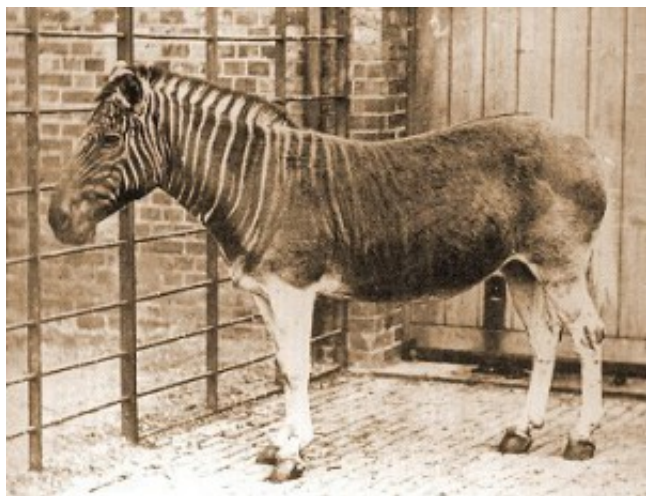




BGP, quand, pourquoi et comment ?



Une fois n'est pas coutume, un article plus technique que vulgarisateur. Si vous ne baignez pas dans les réseaux mais que vous voulez quand même tenter de vous raccrocher aux branches, je vous invite à commencer par [Internet – Comment trouver la bonne route](#) qui vous expliquera les grandes lignes du routage sur Internet sous une forme moins barbare que le présent article ainsi que le début de la série [Comment devenir FAI](#) qui vous expliquera le contexte dans lequel le sujet de cet article est employé.

Nota : je me suis fortement inspiré des efforts documentaires de [Stéphane Bortzmeyer](#). Pour une version encore plus poilue et brute de fondrie du présent article, rendez-vous [ici](#).

BGP est un protocole faisant partie des protocoles de routage. Rien de magique dedans, il ne s'agit que d'un canal de communication établi entre deux équipements (généralement des routeurs) leur permettant de s'échanger un certain nombre d'information concernant la direction que doivent emprunter les flux de données.

Si vous avez lu les articles conseillés ci-dessus, vous savez déjà que BGP est principalement utilisé pour faire communiquer et permettre l'échange de trafic entre deux réseaux autonomes distincts. On pense souvent que BGP est lui même le protocole de transport de l'information, en réalité, il n'en est rien. Les données se baladant de réseau en réseau circulent à côté du flux BGP, pas dedans. L'intérêt principal du partage d'un même média de communication pour les informations de routage et les données elle-mêmes est que si le média en question tombe en panne, les routeurs ne reçoivent plus d'information sur le lien en question et n'y envoient donc plus d'information. C'est la base de la redondance d'Internet.

C'est un passage obligé pour toute organisation exploitant un réseau autonome et souhaitant se relier au reste d'Internet. Voilà pour le quand.

Le pourquoi, à présent. Il existe plusieurs protocoles vaguement similaires à BGP qui sont quasiment tous orientés vers l'efficacité du résultat. Le second en terme de notoriété et d'utilisation dans le monde est OSPF (mais aussi ISIS, RIP, ...). Ces derniers ont pour objectif de faire discuter entre eux tous les éléments de routage d'un réseau en se basant sur une



confiance mutuelle globale. Ils sont donc généralement plus utilisés à l'intérieur d'un réseau ou la confiance est quasi totale dans tous les équipements du réseau. BGP, lui, a été conçu pour cimenter un réseau planétaire dont on ne maîtrise qu'une toute petite partie et qui peut potentiellement contenir des éléments perturbateurs (volontairement ou pas).

Le comment, maintenant. BGP fonctionne bien sûr sur les gros routeurs industriels du marché (Cisco, Brocade, Juniper, ...) mais plusieurs implémentations ont été faites pour les divers systèmes d'exploitation UNIX. Deux sortent du lot, à savoir Quagga et OpenBGP.

N'ayant encore jamais eu l'occasion de tester le second, et cet article s'inscrivant dans la série [Comment devenir son propre FAI](#), je vais bêtement suivre la meute et produire un Niemme howto sur Quagga, et pour pas trop me casser la tête, je ne parlerais que d'IPv4. Pour sortir un peu du lot quand même, je vous apprendrais peut-être que le quagga correspond à deux familles de zèbres dont l'une est éteinte depuis 1883.

