DOCUMENTATION
RESEAU GINFO

Table des matières

I – Possessions du GInfo

Les possessions en termes de matériel du GInfo se résument à deux cas. Les matériels dit actif et les matériels dit passif.

A – Matériels actif

La liste des matériels actifs est :

- 1 Hub D-Link DE-816TP
- 1 Switch Allied Telesyn AT-FS742i
- 2 Switchs Allied Telesyn AT-8350GB
- 4 Switchs Allied Telesyn AT-8326GB
- 2 Serveurs Dell PowerEdge 1600SC
- 2 Serveurs Dell Optiflex GX280
- 1 PC Dell Optiflex GX280 mini
- 3 PC Dell Optiflex GX280 tour
- xxx PC Dell Optiflex 755 mini
- xxx PC Dell Optiflex GX520
- 1 Serveur Dell xxx
- 1 Serveur Dell xxx
- 1 baie de disque

Sur cette liste est utilisé de manière continue 1 serveur Dell PowerEdge 1600SC en tant que cœur de réseau, 1 Serveur Dell xxx en tant que machine de virtualisation, 1 PC Dell Optiflex 755 mini en tant que machine du local, 1 baie de disque en tant que système de stockage.

B – Matériels passifs

Le GInfo possède un certain nombre de matériels passifs dont des câbles, des souris et des claviers (non dénombrés dans ce document).

II – Organisation du réseau

A – Bases et aspects légaux du réseau ainsi que de son accès Internet

Il existe dans le réseau du GInfo (ci-après dénommé réseau ou GInfo) deux sous-réseaux distincts (ci-après dénommé réseaux ou réseau interne).

Les plages d'Internet Protocol v4[1] (ci-après dénommé I.Pv4 ou IP) utilisées sont 10.61.15.0/24 et 10.61.16.0/24 (ci-après dénommés réseau 15 (respectivement réseau 16) ou lan 15(respectivement lan 16)). Il n'existe pour l'heure pas encore d'adressage IPv6[2]

Les différents sous-réseaux sont séparés entre eux et ne communiquent que par le biais du pare-feu du cœur de réseau (ci-après dénommé cœur ou routeur central). Le réseau 15 servant à l'ensemble des serveurs et donc des services proposés par le GInfo (voir III) et le réseau 16 permettant aux membres de venir au local afin de bénéficier d'une connexion haut-débit et d'une convivialité certaine.

L'accès à Internet est assuré par la Direction des Services Informatique de l'École Centrale Marseille (ci-après dénommé DSI ou CRI et ECM) par le biais de leur système de portail captif sur lequel tout membre du GInfo devra se connecter avec son identifiant ECM. L'IP fournit à la DSI au sein du GInfo est 10.61.16.254 . Ceci permet d'assurer la traçabilité des personnes se connectant à Internet au sein du GInfo de manière simple pour les deux parties (GInfo et CRI) afin de répondre notamment à des exigences légales.

