Master 2 RSH

TP4 ADMI

Routage Dynamique

OSPF et BGP

Encadré par:
Ahmad FADEL

Réalisé par:
Fanny PRIEUR / **Lorenzo MAZZOCCHI**

Année 2018/2019

Table des matières

I - Introduction et But du TP 4

Ce TP a pour but l'étude de la configuration automatique du routage dans un réseau grâce au protocole de routage inter-domaine : ***OSPF*** (*Open Shortest Path First*) basé sur l'état de lien. Par la suite, nous connecterons ***OSPF*** avec ***BGP*** (*Border Gateway Protocol*), protocole de d'échange de route externe, pour faire du routage inter-***AS*** (*Autonomous System*).

OSPF : Le protocole **OSPF** (Open Short Path First) a été défini par **IETF** pour résoudre les problèmes posés par l'utilisation de RIP et entre autres le temps de convergence.

Actuellement, ce temps de convergence est environ d'une minute. Ce protocole est beaucoup plus complexe que RIP.

OSPF présente des caractéristiques importantes :

- C'est un protocole ouvert (pas de copyright).
- Il utilise l'algorithme **SPF** (Short Path First) dans ses calculs de route pour déterminer le plus court chemin.
- Principe d adjacence : deux routeurs sont dits adjacents s'ils ont synchronisé leurs bases de données topologiques.
- Le protocole OSPF utilise une base de données distribuées qui permet de garder en mémoire l'état des liaisons.
- Ces informations forment une description de la topologie du réseau et de l'état de l'infrastructure.
- OSPF est un protocole de routage intra-domaine, c'est-à-dire qu il ne diffuse les informations de routage qu'entre les routeurs appartenant à un même système autonome (un ensemble de réseaux qui utilisent un protocole de routage commun et qui dépend d'une autorité d'administration unique).

BGP : Le protocole **BGP** (Border Gateway Protocol) est un protocole d'échange de route externe (un EGP Exterior Gateway Protocol), utilisé notamment sur le réseau Internet. Son objectif principal est d'échanger des informations de routage et d'accessibilité de réseaux (appelés *préfixes*) entre ***Autonomous Systems*** (AS).

Les objectifs principaux de BGP sont les suivants :

- Échanger des routes (trafic) entre des organismes indépendants
    - Opérateurs
    - Fournisseurs de services Internet
- Implémenter la politique de routage de chaque organisme
    - Respect des contrats passés entre les organismes
    - Sûreté de fonctionnement
- Minimiser le trafic induit sur les liens
- Donner une bonne stabilité au routage

AS: Les **systèmes autonomes** (autonomous system) est un ensemble de réseaux informatiques *IP* intégrés à Internet et dont la politique de routage interne (routes à choisir en priorité, filtrage des annonces) est cohérente.

Un *AS* est généralement sous le contrôle d'une entité ou organisation unique, typiquement un fournisseur d'accès à Internet.

Un As c'est :

- Un ensemble de routeurs interconnectés
    - Gérés par une seule unité administrative
    - Présentant une image cohérente des réseaux destinations qu'ils peuvent permettre d'atteindre
- Tout système autonome sera identifié par un numéro
    - AS number défini sur 16 bits (32 bits dans un futur proche)
    - Exemple : AS702 (cf www.cidr-report.org/autmons.html)
    - Remarque : Certains domaines ne possèdent pas de numéro d'AS (petit domaines n'utilisant pas BGP)

Comme pour le TP sur ***RIP***, nous utiliserons les routeurs ***Cisco*** et le logiciel ***Quagga***.

GNU Zebra est une suite de logiciels de routage. Il prend en charge plusieurs protocoles et permet de transformer une machine Unix en routeur. ***Quagga*** est une suite de logiciels de routage implémentant les protocoles ***OSPF*** (v2 & v3), ***RIP*** (v1, v2 & v3), ***BGP*** (v4) et ***IS-IS*** pour les plates-formes de type Unix. Tout comme GNU Zebra, il permet de transformer une machine Unix en routeur.

Quagga est un fork du projet ***GNU Zebra*** (inactif depuis 2005).

Pour ce TP, nous nous concentrerons uniquement sur la configuration des routeurs. Nous n'utilisons pas les machines Windows.

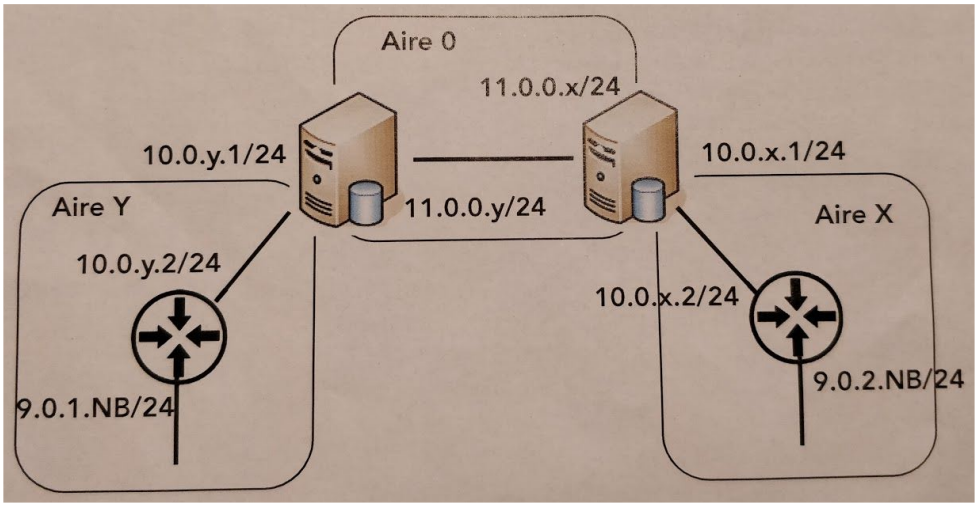

Fig.1.L'architecture à mettre en place pour réaliser la première partie du TP.

