

Configuration d'une interface de réseau sans fil IEEE 802.11

Philippe Latu
philippe.latu(at)inetdoc.net

<http://www.inetdoc.net>

Introduction à la configuration d'une interface de réseau sans fil avec le système GNU/Linux : identification du type d'interface, de ses caractéristiques et manipulation de ses paramètres. Ce support fournit une méthodologie de dépannage simple d'une connexion sur un réseau sans fil IEEE 802.11.

Table des matières

1. Copyright et Licence	2
1.1. Méta-information	2
1.2. Conventions typographiques	2
2. Identification des interfaces disponibles	3
2.1. Comment identifier le périphérique réseau ?	3
2.2. Comment vérifier que l'interface de réseau sans fil est bien gérée ?	3
3. Utilisation du kit wireless-tools	4
3.1. Commande iwconfig	4
3.2. Commande iwlist	5
3.2.1. Comment obtenir la liste des canaux accessible depuis l'interface ?	6
3.2.2. Quelles sont les infrastructures accessibles depuis l'interface ?	6
3.3. Bilan sur le kit wireless-tools	7
4. Utilisation de kismet	7
4.1. Installation de kismet	7
4.2. Configuration de kismet	7
4.2.1. Délégation des droits d'accès avec sudo	7
4.2.2. Configuration du type d'interface	7
4.3. Exécution de kismet	8
4.4. Bilan sur l'utilisation de kismet	9
5. Utilisation de Wireshark	9
6. Travaux pratiques	10
6.1. Travail préparatoire	10
6.2. Configuration de l'interface IEEE 802.11	11
6.3. Analyse des conditions de communications radio	11
6.4. Analyse des trames IEEE 802.11	11
7. Infrastructure Wi-Fi et méthodes d'authentification	12
8. Association sans authentification	13
8.1. Configuration du point d'accès : routeur ISR 877W	13
8.2. Configuration de la station sans outil d'authentification	14
8.3. Configuration de la station avec les outils d'authentification	16
8.4. Configuration de la station pour accéder à un <i>hotspot</i>	20
8.5. Chiffrement du trafic de la station avec ipsec	21
9. Notes sur le support matériel et les <i>firmwares</i>	21
9.1. Interfaces de type Intel	21
9.2. Interfaces de type Broadcom b43	21
10. Documents de référence & outils	23
10.1. Normes & standards	23
10.2. Outils utilisés	24
10.3. Références inetdoc.LINUX	24
10.4. Autres références	24
11. Glossaire des acronymes	24

1. Copyright et Licence

Copyright (c) 2000,2012 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2012 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.3 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

1.1. Méta-information

Cet article est écrit avec *DocBook*¹ XML sur un système *Debian GNU/Linux*². Il est disponible en version imprimable au format PDF : [config.interface.wlan.pdf](#)³.

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. C'est la distribution *Debian GNU/Linux* qui est utilisée pour les tests présentés. Voici une liste des paquets contenant les commandes utilisées dans ce document :

- pciutils - Linux PCI Utilities
- net-tools - The NET-3 networking toolkit
- ifupdown - High level tools to configure network interfaces
- iputils-ping - Tools to test the reachability of network hosts
- kismet - Wireless 802.11b monitoring tool
- wireless-tools - Tools for manipulating Linux Wireless Extensions
- wireshark - network traffic analyzer
- wpa_gui - GUI for wpa_supplicant
- wpa_supplicant - Client support for WPA and WPA2 (IEEE 802.11i)

1.2. Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou *prompt* spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite \$ ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.
- Toute commande précédée de l'invite # nécessite les privilèges du super utilisateur.

¹ <http://www.docbook.org>

² <http://www.debian.org>

³ <http://www.inetdoc.net/pdf/config.interface.wlan.pdf>

2. Identification des interfaces disponibles

Avant de pouvoir configurer une interface, il faut que le pilote de périphérique correspondant ait été chargé en mémoire. Comme une interface réseau est un dispositif matériel, c'est au niveau du noyau Linux que l'opération doit s'effectuer. Soit le pilote d'interface a été inclus dans la partie monolithique du noyau soit il est chargé sous forme de module. C'est cette dernière solution qui est le plus souvent retenue. Un module peut être chargé ou déchargé à volonté sans avoir à redémarrer la machine.

2.1. Comment identifier le périphérique réseau ?

Il existe une grande variété de contrôleurs d'interface réseau sans fil. À chaque composant correspond un pilote logiciel spécifique. Qu'il s'agisse d'une carte additionnelle ou d'un composant intégré sur la carte mère, le contrôleur est toujours un périphérique connecté à un bus PCI, USB ou ISA pour les modèles les plus anciens. Les commandes **lspci** du paquet `pciutils`, **lsusb** du paquet `usbutils` et **lspcmcia** du paquet `pcmciautils` donnent la liste des périphériques reliés respectivement aux bus PCI, USB ou ISA.

Voici quelques exemples caractéristiques obtenus à l'aide des commandes `$ lspci -v`, `$ lsusb` ou `$ lspcmcia -v`.

- Un contrôleur de marque Intel™ intégré sur carte mère

```
0c:00.0 Network controller: Intel Corporation PRO/Wireless 3945ABG [Golan] Network Connection (rev 02)
Subsystem: Intel Corporation Device 1021
Flags: bus master, fast devsel, latency 0, IRQ 31
Memory at f1fff000 (32-bit, non-prefetchable) [size=4K]
Capabilities: [c8] Power Management version 2
Capabilities: [d0] MSI: Mask- 64bit+ Count=1/1 Enable+
Capabilities: [e0] Express Legacy Endpoint, MSI 00
Capabilities: [100] Advanced Error Reporting
Capabilities: [140] Device Serial Number 65-5e-54-ff-ff-3c-1f-00
Kernel driver in use: iwl3945
```

- Un contrôleur mini PCI de marque Intel™ intégré sur carte mère.

```
03:03.0 Network controller: Intel Corporation PRO/Wireless 2915ABG ...
Subsystem: Intel Corporation Unknown device 1021
Flags: bus master, medium devsel, latency 64, IRQ 18
Memory at dceff000 (32-bit, non-prefetchable) [size=4K]
Capabilities: [dc] Power Management version 2
```

- Un contrôleur de marque Broadcom™ sur une carte PCCARD.

```
06:00.0 Network controller: Broadcom Corporation BCM4306 \
                        802.11b/g Wireless LAN Controller (rev 03)
Subsystem: Linksys Device 4320
Flags: bus master, fast devsel, latency 64, IRQ 11
Memory at 2c000000 (32-bit, non-prefetchable) [size=8K]
Capabilities: [40] Power Management version 2
Kernel driver in use: b43-pci-bridge
Kernel modules: ssb
```

- Un contrôleur de marque Realtek™ connecté sur un bus USB.

```
Bus 001 Device 003: ID 0bda:8187 Realtek Semiconductor Corp. RTL8187 Wireless Adapter
```

- Un contrôleur de marque Cisco™ sur une carte PCMCIA.

```
Socket 1 Bridge:      [yenta_cardbus]          (bus ID: 0000:00:04.1)
Configuration:      state: on                ready: yes
                    Voltage: 5.0V Vcc: 5.0V Vpp: 5.0V
Socket 1 Device 0:   [airo_cs]              (bus ID: 1.0)
Configuration:      state: on
Product Name:       Cisco Systems 350 Series Wireless LAN Adapter
Identification:     manf_id: 0x015f card_id: 0x000a
                    function: 6 (network)
                    prod_id(1): "Cisco Systems" (0xa17c320e)
                    prod_id(2): "350 Series Wireless LAN Adapter" (0x3d011600)
```

Certaines interfaces présentent quelques singularités quant à l'emploi de logiciel directement intégré sur les composants ; les *firmwares*. Quelques éléments sur l'obtention de ces *firmwares* sont donné à la [Section 9](#), « *Notes sur le support matériel et les firmwares* ».

2.2. Comment vérifier que l'interface de réseau sans fil est bien gérée ?

Les pilotes logiciels des composants sont chargés dynamiquement lors de l'initialisation du système d'exploitation. Dans la plupart des cas, ils sont chargés en mémoire sous forme de modules. On peut vérifier que ces pilotes logiciels ont bien été chargés en consultant les messages systèmes et la liste des modules chargés en mémoire.

Voici un extrait des messages d'initialisation du système avec le contrôleur Intel™ dont le pilote logiciel est baptisé `ipw2200`. Ces messages sont obtenus à l'aide de la commande **dmesg**.

```
# dmesg |grep -1 ipw2200
```

```
ipw2200: Intel(R) PRO/Wireless 2200/2915 Network Driver, 1.2.0kdmprq
ipw2200: Copyright(c) 2003-2006 Intel Corporation
ACPI: PCI Interrupt 0000:03:03.0[A] -> GSI 17 (level, low) -> IRQ 18
ipw2200: Detected Intel PRO/Wireless 2915ABG Network Connection
ipw2200: Detected geography ZZE (13 802.11bg channels, 19 802.11a channels)
ACPI: PCI Interrupt 0000:00:1e.3[B] -> GSI 17 (level, low) -> IRQ 18
```

On retrouve aussi ce nom de pilote dans la liste des modules chargés en mémoire. Cette liste est obtenue à l'aide de la commande **lsmod**.

```
# lsmod |grep ipw2200
ipw2200                177864  0
ieee80211              33864  1 ipw2200
firmware_class        10240  2 pcmcia,ipw2200
```

3. Utilisation du kit wireless-tools



Note

Les outils présentés ci-dessous doivent être remplacés dans un futur proche par une nouvelle interface de programmation et de configuration baptisée iw. Tant que l'intégration de ce nouvel outil n'est pas achevée dans la distribution Debian GNU/Linux, les informations données dans cette section restent d'actualité. Pour plus de détails sur l'évolution de cette «migration», il faut consulter le fichier de documentation du paquet iw : `/usr/share/doc/iw/README.Debian`.

Le kit wireless-tools contient les outils de configuration d'interface de réseau sans fil IEEE 802.11 au niveau liaison.

Relativement aux réseaux filaires de type Ethernet, il existe un grand nombre de paramètres à configurer au niveau liaison de données sur une interface IEEE 802.11 avant de passer au niveau réseau. Les outils fournis avec le paquet wireless-tools peuvent être utilisés par des logiciels graphiques de configuration réseau ou individuellement.

Voici les informations sur la version utilisée pour les tests présentés dans ce document.

```
$ dpkg -l wireless-tools
Souhait=inconnU/Installé/suppPrimé/Purgé/H=à garder
| État=Non/Installé/fichier-Config/dépaqueté/écheconFig/H=semi-installé
|/ Err?=(aucune)/H=à garder/besoin Réinstallation/X=les deux (État,Err: maj=mauvais)
|/ Nom          Version          Description
+++-----+-----+-----+
ii wireless-tools 29-1             Tools for manipulating Linux Wireless Extensions
```

Dans cette section, on s'intéresse à l'utilisation individuelle des différents outils dont voici la liste.

iwconfig

La commande **iwconfig** est le principal outil de manipulation des paramètres d'une interface de réseau sans fil. Son mode de fonctionnement est calqué sur celui de la commande **ifconfig** qui est utilisée pour le paramétrage au niveau réseau avec le protocole IP.

iwevent

La commande **iwevent** sert à afficher les événements générés par le pilote d'interface ou les évolutions sur le réseau.

iwgetid

La commande **iwgetid** renvoie des valeurs de paramètres individuels de configuration. Si les informations fournies sont identiques à celles affichées par la commande **iwconfig**, **iwgetid** est plus facile à intégrer dans les scripts des outils de configuration réseau interactifs.

iwlist

La commande **iwlist** sert à afficher des informations complémentaires à celles fournies par **iwconfig**.

iwpriv

La commande **iwpriv** sert à afficher (et/ou) configurer les paramètres complémentaires d'une interface. Dans la plupart des cas, il s'agit du support d'extensions qui ne font pas vraiment partie de la norme IEEE 802.11.

iwspy

La commande **iwspy** sert à collecter les statistiques de communication radio sur une **station** ou un **point d'accès**.

