Routage dynamique avec BGP

Stéphane Bortzmeyer
AFNIC

Immeuble International
78181
Saint-Quentin-en-Yvelines
France
bortzmeyer@nic.fr

Gitoyen

Une grande partie de ce cours a pu être faite grâce à l'expérience acquise en concevant et en déployant le réseau de l'opérateur Internet Gitoyen.

$Id: bgp-partial.db,v 1.2 2008/02/12 11:26:38 bortzmeyer Exp $

Ce document explique la conception et la configuration d'un réseau TCP/IP d'opérateur, interconnecté avec d'autres opérateurs et utilisant le routage dynamique, en l'occurrence avec le protocole BGP.

Il se veut pratique : l'accent est mis sur une configuration simple qui devrait couvrir la plupart des cas. Il ne remplace donc pas un cours complet sur BGP.

Table des matières

1. Introduction 1
2. Premier réseau et première configuration 8
3. Configurations plus riches 16
4. Déboguage 22
Bibliographie 28
RFC 28

1. Introduction

1.1. Dois-je utiliser BGP ?

BGP est nécessaire si :

- Vous voulez vous connecter à plusieurs fournisseurs de connectivité de manière propre[1].
- Vous voulez vous connecter à un point d'échange entre opérateurs. De tels points d'échange sont un outil essentiel pour la connectivité Internet d'un pays : ils permettent aux opérateurs d'un pays d'échanger du trafic directement, sans passer par les États-Unis ou bien l'Europe.

1.2. Petit rappel IP

> **Important :** Ce texte n'est qu'un rappel, ce cours nécessite une bonne connaissance d'IP.

Le protocole IP (que cela soit IPv4 ou bien IPv6) repose sur la notion d'adresse. Une machine a une ou plusieurs adresses qui s'écrivent, en IPv4, avec quatre chiffres séparés par des points, comme 80.67.160.1[2] et en IPv6 par plusieurs groupes de chiffres séparés par des deux-points, comme 2001:910::5043:a001[3].

Dans une adresse, les chiffres binaires les plus à gauche identifient le réseau (d'où le nom de préfixe), les chiffres les plus à droite la machine. Le nombre de chiffres pour chaque rôle dépend de la longueur du préfixe. Celle-ci s'indique par un chiffre suivant une barre oblique. Par exemple, 80.67.160.1/27 indique que l'adresse IP en question identifie une machine dans un réseau où les 27 premiers bits désignent le réseau et les 5 derniers la machine (les adresses IPv4 comptent 32 bits). De la même façon, fec0:1:0:ff00::1/64 est une adresse IPv6 sur un réseau où il y a 64 bits pour le réseau et 64 pour la machine (les adresses IPv6 comptent 128 bits).

Il est important de noter que la longueur du préfixe dépend de l'endroit où on se trouve. Du fait de l'agrégation des préfixes, le réseau 80.67.160.0/27 se trouve par exemple annoncé sur Internet comme une partie de 80.67.160.0/19.

Il existe des outils pour faciliter les calculs sur les adresses. En IPv4, netmask permet, par exemple :

```
% netmask 80.67.168.6/19❶
    80.67.160.0/19
% netmask -s 80.67.168.6/27❷
    80.67.168.0/255.255.255.224
% netmask -r 80.67.168.6/27 ❸
    80.67.168.0-80.67.168.31    (32)
```

❶ Quelle est l'adresse du réseau correspondant à cette adresse et cette longueur de préfixe ?

❷ IPv4 utilise encore beaucoup une vieille notation : au lieu de la longueur du préfixe, on affiche un masque où tous les bits "réseau" valent 1 et tous les bits "machine" valent 0.

❸ Quelles sont les adresses IP dans ce réseau ?

1. BGP nécessite certaines ressources virtuelles, mais rares comme des adresses IP et ne peut donc pas être à votre portée, même si vous souhaitez vous connecter à plusieurs opérateurs. Vous serez alors amenés à utiliser des bricolages plus ou moins propres (par exemple le *source routing*).
2. Plus de détails sur l'adressage IPv4 dans RFC 791.
3. Plus de détails sur l'adressage IPv6 dans RFC 2373.

En IPv6, il n'existe malheureusement pas d'outils équivalents mais ipv6calc permet certaines choses.

1.3. Petit rappel routage

Pour transmettre un paquet, une machine IP suit sa table de routage. On peut l'afficher sur quasiment tous les Unix avec **netstat -rn**, qui est la commande la plus portable, ou bien avec **route -n** sur Linux, **route -n show** sur NetBSD, **show ip route** sur IOS, etc. La table de routage est une série d'entrées, et elle fait correspondre à un préfixe, l'adresse IP du routeur où envoyer le paquet. Par exemple, si un routeur a la table de routage suivante :

```
147.94.0.4      194.68.129.102  255.255.255.252 UG    0      0        0 eth1
213.223.128.0   194.68.129.244  255.255.255.248 UG    0      0        0 eth1
192.54.202.64❶   194.68.129.102  255.255.255.240 UG    0      0        0 eth1
194.6.149.64    194.68.129.244  255.255.255.224 UG    0      0        0 eth1
194.6.145.0     194.68.129.244  255.255.255.224 UG    0      0        0 eth1
213.56.168.160  194.68.129.224  255.255.255.224 UG    0      0        0 eth1
195.141.72.128  194.68.129.213  255.255.255.224 UG    50     0        0 eth1
```

❶ Les paquets à destination de 192.54.202.66 (**netmask 192.54.202.66/255.255.255.240** donne 192.54.202.64) seront envoyés à 194.68.129.102. traceroute permet de vérifier cela :

```
traceroute to 192.54.202.66 (192.54.202.66), 30 hops max, 40 byte packets
 1  194.68.129.102  1 ms  0 ms  0 ms
 2  193.51.179.158  1 ms  1 ms  1 ms
...
```

Il y a une légère différence en IPv6 : il existe des adresses spécifiques à un lien (*link local*), leur préfixe est fe80::/10 et ce sont ces adresses qui sont utilisées dans la table de routage.

```
rachel:~ % netstat -r -n -finet6
...
Destination                       Gateway                       Flags     Refs     Use    Mtu  Int
default                           fe80::201:96ff:fe96:dc60%epic0 UG          0       0      -  epi
```

> **Important :** Pour une adresse IP de destination, il peut exister plusieurs entrées dans la table de routage. Si les longueurs de préfixe sont différentes, le routeur choisira la plus spécifique (le préfixe le plus long). Si les longueurs de préfixe sont égales, le résultat dépend du routeur. Avec Linux, si le noyau a été compilé avec l'option *IP: equal cost multipath (CONFIG_IP_ROUTE_MULTIPATH)*[4], les deux chemins pourront être utilisés, répartissant ainsi la charge.

Les tables de routage peuvent être construites manuellement (l'ingénieur système ajoute des commandes dans un fichier de configuration) ou automatiquement par un système de routage dynamique. Dans ce dernier cas, les routeurs échangent de l'information entre eux ("Je suis connecté à un lien qui porte les préfixes 2001:910::/32 et FEC0:1::/32", "J'ai une liaison point-à-point avec 82.3.67.98") et, sur la base de ces informations, construisent chacun sa table de routage.

En anglais, on appelle le routage effectif des paquets *forwarding* et la construction des tables de routage *routing*. Tout routeur IP fait du *forwarding* mais le routage pouvant être statique, tous ne font pas du *routing*. Ces deux fonctions sont typiquement mises en oeuvre par des parties très distinctes du routeur[5].

4. qui nécessite l'option *IP: advanced router*
5. Sur Unix, le noyau effectue le *forwarding*, alors que le *routing* est confié à un programme extérieur comme Quagga.

1.4. Et si ça ne marche pas ?

Si votre configuration TCP/IP ne fonctionne pas, il est utile de demander de l'aide. D'innombrables listes de diffusion[6] existent à cette fin. Mais beaucoup de personnes ont du mal à donner sur une liste de diffusion les informations nécessaires pour érsoudre le problème.

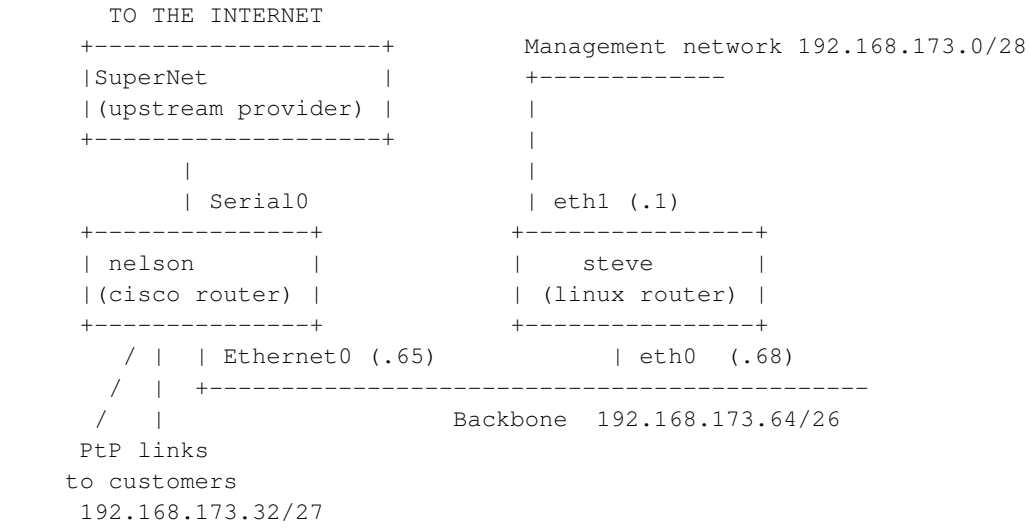
Pour toute liste, les conseils suivants sont utiles :

- Donnez les vraies informations. Il n'existe aucune raison sérieuse[7]de dissimuler les noms et les adresses IP utilisées. Au contraire, si vous indiquez les vraies adresses, les lecteurs pourront les essayer plus facilement.

  Sauf si vous soumettez un problème purement théorique, auquel cas vous devez utiliser des adresses RFC 1918, donnez les vraies adresses.