II - Mise en place de l'architecture

Nous allons mettre en place la configuration vue en Fig.1 dans l'introduction.

A.Configuration préliminaire

- Sur la machine Linux, désactivez toutes les interfaces *eth* de la machine. Vérifiez qu'il n'y a pas de routes. Configurez les interfaces *eth1* (reliant les Linux entre eux), *eth2* (reliant les Linux aux routeurs) respectivement avec les adresses 11.0.0.X/24, 10.0.X.1/24 où X est votre numéro de machine.

La commande permettant de désactiver les différentes interfaces est la suivante :

ifconfig <nom_interface> down

Exemple : ***ifconfig eth0 down***

Pour confirmer que les interfaces sont ***down*** nous pouvons lancer la commande ***ifconfig***. Nous pouvons voir que la seule interface d'activité est la ***lo.***

```
root@localhost:~# ifdown eth0
Killed old client process
Internet Systems Consortium DHCP Client 4.3.1
Copyright 2004-2014 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/78:ac:c0:af:bc:8f
Sending on   LPF/eth0/78:ac:c0:af:bc:8f
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 148.60.12.253 port 67
root@localhost:~# ifdown eth1
ifdown: interface eth1 not configured
root@localhost:~# ifdown eth2
ifdown: interface eth2 not configured
root@localhost:~# ifdown eth3
ifdown: interface eth3 not configured
root@localhost:~# ifdown eth4
ifdown: interface eth4 not configured
root@localhost:~# ifconfig
lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:2104 (2.0 KiB)  TX bytes:2104 (2.0 KiB)
```

Fig.2.Exécution de la commande ifdown et vérification de la désactivation des interfaces avec la commande ifconfig

Pour vérifier que notre machine Linux ne possède aucune route, nous utilisons la commande suivante : ***route -n****.*

Afin de répondre à la topologie demandée, nous avons effectué les commandes suivantes sur nos interfaces situées sur la **machine Linux** et la **machine voisine** :

Sur notre machine :	***ifconfig eth1 11.0.0.3 netmask 255.255.255.0***
	ifconfig eth2 10.0.3.1 netmask 255.255.255.0
Sur la machine voisine :	***ifconfig eth1 11.0.0.4 netmask 255.255.255.0***
	ifconfig eth2 10.0.4.1 netmask 255.255.255.0

```
root@localhost:~# ifconfig eth1 11.0.0.3 netmask 255.255.255.0
root@localhost:~# ifconfig eth2 10.0.3.1 netmask 255.255.255.0
root@localhost:~# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:0d:88:70:dc:d8
          inet adr:11.0.0.3  Bcast:11.0.0.255  Masque:255.255.255.0
          adr inet6: fe80::20d:88ff:fe70:dcd8/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:37
          collisions:0 lg file transmission:1000
          RX bytes:0 (0.0 B)  TX bytes:6590 (6.4 KiB)

eth2      Link encap:Ethernet  HWaddr 00:0d:88:70:dc:d9
          inet adr:10.0.3.1  Bcast:10.0.3.255  Masque:255.255.255.0
          adr inet6: fe80::20d:88ff:fe70:dcd9/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:33
          collisions:0 lg file transmission:1000
          RX bytes:0 (0.0 B)  TX bytes:5992 (5.8 KiB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:26 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:2104 (2.0 KiB)  TX bytes:2104 (2.0 KiB)
```

Fig.3.Détail de l'adressage de notre machine Linux sur les interfaces eth1 et eth2.

III - OSPF

OSPF est un protocole de routage dynamique défini par l'***IETF*** à la fin des années 80, en 1987 pour être exact. Ce protocole a deux principales caractéristiques :

- Il est ouvert, d'où le terme *Open* de ***OSPF***
- Il utilise l'algorithme du plus court chemin ou ***Dijkstra***

N.B: Pour plus de détails, merci de consulter l'introduction.

A.Configuration de OSPF sur un PC Linux (Zebra)

Rappels :

- ❏ Pour lancer Quagga : **/etc/init.d/quagga start**
- ❏ Mot de passe des démons **zebra** et **ospfd** est "**zebra**".
- ❏ Edition du fichier /**etc/quagga/daemons** en modifiant **zebra=yes** et **ospfd=yes**.

Comme pour ***RIP***, ***Quagga*** implémente le protocole ***OSPF***. Le pc Linux sera configuré comme routeur, ce qui nous permettra de voir les informations OSPF échangées entre les routeurs. Il est important de noter que ce **Quagga** appartient au **backbone**.

1. Configurez les interfaces *eth1* et *eth2* comme précédemment dans ***Zebra***.

Afin de répondre à la topologie, nous avons effectués les commandes suivantes sur notre machine :

- ***telnet localhost 2601*** : Permet de nous connecter à ***zebra*** qui est en écoute sur le port 2601.
- ***enable*** : Pour passer en mode administrateur sur ***zebra***.
- ***configure terminal*** : Pour passer en mode de configuration de notre routeur Zebra.
- ***interface ethX*** : Permet d'entrer dans la configurer de l'interface **eth** où **X** est le numéro de l'interface à modifier.
- ***ip address <ip>/24*** : Permet de saisir l'adresse IP et le masque (écrit sous format CIDR) de l'interface à configurer, ici le masque **/24** correspond à **255.255.255.0**. **<ip>** correspond à l'IP souhaitée.
- ***write file*** : Permet de sauvegarder l'intégralité des modifications faite dans le fichier de configuration **zebra.conf**.

Fig.4.Résultat des commandes décrites plus haut permettant de configurer le fichier zebra.conf

2. Pour que le noyau Linux prenne en compte le relais des paquets entre ces interfaces, il faut activer l'option ***ip_forward***.