3.1. Commande iwconfig

Voici trois exemples d'exécution de la commande sans spécification de paramètre. Comme dans le cas de la commande **ifconfig**, l'exécution de la commande **iwconfig** affiche l'ensemble des valeurs courantes des options de l'interface.

Résultats obtenu avec une interface IEEE 802.11b.

```
$ /sbin/iwconfig wlan0
```

```
wlan0 IEEE 802.11-DS ESSID:"wlan.lab" ①
Mode:Managed② Frequency:2.442 GHz③ Access Point: 00:0E:83:88:E8:D4④
Bit Rate:11 Mb/s Tx-Power=20 dBm Sensitivity=0/65535
Retry limit:16 RTS thr:off Fragment thr:off
Power Management:off
Link Quality=100/100 Signal level=-34 dBm Noise level=-90 dBm
Rx invalid nwid:9418 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:54513 Missed beacon:0
```

Résultats obtenu avec une interface IEEE 802.11g.

```
$ /sbin/iwconfig wlan0

wlan0 IEEE 802.11g ESSID:"linux.home" ①
Mode:Managed② Frequency:2.412 GHz③ Access Point: 00:0F:66:DC:3D:31④
Bit Rate:54 Mb/s Tx-Power=20 dBm Sensitivity=8/0
Retry limit:7 RTS thr:off Fragment thr:off
Encryption key:<snipped/> Security mode:open
Power Management:off
Link Quality=99/100 Signal level=-23 dBm Noise level=-88 dBm
Rx invalid nwid:0 Rx invalid crypt:4 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Résultats obtenu avec une interface non associée.

```
$ /sbin/iwconfig wlan0

wlan0 unassociated ESSID:off/any ①
Mode:Managed② Channel=0③ Access Point: Not-Associated④
Bit Rate:0 kb/s Tx-Power=20 dBm Sensitivity=8/0
Retry limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:off
Link Quality:0 Signal level:0 Noise level:0
Rx invalid nwid:0 Rx invalid crypt:4 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:12 Missed beacon:0
```

①①① Informations sur le type de réseau sans-fil et l'identification du service.

La chaîne IEEE 802.11-DS désigne un réseau de type IEEE 802.11b alors que la chaîne IEEE 802.11g désigne directement le type de réseau.

L'acronyme **ESSID** signifie *Extended Service Set Identifier*. La chaîne de 32 caractères maximum correspondante identifie le domaine réseau auquel appartient l'interface.

L'option `ssid` de la commande **iwconfig** sert à configurer le nom de réseau. C'est la première option à paramétrer lors de l'implantation d'une station dans un nouveau réseau. La syntaxe est du type :

```
# iwconfig wlan<i> ssid "<myOwnWLAN>"
```

②②② Informations sur le type d'infrastructure du réseau sans fil.

Dans les trois exemples, l'interface appartient à une infrastructure simple ou étendue. L'option `mode` est positionnée à la valeur `Managed`.

Cette option `mode` peut prendre plusieurs valeurs. Dans le contexte de ce document, on ne s'intéresse qu'aux trois valeurs suivantes :

Ad-Hoc

Dans ce mode, l'interface s'associe directement aux autres stations sans utiliser un point d'accès. C'est le mode à utiliser lorsque l'on souhaite communiquer d'un hôte à l'autre sans information sur la présence d'une infrastructure.

Managed

Dans ce mode, l'interface s'associe à une infrastructure réseau comprenant un ou plusieurs point d'accès et peut gérer les déplacements entre zones de couverture radio (*roaming*).

Monitor

Dans ce mode, l'interface est placée en mode moniteur passif et collecte l'ensemble des trames présentes dans sa zone de couverture radio. C'est dans ce mode que l'on peut capturer et analyser les trames de gestion et de contrôle du réseau sans fil.

La syntaxe d'utilisation de cette option est du type :

```
# iwconfig wlan<i> mode managed
```

Pour plus d'information sur les autres valeurs de l'option `mode`, consulter les pages de manuels de la commande **iwconfig** : `$ man iwconfig`.

3.2. Commande iwlist

Cette commande permet d'obtenir des informations complémentaires à celles fournies par la commande **iwconfig**. La liste des options est donnée à l'aide de la séquence `$ /sbin/iwlist --help`.

Voici quelques exemples d'utilisations courantes de cette commande.

3.2.1. Comment obtenir la liste des canaux accessible depuis l'interface ?

Liste des canaux accessibles depuis une interface réseau IEEE 802.11b simple.

```
$ /sbin/iwlist wlan0 channel
wlan0    14 channels in total; available frequencies :
Channel 01 : 2.412 GHz
Channel 02 : 2.417 GHz
Channel 03 : 2.422 GHz
Channel 04 : 2.427 GHz
Channel 05 : 2.432 GHz
Channel 06 : 2.437 GHz
Channel 07 : 2.442 GHz
Channel 08 : 2.447 GHz
Channel 09 : 2.452 GHz
Channel 10 : 2.457 GHz
Channel 11 : 2.462 GHz
Channel 12 : 2.467 GHz
Channel 13 : 2.472 GHz
Channel 14 : 2.484 GHz
Current Frequency=2.442 GHz (Channel 7)
```

Liste des canaux accessibles depuis une interface réseau IEEE 802.11a/b/g.

```
$ /sbin/iwlist wlan0 channel
wlan0    32 channels in total; available frequencies :
Channel 01 : 2.412 GHz
Channel 02 : 2.417 GHz
Channel 03 : 2.422 GHz
Channel 04 : 2.427 GHz
Channel 05 : 2.432 GHz
Channel 06 : 2.437 GHz
Channel 07 : 2.442 GHz
Channel 08 : 2.447 GHz
Channel 09 : 2.452 GHz
Channel 10 : 2.457 GHz
Channel 11 : 2.462 GHz
Channel 12 : 2.467 GHz
Channel 13 : 2.472 GHz
Channel 36 : 5.18 GHz
Channel 40 : 5.2 GHz
Channel 44 : 5.22 GHz
Channel 48 : 5.24 GHz
Channel 52 : 5.26 GHz
Channel 56 : 5.28 GHz
Channel 60 : 5.3 GHz
Channel 64 : 5.32 GHz
Channel 100 : 5.5 GHz
Channel 104 : 5.52 GHz
Channel 108 : 5.54 GHz
Channel 112 : 5.56 GHz
Channel 116 : 5.58 GHz
Channel 120 : 5.6 GHz
Channel 124 : 5.62 GHz
Channel 128 : 5.64 GHz
Channel 132 : 5.66 GHz
Channel 136 : 5.68 GHz
Channel 140 : 5.7 GHz
Current Frequency=2.412 GHz (Channel 1)
```

3.2.2. Quelles sont les infrastructures accessibles depuis l'interface ?

Recherche des infrastructures de réseau sans fil disponibles dans la zone de couverture radio de l'interface.

```
$ /sbin/iwlist wlan0 scan
wlan0    Scan completed :
Cell 01 - Address: 00:0F:66:DC:3D:31
          ESSID:"linux.home"
          Protocol:IEEE 802.11bg
          Mode:Master
          Channel:1
          Encryption key:on
          Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 6 Mb/s; 9 Mb/s
                   11 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
                   48 Mb/s; 54 Mb/s
          Quality=97/100 Signal level=-28 dBm
          IE: WPA Version 1
              Group Cipher : TKIP
              Pairwise Ciphers (1) : TKIP
              Authentication Suites (1) : PSK
          Extra: Last beacon: 1960ms ago
Cell 02 - Address: 00:0E:83:88:E8:D4
          ESSID:"wlan.lab"
          Protocol:IEEE 802.11b
          Mode:Master
          Channel:6
          Encryption key:off
          Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s
          Quality=92/100 Signal level=-38 dBm
          Extra: Last beacon: 1765ms ago
```

3.3. Bilan sur le kit wireless-tools

Les deux commandes **iwconfig** et **iwlist** sont les deux outils principaux du kit wireless-tools. Ces commandes sont essentielles à la compréhension des mécanismes de fonctionnement du niveau liaison de données d'un réseau sans-fil avant authentification. Les autres commandes sont moins pertinentes dans la mesure où elles correspondent à de la collecte d'informations qui peuvent être obtenues par ailleurs : journaux systèmes, noyau, etc.

4. Utilisation de kismet

Le logiciel **kismet** entre dans la catégorie des «sondeurs réseau» ou *wireless network sniffers*. Il offre de nombreuses possibilités qui sortent du cadre de ce document. Cet outil permet de répondre à un objectif simple : *recenser les équipements IEEE 802.11 actifs dans la zone de couverture radio actuelle*.

Avant de procéder à ses propres tests, il faut chercher à se placer dans les meilleures conditions. Dans le cas des réseaux sans-fils, on cherche à se positionner sur un canal libre de toute interférence. L'utilisation de kismet permet de sélectionner un canal IEEE 802.11 non occupé.

4.1. Installation de kismet

Cet outil est fourni en paquet avec la distribution Debian GNU/Linux. Il suffit donc d'installer ce paquet avec la commande `# apt-get install kismet` et de contrôler les informations correspondantes avec la commande `$ apt-cache show kismet`.

Voici les informations de version sur le paquet kismet utilisé pour le présent document.

```
$ dpkg -l kismet
Souhait=inconnU/Installé/suppPrimé/Purgé/H=à garder
| État=Non/Installé/fichier-Config/dépaqueté/échec-conFig/H=semi-installé
|/ Err?=(aucune)/H=à garder/besoin Réinstallation/X=les deux (État,Err: maj=mauvais)
||/ Nom                Version                Description
+++-----+-----+-----+-----+
ii kismet                2008-05-R1-4+b1       Wireless 802.11b monitoring tool
```

4.2. Configuration de kismet

Comme avec tous les logiciels d'analyse réseau, la configuration de kismet dépend des droits d'accès donnés à l'utilisateur pour prélever les informations directement sur l'interface réseau et du type de cette interface.

4.2.1. Délégation des droits d'accès avec sudo

Pour ce qui est des droits d'accès, soit on exécute le logiciel à partir du niveau super utilisateur, soit on délègue les droits du super utilisateur pour cet outil. Dans le dernier cas, on fait appel à `sudo` pour la délégation des droits.

Voici un extrait du fichier de configuration `/etc/sudoers` de `sudo` dans lequel on a ajouté kismet dans la liste des outils autorisés pour l'utilisateur phil.

```
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.

<snipped/>
phil    ALL=NOPASSWD: /sbin/iwspy, /sbin/iwconfig, /usr/bin/kismet
```

Une fois cette délégation des droits mise en place, on lance le logiciel à l'aide de la commande `$ sudo kismet` depuis n'importe quelle console.

4.2.2. Configuration du type d'interface

On doit éditer le fichier de configuration `/etc/kismet.conf` pour identifier le type d'interface réseau sans fil utilisée au niveau du paramètre `source` suivant la syntaxe `source=type,interface,name[,channel]`.

Les listes des types d'interfaces supportées est donnée dans la section *Capture sources* du fichier de documentation fourni avec le paquet kismet : `/usr/share/doc/kismet/README.gz`.

Voici les trois exemples de paramètres à utiliser qui correspondent aux trois exemples d'interface donnés dans la [Section 2, « Identification des interfaces disponibles »](#).

Intel Corporation PRO/Wireless 2915ABG

```
source=ipw2200,wlan0,MyWlan
```

Broadcom Corporation BCM4306 802.11b/g

```
source=bcm43xx,wlan0,MyWlan
```

Cisco Systems 350 Series Wireless LAN Adapter

```
source=cisco_wifix,eth1:wifi0,MyWlan
```

4.3. Exécution de kismet

Une fois la configuration en place, il ne reste plus qu'à lancer kismet et attendre quelques minutes pour que le recensement des équipements actifs soit complet.

Pour que le logiciel kismet puisse scruter les communications sur l'ensemble des 14 canaux utilisables par les réseaux IEEE 802.11b/g, il faut placer l'interface en mode monitor. Voici les commandes à utiliser pour reconfigurer l'interface réseau sachant qu'aucune configuration n'a été effectuée auparavant.

Intel Corporation PRO/Wireless 2915ABG

```
# iwconfig wlan0 essid any
# iwconfig wlan0 mode monitor
```

Broadcom Corporation BCM4306 802.11b/g

```
# iwconfig wlan0 essid any
# iwconfig wlan0 mode monitor
```

Cisco Systems 350 Series Wireless LAN Adapter

Pour analyser le trafic relatif à l'infrastructure à laquelle l'interface appartient, il faut utiliser l'option rfmon.