- Donnez le maximum de détails sur votre environnement : système d'exploitation utilisé, version, type de réseau (Ethernet ? FDDI ?). N'oubliez pas que vos lecteurs ne sont pas dans votre tête : pour vous, il est évident que votre machine 192.168.7.34 est un routeur Extreme mais les autres ne le savent pas.

- Faites des schémas. Un bon croquis vaut souvent mieux que bien des discours. Comme il n'existe pas de norme largement répandue pour transmettre des dessins[8], le mieux est d'utiliser l'*ASCII art*. Avec le mode picture (http://www.cs.cmu.edu/cgi-bin/info2www?(emacs)Picture) de l'éditeur Emacs, c'est assez facile. En voici un exemple, avec indication des noms des routeurs (ce qui facilite beaucoup les discussions ultérieures) et des interfaces :



- Soyez factuel : donnez les commandes *exactes* que vous avez utilisées, les résultats *exacts* de ces commandes. Faites du copier/coller, pas de la traduction ou du résumé.

6. Par exemple, en France, la liste IP (http://www.services.cnrs.fr/wws/info/ip) ou bien en Afrique la liste d'AFNOG (http://www.afnog.org/meeting3.html#mailinglist), destinée aux opérateurs mais qui voit en pratique beaucoup de questions de base sur IP.
7. Et surtout pas de raisons de sécurité.
8. Je fonde beaucoup d'espoirs sur SVG (http://www.w3.org/Graphics/SVG/Overview.html) mais qui est très peu répandu encore.

1.5. Introduction à BGP

BGP, (*Border Gateway Protocol*) est le protocole standard de l'Internet pour les interconnexions entre opérateurs[9]. Il fait partie de la famille des EGP (*Exterior Gateway Protocol*) dont il est aujourd'hui le seul membre.

BGP ne sert donc que si vous êtes vous-même opérateur, c'est-à-dire si vous avez votre propre politique de routage à l'extérieur de votre AS (*Autonomous System*). La connaissance de BGP est nécessaire si vous travaillez chez un opérateur ou bien si vous gérez un point d'échange entre opérateurs. Elle peut aussi aider à mieux comprendre le routage dans l'Internet mais ce cours se concentre sur l'aspect pratique, plus que sur une compréhension en profondeur.

Qu'est-ce qui différencie les protocoles EGP, comme BGP, des protocoles utilisés à l'intérieur des AS, les IGP (*Interior Gateway Protocol*) comme OSPF ? La différence essentielle n'est pas technique. Elle est administrative.

On n'utilise les IGP qu'à l'intérieur d'une entité (entreprise, association, etc), où des décisions (comme la suppression ou bien l'ajout d'une ligne) peuvent être prises par un service unique. Le but des IGP est donc de trouver la route la plus efficace, en faisant confiance aux autres routeurs.

Au contraire, les EGP comme BGP s'utilisent entre entités distinctes (et même souvent concurrentes). Il n'y a plus de possibilité de prendre une décision qui s'imposera à tous. On n'est souvent même pas prévenu de ce que vont faire les pairs avec lequels on parle BGP. En conséquence de quoi, les EGP reposent sur l'idée de méfiance : le but n'est pas de trouver la meilleure route mais au contraire d'empêcher les routeurs de choisir une route dont on ne voudrait pas.

Cette différence a de nombreuses conséquences pratiques. Par exemple, il n'y a aucun moyen en OSPF de dire que l'on veut envoyer des informations à tel routeur mais pas à tel autre. OSPF nécessite que tous les routeurs d'une zone aient la même base. Au contraire, cette capacité à filtrer l'information envoyée est une des fonctions les plus utilisées de BGP.

La définition d'un AS (*Autonomous System*) est donc administrative. Un AS, l'unité de routage pour BGP, est une entité où le pouvoir de décision est centralisé. Elle communique avec d'autres entités, ayant leurs propres AS et qui n'ont pas de relations hiérarchiques. Les autres entités sont parfois vos fournisseurs de connectivité, parfois vos clients, parfois des pairs mais, dans tous les cas, vous ne pouvez pas leur dicter leur politique de routage.

Avec les autres opérateurs, vous allez pouvoir échanger du trafic. Cela peut se faire sur plusieurs bases contractuelles différentes. On distingue en général le transit, où vous payez un fournisseur pour vous annoncer des routes (et pour acheminer ensuite votre trafic) du *peering* où la relation se fait entre pairs, entre (presque) égaux, et où il n'y a pas en général d'échange d'argent.

Outre votre AS, le monde est rempli d'autres opérateurs. Ils ne sont pas égaux entre eux. Le classement se fait en fonction de l'achat de transit qu'ils effectuent. Si un opérateur est suffisamment grand et a une envergure internationale, il n'achète plus du tout de transit : on parle de *Tier-1*. Tous les autres opérateurs, tout en ayant des clients et en échangeant avec leurs pairs, sont obligés d'acheter du transit pour combler les limites de leur réseau. Par exemple, un opérateur purement européen va devoir acheter du transit à un *Tier-1* pour permettre à ses clients d'accèder au Japon ou au Brésil.

1.6. Quel matériel et logiciel choisir ?

BGP tourne aujourd'hui sur les routeurs de haut de gamme[10] ainsi que sur toute machine Unix.

L'inclinaison personnelle de l'auteur le porte vers les logiciels libres (http://www.april.org/) donc la majorité des exemples concerneront le couple Unix+Zebra (http://www.zebra.org/) avec lequel l'auteur

9. En toute rigueur, il faut signaler que certains gros clients, connectés à plusieurs opérateurs, utilisent également BGP.
10. Sur les routeurs commerciaux, BGP peut ne pas faire partie de la licence de base et peut nécessiter l'achat d'une licence spéciale.

a le plus d'expérience. (Zebra a désormais un successeur, Quagga, Zebra ne semblant plus maintenu.) Ce couple permet de faire du BGP avec un simple PC, et un Unix libre comme Debian (http://www.debian.org/) ou bien FreeBSD (http://www.freebsd.org/).

Le système le plus utilisé pour les routeurs BGP est sans doute IOS (Internetwork Operating System) (http://www.cisco.com/en/US/products/sw/iosswrel/index.html) de Cisco. Les commandes sont très proches de celles de Zebra.

Les routeurs de Juniper (http://www.juniper.net/), plus orientés vers le haut de gamme, ont un langage de commandes très différent.

Parmi les autres marques de routeurs BGP qui mettent beaucoup de documentations en ligne, citons Riverstone (http://www.riverstonenet.com/support/bgp/).

1.7. Petit rappel sur l'utilisation des routeurs IP

> **Important :** Ce texte n'est qu'un rappel sommaire, ce cours nécessite une bonne connaissance des environnements utilisés. Par exemple, si votre routeur est une machine Unix, il faut connaitre le shell, le système des journaux, etc. Par exemple, lorsque j'écris "envoyez le signal HUP au démon", je considère acquis que vous savez envoyer un signal à un démon[11].

1.7.1. Quagga

Quagga (http://www.quagga.net/) est issu du programme Zebra, qui ne semble plus maintenu mais que vous trouverez encore fréquemment en production.

Quagga est composé de plusieurs démons qui acceptent chacun des connexions TCP. Vous pouvez donc utiliser telnet pour vous connecter à chaque démon et effectuer sa configuration[12].

Si cette connexion aux démons est très pratique pour déboguer ou surveiller un routeur, je ne la trouve pas efficace pour écrire ou modifier la configuration du routeur. Je recommande plutôt de modifier les fichiers de configuration avec un éditeur, ce qui permet d'utiliser un bon outil d'édition, de gérer les configurations avec CVS, de préserver les commentaires, etc.

L'emplacement des fichiers de configuration de Quagga dépend de votre Unix et de la manière dont Quagga a été compilé[13]. J'utiliserai les emplacements du système Debian. Tous les fichiers sont alors dans `/etc/zebra`. On trouve notamment :

- `zebra.conf`, la configuration générale[14],
- `ospfd.conf`, la configuration du protocole OSPF,
- `ospf6d.conf`, la configuration du protocole OSPFv6,
- `bgpd.conf`, la configuration du protocole BGP.

11. **man kill**
12. Il existe aussi un système, le vtysh pour se connecter de manière unifiée à tous les démons mais il n'est pas décrit ici.
13. Rappelez-vous que Quagga se nommait Zebra précédemment et beaucoup de fichiers n'ont pas encore changé de nom.
14. Ce fichier est vide la plupart du temps. Il contient des définitions d'interfaces et de routes statiques mais, sur Unix, on les fait typiquement en dehors de Quagga. Le noyau Unix informera ensuite Quagga. On ne remplit ce fichier que pour contourner des bogues du noyau (par exemple avec les adresses *multicast* sur NetBSD).

Normalement, sur Unix, une fois qu'on a modifié un fichier de configuration, on envoit le signal HUP au démon[15]. Dans certaines versions de Quagga, avec certains protocoles (notamment BGP), ce signal a des effets négatifs (comme de couper toutes les sessions BGP) ou nuls. Il est donc recommandé d'utiliser la démarche suivante pour changer la configuration.

1. Tapez les commandes dans la fenêtre de la console du démon.
2. Si elles sont acceptées et produisent bien le résultat attendu, copiez les commandes dans le fichier de configuration, avec les commentaires appropriés.

Une variante, surtout si on connait bien les commandes, est de les mettre dans le fichier d'abord, puis de copier/coller les nouvelles commandes dans la console.

Venons en maintenant à la console d'administration du démon. On l'atteint en faisant un **telnet *routeur démon***[16]. Le nom du démon (ou, plus exactement, le port sur lequel il écoute) est zebra, ospfd, ospf6d ou bien bgp[17].

Une fois connecté à la console du démon, vous avez une invite (configurable) et vous pouvez taper des commandes :

```
monrouteur> show ip ospf route
```

Il n'est pas nécessaire de connaitre toutes ces commandes par coeur. La touche ? vous fera apparaitre une aide contextuelle[18]. Par exemple, si vous avez déjà tapé **show ip ospf** et que vous tapez un ?, vous avez la liste :

```
  border-routers  for this area
  database        Database summary
  interface       Interface information
  neighbor        Neighbor list
  route           OSPF routing table
  <cr>
```

des commandes valides à ce stade (<cr> signifie *Carriage Return* et désigne la touche **Entrée**).