L'utilité de Quagga se résume à discuter avec les routeurs voisins et à transcrire le résultat de ces conversations en ajoutant et en retirant des routes dans la table de routage de la machine (accessible avec netstat -rn généralement)

Nous partons du principe :

- que vous avez déjà réussi à installer un petit routeur sous Linux ou FreeBSD.
- que vous y avez connecté un modem ADSL d'un côté et quelques utilisateurs de l'autre et que ça fonctionne
- que vous avez installé le package Quagga (apt-get install quagga par exemple)
- que vous avez fait le nécessaire auprès du RIPE pour obtenir au moins un AS et un bloc d'IP (ou bien que vous vous êtes mis d'accord avec votre FAI pour utiliser un AS privé et un bloc d'IP du FAI lui-même juste pour tester)
- que vous avez choisi un fournisseur d'accès poilu du type FDN, et que vous avez donc le bonheur de pouvoir distribuer du vrai internet avec des vraies adresses IP routables à vos utilisateurs. Ça tombe bien, puisque de toute façon, les fournisseurs d'accès de la cabale des *agrumes libres qui savent y faire* ne proposent pas d'offre basée sur BGP via l'ADSL et qu'ils ne pourront donc, sauf utilisation de VPN pas jolis, pas vous être utiles dans le cadre du présent article.

Lorsque vous allez demander à votre fournisseur d'accès d'établir un lien BGP, quelles informations allez-vous obtenir ?

- L'adresse IP du routeur BGP du fournisseur (qui a de forte chance d'être la même IP que celle qui est utilisée comme passerelle par défaut par votre routeur, nous utiliserons 80.67.169.41)
- Le numéro d'AS du fournisseur (pour notre exemple, ce sera l'AS 20766)
- Éventuellement un mot de passe MD5 servant à protéger la session BGP d'attaques TCP aveugles

Vous lui aurez vous-même fourni votre numéro d'AS (pour notre exemple, nous utiliserons



l'AS 64512 qui se trouve être un AS privé qui n'existera jamais sur Internet) et éventuellement la liste des blocs d'IP dont vous êtes titulaire (qui peut, si vous avez bien fait votre travail avec le RIPE, être trouvé avec votre numéro d'AS. Pour notre exemple, nous utiliseront 172.16.0.0/24 qui se trouve aussi être un bloc privé qui ne devrait jamais apparaître sur Internet).

Que trouve-t-on avec le package Quagga ? Il y a une pelleté de binaires et de bibliothèques et au moins deux fichiers de configuration : zebra.conf et bgpd.conf. Le premier est la configuration de base du routeur, le second la configuration spécifique à BGP. Ne lancez pas Quagga tout de suite, nous allons commencer par le configurer.

La configuration de base est assez simple (fichier zebra.conf) :

```
!  
hostname mon_petit_routeur.mon_fai_a_moi.fr  
password mot_de_passe  
enable password second_mot_de_passe  
!  
interface em0  
ip address 172.16.0.129/25  
!  
interface em1  
ip address 80.67.169.42/30  
!  
interface em2  
!  
interface lo0  
ip address 172.16.0.1/32  
!  
line vty
```

Trois groupes de lignes sont à remarquer :

- le nom de l'hôte, qui correspond généralement au nom donné à la machine elle-même.
- les mots de passe, utilisés pour verrouiller l'accès d'administration du routeur (enable étant le mode super_utilisateur_qui_peut_tout_casser).
- la liste des interface réseau du routeur avec les adresses IP qui leur sont associées (ici, il y en a trois plus le loopback).

Nota : votre installation par défaut contiendra sûrement plus de directives de configuration que ça, vérifiez simplement que celles-ci sont bien présentes, ça suffira amplement pour commencer.

Concernant les adresses IP, vous remarquerez que l'interface em0 a une adresse IP en début



du second bloc de votre classe (vos utilisateurs auront donc des IP comprises entre 172.16.0.130 et 172.16.0.254 et utiliseront 172.16.0.129 comme route par défaut), que l'em1 a une IP du fournisseur sur un petit bloc /30 et que le loopback a une IP réelle à vous (dans le premier bloc de votre classe). Le loopback possède également bien sur la célèbre 127.0.0.1 mais on y ajoute également une IP du réseau pour pouvoir parler au routeur depuis notre réseau sur une interface qui existera toujours quelle que soit les interconnexions qui apparaîtront ou disparaîtront avec le temps.

