configuration des deux démons de routage. Dans zebra sur R3, on obtient les informations suivantes sur l'interface dummy0.

```
R3-zebra# sh int dummy0
Interface dummy0 is up, line protocol detection is disabled
  index 5 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,NOARP>
  bandwidth 10000 kbps
  inet 10.1.3.3/29 broadcast 10.1.3.7
  inet6 fe80::dc17:c4ff:fe8a:427/64
    0 input packets (0 multicast), 0 bytes, 0 dropped
    0 input errors, 0 length, 0 overrun, 0 CRC, 0 frame
    0 fifo, 0 missed
    3 output packets, 210 bytes, 0 dropped
    0 output errors, 0 aborted, 0 carrier, 0 fifo, 0 heartbeat
    0 window, 0 collisions
```

On a fixé manuellement à 10Mbps dans zebra la bande passante associée à l'interface dummy0. Ce paramètre a une influence sur le calcul de métrique dans les bases de données OSPF.

2. Quelles sont les opérations à effectuer pour installer un service Web en écoute exclusivement sur l'adresse IP de l'interface dummy0 ?

Pour aller au plus court, on installe le paquet apache2.

```
r3:~# aptitude install apache2-mpm-worker
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Lecture de l'information d'état étendu
Initialisation de l'état des paquets... Fait
Lecture des descriptions de tâches... Fait
Les NOUVEAUX paquets suivants vont être installés :
  apache2-mpm-worker apache2-utils{a} apache2.2-common{a} libapr1{a}
  libaprutil1{a} libexpat1{a} libmysqlclient15off{a} libpq5{a} mysql-common{a}
0 paquets mis à jour, 9 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 3748ko d'archives. \
Après dépaquetage, 10,3Mo seront utilisés.
Voulez-vous continuer ? [Y/n/?]
<snipped/>
Paramétrage de apache2-mpm-worker (2.2.9-10) ...
Starting web server: apache2apache2: Could not reliably determine the server's
  fully qualified domain name, using 127.0.1.1 for ServerName
```

On modifie ensuite le fichier de configuration /etc/apache2/ports.conf de façon à limiter l'accès à une adresse IP unique.

```
r3:/etc/apache2# diff -uBb ports.conf.dpkg-dist ports.conf
--- ports.conf.dpkg-dist      2008-11-28 23:37:25.000000000 +0100
+++ ports.conf      2008-11-28 23:37:45.000000000 +0100
@@ -3,7 +3,7 @@
 # /etc/apache2/sites-enabled/000-default

 NameVirtualHost *:80
 -Listen 80
 +Listen 10.1.3.3:80

 <IfModule mod_ssl.c>
```

```
# SSL name based virtual hosts are not yet supported, therefore no
```

Après avoir redémarré l'instance de serveur Web, on vérifie que la nouvelle configuration est bien en place.

```
# /etc/init.d/apache2 restart
<snipped>
# lsof -i |grep apache2
apache2  5885    root      3u  IPv4  10977      TCP 10.1.3.3:www (LISTEN)
apache2  5889  www-data  3u  IPv4  10977      TCP 10.1.3.3:www (LISTEN)
apache2  5894  www-data  3u  IPv4  10977      TCP 10.1.3.3:www (LISTEN)
```

### 3. Comment ajouter la route correspondant au nouveau réseau 10.1.3.0/29 dans le domaine de routage OSPF ?

Comme dans le cas de la mise en place des autres routes dans la configuration du démon `ospfd`, on ajoute une entrée `network` dans l'instance OSPF du routeur R3.

```
R3-ospfd(config)# router ospf
R3-ospfd(config-router)# network 10.1.3.0/29 area 0
```

On vérifie ensuite que cette entrée est bien intégrée dans la base de données OSPF.

```
R3-ospfd(config)# sh ip ospf database
```

### 4. Comment valider l'accès à ce service Web depuis les autres routeurs ?

En respectant l'ordre des protocoles de la pile TCP/IP, on commence par valider la connectivité au niveau réseau avant de passer à la couche transport et enfin au niveau application.

