

# **EXTENSIÓN DE LAS CAPACIDADES DE MONITORIZACIÓN DE JFFNMS EN ENTORNOS WLAN**

**PFC - INGENIERÍA INFORMÁTICA**

**LUIS LUCEÑO FORNÉ**

**CONSULTOR: VICTOR CARCELER HONTORIA**

**JUNIO 2006**

# ÍNDICE

|   |           |
|---|-----------|
| <b>PLAN DE PROYECTO.....</b>  | <b>4</b>  |
| <b>1. DESCRIPCIÓN DEL PROYECTO.....</b>   | <b>4</b>  |
| 1.1 INTRODUCCIÓN.....   | 4         |
| 1.2 OBJETIVOS.....  | 4         |
| 1.3 RESULTADOS ESPERADOS .....  | 4         |
| 1.4 ANÁLISIS DE RIESGOS.....  | 4         |
| 1.5 ALCANCE DEL PROYECTO .....  | 5         |
| <b>2. ORGANIZACIÓN DEL PROYECTO.....</b>  | <b>5</b>  |
| 2.1 RELACIÓN DE ACTIVIDADES.....  | 5         |
| 2.2 CALENDARIO DE TRABAJO .....   | 6         |
| <b>ESTUDIO PARA EL FUNCIONAMIENTO DEL PUNTO DE ACCESO LINKSYS WRT54G ..</b>   | <b>7</b>  |
| <b>1. INTRODUCCIÓN.....</b>   | <b>7</b>  |
| <b>2. LOCALIZACIÓN FIRMWARE COMPATIBLE CON MIB ESTÁNDAR DE PARÁMETROS DE RADIO COMPATIBLE CON WRT54GS.....</b>          | <b>7</b>  |
| <b>3. OBTENCIÓN Y ALMACENAMIENTO DE PARÁMETROS DE RADIO .....</b>   | <b>8</b>  |
| 3.1 ESCOGIENDO EL PRIMER PARÁMETRO: VELOCIDAD ( WL RATE ).....  | 9         |
| 3.2 IMPLEMENTANDO EL PARÁMETRO: CANAL ( WL CHANNEL ) .....  | 14        |
| <b>ESTUDIO DE LAS CARACTERÍSTICAS Y FUNCIONAMIENTO DE JFFNMS .....</b>  | <b>18</b> |
| <b>1. DESCRIPCIÓN DE JFFNSM.....</b>  | <b>18</b> |
| <b>2. FUNCIONAMIENTO.....</b>   | <b>19</b> |
| <b>AMPLIACIÓN JFFNMS PARA MONITORIZAR LOS PARÁMETROS DE FUNCIONAMIENTO WIFI DEL PUNTO DE ACCESO LINKSYS WRT54 .....</b> | <b>20</b> |
| <b>1. INTRODUCCION.....</b>   | <b>20</b> |
| <b>2. CONFIGURACIÓN E IMPLEMENTACION DE PARÁMETROS: VELOCIDAD (RATE) Y CANAL (CHANNEL) .....</b>                        | <b>20</b> |

|           |   |           |
|-----------|---|-----------|
| 2.1       | ZONA.....   | 20        |
| 2.2       | HOST.....   | 21        |
| 2.3       | INTERFACE .....   | 22        |
| 2.4       | LOCALIZACIÓN PARÁMETROS DE LA INTERFACE .....   | 25        |
| 2.5       | POLLER.....   | 27        |
| 2.6       | GRÁFICOS .....  | 28        |
| 2.7       | SLA Y ALARMAS.....  | 30        |
| <b>3.</b> | <b>AMPLIACIÓN DE PARÁMETROS: NIVEL DE SEÑAL (RSSI) Y CLIENTES ASOCIADOS (ASSOCLIST) .....</b> | <b>31</b> |
| 3.1       | NIVEL DE SEÑAL (RSSI) .....   | 31        |
| 3.2       | CLIENTES ASOCIADOS (ASSOCLIST).....   | 34        |
| 3.3       | COMPLETANDO LA AMPLIACIÓN EN JFFNMS .....   | 38        |
|           | <b>RESUMEN Y CONCLUSIONES .....</b>   | <b>42</b> |
|           | <b>BIBLIOGRAFÍA.....</b>  | <b>44</b> |
|           | <b>ANEXOS.....</b>  | <b>45</b> |
| 1.        | <b>ACTUALIZACIÓN FIRMWARE DD-WRT .....</b>  | <b>45</b> |
| 2.        | <b>INSTALACIÓN JFFNMS .....</b>   | <b>47</b> |

## **PLAN DE PROYECTO**

### **1. DESCRIPCIÓN DEL PROYECTO**

#### **1.1 INTRODUCCIÓN**

Dado que ya es una realidad el hecho de la extensión del uso de las redes inalámbricas, resulta cada vez más necesario disponer de herramientas que permitan monitorizar los dispositivos que la componen, equipos con tarjetas inalámbricas, acces points, routers, etc....

