

Rapport Systèmes et Réseaux II

Perion Maxence, Pinon Alexandre

Sommaire

1	Introduction	3
2	Installation d'une distribution GNU/Linux sans interface graphique	3
2.1	Installation via le réseau	3
3	Paramétrage du réseau	4
3.1	Répartition des adresse IP (réseau Interconnexion et réseau privé)	4
3.2	Interfaces réseau du routeur	5
3.3	Interfaces réseau du client	6
3.4	Communication sans table de routage	7
3.5	Mise en place de la table de routage	7
3.6	Règles iptables pour laisser l'accès a internet a une machine . . .	9
4	Service DHCP	9
4.1	Mise en place du DHCP	9
4.2	Système de log pour le DHCP	11
5	Sauvegarde automatique	11
5.1	Rsync	11
5.2	Cron	11
6	Manipulation paquets de Debian	11
6.1	Mise à jour du système	11
6.2	Les paquets	11
7	Interrogation d'un serveur DNS	11
7.1	Les commandes host, dig et nslookup	11
7.2	Fichier de renseignement du serveur DNS	11
7.3	Rôle du fichier /etc/hosts	11
8	Installation d'un DNS	11
8.1	Mise en place du DNS	11
8.2	Test réaliser afin de valider le DNS	11
9	Installation d'un serveur LAMP	12
9.1	Installation et mise en place d'Apache	12
9.2	Installation et mise en place de PostgreSQL	12
9.3	Installation et mise en place de PHP	12
9.4	Installation et mise en place de MySQL	12
9.5	Installation et mise en place d'un PDO	12

10	Script de routage et matrice de filtrage	12
10.1	Scripts de routage au démarrage	12
10.2	Règles iptables de la matrice de filtrage	12
11		12
11.1	12
11.2	12

1 Introduction

L'objectif de ces Travaux Pratiques a été de réaliser un serveur complet pour une agence fictive. Pour cela, nous étions en possession de deux machines: un serveur/routeur ainsi qu'un client, une machine simulant la connexion d'un appareil au réseau de l'agence. Nous sommes partis de zéro et dans un premier temps nous avons installé Debian sur notre serveur, une distribution GNU/Linux, sans interface graphique via le réseau IEM pour avoir un outil de travail. Premièrement, nous avons défini et paramétré notre adressage et routage, c'est à dire la façon d'attribuer les adresses IP et comment communiquer sur les différents réseaux. Afin de s'affranchir des adresses IP et pour s'approcher d'un cadre plus réaliste, nous avons deuxièmement mis en place sur le serveur un service de DNS, permettant d'utiliser des noms de domaines agissant comme des alias pour des adresses IP. Troisièmement, nous avons mis en place ce qui pourrait servir pour héberger un site web pour notre agence grâce à une suite d'outils appelée un LAMP. Pour finir, nous avons mis en place un service d'authentification centralisé LDAP ainsi qu'un service de partage de fichiers Samba.

2 Installation d'une distribution GNU/Linux sans interface graphique

2.1 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 on mit 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 tâches.

3 Paramétrage du réseau

Nous avons ensuite définis la répartition des adresses IP pour chaque groupe (chaque agence) et paramétré en conséquences les différentes interfaces réseaux sur le routeur puis le client. 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.

3.1 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'interconnexion. Afin de mettre en place le réseau d'interconnexion 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'interconnexion.

Groupe	Réseau IEM	Réseau Interconnexion	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 adresses IP de chaque groupe.

Ce petit tableau récapitule les adresses IP allouer pour chaque réseau et pour chaque groupe. Pour notre cas nous sommes dans le groupe 2 ce qui veut dire que notre adresse sur le réseau d'interconnexion est le "192.168.1.2" et notre adresse de réseau privé est le "10.1.2.0".