```
echo "Mode: rfmon" >/proc/driver/aironet/<interface>/Config
```

Pour analyser le trafic relatif à l'ensemble des infrastructures disponibles, il faut utiliser l'option y.

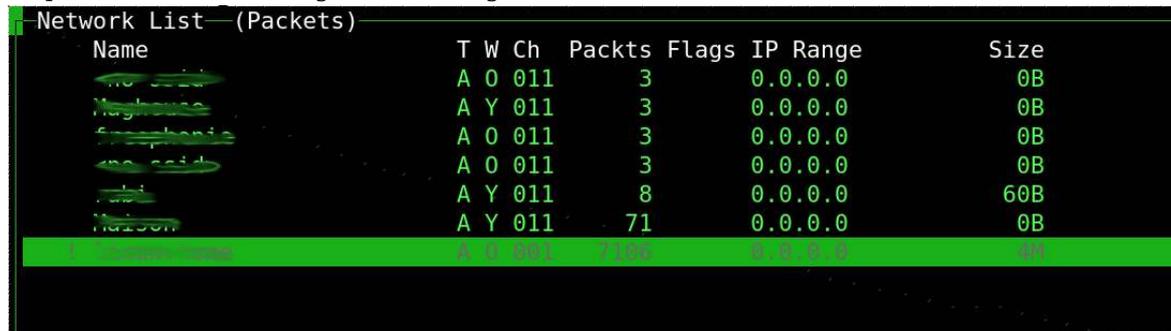
```
echo "Mode: y" >/proc/driver/aironet/<interface>/Config
```

Si on affiche l'état de l'interface en cours d'exécution, on constate qu'aucune valeur SSID n'est affectée, que (la fréquence|le canal) change à chaque exécution et que l'interface n'est pas associée à un point d'accès.

