

## Chapitre 6

# Protocole BGP

### 1. Historique et Objectifs

- EGP (1984) : RFC 904
- BGP-1 (1989) : RFC 1195
- BGP-2 (1990) : RFC 1163
- BGP-3 (1991) : RFC 1267
- BGP-4 (1995) : RFC 1771, 1772 et 1773

#### **Besoins**

- \_ **Topologie** : L'Internet est un réseau maillé entre AS complexe avec « peu » de structures
- \_ **Autonomie des AS** : Chaque AS définit le coût des liens avec des philosophies/approches différentes. Il est impossible de trouver les plus courts chemins
- \_ **Confiance** : des AS peuvent ne pas se faire confiance mutuellement pour propager de « bonnes routes »
- \_ **Opérateurs concurrents**
- \_ **Politiques** : Des AS différents ont des objectifs différents
  - \_ Minimiser le nombre de sauts, utiliser un provider plutôt qu'un autre...

#### **Objectifs :**

Le protocole BGP est le protocole de routage est utilisé sur internet.

Ce protocole est prévu pour réaliser du routage entre des AS (Autonomous System). Il est aussi prévu pour fonctionner sur de très grands réseaux.

- \_ **Échanger des routes** (du trafic) entre organismes indépendants (Opérateurs)
- \_ **Implémenter la politique de routage de chaque organisme**
  - \_ Respect des contrats passés entre organismes
  - \_ Sûreté de fonctionnement
- \_ **Être indépendant des IGP** utilisés en interne à un organisme
- \_ **Supporter un passage à l'échelle** (de l'Internet)
- \_ **Minimiser le trafic induit sur les liens**
- \_ Donner une **bonne stabilité au routage**

**Vocabulaire**

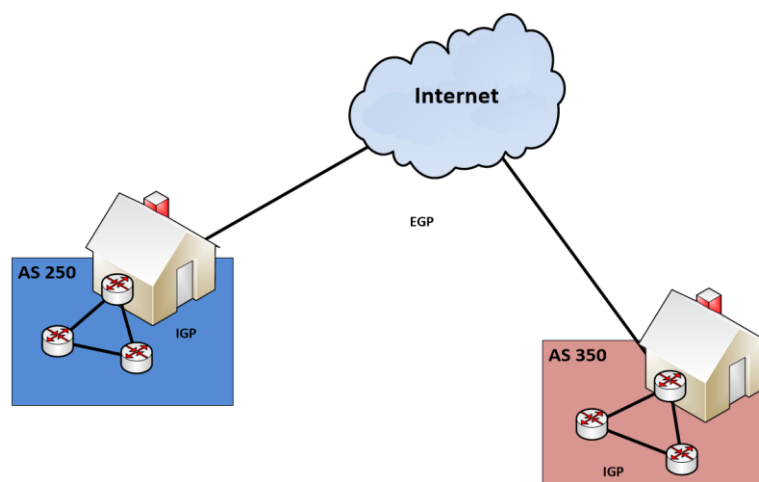
- **Voisin** (Neighbor)
  - Routeur avec qui on a une session BGP
- **NLRI/Préfixe**
  - NLRI - network layer reachability information
  - Informations concernant l'accessibilité (ou pas) d'une route (réseau + masque)
- **Router-ID** (identifiant de routeur)
  - Adresse IP la plus grande du routeur
- **Route/Path** (chemin)
  - Préfixe (NLRI) annoncé par un voisin
- **Transit** - transport de vos données par un réseau tiers, en général moyennant paiement
- **Peering** - accord bi-latéral d'échange de trafic
  - chacun annonce uniquement ses propres réseaux et ceux de ses clients à son voisin
- **Default** - route par défaut, où envoyer un paquet si la table de routage ne donne aucune information plus précise

**2. Caractéristiques de base**

Il existe 2 types de protocole de routage : les **IGP** (Interior Gateway Protocol) et les **EGP** (Exterior Gateway Protocol).

Les **IGP** assurent le routage au sein des AS (Autonomous System). Par exemple : **RIP**, **EIGRP**, **OSPF**.

Les **EGP** assurent le routage entre les AS.



**BGP** est du type **EGP**. C'est d'ailleurs le seul protocole de ce type.

Pour rappel, un AS est un ensemble de réseaux sous une même autorité. Par exemple, un réseau d'entreprise est un AS. BGP permet de connecter cet AS à internet, et donc à d'autres AS.

Pour prendre part au protocole eBGP avec un fournisseur de services Internet, vous devez posséder un numéro de système autonome (ASN) public. Vous devez obtenir un numéro ASN auprès de l'un des registres régionaux répertoriés dans le tableau ci-dessous. Lorsque vous recevez votre propre numéro ASN, vous devez contacter chaque fournisseur de services Internet pour obtenir leur numéro ASN et toute autre information nécessaire.

Région	Nom du registre	Site Web
Amérique du Nord	RIN	www.arin.net
Europe	RIPE NCC	www.ripe.net
Asie Pacifique	APNIC	www.apnic.net
Amérique latine	LACNIC	www.lacnic.net
Afrique	AfriNIC	www.afrinic.net

Les AS sont identifiés par un numéro sur 16 bits unique attribué par les mêmes organismes qui affectent les adresses IP. Il existe une plage de numéros d'AS privés de 64 512 à 65 535 pour ceux qui ne possèdent pas de numéro d'AS public.

A l'intérieur de votre AS, vous pouvez bien faire ce que vous voulez. Vous préférez tel protocole de routage ? Vous estimez que telle métrique est plus pertinente ? Tant mieux, cela ne regarde que vous. La seule contrainte, c'est que vous devrez désigner un ou plusieurs routeurs, à la frontière de votre AS, pour propager les informations d'accessibilité de vos réseaux et collecter les informations d'accessibilité des autres ASBR.