B – Cœur de réseau et connectivité

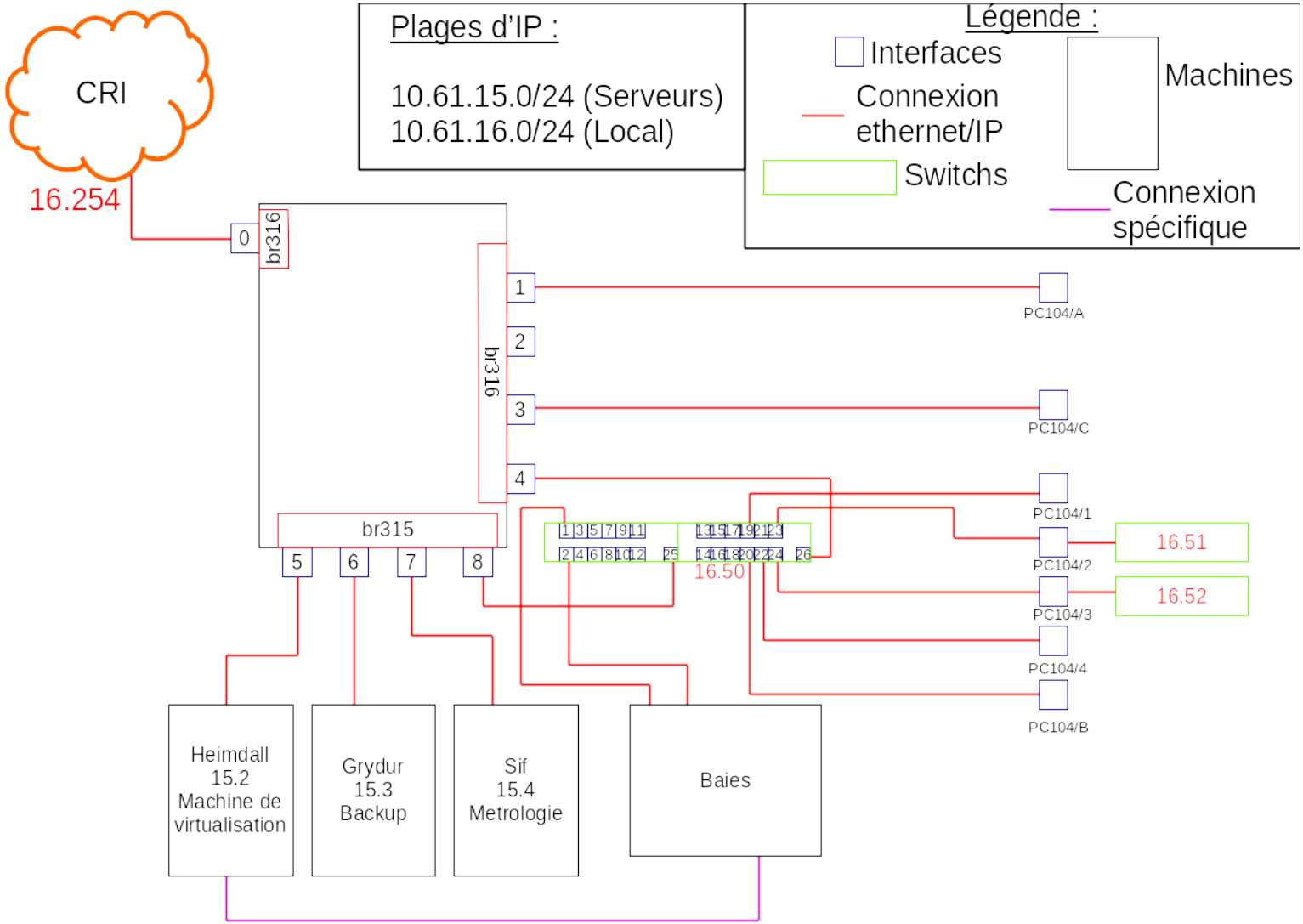
Le cœur possédant 9 interfaces physiques (numérotées de 0 à 8 et nommées eth0,…) celle-ci ont été mise ensemble dans les deux sous-réseaux de la manière qui suit :

- Interfaces eth0, eth1, eth2, eth3 et eth4 dans le bridge[3] br316 (ci-après appelé 316)
- Interfaces eth5, eth6, eth7 et eth8 dans le bridge br315 (ci-après appelé 315)

L'accès à Internet n'est possible dans cette configuration que sur le réseau 316 a l'heure actuelle. Une ou plusieurs solutions sont à l'étude parmi lesquels faire un changement de gateway[4] et router le lan 315 sur le lan 316

C – Schématisation des connections, des interfaces du réseau et des plages d'IP

Le schéma complet du réseau est le suivant :



III – Configuration logicielle du coeur

Le coeur est un outil essentiel pour le fonctionnement proprement dit du réseau au complet en effet son objectif est quadruple.

Il doit tout d'abord assurer par le biais d'un serveur DHCP[5] l'adressage des IP aux personnes (physique ou serveur).

Ensuite il doit aussi assurer, par le biais d'un serveur DNS[6], les noms des domaines internes du réseau afin de permettre aux gens d'accéder aux différents services et aux administrateurs une facilité d'accès sans se soucier de retenir toutes les IP.

De plus il doit permettre aux personnes physiques présentes dans le local d'avoir un accès haut-débit câblé.