Vous pouvez également compléter une commande en cours de saisie avec la touche **Tabulation**.

Ce mécanisme de console en ligne de commande se prête bien à l'automatisation. Par exemple, le programme Python (http://www.python.org/) suivant se connecte à un routeur pour afficher la liste des routes[19] :

```
from telnetlib import Telnet
tn = Telnet(router, 2601) # 2601 is the port of the Quagga console

def write_after_prompt (prompt, text):
    (index, match, read) = tn.expect([prompt], 4)
    if not match:
            raise "Text \"" + prompt + "\" not found"
    # Eat it
    tn.read_until(prompt, 0)
```

15. Debian facilite le choses en créant pour chaque démon un script `/etc/init.d/`*démon* qui accepte toujours les mêmes options, comme **reload** pour recharger une configuration, ce qui permet de gérer de manière uniforme tous les démons, même ceux qui ne suivent pas parfaitement la convention. D'autres systèmes d'exploitation ont un mécanisme analogue, par exemple FreeBSD dans ses toutes dernières versions (5).
16. telnet ne chiffrant pas les communications, il est très fortement recommandé de taper cette commande depuis le routeur lui-même, pour limiter les risques d'écoute.
17. Si aucun de ces noms ne marche, par exemple si vous obtenez `tcp/zebra: unknown service`, c'est que ces noms n'ont pas été mis dans `/etc/services`. Là encore, il est recommandé d'utiliser un système de paquetages qui vous épargne ce genre d'oublis.
18. C'est-à-dire que les commandes ou options affichées sont celles qui sont réellement disponibles à ce stade. C'est d'autant plus utile que les commandes ne sont pas les mêmes pour chaque démon.
19. SNMP permet souvent la même chose.

```
    tn.write(text + '\r\n')

write_after_prompt ("Password:", password)
write_after_prompt (prompt, 'terminal length 0')
write_after_prompt (prompt, 'show ip route')
```

Certaines commandes nécessitent plus de privilège, puisqu'elles modifient l'état du routeur. La commande **enable** permet de passer dans cet état privilégié. Par exemple, pour changer la configuration du routeur ou bien pour réinitialiser une session BGP, vous devrez être en mode **enable**.

1.7.2. IOS

> **Important :** IOS est un logiciel commercial et il est donc recommandé de s'adresser au vendeur pour toute question.
>
> Je suis bien conscient que le vendeur n'est souvent pas très efficace pour le support mais les partisans du logiciel commercial expliquent toujours que l'avantage de ce dernier est la qualité du support : c'est donc l'occasion de tester.

Le langage de commandes d'IOS a été largement imité et ne surprendra donc pas les utilisateurs de Quagga. Comme pour Unix, je recommande d'éditer les fichiers de configuration sur une machine qui puisse les gérer avec un outil comme CVS. Si on gère plusieurs routeurs, il est recommandé d'utiliser un outil comme COSI (http://cosi-nms.sourceforge.net/) ou Rancid (http://www.shrubbery.net/rancid/). Une fois le fichier édité, on peut le charger sur le Cisco avec **configure network**[20].

1.7.3. SNMP

SNMP (*Simple Network Management Protocol*) est un protocole de gestion de réseau. C'est une norme (RFC 1157 et RFC 2578) donc il fonctionnera quel que soit le type de vos routeurs. Vous trouverez énormément d'informations sur SNMP sur le Simple Web (http://www.simpleweb.org/).

SNMP permet à un gérant (*manager*) d'interroger un agent (qui se trouve, dans notre cas, sur un routeur ou un commutateur) et d'obtenir des informations. Par exemple, ici, on utilise le gérant Net-SNMP sur Unix pour demander à un routeur combien il a d'interfaces[21], puis combien d'octets sont passés sur la première interface depuis le démarrage du routeur :

```
% snmpget 192.134.7.246 password interfaces.ifNumber.0
interfaces.ifNumber.0 = 5

% snmpget 192.134.7.246 password interfaces.ifTable.ifEntry.ifInOctets.1
interfaces.ifTable.ifEntry.ifInOctets.1 = Counter32: 34254759
```

20. Il faudra avoir configuré un serveur TFTP avant.
21. Cette information étant un scalaire, il faut terminer l'identificateur de l'objet SNMP par un 0. La seconde information est un vecteur, on termine l'identificateur par l'index.

2. Premier réseau et première configuration

Nous allons commencer par un cas tellement simple qu'il ne justifie pas réellement d'utiliser BGP. Nous avons un AS d'un petit opérateur et celui d'un fournisseur de connectivité plus gros. Si vous êtes dans la sitation du petit opérateur, vous ne pouvez pas commencer à faire du BGP comme cela : il vous faudra un numéro d'AS et des adresses IP[22]

2.1. Un peu de bureaucratie

Contrairement aux IGP (qui ne regardent que vous), les EGP comme BGP nécessitent une coordination centralisée : vos routes vont en effet apparaitre dans un espace commun, la DFZ (*Default-Free Zone*), l'ensemble des routeurs qui n'ont pas de route par défaut. Les ressources étant rares dans la DFZ (surtout en IPv4), une attribution coordonnée est nécessaire, avec son inévitable (?) paperasserie.

Les numéros d'AS sont des entiers stockés sur 16 bits. Il ne peut donc y en avoir que 65535 au niveau mondial, ce qui est très peu. Genuity a obtenu le 1, Gitoyen a eu, beaucoup plus récemment, le 20766 (http://www.ripe.net/perl/whois?form_type=advanced&full_query_string=&searchtext=AS20766&alt_database=RIPE
num&Simple+search=Simple+search).

Les adresses IP, quant à elles, sont encore plus rares, en IPv4 (en IPv6, elles s'obtiennent au contraire très facilement et c'est une puissante motivation pour passer à IPv6).

Ces deux types de ressources s'obtiennent auprès des RIR (*Regional Address Registry*) et il en existe actuellement quatre : ARIN (http://www.arin.net/) en Amérique du Nord, LACNIC (http://www.lacnic.net/) en Amérique du Sud, APNIC (http://www.apnic.net/) en Asie-Pacifique et RIPE-NCC (http://www.ripe.net/) en Europe[23].

Je parlerai surtout du RIPE-NCC (Réseaux IP Européens - *Network Coordination Center*)[24] car c'est celui que je connais le mieux.

Le RIPE-NCC est un registre : il attribue des ressources (adresses IP - v4 et v6 - et numéros d'AS, notamment) et garde trace de ces allocations. Il gère une base de données de ces ressources, accessible publiquement par le protocole whois (RFC 0954). Par exemple, pour connaitre le "propriétaire"[25] de l'AS 20766 : **whois -h whois.ripe.net AS20766**

```
aut-num:      AS20766❶
as-name:      GITOYEN-MAIN-AS
descr:        The main Autonomous System of Gitoyen (Paris, France).
admin-c:      SB4267-RIPE❷
tech-c:       GI1036-RIPE
remarks:      Looking Glass: http://lookinglass.gitoyen.net/❸
import:       from AS12876❹
        action pref=100;
        accept AS-TISCALIFR
export:       to AS12876
        announce AS-GITOYEN
...
```

❶ Le serveur whois présente les données sous un format attribut:valeur. Ici, l'attribut `aut-num` vaut `AS20766`. D'autres protocoles, comme celui en cours d'étude au sein du groupe de travail Crisp

22. Nous verrons plus tard comment utiliser des ressources privées si vous n'arrivez pas à obtenir de telles ressources publiques - un problème fréquent en Afrique, par exemple.
23. Il y a eu des discussions pour la création d'un RIR en Afrique, Afrinic (http://www.afrinic.org/), mais qui n'ont pas encore débouché sur quelque chose de concret.
24. Qu'on appelle souvent RIPE tout court par abus de langage : normalement, le RIPE est une association des opérateurs, le RIPE-NCC étant le RIR.
25. Les RIR précisent bien que les adresses ne sont que prêtées : vous n'en êtes pas le propriétaire.

(http://www.ietf.org/html.charters/crisp-charter.html), de l'IETF, présenteront ces mêmes données de manière différente.

❷ La base contient des informations sociales sur les personnes ou organismes à contacter en cas de problèmes. Ces informations sont accessibles via un *handle*, un identificateur, ici le mien.

❸ La base contient aussi des commentaires en texte libre.

❹ L'objet contient des informations sur la politique de routage de l'AS. Elles sont exprimées en langage RPSL (RFC 2622 et RFC 2650). Ici, on apprend que Gitoyen échange des routes avec l'AS 12876, Tiscali.

Si on veut savoir de qui dépend l'adresse IPv4 192.134.7.250 , on fait un : **whois -h whois.ripe.net 192.134.7.250**

```
inetnum:      192.134.0.0 - 192.134.7.255
netname:      NIC-FR-BLOC
descr:        AFNIC
descr:        c/o INRIA
descr:        Domaine de Voluceau, Rocquencourt❶
descr:        BP 105, 78153 Le Chesnay CEDEX, France
admin-c:      NFC1-RIPE
tech-c:       NFC1-RIPE
...
```

❶ Cette adresse postale n'est plus valable depuis longtemps mais cela illustre bien la méfiance avec laquelle il faut lire le contenu de ces bases.

On voit que cette adresse a été allouée à l'AFNIC. Essayons-en une autre :

```
whois -h whois.ripe.net 81.6.7.35
inetnum:      81.6.7.0 - 81.6.7.63
netname:      HUGIT-G-CH-NET
descr:        Hug-IT, Aadorf, Switzerland
country:      CH
admin-c:      MH2254-RIPE
tech-c:       gnoc3-ripe
...
```

Ici, l'objet RIPE reflète une affectation à un client. Le fournisseur d'accès est (notez l'option -L[26] pour avoir les réseaux moins spécifiques) :

```
whois -h whois.ripe.net -L 81.6.7.35
inetnum:      81.6.0.0 - 81.6.63.255
netname:      CH-GREEN-20020613
descr:        PROVIDER LOCAL REGISTRY
descr:        Green Connection AG
country:      CH
admin-c:      GH36-RIPE
tech-c:       OB22-RIPE
...
```

Pour obtenir des ressources analogues, adresses IP ou bien numéro d'AS, il faut s'adresser à un LIR (*Local Internet Registry*), c'est-à-dire un membre de RIPE qui a autorité pour demander (en les justifiant !) des ressources et qui gère les objets le concernant dans la base du RIPE-NCC[27]. Bien sûr, vous pouvez devenir LIR vous même (http://www.ripe.net/ripencc/new-mem/) mais c'est coûteux et cela nécessite une équipe rompue aux interactions avec une bureaucratie internationale

26. L'ensemble des options est documentée au RIPE (http://www.ripe.net/ripe/docs/databaseref-manual.html#2.0).
27. Les LIR assurent cette tâche avec un sérieux très variable et c'est pour cela que la base est souvent erronnée.