3.2 Interfaces réseau du routeur

Les adresses que nous venons de répartir entre chaque groupe doivent être utiliser par le routeur. C'est pour cela que 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". En effet durant les phases de test il était important de savoir qu'elle interfaces était allumée ou éteinte, il nous est arrivé de relancer une interface à la main via la commande

```
$ ip link set enp3s0 up
```

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 indiqués. Pour l'interface "eno1" qui est relié au réseau IEM, nous avons plusieurs informations supplémentaires ; l'adresse de broadcast ainsi que la passerelle (gateway). De plus nous pouvons maintenant différencier clairement qu'elle interface fait référence à qu'elle utilisation. En effet l'interface "eno1" est relié au réseau IEM car pour le réseau nous avons indiqué l'adresse "172.31.20.1" qui est l'adresse du routeur du réseau IEM fournie par les intervenants de travaux pratiques. Le raisonnement est le même pour les interfaces "enp3s0" et "enp1s0", pour lesquels nous avons indiqué respectivement pour le réseau "10.1.2.0" et "192.158.1.0". Et qui sont donc les réseaux d'interconnexion et privé. Les informations "dns-nameservers" et "pre-up" seront expliqués ultérieurement. Le fichier de configuration des interfaces réseaux est actuellement écrit et prêt à l'emploi, mais ce n'est pas pour autant que les interfaces vont changer. En effet pour que les modifications soient prises en compte il faut redémarrer le service associé en tant que root ou alors avec sudo:

```
$ service networking restart
ou
$ /etc/init.d/networking restart
ou
$ systemctl restart networking
```

Ces 3 commandes permettent de redémarrer le service qui s'occupe des interfaces réseaux.

3.3 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 à celui du routeur. Voici les informations que nous retrouvons sur le fichier et qui renseignent l'adresse, le masque et le réseau de l'interface réseau de notre client.

```
auto enp11s0
iface enp11s0 inet static
    address 10.1.2.2
    netmask 255.255.255.0
    network 10.1.2.0
    gateway 10.1.2.1
```

On voit ici que l'IP renseigné pour le réseau donc le network pour l'interface "enp11s0" est "10.1.2.1", qui est l'adresse du de notre sous réseau. On connaît

désormais l'IP de notre client qui est "10.1.2.2". Comme pour le routeur afin d'appliquer les changements il faut redémarrer le service avec l'une des trois commande donnée pour le routeur. Par la suite toutes ces configuration vont permettre aux client et routeur de communiquer avec le reste des groupes.

3.4 Communication sans table de routage

Nous allons maintenant aborder la communication entre différente machine, car en effet nous avons pour but de simuler un réseau en entreprise, et par conséquent il faut que nos machines puisse communiquer, et que le routeur sois en capacité de rediriger correctement les machines ou elle souhaite aller. 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 car ils sont tous sur le même réseau. 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.

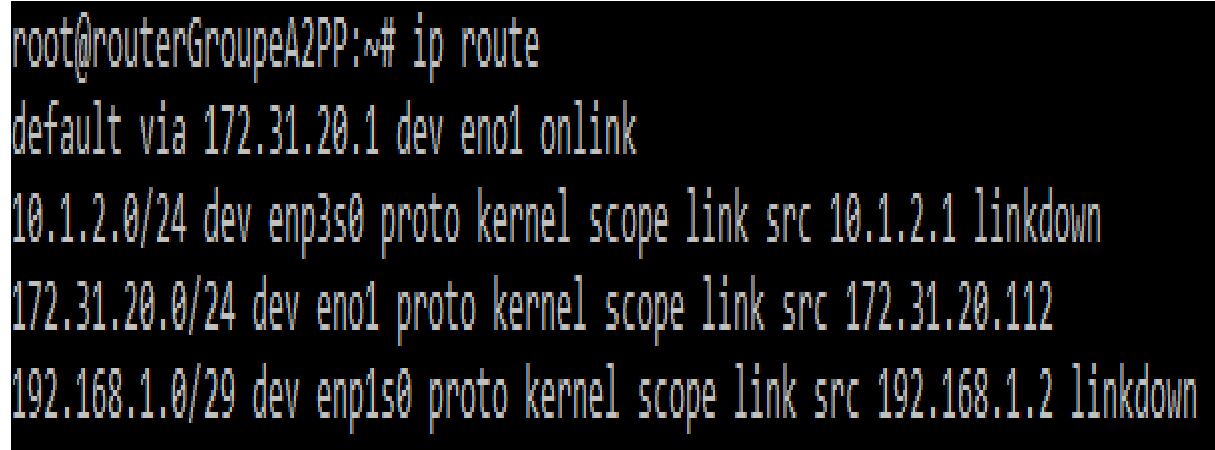
3.5 Mise en place de la table de routage

L'étape suivante est la création de routes afin de faire communiquer les différentes machines de la simulation: le client et le routeur, mais également entre chaque routeurs (les autres agences) et entre différents clients de différentes agences. 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.