Il est utile de connaître le fonctionnement de BGP pour :

- Travailler chez un opérateur internet
- Gérer un point d'échange entre opérateurs
- Se connecter à plusieurs FAI
- Le mettre en place dans le cœur de réseau d'une très grande entreprise

BGP est du type Path Vector. C'est un dérivé du type vecteur de distance.

La métrique utilisée est très complexe, et l'administrateur de l'AS pourra choisir la métrique à utiliser. Il ne s'agit pas d'une métrique simple, mais d'une liste d'attributs associés à la route. (Voir métrique plus loin)

BGP utilise TCP sur le port 179. C'est donc TCP qui se charge d'envoyer les Ack.

La convergence sur un réseau BGP est très lente.

Les MAJ de routage sont envoyées quand il y a du changement, et seulement ce qui a changé est envoyé.

### 3. Relation de voisinage BGP

Pour établir une relation de voisinage BGP entre 2 routeurs, il faut la configurer manuellement sur les deux routeurs.

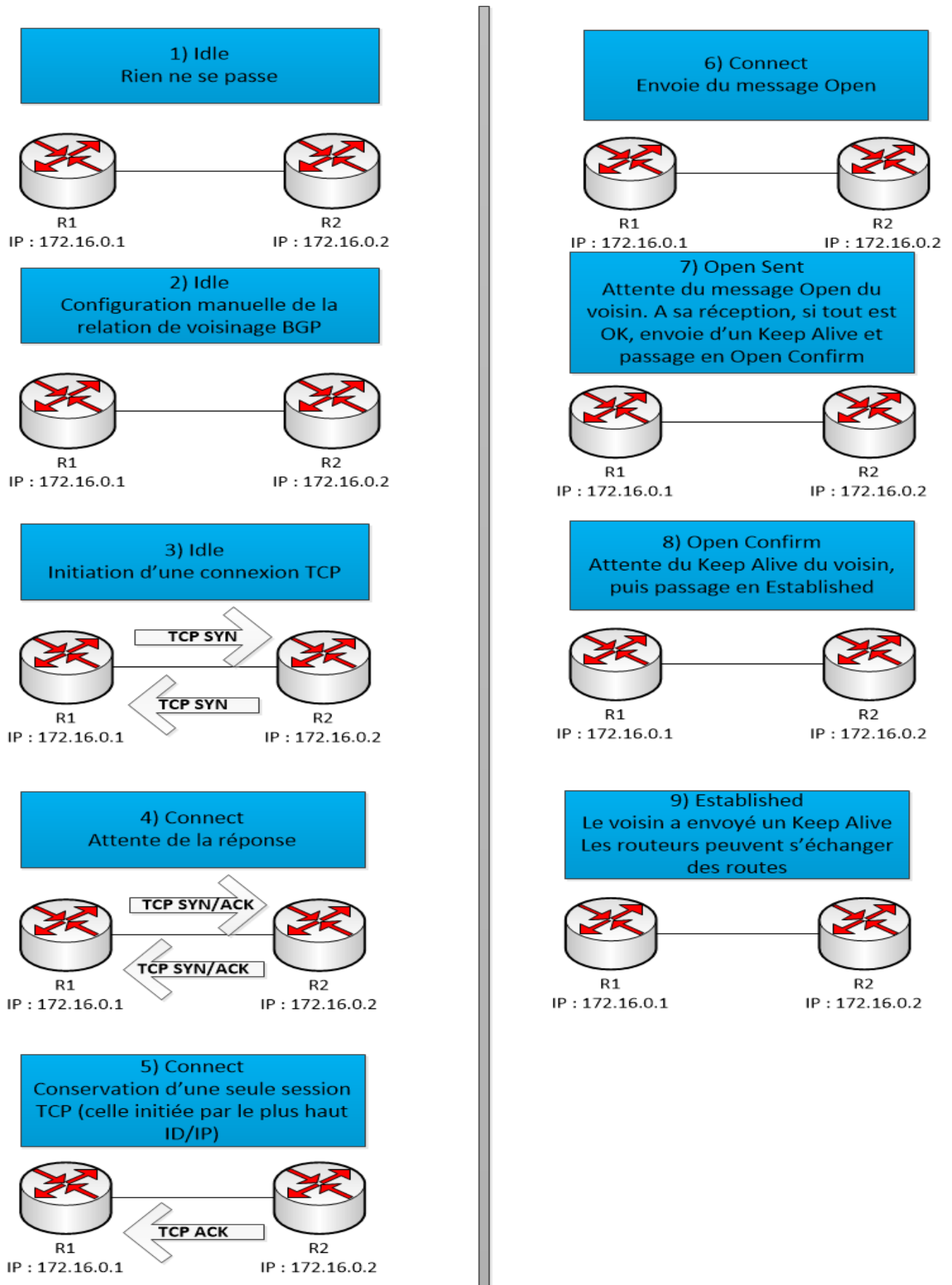
Contrairement à d'autres protocoles où les voisins se détectent automatiquement, en BGP il faut renseigner l'adresse du voisin, de manière à établir la connexion.

Comme dit précédemment, la session utilise le protocole TCP.

Voici les états d'une connexion :

- **Idle** : Premier stade d'une session BGP. Quand une ouverture de session est demandée (configuration, démarrage du routeur), le routeur initie une session TCP, puis passe en mode Connect (pour attendre la réponse)
- **Connect** : Le routeur attend une réponse (TCP Syn-Ack) du voisin ou une demande d'établissement de connexion TCP. En effet, les deux routeurs vont vouloir établir une session en même temps. Une seule session sera conservée (celle initiée par le routeur avec le plus haut ID). Il y aura donc un routeur « chef » de la relation de voisinage. Quand la session est établie, on envoie le message Open et on passe en mode Open Sent
- **Active** (facultatif) : La tentative de connexion avec le voisin a échoué (Timeout). Le routeur fait une nouvelle tentative. En cas d'échec, il retombe en mode Idle. En cas de succès, on passe en mode Open Sent. Ce statut n'est donc atteint qu'en cas d'échec en mode Connect.
- **Open Sent** : Le message Open a été envoyé. Il contient diverses infos (version BGP, ID du routeur, numéro d'AS, Hold Down Timer). A la réception du message Open du voisin, si tout convient, le routeur envoie un KeepAlive au et passe en Open Confirm.
- **Open Confirm** : Attente du KeepAlive du voisin, avant de passer en Established.
- **Established** : La relation de voisinage est établie. Les routeurs peuvent s'échanger des routes.