(http://www.ripe.net/ripe/docs/internet-registries.html). Souvent, vous ferez donc appel à un LIR (la liste figure sur le site Web du RIPE-NCC (http://www.ripe.net/ripencc/mem-services/general/indices/index.html)) qui demandera pour vous. Sans doute vous laissera t-il remplir une partie des formulaires de demande comme celui qui suit, notamment la section où vous justifiez votre demande d'adresses.

```
Reply-To: noc@gitoyen.net❶
From: Stephane Bortzmeyer <bortzmeyer@gitoyen.net>
To: hostmaster@ripe.net
Cc: noc@gitoyen.net
Subject: NEW address assignment (and allocation) needed
X-NCC-RegID: fr.gitoyen


#[ OVERVIEW OF ORGANISATION TEMPLATE ]#

Gitoyen is made of five Internet providers (Netaktiv, Gandi, Globenet,
Placenet and FDN). The legal status is a French GIE (Groupement
d'Intérêt Economique). All of the members are currently connected by
other providers but intend to use only Gitoyen in the near future. We
will need IP addresses for the infrastructure of Gitoyen and for its
members, which will in turn assign addresses to some customers. This
request of assignment is for Gitoyen only.

http://www.gitoyen.net/ (in French only)

#[ REQUESTER TEMPLATE ]#

name: Stéphane Bortzmeyer
organisation: Netaktiv
country: FR
phone:  +33 140137920
e-mail: noc@gitoyen.net

#[ USER TEMPLATE ]#

name: Stéphane Bortzmeyer
organisation: Gitoyen
country: FR
phone:  +33 140137920
e-mail: noc@gitoyen.net

#[ CURRENT ADDRESS SPACE USAGE TEMPLATE ]#❷


                                    Addresses Used
Prefix          Subnet Mask      Size  Current 1-yr  2-yr  Description

                                  0          0    0     0   Totals

II. The Request❸

#[ REQUEST OVERVIEW TEMPLATE ]#

request-size:            64
addresses-immediate:     24
addresses-year-1:        48
addresses-year-2:        64
subnets-immediate:       2
subnets-year-1:          2
```

```
subnets-year-2:          2
inet-connect:            Gitoyen will connect in a few days, at the POPs Sfinx and Telehouse2 in Par
country-net:             FR
private-considered:      Yes
request-refused:         No
PI-requested:            Yes
address-space-returned:  No

#[ ADDRESSING PLAN TEMPLATE ]#

                                    Addresses Used
Prefix          Subnet Mask      Size  Immediate 1-yr  2-yr  Description

0.0.0.0         255.255.255.224   32    12      24    32   Gitoyen main presence points at Sfinx
0.0.0.0         255.255.255.224   32    12      24    32   Gitoyen main presence points at Telehouse

                                  64    24      48    64 Totals

III. Database Information

#[ NETWORK TEMPLATE ]#

inetnum:
netname:  Gitoyen-main
descr: Gitoyen main presence points in Paris
country:  FR
admin-c:  SB4267-RIPE
tech-c:   GI1036-RIPE
status:   ASSIGNED PI
notify:   noc@gitoyen.net


#[TEMPLATES END]#

IV. Optional Information

A request for an AS is pending: NCC#2001052545. We will use BGP and
connect to 3 upstream providers.

We do not had any allocation yet, so it is a request both for an
allocation and an assignment.

We expect to have an AS IPV6 during 2002 and migrate to ipv6 end of
2003.

This request is only for Gitoyen central infrastructure. Requests for
IP addresses for members will follow. (Members already have IP
addresses, unlike Gitoyen, an will return them.)
```

❶ L'essentiel de l'interaction avec le RIPE-NCC se fait par courrier électronique.

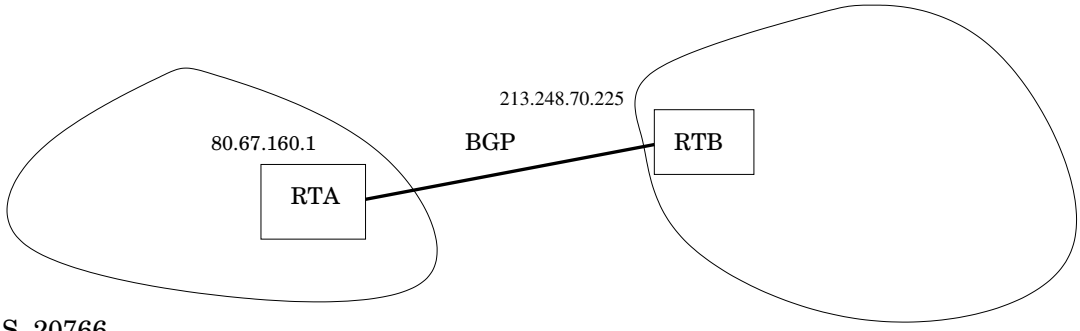
❷ L'infrastructure était entièrement nouvelle. Mais si vous avez déjà un réseau, vous devrez le documenter ici.

❸ Le gros morceau : il est recommandé de bien étudier le plan d'adressage (et de le rédiger) avant de le traduire en formulaire RIPE. Cela vous aidera également pour les discussions en interne ou bien si vous voulez solliciter de l'aide extérieure.

Une fois que vous aurez votre numéro d'AS et vos adresses IP, vous pourrez préparer votre configuration.

2.2. Configuration de BGP

Figure 1. Un seul fournisseur



Votre AS, 20766

L'AS d'un fournisseur, 1299

Elle peut être aussi simple que (sur un Unix avec Quagga) :

```
hostname myrouter
router bgp 20766
  bgp router-id 80.67.160.1
  network 80.67.160.0/19
  neighbor 213.248.70.225 remote-as 1299
log file /var/log/zebra/bgpd.log
```

Ici, on déclare son propre numéro d'AS, on donne au routeur un *router ID*[28], on indique le ou les réseaux que l'on va annoncer et le ou bien les voisins avec qui on va établir une session BGP. Chaque voisin doit être accessible directement, sans routage[29].

C'est tout pour une première configuration. Les points à noter :

- Contrairement à OSPF, on doit annoncer *exactement* le préfixe réseau que l'on veut diffuser. Le routeur ne tiendra pas compte des routes réelles (le routeur qui fait du BGP avec le reste du monde n'a pas forcément une interface avec tous vos réseaux : pour les joindre, il compte sur un IGP, comme OSPF). Cela peut dépendre du routeur, toutefois donc, si votre routeur n'annonce pas la route spécifiée par **network**, vous pouvez toujours vérifier ce point.
- L'adresse IP du voisin et son AS vous ont été communiqués par lui. Le nombre de possibilités de malentendus est très élevé en matière de réseau, surtout lorsqu'un suédois vous dicte une adresse IP en anglais au téléphone. Il est donc prudent de relire plusieurs fois sa configuration.

Une fois la session établie, vous devez voir arriver des routes. Ici, le journal de Quagga, si vous avez inclus un **debug bgp updates** dans la configuration :

```
2003/02/21 06:28:32 BGP: 213.248.70.225 rcvd 193.9.124.0/22
2003/02/21 06:28:32 BGP: 213.248.70.225 rcvd UPDATE w/ attr: nexthop 213.248.70.225, origin i, path
```

En vous connectant à la console de Quagga, vous devriez voir les annonces (le réseau 193.9.124.0/22 a été choisi au hasard) :

```
myrouter> show ip bgp 193.9.124.0
```

28. Il n'est pas nécessaire de spécifier un *router ID*. Le routeur en choisit un automatiquement parmi les adresses IPv4 de la machine. Mais le choisir explicitement comme étant l'adresse "principale" de la machine aidera au déboguage, par exemple lors d'un **show ip bgp neighbor**.
29. La règle exacte est moins stricte, dans certains cas, il est possible d'atteindre un voisin BGP via plusieurs sauts.

```
BGP routing table entry for 193.9.124.0/22❶
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  62.220.128.140 80.67.160.39 208.3.246.225
  1299 5400 15410 5554❷
    213.248.70.225❸ from 213.248.70.225❹ (213.248.71.184❺)
      Origin IGP, localpref 100, valid, external, best
      Last update: Sun Feb 23 17:19:53 2003
```

❶ Nous avons bien une route pour ce réseau. Avec un réseau non annoncé ou bien filtré, nous aurions eu `% Network not in table`.

❷ On voit ici la liste des AS par lesquels est passée l'annonce. BGP ne route pas entre réseaux mais entre AS. Cela n'est pas toujours optimal mais cela permet un strict contrôle de la politique de routage.

Ici, l'annonce a été émise par 5554, qui l'a transmise à 15410, puis à 5400 et 1299 avant qu'elle n'arrive chez nous.

❸ Vous voyez ici le *next hop*, le routeur suivant pour atteindre ce réseau. C'est ce *next hop* qui sera mis comme intermédiaire dans la table de routage.

❹ Vous voyez ici le routeur qui a fait l'annonce BGP. C'est souvent le même que le *next hop*.

❺ Ici, le *router ID* du routeur qui a fait l'annonce.

Avec les commandes ci-dessus, on peut afficher les routes reçues. Normalement, elles sont installées dans la table de routage du routeur (**netstat -rn** sur Unix) et utilisées par celui-ci lors du *forwarding* (**traceroute** ***une-adresse-IP*** pour vérifier).

Maintenant, comment savoir si nos routes à nous sont bien annoncées ? Le plus simple est d'utiliser un *looking glass* (voir Section 4.2.1).