Una de las herramientas GPL más utilizadas para esta monitorización de redes es Just For Fun Network Management System (JFFNMS) basada en el protocolo SNMP. Sin embargo, no cuenta con un soporte especial para entornos WLAN. Por lo tanto se pretende estudiar la manera de extender la aplicación para monitorizar algunos parámetros de interés en las redes WiFi (canal, nivel de señal en el enlace, clientes asociados, velocidad de enlace...).

#### **1.2 OBJETIVOS**

- Estudiar las características y funcionamiento de JFFNMS.
- Estudiar la manera de ampliar las capacidades para monitorizar equipos WLAN.
- Estudiar el funcionamiento del punto de acceso LinkSys WRT54G.
- Ampliar JFFNMS para monitorizar los parámetros de funcionamiento WiFi del punto de acceso LinkSys WRT54.

#### **1.3 RESULTADOS ESPERADOS**

Recoger un conjunto de parámetros de radio de un dispositivo Wireless y poder representarlos en la aplicación de monitorización Just For Fun Network Management System (JFFNMS).

#### **1.4 ANÁLISIS DE RIESGOS**

El desconocimiento a priori, tanto del hardware a monitorizar como del software de monitorización no está exento de que determinadas funcionalidades no pudieran desarrollarse. Sin embargo a medida que transcurren las fases de investigación podremos ir determinando más concretamente si podemos o no culminar los objetivos.

## 1.5 ALCANCE DEL PROYECTO

El proyecto quedará fijado en cuanto al hardware por el router LinkSys Wrt54Gs y en cuanto al software de monitorización por la aplicación JFFNMS sobre plataforma Windows XP Professional.

Los parámetros de radio a monitorizar serán:

- Velocidad de enlace.
- Canal.
- Nivel de señal en el enlace.
- Clientes asociados.

## 2. ORGANIZACIÓN DEL PROYECTO

### 2.1 RELACIÓN DE ACTIVIDADES

- Estudio para el funcionamiento del punto de acceso LinkSys WRT54G:
  - Estudio código fuente sección descripción OID's
  - Identificación OID's de los parámetros de radio
  - Localización firmware compatible con MIB estándar de parámetros de radio compatible con WRT54G
- Estudio de las características y funcionamiento de JFFNMS:
  - Estudio y descripción protocolo SNMP
  - Estudio descripción de MIBS. Localización de editores.
  - Instalación y configuración de JFFNMS
  - Pruebas de detección de dispositivos
  - Estudio funcionalidades
- Ampliación JFFNMS para monitorizar los parámetros de funcionamiento WIFI del punto de acceso LinkSys WRT54:
  - Estudio de la implementación de nuevas funcionalidades en JFFNMS
  - Programación PHP



## ESTUDIO PARA EL FUNCIONAMIENTO DEL PUNTO DE ACCESO

### LINKSYS WRT54G

#### 1. INTRODUCCIÓN

Para poder monitorizar los parámetros del punto de acceso necesitamos en primer lugar, seleccionar de los diversos firmware disponibles para nuestro punto de acceso Linksys WRT54GS, cual de ellos nos permitirá de un forma sencilla, capturar los parámetros de radio que tenemos previstos para nuestro proyecto.

En segundo lugar, cómo podemos obtener con el nuevo firmware instalado los parámetros de radio del AP Linksys WRT54 y guardarlos sin que se pierdan sus valores al reiniciar el router.

En tercer lugar de qué manera podemos guardar los parámetros y en qué lugar de la MIB para que puedan ser accesible desde una aplicación externa como será en nuestro caso JFFNMS.

#### 2. LOCALIZACIÓN FIRMWARE COMPATIBLE CON MIB ESTÁNDAR DE PARÁMETROS DE RADIO COMPATIBLE CON WRT54GS

De las múltiples versiones disponibles de firmware para el router Wrt54GS seleccionamos la versión de DD-WRT para la versión 4 del router Wrt54GS SP1: wrt.v23\_wrt54gsv4.bin.

| Vendor/Project                    | Project               | Latest Revision | Revision Date    |
|-----------------------------------|-----------------------|-----------------|------------------|
| Linksys <a href="#">[website]</a> | Linksys WRT54GS v1-v3 | 4.71.1          | April 25, 2006   |
|                                   | Linksys WRT54GS v4    | 1.06.1          | April 25, 2006   |
|                                   | Linksys WRT54GS v5    | 1.50.5          | February10, 2006 |

| Vendor/Project                   | Project       | Latest Revision | Revision Date       |
|----------------------------------|---------------|-----------------|---------------------|
| DD-WRT <a href="#">[website]</a> | <b>stable</b> | <b>v23-SP1</b>  | <b>May 16, 2005</b> |
|                                  | alpha         | v24             | March 12, 2006      |

|                                |     |       |               |
|--------------------------------|-----|-------|---------------|
| eWRT <a href="#">[website]</a> | 0.4 | final | April 1, 2006 |
|--------------------------------|-----|-------|---------------|

