

Initiation au routage, 3ème partie

Publié par :

Philippe Latu

philippe.latu(at)linux-france.org

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

Historique des versions		
\$Revision: 1321 \$	\$Date: 2008-09-24 10:21:50 +0200 (mer 24 sep 2008) \$	\$Author: latu \$
Année universitaire 2006-2007		
Résumé		
<p>Ce document est le troisième article d'une série rédigée par Pacôme Massol sur l'utilisation d'un système GNU/Linux comme routeur. Le logiciel présenté : Zebra ainsi que son successeur Quagga possèdent de nombreux atouts pour faciliter le développement de solutions d'interconnexion réseau entièrement basées sur du logiciel libre. Ce document a été publié dans le numéro 51 de Linux Magazine en Juin 2003. La version publiée ici pour le projet inetdoc.LINUX ne contient que quelques différences mineures sur la présentation et la configuration du logiciel.</p>		

Table des matières

1. Copyright et Licence	2
1.1. Meta-information	2
2. Avant-propos	2
2.1. Résumé des épisodes précédents	2
3. Les grands principes d'OSPF	3
3.1. La notion de coût	3
3.2. La base de données topologique	4
3.3. L'élection des meilleures routes	4
3.4. La détermination d'une table de routage	5
4. Le fonctionnement d'OSPF un peu plus en détail	5
4.1. État initial	6
4.2. Établir la liste des routeurs voisins : <i>Hello, my name is R1 and I'm an OSPF router.</i>	6
4.3. Élire le routeur désigné et le routeur désigné de secours	6
4.4. Découvrir les routes	6
4.5. Élire les routes à utiliser	7
4.6. Maintenir la base topologique	7
4.7. Conclusion partielle	7
5. Le concept de zone (<i>area</i>)	7
6. Place à la pratique	9
6.1. Situation de départ	10
6.2. Activation du processus de routage	10
6.3. Activation des annonces de routes	10
6.4. Affichage de la configuration	11
6.5. État des routeurs	12
6.6. Quelques éléments sur la sécurité	14
6.6.1. Filtrer la diffusion des routes	14
6.6.2. Protéger les annonces de routes	14
7. Conclusion	15
7.1. Bibliographie	15
7.2. Liens	15

1. Copyright et Licence

```
Copyright (c) 2002-2005 Pacôme Massol
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 sect2 entitled "GNU
Free Documentation License".
```

```
Copyright (c) 2002-2005 Pacôme Massol
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 1.1 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 sect2 intitulée
« Licence de Documentation Libre GNU ».
```

1.1. Meta-information

Cet article est écrit avec *DocBook*² XML sur un système *Debian GNU/Linux*³. Il est disponible en version imprimable aux formats PDF et Postscript : [zebra.ospf.pdf](#)⁴ | [zebra.ospf.ps.gz](#)⁵.

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. Comme la distribution *Debian GNU/Linux* est utilisée pour l'ensemble des supports du projet *inetdoc.LINUX*, voici une liste des paquets contenant les commandes nécessaires :

- net-tools - The NET-3 networking toolkit
- quagga - Unofficial successor of the Zebra BGP/OSPF/RIP routing daemon
- quagga-doc - info files for quagga
- zebra & zebra-doc - anciens paquets non mis à jour depuis 2003.

Les copies d'écran présentées ici correspondent à la publication initiale. Depuis, les versions de Zebra puis de Quagga ont évolué et l'affichage des informations de routage a été modifié. Cependant, ces modifications ne devraient pas gêner les lecteurs.

2. Avant-propos

Ce document est la suite de deux articles publiés respectivement dans *Linux Magazine*⁶ numéros 42 de septembre et 43 d'octobre 2002. Étant donné que cela remonte déjà à quelques temps, un rapide résumé est peut-être nécessaire avant de passer au vif du sujet : le protocole de routage dynamique OSPF et sa mise en oeuvre avec Zebra.

2.1. Résumé des épisodes précédents

Le routeur est un élément essentiel dans l'aiguillage des paquets de données dans un inter-réseau. Pour chaque paquet reçu, il extrait le préfixe réseau de l'adresse IP de destination du paquet et le recherche dans une table qu'il possède en mémoire. Cette table de routage contient essentiellement une liste d'adresses réseau et, pour chacune, le moyen de

² <http://www.docbook.org>

³ <http://www.debian.org>

⁴ <http://www.linux-france.org/prj/inetdoc/telechargement/zebra.ospf.pdf>

⁵ <http://www.linux-france.org/prj/inetdoc/telechargement/zebra.ospf.ps.gz>

⁶ <http://www.linuxmag-france.org/>

l'atteindre, à savoir l'adresse d'un routeur immédiatement voisin et situé sur la route vers la destination. Si le routeur trouve dans cette table le préfixe réseau, il transmet le paquet sur le réseau du routeur voisin concerné. Ce processus sera renouvelé par le routeur voisin et ainsi de suite, de proche en proche le paquet sera orienté vers sa destination.

Seulement voilà, il faut saisir les tables de routage ! Travail fastidieux pour les petits doigts agiles de l'administrateur lorsque les réseaux sont de grande taille. De plus, compte tenu de l'évolution du nombre de réseaux à interconnecter dans le cas d'internet, il est de toute façon devenu impossible de se cantonner au routage statique (souvenez-vous, dans le document *1ère partie, routage statique*⁷ nous nous sommes adonnés aux joies de la saisie de routes statiques avec Zebra).