En quatrième point et en lien avec le troisième point il doit être capable de filtrer les connexions (au sens des accès autorisés ou non à certains ports et à certains sites) par le biais d'un pare-feu[7], intégré au sein du noyau Linux qui est IPTable[8]. Ce pare-feu permet aussi de faire le routage depuis le local vers les services autorisés.

On peut donc constater ici que ce serveur possède une criticité assez élevé et c'est pourquoi il est fortement recommandé, sinon obligatoire de faire des sauvegardes de chaque fichier avant d'en faire une modification par exemple par le biais de la commande cp en appelant le nouveau fichier <même nom>.bak

De plus il y a sur ce serveur un ensemble de script écrits en Python permettant de faire des modifications de manière plus sécurisé. Ces scripts sont disponibles dans /root/scripts

Afin de se simplifier le travail et d'éviter que certains services ne fonctionnent sur plusieurs interfaces il a été décider de créer un bridge sur toutes les interfaces appartenant à un même sous-réseau afin de travailler comme si nous avions qu'une interface au niveau logiciel.

Nous commencerons donc par expliquer les différentes interfaces, ce qu'est un bridge au sens réel du terme et dans notre cas et ensuite nous expliciterons ce qui a été dit plus haut sur le positionnement des interfaces les différent bridges de notre configuration dans cette partie nous parlerons aussi rapidement de la création des bridges et du fichier de configuration des IP statique de la machine ensuite nous parlerons du serveur DHCP, son fonctionnement globale puis sa configuration dans notre cas en particulier,, après dans le iii nous parlerons de la partie DNS, puis pour finir du pare-feu dans le iv.

A – Interfaces physique et bridge

I - Généralités

Le coeur possède donc comme dit précedemment neuf interfaces physiques. Celles-ci ont été mises au sein de deux bridges puisque nous avons deux sous-réseaux (lan 315 et lan 316).

Un bridge, est dans notre cas particulier, un moyen logiciel de fusionner plusieurs interfaces physique au sein d'une même interface logicielle. Cela présente plusieurs avantages notamment celui de pouvoir, dans toutes les configurations des services, traiter l'ensemble de ces interfaces comme une seule interface, nommé d'après le bridge.

Nous avons donc fusionner les interfaces en deux bridges différents car nous possédons deux sous-réseaux. Les noms de ces deux bridges sont **br315** et **br316** et possèdent des IP statique (et non fixe on verra la différence dans la partie DHCP en ii) qui sont **10.61.15.1** et **10.61.16.1**

II – Création d'un nouveau bridge et mise en place de l'adressage IP

Pour créer un nouveau bridge il s'agit de s'assurer d'abord que le paquet *bridge-utils* est installé :

 # dpkg list –installed | grep bridge-utils

Si tel est le cas on peut passer à la suite, sinon il faut d'abord procéder à l'installation[13]. Une fois l'installation faite on créé le bridge et on ajoute les interfaces à l'intérieur :

 # bridge-utils addbr <nom du bridge>

 # bridge-utils addif <nom du bridge> <nom interface 1> <nom interface 2> …

Ensuite il faut modifier le fichier de configuration des interfaces afin de démarrer le bridge au lancement de la machine et de définir une IP, soit dynamiquement, soit de façon statique, le fichier concerné est */etc/network/interfaces* :

 # nano /etc/network/interfaces

Une fois dans le fichier on doit d'abord dire que l'on veut que l'adresse de l'interface (ou des interfaces) physique soit manuel car c'est le bridge qui prend une adresse IP et plus l'interface. Il faut donc ajouter la ligne suivante et ceci pour chaque interface du bridge.

iface <nom de l'interface> inet manual

Ensuite il faut définir le bridge et dire si l'on veut une IP statique ou dynamique :

Pour une IP dynamique :

auto <nom du bridge>

iface <nom du bridge> inet dhcp

bridge_ports <nom de l'interface 1> <nom de l'interface 2> …

Pour de l'adressage statique :

auto <nom du bridge>

iface <nom du bridge> inet static

bridge_ports <nom de l'interface 1> <nom de l'interface 2> …

address <adresse que l'on veut donner en faisant attention aux eventuels conflits>

netmask <le bon netmask correctement calculé>

Il est aussi possible d'ajouter des directives à une interface afin d'effectuer des actions une fois l'interface lancé ou avant son lancement avec les directives post-up et pre-up, cela se fait comme ceci (exemple sur post-up, le cas est le même sur pre-up) :