À partir du routeur R1, on utilise la séquence suivante :

- Niveau réseau : **ping**

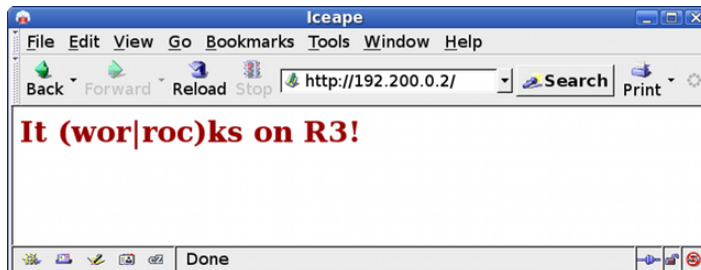
```
r1:~# ping -c 3 10.1.3.3
PING 10.1.3.3 (10.1.3.3) 56(84) bytes of data.
64 bytes from 10.1.3.3: icmp_seq=1 ttl=64 time=3.50 ms
64 bytes from 10.1.3.3: icmp_seq=2 ttl=64 time=0.456 ms
64 bytes from 10.1.3.3: icmp_seq=3 ttl=64 time=0.394 ms

--- 10.1.3.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 0.394/1.451/3.504/1.452 ms
```

- Niveau transport : **tcptracert**

```
r1:~# tcptracert 10.1.3.3
traceroute to 10.1.3.3 (10.1.3.3), 30 hops max, 40 byte packets
 1  10.1.3.3 (10.1.3.3)  0.632 ms  0.563 ms  0.512 ms
```

- Niveau application : navigateur Web



Capture d'écran serveur Web à l'adresse 10.1.3.3 - vue complète<sup>6</sup>

Cette dernière capture d'écran a été réalisée à partir du système hôte sur lequel les instances de machines virtuelles correspondant aux trois routeurs ont été lancées. Les routeurs sur machines virtuelles KVM choisis

<sup>6</sup> <http://www.linux-france.org/fr/melodoc/cours/interco.ospf/images/www-r3.png>

pour mettre la topologie réseau en place n'ont pas d'interface graphique. On accède donc au service Web par le biais de l'interface graphique du système hôte. Voir [Section 9, « Manipulations sur machines virtuelles »](#).

Du point de vue réseau, le système hôte est raccordé au commutateur virtuel VDE sur le port numéro 1 via son interface `tap0` avec l'adresse IP `192.200.0.1/27`.

## 8. Adaptation de la métrique de lien au débit

Par défaut pour le protocole OSPF, le calcul de métrique se fait à partir de l'expression :  $10^8 / \text{Bande\_Passante\_du\_lien}$ .

Cette règle a été établie à une époque où l'utilisation d'un lien à 100Mbps devait être considéré comme une situation d'exploitation futuriste. Aujourd'hui, les liens à 100Mbps sont monnaie courante et les 10Gbps sont à portée de vue.

Cette section traite donc de la configuration des instances de protocole de routage OSPF utilisant des liens avec une capacité supérieure à 100Mbps.

1. Quelle est l'instruction à utiliser pour que le calcul de métrique OSPF se fasse sur la base d'un débit de lien à 1Gbps ?

C'est l'instruction **auto-cost reference-bandwidth** qui permet de fixer une nouvelle référence de coût de lien en Mbps.

```
R2-ospfd(config-router)# auto-cost reference-bandwidth
<1-4294967> The reference bandwidth in terms of Mbits per second
```

Voici un extrait de la configuration du routeur R2 après application d'une nouvelle référence à 1000Mbps soit 1Gbps.

```
router ospf
  ospf router-id 0.0.0.2
  ! Important: ensure reference bandwidth is consistent across all routers
  auto-cost reference-bandwidth 1000
  network 10.1.12.0/26 area 0.0.0.0
  network 10.1.23.0/26 area 0.0.0.0
```

Comme indiqué dans le commentaire ci-dessus, il est indispensable que tous les routeurs de l'aire OSPF aient la même référence de calcul de métrique. La même instruction doit donc être implantée dans la configuration des trois routeurs.

