

Livrable de

S.A.É. 2.03 :

Installation de service réseau

B.U.T. Informatique : semestre 2 (2022)

I.U.T. de Bayonne et du Pays Basque

DARGAZANLI Nicolas, MAURICE Alexandre

en TD I, TP 1.

Table des matières

1. Explication détaillées du plan d'adressage retenu	3
1.1. Plan du réseau	4
2. Paramétrage des machines	5
2.1. Le serveur DHCP:	5
2.2 Les machines administratives	7
2.3. Les machines entrepôt	8
2.4 Le routeur R0	9
2.5 Le routeur R1	10
2.6 Le routeur R2	11
2.7 Le serveur FTP	12
2.8 Le serveur de gestion	13
3. Tests	14
3.1 Test DHCP	14
3.2 Test de connectivité internet	15
3.3 Test du service FTP	16
3.4 Test de l'application de gestion	17

1. Explication détaillées du plan d'adressage retenu

Nous avons conclu que des adresses de classe C étaient suffisantes pour les besoins de la société : celle-ci requérant 3 sous-réseaux, nous devons modifier le masque en y ajoutant 2 bits de sous-réseau, ce qui veut dire qu'un sous-réseau ne sera pas utilisé. L'adresse du réseau sera alors 192.168.0.0, et le masque 255.255.255.192.

Le premier sous-réseau sera celui des serveurs, commençant à 192.168.0.0/26, et terminant à 192.168.0.63/26.

- La machine FTP aura l'adresse 192.168.0.10/26
- La machine de gestion aura l'adresse 192.168.0.20/26
- La carte ethernet eth0 du routeur R2 aura l'adresse 192.168.0.1/26
- La carte ethernet eth1 du routeur R1 aura l'adresse 192.168.0.2/26

Le deuxième sous-réseau sera celui de l'entrepôt, commençant à 192.168.0.64/26, et terminant à 192.168.0.127/26.

- La carte ethernet eth1 du routeur R2 aura l'adresse 192.168.0.65/26
- La carte ethernet eth1 du serveur DHCP aura l'adresse 192.168.0.66/26

Le troisième sous-réseau sera celui de l'administration, commençant à 192.168.0.128/26, et terminant à 192.168.0.191/26.

- La carte ethernet eth0 du routeur R1 aura l'adresse 192.168.0.189/26
- La carte ethernet eth0 du routeur R0 aura l'adresse 192.168.0.190/26
- La carte ethernet eth0 du serveur DHCP aura l'adresse 192.168.0.129/26

Le dernier réseau ne fait pas partie des sous-réseaux, il est uniquement présent dû à une limitation de *Marionnet* ne pouvant pas changer le masque et les derniers octets de l'adresse IP. Nous sommes donc obligés de mettre en place un routeur R0. Sans limitation nous aurions ajouté le modem/passerelle directement dans le sous-réseau de l'administration.

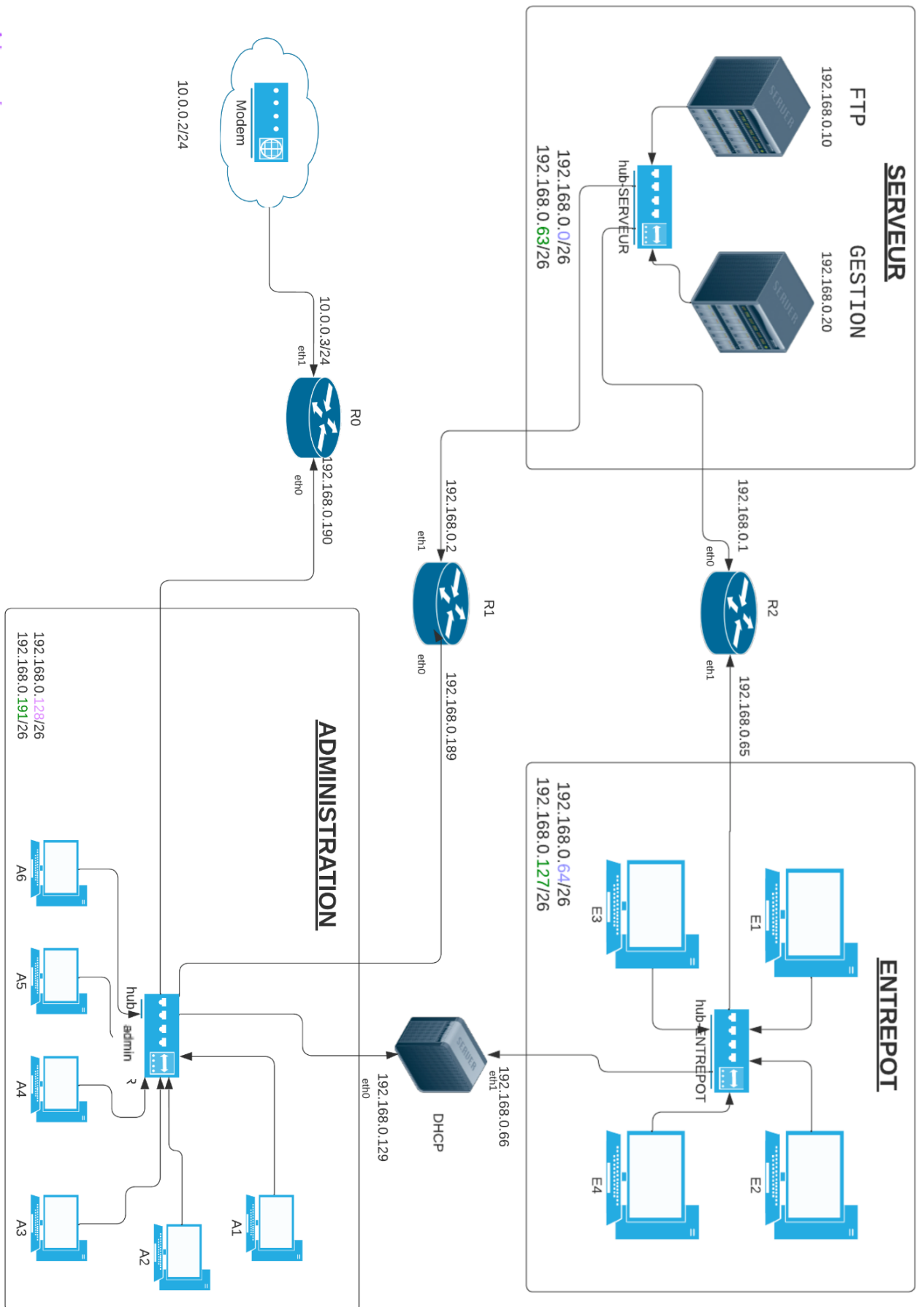
- Le modem aura comme adresse 10.0.0.2/24
- La carte ethernet eth1 du routeur R0 aura l'adresse 10.0.0.3/24

Remarque :

- Les adresses des serveurs FTP et Gestion ont été assignées de manière à les mémoriser facilement
- Les adresses des routeurs et du serveur DHCP ont été assignées en début ou fin de plage d'IP afin de pouvoir facilement les exclure de la configuration DHCP

