République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche Scientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Études de MASTER ACADEMIQUE

Domaine : **Sciences et Technologies**
Filière : **Génie électrique**
Spécialité : **Télécommunication et réseaux**

Présenté par
Rabah AIT RAMDANE
Feriel KHELIFA

Thème

Développement d'une solution d'équilibrage de charge (Load balancer) sous Linux.

Mémoire soutenu publiquement le 08/07/ 2015 devant le jury composé de :

M. Mourad LAZRI
Maître de conférences classe A, UMMTO, Président

M. Fethi OUALLOUCHE
Maître de conférences classe B, UMMTO, Encadreur

M. Mounir SEHAD
Maître de conférences classe B, UMMTO, Examinateur

M. Slimane HAMEG
Maître de conférences classe B, UMMTO, Examinateur

Remerciement

Ce travail est le fruit et l'aboutissement de nos études à l'université Mouloud Mammeri de TiziOuzou.

Il n'aurait pu voir le jour sans le soutien de plusieurs personnes que nous tenons à remercier :

Notre promoteur M. OUALLOUCHE trouve ici l'expression de nos grandes reconnaissances pour les précieux conseils et les encouragements qui nous ont aidés à réaliser ce travail.

Enfin, que tous ceux qui nous ont prêté main forte et contribué à la réalisation de ce travail, et que nous n'avons pas pu désigner nommément, nous excusent et qu'ils sachent que nous ne les avons pas oubliés et que nous les remercions de tout cœur.

Dédicace

Je dédie ce travail à mon très cher père.

Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et L'exemple du dévouement qui n'a pas cessé de m'encourager et De prier pour moi.

Ta prière et ta bénédiction m'ont été d'un grand secours Pour mener à bien mes études.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

Je dédier aussi ce travail à ma maman qu'elle repose en paix et ma grand mère et ma chère copine Amina.

Khelifa feriel

Dédicace

Je dédie ce travail à ma famille qui m'a soutenu durant toutes mes années d'études en m'incitant toujours à aller de l'avant, ainsi qu'à tous mes amis.

Rabah

Résumé

Le développement de plus en plus d'applications qui nécessitent une haute disponibilité du support de communication avec une grande fiabilité, a suscité notre intérêt de développer une solution d'équilibrage de charge sur des liens disponibles tels que « ADSL, 4G, 3G, Satellites et autres ».

Notre travail est divisé en trois parties. La première est consacrée aux généralités sur les normes régissant les réseaux informatiques, le modèle théorique OSI avec ses sept couches et les règles qu'il préconise aux constructeurs de matériels réseaux sont traité dans cette partie, ainsi que le modèle TCP/IP car ce dernier est le plus utilisé dans la pratique à cause de son adoption par le réseau internet.

La deuxième partie fait le point sur les différents algorithmes de routage. Ces algorithmes permettent l'acheminement des données entres les réseaux en prenant en considération plusieurs paramètres comme le nombre de sauts, la bande passante … etc.

La troisième partie est consacrée au développement de la solution d'équilibrage de charge (load balancer). Elle est devisée en deux parties, la première est dédier à la présentation des outils Netfilter et iproute2 , qui sont utilisés pour la mise en œuvre du load balancer, la deuxième partie quant à elle présente la mise en pratique de ces outils pour la réalisation de la solution d'équilibrage de charge ainsi que les résultats des tests.

Mots clés : modèle OSI, TCP/IP, Routage, Load balancer, Netfilter.

Sommaire

Chapitre 2 : Étude des différents protocoles de routages dynamique et les algorithmes de sélection de la meilleure route

Chapitre 3 : Développement et tests

Glossaire

ABR: area border router

ARP : Address Resolution Protocol

AS: autonomous system

ASBR: autonomus system border

ASCII : American Standard Code for Information Interchange

ASN: autonomous system number

BGP : border gateway protocol

BR: backbone routers

BSD:berkelay software distribution

CIDR : Classless Inter-Domain Routing

DEC : Digital Equipment Corporation

DSA : Distributed System Architecture

EBCDIC : Extended Binary Coded Decimal Interchange Code

EIGRP: enhanced interior gateway routing protocol

FTP : File Transfer Protocol

IANA : Internet Assigned Numbers Authority

IGP: interior gateway protocol

IGRP: interior gateway routing protocol

IP : Internet Protocol

IR:router interne

ISO : International Standard Organization

LSA: link state announcement

LSP : link state packet

MAC : Media Access Control

MTU : Maximum Transmission Unit

NFS : Network File System

NIC : Network Interface Card

NIS : Network Information Service

OSI : Open System Interconnection

OSPF: open shortest path first

PPBP : Plus petite bande passante vers le Subnet en Kbits/s

RCP : Rich Client Platform

RFC : Request for Comments

RIP : protocole d'information de routage

ROM : Read Only Memory

RTP: reliable transfer protocol

SNA : Systems Network Architecture

SPF :shortest path first

TCP : Transmission Control Protocol

TI-RPC : Transport Independent Remote Procedure

VLSM: variable length subnet mask

XNS: xeros network system

Liste des figures

Introduction

Introduction

La grande évolution qu'a connue le réseau internet à partir des années 90 [1] a fait de celui-ci un outil de choix pour le partage des données et des ressources informatiques à l'échelle planétaire.

Sa grande diffusion que ça soit pour un usage professionnel ou personnel et les exigences des utilisateurs en matière de qualité de service, ont incité les chercheurs à développer au fil des années une multitude de technologies permettant d'accéder à ce réseau [2].

Du fait que chacune de ces technologies présente des avantages et des limites, des solutions de répartition de charge ont été élaborées afin de pouvoir bénéficier des bienfaits de chaque technologie [3], mais restants toujours onéreuses et peu adapter à une utilisation personnel.

Dans notre travail on se focalise sur l'étude et la réalisation d'une solution de répartition de charge (load balancing) permettant de devisé le flux réseau sur plusieurs accès vers internet.

Notre travail comporte trois chapitres ; le premier chapitre traite des généralités sur les réseaux et des normes permettant leurs bons fonctionnements, le second a été consacré l'étude des différents algorithmes de routage utilisés pour l'interconnexion des réseaux entre eux.

Le dernier chapitre a été devisé en deux parties dont la première présente les outils utilisés pour la mise au point de la solution de load balancing, la seconde a été consacrée à la mise en pratique de la méthode utilisée ainsi qu'à la présentation des résultats obtenus, et nous terminons notre travail par une conclusion et une bibliographie.

CHAPITRE I

Etat de l'art sur le modèle OSI et TCP/IP

1. Préambule

Au début des années 70, chaque constructeur a développé sa propre solution réseau autour d'architecture et de protocoles privés (SNA d'IBM, DECnet de DEC, DSA de Bull, TCP/IP du DoD,...) mais il s'est vite avéré qu'il serait impossible d'interconnecter ces différents réseaux propriétaires si une norme internationale n'était pas établie. C'est ainsi qu'en 1984 [1] que l'International Standard Organization (ISO) a mis en place le modèle de référence OSI Open System Interconnection (Interconnexion de Systèmes Ouverts).

Un système ouvert est un ordinateur, un terminal, un réseau, n'importe quel équipement respectant cette norme et donc apte à échanger des informations avec d'autres équipements hétérogènes et issus de constructeurs différents.

2. Définition

Le modèle OSI (Open System Interconnexion) est une norme qui préconise comment les ordinateurs devraient communiquer entre eux dans un réseau, il décompose les différentes opérations à effectuer pour établir une connexion entre deux machines en 7 étapes successives, qui sont nommées les 7 COUCHES du modèle OSI.

3. Les Couches Du Modèle OSI :

Le modèle OSI est un modèle à sept couches [1] qui décrit le fonctionnement d'un réseau à commutations de paquets. Chacune des couches de ce modèle représente une catégorie de problème que l'on rencontre dans un réseau.la figure ci-après présente la structure de ce modèle :

7	Couche Application
6	Couche Présentation
5	Couche Session
4	Couche Transport
3	Couche Réseau
2	Couche Liaison
1	Couche Physique

Fig.1 : Couches du Modèle OSI

Découper les problèmes en couche présente des avantages lorsque l'on met en place un réseau, il suffit de trouver une solution pour chacune des couches, elle permet également de changer de solution technique pour une couche sans pour autant être obligé de tout repenser.

Ainsi chaque couche garantit à la couche qui lui est supérieure que le travail qui lui a été confié a été réalisé sans erreurs.

En plus que chaque couche ait un rôle qu'elle doit respecter, le modèle OSI impose deux règles liant les couches :

- *chaque couche est indépendante :* les informations utilisées par une couche ne pourront pas être utilisées par une autre.

- *chaque couche ne peut communiquer qu'avec une couche adjacente :* cela assure que pendant l'envoie ou la réception, les données transmises vont parcourir toutes les couches du modèle OSI.

Le tableau suivant présente un résumé du rôle de chaque une de ces couches

N°	***Nom de la couche***	***Rôle***	***Rôle secondaire***	***Matériel utilisé***
1	PHYSIQUE	offre un support de transmission pour la communication	aucun	Hub (Concentrateur)
2	LIAISON DE DONNEES	connecter les machines entre elles sur un réseau local	détecter les erreurs de transmission	Switch (Commutateur)
3	RESEAU	interconnecter les réseaux entre eux	fragmenter les paquets	Routeur
4	TRANSPORT	gérer les connexions applicatives	garantir la connexion	
5	SESSION	Synchronisation des données et organisation du dialogue		
6	PRESENTATION	Codage des données en un langage connu par la couche supérieur		
7	APPLICATION	Désignation du type d'information à transférer		Proxy

Tableau 1. Organisation et rôle des couches du modèle OSI

3.1. La Couche 1 « Physique » :

Cette couche définit les propriétés physiques du support de données. Par exemple, dans le cas de câbles en cuivre, les méthodes de transmission sont différentes que celles utilisées sur une liaison par fibre optique. Selon la qualité du support, les vitesses de transmission sont naturellement très variables. La couche physique est représentée par le matériel de la carte réseau.

3.2. La Couche 2 « Liaison » :

La couche liaison assure La fiabilité de la transmission des données par la couche 1, sur le support réseau, Elle réalise cette fonction par l'établissement de sommes de contrôle (checksum), par la synchronisation de la transmission des données et par différents procédés d'identification et de correction d'erreurs. L'adressage des ordinateurs est réalisé dans cette couche par les adresses définies de manière fixe sur les cartes réseau. Dans le cas des cartes Ethernet, cette adresse est appelée adresse Ethernet ou adresse matérielle (MAC).

La couche liaison est matérialisée et exécutée par un logiciel résidant en ROM sur la carte réseau.

3.3. La Couche 3 « Réseau » :

La couche réseau prend en charge l'optimisation des chemins de transmission entre les ordinateurs distants. Les paquets de données sont transmis grâce à l'établissement d'une connexion logique entre les ordinateurs, qui peut comprendre plusieurs nœuds, L'adressage des ordinateurs est réalisé dans cette couche par des adresses logiques (par exemple des adresses IP) qui doivent être configurées sur chacun des ordinateurs.

Les protocoles chargés de la gestion de cette couche sont le protocole Internet Protocol (IP) de la famille TCP/IP et le protocole Internet Packet Exchange (IPX) de NovelI IPX/SPX

3.4. La Couche 4 « Transport » :

La couche transport prend en charge le pilotage du transport des données entre l'expéditeur et le destinataire. Cette fonction est réalisée par les protocoles TCP (Transmission Control Protocol) et UDP (User Datagram Protocol) de la famille des protocoles TCP/IP, ou par SPX (Sequenced Packet Exchange) de la famille Novell IPX/SPX.

3.5. La Couche 5 « Session » :

Cette couche gère l'échange des données sur la connexion établie par les couches 1 à 4. En particulier, c'est cette couche qui détermine lequel des ordinateurs connectés doit émettre les données et lequel doit les recevoir.

Le procédé Transport Independent Remote Procedure Call (TI-RPC), qui permet des appels de procédures sur des ordinateurs distants, indépendamment du protocole de transport est l'un des protocoles de cette couche. De nombreux procédés de connexion (Login) utilisent également un protocole de cette couche.

3.6. La Couche 6 « Présentation » :

C'est dans cette couche qu'est réalisée l'adaptation de la représentation des données en fonction de l'architecture des ordinateurs. Par exemple. L'échange de données entre un ordinateur central IBM. Qui utilise le codage de caractères EBCDIC, et un PC qui utilise le codage ASCII impose que les données soient d'abord converties au format réseau avant la transmission vers le destinataire. Celui-ci doit alors convertir les données reçues dans le format réseau pour les présenter dans Le format qu'il peut utiliser.

3.7. La Couche 7 « Application » :

La couche application est l'interface entre l'application et le réseau. Elle permet au modèle d'assurer l'indépendance de l'application vis-à-vis des accès réseau, exécutés par les couches inférieures. Certains programmes typiques utilisent cette couche, par exemple FTP ou RCP. Des services système comme NFS (Network File System) ou NIS (Network Information Service) exploitent également cette interface.

Nous trouvons dans cette couche les protocoles applicatifs. Ce sont des protocoles de haut niveau, destinés à permettre le dialogue entre applications serveurs et clientes comme HTTP, FTP, POP et SMTP.

4. Le Modèle TCP/IP

Le modèle TCP/IP reprend l'approche modulaire du modèle OSI (utilisation de modules ou couches) à l'exception que celui-ci n'en possède que 4 couches qui ont des taches plus diverses que celles du modèle OSI. La figure 2 montre une comparaison entre les couches du modèle OSI et celles du modèle TCP/IP

	Modèle OSI	Modèle TCP/IP
7	Couche Application	Application
6	Couche Présentation	
5	Couche Session	
4	Couche Transport	Transport
3	Couche Réseau	Internet
2	Couche Liaison	Accès Réseau
1	Couche Physique	

Fig.2 : comparaison des couches des modèles OSI et TCP/IP

4.1. Architecture :

Le Modèle TCP/IP est structuré en quatre couches de protocoles qui s'appuient sur une couche matérielle comme illustré dans la figure 2.

La couche Accès réseau est l'interface avec le réseau elle permet l'envoie est la réception des données sur le support de transmission.

La couche internet gère la circulation des paquets à travers le réseau en assurant leur routage. Elle comprend aussi les protocoles ICMP (Internet Control Message Protocol) et IGMP (Internet Group Management Protocol)

La couche transport assure tout d'abord une communication de bout en bout en faisant abstraction des machines intermédiaires entre l'émetteur et le destinataire. Elle s'occupe de réguler le flux de données et assure un transport fiable (données transmises sans erreur et reçues dans l'ordre de leur émission) dans le cas de TCP (Transmission Control Protocol). Pour UDP (User Datagram Protocol), il n'est pas garanti qu'un paquet arrive à bon port, il appartient à la couche application de s'en assurer.

La couche application est celle des programmes utilisateurs comme Telnet (connexion à un ordinateur distant), FTP (File Transfert Protocol), SMTP (Simple Mail Transfert Protocol), elle leurs permet d'exploiter les couches inferieurs.

