

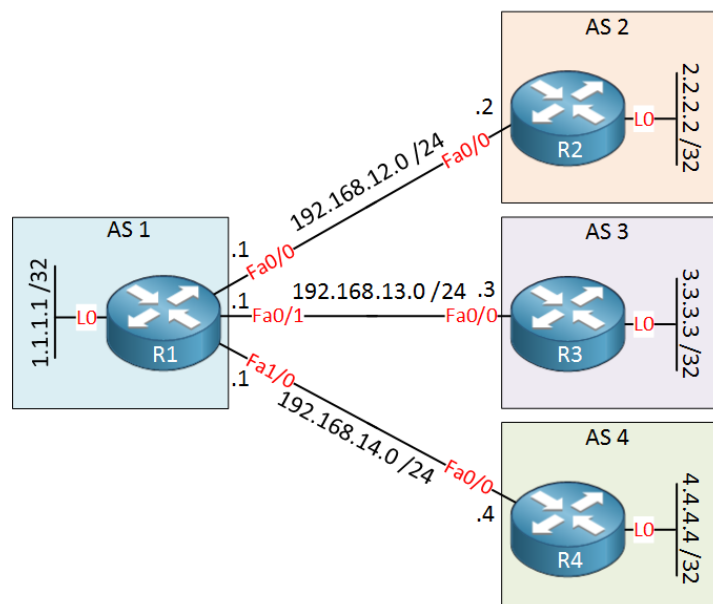
## TP2 : BGP Peer Groups, RR et AS Path filter

**Objectif :** L'objectif de ce TP est de vous montrer des moyens de simplifier la configuration BGP, notamment lorsque le nombre de routeur à configurer augmente. Aussi ce TP fera l'objet d'une synthèse des services MPLS-VPN et MPLS-TE

### Partie 1 : peer groups.

Lorsque vous configurez BGP sur un routeur, il est possible que certains des voisins BGP partagent exactement la même configuration. Cela peut être ennuyeux car vous devez taper exactement les mêmes commandes pour chacun de ces voisins. De plus, lorsque BGP prépare des mises à jour, il le fait séparément pour chaque voisin. Cela signifie qu'il doit utiliser des ressources CPU pour préparer la mise à jour pour chaque voisin.

Pour simplifier la configuration de BGP et réduire le nombre de mises à jour que BGP doit créer, nous pouvons utiliser des groupes de pairs. Nous pouvons ajouter des voisins à un groupe de pairs, puis appliquer toutes nos configurations au groupe de pairs. BGP préparera les mises à jour pour le groupe de pairs qui nécessite moins de ressources CPU que de les préparer pour chaque voisin séparément.



Ci-dessus, nous avons 4 routeurs dans différents systèmes autonomes. R1 est connecté à R2, R3 et R4. Disons que nous avons les exigences suivantes pour ces voisins eBGP :

- Utilisez le multi-saut eBGP avec la commande `ebgp-multihop`
- Créer les session ebgp en utilisant l'interface de bouclage.
- Faire en sorte que le routeur change le next-hop
- Définissez la métrique par défaut (MED) sur 2323.

Voici à quoi ressemblerait la configuration BGP sur R1 sans utiliser la méthode avec "peer group" :

```
R1 (config) #router bgp 1
R1 (config-router) #neighbor 2.2.2.2 remote-as 2
R1 (config-router) #neighbor 3.3.3.3 remote-as 3
R1 (config-router) #neighbor 4.4.4.4 remote-as 4
R1 (config-router) #neighbor 2.2.2.2 update-source loopback 0
R1 (config-router) #neighbor 3.3.3.3 update-source loopback 0
R1 (config-router) #neighbor 4.4.4.4 update-source loopback 0
R1 (config-router) #neighbor 2.2.2.2 ebgp-multihop 2
R1 (config-router) #neighbor 3.3.3.3 ebgp-multihop 2
R1 (config-router) #neighbor 4.4.4.4 ebgp-multihop 2
R1 (config-router) #neighbor 2.2.2.2 next-hop-self
R1 (config-router) #neighbor 3.3.3.3 next-hop-self
R1 (config-router) #neighbor 4.4.4.4 next-hop-self
R1 (config-router) #neighbor 2.2.2.2 route-map SET_MED out
R1 (config-router) #neighbor 3.3.3.3 route-map SET_MED out
R1 (config-router) #neighbor 4.4.4.4 route-map SET_MED out
```

