

Laboratoires services

Rapport laboratoire 6

BULLONI Lucas & WERMEILLE Bastien

21 juin 2018

Table des matières

1	Introduction	3
1.1	Prérequis	3
1.2	Réseau du laboratoire	3
1.2.1	Réseau initial	3
1.2.2	Réseau final	3
2	Déploiement d'un sous-réseau	4
2.1	DHCP	4
2.1.1	Plage d'adresses dynamiques	4
2.1.2	Adresses statiques	4
2.2	DNS	4
2.2.1	Définition de la zone	4
2.2.2	Tests	6
2.2.3	Champs TXT	8
3	Déploiement de 4 sous-réseaux	9
3.1	Backbone	9
3.2	Tests	9
3.3	Échanges RIP	11
4	Autres services	12
4.1	DNS Slave	12
4.1.1	Test	12
4.2	NTP	13
4.2.1	Amélioration précision NTP	14
5	Questions	15
6	Questions pour le rapport	17
7	Conclusion	19

1 Introduction

Dans le cadre d'un laboratoire du cours "Réseau et application", les services de base d'un réseau informatique sont mis en pratiques. Les protocoles à mettre en place sont le DHCP, DNS, serveur web et optionnellement le protocole NTP. La partie NTP a été réalisée. Le but principal est de créer un petit réseau avec un DHCP et DNS fonctionnel, ensuite ce réseau va être dupliquer et sera connecté à tout ses clones via un réseau backbone. Toutes les réseaux sont composés uniquement de machine GNU/Linux. Tout le laboratoire se fera sur NetKit, logiciel de simulation d'environnement réseau virtuel.

1.1 Prérequis

- Un PC Linux avec NetKit
- Laboratoire netkit "qos"

1.2 Réseau du laboratoire

1.2.1 Réseau initial

Le réseau est composé de deux PC, dont un qui est un serveur web, ainsi qu'un serveur qui fait office de DNS et de DHCP. Les deux PC feront aussi office de serveur mail (uniquement dans les enregistrement DNS).

Le nom du réseau est net1.mylan.ch.

[Image sous-réseau]

1.2.2 Réseau final

L'objectif final est de dupliquer le premier réseau 3 fois et d'interconnecter les 4 sous-réseaux via un backbone pour qu'ils puissent communiquer entre eux.

[Image réseau final]

2 Déploiement d'un sous-réseau

La première étape du laboratoire est la configuration du serveur DNS et DHCP. Toutes les manipulations de configurations ont été faites dans le fichier "n1-router.startup" afin que les modifications soient préservées lors du redémarrage du laboratoire.

2.1 DHCP

La configuration DHCP se trouve dans la section du fichier .startup du serveur DHCP/DNS après la ligne :

```
cat » /etc/dhcp3/dhcpd.conf « EOF
```

2.1.1 Plage d'adresses dynamiques

La première étape est la configuration de la plage d'adresse IP dynamique. Toutes les machines configurées avec une adresse IP dynamique prendront une adresse entre 192.168.1.100/24 et 192.168.1.199/24.

L'adresse de broadcast est 192.168.1.255 et la passerelle est 192.168.1.1. Nous avons également ajouté le serveur DNS en prévoyance. Le serveur étant la même machine que le DHCP, l'adresse est également 192.168.1.1, de ce fait, le serveur sera en adresse statique.

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.100 192.168.1.199;  
    option routers 192.168.1.1;  
    option broadcast-address 192.168.1.255;  
    option domain-name-servers 192.168.1.1;  
    option domain-name "net1.mylan.ch";  
}
```

2.1.2 Adresses statiques

le serveur DNS/DHCP est configuré en IP statique, un serveur DHCP ne peut en effet pas s'attribuer une adresse dynamique. Il faut également faire attention à ne pas mettre une adresse statique dans la plage d'adresse dynamique pour ne pas créer de conflit. La configuration du serveur DHCP est faite dans la définition de la zone de la partie précédente.

Le serveur web est également configuré avec une adresse IP statique. En effet, un serveur Web ne va que très rarement changer d'adresse IP afin d'éviter des problèmes de mise à jour DNS. Nous avons décidé de donner l'adresse 192.168.1.42 pour ce serveur. La configuration est la suivante :

```
host tournedix {  
    hardware ethernet 00:00:00:01:00:00;  
    fixed-address 192.168.1.42;  
}
```

2.2 DNS

La configuration DNS est également faite dans le fichier .startup du serveur.

2.2.1 Définition de la zone

Comme énoncé précédemment, la zone (sous-réseau) est composé d'un serveur web et d'un PC simple, et d'un serveur DHCP/DNS. Comme le protocole l'énonce, les entrées DNS sont de type NS, les entrées simples A (pour

IPv4), les alias CNAME et les serveurs mails MX. [cours]

Certaines entrées doivent être configurées dans les deux sens, du passage de l'hostname à l'adresse IP et de l'adresse IP à l'hostname.

la configuration hostname vers adresse ip est faite après la ligne :

```
'cat > /var/lib/bind/net1.mylan.ch.zone « EOF'
```

Et la configuration hostname vers adresse IP est faite après la ligne :