4.2. Adressage

Pour intégrer une machine dans un réseau local, ou pour utiliser cette machine comme serveur d'accès à Internet on doit lui attribué une adresse.

Dans les réseaux il existe une adresse MAC (Media Access Control), qui est écrite en dur dans la ROM de l'interface réseau et donc théoriquement ineffaçable et infalsifiable. Cette adresse est réputée unique et décidée par le constructeur de la carte réseau. Elle est la seule adresse exploitée au niveau de la couche 2 pour l'identification des hôtes qui dialoguent. Cette méthode ne permettant pas l'interconnexion de réseaux, il a été nécessaire d'ajouter dans la couche supérieure (couche 3), une adresse logique qui sera attribuée par l'administrateur du réseau, en coordination avec les organismes chargés de gérer l'attribution de ces adresses.

Dans le cas de réseaux utilisant TCP/IP comme protocole de transport cette adresse est appelé adresse IP.

4.2.1. Définition :

Une adresse IP est constituée de 32 bits (4 octets), devisés pour des raisons de visibilité en quatre groupes de 1 octet chacun. Ces valeurs sont généralement exprimées sous forme de 4 chiffres décimaux compris entre 0 et 255 séparés chacun par un point. Ce mode de représentation est appelé « Dotted Decimal Notation », c'est-à-dire notation décimale pointée.

Une adresse IP est constituée de deux parties :

- Un identifiant réseau (Net-ID).
- Un identifiant machine (Host-ID).

Les adresses Internet sont réparties dans des classes d'adresses allant de A à E, dont seules les classes de A à C sont disponibles pour l'adressage normal.

Les parties de l'adresse séparées par un point n'indiquent pas quelle partie de l'adresse constitue l'identifiant réseau et laquelle désigne l'adresse de l'ordinateur. Les logiciels réseau identifient la classe d'adresses au contenu des 4 premiers bits du premier octet de l'adresse IP.

4.2.2. Les Classes des adresses IP :

Les Adresses de la Classe A :

Les adresses de la classe A sont formées d'un octet pour l'adresse réseau et de trois octets pour l'adresse de l'ordinateur. Les logiciels de communication identifient les adresses de ce type au fait que le bit de poids fort (MSB - Most Significant Bit) possède la valeur 0. Les adresses de la classe A ne peuvent donc retenir que des valeurs comprises entre O et 127. Les deux valeurs extrêmes, 0 et 127, ont une signification particulière. Les adresses de la classe A ne peuvent donc adresser que 126 réseaux comportant chacun 16777214 ordinateurs.

Les Adresses de la Classe B :

Ce type d'adresse est composé de deux octets pour l'adresse du réseau et de deux octets pour l'adresse de l'ordinateur. Les deux premiers bits de poids fort possèdent ici la valeur 10. Dans ce cas, le premier octet ne peut contenir que des valeurs comprises entre 128 et 191. Ce type d'adresse ne peut donc définir que 16382 réseaux comportant chacun 65534 ordinateurs.

Les Adresses de la Classe C :

Une adresse de la classe C est constituée de trois octets pour l'adressage de réseau, le dernier octet étant réservé pour l'adressage de l'ordinateur à l'intérieur du réseau. Les trois bits de poids fort possèdent la valeur 110. Le premier octet peut comporter des valeurs comprises entre 192 et 223.11 est donc possible d'adresser 2097150 réseaux de 254 ordinateurs chacun.

Les Adresses de la Classe D :

Cette classe d'adresses contient les adresses appelées Multicast. Elles permettent d'adresser simultanément des groupes d'ordinateurs. Un ordinateur peut ainsi posséder à la fois une adresse fixe et une adresse Multicast. Il ne peut cependant que recevoir des données par cette adresse, ce qui permet de configurer plusieurs ordinateurs sur la même adresse Multicast. Lorsqu'un paquet de données est transmis à ce type d'adresse, c'est l'ensemble du groupe d'ordinateurs qui est concerné. Les quatre premiers bits possèdent la valeur 1110. Le premier octet ne peut donc contenir que des adresses comprises entre 224 et 239.

Fig.3. Classes d'adresse IP

4.2.3. Les Adresses particulières :

L'Adresse de Loopback

L'adresse de classe A est réservée pour la fonction rebouclage (Loopback) d'un ordinateur. Par définition, toutes les adresses IP dont le premier octet possède la valeur 127 sont réservées aux tests internes du logiciel réseau. Si un paquet de données est adressé par exemple à l'adresse 127.2.5.10, il sera renvoyé à l'intérieur du réseau vers son expéditeur. Il est ainsi possible de vérifier que le logiciel local TCP/IP est installé et configuré correctement. Dans ce type de boucle de test (Loop), le paquet de données parcourt les couches OSI 7 à 3. Les couches matérielles I et 2 ne sont pas prises en compte dans ce test. Ainsi, il n'est pas possible de vérifier le fonctionnement de la carte réseau. En outre, le logiciel réseau vérifie seulement que le premier octet possède la valeur 127. Il est donc indifférent, pour le test de Loopback, que l'adresse ait la valeur 127.45.3.78 ou 127.200. 115.34. Dans les deux cas, c'est uniquement le logiciel réseau de l'ordinateur local qui sera contrôlé.

L'Adresse globale du réseau

Si l'on veut indiquer que l'on ne s'adresse pas à un ordinateur particulier à l'intérieur du réseau, mais à l'ensemble du réseau logique, il faut utiliser l'adresse globale du réseau. Pour cela, tous les bits de la partie Hôte l'adresse IP doivent être mis à « 0 ».

L'Adresse de diffusion (Broadcast)

S'il s'agit d'adresser tous les ordinateurs du réseau, il faut donner la valeur Pour cela, tous les bits de la partie Hôte l'adresse IP doivent être mis à «1».

Par exemple, dans le cas d'une adresse de classe B, deux octets sont utilisés pour adresser le réseau, les deux derniers octets permettent d'adresser un ordinateur dans ce réseau. Pour générer une adresse de diffusion pour ce réseau, il suffit de positionner à 1 tous les bits des octets qui adressent les ordinateurs.

4.3. Le Masque de sous-réseau

Le masque de sous-réseau indique quelle partie de l'adresse Internet est utilisée pour adresser le réseau, et laquelle est réservée à l'adressage d'un ordinateur particulier à l'intérieur du réseau logique. Le masque de sous-réseau n'a en principe aucune influence sur les paquets des données transmis par un ordinateur sur le réseau. Il influence par contre le fonctionnement du logiciel local de réseau, en lui indiquant comment l'adresse IP doit être

interprétée. Il existe un masque de sous-réseau par défaut pour chaque type de classe d'adresses, qui indique comment l'adresse doit être interprétée dans le cas normal.

Le tableau ci-après présente les valeurs correspondantes :

Classe d'adresse	***Valeur du 1er octet***	***Adresse réseau***	***Adresse de diffusion***	***Masque de sous-réseau***
A	1-126	XXX.0.0.0	XXX.255.255.255	255.0.0.0
B	128-191	XXX.XXX.0.0	XXX.XXX.255.255	255.255.0.0
C	192-223	XXX.XXX.XXX.0	XXX.XXX.XXX.255	255.255.255.0

Tableau.2. Masque de sous réseau par défaut

Dans certains cas particuliers, il peut être judicieux, pour assurer la transmission des paquets de données à travers différents ordinateurs intermédiaires (routage), de définir un masque de sous-réseau différent de la valeur par défaut.

4.4. La notation CIDR :

La méthode de distribution d'adresse par classes induit un gaspillage dans la mesure où des entreprises pouvaient se voir attribuer une classe complète et n'en utiliser qu'une partie.

Avec le modèle de notation CIDR (Classless Inter-Domain Routing), la notion de classe n'existe plus, si ce n'est pour les classes réservées à l'usage privé. Les adresses sont désormais distribuées par bloc, sans tenir compte de leur classe originale.

L'IANA distribue actuellement des blocs d'adresses contiguës, délimitées par un masque, toujours de 32 bits, dont les x bits de gauche sont à 1 et les autres à 0. Dans ce modèle de notation, un bloc d'adresse se définit ainsi : Adresse.de.Base/x

À titre d'exemple, l'adresse de classe C 192.168.1.0 avec le masque 255.255.255.0 s'écrirait :

192.168.1.0/24

Ici, nous avons toujours deux adresses remarquables :

- 192.168.1.0 qui symbolise tout le bloc ;

- 192.168.1.255 qui est l'adresse de broadcast pour ce bloc.

La souplesse de cette méthode CIDR réside dans le fait que l'on peut définir un bloc de la manière suivante 192.168.0.0/26

4.5. Les Adresses physiques (MAC)

L'adresse MAC désigne de manière unique une station sur le réseau. À des fins de facilité d'administration, elle est gravée dans l'adaptateur réseau (NIC, Network Interface Card) par le fabriquant. Pour garantir l'unicité d'adresse, c'est l'IEEE qui les attribue. L'IEEE propose deux formats d'adresse : un format long sur 48 bits et un format court sur 16 bits [2]. La figure 4 présente l'adressage IEEE, les bits sont représentés dans l'ordre d'émission sur le support (bits de poids faibles devant).

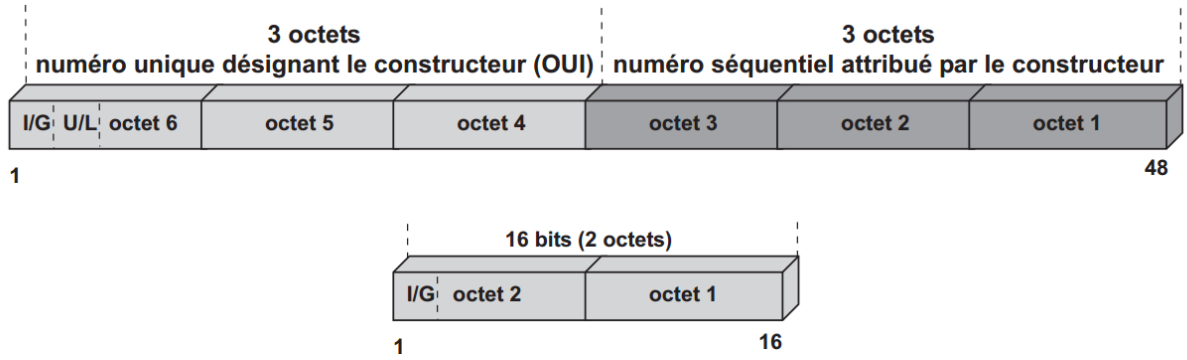


Fig.4. Adressage IEEE

Seul, en principe, l'adressage long est utilisé. Le premier bit (bit I/G) distingue une adresse individuelle ou unicast (I = 0) d'un adressage de groupe (multicast ou broadcast, I = 1). Le bit suivant (bit U/L) détermine si l'adresse qui suit est universelle : adressage IEEE (U = 0) ou local (U = 1). Dans ce dernier cas, c'est à l'administrateur de réseau de gérer l'espace d'adressage et de garantir l'unicité d'adressage. L'adressage IEEE est un adressage à plat, il désigne une machine mais ne permet pas d'en déterminer la position géographique.

Dans l'adressage universel, les 22 bits suivants désignent le constructeur ou le revendeur de l'adaptateur réseau. IEEE attribue à chaque constructeur un ou plusieurs numéros qui l'identifient (OUI, Organization Unit Identifier). Les 24 bits suivants appartiennent à une série séquentielle et sont inscrits dans l'adaptateur sous la responsabilité du fabricant (SN, Serial Number). La RFC 1340 récapitule la liste des numéros attribués.

Il est ainsi théoriquement possible de différencier 2^{24} constructeurs pouvant chacun produire 2^{24} cartes réseau. Du fait de la grande diffusion des systèmes en réseau, il est parfaitement possible qu'un constructeur produise plus de 2^{24} cartes réseau. Dans ce cas, un nouvel identifiant constructeur lui est affecté, lui permettant de produire à nouveau 2^{24} cartes réseau. Il est cependant peu probable que tous les identifiants constructeur puissent être épuisés.

4.6. Conversion d'adresses logiques (IP) en adresses physiques

Comme chaque carte réseau dispose d'une adresse physique unique, exploitable par la couche 2 du modèle OSI. À partir de ce niveau, l'adressage des ordinateurs est réalisé exclusivement grâce à leur adresse logique.

L'administrateur réseau ne pouvant cependant configurer que les adresses Internet (adresses logiques), il est impératif de procéder à la conversion entre ces adresses logiques et les adresses physiques correspondantes.

4.7. Conversion entre adresses physique en adresses logique (IP)

Lors de la transmission d'un paquet de données d'un ordinateur à un autre, chaque ordinateur n'est identifié que par son adresse Ethernet, définie dans la couche 2 du modèle OSI. La correspondance entre l'adresse Ethernet et l'adresse Internet accordée par l'administrateur réseau est réalisée par une simple table, qui associe l'adresse physique sur 48 bits et l'adresse Internet sur 32 bits.

Pendant la phase de développement de TCP/IP, ces tables devaient être gérées manuellement.

Lorsqu'une connexion devait être établie avec un ordinateur enregistré dans la table, celle-ci était parcourue pour identifier son inscription. Ce n'est qu'après cette identification que la connexion pouvait être établie. Ce processus assez statique était réalisable dans des réseaux de petite taille. Mais certains problèmes commençaient déjà à se poser sur des réseaux comportant plus d'une trentaine d'ordinateurs, lors du changement d'une carte réseau, car il était nécessaire d'assurer la mise à jour manuelle des fichiers de correspondance sur tous les ordinateurs.

Lors des développements suivants du protocole TCP/IP, la gestion de cette table a été automatisée grâce à un protocole utilitaire qui fait toujours partie de la pile des protocoles TCP/IP. Ce protocole est appelé l'ARP (Address Resolution Protocol).

4.8. Le protocole ARP :

La manière la plus utilisée aujourd'hui pour assurer la conversion dynamique des adresses Internet en adresses physiques consiste à mettre en œuvre le protocole ARP (Address Resolution Protocol). Ce protocole, implanté sur la couche 3 du modèle OSI, prend en charge la gestion des tables d'adresses décrites dans le passage précédent.

La table gérée par le protocole ARP est appelée Cache ARP. Le protocole de résolution d'adresses fonctionne de la manière suivante :

Avant de transmettre des données par le réseau, le protocole Internet (IP) interroge le mécanisme ARP pour connaître l'existence d'une entrée correspondant à l'adresse Internet cible recherchée.

Le protocole ARP compare l'entrée transmise par le protocole Internet et le contenu du cache ARP :

- Si une entrée est trouvée dans le cache ARP, l'ordinateur cible peut être adressé directement grâce à son adresse physique, et le paquet de données est transmis.

- Si aucune entrée adéquate n'est trouvée dans le cache, une requête d'interrogation (ARP Request) est transmise à tous les ordinateurs du réseau (broadcast), pour les interroger sur l'existence de l'adresse physique correspondante.

Seul l'ordinateur qui possède l'adresse Internet souhaitée répond par une requête de réponse (ARP Reply).

Les deux ordinateurs enregistrent l'information échangée dans leur cache ARP. Dès lors, l'ordinateur cible peut être adressé grâce à son adresse physique, et le paquet de données peut être transmis.