Voici un petit schéma résumant le processus (sans le passage en mode Active) :



## 4. Types de messages BGP

- **Open packet** : Il permet de démarrer une relation de voisinage. C'est le premier message envoyé après qu'une session TCP soit établie. Il contient entre autre la version BGP, l'ID du routeur (l'IP d'une des interfaces), le numéro de l'AS local, et le Hold Down Timer (temps max entre deux messages BGP avant de clore la session). Le message Open peut aussi contenir des informations additionnelles.
- **Keep Alive** : Permet de maintenir la relation BGP. Comme BGP n'envoie des MAJ que lorsque que c'est nécessaire (après un changement), il est fort probable que les routeurs ne discutent pas pendant longtemps (si il n'y a pas de changement). Pour s'assurer que la relation de voisinage tient toujours (et que le voisin est toujours là), des Keep Alive sont envoyés. En générale, le temps entre deux Keep Alive représente 1/3 du Hold Down Timer Max. Si on ne reçoit pas de Keep Alive ou de Update, et que le Hold Down Timer est atteint, la relation est arrêtée. Par défaut le Hold Down Timer est de 180 secondes chez Cisco. Donc 60 secondes entre les Keep Alive.
- **Update** : Permet d'annoncer de nouvelles routes, ou d'en retirer. Quand on annonce une nouvelle route, on donne l'AS Path avec. Il s'agit du chemin que va emprunter le paquet pour arriver à destination. Si le routeur qui reçoit la route pour une destination voit son propre AS dans l'AS Path, il va refuser la route (car cela causerait une boucle de routage).
- **Notification** : Informe le voisin qu'il y a un problème. La relation BGP est arrêtée (ainsi que la session TCP) et le routeur repasse en mode Idle.

## 5. Types de tables BGP

BGP utilise 3 tables pour fonctionner.

**Neighbor Table** : contient la liste des voisins à qui on est connecté. Les relations sont établies à la main.

Voici un exemple de table de voisinage :

```
R1#show ip bgp summary
BGP router identifier 172.16.10.1, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down  State/PfxRcd
172.16.10.2     4    200      3       3        1    0    0 00:00:43      0
```

Les champs contenus dans la table sont les suivants :

- **Neighbor** : l'ID du voisin
- **V** : la version BGP
- **AS** : le numéro de l'Autonomous System du voisin
- **MsgRcvd** : nombre de messages reçus (Update, KeepAlive)
- **MsgSent** : nombre de messages envoyés

- **TblVer** : dernière version de la table de routage envoyée (à comparer avec la version locale)
- **InQ** : nombre de messages en file d'attente venant du voisin
- **OutQ** : nombre de message en file d'attente, à envoyer chez le voisin
- **Up/Down** : depuis combien de temps la relation est UP
- **State/PfxRcd** : nombre de préfixes reçus par le voisin

La deuxième table est la **BGP Table**. Elle contient la liste de toutes les routes reçues, et ce depuis tous les voisins. En effet, si l'on a plusieurs voisins, il se peut que l'on reçoive plusieurs fois la même route (mais venant d'un voisin différent).

Voici un exemple :

```
R1#show ip bgp
BGP table version is 2, local router ID is 172.16.10.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 10.0.2.0/24    172.16.10.2          0           0 200 i
```

Les champs contenus dans la table sont les suivants :

\* : l'étoile signifie que la route est valide

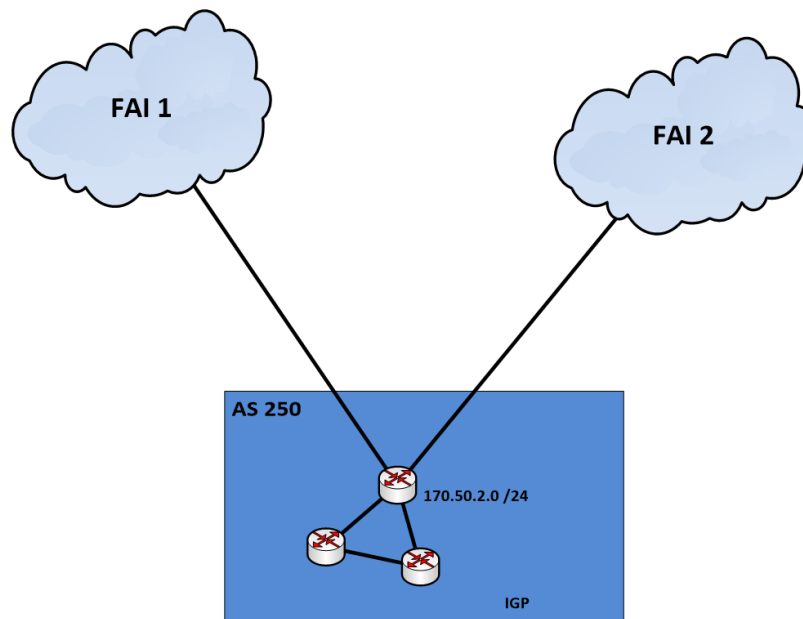
- > : le chevron signifie que c'est la meilleure route disponible pour cette destination (ici nous n'avons qu'une route, donc il n'y a pas le choix)
- **Network** : le réseau de destination
- **Next Hop** : prochain saut pour joindre la destination
- **Metric** : un des attributs constituant la métrique totale
- **LocPrf** : un des attributs constituant la métrique totale
- **Weight** : un des attributs constituant la métrique totale
- **Path** : les AS par lesquels il faudra passer (ici il n'y en a que 1 : le 200)

La troisième table est la **table de routage**, ce qui est logique. Son fonctionnement ne change pas de d'habitude.

