

Introduction au routage dynamique avec OSPF

Philippe Latu

philippe.latu(at)inetdoc.net

<http://www.inetdoc.net>

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 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	3
3. Préparation des routeurs	5
4. Communications entre routeurs	7
5. Configuration OSPF de base	8
6. Adaptation de la métrique de lien au débit	9
7. Publication d'une route par défaut via OSPF	10
8. Ajout de routes fictives	11
9. Manipulations sur machines virtuelles	12
9.1. Préparation des routeurs	12
9.2. Table de routage du système hôte	13
9.3. Configuration du commutateur virtuel vde	13
9.4. Utilisation de l'interface virtio	15
10. Documents de référence	17

1. Copyright et Licence

Copyright (c) 2000,2012 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 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) 2000,2012 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 1.3 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

Ce document est écrit avec *DocBook*¹ XML sur un système *Debian GNU/Linux*². Il est disponible en version imprimable au format PDF : [interco.ospf.q.pdf](http://www.inetdoc.net/pdf/interco.ospf.q.pdf)³.

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.

¹ <http://www.docbook.org>

² <http://www.debian.org>

³ <http://www.inetdoc.net/pdf/interco.ospf.q.pdf>

2. Architecture réseau étudiée

2.1. Topologie type

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

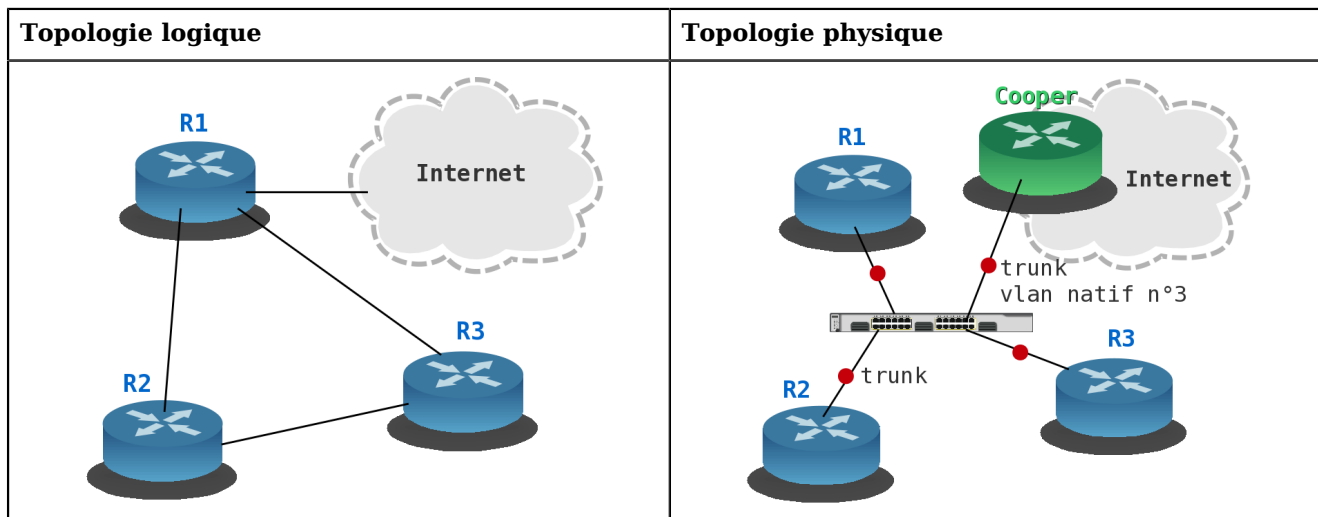
Topologie logique

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 *Routage Inter-VLAN*⁴ 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



Après avoir mis en œuvre la topologie physique en s'appuyant sur le support de la séance de travaux pratiques précédente : *Routage Inter-VLAN*⁵, 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*⁶ donne l'adresse du routeur central connecté à l'Internet.