Pas mal de lignes ! Simplifions la configuration de R1 en définissant un groupe de pairs. Nous devons d'abord préciser le numéro AS pour chaque voisin eBGP séparément :

```
R1 (config) #router bgp 1
R1 (config-router) #neighbor 2.2.2.2 remote-as 2
R1 (config-router) #neighbor 3.3.3.3 remote-as 3
R1 (config-router) #neighbor 4.4.4.4 remote-as 4
```

Nous pouvons maintenant créer le groupe de pairs. Si vous regardez la commande « `neighbor` », vous verrez quelques options :

```
R1 (config-router) #neighbor ?  
  A.B.C.D      Neighbor address  
  WORD        Neighbor tag  
  X:X:X:X::X   Neighbor IPv6 address
```

Nous pouvons spécifier une adresse IPv4 ou IPv6 pour le voisin ou nous pouvons utiliser une étiquette. C'est ce que nous allons utiliser pour définir le groupe de pairs :

```
R1 (config-router) #neighbor ALL_R peer-group
```

Ici on a appelé le groupe de pairs par **ALL\_R**. L'étape suivante consiste à ajouter les voisins à ce groupe de pairs :

```
R1 (config-router) #neighbor 2.2.2.2 peer-group ALL_R  
R1 (config-router) #neighbor 3.3.3.3 peer-group ALL_R  
R1 (config-router) #neighbor 4.4.4.4 peer-group ALL_R
```

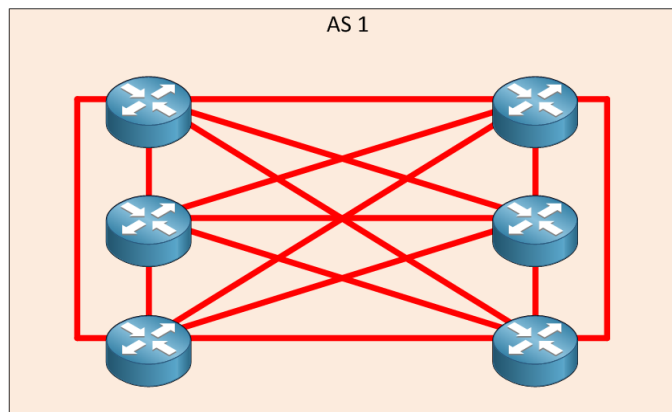
C'est tout ce que vous avez à configurer. Tout le reste que vous souhaitez configurer peut être appliqué au groupe de pairs au lieu de l'appliquer directement au voisin :

```
R1 (config-router) #neighbor ALL_R update-source loopback 0  
R1 (config-router) #neighbor ALL_R ebgp-multihop 2  
R1 (config-router) #neighbor ALL_R ebgp next-hop-self  
R1 (config-router) #neighbor ALL_R route-map SET_MED out
```

Ces quatre commandes sont maintenant appliquées à R2, R3 et R4 grâce à notre peer group. Cela nous évite de taper plusieurs commandes sur les routeurs et ces derniers nécessiteront moins de cycles CPU pour les mises à jour BGP.

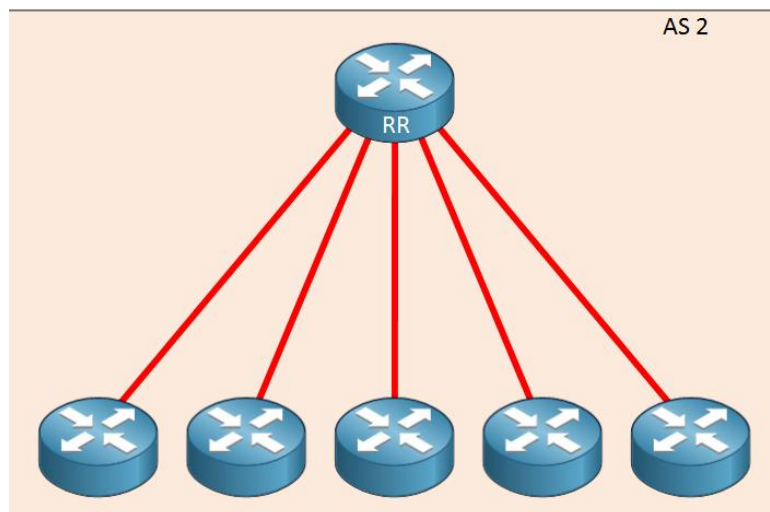
## Partie 2 : Route Reflector

Vous avez vu dans le S3 qu'avec iBGP, chaque routeur doit former un lien de peering avec tous les autres nœuds iBGP de l'AS afin d'assurer des sessions BGP partout dans l'AS (full mesh).



full mesh  $\rightarrow N*(N-1)/2$  sessions !

