Réseaux IP - Projet Intégré

Etudes Préliminaires

Auteurs :
M. Samuel Riedo

Encadrant :
M. François Buntschu

2 avril 2017

Table des matières

Première partie
Routage Statique & Dynamique

1 Introduction

Le plus connu des réseaux est Internet. Ce dernier est composé d'une multitude de réseaux interconnectés entre eux par des routeurs. Lorsqu'une station d'un réseau A souhaite transmettre des données à la station d'un réseau B, le chemin pour y accéder lui est inconnu. Le but du routage consiste à définir une route entre deux stations au travers d'Internet.

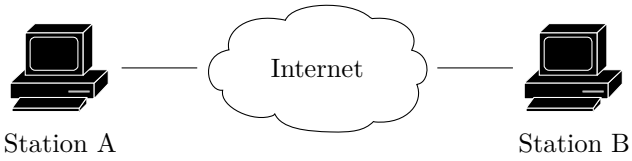

FIGURE 1.1 – Représentation simple d'Internet

Il existe deux moyens de configurer la fonction de routage dans un routeur. La première, appelée routage statique, consiste à configurer manuellement les tables de routages des routeurs. Une table de routage contient des routes, soit une interface de sortie pour une adresse IP.
L'autre manière est de configurer un routage dynamique sur les routeurs. De cette façon, les routeurs vont communiquer périodiquement entre eux au moyen d'un protocole de routage afin de mettre à jour leur table de routage. Le protocole de routage détermine donc comment les routeurs communiquent entre eux, mais également la façon dont il vont calculer la meilleure route.

Remarque. *Généralement, les routeurs utilisent à la fois du routage statique et dynamique pour maximiser l'efficacité du routage et avoir accès à une route par défaut en cas d'échec d'échange d'information via routage dynamique.*

2 Routage Statique

2.1 Fonctionnement

Le routage statique est une forme de routage ou un routeur utilise des configurations manuelles. Dans la plupart des cas, les routes statiques sont configurées par un administrateur réseau qui ajoute les entrées une à une dans la table de routage. Il est également possible pour le routeur de copier une table contenue dans un serveur distant. Dans les deux cas, cette table est statique, c'est-à-dire qu'elle n'est pas modifiée automatiquement pour s'adapter à l'évolution du réseau (une nouvelle route est disponible, une ligne est coupée...).

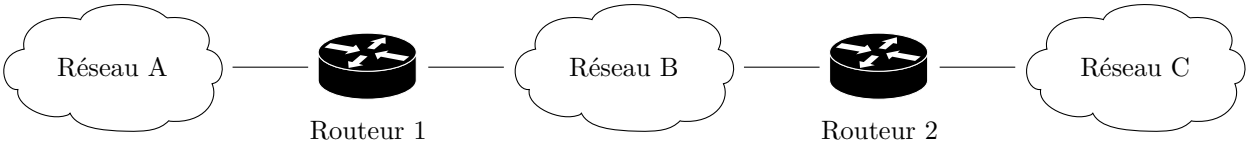

FIGURE 2.1 – Exemple de réseau

Dans le schéma réseau ci-dessus (figure 2.1), le routeur 1 connait les réseaux A et B. Pour accéder au réseau C, il faut que l'administrateur réseau configure manuellement une route dans les tables de routage pour lui indiquer qu'il doit passer par le réseau B. De même, le routeur 2 ne connait que les réseaux B et C. Une configuration statique pour accéder au réseau A via le réseau B est nécessaire.

Avantages	Désavantages
Les routes statiques peuvent être utilisées pour définir une route par défaut si aucune route n'a été trouvée par routage dynamique.	Probabilité d'erreur importante que l'administrateur réseau entre faux une ou plusieurs route(s).
Généralement, le routage statique est plus efficace dans des petits réseaux contenant une ou deux routes.	Aucune adaptabilité à l'évolution du réseau. Si une route n'est plus disponible, le routeur continuera de l'utiliser.
Économie de bande passante, les routeurs n'échangent pas d'information de routage entre eux.	La configuration et la maintenance de grand réseau sont longues et complexes.
Sécurité accrue, aucune information relative au réseau n'est transmise.	

TABLE 2.1 – Avantages & Désavantages du routage statique

Aujourd'hui, le routage statique est principalement uniquement utilisé sur de petits réseaux n'évoluant peu ou en complément du routage dynamique.

2.2 Configuration

Une route statique par défaut est une route qui s'applique à tous les paquets. Une route par défaut identifie l'adresse IP de la passerelle à laquelle le routeur envoie tous les paquets IP qui n'ont pas de route apprise ou statique. Une route statique par défaut est simplement une route statique avec 0.0.0.0/0 comme adresse IPv4 de destination. La configuration d'une route statique par défaut crée une passerelle de dernier recours.

Ajouter une route statique par défaut

```
R4(config)# ip route 0.0.0.0 0.0.0.0 192.168.3.1
```

Supprimer une route par défaut

```
R4(config)# no ip route 0.0.0.0 0.0.0.0 192.168.3.1
```

Ajouter une route statique

```
Router> enable
Router# configure terminal
Router(config# interface s0/0/0
Router(config)# ip route 10.10.20.0 255.255.255.0 192.168.100.1
```

Ajouter deux routes pour une seule destination

```
Router> enable
Router# configure terminal
Router(config)# ip route 197.164.73.0 255.255.255.0 197.164.72.2
Router(config)# ip route 197.164.74.0 255.255.255.0 197.164.72.2
```

Ajouter une route statique via l'interface de sortie

```
Router(config)# ip route 10.10.20.0 255.255.255.0 Serial 0/0/0
```

Affichage de la table de routage

```
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 192.168.3.1 to network 0.0.0.0

10.0.0.0/24 is subnetted, 1 subnets
C 10.0.0.0 is directly connected, Ethernet1/0
S 192.168.2.0/24 [1/0] via 10.0.0.2
C 192.168.3.0/24 is directly connected, FastEthernet2/0
S* 0.0.0.0/0 [1/0] via 192.168.3.1
```

Il est également possible de filter uniquement les routes statiques de la façon suivante :

Affichage des entrées statiques de la table du routage

```
Router(config)# ip route static
```

3 Routage Dynamique

3.1 Fonctionnement

Le routage dynamique, aussi appelé routage adaptatif, utilise un protocole de routage pour partager et recevoir des informations sur le réseau afin de mettre à jour sa table de routage de façon périodique ou événementielle. Le protocole de routage :

- La découverte des réseaux distants.
- L'actualisation des informations de routage.
- Le choix du meilleur chemin vers les réseaux de destination.
- La capacité à trouver un nouveau meilleur chemin si le chemin actuel n'est plus disponible.

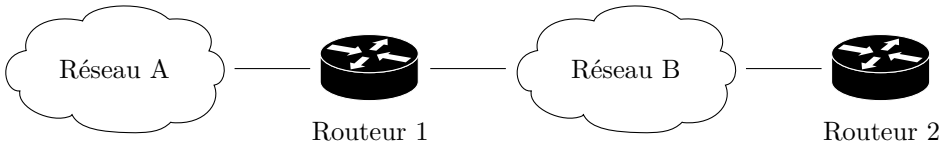

FIGURE 3.1 – Etat du réseau au temps T

Soit le réseau ci-dessus (figure 3.1). Les tables de routages des routeurs 1 et 2 sont remplis via un protocole de routage dynamique. Un nouveau réseau C est par la suite ajouté sur une des interfaces du routeur 2 (figure 3.2). Ce dernier va mettre à jour sa table de routage et informer le routeur 1 que ce nouveau réseau est arrivé et qu'il doit passer par lui pour l'atteindre.
Le principe serait similaire si un réseau ou un routeur venait à être déconnecté du réseau. Les équipements réseaux périphériques au changement modifieraient leur table et informeraient d'autres équipements du changement.

