Module 7 – Filtrage de routes BGP et fonctionnalités avancées

Objectif : À partir du réseau configuré lors du Module 6, utilisation de différentes techniques de configuration des peerings BGP pour mettre en évidence le filtrage de voisins (neighbour filtering) et d'autres fonctionnalités avancées de l'IOS.
Prérequis : Module 6

Topologie:

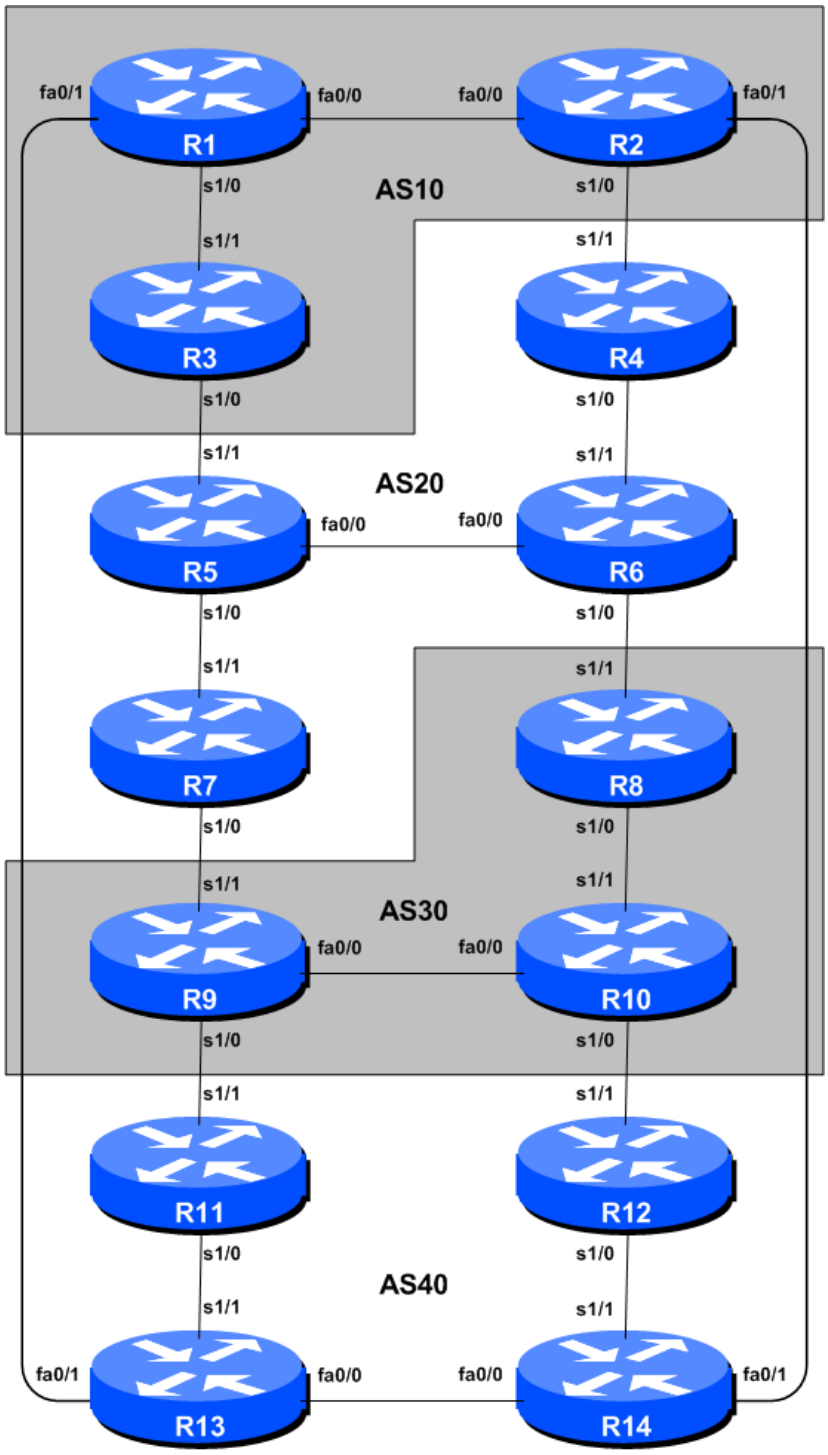


Figure 1 – Numéros d'AS BGP

Notes relatives au TP

Le module précédent a permis de s'initier à la configuration de BGP externe, sans toutefois expliquer comment contrôler quels sont les réseaux annoncés à quels AS. Le but de ce module est donc de vous présenter les types de politiques de routage disponibles en utilisant BGP.