C'est pourquoi, le routage dynamique a été imaginé afin d'alléger la charge d'administration mais aussi pour réaliser des réseaux tolérants aux pannes d'un routeur ou d'une liaison. RIP est un bon exemple de protocole de routage dynamique (reportez-vous au document *2ème partie, routage RIP*⁸ pour découvrir tous ses secrets). Les routeurs supportant RIP s'échangent périodiquement des informations sur les routes qu'ils possèdent (les fameux «vecteurs de distance»). Si une panne se produit, les routeurs immédiatement voisins notent que certaines routes sont devenues inaccessibles et propagent l'information aux autres. Mais hélas, RIP souffre de certaines limitations qui ont poussé l'IETF (*The Internet Engineering Task Force*⁹) à plancher sur un protocole plus robuste, plus efficace, plus paramétrable et supportant des réseaux de grande taille. Cette merveille s'appelle OSPF (*Open Shortest Path First*).

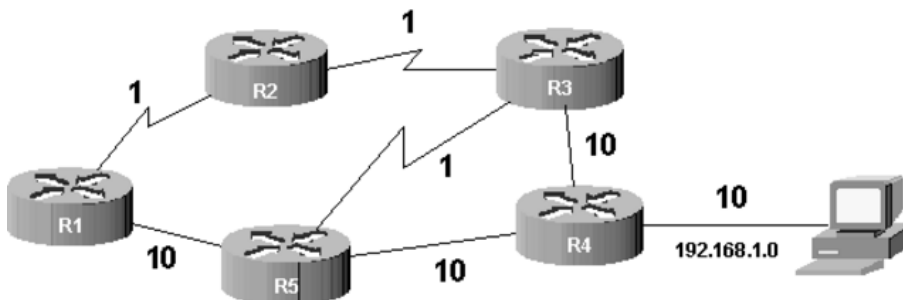
Le fonctionnement de ce protocole est défini dans le document *RFC2328*¹⁰. Les logiciels de Zebra et de son successeur Quagga supportent cette définition du protocole OSPF.

3. Les grands principes d'OSPF

OSPF est un protocole de routage dynamique défini par l'IETF à la fin des années 80. Il a fait l'objet d'un historique relativement complexe de RFCs (Voir *ospf RFC List*¹¹). Ce protocole a deux caractéristiques essentielles :

- Il est ouvert : c'est le sens du terme *Open* de OSPF. Son fonctionnement est connu de tous.
- Il utilise l'algorithme SPF pour *Shortest Path First*, plus connu sous le nom d'algorithme de Dijkstra, afin d'élire la meilleure route vers une destination donnée.

Examinons une topologie qui nous servira de support pour les explications :



Exemple de topologie¹²

Figure 1. Exemple de topologie

3.1. La notion de coût

Supposons que du routeur R1 on cherche à atteindre le réseau 192.168.1.0. Dans une telle situation, Le protocole RIP aurait élu la route passant par R5 puisque c'est la plus courte en termes de saut. Cependant, imaginez que les liens représentés sous forme d'éclairs soient «rapides» (de type FastEthernet à 100 Mbps par exemple) et que les liens

⁷ <http://www.linux-france.org/prj/inetdoc/guides/zebra.statique/>

⁸ <http://www.linux-france.org/prj/inetdoc/guides/zebra.rip/>

⁹ <http://ietf.org/>

¹⁰ <http://www.faqs.org/rfcs/rfc2328.html>

¹¹ <http://rtg.ietf.org/wg/ospf/rfclist>

¹² <http://www.linux-france.org/prj/inetdoc/guides/zebra.ospf/images/spf1.png>

représentés sous formes de segments droits soient «lents» (de type Ethernet à 10 Mbps par exemple). Le choix du protocole RIP n'est plus du tout pertinent !

Le protocole OSPF fonctionne différemment. Il attribue un coût à chaque liaison (appelée *lien* dans le jargon OSPF) afin de privilégier l'élection de certaines routes. Plus le coût est faible, plus le lien est intéressant. Par défaut, les coûts suivants sont utilisés en fonction de la bande passante du lien :

Type de réseau	Coût par défaut
FDDI, FastEthernet	1
Ethernet 10 Mbps	10
E1 (2,048 Mbps)	48
T1 (1,544 Mbps)	65
64 Kbps	1562
56 Kbps	1758
19.2 Kbps	5208

La formule de calcul est simplissime :

$$\text{coût} = \frac{10^8}{\text{bande passante du lien en bps}}$$

La référence 10^8 correspond à un débit maximum de 100Mbps. Dans le cas où l'on utilise des interfaces avec un débit supérieur, il est possible de redéfinir la référence avec une commande du type `auto-cost reference-bandwidth 1000` pour la valeur 10^9 .

Le protocole OSPF privilégie les routes qui ont un coût faible, donc celles qui sont supposées rapides en terme de débit théorique.

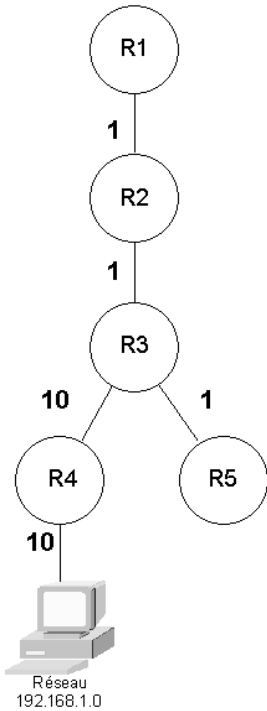
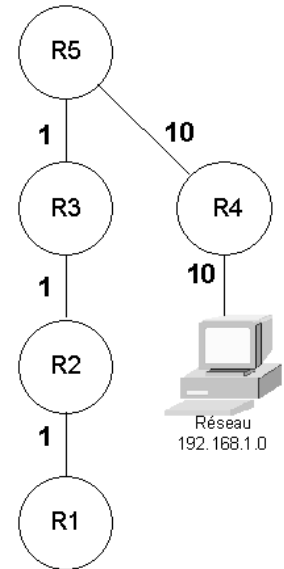
3.2. La base de données topologique

