



Bisoños Usuarios de Linux de Mallorca y Alrededores | Bergantells Usuaris de Linux de Mallorca i Afegitons

Sacando el jugo al Wireless: interconexión de redes con WDS

Por Ricardo Galli Granada, [gallir](http://mmn.uib.es/~gallir/) (<http://mmn.uib.es/~gallir/>)

Creado el 15/12/2002 19:45 y modificado por última vez el 15/12/2002 19:45

En éste artículo veremos como usar y configurar el sistema WDS (*Wireless Distribution System*) del *HostAP*, ya tratado en el artículo [Redes Wireless en Linux](#)⁽¹⁾, para interconectar redes LAN o poner repetidores para ampliar el alcance de nuestra red. Por supuesto, todo en Linux y con el [HostAP](#)⁽³⁾.

Introducción a WDS

Cuando se diseñó el estándar 802.11 se pensó en dos tipos básicos de servicios:

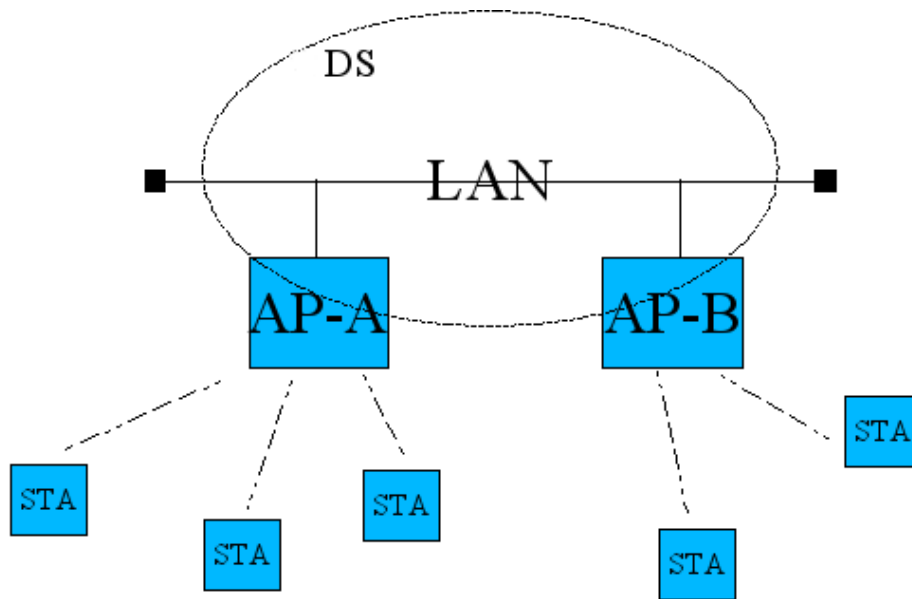
1. BSS (*Basic Service Set*): en este caso sólo hay un punto de acceso y una red inalámbrica definida por las estaciones conectadas a ese único AP.
2. ESS (*Extended Service Set*): en éste caso hay varios APs (como en la figura siguiente) e interesa que las estaciones conectadas a cualquiera de ellos puedan interconectarse entre ellas. El sistema que permite dicha interconexión es el WDS (*Wireless Distribution System*).

El sistema de distribución *wireless* no está del todo definido en el estándar 802.11. De hecho tampoco interesaba definirlo completamente, ya que es conceptualmente muy sencillo y a veces también muy fácil de implementar. De hecho el sistema de distribución (DS, *Distribution System*) está definido por separado ya que el medio puede ser diferente, por ejemplo una red LAN Ethernet.

El DS es sencillamente la forma en que se interconectan varios puntos de acceso (o AP) para permitir la interconexión de las estaciones inalámbricas registradas en los distintos APs. El DS también sirve de base para la implementación de sistemas más sofisticados como el NoCatAuth, *Roaming* con IAPP, Mobility IP, etc.