Dans les étapes 2 à 6, nous allons configurer chaque AS pour en faire un "**AS refusant le transit** (**non-transit AS**)", c'est à dire que l'AS refusera aux AS auxquels il est connecté d'acheminer du traffic de l'un vers l'autre. Cela signifie que la connectivité dans la salle sera rompue - par exemple, AS3 ne sera plus capable de voir les réseaux d'AS10, etc... C'est un choix délibéré ayant pour but de montrer l'efficacité du filtrage BGP.

Ceci est réalisé lors de l'étape 2, en configurant un filtre pour les routes sortantes (outgoing route filter) permettant l'envoi des seuls préfixes locaux vers les peers eBGP. Nous nous assurons également que les peers nous envoient seulement leurs préfixes locaux, grâce à l'ajout d'un filtre entrant (incoming filter). En règle générale, il est bon de configurer des filtres dans les deux sens, afin de protéger contre les erreurs de configurations des deux côtés du peering. Lors de l'étape 4, nous allons utiliser les filtres AS-PATH. Enfin dans l'étape 6, nous arriverons au même résultat en utilisant les communautés BGP.

Important : chaque étape doit être suivie et terminée par tous les participants **avant que la prochaine étape puisse démarrer**, puisque les techniques utilisées changent à chaque étape et qu'elles ne peuvent pas fonctionner dans un environnement incohérent. Ne démarrez pas l'étape suivante sans avoir obtenu le feu vert de vos instructeurs. Si vous démarrez sans attendre, il est très probable que le routage cesse de fonctionner, empêchant ainsi les autres groupes de comprendre les effets réels de leurs configurations.

Important : gardez la configuration utilisée pour le Module 6.

Étapes du TP

1. **Implémentation des politiques BGP.** Avant de faire la moindre configuration dans ce module, il faut impérativement comprendre comment implémenter les politiques BGP. Il est possible d'ajouter des listes de préfixes (prefix lists), des filtres as-path (as-path filters) ou des route-maps directement depuis la CLI. Cependant, ils ne seront pris en compte que pour les updates BGP reçues après l'ajout de ces nouvelles politiques. Ceci s'explique par le fait que BGP envoie des updates incrémentales décrivant les changements aux routes annoncées ou retirées. Pour appliquer une politique de routage à l'intégralité de la table de routage BGP reçue ou envoyée à un peer, la session BGP doit être "réinitialisée". Dans les anciennes versions de l'IOS, cela signifiait détruire la session BGP puis la redémarrer. Cependant, comment on peut l'imaginer, cela causait de graves problèmes de stabilité sur le réseau du fournisseur. Ainsi, le RFC2918 (Route Refresh Capability / Fonction de rafraîchissement de route) a été ajouté à la plupart des implémentations modernes de BGP pour permettre des mises à jour correctes des sessions BGP lorsque des changements de politiques les rendaient nécessaires.

   Pour implémenter les changements de politiques dans tous les exemples à venir, utilisez les commandes suivantes, par exemple pour implémenter de nouvelles politiques entrantes et sortantes sur le peering entre Routeur 1 et Routeur 13:

```
Router1# clear ip bgp 10.10.15.14 out
Router1# clear ip bgp 10.10.15.14 in
```

Note 1: N'oubliez pas les sous-commandes 'out' et 'in' dans les commandes clear ci-dessus - si vous les oubliez, cela forcera une réinitialisation complète de la session BGP. Relisez la présentation BGP si vous n'êtes plus sûr de la raison pour laquelle c'est une mauvaise idée.