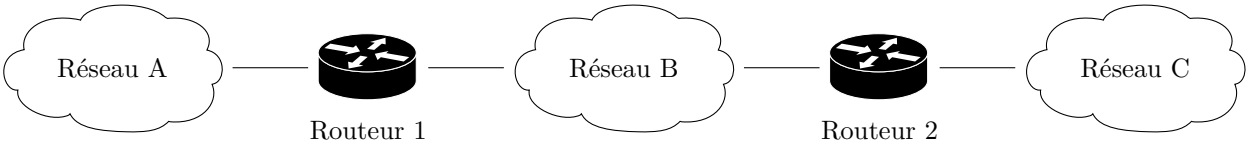

FIGURE 3.2 – Etat du réseau au temps $T + k$

Le réseau a convergé lorsque tous les routeurs disposent d'informations complètes et précises sur le réseau entier. Le temps de convergence est le temps nécessaire aux routeurs pour partager des informations, calculer les meilleurs chemins et mettre à jour leurs tables de routage. Un réseau n'est pas complètement opérationnel tant qu'il n'a pas convergé. Par conséquent, le temps de convergence doit être bref pour la plupart des réseaux.

Avantages	**Désavantages**
Maintenance réduite par les mises à jour automatiques des tables de routage.	Usage de bande passante par les routeurs.
Évolution du réseau plus simple, meilleure flexibilité.	Plus complexe à mettre en place.
Indépendant de la taille du réseau.	Moins sécurisé du fait qu'un hacker peut obtenir des informations sur le réseau simplement en interception les trames des protocoles de routage dynamique.

TABLE 3.1 – Avantages & Désavantages du routage dynamique

3.2 Protocoles

Il existe cinq principaux protocoles de routage. Il peuvent être classés dans différents groupes selon leurs caractéristiques :

Protocoles IGP et EGP

Les protocoles IGP sont utilisés dans des réseaux autonomes (SA), alors que les protocoles de type EGP servent à connecter les différents systèmes autonomes entre eux.

Remarque. *BGP est le seul protocole de type EGP, le terme EGP est de ce fait de manière générale peu utilisé au profit du terme BGP.*

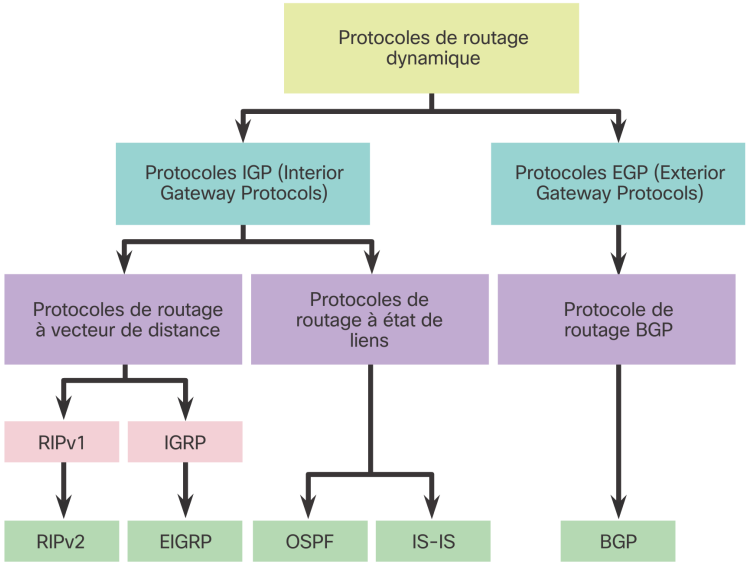

FIGURE 3.3 – Protocoles de routage

Les protocoles IGP sont ensuite divisés entre ceux à vecteurs de distance, et ceux à état de liens. Le premier caractérise une route par une interface de sortie et un nombre de sauts sans connaître le chemin complet entre la source et la destination. Le second permet à chaque routeur de connaitre la topologie complète du réseau en récupérant les informations de tous les autres routeurs. De plus, ces protocoles ne partagent pas périodiquement des informations, mais uniquement lorsqu'une modification de topologie survient.

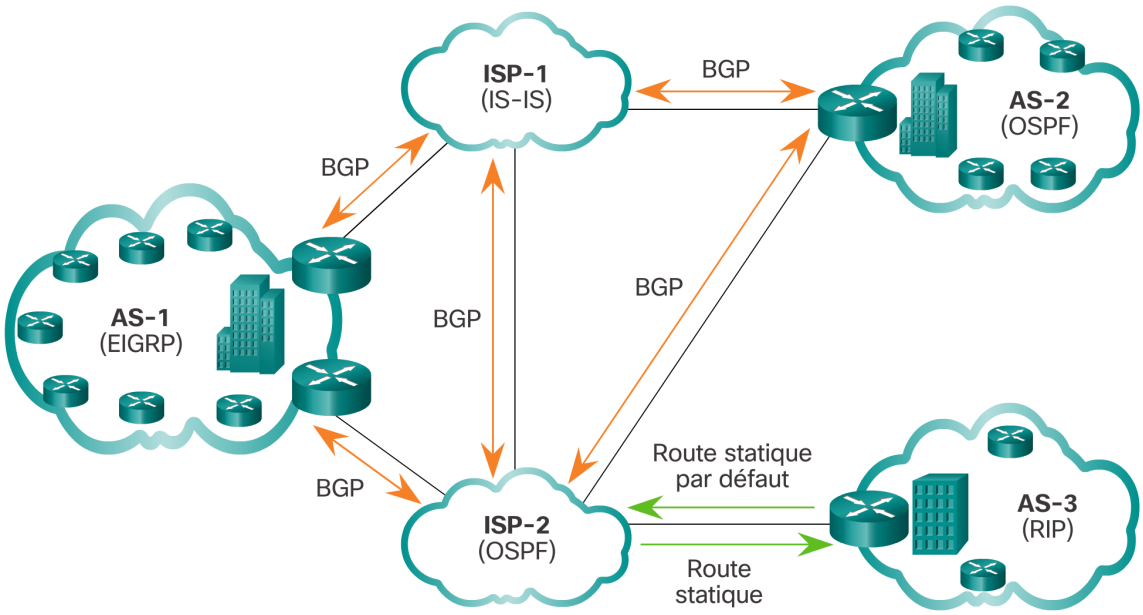

FIGURE 3.4 – Routage EGP & IGP

3.2.1 RIP

3.2.1.1 Introduction

RIP, Routing Information Protocol, est un protocole de routage dynamique de type vecteur distance basé sur l'algorithme de routage de Bellman-Ford. Le metric de routage est le nombre de hops, c'est-à-dire le nombre de routeurs à traverser pour aller d'un point A à un point B. Ce nombre de hops est également utilisé comme solution pour éliminer les boucles en le fixant à un maximum de 15. Cette solution présente néanmoins le défaut de limiter la taille des réseaux sur lesquelles RIP pourra fonctionner, puisque devoir traverser 16 routeurs ou plus entre la source et la destination est impossible.
La première version de RIP, sortie en 1988, transmettait des informations sur le réseau toutes les 30 secondes. Dans les années 80, les tables de routage n'étaient en effet pas grandes et le trafic généré n'était pas problématique. Au fur et à mesure que les réseaux ont grandi, il est apparu que ses informations de routage devenaient problématiques en créant des burst sur le réseau toutes les trente secondes.
De nos jours, le protocole de routage RIP n'est pas fréquemment utilisé, ce dernier étant jugé comme moins bon que ses concurrents. Il reste néanmoins un bon choix dans les petits réseaux pour sa simplicité d'implémentation.

3.2.1.2 RIPv1

Dans sa première version (RFC 1058), RIP utilise du broadcast local pour partager des informations de routage de manière périodique toutes les trente secondes. Comme indiqué précédemment, chaque paquet contient un champ indiquant le nombre de hop déjà effectué. Si ce nombre atteint 16, le paquet est éliminé par les routeurs le recevant. Ce système, utilisé pour éliminer les boucles, sera gardé dans la deuxième version de RIP.