Or il est possible que chaque routeur met en place qu'une seule session BGP avec le routeur qui jouera le rôle de route-reflector (RR).



Résultat obtenu avec route reflector (RR)

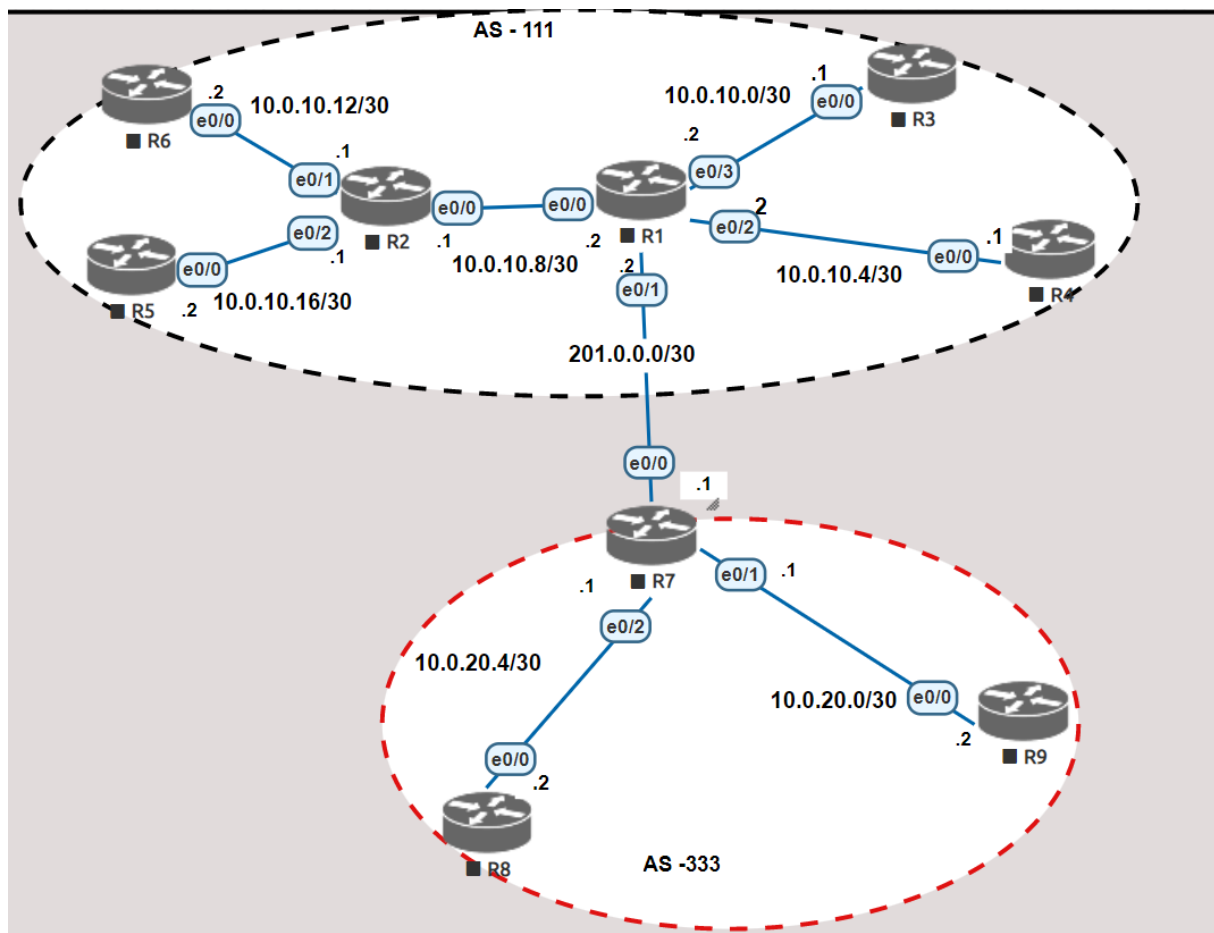
Ainsi, quand un routeur iBGP annonce des préfixes vers le RR, ils seront « reflétés » vers tous les autres routeurs iBGP. Cela simplifie beaucoup la configuration iBGP.