```
$ sudo iwconfig wlan0
wlan0      unassociated  ESSID:off/any
           Mode:Monitor  Frequency=2.457 GHz  Access Point: Not-Associated
           Bit Rate:0 kb/s  Tx-Power=20 dBm   Sensitivity=8/0
```

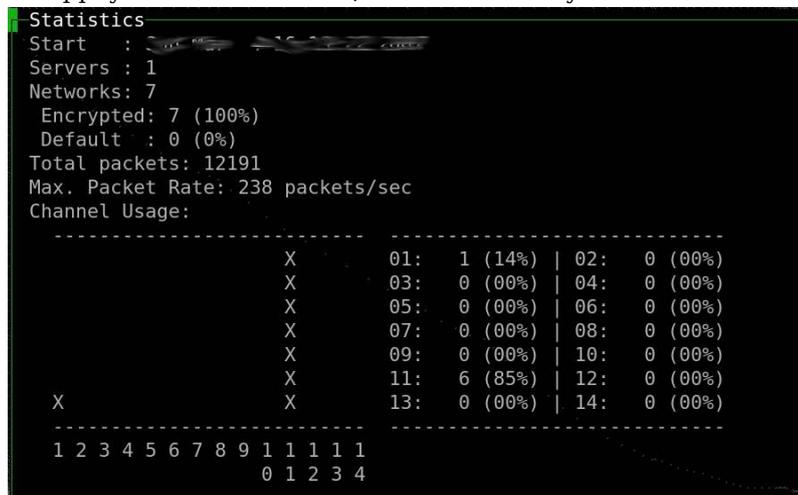
Les deux copies d'écran ci-dessous donnent un exemple des informations obtenues avec kismet.

L'écran principal de kismet affiche la liste des identificateurs de service (SSID), des indicateurs d'états, et la quantité de trames de gestion échangées.



Copie d'écran Kismet - vue complète⁴

En appuyant sur la touche a, on obtient une synthèse sur le taux d'occupation des canaux.

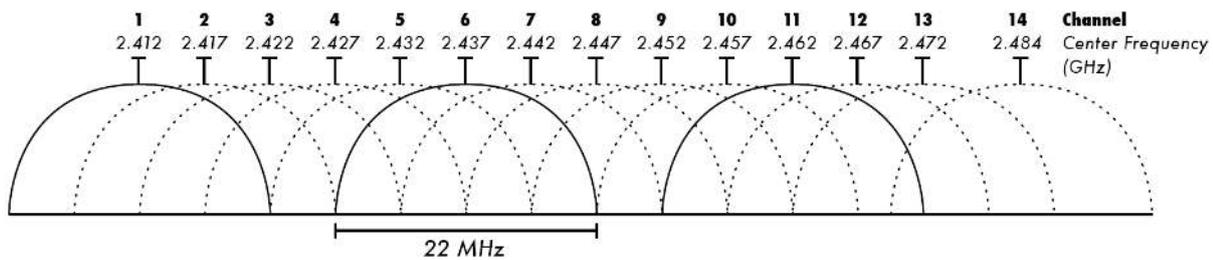


Statistiques Kismet - vue complète⁵

⁴ http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/kismet_network_list.png

⁵ http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/kismet_statistics.png

L'exploitation des données affichées ci-dessus montre que 6 équipements (stations ou points d'accès) sont présents sur le canal 11 et un point d'accès sur le canal 1. Le champ est donc relativement libre pour effectuer des tests sur les canaux non adjacents à ceux qui sont déjà occupés.



Canaux Wifi - source Wikipédia⁶

Pour les besoins des travaux pratiques, on peut configurer le point d'accès pour qu'il utilise l'un des canaux de la liste : 5, 6, 7. De cette façon, les mesures effectuées ne seront pas perturbées par les signaux issus des autres équipements actifs dans la zone de couverture radio.

4.4. Bilan sur l'utilisation de kismet

Kismet se révèle être un outil particulièrement intéressant pour le recensement et l'évaluation du taux d'occupation des canaux dans la zone de couverture radio étudiée. Dans la plupart des cas, les boîtes DSL/Wifi distribuées par les opérateurs Internet sont toutes configurées d'usine sur les mêmes canaux. Une des premières étapes d'optimisation des communications radio consiste à utiliser un canal relativement peu occupé pour limiter les interférences entre points d'accès qui desservent des infrastructures différentes.

Ceci dit, il faut préciser que l'utilisation de Kismet dans un contexte d'infrastructure de type *Hot Spot* permet de capturer l'ensemble du trafic réseau des utilisateurs *en clair*. Il va sans dire que ce genre d'écoute radio n'est pas conforme au bon usage des moyens de télécommunications ; même si c'est un moyen pédagogique très efficace pour sensibiliser les utilisateurs sur les limites de la confidentialité des communications.

5. Utilisation de Wireshark

Cette section a pour but de présenter l'utilisation de l'analyseur réseau *Wireshark* dans le contexte spécifique des réseaux sans-fils IEEE 802.11. Une première présentation de cet analyseur et de son usage pour les protocoles de couche réseau et plus est disponible avec le support *Introduction à l'analyse réseau avec Wireshark*⁷.

La singularité de l'analyse des réseaux sans-fils tient au fait qu'une interface ne peut pas être utilisée en mode infrastructure (*managed*) et en mode d'analyse radio (*monitor*) simultanément.

L'analyse en mode infrastructure ne peut capturer que des trames Ethernet «classiques». Ce contexte est strictement identique à l'analyse sur des réseaux filaires.

L'analyse radio sert à capturer des trames de gestion échangées entre stations et points d'accès. C'est le mode d'analyse qui est principalement utilisé dans le contexte de ce document.

Les opérations de configuration d'interface pour passer en mode analyse radio sont identiques à celles présentées dans la [Section 4, « Utilisation de kismet »](#). Les commandes diffèrent suivant le [modèle d'interface](#).

Voici quelques indications sur l'utilisation de wireshark en mode analyse radio.

Type d'encapsulation, IEEE 802.11 plus radiotap WLAN header

Si cette option de capture de trame n'apparaît pas dans les options, c'est que l'interface réseau n'est pas correctement configurée pour une analyse radio. Il faut alors reprendre les opérations de configuration avec la commande `iwconfig`.



Encapsulation de trame analyse radio - vue complète⁸

Filtrage des trames de type Beacon, `!(wlan.fc.type_subtype == 0x08)`

Suivant la densité des points d'accès, il est très probable que la phase de capture de trame soit noyée dans un flot continu de trame de type *beacon*. Bien souvent, les points d'accès sont très

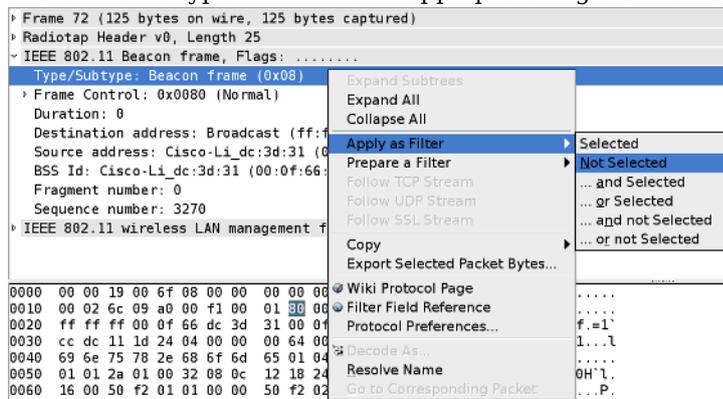
⁶ [http://en.wikipedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_\(802.11b,g_WLAN\).png](http://en.wikipedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).png)

⁷ http://www.inetdoc.net/travaux_pratiques/intro.analyse/

⁸ <http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wireshark-radiotap.png>

mal paramétrés et émettent ces trames à une fréquence trop élevée relativement au fonctionnement optimal de l'infrastructure sans-fil. De plus, ces trames ont pour conséquence d'activer les interfaces sans-fil des stations présentes dans la zone de couverture radio. Ces activations répétées entraînent une consommation plus importante pour les stations fonctionnant sur batteries. Il est donc conseillé d'augmenter la période d'émission des trames *Beacon* pour optimiser le temps de fonctionnement des équipements mobiles sur batteries.

Pour filtrer ces trames non désirées et isoler plus facilement les échanges intéressants, on sélectionne une trame de type *Beacon* et on applique la règle suivante :



Filtrage des trames de type Beacon - vue complète⁹

6. Travaux pratiques

6.1. Travail préparatoire

Avant de lancer les mesures proprement dites, il faut sélectionner un canal IEEE 802.11 libre et configurer le point d'accès en conséquence. On commence donc par procéder à une détection des canaux déjà occupés avec l'aide de [kismet](#).

1. Identifier le type de l'interface de réseau sans fil IEEE 802.11.

Reprendre la démarche présentée dans la section [Section 2.1](#), « Comment identifier le périphérique réseau ? ».

2. Vérifier que l'interface de réseau sans n'est pas configurée pour un accès réseau particulier à l'aide des commandes **ifconfig -a** et **iwconfig**.

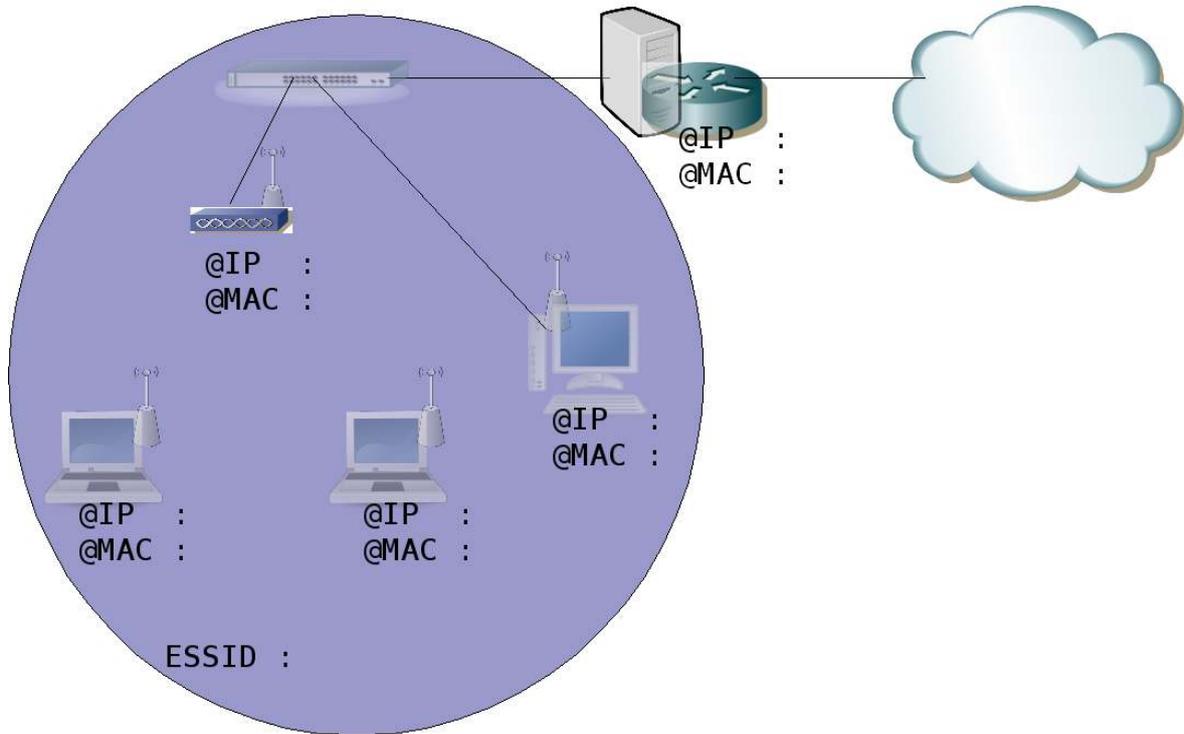
Aucune adresse IP ne doit être affectée à l'interface et elle ne doit pas être associée à un point d'accès.

3. Recenser l'état d'occupation des canaux IEEE 802.11 et choisir un canal libre non adjacent à un canal occupé. Relever aussi la liste des identificateurs de services (SSID) exploités dans la zone de couverture radio.

Reprendre la démarche présentée dans la section [Section 4](#), « Utilisation de kismet ». Le relevé des SSID permet repérer les adresses MAC utilisées par les autres points d'accès.

4. Faire un schéma de l'infrastructure de travaux pratique en identifiant tous les éléments de la norme IEEE 802.11 : DS, STA, AP et SSID.

⁹ <http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wireshark-beacon-filter.png>

Schéma type d'infrastructure réseau sans-fil - vue complète¹⁰

6.2. Configuration de l'interface IEEE 802.11

1. Configurer l'interface IEEE 802.11 pour qu'elle appartienne à l'infrastructure `wlan.lab`.

Visualiser l'état de l'interface à l'aide de la commande **iwconfig** avant et après l'affectation du paramètre `essid`.

2. Quelles sont les informations qui montrent que l'interface est associée à l'infrastructure `wlan.lab` ?

Si l'association est correcte, la commande **iwconfig** doit donner l'adresse MAC du point d'accès.

3. Quelles sont les opérations à réaliser pour configurer l'interface au niveau réseau ?

Visualiser l'état de l'interface à l'aide de la commande **ifconfig** avant et après l'affectation d'une adresse IP.

6.3. Analyse des conditions de communications radio

1. Quel est le niveau de puissance, exprimé en mW, utilisé sur l'interface réseau ? Configurer l'interface pour limiter la puissance d'émission à 1 mW.

Visualiser l'état de l'interface à l'aide de la commande **iwconfig** avant et après l'affectation du paramètre `txpower`.

2. Quelle est l'opération à réaliser pour configurer l'interface en mode d'analyse radio ?

Reprendre les commandes proposées dans la [Section 4, « Utilisation de kismet »](#) en fonction du **type d'interface** utilisée. Visualiser l'état de l'interface avant et après l'affectation du paramètre `mode`.

6.4. Analyse des trames IEEE 802.11

On s'intéresse ici à l'étude des trames spécifiques de la couche liaison des communications sans fils. Pour cette étude, on utilise l'analyseur réseau **wireshark** qui va permettre d'identifier les équipements actifs en présence et les différentes phases de communication.

Ce scénario suppose qu'une première station (STA) soit placée en mode analyse radio et analyse les échanges avec wireshark pendant qu'une seconde station effectue les opérations normales de configuration et d'association avec le point d'accès (AP).

1. Comment valider le fonctionnement de l'analyseur réseau wireshark en mode analyse radio ?

¹⁰ <http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wlan-infrastructure.png>

Relativement au mode de fonctionnement normal, la capture de trame doit offrir le type d'encapsulation IEEE 802.11 plus radiotap WLAN header. Si ce type d'encapsulation n'est pas disponible, c'est que l'interface est en mode infrastructure : *managed*.



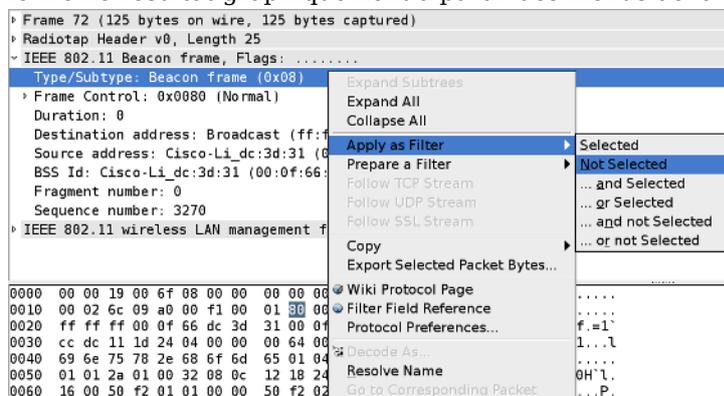
Encapsulation de trame analyse radio - vue complète¹¹

- Quels sont les types d'équipement à l'origine des trames de type *Beacon* et *Probe* ?

À l'aide des adresses MAC déjà identifiées et de la capture des trames, on peut distinguer les émissions des stations (STA) et des points d'accès (AP).

- Comment supprimer l'affichage des trames de type *Beacon* une fois la capture réalisée avec wireshark ?

À l'aide de la fonction de filtrage, il est possible de sélectionner les trames qui ne doivent pas être prises en compte lors de l'analyse. L'expression à saisir est : `!(wlan.fc.type_subtype == 0x08)`. On peut obtenir le même résultat graphiquement à partir des menus de la fenêtre d'analyse des protocoles.



Filtrage des trames de type Beacon - vue complète¹²

- Quelles sont les différentes phases de la reconnaissance d'une station par un point d'accès ?

À partir d'une capture réalisée pendant qu'une station est insérée dans la zone de couverture radio d'un point d'accès, relever les phases :

Probe Request	Probe Response
Authentication	Acknowledgement
Association Request	Association Response

Reconstituer le graphique suivant en indiquant les adresses MAC des équipements présents :



7. Infrastructure Wi-Fi et méthodes d'authentification

Les sections suivantes présentent quelques exemples de configurations d'interfaces de réseaux sans-fils en partant d'un cas pratique d'exploitation simple. On utilise une infrastructure domestique simple pour illustrer les méthodes d'authentification des interfaces Wi-Fi auprès d'un point d'accès.

L'infrastructure étudiée utilise un routeur Cisco Systems™ *ISR 877W*. Ce type de routeur offre des fonctions beaucoup plus étendues que les «boxes» des fournisseurs d'accès internet. Le point important ici étant la granularité dans les possibilités de configuration du système d'exploitation. Ceci dit, n'importe quel point d'accès devrait faire l'affaire dans la mesure où les protocoles d'authentification illustrés sont supportés. Les équipements Cisco sont configurables entièrement en mode console ce qui facilite la rédaction de

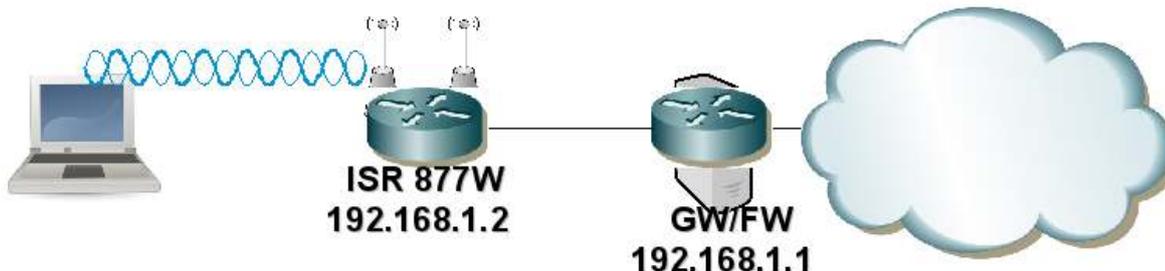
¹¹ <http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wireshark-radiotap.png>
¹² <http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wireshark-beacon-filter.png>

ce document en évitant la multiplication des copies d'écrans de pages Web de serveur intégré dans les équipements moins complets.

La terme «routeur» est usurpé dans cet exemple. En effet, on utilise les fonctions réseaux sans-fils en mode *pont* entre l'interface Wi-Fi et une interface filaire de l'équipement.

Côté GNU/Linux, on s'intéresse aux mêmes possibilités de configuration sur le système d'exploitation du poste client : généralement un ordinateur portable. On distingue la phase de mise au point des paramètres d'authentification de la phase de sauvegarde qui permet de s'associer à la demande à différentes infrastructures de réseaux sans-fils.

Pour chaque cas traité, on donne les configurations du point d'accès (AP) et de la station associée (STA).



Infrastructure Wi-Fi type - vue complète¹³

8. Association sans authentification

Ce premier exemple sert à mettre en place la configuration minimale nécessaire à la validation des communications radio. Il permet aussi d'illustrer le fonctionnement des outils utilisés.

Les caractéristiques générales de cet exemple sont les suivantes :

- Identification du service Wi-Fi (SSID) : open
- Réseau local virtuel (VLAN) utilisé sur le routeur ISR 877W : numéro 2
- Réseau IP utilisé : 192.168.1.0/24

8.1. Configuration du point d'accès : routeur ISR 877W

On commence de façon très classique par la version du système d'exploitation utilisée et par la liste des interfaces réseau reconnues.