Bridging (o WDS) en la LAN

La forma más simple de DS es la conexión de varios APs en las misma red LAN, configurados con un *bridge* a Nivel 2, tal como lo expliqué en el [artículo anterior](#)⁽¹⁾ y como se muestra en la siguiente imagen:



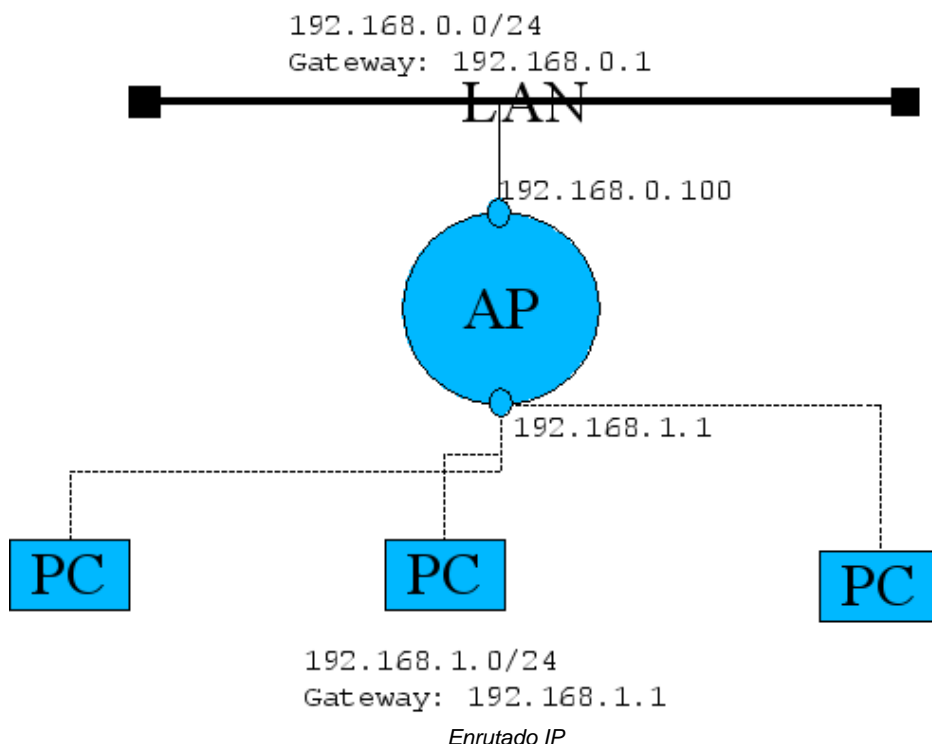
WDS a través de la LAN

La única diferencia es que en este caso tenemos dos ordenadores conectados a la misma LAN, con el *bridging* habilitado en los servidores *wireless*, el propio sistema de *bridging* se encargará de permitir la interconexión entre todas las estaciones y los ordenadores conectados a la LAN.

WDS, o DS inalámbrico

Pero, ¿es posible interconectar LAN o directamente APs a través del enlace inalámbrico?. Sí, **sí es posible**, para ello se ha definido un **formato especial de paquete** que permita el WDS a través del *wireless*.

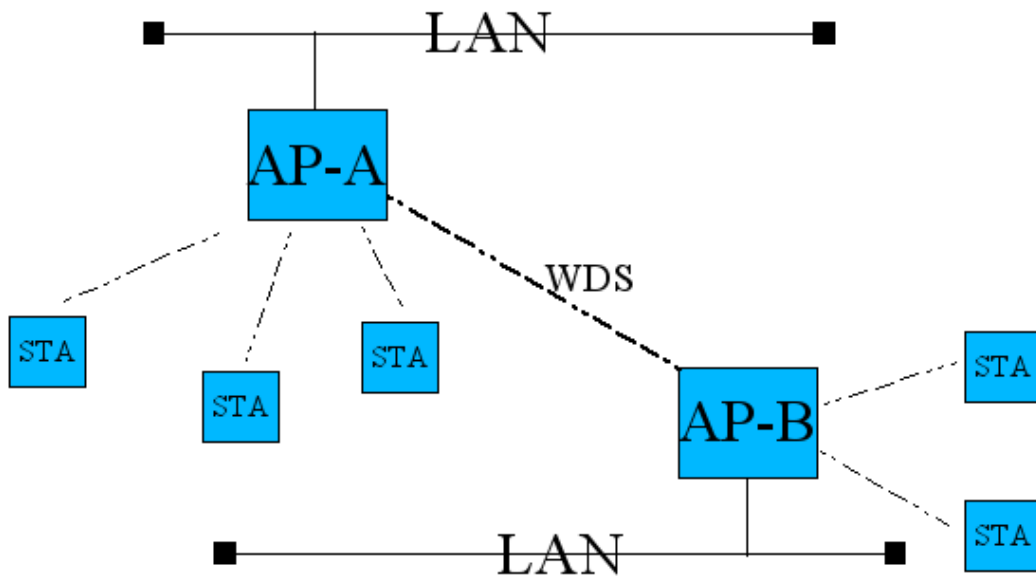
¿Porqué se necesita un formato especial de paquete? ¿Si hay interconexión entre un par de estaciones, una de ellas como AP y la otra como cliente, ¿no bastaría?. Sí, bastaría, **pero sólo si la interconexión se hace a nivel de IP** (similar a la figura siguiente), donde se definen las rutas que se han de seguir manualmente o usando algún algoritmo de enrutamiento dinámico, tipo RIP o OSPF.