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

```
$ ip route
```



```
root@routerGroupeA2PP:~# ip route
default via 172.31.20.1 dev eno1 onlink
10.1.2.0/24 dev enp3s0 proto kernel scope link src 10.1.2.1 linkdown
172.31.20.0/24 dev eno1 proto kernel scope link src 172.31.20.112
192.168.1.0/29 dev enp1s0 proto kernel scope link src 192.168.1.2 linkdown
```

Route de notre routeur.

L'image nous dit que la route par défaut à emprunter pour toutes les machines qui arrive, si aucune des routes décrite ne conviens est la route par l'adresse "172.31.20.1". Ce qui redirige les machines au routeur du réseau IEM. Elle permet également de savoir que la machine avec l'IP "10.1.2.1" est redirigé sur le sous réseau "10.1.2.0/24", de même pour l'adresse "172.31.20.112" qui est redirigé sur le sous réseau "172.31.20.0/24", et "192.168.1.2" redirigé sur "192.168.1.0/29" qui est donc le réseau d'interconnexion.

Les routes statiques sont ajoutés via la commande en tant que root ou utiliser sudo dans le cas contraire:

```
$ ip route add {network} via {IP}
```

Maintenant nous cherchons le moyen de permettre la communication entre notre routeur et le client d'un autre groupe d'amis. Et dans un second temps permettre la communication entre notre client et le client de l'autre groupe. Le groupe avec qui nous effectuons les tests, on pour réseau d'interconnexion l'IP "192.168.1.1", pour réseau privé "10.1.1.0" et l'IP de leur client et la suivante "10.1.1.1".

Donc sur notre routeur il faut appliquer la commande:

```
$ ip route add 10.1.1.0/24 via 192.168.1.1
```

Cette commande permet de dire que tous ce qui arrive sur le routeur "192.168.1.1" donc le routeur de notre groupe de test, est redirigé sur le réseau privé "10.1.1.0". Donc si avec notre routeur nous tapons la commande qui permet de ping leur client:

```
$ ping 10.1.1.1
```


est bien ceci fonctionne, en effet nos paquets sont belle est bien envoyé et réceptionné.

Et dans un second temps sur notre client appliquer la commande:

```
$ ip route add 10.1.1.0/24 via 10.1.2.1
```

Cette commande permet de dire que tous ce qui part de notre client "10.1.2.1", est redirigé sur le réseau privé "10.1.1.0" de notre groupe d'amis. Donc si avec notre client nous tapons la commande qui permet de ping leur client:

```
$ ping 10.1.1.1
```

est bien ceci fonctionne. Tous comme notre routeur nos paquets sont belle est bien envoyé et réceptionné.

3.6 Règles iptables pour laisser l'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 d'accepter les paquets retransmis via le périphérique d'IP interne du pare-feu par l'interface "eno1", et donc de laisser passer les données qui sont destinée au clients qui sont dans le sous réseau "10.1.2.0/24".

Afin de vérifier si notre routeur était capable d'utiliser internet, nous avons installer "links2" qui a était auparavant installé via le gestionnaire de paquet "apt" et avec la commande:

```
$ links2 google.com
```

Et en effet notre routeur était capable de surfer sur le web avec un affichage non graphique. Dans ce cas nous avons appliquer la même procedure pour notre client afin de vérifier si il possédais une connexion internet.

4 Service DHCP

Maintenant partons du principe que nous souhaitons posséder plusieurs clients sur notre réseau privé, il est alors important d'approfondir le paramétrage de notre réseau avec la mise en place d'un service DHCP afin de définir les adresses IP à attribuer pour les machines (clients) se connectant sur notre réseau privé.

