

FreeBSD et réseau WIFI

Jean-Marc LICHTLE *

4 août 2005

Table des matières

1 Introduction	1
1.1 Préambule	1
1.2 Objectif	1
2 Choix des matériels	2
3 Réalisation	2
3.1 Schéma	2
3.2 Installation de FreeBSD sur le P75	2
3.3 Paramétrage du boîtier WIFI	3
3.4 Wifi avec FreeBSD 6.0 et carte SITECOM WL-115	3
3.4.1 Premiers essais	3
3.4.2 Paramétrage du boîtier BELKIN	4
3.4.3 Paramétrage de la carte SITECOM	4
3.4.4 Tests divers	5
4 Conclusion, remerciements	6

Résumé

Utilisation de FreeBSD pour réaliser un routeur entre un réseau ethernet + wifi et une liaison ADSL.

1 Introduction

1.1 Préambule

On se reportera utilement aux articles que j'ai rédigé sur certains thèmes connexes comme :

- Le routage sous Linux et FreeBSD
 - FreeBSD et l'ADSL
 - FreeBSD, des premiers pas à l'utilisation avancée
- documents disponibles sur le site du Mirabellug, le lug de NANCY le pays de la mirabelle.

1.2 Objectif

L'objectif est de décrire la configuration d'un vieux P75 qui doit prendre en charge :

- L'établissement de la connexion ADSL via un modem ethernet ALCATEL Speedtouch.
- Le routage vers un réseau ethernet existant câblé sur un hub NETGEAR.
- Le routage vers un réseau wifi au moyen d'une borne d'accès BELKIN.

* Ingénieur Arts et Métiers promotion CH73

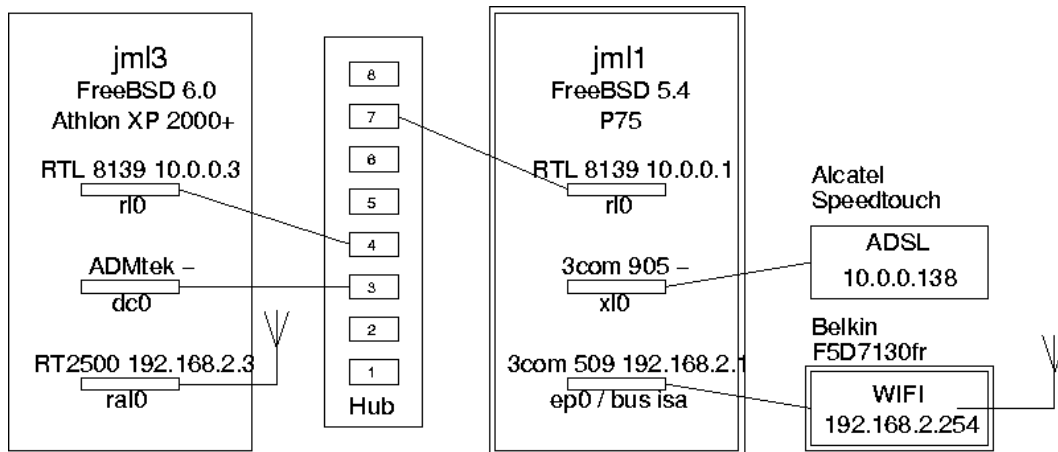


FIG. 1 – Schéma objectif

2 Choix des matériels

J'ai fait de nombreux essais à l'époque où je tournais encore sur FreeBSD 5.3 ou 5.4 sans jamais obtenir de résultat probant. Ma carte NETGEAR WG 311 était malheureusement équipée d'une puce de seconde génération non compatible avec le module existant, ma seconde carte, une SITECOM WL-115 basée sur une puce Ralink 2500 n'était pas mieux reconnue. J'aurais certainement pu compiler un module avec ndis mais les quelques essais que j'ai fait n'ont pas abouti, probablement parce que mon manque d'expérience en matière de wifi ne m'ont pas permis de reconnaître que je suis peut-être passé à deux doigts de la solution. Comme j'avais de toutes façons l'utilisation de ces cartes pour d'autres machines j'ai donc pris le parti de changer mon fusil d'épaule et, plutôt que d'essayer de trouver une carte compatible FreeBSD, je me suis procuré un boîtier faisant la passerelle ethernet-wifi. La suite décrit donc

- La mise en oeuvre d'un boîtier BELKIN F5D7130fr pour en faire le point d'accès d'un réseau BSS (mode infrastructure).
- Partant de ce point d'accès la configuration de la carte SITECOM sous FreeBSD 6.0.