Enrutado IP



Pero en el caso que quisiésemos interconectar dos redes LAN de forma transparente, es decir haciendo *bridging* a Nivel 2 entre ellas (figura siguiente), **no queda más remedio que usar las extensiones WDS del 802.11.**

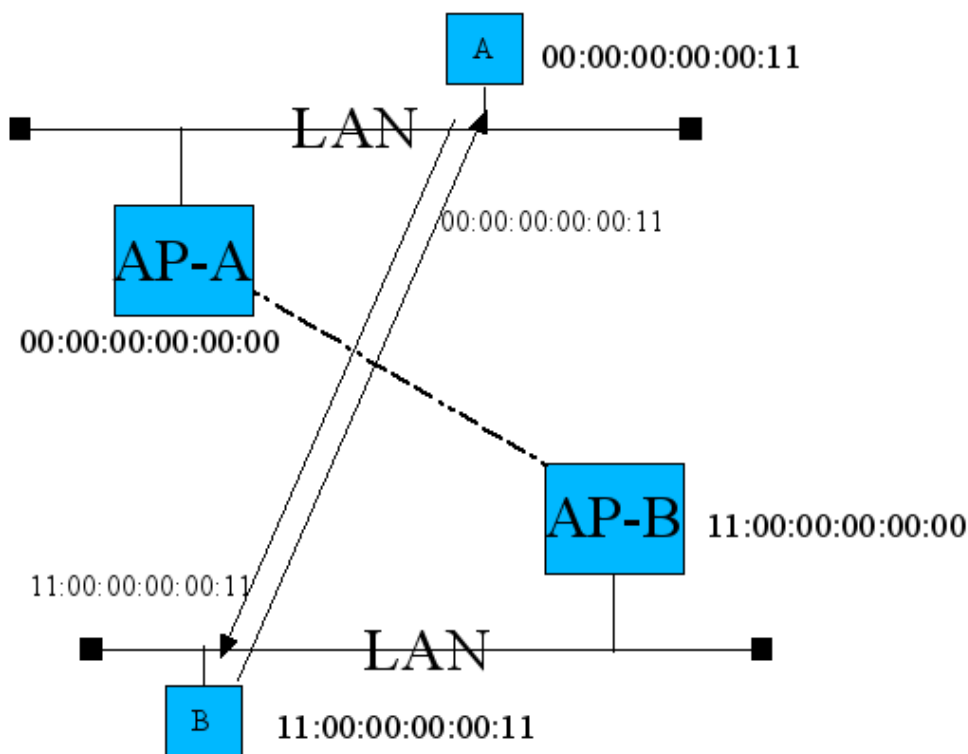


Interconexión de dos LAN a través de wireless

Campos adicionales en el paquete

Las conexiones *wireless* entre dos estaciones se realizan siempre enviando la dirección MAC de la tarjeta *wireless* del origen y del destino. La dirección MAC del destino sirve para que la tarjeta del receptor reciba y procese el paquete localmente. Es decir, **estos tipos de paquetes estándares sólo permiten la conexión entre un par de ordenadores, normalmente un AP y una estación registrada.**

En el caso que se quieran interconectar a nivel 2 un par de redes LAN, estos datos no bastan. Supongamos el siguiente caso, donde un ordenador A envía un paquete de datos a otro ordenador B en otra LAN distinta, interconectada por un enlace inalámbrico:



Envío de paquetes entre A y B

Para que A (con MAC 00:00:00:00:00:11) y B (con MAC 11:00:00:00:00:11) se puedan comunicar a nivel 2 ambas necesitan conocer la dirección MAC de la otra (de eso se encarga el protocolo ARP) y las tramas Ethernet que se envían usan dichas direcciones como origen y destino.