Pour de l'adressage dynamique :

auto <nom du bridge>

iface <nom du bridge> inet dhcp

bridge_ports <nom de l'interface 1> <nom de l'interface 2> …

post-up <la commande à lancer>

Pour de l'adressage statique :

auto <nom du bridge>

iface <nom du bridge> inet static

bridge_ports <nom de l'interface 1> <nom de l'interface 2> …

address <adresse que l'on veut donner en faisant attention aux eventuels conflits>

netmask <le bon netmask correctement calculé voir ii-c>

post-up <la commande à lancer>

B – DHCP

I - Généralités

Le but premier d'un serveur DHCP (Dynamic Host Control Protocol) est d'assurer un adressage IP de manière dynamique afin d'éviter au client final, simple utilisateur du réseau, d'avoir à choisir et fixer son IP lui-même (adressage dit statique) permettant ainsi d'éviter les conflits IP qui sont toujours un peu ennuyant même si ils sont simples à régler et assez rapide.

Il est possible de faire un adressage dynamique c'est-à-dire donner une adresse IP différente au même client à chaque fois qu'il se reconnecte au réseau ou alors faire un adressage fixe c'est-à-dire fixer l'IP du client dans le fichier de configuration du serveur DHCP pour permettre à partir des adresses MAC[9] de la machine du client de donner toujours la même IP à chaque nouvelle demande.

A titre informatif le netmask (ou masque de sous-réseau) est ce qui permet de définir si deux machines sont sur le même sous-réseau. Cela se fait de la manière suivante :

<adresse de la machine 1>

ou logique <netmask>

<adresse du réseau>

Ensuite on calcule pour la deuxième machine et on compare

Un netmask (tout comme une adresse IP) étant juste un ensemble de bits binaires alors on peut le noter de différentes manières :

- La première consiste à le noter comme une adresse IP par un ensemble de quatre chiffres allant de 0 à 255 séparé par des points
- La deuxième consiste à le noter /<le nombre de bits à 1>

II – Configuration du fichier

1 – Directives générales

Pour obtenir un serveur DHCP fonctionnel il faut d'abord s'assurer que le serveur en question est installé :

 # dpkg list –installed | grep isc-dhcp-server

Si tel est le cas on peut passer à la suite, sinon il faut d'abord procéder à l'installation[10]. Une fois l'installation faite on peut passer à la configuration proprement dite. Cette configuration se fait dans trois fichiers essentiellement :

- */etc/dhcp/dhcpd.conf*
- */etc/default/isc-dhcp-server*
- */var/lib/dhcp/dhcpd.leases*

Le fichier « */etc/dhcp/dhcpd.conf* » se décompose en plusieurs directives qu'il convient de définir ci-après :

- *default-lease-time* : Durée par défaut pendant laquelle est gardé une adresse pour un client dans le cadre d'une future reconnexion
- *max-lease-time* : Durée maximale pendant laquelle est gardé une adresse pour un client dans le cadre d'une future reconnexion
- *server-name :* Nom du serveur pour les hôtes du réseau
- *subnet :* Directive permettant la création d'un nouveau sous-réseau (voir ii-B-2)
- *option :* Directive permettant d'ajouter des options au sein d'un sous-réseau

Le fichier « */etc/default/isc-dhcp-server* » n'a pour utilité que de définir sur quelles interfaces le serveur DHCP va écouter. Cela se fait au sein de la partie INTERFACES de ce fichier comme ceci :

INTERFACES= « <liste des interfaces sur lesquelles on écoute »

Attention : il faut noter que ces interfaces doivent posséder une IP statique dans le réseau pour lesquelles on les déclare et que cette IP doit être attribué avant de lancer le serveur DHCP

Le fichier « */var/lib/dhcp/dhcpd.leases* » contient les associations dynamiques qui ont été effectués dans le passé avec notamment la date de début, de fin, l'adresse IP donné et l'adresse MAC de l'hôte