Le coût d'un boîtier BELKIN est certes plus élevé que celui d'une carte simple, environ le double, mais cette solution offre l'avantage d'être, de façon certaine, compatible avec tous les systèmes d'exploitation de la terre à condition qu'ils supportent une carte ethernet quelconque, de préférence 100 M, et qu'ils disposent d'un navigateur. On pourra se procurer assez facilement ces deux matériels, le boîtier BELKIN est vendu par certaines enseignes spécialisées en matériels de bureau / informatique, la carte SITECOM a voisiné un moment dans mon caddie à côté d'une botte de carotte et d'un pack d'eau minérale.

3 Réalisation

3.1 Schéma

Plus qu'un long discours la figure 1 page 2 permet de se rendre compte immédiatement de l'objectif poursuivi. Le P75 jml1 doit être configuré pour servir de routeur entre le réseau ethernet existant, qui comporte d'autres machines que celles figurant sur le schéma, le réseau wifi à créer, et le modem ADSL existant.

3.2 Installation de FreeBSD sur le P75

N'ayant pas besoin de la toute dernière version de FreeBSD (6.0 à l'heure où j'écris ces lignes) j'ai choisi d'installer simplement un 5.4 depuis les CD fournis par la revue LINUX+extra de juillet 2005. Le déroulement de l'installation est classique à deux détails près :

- Ce *!# de BIOS étant vétuste j'ai employé une disquette Smart Boot Manager pour lancer l'installation directement depuis le CD ce qui est bien plus simple et plus universel que la création de disquettes de boot FreeBSD décrite dans les documents "officiels".
- Je n'ai évidemment pas installé d'interface graphique, les 64 Mo de RAM et le disque de 800 Mo n'étant pas vraiment suffisants pour faire tourner efficacement le serveur graphique au demeurant inutile dans ce genre d'application.

Parmi les points auxquels il faut penser j'ai consigné ceux-ci :

- Penser à installer bash, vim, sudo et les autres softs que vous avez l'habitude d'employer comme éventuellement midnight commander ou autre.
- Créer un compte utilisateur (jml dans mon cas) pour les tâches qui ne nécessitent pas de compte administrateur. En pratique ce compte servira surtout à donner un point d'entrée via ssh depuis le PC jml3. Penser éventuellement à fixer le shell par défaut à /usr/local/bin/bash lors de la création du compte.
- Permettre à cet utilisateur de passer root par su. Pour cela il faut que jml fasse partie du groupe wheel. Éditer /etc/group pour ajouter jml à la ligne wheel :* :0 :root et obtenir ainsi wheel :* :0 :root,jml. L'objectif est ici de pouvoir faire, à terme, toute l'administration de jml1 depuis jml3 sans être contraint d'accéder à jml1 physiquement. On peut donc faire disparaître ce dernier sous un table, dans un placard etc.
- Créer un autre compte utilisateur (lambda) non privilégié. Ce compte aura pour but de permettre des connexions via telnet (si vous avez des machines qui utilisent le système d'exploitation de Redmond). Une fois ce point réglé modifier le fichier /usr/local/etc/sudoers pour permettre à cet utilisateur d'arrêter le routeur, pour cela ajouter une ligne lambda ALL=/sbin/init à la fin du fichier. La finalité de ce compte est de permettre à tout utilisateur de l'un des PC du réseau d'arrêter le P75 en fin de journée en tapant sudo init 0 dans une fenêtre telnet ou ssh. Dans le cas de mon installation en effet le P75 n'est pas vraiment accessible, rangé sous un bureau, sans clavier ni écran.
- Donner la possibilité de se connecter via telnet en décommentant les deux lignes de /etc/inetd.conf relative à telnetd.