Si no tuviésemos la extensión WDS sería imposible realizar esta conexión, ya que en AP-A y AP-B perderíamos las direcciones MAC originales que serían reemplazadas por las direcciones MAC de los APs (00:00:00:00:00:00 y 11:00:00:00:00:00 respectivamente). Este problema se soluciona con la extensión WDS, que agrega dos campos adicionales para mantener las direcciones MAC del remitente y destino originales. Por ejemplo, si AP-A envía una trama de A hacia B conectado al AP-B, los campos del paquete *wireless* tendrán:

- **Destinatario:** 11:00:00:00:00:00
- **Origen:** 00:00:00:00:00:00
- **Destinatario original:** 11:00:00:00:00:11
- **Remitente original:** 00:00:00:00:00:11
- ...

Configuración del HostAP para AP-A y AP-B

Ahora veremos como configurar un Linux con el HostAP (y el *bridging* habilitado, leed el [artículo](#)⁽²⁾) y habilitar la extensión WDS para que funcione la configuración anterior.

Supongamos que ambos APs ya están configurados como puntos de acceso. Ahora tenemos que asegurarnos que **ambos estén en el mismo canal** (con el *iwconfig channel #*) y crear una interfaz que será *wlan#wds#* que será el punto a punto con el otro AP. Cada una de dichas interfaces sólo funciona con un AP, **por lo que hay que definir una para cada AP con el que queremos enlazar**. En nuestro ejemplo cada AP tendrá sólo una interfaz adicional.

En los comandos listados a continuación lo que hacemos es:

1. Crear una interfaz wds (en ambos casos serán wlan0wds0) enlazándola con la MAC del otro AP.
2. Configurarla la IP en 0.0.0.0
3. Agregar dicha interfaz al bridge.



Configuración AP-A

```
iwpriv wlan0 wds_add 11:00:00:00:00:00
ifconfig wlan0wds0 0.0.0.0
brctl addif br0 wlan0wds0
```

Configuración AP-B

```
iwpriv wlan0 wds_add 00:00:00:00:00:00
ifconfig wlan0wds0 0.0.0.0
brctl addif br0 wlan0wds0
```

Una vez realizados estos comandos y después de unos breves segundos ya deberías ver como se actualiza el bridge de cada ordenador:

```
ponti:~# iwconfig
```

```
...
```

```
wlan0 IEEE 802.11-b ESSID:"Antoli"
Mode:Master Frequency:2.437GHz Access Point: 00:50:C2:01:96:14
Bit Rate:11Mb/s Tx-Power:7 dBm Sensitivity=1/3
Retry min limit:8 RTS thr:off Fragment thr:off
Encryption key:XXXXXXXXXXXXXXXXXXXXX Encryption mode:restricted
Power Management: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:1907 Invalid misc:1540 Missed beacon:0
```

```
wlan0wds IEEE 802.11-b ESSID:"Antoli"
Mode:Master Frequency:2.437GHz Access Point: 00:50:C2:01:96:14
Bit Rate:11Mb/s Tx-Power:7 dBm Sensitivity=1/3
Retry min limit:8 RTS thr:off Fragment thr:off
Encryption key:XXXXXXXXXXXXXXXXXXXXX Encryption mode:restricted
Power Management:off
```

```
ponti:~# ifconfig
```

```
...
```

```
wlan0 Link encap:Ethernet HWaddr 00:50:C2:01:96:14
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:659454 errors:0 dropped:36941 overruns:0 frame:0
TX packets:79931 errors:1897 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:7410 (7.2 KiB) TX bytes:3679847 (3.5 MiB)
Interrupt:10 Base address:0x100
```

```
wlan0wds0 Link encap:Ethernet HWaddr 00:50:C2:01:96:14
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:60946 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:3926524 (3.7 MiB) TX bytes:42698209 (40.7 MiB)
Interrupt:10 Base address:0x100
```

```
ponti:~#
```

```
ponti:~# brctl show
```

bridge name	bridge id	STP enabled	interfaces
br0	8000.0050c2019614	yes	eth0 wlan0 wlan0wds0

NOTA: la elección de la dirección MAC 00:00:00:00:00:00 ha sido una muy mala idea, porque dicho número tiene un significado especial. Pero es sólo un ejemplo, ya tengo las figuras acabadas y seguramente usaréis lo que viene a continuación, la configuración automática :-)

Enlace automático con los otros AP

Habéis visto que hay que especificar manualmente la dirección MAC de cada AP con el que queremos crear un enlace. Pero hay una forma de hacerlo que sea automático. Esto es muy útil, sobre todo si estáis haciendo pruebas con tarjetas distintas, donde cada una de ellas tienen diferentes MACs, y el procedimiento de cambiar la configuración cada vez es



muy tedioso.