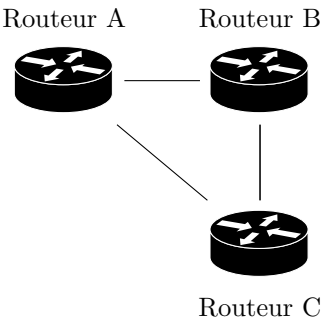

FIGURE 3.5 – Problème des boucles

Soit le schéma ci-dessus (figure 3.5). En sachant que le routeur A traverse le routeur B pour atteindre le troisième routeur, que se passerait-il si la ligne entre les routeurs B et C tombait en panne ?
Le routeur B verrait que la ligne pour aller sur C n'est plus opérationnelle, il essayerait donc de l'atteindre en passant par A. Hors, ce dernier n'a pas connaissance du problème, il renvoie donc les paquets sur B pour atteindre et C, qui va à son tour les lui renvoyer créant ainsi une boucle.
Prévenir ce problème est la raison de l'implémentation du nombre de sauts maximum d'un paquet dans RIP. Après 15 passages entre A et B, les paquets ne pourront plus continuer de transiter sur le réseau et seront perdus.

Remarque. *Certains protocoles tels que l'OSPF informent les routeurs voisins qu'une ligne n'est plus disponible, évitant ainsi ce problème sans que des données fassent plusieurs aller-retour inutiles avant d'être supprimées du réseau.*

RIPv1 est un protocole utilisant du routage par classe (A, B, C, D et E) sans que les informations du subnet soit transmises. En conséquence, il ne supporte pas les sous-réseaux de taille variables (VLSM). Il n'inclut également pas de support pour l'authentification de routeur, le rendant vulnérable à divers types d'attaques.

Avantages	Désavantages
Mise en service simple.	Système anti loop pas optimale comparé à d'autres protocoles
	Pas de support d'authentification.
	Ne supporte pas les sous-réseaux de tailles variables (VLSM).
	Diffusion des mises à jour de routage en broadcast pouvant être problématique pour la sécurité du réseau.

Table 3.2 – Avantages & Désavantages de RIPv1

Exemple de configuration :

Configurer l'interface assignée au réseau 10.10.0.0

```
router# configure terminal
router(config)# router rip
router(config-router)# version 1
router(config-router)# network 10.10.0.0
```

Affichage de la table de routage

```
router# show ip route
```

3.2.1.3 RIPv2

Avec l'évolution des réseaux, la première version de RIP est vite devenue obsolète. Ainsi, en 1993, la version 2 de RIP était définie (RFC 2453). Cette dernière permet le support des masques de sous-réseaux à taille variable (VLSM), c'est-à-dire le routage classless. Le nombre de sauts maximal reste fixé à 16 et le protocole est compatible avec la première version de RIP.
Un autre apport de cette seconde version est le partage d'information de routage de manière unicast avec l'adresse *224.0.0.9* de classe D (réservé pour la multidiffusion), et non en broadcast afin de minimiser l'impact sur le réseau. Il est également possible d'authentifier de manière cryptée ou non la source d'un paquet RIPv2.

Remarque. *RIPv2 ne supporte pas IPv6. Une autre version du protocole, appelé RIPng a été conçu pour pallier à ce manque.*

Exemple de configurations :

Activer RIP pour un réseau

```
router(config)# router rip
router(config-router)# network 192.168.10.0
```

Une interface passive ne transmet pas de paquet RIP. Par défaut, aucune interface n'est passible.

Ajouter une interface passive

```
router(config-router)# passive-interface g0/0
```

Propager une route par défaut

```
router(config)# ip route 0.0.0.0 0.0.0.0 exit-interface next-hop-ip
router(config-router)# default-information originate
```

3.2.2 OSPF

Open Shortest Path First est un protocole de routage dynamique IGP à état de liens. Contrairement à RIP, il n'implémente pas un nombre de sauts maximum, ce qui ne limite pas son implémentation selon la taille de réseau. Il permet également une convergence plus rapide que ce premier et supporte nativement VLSM. D'un point de vue performance, OSPF se classe également dans le haut du panier en utilisant notamment du multicast et non du broadcast pour propager les informations sur l'état du réseau. Décliné en deux versions, OSPFv2 pour l'IPv4 et OSPFv3 pour l'IPv6, OSPF est aujourd'hui le protocole utilisé dans les systèmes autonomes.

Avantages	Désavantages
Pas de limitation de taille de réseau.	Utilise beaucoup de ressource CPU du routeur.
Supporte VLSM.	Utilise beaucoup de mémoire.
Partage d'état de liens de façon multicast.	Si une route change fréquemment entre disponible et indisponible dans un grand réseau utilisant OSPF, des burst de LSP afin de mettre constamment à jour tous les routeurs utiliseraient beaucoup de bande passante.
Meilleure convergence que RIP.	
Meilleur équilibrage de charge que RIP.	
Prend en charge l'authentification des routeurs.	

Table 3.3 – Avantages & Désavantages d'OSPF

3.2.2.1 Fonctionnement d'un protocole à état de liens

Dans un protocole à état de liens, un lien représente l'interface d'un routeur. Les informations relatives à l'état de ces liens sont appelées état de liens. Tous les routeurs de la topologie vont suivre le processus suivant pour converger leur état :

1. Chaque routeur reçoit des informations sur les réseaux qui lui sont directement connectés. Seules les interfaces actives (up) sont incluses dans une des instructions de configuration du routeur participent au processus d'état de liens.
2. Chaque routeur détecte ses voisins sur les réseaux directement connectés en échangeant des paquets Hello. Hello est un Neighbor Discovery Protocol, soit un protocole de couche 2 responsable entre autre de la découverte d'autres hôtes sur le même lien en déterminant leur adresse. Les paquets Hello continuent d'être échangé entre deux routeurs s'étant mutuellement découvert afin de surveiller l'état du voisin. Si un routeur R1 ne reçoit plus de paquet Hello d'un routeur R2, R1 considère R2 comme injoignable.
3. Un paquet LSP (Link-State Packet) contenant l'état de chacun des liens connectés directement est construit par routeur. Ce paquet contient notamment l'ID du voisin, le type de lien et la bande passante.
4. Les paquets LSP sont transférés à tous les voisins de chaque routeur (inondation du réseau). Un routeur recevant un LSP stocke les informations contenues dans le paquet dans une base de données locale, puis fait suivre ce paquet à tous ses voisins excepté la source du paquet. Ce processus crée un effet de diffusion à partir de tous les routeurs de la zone de routage.

   Un paquet LSP est uniquement envoyé lors du démarrage initial du protocole de routage sur un routeur, ou lorsque la topologie a été modifiée (par exemple en cas d'activation ou désactivation d'un lien).
5. Grâce à sa base de données, chaque routeur élabore une carte complète de la topologie et calcule le meilleur chemin vers chaque réseau de destination. L'algorithme SPF (Shortest Path First), tel que l'algorithme de Dijkstra, permet de construire la carte de la topologie et a déterminé le meilleur chemin en fonction du coup (basé sur le débit de l'interface) vers chaque réseau lorsque la base de données d'états de liens est complète.

Remarque. *Bien que le coût soit calculé automatiquement selon le débit de l'interface de sortie, il est possible de le déterminer manuellement avec la commande suivante :*

Définir manuellement le coût d'une interface

```
router(config)# int f1/0
router(config-if)# ip ospf cost 12345
```

3.2.2.2 Fonctionnement d'OSPF

Le fonctionnement décrit au point 3.2.2.1 est applicable à tous les protocoles à états de liens. Néanmoins, OSPF présente certaines caractéristiques lui étant propres. Ainsi, il crée et met à jour non pas une, mais trois bases de données :

— **Base de données de contiguïté (table de voisinage)** - Crée une table de tous les routeurs voisins avec laquelle une connexion bidirectionnelle a été établie. Il est possible de l'afficher avec la commande suivante :

Afficher les voisins d'un routeur

```
router# show ip ospf neighbor
```

— **Base de données d'états de liens (LSDB)** - Liste les informations relatives à tous les routeurs du réseau. Cette table est identique pour les routeurs d'une même zone (voir 3.2.2.3). Il est possible de l'afficher avec la commande suivante :

Afficher la table d'état de liens

```
router# show ip ospf
```

— **Base de données de réacheminement (table de routage)** - Liste des routes créée par l'algorithme de routage. Cette table est unique pour chaque routeur.