Pour chaque destination apprise par BGP, la meilleure route sera placée dans la table de routage.

## 6. Multihoming

Le principe du Multihoming c'est d'être connecté à plusieurs FAI.



Cela permet d'avoir de la redondance (haute disponibilité), et de faire de la répartition de charge.

Cela à un intérêt sur les paquets entrants et sortants :

**Paquet sortant** : nous pouvons répartir la charge sur les deux FAI. Nous pourrions alors profiter de nos deux lignes internet. De plus, selon le réseau de destination à joindre (sur Internet), il est possible qu'un FAI offre une meilleure route. Nous pourrions alors le favoriser dans le choix du Next Hop.

Sans BGP, nous pouvons tout de même faire de la répartition de charge, par contre, nous ne pouvons pas savoir quel FAI offre le meilleur chemin (il faudra choisir au hasard).

**Paquet entrant** : nous pouvons aussi répartir la charge en entrée en annonçant la moitié de nos sous réseaux internes à un FAI, et l'autre moitié au deuxième FAI.

Dans l'exemple donnée, nous pouvons annoncer 170.50.2.0 /25 au FAI 1, et 170.50.2.128 /25 au FAI 2.

Si nous avons deux routeurs qui font face à internet, nous pouvons privilégier l'un ou l'autre en fonction du réseau interne à joindre. Ainsi, un paquet venant d'internet sera automatiquement dirigé vers le routeur (d'entrée sur notre réseau) offrant le meilleur chemin.

## Voyons maintenant les 3 façons d'implémenter BGP pour se connecter à Internet

### Cas N°1 : route par défaut



Ici nous avons deux FAI, qui nous envoient chacun une route par défaut.

Il n'est donc pas possible de favoriser un FAI ou un autre en fonction du réseau à joindre.

Par contre, nous pouvons tout de même utiliser les deux FAI simultanément (répartition de charge), ou en garder un en secours.

Nous pouvons aussi influencer le choix du FAI pour les paquets entrants (voir précédemment), selon les réseaux (internes) que l'on annonce aux FAI.

Cette solution ne consomme que très peu de ressource.

### **Cas N° 2 : MAJ partielles**

Ici les FAI nous envoient une partie de leur table de routage (table de routage de tout internet).

Nous pouvons choisir les routes (pour quelles destinations) que nous souhaitons recevoir.

Par exemple, si nous avons des bureaux distants ou des serveurs dans un Data Center, nous y enverrons régulièrement du trafic. Nous souhaitons donc savoir quel FAI nous offre le meilleur chemin vers ces destinations favorites.

Pour les autres destinations, nous choisirons aléatoirement un des FAI.

### **Cas N° 3 : MAJ complètes**

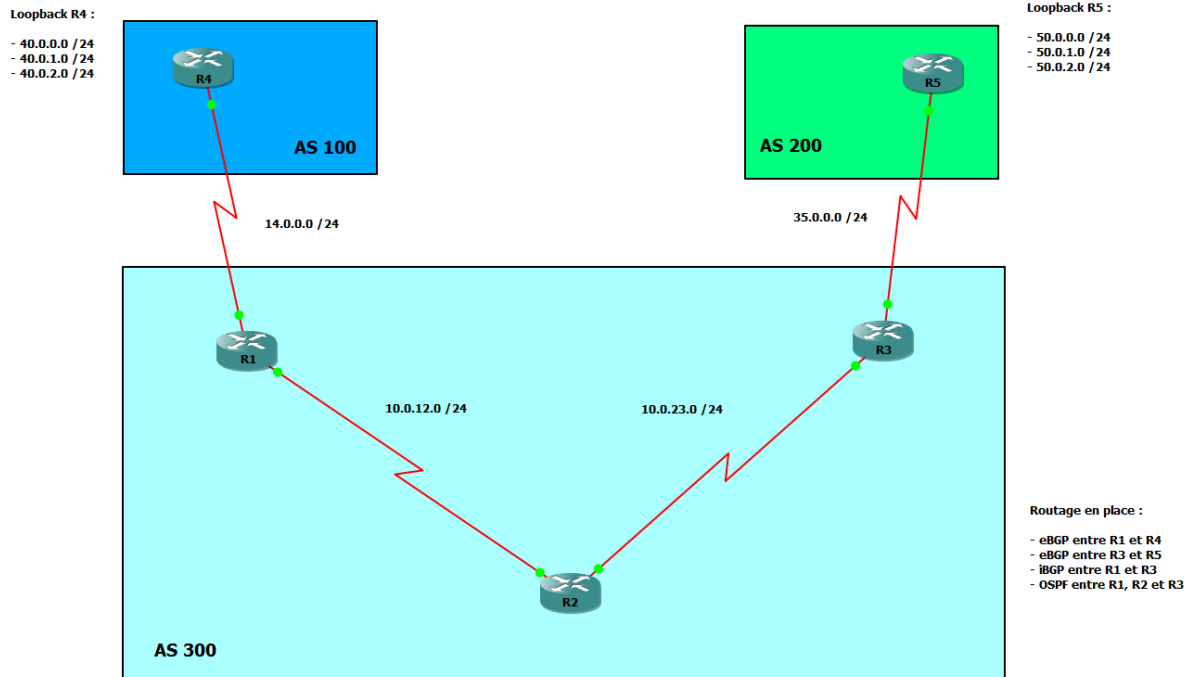
Ici les 2 (ou plus) FAI nous envoient chacun leur table de routage complète (table de routage de tout internet).

