République Algérienne Démocratique Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Mathématiques et de l'Informatique

Département Informatique

Thème

Qualité de services dans IPtables

Pour L'obtention du diplôme de Master
Spécialité : Réseaux et Télécoms

Rédigé par : M^{elle}. ALICHE Roza

M^{elle}. HENNI Djamila

Dirigé par : Mr. DEHMANI Yousef

Année universitaire: 2015-2016

Dédicace

Nous tenons à dédier ce modeste travail de fin d'étude à nos chers parents, que nulle dédicace ne puisse exprimer nos sincères sentiments, pour leur patience illimitée, leur encouragement contenu, leur aide, en témoignage de nos profonds amour et respect pour leurs grands sacrifies. Que dieu les protège

Nous dédions aussi ce travail à nos frères et sœurs, nos familles, nos amis, tous nos professeurs qui nous ont enseigné et à tous ceux qui nous sont chers.

Remerciement

Nous remercions tout premièrement dieu le tout puissant pour la volonté, la santé et la patience, qu'il nous a donné durant toutes ces longues années, notre grand salut sur le premier éducateur notre prophète Mohamed (satisfaction et salut de dieu soit sur lui)

Ainsi, nous tenons également à exprimer nos vifs remerciements à notre encadreur Monsieur DAHMANI Youcef pour avoir d'abord proposé ce thème, pour suivi continuel tout le long de la réalisation de ce travail, et qui n'a pas cessé de nous donner ses conseils.

Nous tenons à remercier vivement toutes personnes qui nous ont aidés à élaborer et réaliser ce projet de fin d'étude, ainsi à tous ceux qui nous ont aidés de prés ou de loin à accomplir ce travail.

Nos remerciements vont aussi à tous nos enseignants de département d'informatique à la faculté I b n K h a l d o u n, nos remerciements à tous les membres du jury qui ont acceptés de juger notre travail.

En fin, nous tenons à exprimer notre reconnaissance à tous nos amis et collègues pour le soutien moral et matériel…

Résumé

Les travaux menés dans ce projet de fin d'étude s'intéressent à l'étude de la qualité de service en IPtables particulièrement à la distribution de la bande passante disponible qui est un paramètre très important pour le bon fonctionnement de plusieurs applications réseaux .

Un rappel sur la bande passante, La transmission de données sur un support physique se fait par propagation d'un phénomène vibratoire. Ce phénomène est caractérisé par un intervalle de temps appelé période, Plus la largeur est importante, plus la quantité d'information circulant sur le support sera élevée. La largeur de la bande passante dépend du type de support utilisé.

La consommation d'internet sa se défère d'un utilisateur à un autre selon leur activités qui font augmenter la quantité de données transférées, ce qui n'est pas évident de diviser le débit équitablement entre un internaute qui est intéressé à la navigation de base, c'est-à-dire la consultation de site web et l'envoi ou la réception de courriels sans fichiers joints, qui ne génère pas un flot important de transfert de données. et un autre qui s'intéresse aux activités qui font augmenter rapidement la consommation internet voici quelques exemples:

- Le partage de fichiers à l'aide de logiciels de type poste à poste (Peer-To-Peer)
- La lecture en continu de fichiers visuels (streaming), comme les communications par l'entremise d'une webcam (Skype, MSN)
- Les vidéoconférences
- Les sites de vidéos en ligne comme Youtube
- Le téléchargement de films et de fichiers de musique
- L'accès aux stations de radio (audio streaming)
- Les jeux en réseau sur console ou sur ordinateur

Mots-clés : réseau informatique, collision, distribution, vitesse, sécurité, bande passante, qualité de service

Liste des figures

Sommaire

Chapitre 3 : Qualité de service dans IPtables

Chapitre 4 : Conception et réalisation

Chapitre1 : Le modèle TCP/IP

Introduction

Aujourd'hui, la plupart des entreprises, quelle que soient leur taille et leur fonctionnalité ont besoin d'un système d'information , qu'est un ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information. Pour cela, l'utilisation d'ordinateurs par la plupart des entreprises a facilité la création d'un réseau informatique interne et autre externe.

Sur un réseau, l'information est morcelée en petits paquets, avant d'être transmise d'un ordinateur à un autre. Un paquet est un segment de données comprenant un en-tête, les données utiles et des éléments de contrôle devant être transmis.

Chaque réseau est régi par un protocole une série de règles et de formats pour l'envoi et la réception de données, un même réseau peut utiliser plusieurs protocoles, un des protocoles les plus connus est le TCP/IP, ce dernier représente l'ensemble des règles de communication sur internet et se base sur la notion adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données.

Ce chapitre fournit principalement un aperçu sur la série des protocoles TCP/IP afin de constituer un prérequis adéquat et nécessaire pour la compréhension du fonctionnement des réseaux informatiques.

I. Modèle TCP/IP

1. Principe architectural

Le modèle TCP/IP (Transmission Control Protocol / Internet Protocol) est constitué d'un ensemble de protocoles permettant à plusieurs ordinateurs de communiquer entre eux, c'est-à-dire un ensemble des règles et des conventions relatives à respecter pour émettre et recevoir des données, ces règles régissent le contenu, le format, la synchronisation, la mirent en séquence et le contrôle des erreurs dans les messages échanger entre les périphériques du réseau. **[5]**

L'interconnexion réseau est un problème complexe, et pour résoudre ce problème, il faut le séparer et traiter par niveaux ou couches. La fonction de chaque couche est de fournir des services à son homologue de niveau supérieur en occultant ses traitements propres.

Le modèle TCP/IP comporte 4 couches, il a été créé afin de répondre à un problème pratique, il est très proche du modèle OSI qui comporte 7 couches, ce dernier qui a été mis au

point par l'ISO (International Standard Organisation) correspond à une approche plus théorique, il aide donc à comprendre le fonctionnement de TCP/IP. La figure 5 compare les deux architectures. **[1]**

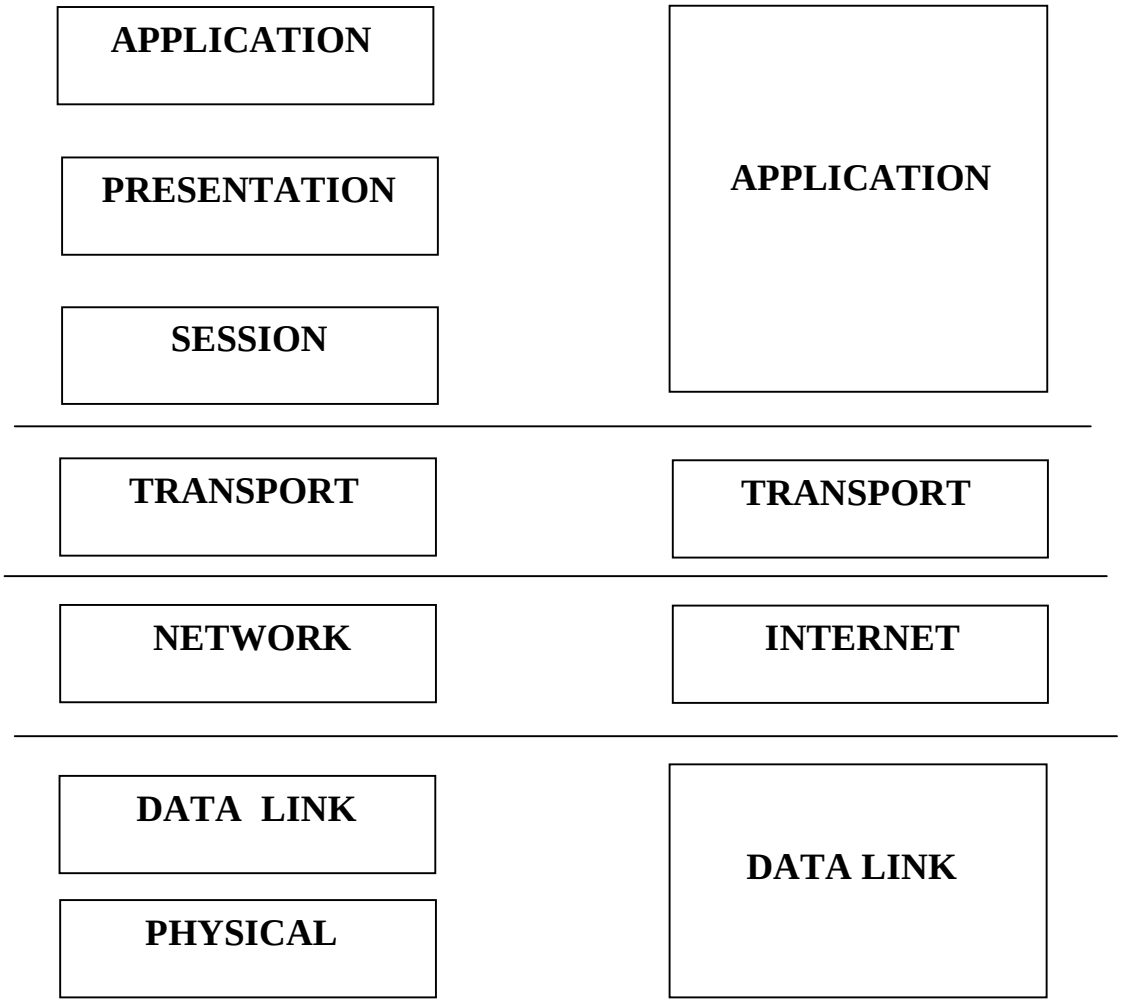

Figure 1 : Le modèle OSI et l'architecture TCP/IP

2. La description générale de la pile et les applications TCP/IP

Sur réseau le modèle le plus utilisable est TCP/IP qui est un suit des protocoles reliés entre eux, se suit des protocoles former la pile TCP/IP. **[1]**

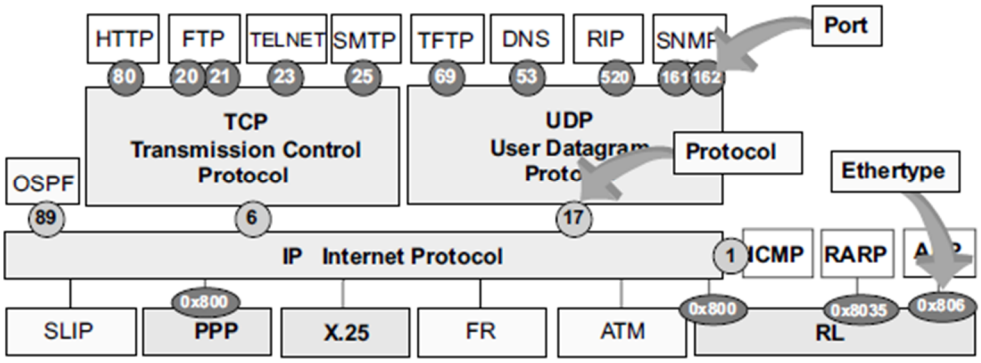

Figure 2 : Les protocoles et les applications de TCP/IP **[1]**

2.1. Couche application

C'est dans la couche application que se situent la plupart des programmes réseau.

Les applications fonctionnent généralement au-dessus de TCP ou d'UDP, et sont souvent associées à un port bien connu. **[1]**

FTP (File Transfer Protocol/ Port TCP 21) *:* est un système de manipulation de fichiers à distance (transfert, suppression, création...).

SMTP (Simple Mail Transfer Protocol/ Port TCP 25) : offre un service de courrier électronique.

HTTP (HyperText Transport Protocol/ Port TCP 80) : protocole de transfert hypertexte, un protocole de communication client-serveur développé pour le World Wide Web.

TELNET (TELetypewriter NETwork Protocol (ARPA) ou TERminaL NETwork Protocol/ Port TCP 23) : système de terminal virtuel permet l'ouverture de session avec des applications distantes.

DNS (Domain Name System/ Port UDP 53) : recherche de correspondance entre noms et adresses IP.

RIP (Routing Information Protocol/ Port UDP 520) : est le premier protocole de routage (vecteur distance) utilisé dans Internet.

SNMP (Simple Network Management Protocol/ Port UDP 161 et 162) : est devenu le standard des protocoles d'administration de réseau.

TFTP (Trivial FTP/ Port UDP 69) : est une version allégée du protocole FTP.

2.2. Couche transport

Les protocoles de la couche de transport peuvent résoudre des problèmes comme la fiabilité des échanges et assurer que les données arrivent dans l'ordre correct.

Dans la suite de protocoles TCP/IP, les protocoles de transport déterminent aussi à quelle application chaque paquet de données doit être délivré. **[2]**

2.2.1. Protocole TCP « Le mode connecté »

Le protocole TCP est un protocole de transport « fiable », orienté connexion, qui fournit un flux d'octets fiable assurant l'arrivée des données sans altérations et dans l'ordre, avec retransmission en cas de perte, et élimination des données dupliquées. Il gère aussi les données « urgentes » qui doivent être traitées dans le désordre. TCP essaie de délivrer toutes les données correctement et en séquence - c'est son but et son principal avantage sur UDP, même si ça peut être un désavantage pour des applications de transfert ou de routage de flux en temps réel, avec des taux de pertes élevées au niveau de la couche réseau.**[5]**

2.2.2. Le protocole UDP « le mode non connecté »

Un protocole simple, sans connexion, « non fiable » ce qui ne signifie pas qu'il est particulièrement peu fiable, mais qu'il ne vérifie pas que les paquets soient arrivés à destination, et ne garantissent pas leur arrivée dans l'ordre. **[5]**

2.2.3. Couche interréseau

La couche réseau ou couche IP (Internet protocol) gère la circulation des paquets à travers le réseau en assurant leur routage, ainsi la gestion de leurs fragmentations et leurs réassemblages à la réception. Elle comprend aussi les protocoles ICMP (Internet control message protocol), ARP (address resolution protocol) et RARP (reverse address resolution protocol).