Note 2: Plutôt que d'utiliser l'adresse IP du voisin pour rafraîchir la session BGP, il est également possible d'utiliser son numéro d'AS (ASN). Puisque le routeur 13 est dans l'AS 40, l'autre jeu de commandes possible serait :

```
Router1# clear ip bgp 40 out
Router1# clear ip bgp 40 in
```

Point de contrôle #1 : *appelez l'assistant pour qu'il vérifie la connectivité. Chaque groupe doit vérifier ses peerings pour voir les effets de cette étape. Utilisez la commande "show ip bgp" pour voir la table BGP - assurez-vous que les préfixes externes appris par BGP ont maintenant une adresse de prochain saut locale. Utilisez la commande "trace" pour montrer que la connectivité réseau n'est pas affectée.*

Filtrage en utilisant les listes de préfixes (prefix-lists)

2. **Configuration des filtres de préfixes basés sur les adresses réseau.** Pour cette configuration, nous allons utiliser les listes de préfixes, qui sont une méthode de contrôle des informations de réseaux échangées au sein des peerings BGP. Le but de cette manoeuvre est de configurer les peerings eBGP de sorte que seuls les réseaux des **AS voisins** soient échangés.

Exemple: Routeur R13 (peering avec R1)

```
!
ip prefix-list out-peer permit 10.40.0.0/20
ip prefix-list out-peer deny 0.0.0.0/0 le 32
!
ip prefix-list in-peer permit 10.10.0.0/20
ip prefix-list in-peer deny 0.0.0.0/0 le 32
!
router bgp 40
 no synchronization
 network 10.40.0.0 mask 255.255.240.0
 neighbor 10.10.15.13 remote-as 10
 neighbor 10.10.15.13 description eBGP peering avec Routeur1
 neighbor 10.10.15.13 prefix-list out-peer out
 neighbor 10.10.15.13 prefix-list in-peer in
 no auto-summary
!
```

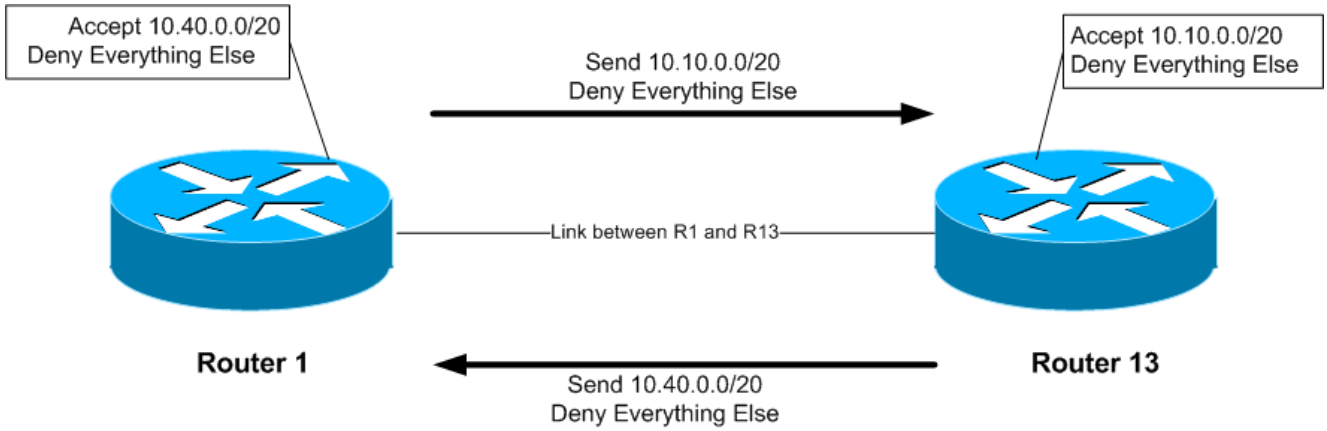
Exemple: Routeur R1 (peering avec R13)

```
!
ip prefix-list out-peer permit 10.10.0.0/20
ip prefix-list out-peer deny 0.0.0.0/0 le 32
!
ip prefix-list in-peer permit 10.40.0.0/20
```

```
ip prefix-list in-peer deny 0.0.0.0/0 le 32
!
router bgp 10
 no synchronization
 network 10.10.0.0 mask 255.255.240.0
 neighbor 10.10.15.14 remote-as 40
 neighbor 10.10.15.14 description Peering avec Routeur13
 neighbor 10.10.15.14 prefix-list out-peer out
 neighbor 10.10.15.14 prefix-list in-peer in
 no auto-summary
!
```

Note : une liste de préfixes IOS possède toujours une dernière entrée implicite *deny any* même si elle n'est pas mentionnée dans la configuration. Certains FAIs ajoutent l'entrée implicite *deny any* parce qu'ils considèrent que c'est une bonne habitude, et une mesure de sécurité.