Nous pourrons donc choisir le meilleur FAI pour chaque paquet sortant.

Cette solution consomme beaucoup de ressource. Surtout beaucoup de RAM (plusieurs centaines de MO)

## **7. Configuration Basique De BGP**

### **7.1 La topologie à utiliser**



Voici la topologie que nous utiliserons tout au long de cette mise en pratique.

Le but sera de voir les configurations de base de BGP.

Trois « Autonomous System » sont à connecter grâce à BGP.

La configuration de base à mettre en place est la suivante :

- Adresses IP
- Interfaces Loopback
- OSPF sur R1, R2 et R3

## 7.2 Relations de voisinage BGP

### 7.2.1 Relation eBGP

Tout d'abord, entrer dans le mode BGP en spécifiant le numéro d'AS :

```
R1 (config) #router bgp 300
```

Puis ajouter une relation de voisinage :

```
R1 (config-router) #neighbor 14.0.0.4 remote-as 100
```

On spécifie l'IP du voisin ainsi que son AS

Il faut maintenant faire de même sur R4 :

```
R4 (config) #router bgp 100

R4 (config-router) #neighbor 14.0.0.1 remote-as 300
```

Comme nous l'avons dit, BGP est un **protocole lent**.

Après un **certain temps**, la **relation BGP va apparaître** :

```
R4 (config-router) #
*Mar 1 00:31:38.583: %BGP-5-ADJCHANGE: neighbor 14.0.0.1 Up
R4 (config-router) #
```

Nous pouvons voir la liste des relations comme ceci :

```
R4#show ip bgp summary
BGP router identifier 40.0.2.1, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
14.0.0.1         4    300      7       7        1    0    0 00:03:18      0
```

**Faisons de même pour R3 et R5 :**

```
R3 (config) #router bgp 300

R3 (config-router) #neighbor 35.0.0.5 remote-as 200
```

```
R5 (config) #router bgp 200

R5 (config-router) #neighbor 35.0.0.3 remote-as 300
```

Les relations sont créées. Ce sont des relations eBGP car elles se mettent en place entre deux Autonomous System.

### 7.2.2 Relation iBGP

Il est aussi possible de mettre en place des relations iBGP (au sein d'un même AS).

Nous allons donc mettre en place une relation iBGP entre R1 et R3. Le but est qu'ils s'échangent leurs routes BGP sans qu'il soit nécessaire d'utiliser de la redistribution.

Vous pourrez constater qu'il n'est pas nécessaire que les routeurs soient directement connectés pour qu'une relation se forme.

Pour une relation iBGP, il est conseillé d'utiliser une IP de Loopback comme IP de voisin.

Pourquoi ? Car si nous utilisons une IP d'interface, et que celle-ci tombe en panne, la relation BGP disparaît. Bien sûr, cela est utile que si notre réseau dispose de liens redondants vers ce voisin.

Si nous avons un lien direct entre R1 et R3 (en plus du lien R1 -> R2 -> R3), et que nous utilisons 10.0.23.3 comme IP de voisin pour R3, la relation BGP disparaîtrait si le lien R1 -> R2 -> R3 tombait (alors qu'il reste le lien R1 -> R3).

Ici nous n'avons qu'un lien reliant R1 et R3. Mais voyons tout de même comment faire une relation de voisinage avec une IP de Loopback.

Sur R1 et R3, ajouter une interface de Loopback. Ne pas oublier d'inclure cette interface dans le processus OSPF.

R1(config)#interface loopback 0

```
R1 (config-if) #ip address 1.1.1.1 255.255.255.255  
  
R1 (config) #router ospf 1  
  
R1 (config-router) #network 1.1.1.1 0.0.0.0 area 0
```

```
R3 (config) #interface loopback 0  
  
R3 (config-if) #ip address 3.3.3.3 255.255.255.255  
  
R3 (config) #router ospf 1
```

```
R3(config-router)#network 3.3.3.3 0.0.0.0 area 0
```

Ensuite, configurer la relation BGP :

```
R1(config)#router bgp 300
```

```
R1(config-router)#neighbor 3.3.3.3 remote-as 300
```

R3(config)#router bgp 300

```
R3(config-router)#neighbor 1.1.1.1 remote-as 300
```

Un « **Show ip bgp summary** » vous montrera que la relation n'est pas UP :

```
R1(config-router)#do show ip bgp summ
BGP router identifier 14.0.0.1, local AS number 300
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
3.3.3.3	4	300	0	0	0	0	0	never	Active
14.0.0.4	4	100	38	38	1	0	0	00:34:49	0

Cela tient au fait que quand R3 envoie un message à R1, le message a pour source 10.0.23.3 (l'IP de l'interface qui envoie le message).

Or R1 s'attend à recevoir des messages de 3.3.3.3

Il faut donc changer la source des messages pour cette relation de voisinage :

```
R1(config-router)#neighbor 3.3.3.3 update-source loopback 0
```

```
R3(config-router)#neighbor 1.1.1.1 update-source loopback 0
```

```
R1#show ip bgp summary
BGP router identifier 14.0.0.1, local AS number 300
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
3.3.3.3	4	300	5	5	1	0	0	00:01:34	0
14.0.0.4	4	100	44	44	1	0	0	00:40:00	0

### 7.2.3 Relation eBGP avec IP de Loopback

Finissons par une petite particularité, la mise en place de relation eBGP avec une IP de Loopback.

Si vous faites ceci, il faut ajouter une commande (en plus de la relation de voisinage avec IP de Loopback). La voici en exemple pour R5 :

```
R3(config-router)#neighbor 5.5.5.5 ebgp-multihop 2
```

Il est nécessaire d'utiliser cette commande, car par défaut, une relation eBGP ne peut pas s'établir sur une distance de plus de 1 saut.