2.3. Plusieurs voisins

Bien sûr, BGP avec un seul voisin n'a guère d'intérêt. On va donc configurer deux sessions :

```
hostname myrouter
router bgp 20766
  bgp router-id 80.67.160.1
  network 80.67.160.0/19
  neighbor 213.228.3.248 remote-as 13049
  neighbor  213.228.3.227  remote-as  8975
log file /var/log/zebra/bgpd.log
```

Maintenant, nous avons deux sessions différentes et les routes sont apprises par deux voisins différents.

```
myrouter> show ip bgp 193.9.124.0
BGP routing table entry for 193.9.124.0/22
Paths: (2 available❶, best #2❷, table Default-IP-Routing-Table)

  13049 1299 5400 15410 5554
    213.228.3.248❸
      Origin IGP, localpref 100, valid, external
      Last update: Sun Feb 23 17:19:53 2003

  8975 5400 15410 5554
    213.228.3.227
      Origin IGP, metric 1625, localpref 100, valid, external, best
```

```
      Last update: Sun Feb 23 17:20:02 2003
```

❶ Nous avons maintenant plusieurs routes possibles (ici, deux). Tous les voisins n'annoncent pas forcément toutes les routes et donc, même avec plusieurs voisins, on peut n'avoir qu'une seule route.

❷ Comme il y a plusieurs routes possibles, il va falloir en choisir une. L'algorithme de sélection est décrit en Section 3.2. Ici, c'est l'*AS path* le plus court qui a emporté la décision.

❸ Ici, l'adresse IP du voisin.

Pour voir les routes apprises d'un voisin particulier :

```
myrouter> show ip bgp neighbors 213.248.70.225 received-routes
BGP table version is 0, local router ID is 80.67.160.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 3.0.0.0❶          213.248.70.225                 0 1299 7018 80❷ i
*> 6.1.0.0/16       213.248.70.225                   0 1299 701 668 7170 1455 i
```

❶ Malheureusement, Quagga, comme IOS, n'affiche pas la longueur du préfixe si celui-ci correspond aux anciennes classes A, B et C. Le préfixe 3.0.0.0 correspondant à une ancienne classe A, et le préfixe annoncé étant de longueur 8, la taille des anciennes classes A, Quagga n'indique pas la longueur.

❷ Ici, l'*AS path*.

2.4. BGP avec IPv6

BGP fonctionne de la même façon avec IPv6. Notons que la configuration de Quagga est un peu plus difficile, il faut spécifier l'*address family* :

```
router bgp 65532
 neighbor fec0:ff:200::1 remote-as 100
 address-family ipv6
  network fec0:fe::/32
  neighbor fec0:ff:200::1 activate
 exit-address-family
```

On voit alors des sessions BGP au dessus d'IPv6 :

```
myrouter> show ipv6  bgp summary
BGP router identifier 10.4.200.2, local AS number 65532
5 BGP AS-PATH entries
0 BGP community entries

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
fec0:ff:200::1  4   100      20      22        0    0    0 00:17:43        1

Total number of neighbors 1
```

et des routes IPv6 :

```
myrouter> show ipv6  bgp
BGP table version is 0, local router ID is 10.4.200.2
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network                                Metric LocPrf Weight Path
*> 2001:910::/32                               0             0 100 i
     fec0:ff:200::1(fe80::206:5bff:fef1:529b)
*> fec0:fe::/32                                              32768 i
     ::

Total number of prefixes 2
```

2.5. Autres routeurs

D'autres systèmes d'exploitation auront des commandes différentes. Pour donner une idée, voici une configuration sur un routeur Extreme :

```
config bgp add network 192.168.2.0/24
config bgp as-number 65002
config bgp routerid 192.168.1.2
enable bgp
create bgp neighbor 192.168.1.1 remote-AS-number 65000
enable bgp neighbor all
```

3. Configurations plus riches

3.1. Filtrage des annonces

On l'a vu, la configuration de BGP est extrêmement simple. En quelques lignes, on peut annoncer son réseau au monde entier. Mais peu d'opérateurs utilisent une configuration aussi simple. En effet, elle les exposerait à un gros risque : celui d'une erreur chez un voisin qui leur enverrait des routes que le voisin ne pourrait pas traiter. Par exemple, si un voisin BGP annonce la totalité des routes de l'Internet, et si en plus son *AS path* est erroné et très court, tout le trafic depuis votre réseau va lui être envoyé. Cela saturera probablement sa ligne. Pire, s'il ne route pas correctement ces préfixes, il aura créé un trou noir : il annonce qu'il route pour ces réseaux mais il ne le fait pas. De telles erreurs sont relativement courantes. Lorsqu'on a des dizaines de voisins, elles se produisent plusieurs fois par an.

Inutile de dire que la réciproque est vraie : si vous annoncez à tort des routes, vous allez perturber vos voisins et mettre en danger les bonnes relations que vous avez avec eux.

Pour s'en garder, on limite souvent le nombre de préfixes que le voisin peut envoyer. Par exemple, si on sait que le voisin est un tout petit opérateur, on peut :

```
neighbor 10.5.6.7 maximum-prefixes 10
```

On le limite ainsi à dix préfixes maximum. S'il dépasse ce nombre d'annonces, la session BGP est coupée automatiquement.

On ne peut pas toujours utiliser cette technique[30]et, de toute façon, il vaut mieux mettre ceintures et bretelles : on va empêcher certaines annonces aberrantes de nous atteindre.

Première technique : on crée une liste de préfixes que l'on n'acceptera pas (syntaxe Quagga ou IOS) :

```
! Filtrage standard pour annonces entrantes Transit
! Tout permettre sauf routes aberrantes et routes à nous (80.67.160.0/19)
ip prefix-list transit-in deny 80.67.160.0/18 ge 19❶
! Les annonces trop générales sont probablement des erreurs. Refuser
! ce qui a moins de 6 bits.
ip prefix-list transit-in deny 0.0.0.0/0 le 6
! Ensuite, on filtre le RFC 1918
ip prefix-list transit-in deny 192.168.0.0/15 ge 16
ip prefix-list transit-in deny 172.16.0.0/11 ge 12
ip prefix-list transit-in deny 10.0.0.0/7 ge 8❷
ip prefix-list transit-in deny 127.0.0.0/7 ge 8
! On accepte tout le reste
ip prefix-list transit-in permit any
```

❶ Quagga exige que la longueur maximale (ici 19) soit strictement supérieure à celle du préfixe. Ici, le vrai préfixe est 80.67.160.0/19. On n'acceptera aucune annonce de ce réseau avec un préfixe de longueur égale ou supérieure (*ge*) à 19.

❷ Si vous avez configuré **debug bgp updates**, vous verrez dans le journal de Quagga des choses comme `2003/03/03 15:17:02 BGP: 10.4.200.2 rcvd UPDATE about 10.0.0.0/8 -- DENIED due to: filter`; si votre voisin tente d'envoyer des routes invalides. Prévenez-le gentiment.

Cette liste (que nous avons nommée transit-in) s'applique ensuite aux voisins :

```
neighbor 213.248.70.225 prefix-list transit-in in
```

Le "in" à la fin de la commande indique qu'on applique ce filtrage en entrée, aux annonces qui nous sont envoyées par 213.248.70.225.

Si le voisin n'est pas fournisseur de transit ou bien si on a au moins deux fournisseurs de transit, il est également prudent d'interdire l'annonce de la route par défaut.

```
ip prefix-list peer-in deny 80.67.160.0/18 ge 19
...
ip prefix-list peer-in deny 0.0.0.0/0
ip prefix-list peer-in permit any
```

En IPv6, un filtre recommandé est :

```
! Filtrage standard pour annonces entrantes Transit
! Interdire routes aberrantes et routes à nous (2001:0910::/32)
! puis, contrairement à ce qui se fait pour IPv4, autoriser les allocations
! connues (6bone, RIR, etc) et interdire tout le reste.
! Filtres de Gert (http://www.space.net/~gert/RIPE/ipv6-filters.html)
ipv6 prefix-list transit-ip6-in deny 2001:0910::/31 ge 32
! Adresses privées
ipv6 prefix-list transit-ip6-in deny FEC0::/7 ge 8
ipv6 prefix-list transit-ip6-in deny ::/0
ipv6 prefix-list transit-ip6-in permit 3ffe::/18 ge 24 le 24
ipv6 prefix-list transit-ip6-in permit 3ffe:4000::/18 ge 32 le 32
ipv6 prefix-list transit-ip6-in permit 3ffe:8000::/22 ge 28 le 28
ipv6 prefix-list transit-ip6-in permit 2001::/16 ge 28 le 35
```

30. Par exemple, un fournisseur de transit IP nous annoncera la totalité des routes de l'Internet. On ne va pas le limiter.

```
! Lire le RFC 3056 "6to4 prefixes more specific than 2002::/16 must not be propagated
ipv6 prefix-list transit-ip6-in permit 2002::/16
ipv6 prefix-list transit-ip6-in deny any
```

Les listes de préfixe agissent sur les adresses IP des réseaux (préfixes) annoncés. Il existe aussi des **filter-list** pour filtrer sur les AS.