2.2.4. Protocole IP

Comme UDP, le protocole IP est un protocole sans connexion qui ne contrôle pas les erreurs de transmission, il s'attache avant tout d'acheminement des paquets TCP à travers un ensemble de réseaux interconnectés.

Les principales fonctions de ce protocole : **[5]**

a. Adressage Internet :

Chaque ordinateur du réseau Internet dispose d'une unique adresse IP codée sur 32 bits. Autrement dit, chaque interface dispose d'une adresse IP particulière. En effet, un même routeur interconnectant 2 réseaux différents possède une adresse IP pour chaque interface de réseau. Une adresse IP est toujours représentée dans une notation décimale constituée de 4 nombres (1 par octet) compris entre 0 et 255 chacun et séparés d'un point. Une adresse IP est constituée d'une paire (identificateur de réseau, identificateur de machine) et appartient à une classe donnée (A, B, C, D ou E) selon la valeur de son premier octet, comme détaillé dans figure 7 :

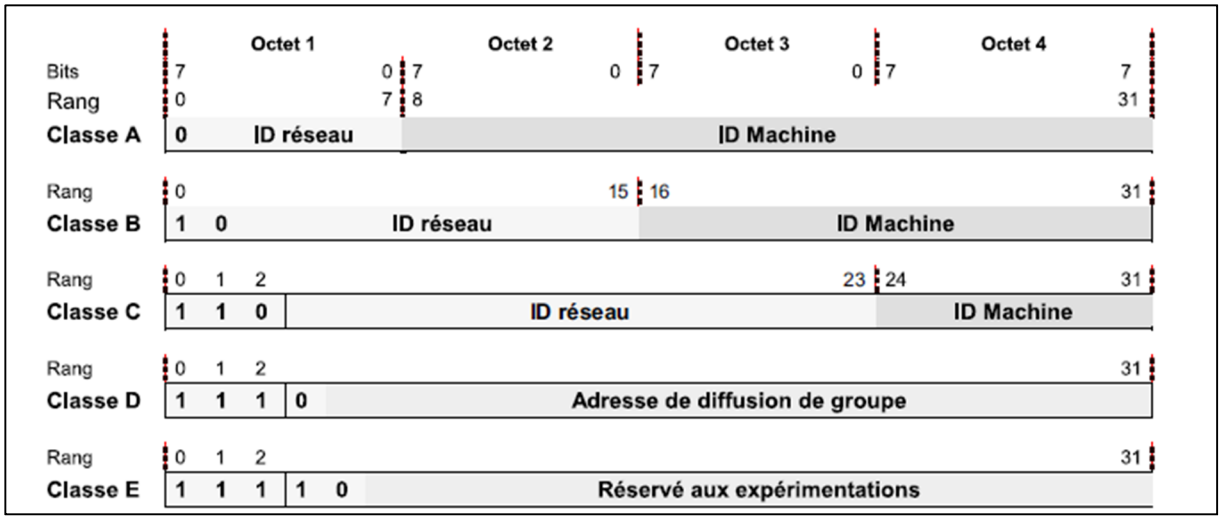

Figure 3 : Les classes des adresses IP **[5]**

b. Fragmentation :

La taille maximale d'un datagramme IP est de 65536 octets (équivalent de 64 MO). Toutefois cette valeur n'est jamais atteinte car les réseaux n'ont pas une capacité suffisante pour envoyer de si gros paquets. En plus, les réseaux ont leurs propres caractéristiques physiques dont la taille maximale des trames appelées MTU (Maximum Transfer Unit) exprimé en octets. La taille maximale d'un MTU est spécifique à chaque type de réseaux et elle est normalisé dans le cas des LAN (Ethernet 1500 octets, Token Ring 5000 octets). La couche interréseau doit assurer la fragmentation des paquets issus de la couche transport, si ces derniers ont une taille plus importante que le MTU du réseau. La fragmentation d'un paquet est contrôle par les champs : longueur totale, identification, drapeaux et le champ position du fragment. Ces quatre champs véhiculent suffisamment d'information pour permettre au récepteur de réassembler ces datagrammes. **[2]**

c. Routage :

Chaque datagramme IP contient l'adresse IP de l'émetteur et du destinataire, il ne contient pas l'identification d'une route par laquelle il passerait, c'est à la couche inter réseau de déterminer le chemin de la station destinataire à partir de l'adresse IP, c'est ce qu'on appelle le routage. Les appareils chargés de déterminer le chemin que les datagrammes vont emprunter pour arriver à la destination sont appelés routeurs.**[2]**

2.2.5. Protocole ICMP

ICMP utilisé par IP et par des protocoles de niveau supérieur, envoie et reçoit des informations d'état concernent les transmissions en cours. Il sert à la gestion du protocole IP, il permet par exemple, de collecter les erreurs qui surviennent lors de l'émission de message (réseau occupé, échéances temporelles…).

Les routeurs utilisent fréquemment ICMP pour contrôler le flux ou la vitesse des données qui leur arrive ou qu'ils envoient.

Il existe deux types de messages ICMP : des messages de commandes et des messages d'erreur. Ces messages ICMP rendent compte d'erreurs et renvoient systématiquement les 64 premiers bits du datagramme fautif. **[3]**

2.2.6. Les protocoles ARP et RARP

Le protocole IP, ainsi que ses adresses peuvent être utilisées sur des architectures matérielles différentes (réseau Ethernet, token-ring, etc.) ayant chacune sa propre adresse physique. Il y a lieu d'établir des correspondances bi-univoques entre adresse IP et adresses matérielles des ordinateurs d'un réseau. Ceci est l'objet des protocoles ARP et RARP. ARP fournit une correspondance dynamique entre une adresse IP connue et l'adresse matérielle correspondante, RARP faisant l'inverse. **[4]**

2.3. La couche accès au réseau

Le datagramme IP issu la couche inter réseau, ne peut être émit directement sur le câble réseau. Car il n'offre aucun moyen de reconnaitre son début ou sa fin. Pour effectuer la transmission, la couche accès au réseau utilise des trames à fin d'encapsuler le datagramme dans une trame et permettre ainsi la reconnaissance du début et de la fin.

SLIP (Serial Line Interface Protocol) : protocole d'encapsulation des paquets IP, ce Protocole n'assure que la délimitation des trames.

PPP (Point to Point Protocol) : protocole d'encapsulation des datagrammes IP, il assure la délimitation des trames, identifie le protocole transporté et la détection d'erreurs. **[1]**

3. Encapsulation des données

Lorsqu'une application envoie des données à l'aide de TCP/IP, les données traversent chaque couche de haut en bas pour aboutir au support physique ou elles sont alors émises sous forme de suites de bits. L'encapsulation illustrée dans la figure 16 consiste pour chaque couche à ajouter des informations aux données en commençant par des en-têtes, voire en ajoutant des informations de remorque.

L'unité de données que TCP envoie a IP est appelée un segment TCP. L'unité de données qu'UDP envoie a IP est appelée un datagramme UDP. L'unité de données qu'IP envoie à l'interface réseau est appelée un paquet IP. Ce flux de bits qui s'écoule le long du réseau est appelé une trame. **[4]**

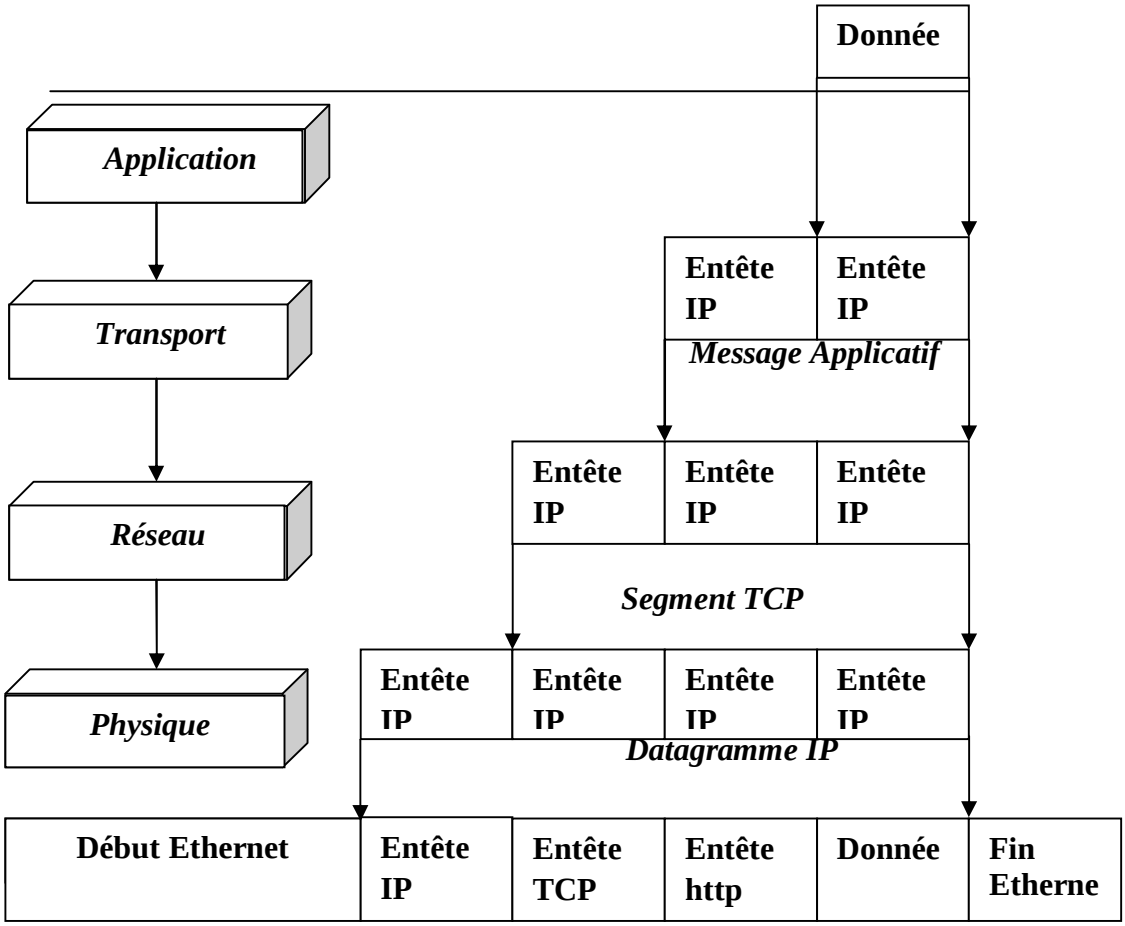

Figure 4 : Encapsulation des données par la pile des protocoles TCP/IP

4. Démultiplexage

Lorsqu'une trame est reçue sur la machine de destination, elle parcourt la pile de protocoles de bas en haut, et a chaque niveau, les en-têtes sont remplacés par la boite de protocole appropriée. Chaque boite de protocole se base sur certains identificateurs figurant dans son en-tête pour déterminer quelle boite de la couche supérieure recevant les données. Ce processus est appelé démultiplexage. La figure en montre le fonctionnement. **[4]**

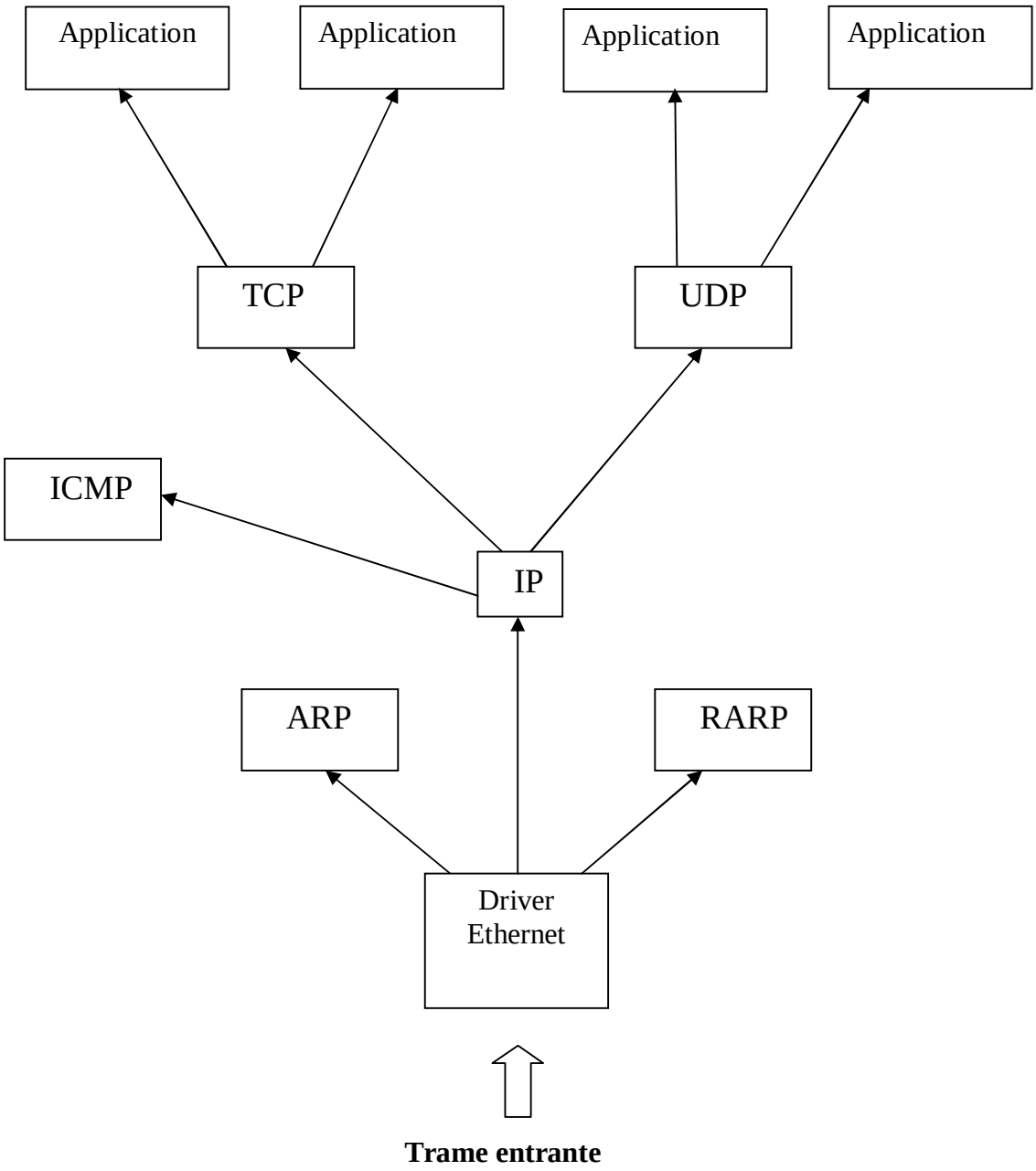

Figure 5 : Le démultiplexage d'une trame Ethernet reçue

Conclusion

Ce chapitre introduit les réseaux informatiques, un rappel sur les différents types de réseau selon leur taille, l'interconnexion entre les machines par des câbles et concentrateurs réseaux.