Il faudra appliquer la même commande sur R5.

Il faut aussi que les deux routeurs sachent comment joindre l'IP de Loopback. Dans l'exemple, R5 doit avoir une route vers 3.3.3.3 (et inversement pour R5).

L'utilisation d'une IP de Loopback pour une relation eBGP est utile si vous possédez un lien redondant (double lien) vers le voisin.

### 7.3 Distribution de route

## Solution 1 : La commande Network

Les relations de voisinage sont en place, c'est une bonne chose.

Par contre, il manque quelque chose :

```
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
O      1.1.1.1 [110/129] via 10.0.23.2, 03:44:59, Serial0/0
    35.0.0.0/24 is subnetted, 1 subnets
C      35.0.0.0 is directly connected, Serial0/1
    3.0.0.0/32 is subnetted, 1 subnets
C      3.3.3.3 is directly connected, Loopback0
    10.0.0.0/24 is subnetted, 2 subnets
O      10.0.12.0 [110/128] via 10.0.23.2, 03:44:59, Serial0/0
C      10.0.23.0 is directly connected, Serial0/0
```

Les routes ne sont pas redistribuées par BGP.

La commande « Neighbor » ne fait qu'établir la relation BGP.

Pour annoncer des réseaux (les inclurent dans le processus BGP), voici la procédure :

```
R4(config)#router bgp 100

R4(config-router)#network 40.0.0.0 mask 255.255.255.0

R4(config-router)#network 40.0.1.0 mask 255.255.255.0

R4(config-router)#network 40.0.2.0 mask 255.255.255.0
```

Contrairement aux autres protocoles, la commande « **Network** » ne fait qu'ajouter des réseaux au processus. Elle ne permet pas de créer une relation avec un voisin.

Voici le résultat :

```

R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
C      1.1.1.1 is directly connected, Loopback0
    3.0.0.0/32 is subnetted, 1 subnets
O      3.3.3.3 [110/129] via 10.0.12.2, 05:00:45, Serial0/0
    40.0.0.0/24 is subnetted, 3 subnets
B      40.0.0.0 [20/0] via 14.0.0.4, 00:06:22
B      40.0.1.0 [20/0] via 14.0.0.4, 00:05:51
B      40.0.2.0 [20/0] via 14.0.0.4, 00:05:51
    10.0.0.0/24 is subnetted, 2 subnets
C      10.0.12.0 is directly connected, Serial0/0
O      10.0.23.0 [110/128] via 10.0.12.2, 05:00:47, Serial0/0
    14.0.0.0/24 is subnetted, 1 subnets
C      14.0.0.0 is directly connected, Serial0/1

```

A ce stade, vous vous demandez sûrement pourquoi nous n'avons pas annoncé le réseau 40.0.0.0 /22.

Simplement parce que BGP ne peut annoncer une route que si elle est présente dans la table de routage.

Or, voici la table de routage de R4 :

```

R4(config-router)#do show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    40.0.0.0/24 is subnetted, 3 subnets
C      40.0.0.0 is directly connected, Loopback0
C      40.0.1.0 is directly connected, Loopback1
C      40.0.2.0 is directly connected, Loopback2
    14.0.0.0/24 is subnetted, 1 subnets
C      14.0.0.0 is directly connected, Serial0/0

```

La route 40.0.0.0 /22 n'existe pas. Nous ne pouvons donc pas l'annoncer avec BGP.



Ensuite, annoncez les réseaux de R5 :

```
R5(config-router)#network 50.0.0.0 mask 255.255.255.0  
  
R5(config-router)#network 50.0.1.0 mask 255.255.255.0  
  
R5(config-router)#network 50.0.2.0 mask 255.255.255.0
```

## Solution 2 : la redistribution

La deuxième solution consiste à utiliser la redistribution de route.

Ici rien de bien nouveau. Voyons tout de même un exemple pour BGP.

Nous allons ajouter deux interfaces de Loopback sur R5, de manière à créer de nouvelles routes.

```
R5(config)#interface loopback 3  
  
R5(config-if)#ip address 50.1.0.1 255.255.255.0  
  
R5(config-if)#interface loopback 4  
  
R5(config-if)#ip address 50.1.1.1 255.255.255.0
```

Le but est que R5 annonce ces routes en BGP.

Profitons-en pour faire une redistribution propre :

```
R5(config)#access-list 10 permit 50.1.0.0 0.0.0.255  
  
R5(config)#access-list 10 permit 50.1.1.0 0.0.0.255
```

```
R5(config)#route-map BGPRedistribution  
  
R5(config-route-map)#match ip address 10
```

```
R5 (config) #router bgp 200
```

```
R5 (config-router) #redistribute connected route-map BGPRedistribution
```

Voici le résultat :

```
R3#show ip route
      1.0.0.0/32 is subnetted, 1 subnets
O       1.1.1.1 [110/129] via 10.0.23.2, 00:55:20, Serial0/0
      50.0.0.0/24 is subnetted, 5 subnets
B       50.0.2.0 [20/0] via 35.0.0.5, 00:54:20
B       50.1.1.0 [20/0] via 35.0.0.5, 00:01:30
B       50.0.0.0 [20/0] via 35.0.0.5, 00:02:31
B       50.1.0.0 [20/0] via 35.0.0.5, 00:02:01
B       50.0.1.0 [20/0] via 35.0.0.5, 00:02:01
      35.0.0.0/24 is subnetted, 1 subnets
C       35.0.0.0 is directly connected, Serial0/1
      3.0.0.0/32 is subnetted, 1 subnets
C       3.3.3.3 is directly connected, Loopback0
      10.0.0.0/24 is subnetted, 2 subnets
O       10.0.12.0 [110/128] via 10.0.23.2, 00:56:26, Serial0/0
C       10.0.23.0 is directly connected, Serial0/0
```

## 7.4 Distribution des routes par iBGP

Nous avons mis en place la distribution de route par BGP.