En sortie, on peut penser que rien n'est nécessaire, puisqu'on liste explicitement les réseaux annoncés avec **network**. Mais il ne faut pas oublier que votre routeur transmet aussi les annonces qu'il a reçu. Comme ce n'est pas toujours souhaité, on filtre :

```
ip prefix-list announce-out permit 80.67.160.0/19
ip prefix-list announce-out deny any
```

Ici, la liste announce-out ne permet qu'une seule annonce, 80.67.160.0/19. On l'applique à chaque voisin :

```
neighbor 213.248.70.225 prefix-list announce-out out
```

Pour être sûr de ne pas laisser échapper des routes vers d'autres AS, si on n'est pas soi-même opérateur de transit, on filtre aussi sur l'AS :

```
! N'annoncer que les routes nous ayant pour origine : AS path vide
ip as-path access-list 1 permit ^$
```

La syntaxe utilisée est celle des expression rationnelles. On applique ce filtre :

```
neighbor 213.248.70.225 filter-list 1 out
```

Si vous êtes un opérateur, vous pouvez filtrer les annonces de vos clients. Voici un exemple où le client (nommé ici "AS20766") annonce deux préfixes :

```
ip prefix-list AS20766 description Gitoyen
ip prefix-list AS20766 permit 80.67.160.0/19 le 24
ip prefix-list AS20766 permit 217.24.80.0/20 le 24
```

3.2. Sélection de routes

Si on a plusieurs voisins, et que la même route est annoncée par certains d'entre eux, comment BGP choisit-il ? L'algorithme appliqué est bien décrit dans la documentation de Cisco (http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt2/1cdbgp.htm#1000898). Pour le résumer (certains critères ont été omis), les critères suivants sont utilisés :

1. On choisit la route avec la plus forte préférence locale. C'est un paramètre local à l'AS qui est configuré avec **set local-preference** ***un-nombre***. Il est affiché lors d'un **show ip bgp**. Par exemple, on préfère en général les liens de *peering*, gratuits, aux liens de transit.
2. On choisit la route avec l'*AS path* le plus court. Rappelez-vous que BGP route entre AS, pas entre réseaux. Rien ne garantit que la route avec l'*AS path* le plus court soit la meilleure, que ce soit en débit ou en latence. Mais c'est la seule information de topologie que BGP connaisse.

3. On choisit les routes extérieures à l'AS : c'est l'algorithme de la patate chaude, où on va chercher à faire sortir le paquet le plus tôt possible.

Voici un exemple de configuration de la préférence locale. On a deux voisins, 10.200.3.1 et 172.22.131.65, et on souhaite privilégier le premier, le second n'étant accessible que par un lien très lent, qu'on ne veut utiliser qu'en cas de panne du premier.

```
router bgp 300
 neighbor 172.22.131.65 route-map slowlink in

route-map slowlink permit 10
  set local-preference 1
  ! La préférence vaut 100 par défaut
```

Les routes annoncées par 172.22.131.65 seront alors gardées en réserve :

```
myrouter> show ip bgp 17.228.3.1
BGP routing table entry for 17.228.3.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  172.22.131.65
  100 200
    10.200.3.1 from 10.200.3.1 (192.134.7.246)
      Origin IGP, localpref 100, valid, external, best
      Last update: Mon Mar  3 16:50:44 2003

  200
    172.22.131.65 from 172.22.131.65 (10.4.200.2)
      Origin IGP, localpref 1, valid, external
      Last update: Mon Mar  3 16:50:23 2003
```

Il est assez facile de contrôler le trafic sortant et de déterminer par où il sort. Avec le trafic entrant, c'est bien plus difficile[31]. La décision est prise par un AS qui n'est pas le vôtre. La seule méthode générale est de jouer sur la longueur de l'*AS path*. Vous pouvez en effet ajouter votre propre numéro d'AS dans l'*AS path* pour décourager plus ou moins l'utilisation d'un lien.

Par exemple, si vous avez une connexion à un point d'échange local, dont l'utilisation est gratuite, et une autre à un fournisseur de transit cher, vous allez chercher à décourager l'utilisation du transit. Pour cela :

```
route-map expensive-transit-out permit 10
  ! Rallonger artificiellement l'AS_PATH pour défavoriser le transit, plus cher
  set as-path prepend 20766
```

et vous appliquez cette route-map avec :

```
neighbor 2001:6c0:800:2000::2 route-map expensive-transit-out out
```

Vos annonces vont ainsi être allongées d'une unité (vérifiez-le avec un *looking glass*).

31. Et rappelez-vous toujours que le trafic n'est pas forcément symétrique.

3.3. Les points d'échange

Vous n'avez pas forcément plusieurs fournisseurs de transit, car cela coûte cher. Mais vous aurez quand même besoin de BGP si vous vous connectez à un point d'échange. Un point d'échange est un endroit où les opérateurs Internet se connectent pour échanger du trafic. Physiquement, c'est en général un ou plusieurs commutateurs Ethernet, chaque opérateur se connectant à un port. Le point d'échange leur attribue des adresses IP et ils peuvent à partir de là se parler directement. Pour échanger du trafic, ils montent des sessions BGP entre eux. Par exemple, le FreeIX (http://www.freeix.net/), le Pouix (http://www.pouix.net/) ou bien le Kixp (http://www.kixp.net) sont des points d'échange.

À un point d'échange, les opérateurs sont des pairs : il n'y a plus de relation client-fournisseur. Il faut donc veiller à sa configuration.

Les points d'échange peuvent fournir d'autres services comme un *route server*, une machine qui établit des sessions BGP avec plusieurs opérateurs et redistribue les routes à chacun, évitant ainsi d'établir une session BGP par pair.

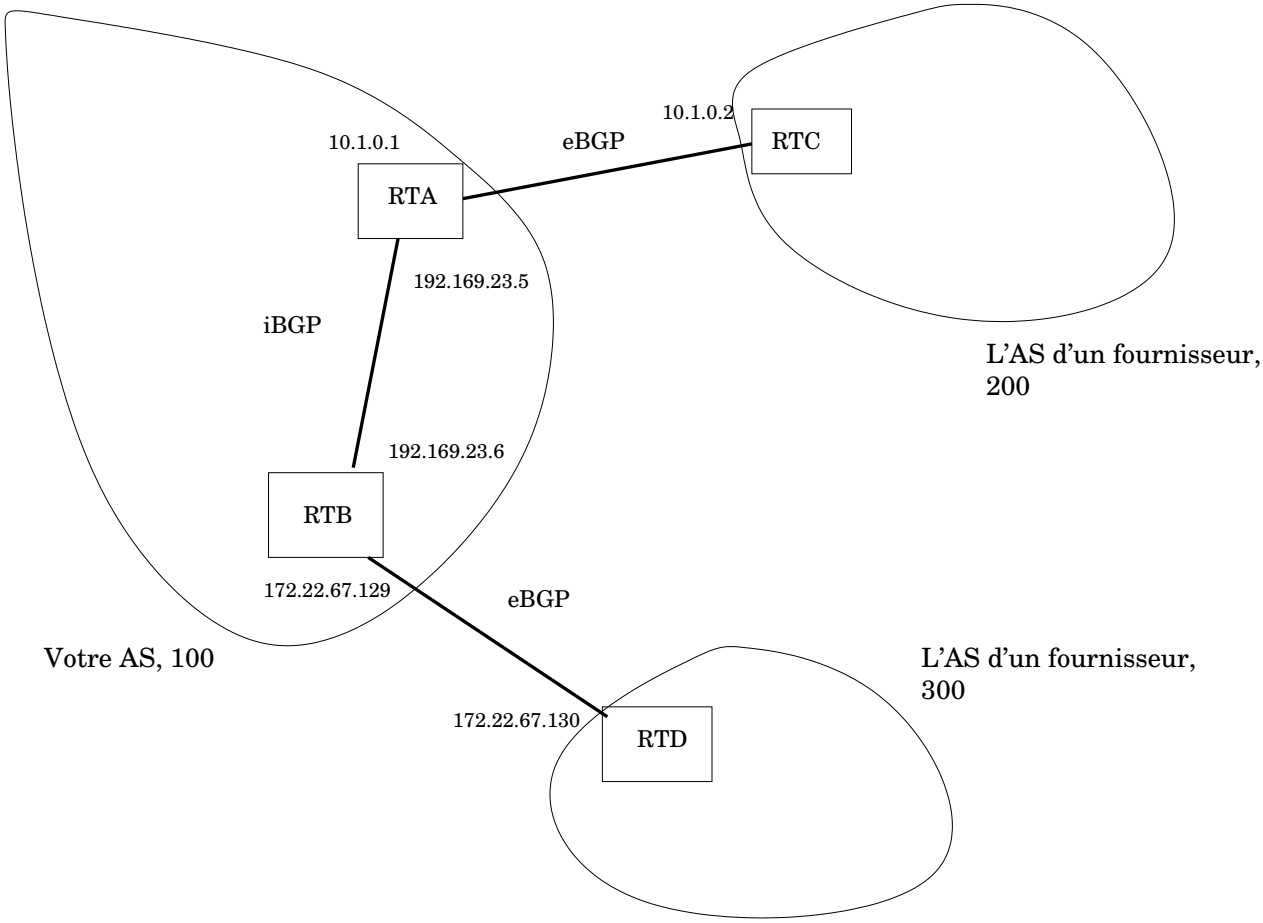
3.4. iBGP

Jusqu'à présent, vous n'aviez qu'un seul routeur, éventuellement connecté à plusieurs fournisseurs. Mais dans la réalité, on a souvent plusieurs ASBR (*Autonomous System Boundary Router*) qui font du BGP avec l'extérieur. Soit parce qu'on répartit la charge et les risques[32], soit parce que l'AS est géographiquement étendu et qu'on a un fournisseur à Rabat et un autre à Casablanca.

Dans ce cas, il faut synchroniser les routeurs BGP entre eux. Cela se fait typiquement en établissant des sessions BGP entre eux. Comme ces sessions se font à l'intérieur du même AS, on parle de iBGP (*Internal BGP*), les sessions BGP entre AS, que nous avons déjà utilisées, se nommant eBGP (*External BGP*). Dès que le numéro d'AS est le même aux deux bouts, le routeur BGP sait qu'il fait de l'iBGP.

32. Il ne serait pas très utile d'avoir plusieurs fournisseurs pour la redondance, si on n'a qu'un seul routeur, dont la panne couperait les liens avec tous les fournisseurs.

Figure 2. iBGP dans l'AS



La configuration correspondant à ce schéma est la suivante :

```
! RTA
router bgp 100
  bgp router-id 192.169.23.5
  ! Session eBGP
  neighbor 10.1.0.2 remote-as 200
  ! Session iBGP
  neighbor 192.169.23.6 remote-as 100
```

Important : En iBGP, le *next hop* (routeur suivant) n'est pas modifié, par défaut. Cela veut dire que les routeurs de vos fournisseurs ou pairs doivent être joignables de tout l'AS, via OSPF ou via une route statique. Si vous oubliez cela, vous aurez des routes non installées (voir l'exercice à ce sujet).