Pour éviter de devoir modifier manuellement le cache ARP, les entrées n'y sont pas permanentes. Une temporisation (ARP-Timer) définit la durée maximale de maintien. Si une entrée n'est pas utilisée pendant un certain temps, le temporisateur se déclenche et supprime l'entrée correspondante du cache.

A chaque utilisation d'une entrée de la table par une adresse IP, le temporisateur est remis à 0, pour éviter d'interroger à chaque fois une adresse fréquemment utilisée.

Ce procédé fonctionne entièrement en arrière-plan et ne nécessite aucune action de maintenance ou de configuration.

4.9. Format des datagrammes IP :

IP (Internet Protocol) a été défini par la RFC 791, c'est un protocole de niveau réseau qui a pour objet l'interconnexion de réseaux hétérogènes.

Les messages transmis par IP sont appelés des datagrammes (Figure5). Certains datagrammes sont des fragments d'un datagramme qui a dû être fragmenté.

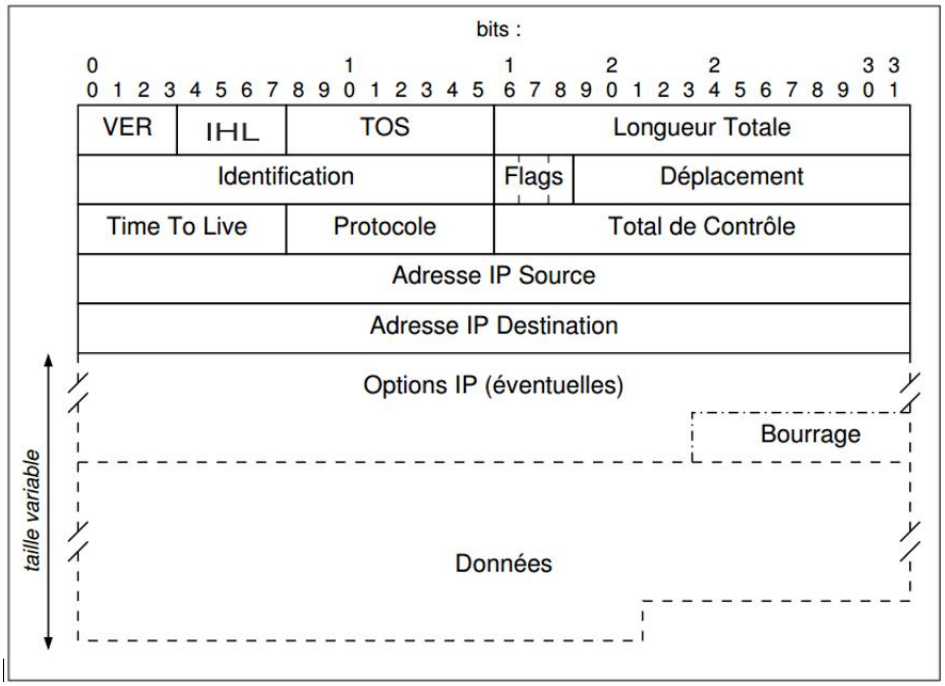


Fig.5.Format d'un datagramme IP

Comme tous les protocoles réseaux d'Internet, tous les champs du datagramme IP sont exprimés en ordre réseau (Network Byte Order), c'est-à-dire que si une valeur tient sur plusieurs octets, le premier octet transmis est l'octet de poids fort. Sur un octet, le premier bit transmis est le bit de poids fort. Ceci est précisé par la RFC 1700.

Le numéro de version sur 4 bits **(VER)** permet d'identifier le format du datagramme, c'est-à-dire la version du protocole IP utilisé. La présence de cette information autorise la cohabitation de plusieurs versions de protocole dans les systèmes intermédiaires, ce qui est indispensable lors de la mise à jour d'une version du protocole IP. La version courante est la version 4, cependant la version 6 (IPv6) est en cours de déploiement dans l'Internet.

Le champ longueur d'en-tête sur 4 bits (**IHL**, Internet Head Lenght) indique, en multiple de mots de 32 bits, la longueur de l'en-tête. La valeur courante, lorsqu'aucune option n'est invoquée, est 5 (20 octets)

Le champ type de service sur 8 bits (**TOS**, Type Of Service) spécifie, à la passerelle inter réseau, le type d'acheminement attendu.

Le champ Longueur Totale sur 16 bits indique la taille totale en octets du datagramme (ou du fragment). Ainsi, un datagramme ne peut pas excéder 65535 octets ($2^{16} - 1$). La norme impose à toute implémentation de pouvoir traiter des datagrammes d'au moins 576 octets. Si un datagramme devant traverser un réseau est de taille supérieure à ce que le réseau peut

transmettre il doit être fragmenté par le routeur ou la station l'injectant dans le réseau. Fragmenter veut dire que le datagramme sera découpé en datagrammes plus petits qui pourront être transmis. Ils auront pour Longueur Totale la taille des données qu'ils transportent plus la longueur de l'en-tête indiquée dans le champ IHL. Le datagramme original sera reconstitué par le destinataire.

Le champ identification (ID) sur 16 bits : la valeur du champ ID, attribuée par la source, est générée de manière aléatoire par un algorithme initialisé par l'heure système.

En cas de fragmentation, l'ID est recopiée par les systèmes intermédiaires dans tous les fragments du datagramme d'origine. L'ID permet, à l'hôte destinataire, d'identifier les différents fragments d'un même datagramme, il facilite ainsi le réassemblage.

Le champ Flags est composé de 3 bits dont le premier n'est pas utilisé. Le bit suivant dit bit **DF** (Don't Fragment) demande au système intermédiaire de ne pas fragmenter le datagramme (bit à 1). Ce bit est utilisé, par exemple, quand le système d'extrémité est incapable de réassembler les différents fragments.

Le système intermédiaire qui reçoit un tel datagramme doit soit le router dans sa totalité sur un sous-réseau où le MTU est compatible soit le détruire. En cas de destruction, il en avertit la source par un message ICMP. Enfin, le bit **MF** (More Fragment) est positionné à 1 dans tous les fragments d'un même datagramme d'origine pour indiquer qu'un fragment suit. Il est à 0 dans le dernier fragment ou lorsqu'un datagramme n'a pas subi de fragmentation.

Le champ déplacement (offset, 13 bits) indique, en cas de fragmentation, la position du fragment dans le datagramme d'origine. Ce champ indique la position du premier bit du fragment dans le datagramme d'origine, en multiple de 8 octets. En conséquence, tous les fragments, sauf le dernier, ont une longueur multiple de 8.

Le champ duré de vie (TTL, Time To Live) sur 8 bits détermine, en seconde, la durée de vie d'un datagramme. Cette valeur est décrémentée toutes les secondes ou à chaque passage à travers une passerelle. Lorsque le TTL est égal à 0, le datagramme est détruit. La passerelle qui détruit un datagramme envoie un message d'erreur ICMP à l'émetteur. Aucune estampille de temps ne figurant dans l'en-tête IP, les passerelles (routeur) n'ont pas la possibilité de mesurer le temps écoulé, elles se contentent alors de décrémenter ce champ de 1 unité.

Le champ protocole, 8 bits, Sert au démultiplexage car indique à quel protocole il faut remettre les données transportées dans le datagramme. Les valeurs possibles de ce champ et leur signification sont décrites dans la RFC 1700

Le champ Checksum (Contrôle de l'entête, 16 bits), Permet de contrôler l'intégrité de l'entête (mais pas des données). Si le Checksum calculé par le destinataire est différent de celui figurant dans le datagramme, celui-ci est détruit.

Les champs Adresse IP Source et Destination, 32 bits chacun, ce sont des entier non signé identifiants respectivement l'adresse IP de l'émetteur et du récepteur du datagramme. On représente une telle adresse en notation décimale pointée.

Le champ Options (taille variable, pouvant être nulle), il comprend la découverte du MTU, l'enregistrement d'une route suivie par un datagramme, le routage à la source, etc. En cas de fragmentation, certaines options sont copiées dans tous les datagrammes (comme le routage à la source), d'autres ne le sont que dans le premier (comme enregistrement de la route).

Le champ Bourrage (Taille variable, pouvant être nulle), n'est présent que pour compléter la taille des options jusqu'à un multiple de 4 octets. Ceci parce que la taille de l'en-tête est IHL × 4 octets.

Le Champ Données (taille variable), il contient les données véhiculées par le datagramme. Sur la station destinataire du datagramme, ces octets seront communiqués à l'entité (protocole) indiquée par le champ Protocole si le Checksum est confirmé. La taille maximale de ce champ est 65535 moins la longueur de l'en-tête. Le champ de données d'un datagramme IP peut contenir un segment TCP, un message ICMP, ARP, RARP ou encore OSPF.

5. Discussion

Dans ce chapitre nous avons vu les concepts de base des réseaux informatique ainsi que les normes régissant leurs gestion on citant le modèle théorique de référence OSI avec ses couches, le modèle TCP/IP étant celui utilisé par le réseau internet a été détaillé dans ce chapitre. le chapitre suivant sera consacré aux algorithmes de routage permettant l'interconnexion des différents réseaux entre eux.

CHAPITRE II

Etude des différents protocoles de routages dynamique et les algorithmes de sélection de la meilleure route

1. Préambule

Afin que les informations puissent circuler d'un réseau à un autre, il faut que le dispositif employé identifie comment transporter ces informations. Le routage est le procédé par lequel les paquets sont acheminés d'un nœud vers un autre.

2. Principe du routage

Pour transférer des paquets d'un serveur expéditeur à un serveur destinataire, la couche réseau doit tout d'abord déterminer le parcours ou route, à emprunter. Que celle-ci propose un service à datagrammes (auquel cas les différents paquets envoyés par un même expéditeur à un même destinataire peuvent emprunter des parcours différents) ou un service à circuit virtuels (auquel cas tous les paquets échangés entre un même couple de systèmes distants empruntent le même parcours), elle doit impérativement déterminer une route précise pour chaque paquet qu'elle envoie. Cette fonction incombe tout naturellement au protocole de routage de la couche réseau.

Un serveur est généralement directement connecté à routeur spécifique, qui est son routeur par défaut. Lorsque l'expéditeur envoie un paquet, c'est le premier routeur qu'il rencontre sur son parcours.

Tout protocole de routage repose sur un algorithme (dit algorithme de routage) chargé de déterminer le parcours qu'un paquet doit suivre entre le routeur source et le routeur destinataire. Le rôle d'un tel algorithme est très simple : en présence d'un groupe de routeurs donné, reliés les uns aux autres par des liaisons physiques, sa mission consiste à trouver le bon parcours entre les routeurs source et destinataire. [4]

3. Type d'algorithmes de routage

D'une manière générale, les algorithmes de routage se divisent en algorithmes globaux ou décentralisés :

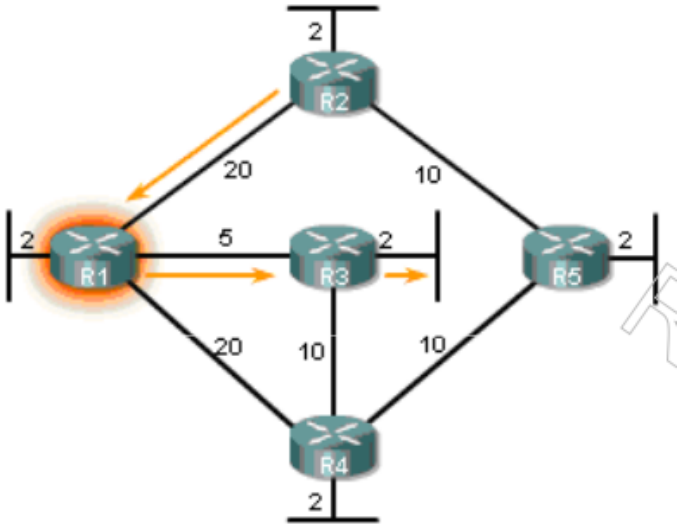
3.1. Les algorithmes globaux

Un algorithme de routage global détermine le parcours le moins onéreux entre la source et la destination au moyen d'une connaissance globale du réseau. Autrement dit, l'algorithme interprète les liaisons entre les nœuds et leurs coûts comme étant des entrées. Mais avant de pouvoir effectuer ce calcul, celui-ci doit tout d'abord avoir accès à ces informations. Le calcul en lui-même peut être effectué en un point donné ou en plusieurs endroits. La

principale différence entre ces deux solutions réside dans le fait qu'un algorithme global dispose d'informations complètes concernant le coût associé aux différentes liaisons. Dans la mesure où les algorithmes doivent être informés de la valeur de chaque liaison du réseau, on parle souvent d'algorithme d'état de lien.

3.1.1. Algorithme de routage par état de lien

Un algorithme par état de lien (Link state) utilise la connaissance de la topologie du réseau et du coût des différentes liaisons pour ses calculs. Par ce protocole, chaque nœud du réseau diffuse à l'ensemble de ses homologues l'identité et la valeur des liaisons auxquelles il est attaché. Cette diffusion d'état de lien peut s'effectuer sans que les différents nœuds aient initialement connaissance de l'existence de tous les autres nœuds du réseau. Un nœud donné doit simplement connaître l'identité de ses voisins directs ainsi que la valeur de la liaison y conduisant. Ainsi, il peut découvrir le restant de la topologie du réseau grâce aux informations d'état de lien qui lui sont communiquées en retour. Grâce à ces informations d'état de lien, tous les nœuds disposent des mêmes informations sur le réseau, chacun pouvant utiliser l'algorithme individuellement et obtenir ainsi les mêmes résultats que ses homologues.



Destination	Shortest Path	Cost
R2 LAN	R1 to R2	22
R3 LAN	R1 to R3	7
R4 LAN	R1 to R3 to R4	17
R5 LAN	R1 to R3 to R4 to R5	27

Fig.6. Protocole de routage par état de lien

3.1.2. Protocoles de routage état de lien

Les protocoles de routage par état de lien (link state protocols) s'appuient, pour construire les tables de routage, sur une base de données topologique. Cette base de données est élaborée à partir des paquets d'état de lien, paquets qui passent dans tous les routeurs pour informer de l'état du réseau. L'algorithme du plus court chemin, appelé algorithme SPF (shortest path first), utilise cette base de données pour la construction des table de routage. Ainsi l'algorithme SPF maintient une base de données complexe, contenant des informations sur la topologie du réseau.

3.1.3. Processus du routage état de lien

1. Chaque routeur prend connaissance de ses propres liens et des réseaux qui lui sont connectés directement. Cette opération s'effectue en détectant qu'une interface se trouve dans l'état up.

2. Chaque routeur est responsable de la détection de ses voisins sur les réseaux connectés directement. Les routeurs à état de liens effectuent cette détection en échangeant des paquets Hello avec d'autres routeurs à état de liens situés sur des réseaux connectés directement.

3. Chaque routeur construit un paquet LSP (Link-State Packet) contenant l'état de chacun des liens connectés directement. Il procède en enregistrant toutes les informations pertinentes sur chaque voisin, notamment l'ID du voisin, le type de lien et la bande passante.