Pour la mise en place d'un RR il suffit juste de rajouter la commande suivante dans la configuration BGP au niveau du routeur qui va jouer ce rôle et cela pour chaque voisin : **neighbor x.x.x.x route-reflector-client**

## Travail à faire :

Récupérer la maquette présente sur moodle puis réaliser la configuration suivante :

1. L'ensemble des routeurs de chaque AS doivent être configurés avec du OSPF
2. Chaque routeur possède une interface de loopback avec l'adresse IP X.X.X.X/32 avec X le numéro du routeur
3. Configurer du BGP au sein de chaque AS. Les routeurs R1 et R7 jouent le rôle de RR. Vous utilisez la méthode avec le peer-group pour simplifier votre configuration
4. Bien évidemment la communication entre R1 et R7 passe avec du eBGP.
5. Sur le routeur R6 et R9 vous rajoutez une interface lo1 avec l'adresse 58.58.58.1/28 et 57.57.57.1/28 respectivement pour simuler un réseau client qui va être partagé dans les annonces BGP.



## Question :

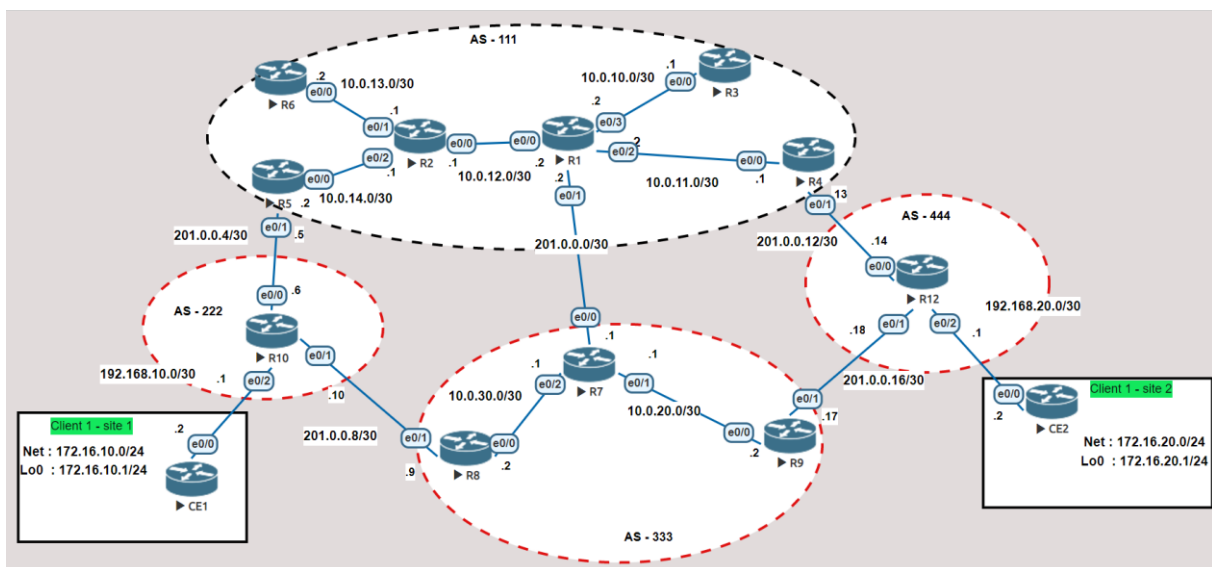
- 1- Combien de sessions BGP sont actives sur R1 ? et sur R2 ?

- 2- Est-ce que le R8 possède une entrée vers 58.58.58.0/28 dans sa table de routage ?
- 3- Est-ce que le R3 possède une entrée vers 57.57.57.0/28 dans sa table de routage ?
- 4- Un ping étendu entre R6 et R9 en utilisant les lo 1 doit fonctionner. Si ce n'est pas le cas apporter les corrections nécessaires.