1.1. Plan du réseau

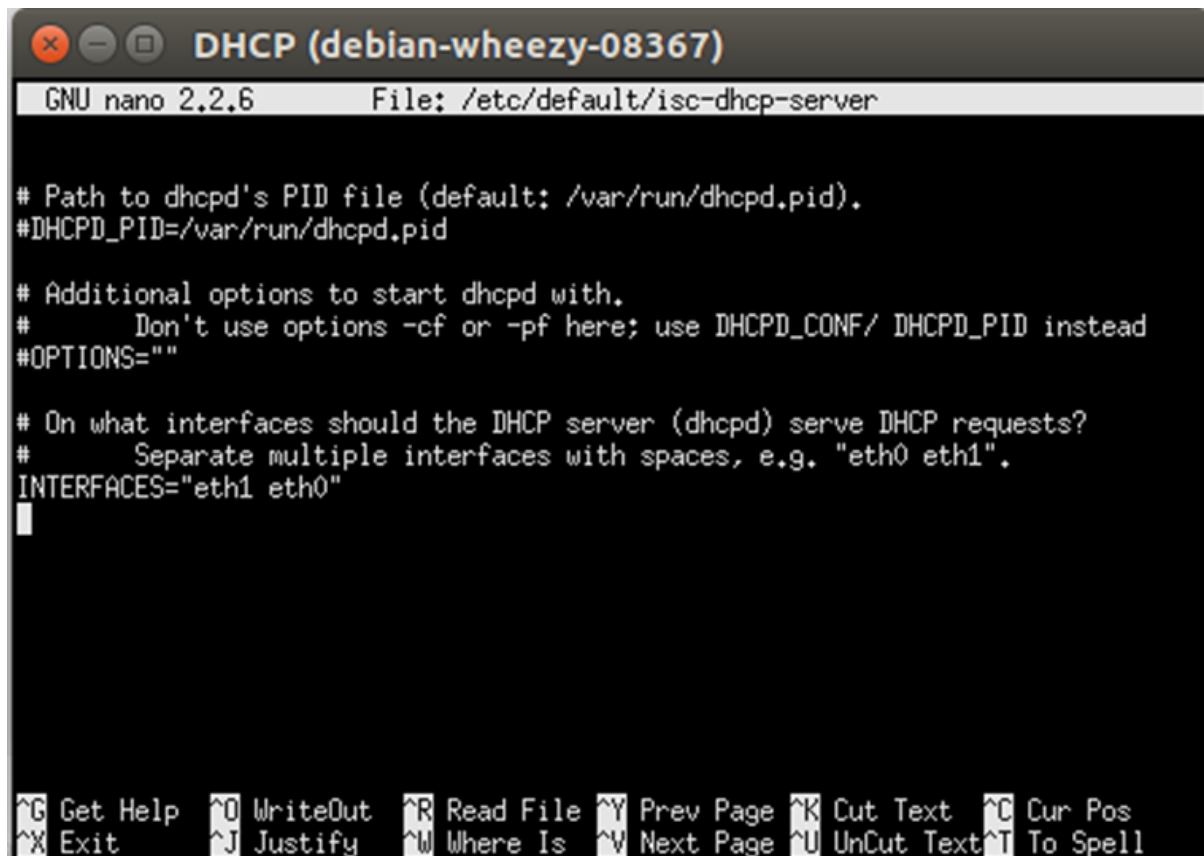
Adresse réseau
Adresse Broadcast



2. Paramétrage des machines

2.1. Le serveur DHCP

Dans un premier temps, nous devons configurer le serveur DHCP afin d'attribuer les IP aux prochaines machines. Dans le fichier `/etc/default/isc-dhcp-server`, nous indiquerons les interfaces sur lesquelles le serveur DHCP doit écouter et intervenir, soit `eth0` pour le réseau administratif et `eth1` pour le réseau de l'entrepôt.



```

DHCP (debian-wheezy-08367)
GNU nano 2.2.6      File: /etc/default/isc-dhcp-server

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

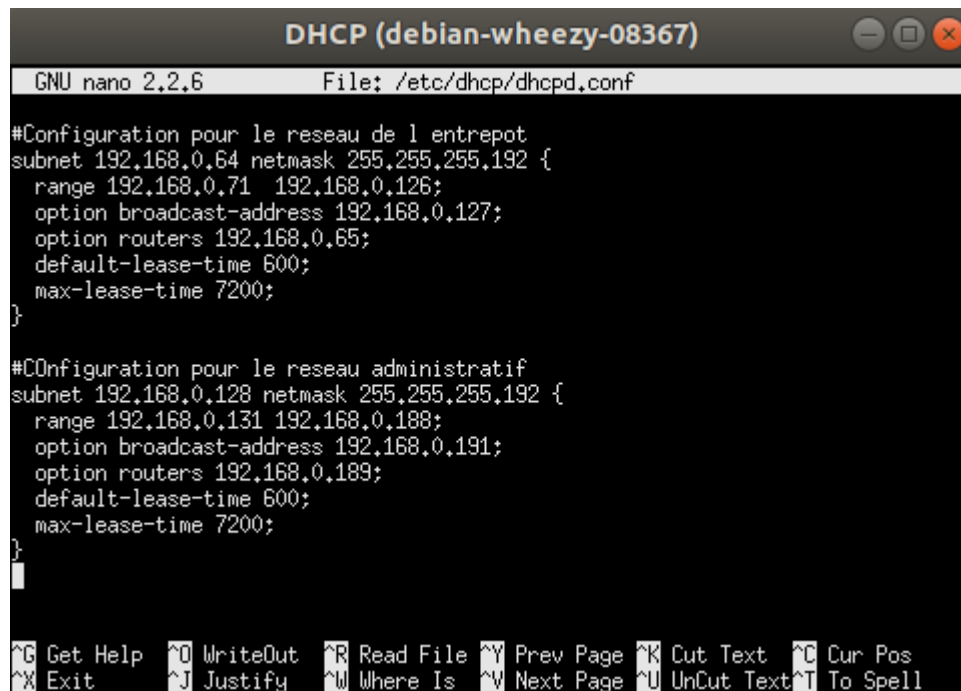
# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

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

```

[^]G Get Help [^]O WriteOut [^]R Read File [^]Y Prev Page [^]K Cut Text [^]C Cur Pos
[^]X Exit [^]J Justify [^]W Where Is [^]V Next Page [^]U UnCut Text [^]T To Spell

Nous allons maintenant configurer les réseaux sur lequel le serveur DHCP va intervenir, en ajoutant le réseau de l'entrepôt, avec le routeur par défaut, sa plage d'adresse et les informations générales du réseau. De même pour le réseau administratif.



```
GNU nano 2.2.6 File: /etc/dhcp/dhcpd.conf

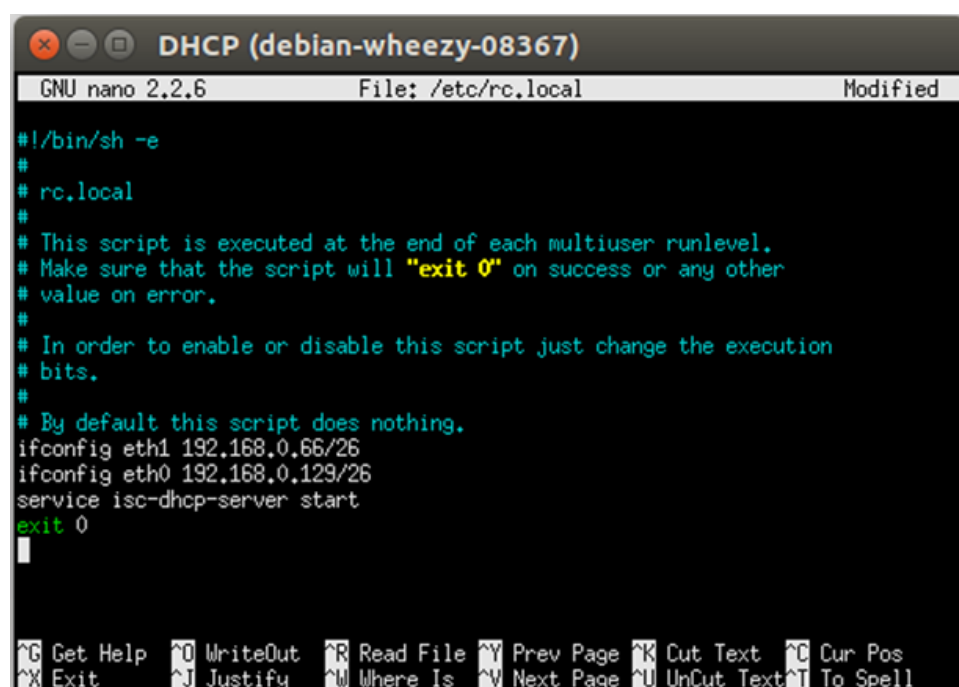
#Configuration pour le reseau de l entrepot
subnet 192.168.0.64 netmask 255.255.255.192 {
    range 192.168.0.71 192.168.0.126;
    option broadcast-address 192.168.0.127;
    option routers 192.168.0.65;
    default-lease-time 600;
    max-lease-time 7200;
}

#Configuration pour le reseau administratif
subnet 192.168.0.128 netmask 255.255.255.192 {
    range 192.168.0.131 192.168.0.188;
    option broadcast-address 192.168.0.191;
    option routers 192.168.0.189;
    default-lease-time 600;
    max-lease-time 7200;
}

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Remarque : nous avons décidé de laisser certaines IP non utilisées hors de la plage dans le cas où l'on souhaiterait ajouter de nouvelles machines avec des IP fixes plus tard.

Puis nous allons adresser des adresses IP fixes du serveur ainsi que démarrer le service DHCP à chaque démarrage dans le fichier de démarrage rc.local de la machine. Nous tenons à faire remarquer que le serveur a été configuré et était fonctionnel avant l'apparition de la documentation DHCP disponible sur Elearn, nous avons donc été dans l'obligation de chercher ces informations sur internet. Certaines différences sont donc notables mais le fonctionnement est identique.

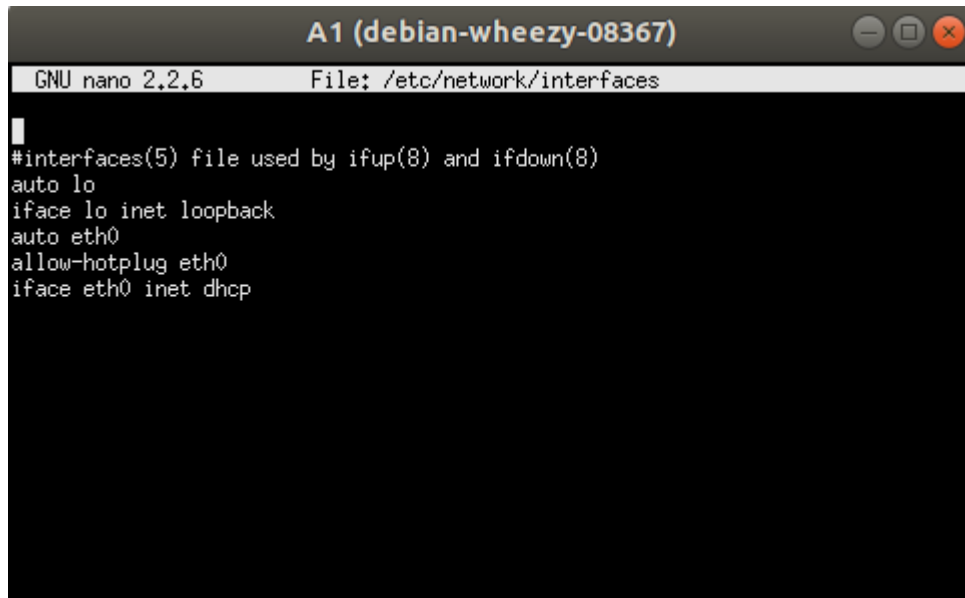