4. Chaque routeur inonde tous les voisins avec des paquets LSP. Les voisins stockent tous les paquets LSP reçus dans une base de données. Ensuite, ils diffusent les paquets LSP à leurs voisins jusqu'à ce que tous les routeurs de la zone aient reçu ceux-ci. Chaque routeur stocke une copie de chaque LSP reçu de ses voisins dans une base de données locale

5. Chaque routeur utilise la base de données pour élaborer une carte complète de la topologie et calcule le meilleur chemin vers chaque réseau de destination. Le routeur possède ainsi une carte complète s'apparentant à une carte routière de l'ensemble des destinations de la topologie et des routes pour les atteindre. L'algorithme SPF sert à construire la carte de la topologie et à déterminer le meilleur chemin vers chaque réseau

3.1.4. Protocole de routage OSPF (priorité au plus court chemin)

L'OSPF est l'un des protocoles de routage les plus utilisés aujourd'hui. Il possède de très bons atouts, et convient très bien aux grands réseaux. De plus, c'est un protocole standard, et donc utilisable par tous les constructeurs.et il sert au routage interne.

A la base, il s'agit cependant toujours d'un protocole à état de lien ayant recours à la technique de l'inondation d'information et à un algorithme de dijkastra de chemin à moindre coût. Avec l'OSPF, les routeurs élaborent une carte topologique complète du système autonome, puis ils utilisent l'algorithme pour établir un arbre de plus court chemin vers tous les réseaux du système, se constituant eux-mêmes en nœud racine. C'est à partir de cet arbre qu'on obtient ensuite le contenu des tables de routage, tandis que la valeur des liaisons individuelles est définie par l'administration de réseau. L'OSPF n'impose pas de règles strictes concernant la méthode à employer pour la valorisation des liaisons. Il procure simplement les mécanismes permettant de déterminer le chemin de moindre coût à partir des valeurs données. Avec l'OSPF les routeurs communiquent leurs informations de routage à tous les routeurs de leur système autonome, et pas seulement à leurs voisins directs. Ces informations sont communiquées à chaque changement d'état d'une liaison, et ceci de manière périodique, même si la liaison n'a pas changé d'état entre temps. Les annonces OSPF sont contenues dans des messages véhiculés directement par la couche IP, avec un protocole de couche supérieur valant 89 pour l'OSPF. Ainsi, le protocole OSPF est lui-même en charge de fonctions telles que la fiabilité du transfert de messages et la diffusion d'indications d'état de lien. Il vérifie également le bon état des liaisons (via des messages HELLO échangés entre routeurs voisins) et permet aux routeurs OSPF d'avoir accès à la base de données de leurs voisins concernant l'état des liens au sein du réseau tout entier.

A. Les différents types de zones OSPF

Chaque area, constituée d'un ensemble de routeurs, forme un domaine logique. Sur le schéma des area standards et la Backbone area0. Cette dernière assure l'interconnexion des autres area. Chaque area doit impérativement être reliée à la Backbone area. La Backbone area, en plus de transmettre les paquets d'une area l'autre doit également faire parvenir à chaque area les informations concernant les autres area, comme par exemple le coût pour atteindre chacune de celles-ci. Les LSA par contre ne sont diffusés qu'à l'intérieur de l'area concernée par la mise à jour.

Les routeurs OSPF envoient un LSA (Link state Announcement ou Advertisment)

– quand l'état d'une ligne change

– ou toutesles30 minutes.

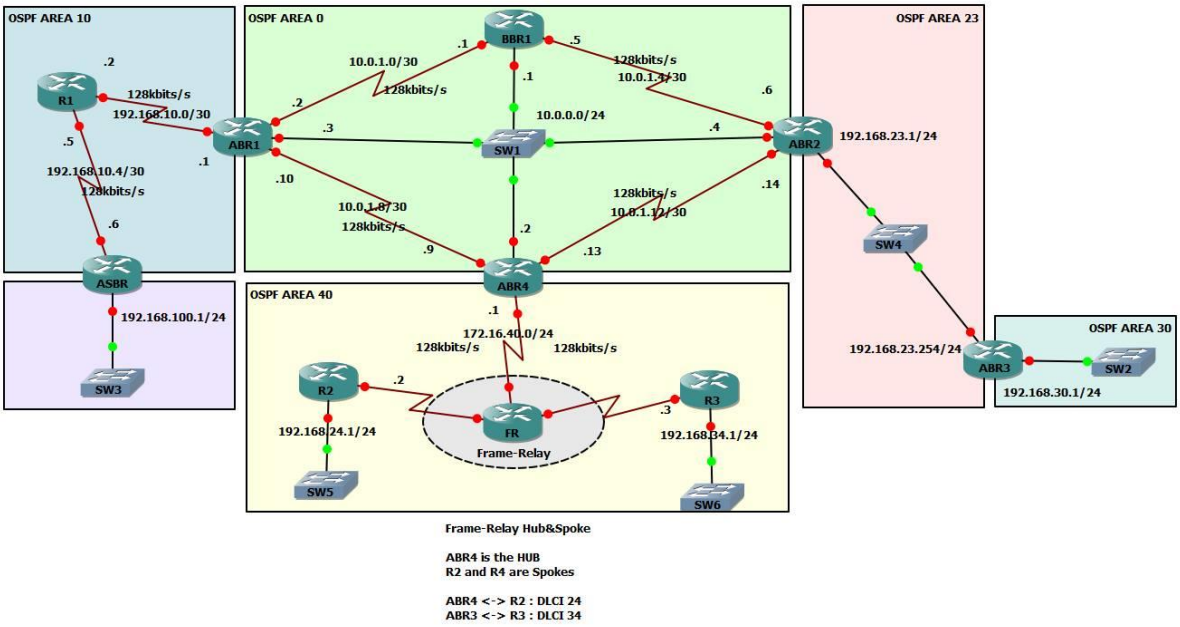


Fig.7. Schéma des area standards et la Backbone area0.

B. Les différents types de routeurs OSPF

OSPF distingue différents types de routeurs en fonction de leur situation dans les différentes area. Chaque routeur doit appartenir au moins à une area. S'il appartient à plusieurs area, il doit obligatoirement appartenir à la Backbone area. Il doit maintenir à jour la topologie de chaque area dont il fait partie ainsi que la table de routage associée à celle-ci. Les routeurs appartenant à plusieurs areas auront donc plusieurs topologies et tables de routage en mémoire. On y distingue les routeurs internes (IR) qui n'appartiennent qu'à une area qui n'est pas la Backbone area ; les Backbone routeurs (BR) qui ne sont connectés qu'à la Backbone area, ce sont donc les routeurs internes à l'area 0. On y voit également les Area Border Router (ABR) qui sont connectés à plusieurs area (dont l'area 0), ce sont eux qui effectuent l'agrégation de routes pour l'area. Il peut y en avoir plusieurs pour une même area. Enfin les Autonomous System Border Router (ASBR) assurent l'interconnexion avec un autre système Autonome qui utilise éventuellement un protocole différent d'OSPF.

C. La métrique de l'OSPF

Quand un routeur OSPF est initialisé, il tente de se faire connaître aux autres routeurs en envoyant un message Hello à l'adresse multicast 224.0.0.5. Si le réseau ne gère pas la diffusion, une configuration manuelle avec l'adresse du routeur voisin s'impose.

La métrique utilisée associe un coût à chaque lien sur base de sa bande passante (BW) de la façon suivante :

Cout =108 BW

108 est la bande passante de référence et BW est la valeur nominale, en bits/s, pour l'interface considérée.

Par défaut toutes les interfaces ont la même bande passante et la métrique est donc équivalente au nombre de sauts utilisé par RIP. Il faut donc configurer chaque interface pour tenir compte des spécificités de chaque lien. Le coût d'un lien est donc uniquement lié à son type.

Le coût associé à un chemin est la somme des coûts des interfaces traversées. La métrique utilisée est déjà meilleure que celle de RIP mais elle ne prend en compte aucun critère lié au caractère dynamique du réseau comme par exemple la charge sur les liens ou le délai de transfert des paquets. OSPF permet lui aussi le partage de charge entre routes de même coût. Il est possible de définir des métriques multiples mais il faudra calculer autant de tables que de métriques. Tous les nœuds doivent utiliser les mêmes métriques.

3.1.5 Le protocole EIGRP

L'EIGRP (Enhanced Interior Gateway Routing Protocol) est un protocole propriétaire Cisco. Il a été développé en améliorant le protocole IGRP, l'ancêtre de l'EIGRP.

Il s'agit d'un protocole de routage interne (IGP), Cela veut dire qu'il est capable de réaliser du routage au sein d'un Autonomous System. Pour faire simple, un AS c'est un ensemble de réseaux sous une même autorité. Sur ces réseaux nous utilisons des protocoles dits IGP (Interior Gateway Protocol). Entre les AS, c'est-à-dire sur internet, nous utilisons des protocoles EGP (tel que BGP).[5]

L'EIGRP est un protocole de routage à vecteur de distance avancé qui joue le rôle d'un protocole à état de liens lors de la mise à jour des voisins et de la gestion des informations de routage. Par rapport aux protocoles à vecteur de distance simples, l'EIGRP offre notamment les avantages suivants :

- Supporte le VLSM (Variable Length Subnet Mask), c'est donc un protocole de routage « classless ».
- Fonctionne sur base de l'algorithme DUAL pour une sélection efficace des routes tout en évitant les boucles.
- Relations d'adjacence avec les routeurs voisins.
- Transmissions des messages en multicast (224.0.0.10) et unicast
- Supporte plusieurs protocoles de la couche réseau : IPv4, IPv6, AppleTalk, IPX, …
- Load-balancing et notamment sur routes ayant des métriques différentes.
- « Summarization » à n'importe quel endroit du réseau.
- « Auto-summarization », par défaut, entre réseaux majeurs (entre deux réseaux classfull).
- Echange des messages entre routeurs assuré par RTP (Reliable Transfer Protocol).
- Métrique tenant compte de la bande passante et du délai des interfaces. D'autres paramètres peuvent être configurés en plus (fiabilité et charge de l'interface).
- Distance administrative pour les routes internes : 90 (valeur par défaut)
- Distance administrative pour les routes externes : 170 (valeur par défaut)

A. La métrique du protocole EIGRP

Par défaut, EIGRP calcule la métrique d'un subnet en tenant compte de la bande passante et du délai des interfaces. D'autres éléments peuvent être configurés pour entrer dans le calcul : la fiabilité de l'interface et la charge de l'interface.

La métrique (avec les paramètres par défaut) se calcule comme suit :

$$\text{Métrique} = ((10.000.000 / \text{PPBP}) + \text{S (délais)}) * 256 \ldots\ldots (1)$$

PPBP : Plus petite bande passante vers le subnet en Kbits/s
S (délais) : Somme des délais des interfaces vers le subnet exprimé en 10µs (dizaine de µs)

EIGRP utilise donc 4 attributs pour juger les routes qu'il apprend :

– **K1** : Bande passante (utilisé par défaut)

– **K3** : Délai (utilisé par défaut)

– **K2** et **K4** : fiabilité

– **K5** : charge

Le MTU est annoncé, mais n'est pas utilisé dans le calcul de la métrique.

B. Fonctionnement

Tout d'abord, EIGRP fonctionne sur la base d'un numéro de système autonome « Autonomous System Number » ou « ASN ». C'est-à-dire qu'il pourra uniquement communiquer avec les routeurs où EIGRP est configuré pour le même ASN.

Ensuite, une fois qu'on l'a activé sur une interface, que ce soit de manière dynamique ou statique, EIGRP tente de découvrir des voisins potentiels pour cela il y envoi des messages « HELLO ».
Lorsque deux routeurs reçoivent des messages HELLO l'un de l'autre, ils vérifient alors les conditions d'adjacence afin de décider si oui ou non ils deviendront des voisins EIGRP (Neighbors).

Pour que deux routeurs deviennent voisins EIGRP ils doivent remplir les conditions suivantes :

- Fonctionner dans le même AS (Autonomous System), donc être configuré avec le même ASN.
- Les deux routeurs doivent pouvoir s'envoyer et recevoir des packets IP.
- Les interfaces doivent être configurées avec une adresse IP dans le même subnet.
- L'interface concernée ne doit pas être configurée comme passive.
- Les valeurs K (valeurs qui définissent le calcul de la métrique) doivent correspondre.
- L'authentification EIGRP (si configurée) doit être passée avec succès.

Si ces différentes conditions sont vérifiées, les deux routeurs se considèrent alors comme voisins EIGRP, ajoutent cette relation dans leur table de voisinage, et commencent à s'échanger des informations.

Lorsqu'une relation de voisins vient de s'établir, chaque routeur commence par envoyer la totalité de ses routes connues pour lesquelles il a une interface active et configurée dans EIGRP. Par la suite, seules les modifications seront envoyées.

Afin de garantir une certaine stabilité, les routeurs s'échangent en permanence des messages HELLO. Ces messages HELLO sont envoyés à intervalles réguliers et ont une durée de vie. Si un des deux routeurs n'a pas reçu de nouveau HELLO avant que la durée de vie du précédent soit écoulée, le routeur voisin est considéré comme défaillant, l'adjacence est rompue et les routes reçues par ce voisin sont retirées de la table de routage.

Chaque routeur garde en mémoire toutes les informations sur les routes reçues de ses voisins et il les stocke dans sa table de topologie. EIGRP utilise ensuite l'algorithme DUAL pour sélectionner la meilleure route vers chaque sous-réseau, calcule la métrique à y associer et place le résultat dans sa table de routage.

C. La configuration d'EIGRP

1. Activer EIGRP de manière dynamique sur les différentes interfaces du routeur (Configurer EIGRP pour fonctionner dans un ASN. (router Eigrp <*asn*>)
2. network <*subnet*> *[masque inverse*]

Par exemple R1 connecté à un subnet 192.168.0.0/24 via son interface Fa0/0 (adresse IP 192.168.0.1 /24), également connecté à R2 via son interface S0/0 (adresse IP 172.16.0.1/30). R2 ayant son interface S0/0 configurée avec une adresse ip 172.16.0.2/30.

Sur le routeur R1 :

```
R1#configure terminal
R1(config)#router eigrp 10
R1(config-router)#network 192.168.0.0
R1(config-router)#network 172.16.0.0 0.0.0.3
R1(config-router)#exit
R1(config)#exit
R1#
```

Sur le routeur R2 :

```
R2#configure terminal
R2(config)#router eigrp 10
R2(config-router)#network 172.16.0.0
R2(config-router)#exit
R2(config)#exit
```

Donc EIGRP bien activé sur les deux routeurs pour l'ASN 10.

Sur R1 l'interface Fa0/0 et S0/0 bien activé via les commandes network.

Sur R2, EIGRP est activé pour l'interface S0/0.