Important : Cela suppose que vos routeurs sont capables de suivre récursivement une route (avec IOS, cela nécessite **no synchronization** dans **router bgp** ***AS***). Sinon (http://www.cisco.com/warp/public/459/bgp_noad.html), les routes seront bien apprises par iBGP mais pas installées dans la table de routage (si vous avez fait un **debug ip bgp updates**, vous verrez des `BGP: nettable_walker 172.22.0.0/255.255.0.0 not synchronized`). Vous devrez alors utiliser la technique **next-hop-self** décrite plus loin.

Si vous avez peu de routeurs iBGP, il peut être plus intéressant de changer le *next hop*. Dans ce cas, le routeur qui annonce sera celui qui route. Cela se fait avec :

```
neighbor 192.169.23.6 next-hop-self
```

show ip bgp ***adresse-IP*** vous donnera le *next hop* pour cette adresse IP.

> **Important :** *Tous* les routeurs iBGP d'un AS doivent avoir une session iBGP avec *tous* les autres, pour que l'AS soit complètement connecté.

3.5. Avec des ressources privées

Compte-tenu de la faible taille de certaines ressources, comme les numéros d'AS (16 bits seulement), il est parfois très difficile d'obtenir des numéros d'AS publics. On peut toutefois faire du BGP avec des numéros d'AS privés (de 64512 à 65535) mais cela nécessite du soin pour éviter que des *AS path* avec ces numéros se retrouvent dans la table de routage globale.

La solution la plus simple est de supprimer ces numéros privés lors des annonces extérieures :

```
neighbor fec0:1::2 remove-private-AS
```

On trouvera tous les détails chez Cisco (http://www.cisco.com/warp/public/459/32.html).

3.6. Interaction entre BGP et OSPF

Pour l'instant, nous avons complètement séparé le routage externe avec BGP du routage interne avec OSPF. C'est la méthode recommandée. Mais il est parfois utile de mélanger partiellement les deux. Tous les routeurs permettent de redistribuer l'information acquise par OSPF dans BGP ou l'inverse.

> **Avertissement**
>
> Redistribuer l'information interne dans BGP est très dangereux : n'importe quelle oscillation ou panne de votre réseau sera visible à l'extérieur.
>
> C'est en outre inutile : vous avez typiquement beaucoup moins de routeurs externes que de routeurs internes et la topologie que vous faites connaitre à l'extérieur est donc plus simple.

4. Déboguage

Si les choses ne marchent pas comme attendu, il faut procéder avec méthode. Si vous envisagez de demander sur une liste de diffusion [33], il va falloir pouvoir donner toute l'information issue de ce processus de déboguage.

33. Une très bonne idée : je suis toujours stupéfait que tant de responsables réseau passent des semaines à chercher seuls alors que la compétence de dizaines d'autres ingénieurs expérimentés est à leur disposition gratuitement. Une raison probable est que demander de l'aide à des égaux (pas à un enseignant ou à un consultant payé pour cela) sur une liste publique ou même semi-publique est une compétence nécessaire et qui ne s'apprend pas dans les écoles, malheureusement.

4.1. Tester la connectivité de base

D'abord, il va falloir tester la connectivité de base de vos routeurs. Si deux routeurs ne peuvent pas se pinguer, BGP ne marchera certainement pas[34]

Si ping échoue (naturellement, vous testez ping avec une adresse IP comme argument, pas un nom, pour ne pas dépendre du DNS), entamez la procédure de déboguage classique en cas de routage statique.

Une fois que ping entre routeurs voisins fonctionne, vous pouvez regarder la table de routage. traceroute vous montrera le chemin pris par les paquets[35]. Les commandes **route -n** sur Linux, **route -n show** sur NetBSD ou encore **show ip route** sur IOS vous montreront la table de routage (FIB, *Forwarding Information Base*) utilisée par le routeur.

4.2. Tester les problèmes spécifiquement BGP

La première chose à tester est l'établissement de la session BGP. BGP tourne au dessus de TCP et une session BGP nécessite une session TCP. Une fois celle-ci établie, les deux routeurs se synchronisent et s'envoient des préfixes.

```
myrouter> show ip bgp summary
BGP router identifier 80.67.168.1, local AS number 20766
22155 BGP AS-PATH entries
73 BGP community entries
Neighbor❶        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
213.228.3.217   4 21502       0       0        0    0    0 00:01:33  Active❷
213.228.3.218   4 15703  141649  141414        0    0    0 03w6d06h❸ 17❹
...
```

❶ C'est l'adresse IP du voisin qui est indiquée, pas son *router ID*.

❷ *Active* signifie que la session n'a pas encore pu être établie. Toute valeur autre qu'un chiffre indique un problème.

❸ Les sessions BGP peuvent durer très longtemps (ici, plus de trois semaines).

❹ Quant la session est établie, le nombre de préfixes reçu est affiché ici.

L'examen du journal de Quagga[36] nous donne des informations supplémentaires par exemple :

```
2003/02/17 14:19:48 BGP: MAXPFXEXCEED: No. of prefix received from 10.228.3.227 (afi 1): 100 exceed
```

(un voisin BGP a dépassé la valeur du paramètre **maximum-prefixes** : la session BGP devra être redémarrée manuellement avec **clear bgp** ***addresse-IP***).

Outre les mécanismes de déboguage des routeurs (journal de Quagga, commandes **debug ?** d'IOS), les traditionnels outils de déboguage réseau comme tcpdump et ethereal sont souvent très précieux. tcpdump est disponible partout et permet de voir facilement si des paquets BGP circulent :

```
%  sudo tcpdump -i eth1 -n port bgp
11:09:54.874825 194.68.129.170.179 > 194.68.129.186.58511: P 889028570:889028589(19) ack 3950494887
11:09:54.874893 194.68.129.186.58511 > 194.68.129.170.179: . ack 19 win 63066 <nop,nop,timestamp 725
```

34. Il y a une exception : en présence de filtres (comme les ACL d'IOS ou bien le Netfilter de Linux), certains protocoles marchent et d'autres pas. Compte tenu de la complexité supplémentaire que cela entraine, je recommande de faire vos premiers essais BGP sur une machine sans filtres.
35. Rappelez-vous toujours que traceroute ne montre que le chemin aller. Si vous avez plusieurs routes possibles, rien ne garantit que le chemin retour soit identique.
36. L'endroit exact où se trouve ce journal dépend de votre configuration. C'est souvent `/var/log/zebra/bgpd.log`. Je rappelle que **tail -f** est la façon la plus pratique de lire un fichier journal.

On peut aussi recueillir des informations avec Section 1.7.3. RFC 1657 décrit la MIB associée à BGP. Ici, on interroge un routeur Unix+Quagga [37](15 est le point de départ de la MIB BGP, voir le RFC ci-dessus) :

```
%snmpwalk -m /usr/share/mibs/BGP4-MIB.txt 80.67.160.1  password 15
...
bgp.bgpPeerTable.bgpPeerEntry.bgpPeerState.194.68.129.224 = established(6)
bgp.bgpPeerTable.bgpPeerEntry.bgpPeerState.194.68.129.225 = established(6)
```

ce qui permet de voir, entre outres, que les session BGP avec nos voisins sont dans un état normal[38] .

4.2.1. Les *looking glasses*

Un *looking glass* est un système public permettant d'obtenir des informations BGP. C'est un outil indispensable pour voir les tables BGP d'un autre point de vue. Par exemple, le *looking glass* va vous permettre de voir si vos annonces sont bien propagées.

L'interface d'un *looking glass* peut être en ligne de commande ou bien via le Web. On trouve les adresses de beaucoup de *looking glass* sur traceroute.org (http://www.traceroute.org/).

Prenons un exemple, le *looking glass* de Netlantis (http://www.netlantis.org/), qui a des sessions BGP avec de très nombreux routeurs en Europe :

```
% telnet 62.220.128.140
zebra>show ip bgp 192.134.7.250
BGP routing table entry for 192.134.4.0/22
Paths: (28 available, best #28, table Default-IP-Routing-Table)
...
  3303 2200 2485
    164.128.32.11 from 164.128.32.11 (164.128.32.11)
      Origin IGP, localpref 100, valid, external, best
      Last update: Mon Feb 24 12:04:57 2003
```

On voit ici que le réseau est bien annoncé, 192.134.4.0/22 est reçu via de nombreuses voies. Toutes les annonces se font via l'AS 2200 (à l'heure où j'écris, c'est en effet uniquement via ce fournisseur que ce réseau est annoncé).

Le *looking glass* de Swinog est neutre au sens où il ne dépend pas d'un opérateur particulier, il tient ses données des sessions BGP qu'il établit avec de nombreux opérateurs. Si maintenant nous sommes intéressés par l'état de nos annonces chez un opérateur particulier, ici Tiscali :

```
% telnet     route-server.ip.tiscali.net
route-server.ip.tiscali.net>show ip bgp       192.134.7.250
BGP routing table entry for 192.134.4.0/22, version 7626472
Paths: (1 available, best #1)
  Not advertised to any peer
  3257 5511 2200 2485
    213.200.64.93 from 213.200.64.93 (213.200.87.18)
      Origin IGP, metric 110, localpref 100, valid, external, best
      Community: 3257:4130 3257:5033
```

On voit que Tiscali a bien reçu notre annonce.

37. L'option -m est nécessaire si les MIB ne sont pas installées là où SNMP les attend.
38. On peut ensuite développer d'utiles applications. Par exemple, Marc Hauswirth a écrit un script de surveillance des sessions BGP pour le logiciel de contrôle mon (http://www.kernel.org/software/mon/), script qui utilise SNMP et qui permet d'être automatiquement prévenu si une session BGP tombe.

D'autres *looking glass* sont accessibles via le Web, en remplissant un formulaire.

4.2.2. Divers

traceroute.org (http://www.traceroute.org/) permet également de faire un traceroute inverse, c'est-à-dire de voir le chemin qu'empruntent les paquets pour venir chez vous[39].