La commande a utiliser est : ***echo "1" > /proc/sys/net/ipv4/ip_forward***
Cette commande écrit le chiffre 1 dans le fichier ip_forward. Cette commande permet d'activer l'IP forwarding en mettant la valeur 1. Sinon, l'IP Forwarding est désactivé par défaut avec la valeur 0.

Cette commande sert sous linux à activer les fonctions de routage entre les interfaces réseau du système. Cela n'a de sens que dans le cas d'une machine possédant plusieurs interfaces réseau. La commande autorise le système à rediriger un paquet de données arrivé par une interface réseau vers une autre interface réseau, conformément à la table de routage du système.

3. Faite en sorte que ***Quagga*** lance ***zebra*** et ***ospfd*** avec ***init.d***.

Pour vérifier que ***Quagga*** exécute bien ***zebra*** et ***ospfd*** lors de son lancement, nous devons vérifier les données écrites dans le fichier **daemons** situé sous l'arborescence **/etc/quagga/daemons**. Il faut que les deux paramètres **zebra** et **ospfd** soient définis à "**yes**".

4. Connectez-vous à ***ospfd***.

Pour connaître le port de **ospfd**, nous devons utiliser la commande suivante : ***netstat -antp | grep zebra***.

```
root@localhost:/etc/quagga# netstat -antp | grep ospfd
tcp        0      0 127.0.0.1:2604          0.0.0.0:*               LISTEN      1576/ospfd
```

Fig.5. Résultat de la commande permettant de connaître le port de ospfd.

Le port d'écoute de ***ospfd*** est le port **2604**. Connaissant le port à utiliser, nous pouvons nous connecter en exécutant la commande suivante : ***telnet localhost 2604***.

5. Expliquez brièvement le rôle de ces commandes.

- **« configure terminal »** : Passer en mode administrateur sur le routeur.
- **« router ospf »** : Cette commande active le mode ***OSPF***.
- **« network 11.0.0.3/24 area 0»** : Adresse réseau correspondant à l'interface *eth1* et à l'adresse IP 11.0.0.3/24 sur l'area 0.
- **« network 10.0.3.0/24 area 3»** : Adresse réseau correspondant à l'interface *eth2* et à l'adresse IP 10.0.3.0/24 sur l'area 3 composant le routeur physique *Cisco*.
- **« redistribute connected »** : Commande permettant de définir ***OSPF*** pour qu'il propager ces routes statiques.
- **« write file »** : Commande pour sauvegarder les configurations faites.

6. Quels sont les paquets ***OSPF*** transmis ? Y'a-t-il des réponses ? Quelle est l'adresse de destination et pourquoi ?

Les paquets transmis sont de type **"Hello Packet".** Périodiquement, des paquets **LS Update** et **LS Acknowledge** sont envoyés. L'adresse de destination est l'adresse IP **224.0.0.5** qui est une **adresse multicast** dans le protocole ***OSPF***.

7. ***Zebra*** possède des commandes pour visualiser la base de données de topologie ***OSPF*** ainsi que celles de ces voisins. Ces deux commandes doivent être lancées à la racine.

Afin de visualiser la base de donnée de topologie d'***OSPF***, nous utilisons la commande disponible dans ***Zebra***, ***show ip ospf database***.

```
ospfd# show ip ospf datab
ospfd# show ip ospf database

       OSPF Router with ID (11.0.0.3)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
11.0.0.3        11.0.0.3        362 0x80000004 0xac6f 1

                Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.0.3.0        11.0.0.3        373 0x80000001 0xb681 10.0.3.0/24

                Router Link States (Area 0.0.0.3)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
11.0.0.3        11.0.0.3        333 0x80000004 0xc059 1

                Summary Link States (Area 0.0.0.3)

Link ID         ADV Router      Age  Seq#       CkSum  Route
11.0.0.0        11.0.0.3        373 0x80000001 0xca6f 11.0.0.0/24
```

Fig.6. Résultat de la commande show ip ospf database

Pour visualiser la topologie voisine, il faut lancer la commande ***show ip ospf neighbor***. Bien évidemment ces deux commandes doivent être lancer à la racine.

```
ospfd# show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface                RXmtL RqstL DBsmL
```

Fig.7. Résultat de la commande show ip ospf neighbor

Dans les figure 6 et 7, nous sommes censé voir notre voisin dans les résultats. Notre voisin est 11.0.0.4.

B.OSPF sur le routeur Cisco

Il faut à présent configurer le routeur Cisco pour qu'il implémente le protocole OSPF. Chaque interfaces doit alors appartenir à une **area différente**.

1. Configurez les interfaces **GigaEthernet 0/0** avec l'adresse **10.0.x.2/24** (reliant le routeur au switch, donc au Quagga voisin) et l'interface **GigaEthernet 0/1** avec l'adresse **9.0.2.0/24** pour le poste x (poste 4) et **9.0.1.0/24** pour le poste y (poste 3). Voici la liste des commandes utilisées :

- **minicom** : Accéder au terminal du routeur *Cisco*.
- **enable** : Passer en mode administrateur.
- **configure terminal** : Passer en mode configuration. Uniquement possible en tant qu'administrateur.
- **interface gigaethernet 0/0** : Modification de l'interface **GigaEthernet 0/0**.
- **ip address 10.0.3.2 255.255.255.0** : Donne cette adresse à l'interface
- **no shutdown** : Pour activer l'interface. Par défaut, les interfaces des routeurs sont éteintes.
- **exit** : Sortir de l'interface **GigaEthernet 0/0**.
- **interface gigaethernet 0/1** : Configurer l'interface GigaEthernet 0/1.
- **ip address 9.0.1.0 255.255.255.0** Donne cette adresse à l'interface
- **no shutdown** : Pour activer l'interface. Par défaut, les interfaces des routeurs sont éteintes.