```
Cisco IOS Software, C870 Software (C870-ADVIPSERVICESK9-M), Version 12.4(20)T, RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Fri 11-Jul-08 06:08 by prod_rel_team

<snipped/>

Cisco 877W (MPC8272) processor (revision 0x300) with 118784K/12288K bytes of memory.
Processor board ID FCZ113860FX
MPC8272 CPU Rev: Part Number 0xC, Mask Number 0x10
4 FastEthernet interfaces
1 ATM interface
1 802.11 Radio
128K bytes of non-volatile configuration memory.
28672K bytes of processor board System flash (Intel Strataflash)
```

On décrit ensuite les éléments de configuration minimale à partir des commandes ci-dessous.

- Configuration du pont au niveau global.

```
! Activation de la fonction Integrated Routind and Bridging
bridge irb
! Choix du protocole Spanning Tree
bridge 2 protocol ieee
! Activation du routage des paquets sur l'interface virtuelle de pont
! BVI: Bridge Virtual Interface
bridge 2 route ip
```

- Configuration de l'interface de pont.

```
! Interface virtuelle activée automatiquement
interface BVI2
ip address 192.168.1.2 255.255.255.0
```

¹³ <http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wlan-cookbook.png>

- Configuration de l'identificateur de service Wi-Fi (SSID) au niveau global.

```
dot11 ssid open
vlan 2
authentication open
guest-mode
```

- Configuration de l'interface Wi-Fi.

```
interface Dot11Radio0
! Adresse IP affectée au niveau pont (BVI2)
no ip address
! Choix du mode de chiffrement minimal
encryption mode wep optional
! Correspondance avec le SSID défini
ssid open
! Sous interface propre au VLAN 2 et appartenant au pont 2
interface Dot11Radio0.2
encapsulation dot1Q 2
bridge-group 2
```

- Configuration de l'interface de réseau local virtuel (*Switched Virtual Interface* ou SVI).

```
interface Vlan2
! Adresse IP affectée au niveau pont (BVI2)
no ip address
! Interface appartenant au pont 2
bridge-group 2
```

- Configuration des interfaces filaires dans le VLAN numéro 2.

```
int range fa0 - 3
switchport access vlan 2
```

8.2. Configuration de la station sans outil d'authentification

On procède de la même façon que dans la section précédente sur la configuration du point d'accès. On donne les caractéristiques de la station utilisée un ordinateur portable.

- Interface réseau sans-fil reconnue par le noyau Linux :

```
$ lspci | grep -i network
0c:00.0 Network controller: Intel Corporation PRO/Wireless 3945ABG Network Connection (rev 02)
```

Sur un ordinateur portable «moderne», l'interface de réseau sans fil est intégrée sur la carte mère. On retrouve cette interface dans la liste des périphériques connectés au bus PCI.

- Liste des interfaces ayant des fonctions réseau sans-fil :

```
# iwconfig
lo          no wireless extensions.

eth0       no wireless extensions.

wmaster0   no wireless extensions.

wlan0      IEEE 802.11  ESSID:""
           Mode:Managed  Frequency:2.447 GHz  Access Point: Not-Associated
           Tx-Power=15 dBm
           Retry min limit:7   RTS thr:off   Fragment thr=2352 B
           Encryption key:off
           Link Quality:0  Signal level:0  Noise level:0
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

On relève qu'à ce stade de la configuration aucun identifiant de cellule Wi-Fi n'est affecté (ESSID:"") et que l'interface n'est associée à aucun point d'accès : Access Point: Not-Associated.

- Liste des interfaces disponibles à la configuration :

```
# ifconfig
<snipped/>
wlan0      Link encap:Ethernet  HWaddr 00:1f:3c:54:5e:65
           UP BROADCAST MULTICAST  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 lg file transmission:1000
           RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0:avahi Link encap:Ethernet  HWaddr 00:1f:3c:54:5e:65
           inet adr:169.254.5.5  Bcast:169.254.255.255  Masque:255.255.0.0
           UP BROADCAST MULTICAST  MTU:1500  Metric:1

wmaster0   Link encap:UNSPEC  HWaddr 00-1F-3C-54-5E-65-00-00-00-00-00-00-00-00-00-00
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 lg file transmission:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

On relève qu'aucune adresse IP n'a été affectée à l'interface Wi-Fi.

Une fois que l'on a validé le fonctionnement de l'interface de réseau sans fil, on passe aux tests de communication. Comme nous sommes dans un cas extrêmement simple, on peut se contenter d'une configuration à la console.

- On désactive les paramètres activés lors de l'initialisation du système d'exploitation.

```
# ifdown wlan0
There is already a pid file /var/run/dhclient.wlan0.pid with pid 5901
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/

wmaster0: unknown hardware address type 801
wmaster0: unknown hardware address type 801
Listening on LPF/wlan0/00:1f:3c:54:5e:65
Sending on LPF/wlan0/00:1f:3c:54:5e:65
Sending on Socket/fallback
```

- On affecte l'identifiant de réseau Wi-Fi.

```
# iwconfig wlan0 essid open
# iwconfig
<snipped/>

wlan0 IEEE 802.11 ESSID:"open"
Mode:Managed Frequency:2.447 GHz Access Point: Not-Associated
Tx-Power=15 dBm
Retry min limit:7 RTS thr:off Fragment thr=2352 B
Encryption key:off
Link Quality:0 Signal level:0 Noise level:0
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

- On vérifie que l'association avec le point d'accès se fait bien ; dès que l'interface est active.

```
# ifconfig wlan0 up
# iwconfig
<snipped/>

wlan0 IEEE 802.11 ESSID:"open"
Mode:Managed Frequency:2.447 GHz Access Point: 00:1D:45:B7:EF:00
Bit Rate=54 Mb/s Tx-Power=15 dBm
Retry min limit:7 RTS thr:off Fragment thr=2352 B
Encryption key:off
Link Quality=100/100 Signal level=-19 dBm Noise level=-60 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

On relève l'adresse MAC du point d'accès auquel l'interface est associée (Access Point: 00:1D:45:B7:EF:00) ainsi que les paramètres qualitatifs de la couverture radio. Voir [Section 3.1, « Commande iwconfig »](#).

- La configuration du niveau réseau ne présente pas difficulté. Elle se fait soit manuellement soit à l'aide du client DHCP lorsque ce service est disponible. Voir la référence [Configuration d'une interface réseau](#).

```
# dhclient3 wlan0
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/

wmaster0: unknown hardware address type 801
wmaster0: unknown hardware address type 801
Listening on LPF/wlan0/00:1f:3c:54:5e:65
Sending on LPF/wlan0/00:1f:3c:54:5e:65
Sending on Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 8
DHCPOFFER from 192.168.1.1
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.1
bound to 192.168.1.103 -- renewal in 690 seconds.
```



Comment éviter les *HotSpots* ?

Suivant le contexte de couverture radio, il est fort probable que les manipulations soient perturbées par la présence de point d'accès de type *HotSpots*. Ces points d'accès émettent des trames de type *beacon* à un rythme assez élevé.

Pour éviter une association à un point d'accès *HotSpot*, on peut utiliser les commandes de configuration dans l'ordre suivant :

1. Désactivation de l'interface : **ifdown wlan0**
2. Affectation du SSID voulu : **iwconfig wlan0 essid open**
3. Activation de l'interface : **ifconfig wlan0 up**
4. Réaffectation du SSID voulu : **iwconfig wlan0 essid open**

En tout état de cause, un recensement des points d'accès présents dans la zone de couverture radio permet de repérer les «éléments perturbateurs». Voir [Section 4, « Utilisation de kismet »](#).

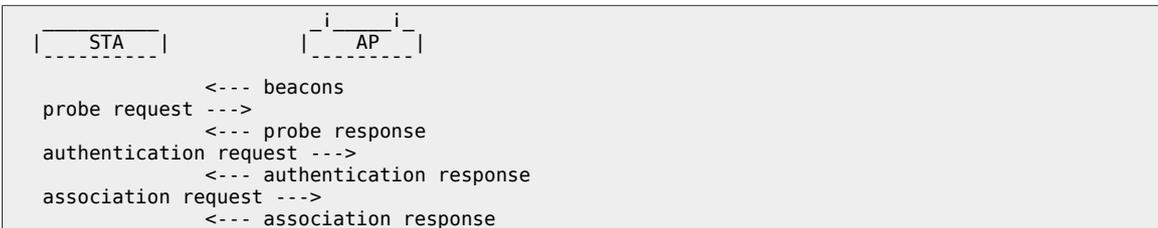
Le tour est joué ! Il est maintenant possible de communiquer avec l'Internet «en clair» dans la zone de couverture radio du point d'accès sachant qu'il est lui même raccordé à une passerelle via le réseau filaire. Cette configuration correspond presque aux *hotspots* disponibles dans les lieux publics. Il manque le portail captif opérateur qui permet «d'extorquer des sommes exorbitantes» aux usagers isolés qui n'ont pas d'autre solution de communication.

Enfin, il reste à afficher les messages des systèmes permettant de confirmer que les opérations de configuration se sont bien déroulées.

- Côté station, les messages systèmes donnent les informations sur le fonctionnement de l'interface réseau sans fil IEEE 802.11.

```
# dmesg
<snipped/>
Registered led device: iwl-phy0:radio
Registered led device: iwl-phy0:assoc
Registered led device: iwl-phy0:RX
Registered led device: iwl-phy0:TX
ADDRCONF(NETDEV_UP): wlan0: link is not ready
wlan0: Initial auth_alg=0
wlan0: authenticate with AP 00:1d:45:b7:ef:00
wlan0: RX authentication from 00:1d:45:b7:ef:00 (alg=0 transaction=2 status=0)
wlan0: authenticated
wlan0: associate with AP 00:1d:45:b7:ef:00
wlan0: RX AssocResp from 00:1d:45:b7:ef:00 (capab=0x421 status=0 axml:id=4)
wlan0: associated
wlan0: switched to short barker preamble (BSSID=00:1d:45:b7:ef:00)
ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
wlan0: no IPv6 routers present
```

Le message `wlan0: link becomes ready` déclenche les opération de configuration de niveau réseau une fois la partie association effectuée. On retrouve les étapes du synoptique ci dessous.



- Côté point d'accès, les mêmes éléments sont présents dans les journaux systèmes et dans la table des associations.

```
#sh dot11 associations

802.11 Client Stations on Dot11Radio0:

SSID [open] :

MAC Address      IP address      Device          Name            Parent          State
001f.3c54.5e65  192.168.1.103  unknown        -               self            Assoc

#sh dot11 statistics client-traffic
Clients:
4-001f.3c54.5e65 pak in 9387 bytes in 1288304 pak out 5915 bytes out 2475207
dup 170 decrypt err 0 mic mismatch 0 mic miss 0
tx retries 280 data retries 280 rts retries 0
signal strength 32 signal quality 130
```

8.3. Configuration de la station avec les outils d'authentification

Dans la section précédente, toutes les opérations réalisées à la console sont à répéter à chaque réinitialisation du système sachant qu'aucune sauvegarde de la configuration n'a été faite. L'objectif de cette section est justement de mettre en place une configuration sauvegardée. De plus, on présente les outils qui vont permettre d'utiliser différentes méthodes d'authentification par la suite.

Pour être capable de sauvegarder une configuration avec ou sans authentification il faut d'abord spécifier qu'une interface est de type Wi-Fi et qu'elle a recours à un *supplicant*.

Sur un système Debian GNU/Linux, c'est le fichier `/etc/network/interfaces` qui contient les paramètres de niveau 3 (IP) des interfaces réseau. Les paquets `wpa_supplicant` et `wpagui` fournissent le logiciel nécessaire au processus conduisant à l'**association** de la station au point d'accès au niveau 2 (MAC).

De façon classique, on obtient les informations sur les paquets relatifs au *supplicant* en interrogeant le gestionnaire de paquets.