Certains traceroute, comme celui d'IOS lorsque le routeur est un routeur BGP, ou bien lft (http://www.mainnerve.com/lft/) sur Unix, permettent d'afficher la liste des AS parcourus. Ici, un exemple avec lft :

```
% lft -A www.afnog.org

Tracing ___________________________________________!_!_!!__!___!_!___!___.

TTL  LFT trace to wawa.eahd.or.ug (216.129.132.164):80/tcp
 1   [AS20766] meowth.gitoyen.net (80.67.160.1) 0.5ms
 2   [AS20766] voltaire-gw.gitoyen.net (80.67.160.34) 1.1ms
 3   [AS6461] 132.ge3-0.er1a.cdg2.fr.mfnx.net (62.4.73.26) 1.4ms
 4   [AS6461] pos0-3.cr1.cdg2.fr.mfnx.net (208.184.231.206) 1.6ms
 5   [AS6461] so-5-0-0.cr1.lhr3.uk.mfnx.net (64.125.31.154) 11.4ms
 6   [AS6461] so-0-0-0.cr2.lhr3.uk.mfnx.net (208.184.231.146) 11.4ms
 7   [AS6461] so-7-0-0.cr2.lga1.us.mfnx.net (64.125.31.182) 83.0ms
 8   [AS6461] so-0-0-0.cr2.lga2.us.mfnx.net (208.184.232.198) 82.9ms
 9   [AS6461] pos1-0.pr1.lga2.us.mfnx.net (216.200.127.170) 82.8ms
10   [AS6461] uunet-abovenet-oc12.lga2.above.net (208.184.231.246) 83.8ms
11   [AS701] 526.at-5-1-0.XR2.NYC8.ALTER.NET (152.63.23.78) 84.0ms
12   [AS701] 282.ATM7-0.XR2.TTN1.ALTER.NET (152.63.30.98) 88.4ms
13   [AS701] POS11-0-0.SR1.TTN1.ALTER.NET (152.63.15.213) 88.3ms
14   [AS701] darkom-gw.customer.alter.net (157.130.222.162) 87.8ms
15   [AS11908] host-66-133-0-46.verestar.net (66.133.0.46) 87.9ms
16   [AS11908] lci-gw10-s0-230.lightband.net (216.129.151.114) 97.9ms
17   [AS11908] afsat-gw.lightband.com (216.129.134.226) 98.6ms
18   [ASN?] 192.168.200.2 746.5ms
19   [AS11908] [target] wawa.eahd.or.ug (216.129.132.164):80 930.6ms
```

On note que le routeur 18 a une adresse privée (RFC 1918) et que son AS ne peut donc pas être trouvé.

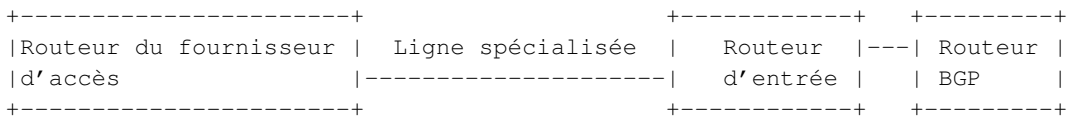
4.3. Des exemples de déboguage

Les exemples de cette section sont présentés sous une forme de questions à une liste de diffusion, avec les informations nécessaires puis les réponses[40].

Q : Ma session BGP ne s'établit pas et je ne vois même pas les paquets avec tcpdump

J'ai la configuration suivante :

39. Rappelez-vous que le chemin n'est pas forcément symétrique.
40. Cela n'a rien à voir avec BGP mais je regrette l'abondance, sur les listes de diffusion techniques, de messages mal écrits, avec insuffisamment d'informations. J'essaie donc de montrer le bon exemple.



Je ne peux pas utiliser le routeur BGP comme routeur d'entrée car il n'a pas la bonne carte pour se connecter à la ligne spécialisée. Et je ne peux pas utiliser le routeur d'entrée comme routeur BGP car il n'a pas assez de mémoire.

La session entre les deux routeurs BGP ne s'établit pas. Sur mon routeur BGP, un tcpdump ne montre pas de paquets BGP entrants, juste les demandes de mon routeur.

Pourtant, les deux routeurs peuvent se pinguer et je peux faire un **telnet *routeur-FAI* bgp** depuis mon routeur BGP : la connexion TCP s'établit bien.

R : C'est normal : par défaut, eBGP est mono-saut ce qui veut dire qu'il n'accepte pas de routeurs intermédiaires. En outre, votre routeur met la durée de vie des paquets (TTL) à 1[41]. Si vous ne pouvez pas convaincre le routeur d'entrée de fonctionner en simple pont (couche 2), la seule solution est de passer en *eBGP multihop* avec **neighbor *routeur-FAI* ebgp-multihop 2** (2 étant le nombre de sauts, que vous pouvez vérifier avec traceroute).

> **Avertissement**
>
> J'ajoute également que la configuration choisie n'est pas idéale. En séparant le contrôle (BGP) du routage effectif, vous compliquez votre réseau et vous serez probablement obligé d'utiliser des astuces assez complexes pour que tout se passe bien quand même.

Q : La session BGP ne s'établit pas, les voisins restent dans l'état Active/Idle.

Mes deux routeurs BGP (des PC/Unix avec Quagga) ne peuvent pas établir une session. Ils sont sur le même Ethernet et peuvent se pinguer. **show ip bgp summary** montre :

```
BGP router identifier 192.134.7.245, local AS number 65432
1 BGP AS-PATH entries
0 BGP community entries

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.134.7.241   4   200       6       6        0    0    0 00:00:02 Idle
```

Voici la configuration de 192.134.7.245 :

```
router bgp 65432
 bgp router-id  192.134.7.245
 network 172.17.1.0/23
 neighbor 192.134.7.241 remote-as 200
```

et de 192.134.7.241 :

```
router bgp 200
  bgp router-id 192.134.7.241
  network 10.1.0.0/16
```

41. telnet ne le fait pas et c'est pour cela que telnet marche.

```
  neighbor 192.134.7.245 remote-as 100
```

R : 192.134.7.241 pense que l'AS de 192.134.7.245 est 100 alors que 192.134.7.245 a déclaré son AS comme étant 65432. Je ne crois pas que Quagga signale cette erreur dans son journal ou dans la sortie de **show ip bgp neighbors**. Mais un tcpdump ou bien un ethereal vous aurait montré le problème (ici, avec tcpdump) :

```
192.134.7.241.179 > 192.134.7.245.46151: P 1:24(23) ack 46 win 17520:\
      BGP (NOTIFICATION: error OPEN Message Error, subcode Bad Peer AS) [ttl 1]
```

Q : La session BGP est établie mais aucune route n'est envoyée.

show ip bgp summary montre zéro préfixe :

```
requin> show ip bgp summary
BGP router identifier 192.134.7.245, local AS number 100
1 BGP AS-PATH entries
0 BGP community entries

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.134.7.241   4   200       6      10        0    0    0 00:00:01        0
```

La configuration de 192.134.7.241 (un PC/NetBSD avec Zebra) est :

```
router bgp 200
bgp router-id 192.134.7.241
neighbor 192.134.7.245 remote-as 100
```

et j'ai comme routes :

```
% route -n show
Routing tables

Internet:
Destination       Gateway           Flags
default           192.134.7.254     UG
10.4.200.0        192.134.7.245     UG
192.134.7.240     link#1            U
172.17.0.0        link#2            U
```

Pourquoi ne sont-elles pas annoncées en BGP ?

R : Par défaut, BGP n'annonce aucune route, pour des raisons de sécurité. Vous devez mettre des directives **network** dans votre fichier de configuration pour avoir des annonces.

Faites ensuite un **show ip bgp neighbour *192.134.7.245* advertised-routes** sur le routeur censé faire les annonces. Vous devriez voir quelque chose du genre :

```
BGP table version is 0, local router ID is 192.134.7.241
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 172.17.0.0       192.134.7.241            0         32768 i

Total number of prefixes 1
```

Q : Mes routes apprises en iBGP ne se retrouvent pas dans la table de routage.

Un routeur iBGP (un PC/Linux avec Zebra) reçoit bien une route en iBGP mais elle n'est pas installé dans le noyau (je ne la vois pas avec **route -n**) :

```
laperouse> show ip bgp 10.30.0.0
BGP routing table entry for 10.30.0.0/16
Paths: (1 available, no best path)
  Not advertised to any peer
  300
    192.134.7.246 (inaccessible) from 172.22.131.64 (192.134.7.241)
      Origin IGP, metric 0, localpref 100, valid, internal
      Last update: Wed Feb 26 14:02:40 2003
```

Pourquoi ? Est-ce parce que le *next hop* est marqué *inaccessible* ?

R : Oui. Le routeur 192.134.7.246 (le *next hop*) n'est pas accessible via une entrée de la table de routage. Ajoutez une route statique ou bien configurez OSPF pour annoncer cette route ou encore passez en **next-hop-self**.

Bibliographie

[faq] Denis Ovsienko, *Unofficial GNU Zebra FAQ*, 2002.

http://pilot.org.ua/zebra

[huitema] Christian Huitema, *Routing in the Internet*, 2000, Prentice-Hall.

[quagga] Paul Jakma, *Quagga Home Page*, 2003.

http://www.quagga.net/

[stewart] John W. Stewart, *BGP4: Inter-Domain Routing in the Internet*, 1998, Addison-Wesley.

[halabi] Sam Halabi, Bassam Halabi, *Internet Routing Architectures*, 2000, Cisco Press.

[beijnum] Iljitsch van Beijnum, *BGP, Building Reliable Networks with the Border Gateway Protocol* , 2002, O'Reilly.

RFC

[RFC 0954] K. Harrenstien, M. Stahl, E. Feinler, *NICNAME/WHOIS*, 1985.

[RFC 1657] S. Willis, J. Burruss, J. Chu, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2*, 1994.

[RFC 1771] Y. Rekhter, T. Li, *A Border Gateway Protocol 4 (BGP-4)*, 1995.

[RFC 1918] Y. Rekhter, R. Moskowitz, D. Karrenberg, G. Groot, E. Lear, *Address Allocation for Private Internets*, 1996.

[RFC 2622] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, M. Terpstra, *Routing Policy Specification Language (RPSL)*, 1999.

[RFC 2650] D. Meyer, J. Schmitz, C. Orange, M. Prior, C. Alaettinoglu, *Using RPSL in Practice*, 1999.

[RFC 3056] B. Carpenter, K. Moore, *Connection of IPv6 Domains via IPv4 Clouds*, 2001.