Du côté de la configuration BGP, ça se corse un peu mais pas tant que ça :

```
!  
hostname mon_petit_routeur.mon_fai_a_moi.fr  
password mot_de_passe  
enable password second_mot_de_passe  
!  
ip route 172.16.0.128/25 em0  
!  
router bgp 64512  
bgp router-id 172.16.0.1  
  network 172.16.0.0 mask 255.255.255.0  
  neighbor 80.67.169.41 remote-as 20766  
  neighbor 80.67.169.41 description Gitoyen  
  neighbor 80.67.169.41 ebgp-multihop 255  
  neighbor 80.67.169.41 activate  
!
```

Qu'avons-nous la ?

- Les trois même lignes que le début du zebra.conf qui ont la même utilité
- La déclaration d'une route statique vers votre classe IP sur l'interface connecté à vos utilisateurs (bien qu'elle devrait déjà être gérée par le système d'exploitation, ça ne fait pas de mal de l'ajouter, ne serait-ce que pour la clarté)
- La déclaration d'une instance BGP précisant votre numéro d'AS
- L'adresse IP d'identification du routeur BGP (la même que celle que vous avez indiqué sur le loopback dans le zebra.conf)
- La déclaration du bloc d'IP que vous gérez et qui sera donc envoyé à vos voisins BGP
- La déclaration de votre premier voisin BGP contenant l'adresse IP indiquée par votre fournisseur, son numéro d'AS, une description texte qui n'a d'intérêt que pour vous y retrouver dans la configuration et une déclaration ebgp-multihop autorisant d'établir la connexion BGP avec un routeur qui n'est pas immédiatement adjacent (votre fournisseur vous dira s'il est nécessaire d'activer cette option)

Ça y est, vous pouvez lancer Quagga. Si tout fonctionne comme prévu, quelques secondes après le lancement vous devriez voir gonfler la table de routage de votre machine (netstat



-rn) jusqu'à atteindre quelques 322000 routes différentes et Internet devrait être au courant de l'existence de votre bloc 172.16.0.0/24, vos utilisateurs devraient donc pouvoir accéder à Internet.

Comment aller un peu plus loin ? Connectez-vous à votre routeur BGP en telnet

```
telnet 172.16.0.1 bgpd
Connected to mon_routeur.mon_fai_a_moi.fr
Escape character is '^]'.

```

```
Hello, this is Quagga (version 0.99.7).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

```

```
User Access Verification

```

```
Password:
mon_routeur.mon_fai_a_moi.fr>

```

Vous y êtes. Le mot de passe tapé correspond évidemment à celui que vous avez indiqué dans le début du fichier bgpd.conf. Nous allons commencer par vérifier si votre session BGP a bien été établie :

```
mon_routeur.mon_fai_a_moi.fr> show ip bgp sum

```

```
BGP router identifier 172.16.0.1, local AS number 64512
332657 BGP AS-PATH entries
0 BGP community entries

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	St
80.67.169.41	4	20766	10525	10232	0	0	0	24m	
332657									

```
Total number of neighbors 1

```

Bingo, tout fonctionne à priori, puisque notre voisin 80.67.169.41 semble connecté et nous envoie tout plein de routes. Nous allons en analyser une de plus près (en prenant par exemple l'une des IP de www.orange.fr)

```
mon_routeur.mon_fai_a_moi.fr> show ip bgp 193.252.122.103

```

```
BGP routing table entry for 193.252.122.0/24

```



```
Paths: (1 available, best #1, table Default-IP-Routing-Table)
 20766 5511 24600
   80.67.169.41 from 80.67.169.41 (80.67.169.41)
     Origin IGP, localpref 100, valid, external, best
     Last update: Sun Aug 01 12:13:51 2010
```

On apprend donc que l'ip de www.orange.fr appartient à un bloc /24, que nous n'avons qu'un seul chemin connu pour y aller (normal, nous n'avons qu'un seul fournisseur pour l'instant), que l'origine de ce bloc d'IP est dans l'AS 24600 et qu'il faut passer par l'AS 5511 après celui de notre fournisseur pour y aller. D'autres informations suivent dont vous trouverez l'explication en fouillant un peu.

BGP dispose d'une syntaxe de configuration très riche permettant de faire des choses hors du commun sans pour autant que beaucoup de documentations n'existent sur le sujet. N'hésitez pas à poser vos questions, j'essaierai d'étoffer les exemples avec le temps et de détailler les explications qui sont pour l'instant plus que succinctes.