3.2. Algorithmes de routage décentralisés

Avec un algorithme de routage décentralisé, le calcul du parcours le moins onéreux se fait de manière itérative et distribuée. Aucun nœud ne connaissant le coût de toutes les liaisons du réseau, le calcul se fait au niveau de chaque nœud en partant de leur connaissance du coût des liaisons auxquelles ils sont reliés. Puis, au moyen d'un calcul par itération et grâce à un échange d'informations avec les nœuds voisins, chacun parvient à évaluer progressivement le parcours le moins onéreux vers une destination ou un groupe donné de destinations. Un de ces algorithmes, connu sous le nom d'algorithme à vecteur de distance. En effet, un nœud n'est jamais en mesure de connaître le parcours complet allant de l'expéditeur au destinataire. En revanche, il sait vers quel voisin il doit transmettre les paquets reçus afin que ceux-ci empruntent le parcours le moins onéreux vers leur destination, de même qu'il en connaît le coût parcours le moins onéreux vers leur destination, de même qu'il en connaît le coût total.

Les différents algorithmes peuvent être répartis en algorithmes statiques ou dynamiques. Pour les premiers, les parcours changent très peu, et les modifications proviennent souvent d'une intervention humaine (modification manuelle du contenu de la table de routage d'un routeur). Pour les seconds, en revanche, les parcours s'adaptent automatiquement aux changements de la topologie ou aux variations du taux d'encombrement du réseau. Un algorithme dynamique peut fonctionner de manière soit périodique, soit immédiate. Ces algorithmes sont généralement plus prompts à prendre en compte les changements survenant au sein du réseau.

L'internet n'a normalement recours qu'à deux types d'algorithmes de routeur : un algorithme d'état de lien dynamique et global et un algorithme à vecteur de distance également dynamique, mais décentralisé.

3.2.1. Algorithme de routage à vecteur de distance

Alors que l'algorithme d'état de lien repose sur une connaissance de la topologie globale du réseau, l'algorithme à vecteur de distance est de nature itérative, asynchrone et distribuée. Il est de nature distribuée parce que les calculs se font au niveau de chaque nœud individuel, à partir des informations fournie par les nœuds voisins directs, et parce que les résultats sont partagés de la même manière. Il est de nature itérative, car ce processus se répète jusqu'à ce qu'il n'y ait plus d'informations à échanger entre nœuds voisins. De fait, cet algorithme s'interrompt de lui-même. Enfin, il est de nature asynchrone dans la mesure où il n'impose pas à tous les nœuds de travailler ensemble.

3.2.2. Protocole de routage par vecteur de distance

Ces algorithmes reposent sur un échange périodique des tables de routage entre routeurs, ainsi que l'accumulation des vecteurs de distance (vecteur désigne la direction et distance l'éloignement). Les modifications de topologie du réseau sont prévenues par des mises à jour régulières de ces données entre les routeurs.

Ainsi, chaque routeur reçoit une copie de la table de routage de son voisin immédiat. Le routeur B dans la figure reçoit l'information du routeur A. le routeur B augmente le vecteur de distance associé à l'aide d'une métrique adapté, puis envoie la table résultante à son autre voisin, le routeur C. ce même procédé est répété, de proche en proche, entre tous les routeurs immédiatement voisin.

Ainsi l'accumulation des distances réseau permet à l'algorithme de maintenir une base de données contenant les informations sur la topologie de l'inter-réseau. Cela dit, les algorithmes utilisant les vecteurs de distance ne permettent pas à un routeur de connaître la topologie exacte d'un réseau. [6]

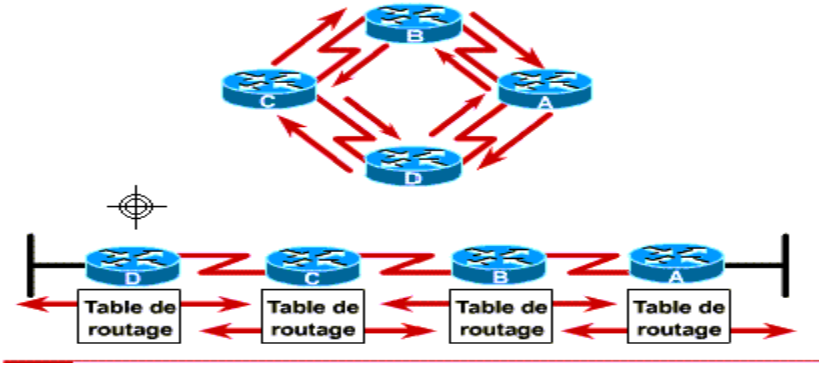


Fig.8 : protocole de routage par vecteur de distance

3.2.3. Processus du routage à vecteur de distance

Pendant que le processus de découverte du réseau par les vecteurs de distance suit son cours, les routeurs découvrent les meilleurs chemins d'atteindre les réseaux qui ne leur sont pas directement connectés pour cela, ils se basent sur les métriques accumulées de chaque voisin.

Le routeur A prend connaissance des autres réseaux par les informations reçues du routeur B. dans la table de routage, chacune des entrées relatives à ces autres réseaux possède un vecteur de distance. Celui-ci est obtenu par accumulation et permet de connaître l'éloignement du réseau dans la direction donnée.

Il peut avoir plusieurs chemins pour une même destination réseau donnée. Lors de la mise à jour de la table de routage, l'objectif premier de l'algorithme du protocole de routage est la détermination de la meilleure route de routage par vecteur de distance utilisent des algorithmes différent. Typiquement, plus la métrique associée est fiable et meilleure est la route.

Le calcul des métriques peut être basé sur un seul ou plusieurs paramètres caractérisant le chemin. Voici une liste des métriques les plus couramment employées par les routeurs :

- **Compte des sauts (hop count)**.il s'agit du nombre de routeurs qu'un paquet devra traverser.

Nombre de battement (ticks).il s'agit du délai associé à une liaison de donnée exprimé en nombre de battement d'horloge de type IBM (approximativement 55 millisecondes par battement).

- **Coût (cost**) : il s'agit d'une valeur arbitraire, généralement basée sur la bande passante, la dépense pécuniaire ou toute autre mesure pouvant être assignée par l'administrateur réseau.
- **Bande passante (bandwidth) :** Il s'agit de la capacité d'écoulement de donnée d'une liaison.
- **Délai (delau) :** il s'agit du temps nécessaire pour transmettre un paquet de la source à la destination.
- **Charge (load) :** Il s'agit de la somme d'activité supportée par une ressource réseau
- **Fiabilité (reliability) :** il s'agit généralement d'une valeur relative au taux d'erreur de transmission rencontré sur chaque lien réseau.
- **MTU (maximum transmission unit**) : il s'agit de la longueur maximale de trame, exprimée en octets, acceptable sur tous les liens rencontrés le long du chemin.

Lorsque, dans un réseau utilisant un protocole à vecteur de distance, la topologie évolue, une mise à jour des tables de routage doit avoir lieu. Comme lors du processus de découverte du réseau, la mise à jour des tables s'effectue pas à pas. De poche en poche entre routeur.

Ainsi, les algorithmes à vecteur de distance appellent tous les routeurs à transmettre leur table de routage complète à chacun de leurs voisins adjacents ou directement connectés. Ces tables de routage contiennent les informations du coût total des routes (défini par leur métrique), ainsi que l'adresse logique du premier routeur rencontré sur le chemin menant au réseau associé.

Lorsqu'un routeur reçoit une mise à jour en prévenance d'un routeur de voisinage, il compare les données reçues avec sa propre table de routage. Pour cela, il établit une nouvelle métrique en ajoutant le coût nécessaire pour atteindre le routeur voisin aux nouveaux coûts des routes, transmis par ce même voisin ensuite, si le routeur en déduit une meilleure route (dont la métrique totale est inférieure), comparée à l'ancienne, pour atteindre un réseau à travers ce voisin, il met à jour sa propre table de routage.

3.2.4. RIP (protocole d'information de routage)

Malgré son ancienneté, le protocole RIP est toujours largement utilisé aujourd'hui. Ses origines remontent au temps de l'architecture XNS (Xerox Network Systems), à laquelle il doit d'ailleurs son nom. Sa généralisation est principalement due à l'assimilation de la version BSD (Berkeley Software Distribution) du système d'exploitation UNIX en 1982, compatible avec TCP/IP. La version originale de ce protocole est définie dans le RFC 1058 et la version à compatibilité ascendante dans le RFC 2453.

RIP est un protocole à vecteur de distance. Sa version originale a recours au nombre de bonds comme mesure du coût d'un chemin donné, attribuant à chaque liaison la même valeur unitaire. Le coût d'un chemin ne pouvant dépasser une valeur de 15 bonds. On sait qu'en présence d'un protocole à vecteur de distance les routeurs s'échangent des informations de routage avec leurs voisins directs. Avec RIP, ces échanges d'informations d'actualisation ont lieu environ toutes les 30 secondes grâce au message de réponse RIP. Celui-ci, envoyé par un routeur ou par un serveur, contient une liste comprenant jusqu'à 25 réseaux de destination au sein du système autonome, ainsi que des indications sur la distance les séparant de l'expéditeur. Ces messages sont aussi connu sous le nom d'annonces RIP. Les principales caractéristiques du RIP :

- C'est un protocole de routage par vecteur de distance.
- La métrique utilisée pour la sélection des chemins est le compte des sauts.
- La valeur métrique maximale attribuée est égale à 15 sauts.
- Les mises à jour de routage dans leur totalité, sont diffusées toutes les 30 secondes par défaut.
- RIP peut réaliser un équilibrage de charge sur jusqu'à six chemins aux coûts identiques (quatre chemin par défaut).
- RIP-1 impose que, pour chaque numéro de réseau de classe principale dans la hiérarchie, un seul masque de sous-réseau soit utilisé. Il s'agit d'un masque de sous-réseau à taille fixe. Le RIP-1 standard ne permet pas les mises à jour déclenchées.
- RIP-2 permet l'utilisation de masques de sous-réseau à taille variable sur l'interréseau. De tels masques sont appelés VLSM (Variable-Length Subnet Mask). Contrairement à RIP-1, RIP-2 supporte les mises à jour déclenchées.

Le protocole RIP a évolué au fil des années pour passer d'un protocole de routage par classes, RIP Version 1 (RIP v1), à un protocole de routage sans classe, RIP Version 2 (RIP v2). La version RIP v2 présente les améliorations suivantes :

- Possibilité de transmettre des informations supplémentaires sur le routage de paquets.

- Mécanisme d'authentification visant à sécuriser la mise à jour de tables.

- Prise en charge des masques de sous-réseau de longueur variable (VLSM).

Le protocole RIP permet d'empêcher les boucles de routage infinies grâce à la définition d'un nombre maximum de sauts autorisé sur un chemin entre la source et une destination. Le nombre maximum de sauts sur un chemin est 15. Lorsqu'un routeur reçoit une mise à jour de routage contenant une nouvelle entrée ou une entrée modifiée, la valeur métrique augmente de 1 et représente un saut sur le chemin. Si la métrique dépasse alors 15, on considère que cela correspond à l'infini et que le réseau de destination est inaccessible.

A. Configuration du protocole RIP

La commande router RIP permet de sélectionner le protocole RIP comme protocole de routage. La commande network permet d'indiquer au routeur les interfaces sur lesquelles exécuter RIP. Le processus de routage associe les interfaces spécifiques aux adresses réseau, puis commence à envoyer et à recevoir les mises à jour RIP sur ces interfaces.

Le protocole RIP envoie des messages de mise à jour de routage à intervalles réguliers. Lorsqu'un routeur reçoit une mise à jour de routage avec modification d'une entrée, il met à jour sa table de routage en conséquence.

La valeur métrique reçue pour le chemin est incrémentée de 1 et l'interface source de la mise à jour apparaît comme saut suivant dans la table de routage. Les routeurs RIP conservent uniquement la meilleure route vers une destination mais ils peuvent également gérer plusieurs chemins de coût égal vers une destination.

La plupart des protocoles de routage utilisent une combinaison de mises à jour soit périodiques, soit déclenchées par des changements sur le réseau. RIP utilise des mises à jour périodiques, mais la mise en œuvre de RIP par Cisco envoie des mises à jour dès qu'un changement dans la topologie est détecté. Les changements dans la topologie déclenchent aussi des mises à jour immédiates sur les routeurs IGRP, quelque soit l'état des compteurs périodiques. Sans ces mises à jour, RIP et IGRP ne fonctionneraient pas de façon satisfaisante.

Après avoir mis à jour sa table de routage en accord avec la modification de la configuration, le routeur commence à transmettre des mises à jour de routage pour informer les autres routeurs du réseau. L'envoi de ces mises à jour, appelées mises à jour déclenchées, est indépendant de l'envoi de mises à jour régulières par les routeurs RIP.

B. Equilibrage de charge RIP

L'équilibrage de charge est un concept permettant à un routeur de bénéficier de plusieurs « meilleurs chemins » vers une destination donnée. Ces chemins peuvent être définis de manière statistique par un administrateur réseau ou calculés par un protocole de routage dynamique tel que RIP.

RIP est capable de gérer un équilibrage de charge sur plus de six chemins de coût égal avec quatre chemins par défaut. RIP réalise ce qu'on appelle un équilibrage de charge de recherche séquentielle. En d'autres termes, RIP envoie tour à tour les paquets sur les chemins parallèles.

La figure présente un exemple de routes RIP à quatre chemins de coût égal. Au démarrage, le routeur utilise un pointeur d'interface qui pointe sur l'interface connectée au routeur 1. Ensuite, le pointeur d'interface boucle sur les interfaces et les routes d'une façon déterministe selon le modèle 1-2-3-4-1-2-3-4-1, etc. Comme la métrique utilisée pour le protocole RIP est le nombre de sauts, aucune importance n'est accordée au débit des liaisons. Par conséquent, le chemin présentant un débit de 56 Kbits/s ne sera pas privilégié par rapport à celui de 155 Mbits/s.