2. Lancer Wireshark sur l'interface **eth2** du pc Debian.

Ici, nous n'avions besoin que de lancer Wireshark sur l'interface eth2 de notre machine Debian afin de commencer les captures des paquets OSPF qui seront demandé par la suite.

3. Configuration de la prochaine area

- **configure terminal**: Passer en mode administrateur.
- **router ospf x** : Activation du processus x du routage ***OSPF***. Exemple : ***router ospf 3*** créera un processus ID 3 pour le protocole ***OSPF***.
- **network 10.0.x.0 255.255.255.0 area x** : Réseau sur lequel on souhaite activer le routage dynamique en indiquant la zone rattaché au réseau.
- **network 9.0.NB.0 255.255.255.0 area x** : Réseau 9.0.NB.0 est rattaché à l'aréa x.
- **redistribute connected** : Pour redistribuer des routes d'un domaine de routage vers un autre domaine de routage.

4. Quels sont les paquets OSPF transmis ? Y a t’il des réponses ?

Les paquets transmis à cet instant sont les suivants :

- **DB Description** : Décrit le contenu des bases de données d'état de liens (*link-state database*) des routeurs OSPF.
- **Hello Packet** : Établit et maintient les informations de contiguïté (adjacency information) avec les voisins.
- **LS Update** : Transporte les *link-state advertisements*, les LSA, aux routeurs voisins.
- **LS Acknowledge** : Accusés de réception des LSA des voisins.

    - **Type 1 - Router LSA** - Le routeur annonce sa présence et liste les liens vers les autres routeurs ou réseaux dans la même zone, avec leur métrique. Les LSAs Type 1 sont envoyés par inondation seulement au sein de la zone.

    - **Type 2 - Network LSA** - Le DR (*designated router*) envoie sur un segment multi-accès (comme Ethernet) la liste des routeurs qui sont sur le même segment. Ils ne sont propagés uniquement au sein de la zone. On y trouve l'adresse du DR.

    - **Type 3 - Summary LSA** - Un ABR envoie des informations résumées d'une autre zone.

    - **Type 4 - ASBR-Summary LSA** - Ils ajoutent une information de l'ABR concernant une route externe propagée par inondation dans toutes les zones de type 5 External LSA.

    - **Type 5 - External LSA** - Ce type de LSA contient des informations injectées dans OSPF via un autre processus. Ils sont envoyés par inondation dans toutes les zones (sauf les zones stub et NSSA).

    - **Type 6 - Group Membership LSA -** Le LSA type 6 n'est pas supporté par les routeurs Cisco. Il est défini pour MOSPF.

    - **Type 7 -** Les routeurs dans une **Not-so-stubby-area (NSSA)** - Le LSA type 7 est généré par un routeur ASBR pour remplacer le LSA de type 5 dans le cas où vous disposez d'une aire *NSSA* ou *totally NSSA*. Les routes seront donc connues comme NSSA externe de type 1 ou 2. Dans votre table de routage, vos routes seront précédées de **O N1** et **O N2**.

    - **Type 8** - Un LSA uniquement avec lien local pour OSPFv3. Un LSA de type 8 est utilisé pour donner des informations sur les adresses de liens locaux et une liste d'adresses IPv6 sur le lien.

    - **Type 9** - Un LSA "opaque" à lien-local (défini par la norme RFC 2370) dans OSPFv2 et le LSA préfixe intra-zone dans OSPFv3. C'est le LSA OSPFv3 qui contient les préfixes.

    - **Type 10** - Un LSA "opaque" à zone locale tel que défini par la RFC 2370. Les LSA opaques contiennent des informations qui devraient être inondées par d'autres routeurs même si le routeur n'est pas en mesure de comprendre les informations étendues. En règle générale, les LSA de type 10 sont utilisés pour les extensions d'ingénierie du trafic vers OSPF, inondant les informations supplémentaires relatives aux liens au-delà de leur métrique, telles que la bande passante et la couleur des liens.

    - **Type 11** - Un AS "opaque" LSA défini par la RFC 5250. Similaire au LSA Type 5.

- **LS Request** : Demande des éléments spécifiques des bases de données d'état de liens (*link-state database*) des routeurs ***OSPF***.

5. **Attendez que votre binôme soit au même point que vous.** Arrêtez Wireshark et donner la signification des paquets OSPF qui circulent entre le routeur et le PC.

Après arrêt de la capture Wireshark nous avons observés 5 types de paquets différents ***OSPF***, “Hello packet”, “DB description”, “LS Request”, “LS Update” et “LS Acknowledge”.

6. Quelle est la signification des paquets DB Desc, LSU et LSR ?

Voici la signification des paquets DB Description, Link-state Update et Link-state Request du protocole ***OSPF*** :

- **DB Description** : Décrit le contenu des bases de données d'état de liens (*link-state database*) des routeurs ***OSPF***.

Ces paquets sont utilisés au moment de l'état **ExStart**, les routeurs vont déterminer qui commence à envoyer les informations. Ici, le principe est d'établir une relation Maître/Esclave entre deux routeurs.

Le routeur qui déclare la plus haute ID (la priorité n'intervient pas) commencera et orchestrera l'échange en tant que maître.

Les routeurs sont maintenant prêts à s'engager dans le processus **Exchange**.

Le maître mène l'esclave à un échange de paquets **Database Description (DBDs)** qui décrivent la base de données de liens de chaque routeur dans les détails.

Ces descriptions comportent le type d'état de lien, l'adresse du routeur qui fait l'annonce, le coût du lien et un numéro de séquence.

- **LS Request** : Demande des éléments spécifiques des bases de données d'état de liens (*link-state database*) des routeurs ***OSPF***.

- **LS Update** : Transporte les *link-state advertisements*, les ***LSA***, aux routeurs voisins.