```
'cat > /var/lib/bind/1.168.192.in-addr.arpa.zone « EOF'
```

Serveur DNS/DHCP Comme énoncé précédemment, le serveur DNS doit être de type NS. Mais cet enregistrement ne permet pas de spécifier son hostname, il faut donc également ajouter une adresse de type A pour pouvoir y accéder via son hostname. Un alias a également été fait pour que la machine puisse être accédée avec le nom 'routeur' et 'DNS'.

La configuration est donc la suivante :

```
@ IN NS dns.net1.mylan.ch.  
DNS IN A 192.168.1.1  
@ IN A 192.168.1.1  
routeur IN CNAME DNS
```

et dans l'autre zone :

```
@ IN NS dns.net1.mylan.ch.
```

Serveur Web Le serveur web sera accessible avec le nom www et pc1. Deux enregistrements de type A ont donc été faits :

```
pc1 IN A 192.168.1.42  
www IN A 192.168.1.42
```

Il faut également faire la correspondance inverse afin d'y accéder avec l'adresse complète (URL). La configuration est comme ceci :

```
42 IN PTR www.net1.mylan.ch.
```

Serveur mail Il a également fallu configurer une entrée DNS de type MX pour chaque PC, mais avec une priorité différente.

- pc1 : Priorité 0
- pc2 : Priorité 10

La configuration est la suivante :

```
@ IN MX 0 pc1  
@ IN MX 10 pc2.net1.mylan.ch.
```

2.2.2 Tests

DHCP dynamique Sur la prise d'écran ci-dessous, on peut remarquer que pc2 a bien pris une adresse dans la plage 192.168.1.100/24 - 192.168.1.199/24. La machine ayant pris l'adresse 192.168.1.101. L'adresse 192.168.1.100 n'a pas été prise, car au moment du test, pc1 était aussi en adresse dynamique.

```
n1-pc2:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:00:02:00:00
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::200:ff:fe02:0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:29 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3104 (3.0 KiB)  TX bytes:1298 (1.2 KiB)
          Interrupt:5
```

FIGURE 1 – Test DHCP dynamique

DHCP statique On a également testé l'adresse fixe pour pc1. Le tout étant fonctionnel :

```
n1-pc1 login: root (automatic login)
n1-pc1:~# dhclient eth0
There is already a pid file /var/run/dhclient.pid with pid 515
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/00:00:00:01:00:00
Sending on   LPF/eth0/00:00:00:01:00:00
Sending on   Socket/fallback
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.1
bound to 192.168.1.42 -- renewal in 279 seconds.
n1-pc1:~# host pc1.net1.mylan.ch
pc1.net1.mylan.ch      A      192.168.1.42
n1-pc1:~# host 192.168.1.42
Name: pc1.net1.mylan.ch
Address: 192.168.1.42

n1-pc1:~#
```

FIGURE 2 – Test DHCP statique pour pc1

Le log du serveur DHCP a également été vérifié afin d'être sûr qu'il n'y ait pas d'erreur.

```
n1-routeur:~# tail /var/log/syslog
May 16 13:36:33 n1-routeur dhcpd: DHCPREQUEST for 192.168.1.100 from 00:00:00:01:00:00 (pc1) via eth0
May 16 13:36:33 n1-routeur dhcpd: DHCPACK on 192.168.1.100 to 00:00:00:01:00:00 (pc1) via eth0
May 16 13:36:43 n1-routeur named[499]: client 192.168.1.1#45194: updating zone 'net1.mylan.ch/IN': adding an RR at 'pc2.net1.mylan.ch' A
May 16 13:36:43 n1-routeur named[499]: client 192.168.1.1#45194: updating zone 'net1.mylan.ch/IN': adding an RR at 'pc2.net1.mylan.ch' TXT
May 16 13:36:43 n1-routeur dhcpd: Added new forward map from pc2.net1.mylan.ch to 192.168.1.101
May 16 13:36:43 n1-routeur named[499]: client 192.168.1.1#39077: updating zone '1.168.192.in-addr.arpa/IN': deleting rrset at '101.1.168.192.in-addr.arpa' PTR
May 16 13:36:43 n1-routeur named[499]: client 192.168.1.1#39077: updating zone '1.168.192.in-addr.arpa/IN': adding an RR at '101.1.168.192.in-addr.arpa' PTR
May 16 13:36:43 n1-routeur dhcpd: added reverse map from 101.1.168.192.in-addr.arpa. to pc2.net1.mylan.ch
May 16 13:36:43 n1-routeur dhcpd: DHCPREQUEST for 192.168.1.101 from 00:00:00:02:00:00 (pc2) via eth0
May 16 13:36:43 n1-routeur dhcpd: DHCPACK on 192.168.1.101 to 00:00:00:02:00:00 (pc2) via eth0
n1-routeur:~#
```

FIGURE 3 – Log DHCP

DNS Sur l'image ci-dessous on peut voir que les entrées DNS de la zone et que le serveur DNS est bien configuré correctement.

```
n1-pc2:~# host -t ns net1.mylan.ch
net1.mylan.ch          NS      dns.net1.mylan.ch
n1-pc2:~# host -t ns net1.mylan.ch
net1.mylan.ch          NS      dns.net1.mylan.ch
n1-pc2:~# host -t a net1.mylan.ch
net1.mylan.ch          A      192.168.1.1
n1-pc2:~#
```

FIGURE 4 – Test DNS serveur

Et sur l'image suivante on peut voir que le l'alias pour le routeur est également fonctionnel.

