

# Rapport système et réseaux II

Perion Maxence, Pinon Alexandre

## Contents

<b>1</b>	<b>Installation d'une distribution GNU/Linux sans interface graphique</b>	<b>3</b>
1.1	Installation via le réseau . . . . .	3
<b>2</b>	<b>Paramétrage du réseau</b>	<b>4</b>
2.1	Répartition des adresse IP (réseau Interconnexion et réseau privé)	4
2.2	Interfaces réseau du routeur . . . . .	5
2.3	Interfaces réseau du client . . . . .	6
2.4	Communication sans table de routage . . . . .	6
2.5	Mise en place de la table de routage . . . . .	7
2.6	Règles iptables pour laisser accès a internet a une machine . . . . .	7
<b>3</b>	<b>Service DHCP</b>	<b>8</b>
3.1	Mise en place du DHCP . . . . .	8
3.2	Fichier de log pour le DHCP . . . . .	8
<b>4</b>	<b>Sauvegarde automatique</b>	<b>8</b>
4.1	Rsync . . . . .	8
4.2	Cron . . . . .	8
<b>5</b>	<b>Manipulation paquets de Debian</b>	<b>8</b>
5.1	Mise à jour du système . . . . .	8
5.2	Les paquets . . . . .	8
<b>6</b>	<b>Interrogation d'un serveur DNS</b>	<b>8</b>
6.1	Les commandes host, dig et nslookup . . . . .	8
6.2	Fichier de renseignement du serveur DNS . . . . .	8
6.3	Rôle du fichier /etc/hosts . . . . .	8
<b>7</b>	<b>Installation d'un DNS</b>	<b>8</b>
7.1	Mise en place du DNS . . . . .	8
7.2	Test réaliser afin de valider le DNS . . . . .	8
<b>8</b>	<b>Installation d'un serveur LAMP</b>	<b>9</b>
8.1	Installation et mise en place d'Apache . . . . .	9
8.2	Installation et mise en place de PostgreSQL . . . . .	9
8.3	Installation et mise en place de PHP . . . . .	9
8.4	Installation et mise en place de MySQL . . . . .	9
8.5	Installation et mise en place d'un PDO . . . . .	9

<b>9</b>	<b>Script de routage et matrice de filtrage</b>	<b>9</b>
9.1	Scripts de routage au démarrage . . . . .	9
9.2	Règles iptables de la matrice de filtrage . . . . .	9
<b>10</b>		<b>9</b>
10.1	. . . . .	9
10.2	. . . . .	9

Introduction : général

Introduction tp1 : dans un premier temps nous avons installation de Debian GNU/Linux via le réseaux pour avoir tout simplement un outil de travail, vient ensuite la répartition des adresses IP pour chaque groupe. Ensuite nous allons aborder le paramétrage des interfaces réseaux sur le routeur puis le client. L'étape suivante et évidemment la création de route afin de faire communiquer le client et le routeur. Vient ensuite la mise en place d'un service DHCP. Sauvegarde automatique avec rsync. Et finalement un approfondissement de l'utilisation des paquets sous Debian. Afin de mettre en place toutes cette architecture nous sommes en possession d'un serveur faisant office de routeur et d'un client.