Après cette installation il va falloir configurer la prise en charge du modem ADSL. Voir le document cité en introduction que j'ai consacré à ce sujet et qui a fait l'objet d'une toute récente mise à jour. Je résume ci-dessous les étapes de cette configuration :

- Configurer la carte ethernet faisant l'interface avec le modem en ajoutant une ligne dans /etc/rc.conf.
- Configurer le chargement des modules nécessaires via /boot/loader.conf
- Modifier /etc/ppp/ppp.conf pour assurer la prise en charge de la connexion.
- Paramétrer la résolution de noms (/etc/resolv.conf).
- Arrivé à ce stade tester la connexion (et la translation d'adresse).
- Paramétrer /etc/rc.conf pour assurer une connexion automatique au boot.

3.3 Paramétrage du boîtier WIFI

Le paramétrage du boîtier se fait très simplement en connectant celui-ci à une carte ethernet d'un PC faisant tourner un navigateur quelconque et en chargeant la page d'accueil du serveur web intégré au boîtier à l'adresse 192.168.2.254. Depuis cette page d'accueil vous avez la possibilité de modifier l'adresse par défaut du boîtier, les paramètres de sécurité etc. Voir à ce sujet la documentation, bien faite, fournie avec le boîtier. Ce qu'il m'importe de souligner ici est que, dès sa sortie d'emballage et sous réserve que les connexions soit faites correctement comme figuré sur le schéma 1, avec une carte ethernet configurée à la bonne adresse (à choisir dans la gamme 192.168.2.1 - 192.168.2.253, vous devriez obtenir un fonctionnement correct même si à ce stade vous partagez votre connexion ADSL avec toute personne du voisinage qui détecterait la présence de votre point d'accès.

3.4 Wifi avec FreeBSD 6.0 et carte SITECOM WL-115

3.4.1 Premiers essais

Ma machine principale fait tourner actuellement FreeBSD 6.0. Cette version présente cet avantage de disposer d'un module prenant en charge la puce Ralink 2500 qui équipe la carte SITECOM WL-115 notée ral0 sur le schéma. Toujours dans une utilisation tout à fait basique il suffit que je tape ifconfig

ral0 192.168.2.3 pour que la connexion wifi s'établisse. J'ai donc, avec le schéma 1 une double possibilité d'accéder à la liaison ADSL, soit en passant par la carte ethernet, soit en utilisant la carte WL-115. J'utilise l'une ou l'autre en fonction des routes que j'ai fixé. En clair, dans mon schéma, je dois taper :

- route add default 10.0.0.1 si je veux passer via l'ethernet câblé,
- route add default 192.168.2.1 si je veux passer via la carte wifi.

L'intérêt ? Je suis un épouvantable tyran et je peux donc, en tirant la prise du boîtier, couper la communication Internet des gosses s'ils me mangent trop de bande passante..... Je passe alors via le réseau câblé.

3.4.2 Paramétrage du boîtier BELKIN

Le problème essentiel posé par l'utilisation basique décrite ci-dessus est que le point d'accès permet de pénétrer le réseau depuis n'importe quelle machine équipée d'une carte wifi et située dans la zone de propagation des ondes. La première chose à faire est donc de mettre en place le codage wep lequel va constituer une première barrière limitant les risques d'intrusion. Deux longueurs de code sont proposées, sur 40 et 104 bits. Accessoirement on pourra changer le SSID (le nom de réseau), la gamme d'adresses IP employées sur le réseau etc. Concernant les réglages à effectuer sur le boîtier BELKIN je vous renvoie à la documentation du constructeur laquelle est tout à fait claire sur ce sujet. Je développerais ici le réglage de la carte SITECOM pour qu'elle se connecte avec le boîtier BELKIN sur lequel j'ai choisi au préalable d'effectuer les réglages suivants :

- Changement du SSID par défaut, "belkin54g" en "rezo"
 - Mise en place d'une clef wep de 40 bits soit 10 caractères hexa, par exemple 12345abcdef.
- Il va sans dire que le SSID et la clef ne sont pas exactement ceux que j'ai utilisé, inutile de parcourir les routes de France un portable sur les genoux pour essayer de pénétrer mon rezo !