Avec le protocole OSPF, tous les routeurs d'un même réseau (on parle de «zone» dans le vocabulaire OSPF, ceci vous sera expliqué avant la mise en pratique) travaillent sur une base de données topologique identique qui décrit le réseau. Cette base a été constituée pendant une première phase de découverte qui vous sera expliquée un peu plus loin. Examinons la base de données suivante qui décrit la topologie de la [Figure 1, « Exemple de topologie »](#) :

Arc	Coût
R1, R2	1
R1, R5	10
R2, R3	1
R3, R4	10
R3, R5	1
R4, R5	10
R4, 192.168.1.0	10

3.3. L'élection des meilleures routes

L'algorithme SPF de Dijkstra va traiter cette base de données afin de déterminer les routes les moins coûteuses. Une fois le traitement réalisé, chaque routeur se voit comme la racine d'un arbre contenant les meilleures routes. Par exemple :

Topologie vue de R1¹³Topologie vue de R5¹⁴

Dans l'exemple, entre R1 et 192.168.1.0, la meilleure route passe par R2, R3 et R4 pour un coût total de $1 + 1 + 10 + 10$ soit 22.

3.4. La détermination d'une table de routage

La base de données topologique décrit le réseau mais ne sert pas directement au routage. La table de routage est déterminée par l'application de l'algorithme du SPF sur la base topologique. Sur R1, voici un extrait de la table de routage calculée par SPF au sujet du réseau 192.168.1.0 :

Réseau de destination	Moyen de l'atteindre	Coût
192.168.1.0	R2	22

Sur R5, on aura l'extrait suivant :

Réseau de destination	Moyen de l'atteindre	Coût
192.168.1.0	R4	20

4. Le fonctionnement d'OSPF un peu plus en détail

Pour administrer un réseau OSPF correctement, il est indispensable de comprendre le fonctionnement interne du protocole.

à l'intérieur d'une même zone, les routeurs fonctionnant sous OSPF doivent préalablement remplir les tâches suivantes avant de pouvoir effectuer leur travail de routage :

1. Section 4.2, « Établir la liste des routeurs voisins : *Hello, my name is R1 and I'm an OSPF router.* »,
2. Section 4.3, « Élire le routeur désigné et le routeur désigné de secours »,
3. Section 4.4, « Découvrir les routes »,

¹⁴ <http://www.linux-france.org/prj/inetdoc/guides/zebra.ospf/images/spf3.png>

¹³ <http://www.linux-france.org/prj/inetdoc/guides/zebra.ospf/images/spf2.png>

4. Section 4.5, « Élire les routes à utiliser »,
5. Section 4.6, « Maintenir la base topologique ».

4.1. État initial

Le processus de routage OSPF est inactif sur tous les routeurs de la Figure 1, « Exemple de topologie ».

4.2. Établir la liste des routeurs voisins : *Hello, my name is R1 and I'm an OSPF router.*

Les routeurs OSPF sont bien élevés. Dès qu'ils sont activés, ils n'ont qu'une hâte : se présenter et faire connaissance avec leurs voisins. En effet, lorsque le processus de routage est lancé sur R1 (commande `router ospf`), des paquets de données (appelés paquets HELLO) sont envoyés sur chaque interface où le routage dynamique a été activé (commande `network`).

L'adresse *multicast* 224.0.0.5 est utilisée, tout routeur OSPF se considère comme destinataire. Ces paquets ont pour but de s'annoncer auprès de ses voisins. Deux routeurs sont dits voisins s'ils ont au moins un lien en commun. Par exemple, sur la Figure 1, « Exemple de topologie », R1 et R2 sont voisins mais pas R1 et R3.

Lorsque le processus de routage OSPF est lancé sur R2, celui-ci récupère les paquets HELLO émis par R1 toutes les 10 secondes (valeur par défaut du temporisateur appelé *hello interval*). R2 intègre l'adresse IP de R1 dans une base de données appelée «base d'adjacences» (*adjacencies database*). Cette base contient les adresses des routeurs voisins. Vous pourrez visionner son contenu grâce à la commande `show ip ospf neighbor`. R2 répond à R1 par un paquet IP *unicast*. R1 intègre l'adresse IP de R2 dans sa propre base d'adjacences. Ensuite, généralisez ce processus à l'ensemble des routeurs de la zone.

Cette phase de découverte des voisins est fondamentale puisque OSPF est un protocole à état de liens. Il lui faut connaître ses voisins pour déterminer s'ils sont toujours joignables et donc déterminer l'état du lien qui les relie.

4.3. Élire le routeur désigné et le routeur désigné de secours

Dans une zone OSPF composée de réseaux de diffusion (*broadcast networks*) ou de réseau à accès multiples sans diffusion (*non broadcast multiple access networks* ou NBMA), l'un des routeurs doit être élu «routeur désigné» (DR pour *Designated Router*) et un autre «routeur désigné de secours» (BDR pour *Backup Designated Router*). Le «routeur désigné» (DR) est un routeur particulier qui sert de référent pour la base de données topologique représentant le réseau.

Pourquoi élire un routeur désigné ? Cela répond à trois objectifs :

- réduire le trafic lié à l'échange d'informations sur l'état des liens (car il n'y a pas d'échange entre tous les routeurs mais entre chaque routeur et le DR),
- améliorer l'intégrité de la base de données topologique (car cette base de données doit être unique),
- accélérer la convergence (souvenez-vous, c'était le talon d'Achille de RIP).

Comment élire le DR ? Autrement dit, qui va se taper la corvée d'expliquer à ses petits camarades la topologie du réseau ? On ne demande pas qui sait parler anglais ou couper les cheveux comme au temps de la conscription. Mais comme il faut bien un critère, le routeur élu est celui qui a la plus grande priorité (*Router ID* ou RID). La priorité est un nombre sur 8 bits fixé par défaut à 1 sur tous les routeurs. Pour départager les routeurs ayant la même priorité, celui qui est élu a la plus grande adresse IP sur une interface de boucle locale (*loopback interface*) ou sur un autre type d'interface active. Le BDR sera le routeur avec la deuxième plus grande priorité.

