Documentation relative au réseau du GInfo et à son fonctionnement

Pierre Delmas
GInfo Centrale Marseille

20 avril 2017

Table des matières

0.1 Remerciements

Merci à tous ceux qui nous ont soutenu moi-même et Sylvain Bentz lors de la migration et avant pendant la période du maintien du réseau du GInfo.
Merci aussi aux membres de la Direction des Sytèmes d'Information et en particulier à Geoffroy Desvernay pour ses précieux conseils et son temps.

Chapitre 1

Introduction

La présente documentation datée de 20 avril 2017a été écrite dans l'état du réseau connu par l'auteur au 30 Mars 2014.

Cette documentation à pour objectif de présenter de manière concise et rapide le fonctionnement du réseau les différentes parties prenantes (notamment une liste des principaux services disponibles et à venir).

Toutefois bien qu'une explication sommaire des différents fichiers nécessaire au fonctionnement d'un service sera présenté le but n'est pas d'expliquer le fonctionnement de tous les fichiers et la manière de les écrire pour cela on pourra se référer à la bibliographie disponible en fin du présent dossier et dont les références nécessaires au moment nécessaire seront données dans le présent document.

Toutefois il sera pris le temps de faire des explications relativement complète sur le fonctionnement des services et sur leur utilité.

De plus et afin de commencer ce document une explication du protocole IP, le plus utilisé à ce jour dans les systèmes réseau poru la communication entre machines et celui qui est actuellement en place sur le réseau du GInfo (ci-après dénommé réseau) sera faite, ainsi qu'une présentation sommaire des différentes couches du modèle OSI, aujourd'hui le modèle d'encapsulation et de transfert de données établi comme norme, ainsi qu'une courte section sur l'adressage MAC.

Pour conclure il est rappeler aux personnes voulant s'intéresser au réseau que pratiquement toutes les documentations et les écrits, tutoriels et autres articles disponibles de qualité sont écrits en langue anglaise et que tous les fichiers de configuration sont rédigés dans cette langue. Il peut donc être nécessaire, voir très utile d'avoir une connaissance correcte voir avancé de cette langue.

Chapitre 2

Protocole IP, adresse MAC et modèle OSI

2.1 Protocole IP

Aujourd'hui le protocole le plus répandu est IP [13] (Internet Protocol). Ce protocole repose sur un système d'adresses communément appelé adresse IP. Cette adresse est composé d'un ensemble de 32 bits classiquement mis sous forme décimale par paquet de 8 bits (octet) avec donc des valeurs comprises entre 0 et 255 séparées par un point. Cette adresse est unique pour chaque machine d'un même réseau.

De plus on associe à cette adresse deux autre valeurs codées elles aussi sur 32 bits et donc mises en base décimale par paquet de 8 bits avec des valeurs comprises entre 0 et 255 séparées par des points. Ces deux valeurs sont le masque de sous-réseau et la gateway. Le masque [19] et la gateway [12] sont identiques pour chaque machine d'un même réseau. Le masque permet aux machines de vérifier que le message qui leur parvient vient bien du même réseau que le leur (par une opération binaire de type XOR) et la gateway correspond à l'adresse de la machine a contacter pour pouvoir communiquer avec le monde extérieur.
Le masque est donc par défaut un ensemble de 1 suivi par un certain de nombre de zéros, il est strictement impossible et me interdit de mettre un 1 après un zéro. Il est ici à noté que l'on peut aussi marqué le masque par {nombre de bits à un 1 dans le masque}
Le terme gateway peut donc à la fois définir une machine physique qui présente la propriété particulière d'avoir deux cartes réseau branchés sur deux réseaux différents et une adresse au sein d'un réseau.
On notera de plus que deux adresses sur le réseau sont interdites d'utilisation, la première qui correspond à l'adresse du réseau lui-même, il s'agit de la première adresse finissant par 0 (adresse IP quelconque XOR masque du sous-réseau = première adresse du réseau) et la dernière adresse du réseau qui correspond à l'adresse de broadcast qui permet à une machine ne connaissant pas encore l'adresse de son destinaire d'envoyer le message à tout le monde sur le réseau les machines se chargeant de décider si le message qui est reçu leur est destiné ou pas à l'aide d'un en-tête toujours présent sur les paquet transmis contenant l'adresse du destinataire et de la source.