- 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	sw5.infra.stri	alderaan	R1	3	eth0	172.16.1.1/20

⁴ <http://www.inetdoc.net/articles/inter-vlan-routing/>

⁵ <http://www.inetdoc.net/articles/inter-vlan-routing/>

⁶ http://www.inetdoc.net/travaux_pratiques/infra.tp/

Groupe	Commutateur	Poste	Rôle	VLAN	Interface	Réseau
				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	eth0.313	10.1.13.3/26
				323	eth0.323	10.1.23.3/26
2	sw6.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	sw7.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	sw8.infra.stri	mustafar	R1	3	eth0	172.16.4.1/20
				372	eth0.372	10.4.12.1/26
				373	eth0.373	10.4.13.1/26
		naboo	R2	372	eth0.372	10.4.12.2/26
				383	eth0.383	10.4.23.2/26
		tatooine	R3	373	eth0.373	10.4.13.3/26
383	eth0.383			10.4.23.3/26		

Le positionnement des 4 commutateurs est référencé dans le support *Architecture réseau des travaux pratiques*⁷.

⁷ http://www.inetdoc.net/travaux_pratiques/infra.tp/

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 brasser les postes sur les commutateurs non programmés.

```
$ dpkg -l quagga vlan | grep ^ii
ii quagga 0.99.17-2 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 de réseau local*⁸.

```
# 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:~# grep -v '^#' /etc/quagga/daemons
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
```

⁸ http://www.inetdoc.net/travaux_pratiques/config.interface.lan/

```
!  
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# grep -l bandwidth zebra.conf  
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 ?

Au niveau liaison, utiliser la commande fournie avec le paquet `vlan` pour affecter les numéros de VLANs à l'interface de chaque routeur. Au niveau réseau, configurer les adresses IP sur les sous-interfaces créées.

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

Lancer les tests ICMP usuels entre chaque routeur sur chaque lien actif.

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 ?

Utiliser une commande usuelle de visualisation de la table de routage (**route** ou **ip**) puis afficher la même table de routage à partir de la connexion au démon zebra avec la commande usuelle du système Cisco™ IOS.

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

Reprendre l'instruction présentée dans le document *Configuration d'une interface de réseau local : activation du routage*⁹.

L'opération doit être répétée sur chacun des trois routeurs pour que le protocole de routage dynamique puisse fonctionner normalement.

⁹ http://www.inetdoc.net/travaux_pratiques/config.interface.lan/config.interface.lan.procs.html#change_proc

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`, lancer la commande de visualisation globale du protocole.
Cette commande est utilisable sur chacun des trois routeurs.
2. Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF et fixer manuellement l'identifiant du routeur ?
La liste de identifiants à utiliser pour les trois routeurs est la suivante.

R1	0.0.0.1
R2	0.0.0.2
R3	0.0.0.3

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
3. Quelles sont les opérations à effectuer pour activer le protocole de routage OSPF pour les réseaux connus de chaque routeur ?
Dans la configuration du démon `ospfd`, ajouter une entrée de réseau pour chaque lien connu du routeur. La liste des liens connus correspond aux entrées marquées c de la table de routage visualisée à partir du démon `zebra`.
 4. Quelle est la commande qui permet de visualiser l'état des interfaces actives du routeur vis-à-vis du protocole de routage OSPF ?
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.
 5. À partir des informations affichées dans la question précédente, retrouver l'identifiant de routeur et le type de réseau, repérer et identifier la présence d'un autre routeur sur le même réseau.
Pour chaque interface vue du démon de routage OSPF, repérer les informations relatives au type de réseau et au décomptage des routeurs voisins.
 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 ?
Visualiser les listes des routeurs voisins puis la liste des routes présentes dans la base de données du démon `ospfd`. Faire la correspondance entre les métriques affichées et les bandes passantes de chaque lien.
 7. Quel est le mode d'affichage de la table de routage du système qui offre le plus d'informations ?
Identifier le «lieu de la synthèse» de tous les canaux d'information sur la table de routage d'un routeur.
 8. Comment visualiser la table de routage au niveau système ?
Utiliser une commande usuelle de visualisation de la table de routage.

6. 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 vont bientôt le devenir.

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 ?

Rechercher, le mot clé *bandwidth* dans la liste des instructions de configuration du démon *ospfd*.