```
GNU nano 2.2.6 File: /etc/rc.local Modified

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth1 192.168.0.66/26
ifconfig eth0 192.168.0.129/26
service isc-dhcp-server start
exit 0
```

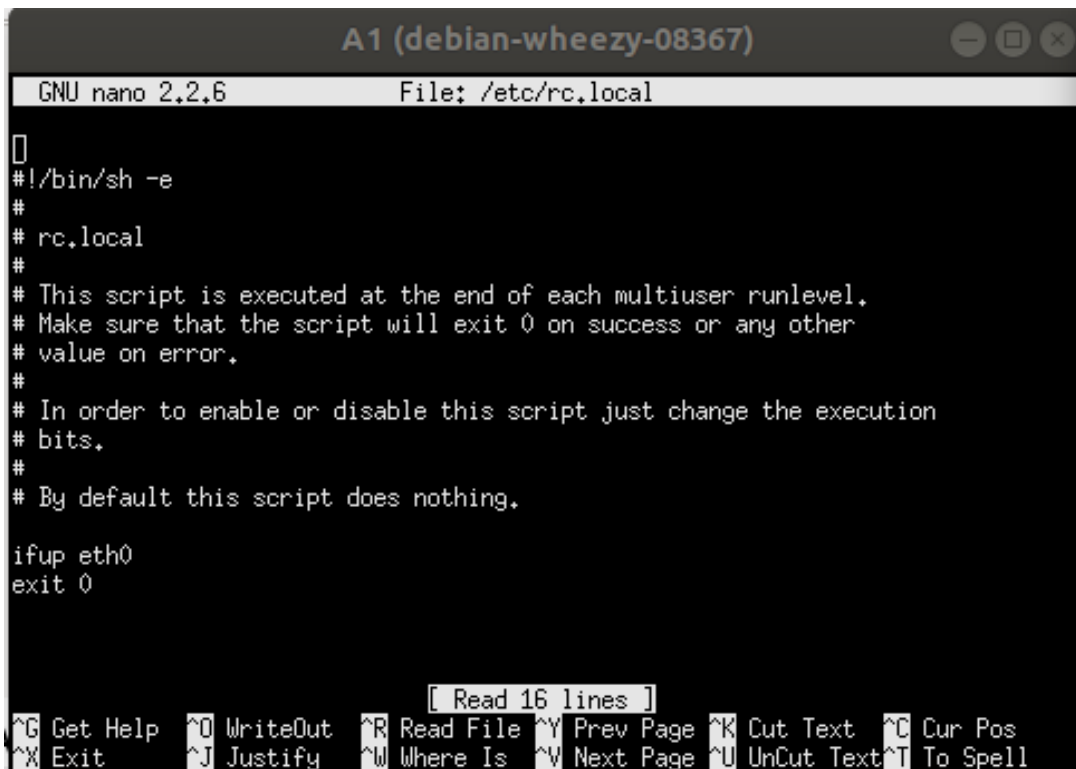
2.2 Les machines administratives

Dans un premier temps nous allons activer la recherche DHCP sur les machines administratives en ajoutant ces prochaines lignes dans le fichier `/etc/network/interfaces`.

A terminal window titled 'A1 (debian-wheezy-08367)' showing the nano 2.2.6 editor editing the file /etc/network/interfaces. The file content is as follows:

```
#interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```

Nous ajoutons ensuite dans le fichier démarrage de la machine la commande d'activation de la carte réseau

A terminal window titled 'A1 (debian-wheezy-08367)' showing the nano 2.2.6 editor editing the file /etc/rc.local. The file content is as follows:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will exit 0 on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

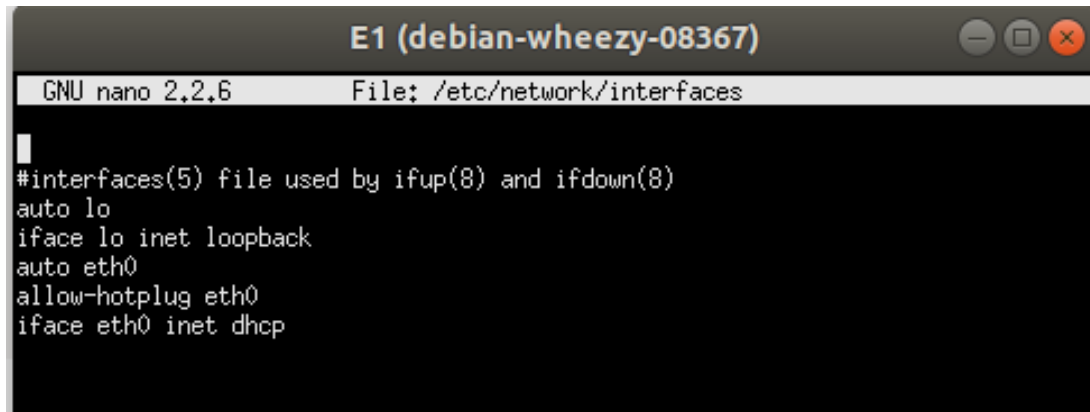
ifup eth0
exit 0
```

At the bottom of the window, there is a status bar with the text '[Read 16 lines]' and a row of keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Page, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where Is, ^V Next Page, ^U UnCut Text, ^T To Spell.

Ces étapes seront répétées pour toutes les autres machines du sous-réseau.

2.3. Les machines entrepôt

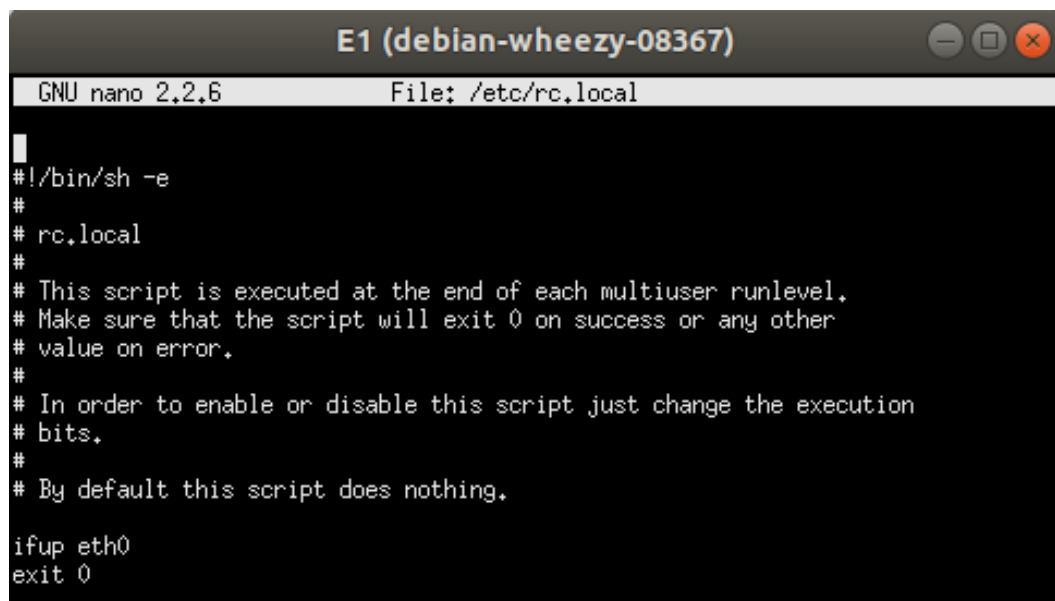
Dans un premier temps nous allons activer la recherche DHCP sur les machines de l'entrepôt en ajoutant ces prochaines lignes dans le fichier `/etc/network/interfaces`.



The screenshot shows a terminal window titled "E1 (debian-wheezy-08367)". The window displays the contents of the file `/etc/network/interfaces` using the GNU nano 2.2.6 editor. The file contains the following text:

```
#interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```

Nous ajoutons ensuite dans le fichier démarrage de la machine la commande d'activation de la carte réseau