La disposition physique des ordinateurs d'un réseau est appelé topologie, on distingue entre la topologie physique qui concerne le câblage et la topologie logique qui concerne la façon dont les signaux circulent sur le câble.

Dans la deuxième partie de ce chapitre, nous avons vu des protocoles de communication des réseaux, à savoir la série des protocoles TCP/IP. Un administrateur doit savoir que les usurpations visent principalement TCP/IP. Il se doit donc de connaitre les protocoles de communication en général et les protocoles TCP/IP en particulier afin de mieux aborder les techniques de sécurité.

Dans le chapitre suivant, nous allons présenter principalement la sécurité réseau, en particulier IPtables comme une solution complète de pare-feu.

Chapitre 2 : Iptables

Introduction

De nos jours, l'une des méthodes fondamentales de sécurisation des réseaux informatiques est le filtrage des paquets. Aujourd'hui tout le monde sait ce que c'est qu'un firewall ainsi que son utilité sur un réseau, un serveur ou même un ordinateur personnel. En gros, c'est la partie du système qui s'occupe de gérer les accès à la machine et au réseau et donc ainsi de les protéger des intrusions venant d'un autre réseau ou d'une autre machine.
IPtables est un outil linux pour gérer le pare-feu qui est intégré au noyau Linux 2.4, l'architecture du noyau pour le système de pare-feu s'appelle 'Netfilter'.

Le but de ce chapitre est de comprendre le fonctionnement d'IPtables, nous allons présenter Netfilter pour connaitre notre système de base, ainsi que la description et les activités d'Iptables.

I. Netfilter

1. Présentation

Vous aurez remarqués que dans le mot Netfilter, il y a Net et il y a Filter donc à priori ça va parler d'internet et de filtrage qui rendait compte au firewall.

Netfilter est le module qui fournit sous linux depuis la version 2.4 les fonctions du pare-feu, de traduction d'adresse (NAT) et d'historisation du trafic réseau. Il fonctionne en mode noyau car il intercepte et manipule les paquets IP avant et après le routage. **[6]**

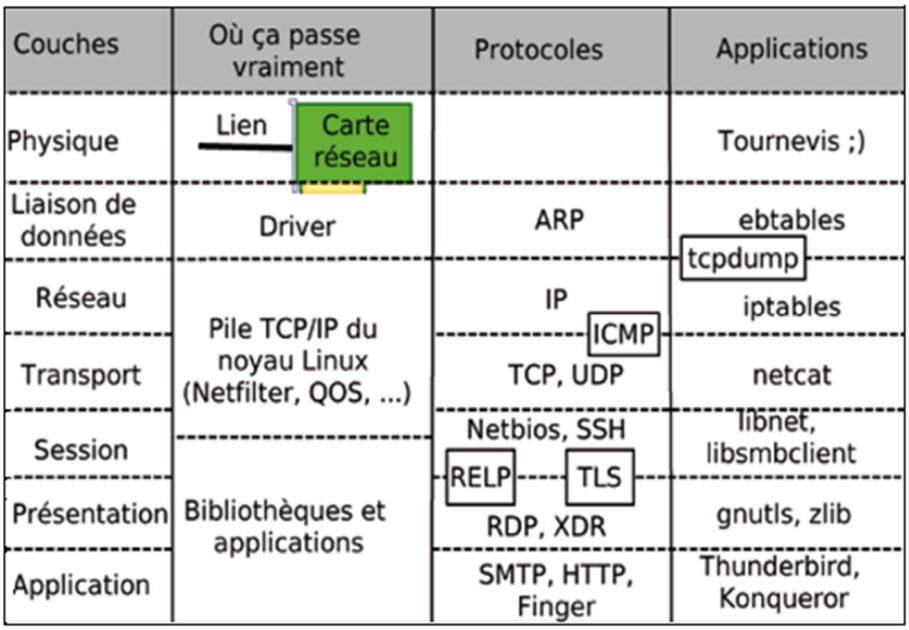

Figure 6 : Emplacement Netfilter **[7]**

Afin de présenter le Netfilter, on prend ce dernier par le bout réseau pour découvrir tous ses fonctionnalités. Pour savoir comment un paquet arrive depuis une autre machine vers vos applications, il se passe plusieurs étapes :

1. Une fois que la carte s'assure de l'intégrité de donnés, elle envoie ce signal à **la couche de liaison** qui s'assure grâce à l'adresse physique (MAC) de la carte que le signal lui est bien destiné.
2. Le signal remonte au niveau du réseau, et le couple IP source/destination est utilisé. A cette étape, il nous reste **routé** vers une autre machine.
3. Puis, ce signal est **transporté** vers un protocole de transport (généralement **tcp** ou **udp**).
4. Enfin, les données sont délivrées aux **applications** à l'écoute sur ce port et cette adresse IP.

1.1 Les hooks Netfilter

L'architecture de Netfilter comprend une interface uniforme. Ce qui réduit le cout impliqué pour mettre en œuvre des nouvelles fonctions, il est appelé *hook Netfilter,* ce qui signifie qu'il fournit un *hook* pour le code paquet-filter. Cette section décrit les composants de cette architecture et sa mise en œuvre dans le noyau Linux, En fait cette section fournit de brèves instructions pour faciliter votre écriture de vos propres modules du Netfilter.

Netfilter offre une série de cinq hooks placés sur la pile réseau du noyau à chaque fois que l'on est dans un cas suivant : **[8]**

- Un paquet arrive sur l'interface réseau (**NF_IP_PRE_ROUTING**)
- Le paquet passe l'étape de routage et arrive en locale (**NF_IP_LOCAL_IN**)
- Le paquet n'est pas destiné à une interface locale (**NF_IP_FORWARD**)
- Le paquet ressort par l'interface réseau (**NF_IP_POST_ROUTING**)
- Le paquet ressort de l'interface locale (**NF_IP_LOCAL_OUT**)

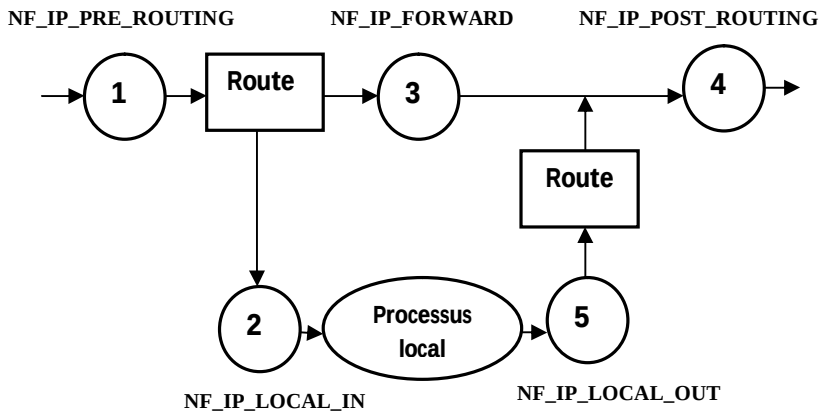

Figure 7: Les hooks netfilter

1.2 Les objectifs du Netfiler

– Monter un firewall filtrant basé sur les paquets mais aussi sur le statut des connexions engendrées par les paquets (le suivi de connexion)

– Utiliser NAT (Network Address Translation) et le masquerading afin de partager un accès internet à plusieurs machines

– Utiliser NAT pour faire du proxy transparent (évite d'avoir à paramètre le proxy sur les clients/navigateurs web)

– Mettre en place (notamment en permettant le marquage des paquets via la table mangle) la possibilité d'utiliser tc (trafic control) + iproute2 dans le but d'obtenir un routeur sophistiqué permettant le QoS (Quality Of Service, pour privilégié certains services, mettre en place des limites d'utilisation de bande passante sur un utilisateur, sur un groupe...)

– Manipuler des paquets. **[13]**

II. IPtables

1. Définition d'Iptables

Iptables est un utilitaire Linux (ligne de commande) intégré au noyau système depuis la version 2.4. Il est un front-end à Netfilter qui permet de faire de la sélection de paquet et de les envoyer à une *cible* déterminée. Cela peut être du filtrage ou du NAT.

Iptables est utilisé pour définir, maintenir et visualiser les règles de filtrage IP du noyau, En d'autres termes IPtables permet de manipuler des règles qui définissent des *chaînes* qui sont imbriquées dans des tables. **[8]**

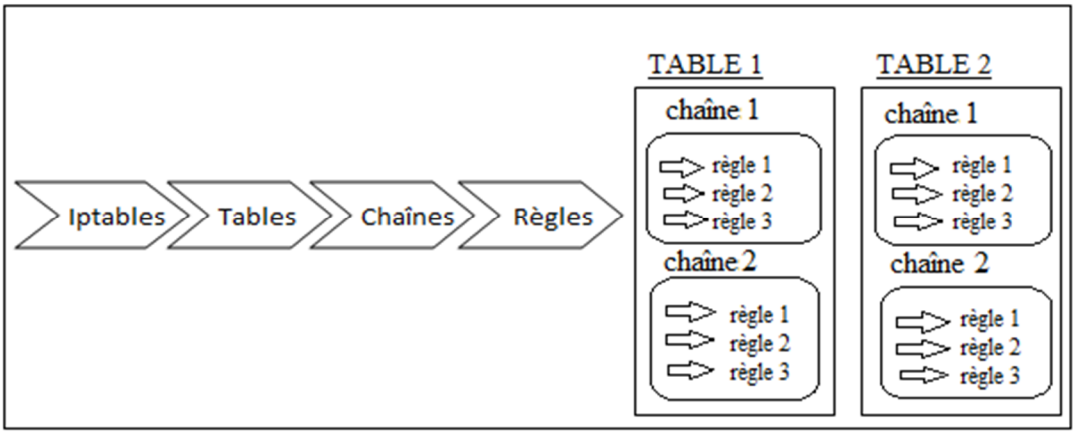

Figure 8 : Structure d'Iptables

2. Tables et chaînes

Il y a quatre tables au total. Chaque table contient plusieurs chaînes prédéfinies. Elles peuvent aussi être personnalisées.

- Filter : INPUT, PORWARD, OUTPUT
- Nat : PREROUTING, POSTROUTING, OUTPUT
- Mangle : PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING
- Raw : PREROUTING, OUTPUT

2.1 Fonctionnement générale

Tous les paquets inspectés par l'Iptables passent à travers une séquence de tables intégrées qui sont des sortes de files d'attente de traitement des paquets. Chacune de ces tables est dédiée à un type particulier d'activité contrôlée par des chaînes de transformation de paquets et/ou des chaînes de filtrage.

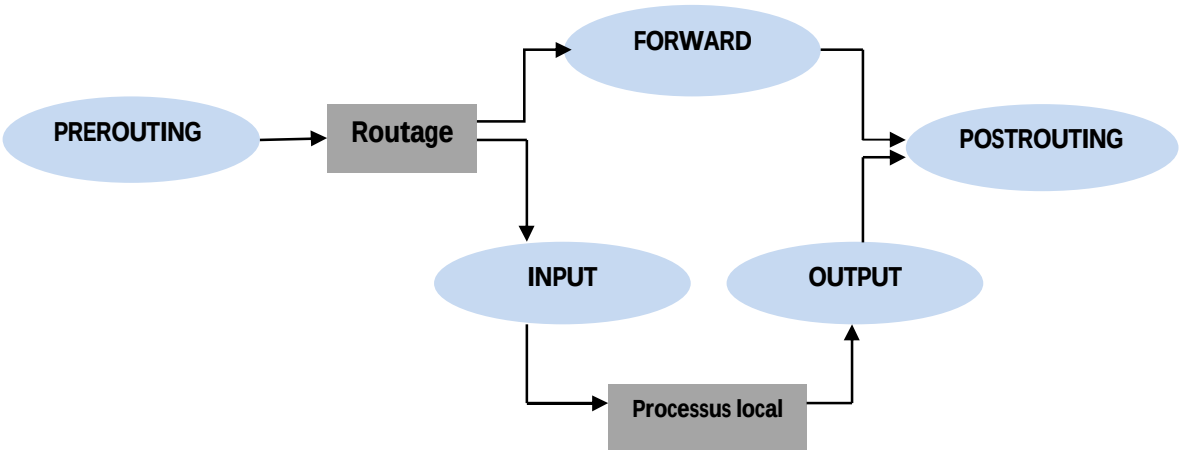

Figure 9 : Les chaînes iptables

La **figure 13** illustre le parcours typique d'un paquet IP dans un système (i.e. machine). A la réception d'un paquet IP le système commence par consulter le processus du routage pour déterminer la destination du paquet. Il est possible d'agir sur un paquet avant de considérer le processus du routage en associant un règle à la *chaîne* prédéfinie **PREROUTING**. Si le paquet traverse cette chaîne, le routage va être effectué. **[9]**

Deux cas sont possibles :

- Le paquet est destin à la machine elle-même. Dans ce cas le paquet va être envoyé à un processus local. Avant que le paquet soit livré à ce processus il doit traverser la *chaîne* **INPUT**. Autrement dit, on dispose de la possibilité de désigner une règle de filtrage qui concerne les paquets destinés à la machine elle-même.
- Le paquet est destiné à un autre réseau. Dans ce cas, si le drapeau du routage est activé sur la machine locale, alors le paquet va être propagé vers le réseau cible mains en traversant d'abords, les *chaînes* prédéfinies **FORWORD** puis **POSTROUTING.**

Un paquet généré par la machine locale commence par traverser la *chaîne* prédéfinie **OUTPUT** puis **POSTROUTING.**

A la traversé d'une chaîne, les règles de filtrage associées à cette chaîne vont être examinée dans l'ordre. La première règle dont la partie paquet *cible* correspond au paquet à traiter sera exécutée. Ainsi, comme pour les règles de routage, l'ordre des règles de filtrage est important.