Ces tables contiennent une liste des routeurs voisins permettant d'échanger les informations de routage, et elles sont conservées et mises à jour dans la mémoire vive. Afin de les créer et de les maintenir à jour, plusieurs types de paquets sont échangés entre les routeurs :

— **Paquet Hello** - Découvre les routeurs voisins
— **Paquet DBD de description de base de données** - Vérifie la synchronisation de la base de données entre les routeurs. Contient une liste abrégée de la LSDB.
— **Paquet LSR de demande d'état de liens** - Demande des renseignements sur une entrée spécifique de la LSDB.
— **Paquet LSU de mise à jour d'état de liens** - Réponse aux paquets LSR.
— **Paquet LSAck d'accusé de réception d'état de liens** - Confirme la réception d'un LSU.

Ces quatres derniers types de paquets sont des LSA (Link State Advertisement).

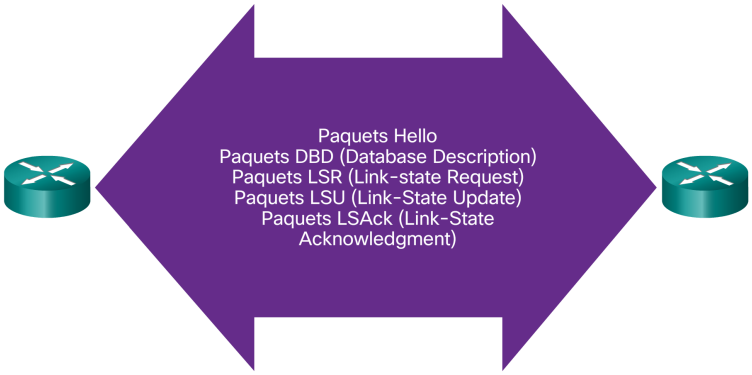

FIGURE 3.6 – Type de paquets échangé dans un réseau OSPF

3.2.2.3 Zone

Dans des grands systèmes autonomes, chaque routeur OSPF doit garder le LSA de chaque routeur dans sa base de données d'états de liens. Plus le réseau est grand, plus cette table le devient. En conséquence, les routeurs ont besoin de plus de mémoire et l'algorithme SPF prend beaucoup de temps processeur. Afin de réduire la taille des LSBD et le temps de l'algorithme SPF, il est possible de diviser un système autonome en zone (ou air).
Une zone est un ensemble de routeurs qui partage la même base de données d'états de liens caractérisés par un ID de 32 bits en notation décimale divisée en 4 nombres séparés par des points (comme une adresse IP). Il est néanmoins courant de donner un ID en notation décimal sans point aux zones (zone 41 au lieu de 36.54.7.23 par exemple). Les paquets LSA échangés par les routeurs restent confinés dans une aire sans en sortir.
Chaque système autonome utilisant OSPF dispose d'une aire backbone, avec l'ID 0.0.0.0, ou simplement 0. Cette zone est l'équivalent d'un hub auquel toutes les zones périphériques sont connectées. Elle est chargée de distribuer les informations de routage (LSA) et faire transiter les données entre zones.
Toutes les zones n'étant pas backbone sont des aires périphériques. Chaque aire périphérique doit être reliée à l'aire backbone et ne peut pas être directement connectée à une autre aire backbone.

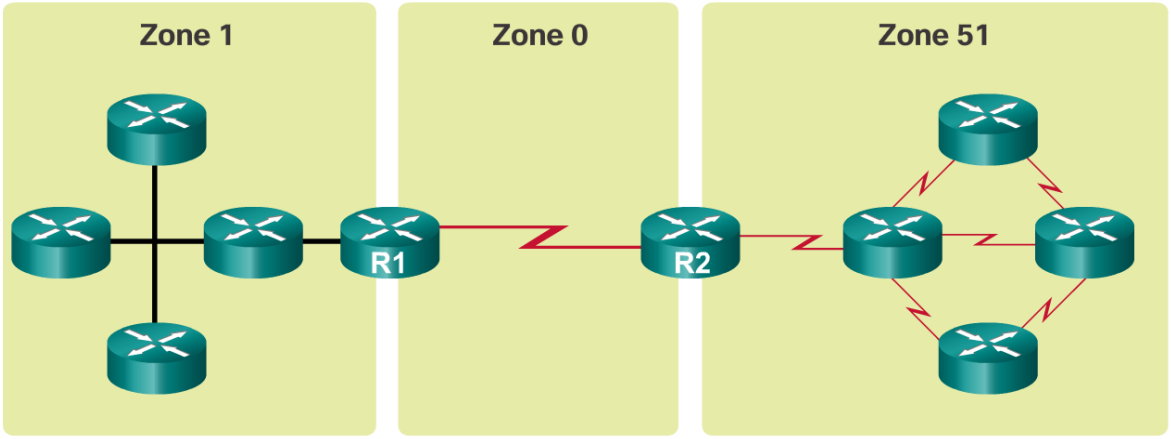

FIGURE 3.7 – Séparation d'un système autonome en trois zones

Soit le schéma ci-dessus (figure 3.7), les routeurs R1 et R2 sont appelés routeurs ABR (Area Border Router), du fait qu'ils relient deux zones entre eux. La zone 0 est l'aire backbone alors que les zones 1 et 51 sont des aires périphériques.
Les routeurs d'un réseau OSPF peuvent être caractérisés selon quatre types :

- **Internal Router** - Un routeur dont toutes les interfaces sont connectées à la même zone. Il n'a qu'une seule LSBD.
- **Are Border Router (ARB)** - Un routeur avec des interfaces connectées à différentes zones. Il possède une LSBD pour chaque aire à laquelle il est connecté.
- **Backbone Router** - Un routeur avec une interface sur l'aire backbone. Tous les routeurs ARB et les routeurs internes de l'aire backbone sont des routeurs backbone.
- **AS Boundery Router (ASBR)** - Un routeur échange des informations avec des sources en dehors du système autonome.

Remarque. *Dans un petit réseau, il est possible de créer une seule zone, qui sera donc l'aire backbone.*

3.2.2.4 Configuration

Configuration simple

```
router> enable
router# configure terminal
router(config)# router ospf process-id
router(config-router)# network wildcard-mask area area-id
```

Le process-id est un simple identifiant. Plusieurs process-id peuvent être utilisé sur le même routeur si plusieurs configurations d'OSPF différentes doivent être faites. Cet ID n'influence en rien la manière de fonctionner d'OSPF et n'est pas partagé sur le réseau.
Exemple

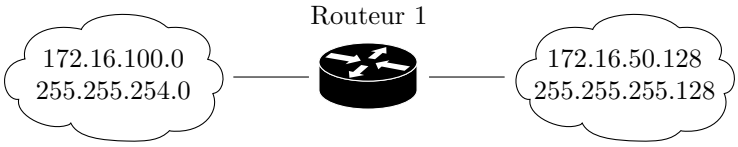

FIGURE 3.8 – Exemple de réseau

Configuration du réseau

```
router# configure terminal
router(config)# router ospf 10
router(config-router)# network 172.16.100.0 0.0.1.255 area 0
router(config-router)# network 172.16.50.128 0.0.0.127 area 0
```

Remarque. *Le mask est un wildcard mask. Pour 200, entrer 55 dans la configuration du routeur (*$255-200=55$*).*

3.2.3 BGP

Border Gateway Protocol est un protocole de routage dynamique EGP standardisée par l'IETF utilisé entre des systèmes autonomes ainsi qu'entre les fournisseurs d'accès internet (ISP). En raison de sa grande adaptabilité, il est utilisé comme principal protocole pour relier toutes les SA constituant Internet (figure 3.4). Aujourd'hui, la table de routage BGP contient plus de 90'000 routeurs. Pour parvenir à ce résultat, BGP utilise de nombreux paramètres de routage, appelé attributs, afin de déterminer la politique de routage et maintenir un environnement de routage stable :

- **Weight** - Attribut local d'un routeur. S'il existe deux chemins pour la même destination, celle avec le plus haut poids sera préférée.
- **Local preference** - Utilisé pour précisé une préférence sur un point de sortie d'un SA.
- **Next Hop**
- ...