Note : ces listes de préfixes sont seulement appliquées aux peerings avec les autres AS. On les appelle des peerings externes (utilisés par eBGP). Il n'y a généralement pas besoin d'appliquer ce genre de filtres aux peerings iBGP.



Point de contrôle #2 : *appelez l'assistant pour qu'il vérifie la connectivité. Chaque groupe doit vérifier ses peerings pour voir les effets de cette étape.*
Utilisez les commandes "show ip bgp neigh x.x.x.x advertise | route".

ARRÊTEZ ET ATTENDEZ ICI

3. **Retrait de la configuration précédente.** Cette étape illustre comment retirer la configuration définie à l'étape précédente. C'est indispensable avant de passer à la suite.

Exemple: Routeur R1

```
Router1#conf t
Router1(config)#router bgp 10
!
! On enleve d'abord la liste de prefixes du peering BGP avec R13
!
Router1(config-router)#no neighbor 10.10.15.14 prefix-list out-peer out
Router1(config-router)#no neighbor 10.10.15.14 prefix-list in-peer in
!
```

```
! Maintenant on enleve les listes ells-memes
!
Router1(config)#no ip prefix-list out-peer
Router1(config)#no ip prefix-list in-peer
!
! Voila, la configuration est nettoyee, comme il faut.
!
Router1(config)#end
!
! Puis on rafraichit le peering BGP pour enlever l'ancienne politique
!
Router1#clear ip bgp 40 out
Router1#clear ip bgp 40 in
Router1#
```

Filtres AS-PATH

4. **Configuration de filtres de préfixes basés sur l'attribut AS path :** Durant cette étape nous allons utiliser les listes d'accès AS path, qui sont un autre moyen de contrôler les réseaux échangés dans les peerings BGP.

Exemple : Routeur R14 (peering avec R2)

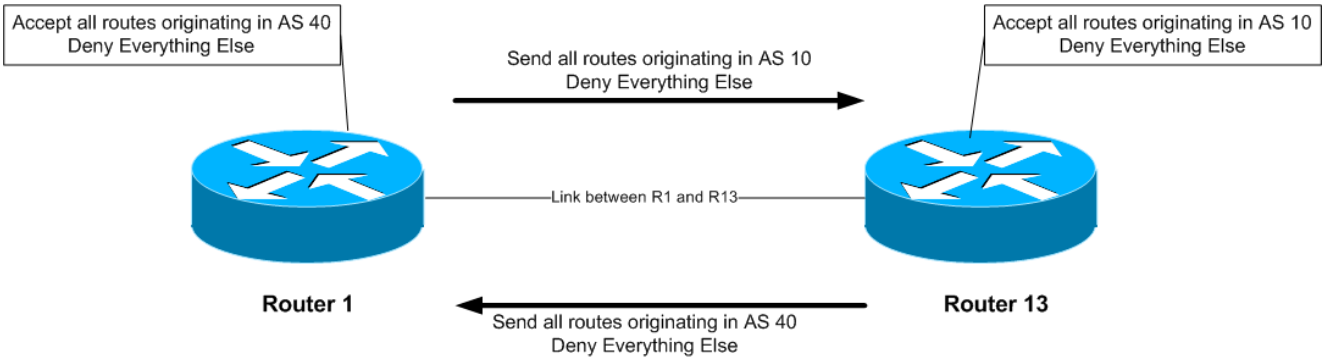
```
ip as-path access-list 2 permit ^$
ip as-path access-list 3 permit ^10$
!
router bgp 40
 neighbor 10.40.15.18 remote-as 10
 neighbor 10.40.15.18 filter-list 2 out
 neighbor 10.40.15.18 filter-list 3 in
!
```

Exemple : Routeur R2 (peering avec R14)

```
ip as-path access-list 2 permit ^$
ip as-path access-list 3 permit ^40$
!
router bgp 10
 neighbor 10.40.15.17 remote-as 40
 neighbor 10.40.15.17 filter-list 2 out
 neighbor 10.40.15.17 filter-list 3 in
!
```

Q. Pourquoi est-ce que la liste de filtres sortants arrive à une correspondance pour l'as-path null et non le numéro d'AS local, dans les exemples ci-dessus ?

A. Parce que l'attribut AS-PATH est positionné après application des listes de préfixes, filtres d'as-path et route-maps. Si l'AS local était inclus dans la configuration des listes de filtres sortants, les préfixes seraient ignorés parce que l'attribut AS-PATH n'est pas encore positionné à ce moment.