Afin de s'assurer que votre routeur préféré sera élu DR, il suffit de lui affecter une priorité supérieure à 1 avec la commande `ospf priority`. Vous devrez faire ceci avant d'activer le processus de routage sur les routeurs car, une fois élu, le DR n'est jamais remis en cause même si un routeur avec une priorité plus grande apparaît dans la zone.

4.4. Découvrir les routes

Il faut maintenant constituer la base de données topologique. Les routeurs communiquent automatiquement les routes pour les réseaux qui participent au routage dynamique (ceux déclarés avec la commande `network`). Zebra et son

successeur Quagga étant multiprotocoles, ils peuvent également diffuser des routes provenant d'autres sources que OSPF, grâce à la commande `redistribute`.

Chaque routeur (non DR ou BDR) établit une relation maître/esclave avec le DR. Le DR initie l'échange en transmettant au routeur un résumé de sa base de données topologique via des paquets de données appelés LSA (*Link State Advertisement*).

Ces paquets comprennent essentiellement l'adresse du routeur, le coût du lien et un numéro de séquence. Ce numéro est un moyen pour déterminer l'ancienneté des informations reçues. Si les LSA reçus sont plus récents que ceux dans sa base topologique, le routeur demande une information plus complète par un paquet LSR (*Link State Request*). Le DR répond par des paquets LSU (*Link State Update*) contenant l'intégralité de l'information demandée. Ensuite, le routeur (non DR ou BDR) transmet les routes meilleures ou inconnues du DR.

L'administrateur peut consulter la base de données topologique grâce à la commande `show ip ospf database`.

4.5. Élire les routes à utiliser

Lorsque le routeur est en possession de la base de données topologique, il est en mesure de créer la table de routage. L'algorithme du SPF est appliqué sur la base topologique. Il en ressort une table de routage contenant les routes les moins coûteuses.

Il faut noter que sur une base de données topologique importante, le calcul consomme pas mal de ressources CPU car l'algorithme est relativement complexe.

4.6. Maintenir la base topologique

Lorsqu'un routeur détecte un changement de l'état d'un lien (cette détection se fait grâce aux paquets HELLO adressés périodiquement par le routeur à ses voisins), celui-ci émet un paquet LSU sur l'adresse *multicast* 224.0.0.6 : le DR et le BDR de la zone se considèrent comme destinataires.

Le DR et le BDR intègrent cette information à leur base topologique. Le DR et diffuse l'information sur l'adresse 224.0.0.5 (tous les routeurs OSPF sans distinction). C'est le protocole d'inondation. Toute modification de la topologie déclenche une nouvelle exécution de l'algorithme du SPF et une nouvelle table de routage est constituée.

4.7. Conclusion partielle

Voilà pour les principes fondamentaux d'OSPF mais des notions importantes restent à évoquer si vous souhaitez déployer OSPF sur de grands réseaux (en particulier sur le fonctionnement d'OSPF sur un réseau point à point et sur l'agrégation de routes). Si vous voulez approfondir, reportez-vous au livre de **C. Huitema cité en bibliographie** qui, bien qu'un peu ancien est très complet sur la question. Bien sûr, vous pouvez toujours vous plonger dans les différentes RFC qui constituent OSPF (la **RFC2328**¹⁵ en particulier) et dont la lecture est toujours aussi agréable et passionnante ! (je plaisante, bien sûr).

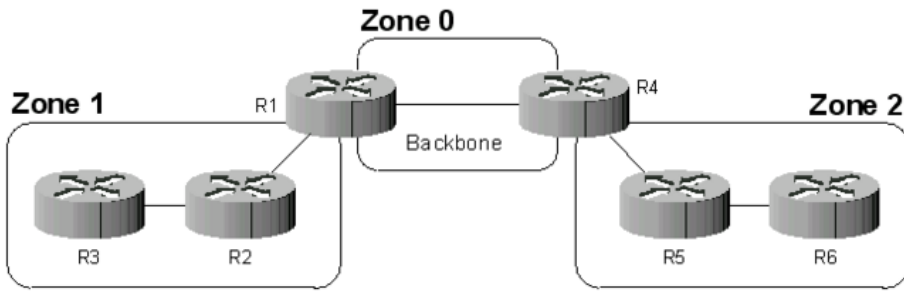
Avant d'attaquer la pratique, un dernier concept : les zones OSPF.

5. Le concept de zone (*area*)

Contrairement à RIP, OSPF a été pensé pour supporter de très grands réseaux. Mais, qui dit grand réseau, dit nombreuses routes. Donc, afin d'éviter que la bande passante ne soit engloutie dans la diffusion des routes, OSPF introduit le concept de zone (*area*). Le réseau est divisé en plusieurs zones de routage qui contiennent des routeurs et des hôtes.

Chaque zone, identifiée par un numéro, possède sa propre topologie et ne connaît pas la topologie des autres zones. Chaque routeur d'une zone donnée ne connaît que les routeurs de sa propre zone ainsi que la façon d'atteindre une zone particulière, la zone numéro 0. Toutes les zones doivent être connectées physiquement à la zone 0 (appelée *backbone* ou réseau fédérateur). Elle est constituée de plusieurs routeurs interconnectés. Le *backbone* est chargé de diffuser les informations de routage qu'il reçoit d'une zone aux autres zones. Tout routage basé sur OSPF doit posséder une zone 0.