## 1 Installation d'une distribution GNU/Linux sans interface graphique

### 1.1 Installation via le réseau

La première étape pour commencer à travailler a tout simplement était d'installer une distribution Debian GNU/Linux sans interface graphique. En effet cette distribution est un système d'exploitation et va nous permettre pour la suite des manipulations d'avoir une base de travail. Afin d'installer cette distribution les administrateurs système et réseau nous ont mis à disposition un boot via le réseau. En temps normal pour installer un système d'exploitation qu'elle qu'il soit, les utilisateurs lambda utilisent des clés USB ou disque dur qui permette de boot sur celle-ci. Le principe est le même sauf que cette installation se fait par le réseau IEM qui est le réseau de l'université de Bourgogne. Pour lancer l'installation nous devons brancher notre serveur sur le réseau et la démarrer depuis le bios en sélectionnant le bouton "PXE IEM". Une fois l'installation démarrée nous devons suivre les étapes d'installation, tel que le nom et mot de passe du root, la choix des partition du disque, le nom de domaine (sartre.iem)... Le disque est composé de 3 partitions, il est possible d'afficher les partitions et des détails supplémentaire avec la commande :

```
$ sudo fdisk -l
```

```

root@routerGroupeA2PP:/etc/network# root@routerGroupeA2PP:/etc/network# sudo fdisk -l
Disque /dev/sda : 238,47 GiB, 256060514304 octets, 500118192 secteurs
Modèle de disque : SAMSUNG SSD SM84
Unités : secteur de 1 x 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x82be967b

Périphérique Amorçage      Début          Fin    Secteurs Taille Id Type
/dev/sda1      *              2048 498116607 498114560 237,5G 83 Linux
/dev/sda2              498118654 500117503   1998850   976M  5 Étendue
/dev/sda5              498118656 500117503   1998848   976M  82 partition d'échange Linux / Solaris
root@routerGroupeA2PP:/etc/network#

```

### Les 3 partitions d'un disque.

Ici nous pouvons voir distinctement les différentes partition. Dont la partition général de 237.5 Go qui permet de stocker différentes chose tel que des paquets pour l'utilisation de Debian ou encore différents types de fichiers. De plus nous pouvons voir la partition qui permet l'échange entre Linux et Solaris, mais également un partition étendue qui est un conteneur de partition logiques. Maintenant que Debian est installé sur notre routeur il est prêt à être configuré et utilisé pour différentes taches.

## 2 Paramétrage du réseau

Avant tout paramétrage il est important de préciser que nous avons branché le réseau IEM sur la carte réseau intégré à la carte mère sur routeur. La première carte réseau externe à était branchée au switch afin de permettre la communication avec les autres groupes. Dernièrement la deuxième carte réseau externe est branchée sur le client. Ici le but de la manipulation est de mettre en place un réseau d'entreprises et d'une infrastructure de services.

### 2.1 Répartition des adresse IP (réseau Interconnexion et réseau privé)

Chaque groupe de TP possède une IP et un masque de classe C pour le réseau d'intercommunication. Afin de mettre en place le réseau d'intercommunication nous avons pris une plage IP compris entre 192.168.1.1 et 192.168.1.5 avec ce masque 255.255.255.248, 248 car nous avons besoin de 3 bits pour définir l'ensemble des routeurs qui sont dans le sous-réseau. En effet nous voulions distribuer les IP pour 6 routeurs, 5 pour les groupes de TP et 1 pour le réseau d'intercommunication.

Groupe	Réseau IEM	Réseau Intercommunication	Réseau privé
1	172.31.20.111 255.255.255.0	192.168.1.1 255.255.255.248	10.1.1.0 255.255.255.0
2	172.31.20.112 255.255.255.0	192.168.1.2 255.255.255.248	10.1.2.0 255.255.255.0
3	172.31.20.113 255.255.255.0	192.168.1.3 255.255.255.248	10.1.3.0 255.255.255.0
4	172.31.20.114 255.255.255.0	192.168.1.4 255.255.255.248	10.1.4.0 255.255.255.0
5	172.31.30.115 255.255.255.0	192.168.1.5 255.255.255.248	10.1.5.0 255.255.255.0

Tableau des adresse IP de chaque groupe.

Ce petit tableau récapitule les adresses IP allouer pour chaque réseau et pour chaque groupe.