7. En interprétant ces paquets, déduisez les étapes de la création de la table de routage en OSPF ? Relevez l'état de la base de donnée topologie OSPF sur le routeur PC (Zebra) et le routeur. Sur le Cisco la commande est **show ip ospf database**. Que représentent les différentes tables affichés ?

Un routeur ***OSPF*** peut prendre en charge trois types d'opérations :
- opérations dans une zone,
- connexions inter-zones
- et connexions avec d'autres systèmes autonomes (***AS***).

Nous nous intéresserons à ce qui se passe dans une zone (area).

Fonctionnement dans une zone :
- Pour chaque zone, une table d'états de lien est construite et maintenue.
- La table de routage est construite à partir de cette base de données.
- Ce résultat est obtenu grâce à l'application de l'algorithme de routage ***OSPF***.

Étape 1 : *Découverte des voisins*
D'abord, l'interface d'un routeur doit trouver ses voisins et entretenir une relation avec chaque voisin L2. Il utilise des paquets ***Hello***. Dès son initialisation ou à la suite d'un changement dans la topologie, un routeur va générer un *link-state advertisement (LSA)*. Cette annonce va représenter la collection de tous les états de liens de voisinage du routeur.

Étape 2 : *Inondations et mises à jour*
Tous les routeurs de la zone (*area*) vont s'échanger ces états de liens par inondation (*flooding*). Chaque routeur qui reçoit des mises à jour d'état de lien (*link-state update*), en gardera une copie dans sa *link-state database* et propagera la mise à jour auprès des autres routeurs.

Étape 3 : *Calcul des routes*
Après que la base de données de chaque routeur a été complétée, chacun va calculer l'arbre du chemin le plus court (*Shortest Path Tree*) vers toutes les destinations avec l'algorithme de ***Dijkstra***. Il construira alors la table de routage (*routing table*), appelée aussi *forwarding database*, en choisissant les meilleures routes à inscrire.

Étape 4 : *Maintenance des routes*
S'il n'y a pas de modification topologique, ***OSPF*** sera très discret. Par contre en cas de changement, il y aura échange d'informations (par des paquets d'état de lien) et l'algorithme de ***Dijkstra*** calculera de nouveau les chemins les plus courts afin de les inscrire dans la table de routage.

On peut obtenir plus de détails en utilisant la commande **show ip ospf**. On peut préciser par exemple les informations pour chaque interface Ethernet déclarée dans la configuration OSPFD.

8. Quelles sont les informations que les routeurs du backbone ont sur la topologie de l'area x et de l'area y?

Contrairement à **RIP**, **OSPF** a été pensé pour supporter de très grands réseaux. Mais, qui dit grand réseau, dit nombreuses routes.

Afin d'éviter que la bande passante ne soit engloutie dans la diffusion des routes, **OSPF** introduit le concept de zone (area).

Le réseau est divisé en plusieurs zones de routage qui contiennent des routeurs et des hôtes.

Chaque zone, identifiée par un numéro, possède sa propre topologie et ne connaît pas la topologie des autres zones.

Chaque routeur d'une zone donnée ne connaît que les routeurs de sa propre zone ainsi que la façon d'atteindre une zone particulière, la zone numéro 0.

Toutes les zones doivent être connectées physiquement à la zone 0 (appelée backbone ou réseau fédérateur).

Elle est constituée de plusieurs routeurs interconnectés. Le backbone est chargé de diffuser les informations de routage qu'il reçoit d'une zone aux autres zones.

Tout routage basé sur **OSPF** doit posséder une zone 0.

Dans notre TP, le réseau est découpé en trois zones dont le backbone. Les routeurs de la zone 1, par exemple, ne connaissent pas les routeurs de la zone 2 et encore moins la topologie de la zone 2.

L'intérêt de définir des zones est de limiter le trafic de routage, de réduire la fréquence des calculs du plus court chemin par l'algorithme **SPF** ainsi que d'avoir une table de routage plus petite (ce qui accélère la convergence).

Les routeurs **R1** (Linux 11.0.0.3) et **R2** (Linux 11.0.0.4) sont particuliers puisqu'ils sont «à cheval» sur plusieurs zones (on les appelle **ABR** pour **Area Border Router** ou **routeur de bordure de zone**).

Ces routeurs maintiennent une base de données topologique pour chaque zone à laquelle il sont connectés. Les **ABR** sont des points de sortie pour les zones ce qui signifie que les

informations de routage destinées aux autres zones doivent passer par l'**ABR** local à la zone.

L'**ABR** se charge alors de transmettre les informations de routage au backbone.

Les **ABR** du backbone distribueront ensuite ces informations aux autres zones auxquelles ils sont connectés.

9. Quelles sont les informations que les routeurs de l'area y ont de l'area x et de l'area 0 ?

Les routeurs de la zone Y ont toutes les informations des zones X et 0 en se référant à ce qui a été expliqué à la question précédente.

10. Qui est le routeur désigné de votre area ?

Afin de voir qui est notre **Designated Router** (DR), nous pouvons le voir avec la commande ***show ip ospf neighbor*** en regardant l'état du lien du voisin. Si on voit **FULL/DR** dans le champ State par exemple, cela signifie que c'est un **Designated Router**.

Nous pouvons le voir aussi en regardant avec la commande ***show ip ospf interface*** où l'on peut voir le **Designated Router**.

Comment élire le **DR** ? Le routeur élu est celui qui a la plus grande priorité (**Router ID** ou **RID**).

La priorité est un nombre sur 8 bits fixé par défaut à 1 sur tous les routeurs. Pour départager les routeurs ayant la même priorité, celui qui est élu à la plus grande adresse IP sur une interface de boucle locale (**loopback interface**) ou sur un autre type d'interface active.

Le **BDR** sera le routeur avec la deuxième plus grande priorité.