En plus des chaînes prédéfinies citées ci-haut, on peut définir des chaînes utilisateurs. Ce sont des chaînes logiques qui permettent de structurer le processus du filtrage. **[9]**

2.2 Les tables

2.2.1 La table Mangle

Le rôle principal de cette table devrait être de modifier des paquets. En d'autres termes, vous pouvez utiliser en toute liberté les correspondances de la table mangle, qui permettent de changer les champs de TOS pour les types de service (QoS), et d'autres.

- TOS
- TTL
- MARK
- SECMARK
- CONNSECMARK

Elles ne doivent pas être utilisées en dehors de cette table. **[10]** (On va découvrir cette table en détail dans le chapitre suivant).

2.2.2 La table Nat

Cette table devrait être utilisée seulement pour effectuer de la traduction d'adresse réseau (NAT) 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 rencontra cette chaîne. Ensuit, les autres paquets subiront automatiquement le même sort que le premier. Voici les cibles actuelles capables d'accomplir ce genre de choses :

- DNAT
- SNAT
- MASQUERADE
- REDIRECT

La cible **DNAT** est généralement utile dans le cas où vous détenez une adresse IP publiques et que vous désirez rediriger les accès vers un pare-feu localisé sur autre hôte (par exemple, dans une zone démilitarisée ou DMZ). Concrètement, on change l'adresse de destination du paquet avant de le router à nouveau vers l'hôte désigné.

La cible **SNAT** est quant à elle employée pour changer l'adresse de source des paquets. La plupart des temps, vous dissimulerez votre réseau local ou votre DMZ, etc. Un très bon exemple se serait donné par un pare-feu pour l'adresse externe est connue, mais qui nécessite de substituer les adresses IP du réseau local avec celle du pare-feu. Avec cette cible, le pare-feu effectuera automatiquement sur les paquets du SNAT dans un sens et du **SNAT inverse** dans l'autre, rendant possible les connexions d'un réseau local sur l'internet.

La cible **MASQUERADE** s'utilise exactement de la même façon que la cible SNAT, mais la cible MASQUERADE demande un peut 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. Par conséquence, la cible MASQUERADE permet de faire fonctionner un système d'adressage IP dynamique sous DHCP, que votre FAI devrait vous procurer pour des connexions à l'internet de type PPP, PPPoE ou SLIP. **[10]**

2.2.3 La table Raw

La table Raw est principalement utilisée pour placer des marques sur les paquets qui ne doivent pas être vérifiés par le système de traçage de connexion. Ceci est effectué en utilisant la cible NOTRACK sur le paquet. Si une connexion rencontre une cible NOTRACK, conntrack ne tracera pas cette connexion. Ceci était impossible à résoudre sans l'ajout d'une nouvelle table, car aucune des autres tables n'est appelée jusqu'à ce que conntrack ait été lancé sur les paquets, et ait été ajouté aux tables conntrack, ou vérifié sur une connexion existence.

Cette table ne supporte que les chaînes PREROUTING et OUTPUT. Aucune autre chaîne n'est nécessaire, car c'est le seul endroit où vous pouvez opérer sur les paquets avant qu'ils soient vérifiés par le traçage de connexion. **[10]**

2.2.4 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 (avec la cible DROP) ou les accepter (avec ACCEPT) suivant leur contenu. Bien que entendu, il est possible de réaliser préalablement du filtrage ; malgré tout, cette table a été spécialement conçue pour ça. Presque toutes les cibles sont utilisables dans celle-ci. D'autres informations seront données sur la table filter, cependant vous savez maintenant que c'est l'emplacement idéal pour effectuer votre filtrage principal. **[10].**

3. Gestion des chaînes

3.1 Politique de filtrage

Le principe du filtrage de paquets est simple: le kernel analyse l'en-tête de chaque paquet qui traverse l'interface externe (modem, carte Ethernet ...).

Un filtre IP opère principalement au niveau de la couche 2 de la pile TCP/IP. IPtables cependant peut également travailler au niveau de la couche Mais par définition un filtre IP travaille sur la seconde couche. **[12]**

3.1.1 Les termes du filtrage IP

Voici une liste des termes les plus couramment utilisés dans le filtrage IP.

– Effacement/refus (Drop/Deny) – Quand un paquet est effacé ou refusé, il est tout simplement supprimé.

– Rejet (Reject) – De même façon basique, c'est la même chose que effacer/refuser, sauf qu'une notification est envoyée à l'hôte expéditeur du paquet rejeté.

– Etat (State) – Un état spécifique d'un paquet par rapport à l'ensemble d'un flux de paquets. Par exemple, si le paquet est le premier que voit le pare-feu ou dont il a connaissance, il est considéré comme nouveau (le paquet SYN dans une connexion TCP), ou s'il fait partie d'une connexion déjà établie dont la connaissance le pare-feu. Les états sont connus par le traçage de connexion, qui garde les traces de toutes les sessions.

– Match – Ce terme peut avoir deux sens différents quand il est employé avec Iptables. Le premier sens est une simple correspondance qui indique à la règle, ce que l'en-tête doit contenir. Le second indique si la règle entière est une correspondance. Si le paquet s'accorde à toute la règle, les instructions saut et cibles seront exécutées (ex. le paquet sera supprimé).

– Saut (Jump) – L'instruction jump est très proche d'une cible. Une instruction jump est écrite exactement de la même façon qu'une cible dans Iptables, sauf qu'au lieu d'écrire un nom de cible, vous écrivez un nom de chaîne.

– Traçage de connexion – Un pare-feu qui implémente le traçage de connexion est capable de suivre les connexions/flux.

– Acceptation (Accept) – Pour accepter un paquet et le laisser passer à travers les règles du pare-feu. C'est l'opposé des cibles effacement et refus, de même pour la cible rejet.

– Gestion des règles (Policy) – Il existe deux sortes de gestion des règles dans l'implémentation d'un pare-feu. En premier nous avons la gestion des chaînes qui indiquent le comportement par défaut du pare-feu. Le second type de gestion des règles est la gestion de sécurité établie par une stratégie d'ensemble d'une entreprise.

3.1.2 Le mécanisme du filtrage de paquet

Les sorts les plus usuels des paquets sont:

✓ ACCEPT consiste à accepter le paquet, il passe donc au firewall

✓ DENY refuser le paquet et faire comme si le paquet n'avait pas été reçu. un paquet refusé n'engendre donc pas de message d'erreur du type ICMP

✓ REJECT rejeter le paquet, le rejet est identique au refus mais un message d'erreur du type ICMP est transmis à l'émetteur du paquet.

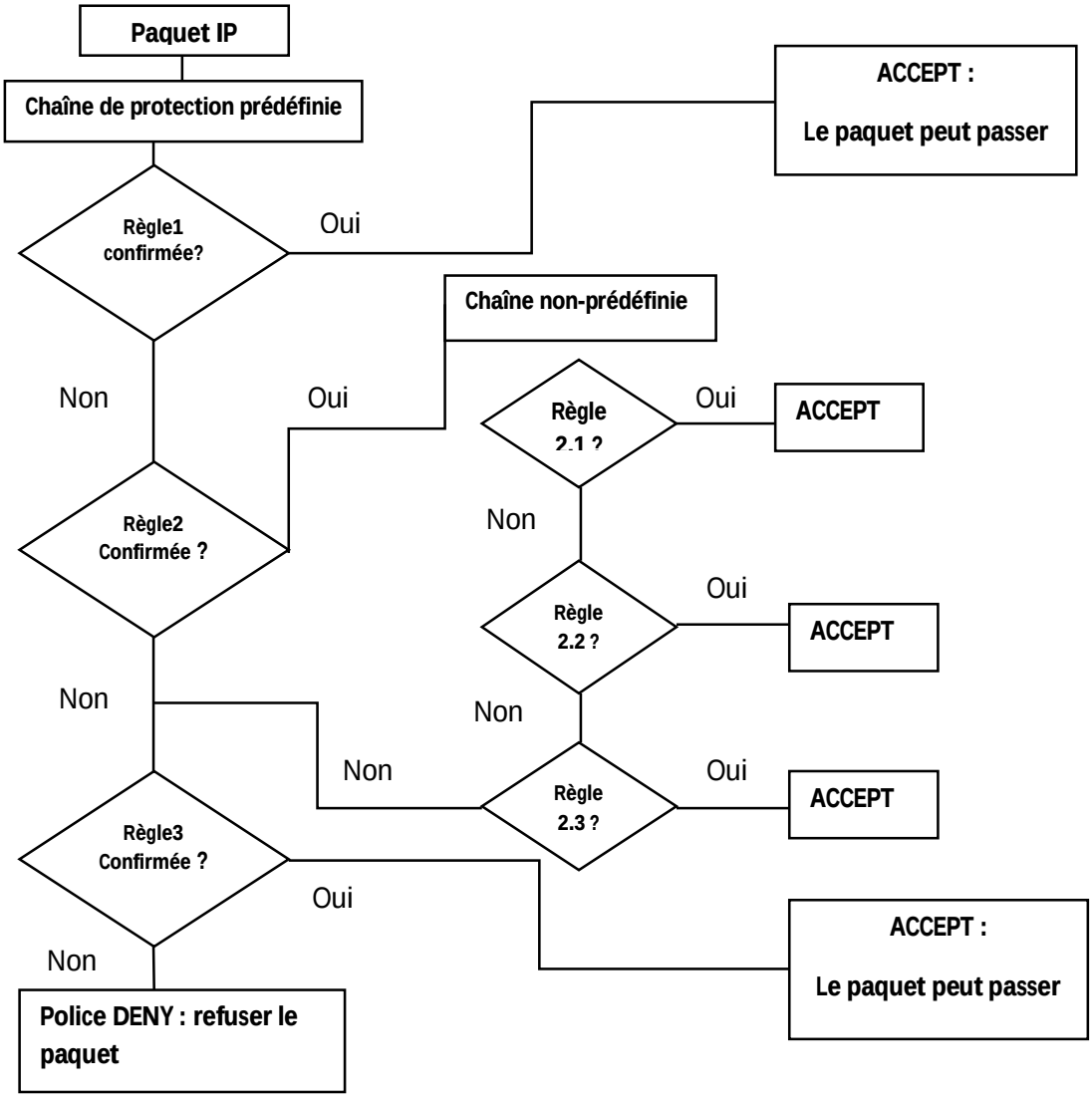

Figure 10: Le mécanisme du filtrage de paquet

Les paquets sont comparés successivement aux règles de la chaîne correspondante jusqu'à ce qu'une règle soit trouvée à laquelle le paquet correspond ou jusqu'à ce que la fin de la liste soit atteinte. Dans ce cas, la policy est consultée pour décider du sort du paquet. La policy est définie pour chaque chaine et elle dit ce que le noyau doit faire du paquet (accepter, rejeter ou refuser) si aucune règle de la chaine n'est appliquée. Dans les chaines figurent alors surtout les règles pour les paquets qui doivent être acceptés. Il est aussi possible d'accepter tous les paquets par défaut et filtrent les paquets non souhaités, de ne pas oublier une et de toujours reconfigurer le firewall quand les hackers ont trouvé une nouvelle méthode d'intrusion, la policy DENY est le seul choix acceptable pour un utilisateur final.

Si l'implémentation du filtre IP suit strictement la définition, il devrait être capable, en d'autres termes, de filtrer les paquets basées sur leurs en-têtes IP (adresses sources et destinations, TOS/DSCP/ECN, TTL, protocole, etc.). Toutes choses actuellement dans l'en-tête IP. Cependant, l'implémentation d'IPtables n'est pas strictement en accord avec la définition, il est aussi capable de filtrer les paquets basés sue d'autres en-têtes se trouvant dans le paquet (TCP, UDP, etc.), et l'adresse MAC.

Il y a une chose cependant sur laquelle IPtables est plutôt strict aujourd'hui. Il ne suit pas les flux ou les morceaux de données (via numéros d'interclassement, numéros de port, etc.) à peut prés comme la vraie pile TCP/IP. Ceci est appelé traçage de connexion, et grâce à ça nous pouvons effectuer de la translation (marquage/traduction) d'adresse source et destination (généralement appelé SNAT et DNAT), aussi bien que de la vérification d'état de paquet.

3.2 La traduction d'adresse réseau (NAT)

Un serveur **NAT** (pour *Network Address Traslation*) traduit les adresses IP sources et/ou destination des paquets en adresse différentes en jouent les numéros des ports. Le serveur NAT reçoit le paquet, change l'en-tête IP (il réécrit l'adresse source ou destination et recalcule la somme de contrôle du paquet) puis change l'en-tête de transport (il donne un numéro de port non utilisé et recalcule la somme de contrôle si celles-ci a un sens). **[11]**

La partie *netfilter* de linux permet de mettre en place facilement ces traductions d'adresses : en n'a pas en particulier à se préoccuper explicitement de la gestion des numéros de port ou du calcul des sommes de contrôle, effectués pour nous par le noyau linux. On se sert pour cela de la table **Nat** et de l'une des cinq cibles suivantes :

– La cible principale est **SNAT**, employée pour changer l'adresse source des paquets. Avec cette cible, le serveur **NAT** effectuera automatiquement sur les paquets de la traduction d'adresse source dans un sens et de la traduction d'adresse source inverse dans l'autre sens.