2. Comment modifier le débit d'un lien à 1Gbps ?

Comme indiqué dans la [Section 3, « Préparation des routeurs »](#), c'est dans la configuration du démon zebra que l'on attribue la bande passante d'une interface réseau. Cette opération est nécessaire compte tenu du fait que la bande passante d'une même interface Ethernet eth0 peut varier de 10Mbps à 10Gbps.

Dans toutes les sections précédentes, la bande passante des interfaces était fixée à 100Mbps. Pour tenir compte de la nouvelle référence de calcul de métrique, on applique une nouvelle valeur de bande passante fixée à 1Gbps sur les interfaces Ethernet des trois routeurs. Voici un extrait du fichier de configuration du démon zebra du routeur R2.

```
interface eth0
  bandwidth 1000000
  ipv6 nd suppress-ra
  !
interface eth0.12
  bandwidth 1000000
  ipv6 nd suppress-ra
  !
interface eth0.23
  bandwidth 1000000
  ipv6 nd suppress-ra
```

Comme la valeur de base de bande passante est le kilobit par seconde, il faut 1 million de Kbps pour faire 1Gbps ou  $10^9$ bps.

3. Comment peut-on identifier le débit d'un lien dans la configuration OSPF ?

C'est l'instruction **sh ip ospf interface** qui permet d'afficher la valeur de la bande passante attribuée aux interfaces actives dans le routage OSPF.

Voici le résultat obtenu sur le routeur R2.

```
R2-ospfd# sh ip ospf interface
eth0 is up
  ifindex 2, MTU 1500 bytes, BW 1000000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  OSPF not enabled on this interface
eth0.12 is up
  ifindex 3, MTU 1496 bytes, BW 1000000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.12.2/26, Broadcast 10.1.12.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 0.0.0.2, Interface Address 10.1.12.2
  Backup Designated Router (ID) 0.0.0.1, Interface Address 10.1.12.1
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 8.604s
  Neighbor Count is 1, Adjacent neighbor count is 1
eth0.23 is up
  ifindex 4, MTU 1496 bytes, BW 1000000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.1.23.2/26, Broadcast 10.1.23.63, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 0.0.0.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 0.0.0.3, Interface Address 10.1.23.3
  Backup Designated Router (ID) 0.0.0.2, Interface Address 10.1.23.2
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 1.144s
  Neighbor Count is 1, Adjacent neighbor count is 1
lo is up
  ifindex 1, MTU 16436 bytes, BW 0 Kbit <UP,LOOPBACK,RUNNING>
  OSPF not enabled on this interface
```

4. Quel est le coût d'accès au pseudo service Internet (réseau 10.1.3.0/29) après modification de la référence de calcul de métrique ? Justifier la valeur de métrique obtenue.

Il est possible d'obtenir les informations de calcul de métrique soit à partir du démon de routage OSPF ospfd, soit à partir du démon de routage statique zebra.

- Démon de routage OSPF:

```
R2-ospfd# sh ip ospf route
===== OSPF network routing table =====
N   10.1.3.0/29           [101] area: 0.0.0.0
                                via 10.1.23.3, eth0.23
N   10.1.12.0/26         [1] area: 0.0.0.0
                                directly attached to eth0.12
N   10.1.13.0/26         [2] area: 0.0.0.0
                                via 10.1.12.1, eth0.12
                                via 10.1.23.3, eth0.23
N   10.1.23.0/26         [1] area: 0.0.0.0
                                directly attached to eth0.23

===== OSPF router routing table =====
R   0.0.0.1              [1] area: 0.0.0.0, ASBR
                                via 10.1.12.1, eth0.12

===== OSPF external routing table =====
N E2 0.0.0.0/0          [1/10] tag: 0
```

```
via 10.1.12.1, eth0.12
```

