

Routeur Raspberry via mobile 4G Free en complément et/ou secours Freebox

Introduction:

L'installation du Raspberry PI 4b 4Go a été réalisée avec le kit complémentaire Bruphny, qui inclut en plus du boîtier entre autres une clef USB support de micro SD Card et un ventilateur 35mm dont les 2 broches d'alimentation sont séparées, permettant ainsi d'alimenter ledit ventilateur en 5v ou 3,3v... Noter que via un mobile Free, seul le trafic ipv4 est possible, donc pas de route ipv6 par le mobile! Par ailleurs, il est préférable de couper le wifi sur le mobile...

- 1) Réaliser la micro SD Card de boot sur le PC (en utilisant éventuellement la clef USB lecteur de micro SD fournie dans le kit) et l'ancrer l'utilitaire de création de la fondation Raspberry: https://downloads.raspberrypi.org/imager/imager_1.4.exe
Si l'on veut n'utiliser le Raspberry que pour cet usage de routeur, sélectionner la version Raspberry-Pi-os mini...
- 2) Si on veut personnaliser la Sd Card, pour activer le SSH directement après le boot
 - a. Réinsérer la Sd Card (ou la clef usb qui la contient) dans le PC
 - b. Dans Windows, créer un fichier vide ssh dans la partition boot (la seule accessible depuis Windows, étant en Fat32, donc pas de problème)
 - c. Si l'on veut disposer directement d'une adresse IP fixe sur le Raspberry directement après l'installation, il faut modifier un fichier "dhcpd.conf" dans la partition rootfs... Étant donné que cette partition est en Ext4, on ne pourra réaliser cette modification que si l'on dispose d'un Linux. Si ce n'est pas le cas, on fera les modifications après l'installation. Ci-dessous, je dispose d'une telle machine sur mon PC dans une VirtualBox
 - d. Lancer la machine virtuelle Linux, ou accéder à un Linux...
 - e. Insérer le lecteur USB de cartes SD, et le monter dans la machine Linux via l'interface Virtualbox (ou non!)
 - f. Attendre que les devices /dev/sdx1 (boot) et /dev/sdx2 (rootfs) soient montés
se mettre sous root par un `sudo -i` en entrant le mot de passe qui va bien
 - g. Pour disposer d'une IP fixe, modifier dans le rootfs le fichier dhcpd.conf. On en profite pour décrire le "device" de la future connexion à l'Mobile, en lui fixant un métrique sympa. Ceci permettra d'activer le routage par défaut avec une priorité sympa sur l'interface Smartphone. Ce routage disparaît automatiquement si on débranche le Phone du Raspberry!
Il est à noter qu'avec les dernières versions de Debian et autre Linux, la gestion réseau est particulièrement dynamique, et il reste très peu de paramétrage à faire...
 - h. Aller dans /media/jacques/boot et créer un fichier ssh

```
cd /media/jacques/boot  
touch ssh
```
 - i. Aller dans /media/jacques/rootfs/etc et modifier le fichier dhcpd.conf

```
cd /media/jacques/rootfs/etc  
vim dhcpd.conf
```

```
interface eth0  
static ip_address=192.168.1.2/24  
static routers=192.168.1.254  
static domain_name_servers=192.168.1.254  
  
interface telmob  
metric 99
```
 - j. Démonter les 2 file-system précédemment montés.

```
umount /media/jacques/boot  
umount /media/jacques/rootfs
```
 - k. Retirer la clef USB, puis la carte SD de la clef.
- 3) Insérer la carte micro SD dans le Raspberry Pi, le connecter au réseau, connecter clavier et souris et l'écran HDMI.
- 4) Mettre le Raspberry sous tension, l'installation de base est lancée, à partir de la Micro SD préalablement réalisée. Lors du premier boot et au début de l'installation, le file-system rootfs est automatiquement agrandi pour occuper la taille maxi possible sur la Sd Card
- 5) A la fin de l'installation, si l'on a un écran HDMI et un clavier connecté en USB sur le Raspberry
 - a. Se loguer sur la machine via le clavier console avec l'utilisateur pi et mot de passe raspberry.
Attention, le clavier console est en British (QWERTY) à la fin du boot!

login : pi
password : raspberry (Attention, clavier toujours en QWERTY, donc mot de passe rqsberry!)

- b. Lancer ensuite rasp-config en mode administrateur, taper :

sudo raspi-config (clavier toujours en QWERTY, donc rqspi)config!)

- c. Dans rasp-config, changer le mot de passe utilisateur par le choix 4 et l3, mettre le bon Time zone et le bon clavier. Pour que le clavier passe en français, il est nécessaire que la manipulation de changement soit faite depuis le clavier console... Donc, un clavier Azerty doit être connecté en USB sur le Raspberry.