The screenshot shows a terminal window titled "E1 (debian-wheezy-08367)". The window displays the contents of the file `/etc/rc.local` using the GNU nano 2.2.6 editor. The file contains the following text:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will exit 0 on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

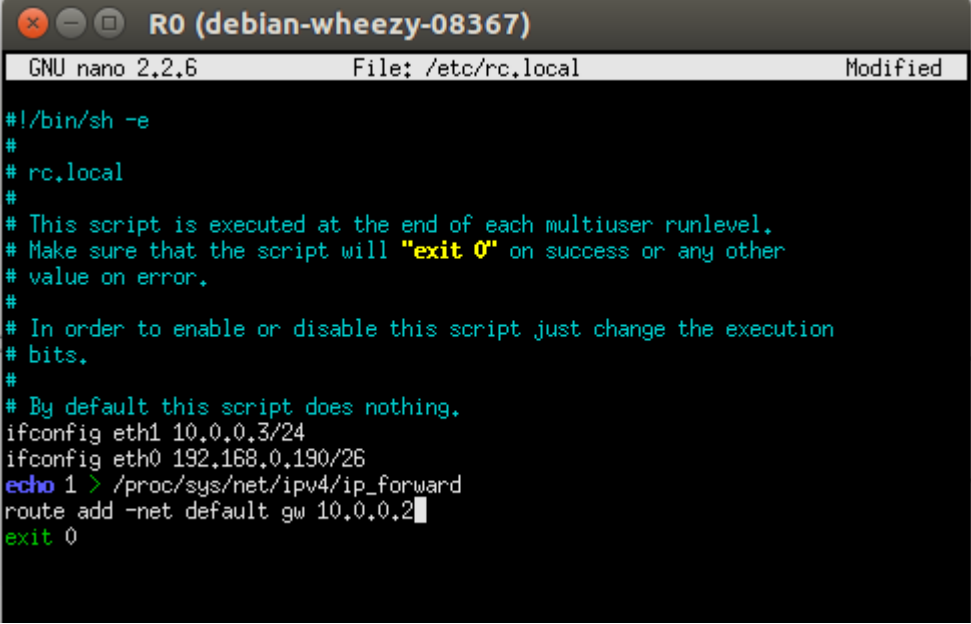
ifup eth0
exit 0
```

Ces étapes seront répétées pour toutes les autres machines du sous-réseau.

2.4 Le routeur R0

Dans le projet *Marionnet* nous utilisons des machines standard avec deux cartes réseau, la principale raison est pour pouvoir s'en servir comme machines espion ainsi que vérifier si les routages sont bon.

Les routeurs n'ont pas beaucoup de configuration nécessaire, il suffit de leurs donner des adresses IP valides qui ne risque pas d'être ré-attribuées par le serveur DHCP.

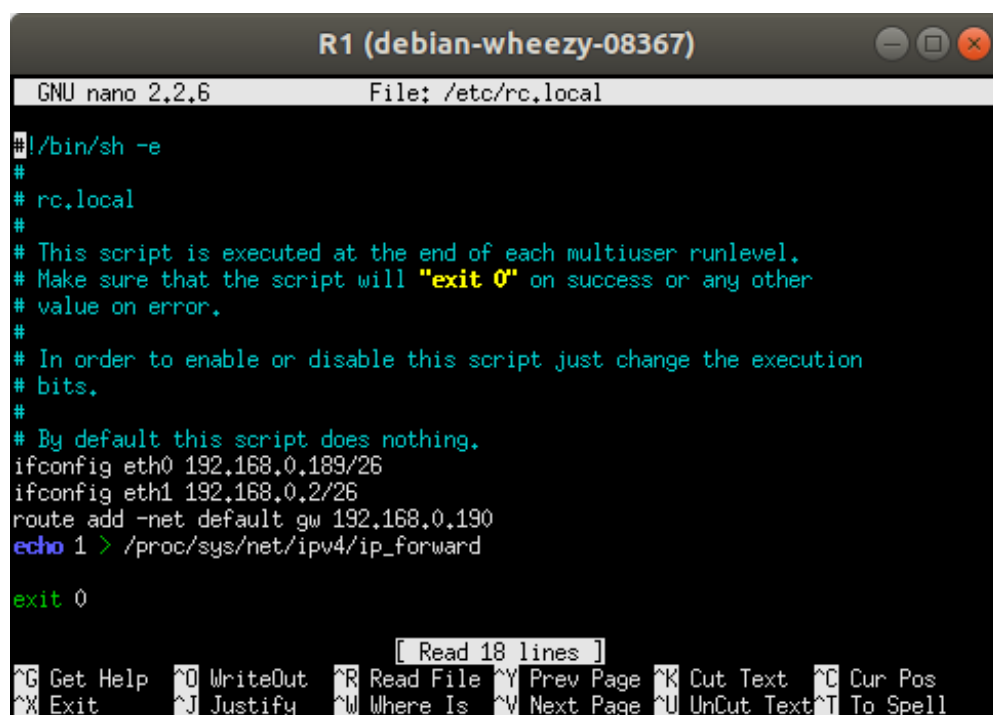


```
R0 (debian-wheezy-08367)
GNU nano 2.2.6      File: /etc/rc.local      Modified
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth1 10.0.0.3/24
ifconfig eth0 192.168.0.190/26
echo 1 > /proc/sys/net/ipv4/ip_forward
route add -net default gw 10.0.0.2
exit 0
```

Comme dit précédemment dans l'explication détaillée, l'interface eth1 permettra d'accéder à internet avec son IP **10.0.0.3**. L'interface eth0 est reliée au réseau administratif. Ces IP étant fixées localement par la machine, il a fallu les configurer à chaque démarrage d'où l'inclusion dans le fichier **rc.local**. De plus nous ajoutons la commande permettant d'activer la fonction de routage de la machine, cette commande n'est pas nécessaire pour chaque démarrage mais nous avons remarqué un phénomène étrange qui re-désactivait cette fonctionnalité donc nous l'exécutons à chaque démarrage par sécurité. De plus étant donné qu'il s'agit du routeur permettant un accès vers internet il nous faut ajouter un routage par défaut dirigeant vers le modem d'adresse **10.0.0.2**.

2.5 Le routeur R1

Le routeur R1 aura lui aussi des adresses IP fixées localement, elles devront donc aussi être reconfigurées à chaque démarrage. En interface eth0, on configure l'adresse 192.168.0.189 concernant le sous-réseau administratif, et l'interface eth1 aura comme adresse 192.168.0.2 concernant le réseau de serveurs. Celui-ci étant le routeur par défaut du réseau administratif. Il doit avoir un routage par défaut vers le routeur R0 chargé du trafic sortant vers le monde extérieur pour permettre aux machines d'avoir accès à internet. Les deux routeurs étant sur le même réseau les premières requêtes seront redirigées.



The screenshot shows a terminal window titled "R1 (debian-wheezy-08367)". The terminal is running the GNU nano 2.2.6 text editor, editing the file /etc/rc.local. The script content is as follows:

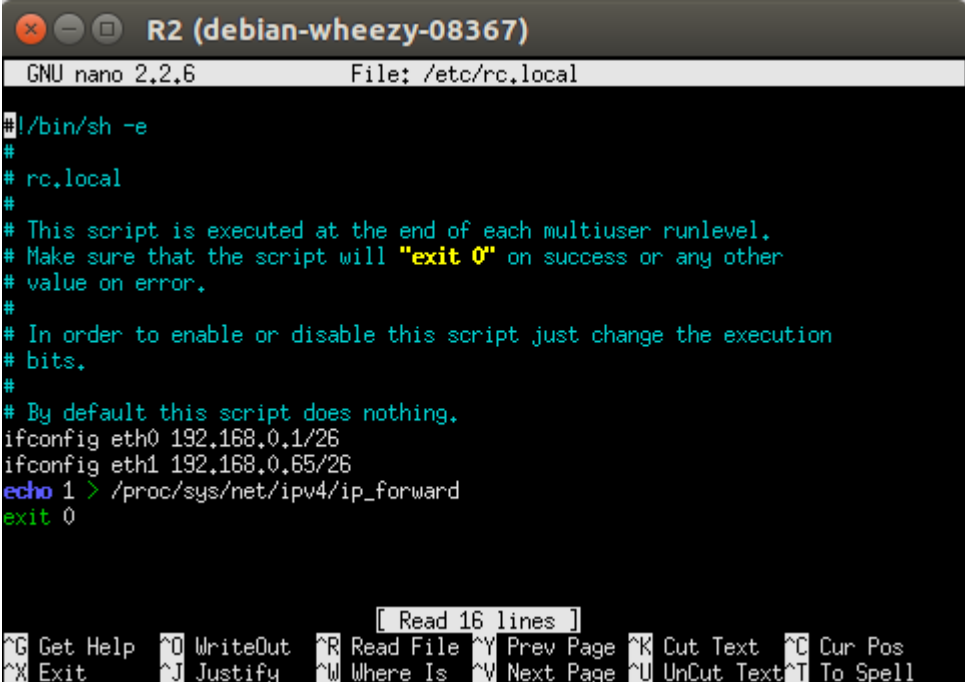
```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth0 192.168.0.189/26
ifconfig eth1 192.168.0.2/26
route add -net default gw 192.168.0.190
echo 1 > /proc/sys/net/ipv4/ip_forward