Pour vérifier que l'expression régulière fonctionne comme prévu, utilisez la commande exec "***show ip bgp regexp*** *<expression-reguliere>*" pour montrer tous les chemins qui correspondent à l'expression régulière spécifiée. N'oubliez pas que la commande "*clear ip bgp <neighbour address> [in|out]*" est requise pour implémenter ce filtre.

Notez bien que les filtres AS Path autorisent tous les préfixes émis par l'AS voisin - dans l'exemple précédent qui utilisait les filtres de préfixes, seuls les agrégés de l'AS voisin pouvaient passer les filtres.

Point de contrôle #3 : *appelez l'assistant pour qu'il vérifie la connectivité. Une fois que l'instructeur donne le feu vert à la classe, passez à l'étape suivante.*

ARRÊTEZ ET ATTENDEZ ICI

5. **Retrait de la configuration précédente.** Cette étape illustre comment retirer la configuration définie à l'étape précédente. C'est indispensable avant de passer à la suite.

Exemple : Routeur 2

```
Router1#conf t
Router1(config)#router bgp 10
!
! On enleve la liste de filtres du peering BGP avec R14
!
Router1(config-router)#no neighbor 10.40.15.17 filter-list 2 out
Router1(config-router)#no neighbor 10.40.15.17 filter-list 3 in
!
! Puis les listes de filtres elles-memes
!
Router1(config)#no ip as-path access-list 2
Router1(config)#no ip as-path access-list 3
!
! Voila, la configuration est nettoyee, comme il faut.
!
Router1(config)#end
!
! Puis on rafraichit le peering BGP pour enlever l'ancienne politique
!
Router1#clear ip bgp 10.40.15.17 in
Router1#clear ip bgp 10.40.15.17 out
Router1#
```

Communautés BGP pour le filtrage (et route-maps)

6. **Introduction aux concepts de communautés BGP et route-maps.** Cette étape explique comment utiliser ces concepts pour le tagging, l'identification et finalement le filtrage de préfixes. Nous obtiendrons des résultats similaires à ceux des étapes précédentes.

   Sur chaque routeur, configurez BGP de sorte à ce qu'il envoie à ses peers externes une communauté pour **tous les préfixes** appartenant à l'AS local. La communauté doit suivre le format *[Numéro d'AS]:[Numéro de Routeur]*. Par exemple, Routeur8 doit utiliser la communauté 30:8.

   Exemple : Routeur R8

```
! Affiche les communautes en utilisant le format 16-bits:16-bits plutot que
! un seul entier de 32-bits.
!
ip bgp-community new-format
!
ip prefix-list out-match permit 10.30.0.0/20 le 26
!
route-map outfilter permit 10
 match ip address prefix-list out-match
 set community 30:8
!
router bgp 30
 neighbor 10.20.15.17 remote-as 20
 neighbor 10.20.15.17 route-map outfilter out
 neighbor 10.20.15.17 send-community
!
```

   Notes :
   1) La commande *show ip bgp <network>* permet de voir l'attribut de communauté dans la table BGP.
   2) L'attribut de communauté est un champ défini sur 32 bits. Par contention établie à l'IETF, ce champ est séparé en deux sous-champs de 16 bits chacun pour une interprétation plus aisée. Les 16 bits de poids fort contiennent le numéro d'AS tandis que les 16 bits de poids faible représentent une valeur dont la signification est bien définie pour les deux AS participant au peering. Les seules exceptions sont les chaînes de caractères définies spécifiquement, telles que ***no-export*** et ***local-as***.

   Q: Pourquoi faut-il mettre *send-community* pour les peerings eBGP ?
   A: Puisque les valeurs de communauté ne sont pas propagées par défaut, vous devez le configure explicitement sur le routeur.

Point de contrôle #4 : *appelez l'assistant pour qu'il vérifie la connectivité. Chaque groupe doit vérifier ses peerings pour voir les effets de cette étape.*

ARRÊTEZ ET ATTENDEZ ICI

7. **Retrait de la configuration précédente.** Cette étape illustre comment retirer la configuration définie à l'étape précédente. C'est indispensable avant de passer à la suite.