- 6) Ensuite upgrader le raspberry par la commande

sudo apt-get update -y && sudo apt-get upgrade -y && sudo reboot

- 7) Installer vim

sudo apt-get install -y vim

- 8) Petite étape obligatoire, si le Smartphone est un Iphone: Installer des packages complémentaires pour que l'Iphone soit "visible" par le Raspberry, particulièrement en ce qui concerne la partie réseau, car sans ces packages, l'Iphone est bien visible en USB, mais invisible avec la commande ip a!

sudo apt-get install -y iddeviceinstaller python-imobiledevice libimobiledevice-utils python-plist ifuse libusbmuxd-tools

- 9) Forçage du routage

- a. Modifier le fichier .etc/sysctl.conf

sudo vim /etc/sysctl.conf

```
# configurer le Rasp pour router IPV4 et IPV6 décommenter les 2 lignes
net.ipv4. ip_forward=1
net.ipv6. conf.all.forwarding=1
```

- b. Prise en compte, lancer la commande

sudo sysctl -p /etc/sysctl.conf

- 10) Une fois les bons utilitaires installés, le mobile prend le nom "eth1" dans ip a. Soit on garde cette appellation, soit on personnalise le device, en créant une règle complémentaire dans udev. Si l'on personnalise, créer un fichier de description de device, qui appellera la connexion de l'mobile "telmob", et non eth1 (pour le fun!). Dans les fichiers de paramétrage suivants, on utilisera en tant que device "telmob".

Si l'on garde eth1, modifier tous les fichiers avec l'utilisation de "telmob" en "eth1", y compris dans dhcpd.conf!

Les 2 lignes dans les règles spéciales UDEV correspondent à un Iphone 6s et un Nokia 5.1. Pour trouver les bons identifiants Vendor et Produit, faire simplement un `sudo lsusb` une fois le Smartphone connecté sur le port USB...

- a. Création de la règle udev spéciale Iphone 5 et 6, Choisir la bonne ligne pour créer le fichier!

Nota: La première ligne est pour un Iphone 6S, et la deuxième pour un Nokia 5.1 (2 smartphones à la maison...)

sudo vim /etc/udev/rules.d/80-mobile.rules

```
SUBSYSTEM=="net", ACTION=="add", ATTRS{idVendor}=="05ac", ATTRS{idProduct}=="12a8", NAME="telmob"
SUBSYSTEM=="net", ACTION=="add", ATTRS{idVendor}=="08ed", ATTRS{idProduct}=="2004", NAME="telmob"
```

- b. Valider la nouvelle règle en faisant

sudo udevadm control -R ou sudo udevadm trigger

- c. Et contrôler par

sudo udevadm test /dev/net

- 11) Forçage POSTROUTING pour mobile dans iptable

Là, on fait un peu dans le "moche", en forçant dans les iptables en fin de boot le POSTROUTING telmob en créant sur /home/pi un fichier "reseau.sh" pour activer le masquering vers le réseau mobile au boot. Ce POSTROUTING restera en permanence dans iptables. Cette règle iptables sera transparente lorsque le mobile est déconnecté, le routage par défaut du mobile étant automatiquement annulé lorsque ce dernier est déconnecté du Raspberry... La gestion réseau "moderne" façon utilisateur facilite bien les choses!

- a. Le fichier est placé dans /home/pi par commodité. Il comprend un test seulement utile si l'on exécute le script en manuel...

sudo vim /home/pi/reseau.sh

```
#!/bin/bash
#
if !( iptables -t nat -nvL POSTROUTING | grep -q telmob ); then
```

```

        iptables -t nat -A POSTROUTING -o telmob -j MASQUERADE
    fi
#

```

- b. Rendre le fichier exécutable

```
sudo chmod ugo+x /home/pi/reseau.sh
```

- c. Modifier le fichier /etc/rc.local, en ajoutant la commande avant le "exit 0" pour exécuter le fichier /home/pi/reseau.sh

```
sudo vim /etc/rc.local
```

12) Mise en place d'un server DHCP à l'ancienne!

- a. Installer le serveur DHCP

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

- b. Modifier le fichier conf de isc-dhcp-server, ne pas activer pour l'IPv6

```
sudo vim /etc/default/isc-dhcp-server
```

```

INTERFACESv4= "eth0"
INTERFACESv6= ""

```

- c. Editer le fichier /etc/dhcp/dhcpd.conf

```
sudo vim /etc/dhcp/dhcpd.conf
```

```

# dhcpd.conf
# Configuration file for ISC dhcpd
#
default-lease-time 600;
max-lease-time 7200;
# have support for DDNS.
ddns-update-style none;
authoritative;
option domain-name-servers 192.168.1.2, 192.168.1.254;
option broadcast-address 192.168.1.255;
subnet 192.168.1.0 netmask 255.255.255.0 {
    option domain-name "jfu-domain.net";
    range 192.168.1.106 192.168.1.120;
}
option routers 192.168.1.2;

```

- d. Intégrer des baux statiques à dhcpd.conf si nécessaire

- e. Redémarrer le serveur DHCP

```
sudo service isc-dhcp-server restart
```

- f. En passant, arrêter le service DHCP s'il y en a un autre qui "joue" sur une autre machine, par exemple sur la box.