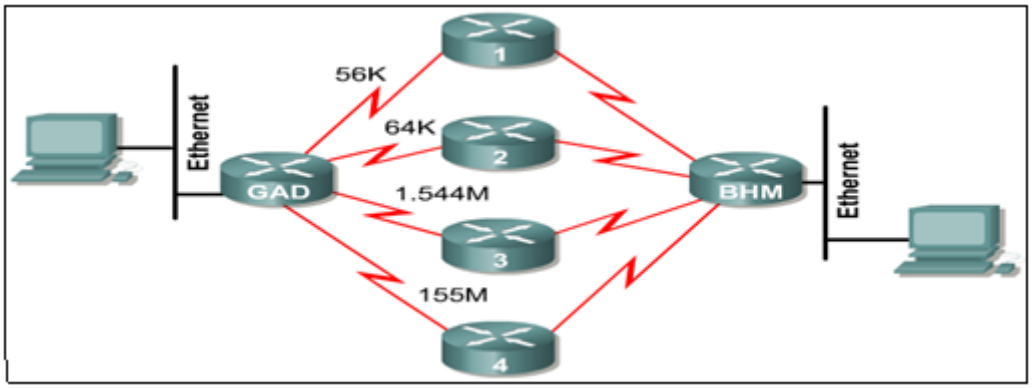


Fig.9. exemple de route Rip à coût égale

3.2.5. IGRP (interior gateway routing protocol)

Le protocole IGRP est un protocole IGP (Interior Gateway Protocol) à vecteur de distance. Les protocoles de routage à vecteur de distance comparent les routes de façon mathématique en mesurant les distances. Cette mesure est appelée vecteur de distance. Les routeurs utilisant des protocoles à vecteur de distance doivent envoyer, à intervalles réguliers, une partie ou l'intégralité de leur table de routage sous forme de message de mise

à jour à tous les routeurs voisins. Lors de la diffusion des informations de routage sur l'ensemble du réseau, les routeurs exécutent les fonctions suivantes :

- Identification de nouvelles destinations - Apprentissage des pannes

Le protocole IGRP est un protocole de routage à vecteur de distance mis au point par Cisco. Il envoie les mises à jour de routage toutes les 90 secondes et donne aux réseaux des informations sur un système autonome particulier. Les principales caractéristiques de la conception du protocole IGRP sont les suivantes :

- **Une extensibilité accrue :**l'extensibilité a été améliorée pour permettre le routage de réseau plus vaste que ceux permis par RIP . ainsi, là où RIP impose une limite de 15 saut, IGRP le supplante avantageusement avec une valeur maximale de 100 sauts par défaut, valeur qui peut être étendue par configuration jusqu'à 255 sauts.
- **Une métrique sophistiquée :** IGRP utilise une métrique composite qui fournit une grande flexibilité dans la sélection des routes. Par défaut, la métrique est composée des paramètres de délai inter-réseau et de bande passante. Mais des paramètres comme la fiabilité, la charge et le MTU peuvent tout aussi bien être inclus dans le calcul, si nécessaire.
- **Le support des chemins multiples :** IGRP peut maintenir jusqu'à six chemins différents entre une source et une destination du réseau ; et, contrairement à RIP, les chemins peuvent avoir des coûts différents. Les chemins multiples peuvent être utilisés pour améliorer la bande passante disponible ou pour sécuriser les routes.

A. Les métriques IGRP

IGRP utilise une métrique de routage composite. Cette complexité offre une plus grande finesse dans le choix d'un chemin pour une destination donnée que ne le permet la métrique de RIP, basée uniquement sur le nombre de sauts. Le chemin ayant la plus petite métrique est considéré comme la meilleure route. Par défaut, les métriques IGRP sont pondérées à l'aide des constante K1 à K5. Ces constantes permettent la conversion d'un vecteur métrique IGRP en une valeur scalaire.

La métrique IGRP inclut les paramètres suivant :

- **La bande passante :** la bande passante la plus faible rencontrée sur le chemin.
- **Le délai :** le cumul des délais induits par les interfaces le long du chemin.
- **La fiabilité :** la fiabilité de la connexion entre source et la destination. Déterminée à partir de données échangées sur sa durée de vie.

- **La charge :** basée sur la charge transmissible, en bits par seconde. Sur une liaison entre source et destination.
- **MTU :** l'unité de transfert maximale utilisable sur le chemin (c'est-à-dire la plus grande trame acceptable)

Le protocole IGRP utilise une métrique composée. Celle-ci est basée sur la bande passante, le délai, la charge et la fiabilité. Seuls la bande passante et le délai sont pris en compte par défaut. Les autres paramètres ne sont pris en considération que s'ils sont activés via la configuration. Le délai et la bande passante ne sont pas des valeurs mesurées mais des valeurs définies au moyen des commandes d'interface de délai et de bande passante. La fiabilité et la charge sont sans unité et peuvent prendre une valeur comprise entre 0 et 255. La bande passante peur prendre des valeurs reflétant des vitesses allant de 1200 bps à 10Gbps. Quant au délai, il peut prendre une valeur de 1 à 2* 10^23.

B. Routage IGRP

Pour initier un processus de routage IGRP, on utilise les commandes router igrp et network. Rappelons qu'IGRP requiert un numéro de système autonome. Ce numéro doivent utiliser le même numéro, sous peine, dans le cas contraire, de ne pas s'échanger les informations de routage. La commande network permet d'identifier un réseau principal auquel le routeur est directement connecté. Le processus de routage associe les adresses des interfaces aux numéros des réseaux devant être informés, et commence le traitement des paquets sur ces interfaces. La syntaxe des commandes router igrp et network est la suivante :

Router (config-router)#router igrp numéro_système_autonome

Router (config-router)#network numéro-de-réseau

La figure 10 présente un exemple de configuration du protocole IGRP avec le système autonome 101.

```
RouterA(config)#router igrp 101
RouterA(config-router)#network 192.168.1.0
RouterA(config-router)#network 192.168.2.0

RouterB(config)#router igrp 101
RouterB(config-router)#network 192.168.2.0
RouterB(config-router)#network 192.168.3.0
```

Fig10. : Configuration du protocole IGRP

C. L'équilibrage de charge IGRP à coût inégaux

IGRP supporte les chemins multiples entre source et destination, en s'appuyant sur sa métrique composite. Ainsi, un seul flux de trafic peut être écoulé sur deux lignes à bandes passantes égales. Dans tel cas, les données employées tour à tours sur les deux lignes et, si l'une d'elles devient inaccessible, la totalité du trafic est automatiquement dirigée sur l'autre.

En outre, on peut utiliser plusieurs chemins, même si les métriques des routes sont différentes. Ainsi, si par exemple un chemin est trois fois meilleur qu'un autre, parce que son coût est trois plus faible, il sera utilisé trois fois plus souvent que l'autre.

L'équilibrage de charge à coût inégaux permet une distribution du trafic sur jusqu'à six chemins de coûts différent, fournissant un débit de bout en bout, ainsi qu'une fiabilité accrue.

4. BGP (Border Gateway Protocol)

BGP se définit plus précisément comme un protocole à vecteur de chemin, ceci parce que les routeurs BGP voisins, connu sous le nom de partenaire BGP, s'échange des informations détaillées sur les chemins à emprunter (incluant par exemple la liste des systèmes autonomes que traverse un chemin pour atteindre une destination donnée) plutôt que des indications sur la distance exprimée par leur coût. A l'instant d'un protocole à vecteur de distance classique, BGP est un protocole distribué, dans le sens où les routeurs BGP ne communiquent qu'avec leurs voisins directs.

Les informations générales concernant les chemins desservant des destinations éloignées se propagent d'un AS en AS, via l'échange d'informations de routage BGP entre des binômes constitués par des partenaires BGP voisins. Le routage BGP se fait en direction de réseaux de destination plutôt qu'en direction de routeurs ou de serveurs. Une fois qu'un datagramme atteint son réseau de destination, il est pris en charge par le protocole de routage interne et transmis jusqu'à son destinataire final.

5. Discussion :

Tout au long de ce chapitre, nous avons vu les algorithmes de routage par état de lien (LS, link state) et à vecteur de distance (DV, distance vector) et leur propriétés. Les algorithmes LS maintiennent une base de données complexe, contenant des informations sur la topologie du réseau. Contrairement aux algorithmes par vecteurs de distance qui ne possèdent que des informations sur les réseaux et les routeurs avoisinant.

CHAPITRE III

Développement et tests

Préambule

Ce chapitre a été devisé en deux parties, dans la première nous avons présenté l'architecture et les principales fonctionnalités de Netfilter et iproute2, la deuxième a été consacrée à la mise en application de ces outils pour la réalisation de notre solution d'équilibrage de charge.

Partie 1

1. Préambule

Les systèmes d'exploitations basé sur les versions 2.6 et supérieur du noyau Linux sont particulièrement riches dans le domaine de la gestion des réseaux et il devient possible de réaliser avec ces systèmes des passerelles et des firewalls aussi performants sinon plus, que certains matériels spécialisés.

Netfilter permet de faire beaucoup plus de choses en matière de filtrage de paquets et de translation d'adresses de manière plus performante que ses prédécesseurs, quant à Iproute2 il constitue une nouvelle approche de la gestion des routes inter réseaux.

2. Netfilter :

Netfilter est un ensemble de hooks (accroches) implémenté au sein du noyau Linux. Il est utilisé pour l'interception et la manipulation des paquets réseau lors des appels des routines de réception ou d'émission des paquets par les interfaces réseau.

À travers ces points d'accroche, Netfilter est capable de :

- effectuer des filtrages de paquets, principalement pour assurer des fonctions de Firewall.

- effectuer des opérations de NAT (Network Address Translation) Ces fonctions sont particulièrement utiles lorsque l'on veut faire communiquer tout ou une partie d'un réseau privé, monté avec des adresses IP privées avec l'Internet.

- effectuer des opérations de marquage des paquets, pour leur appliquer un traitement spécial. Ces fonctionnalités sont particulièrement intéressantes sur une passerelle de réseau d'entreprise ou d'un réseau domestique.

Le schéma de la figure suivante représente ces points d'accroches

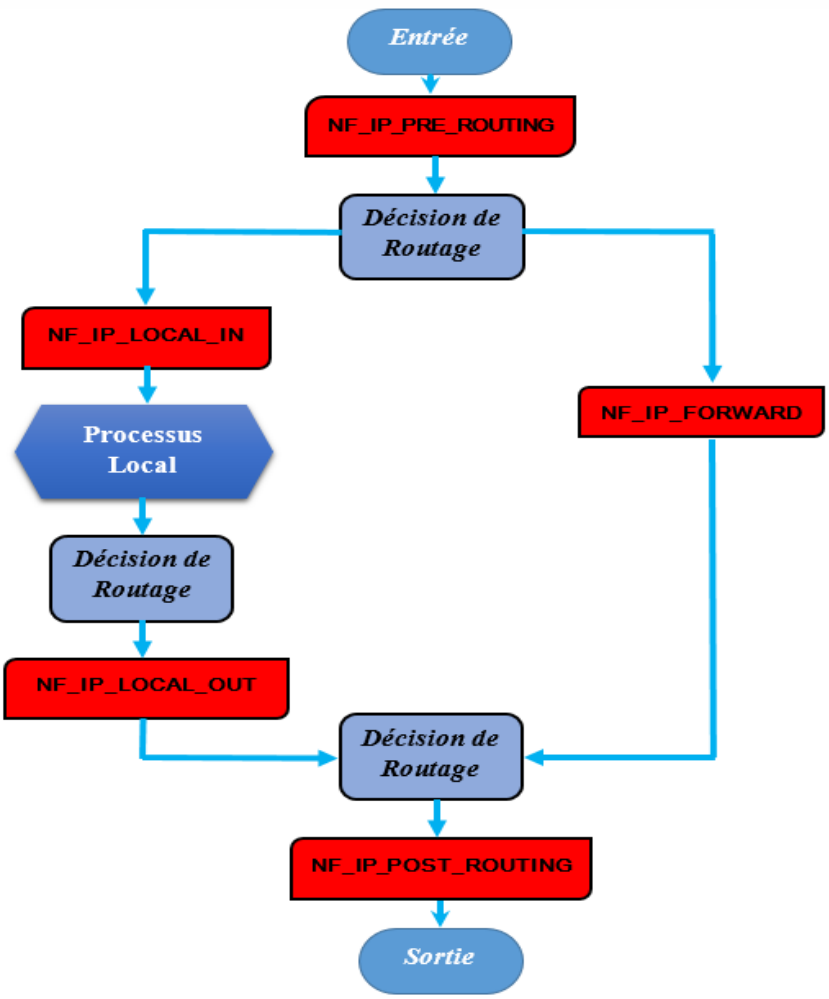


Fig.11. Points d'accroche de Netfilter

Le paquet arrive vers la machine après avoir passé de simples tests de validité (comme IP checksum…etc.), il passe alors dans premier point d'accroche de Netfilter NF_IP_PRE_ROUTING.

Ensuite il entre dans le code de routage qui décide si le paquet est destiné à une autre interface ou à un processus local. Le code de routage peut éventuellement détruire le paquet s'il n'est pas routable.

Si le paquet est destiné à la machine locale il traverse le point d'accroche NF_IP_LOCAL_IN avant d'être transmis au processus local, si par contre il est destiné à une autre interface réseau il traverse le point d'accroche NF_IP_FORWARD avant qu'il le transmette à son tour au point d'accroche NF_POST_ROUTING pour qu'il soit envoyé sur le support de transmission.

Les paquets générés par les processus local passent par le point d'accroche NF_IP_LOCAL_OUT qui est placé après la décision de routage, ce qui permet l'extraction des informations comme l'IP source ou de destination et de les modifier.

Pour pouvoir intervenir sur tous ces points d'accroches « hooks » Netfilter fait appel à iptables.

Iptables est l'interface en ligne de commande qui permet, d'écrire des chaînes de règles dans les tables de Netfilter. Netfilter possède trois tables qui correspondent aux trois principales fonctions cité précédemment.

2.1. Les tables et leurs chaînes :

Il existe trois tables qui vont servir à contenir des règles de filtrage et qui interviennent chacune sur certains points d'accroche « hooks » comme indiqué sur la figure suivante :

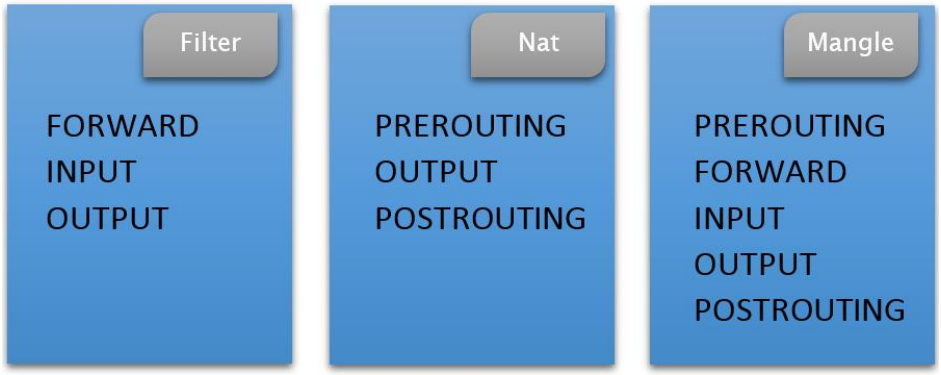


Fig.12. Tables utilisées par iptables

2.1.1. La table Filter

La table filter sert principalement à filtrer les paquets. On peut établir une correspondance avec des paquets et les filtrer comme on le désire. C'est l'endroit prévu pour intervenir sur les paquets et analyser leur contenu, c'est-à-dire les détruire ou les accepter suivant leur contenu. Bien entendu, il est possible de réaliser préalablement du filtrage ; malgré tout, cette table a été spécialement conçue pour ça [7].

Cette table intervient sur trois chaînes suivantes :

La chaîne FORWARD : les paquets qui traversent l'hôte, suivant les routes implantées, seront filtrés ici.

La chaîne INPUT : Cette chaîne décidera du sort des paquets entrant localement sur l'hôte.

La chaîne OUTPUT : Ici, ce ne sont que les paquets émis par l'hôte local qui seront filtrés.

2.1.2. La table Nat

Cette table est utilisée seulement pour effectuer de la traduction d'adresse réseau (NAT, Network Address Translation) sur différents paquets. Autrement dit, elle ne devrait servir qu'à traduire le champ de l'adresse source d'un paquet ou celui de l'adresse destination. Précisons à nouveau que seul le premier paquet d'un flux rencontrera cette chaîne.