3.4.3 Paramétrage de la carte SITECOM

Curieusement pour une utilisation sans protection comme décrite plus haut le fonctionnement de la carte SITECOM est satisfaisant. Dès qu'on en arrive à utiliser le codage wep on se rend compte que les choses vont moins bien et qu'il faut charger à la main le module qui prend en charge le traitement du wep, wlan_wep. Un petit

```
# kldload wlan_wep
```

sous compte root et tout rentre dans l'ordre. Ce chargement peut être automatisé en ajoutant une ligne "kldload wlan_wep" à la fin du fichier /etc/rc.conf.

Moyennant le chargement du module wlan_wep la mise en réseau de la carte SITECOM se fait simplement via la syntaxe suivante :

```
ifconfig ral0 192.168.2.3 ssid rezo wepkey 0x12345abcdef weptxkey 1 wepmode on
```

A partir de ce moment un

```
# ifconfig ral0
```

vous produira un affichage semblable à celui-ci :

```
ral0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet6 fe80::20c:XXXX:XXXX:XXXX%ral0 prefixlen 64 scopeid 0x3
inet 192.168.2.3 netmask 0xfffff00 broadcast 192.168.2.255
ether 00:0c:f6:XX:XX:XX
media: IEEE 802.11 Wireless Ethernet autoselect (OFDM/54Mbps)
status: associated
ssid rezoXXXX channel 1 bssid 00:11:50:XX:XX:XX
authmode OPEN privacy ON deftxkey 1
wepkey 1:40-bit <XXXXXXXXXX>
txpowmax 100 protmode CTS bintval 100
```

dans lequel les "X" sont bien naturellement des masques. Vous pouvez vérifier très simplement en faisant un ping sur 192.168.2.254 que la liaison wifi est établie. A cet endroit il convient de faire plusieurs remarques :

- Je n'ai pas réglé le channel qui reste affiché à 1. Le fonctionnement est toutefois correct alors que le boîtier BELKIN est resté sur son réglage par défaut, le channel 11. Nous sommes sur un réseau en mode infrastructure et c'est donc le point d'accès qui fixe le canal. Il en serait tout autrement sur un réseau ad-hoc.
- Le mot important ici est "associated". Si vous faite l'impasse sur "wepmode 1" vous obtenez également un affichage "associated" et pourtant les pings ne fonctionneront pas, signe que ce paramètre est indispensable. Par cette syntaxe on précise en fait qu'on veut utiliser la première clef wep du boîtier BELKIN lequel autorise quatre clefs de 40 bits.
- La syntaxe de la ligne de commande mentionnée plus haut est la plus simple que j'ai trouvée et qui soit fonctionnelle. On pourrait toutefois raccourcir encore en utilisant d'autres mots clefs que ifconfig prend en compte par souci de compatibilité avec les autres BSD. On y trouve des raccourcis assez surprenant et qui sont très bien documentés dans la page man de ifconfig.

Le paramétrage du boîtier BELKIN lui permet, en décochant la case "Broadcast SSID", de rester discret et de ne pas se manifester par l'envoi périodique d'informations le concernant sur les ondes. Dans ces conditions je ne suis pas arrivé à faire fonctionner la carte SITECOM qui ne voulais pas "s'accrocher" au réseau. Pour être plus précis j'ai obtenu quelques connexions correctes en mettant la carte en mode promisc et en attendant qu'une trame venant fortuitement d'un autre PC vienne débloquent la situation. Dès qu'une trame était reçue je voyais s'afficher sur la console un message m'indiquant que la carte venait de passer à l'état UP, ce qui est paradoxal puisque la carte était censée être UP avant cette démarche, du moins c'est ce que me confirmait ifconfig. Il reste donc un point à analyser ici.