Les voisins BGP échangent des informations dès que la connexion TCP entre eux est établie. Lorsqu'un changement est détecté, seul ce changement est transféré et non toute la table entre les routeurs BGP. Aucune information périodique n'est également transmise.

3.2.4 IS-IS

Intermediate System to Intermediate System (IS-IS) est un protocole de routage dynamique IGP à état de liens, tel qu'OSPF. Ces deux protocoles ont beaucoup d'autres points communs, tels que la création d'une base de données de la topologie du réseau, l'inondation de paquet d'états de liens ou l'utilisation de l'algorithme de Dijkstra pour déterminer la meilleure route. Les deux supportent VLSM, peuvent utiliser du multicast pour découvrir leurs voisins et permettent aux routeurs de s'authentifier.
La grande différence est que OSPF est un protocole de couche 3 alors qu'IS-IS, lui, est un protocole de couche 2. L'adoption massive d'IP à donc eu comme conséquence la popularité d'OSPF, ce dernier ayant été créer pour IP. Cependant, IS-IS est plus économe et s'adapte mieux aux grands réseaux. IS-IS permet donc de fonctionner avec plus de routeurs dans une seule aire qu'OSPF. De plus, il est multi protocoles, ce qui lui permet de ne pas router uniquement des paquets IPv4 et de s'être facilement adapté à IPv6 contrairement à OSPF qui à du sortir une nouvelle version, OSPFv3, pour en être capable.
IS-IS diffère également d'OSPF dans la manière dont les aires sont définies et routées. Une notion de level est appliquée au routeur :

— **Level 1** : Routeur dans une aire, ne communique qu'avec d'autres routeurs level 1.
— **Level 2** : Route entre aires, ne communique qu'avec d'autres routeurs level 2.
— **Level 1-2** : Les deux, communique avec des routeurs level 1 ou 2.

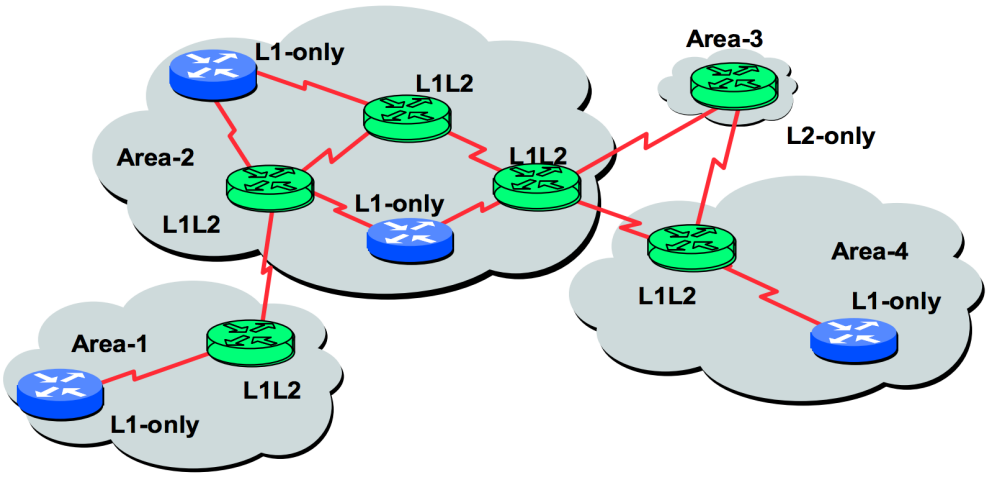

FIGURE 3.9 – Level des routeurs dans un réseau IS-IS

Dans OSPF, les interfaces font chacune partie d'une seule aire et les ABR les séparent alors que dans IS-IS la frontière de l'aire est l'interface elle-même. Il n'y a donc pas de notion d'aire backbone ici.

Avantages	Désavantages
Support multi protocoles.	Besoin d'activer CLNS sur le réseau.
S'adapte bien aux petits comme aux grands réseaux.	Traite les interfaces plus lentes avec le même métrique que les plus rapides.
Pas d'aire backbone sur laquelle toutes les données interzones doivent passer.	

TABLE 3.4 – Avantages & Désavantages d'IS-IS

4 Routage Privé

Lorsqu'un réseau local à besoin de communiquer avec un réseau distant, il est possible de le faire de deux manière différentes :

— Au travers d'un réseau publique tel Internet.
— Au travers d'un réseau privé.

Un réseau privé consiste généralement en un VPN permettant de créer un réseau privé par dessus un réseau publique. Cette technique à plusieurs avantages dont une réduction des coûts, une facilité de mise en place et une sécurité souvent identique à celle d'un réseau physiquement privé.

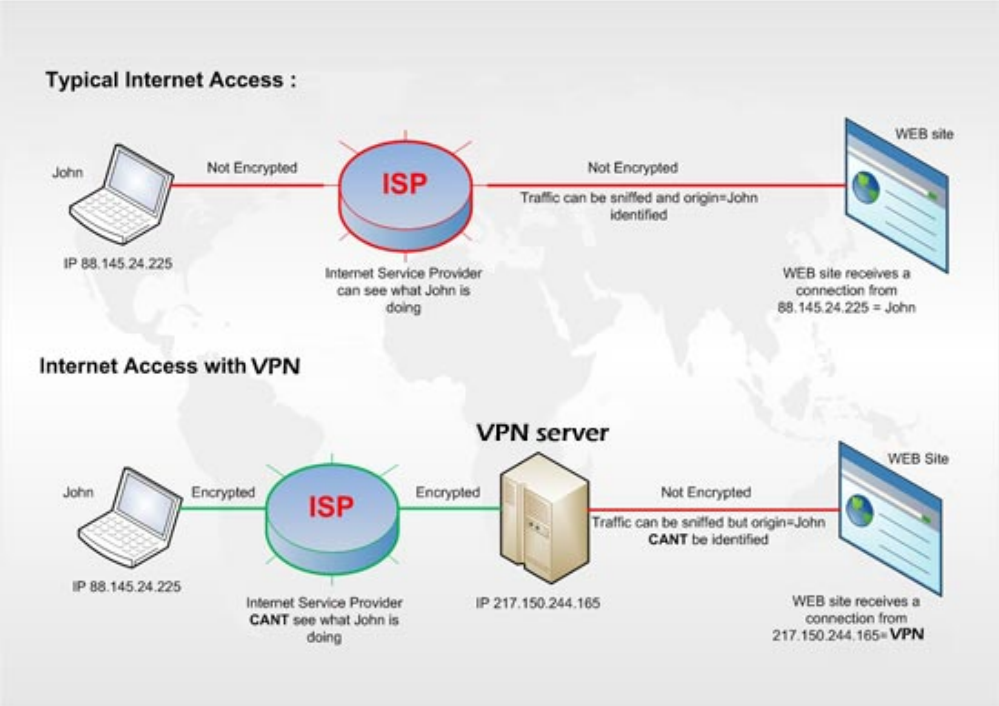

FIGURE 4.1 – Fonctionnement d'un VPN

Du côté de la mise en oeuvre, il doit y avoir un moyen de distinguer le trafic destiné au réseau public et le trafic destiné au réseau privé. Avec IPSec VPN, Split Tunnel décide quel trafic est destiné au Réseau Public et quel trafic est destiné au Réseau Privé.
Avec le protocole de routage, il pourrait être nécessaire d'implémenter le tunnel GRE (Generic Routing Encapsulation) en plus du tunnel IPSec. GRE est un protocole développé par Cisco permettant d'encapsuler des trames couche 2 telles que de l'Ethernet dans un réseau IP.

Remarque. *GRE n'est pas encrypté.*

Deuxième partie
OpenStack & Cloud

1 Introduction

Openstack est une plate-forme gratuite et open source pour le cloud-computing, basé sur une architecture modulaire composée de nombreux services principaux et secondaires regroupés en nodes(infrastructure en tant que service, IAAS). Chacun de ses nodes permet de contrôler les différentes ressources d'une machine virtuelle que sa puissance de calcule, son stockage ou son réseau. Il permet de déployer des instances de machines virtuelles gérant différentes tâches. Grâce à ses instances, plusieurs utilisateurs peuvent utiliser plusieurs services du cloud simultanément. Lancé en 2010 par Rackspace et la NASA, le projet est aujourd'hui soutenu par de nombreuses entreprises telles que Canonical, AT&T, Cisco, IBM ou encore VMware.