```
# dpkg -l wpa* |grep ^ii
ii wpagui 0.6.4-1 GUI for wpa_supplicant
ii wpa_supplicant 0.6.4-1 Client support for WPA and WPA2 (IEEE 802.11i)
```

La principale ressource documentaire se trouve dans le fichier `/usr/share/doc/wpa_supplicant/README.Debian.gz` du paquet `wpa_supplicant`. Le mode de configuration le plus intéressant et celui utilisé dans ce document est baptisé *Mode #2: Roaming Mode*. Ce mode offre plusieurs fonctionnalités intéressantes.

Comme son nom l'indique, le mode «vagabondage» (*roaming*) doit supporter plusieurs configurations suivant la zone de couverture radio dans laquelle se trouve la station : lieu public, infrastructure d'entreprise, réseau domestique, etc. Le système de la station doit donc pouvoir changer de configuration dynamiquement sans qu'il soit nécessaire de le réinitialiser. Dans le contexte de ce document, il est possible d'interagir avec le *supplicant* et de tester les paramètres de configuration au cas par cas.

Voici une copie du fichier `/etc/network/interfaces` faisant appel au *supplicant* en mode *roaming*. Aucune définition d'infrastructure de réseau sans-fil n'a encore été implantée dans ce fichier.

```
# The loopback network interface
auto lo
iface lo inet loopback

# The Ethernet network interface
allow-hotplug eth0
iface eth0 inet dhcp

# The Wi-Fi network interface
allow-hotplug wlan0
iface wlan0 inet manual❶
    wpa-driver wext❷
    wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf❸
```

Ce fichier comprend trois interfaces :

- L'interface de boucle locale `lo` nécessaire aux communications réseau (TCP/IP) entre les processus locaux exécutés sur la station.
- L'interface filaire Ethernet `eth0` dont la configuration réseau est obtenue dynamiquement via le service **DHCP**.
- L'interface Wi-Fi `wlan0` est configurée en mode manuel. Ses paramètres sont détaillés ci-après.

- ❶ L'interface de réseau sans-fil est placée en mode manuel pour la configuration du niveau réseau sachant que l'on doit faire appel à un service particulier pour la configuration au niveau liaison : le *supplicant*. C'est cet outil qui est responsable de la gestion des associations entre la station (STA) et les différents point d'accès (AP) accessibles.

La configuration du niveau réseau se fait à partir d'un identifiant d'interface partagé entre les fichiers de configuration du *supplicant* et celui des interfaces ; c'est à dire le fichier présenté ci-dessus (`/etc/network/interfaces`).

- ❷ Le paramètre `wpa-driver` désigne l'interface logicielle d'échange entre le *supplicant* et le pilote d'interface réseau. La valeur `wext` correspond au mode d'échange par défaut sachant que l'interface utilisée dans cet exemple est de marque Intel™.
- ❸ Le paramètre `wpa-roam` désigne le fichier de configuration du *supplicant*. C'est ce fichier qui doit contenir la liste des réseaux sans-fil auxquels la station est susceptible de se connecter. Pour chacun des réseaux sans-fils utilisables, un certain nombre de paramètres doivent être présents. Les plus importants sont l'**identifiant** de cellule ou d'infrastructure et les méthodes d'authentification.

Voici une copie du fichier de configuration du *supplicant* avant introduction de définition de réseau sans-fil.

```
# cat /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant❶
update_config=1❷
```

- ❶ Le paramètre `ctrl_interface` désigne le répertoire contenant le *socket* de communication avec l'interface réseau sans-fil.
- ❷ La valeur du paramètre `update_config` autorise la mise à jour dynamique du fichier de configuration à partir des logiciels `wpa_gui` et `wpa_cli`.

Avant de se lancer dans les opérations de configuration avec l'interface graphique fournie par le paquet `wpagui`, il faut que l'utilisateur normal ait la capacité à utiliser cette interface. Dans ce but, on doit déléguer les droits d'utilisation du programme `/usr/sbin/wpa_gui` en suivant la démarche donnée dans la **Section 4.2.1, « Délégation des droits d'accès avec sudo »**. La ligne concernant l'utilisateur normal `phil` du fichier `/etc/sudoers` devient :

```
phil ALL=NOPASSWD: /sbin/iwspy, /sbin/iwconfig, /usr/bin/kismet, /usr/sbin/wpa_gui
```

À partir de cette étape, on peut lancer les opérations de configuration en dialoguant avec le *supplicant* via le *socket* dédié à l'interface. Voici un exemple d'identification de l'exécution du processus *supplicant* et du *socket* de communication avec l'interface réseau wlan0.

```
# ps auxww |grep wpa |grep -v grep
root      2801  0.0  0.0 21020 1020 ?        S<s  09:42   0:00 \
/sbin/wpa_supplicant -B -P /var/run/wpa_supplicant.wlan0.pid \
-i wlan0 -D wext -q -f /var/log/wpa_supplicant.wlan0.log \
-C /var/run/wpa_supplicant

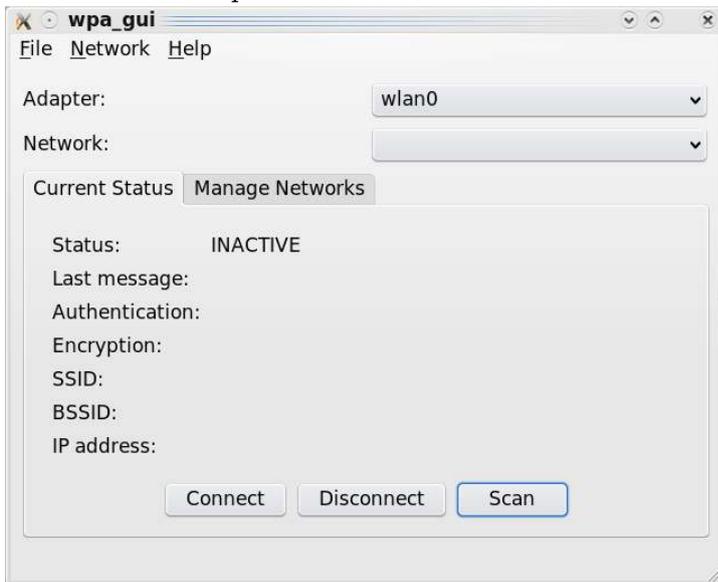
# ll /var/run/wpa_supplicant/wlan0
srwxrwx--- 1 root root 0 aou 20 09:42 /var/run/wpa_supplicant/wlan0
```

Voici les copies d'écran présentant la mise en place de la configuration minimale sans authentification.

1. Lancement de wpa_gui.

```
$ sudo wpa_gui
Selected interface 'wlan0'
Trying to connect to '/var/run/wpa_supplicant/wlan0'
```

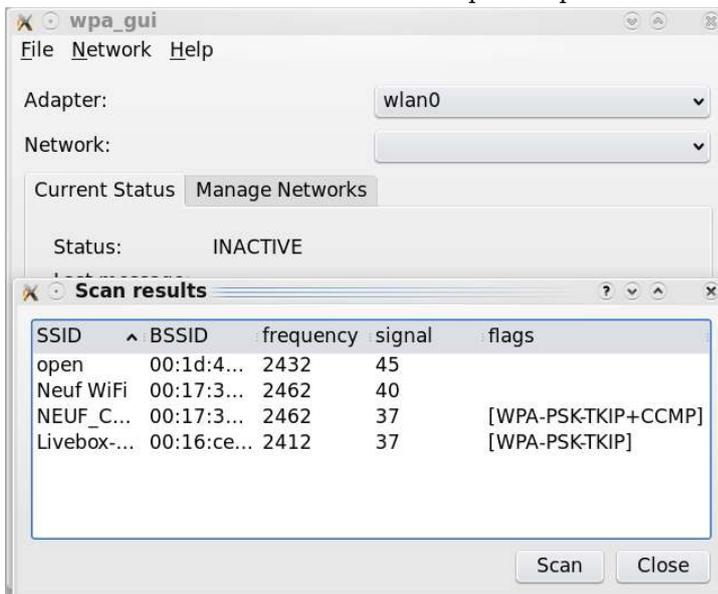
2. Une fois l'application lancée, la première opération à faire est de lancer un recensement des points d'accès visibles depuis la station en utilisant le bouton SCAN.



Copie d'écran initial wpa_gui - vue complète¹⁴

3. Un nouvel appui sur le bouton SCAN lance la reconnaissance des points d'accès accessibles.

Le résultat du recensement donne la liste des cellules visibles de la station avec leurs identifiants SSID ainsi que les modes d'authentification supportés. Ces informations sont issues des trames de requête *probe* émises par la station en direction des différents points d'accès «visibles». Cette «visibilité» dépend elle-même des trames *beacon* émises par les points d'accès.

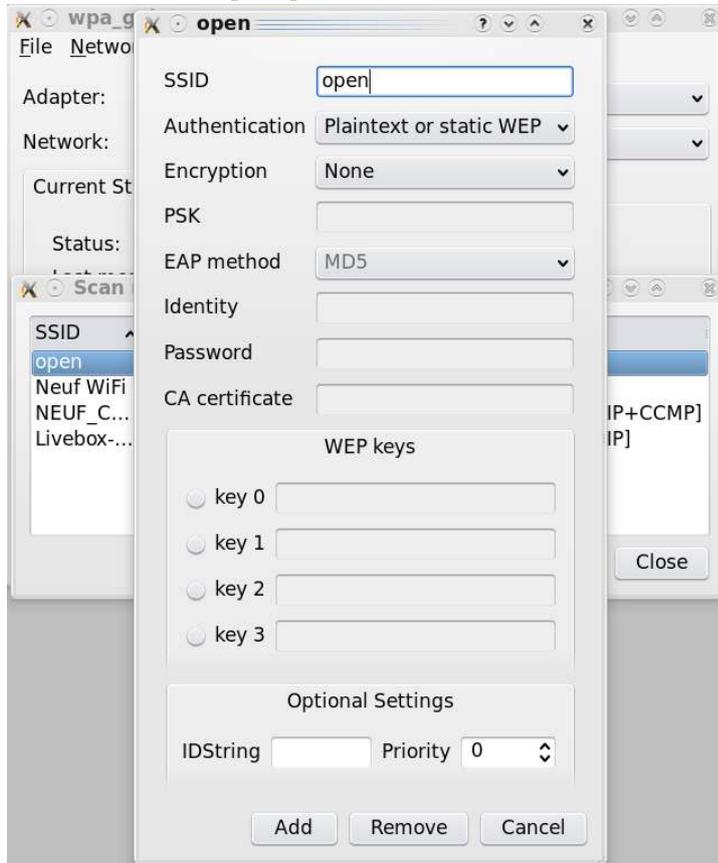


¹⁴ http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wpa_gui0.png

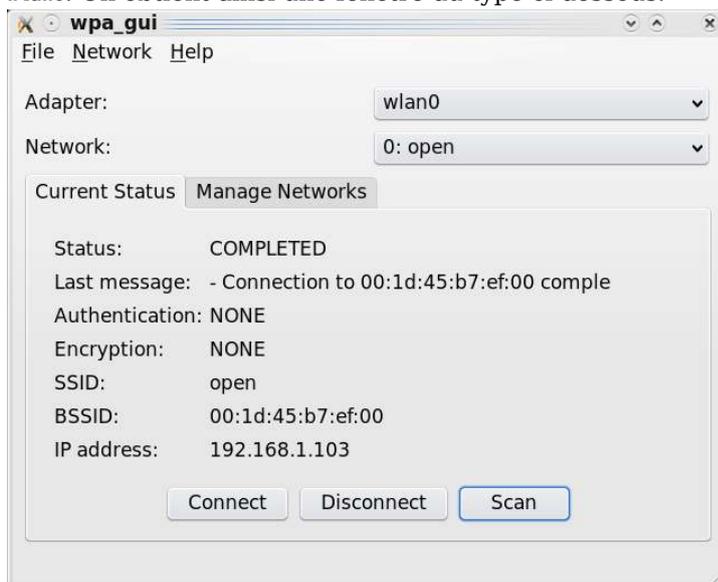
Copie d'écran scan wpa_gui - vue complète¹⁵

4. Un double click sur le SSID du réseau Wi-Fi voulu ouvre une fenêtre de configuration des paramètres d'authentification.

Dans le cas présent, on utilise l'identifiant open et il n'est pas nécessaire de modifier les paramètres d'authentification puisque l'on accède à un réseau ouvert.

Copie d'écran authentification wpa_gui - vue complète¹⁶

5. Enfin, une fois l'association réalisée, il ne reste plus qu'à lancer le client DHCP pour solliciter le serveur, obtenir une configuration réseau IP correcte et une référence de serveur de noms de domaines. Sachant qu'aucune configuration de niveau réseau n'a encore été sauvegardée dans le fichier /etc/network/interfaces, il est nécessaire de lancer manuellement la requête DHCP, à l'aide de la commande : `# dhclient3 wlan0`. On obtient ainsi une fenêtre du type ci-dessous.

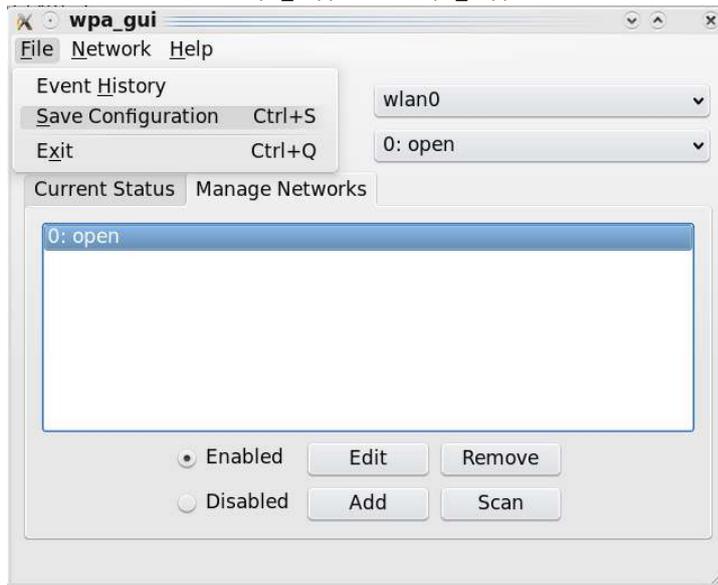
Configuration complète wpa_gui - vue complète¹⁷

¹⁵ http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wpa_gui1.png

¹⁶ http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wpa_gui2.png

¹⁷ http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wpa_gui3.png

6. Pour sauvegarder les paramètres de la configuration courante, on passe par le menu File puis par l'option Save Configuration. Cette opération est possible grâce au paramètre `update_config=1` préalablement placé dans le fichier `/etc/wpa_supplicant/wpa_supplicant.conf`.



Sauvegarde de configuration avec `wpa_gui` - vue complète¹⁸

Le résultat des manipulations réalisées via l'interface graphique du programme `wpa_gui` se retrouve dans le fichier de configuration du *supplicant*.