R4 annonce ses routes à R1, et R5 annonce ses routes à R3.

Mais il y a un problème que vous aurez peut être remarqué : R1 et R3 ne s'échange pas de route par BGP.

La relation iBGP entre R1 et R3 ne semble pas permettre la redistribution des routes apprises par eBGP.

Vous pouvez voir cela sur la table de routage juste au-dessus. R3 ne connaît pas les réseaux 40.0.0.0 /22

En réalité, R1 connaît les routes, mais il ne les place pas dans sa table de routage.

```

R1#show bgp
BGP table version is 4, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 40.0.0.0/24     14.0.0.4             0           0 100 i
*> 40.0.1.0/24     14.0.0.4             0           0 100 i
*> 40.0.2.0/24     14.0.0.4             0           0 100 i
* 150.0.0.0/24     35.0.0.5             0        100   0 200 i
* 150.0.1.0/24     35.0.0.5             0        100   0 200 i
* 150.0.2.0/24     35.0.0.5             0        100   0 200 i
* 150.1.0.0/24     35.0.0.5             0        100   0 200 ?
* 150.1.1.0/24     35.0.0.5             0        100   0 200 ?

```

Il n'y a pas de chevron devant la route. Elle n'est donc pas mise dans la table de routage.

Mais pourquoi donc ?

Il y a deux raisons.

**1<sup>er</sup> raison** : Un routeur n'apprend pas une route par iBGP (et ne la redistribue pas) tant qu'il n'a pas appris cette même route par un protocole de routage interne (type OSPF).

Dans notre topologie, R1 n'apprendra pas la route pour 50.0.0.0 /24 par iBGP (donc venant de R3), tant qu'il n'a pas reçu une MAJ OSPF contenant une route vers cette même destination.

Cette règle était valable sur les anciens routeurs Cisco. Pour l'annuler, il faut entrer la commande « **no synchronization** ».

Un « **Show run** » vous permettra de savoir si la commande est activée par défaut.

Si ce n'est pas le cas, il faut l'entrer sur R1 et R3.

```

R1#show run | s bgp
router bgp 300
  no synchronization
  bgp log-neighbor-changes
  neighbor 3.3.3.3 remote-as 300
  neighbor 3.3.3.3 update-source Loopback0
  neighbor 14.0.0.4 remote-as 100

```

**2<sup>e</sup> raison** : Quand eBGP annonce une route, il change le Next-Hop. Quand iBGP annonce une route, il ne change pas le next hop.

Qu'est-ce que cela pose comme problème ?

Et bien R3 va annoncer une route à R1, ayant pour destination 50.0.0.0 /24, et pour Next Hop 35.0.0.5.

Hors R1 n'a pas la moindre idée de comment joindre 35.0.0.5.

Il ne va donc pas placer cette route dans sa table de routage.

La commande « **Show bgp** » vous permet de voir les routes avec leurs Next-Hop.

```
R1#show bgp
BGP table version is 4, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight Path
*> 40.0.0.0/24     14.0.0.4           0             0 100 i
*> 40.0.1.0/24     14.0.0.4           0             0 100 i
*> 40.0.2.0/24     14.0.0.4           0             0 100 i
* i50.0.0.0/24     35.0.0.5           0          100   0 200 i
* i50.0.1.0/24     35.0.0.5           0          100   0 200 i
* i50.0.2.0/24     35.0.0.5           0          100   0 200 i
* i50.1.0.0/24     35.0.0.5           0          100   0 200 ?
* i50.1.1.0/24     35.0.0.5           0          100   0 200 ?
```

Il va donc falloir changer le Next-Hop au moment où R3 annonce les routes à R1.

La manipulation est simple :

```
R3(config)#router bgp 300

R3(config-router)#neighbor 1.1.1.1 next-hop-self
```

R3 va changer le Next-Hop par son IP quand il va redistribuer des routes à R1.

Pareil pour R1 :

```
R1(config)#router bgp 300

R1(config-router)#neighbor 3.3.3.3 next-hop-self
```

Voyons si cela a marché :

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

  1.0.0.0/32 is subnetted, 1 subnets
C      1.1.1.1 is directly connected, Loopback0
  50.0.0.0/24 is subnetted, 5 subnets
B      50.0.2.0 [200/0] via 3.3.3.3, 00:03:35
B      50.1.1.0 [200/0] via 3.3.3.3, 00:03:35
B      50.0.0.0 [200/0] via 3.3.3.3, 00:03:35
B      50.1.0.0 [200/0] via 3.3.3.3, 00:03:35
B      50.0.1.0 [200/0] via 3.3.3.3, 00:03:35
  3.0.0.0/32 is subnetted, 1 subnets
O      3.3.3.3 [110/129] via 10.0.12.2, 01:30:24, Serial0/0
  40.0.0.0/24 is subnetted, 3 subnets
B      40.0.0.0 [20/0] via 14.0.0.4, 01:29:26
B      40.0.1.0 [20/0] via 14.0.0.4, 01:29:26
B      40.0.2.0 [20/0] via 14.0.0.4, 01:29:26
  10.0.0.0/24 is subnetted, 2 subnets
C      10.0.12.0 is directly connected, Serial0/0
O      10.0.23.0 [110/128] via 10.0.12.2, 01:30:27, Serial0/0
  14.0.0.0/24 is subnetted, 1 subnets
C      14.0.0.0 is directly connected, Serial0/1
```

En effet, R1 connaît maintenant les routes redistribuées par R3 (et inversement pour R1).

Au passage, vous noterez que R4 a aussi appris les routes (pareil pour R5).