## 2.2 Interfaces réseau du routeur

Nous allons maintenant expliquer comment nous avons paramétré les interfaces réseau sur le routeur. Pour cela il suffit simplement de se rendre dans le fichier `/etc/network/interfaces` et de modifier le fichier comme ceci :

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eno1
iface eno1 inet static
    address 172.31.20.112
    netmask 255.255.255.0
    network 172.31.20.1
    broadcast 172.31.20.255
    gateway 172.31.20.1
    dns-nameservers 172.31.21.35 193.50.50.6
pre-up /etc/firewall-free.sh

#reseau prive
auto enp3s0
iface enp3s0 inet static
    address 10.1.2.1
    netmask 255.255.255.0
    network 10.1.2.0

#interconnexion
auto enp1s0
iface enp1s0 inet static
    address 192.168.1.2
    netmask 255.255.255.248
    network 192.168.1.0
```

Fichier de configuration des interfaces réseaux du routeur.

Il est également possible d'afficher les interfaces réseaux dans le shell avec la commande :

```
$ ip a
```

ce qui a pour intérêt de connaître rapidement les macs adresse et/ou IP de chacun, mais permet également de savoir si une interface est en route "UP" ou pas "DOWN". Pour le fichier de configuration ci-dessus nous pouvons voir les 3 interfaces donc les 3 port Ethernet que possède notre routeur, ainsi que l'interface

"lo" qui représente le loopback. Pour chacune d'entre elle leurs adresses (address), leurs masques (netmask) et leurs réseaux (network) sont indiqué. Pour l'interface "eno1" qui est relié au réseau IEM, nous avons plusieurs information supplémentaire ; l'adresse de broadcast ainsi que la passerelle (gateway). Les informations "dns-nameservers" et "pre-up" seront expliqué ultérieurement. REDEMARAGE !

## 2.3 Interfaces réseau du client

Nous allons maintenant expliquer comment nous avons paramétré les interfaces réseau sur le client. Comme pour le routeur le fichier de configuration se trouve dans /etc/network/interfaces. Par conséquent la forme du fichier est similaire a celui du routeur. Voici les informations que nous retrouvons sur le fichier et qui renseigne l'adresse, le masque et le réseau de l'interface réseau de notre client.

```
auto eno1
iface eno1 inet static
    address 172.31.20.112/24
    netmask 255.255.255.0
    network 172.31.20.1
```

## 2.4 Communication sans table de routage

Par défaut le client et le routeur peuvent communiquer car il sont branché physiquement entre eux. Il en est de même pour tous les routeurs qui sont brancher sur le réseau IEM. Afin de tester les communication entre différentes machines, que se soit routeurs ou clients, il nous suffit simplement de ping la machine avec sont IP et avec cette commande :

```
$ ping ip_machine
```

Par exemple avec notre routeur nous allons ping le routeur d'un groupe d'amis via le réseau IEM :

```
root@routerGroupeA2PP:~# ping 172.31.20.111
PING 172.31.20.111 (172.31.20.111) 56(84) bytes of data.
64 bytes from 172.31.20.111: icmp_seq=1 ttl=64 time=1.94 ms
64 bytes from 172.31.20.111: icmp_seq=2 ttl=64 time=0.689 ms
64 bytes from 172.31.20.111: icmp_seq=3 ttl=64 time=0.721 ms
64 bytes from 172.31.20.111: icmp_seq=4 ttl=64 time=0.686 ms
64 bytes from 172.31.20.111: icmp_seq=5 ttl=64 time=0.734 ms
64 bytes from 172.31.20.111: icmp_seq=6 ttl=64 time=0.703 ms
^C
--- 172.31.20.111 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5095ms
rtt min/avg/max/mdev = 0.686/0.912/1.941/0.460 ms
```

Ping de notre routeur à un autre.

Nous pouvons voir ici que les paquets sont bien envoyé et reçu par notre correspondant. Donc pas de soucis pour ping un autre routeur. Mais si notre routeur ou notre client veut ping ou plus couramment communiquer avec un client d'un autre groupe qui est donc sur un réseau privé, on va devoir lui indiquer le chemin pour aller jusqu'à celui-ci.

## 2.5 Mise en place de la table de routage

Dans la configuration actuelle il nous est impossible de ping le client qui passe par un autre routeur que le notre, en effet pour arriver a faire ceci il nous faut mettre en place une table de routage. La table de routage permet d'indiquer les routes a prendre par tel ou tel machines.

### Route de notre routeur.

Il est également possible d'afficher les routes dans le shell avec la commande :