En el caso de que queramos hacer los enlaces automáticamente, **primero hay que crear todas las interfaces *wlan#wds#* que hagan falta, una para cada posible AP**. Luego hay que habilitar en enlace automático con el comando *iwpriv* (incluido en el paquete *wireless-tools*) y el *prism2_param* que viene en el subdirectorio *utils* del *hostap*.

El procedimiento completo para todos los APs es el siguiente (en el ejemplo creamos dos interfaces *wds*, para enlazar con dos APs distintos):

```
# la primera interfaz wds
iwpriv wlan0 wds_add 00:00:00:00:00:00
ifconfig wlan0wds0 0.0.0.0
brctl addif br0 wlan0wds0
# la segunda interfaz wds
iwpriv wlan0 wds_add 00:00:00:00:00:00
ifconfig wlan0wds1 0.0.0.0
brctl addif br0 wlan0wds1
# habilitamos en enlace automático
prism2_param wlan0 autom_ap_wds 1
prism2_param wlan0 other_ap_policy 1
# agregamos ambas interfaces al bridge
brctl addif br0 wlan0wds0
brctl addif br0 wlan0wds1
```

Si todo funciona correctamente, podéis mirar en el */var/log/syslog* y tenéis que ver un mensaje como el mostrado a continuación cuando se crea la conexión entre los dos AP:

```
... wlan0: adding automatic WDS connection to AP 00:50:c2:01:96:14
... wlan0: using pre-allocated WDS netdevice wlan0wds0
```

Por supuesto, siempre es de mucha ayuda en estos casos ver el estado de las tablas del *bridge*:

```
ponti:~# brctl showmacs br0
port no mac addr is local? ageing timer
1 00:04:76:26:96:c7 no 0.12
3 00:05:5d:d6:4b:da no 0.43
3 00:30:65:1d:e6:3a no 222.69
1 00:40:43:05:66:00 no 19.21
3 00:50:c2:01:96:14 yes 0.00
2 00:50:c2:01:96:14 yes 0.00
1 00:60:08:b3:6c:b7 yes 0.00
```

Repetidor

Por si no os habéis dado cuenta, si no le configuráis una interfaz LAN (ethernet), **el AP funcionará solamente como repetidor *wireless***. Fácil y guapo ¿no?. Pero tened en cuenta que ambos están en el mismo canal, por lo que el tráfico se duplica o triplica y además se crean interferencias (ver las consideraciones finales). Pero sirve muy bien para poner a ambos a suficiente distancia y así ampliar el alcance de la red inalámbrica (mientras mayor sea la distancia entre las estaciones conectadas a ambos AP, menor será la interferencia entre ellas).

Configuración en Debian

Si, ya lo sé, hacer eso cada vez es un coñazo :-), por eso os muestro mi ejemplo de cómo lo he especificado en el */etc/network/interfaces* de una Debian (que es la única distro que uso para las pruebas, el *apt-get* es insuperable y siempre tengo las últimas versiones :-).

En mi caso se tratan de dos portátiles, cada uno con una Ethernet y una Conceptronic PCMCIA (insisto, antes de hacer estas pruebas, asegurarnos que todo funciona correctamente en modo *Master* leyendo el [artículo](#)⁽¹⁾ que mencioné anteriormente).

Como *ifup/ifdown* son bastantes estrictos (demasiado), puede dar error si se especifica la dirección 0.0.0.0 en la línea del *address*, por lo que a continuación se hace ésto manualmente con el *ifconfig* una vez que la interfaz ya está en



marcha (especificando la *stanza* [sic] **up**) para ponerla en modo promiscuo.

Aseguraros que el comando `prism2_param` (es un script del shell) está en el PATH del root, sino fallará al ejecutarse el *ifup* y quedarán las interfaces mal configuradas (verificar que no haya errores en el `/var/log/syslog`).

Como podéis ver, lo que se hace es crear automáticamente al arranque (*auto br0*) el *bridge br0*. Luego para cada interfaz de red (*eth0* y *wlan0*) las agrega al *bridge br0* en cuanto están configuradas. En el caso de la interfaz *wlan0* también crea el *wlan0wds0*, especifica la política de conexión automática, y también agrega esta interfaz al *bridge br0*.