Cette table intervient au niveau des chaines suivantes :

La chaîne PREROUTING : Permet de faire de la translation d'adresse de destination.

La chaîne POSTROUTING : Elle permet de faire de la translation d'adresse de la source.

La chaîne OUTPUT : Celle-ci permet de modifier l'adresse source ou de destination des paquets générés localement (par la machine elle-même).

2.1.3. La table Mangle :

Cette table permet le marquage des paquets entrants et générés localement .Le marquage de paquets va permettre un traitement spécifique des paquets marqués dans les tables de routage avec IPROUTE 2. Cette table peut intervenir au niveau de n'importe quelle chaine, néanmoins il est déconseiller de l'utiliser pour réaliser d'autres taches dédier aux autres tables.

2.1.4. Les chaînes

Les chaînes sont des ensembles de règles que nous allons écrire dans chaque table. Ces chaînes vont permettre d'identifier des paquets qui correspondent à certains critères.

2.2. Les cibles

Les cibles sont des sortes d'aiguillage qui dirigeront les paquets satisfaisant aux critères. Iptables possède plusieurs celles utilisées dans notre travaille sont :

2.2.1. Cibles de la table Nat

La cible SNAT : elle est employée pour changer l'adresse de source des paquets.

La cible MASQUERADE : elle s'utilise exactement de la même façon que la cible SNAT, mais la cible MASQUERADE demande un peu plus de ressources pour s'exécuter. L'explication vient du fait que chaque fois qu'un paquet atteint la cible MASQUERADE, il vérifie automatiquement l'adresse IP à utiliser, au lieu de se comporter comme la cible SNAT qui se réfère simplement à l'unique adresse IP configurée.

2.2.2. Cibles de la table Mangle

Cible MARK

La cible MARK sert à placer les valeurs de marquage Netfilter qui sont associées à des paquets spécifiques. Cette cible n'est valide que dans la table mangle, et ne fonctionne pas en dehors de celle-ci. Les valeurs MARK peuvent être utilisées conjointement avec les possibilités de routage avancé de Linux pour envoyer différents paquets à travers différentes routes et indiquer d'utiliser différentes disciplines de files d'attente, etc.

Cible CONNMARK

La cible CONNMARK sert à placer une marque sur une connexion, comme le fait la cible MARK. Elle peut alors être utilisée avec la correspondance connmark pour sélectionner la connexion dans le futur

3. Iproute2

Dans les systèmes d'exploitation existants, au fur et à mesure que de nouveaux concepts réseau apparaissaient, les développeurs sont parvenus à les greffer sur les structures existantes [8]. Ce travail constant d'empilage de couches a conduit à des codes réseau aux comportements étranges. Dans le passé, Linux émulait le mode de fonctionnement de SunOS, ce qui n'était pas l'idéal.

Les systèmes se basant sur un noyau linux 2.2 et plus bénéficient d'un sous-système réseau complètement réécrit. Ce nouveau codage de la partie réseau apporte à Linux des performances et des fonctionnalités qui n'ont pratiquement pas d'équivalent parmi les autres systèmes d'exploitation.

Iproute2 rassemble l'ensemble des outils nécessaire à la gestion et la configuration du routage dans les systèmes d'exploitation linux, il permet de formuler clairement des fonctionnalités impossibles à implémenter dans le sous-système réseau précédent.

Iproute2 donne accès à diverses commande celle utilisé dans se travaille sont les suivantes :

ip rule : cette commande permet la gestions des règles de routage, autrement dit elle permet l'ajout la suppression et la modification des règles régissant la politique de routage.

ip route : cette commande nous permet d'intervenir sur les tables de routage.

Partie 2

1. Présentation

Cette partie décris la méthode utilisée pour gérer de manière sélective le load-balancing entre plusieurs accès à internet.

Dans notre cas on dispose de deux accès internet distincts, et l'on souhaite pouvoir rediriger (re-router) arbitrairement certaines connexions réseau vers l'un ou l'autre de ces accès.

Ces accès internet peuvent prendre de nombreuses formes : ADSL, fibre optique, Wifi,PPP, VPN, ...

De manière générale on se trouve face à deux cas distincts :

- Une seule passerelle internet est accessible via une interface réseau.
- Plusieurs passerelles internet sont accessibles via une même interface réseau.

Dans notre travail on mettra on application le cas où une seule passerelle est accessible via une interface.

2. Outils utilisés

Pour mettre en œuvre cette méthode on a utilisé les outils suivant :

- Tables de routage (iproute2) pour router les connexions vers l'une ou l'autre passerelle.
  - gestion des tables de routage (ip route)
  - gestion des règles de routage (ip rule)
- Marquage de paquets (iptables -j MARK) pour identifier quelle table de routage sera utilisée.
- Marquage de connexions (iptables -j CONNMARK) pour déterminer les stratégies de routage et gérer l'affinité d'une connexion avec un accès donné.
- Translation NAT (iptables -j MASQUERADE) pour assurer le transit correct des paquets re_routés.

3. Configuration réseau utilisée

Dans notre cas on dispose de la configuration réseau présenté dans la figure suivante :

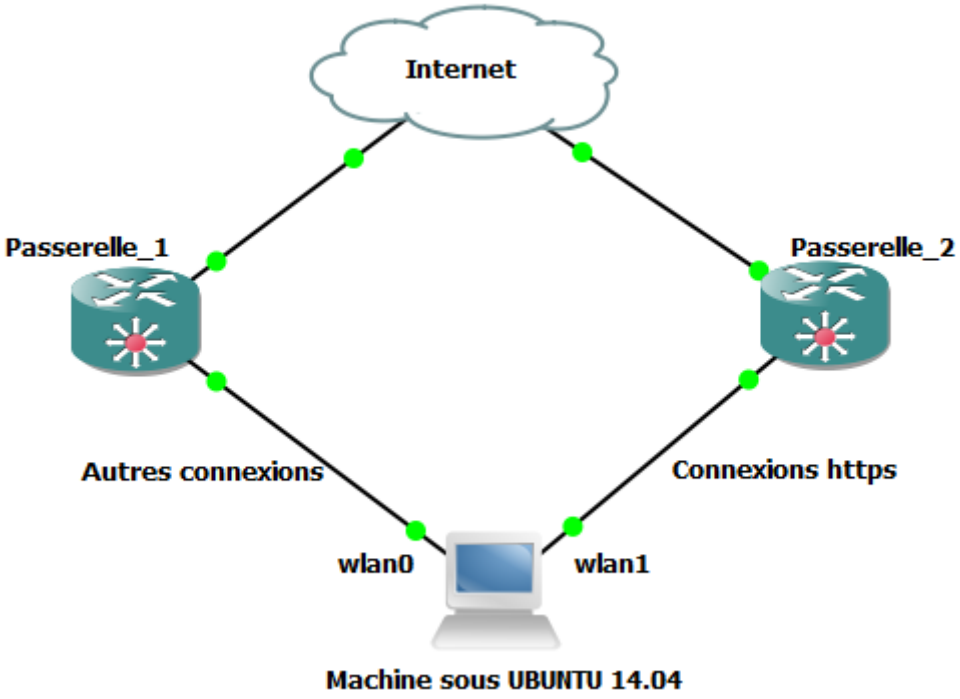


Fig.13.Configuration utilisée

Les deux accès internet son accecible via les deux interfaces wifi wlan0 et wlan1configurer de la manière suivante :

wlan0 : adresse IP 192.168.1.36/24, passerelle 192.168.1.1, correspondant à l'accès internet par défaut utilisé actuellement.

wlan1 : adresse IP 192.168.5.101/24, passerelle 192.168.5.1, correspondant à un accès internet non utilisé actuellement.

Il est possible avec la méthode qu'on présente ici d'utilisé plus que deux accès internet, il suffit juste d'adapter la configuration et les diverses règles en conséquence.

4. Stratégies de routage

Le but de cette méthode étant de router arbitrairement certaines connexions via l'accès internet de notre choix, les stratégies nous permettons de choisir le critère de routage étant diverses comme :

- sélection des paquets selon le protocole et le port utilisé

- sélection des paquets d'un utilisateur spécifique

- sélection des paquets générer par une application spécifique.

Notre choix c'est porter sur la première stratégie, où nous allons router le trafic https (à destination du port TCP 443) vers l'accès internet fourni par la passerelle 192.168.5.1 sur l'interface wlan1.

Il est aussi possible de mettre en place plusieurs stratégies actives en parallèle, on ajoutant certaines règles.

5. Détermination des diverses constantes

Pour qu'on puisse reconnaitre les paquets a redirigé, on doit leurs attribuer certaines valeurs spécifiques (marques) selon la stratégie cité précédemment.

Ces valeurs sont :

- Une valeur utilisée pour le marquage des paquets dans iptables (entre 1 et 2^{32}) qui servira entre autres à nous aiguiller vers la bonne passerelle (via une table de routage spécifique).
- Une valeur utilisée pour le marquage des connexions dans iptables (entre 1 et 2^{32}) qui servira à gérer l'affinité d'une connexion avec un accès internet donné.

Dans notre cas on a choisi d'utilisé la même valeur pour marquer les paquets à envoyer via l'interface wlan1 = 78.

6. Tables de routage

6.1 Création de la table de routage

Pour pouvoir routé nos paquets marqué nous devons crée une table de routage spécifique, pour cela on a ajouté une entrée dans le fichier **/etc/iproutes2/rt_tables** comme présenter ci-après :

```
rar@RaR-PC:~/Bureau$ cat /etc/iproute2/rt_tables
#
# reserved values
#
255     local
254     main
253     default
0       unspec
#
```

Fig.14. Table de routage par défaut

La figure 14 montre l'ensemble des tables de routage par défaut (local, main, default) utilisé par le système d'exploitation.

```
rar@RaR-PC:~/Bureau$ sudo gedit /etc/iproute2/rt_tables
```

Fig.15. Ouverture du fichier rt_tables avec l'éditeur de texte gedit

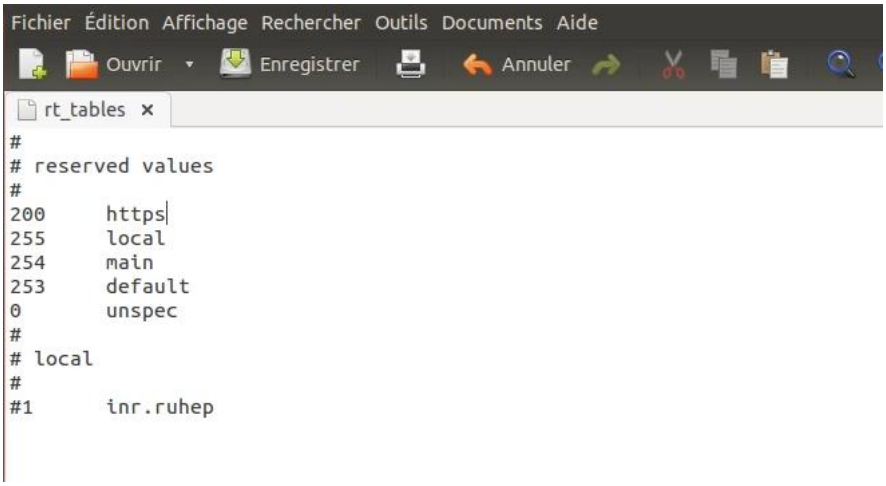


Fig.16. Ajouts de la table de routage https

6.2 Création des routes dans les tables de routage spécifiques

Par défaut, notre machine dispose d'une table de routage principale (la 254, nommée main) comme le montre la figure 14 qui comporte toutes les informations nécessaires pour utiliser les réseaux locaux ainsi que l'accès internet par défaut (wlan0 dans notre cas).

Pour chacun des accès internet on va recopier cette table en remplaçant la passerelle par défaut par la passerelle correspondante à l'accès en question. Le script suivant permet l'automatisation de cette tâche.

```
1  #!/bin/bash
2  #ce scrite permet l'automatisation de l'ajouter des routes a notre
3
4  TABLE="$1"
5  IFACE="$2"
6  GATEWAY="$3"
7    ip route flush table $TABLE
8    ip route show table main \
9      | grep -Ev ^default \
10     | while read ROUTE ;
11   do
12     ip route add table $TABLE $ROUTE
13   done
14   ip route add table $TABLE default via $GATEWAY dev $IFACE
```

Fig.17. Script gérant l'ajout des routes à la table

La figure 18 montre le contenue de la table https après l'exécution du script

```
rar@RaR-PC:~/Bureau$ ip route show table https
default via 192.168.5.1 dev wlan1
192.168.1.0/24 dev wlan0  proto kernel  scope link  src 192.168.1.36  metric 9
192.168.5.0/24 dev wlan1  proto kernel  scope link  src 192.168.5.101  metric 9
```

Fig.18. Contenue de la table https

7. Marquage des paquets et des connexions

Pour qu'on puisse sélectionner les paquets qui doivent être envoyés via l'interface wlan1 un marquage spécifique dans iptables doit leur être attribué.

```
rar@RaR-PC:~/Bureau$ sudo iptables -t mangle -A OUTPUT -p tcp --dport 443 -jMARK
 --set-mark 78
```

Cette commande permet de marquer tous les paquets sortant de notre machine via le port 443 et utilisant le protocole TCP par la marque choisie ici 78.

```
rar@RaR-PC:~/Bureau$ sudo iptables -t mangle -A PREROUTING -i wlan1 -m state --s
tate NEW -j CONNMARK --set-mark 78
```

Cette commande permet de marquer toutes les connexions entrantes via l'interface wlan1 par la marque choisie.

```
rar@RaR-PC:~/Bureau$ sudo iptables -t mangle -A OUTPUT -i wlan1 -j CONNMARK --re
store
```

Cette commande permet la restauration de la marque sur les connexions sortantes issues des connexions entrantes marquées précédemment.

8. Translation NAT

Toutes les connexions sortantes via l'interface wlan1 doivent être NATée, ce qui veut dire que l'adresse de l'interface wlan1 va être inscrite dans les champs adresse source des paquets destiné à être envoyer via cette interface. La commande suivante permet d'effectue cette opération

```
rar@RaR-PC:~/Bureau$ sudo iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE
```

9. Ajout des règles liant la table de routage au marquage

Les systèmes d'exploitation linux utilise par défaut les règles de routage par défaut suivantes

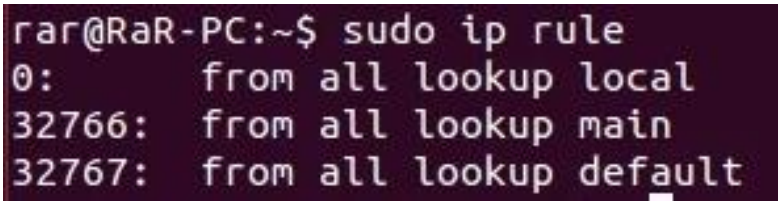


Fig.19. Règles de routage par défaut

Cette figure montre que par défaut tous les paquets destinés à quitter la machine vers le réseau extérieur sont routés selon les routes définit dans la table main

Pour crée une exception pour nos paquets on doit ajouter une règle liant notre marquage de paquets a la table https. La figure 20 montre la commande utilisée pour crée ce lien

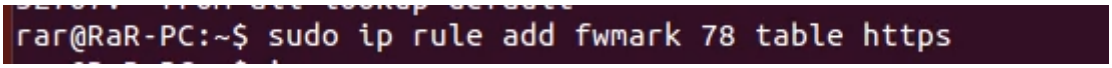


Fig.20. Création du lien entre le marquage et la table https

La figure 21 nous montre que tous les paquets portant la marque 78 vont être routés selon les routes définies dans la table https.