exit 0
```

At the bottom of the terminal, there is a status bar with the text "[Read 18 lines]" and a list of keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Page, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where Is, ^V Next Page, ^U UnCut Text, ^T To Spell.

2.6 Le routeur R2

Le routeur R2 aura comme adresse sur l'interface eth0 : 192.168.0.1, localisé dans le sous-réseau des serveurs. Sur l'interface eth1, le réseau de l'entrepôt il aura comme adresse 192.168.0.65. De même que les autres routeurs, ces adresses sont assignées localement et ne peuvent pas être attribuées par le serveur DHCP.



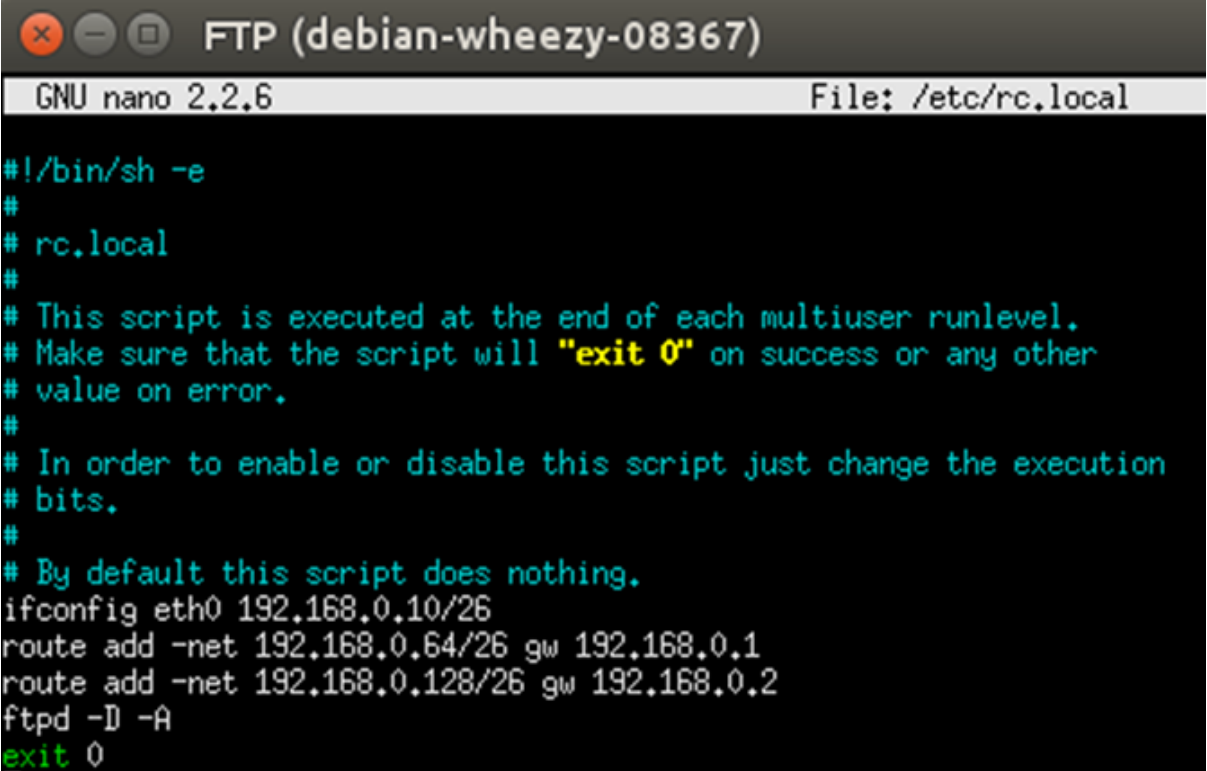
```
R2 (debian-wheezy-08367)
GNU nano 2.2.6      File: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth0 192.168.0.1/26
ifconfig eth1 192.168.0.65/26
echo 1 > /proc/sys/net/ipv4/ip_forward
exit 0

[ Read 16 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^W Next Page ^U UnCut Text ^T To Spell
```

2.7 Le serveur FTP

Le serveur FTP comme son nom l'indique héberge le service de partage de fichier. Son adresse IP est attribuée localement sans crainte de conflit car le serveur DHCP n'accède pas à ce réseau. Par hasard et moyen mémotechnique son adresse sera **192.168.0.10**. Pour répondre aux requêtes des utilisateurs sur les autres réseaux, il doit y avoir deux règles de routage vers lesdits réseaux grâce aux deux routeurs R1 et R2. Le service FTP est démarré à chaque démarrage grâce à la commande **ftpd -D -A**. l'option -D permet de lancer le service en mode daemon facilitant l'utilisation, l'option -A force le mode anonyme et rejette tout autre compte qui essaye de se connecter. Il nous a fallu pour cela créer un nouvel utilisateur du nom de "**ftp**" avec la commande **adduser ftp**. Pour se connecter les utilisateurs ont juste à entrer comme nom :**ftp** et entrer n'importe quoi en mot de passe.

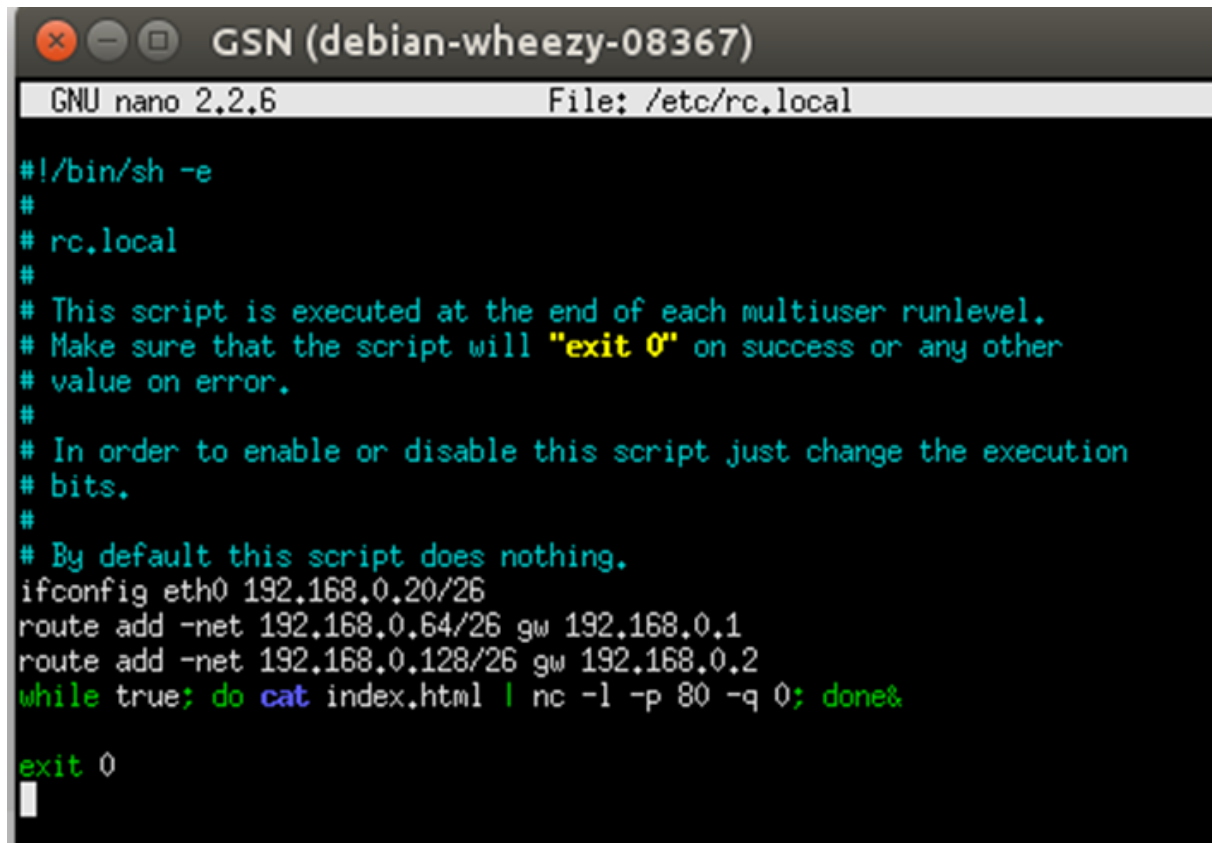


```
FTP (debian-wheezy-08367)
GNU nano 2.2.6 File: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth0 192.168.0.10/26
route add -net 192.168.0.64/26 gw 192.168.0.1
route add -net 192.168.0.128/26 gw 192.168.0.2
ftpd -D -A
exit 0
```

2.8 Le serveur de gestion

Le serveur de gestion héberge un site web d'application de gestion, comme le serveur FTP on lui attribue une adresse localement à chaque démarrage ici **192.168.0.20**. On ajoute les mêmes règles de routage pour répondre aux requêtes venant des deux réseaux.