```

R4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    50.0.0.0/24 is subnetted, 5 subnets
B       50.0.2.0 [20/0] via 14.0.0.1, 00:07:13
B       50.1.1.0 [20/0] via 14.0.0.1, 00:07:13
B       50.0.0.0 [20/0] via 14.0.0.1, 00:07:13
B       50.1.0.0 [20/0] via 14.0.0.1, 00:07:13
B       50.0.1.0 [20/0] via 14.0.0.1, 00:07:13
    40.0.0.0/24 is subnetted, 3 subnets
C       40.0.0.0 is directly connected, Loopback0
C       40.0.1.0 is directly connected, Loopback1
C       40.0.2.0 is directly connected, Loopback2
    14.0.0.0/24 is subnetted, 1 subnets
C       14.0.0.0 is directly connected, Serial0/0

```

## 7.5 Les trous noirs causés par BGP

Aviez-vous remarqué que vous avez créé un trou noir ? Ou plutôt que je vous ai fait créer un trou noir ?

Bon certes, le terme est un peu fort, mais tout de même.

Pour constater les dégâts, faites un ping depuis R4 vers 50.0.0.1.

Normalement, il devrait passer, non ?

```

R4#ping 50.0.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 50.0.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

```

Et bien non ...

Et pourquoi ? Car R2 ne connaît pas la route vers 50.0.0.0 /22.

Et pourquoi ne connaît-il pas la route ? Car celle-ci a été annoncée en BGP. La MAJ est venue de R5, elle est passée par R3 puis a traversé R2 (sans que celui-ci ne la prenne en compte), puis elle est arrivée à R1 (puis R4).

Au final, R2 a bien reçu la MAJ. Sauf que comme il n'utilise pas BGP (et donc qu'il n'a pas de relation de voisinage), il ne la pas assimilée. Il s'est contenté de la diriger vers sa destination (R1).

Mais comment empêcher le trou noir ?

### **3 solutions :**

- Mettre en place un lien direct entre R1 et R3
- Redistribuer les routes BGP dans OSPF (R2 va donc les assimiler)
- Mettre en place BGP sur R2

La mise en place d'un lien est la solution la plus simple est la plus performante. Il faut tout de même que le câblage soit réalisable.

La redistribution de route est une bonne solution si R2 est capable de supporter toutes les routes. Dans cet exemple, il n'y a pas beaucoup de route. Mais imaginez une topologie avec de très grandes tables de routage (ce qui arrive quand on se connecte à un FAI).

Il faut donc voir si R2 possède suffisamment de ressources.

De plus, le routeur de destination doit connaître la route de retour.

Par exemple, pour aller de R1 à R5, la redistribution permet à R1 de connaître 50.0.0.0 /22, mais il faut que R5 connaisse la route pour revenir à 10.0.12.0 /24.

Il en est de même pour l'utilisation de BGP. A la différence près que BGP conserve toutes les routes reçues. Si R1 et R3 envoient des routes pour la même direction, R2 va toutes les conserver. Puis il va mettre la meilleure dans la table de routage.

Si R1 et R3 envoient des routes similaires, cette solution est plus gourmande que la redistribution.

Libre à vous de choisir la solution adéquate.

Vous devriez être capable de faire fonctionner les trois.

Pour ma part, je vais faire simple. Je vais placer un lien entre R1 et R3.

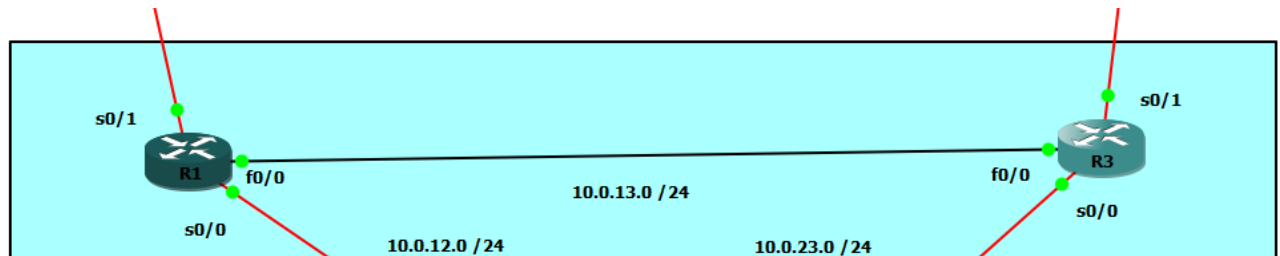
Néanmoins, je vous encourage à tenter une des deux autres solutions.

Il y a un ou deux pièges à éviter. La mise en place peut donc être intéressante.

Il faut bien faire attention. Pour que les solutions fonctionnent il faut que les routes soient connues de tous les routeurs.

Voici pour la mise en place d'un lien :

Ajouter le lien :



Configurer les interfaces.

Mettre en place OSPF entre R1 et R3 (pour que le meilleur chemin vers 3.3.3.3 soit le lien Ethernet).

Inclure les réseaux 14.0.0.0 / 24, 35.0.0.0 / 24 et 10.0.13.0 / 24 dans le processus BGP.

Voici le résultat :

```
R4#ping 50.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 50.0.0.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/22/40 ms
```



## Contenu

1. Historique et Objectifs.....	1
2. Caractéristiques de base .....	2
3. Relation de voisinage BGP .....	4
4. Types de messages BGP .....	6
5. Types de tables BGP .....	6
6. Multihoming .....	7
7. Configuration Basique De BGP .....	9
7.1 La topologie à utiliser .....	9
7.2 Relations de voisinage BGP .....	10
7.2.1 Relation eBGP .....	10
7.2.2 Relation iBGP .....	12
7.2.3 Relation eBGP avec IP de Loopback .....	14
7.3 Distribution de route .....	14
7.4 Distribution des routes par iBGP .....	18
7.5 Les trous noirs causés par BGP .....	22