– La cible **DNAT** est employée de même pour changer l'adresse de destination.

– La cible **MASQUERADE** (*camouflage en français*) est l'analogue de la cible SNAT mais s'utilise lorsque les adresses sont données par DHCP. On spécifiera donc l'interface réseau au lieu d'une adresse IP.

– La dernière cible s'appelle **REDIRECT**.

3.2.1 Traduction d'adresse source

La traduction d'adresse réseau source est utilisée le plus fréquemment lorsque nous n'avons pas les moyens ou ne voyons pas l'intérêt d'avoir une adresse IP publique pour chacune des postes d'un réseau local. **[11]**

3.2.2 Traduction d'adresse de destination

La traduction d'adresse de destination est utile lorsque, dans la configuration qu'on veut qu'un serveur visible de l'extérieur par une adresse. **[11]**

3.2.3 Camouflage

Le camouflage d'adresse IP est un cas particulier de traduction d'adresse source à utiliser pour les adresses IP allouées automatiquement, comme par la liaison téléphonique avec les fournisseurs d'accès. Dans ce cas on ne spécifie pas explicitement l'adresse source publique mais l'interface : *Netfilter* utilisera l'adresse source de l'interface par laquelle le paquet sort. De plus, si le lien est rompu, les connexions (qui sont de toute façon perdues) sont oubliées, ce qui évite problèmes éventuels quand la connexion est rétablie avec une nouvelle adresse IP. **[11]**

Conclusion

Au cours de ce chapitre, nous avons découvris les nombreuses options et méthodes d'iptables qui permettent de faire du filtrage, nat et de la qualité de service…

Le principe de fonctionnement d'Iptables est donc d'écrire un programme qui récupère toutes les trames entrantes et sortantes, les analyse et les fait suivre ou non suivant le filtrage désiré. On intérêt à utiliser un modèle en couches qui facilite le travail.

Dans le prochain chapitre nous allons voir qu'est ce qu'une qualité de service et comment il est possible d'appréhender cette dernière au travers l'iptables.

Chapitre3 : Qualité de service dans IPtables

Introduction:

La qualité de service se retrouve dans tous les domaines, elle peut apporter des réponses à de nombreux problèmes que rencontrent les administrateurs systèmes ou les fournisseurs d'accès. Malgré les évolutions constantes en matière de bande passante, les trafics réseaux sont de plus en plus importants dans beaucoup de systèmes informatiques, dans une configuration simple de réseau ces flux sont émis dans l'ordre sans se soucier de savoir si l'application ou le service a besoin d'une bande passante minimum ou un délai de transmission très court. C'est pourquoi, ces dernières années, "le terme de qualité de service" s'est grandement développé au sein des réseaux.

Dans ce chapitre nous allons intéresser aux mécanismes permettant d'améliorer le niveau de QoS en Iptables mais aussi de sécurité qui permet de rendre plus performantes les communications de données échangées en environnement réseau de l'entreprise.

I. La qualité de service (QoS)

La qualité de service est un concept de gestion qui a pour but d'optimiser les ressources d'un réseau et de garantir de bonnes performances aux applications critiques pour l'entreprise. Elle permet d'offrir aux utilisateurs des débits et des temps de réponse différenciés par application suivant les protocoles mis en œuvre au niveau de la structure.

En définitive, la QoS sert à la gestion de la bande passante et englobe tous les mécanismes permettant de différencier les types de trafic, ceux-ci pouvant être classés et administrée différemment à travers le réseau. Mais, avant de parler de qualité de service, il convient de s'interroger sur les critères qui caractérisent le service dans le réseau permettant de délivrer un service de bonne qualité.

Au début de L'internet, la préoccupation majeure était de transmettre les paquets de données à leur destination. Ensuite, des mécanismes liés au protocole TCP/IP ont été développés pour faire face aux conséquences de la perte de paquets ou de la saturation du réseau. Mais depuis le début des années 1990, la communauté des fournisseurs de services qui administre l'internet est confronté non seulement aux problèmes de croissance explosive mais aussi à des aspects de globalisation et stabilité du réseau. **[18]**

La méthode utilisée jusque-là, consistant à fournir des réseaux surdimensionnés ne peut plus s'appliquer indéfiniment. Et l'évolution de l'internet ne permet pas d'offrir une qualité de service constante, ni de donner des priorités à certains types de trafic. C'est pourquoi, les

architectes du réseau, les constructeurs et les fournisseurs de services concentrent aujourd'hui leurs efforts sur la définition et la mise en œuvre de ce concept Quality of Service. **[18]**

La qualité de service est la capacité à véhiculer dans de bonnes conditions un type de trafic donné, en matière de débit, délais de transmission, disponibilité et taux de perte. **[17]**

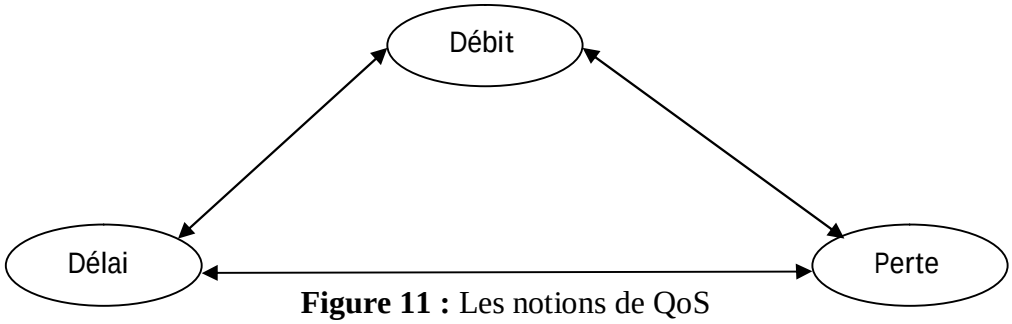

Figure 11 : Les notions de QoS

1. Caractéristiques de la QoS

La qualité de service d'un réseau désigne sa capacité à transporter dans de bonnes conditions des flux issus de différentes applications. Ces flux doivent être traités différemment selon leurs caractéristiques et leurs objectifs fixés: **[18]**

- La fiabilité : Le service d'acheminement des paquets doit être fiable.
- La bande passante : Suffisante pour absorber les flux générés par les applications.
- Le délai : Rapide pour les applications qui le nécessitent.
- La régulation : Trafic régulier pour les applications qui le nécessitent.
- Taux d'erreur : Le plus faible possible, garanti aux utilisateurs.

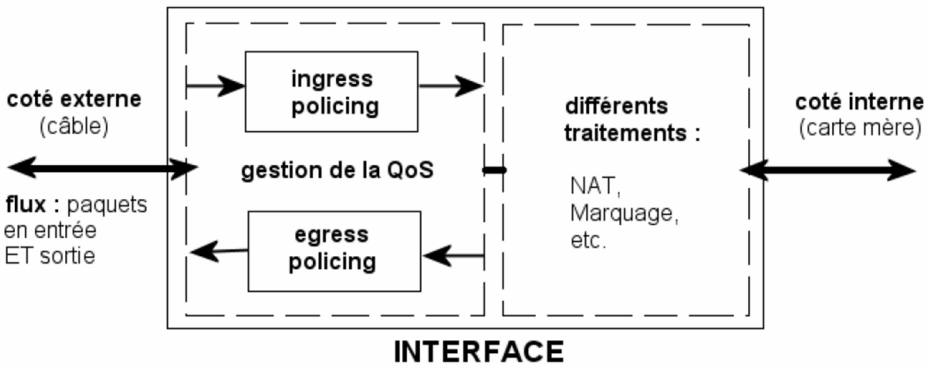

Figure 12 : Interface supportant la QoS **[18]**

Les paquets arrivent sur l'interface par l'intermédiaire du câble réseau (ou ondes dans le cas du WIFI). Ils passent par la gestion de la QoS (paquets entrants / paquets sortants) puis par les différents traitements (NAT, Marquage, etc.) et sortent ensuite de l'interface. Même chose dans l'ordre inverse pour les paquets provenant de la machine (routée : acheminés d'un sous-réseau à un autre par une machine appelée routeur). **[18]**

2. Mécanismes de QoS

2.1. Gestion du trafic :

La gestion du trafic travaille sur les paquets sortant du noyau. Il n'a pas, initialement, pour objectif de contrôler le trafic des paquets entrants. Cette portion de code du noyau se situe entre la couche IP et le pilote du matériel qui transmet sur le réseau. On est donc très bas dans les couches. En réalité, c'est 'trafic control' qui est constamment en charge de transmettre au driver de la carte réseau le paquet à envoyer.

Cela signifie, en fait, que le module TC est en permanence activée dans le noyau, même quand vous pensez pas l'utiliser. Par défaut, ce module maintient une *queue* (prononcer kiou, une file d'attente) similaire à FIFO dans laquelle le premier paquet entré est donc le premier sortie.

La base de TC est la Queuing Discipline qui représente la politique du module appliquée à une *queue.* il existe différentes méthodes de pile. on retrouve les méthodes FIFO, FIFO à plusieurs files, FIFO avec hash et round robin (SFQ). On a également un système Token Bucket Filter (TBF) qui attribue des jetons à une Queuing Discipline pour en limiter le débit. **[19]**

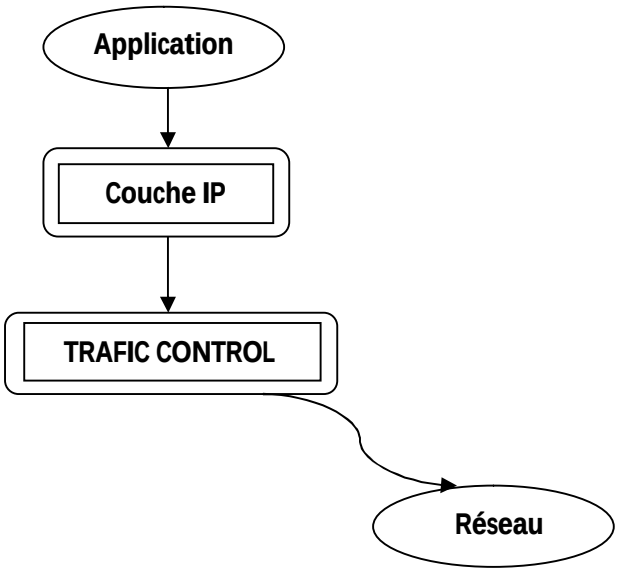

Figure 13 : gestion du trafic

2.2. La priorisation des flux

Les équipements actifs des réseaux pratiquent 'la neutralité des flux', c'est-à-dire qu'ils ne donnent aucune priorité aux trames qui circulent. Ceux-ci entrent dans les commutateurs et y sont traités à la volée. une file d'attente de sortie est associée à chacun des ports du commutateur et le principe du "premier arrivé, premier sorti" (FIFO) est appliquée en cas de concurrence d'accès au port de sortie entre plusieurs trames. Du trafic de faible importance peut donc consommer de la bande passante au détriment de trafics plus importants, voire critiques.

Par exemple, on donnera une priorité aux flux liés à la supervision par rapport à la téléphonie IP ou la vidéo, et une priorité à ces flux multimédias par rapport aux téléchargements FTP. Si un téléchargement dure quelques secondes de plus, ce n'est pas très grave. Par contre, si un flux de téléphonie IP est congestionné, cela peut altérer la qualité de transmission et rendre une conversation inintelligible.

En mode QoS, une série de bits inclus dans la trame ou le paquet IP qui circule indique que cette trame ou ce paquet a une priorité définie entre 0 à 7, selon son degré d'urgence d'acheminement (7 étant le plus urgent), la priorisation ne peut se gérer qu'à l'intérieur de l'équipement : on ne peut pas organiser de file d'attente en entrée, bien évidemment elle va permettre de traiter et de réémettre les trames non pas en fonction de leur ordre d'arrivée, mais en fonction de leur priorité. **[20]**

II. Qualité de service dans Iptables

1. Gestion de la bande passante

Pour faire de la qualité de service, la gestion de la bande passante semble le moyen le plus facile. Dans ce but, les distributions Linux offrent la possibilité de contrôler le trafic pour une bande passante. De plus, de nombreux scripts ont été écrits afin de faciliter la configuration de cette gestion de bande passante.

Linux utilise deux unités du contrôle du trafic pour la gestion de la bande passante :

- Les filtres qui placent le trafic dans les files d'attente (fwmark, u32)
- Les files d'attente qui décident des flux prioritaires (CBQ, HTB, RED, TBF, SFQ...)

CBQ.init : file d'attente CBQ

- Convient à de petits débits
- Nécessite de connaître la taille moyenne des paquets et la vitesse maximale de la connexion
- Utilise le temps d'inactivité de la connexion pour calculer une approximation du débit utilisé.

HTB.init : file d'attente HTB

- Convient à de gros débits
- Consomme peu de ressources
- Ne fais pas d'approximation en ce qui concerne le calcul du débit
- Nécessite de connaitre le débit maximal de votre connexion.

Wondershaper : file d'attente HTB

- Maintien une bonne réactivité pour le trafic interactif (ssh, telnet...)
- Surfer sans souci lors du gros downloade
- S'assure que l'uploade ne défavorise pas le downloade et inversement.

1.1. La commande TC

La commande TC est la commande Linux permettant de gérer la bande passante d'une machine. Le plus gros inconvénient de cette commande, comme nous allons pouvoir le voir, est sa complexité d'écriture.