- Démon de routage statique :

```
R2-zebra# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/10] via 10.1.12.1, eth0.12, 07:11:46
O>* 10.1.3.0/29 [110/101] via 10.1.23.3, eth0.23, 07:11:46
O   10.1.12.0/26 [110/1] is directly connected, eth0.12, 07:12:32
C>* 10.1.12.0/26 is directly connected, eth0.12
O>* 10.1.13.0/26 [110/2] via 10.1.12.1, eth0.12, 07:11:46
   *                       via 10.1.23.3, eth0.23, 07:11:46
O   10.1.23.0/26 [110/1] is directly connected, eth0.23, 07:12:27
C>* 10.1.23.0/26 is directly connected, eth0.23
C>* 127.0.0.0/8 is directly connected, lo
```

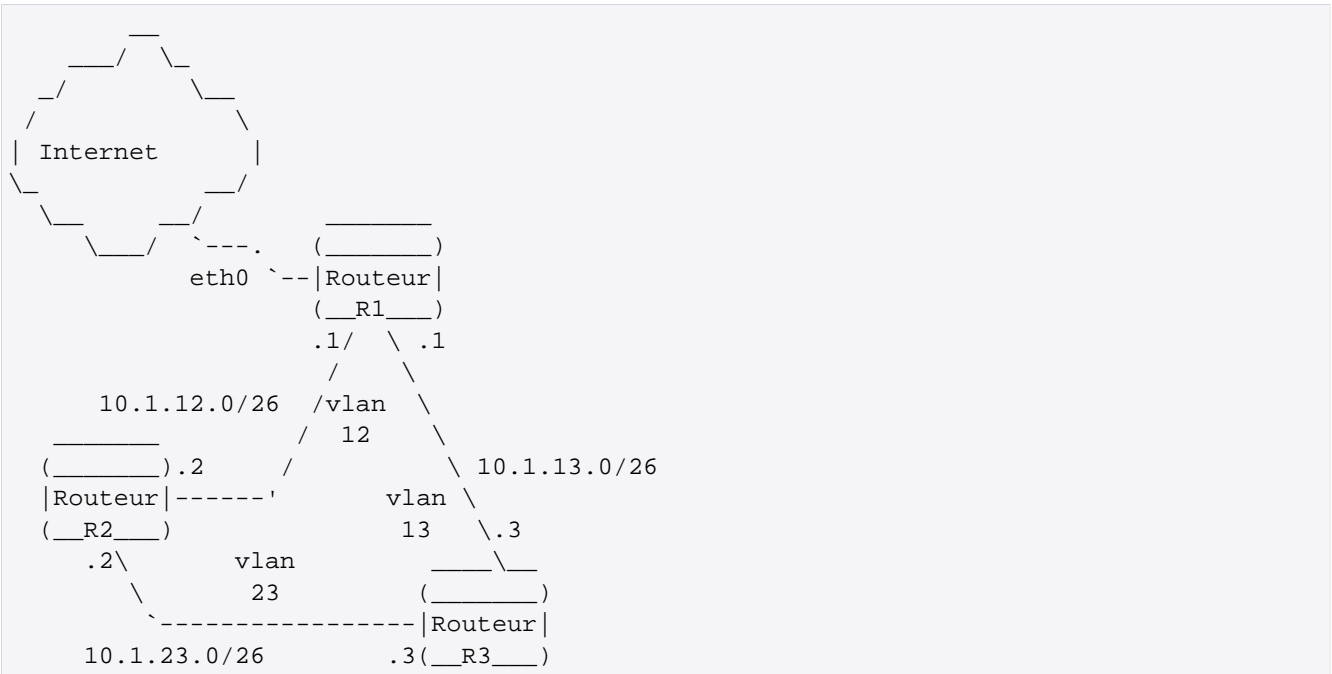
On obtient la valeur 101 vu du routeur R2. Cette métrique correspond à la somme des coûts de deux liens : 1 lien à 1Gbps ( $10^9/10^9 = 1$ ) + 1 lien à 10Mbps ( $10^9/10^7 = 100$ ) = 101.