```
n1-pc2:~# ping routeur.net1.mylan.ch
PING dns.net1.mylan.ch (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.142 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.356 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.562 ms
^C
--- dns.net1.mylan.ch ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.142/0.353/0.562/0.172 ms
n1-pc2:~#
```

FIGURE 5 – Test alias

Les tests des entrées pour pc1 et pc2 ont également été faits et sont un succès comme on peut le constater sur l'image ci-dessous.

```
n1-pc1:~# host pc1.net1.mylan.ch
pc1.net1.mylan.ch      A      192.168.1.42
n1-pc1:~# host 192.168.1.42
Name: pc1.net1.mylan.ch
Address: 192.168.1.42
```

FIGURE 6 – Test entré DNS PC

L'image montre uniquement un test sur pc1, mais un test à également été fait sur pc2 et produit le même résultat avec les bonnes adresses IP et hostname.

Serveur web Le serveur web a été testé avec la commande 'Lynx' comme énoncé dans le document du laboratoire et le test est aussi un succès. [doc-labo]

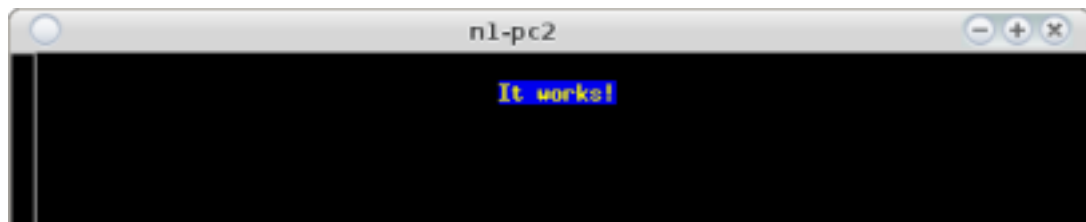


FIGURE 7 – Serveur Web

2.2.3 Champs TXT

Le champ TXT est une simplement des informations textuelles sur le serveur. Ce champ TXT généré automatique contient. [TXT]

L'entrée générée ci-dessous contient un hash qui permet de vérifier l'authenticité du DNS.

```
n1-pc2
n1-pc2:~# host -t TXT pc2.net1.mylan.ch
pc2.net1.mylan.ch      TXT      "00952356d21fa5cf0e773bcd81ec3cb33"
n1-pc2:~#
```

FIGURE 8 – Entrée TXT

3 Déploiement de 4 sous-réseaux

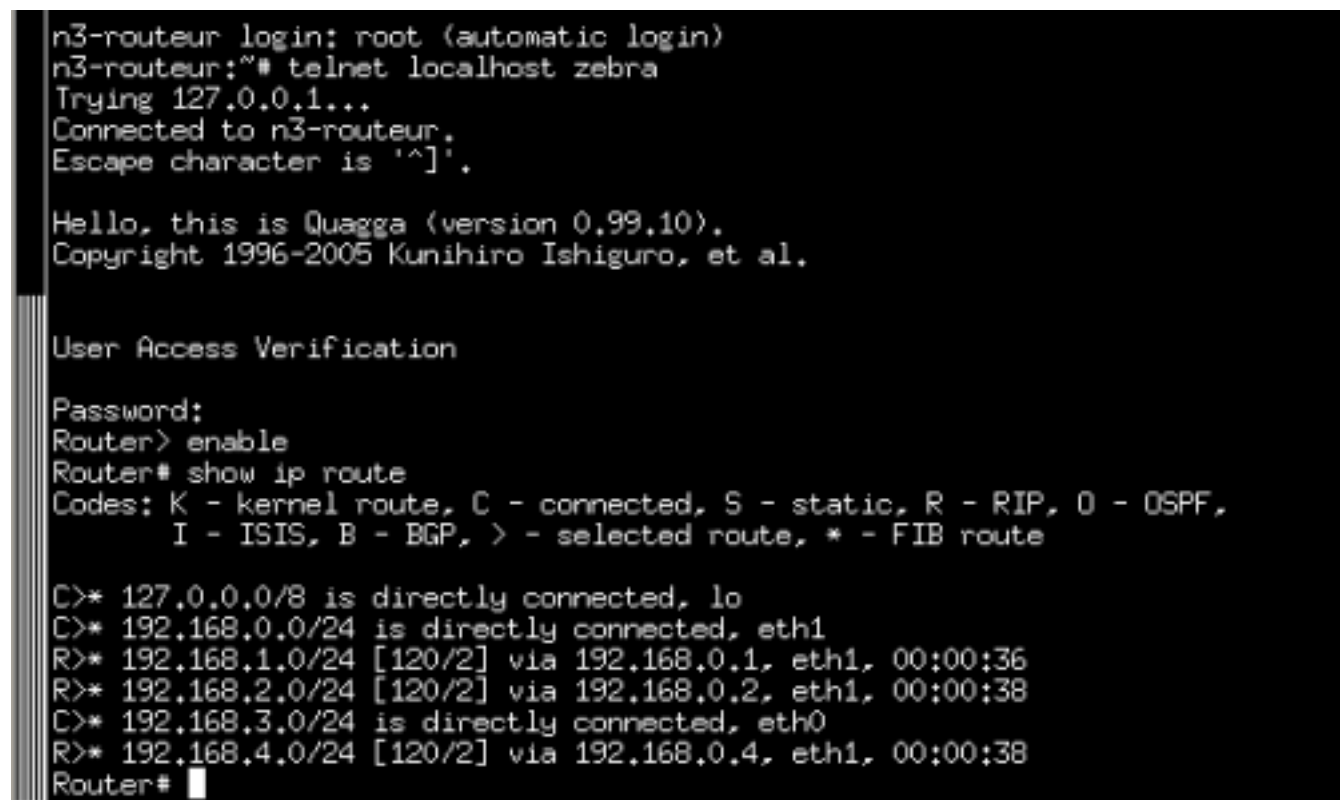
L'étape suivante est de dupliquer le sous-réseau 3 fois et de les interconnectés. Les routeurs auront donc deux interfaces. Une pour le réseau internet et une autre pour l'accès aux autres réseaux. Ces interfaces seront d'adresse 192.168.0.X/24.

3.1 Backbone

Le routage entre les différents sous-réseaux est fait avec Zebra, logiciel qui s'occupe de faire du routage TCP/IP [Zebra]. Mais le logiciel semble ne plus être maintenu et a été remplacé/continué sous le nom Quagga, comme on peut le constater dans l'image à la fin de cette section.

Afin d'ajouter Zebra à notre réseau, nous avons simplement ajouter la confirmation du document du laboratoire [doc-labo], celle-ci ne demandant pas d'adaptation.

Ensuite le réseau net1 a été dupliqué 3 fois grâce au script 'duplicate-subnet.sh' fournit avec le laboratoire. Nous avons également testé que tous les sous-réseaux fonctionne en faisant simplement des pings entre les machines. Comme énoncé dans le document, on se connectant en SSH à Zebra, on peut voir des routes échangées (celles préfixée par R) et d'autre directe (préfixées par C)