```
GNU nano 2.2.6      File: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ifconfig eth0 192.168.0.20/26
route add -net 192.168.0.64/26 gw 192.168.0.1
route add -net 192.168.0.128/26 gw 192.168.0.2
while true; do cat index.html | nc -l -p 80 -q 0; done&

exit 0
```

Nous avons ensuite élaboré un fichier HTML `index.html` représentant l'interface de l'application de gestion. Nous utilisons le programme netcat pour transférer cette page, cette méthode n'est pas l'idéal pour héberger un site web mais est en accord avec les consignes données.

Pour cela nous utilisons la commande: `while true; do cat index.html | nc -l -p 80 -q 0; done&`. Cette commande crée une boucle dans laquelle on fait passer le résultat de la commande `cat index.html` qui affiche simplement le contenu du fichier `index.html` dans la commande netcat. La commande netcat va donc écouter (-l) sur le port 80 (-p 80) réservé au protocole HTTP.

Dès qu'une connexion est établie avec un client le programme transfère le contenu reçu en stdin (ici donc le contenu du fichier HTML), une fois le transfert du document terminé la connexion est arrêtée immédiatement par le serveur (-q 0). Notons `&` au marqueur de fin de boucle pour qu'elle se lance en arrière plan son absence gèlera la machine au prochain redémarrage car la fin du fichier `rc.local` ne sera jamais atteinte.

De plus, une fois la connexion terminée par le serveur, un nouveau tour de boucle reprend ce qui permet de relancer le programme netcat permettant de ne même pas avoir à interagir avec le serveur de gestion une fois démarré et une fois une connexion terminée.

3. Tests

3.1 Test DHCP

```
A1 (debian-wheezy-08367)
[0 root@A1 ~]$ ifup eth0
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/02:04:06:06:7a:eb
Sending on   LPF/eth0/02:04:06:06:7a:eb
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 5
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPOFFER from 192.168.0.129
DHCPACK from 192.168.0.129
bound to 192.168.0.137 -- renewal in 292 seconds.
[0 root@A1 ~]$
```

```
E1 (debian-wheezy-08367)
[0 root@E1 ~]$ ifup eth0
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/02:04:06:1b:2f:ae
Sending on   LPF/eth0/02:04:06:1b:2f:ae
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 4
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPOFFER from 192.168.0.66
DHCPACK from 192.168.0.66
bound to 192.168.0.72 -- renewal in 243 seconds.
[0 root@E1 ~]$
```

On remarque donc les différentes requêtes DHCP entre les machines et le serveur DHCP, commençant par une requête **DHCPDISCOVER** visant à chercher le serveur DHCP puis une requête **DHCPREQUEST** demandant une IP disponible. Vient ensuite une requête **DHCPOFFER** venant du serveur DHCP qui lui propose donc une IP valide et disponible. Finalement, la requête **DHCPACK** qui confirme les informations transmises. Nous ajoutons ensuite dans le fichier démarrage de la machine la commande d'activation de la carte réseau du sous-réseau. Ces étapes seront répétées pour toutes les autres machines du sous-réseau

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x448f8f74
2	0.000590000	192.168.0.129	192.168.0.137	ICMP	62	Echo (ping) request id=0x1471, seq=0/0, ttl=64
3	0.011751000	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
4	0.768801000	::	ff02::1:ff06:7aeb	ICMPv6	78	Neighbor Solicitation for fe80::4:6ff:fe06:7aeb
5	1.002983000	192.168.0.129	192.168.0.137	DHCP	342	DHCP Offer - Transaction ID 0x448f8f74
6	1.004921000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x448f8f74
7	1.009045000	192.168.0.129	192.168.0.137	DHCP	342	DHCP ACK - Transaction ID 0x448f8f74
8	2.416180000	fe80::4:6ff:fe06:7aeb	ff02::2	ICMPv6	70	Router Solicitation from 02:04:06:06:7a:eb
9	4.759665000	fe80::4:6ff:fe06:7aeb	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
10	5.015575000	BbnInter_06:7a:eb	BbnInter_06:7a:eb	ARP	42	Who has 192.168.0.137? Tell 192.168.0.129
11	5.015731000	BbnInter_06:7a:eb	BbnInter_06:7a:eb	ARP	42	192.168.0.137 is at 02:04:06:06:7a:eb
12	6.423629000	fe80::4:6ff:fe06:7aeb	ff02::2	ICMPv6	70	Router Solicitation from 02:04:06:06:7a:eb
13	10.428392000	fe80::4:6ff:fe06:7aeb	ff02::2	ICMPv6	70	Router Solicitation from 02:04:06:06:7a:eb

Frame 1: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0
Ethernet II, Src: BbnInter_06:7a:eb (02:04:06:06:7a:eb), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol

3.2 Test de connectivité internet

```
A1 (debian-wheezy-08367)
Bienvenue Serveur HTTP Iparla

E1 (debian-wheezy-08367)
[0 root@E1 ~]$ lynx 10.3.224.215
Looking up '10.3.224.215' first

Looking up 10.3.224.215 first
Looking up 10.3.224.215
Making HTTP connection to 10.3.224.215
Alert!: Unable to connect to remote host.

lynx: Can't access startfile http://10.3.224.215/
[1 root@E1 ~]$

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
Arrow keys: Up and Down to move, Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

```
FTP (debian-wheezy-08367)
[0 root@FTP ~]$ lynx 10.3.224.215
Looking up '10.3.224.215' first

Looking up 10.3.224.215 first
Looking up 10.3.224.215
Making HTTP connection to 10.3.224.215
Alert!: Unable to connect to remote host.

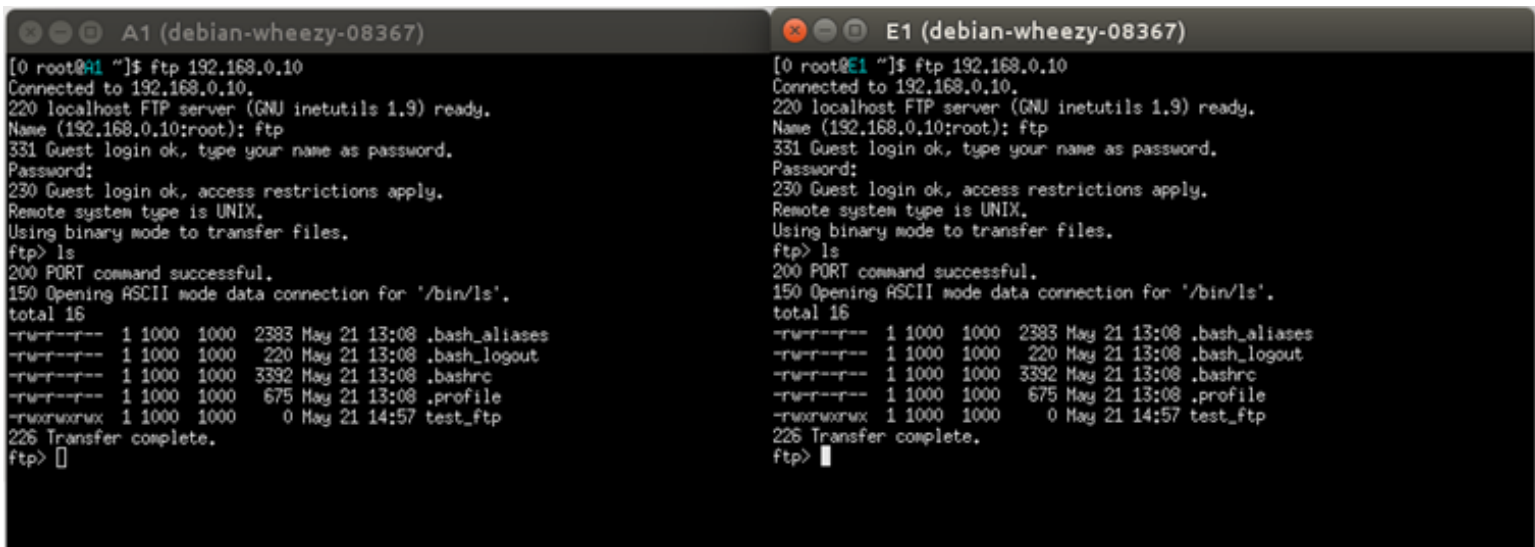
lynx: Can't access startfile http://10.3.224.215/
[1 root@FTP ~]$
```

On confirme donc que seul les machines administrative ont un accès vers le monde extérieur, ici tester avec la connexion à la page web du serveur Iparla

Notons la redirection proposé par le routeur R1 vers le routeur R0 pour le client A1, Le routage pointant vers le même réseau, la communication avec le routeur R1 n'est pas nécessaire.