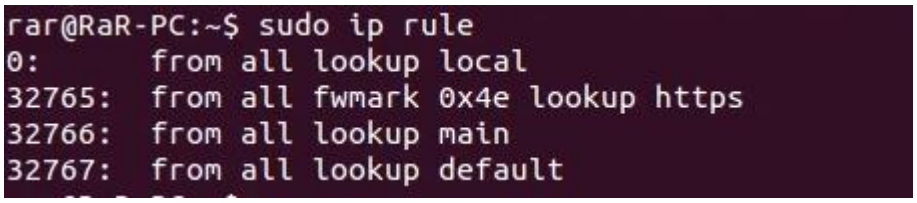


Fig.21. Règles de routage après ajout de l'exception

Les figures 22 et 23 montrent des captures d'écran du logiciel de capture de paquets Wireshark, et le résultat d'une commande **traceroute** qui permet de tracer les connexions, elles nous permettent de constater que tout le flux réseau passe exclusivement via l'interface wlan0 (passerelle 192.168.1.1).

Les deux figures 24 et 25 nous montrent qu'après application de la méthode d'équilibrage de charge, que les connexions réseau utilisant le protocole TCP sur le port 443 sont re-router vers l'interface wlan1 (passerelle 192.168.5.1)

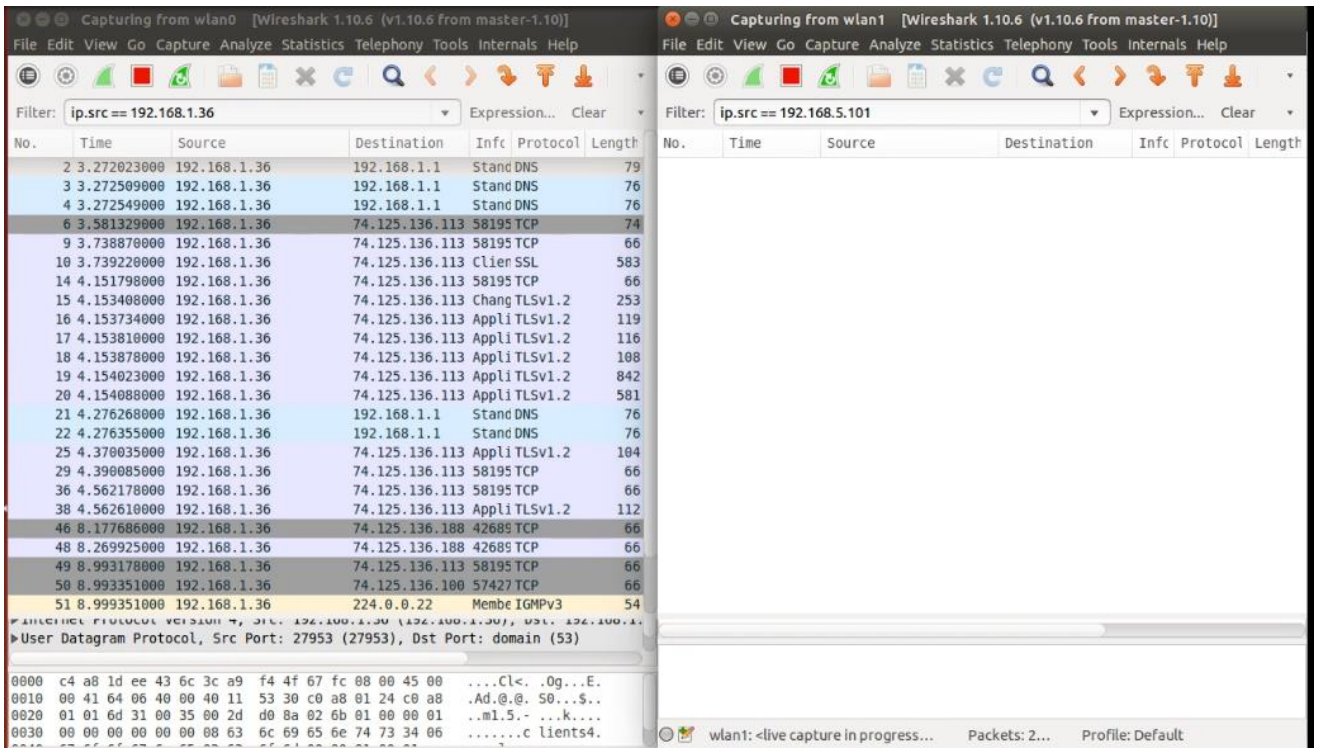


Fig.22. Capture du logiciel Wireshark avant application de la méthode

```
rar@RaR-PC:~/Bureau$ sudo traceroute www.google.dz -T -p 443
traceroute to www.google.dz (216.58.208.35), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.085 ms  2.176 ms  3.298 ms
 2  * * *
 3  10.200.123.66 (10.200.123.66)  776.534 ms  786.454 ms  806.247 ms
 4  10.192.0.233 (10.192.0.233)  816.506 ms  856.263 ms  876.210 ms
 5  10.192.0.251 (10.192.0.251)  907.127 ms  916.177 ms  956.119 ms
 6  172.17.2.177 (172.17.2.177)  967.034 ms  79.353 ms *
 7  72.14.214.14 (72.14.214.14)  139.269 ms  159.362 ms  169.217 ms
 8  209.85.252.36 (209.85.252.36)  189.269 ms  219.754 ms  259.258 ms
 9  216.239.43.42 (216.239.43.42)  260.101 ms  260.206 ms  269.283 ms
10  72.14.232.211 (72.14.232.211)  269.590 ms 216.239.48.73 (216.239.48.73)  272.164
11  66.249.95.22 (66.249.95.22)  131.505 ms 216.239.51.2 (216.239.51.2)  119.772 ms 2
12  216.239.51.235 (216.239.51.235)  161.419 ms  139.230 ms  109.734 ms
13  www.google.dz (216.58.208.35)  110.956 ms  110.024 ms  102.411 ms
rar@RaR-PC:~/Bureau$ sudo traceroute www.google.dz -T -p 80
traceroute to www.google.dz (216.58.210.195), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  4.187 ms  4.273 ms  4.387 ms
 2  * * *
 3  10.200.123.66 (10.200.123.66)  798.861 ms  808.711 ms  828.703 ms
 4  10.192.0.233 (10.192.0.233)  838.790 ms  898.855 ms  899.027 ms
 5  10.192.0.251 (10.192.0.251)  938.815 ms  948.975 ms  978.668 ms
 6  172.17.2.177 (172.17.2.177)  988.765 ms  69.449 ms  89.231 ms
 7  72.14.214.14 (72.14.214.14)  129.322 ms  169.203 ms  180.380 ms
 8  209.85.252.36 (209.85.252.36)  209.313 ms  219.050 ms  249.169 ms
 9  64.233.174.55 (64.233.174.55)  259.290 ms  299.866 ms  309.086 ms
10  mrs04s09-in-f195.1e100.net (216.58.210.195)  339.210 ms  349.037 ms  309.747 ms
rar@RaR-PC:~/Bureau$
```

Fig.23. Résultats de la commande traceroute

```
rar@RaR-PC:~/Bureau$ sudo traceroute www.google.dz -T -p 443
traceroute to www.google.dz (216.58.208.99), 30 hops max, 60 byte packets
 1  192.168.5.1 (192.168.5.1)  3.880 ms  4.543 ms  5.133 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  209.85.253.10 (209.85.253.10)  187.014 ms  206.299 ms  236.121 ms
10  72.14.232.76 (72.14.232.76)  246.621 ms  247.215 ms  247.795 ms
11  * * *
12  209.85.253.115 (209.85.253.115)  276.025 ms  276.616 ms  119.209 ms
13  * * *
14  216.239.51.223 (216.239.51.223)  128.878 ms  138.431 ms  119.540 ms
15  www.google.dz (216.58.208.99)  128.954 ms  110.701 ms  109.868 ms
rar@RaR-PC:~/Bureau$ sudo traceroute www.google.dz -T -p 80
traceroute to www.google.dz (216.58.208.99), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.721 ms  2.785 ms  3.380 ms
 2  * * *
 3  10.200.123.66 (10.200.123.66)  649.832 ms  649.969 ms  659.102 ms
 4  10.192.0.233 (10.192.0.233)  669.291 ms  689.132 ms  709.344 ms
 5  10.192.0.251 (10.192.0.251)  739.202 ms  769.654 ms  789.083 ms
 6  172.17.2.177 (172.17.2.177)  809.146 ms  67.950 ms  87.811 ms
 7  72.14.214.14 (72.14.214.14)  117.751 ms  167.669 ms  177.543 ms
 8  209.85.252.194 (209.85.252.194)  207.643 ms  217.620 ms  247.570 ms
 9  209.85.253.10 (209.85.253.10)  277.622 ms  307.564 ms  317.552 ms
10  72.14.232.76 (72.14.232.76)  368.446 ms  367.754 ms  301.058 ms
11  72.14.234.10 (72.14.234.10)  300.326 ms  270.941 ms  220.794 ms
12  209.85.253.115 (209.85.253.115)  230.380 ms  229.971 ms  220.088 ms
13  72.14.238.132 (72.14.238.132)  190.305 ms  161.053 ms  131.430 ms
```

Fig.24. Résultats de la commande traceroute après application de la méthode

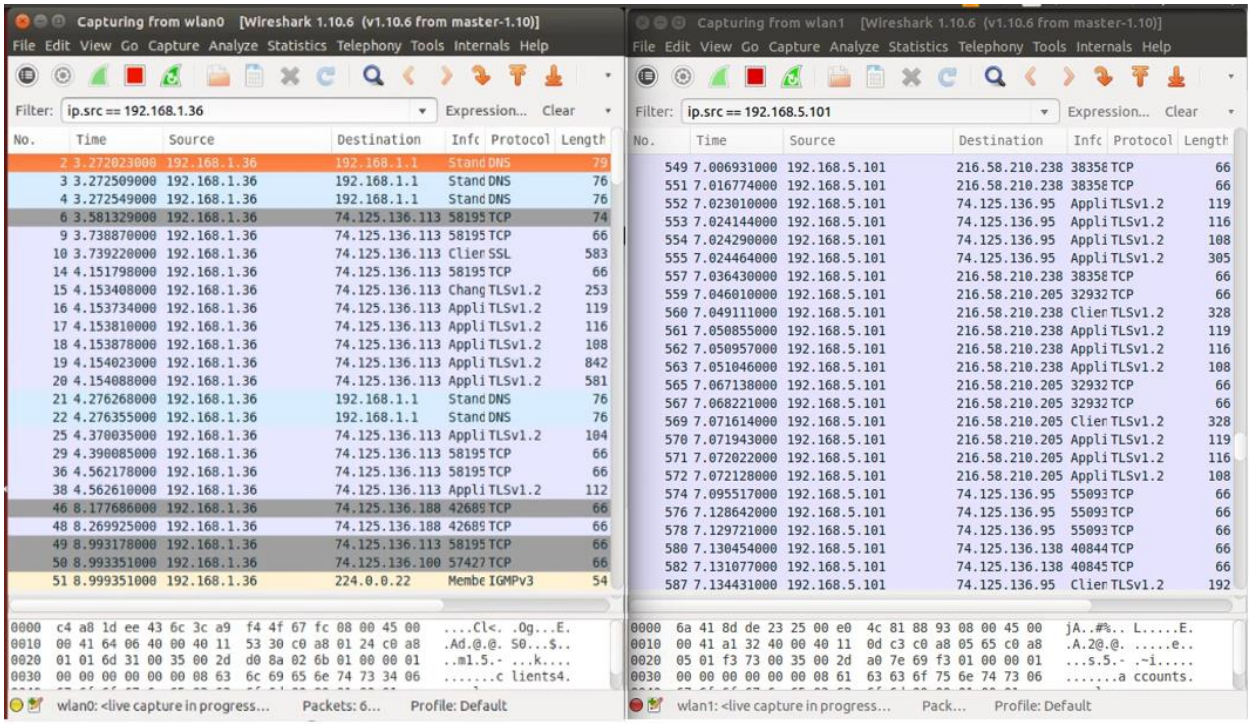


Fig.25. capture du logiciel Wireshark après application de la méthode

Discussion

Nous avons vue dans ce chapitre les outils qui nous on permit de mettre en œuvre notre solution d'équilibrage de charge et leurs mises en pratique ; la méthode utilisé présente l'avantage d'être adaptable a divers cas comme une utilisation en réseau où il est possible de filtrer provenant d'un réseau local allant vers internet, elle permet aussi une sélection précise des connexions a gérée.

Conclusion

Conclusion

Le développement de plus en plus d'applications qui nécessitent une haute disponibilité du support communication avec une grande fiabilité, a suscité notre intérêt de développer une solution d'équilibrage de charge sur des liens disponibles tels que « ADSL, 4G, 3G, Satellites et autres ».

Dans ce travail nous nous sommes focalisé sur l'étude et la réalisation d'une solution d'équilibrage de charge (load balancer) sous Linux. À cet effet, nous avons partagé la charge imposée à un lien principal sur les divers liens disponible. Nous avons pris comme critère de répartition, le protocole et le port de connexion utilisé. Cette méthode présente l'avantage de permettre une sélection précise des connexions à rediriger et de pouvoir être appliquée dans diverses configurations réseau.

En guise de perspectives et afin de permettre une meilleure flexibilité de distribution de la charge, nous proposons l'application d'algorithmes permettant l'automatisation de la solution et la détection de l'état des liens disponibles (Fail-Over).

Bibliographie

[1] Eric Lalitte et R.Guichad, 04/2013, Apprenez le fonctionnement des réseaux TCP/IP, Ed. IN'TECH INFO

[2] Claude Servin, 2003, RÉSEAUX ET TÉLÉCOMS, Ed. DUNOD

[3] Willy Tarreau et Wilfried Train, 2012, Le Load Balancing, Ed. FIRST Interactive

[4] James Kurose et Keith W.ross, 06/2007, analyse structurée des réseaux, Ed. Pearson.

[5] André VAUCAMPS, 10/2010, Cisco protocoles et concepts de routage, Ed. eni editions.

[6] Steve McQuerry, 12/2007, Interconnecting cisco network devices, Ed. cisco press.

[7] Philippe Latu, Iptables Tutorial 1.2.2, www.inetdoc.net

[8] Bert Hubert et al, Routage avancé et contrôle de trafic sous Linux, www.inetdoc.net