Le lien à 1Gbps correspond à la liaison entre les routeurs R2 et R3. Le lien à 10Mbps correspond à la liaison entre le routeur R3 et son interface virtuelle dummy0 sur laquelle le service Web est en écoute.

## 9. Manipulations sur machines virtuelles

Il est possible de réaliser l'ensemble des manipulations de ce support à l'aide de trois instances de machines virtuelles et du commutateur virtuel *Virtual Distributed Ethernet* présenté dans l'article *Virtualisation système et enseignements pratiques*<sup>7</sup>.

Voici quelques éléments sur la mise en œuvre de cette «infrastructure de travaux pratiques».



### 9.1. Préparation des routeurs

Partant de la liste des images de machines virtuelles téléchargeables à partir du serveur Web de l'infrastructure de de travaux pratiques STRI (<http://www.stri/vm/>), on copie 3 images disques identiques.

```

$ cd ~/vm
$ mkdir ospf
$ cp vm0-debian-i386.qcow2 ospf/r1.qcow2
$ cp vm0-debian-i386.qcow2 ospf/r2.qcow2
$ cp vm0-debian-i386.qcow2 ospf/r3.qcow2

```

Ensuite, on crée un script *shell* de lancement des instances de «routeurs» dans lequel on fixe les paramètres d'initialisation de ces mêmes «routeurs».

Attention ! Ce script ne doit être lancé qu'après l'initialisation du commutateur virtuel pour que le brassage des routeurs sur les ports du commutateur puisse se faire correctement.

```

$ cd ~/vm/ospf
$ cat startup.sh

# Décommenter une ligne MODEL= pour choisir le type de virtualisation
# Paravirtualisation
MODEL=",model=virtio"
# Virtualisation
#MODEL=""

kvm -name r1 \
-m 512 -hda r1.qcow2 \
-net vde,vlan=1,sock=/tmp/vde.ctl,port=2 \
-net nic,vlan=1$MODEL,macaddr=52:54:00:12:34:01 &

```

<sup>7</sup> <http://www.linux-france.org/prj/inetdoc/articles/vm/>

```
kvm -name r2 \
-m 512 -hda r2.qcow2 \
-net vde,vlan=2,sock=/tmp/vde.ctl,port=3 \
-net nic,vlan=2$MODEL,macaddr=52:54:00:12:34:02 &
kvm -name r3 \
-m 512 -hda r3.qcow2 \
-net vde,vlan=3,sock=/tmp/vde.ctl,port=4 \
-net nic,vlan=3$MODEL,macaddr=52:54:00:12:34:03 &
```

On note que le sixième octet des adresses MAC de chaque instance correspond au numéro du routeur ce qui permet de valider le brassage sur le commutateur virtuel en affichant le contenu de la table CAM.

Remarque importante : l'option `vlan=x` ne correspond absolument pas à l'attribution d'un numéro de VLAN au niveau du commutateur *VDE*. Il s'agit ici de mettre en correspondance l'affectation d'un port du commutateur via son *socket* et les caractéristiques de l'interface réseau utilisée par l'instance de système virtuel (modèle et adresse MAC dans notre cas).

## 9.2. Configuration du commutateur virtuel vde

Comme indiqué dans l'article cité en référence ci-dessus, on lance le commutateur virtuel comme suit :

```
$ vde_switch -d --tap tap0 -s /tmp/vde.ctl -M /tmp/vde.mgmt
```

Dans cette configuration, l'interface `tap0` est automatiquement brassée sur le port 1 du commutateur et elle utilise le VLAN numéro 0 par défaut.

Une fois que l'on a lancé le script de lancement des instances de machines virtuelles à l'aide de la commande : `$ sh ./startup.sh`, on obtient les informations suivantes à partir de l'interface de configuration du commutateur.