2.2 Adressage MAC

L'adressage MAC [15] (Media Access Control) est une adresse unique à chaque carte réseau au monde. Elle est composé de 24 octets, généralement regroupé sous la forme de 12 chiffres hexa-décimaux par paquet de deux séparés par un séparateur généralement des deux-points ou des tirets. Afin d'assurer l'unicité de chaque carte malgré le nombre de fabricant les 12 premiers octets sont attribués pour un fabricant et les douzes derniers octets sont choisi par le constructeur suivant leurs propres algorithmes.

2.3 Modèle OSI

Le modèle OSI [16] (Open Systems Interconnection) se base sur 7 couches 4 couches physiques et 3 logiciels. Lors d'une communication toutes les couches ne sont pas utilisés. Le protocole IP évoqué dans la section supérieure relève de la couche 4. L'adressage MAC relève lui de la couche 3.

Chapitre 3

Organistaion du réseau

3.1 Aspects légaux

Il existe dans le réseau du GInfo (ci-après dénommé réseau ou GInfo) un seul sous-réseau sur le réseau 10.61.16.0/24 (nommé ci-après lan316).

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.

3.2 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 un même sous-réseau par le biais d'un bridge [9].

3.2.1 Généralités

Le coeur possède donc comme dit précedemment neuf interfaces physiques. Celles-ci ont été mises au sein d'un même bridge puisque nous avons deux sous-réseaux (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. Le nom de ce bridge est br316 et possède une IP statique (et non fixe on verra la différence dans la partie DHCP) qui est 10.61.16.1

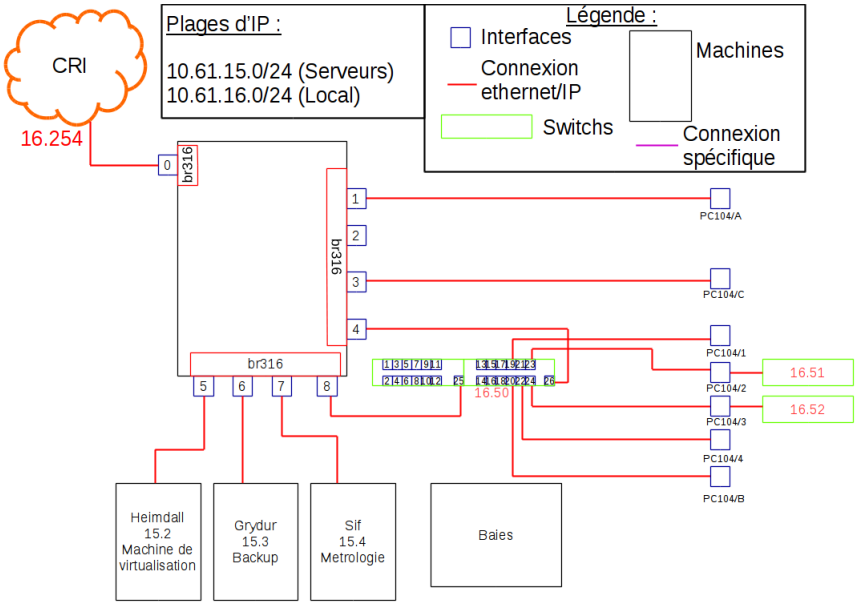

FIGURE 3.1 – Branchements

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

Pour créer un nouveau bridge[1] 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. 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} ...

addresse {*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*}
post-up {*la commande à lancer*}

Chapitre 4

Service DHCP

4.1 Généralités

Le but premier d'un serveur DHCP[10] (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 de la machine du client de donner toujours la même IP à chaque nouvelle demande.