```
n3-routeur login: root (automatic login)
n3-routeur:~# telnet localhost zebra
Trying 127.0.0.1...
Connected to n3-routeur.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.10).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router> enable
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.0.0/24 is directly connected, eth1
R>* 192.168.1.0/24 [120/2] via 192.168.0.1, eth1, 00:00:36
R>* 192.168.2.0/24 [120/2] via 192.168.0.2, eth1, 00:00:38
C>* 192.168.3.0/24 is directly connected, eth0
R>* 192.168.4.0/24 [120/2] via 192.168.0.4, eth1, 00:00:38
Router#
```

FIGURE 9 – Zebra

Le réseau final sera donc celui présenté dans l'introduction :
[IMAGE]

3.2 Tests

Ensuite nous avons testé en faisant des pings depuis pc1 de net3 à une machine du réseau 1 et du réseau 2. Comme on peut le voir sur la prise d'écran ci-dessous, cela fonctionne correctement.

```
n3-pc1 login: root (automatic login)
n3-pc1:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=15.5 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=0.581 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=0.614 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2009ms
rtt min/avg/max/mdev = 0.581/5.575/15.530/7.039 ms
n3-pc1:~# ping pc2.net2.mylan.ch
PING pc2.net2.mylan.ch (192.168.2.100) 56(84) bytes of data.
64 bytes from pc2.net2.mylan.ch (192.168.2.100): icmp_seq=1 ttl=62 time=10.3 ms
64 bytes from pc2.net2.mylan.ch (192.168.2.100): icmp_seq=2 ttl=62 time=0.262 ms
64 bytes from pc2.net2.mylan.ch (192.168.2.100): icmp_seq=3 ttl=62 time=0.447 ms
^C
--- pc2.net2.mylan.ch ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2010ms
rtt min/avg/max/mdev = 0.262/3.680/10.331/4.703 ms
n3-pc1:~#
```

FIGURE 10 – Tests backbone

On peut également voir qu'un ping est fait avec l'adresse IP et un autre avec le hostname. Ce qui valide également le test de fonctionnement DNS.

Un traceroute a également été fait de pc1.net1 à pc1.net4. On peut voir que le paquet passe bien par les interfaces réseaux du backbone.