¹⁵ <http://www.faqs.org/rfcs/rfc2328.html>



Réseau découpé en trois zones¹⁶

Figure 2. Réseau découpé en trois zones

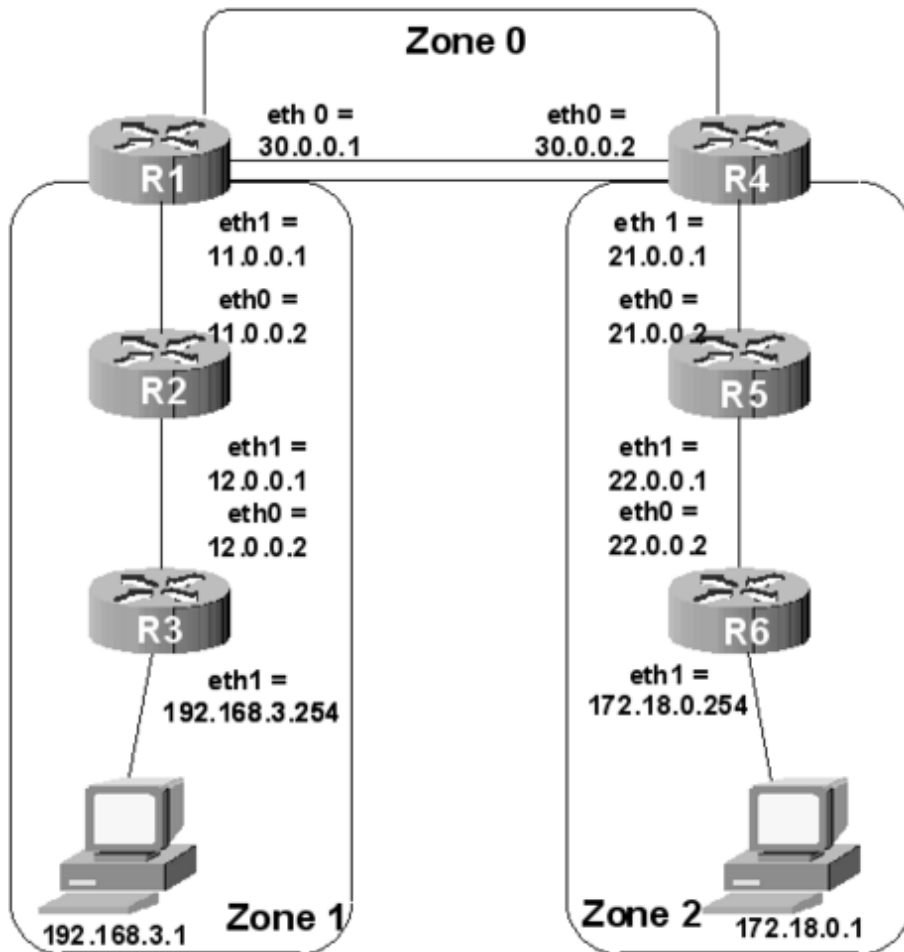
Sur la **Figure 2**, « Réseau découpé en trois zones », le réseau est découpé en trois zones dont le *backbone*. Les routeurs de la zone 1, par exemple, ne connaissent pas les routeurs de la zone 2 et encore moins la topologie de la zone 2. L'intérêt de définir des zones est de limiter le trafic de routage, de réduire la fréquence des calculs du plus court chemin par l'algorithme SPF ainsi que d'avoir une table de routage plus petite (ce qui accélère la convergence). Les routeurs R1 et R4 sont particuliers puisqu'ils sont «à cheval» entre plusieurs zones (on les appelle ABR pour *Area Border Router* ou routeur de bordure de zone). Ces routeurs maintiennent une base de données topologique pour chaque zone à laquelle ils sont connectés. Les ABR sont des points de sortie pour les zones ce qui signifie que les informations de routage destinées aux autres zones doivent passer par l'ABR local à la zone. L'ABR se charge alors de retransmettre les informations de routage au *backbone*.

Les ABRs du *backbone* ensuite redistribueront ces informations aux autres zones auxquelles ils sont connectés.

¹⁶ <http://www.linux-france.org/prj/inetdoc/guides/zebra.ospf/images/spf4.png>

6. Place à la pratique

Nous allons travailler avec le réseau suivant :



Topologie de travail¹⁷

Figure 3. La topologie de travail

Le réseau a été découpé en trois zones. Vous remarquerez que la zone 0 permet de fédérer l'ensemble du réseau. Il s'agit du *backbone* dont nous avons déjà discuté. Le découpage de ce réseau en trois zones est un cas d'école dont le but est d'examiner la configuration d'OSPF dans un contexte multi-zone.

Généralement, on considère qu'une zone peut accueillir plusieurs dizaines de routeurs.

Pour ne pas surcharger ces lignes inutilement, nous nous en tiendrons ici à la configuration de R1, R2 et R3. Vous verrez que la configuration n'est pas très complexe. Par symétrie, il est facile de l'adapter aux autres routeurs.

Pour votre service, chers lecteurs, j'ai mis en ligne une carte cliquable (*Zebra et OSPF*¹⁸) qui vous permettra de consulter l'état, la configuration complète et la table de routage des six routeurs.

Enfin, avant de commencer, vous trouverez une traduction partielle de la documentation de Zebra à la page : *GNU Zebra*¹⁹.

¹⁷ <http://www.linux-france.org/prj/inetdoc/guides/zebra.ospf/images/spf5.png>

¹⁸ <http://pmassol.net/lm/ospf.html>

¹⁹ <http://pmassol.net/zebra.html>

6.1. Situation de départ

Vous devez créer des fichiers de configuration rudimentaires sur chaque routeur pour les services Zebra ou Quagga :

- Fichier `/etc/zebra/zebra.conf` ou `/etc/quagga/zebra.conf` pour le démon principal.