Bien entendu vous avez la possibilité de mémoriser le paramétrage de la carte ral0 dans /etc/rc.conf en y ajoutant une ligne :

```
ifconfig_ral0="192.168.2.3 etc.."
```

3.4.4 Tests divers

Les logiciels de paramétrage de cartes wifi disponibles sur le système d'exploitation édité à Redmond (USA) offrent un certain nombre de fonctionnalités permettant de tester la qualité de la connexion établie, de chercher les réseaux présents dans le voisinage etc. De telles possibilités sont également présentes sur LINUX. J'ai testé ainsi la carte SITECOM avec KNOPPIX 3.9, elle est reconnue sans aucun problème de façon totalement automatique et le logiciel kwifimanager offre des fonctionnalités tout à fait au niveau de ce qu'offre le système de Redmond.

Sans recourir à ces programmes, certes très bien faits, on peut obtenir des renseignements tout aussi précis directement en ligne de commande. Pour "sniffer" les réseaux des voisins taper :

```
# ifconfig ral0 scan
```

On obtient après une courte pause l'affichage suivant :

SSID	BSSID	CHAN	RATE	S:N	INT	CAPS
rezoXXXX	00:11:50:XX:XX:XX	11	54M	96:0	100	EP
NETGEAR	00:0f:b5:XX:XX:XX	07	54M	40:0	200	ES

On y retrouve le réseau managé par la borne BELKIN (rezoXXXX) ainsi qu'un réseau inconnu, probablement un voisin.

Pour obtenir des renseignements précis sur l'état de la carte SITECOM il suffit de taper :

```
# wicontrol -i ral0
```

Ce à quoi la machine répond par :

```
NIC serial number: [ ]
Station name: [ jml3.XXXX ]
SSID for IBSS creation: [ rezoXXXX ]
Current netname (SSID): [ rezoXXXX ]
```

```
Desired netname (SSID): [ rezoXXXX ]
Current BSSID: [ 00:11:50:XX:XX:XX ]
Channel list: [ 7ffe ]
IBSS channel: [ 1 ]
Current channel: [ 11 ]
Comms quality/signal/noise: [ 0 85 0 ]
Promiscuous mode: [ Off ]
Intersil-Prism2 based card: [ 1 ]
Port type (1=BSS, 3=ad-hoc): [ 1 ]
MAC address: [ 00:0c:f6:XX:XX:XX ]
TX rate (selection): [ 0 ]
TX rate (actual speed): [ 1 ]
RTS/CTS handshake threshold: [ 2312 ]
Create IBSS: [ Off ]
Access point density: [ 1 ]
Power Mgmt (1=on, 0=off): [ 0 ]
Max sleep time: [ 100 ]
WEP encryption: [ On ]
TX encryption key: [ 1 ]
Encryption keys: [ 0XXXXXXXXXX][ ][ ][ ]
```

Il est aisé d'y lire tous les éléments de paramétrage initiaux ainsi que quelques renseignements comme le niveau de signal et de bruit. Les "X" sont bien entendu des masques. Je vous renvoie à l'étude de la page de manuel de wicontrol dans laquelle vous pourrez lire que wicontrol a été conçu pour fonctionner avec des cartes faisant intervenir des puces Lucent, Intersil et Atheros. Il en découle que tout ne fonctionne pas avec la carte SITECOM laquelle comporte une puce Ralink.

4 Conclusion, remerciements

Mon objectif était, en m'appuyant sur quelques documentations existantes, citées en préambule, de démontrer que la création d'un point d'accès wifi ouvrant vers le web est une chose tout à fait possible en utilisant un vieux PC et FreeBSD. Il reste que le résultat ainsi obtenu est encore perfectible et qu'il conviendra que l'utilisateur qui voudrait améliorer le dispositif décrit se penche sur les sujets suivants :

- Sécurisation WPA, possible avec ce boîtier.
- Sécurisation par mise en place de liste de MAC autorisée (ou interdites).
- Filtrage des transferts au niveau du routeur.

A l'un ou l'autre endroit du texte j'ai donné quelques conseils qui feraient, à juste titre, hurler des spécialistes de la sécurité comme l'activation de telnetd. Je n'ai fait ici que lever un petit coin du tapis, il reste un peu de travail...

Je tiens à remercier ici M. Damien BERGAMINI (<http://damien.bergamini.free.fr>) qui a écrit le driver pour puce RALINK 2500 pour la famille *BSD. La documentation mise en ligne sur son site a été une aide précieuse pour débroussailler ce sujet.

jml