Dans notre cas, nous avons un **Designated Router** dans la zone 0 et c'est l'hôte d'adresse IP 11.0.0.4 (**R2**). Son **BDR** est l'hôte d'adresse IP 11.0.0.3 (**R1**).

Dans les autres zones, nous n'avons qu'un seul routeur par zone. C'est ce routeur qui sera le **Designated Router** puisqu'il possède les adresses IP les plus grandes comparés aux routeurs **R1** et **R2**.

11. Affichez dans Zebra le(s) routeur(s) de bordure de l'area x ou y.

Les routeurs de bordures sont les routeurs Quagga soit les machines Linux.

- L'hôte Linux définit comme R1 est le routeur de bordure de la zone 0 et de la zone Y.
- L'hôte Linux définit comme R2 est le routeur de bordure de la zone 0 et de la zone X.

12. Sur le routeur, regardez la table de routage. Quelles sont les routes OSPF ?

Afin de connaître les différentes routes **OSPF** présentes sur un de nos routeurs, il suffit de faire la commande ***show ip route***.

Nous obtenons un début de résultat ci-après :
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
B - BGP, > - selected route, * - FIB route

Afin de déterminer quelle route est **OSPF** ou non, il suffit de se référer au **O** présent devant l'une de nos routes.

Dans notre cas, les routes OSPF sont les routes reliant chaque zone entre-elles.

IV - BGP

*Note : Nous n'avons pas pu fournir tout au long de cette partie (**BGP**) des captures d'écran de nos travaux puisque nos outils (GNS3, Cisco Packet Tracer, etc…) n'étaient pas optimaux dans leur fonctionnement lors de la réalisation et l'exécution de ce TP.*

A.Configuration de BGP dans le Cisco

Pour configurer un routeur BGP celui-ci à besoin de connaître quatre choses:

- De quel AS (autonomous system) il est le routeur de bordure.
- Qui sont ces voisins.
- Quel préfixe réseau il doit annoncer.
- Donner son adresse de loopback.

1. Configurer l'interface de loopback lo0 avec l'adresse IP 50.0.0.X/32. Il n'est pas nécessaire de faire ***no shutdown*** pour cette interface.

   Pour configurer l'interface de loopback lo0, il faut utiliser les commandes ci-après :

   a. Pour l'**Area Y** (Y=3) :
   Router**#configure terminal**
   Router(config)**#interface loopback 0**
   Router(config-if)**#ip address 50.0.0.3 255.255.255.255**

   b. Pour l**'Area X** (X=4) :
   Router**#configure terminal**
   Router(config)**#interface loopback 0**
   Router(config-if)**#ip address 50.0.0.4 255.255.255.255**

   c. Pour l'**AS 65012** :
   Router**#configure terminal**
   Router(config)**#interface loopback 0**
   Router(config-if)**#ip address 50.0.0.252 255.255.255.255**

   d. Pour l'**AS 65010** :
   Router**#configure terminal**
   Router(config)**#interface loopback 0**
   Router(config-if)**#ip address 50.0.0.254 255.255.255.255**

   e. Pour l'**AS 65011** :
   Router**#configure terminal**
   Router(config)**#interface loopback 0**
   Router(config-if)**#ip address 50.0.0.253 255.255.255.255**

2. Dans le contexte ***router bgp***, configurer votre identifiant d'**AS**.

   Pour configurer l'AS, il faut utiliser les commandes suivantes :
   a. Pour l'**Area Y** (Y=3) **AS 65000** : Router(config)**# router bgp 65000**
   b. Pour l**'Area X** (X=4) **AS 65000** : Router(config)**# router bgp 65000**
   c. Pour l'**AS 65012** : Router(config)**# router bgp 65012**
   d. Pour l'**AS 65010** : Router(config)**# router bgp 65010**
   e. Pour l'**AS 65011** : Router(config)**# router bgp 65011**

3. Il faudrait aussi associer l'identifiant du routeur à son adresse loopback. Pour cela, utiliser la commande bgp router-id.

   Nous associons l'identifiant du routeur à son adresse **loopback** pour chaque routeur:
   a. Pour l'**Area Y** (Y=3) **AS 65000** : Router(config-router)**#bgp router-id 50.0.0.3**
   b. Pour l**'Area X** (X=4) **AS 65000** : Router(config-router)**#bgp router-id 50.0.0.4**
   c. Pour l'**AS 65012** : Router(config-router)**#bgp router-id 50.0.0.252**
   d. Pour l'**AS 65010** : Router(config-router)**#bgp router-id 50.0.0.254**
   e. Pour l'**AS 65011** : Router(config-router)**#bgp router-id 50.0.0.253**

4. Puis, dans le même contexte, déclarer vos voisins (AS) grâce à la commande (neighbor). C'est-à-dire le 9.0.1.254 (resp. 9.0.2.254). N'oubliez pas que les deux routeurs du même banc doivent établir une session iBGP.

   A présent, nous allons déclarer nos voisins (**AS**), grâce à la commande neighbor :
   a. Pour l'**Area Y** (Y=3) **AS 65000** : Router(config)**#router bgp 65000**
      Router(config-router)**#neighbor 9.0.1.254 remote-as 65012**

      Il existe deux modes de fonctionnement de **BGP** : ***Interior BGP*** (**iBGP**) et ***Exterior BGP*** (**eBGP**). **iBGP** est utilisé à l'intérieur d'un **Autonomous System** alors que **eBGP** est utilisé entre deux **AS**.

      Après avoir configurer l'**eBGP**, il faut configurer l'**iBGP.**

      Router(config)**#interface loopback 0**
      Router(config-if)**#ip address 10.0.3.2 255.255.255.0**
      Router(config)**#router ospf 1**
      Router(config-router)**#network 10.0.3.2 0.0.0.0 area 3**

   b. Pour l**'Area X** (X=4) **AS 65000** : Router(config)**#router bgp 65000**
      Router(config-router)**#neighbor 9.0.2.254 remote-as 65011**

      Après avoir configurer l'**eBGP**, il faut configurer l**'iBGP** car l'**Area** 4 est dans le même **AS** que l'**Area** 3 qui est l'**AS** 65000.