Exemple pour R1 :

```
hostname R1(ZEBRA)
password foo
```

- Fichier `/etc/zebra/ospfd.conf` ou `/etc/quagga/ospfd.conf` pour le démon `ospfd`

Exemple pour R1 :

```
hostname R1(OSPF)
password foo
```

Vous devez ensuite démarrer (ou redémarrer) les deux démons `zebra` et `ospfd` sur chaque routeur.



Précaution avec une configuration manuelle

Veillez bien à respecter l'ordre d'exécution des services : d'abord `zebra -d` puis `ospfd` ensuite.

Enfin, sur R1 entrez dans le terminal de configuration de `ospfd` via **telnet** sur le port 2604 :

```
Linux# telnet localhost ospfd

Hello, this is zebra (version 0.91a).
Copyright 1996-2001 Kunihiro Ishiguro.

User Access Verification
Password:
R1(OSPF)> enable
R1(OSPF)#
```

Si vous avez envie de suivre précisément les échanges de messages entre routeurs, Zebra propose un puissant mécanisme de débogage grâce à la commande **debug** (je vous laisse découvrir tous ses paramètres). Supposons que je veuille garder une trace de tous les messages HELLO émis par R1 :

```
R1(OSPF)# conf t
R1(OSPF)(config)# log file /var/log/zebra/ospfd.log
R1(OSPF)(config)# debug ospf packet hello send detail
```

Il faut que le répertoire `/var/log/zebra` existe.

6.2. Activation du processus de routage

Dans le mode `config`, nous allons activer le processus OSPF :

```
R1(OSPF)(config)# router ospf
R1(OSPF)(config-router)#
```

6.3. Activation des annonces de routes

Le processus de routage OSPF est activé mais rien ne se passe. Comme pour RIP, il faut indiquer sur quel(s) réseau(x) on souhaite que le routage dynamique soit opérationnel. Ceci se fait par la commande **network**. Mais, nouveauté par rapport à RIP qui n'intègre pas le concept de zone, il faut indiquer à quelle zone sera rattaché le réseau. Sur la [Figure 3, « La topologie de travail »](#), on voit que R1 est relié à deux réseaux. Le réseau 30.0.0.0/8 est attaché à la zone 0 et le réseau 11.0.0.0/8 à la zone 1. La configuration se fait donc de cette manière :

```
R1(OSPF)(config-router)# network 30.0.0.0/8 area 0
R1(OSPF)(config-router)# network 11.0.0.0/8 area 1
```

Que se passe-t-il sur le réseau ? R1 envoie des paquets HELLO sur les interfaces pour lesquelles la commande **network** a été saisie. Mais personne n'est là pour les écouter. Activez le routage sur R2 en adaptant les commandes aux spécificités du routeur. Je vous aide un peu. Sur R2, vous réaliserez les configurations suivantes :

```
R2(OSPF)(config-router)# network 11.0.0.0/8 area 1
R2(OSPF)(config-router)# network 12.0.0.0/8 area 1
```

Enfin, sur R3, vous réaliserez les configurations suivantes :

```
R3(OSPF)(config-router)# network 12.0.0.0/8 area 1
R3(OSPF)(config-router)# network 192.168.3.0/24 area 1
```

Mais sur R3, il y a une particularité. Le réseau 192.168.3.0/24 contient des ordinateurs mais aucun routeur. La commande **network** va diffuser sur ce réseau des annonces de routes ce qui consomme inutilement de la bande passante. Par conséquent, nous allons désactiver cette diffusion :

```
R3(OSPF)(config-router)# passive-interface eth1
```

Ainsi, aucune route n'est diffusée sur cette interface. De même, aucune annonce de route ne sera prise en compte. Le réseau sera considéré comme étant d'extrémité (*stub*).

6.4. Affichage de la configuration

Affichons la configuration complète de R1 :

```
R1(OSPF)(config-router)# end
R1(OSPF)# show running-config
```

```
Current configuration:
!
hostname R1(OSPF)
password foo
!
!
!
interface lo
!
interface eth0
!
interface eth1
!
router ospf
network 11.0.0.0/8 area 1
network 30.0.0.0/8 area 0
!
line vty
!
end
```

Affichons la configuration complète de R3 :

```
R3(OSPF)# show running-config
```

```
Current configuration:
!
hostname R3(OSPF)
password foo
!
!
!
interface lo
!
interface eth0
!
```

```

interface eth1
!
router ospf
passive-interface eth1
network 12.0.0.0/8 area 1
network 192.168.3.0/24 area 1
!
line vty
!
end

```

J'espère que vous avez la même configuration. Si ce n'est pas le cas, vous pouvez annuler une ligne contenant une erreur en vous remettant au même endroit où vous avez saisi la commande et en saisissant à nouveau la commande, mais en la faisant précéder de `no`.

Pour enregistrer la configuration, je vous rappelle que l'on saisit :

```
R1(OSPF)# copy running-config startup-config
```

Reproduisez maintenant ces manipulations sur l'ensemble des routeurs du réseau.

6.5. État des routeurs

Nos petits routeurs ont, en principe, bien travaillé. Dans chaque zone, ils ont élu leur chef, le routeur désigné (DR), ils ont échangé leurs connaissances et calculé une magnifique table de routage, ultra-optimale. En résumé, les deux stations d'extrémité de la figure doivent pouvoir s'atteindre avec une commande **ping**. Si jamais ce n'est pas le cas, c'est que probablement vous vous êtes trompé dans une configuration. Dans ce cas, reprenez la configuration de chaque appareil. Utilisez les outils **ping**, **tcpdump** et **traceroute** pour contrôler votre configuration et suivre les paquets. Et n'oubliez pas que dans un **ping**, il y a un aller mais aussi un retour !