2 – Création d'un nouveau sous-réseau dans */etc/dhcp/dhcpd.conf*

Afin de créer un nouveau sous-réseau il ajouter un subnet dans le fichier de configuration. Un nouveau sous-réseau ressemble à ceci :

subnet <adresse du réseau> netmask <le bon netmask au foramt 0 à 255>

{

<toutes les options que l'on souhaite mettre>

pool

{

range <la première IP dynamique> <la derniere IP dynamique> ;

}

group

{

}

}

Les options les plus courantes sont :

- *domain-name :* Pour définir le nom de domaine que le client voit affiché dans son fichier de résolution des noms d'hôtes
- *domain-name-server :* IP de la machine servant à faire les résolutions de noms de domaine
- *broadcast-adress :* Pour définir l'adresse de broadcast
- *routers :* Pour définir la gateway du réseau (machine par laquelle passe le flux Internet)

La partie *group* permet d'ajouter un nouvel hôte pour donner une IP fixe en le mettant à l'intérieur de cette manière :

host <nom d'hôte>

{

hardware ethernet <adresse MAC> ;

fixed-address <adresse IP> ;

}

III – Service

Le service associé au serveur DHCP est *isc-dhcp-server.service* qui se contrôle via systemctl sans problème

C – DNS

I – Généralités

Le but premier d'un serveur DNS est d'éviter au client final, simple utilisateur du réseau, d'avoir à se souvenir des adresses IP pour pouvoir accéder à un service sur l'in des serveurs. Cela permet aussi aux administrateurs réseau d'éviter de retenir toutes les IP. Cela permet aussi en cas de changement d'adresse IP pour l'un des services de pas avoir à le notifier car on fait juste le changement dans le serveur et le changement est totalement transparent pour les clients.

Nous utilisons au sein de notre réseau le serveur *bind9*. Il faut donc vérifier que ce paquet est installé. Si il n'est pas installé il faut procéder à son installation[11] avant de continuer.

Il s'agit donc d'un ensemble de fichiers contenant des tas de directives différentes. La liste de ces fichiers est la suivante :

- */etc/bind/named.conf*
- */etc/bind/named.conf.options*
- */etc/bind/named.conf.default-zones*
- */etc/bind/named.conf.local*
- */var/cache/bind/db.ginfo.local*
- */var/cache/bind/db.ginfo.local.inv*

Nous n'expliciterons dans cette partie que les fichiers */etc/bind/named.conf.local*, */var/cache/bind/db.ginfo.local*, */var/cache/bind/db.ginfo.local.inv* . Il est possible de trouver comment configurer les autres fichiers dans le guide d'installation[11]

II – Configuration

i – Fichier */etc/bind/named.conf.local*

Ce fichier permet juste de créer une nouvelle zone DNS c'est-à-dire de créer une nouvelle dans laquelle il est possible de chercher une machine et son IP associé.

Il existe deux types de zones DNS les zones directes et les zones inverses. Les zones directes permettent de retrouver une IP à partir du nom de domaine, c'est là le cas le plus fréquent car quand un client va chercher un service il le fera par son nom de domaine. Les zones inverses elle permettent de retrouver un nom de domaine à partir de son IP cas très utile pour l'administration des systèmes et des réseaux.

Pour créer une nouvelle zone directe :

```
zone « <nom de la zone> »
{
        type master;
        file « <le fichier contenant les associations> »
        forwarders {};
};
```

Pour créer une zone inverse :

```
zone "<Partie fixe dans l'IP du réseau associé>.in-addr.arpa"
{
        type master;
        file "<fichier contenant les associations>";
        forwarders {};
};
```

Attention : Les fichiers contenant les associations ne peuvent être le même car ils sont écrits de manière différente.

ii – Fichiers d'association

Les fichiers d'associations contiennent tous les deux en en-tête la partie suivante :

```
$TTL    3600
@       IN      SOA     <nom d'hote du serveur>.<domaine>     root.<domaine>. (
            <date de la dernière modification>      ;Serial
            3600        ;Refresh [1h]
            600         ;Retry [10m]
            86400       ;Expire [1d]
            600)        ;Negative Cache TTL [1h]
;
@       IN      NS      <nom d'hote du serveur>.<domaine>.
```

a – Le fichier direct

Ajouter un hôte :