89	76.904517000	fe80::4:611:fe06:7aeb	1102::2	ICMPv6	70 Router Solicitation from 02:04:06:06:7a:eb
90	78.171364000	BbnInter_f9:5d:b7	Broadcast	AoE	32 Query Config Information Request
91	81.437165000	BbnInter_2d:e9:94	BbnInter_4d:24:d3	ARP	42 Who has 192.168.0.129? Tell 192.168.0.140
92	81.437614000	BbnInter_4d:24:d3	BbnInter_2d:e9:94	ARP	42 192.168.0.129 is at 02:04:06:4d:24:d3
93	89.957604000	BbnInter_06:7a:eb	Broadcast	ARP	42 Who has 192.168.0.189? Tell 192.168.0.137
94	89.957616000	BbnInter_ab:28:81	BbnInter_06:7a:eb	ARP	42 192.168.0.189 is at 02:04:06:ab:28:81
95	89.958326000	192.168.0.137	10.0.0.2	ICMP	98 Echo (ping) request id=0x07df, seq=1/256, ttl=64
96	89.958350000	192.168.0.137	10.0.0.2	ICMP	98 Echo (ping) request id=0x07df, seq=1/256, ttl=63
97	89.959235000	10.0.0.2	192.168.0.137	ICMP	98 Echo (ping) reply id=0x07df, seq=1/256, ttl=254
98	90.962092000	192.168.0.137	10.0.0.2	ICMP	98 Echo (ping) request id=0x07df, seq=2/512, ttl=64
99	90.962129000	192.168.0.189	192.168.0.137	ICMP	126 Redirect (Redirect for host)
100	90.962150000	192.168.0.137	10.0.0.2	ICMP	98 Echo (ping) request id=0x07df, seq=2/512, ttl=63
101	90.963679000	10.0.0.2	192.168.0.137	ICMP	98 Echo (ping) reply id=0x07df, seq=2/512, ttl=254
102	90.964108000	BbnInter_06:7a:eb	Broadcast	ARP	42 Who has 192.168.0.190? Tell 192.168.0.137
103	90.964577000	BbnInter_06:7a:eb	Broadcast	ARP	42 Who has 192.168.0.190? Tell 192.168.0.137

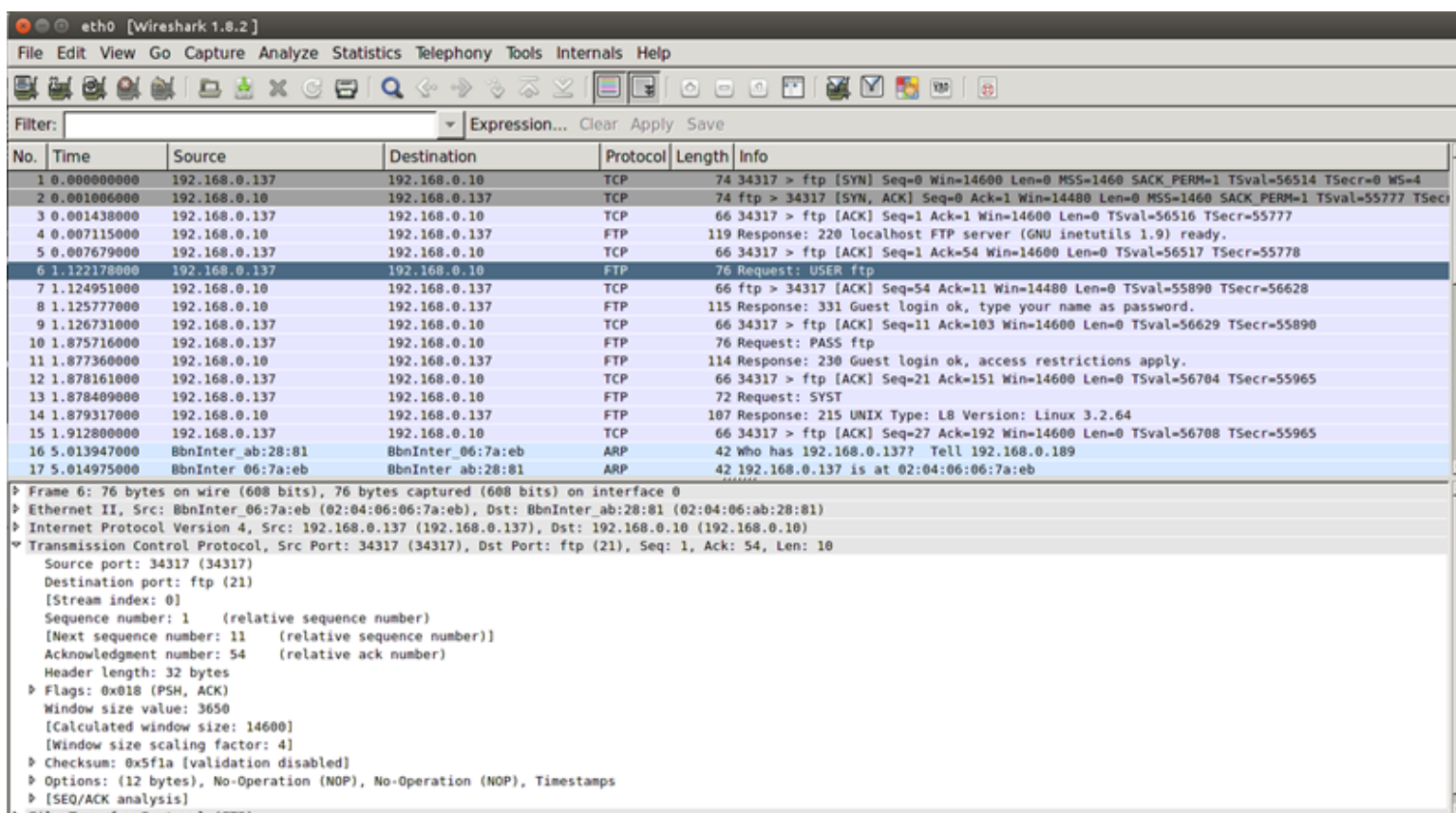
3.3 Test du service FTP



```
[0 root@A1 ~]$ ftp 192.168.0.10
Connected to 192.168.0.10.
220 localhost FTP server (GNU inetutils 1.9) ready.
Name (192.168.0.10:root): ftp
331 Guest login ok, type your name as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 16
-rw-r--r-- 1 1000 1000 2383 May 21 13:08 .bash_aliases
-rw-r--r-- 1 1000 1000 220 May 21 13:08 .bash_logout
-rw-r--r-- 1 1000 1000 3392 May 21 13:08 .bashrc
-rw-r--r-- 1 1000 1000 675 May 21 13:08 .profile
-rwxrwxrwx 1 1000 1000 0 May 21 14:57 test_ftp
226 Transfer complete.
ftp>
```

```
[0 root@E1 ~]$ ftp 192.168.0.10
Connected to 192.168.0.10.
220 localhost FTP server (GNU inetutils 1.9) ready.
Name (192.168.0.10:root): ftp
331 Guest login ok, type your name as password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for '/bin/ls'.
total 16
-rw-r--r-- 1 1000 1000 2383 May 21 13:08 .bash_aliases
-rw-r--r-- 1 1000 1000 220 May 21 13:08 .bash_logout
-rw-r--r-- 1 1000 1000 3392 May 21 13:08 .bashrc
-rw-r--r-- 1 1000 1000 675 May 21 13:08 .profile
-rwxrwxrwx 1 1000 1000 0 May 21 14:57 test_ftp
226 Transfer complete.
ftp>
```

Dans ces captures d'écran on confirme le fonctionnement du service pour les machines administratives ainsi que pour celle de l'entrepôt. On note que le mode anonyme restreint les utilisateurs au répertoire racine de l'utilisateur **ftp** afin d'empêcher l'accès des fichiers qui ne doivent pas être partagés.