```
# cat /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
update_config=1

network={
    ssid="open"
    key_mgmt=NONE
    id_str=""
}
```

Cette configuration doit être complétée au niveau réseau en faisant le lien entre le fichier de configuration du *supplicant* (`/etc/wpa_supplicant/wpa_supplicant.conf`) et le fichier de configuration de niveau réseau des interfaces (`/etc/network/interfaces`) via la chaîne de caractères d'identification désignée par le paramètre `id_str`.

Dans cet exemple, on utilise la chaîne `open_testPod` et les fichiers de configuration sont édités comme ci-dessous.

- Fichier `/etc/wpa_supplicant/wpa_supplicant.conf` :

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1

network={
    id_str="open_testPod"
    ssid="open"
    key_mgmt=NONE
}
```

- Fichier `/etc/network/interfaces` :

```
# The Wi-Fi network interface
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-driver wext
    wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

# id_str="open_testPod"
iface open_testPod inet dhcp
```

8.4. Configuration de la station pour accéder à un *hotspot*

Dans les lieux publics, la tendance est au développement de la couverture radio Wi-Fi via un accès de type *hotspot*. Les opérateurs de téléphonie mobile sont très présents sur ce segment de marché. L'idée force consiste à dupliquer le modèle économique de la téléphonie mobile sur les accès aux réseaux sans-fils. Dans ce contexte, la station doit s'associer à n'importe quel point d'accès, obtenir une configuration réseau complète et se retrouver «piégée» par un portail captif qui intercepte toutes les communications réseau IP. Ce portail

¹⁸ http://www.linux-france.org/prj/inetdoc/cours/config.interface.wlan/images/wpa_gui4.png

captif tient le rôle de point de facturation en fonction du temps de communication de la même façon qu'une carte téléphonique prépayée.

Du point de vue configuration de l'interface de station, on reprend les mêmes éléments que dans la section précédente en omettant de préciser l'identifiant *Service Set Identifier*. Les fichiers de configuration sont complétés de la façon suivante :

- Fichier `/etc/wpa_supplicant/wpa_supplicant.conf` :

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1

network={
    id_str="open_testPod"
    ssid="open"
    key_mgmt=NONE
}

network={
    id_str="hotspot"
    ssid=""
    key_mgmt=NONE
}
```

- Fichier `/etc/network/interfaces` :

```
# The Wi-Fi network interface
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-driver wext
    wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

# id_str="open_testPod"
iface open_testPod inet dhcp

# id_str="hotspot"
iface hotspot inet dhcp
```

8.5. Chiffrement du trafic de la station avec ipsec

Dans le contexte des sections précédentes, le trafic réseau IP émis depuis la station circule «en clair» sur les ondes radio.

```
# dpkg -l ipsec-tools racoon |grep ^ii
ii ipsec-tools      1:0.7.1-1          IPsec tools for Linux
ii racoon           1:0.7.1-1          IPsec IKE keying daemon
```

9. Notes sur le support matériel et les firmwares

Après d'âpres discussion au sein de la communauté des développeurs Debian, il a été décidé de publier les logiciels binaires dans des paquets de la catégorie *non-free*. Les *firmwares* dont l'installation est présentée ici entrent justement dans cette liste de paquets dédiés.

9.1. Interfaces de type Intel

Les interfaces de marque Intel™ nécessitent un *firmware* spécifique pour fonctionner correctement. Ce logiciel binaire est aujourd'hui distribué via un paquet baptisé `firmware-iwlwifi` qui supporte les différentes familles de contrôleurs de la marque.

Les copies d'écran ci-dessous montrent qu'il n'est plus nécessaire de télécharger manuellement ces *firmwares* à partir de sites différents pas toujours faciles à identifier.

```
$ dpkg -l firmware-iwlwifi | grep -2 ^ii
||/ Nom          Version      Description
+++-----
ii firmware-iwlwifi 0.16        Binary firmware for Intel Wireless 3945, 4965 and 5000-series cards
```

```
$ dpkg -L firmware-iwlwifi | grep ucode
/lib/firmware/iwlwifi-4965-2.ucode
/lib/firmware/iwlwifi-3945-1.ucode
/lib/firmware/iwlwifi-4965-1.ucode
/lib/firmware/iwlwifi-5000-1.ucode
/lib/firmware/iwlwifi-3945-2.ucode
```

Le logiciel binaire est appelé automatiquement lors du chargement du pilote d'interface en mémoire pendant l'initialisation du système.

```
$ dmesg |grep iwl3945 |grep firmware
iwl3945 0000:0c:00.0: firmware: requesting iwlwifi-3945-2.ucode
iwl3945 loaded firmware version 15.28.2.8
```

Une fois ce chargement en mémoire effectué l'interface `wlan0` doit être prête pour les étapes de configuration suivantes et accessible via la commande `iwconfig`.

9.2. Interfaces de type Broadcom b43

Avec l'arrivée du noyau 2.6.26, les interfaces PC-CARD de type BCM4306 de la marque Broadcom™ sont intégrées dans les outils `mac80211` sous le nom `b43`.

Ce même noyau 2.6.26 est intégré dans la version stable de la distribution Debian GNU/Linux baptisée *Lenny*. Il est donc justifié de préciser quelques éléments sur l'utilisation de ce type d'interface avec cette nouvelle génération de logiciel de pilotage.

Tout d'abord, il est préférable de reprendre la reconnaissance et le recensement des interfaces réseau à zéro pour que les entrées Wi-Fi soient correctement positionnées.

C'est le démon `udev` qui à la charge de «convertir» les références des composants électriquement actifs reconnus par le noyau en entrées de type périphérique dans le système d'exploitation. Ce démon est fourni par le paquet du même nom et les règles de nommage des interfaces réseau sont placées dans le répertoire `/etc/udev/rules.d`. Voici les informations sur l'état du système utilisé pour la rédaction de cette section.

```
# dpkg -l udev |grep ^ii
ii udev          0.125-6                /dev/ and hotplug management daemon

# ll /etc/udev/rules.d/ |grep net
-rw-r--r-- 1 root root 537 a00 31 14:42 70-persistent-net.rules
-rw-r--r-- 1 root root 3,1K a00 31 14:40 75-persistent-net-generator.rules
```

Si l'interface Wi-Fi a précédemment été configurée sous la référence `bcm43xx`, il faut effacer le fichier `70-persistent-net.rules` à l'aide de la commande **# rm /etc/udev/rules.d/70-persistent-net.rules** et redémarrer le système. On se croirait sous Windoze !

Une fois que le nouveau jeu de règles est activé, on obtient les entrées suivantes dans le nouveau fichier `70-persistent-net.rules`.

```
# cat /etc/udev/rules.d/70-persistent-net.rules
# This file was automatically generated by the /lib/udev/write_net_rules
# program run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single line.

# PCI device 0x8086:0x1229 (e100)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?* ", ATTR{address}=="00:d0:59:9d:29:c6", \
    ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x14e4:0x4320 (b43)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?* ", ATTR{address}=="00:12:17:b6:9c:98", \
    ATTR{type}=="1", KERNEL=="wlan*", NAME="wlan0"
```

Avec ces règles, la liste des interfaces obtenues via les commandes **ifconfig** (et/ou) **iwconfig** fait bien apparaître les entrées `wlan0` et `wmaster0`.

- Commande de configuration du niveau Wi-Fi :

```
# iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

wmaster0    no wireless extensions.

wlan0       IEEE 802.11 ESSID:"open"
            Mode:Managed Frequency:2.427 GHz Access Point: 00:1D:45:B7:EF:00
            Bit Rate=54 Mb/s   Tx-Power=27 dBm
            Retry min limit:7   RTS thr:off   Fragment thr=2352 B
            Encryption key:off
            Link Quality=91/100  Signal level=-37 dBm  Noise level=-69 dBm
            Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
            Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

- Commande de configuration du niveau réseau IP :

```
# ifconfig
eth0        Link encap:Ethernet HWaddr 00:d0:59:9d:29:c6
            UP BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 lg file transmission:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

lo          Link encap:Boucle locale
            inet adr:127.0.0.1 Masque:255.0.0.0
            adr inet6: ::1/128 Scope:Hôte
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:88 errors:0 dropped:0 overruns:0 frame:0
            TX packets:88 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 lg file transmission:0
            RX bytes:5312 (5.1 KiB) TX bytes:5312 (5.1 KiB)

wlan0       Link encap:Ethernet HWaddr 00:12:17:b6:9c:98
            inet adr:192.168.1.6 Bcast:192.168.1.255 Masque:255.255.255.0
            adr inet6: fe80::212:17ff:feb6:9c98/64 Scope:Lien
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:13362 errors:0 dropped:0 overruns:0 frame:0
            TX packets:15606 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 lg file transmission:1000
            RX bytes:8907944 (8.4 MiB) TX bytes:1824997 (1.7 MiB)
```

```
wmaster0 Link encap:UNSPEC HWaddr 00-12-17-B6-9C-98-77-6C-00-00-00-00-00-00-00
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Le logiciel de pilotage intégré à l'interface ou *firmware* est fourni par le constructeur LinkSys™. Le contenu de ce logiciel est protégé puisque seuls les binaires sont fournis. Il n'est donc pas possible de le recompiler à partir des sources comme on le fait couramment avec le noyau Linux. Cependant, l'interface entre le noyau et ce logiciel binaire est possible. C'est ce qui est fait dans le cas du pilote b43. C'est un module du noyau Linux qui fait appel au *firmware* constructeur pour utiliser les fonctions Wi-Fi.

La distribution Debian GNU/Linux fournit un paquet baptisé b43-fwcutter qui permet de télécharger le *firmware* et de l'installer dans l'arborescence du système d'exploitation. Voici les informations le paquet et sur l'arborescence d'installation.