<nom d'hote> IN A <adresse IP>

Note : Il est possible d'attribuer plusieurs IP à un même nom d'hote

Créer un alias de nom :

<nom d'hote> IN CNAME <nom d'hote vers lequel on veut pointer>

b – Le fichier inverse

Ajouter un hôte :

<partie de l'IP qui change> IN PTR <hote>.<domaine>

Note : Il est possible d'attribuer plusieurs IP à un même nom d'hote sur un même domaine

C – Service

Le service utilisé pour le serveur DNS choisi est *bind9.service* . Il est contrôlable depuis systemctl

D – Pare-feu

I – Généralités

Le but d'un pare-feu(ou firewall) est de protéger l'accès à Internet des clients et de protéger son réseau des éventuels intrus venant de l'extérieur.

Il existe plusieurs technologies pour les pare-feu les plus répandus étant IPTable (noyau Linux) et pf (noyau BSD).

Notre coeur étant un noyau Linux (Debian), le choix d'IPTables s'est imposé de lui-même. Nous verrons dans le C comment ajouter des règles et le fichier de configuration d'IPTables contenant les règles.

On notera aussi la présence d'une directive

post-up /sbin/iptables-restore < /etc/iptables/iptables.rules

dans le fichier */etc/network/interfaces* après le démarrage de la dernière interface afin de mettre en place les règles au démarrage. Cela aurait aussi pu être fait via un script systemd mais la solution actuelle est la plus simple et la plus rapide à mettre en œuvre.

II – Règles

Afin de créer un pare-feu efficace mais non limitatif au sein du GInfo il a été décidé d'un ensemble de règles assez solides mais relativement souples.

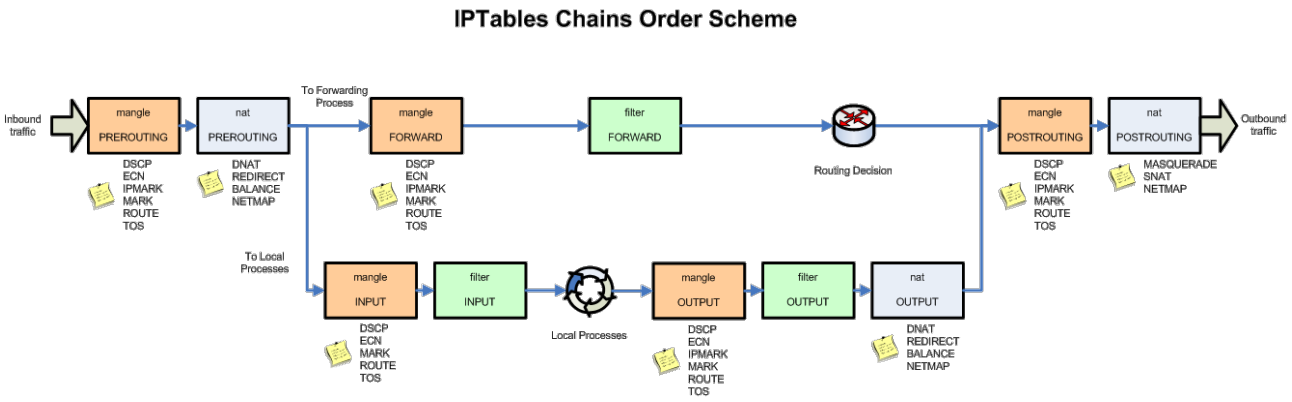
Les règles suivantes ont été décidé :

- Les clients finaux du réseau 316 doivent avoir un accès illimité à Internet
- Le débit total utilisés doit être de 10MBits/s maximum en moyenne avec des pics à 25MBits/s permis sur une durée inférieur à 20s
- Les services doivent être accessibles depuis l'intérieur du local mais pas depuis Internet
- Les serveurs ne sont pas routés vers Internet

III – IPTables

Iptables se compose de quatre tables : filter (pour filtrer), nat(pour faire du routage), mangle(pour modifier des paquets à la volée) et raw(pour marquer les paquets à ne pas vérifier).

Le schéma suivant résume le passage d'un paquet au sein d'IPTables :



Le fichier de configuration d'IPTables n'est par défaut pas créé. Il faut donc créé le dossier puis le fichier :

 # mkdir /etc/iptables

 # touch /etc/iptables/iptables.rules

Pour le moment on le laisse vide car on le remplira par une sauvegarde des règles après le premier essai.