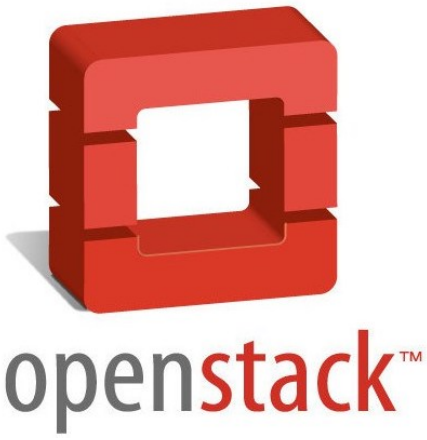

FIGURE 1.1 – Logo d'OpenStack

Un cloud OpenStack est composé de un ou plusieurs Network Nodes, Compute Nodes, Storage Nodes et Cloud Controller Nodes, chacun ayant un rôle bien défini :

- **Compute Nodes** - Le noeud de calcul gère un hyperviseur (KVM, ESX, Hyper-V ou XenServer). Le noeud de calcul gère le service de calcul (nova), la télémétrie (ceilometer) et le service d'agent de réseau ouvert vSwitch (neutron).
- **Network Nodes** - Le noeud réseau exécute des services de réseau (neutron). Il exécute les services neutroniques pour le layer 3, les métadonnées, DHCP et Open vSwitch. Le noeud réseau gère tous les réseaux entre les autres noeuds ainsi que le routage des instances. Il fournit des services tels que DHCP et IP qui permettent aux instances de se connecter à des réseaux publics.
- **Storage Nodes** - Le noeud de stockage exécute des services de stockage. Il gère le service d'image (glance), le stockage de blocs (cinder), le stockage d'objets (swift) et dans le futur stockage de fichiers partagés (manila).
- **Controller Nodes** -Panneau de contrôle Tableau de bord de l'environnement OpenStack. Le panneau de contrôle gère l'identité (keystone), le tableau de bord (Horizon), la télémétrie (ceilometer), l'orchestration (heat) et le service serveur de réseau (neutron).

La mission d'OpenStack est d'offrir une plate-forme de cloud computing open source pour les clouds privés et publics de grandes ou petites tailles, facilement implémentable et évolutif.

2 Cloud Computing

Le cloud computing est un procédé permettant de fournir des ressources et des données à d'autres machines à la demande. Le ou les serveurs fournissant ses services doivent avoir un logiciel conçu pour cet usage, c'est-à-dire en étant fiable et s'adaptant à la demande. OpenStack permet de fournir de nombreux services en fonctionnement telle une plateforme sur laquelle le développeur crée une application (de manière similaire à Windows, OSX ou Linux pour les ordinateurs personnels fournissant une interface entre les applications et le hardware).

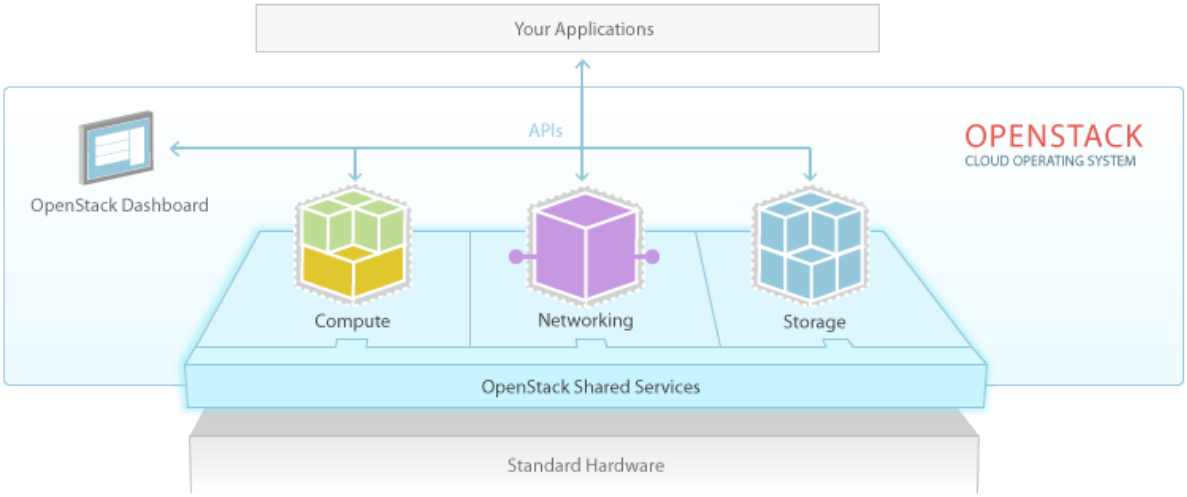

FIGURE 2.1 – Principe d'OpenStack

Le cloud computing est une des tendances IT les plus en vogue. Les infrastructures virtualisées offrent des avantages conséquents en comparaison aux datacenters traditionaux sur les performances, l'adaptabilité et la sécurité. Il existe néanmoins deux types de clouds ayant chacun leurs avantages et leurs inconvénients.

2.1 Cloud Publique

Les clouds publics appartiennent à des entreprises qui louent leurs services. Les clients bénéficient généralement d'un coût plus bas du fait que les coûts sont répartis entre tous les utilisateurs. Un second avantage est qu'il est souvent plus simple d'agrandir la taille du cloud, car il est déjà basé sur une infrastructure principale plus grande où il suffit d'allouer plus de ressource à l'un ou l'autre utilisateur.

Avantages	Désavantages
Moins cher, car basé sur un système "Pay as you go".	Impossible de parfaitement l'adapter à ses besoins si l'entreprise ne fournit pas le service voulu.
Mise en place et gestion simplifiée.	Moins sécurisé, les données n'étant pas stockées dans des serveurs appartenant au client.

TABLE 2.1 – Avantages & Désavantages des clouds publics

2.2 Cloud Privé

Afin d'avoir un contrôle total sur les données, un cloud privé est privilégié. Les compagnies avec des données sensibles ou des besoins spécifiques ne peuvent donc pas forcément utiliser un cloud public où les données sont dépendantes d'une solution externe.

Avantages	Désavantages
Répond à des besoins précis et spécifiques.	Plus cher.
Sécurité accrue.	Besoin de personnel compétent dans le domaine.
Customizable.	

TABLE 2.2 – Avantages & Désavantages des clouds privés

2.3 Types de services

En plus d'être soit privé, soit public, un cloud peut être catégorisé selon la manière dont il offre ses services :

- **Infrastructure as a Service (IaaS)** - Offre un service de bas niveau ou le client peut choisir son système d'exploitation et les applications qu'il souhaite y installer. Des machines virtuelles peuvent être dynamiquement louées sur de courtes périodes. Il est également possible de louer un ensemble de VM formant une infrastructure. OpenStack fonctionne sur ce principe.
- **Platform as a Service (PaaS)** - Le système d'exploitation est déjà installé et l'infrastructure est gérée par le fournisseur. Le client n'installe que les outils qui lui sont nécessaires par dessus. Exemple : hébergement web via LAMP.
- **System as a Service (SaaS)** - Dans ce dernier cas de figure, le système d'exploitation et les outils sont déjà installés et prêts à être utilisé. Le client ne s'occupe de rien. Exemples : Applications Web, Office 365.