La commande TC se décompose en trois parties :

- La configuration des disciplines de file d'attente
- La configuration des classes de trafic
- La configuration du filtrage de flux

1.1.1. Configuration des files d'attente

La commande se décompose en plusieurs parties :

– Le nom de la commande Tc

– l'action à faire sur cette queue

– l'interface réseau sur laquelle elle s'applique introduite par dev

– le point de montage de cette queue : soit en root (première) soit en parent (queue fils d'une autre queue)

– le nom de la queue (un numéro) introduit par handle

– et enfin la discipline correspondante (l'ordonnancement) accompagné de ses paramètres (comme le débit ou encor la classe par défaut)

Cette commande permet donc de créer des queues ordonnancement avec la possibilité de mettre en place une hiérarchie dans ces dernières.

1.1.2. Configuration des classes

Cette commande est quasiment identique à la précédente dans sa description.

– Le début de la commande est identique (« tc class add dev »)

– Ensuite on associe la classe à un ordonnanceur par parent et on lui attribue un identifiant (par classid)

– On ajoute la technique d'ordonnancement de la queue mère

– Puis on définit les paramètres associés à la classe comme par exemple le niveau de priorité, le débit de la classe …

1.1.3. Configuration des filtres

Le début est également identique aux précédentes

– On définit sur quelle queue s'applique le filtrage

– Ensuite les différentes caractéristiques de filtrage sont ajoutées en commençant par le protocole, puis d'autres comme l'adresse source ou encore un champ particulier d'une trame.

– Enfin un identifiant est attribué au trafic

1.1.4. Les classifieurs

Les classifieurs sont des options pour la commande Tc filter afin de repérer des champs particuliers dans une trame. Ces classifieurs sont au nombre de trois : route, **Fw** et **u32**.

a. Le classifieur route

Ce classifieur permet de filtrer les trafics en fonction du marquage sur la table de routage. Il doit donc être utilisé conjointement avec la commande IP.

Lors de déclarations de routes avec la commande IP, il faut attribuer à ces routes un numéro, appelé REALMNUMBER, introduit par REALM.

Le filtrage des paquets se fait donc sur ce numéro, avec des options comme **from** (en provenance de) ou **to** (à destination de).

b. Le Classifieur fw :

Ce Classifieur se base sur les marquages effectués à l'aide d'IPtables/Netfilter. Pour faire ce marquage, il est possible de marquer le paquet passant au travers des règles de firewall ou uniquement en utilisant la table mangle (table d'Iptables servant à marquer des paquets) afin de marquer les paquets sans faire d'autre manipulation.

c. Le Classifieur u32

Ce classifieur utilise uniquement la commande Tc. Le principe de ce classifieur est simple puisqu'il s'agit de filter les paquets en fonction d'un champ particulier de la trame, champ pointés par le marqueur u32.

1.1.5. La commande Iptables

Le noyau Linux supporte directement QoS grâce à IPtables, c'est ce qui vous donnera la plus grande flexibilité. Comme on a perçu dans le chapitre précédant la table principale pour la gestion de la qualité de service est la Table Mangle.

Lorsqu'un paquet IP est envoyé depuis une application, il transite par la table MANGLE d'iptables. A ce moment, il est possible d'effectuer des modifications sur le paquet IP, comme une réécriture d'adresse, ou la modification d'un champ IPv4 .pour la gestion de la QoS on modifie la valeur du champ TOS du paquet IPv4. Lorsque les modules de QoS sont chargés dans le noyau linux le paquet est ensuite traité par le module de gestion de la QoS. **[23]**

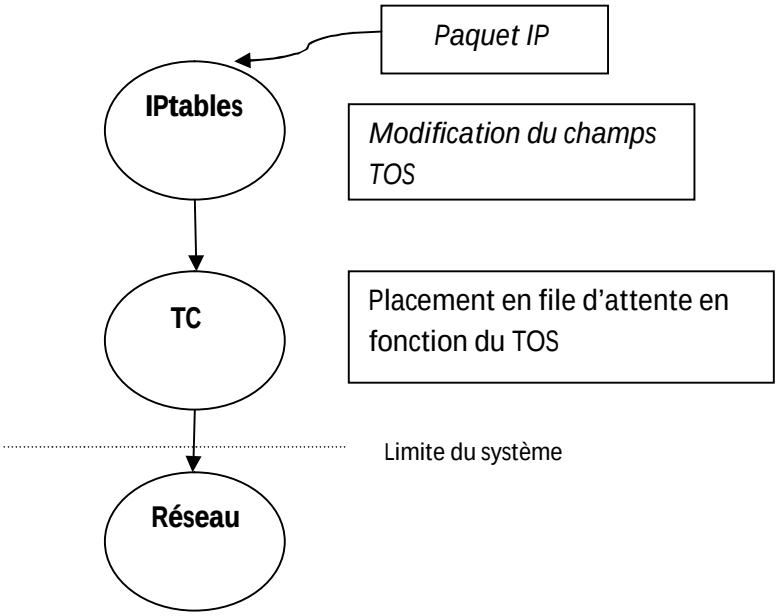

Figure 14 : Traitement d'un paquet pour la QoS

1.2.1. La Table Mangle :

Comme il a déjà été précisé, le rôle principal de cette table devrait être de modifier des paquets. En d'autres termes, vous pouvez utiliser en toute liberté les correspondances de la table mangle, qui permettent de changer les champs de TOS (type de service), et d'autres. Par contre cette table n'est pas utilisable pour effectuer de filtrage, de même les opérations de DNAT, SNAT ou de masquerading ne fonctionnent pas dans Mangle. **[14]**

Les cibles suivantes sont valides uniquement dans la table mangle. elles ne doivent pas être utilisées en dehors de cette table.

- TOS
- TTL
- MARK
- SECMARK
- CONNSECMARK

1.2.1.1. La cible TOS

Permet de définir et modifier le champ de type de service d'un paquet. C'est utile pour définir des stratégies réseau concernant le choix de routage des paquets nous traitons plus en détail ultérieurement.

a. Caractéristique de la cible TOS:

La cible TOS sert à disposer le champ type de service dans un en-tête IP. Le champ TOS consiste en 8 bits utilisés pour aider au routage de paquets. C'est un des champs qui peut-être utilisé directement dans **iproute2** et son sous-système pour les stratégies de routage, notez de plus que si vous maintenez plusieurs pare-feu et routeurs séparés, c'est le seul moyen pour propager les informations de routage dans les paquets entre ces routeurs et pare-feu, comme on a mentionné précédemment la cible MARK laquelle dispose un MARK associé à un paquet spécifique et disponible seulement dans le noyau, et ne peut pas être propagée avec le paquet. Si vous avez besoin de propager des informations de routage pour un paquet spécifique ou un flux, vous devrez donc placer le champ TOS, qui a été créé pour ça.

Il existe un grand nombre de routeurs sur internet qui font du mauvais travail à ce sujet, ce qui fait qu'il peut être moins utile maintenant d'essayer de faire de l'analyse TOS avant d'envoyer les paquets sur Internet. Dans le meilleur des cas les routeurs ne font pas attention au champ TOS. Dans le pire, ils examineront le champ TOS et feront de mauvaises choses, cependant le champ TOS peut être placé avec une grande utilité si vous avez un grand WAN ou LAN avec de multiples routeurs, vous avez la possibilité de donner aux paquets différentes routes et préférence, basées sur leur valeur TOS. **[22]**

b. La composition TOS

Voila la composition du champ services TOS

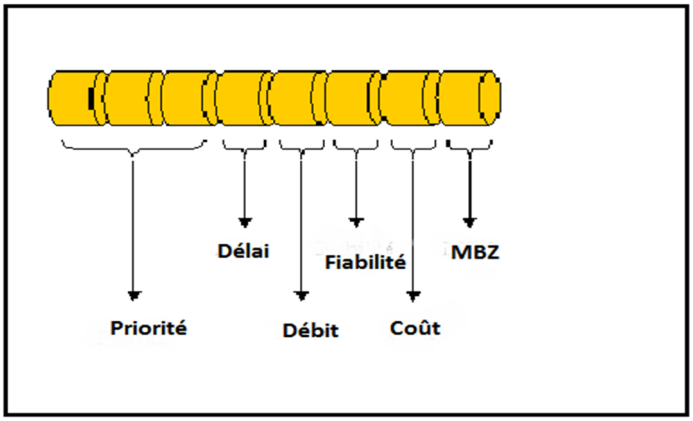

Figure 15: le champ TOS [21]

- **Priorité**

Le champ priorité (précédence) est codé sur 3 bits. Il indique la priorité que possède le paquet. Voici les correspondances des différentes combinaisons :

- -0- 000 Routine
- -1- 001 Prioritaire
- -2- 010 Immédiat
- -3- 011 Urgent
- -4- 100 Très urgent
- -5- 101 Critique
- -6- 110 Supervision interconnexion
- -7- 111 Supervision réseau

- **Délai**

Le champ Délai (Delay) est codé sur 1 bit. Il indique l'importance du délai d'acheminement du paquet. Voici les différentes combinaisons :

- -0- Normale
- -1- Bas

- **Débit**

Le champ débit (Throughput) est codé sur 1 bit. Il indique l'importance du débit acheminé. Voici les correspondances des différentes combinaisons :

- -0- Normal
- -1- Haut

- **Fiabilité**

Le champ Fiabilité (Reliability) est codé sur 1 bit. Il indique l'importance de la qualité du paquet. Voici les correspondances des différentes combinaisons :

- -0- Normal
- -1- Haut

- **Coût**

Le champ Coût est codé sur 1 bit. Il indique le coût du paquet. Voici les correspondances des différentes combinaisons :

- -0- Normal
- -1- Faible

- **NBZ**

Le champ NBZ (Must Be Zero) est codé sur 1 bit. Comme son nom l'indique, il doit être mis à 0.

1.2.1.2. La cible TTL

Permet de modifier le champ durée de vie ou TTL (Time To Live) d'un paquet. Il est possible par exemple de spécifier aux paquets d'avoir un champ TTL spécifique. Ceci peut se justifier lorsque vous ne souhaitez pas être rejeté par certains fournisseurs d'accès à Internet (FAI) trop indiscrets. En effet, il existe des FAI qui désapprouvent les utilisateurs sur une même connexion, et de fait, quelques-uns de ces FAI sont connus pour vérifier si un même hôte génère différentes valeurs TTL, supposant ainsi que plusieurs machines sont branchées sur la même connexion

1.2.1.3. La cible MARK

Permet d'associer des valeurs de marquage particulier aux paquets. Elles peuvent ensuite être identifiées par les programmes iproute2 pour appliquer un routage différent en fonction de l'existence ou de l'absence de telle ou telle marque. On peut ainsi réaliser de la restriction de bande passante et de la gestion de priorité (Class Based Queuing).

1.2.1.4. La cible SECMARK

Peut être utilisée pour placer des marques dans un contexte de sécurité sur des paquets dans SELinux ou tout autre système de sécurité capable de gérer marques. Autre système de sécurité capable de gérer ces marques.

CONNSECMARK sert à copier un contexte de sécurité vers ou depuis un simple paquet ou vers une connexion complète. Elle est utilisée par SELinux ou autres système de sécurité pour affiner cette sécurité au niveau connexion.

Conclusion

Au cours de ce chapitre, nous avons abordé les différents aspects liés à la qualité de service qui devient un impératif pour tout entreprise qui se respecte pour maintenir sa place dans le marché ou les clients deviennent de plus en plus exigeants et cela du fait de la mondialisation et de la libre circulation des services. ainsi l'ensemble des techniques permettant d'assurer la qualité du service réseau à titre d'exemple, nous avons décrit comment contrôler la bande passante et encore assigner des priorités aux flux réseau afin de pouvoir garantir de la QoS.

Dans la deuxième partie de ce chapitre, nous avons vu la solution Iptables pour garantir la qualité de service, ainsi la table Mangle qui permet de changer les champs TOS pour assurer le type de service.

Dans le chapitre qui suit nous allons réaliser notre application de fonctionnement de l'application réalisée qui est un script de configuration de gestion de bande passante sur un noyau linux.

Chapitre 4 : Conception et réalisation

Introduction

La qualité de service est une exigence qui reboote sur les réseaux actuels ayant un rôle de fournir des services prévisibles, mesurables et parfois garantis. L'architecture réseau à commutation de paquets ne garantit pas que tous les paquets composant un message particulier arriveront à temps, dans l'ordre voulu, ni même qu'ils arriveront. Les réseaux ont également besoin de processus leur permettant de gérer l'encombrement du réseau, lorsque plusieurs communications sont initiées simultanément sur le réseau, la demande de bande passante peut excéder la quantité disponible, créant ainsi un encombrement du réseau qui a simplement plus de bits à transmettre que ce que la bande passante du canal de communication peut prendre en charge. **[20]** Pour faire de la qualité de service, la gestion de la bande passante semble le moyen le plus facile, dans ce but le noyau linux supporte directement la qualité de service grâce à iptables. Ce chapitre a pour objectif de mètre en œuvre la qualité de service dans un réseau à travers un scénario afin de répondre à des besoins précis.

I. L'environnement de travail

Pour la réalisation de notre projet, on a utilisé la virtualisation comme une technologie.

La virtualisation consiste à faire fonctionner plusieurs systèmes d'exploitation sur une seule machine, tout en donnant l'illusion parfaite qu'elles se situent sur des ordinateurs physiques différents. C'est une technologie qui devient de plus en plus efficace avec les microprocesseurs récents, qui sont généralement dotés de technologies de virtualisation améliorant les performances.

Il existe de nombreux outils (appelés **Hyperviseurs** : VirtualBox, VMWare, Virtual PC, etc.). Dans notre cas on a utilisé le Hypervisur VirtualBox.