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

Normalement, le débit d'un lien est directement extrait des paramètres de l'interface connectée au lien. Dans le cas d'interface qui n'ont «aucune réalité physique», ce débit peut être attribué arbitrairement par configuration. On doit rechercher dans les options des démons *zebra* et *ospfd* le moyen d'attribuer un débit aux sous-interfaces de VLANs.

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

Visualiser les paramètres des interfaces réseau depuis la console du démon *ospfd*.

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.

À partir des informations de la base de données du démon *ospfd*, faire la somme des métriques de chaque lien entre les deux extrémités en communication.

7. 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 ?
Rechercher le mot clé `default` dans la liste des commandes relatives au démon `ospfd`.
2. Comment la publication de route par défaut apparaît-elle sur les différents routeurs OSPF ?
Relevez la métrique de la route par défaut sur les routeurs qui n'ont pas une connexion directe vers l'Internet.

8. 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.

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` et on édite la configuration du service de façon à limiter l'accès à l'adresse IP voulue.

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.

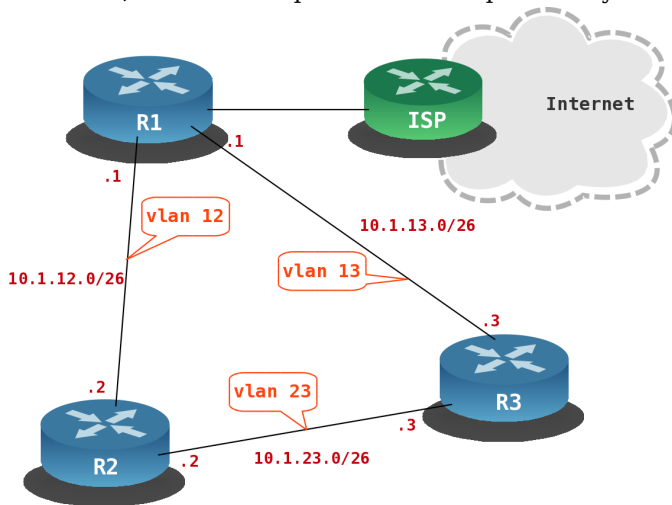
4. Comment valider l'accès à ce service Web depuis les autres routeurs ?

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.

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 enseignement*¹⁰.

Voici quelques éléments sur la mise en œuvre de cette «infrastructure de travaux pratiques». Dans la figure ci-dessous, le routeur baptisé *ISP* correspond au système hôte sur lequel les systèmes virtuels sont exécutés.



9.1. Préparation des routeurs

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

```
$ cd ~/vm
$ mkdir ospf
$ qemu-img create -b vm0-debian-i386.raw -f raw ospf/r1.raw
$ qemu-img create -b vm0-debian-i386.raw -f raw ospf/r2.raw
$ qemu-img create -b vm0-debian-i386.raw -f raw ospf/r3.raw
```

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 ospf-lab.sh

#!/bin/bash

../scripts/startup.sh r1.raw 512 2
../scripts/startup.sh r2.raw 512 3
../scripts/startup.sh r3.raw 512 4
```

Le code du script *startup.sh* donné ci-dessous est extrait de l'article cité plus haut.

```
#!/bin/bash
# $Id: startup.sh 1614 2011-03-17 22:41:04Z latu $

#RedOnBlack='\E[31;40m'
RedOnBlack='\E[31m'

vm=$1
shift
memory=$1
shift
port=$1
shift

if [[ -z "$vm" || -z "$memory" || -z "$port" ]]
then
    echo "ERREUR : paramètre manquant"
    echo "Utilisation : $0 <fichier image> <quantité mémoire en Mo> <port commutateur [2..32]>"
    exit 1
fi

if (( $memory < 128 ))
then
    echo "ERREUR : quantité de mémoire RAM insuffisante"
    echo "La quantité de mémoire en Mo doit être supérieure ou égale à 128"
```

¹⁰ <http://www.inetdoc.net/guides/vm/>