Ajouter une règle à la fin d'une table :

 # iptables -t <ensembles de tables> -A <table> <options> -j <règle>

Ajouter une règle à une position déterminé :

 # iptables -t <ensemble de tables> -I <position> <options> -j <règle>

Supprimer une règle :

 # iptables -t <ensemble de tables>

Vider une table :

 # iptables -t <ensemble de tables> -F <table>

Sauvegarder les règles :

 # iptables-save > /etc/iptables/iptables.rules

Restaurer des règles :

 # iptables-restore < /etc/iptables/iptables.rules

On pourra trouver des informations et des conseils supplémentaires sur le site de LéaLinux[12]

IV – Services

Les services proposés par le GInfo sont de plusieurs types.

Il y a un service Web, un service PXE[14], un service LDAP[15], un service SQL[16], un service Samba[17]

Il ne sera pas explicité dans cette documentation les systèmes et la configuration de ces services chacun d'eux faisant l'objet d'une documentation séparé accessible au format papier au sein du GInfo. Nous ne parlerons donc que des éléments de base du fonctionnement, c'est-à-dire à quoi servent t-ils et les ports utilisés.

A – Web

Un serveur Web est un serveur capable de répondre aux requêtes de type HTTP et HTTPS. Les ports classiques sont le port 80 pour HTTP et 443 pour HTTPS. Au sein du GInfo notre serveur Web ne possède qu'un seul port ouvert le 8080 qui est donné par le CRI.

B – PXE

Un serveur PXE est un serveur permettant de démarrer des machines à travers le réseau sans que la personne ne possède de clé USB ou de CD de boot. Un serveur PXE est donc aussi un serveur TFTP car il doit être capable une fois le boot effectué de fournir le système de démarrage. Le port utilisé pour PXE sont les 4011, 1758, 1759 et le port pour le serveur TFTP est le 69.

C – LDAP

Un serveur LDAP est un serveur d'annuaire permettant aux clients de posséder un compte au sein du réseau afin d'avoir une même session sur toutes les machines du GInfo. Les ports utilisés sont 389 pour LDAP et 636 pour LDAPS.

D – SQL

Un serveur SQL est un serveur capable de faire du stockage et du traitement de données via le langage SQL. Le serveur SQL installé est MySQL. Le port utilisé est 3306.

E – Samba

Un serveur Samba est un serveur capable de faire du stockage et du partage de fichier. Les port utilisés sont 137,138,139 et 445.

Bibliographie

1 : I.Pv4 (https://fr.wikipedia.org/wiki/IPv4)

2 : I.Pv6 (https://fr.wikipedia.org/wiki/Adresse_IPv6)

3 : Bridge (https://fr.wikipedia.org/wiki/Pont_(r%C3%A9seau))

4 : Gateway (https://fr.wikipedia.org/wiki/Passerelle_(informatique))

5 : Dynamic Host Configuration Protcol
(https://fr.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)

6 : Domain Name System (https://fr.wikipedia.org/wiki/Domain_Name_System)

7 : Pare-feu (https://fr.wikipedia.org/wiki/Pare-feu_(informatique))

8 : IPTables (https://fr.wikipedia.org/wiki/Iptables)

9 : Adresse MAC (https://fr.wikipedia.org/wiki/Adresse_MAC)

10 : Installation d’un serveur DHCP (https://wiki.debian.org/fr/DHCP_Server)

11 : Installation d’un serveur DNS (https://wiki.debian.org/fr/Bind9)

12 : Site de LéaLinux sur IPTables (http://lea-linux.org/documentations/Iptables)

13 : Installation des bridges (https://wiki.debian.org/fr/BridgeNetworkConnections)

14 : Preboot-eXecution Environement
(https://fr.wikipedia.org/wiki/Preboot_Execution_Environment)

15 : Lightweight Directory Access Protocol
(https://fr.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol)

16 : Structured Query Language (https://fr.wikipedia.org/wiki/Structured_Query_Language)

17 : Samba (https://fr.wikipedia.org/wiki/Samba_(informatique))