      Router(config)**#interface loopback 0**

Router(config-if)**#ip address 10.0.4.2 255.255.255.0**
Router(config)**#router ospf 1**
Router(config-router)**#network 10.0.4.2 0.0.0.0 area 4**

c. Pour l'**AS 65012** : Router(config)**#router bgp 65012**
Router(config-router)**#neighbor 7.0.0.1 remote-as 65010**
Router(config-router)**#neighbor 9.0.1.3 remote-as 65000**

d. Pour l'**AS 65010** : Router(config)**#router bgp 65010**
Router(config-router)**#neighbor 8.0.0.2 remote-as 65011**
Router(config-router)**#neighbor 7.0.0.2 remote-as 65012**

e. Pour l'**AS 65011** :Router(config)**#router bgp 65011**
Router(config-router)**#neighbor 8.0.0.1 remote-as 65010**
Router(config-router)**#neighbor 9.0.2.4 remote-as 65000**

5. A présent, il faut déclarer les routes qu'on veut afficher à l'extérieur de l'**AS**.

   Pour cela, on a deux façons de faire :

   Soit utiliser une déclaration statique regarder du côté de la commande **network**, toujours sous le contexte **bgp**.

   Soit de façon dynamique en redistribuant les routes connectés et les routes apprises par **OSPF** avec la commande **redistribute**.

   Donc il faudrait déclarer le réseau 50.0.0.X/32 (resp. 50.0.0Y/32), redistribuer les routes connectées et celles obtenues avec OSPF.

   a. Pour l'**AS 65012** nous utilisons la commande **network**:
   Router(config)**#router bgp 65012**
   Router(config-router)**#network 9.0.1.254 mask 255.255.255.0**
   Router(config-router)**#network 7.0.0.2 mask 255.255.255.0**

   b. Pour l'**AS 65010** nous utilisons aussi la commande **network** :
   Router(config)**#router bgp 65010**
   Router(config-router)**#network 8.0.0.1 mask 255.255.255.0**
   Router(config-router)**#network 7.0.0.1 mask 255.255.255.0**

   c. Pour l'**AS 65011** nous utilisons aussi la commande **redistribute** :
   Router(config)**#router bgp 65011**
   Router(config-router)**#nredistribute connected route-map BGPRedistribution**

6. Il faudrait aussi exécuter ces deux commandes dans le contexte bgp : no synchronisation, no auto-summary. Quels sont les buts de ces deux commandes ?

   Il ne faut pas oublier de redistribuer les routes pour la **relation iBGP** entre l'**Area 3** et l'**Area 4**.

   En effet, 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 de type **OSPF**.

   Dans notre topologie l'**Area** 3 n'apprendra pas la route 9.0.2.254/24 venant de l'**AS** 65011 par **iBGP** tant qu'il n'a pas reçus une mise à jour **OSPF** contenant une route vers cette même destination.

   no synchronisation : Un routeur **BGP** avec la synchronisation activée n'annoncera pas les routes apprises par **iBGP** à d'autres homologues **eBGP** s'il ne peut pas valider ces routes dans son **IGP**.

   En supposant qu'**IGP** dispose d'une route vers les routes apprises par **iBGP**, le routeur annoncera les routes **iBGP** aux homologues **eBGP**.

   Dans le cas contraire, le routeur traite la route comme n'étant pas synchronisée avec IGP et ne l'annonce pas.

   La désactivation de la synchronisation à l'aide de la commande no synchronization sous le routeur **BGP** empêche ce dernier de valider les routes **iBGP** dans **IGP**.

   no auto-summary : Désactive le résumé automatique, ne résume plus les réseaux dans leur adresse par classe au niveau des routeurs de périphérie.

   a. Pour l'**Area Y** (Y=3) **AS 65000** :
      no synchronization
      no auto-summary

   b. Pour l'**Area X** (X=4) **AS 65000** :
      no synchronization
      no auto-summary

7. Sur le Cisco, vous pouvez contrôler votre configuration. A la racine du routeur, tapez la commande: **show ip bgp neighbors.**

   Comme indiqué dans la consigne, il suffit de se mettre en enable à la racine du routeur et de faire la commande show ip bgp neighbors pour vérifier nos configurations.

8. Afficher la table de routage des routeurs Cisco. qu'est-ce que vous constatez ?

   Afin d'afficher la table de routage des routeurs Cisco il suffit de taper la commande **show ip route** ou **show bgp**.

9. Dans les routes bgp, on voit des préfixes étiquetés avec un i. Pourquoi ?

   Dans les routes bgp, on remarque des préfixes étiquetés avec un i. Ces routes n'ont pas été mises dans la table de routage du routeur.

   Il y a deux raisons :

   La première est qu'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 tel que **OSPF**.

   La seconde est que, quand **eBGP** annonce une route, il change le Next-Hop. Quand iBGP annonce une route, il ne change pas le Next-Hop.

10. Sur les routeurs Quagga, afficher la table de routage. Quels routeurs BGP utilise-t-on pour sortir vers les AS externes ?

    Pour sortir vers les ASs externes, on utilise les routeurs BGP de l'Area 3 ou de l'Area 4.

11. Sur les routeurs bgp afficher la table de routage bgp. Quelles informations peut-on voir ? Est-ce qu'on voit le PATH AS ?

    Si l'on affiche la table de routage bgp avec la commande **show bgp** on peut voir les networks, next-hop, metric, locprf, weight et le path. Le path permet de savoir par quel **AS** nous passons.

B.Choix du routeur eBGP

A présent nous allons modifier la configuration des routeurs Cisco pour mettre en place l'attribut **localpreference** de **bgp**.