Afin d'illustrer ce dont nous avons discuté dans la toute première partie de cet article, examinons l'état du routeur R1. Nous pouvons faire un diagnostic très complet de l'appareil en utilisant les nombreuses sous-commandes de **show ip ospf**. Vous constaterez que les informations fournies par `ospfd` sur son état sont beaucoup plus conséquentes que celles que l'on pouvait extirper de `ripd`.

Dans un premier temps, je vous propose d'examiner l'état de santé général du routeur R3 :

```

R3(OSPF)# show ip ospf
 OSPF Routing Process, Router ID: 192.168.3.254
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Refresh timer 10 secs
 Number of external LSA 0
 Number of areas attached to this router: 1

 Area ID: 0.0.0.1
  Shortcutting mode: Default, S-bit consensus: no
  Number of interfaces in this area: Total: 2, Active: 2
  Number of fully adjacent neighbors in this area: 1
  Area has no authentication
  Number of full virtual adjacencies going through this area: 0
  SPF algorithm executed 13 times
  Number of LSA 9

```

Le premier bloc décrit le fonctionnement général du routeur : l'ID du routeur (égale à sa plus grande adresse IP), conformité aux RFC, valeurs des temporisateurs. Une seule zone est attachée à ce routeur. C'est la zone 1 (exprimée en notation décimale pointée). Notre routeur a deux interfaces dans la zone, il n'a qu'un seul voisin. L'algorithme du SPF a été exécuté 13 fois. La base de données topologique contient neuf états de liens (LSA). Si notre routeur était attaché à plusieurs zones, le deuxième bloc serait répété autant de fois que de zones. Vous pourrez le constater sur R1.

Maintenant, listons nos informations sur les routeurs voisins :

```
R3(OSPF)# show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface	RXmtL	RqstL	DBsmL
12.0.0.1	1	Full/Backup	00:00:34	12.0.0.1	eth0	0	0	0

Déchiffrons ces informations. La différence entre la colonne **ID** et la colonne *Address*, c'est que l'**ID** identifie l'appareil dans le réseau alors que l'adresse correspond à l'interface à laquelle nous sommes relié avec ce routeur. La colonne *State* nous apprend deux choses : il est synchronisé avec le routeur désigné (DR) grâce à la mention *Full*, c'est le routeur désigné de secours (BDR) de la zone grâce à l'indicateur *Backup*. Ce routeur sera déclaré comme inactif si nous ne recevons pas de message HELLO d'ici 34 secondes (*Dead Time*).

Voyons le contenu de la base de données topologique de R3 :

```
R3(OSPF)# show ip ospf database
```

```
OSPF Router with ID (192.168.3.254)
Router Link States (Area 0.0.0.1)
```

Link ID	ADV Router	Age	Seq#	CkSum	Link count
12.0.0.1	12.0.0.1	981	0x80000006	0xf9e2	2
30.0.0.1	30.0.0.1	952	0x80000003	0xb13e	1
192.168.3.254	192.168.3.254	1063	0x80000005	0x15b7	2

```
Net Link States (Area 0.0.0.1)
```

Link ID	ADV Router	Age	Seq#	CkSum
11.0.0.2	12.0.0.1	981	0x80000001	0xda39
12.0.0.2	192.168.3.254	1063	0x80000001	0x5d0b

```
Summary Link States (Area 0.0.0.1)
```

Link ID	ADV Router	Age	Seq#	CkSum	Route
21.0.0.0	30.0.0.1	837	0x80000001	0x0d08	21.0.0.0/8
22.0.0.0	30.0.0.1	702	0x80000001	0x64a5	22.0.0.0/8
30.0.0.0	30.0.0.1	976	0x80000001	0x33e2	30.0.0.0/8
172.18.0.0	30.0.0.1	599	0x80000001	0x4a0d	172.18.0.0/24

Ces trois tableaux présentent de façon synthétique l'ensemble des LSA stockés dans la base topologique.

- Le premier tableau contient des LSA diffusés par chaque routeur. Ils décrivent l'état des interfaces de chaque routeur.
- Le deuxième tableau contient des LSA diffusés par le routeur désigné (DR). Ils décrivent la liste des routeurs présents dans chaque réseau.
- Le dernier tableau contient un résumé des routes diffusées par le routeur de bordure de zone (ABR). Ce sont des routes qu'il a reçu via le réseau fédérateur (*backbone*) par les routeurs des autres zones.

L'âge et le numéro de séquence sont utilisés pour mettre à jour la base lorsque des LSA sont reçus. La somme de contrôle (*checksum* noté CkSum est utilisée pour contrôler l'intégrité des LSA.

Pour obtenir des informations détaillées sur chaque LSA, vous pouvez compléter la commande **show ip ospf database** par les options suivantes : **router**, **network** ou **summary**. Par exemple : **show ip ospf database router 192.168.3.254** (qui correspond à la troisième ligne du premier tableau) vous apprendra que ce router est relié à deux réseaux : un de transit (12.0.0.0/8) et un d'extrémité (*stub*) 192.168.3.0/24.

Enfin, si vous voulez consulter la table de routage obtenue après traitement par SPF des différents LSA, vous n'aurez qu'à saisir un **show ip ospf route**.



Rappel important

Il existe une différence entre cette table et celle utilisée par le démon zebra pour le routage proprement dit. Souvenez-vous que Zebra et Quagga sont multi-protocole et qu'ils ont une architecture modulaire (voir

LM 43 : *2ème partie, routage RIP*²⁰). Chaque démon calcule une table de routage à partir des informations dont il dispose (et qui ne sont pas nécessairement les mêmes pour chaque démon). Ensuite, ils transmettent chacun leur table au démon zebra qui en fait la synthèse. Cette synthèse constitue la véritable table de routage utilisée pour router les paquets.

Nous avons fait un tour d'horizon des principales commandes de Zebra permettant de surveiller l'état de ospfd. Il y en a encore beaucoup d'autres que je vous laisse découvrir (faites un `show ip ospf ?` par exemple). Il nous reste à observer la table de routage obtenue par Zebra. Quittez ospfd et connectez-vous sur le démon zebra via la commande `telnet localhost 2601` :