```

exit 1
fi

macaddress="52:54:00:12:34:$port"

echo -e "$RedOnBlack"
echo "~> Machine virtuelle : $vm"
echo "~> Mémoire RAM       : $memory"
echo "~> Port commutateur  : $port"
echo "~> Adresse MAC       : $macaddress"
tput sgr0

# -vga vmware \
kvm \
-daemonize \
-name $vm \
-m $memory \
-rtc base=localtime,clock=host \
-drive file=$vm,if=virtio,media=disk,boot=on \
-k fr \
-usb -usbdevice tablet \
-soundhw es1370 \
-net vde,vlan=1,sock=/var/run/vde2/tap0.ctl,port=$port \
-net nic,vlan=1,model=virtio,macaddr=$macaddress \
$*

```

9.2. Table de routage du système hôte

Pour que les réseaux de l'aire OSPF puissent communiquer avec le système hôte et l'Internet, il est nécessaire de compléter la table de routage du système hôte. Dans ce contexte le système hôte joue le rôle d'un routeur central et la technique usuelle employée pour répondre au besoin d'interconnexion consiste à implanter une «super route» qui rassemble tous les réseaux de l'aire en une seule entrée.

L'aire OSPF étudiée contient quatre réseaux :

- 10.1.3.0/29 - réseau fictif ajouté sur R3,
- 10.1.12.0/26 - réseau correspondant au lien entre R1 et R2,
- 10.1.13.0/26 - réseau correspondant au lien entre R1 et R3,
- 10.1.23.0/26 - réseau correspondant au lien entre R2 et R3,

L'utilisation de l'outil `ipcalc` permet de vérifier qu'un masque de 19 bits permet d'englober ces quatre réseaux en une seule entrée.

```

$ ipcalc 10.1.0.0/19
Address:  10.1.0.0          00001010.00000001.000 00000.00000000
Netmask:  255.255.224.0 = 19 11111111.11111111.111 00000.00000000
Wildcard: 0.0.31.255      00000000.00000000.000 11111.11111111
=>
Network:  10.1.0.0/19      00001010.00000001.000 00000.00000000
HostMin:   10.1.0.1        00001010.00000001.000 00000.00000001
HostMax:   10.1.31.254     00001010.00000001.000 11111.11111110
Broadcast: 10.1.31.255    00001010.00000001.000 11111.11111111
Hosts/Net: 8190           Class A, Private Internet

```

On complète donc la table de routage du système hôte avec l'instruction suivante :

```
# ip route add 10.1.0.0/19 dev tap0
```

Le fait de désigner la destination par l'interface `tap0` optimise le traitement dans la mesure où le processus de commutation de paquet connaît directement l'interface sur laquelle placer les paquets IP sans passer par un examen de la table de routage ou plus exactement les *hashes* de la *Forwarding Information Base* (FIB).

Une fois cette nouvelle entrée de la table de routage du système hôte en place, on peut valider l'accessibilité des réseaux de l'aire en testant le service Web factice implanté sur le routeur R3 depuis le système hôte.

```

# nmap -A -p80 10.1.3.3

Starting Nmap 5.21 ( http://nmap.org ) at 2010-11-18 22:44 CET
Nmap scan report for 10.1.3.3
Host is up (0.0019s latency).
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.2.16 ((Debian))
|_html-   title: Site doesn't have a title (text/html)
<snipped/>

```

9.3. Configuration du commutateur virtuel vde

Comme indiqué dans l'article cité en référence ci-dessus, le lancement du commutateur virtuel *Virtual Distributed Ethernet*¹¹ est intégré à la configuration du système hôte. Voici un extrait du fichier `/etc/network/interfaces` relatif à la configuration de l'interface `tap0`.

```
# The tap0 network interface
iface tap0 inet static
```

¹¹ <http://vde.sourceforge.net/>

```

address 192.0.2.1
netmask 255.255.255.224
network 192.0.2.0
broadcast 192.0.2.31
vde2-switch -
post-up /etc/init.d/isc-dhcp-server restart
post-up /etc/init.d/ntp restart
post-up /etc/init.d/bind9 reload

```

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 ./ospf-lab.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, on visualise les connexions sur les 4 ports occupés.