1. Dans le contexte bgp, rajouter une route-map comme suit : neighbor 9.0.1.254 (resp. 9.0.2.254) route-map sortie in. Quel est le but d'une route-map sur les routeurs ?

   Nous entrons la commande **neighbor 9.0.1.254 route-map sortie in**. La commande route-map permet de forcer pour certains flux, le chemin en ne suivant pas les routes qui sont dans la table de routage.

2. Dans le contexte configure terminal, on tape la commande **route-map sortie permit 10**. Dans le nouveau contexte appelé **route-map** on peut à présent déclarer le

localpreference. Pour rappel, plus le **localpreference** est grand, plus le routeur à de priorité.

La **Local Preference** est un peu comme le poids. Elle permet de choisir un routeur favori, et de l'annoncer dans l'**AS**.

Par contre, la **Local Preference** n'est pas annoncée hors de l'**AS**.

3. Afficher à présent les tables de routage bgp des deux Cisco du banc. Est-ce que les tables ont changé ?

   On affiche la table **bgp** avec un **show bgp** et l'on remarque qu'en effet la table à changer et que l'adresse de plus haute priorité se situe tout en haut de la table.

   BGP a parcourus la liste des attributs, afin de différencier les routes en se demandant, quelle route a le meilleur poids ? Aucune des deux.

   Quelle route a la meilleure Local Preference ? Celle d'adresse IP 9.0.1.254. L'hôte 9.0.1.254 sera choisi comme Next Hop.

   En sommes, pour ne favoriser que certaines d'entre-elles, il faudra utiliser une route-map.

4. Regarder les tables de routages de Quagga. Est-ce que ça a changé ?

   Oui, la table de routage de quagga à changé car il passe maintenant par les routes avec la plus haute priorité.

5. Dans le contexte de ce TP, a-t-on besoin d'utiliser un MED pour définir le routeur d'entrée à AS ?

   Dans le contexte de ce TP nous n'avons pas besoin d'un **MED** (Multi-Exit-Discriminator, aussi appelé **Metric**). Le **MED** permet de favoriser un routeur pour l'entrée dans l'AS.

   La métrique la plus faible est la meilleure niveau priorité dans un **MED**.
   Pour cela, deux solutions :
   - Utiliser la commande « default-metric » : Applique l'attribut à toutes les routes annoncées.
   - Utiliser une Route Map et appliquer une métrique (MED) sur certains réseaux (routes) annoncés

     Sauf que cette commande ne permet que de modifier la métrique par défaut. Or certaines routes sont annoncées avec une métrique (donc la

métrique par défaut n'aura d'effet). La métrique (ou **MED**) dépend de la façon d'annoncer les routes (commande **Network**, **Redistribute**, etc…)

V - Remise en état du matériel

- Penser à la remise en état du routeur **Cisco** et **Switch**:
    - Si la startup-configuration n'a pas été modifier, **éteindre le routeur**.
    - Sinon, se connecter dessus avec **minicom**, passer en mode **enable** et lancer la commande **reload**.
    - Pour le Switch idem en lançant **reload**.

- Penser à la remise en état du PC **Linux**:
    - Vérifiez que l'interface **eth0** est bien **up**.
    - Lancez les scripts **/script/init_machine.sh** et **/script/init_reseau.sh**.

- Penser à remettre le **câblage** en état:
    - Tous les câbles doivent être débranchés de leur interface, **sauf eth0** qui ne doit pas l'être.

VI - Conclusion

Le TP4 du module d'ADMI nous à permis d'aborder le routage dynamique en étudiant les protocoles **OSPF** et **BGP**.

Nous avons appris à configurer une topologie simple dans des zones en utilisant le protocole **OSPF** qui utilise l'algorithme du plus court chemin de **Djikstra**.

OSPF est un protocole de routage dynamique moderne, robuste et conçu pour les grands réseaux. On constate qu'il est nettement plus complexe que **RIP** dans son fonctionnement interne. L'un des inconvénients de ce protocole est qu'il peut être (très) gourmand en puissance de calcul et en mémoire lorsque le réseau comporte beaucoup de routes ou qu'il y a de fréquentes modifications de topologie.

De plus, nous avons vu que **OSPF** est un protocole **IGP** (**Interior Gateway Protocol**), c'est-à-dire qu'il agit au sein d'un système autonome. Un **AS** (**Autonomous System**) est un ensemble de réseaux gérés par un administrateur commun.

Cependant, pour assurer le routage entre les systèmes autonomes, un protocole de type **EGP** (**Exterior Gateway Protocol**) doit être mis en œuvre. Dans le cas de l'Internet, c'est généralement **BGP** (**Border Gateway Protoco**l) qui assume cette mission.

C’est pourquoi nous avons utilisé le protocole **BGP**. Cette phase incluait la notion d’**iBGP** (***Interior BGP***) et d’**eBGP** (***Exterior BGP)***. **iBGP** est utilisé à l'intérieur d'un **Autonomous System** alors que **eBGP** est utilisé entre deux **AS**.

Les connexions **eBGP** sont établies sur des connexions point-à-point ou sur des réseaux locaux. Si la liaison physique est rompue, la session **eBGP** l’est également, et tous les préfixes appris par celle-ci sont annoncés comme supprimés et retirés de la table de routage.

A l’inverse, les connexions **iBGP** sont généralement établies entre des adresses IP logiques, non associées à une interface physique particulière, les adresses loopback.

Pour conclure, ce TP nous a ouvert une nouvelle vision sur le routage avec la mise en place des routages dynamiques au travers de l’utilisation des protocoles **OSPF** et **BGP**. De plus, cela nous a permis de confirmer certaines de nos connaissances et d’acquérir de nouvelles façons de configurer les routeurs nous montrons qu’il est possible de faire énormément de choses avec ces outils, même s’ils sont de vieille génération.