- Avant la mise en place des VLANs sur les routeurs et le commutateur.

```
$ unixterm /tmp/vde.mgmt
VDE switch V.2.2.2
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

vde$ hash/print
0000 DATA END WITH '.'
Hash: 0048 Addr: 52:54:00:12:34:03 VLAN 0000 to port: 004 age 39 secs
Hash: 0055 Addr: 52:54:00:12:34:01 VLAN 0000 to port: 002 age 87 secs
Hash: 0088 Addr: 52:54:00:12:34:02 VLAN 0000 to port: 003 age 63 secs
.
1000 Success
```

- Après la mise en place des VLANs sur les routeurs et le commutateur.

```
vde$ hash/print
0000 DATA END WITH '.'
Hash: 0007 Addr: 52:54:00:12:34:01 VLAN 0013 to port: 002 age 9 secs
Hash: 0032 Addr: 52:54:00:12:34:03 VLAN 0013 to port: 004 age 9 secs
Hash: 0064 Addr: 52:54:00:12:34:03 VLAN 0023 to port: 004 age 9 secs
Hash: 0104 Addr: 52:54:00:12:34:02 VLAN 0023 to port: 003 age 9 secs
Hash: 0119 Addr: 52:54:00:12:34:01 VLAN 0012 to port: 002 age 3 secs
.
1000 Success
```

On valide ainsi le brassage des routeurs.

**Tableau 3. Brassage commutateur virtuel**

Port VDE	Routeur	Liaison
1	R1	Internet   Système hôte
2	R1	trunk R2 + R3
3	R2	trunk R1 + R3

Port VDE	Routeur	Liaison
4	R3	trunk R1 + R2

Les mêmes informations se retrouvent via l'interface de configuration du commutateur virtuel.

```
vde$ port/print
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
  IN: pkts      0      bytes      0
  OUT: pkts     52     bytes    13032
  -- endpoint ID 0007 module tuntap      : tap0
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
  IN: pkts     18     bytes    4572
  OUT: pkts     34     bytes    8460
  -- endpoint ID 0003 module unix prog   : vdeqemu user=phil PID=23102 \
                                           SOCK=/tmp/vde.ct1/.23102-00000
Port 0003 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
  IN: pkts     16     bytes    3888
  OUT: pkts     31     bytes    8474
  -- endpoint ID 0009 module unix prog   : vdeqemu user=phil PID=23173 \
                                           SOCK=/tmp/vde.ct1/.23173-00000
Port 0004 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
  IN: pkts     18     bytes    4572
  OUT: pkts     25     bytes    6966
  -- endpoint ID 0011 module unix prog   : vdeqemu user=phil PID=23244 \
                                           SOCK=/tmp/vde.ct1/.23244-00000
.
1000 Success
```

Une fois le brassage en place, on peut passer à la configuration des VLANs ; toujours via l'interface de configuration du commutateur virtuel. On sauvegarde la liste des instructions dans un fichier de configuration du commutateur : `vde.conf` dans notre exemple. On charge ensuite ces instructions dans le commutateur virtuel via la commande `vde $ load vde.conf`.

```
vlan/create 12
vlan/create 13
vlan/create 23
vlan/addport 12 2
vlan/addport 13 2
vlan/addport 12 3
vlan/addport 23 3
vlan/addport 13 4
vlan/addport 23 4
```

Cette syntaxe s'approche plus du mode Hewlett Packard™ que du mode Cisco™. Les trois premières lignes servent à créer les VLANs suivant la dénomination :

VLAN numéro 12	Liaison R1 / R2
VLAN numéro 13	Liaison R1 / R3
VLAN numéro 23	Liaison R2 / R3

Ensuite, les six lignes suivantes servent à désigner les VLANs ou les trames étiquetées à véhiculer vers les ports. Par exemple, le port numéro 2 reçoit les trames avec les balises des VLANs 12 et 13.