```
$ ip route
```

L'image nous dit que la route par défaut a emprunter si aucune des routes décrite ne convien est `ip 0...`

## 2.6 Règles iptables pour laisser accès a internet a une machine

L'architecture de notre réseau commence a prendre forme, mais il est maintenant temps de permettre au client de notre réseau privé d'avoir accès a internet. Car oui pour le moment seul le routeur est en possession d'une connexion internet car il est encore une fois connecté au réseau IEM. Pour réaliser cette manoeuvre appelé une translation d'adresse, nous allons utilisé les règles "iptables" :

```
$ iptables -t nat -A POSTROUTING -s 10.1.2.0/24 -o eno1 -j MASQUERADE
```

Cette commande permet au routeur de filtrer et donc de laisser passer les clients qui arrivent du sous réseau 10.1.2.0/24.

Afin de savoir si notre routeur était capable d'utiliser internet, nous avons installer "links2" et avec la commande :

```
$ links2 google.com
```

il nous étaient possible de surfer sur le web avec un affichage non graphique.

## **3 Service DHCP**

### **3.1 Mise en place du DHCP**

### **3.2 Fichier de log pour le DHCP**

## **4 Sauvegarde automatique**

### **4.1 Rsync**

### **4.2 Cron**

## **5 Manipulation paquets de Debian**

### **5.1 Mise à jour du système**

### **5.2 Les paquets**

Conclusion tp 1

Introduction tp 1bis

## **6 Interrogation d'un serveur DNS**

### **6.1 Les commandes host, dig et nslookup**

### **6.2 Fichier de renseignement du serveur DNS**

### **6.3 Rôle du fichier /etc/hosts**

## **7 Installation d'un DNS**

Pour la suite des explication il est important de préciser que nous avons choisie comme nom de domaine : agence.altantide.

### **7.1 Mise en place du DNS**

Création des deux zones, (named.conf.local et named.conf.default-zone), copie du fichier "db.local" en "db.agence.atlantide", copie de "db.127" en "db.agence.atlantide.inv" pour la zone de recherche inversé. Ajout des forwarders dans "named.conf.options". Redémarrage avec systemctl restart bind 9. Changement du dhcp pour prendre en compte le serveur DNS.

### **7.2 Test réaliser afin de valider le DNS**

named-checkzone et named-checkconf pour vérifier la syntaxe /var/log/syslog pour voir les erreurs si présente au démarrage depuis le serveur : nslookup ip-client depuis le serveur : nslookup nom-client depuis un client connecté en filaire au routeur : nslookup ip-client depuis un client connecté en filaire au routeur : nslookup nom-client vers le serveur d'une autre agence : vers le client d'une autre agence :

Conclusion tp 1bis



Introduction tp 2

## **8 Installation d'un serveur LAMP**

### **8.1 Installation et mise en place d'Apache**

Test, localisation de la page web, création du compte utilisateur développeur, modification du serveur pour que www soit associé au dev web (créé dans /srv), les bons droits, mise en place du virtual hosts pointant sur /srv/www

### **8.2 Installation et mise en place de PostgreSQL**

Déroulement...

### **8.3 Installation et mise en place de PHP**

Programme PHP qui se connecte à PostgreSQL et affiche des infos pour la vérification.

### **8.4 Installation et mise en place de MySQL**

Comme pour PostgreSQL et php

### **8.5 Installation et mise en place d'un PDO**

## **9 Script de routage et matrice de filtrage**

### **9.1 Scripts de routage au démarrage**

### **9.2 Règles iptables de la matrice de filtrage**

Conclusion tp 2

Introduction 3

## **10**

### **10.1**

### **10.2**

Conclusion tp 3