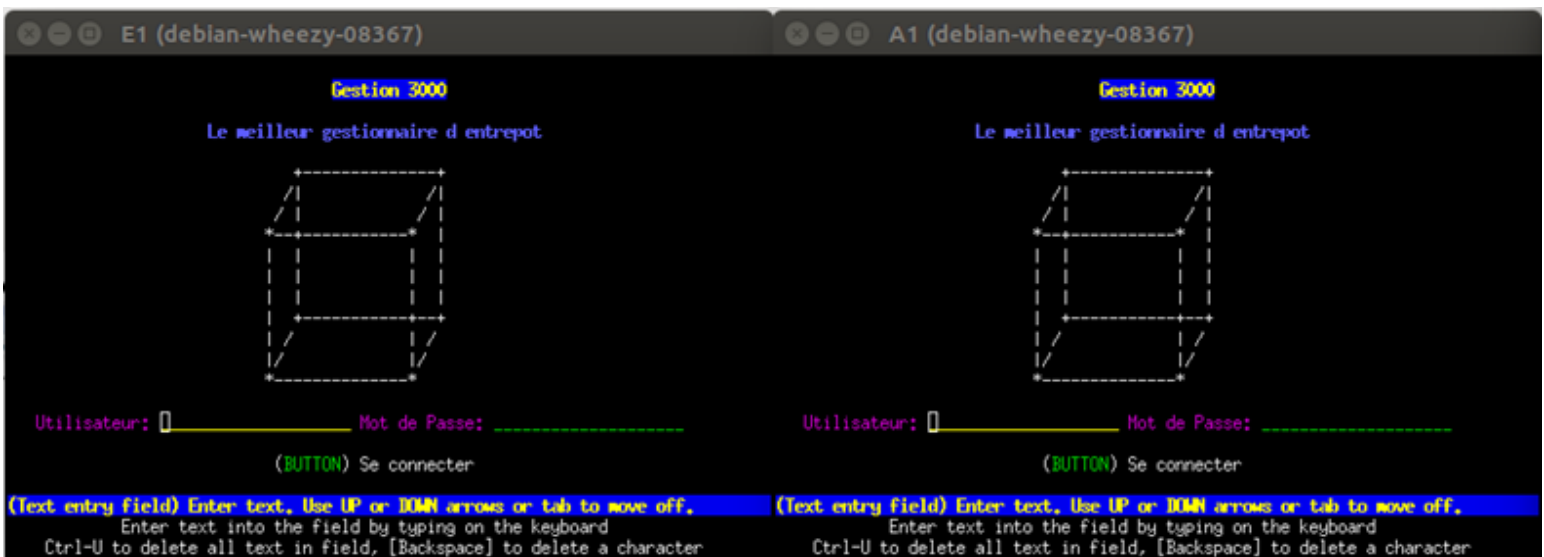


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.137	192.168.0.10	TCP	74	34317 > ftp [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=56514 TSecr=0 WS=4
2	0.001000000	192.168.0.10	192.168.0.137	TCP	74	ftp > 34317 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=55777 TSecr=0
3	0.001430000	192.168.0.137	192.168.0.10	TCP	66	34317 > ftp [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=56516 TSecr=55777
4	0.007115000	192.168.0.10	192.168.0.137	FTP	119	Response: 220 localhost FTP server (GNU inetutils 1.9) ready.
5	0.007679000	192.168.0.137	192.168.0.10	TCP	66	34317 > ftp [ACK] Seq=1 Ack=54 Win=14600 Len=0 TSval=56517 TSecr=55778
6	1.122170000	192.168.0.137	192.168.0.10	FTP	76	Request: USER ftp
7	1.124951000	192.168.0.10	192.168.0.137	TCP	66	ftp > 34317 [ACK] Seq=54 Ack=11 Win=14480 Len=0 TSval=55890 TSecr=56628
8	1.125777000	192.168.0.10	192.168.0.137	FTP	115	Response: 331 Guest login ok, type your name as password.
9	1.126731000	192.168.0.137	192.168.0.10	TCP	66	34317 > ftp [ACK] Seq=11 Ack=103 Win=14600 Len=0 TSval=56629 TSecr=55890
10	1.875716000	192.168.0.137	192.168.0.10	FTP	76	Request: PASS ftp
11	1.877360000	192.168.0.10	192.168.0.137	FTP	114	Response: 230 Guest login ok, access restrictions apply.
12	1.878161000	192.168.0.137	192.168.0.10	TCP	66	34317 > ftp [ACK] Seq=21 Ack=151 Win=14600 Len=0 TSval=56704 TSecr=55965
13	1.878409000	192.168.0.137	192.168.0.10	FTP	72	Request: SYST
14	1.879317000	192.168.0.10	192.168.0.137	FTP	107	Response: 215 UNIX Type: L8 Version: Linux 3.2.64
15	1.912800000	192.168.0.137	192.168.0.10	TCP	66	34317 > ftp [ACK] Seq=27 Ack=192 Win=14600 Len=0 TSval=56708 TSecr=55965
16	5.013947000	BbnInter_ab:28:81	BbnInter_06:7a:eb	ARP	42	Who has 192.168.0.137? Tell 192.168.0.189
17	5.014975000	BbnInter_06:7a:eb	BbnInter_ab:28:81	ARP	42	192.168.0.137 is at 02:04:06:06:7a:eb

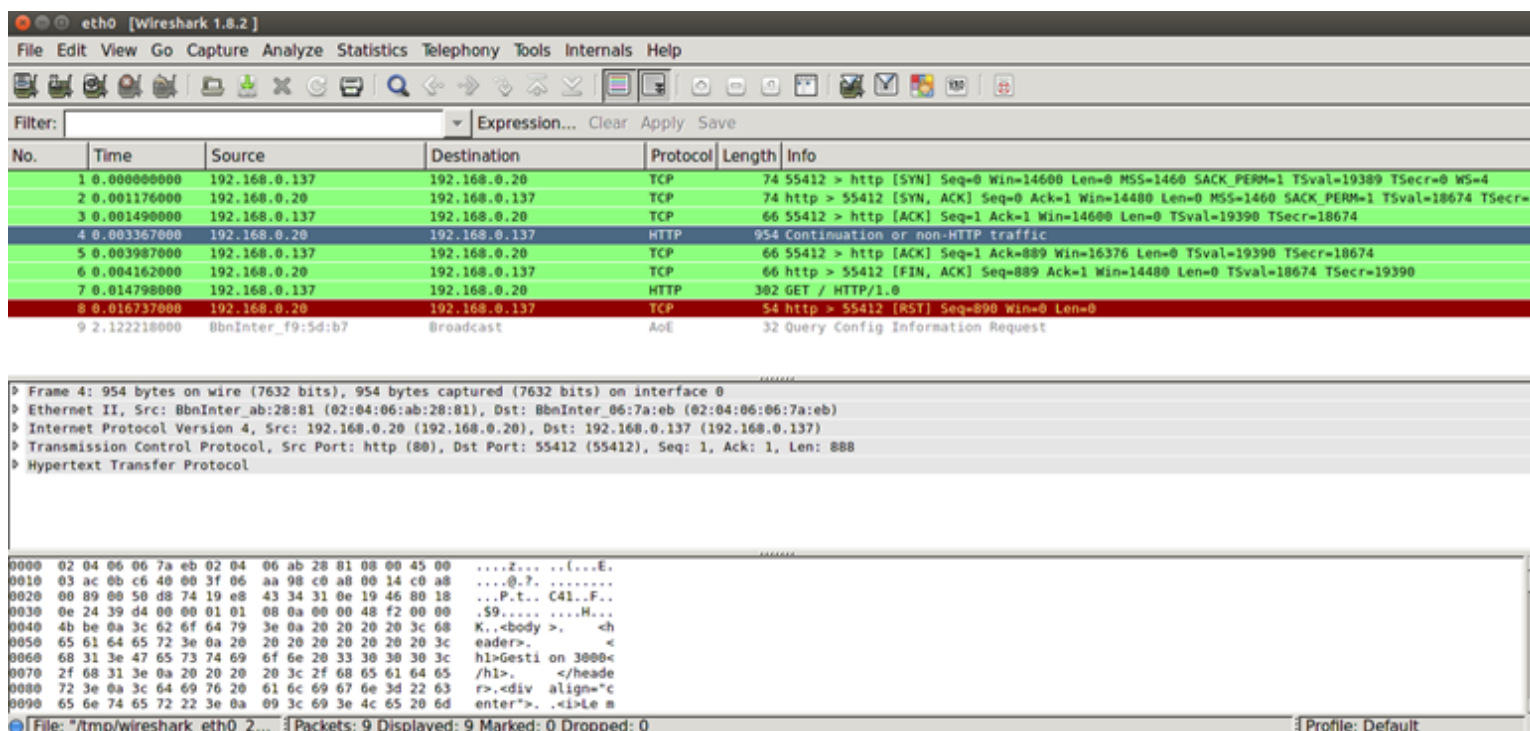
Frame 6: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Ethernet II, Src: BbnInter_06:7a:eb (02:04:06:06:7a:eb), Dst: BbnInter_ab:28:81 (02:04:06:ab:28:81)
Internet Protocol Version 4, Src: 192.168.0.137 (192.168.0.137), Dst: 192.168.0.10 (192.168.0.10)
Transmission Control Protocol, Src Port: 34317 (34317), Dst Port: ftp (21), Seq: 1, Ack: 54, Len: 10
Source port: 34317
Destination port: ftp (21)
[Stream index: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 11 (relative sequence number)]
Acknowledgment number: 54 (relative ack number)
Header length: 32 bytes
Flags: 0x018 (PSH, ACK)
Window size value: 3650
[Calculated window size: 14600]
[Window size scaling factor: 4]
Checksum: 0x5f1a [validation disabled]
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]

Dans cette capture Wireshark de R1 on retrouve les requêtes du protocole de transfert de fichier entre A1 et le serveur FTP , n'étant pas chiffré ces informations sont visibles en clair.

3.4 Test de l'application de gestion



Sur cette capture d'écran, on confirme le bon fonctionnement du service, la simple exécution de la commande **lynx 192.168.0.20** nous affiche directement le contenu grace a l'option -q 0 de netcat.



Cette capture WireShark de R0 met en évidence la communication entre le serveur de gestion et la machine A1, On y retrouve le port http 80 et le port utilisé par lynx 55412. De plus, nous pouvons récupérer le code HTML de la page qui s'affiche car celle-ci n'est pas chiffrée.