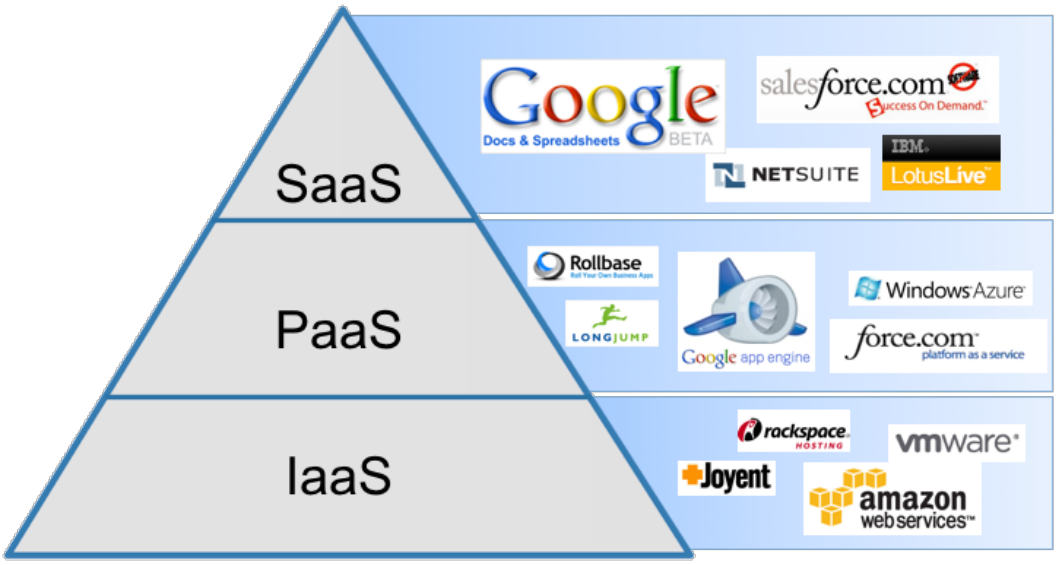

FIGURE 2.2 – Types d'architectures

3 Core Services

Les relations entre les services d'OpenStack peuvent être visualisées selon le schéma ci-dessus :

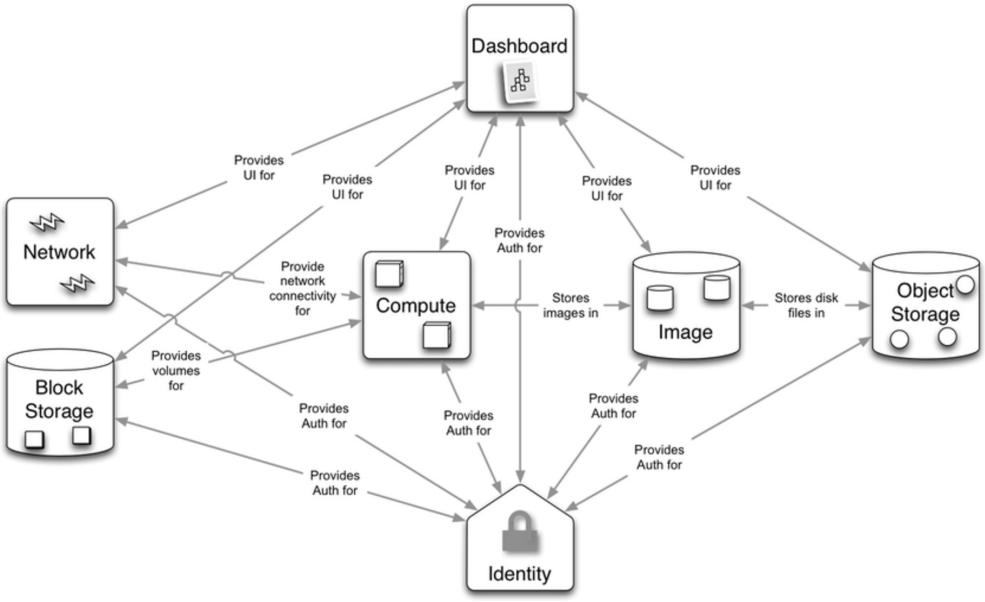

FIGURE 3.1 – Structure d'OpenStack

3.1 Swift

Swift est un service de stockage d'objet redondant et évolutif. Les objets et les fichiers sont écrits sur plusieurs disques répartis sur plusieurs serveurs différents. L'agrandissement de l'espace de stockage se fait simplement en augmentant le nombre de serveurs à disposition. Dans le cas d'une panne sur un disque ou sur l'entier d'un serveur, Swift se charge de répliquer les données au sein du cluster. L'intégrité des données est également gérée automatiquement par Swift. Puisque toute la logique de Swift est applicative, elle permet l'utilisation de matériel peu coûteux et non spécialisé.

3.2 Cinder

Contrairement à Swift, Cinder est un service offrant du stockage persistant en mode bloc. Contrairement à du stockage d'objet, les fichiers sont ici séparés en plusieurs blocs de même taille étant chacun stockés à une adresse. L'avantage de ce type de stockage est plus performant, car on peut modifier qu'un bloc d'un fichier sans devoir le charger en entier. En revanche, le stockage d'objet permet d'associer des metadata à chaque objet, ce qui le rend plus modulable.
Cinder gère l'attachement et le détachement de périphériques de stockage sur le serveur (mount, unmount). En plus du stockage local sur le serveur, Cinder peut utiliser de multiples plateformes de stockage telles que Ceph, EMC (ScaleIO, VMAX et VNX), GlusterFS, Hitachi Data Systems, IBM Storage (Storwize family, SAN Volume Controller, XIV Storage System, et GPFS), NetApp, HP (StoreVirtual et 3PAR) et bien d'autres.

3.3 Neutron

Neutron est un service permettant de gérer et manipuler les réseaux et l'adressage IP dans un environnement OpenStack. Les utilisateurs peuvent gérer et créer leurs propres réseaux, contrôler le trafic, la sécurité et connecter des instances à un ou plusieurs réseaux. L'adresse par instance peut être faite de manière statique ou dynamique via DHCP. Les adresses sont partagées entre les instances de machines virtuelles afin de leur donner une connectivité Internet uniquement lorsqu'elles en ont besoin.
Neutron permet de déployer des réseaux de types flat network, VLAN, VXLAN ou à tunnel GRE. Il est également possible d'utiliser des modules complémentaires pour commuquer avec des équipements ou logiciels de gestion de réseaux, tels que OpenVswitch, LinuxBridge ou Cisco Nexus.

3.4 Horizon

Horizon est le tableau de bord d'OpenStack. Il permet aux utilisateurs de gérer le cloud au travers d'une interface web graphique. Cette application, écrite en Python, peut être modifiée pour y faire apparaître par exemple son logo du fait qu'elle est open source et que son code est donc modifiable à souhait (comme tous les modules d'OpenStack).

3.5 Nova

Nova gère les ressources de calcul d'une ou plusieurs infrastructures en contrôlant les hyperviseurs. Un hyperviseur est une plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une même machine physique en même temps. Il en existe plusieurs, le mieux supporté par Nova étant KVM bien que Xen, ESX et Hyper-V sont également pris en charge.

3.6 Keystone

Keystone est un annuaire central contenant à la fois la liste des services et la liste des utilisateurs d'OpenStack. Il permet également de contrôler les rôles et autorisations de ces derniers. Tous les services et utilisateurs d'OpenStack utilisent Keystone pour s'authentifier entre eux.

3.7 Glance

OpenStack permet la découverte, l'envoi et la distribution d'image disque vers les instances au moyen du service d'image Glance. Glance permet également de stocker ces images créant un catalogue illimité de backups notamment en utilisant Swift. Le service implémente une interface REST pour requêter des images disques.

3.8 Ceilometer

Ceilometer est un service de télémétrie d'OpenStack mesurant différentes données sur l'utilisation du cloud comme le nombre d'instances lancé dans un projet et depuis combien de temps. Il est fréquemment utilisé pour fournir les métriques du service de facturation.

3.9 Heat

Heat est le composant d'orchestration d'OpenStack. L'orchestration est un procédé permettant de créer un service accessible pour manager l'entier du cycle de vie d'une infrastructure ou d'applications dans un cloud OpenStack. Heat permet de décrire une infrastrure sous forme de template appelé stack. Ce template est contenu dans un fichier texte lisible par un humain. Les stacks de Heat peuvent également s'adapter automatiquement grâce aux données fournies par Ceilometer. Si l'on souhaite modifier une infrastructure, il suffit simplement de modifier le template pour actualiser le stack actuel.