Todo muy simple y lineal (en realidad se podrían poner más cosas, especialmente para cuando se baja la interfaz y para hacer un *reset*, pero eso ya os dejo como trabajo para casa :-)

```
# /etc/network/interfaces

# The loopback interface
auto lo
iface lo inet loopback

auto br0
iface br0 inet static
    address 192.168.0.3
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    bridge_ports none
    bridge_stp on

iface eth0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    up /usr/sbin/brctl addif br0 eth0

iface wlan0 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    wireless_essid Antoli
    wireless_mode Master
    wireless_key s:poner_tu_clave
    wireless_channel 6
    up ifconfig wlan0 0.0.0.0
    up /usr/sbin/brctl addif br0 wlan0
    up iwpriv wlan0 wds_add 00:00:00:00:00:00
    up ifconfig wlan0wds0 0.0.0.0
    up prism2_param wlan0 autom_ap_wds 1
    up prism2_param wlan0 other_ap_policy 1
    up /usr/sbin/brctl addif br0 wlan0wds0
```

Consideraciones finales

Por supuesto no todo es maravilloso, algunos de los problemas conocidos y que me he encontrado son:

– **Ambos AP tienen que usar el mismo canal**, por lo que el tráfico se duplica o triplica. Por ejemplo supongamos que hay tráfico entre A, una estación *wireless* conectada al AP–A y la estación B, conectada al AP–B. Los paquetes de A hacia B primero van hacia AP–A, luego a AP–B y recién a B, es decir que triplicamos en tráfico en el mismo canal, así que **no lograréis pasar de los 2 Mbps**.

```
gallir@sofi:~$ wget http://192.168.0.10/~gallir/www.tgz
...
Connecting to 192.168.0.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15,763,198 [application/x-tar]

100%[====...=====] 15,763,198  161.43K/s  ETA 00:00

23:17:22 (161.43 KB/s) - `www.tgz.10' saved [15763198/15763198]
```



– En mis pruebas he observado que **si los AP están muy cerca entre ellos se producen muchos descartes de paquetes** haciendo que las tarjetas estén bajando y subiendo continuamente la velocidad de transmisión en caso de tráfico elevado:

```
...
... wlan0: STA 00:05:5d:d6:4b:da TX rate raised to 55
... wlan0: STA 00:05:5d:d6:4b:da TX rate raised to 110
... wlan0: STA 00:05:5d:d6:4b:da TX rate lowered to 55
```

– Si una estación está primero registrado en el AP–A y luego se registra en AP–B, se pierde la conectividad entre la estación y AP–A durante unos cuantos minutos (5, aproximadamente). Este efecto sólo ocurre con la conectividad entre la estación y el AP–A, no así entre la estación móvil y las otras registradas en el AP–A (sí, es muy extraño). **Este problema se solucionará con el protocolo IAPP**, que está en desarrollo en el HostAP, para notificar inmediatamente de las bajas y registros a nuevos APs.

– Por un bug en las versiones anteriores al 1.4.9 del firmware STA de las tarjetas PRISM2, el formato del paquetes WDS no es estándar, **por lo que esas tarjetas no funcionarán con otros AP que no sean del HostAP** (y OpenAP, que tienen un formato similar).

– En las versiones nuevas del HostAP (del CVS) ya se puedes recibir paquetes WDS con el formato estándar, pero sólo en modo *Managed* (en éste artículo lo hacemos todo en modo *Master*).

– Como las tarjetas tienen una sola radio **es imposible con una única tarjeta usar dos canales distintos**, que sería lo ideal para interconectar dos redes o hacer de repetidor con el WDS y dar conectividad a las estaciones clientes por otro canal distinto para evitar las limitaciones de ancho de banda e interferencias. Dada esta limitación, lo ideal sería poner dos tarjetas en un ordenador, usar una para el WDS y la otra para las estaciones inalámbricas.

Por último, tened en cuenta que estamos haciendo ésto para conectar redes LAN (o hacer de repetidor) en forma transparente para tener una sólo red IP. Si tenéis redes IP distintas, el problema es más sencillo de solucionar a nivel de *wireless*, aunqu un poco más complejo a nivel de encaminamiento IP.

En otro artículo hablaremos de configuración de **DHCP, autenticación 802.1X, Radius e IAPP**, pero eso después que lo haya terminado de instalar y configurar en mi laboratorio, es decir en mi casa :–)

Lista de enlaces de este artículo:

1. <http://bulmalug.net/body.phtml?nIdNoticia=1309>
2. <http://bulmalug.net/body.phtml?nIdNoticia=1309&nIdPage=4>
3. <http://hostap.epitest.fi/>

E–mail del autor: gallir@uib.es

Podrás encontrar este artículo e información adicional en: <http://bulmalug.net/body.phtml?nIdNoticia=1624>