```
n1-pc1 login: root (automatic login)
n1-pc1:~# traceroute pc1.net4.mylan.ch
traceroute to pc1.net4.mylan.ch (192.168.4.42), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0 ms  0 ms  0 ms
 2  192.168.0.4 (192.168.0.4)  0 ms  0 ms  0 ms
 3  www.net4.mylan.ch (192.168.4.42)  6 ms  0 ms  0 ms
n1-pc1:~#
```

FIGURE 11 – Traceroute

3.3 Échanges RIP

Afin d'éviter les boucles de routages, les routeurs communiquent entre eux avec le protocole RIP [RIP]. Nous avons vérifié cela en capturant les paquets RIP. Comme on peut le voir sur l'image ci-dessous :

0.000000	192.168.0.1	224.0.0.9	RIPv2	66 Response
10.177421	192.168.0.3	224.0.0.9	RIPv2	66 Response
12.128346	192.168.0.2	224.0.0.9	RIPv2	66 Response
13.192698	192.168.0.4	224.0.0.9	RIPv2	66 Response
36.011300	192.168.0.1	224.0.0.9	RIPv2	66 Response
42.138737	192.168.0.2	224.0.0.9	RIPv2	66 Response
42.201865	192.168.0.4	224.0.0.9	RIPv2	66 Response
46.178139	192.168.0.3	224.0.0.9	RIPv2	66 Response

FIGURE 12 – RIP

Le protocole est RIPv2, qui est simplement la deuxième amélioration du protocole de base.

4 Autres services

4.1 DNS Slave

Un DNS Slave est un DNS qui va baser tout ses enregistrement sur un autre serveur DNS [**Slave**]. Comme demander dans le document, le DNS du réseau 3 va devenir le DNS slave du réseau 2, qui sera donc son master. Pour cela il faut donc modifier la configuration du DNS du réseau 2 pour lui indiquer qu'il a un slave et également modifier la configuration du réseau 3 pour lui indiquer qu'il a un master.

Pour cela il faut premièrement ajouter un enregistrement DNS au réseau 2 pour lui indiquer l'existence du serveur 3.

```
dns IN A 192.168.3.1
;et sa correspondance
@ IN NS dns.net3.mylan.ch
```

Ensuite il faut indiquer au serveur 3 qu'il est un slave en ajoutant ces lignes après la ligne 'cat » /etc/bind/named.conf.local «EOF' :

```
zone "net2.mylan.ch" IN {
    type slave;
    masters { 192.168.2.1; };
    file "/var/lib/bind/net2.mylan.ch.slave";
};

zone "2.168.192.in-addr.arpa" IN {
    type slave;
    masters { 192.168.2.1; };
    file "/var/lib/bind/2.168.192.in-addr.arpa.slave";
};
```

4.1.1 Test

On a pu testé que le DNS 3 est bien le slave de DNS 2 en faisant la commande :

```
host -t ns net2.mylan.ch 192.168.0.1
```

et en répétant l'opération pour tous les sous-réseaux 192.168.0.X. On peut constater sur l'image ci-dessous que ça a bien fonctionné.



```
n3-routeur:~# host -t ns net2.mylan.ch 192.168.0.1
net2.mylan.ch      NS      dns.net2.mylan.ch
net2.mylan.ch      NS      dns.net3.mylan.ch
n3-routeur:~# host -t ns net2.mylan.ch 192.168.0.2
net2.mylan.ch      NS      dns.net2.mylan.ch
net2.mylan.ch      NS      dns.net3.mylan.ch
n3-routeur:~# host -t ns net2.mylan.ch 192.168.0.3
net2.mylan.ch      NS      dns.net2.mylan.ch
net2.mylan.ch      NS      dns.net3.mylan.ch
n3-routeur:~# host -t ns net2.mylan.ch 192.168.0.4
net2.mylan.ch      NS      dns.net3.mylan.ch
net2.mylan.ch      NS      dns.net2.mylan.ch
n3-routeur:~#
```

FIGURE 13 – Test slave

4.2 NTP

NTP est un protocole de synchronisation de temps qui permet d'avoir une précision à quelques millisecondes près [NTP]. Nous avons donc testé ce protocole en l'installant sur la machine debian. Le paquet n'étant pas présent dans le laboratoire netkit, nous n'avons pas pu tester tout ce qui était relatif à notre petit sous-réseau.

Premièrement, il a fallu installer NTP avec la commande 'apt-get install ntp'.

Ensuite il nous avons pu visualiser les strates avec la commande 'ntpq -p'. Les strates sont les serveurs de références sur le quels notre serveur va se synchroniser. Les strates sont organisés sous-forme d'arbre, ayant pour racine une horloge atomique.

```
pclabo@poste-404:~/NETKIT/LABOS/services/services$ ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
*ifconfig.com.ua .GPS.          1 u  -   64   1   62.438 -2.571  0.106
gintonic.woody. 195.186.4.101    3 u   1   64   1   16.368 -4.186  0.552
nsl.nexellent.n 194.242.34.149    2 u   1   64   1    3.559 -5.674  0.568
static.dsl.as87 131.188.3.222    2 u   2   64   1    4.317 -3.317  0.204
pclabo@poste-404:~/NETKIT/LABOS/services/services$ ntpdc -p
      remote           local      st poll reach  delay  offset  disp
=====
*gandalf.teleinf 157.26.77.44      2   64   3 0.00020 -0.002589 0.25174
pclabo@poste-404:~/NETKIT/LABOS/services/services$
```

FIGURE 14 – Strates

Les paquets de synchronisation ont également été capturés avec wireshark. On peut voir la deuxième prise d'écran les données utiles à l'heure, comme par exemple le délai avec la racine ou l'heure exacte.

3	1.835674	157.26.77.44	78.26.180.80	NTP	90	NTP Version 4, client
4	1.898433	78.26.180.80	157.26.77.44	NTP	90	NTP Version 4, server
7	4.835556	157.26.77.44	217.147.208.1	NTP	90	NTP Version 4, client
8	4.839389	217.147.208.1	157.26.77.44	NTP	90	NTP Version 4, server
10	5.835618	157.26.77.44	212.25.15.129	NTP	90	NTP Version 4, client
11	5.840119	212.25.15.129	157.26.77.44	NTP	90	NTP Version 4, server
15	7.835677	157.26.77.44	157.161.57.2	NTP	90	NTP Version 4, client
16	7.852466	157.161.57.2	157.26.77.44	NTP	90	NTP Version 4, server
21	9.612063	157.26.77.44	157.26.77.13	NTP	90	NTP Version 4, client
22	9.612462	157.26.77.13	157.26.77.44	NTP	90	NTP Version 4, server
25	11.612083	157.26.77.44	157.26.77.13	NTP	90	NTP Version 4, client
26	11.612375	157.26.77.13	157.26.77.44	NTP	90	NTP Version 4, server
31	13.612079	157.26.77.44	157.26.77.13	NTP	90	NTP Version 4, client
32	13.612385	157.26.77.13	157.26.77.44	NTP	90	NTP Version 4, server
35	15.611910	157.26.77.44	157.26.77.13	NTP	90	NTP Version 4, client

FIGURE 15 – NTP : Paquets de synchronisation (1)