   Exemple : Routeur 9

```
Router8#conf t
Router8(config)#router bgp 30
!
! On enleve d'abord la route-map du peering eBGP
!
Router8(config-router)#no neighbor x.x.x.x route-map outfilter out
!
! Puis la route-map elle-meme
!
Router8(config)#no route-map outfilter
!
! Et maintenant la liste de prefixes utilise par la route-map
!
Router8(config)#no ip prefix-list out-match
!
! Voila, la configuration est nettoyee, comme il faut.
!
Router8(config)#end
!
! Puis on rafraichit le peering BGP pour enlever l'ancienne politique
!
Router8#clear ip bgp 10.20.15.17 out
Router9#
```

Communautés BGP

8. **Configuration des communautés BGP.** Lors de l'étape 6, la communauté d'appartenance d'un réseau était générée au point de peering où les deux routeurs BGP pouvaient communiquer. Malgré l'intérêt pédagogique de cette technique, la façon de faire usuelle, est d'attacher l'information de communauté à un réseau lorsque le réseau est injecté dans la table de routage BGP.

   Chaque groupe doit assigner une communauté au ***bloc réseau client*** (le /26) dont il est l'origine (originator) pour BGP. Relisez la documentation de BGP pour trouver comment faire. Chaque routeur doit assigner une communauté de format *[Numéro d'AS]:[Numéro de Routeur]* exactement comme à l'étape précédente.

   Exemple : Routeur R3

```
ip bgp-community new-format
!
route-map community-tag permit 10
 set community 10:3
!
router bgp 10
 no synchronization
 network 10.10.0.128 mask 255.255.255.192 route-map community-tag
 neighbor 10.20.15.1 remote-as 20
 neighbor 10.20.15.1 send-community
 no auto-summary
!
ip route 10.10.0.128 255.255.255.192 null0
```

Vérifiez que le routeur apparaisse bien avec sa communauté dans la table de routage BGP.

Q: Pourquoi est-ce que les peers externes, mais pas les internes, sont-ils les seuls à voir la communauté configurée sur le réseau ?
A: Voir précédemment. Tous les peerings nécessitent la sous-commande BGP send-community pour pouvoir propager l'attribut de communauté à ses peers.

9. **Communautés sur les peerings BGP internes.** Comme pour l'étape précédente, paramétrez maintenant les peerings internes pour que l'attribut de communauté de votre réseau soit envoyé aux peers locaux.

   Astuce : ajoutez simplement la ligne de configuration *neighbor x.x.x.x send-community* pour tous les peerings iBGP. N'oubliez pas de rafraîchir les sessions de peering BGP pour que les changements de configuration soient implémentés.

Point de contrôle #5 : *appelez l'assistant et montrez comment la communauté a été ajoutée en utilisant la commande show ip bgp. Montrez également que vous pouvez voir les communautés configurées par vos peers internes et externes.*

10. **Configuration du filtre de préfixe entrant basé sur l'attribut de communauté.** L'objectif ici est d'accepter seulement les réseaux reçus par le peering BGP externe du voisin. (C'est similaire à ce qui était fait aux étapes 2 et 4 avec les filtrages par préfixe ou AS path). Par exemple, R13 devrait seulement accepter les réseaux dont R1 est origine, et devrait utiliser l'information de communauté que R1 a ajouté au réseau pour y parvenir.

    Exemple : Routeur R13

```
ip community-list 3 permit 10:1
!
route-map infilter permit 10
 match community 3
!
router bgp 40
 neighbor 10.10.15.13 route-map infilter in
!
```

    Le choix du numéro de liste de communauté (community-list) revient à chaque groupe. En effet, il n'est annoncé dans aucun peering BGP ni utilisé de la moindre façon à part pour identifier la liste de communauté (de façon similaire au numéro d'access-list).

11. **Mise en place de l'attribut local-preference sur les routes eBGP reçues.** Dans cet exemple, une préférence locale va être mise en place pour les routes correspondant au filtre de communauté mis en place à l'étape 10. Conservez la route-map utilisée à l'étape 10 - une ligne de configuration supplémentaire va y être ajoutée. Il faut également permettre les autres réseaux reçus par les filtres et dont la préférence locale a la valeur par défaut.

    Q. Pourquoi ?
    A. Sans la seconde directive, la route-map implémente une directive deny par défaut et aucun autre préfixe n'est autorisé.