Comme dans les cas précédents, on retrouve ces affectations via l'interface de configuration du commutateur virtuel.

```
vde$ vlan/print
0000 DATA END WITH '.'
VLAN 0000
  -- Port 0001 tagged=0 active=1 status=Forwarding
  -- Port 0002 tagged=0 active=1 status=Forwarding
```

```

-- Port 0003 tagged=0 active=1 status=Forwarding
-- Port 0004 tagged=0 active=1 status=Forwarding
VLAN 0012
-- Port 0002 tagged=1 active=1 status=Forwarding
-- Port 0003 tagged=1 active=1 status=Forwarding
VLAN 0013
-- Port 0002 tagged=1 active=1 status=Forwarding
-- Port 0004 tagged=1 active=1 status=Forwarding
VLAN 0023
-- Port 0003 tagged=1 active=1 status=Forwarding
-- Port 0004 tagged=1 active=1 status=Forwarding
.
1000 Success

```

### 9.3. Utilisation de l'interface virtio

Pour accéder à des débits réseau plus élevés sur les instances de machines virtuelles, on doit utiliser l'interface *virtio* du noyau Linux. Il s'agit de mettre en place une *paravirtualisation* dans laquelle le noyau de l'instance de machine virtuelle est «modifié» pour accéder directement aux ressources du système hôte.

Pour utiliser cette interface de *paravirtualisation*, il est nécessaire d'intervenir sur la configuration système des routeurs R1, R2 et R3. Comme les paquets de noyaux fournis par la distribution *Debian GNU/Linu* possèdent toutes les fonctions nécessaires, il n'est pas nécessaire de recompiler quoi que ce soit.

1. Il faut compléter la liste des modules accessibles par le noyau avant le montage de la racine du système d'exploitation. Une fois ces modules disponibles, le système d'exploitation fait appel aux fonctions de paravirtualisation lors de l'initialisation des périphériques.

On ajoute la liste des modules donnée dans la copie d'écran ci-dessous dans le fichier `/etc/initramfs-tools/modules`.

```

# cat /etc/initramfs-tools/modules | grep -v ^#
virtio
virtio_balloon
virtio_pci
virtio_rng
virtio_net
virtio_blk

```

Pour que cette nouvelle liste de modules soit prise en compte, il faut générer un nouveau fichier `initramfs` à l'aide de la commande suivante :

```

r2:~# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-2.6.26-1-686

```

2. Il faut préciser que le modèle d'interface réseau à utiliser est justement de type `virtio` lors du lancement des instances de machines virtuelles. Cette opération se fait via la directive `-net nic,model=virtio`.

Voir [Section 9.1, « Préparation des routeurs »](#).

3. *Point très important !* Le routage inter-VLAN ne peut fonctionner correctement qu'après correction manuelle de la taille maximum de paquet sur les interfaces des routeurs : *Maximum Transmit Unit* ou MTU.

Ce problème historique d'adaptation de la charge utile de paquet IP est «réapparu» avec l'utilisation de la paravirtualisation. Il est nécessaire de décompter les 4 octets IEEE 802.1Q ajoutés à la trame Ethernet standard pour que les communications fonctionnent correctement entre routeurs.

La valeur usuelle du paramètre MTU pour une interface Ethernet est de 1500 octets. On doit donc appliquer une valeur des 1496 octets pour tenir compte des 4 octets utilisés par les informations sur le réseau local virtuel (VLAN) associé à l'interface.

Du point de vue configuration système, le fichier `/etc/network/interfaces` du routeur R2 devient :

```
# cat /etc/network/interfaces | grep -v ^#
auto lo
iface lo inet loopback

auto eth0.12
iface eth0.12 inet static
    address 10.1.12.2
    netmask 255.255.255.192
    mtu 1496

auto eth0.23
iface eth0.23 inet static
    address 10.1.23.2
    netmask 255.255.255.192
    mtu 1496
```

L'affichage de l'état d'une interface fait apparaître cette nouvelle valeur du paramètre MTU :

```
# ifconfig eth0.12
eth0.12  Link encap:Ethernet  HWaddr 52:54:00:12:34:02
         inet adr:10.1.12.2  Bcast:10.1.12.63  Masque:255.255.255.192
         adr inet6: fe80::5054:ff:fe12:3462/64  Scope:Lien
         UP BROADCAST RUNNING MULTICAST  MTU:1496  Metric:1
         RX packets:845564 errors:0 dropped:0 overruns:0 frame:0
         TX packets:734140 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 lg file transmission:0
         RX bytes:957985029 (913.6 MiB)  TX bytes:824013850 (785.8 MiB)
```