3.10 Trove

Les bases de données relationnelles et NoSQL sont gérées dans OpenStack au moyen de Trove. À ce jour les services de base de données supportés sont les suivants : MySQL, Redis, PostgreSQL, Mongodb, Cassandra, Couchbase et Percona.

4 Architecture

4.1 Controller Node

Le noeud contrôleur héberge le service d'identité, le service d'image, la partie management de compute et du réseau, plusieurs agents réseau, et le dashboard. Il inclut également les services support comme une base de données SQL, la file de message, et NTP (Network Time Protocol, méthode pour maintenir une horloge juste en communiquant avec une source précise).
En option, le noeud contrôleur peut faire tourner des parties de services de stockage par blocs, de stockage d'objets, d'orchestration et de télémétrie.
Il requiert au minimum deux interfaces réseau.

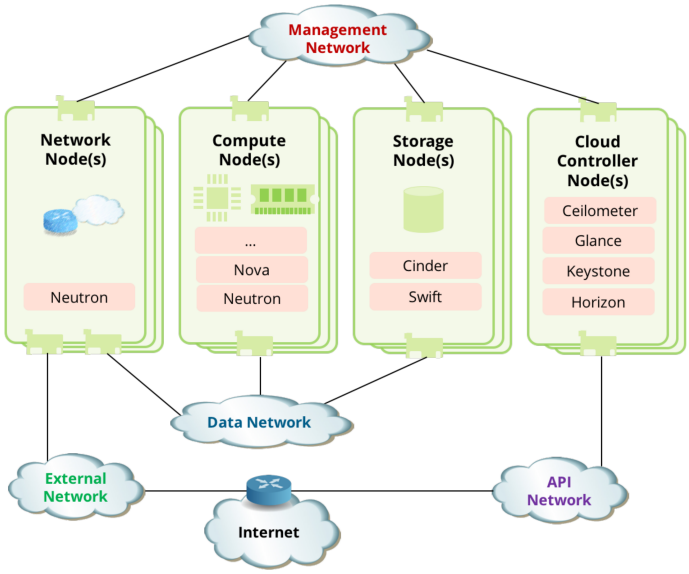

FIGURE 4.1 – Four-Node architecture

5 Compute Node

Le noeud compute exécute la partie hyperviseur de compute qui fait fonctionner les instances. Par défaut, compute utilise l'hyperviseur KVM. Le noeud compute héberge également un agent du service réseau qui connecte les instances aux réseaux virtuels et fournit des services de firewalling aux instances via les groupes de sécurité.
Il est possible de déployer plus d'un noeud compute. Chaque noeud nécessite au minimum deux interfaces réseau.

5.1 Stockage Node

Le noeud optionnel de Stockage par Blocs contient les disques que le service de Stockage par Blocs provisionne pour les instances.
Pour simplifier, le trafic du service entre les nœuds compute et ce nœud utilise le réseau de management. Les environnements de production devraient implémenter un réseau de stockage séparé pour accroitre la performance et la sécurité.
Vous pouvez déployer plus d'un nœud stockage. Chaque nœud nécessite au minimum une interfaces réseau. Stockage Objet Le noeud optionnel de stockage objet contient les disques que le service de stockage objet utilise pour stocker les comptes, les conteneurs et les objects.
Pour simplifier, le trafic du service entre les nœuds compute et ce nœud utilise le réseau de management. Les environnements de production devraient implémenter un réseau de stockage séparé pour accroitre la performance et la sécurité.
Ce service nécessite deux nœuds. Chaque nœud doit avoir au minimum une interface réseau. Vous pouvez déployer plus de deux nœuds de stockage objet.

5.2 Network Node

5.2.1 Réseau Fournisseur

L'option réseaux fournisseurs (Provider Networks) déploie le service réseau d'OpenStack de la façon la plus simple possible avec essentiellement des services de couche 2 (bridging/switching) et une segmentation des réseaux en VLAN. Principalement, il fait le lien entre les réseaux virtuels et les réseaux physiques et dépend de l'infrastructure réseau physique pour les services de couche 3. De plus, un service DHCP fournit les informations d'adresse IP aux instances. Les réseaux fournisseurs ne permettent pas de créer de nouveau VLAN mais uniquement d'utiliser ceux fournit par le fournisseur.

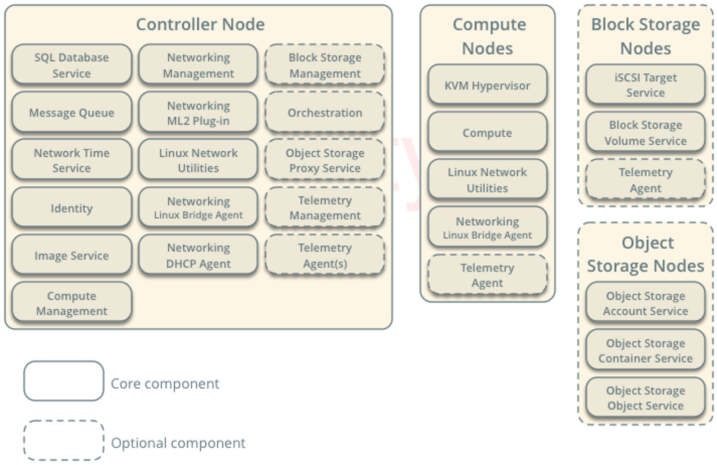

FIGURE 5.1 – Réseau fournisseur

5.2.2 Réseau Libre-Service

L'option de réseaux libres-services (Self-Service Network) améliore l'option de réseaux fournisseurs avec des services de couche 3 qui permettent la création de réseaux libres-services en utilisant des techniques de segmentation comme VXLAN. Essentiellement, cela permet de router les réseaux virtuels vers les réseaux physiques via le NAT. De plus, cette option sert de base aux services avancés comme LBaaS et FWaaS.

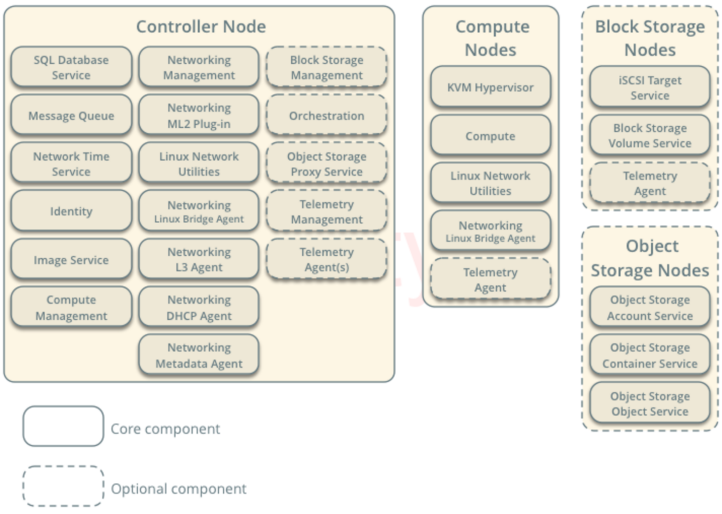

FIGURE 5.2 – Réseau libres-services

Liste des tableaux

Table des figures

6 Références

Routage statique et dynamique - it-connect.fr
Routage - wikipedia.org
Configuration du routage static - routeur.clement.com
Protocole RIP - idum.fr
Protocole RIP - wikipedia.org
Routing Loop Problem - wikipedia.org
Configurating RIP - cisco.com
Request for Comment - wikipedia.org
IETF - wikipedia.org
ISOC - wikipedia.org
OSPF - wikipedia.org
Avantages & désavantages d'OSPF - ipsit.bu.edu
OSPF Area - microsoft.com
BGP - cisco.com
Configuration OSPF - pearsonitcertification.com
OpenStack - opensource.com
openstack.org
OpenStack - wikipedia.org
Cloud privé et publique - mashable.com
Nodes définitions - keithtenzer.com
Types de cloud - loria.fr
Stockage de blocs et d'objets - druva.com
IS-IS - wikipedia.org
IS-IS pros and cons - networkenchancers.blogspot.ch Routage privé - networkenchancers.blogspot.ch

Fribourg, le 2 avril 2017

Samuel Riedo