Exemple :

```
route-map infilter permit 10
 match community 3
 set local-preference 120
!
route-map infilter permit 20
```

Souvenez-vous qu'après la définition d'une nouvelle politique, la session BGP doit être rafraichie pour que cette nouvelle politique puisse être mise en oeuvre. Comme le routeur ne conserve pas systématiquement toutes les mises à jour reçues de son peer, cette étape est nécessaire. Vous pouvez dans ce but utiliser la commande exec "***clear ip bgp*** <*peer address*> ***in***".

Point de contrôle #6 : *appelez l'assistant et montrez que les routes dont vos peers eBGP sont origines ont maintenant une préférence locale de 120. Montrez également que les autres routes ont la préférence par défaut de 100.*

Groupes de peering BGP (peer-groups)

12. **Configuration de la fonctionnalité de groupes de peering pour les peers iBGP.** Les groupes de peering BGP aident a réduire la charge machine du routeur en générant et propageant des mises à jour vers les peers qui ont la même politique. Cette étape configure des groupes de peering BGP for les peers iBGP dans chaque AS. Remplacez la configuration individuelle de chaque peer iBGP par une configuration de groupe de peering, comme dans l'exemple ci-dessous.

Exemple : Routeur R9

```
router bgp 30
 neighbor ibgp-peers peer-group
 neighbor ibgp-peers description Groupe de peering utilise pour iBGP
 neighbor ibgp-peers remote-as 30
 neighbor ibgp-peers update-source loopback 0
 neighbor ibgp-peers send-community
 neighbot ibgp-peers next-hop-self
 neighbor ibgp-peers password cisco
!
```

Note : l'ancienne configuration antérieure au groupe de peering doit être retirée avant de convertir vers la configuration utilisant les groupes de peering.

```
router bgp 30
 no neighbor 10.30.15.224
 no neighbor 10.30.15.226
!
 neighbor 10.30.15.224 peer-group ibgp-peers
 neighbor 10.30.15.226 peer-group ibgp-peers
!
```

Q: Quels sont les avantages procurés par les groupes de peering ?

A: Les groupes de peering BGP permettent d'utiliser une configuration commune sur plusieurs peers BGP. L'application la plus courante est pour iBGP. Tous les peers BGP internes dans un

réseau d'un même FAI ont tendance à avoir la même relation les uns avec les autres. Plutôt que de définir une configuration conséquente par peer, et d'avoir à changer chaque peering quand il faut faire des modifications, la configuration peut être définie dans un groupe de peering, et seul le groupe doit être changé pour modifier la configuration de tous les peers iBGP. Cela réduit considérablement la surcharge de travail induite par toute modification, la charge processeur du routeur et simplifie grandement les configurations et leur visualisation.

Il est fortement recommandé d'utiliser la sous-commande peer-group de façon systématique pour la configuration de peers BGP. Comme dit précédemment, la plupart des peers iBGP ont la même configuration, il est donc très important de simplifier les configurations pour tous les intervenants du réseau. De plus, une configuration qui utilise beaucoup de groupes de peering est habituellement plus lisible qu'une configuration ayant une section distincte par peer, en particulier dans les réseaux ayant de nombreux peers.

Note : à l'avenir dans ce workshop, toute configuration devra être faite avec des groupes de peering autant que possible (surtout pour les configurations iBGP).

13. **Résumé :** Dans ce module, nous avons découvert certaines des fonctionnalités de configuration des peerings BGP disponibles dans l'IOS Cisco. Nous encourageons le lecteur à essayer toutes sortes de permutations des exemples donnés au cours du TP. L'utilisation des communautés est en train de gagner en popularité maintenant que la fonctionnalité est reconnue pour son utilité dans le contrôle des politiques de routage entre différents AS. Les rafraîchissements de route et les groupes de peering sont également beaucoup utilisés dans les backbones de FAI car ils facilitent énormément l'administration et la configuration d'un réseau opérationnel.

Questions de revisions :

1. Pourquoi l'utilisation des rafraichissements de routes est-elle la meilleure méthode d'implémentation des nouvelles politiques BGP ?

2. Pourquoi l'utilisation des filtres AS-PATH fournit-elle moins de granularité queles filtres de préfixes pour une session BGP ? Quelle est la meilleure solution pour un réseau de FAI, et pourquoi ?

3. Quand faut-il mettre l'attribut de communauté sur un préfixe BGP ?

4. Est-ce que l'IOS envoie les communautés BGP par défaut pour iBGP ? et pour eBGP ? Dans la négative, de quoi les opérateurs doivent-ils se souvenir ?