Dans un contexte de virtualisation «pure», cette manipulation n'est pas nécessaire dans la mesure où le noyau de l'instance de machine virtuelle utilise le même pilote de périphérique réseau que celui d'un système non virtualisé. Les pilotes de cartes réseau Ethernet du noyau Linux intègrent directement l'utilisation des VLANs depuis longtemps. Il est donc logique que le routage inter-VLAN soit totalement transparent dans un contexte de virtualisation à 100%. Avec l'utilisation de la *paravirtualisation*, le chantier reste ouvert.

Une fois la configuration de la paravirtualisation en place, on peut utiliser l'outil *iperf* pour mesurer la capacité réseau entre les routeurs R1 et R2. Voici un échantillon de résultat obtenu.

```
etu@r1:~$ iperf -c 10.1.12.2
-----
Client connecting to 10.1.12.2, TCP port 5001
TCP window size: 16.0 KByte (default)
-----
[  3] local 10.1.12.1 port 54197 connected with 10.1.12.2 port 5001
[ ID] Interval           Transfer     Bandwidth
[  3]  0.0-10.0 sec      433 MBytes   364 Mbits/sec
```

Ce résultat correspond au débit maximum par VLAN sur le lien. Sachant que deux VLANs sont actif sur le lien «physique», le débit de 364Mbps représente à peu près la moitié du débit total du canal de communication.

## 10. Documents de référence

### *Configuration d'une interface réseau*

Le support *Configuration d'une interface de réseau local*<sup>8</sup> présente les opérations de configuration d'une interface réseau et propose quelques manipulations sur les protocoles de la pile TCP/IP

### *Initiation au routage, 1ère partie*

L'article *Initiation au routage, 1ère partie*<sup>9</sup> introduit l'utilisation de quagga et de son premier démon baptisé zebra. Ce démon permet de mettre en place un *routage statique* associé à la table de routage définie dans la configuration du système.

<sup>8</sup> <http://www.linux-france.org/prj/inetdoc/cours/config.interface.lan/>

<sup>9</sup> <http://www.linux-france.org/prj/inetdoc/guides/zebra.statique/>

*Initiation au routage, 3ème partie*

L'article *Initiation au routage, 3ème partie*<sup>10</sup> introduit l'utilisation du protocole OSPF sur plusieurs aires. Ce n'est pas l'objectif de ce support qui se limite au routage dynamique dans un système autonome unique ; donc une aire unique.

*Introduction au routage inter-VLAN*

Le support *Introduction au routage Inter-VLAN*<sup>11</sup> introduit le principe du routage inter-VLAN ainsi que ses conditions d'utilisation. C'est aussi un support de travaux pratiques dans lequel on n'utilise que du routage statique et de la traduction d'adresses sources (S-NAT) pour acheminer le trafic utilisateur entre les différents réseaux.

*Open Shortest Path First (OSPF)*

La page consacrée au protocole sur le site Cisco™ : *Open Shortest Path First (OSPF)*<sup>12</sup> regroupe des ressources importantes sur la conception d'architecture réseau utilisant ce protocole.

On peut aussi citer les supports de formation *Cisco Networking Academy* qui sont d'une excellente qualité sur l'initiation à l'utilisation des protocoles de routage. Malheureusement, ce ne sont pas des documents libres d'utilisation.

---

<sup>10</sup> <http://www.linux-france.org/prj/inetdoc/guides/zebra.ospf/>

<sup>11</sup> <http://www.linux-france.org/prj/inetdoc/cours/routage.inter-vlan/>

<sup>12</sup> [http://www.cisco.com/en/US/tech/tk365/tk480/tsd\\_technology\\_support\\_sub-protocol\\_home.html](http://www.cisco.com/en/US/tech/tk365/tk480/tsd_technology_support_sub-protocol_home.html)