- g. Démarrer une machine en DHCP et tester si le serveur 1.1.1.1 est joignable, le routage par défaut se fera au travers du Raspberry, qui renvoie par défaut vers la box

```
ping -4 1.1.1.1
```

13) Là, on peut connecter le mobile. Il sera alors reconnu par le Rasp. Il faut alors autoriser la connexion sur le mobile au Raspberry

14) Vérifier la présence du mobile sur le Raspberry en faisant:

```
lsusb
```

```
ip a
```

Normalement, le Mobile apparaît dans la liste des devices IP, en tant que telmob, si la règle udev créée à été prise en compte.

Si le mobile apparaît en tant que eth1, on peut réaliser un reboot, qui prendra en compte la règle udev créée

15) On vérifie que depuis le PC en forçant son paramétrage IP, on va vers l'une ou l'autre gateway, selon que l'Mobile est connecté ou pas !

16) On pousse le bouchon plus loin, en installant bind, pour disposer d'un serveur DNS cache...

- a. Sur le Raspberry en version "mini", pas de dnsmasq, sinon, éventuellement désinstaller dnsmasq avant!

```
sudo apt-get install bind9 bind9-doc dnsutils
```

- b. Modifier le fichier options /etc/bind/named.conf.options

```
sudo vim /etc/bind/named.conf.options
```

```
dnssec-validation auto;
auth-nxdomain no;
listen-on-V6 { any; };
allow-query { internals; };
allow-transfer { none; };
allow-recursion { internals; };
forwarders { 192.168.1.254; 1.1.1.1; };
```

- c. Modifier le fichier /etc/bind/named.conf.local

```
sudo vim /etc/bind/named.conf.local
```

```
acl internals { 127.0.0.0/8; 192.168.1.0/24; };
```

- d. Avant de redémarrer, vérification avec named-checkconf

```
sudo named-checkconf
```

Relancer bind par un

```
sudo service bind9 restart
```

- 17) Couper complètement le bluetooth et le Wifi sur le Raspberry: On limite la consommation!

```
echo "dtoverlay=disable-wifi" | sudo tee -a /boot/config.txt
echo "dtoverlay=disable-bt" | sudo tee -a /boot/config.txt
```

- 18) Vérifier que depuis le PC, on a accès au net via la 4G Mobile

```
ping -4 1.1.1.1
ping -4 google.fr
traceroute -d -4 www.sncf.com
accéder à www.franboise314.fr via firefox...
```

- 19) Une fois tout ceci fait et que tout est OK, il est temps de penser à la santé de la SD Card! Pour cela, réduire drastiquement les écritures sur cette dernière, en déplaçant en RAM les répertoires très mis à jour, hors du boot, on sauvegarde /etc/fstab

```
sudo cp -p /etc/fstab /etc/fstab_sav
```

- 20) On modifie /etc/fstab en ajoutant les lignes suivantes, puis on reboote!

```
sudo vim /etc/fstab
```

```
tmpfs /tmp tmpfs defaults,noatime,nosuid,size=64m 0 0
tmpfs /var/tmp tmpfs defaults,noatime,nosuid,size=64m 0 0
tmpfs /var/log tmpfs defaults,noatime,nosuid,mode=0755,size=64m 0 0
```

On peut noter que l'autre arborescence mise à jour fréquemment est /run... Cette arborescence est déjà en file-system tmpfs, donc pas d'écriture sur la SD Card...

- 21) Pour l'utilisation en tant que routeur, pas besoin de swapfiles dans /var, on le désactive d'abord, puis on le supprime..

```
dphys-swapfile swapoff
dphys-swapfile uninstall
```

- 22) On peut compléter l'installation par un couple clef-privée – clef publique pour le SSH...

- a. Il est préférable de se loguer sur le raspberry via putty avec l'utilisateur pi et le bon mot de passe
Pour avoir une directory .ssh avec les bons droits et des clefs qui vont bien, utiliser ssh-keygen.

```
ssh-keygen -t ed25519
```

Puis créer dans .ssh le fichier authorized_keys avec les mêmes droits que "id_ed25519.pub"

- b. Récupérer le fichier "id_ed25519", faire

```
view .ssh/id_ed25519
```

- c. Côté Windows, sélectionner à la souris la totalité du contenu de id_25519, et sauvegarder dans un fichier texte sur le bureau.
Côté Raspberry, quitter le view, et réaliser un vim de authorized_keys

```
vim .ssh/authorized_keys
```

- d. Lancer ensuite Puttygen en choisissant "Conversions", puis Import dans la barre de menus. Choisir alors le fichier sur le bureau, Ensuite, éventuellement changer la passphrase et sauvegarder par "Save private key". La clef privée est sauvegardée au format Putty.
Sauvegarder selon la formule habituelle du copié-collé la clef publique dans /home/pi/authorized_keys du raspberry.
Sauvegarder le contenu d'authorized_keys...