1. Présentation du VirtualBox

VirtualBox ou machine virtuelle est un logiciel gratuit, open source et multi-platformes. En utillisant les resources matèrielles de l'ordinateur (*système hôte*), VirtuelBox permet la création d'un ou plusieurs ordinateurs virtuels dans lesquels s'installent d'autres systèmes d'exploitation (*système invité*). **[25]**

2. Les modes réseaux VirtualBox

VirtualBox fournit jusqu'à huit cartes Ethernet PCI virtuelles pour chaque machine viertuelle. Pour chaque carte sélectionnée, vous puvez choisir le mode de virtualisation effectué par la carte virtuelle par rapport à votre matèriel réseau physique sur l'hôte. Les principaux modes sont :

- NAT : - Les machines virtuelles communiquent entre elles
  - Les machines virtuelles communiquent avec l'hôte et l'extérieur
  - L'hôte et l'extérieur ne voient pas les VM (ex : un ping ne fonctionne pas).
- Réseau Interne : - Les machines virtuelles communiquent entre elles
  - Les machines virtuelles communiquent avec l'hôte
  - Les machines virtuelles ne communiquent pas avec l'extérieur
- Réseau Privé Hôte : - les machines virtuelles communiquent entre elles
  - Les machines virtuelles communiquent avec l'hôte
  - Les machines virtuelles ne communiquent pas avec l'extérieur
- Pont: - les machines virtuelles sont sur le même réseau que l'hôte

II. Scénario

1. Les moyens utilisés

- ✓ Une machine serveur (ubuntu server) pour gérer les flux de données
- ✓ Un serveur FTP (proftpd) installé sur la machine physique (ubuntu desktop)
- ✓ 2 machines clientes (ubuntu desktop)

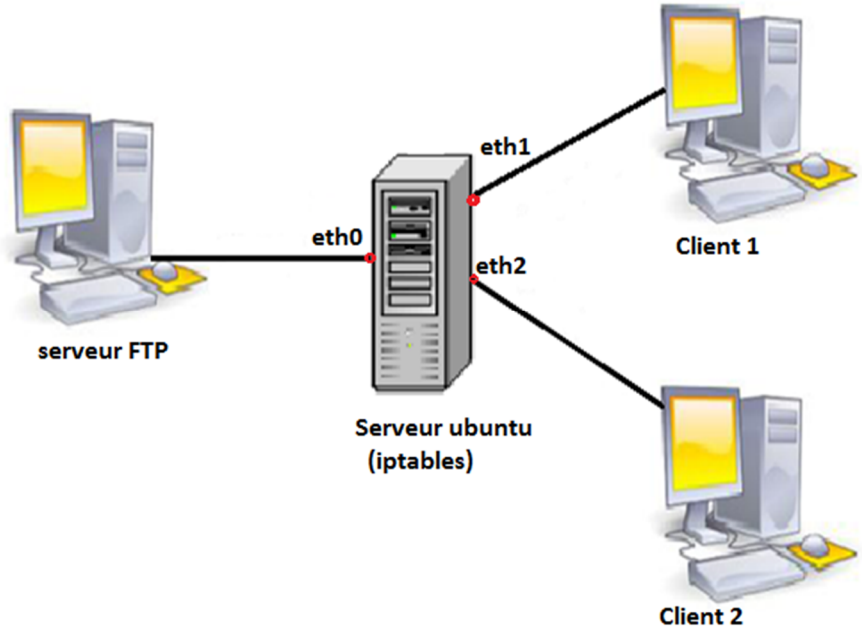

Figure 16 : Scénario représentatif

2. Objectif

Ce que l'on souhaite :

- Autoriser l'accès des clients au serveur FTP
- Limiter la bande passante de téléchargement FTP pour le client 1 seulement

III. Les étapes de réalisation du projet

Après la création des machines virtuelles et l'installation de ses systèmes d'exploitation, voici les étapes principales à suivre :

1. Installation et configuration du serveur FTP

Pour cela nous allons utiliser le serveur proftpd qui est un serveur libre et le plus puissant et le plus sécurisé. [26]

Afin d'installer le serveur sur notre distribution de linux (ubuntu 14.04), nous allons utiliser comme d'habitude le terminal, moyen le plus simple et plus rapide pour y arriver.

✓ Ouvrir le terminal et tapez la commande suivante après être passé en super utilisateur grâce à la commande *sudo su* :

apt-get install proftpd

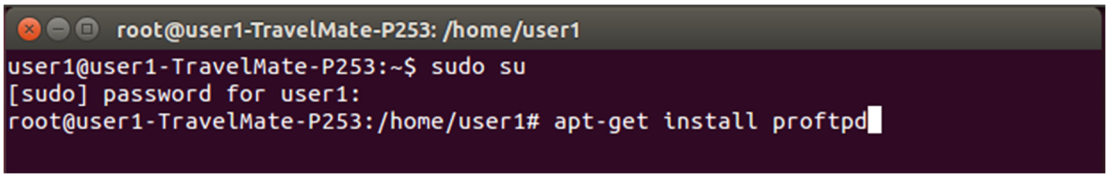

Figure 17 : Installation proftpd

✓ On vous demandera ensuite si vous voulez installez le serveur en mode indépendant ou depuis inetd choisissez **indépendant**

✓ Le serveur sera ensuite installé. La configuration de proftpd se fait dans un seul fichier proftpd.conf qui se trouve dans /etc/proftpd/. Nous allons donc nous rendre dans le dossier proftpd et modifier le fichier proftpd.conf (modifier le nom du serveur).

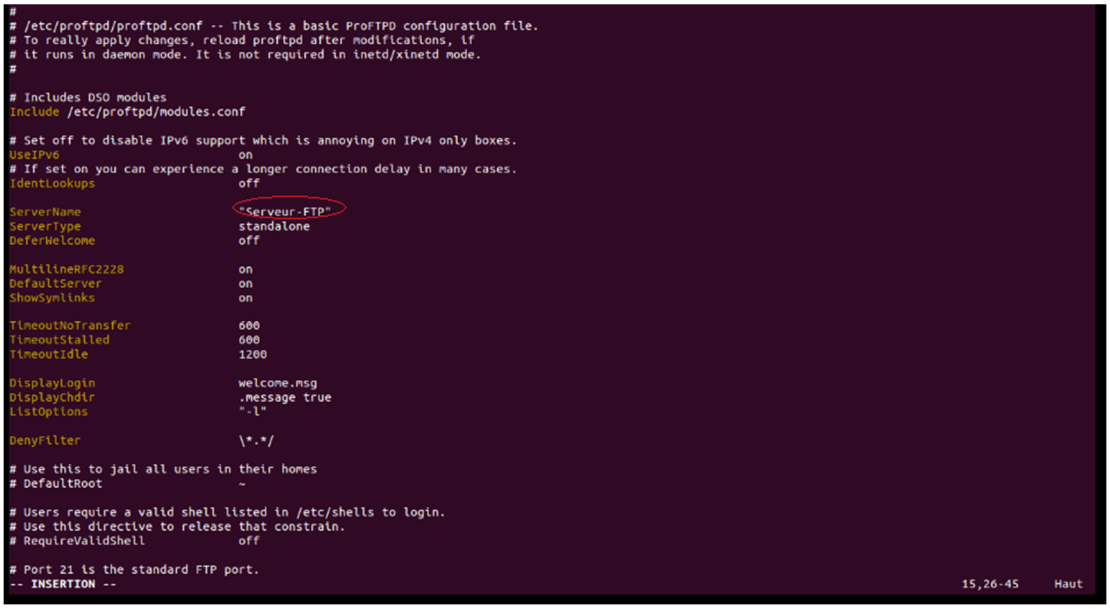

Figure 18 : Configuration proftpd

✓ Afin de quitter et enregistrer le fichier, il faut utiliser la commande suivante :

:wq!

✓ Nous allons redémarrer le serveur afin de prendre en compte les changements grace à la commande :

service proftpd restart

✓ Afin de vérifier si le serveur fonctionne rendez-vous à l'adresse ftp://127.0.0.1 dans votre navigateur, on vous demandera un nom d'utilisateur et un mot de passe qui est celui de votre compte utilisateur :

Figure 19 : Authentification FTP

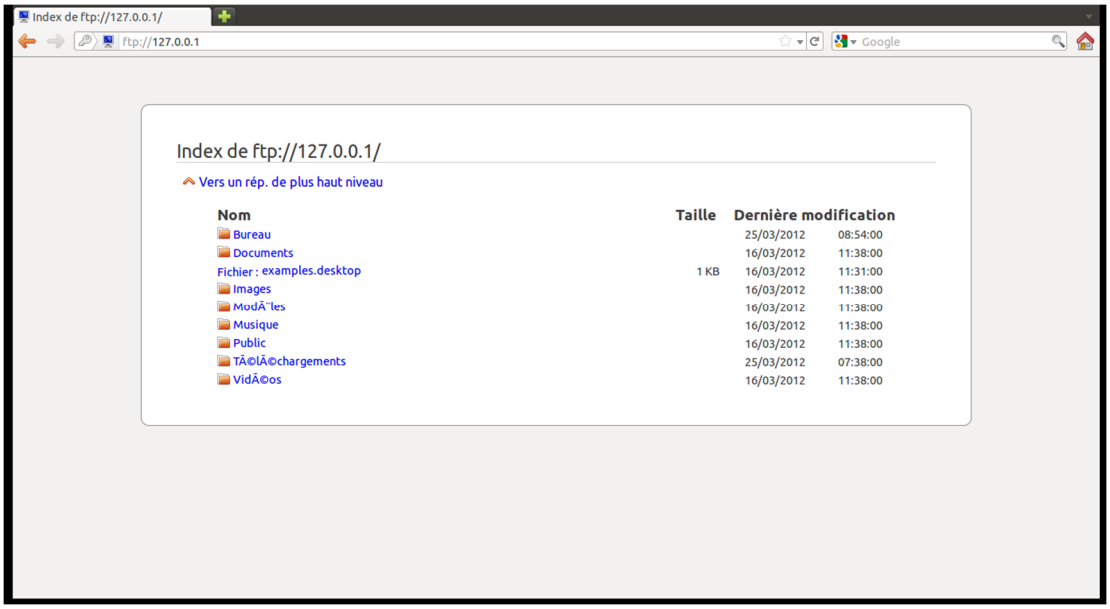

Figure 20 : Index de FTP

2. La configuration réseau

2.1. Configuration des interfaces

✓ Dans notre serveur ubuntu, nous allons crées 3 interfaces (eth0 : pour l'hôte (serveur ftp), eth1 : pour client 1 et eth2 : pour client2). Pour cela nous allons premièrement activer 3 adapteurs (3 cartes réseaux) en cliquant sur la **configuration** (pour le serveur) et choisissant **Réseau**

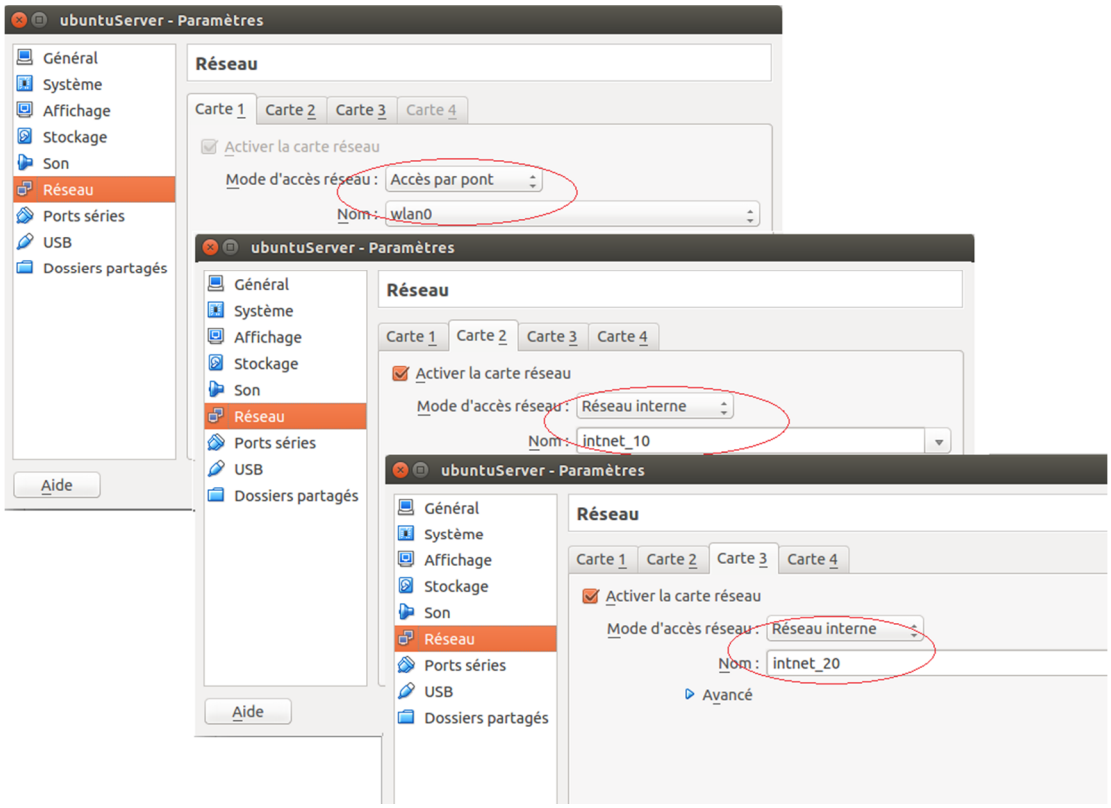

Figure 21 : Activation des 3 cartes réseaux du serveur

- ✓ Puis, pour la création de ses interfaces, éditez le fichier de configuration des interfaces : */etc/network/interfaces* et taper les informations des adresses