```

Network Time Protocol (NTP Version 4, client)
  ▶ Flags: 0x23, Leap Indicator: no warning, Version number: NTP Version 4
    Peer Clock Stratum: secondary reference (2)
    Peer Polling Interval: 6 (64 sec)
    Peer Clock Precision: 0.000000 sec
    Root Delay: 0.062408447265625 seconds
    Root Dispersion: 0.22564697265625 seconds
    Reference ID: 78.26.180.80
    Reference Timestamp: May 30, 2018 13:47:30.147251473 UTC
    Origin Timestamp: May 30, 2018 13:48:29.986006719 UTC
    Receive Timestamp: May 30, 2018 13:48:29.992943451 UTC
    Transmit Timestamp: May 30, 2018 13:49:36.989138195 UTC

```

FIGURE 16 – NTP : Paquets de synchronisation (2)

4.2.1 Amélioration précision NTP

Si une précision à la milliseconde n'est pas suffisant, il existe un protocole le protocole PTP qui permet d'atteindre une précision à la nanoseconde [PTP]. Mais ce protocole requiert des appareils le supportant. Car la plupart des machine ne permettent pas d'avoir une telle précision.

5 Questions

1. Quel est le rôle du mot clé authoritative dans la configuration du serveur DHCP ?

Dans le cas où l'on a configuré le sous-réseau "aa.bb" dans le DHCP, une requête venant d'un sous-réseau "non identifié" "aa.bb.cc" sera ignorée si ce mot clé n'est pas activé, s'il est activé, le serveur répondra avec un NACK

2. Pour quelle raison aurait-on tendance à configurer une durée de bail DHCP courte (p.ex. 1h) ? ou longue ? (p.ex. 1 semaine)

- bail court : quand il y a beaucoup de clients et pas beaucoup d'adresses IP disponibles dans la plage
- bail long : quand il y a de grandes plages d'adresses IP et pas beaucoup de clients

3. Peut-on imaginer un réseau dans lequel aucune adresse de type hard static n'existe ?

Non, il faut au moins que le routeur ait une adresse hard static, afin que les différents périphériques puissent joindre le serveur. TODO check

4. Quel est l'avantage de configurer des imprimantes, machines virtuelles ou éventuellement serveurs en adresse de type DHCP static ?

Ces périphériques sont toujours censés être disponibles ! Les mettre en adresse dynamique n'apporte rien. Ces périphériques doivent être en adresse statique afin de toujours pouvoir être atteint de la même manière. Certains programmes par exemple sauvegardent l'adresse IP de certains serveurs au lieu du nom, ce qui poserait des problèmes pour ceux-ci. La configuration statique permet également de limiter le nombre de requêtes inutiles à chaque changement d'adresse des ces différents serveurs/imprimantes/...

5. Peut-on mettre les plages d'adresses DHCP dynamiques là où se trouvent des adresses hard static ?

Non, le serveur DHCP risquerait d'allouer dynamiquement une adresse statique déjà allouée. Si nous voulions pouvoir faire cette manipulation qui n'apporte pas grand-chose, il faudrait que le serveur DHCP sache que ces adresses sont déjà allouées ce qui revient à modifier la plage d'adresses DHCP disponible pour allocation.

6. Qu'est-ce que Cisco appelle des manual bindings ?

Ce sont des adresses IP (un pool) liées à certaines adresses MAC connues par le serveur DHCP.
TODO completer

7. Expliquez le concept derrière la configuration DNS suivante (faites abstraction du CNAME, mais tenez compte du TTL et du fait qu'il y a 2 champs A différents) :