4.1 Mise en place du DHCP

Avant toute chose, nous avons cherché et installé le bon paquet pour l'utilisation du DHCP:

```
$ apt-get install isc-dhcp-server
```

Une fois l'installation effectuée nous nous sommes rendu dans le fichier "/etc/default/isc-dhcp-server" qui a été installé grâce à la commande donnée précédemment. Dans ce fichier il faut renseigner sur quelle interface réseau nous voulons travailler pour la suite de la mise du DHCP.

```
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp3s0"
INTERFACESv6=""
```

Interface du DHCP.

Comme nous voulons allouer automatiquement les adresses IP de chaque client connecté à notre réseau privé nous avons utilisé l'interface "enp3s0". Ceci étant fait nous voulons paramétrer en détails le serveur DHCP, comme par exemple la plage d'adresse IP qui va être subnet, ou encore des adresses fixes pour des postes en particulier. La configuration du serveur DHCP se fait sur le fichier "/etc/dhcp/dhcpd.conf".

```
#logs separates
deny unknown-clients;
log-facility local7;

subnet 10.1.2.0 netmask 255.255.255.0 {
    option routers 10.1.2.1;
    option domain-name "agence.atlantide";
    option domain-name-servers 192.168.1.2;
}

host client1 {
    hardware ethernet f0:4d:a2:a0:e1:af;
    fixed-address 10.1.2.2;
}

host client2 {
    hardware ethernet f0:4d:a2:a0:e2:7d;
    fixed-address 10.1.2.2;
}
```

Configuration du serveur DHCP.

Sur cette capture d'écran nous pouvons voir que le réseau subnet est tout naturellement notre réseau privé avec l'adresse "10.1.2.0" et le masque "255.255.255.0". Nous attribuons une adresse fixe à plusieurs clients car durant nos tests il est arrivé que nous ayons des clients différents. Afin d'attribuer une adresse fixe

il suffit d'indiquer l'adresse qu'il va récupérer "fixed-address" ainsi que sa mac adresse "hardware ethernet". Nous avons également renseigné différentes informations concernant le DNS mais nous aborderons le sujet plus tard.

Tous comme les interfaces réseau quand une modification est apportée il faut redémarrer le service associé afin de prendre en compte les changements effectués, avec la commande:

```
$ service isc-dhcp-server restart
```

Il est également possible de démarrer le service si il est arrêté avec:

```
$ service isc-dhcp-server start
```

Ou a contrario le stopper si besoin avec:

```
$ service isc-dhcp-server stop
```

Il est également possible d'afficher les erreurs en cas de soucis avec:

```
$ cat /var/log/syslog
```

Et finalement on peut afficher l'interface d'écoute du démon avec:

```
$ ps ax | grep dhcpd
```

Une fois que tout est mis en place et que tout est opérationnel, il nous faut une trace de ce qu'il va se passer. C'est pour cela que nous avons besoin de mettre en place un système de log pour serveur DHCP.

4.2 Système de log pour le DHCP

5 Sauvegarde automatique

5.1 Rsync

5.2 Cron

6 Manipulation paquets de Debian

Sous Debian il existe de multiples gestionnaires de paquets, mais le plus couramment utilisé est le gestionnaire "apt".

6.1 Mise à jour du système

6.2 Les paquets

Conclusion tp 1

Introduction tp 1bis

7 Interrogation d'un serveur DNS

7.1 Les commandes host, dig et nslookup

7.2 Fichier de renseignement du serveur DNS

7.3 Rôle du fichier /etc/hosts

8 Installation d'un DNS

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

8.1 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. "option domain-name "agence.atlantide" fournit le nom de domaine, dans ce cas il sert a faire référence aux ordinateurs du réseau par leurs nom sans ajouter le nom de domaine.

8.2 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

9 Installation d'un serveur LAMP

9.1 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 dabs /srv), les bon droits, mise en place du virtuals hosts pointant sur /srv/www

9.2 Installation et mise en place de PostgreSQL

Déroulement...

9.3 Installation et mise en place de PHP

Programme PHP qui se connecte a PostgreSQL et affichage infos pour la vérification.

9.4 Installation et mise en place de MySQL

Comme pour PostgreSQL et php

9.5 Installation et mise en place d'un PDO

10 Script de routage et matrice de filtrage

10.1 Scripts de routage au démarrage

10.2 Règles iptables de la matrice de filtrage

Conclusion tp 2

Introduction 3

11

11.1

11.2

Conclusion tp 3