```

$ vdeterm /var/run/vde2/tap0.mgmt
VDE switch V.2.2.3
(C) Virtual Square Team (coord. R. Davoli) 2005,2006,2007 - GPLv2

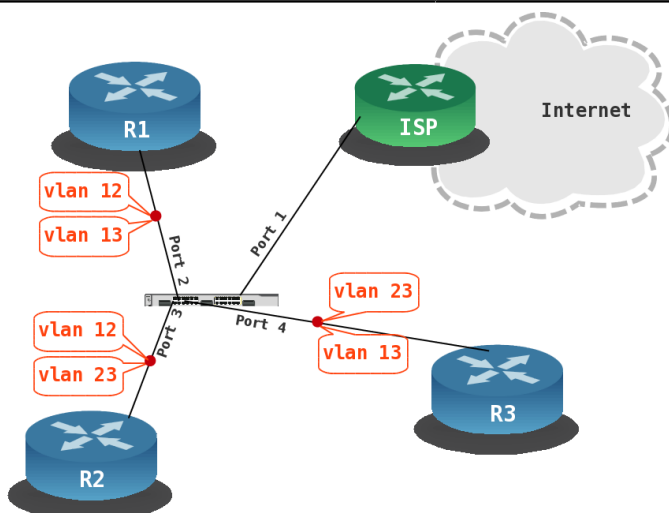
vde$ port/print
0000 DATA END WITH '.'
Port 0001 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
Current User: NONE Access Control: (User: NONE - Group: NONE)
IN:  pkts      2288      bytes      1288845
OUT: pkts      1239      bytes      221354
-- endpoint ID 0007 module tuntap      : tap0
Port 0002 untagged_vlan=0000 ACTIVE - Unnamed Allocatable
Current User: ph1l Access Control: (User: NONE - Group: NONE)
IN:  pkts      1870      bytes      275892
OUT: pkts      2831      bytes      1328457
-- endpoint ID 0011 module unix prog   : QEMU user=ph1l PID=2982 \
SOCK=/var/run/vde2/tap0.ctl/.02982-00000
Port 0003 untagged_vlan=0099 ACTIVE - Unnamed Allocatable
Current User: ph1l Access Control: (User: NONE - Group: NONE)
IN:  pkts      639      bytes      55474
OUT: pkts      645      bytes      57546
-- endpoint ID 0003 module unix prog   : QEMU user=ph1l PID=2986 \
SOCK=/var/run/vde2/tap0.ctl/.02986-00000
Port 0004 untagged_vlan=0099 ACTIVE - Unnamed Allocatable
Current User: ph1l Access Control: (User: NONE - Group: NONE)
IN:  pkts      642      bytes      55372
OUT: pkts      638      bytes      57080
-- endpoint ID 0009 module unix prog   : QEMU user=ph1l PID=2989 \
SOCK=/var/run/vde2/tap0.ctl/.02989-00000
.
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
4	R3	trunk R1 + R2



Une fois le brassage en place, on peut passer à la configuration des VLANs ; toujours via l'interface de configuration du commutateur virtuel. Il est possible d'utiliser un fichier de sauvegarde de la liste des instructions de configuration du commutateur : `vde.conf` dans notre exemple. On charge ces instructions dans le commutateur virtuel via la commande `vde$ load /home/ph11/vm/ospf/vde.conf`.

```
vde$ load /home/ph11/vm/ospf/vde.conf
vde[/home/ph11/vm/ospf/vde.conf]: vlan/create 12
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/create 13
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/create 23
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/addport 12 2
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/addport 13 2
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/addport 12 3
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/addport 23 3
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/addport 13 4
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/addport 23 4
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: plugin/add /usr/lib/vde2/plugins/pdump.so
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: vlan/create 99
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: port/setvlan 3 99
1000 Success
vde[/home/ph11/vm/ospf/vde.conf]: port/setvlan 4 99
1000 Success
1000 Success
```

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
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
VLAN 0099
-- Port 0003 tagged=0 active=1 status=Forwarding
-- Port 0004 tagged=0 active=1 status=Forwarding
.
1000 Success
```

9.4. Utilisation de l'interface virtio

Pour accéder à des débits réseau plus élevés sur les instances de machines virtuelles, on utilise 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 n'est pas nécessaire d'intervenir sur la configuration système des routeurs R1, R2 et R3. Les paquets de noyaux fournis par la distribution *Debian GNU/Linux* possèdent toutes les fonctions nécessaires.

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.

Sur le routeur R2, on lance la partie serveur de l'outil *iperf*.