```
R3(Zebra)> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
      B - BGP, > - selected route, * - FIB route

O>* 11.0.0.0/8 [110/20] via 12.0.0.1, eth0, 00:12:48
O   12.0.0.0/8 [110/10] is directly connected, eth0, 00:14:09
C>* 12.0.0.0/8 is directly connected, eth0
O>* 21.0.0.0/8 [110/40] via 12.0.0.1, eth0, 00:11:18
O>* 22.0.0.0/8 [110/50] via 12.0.0.1, eth0, 00:10:16
O>* 30.0.0.0/8 [110/30] via 12.0.0.1, eth0, 00:12:13
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.18.0.0/24 [110/60] via 12.0.0.1, eth0, 00:08:48
O   192.168.3.0/24 [110/10] is directly connected, eth1, 00:14:19
C>* 192.168.3.0/24 is directly connected, eth1
```

Les routes notées 'O' ont été découvertes par OSPF. Entre crochets, on observe la distance administrative du protocole (110 par défaut pour OSPF) et le coût de la route pour accéder au réseau. Dans le cas de la topologie étudiée, il n'y a que des réseaux à 10 Mbits/s, donc avec un coût par défaut de 10 pour chaque lien.

6.6. Quelques éléments sur la sécurité

6.6.1. Filtrer la diffusion des routes

Le premier inconvénient d'un protocole de routage dynamique comme OSPF est sa volubilité. Il a tendance à dévoiler tout un tas d'informations sur les réseaux qu'un administrateur consciencieux n'a pas forcément envie de révéler. Pour limiter la diffusion des routes au strict minimum, ospfd intègre, à l'instar de rripd, un mécanisme de liste de contrôle d'accès (*access-lists*).

Reportez-vous à l'article *2ème partie, routage RIP*²¹ publié dans le numéro 43 de *Linux Magazine*²². La configuration est strictement identique. J'en profite pour faire un peu de publicité : si vous êtes intéressés par les problèmes de sécurité, je vous conseille l'excellent magazine *Multi-system & Internet Security Cookbook (MISC)*²³. La série d'articles sur la «protection de l'infrastructure réseau IP» constitue sur certains points un approfondissement intéressant.

6.6.2. Protéger les annonces de routes

Le deuxième inconvénient d'un protocole de routage dynamique comme OSPF est sa naïveté. Il croit tout ce qu'on lui dit ! Un petit malin pourrait s'amuser à diffuser des routes farfelues à vos routeurs, ce qui pourrait provoquer des dénis de service. Pour pallier à cela, on peut activer l'authentification des annonces sur une zone. Voici les manipulations à réaliser sur chaque routeur :

```
Routeur(OSPF)(config-router)# area 1 authentication message-digest
```

Ensuite, pour chaque interface participant à la diffusion des routes :

```
Routeur(OSPF)(config)# int ethx
Routeur(OSPF)(config-if)# ospf message-digest-key 1 md5 motdepasse
```

²⁰ <http://www.linux-france.org/prj/inetdoc/guides/zebra.rip/>

²¹ <http://www.linux-france.org/prj/inetdoc/guides/zebra.rip/>

²² <http://www.linuxmag-france.org/>

²³ <http://www.miscmag.com/>

Vous adapterez le «motdepasse» à vos besoins. Ce mot de passe doit bien sûr être connu de tous les routeurs.

7. Conclusion

OSPF est un protocole de routage dynamique moderne, robuste et conçu pour les grands réseaux. On constate qu'il est nettement plus complexe que RIP. Pas forcément dans sa configuration mais dans son fonctionnement interne. Un inconvénient de ce protocole est qu'il peut être gourmand en puissance de calcul et en mémoire lorsque le réseau comporte beaucoup de routes ou qu'il y a de fréquentes modifications de topologie.

OSPF est un protocole IGP (*Interior Gateway Protocol*), c'est-à-dire qu'il agit au sein d'un système autonome. Un AS (*Autonomous System*) est un ensemble de réseaux gérés par un administrateur commun. Chaque système autonome possède un numéro identifiant sur 16 bits délivré par l'IANA (*Internet Assigned Numbers Authority*) ou ses délégations. Classiquement, les multinationales, les opérateurs de télécom ou les fournisseurs d'accès à Internet détiennent un numéro de système autonome.

Pour assurer le routage entre les systèmes autonomes, un protocole de type EGP (*Exterior Gateway Protocol*) doit être mis en oeuvre. Dans le cas de l'Internet, c'est généralement BGP (*Border Gateway Protocol*) qui assume cette mission. BGP, protocole supporté par Zebra, constitue un vaste terrain d'investigation.

Pour terminer, je voudrais saluer tous mes étudiants de BTS Informatique de gestion à Loudun qui m'ont aidé dans la réalisation de ce document.

7.1. Bibliographie

TCP/IP : Architecture, protocoles et applications

Douglas COMER, DUNOD. ISBN: 2-10-005384-1 (09/2001) 830 p.

Le routage dans l'Internet

Christian HUITEMA, EYROLLES. ISBN: 2-212-08902-3 (10/1994) 418 p.

7.2. Liens

- [Zebra](#)²⁴
- [Quagga](#)²⁵
- [Linux Magazine](#)²⁶
- Version originale du document et page personnelle de [Pacôme Massol](#)²⁷

²⁴ <http://www.zebra.org/>

²⁵ <http://www.quagga.net/>

²⁶ <http://www.linuxmag-france.org/>

²⁷ <http://www.pmassol.net/>