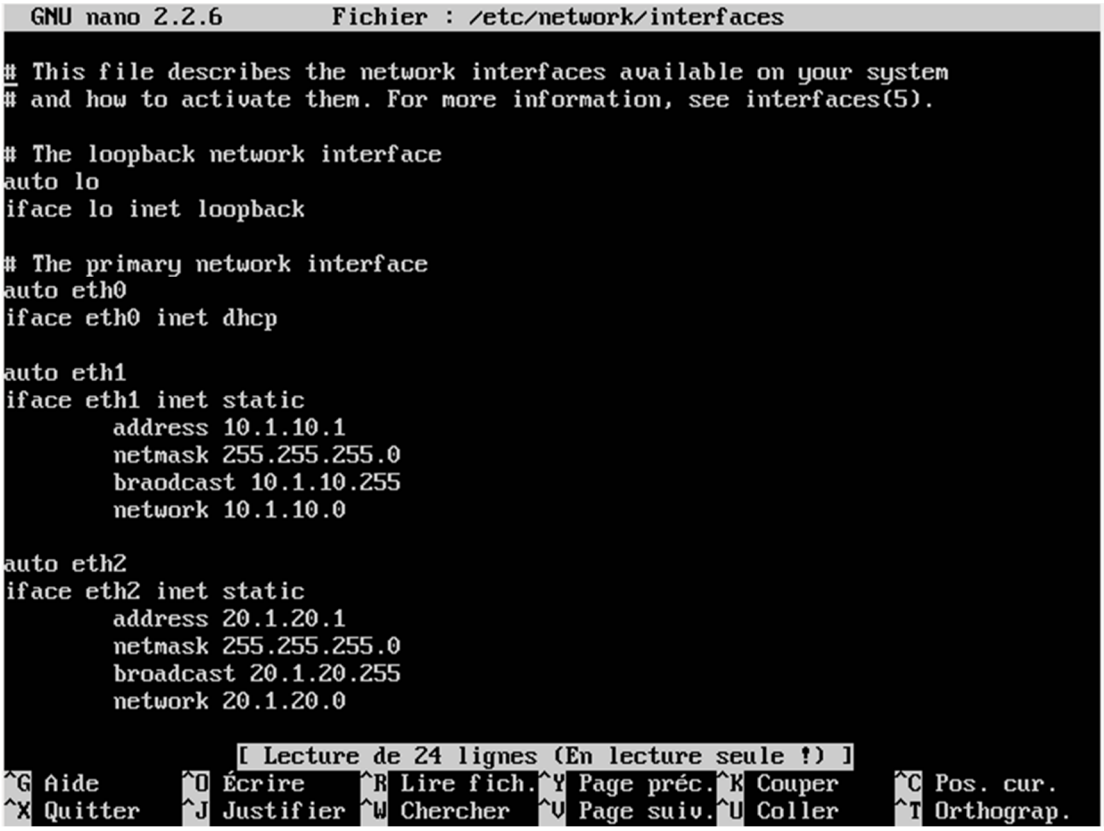

Figure 22 : Création des interfaces

✓ Ensuite, on a donné des adresses aux clients, les adresses sont dans le même réseau que leurs interfaces : Adresse_client1 : 10.1.10.100

Adresse_client2 : 20.1.20.100

2.2. Le routage

✓ Nous allons installer le logiciel **bird** pour obtenir les routes BGP du routeur du collège par la commande apt-get install bird

✓ Une fois installé, modifiez le fichier /etc/bird.conf.

✓ On relance **bird** par la commande /etc/init.d/bird restart

✓ Sur le routeur du collège, on tape les commandes Cisco pour tracer la table de routage **ospf-bgp** qu'on a utilisé via le logiciel **bird** pour remplir la nouvelle table.

3. Gestion de la Qos (limitation de la bande passante FTP pour le client 1)

Tout d'abord, pour activer la gestion de la qualité de service sous Linux, il faut que les options CBQ,SFQ et HTB aient été compilés comme module du noyau. Pour notre cas, on a chargé le module HTB par la commande suivante :

modprobe sch_htb

Ensuite, pour pouvoir utiliser la commande **TC**, qui permet de gérer les files d'attente au niveau du noyau Linux, il faut installer le paquetage iproute :

```
apt-get install iproute
```

Voici les commandes principales qu'on a utilisées dans notre scripte:

```
## Initialisation des tables
  # Initialisation de la table Filter
iptables  -t filter -F
iptables  -t filter -X
iptables  -t filter -P INPUT DROP
iptables  -t filter -P OUTPUT DROP
iptables  -t filter -P FORWARD DROP
  # Initialisation de la table Nat
iptables -t nat -F
iptables -t nat -X
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
  # Initialisation de la table Mangle
iptables -t mangle -F
iptables -t mangle -X
iptables -t mangle -P PREROUTING ACCEPT
iptables -t mangle -P INPUT ACCEPT
iptables -t mangle -P OUTPUT ACCEPT
iptables -t mangle -P FORWARD ACCEPT
iptables -t mangle -P POSTROUTING ACCEPT
## Initialisation de TC
tc qdisc del dev eth1 root
## Création des classes pour la bande passante
  # Création de la classe principale
tc qdisc add dev eth1 root handle 13 :0 htb default 1
  # Ses classes permettant de brider le debit
tc class add dev eth1 handle 13 :0 classid 13.1 htb rate 54000 bit
tc class add dev eth1 handle 13:2 classid 13:2    htb rate 5000 bit
  # Et on conserve les prio en fonction du TOS
```

```
tc qdisc add dev eth1 parent 13 :1 handle 131 prio
tc qdisc add dev eth1 parent 13:2 handle 132 prio
  # Les flux FTP marqués (21) sortant sont envoyés dans la classe 2 qui bride
tc filter add dev eth1 parent 13 :  protocol ip handle 21 fw classid 13:2
## Création des règles
  # Création de la chaîne et association sur les paquets sortants du routage vers le client
iptables -t mangle -N TRAFFIC-ENTRANT
iptables -t mangle -I POSTROUTING -o eth1 -j TRAFFIC-ENTRANT
   # Association traffic <> Mark selon le type de flux
iptables -t mangle -A TRAFFIC-ENTRANT -p tcp -m lenght --lenght :64 -j MARK --s-mark
20
iptables -t mangle -A TRAFFIC-ENTRANT -p tcp --sport 20 -j MARK --set-mark 21
iptables -t mangle -A TRAFIIC-ENTRANT -p tcp --dport 20 -j MARK --set-mark 21
```

4. Vérification du résultat obtenu

Pour vérifier le résultat obtenu, on télécharge le meme fichier (meme taille de fichier) par le client 1 et client 2, et on voir la déférence du temps de téléchargement pour client 1 et client 2

- ✓ Dans le navigateur des clients, tapez l'adresse suivant :

ftp://@_wlan0

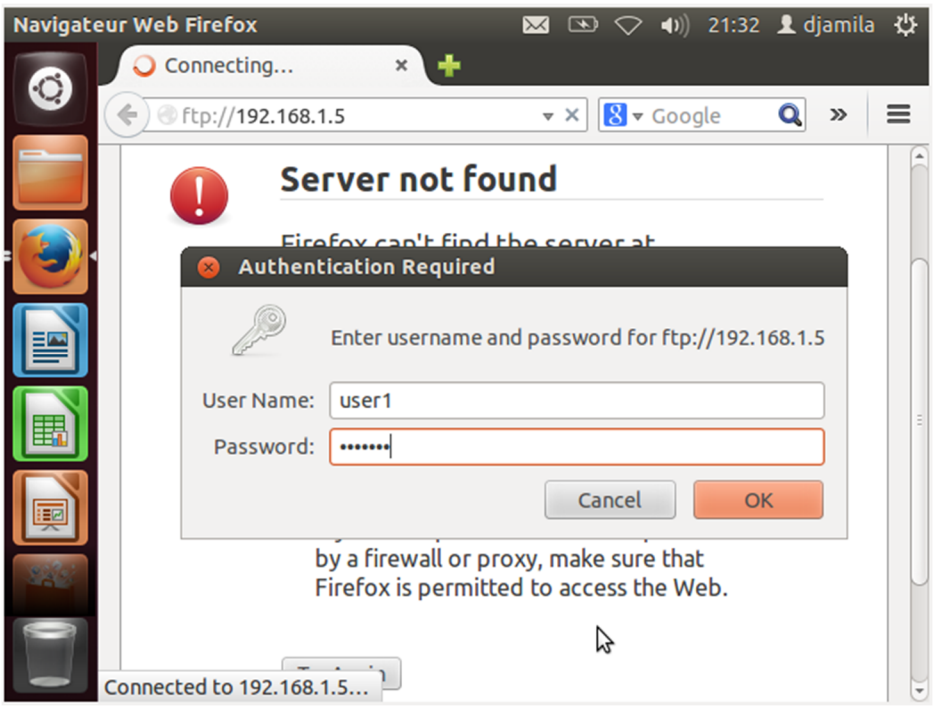

Figure 23 : Authentification FTP par client

- ✓ Puis sélectionner le fichier à télécharger

Voilà les propriétés de téléchargement pour chaque client :

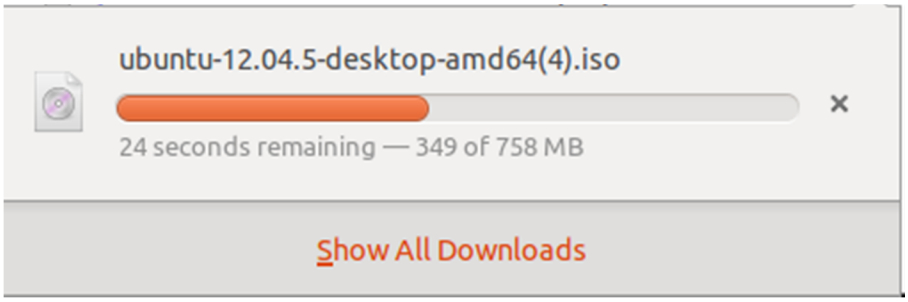

Figure 24 : Téléchargement client 2

Conclusion

Iptables gère plus finement les flux. Nous avons vu dans ce chapitre les étapes qui nous ont permis de créer une façade de la Qualité de Service grâce au couple Iptables/TC. Cette façade a touchée principalement les deux champs (TOS et MARK) de la table mangle et alors on a obtenu finalement une bonne qualité de service selon nos besoins.

Conclusion générale

Mettre un firewall sur son système est aujourd'hui un pré requis indispensable pour naviguer, protéger et gérer le réseau. Le protocole TCP/IP n'a pas d'aptitude à connaître les performances de son réseau, il commence à envoyer des paquets, de plus en plus rapidement et quand des paquets commencent à se perdre, il ralentit. La plupart des files d'attentes fonctionnent selon le modèle suivant : elles reçoivent des paquets, les positionnent en file d'attente jusqu'à un certain point, et ensuite, éliminent tout paquet qui arrive dans le cas où la file d'attente est pleine. Si on travail on UDP, les paquets ne sont plus retransmis, si c'est du TCP, l'émetteur renverra les paquets perdus. Le débit s'en trouve alors ralenti. L'engouement pour internet et les réseaux en générale amène à développer de nouvelles techniques pour parer aux limites du protocole TCP/IP. La qualité de services est l'une de ces techniques qui permet de jouer sur le débit et de prioriser certains flux afin d'obtenir un service garanti aux protocoles et sous réseau qui le nécessite.

Iptables est une solution complète de pare-feu. Il assure de la qualité de service grâce à la table Mangle et ses champs (particulièrement les champs TOS et MARK)

Notre objectif est de présenter la qualité de service offrit par l'iptables, tel que l'intérêt de cette QoS est associé à la priorisation des flux et la gestion de la bande passante.

Finalement, pour que notre firewall soit efficace, il doit posséder plusieurs interfaces réseaux pour faire un filtrage entre plusieurs zones. L'idéal c'est qu'à chaque zone du SI (système d'informatique) correspondre une interface de cette façon et grâce au couple iptables/TC, nous avons filtrés et gérés plus finement notre flux de données et nous avons donc réalisé notre façade de qualité de service avec succès.

Bibliographe

[1] Jean-Pierre Arnaud, Réseaux & télécoms, 3e édition, Paris, 2009

[2] R.mezari & M.Lahdir, *Réseaux Locaux, Alger, 2006*

[3] Stevens R, TCP/IP illustré, les protocoles, volume 1, vuibert informatique, paris, 1998

[4] Zouheir Trabelsi, L'espionnage dans les réseaux TCP /IP paris, 2005

Webographie

[5] https://fr.scribd.com/doc/82449229/La-securite-des-reseaux-informatique

[6] http://kijack.perso.neuf.fr/guill.web-/Tomnat.html

[7] http://webadonf.net/2011/03/regles-de-filtrage-avec-iptables/

[8] http://dindinx.net/formation-netfilter/formation-iptables.html

[9] http://www-lipn.univ-paris13.fr/~kanawati/R4/TP/RT-R4-TP6.pdf

[10] https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html

[11] http://lacl.u-pec.fr/cegielski/sec/ch5.pdf

[12] https://www.frozentux.net/iptables-tutorial/fr/c461.html

[13] https://www.netfilter.org

[14] http://www.inetdoc.net/guides/iptables-tutorial/mangletable.html

[15] http://fr.slideshare.net/miyamiya/qos-of-wlan-wifi

[16] http://webcache.googleusercontent.com/search?q=cache

[17]http://wiki.linuxwall.info/doku.php/fr:ressources:dossiers:networking:traffic_control$

[18] http://www.reseaucerta.org/docs/cotecours/C%C3%B4t%C3%A9%20cours-QoS-V1.0.pdf

[19] http://www.inetdoc.net/guides/iptables-tutorial/tostarget.html

[20] https://www.youtube.com/watch?v=etuEEc86LoU

[21] http://www.frameip.com/entete-ip/

[22] http://www.inetdoc.net/guides/iptables-tutorial/tostarget.html

[23] http://files.vivien.boistuaud.net/public/ingenieurs2000/ir3-diffserv-qos-linux-tp_boistuaud_fraux_herr.pdf

[24] https://doc.ubuntu-fr.org/virtualisation

[25] https://doc.ubuntu-fr.org/virtualbox

[26] http://www.lolokai.com/blog/2012/04/04/installation-dun-serveur-ftp-sous-ubuntu-11-10/

[27] https://doc.ubuntu-fr.org/routage