```
etu@r2:~$ iperf -s -w 65536
```

```
-----  
Server listening on TCP port 5001  
TCP window size: 128 KByte (WARNING: requested 64.0 KByte)  
-----
```

Sur le routeur R1, on lance la partie cliente du même outil. On fixe le temps de transmission à 180 secondes avec 4 *threads* en parallèle.

```
etu@r1:~$ iperf -c 10.1.12.2 -w 65536 -t 180 -P 4  
-----  
Client connecting to 10.1.12.2, TCP port 5001  
TCP window size: 128 KByte (WARNING: requested 64.0 KByte)  
-----  
[ 4] local 10.1.12.1 port 35103 connected with 10.1.12.2 port 5001  
[ 5] local 10.1.12.1 port 35104 connected with 10.1.12.2 port 5001  
[ 3] local 10.1.12.1 port 35102 connected with 10.1.12.2 port 5001  
[ 6] local 10.1.12.1 port 35105 connected with 10.1.12.2 port 5001  
[ ID] Interval      Transfer      Bandwidth  
[ 4]  0.0-180.0 sec  7.88 GBytes   376 Mbits/sec  
[ 5]  0.0-180.0 sec  7.94 GBytes   379 Mbits/sec  
[ 3]  0.0-180.0 sec  7.67 GBytes   366 Mbits/sec  
[ 6]  0.0-180.0 sec  7.72 GBytes   369 Mbits/sec  
[SUM] 0.0-180.0 sec 31.2 GBytes  1.49 Gbits/sec
```

Avec un débit total de 1.49 Gbits/sec, on peut dire que les conditions de transmission réseau entre deux instances de systèmes virtualisés sont très bonnes.

10. Documents de référence

Architecture réseau des travaux pratiques

Le support *Architecture réseau des travaux pratiques*¹² présente la topologie physique de la salle de travaux pratiques avec la *Disposition des équipements dans l'armoire de brassage*¹³ ainsi que les configurations par défaut des équipements. On y trouve aussi le plan d'adressage IP utilisé avec les autres supports de travaux pratiques, le plan de numérotations des VLANs et les affectations des groupes de ports des commutateurs.

Configuration d'une interface réseau

Le support *Configuration d'une interface de réseau local*¹⁴ 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*¹⁵ 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.

Initiation au routage, 3ème partie

L'article *Initiation au routage, 3ème partie*¹⁶ 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 *Routage Inter-VLAN*¹⁷ 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)*¹⁸ 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.

Virtualisation système et enseignement

Le support *Virtualisation système et enseignement*¹⁹ présente la solution de virtualisation intégrée au noyau Linux : KVM. Associée au commutateur *Virtual Distributed Ethernet*²⁰, cette solution permet de construire des maquettes de travaux pratiques très complètes en offrant de nombreuses fonctions réseau «réelles» dont une table CAM²¹.

¹² http://www.inetdoc.net/travaux_pratiques/infra.tp/

¹³ http://www.inetdoc.net/travaux_pratiques/infra.tp/infra.tp.interco.html#infra.tp.interco.cabling

¹⁴ http://www.inetdoc.net/travaux_pratiques/config.interface.lan/

¹⁵ <http://www.inetdoc.net/guides/zebra.statique/>

¹⁶ <http://www.inetdoc.net/guides/zebra.ospf/>

¹⁷ <http://www.inetdoc.net/articles/inter-vlan-routing/>

¹⁸ http://www.cisco.com/en/US/tech/tk365/tk480/tsd_technology_support_sub-protocol_home.html

¹⁹ <http://www.inetdoc.net/guides/vm/>

²⁰ <http://vde.sourceforge.net/>

²¹ http://en.wikipedia.org/wiki/CAM_Table