4.2 Configuration

4.2.1 Directives générales

Pour obtenir un serveur DHCP[2] 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. 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
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

4.2.2 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 pour sortir du réseau)

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'hote*}
{
        hardware ethernet {*adresse MAC*} ;
        fixed-address {*adresse IP*} ;
}

4.2.3 Service

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

Chapitre 5

Service DNS

5.1 Généralités

Le but premier d'un serveur DNS[11, 3] 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'un 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 DHCP et le serveur DNS 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 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 en ligne de Debian.

5.2 Configuration

5.2.1 Fichier /etc/bind/named.conf.local

Ce fichier permet juste de créer une nouvelle zone DNS c'est-à-dire de créer une nouvelle zone 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 elles 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.

5.2.2 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 avec l'heure et la minute} ;Serial
3600 ;Refresh [1h]
600 ;Retry [10m]
86400 ;Expire [1d]
600) ;Negative Cache TTL [1h]
;
@ IN NS {nom d'hote du serveur}.{domaine}.

Le fichier direct

Ajouter un hote :
{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}

Le fichier inverse

Ajouter un hote :
{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

5.3 Service

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

Chapitre 6

Services sur d'autres serveurs

Les services proposés par le GInfo sont de plusieurs types. Il y a un service Web, un service PXE, un service LDAP, un service SQL, un service Samba 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.

6.1 Web

Un serveur Web[20, 4] 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. Nous distribuons donc HTTPS par ceux ports

6.2 PXE

Un serveur PXE[6, 7] 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. Les port utilisés pour PXE sont les 4011, 1758, 1759 et le port pour le serveur TFTP est le 69.

6.3 LDAP

Un serveur LDAP[14, 5] 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.

6.4 SQL

Un serveur SQL[18, 4] 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.

6.5 Samba

Un serveur Samba [17, 8] est un serveur capable de faire du stockage et du partage de fichier. Les port utilisés sont 137,138,139 et 445.

Bibliographie

[1] Debian Community. Debian bridge page. https ://wiki.debian.org/fr/BridgeNetworkConnections.

[2] Debian Community. Debian dhcp page. https ://wiki.debian.org/fr/DHCP_Server.

[3] Debian Community. Debian dns page. https ://wiki.debian.org/fr/Bind9.

[4] Debian Community. Debian lamp page. https ://wiki.debian.org/fr/Lamp.

[5] Debian Community. Debian ldap page. https ://wiki.debian.org/LDAP/OpenLDAPSetup.

[6] Debian Community. Debian pxe page. https ://en.wikipedia.org/wiki/Preboot_Execution_Environment.

[7] Debian Community. Debian pxe page. https ://wiki.debian.org/PXEBootInstall#Set_up_TFTP_server.

[8] Debian Community. Debian samba page. https ://wiki.debian.org/SambaServerSimple.

[9] Wikipedia Community. Wikipedia bridge page. https ://en.wikipedia.org/wiki/Bridging_(networking).

[10] Wikipedia Community. Wikipedia dhcp page. https ://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Proto

[11] Wikipedia Community. Wikipedia dns page. https ://en.wikipedia.org/wiki/Domain_Name_System.

[12] Wikipedia Community. Wikipedia gateway page. https ://en.wikipedia.org/wiki/Gateway_(telecommunications).

[13] Wikipedia Community. Wikipedia ip page. https ://en.wikipedia.org/wiki/Internet_Protocol.

[14] Wikipedia Community. Wikipedia ldap page. https ://en.wikipedia.org/wiki/Lightweight_Directory_Access_Proto

[15] Wikipedia Community. Wikipedia mac page. https ://en.wikipedia.org/wiki/MAC_address.

[16] Wikipedia Community. Wikipedia osi page. https ://en.wikipedia.org/wiki/OSI_model.

[17] Wikipedia Community. Wikipedia samba page. https ://en.wikipedia.org/wiki/Samba_(software).

[18] Wikipedia Community. Wikipedia sql page. https ://en.wikipedia.org/wiki/SQL.

[19] Wikipedia Community. Wikipedia subnet mask page. https ://en.wikipedia.org/wiki/IPv4_subnetting_reference.

[20] Wikipedia Community. Wikipedia webserver page. https ://en.wikipedia.org/wiki/Web_server.