```
“ $ dig -t a www.yahoo.com www.yahoo.com. 216 IN CNAME fd-fp3.wg1.b.yahoo.c fd-fp3.wg1.b.yahoo.com. 38 IN
A 46.228.47.114 fd-fp3.wg1.b.yahoo.com. 38 IN A 46.228.47.115
#lancez plusieurs fois la commande pour voir si quelque chose change “ on a un serveur dupliqué donc deux
serveurs à disposition. Cela permet de répartir la charge sur plusieurs serveurs différents et d'avoir de la redondance
en cas de panne.
```

8. Consultez sur un serveur DHCP le fichier “/var/lib/DHCP/dhcpd.leases” : que contient-il ? qu'en déduisez-vous sur la volatilité des adresses IP dynamiques ?

Contiens les bails DHCP. Les adresses IP dynamiques qui ont été allouées possèdent une "durée de vie" limitée qui expire au bout d'un certain moment. Une fois le bail expiré, celui-ci est supprimé et le périphérique qui possédait e bail doit en redemander un.

9. Capturez et expliquez ce qui se passe avec : “host big-entry.alphanet.ch” (sur machine réelle, ou “nslookup”)

Il y a de nombreuses adresses IP qui s'affichent : “big-entry.alphanet.ch has address 192.168.24.25”

il y a plusieurs serveurs physiques qui répondent à ce nom (plusieurs entrées avec le même nom dans la résolution d'adresses directe DNS)

TODO ...

10. Expliquez la résolution inverse (adresse IP vers nom)

La résolution inverse permet de récupérer le Fully qualified domain name (FQDN) à partir d'une adresse IP. Le FQDN est le nom de domaine qualifié qui correspond au nom de domaine épelé sans abréviation, y compris le potentiel sous-domaine, le nom de domaine et l'extension du nom de domaine. Par exemple : *www.domaine.ch*.

11. Comment un serveur de nom trouve-t-il quels serveurs gèrent la racine “.“ ?

Le nombre de serveurs racines est limité dans le monde et ceux-ci sont accessibles publiquement.

12. A quoi sert l'option DNS (bind/named) “forwarders” ?

L'option “forwarders” permet de rediriger les requêtes qui ne sont pas résolues par notre serveur vers un serveur DNS distant (serveurs DNS de votre FAI par exemple).

Cela permet d'utiliser le cache d'un serveur déjà existant et donc d'obtenir des temps d'accès plus rapides. Si la requête DNS n'est pas résolue par le serveur DNS “distant” alors la requête sera envoyée aux serveurs DNS racine.

13. Vous modifiez une zone gérée par un master et un slave DNS, que devez-vous absolument changer ?

Il est nécessaire de mettre à jour les informations des 2 serveurs DNS, du master et du slave.

14. Que pensez-vous de la sécurité du protocole DNS ? Quelles possibilités existent pour l'améliorer ?

Ce protocole n'est pas sûr et possède plusieurs problèmes de sécurité tel le "DNS Spoofing" ou "DNS cache Poisoning", qui consiste à insérer de faux enregistrements DNS dans le serveur afin que celui-ci retourne une réponse erronée au client qui va se retrouver sur le mauvais site. Il existe une version sécurisée de ce protocole, DNSSEC qui est comme DNS, mais qui inclut du cryptage.

15. Vous venez de configurer deux serveurs DNS pour la zone “mondomaine.ch”, et ils sont associés à un registrar : expliquez ce que le registry 1 doit faire pour que cela fonctionne, techniquement (indication : “whois alphanet.ch” ; de plus, voir les entrées “NS” sur “n1-routeur” de “mylan.ch” vers les sous-domaines)

Faire remonter l'existence de ce réseau au DNS supérieur qui s'occupe de la zone.

16. Pourquoi est-ce important d'avoir des champs PTR existants et cohérents ?

Les champs PTR permettent d'associer une adresse IP à un nom d'hôte. Un PTR correspond à un enregistrement inversé. Les champs PTR sont indispensables pour les serveurs de messageries par exemple. Les serveurs de messageries contrôlent que l'adresse IP correspond au nom du serveur source. Dans le cas où le PTR n'est pas configuré alors la vérification ne fonctionne pas et l'email est rejeté.

17. A quoi servent les champs de type SRV ? et TXT ? et NAPTR ? (donnez un exemple pour chacun)

- SRV : indique quel serveur s'occupe d'un service spécifique : SIP
- TXT : texte libre. Exemple : notification de règles SPF (Shortest Path First)
- NAPTR : conversion de valeurs. Exemple : n° de téléphone en URI SIP

18. A-t-il été nécessaire de changer le protocole DNS pour supporter les domaines accentués, où seule une modification du client a suffi ? (indication : punycode, exemple : "linux-neuchâtel.ch" dans votre navigateur)

Non, il n'a pas été nécessaire de modifier le protocole DNS du côté du serveur. Nous utilisons à la place le punycode qui est une syntaxe de codage qui est utilisée avec les noms de domaines internationaux.

19. A quoi sert le DNSSEC ?

Sécuriser le protocole 'DNS' avec l'ajout d'un chiffrement des données.

6 Questions pour le rapport

1. Expliquez ce que l'on doit faire sur le maître et le secondaire pour activer un secondaire (slave)

La configuration se déroule en 2 étapes. La première consiste à ajouter un enregistrement NS référençant le serveur DNS esclave sur le serveur maître. Finalement, il faut ajouter une zone du côté du slave qui référence le serveur avec comme type "slave".

Il s'agit de définir la zone DNS sur le serveur maître en précisant le type master et du côté du serveur DNS esclave, configurer la même zone en précisant l'adresse du serveur maître.

2. Sécurité d'un serveur de nom