🚦 Point de validation 1 : montrer le résultat du ping à votre intervenant de TP

### Partie 3 : AS Path filter.

A présent vous allez rajouter deux routeurs pour former les AS 222 et 444, auxquels un client présent sur deux sites est connecté. La topologie devient comme suit :



1. Mettez en places les sessions BGP nécessaire pour interconnecter R10 et R12 aux AS 111 et 333.
2. Est-ce que R10 et R12 possèdent des entrées vers 57.57.57.0/28 et 58.58.58.0/28 ? Si ce n'est pas le cas apporter les corrections nécessaires.
3. Les routeurs R10 et R12 possèdent combien de routes pour aller vers 57.57.57.0/28 et 58.58.58.0/28 ? Les routes passent par quels AS ?

4. Arrêter les interfaces e0/1 du R8 et e0/0 du R7 puis vérifier que R10 et R12 possèdent toujours des routes pour aller vers 57.57.57.0/28 et 58.58.58.0/28

✚ N.B pour une prise en charge rapide des nouveaux changements, appliquer la commande **clear ip bgp \* soft** sur les routeurs impactés par le changement.

5. Remettez les interfaces e0/1 du R8 et e0/0 du R7 en marche puis appliquer un filtrage de telle sorte que les AS 222 et 444 n'acceptent que les préfixes venant directement de l'AS 111.

✚ **Point de validation 2 : montrer le résultat de la commande 'show ip bgp' appliquée sur R10 et R12 à votre intervenant de TP**

## Partie 4 : MPLS-VPN

1. Vous allez à présent configurer une session MPLS-VPN entre les deux sites du client. Une première session passera par l'AS333 et une 2<sup>e</sup> passera par l'AS 111. Donner les étapes et la configuration à mettre en place.
2. Assurez-vous qu'un ping étendu entre CE1 et CE2 passe sans problème. Faites un traceroute et noter le chemin emprunter.
3. Arrêter le lien entre R10 et R8 puis vérifier que le ping étendu entre CE1 et CE2 fonctionne toujours. Faites un traceroute et noter le chemin emprunter.

✚ **Point de validation 3 : montrer le résultat du traceroute à votre intervenant de TP**

## Partie 5 : MPLS-TE

À présent vous allez rajouter le service MPLS-TE au niveau de l'AS 111.

1. Configurer un tunnel MPLS-TE entre R5 et R4 qui respecte les exigences suivantes :
  - Le calcul du chemin se fera d'une manière dynamique, ainsi le tunnel sera établi selon le meilleur chemin obtenu par le protocole IGP.

- La priorité de l'installation et du maintien du tunnel sera de 2 et 2 respectivement.
- La contrainte à satisfaire pour ce tunnel sera d'avoir une bande passante de 5 Mbps. La bande passante réservée au niveau des routeurs est de 7 Mbits.
- Le transfert du flux dans le tunnel va utiliser l'option autoroute avec la variante Forwarding adjacency

La communication entre les deux sites du client passe dans notre cas par l'AS 333. Or on souhaite influencer ce choix pour passer par l'AS 111 étant donné qu'on a un tunnel MPLS-TE réservé dessus. Pour cela on va modifier le poids des chemins vers 172.16.20.0/24 sur R10 ainsi que pour 172.16.10.0/24 sur R12. Pour rappel, le chemin avec le poids le plus élevé c'est celui qui sera utilisé par BGP.


Voici un exemple de modification du poids des routes BGP :

```
route-map POIDS permit 1
  match ip address YY
  set weight valeur

router bgp XX
address-family vpnv4
neighbor X.X.X.X route-map POIDS out
```

Ici on définit un route-map qui modifie le poids des entrées qui vérifient les préfixes définis dans la liste YY. Puis cette route-map est associée au voisin BGP soit pour le flux entrant (in) ou le flux sortant (out)

2. Appliquer une modification des poids des routes de telle sorte que la communication MPLS-VPN entre les deux sites du client passe par le tunnel TE.

 **Point de validation 4 : montrer le résultat des commandes suivantes à votre intervenant de TP :**

- 'show bgp vpnv4 unicast all' appliquée sur R10 et R12,
- 'show mpls traffic-eng tunnels brief' appliquée sur R5.
- 'show mpls forwarding-table' sur R5
- 'traceroute 172.16.20.1 source lo0' appliquée sur CE1.