```
# dpkg -l b43* |grep ^ii
ii  b43-fwcutter      1:011-5      Utility for extracting Broadcom 43xx firmware

# ls -l /lib/firmware/b43*
```

L'utilisation du *firmware* se retrouve dans les messages systèmes lors de l'activation de l'interface ; au démarrage par exemple. La commande **# dmesg | less** permet de parcourir les messages correspondants. Voir les lignes débutant par *firmware*: dans la copie d'écran ci-dessous.

```
b43-pci-bridge 0000:06:00.0: enabling device (0000 -> 0002)
ACPI: PCI Interrupt 0000:06:00.0[A] -> Link [C142] -> GSI 11 (level, low) -> IRQ 11
PCI: Setting latency timer of device 0000:06:00.0 to 64
ssb: Sonics Silicon Backplane found on PCI device 0000:06:00.0
b43-phy0: Broadcom 4306 WLAN found
phy0: Selected rate control algorithm 'pid'
Broadcom 43xx driver loaded [ Features: PMLR, Firmware-ID: FW13 ]
<snipped>
input: b43-phy0 as /class/input/input6
firmware: requesting b43/ucode5.fw
firmware: requesting b43/pcm5.fw
firmware: requesting b43/b0g0initvals5.fw
firmware: requesting b43/b0g0bsinitvals5.fw
b43-phy0: Loading firmware version 410.2160 (2007-05-26 15:32:10)
Registered led device: b43-phy0::tx
Registered led device: b43-phy0::rx
Registered led device: b43-phy0::radio
NET: Registered protocol family 10
lo: Disabled Privacy Extensions
ADDRCONF(NETDEV_UP): eth0: link is not ready
ADDRCONF(NETDEV_UP): wlan0: link is not ready
wlan0: Initial auth_alg=0
wlan0: authenticate with AP 00:1d:45:b7:ef:00
wlan0: RX authentication from 00:1d:45:b7:ef:00 (alg=0 transaction=2 status=0)
wlan0: authenticated
wlan0: associate with AP 00:1d:45:b7:ef:00
wlan0: RX AssocResp from 00:1d:45:b7:ef:00 (capab=0x421 status=0 axml:id=2)
wlan0: associated
wlan0: switched to short barker preamble (BSSID=00:1d:45:b7:ef:00)
ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
```

Ces messages systèmes sont la conséquence du chargement des modules de pilotage du noyau Linux. On obtient la liste des modules relatifs à l'interface à l'aide de la commande donnée ci-après.

```
# lsmod | grep b43
b43                110748  0
rfkill             5652    3 rfkill_input,b43
rng_core           3940    1 b43
mac80211           139680  1 b43
led_class           3908    1 b43
input_polldev      3752    1 b43
ssb                33476   1 b43
pcmcia             29548   2 b43,ssb
firmware_class     6816    2 b43,pcmcia
pcmcia_core        31892   5 b43,ssb,pcmcia,yenta_socket,rsrc_nonstatic
```

À partir de là, les opérations de configuration de l'interface de réseau sans-fil n'ont rien de spécifique au modèle de composant utilisé. Il n'y a plus qu'à se référer aux sections ci-dessus.

10. Documents de référence & outils

10.1. Normes & standards

IEEE 802.11 Standard

La page [IEEE 802.11 LAN/MAN Wireless LANS](http://standards.ieee.org/getieee802/802.11.html)¹⁹ permet de télécharger les standards publiés par le sous-comité 802.11 de l'*Institute of Electrical and Electronic Engineers*²⁰.

¹⁹ <http://standards.ieee.org/getieee802/802.11.html>

²⁰ <http://www.ieee.org/>

10.2. Outils utilisés

Kismet

Le logiciel *Kismet 802.11 layer2 wireless network sniffer*²¹ est utilisé dans ce document pour recenser les équipements sans fils actifs dans la zone de couverture radio. Il offre de nombreuses autres possibilités.

Wireshark

L'analyseur de réseau *Wireshark: The World's Most Popular Network Protocol Analyzer*²² est un outil essentiel pour l'étude et l'analyse des formats de trames IEEE 802.11. La page *WLAN (IEEE 802.11) capture setup*²³ donne les instructions sur la configuration d'une interface réseau sans fil avant capture.

Pour obtenir plus d'informations sur l'utilisation de *Wireshark*, consulter le support *Introduction à l'analyse réseau avec Wireshark*²⁴.

10.3. Références inetdoc.LINUX

Modélisations réseau

L'article *Modélisations réseau*²⁵ présente les caractéristiques générales des réseaux de télécommunications et les deux principales modélisations utilisées dans les réseaux contemporains.

Configuration d'une interface réseau

Le support *Configuration d'une interface de réseau local*²⁶ présente les commandes de configuration d'une interface réseau filaire ainsi que les manipulations sur les tables de routage et la résolution des noms de domaines.

10.4. Autres références

Page Wikipédia IEEE 802.11

L'article *IEEE 802.11*²⁷ présente les caractéristiques de la norme IEEE 802.11 de façon correcte même si quelques corrections sont à apporter.

11. Glossaire des acronymes

Access Point, Wireless Access Point, AP, WAP

Dans le standard IEEE 802.11, un point d'accès est un équipement actif qui accède aux réseaux filaire et radio. Il offre des services de communications aux **stations** équipées d'interfaces de réseau sans fil dans sa zone de couverture radio.

association, processus d'association

La notion d'*association* dans les réseaux sans-fils du type IEEE 802.11 correspond à l'établissement d'une liaison point à point entre une **station** et un **point d'accès**. Cette association se fait au niveau liaison une fois que toutes les conditions sont réunies :

- L'interface de la station à été configurée avec le même **identifiant** que celui desservi par le point d'accès.
- Les débits disponibles sur l'interface de la station sont compatibles avec ceux délivrés par le point d'accès.
- Le processus d'authentification de la station auprès de l'infrastructure de réseau sans fil à abouti avec succès.

beacon, Wi-Fi beacons

Dans le standard IEEE 802.11, une trame *beacon* est une trame de diffusion émise par le point d'accès (AP) à destination de toutes les stations (STA). Le rôle de ces *beacons* est de fournir les caractéristiques de la cellule Wi-Fi : l'identifiant SSID, la liste des débits disponibles ainsi que les modes et méthodes d'authentification.

DHCP, Dynamic Host Control Protocol

Le protocole DHCP est utilisé par les stations pour obtenir automatiquement les paramètres de configuration au niveau réseau ainsi et les paramètres du service de noms de domaines. Généralement, au niveau réseau les paramètres sont : adresse IP, masque réseau et adresse IP de passerelle de sortie du réseau. Pour la résolution des noms de domaines, le service DHCP doit fournir au moins une adresse IP de serveur DNS.

Distribution System, DS

Dans le standard IEEE 802.11, l'acronyme DS qualifie l'infrastructure filaire de connexion des points d'accès. Dans les réseaux sans-fils contemporains, cette infrastructure relie tous les points d'accès à des

²¹ <http://www.kismetwireless.net/>

²² <http://www.wireshark.org/>

²³ <http://wiki.wireshark.org/CaptureSetup/WLAN>

²⁴ http://www.inetdoc.net/travaux_pratiques/intro.analyse/

²⁵ <http://www.inetdoc.net/articles/modelisation/>

²⁶ http://www.inetdoc.net/travaux_pratiques/config.interface.lan/

²⁷ http://en.wikipedia.org/wiki/IEEE_802.11

commutateurs dédiés chargés du contrôle de la couverture radio en fonction de l'évolution dynamique des conditions de propagation : taux d'occupation des bâtiments, nombre de stations, interférences, détection de points d'accès étrangers, etc.

Media Access Control, MAC, Adresse MAC

Dans un premier temps, cet acronyme désigne la sous-couche la plus basse de la couche liaison de données du modèle contemporain (Voir *Modélisations réseau*). Sa définition a été donnée dans les standards IEEE 802. Cette sous-couche est placée au dessus de la couche physique. Elle est responsable de la délimitation de la suite de bits en une trame sans erreur.

Dans un deuxième temps, l'acronyme désigne aussi de format d'adressage utilisé dans la trame. Une adresse MAC est constituée de six octets notés en hexadécimal séparés par deux points. On retrouve ce type d'adresses principalement dans les réseaux Ethernet IEEE 802.3 et sans fils IEEE 802.11.

probe, probe request, probe response

Les trames de type *probe* sont utilisées par les stations lorsqu'elles veulent connaître les caractéristiques d'un point d'accès visible depuis leur interface Wi-Fi en vue d'une association. Les trames de requête (*probe request*) sont généralement émises sur tous les canaux en utilisant un identifiant *Service Set Identifier* recherché et en donnant la liste des modes d'authentification et des débits supportés. Le point d'accès sollicité réponds (*probe response*) alors en renvoyant une trame contenant les mêmes informations indiquant ce qu'il supporte de son côté.

Cet échange de trames *probe request / probe response* est un préalable à toute association.

Service Set Identifier, SSID, BSS, ESS, ESSID

Le premier acronyme désigne le nom de l'infrastructure réseau auquel une interface appartient. Ce nom est une chaîne de 32 caractères maximum. Comme une infrastructure de réseau sans fil constitue un «ensemble de services», on a baptisé l'identification de cette infrastructure : *Service Set Identifier*.

Les deux acronymes suivants sont relatifs au type d'infrastructure. Par définition, une infrastructure de type *Basic Service Set* ne comprend qu'un *point d'accès (AP)* qui gère seul les *stations (STA)* présentes dans la zone de couverture radio. En réalisant une interconnexion des points d'accès via un réseau filaire, on obtient une infrastructure de type *Extended Service Set*.

Enfin, la commande **iwconfig** fait apparaître l'acronyme ESSID qui identifie le nom de l'infrastructure quel que soit son type.

Station, STA

Dans le standard IEEE 802.11, une station est un hôte du réseau sans-fil. Un hôte doit être authentifié et associé à un *point d'accès* pour accéder aux autres réseaux (généralement l'Internet).

supplicant

Le terme *supplicant* est utilisé dans le standard IEEE 802.1X pour désigner un hôte de segment réseau point à point qui cherche à s'authentifier auprès d'un service d'authentification (*authenticator*). Ce service d'authentification est situé à l'autre extrémité de la liaison point à point. Dans la pratique, le *supplicant* est un logiciel installé sur le poste utilisateur. Le système d'exploitation de l'utilisateur lance ce logiciel qui soumet ses paramètres (clés, certificats, mot de passe, etc.) pour accéder à un réseau sécurisé. Lorsque l'authentification est validée, le service *authenticator* doit connecter le poste utilisateur au réseau.

Sur les systèmes GNU/Linux, le logiciel de référence est : *Linux WPA/WPA2/IEEE 802.1X Supplicant*²⁸.

²⁸ http://hostap.epitest.fi/wpa_supplicant/