Indications, http://docstore.mik.ua/orelly/networking_2ndEd/dns/ch11_02.htm

(a) Pourquoi les serveurs récursifs sont-ils dangereux ?

Car un attaquant peut faire une attaque de type "DNS Spoofing" sans avoir à se trouver dans le réseau. Il peut faire une requête DNS au roteur vers site dont il veut détourner le trafic et immédiatement après envoyer des réponses DNS au serveur DNS avec une réponse pour ce site avec l'adresse IP de son site malveillant.

Cela lui permet de faire croire à l'utilisateur qu'il se trouve sur le site qu'il a demandé alors qu'il ne l'est pas.

(b) Pourquoi est-ce une bonne pratique de séparer les serveurs autoritaires d'une zone sur Internet d'un serveur récursif pour un sous-réseau ?

Afin que le serveur récursif ne soit accessible que depuis le sous-réseau et pas depuis l'extérieur. Cela permet de limiter les attaques par spoofing.

(c) Qu'est-ce qu'une attaque de trafic amplification sur le DNS ? Comment l'éviter ?

Il s'agit d'envoyer des paquets DNS à des serveurs récursifs ouverts avec comme provenance l'adresse du serveur à attaquer. La requête DNS pourrait être `dig ANY isc.org @x.x.x.x`. Cette requête de 64 bytes peut retourner environ 3'223 bytes, ce qui correspond à une multiplication du trafic(amplification) par 50.[DDOS][amplification][attaques]

Le comble de ce problème est que la requête en question est possible grâce au protocole DNSSEC qui permet d'améliorer la sécurité des réseaux. Cette requête permet de retourner la liste des clés DNSSEC.

Une manière d'éviter cette problématique est de fermer l'accès des serveurs DNS récursif afin d'empêcher les attaquants de les détourner.

(d) Avancés : évaluez la résistance d'une configuration BIND9 de base face aux attaques de spoofing basées sur le devinage du champ ID de la requête ?

Indications :

- Est-ce que le numéro de port de la requête est également devinable ? si oui, quel est l'impact sécurité et comment corriger le problème ?
- Que peut-il se passer si l'on met un serveur BIND9 avec numéro de port variable et aléatoire derrière un firewall NAT/PAT, par exemple un routeur ADSL ?

Oui, le port de la requête est devinable, TODO

3. Qu'est-ce que la configuration de views BIND ?

Permet de présenter une configuration du serveur DNS selon la provenance de la requête. Certains périphériques auront accès à une version et d'autres à une toute autre configuration.[[views](#)]

4. Comparez les protocoles de routage interne OSPF et RIPv2

OSPF[[OSPF](#)] se base sur la rapidité d'un chemin alors que RIP[[RIP](#)] fonctionne avec les "hop count". OSPF envoie des messages "link-state advertisement" (lsa) qui contiennent la liste des routeurs atteignable par lui-même à tous ses voisins, propagé ensuite à tous les routeurs du réseau. Cet ensemble de LSA permet de connaître la structure du réseau. Chaque routeur utilisant finalement l'algorithme de Dijkstra afin de trouver le chemin le plus court vers chaque réseau.

RIP quant à lui ne prend en compte que le nombre de sauts en chaque routeur. Il envoie des messages toutes les 30 secondes à tous ses voisins avec de leur communiquer la liste des distances qui le sépare des différents réseaux connus. L'avantage de ce protocole est qu'il est simple à mettre en place et supporté par un grand nombre de routeurs. D'un autre côté, celui-ci a une très lente convergence dans de grands réseaux et peut être inefficace dans certains cas, car il ne prend pas en compte la vitesse des lignes.[[RIP-VS-OSPF](#)]

RIP consomme une partie de la bande passante en continu alors que OSPF lui en consomme principalement lors de l'initialisation du réseau, jusqu'à ce que tout soit calculé puis très peu de paquets sont envoyés par la suite.

5. La délégation de zone inverse est par classe A-C, qui sont obsolètes : des délégations de granularité inférieure à /24 sont aujourd'hui nécessaires : consultez le RFC 2317 pour expliquer la solution moderne

La solution "Moderne" consiste à faire déléguer la résolution inverse de ces réseaux par le FAI à notre propre serveur DNS. Pour ce faire, nous allons en collaboration avec le FAI définir un label pour nommer la zone. Le FAI va ainsi déléguer la résolution-inverse pour la zone contenant nos sous-réseaux à notre propre serveur DNS. Le FAI va également créer un CNAME pour chacun de nos sous-réseaux.

Ainsi, la résolution DNS standard nom vers adresse sera disponible du côté du FAI et la résolution inverse sera effectuée sur notre propre serveur DNS[[RFC-2317-ietf](#)][[RFC-2317-divers](#)].

6. Quand est-ce qu'un CNAME est impossible ? (indication : peut-il coexister avec un même noeud ou une feuille du même niveau) quelles sont les limitations supplémentaires sur CNAME ?

La première limitation est la suivante, un CNAME[CNAME] doit toujours pointer sur un autre nom domaine et jamais sur une adresse. La seconde est qu'aucun autre enregistrement ne doit redéfinir le label du CNAME ! Il ne faut pas non plus utiliser de CNAME pour la racine du domaine, mais plutôt utiliser un ALIAS[ALIAS].

Un CNAME ne devrait en théorie pas pointer sur un autre CNAME, cela pourrait causer une boucle DNS non résoluble dans le cas où les deux CNAME se pointent l'un l'autre.[cname-limitations]

7 Conclusion