**HyperWRT** [\[website\]](#)

|        |       |                   |
|--------|-------|-------------------|
| stable | 2.0   | February 7, 2005  |
| beta   | 2.1b1 | February 18, 2005 |

**Sveasoft** [\[website\]](#)

|             |                  |                |
|-------------|------------------|----------------|
| stable      | Alchemy 1.0      | April 21, 2005 |
| pre-release | Talisman 1.2 RC1 | May 7, 2006    |

**Thibor** [\[website\]](#) - HyperWRT update

|                           |              |
|---------------------------|--------------|
| HyperWRT Thibor Thibor15c | May 12, 2006 |
|---------------------------|--------------|

Fig. 1: Lista de firmwares compatibles con WRT54GS

El firmware de DD-WRT permite obtener vía SNMP valores del router Linksys WRT54GS.

En el siguiente acceso:

- [http://www.dd-wrt.com/wiki/index.php/What\\_is\\_%22DD-WRT%22%3F#v23\\_File\\_Versions](http://www.dd-wrt.com/wiki/index.php/What_is_%22DD-WRT%22%3F#v23_File_Versions)

Se enumeran todas y cada una de las funcionalidades de la versión que he seleccionado, si bien en nuestro caso la más importante es el soporte de SNMP.

Una vez seleccionado la fabricante y versión del firmware, procedemos a sus instalación: (Ver Anexo)

### 3. OBTENCIÓN Y ALMACENAMIENTO DE PARÁMETROS DE RADIO

Los parámetros que necesitamos están bajo el conjunto de comandos: wl

Así pues disponemos de los siguientes comandos wlan:

```

~ # wl cmds
a_rate          del_ie          powerindex      scan_channel_time
a_mrate         eap             phyreg          scan_unassoc_time
ap              evm             plcphdr         scan_home_time
atten           frag            phytype         scan_passive_time
antdiv          fgacurcy       primary_key     scan_nprobes
addwep          fasttimer      passive         ssid
aes             frameburst     prb_resp_timeout shortslot
authorize       gmode          phylist         shortslot_override
assoc           gmode_protection_pktcnt shortslot_restrict
assoclist       gmode_protection_control pwr_percent sch_timeout
aciargs         gmode_protection_cts quiet slowtimer
auth_sta_list   gmode_protection_override restart sta_info
auth_sta_list   glacialtimer   reboot          tssi
add_ie          infra          radio           txpwr
bg_rate         int            revinfo         txpwr1
bg_mrate        ignore_bcns    rts             txpathpwr
bssid           isup           rate            txpwrlimit
band            interference   radioreg        txant
bands           join           ratedump        tkip
bi              keys           rateparam       tsc
cmds            lodump         rmwep           tkip_countermeasures
clk             lrl            regulatory      up
cwmin           locale         rateset         ucflags
cwmax           lazywds        roam_trigger    upgrade
channel         longtrain      roam_delta      ver
chanlist        lbt            roam_scan_period wake
channels        legacy_erp     rssi            wepstatus
channels_in_country list_ie        radar           wep
curpower        msglevel      rssidump        wsec_test
channel_ga       monitor        rm_req          wsec_restrict
channel_ga_start mrate         rm_rep          wsec
country         macreg         rand            wpa_auth
crsuprs         mac            srddump         wds
csa             macmode        srl             wet
constraint      measure_req   shm            wds_remote_mac
cap             malloc_dump   scbdump         wds_wpa_role_old
chan_info       nvdump        set_pmk         wds_wpa_role
down            nvset         scan            wme
dump            nvget         spect           wme-ac
deauthorize     noise         scanresults
deauthenticate  out           status
disassoc        PM            scansuppress
dtim            promisc       suprates
~ #

```

Fig. 2.1: Comandos wl.

### 3.1 Escogiendo el primer parámetro: velocidad ( wl rate )

Consultamos en la documentación del firmware dónde se encontraban los parámetros de radio y cómo se ejecutaban así que localizamos el comando wl rate y obtuvimos la siguiente información:

*rate* force a fixed rate:

- valid values for 802.11a are (6, 9, 12, 18, 24, 36, 48, 54)
- valid values for 802.11b are (1, 2, 5.5, 11)
- valid values for 802.11g are (1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54)
- -1 (default) means automatically determine the best rate

Fig. 2.2 Sintaxis comando: *rate*.

Comprobamos el comando en cuestión:

- `wl rate`

```

C:\ Telnet 192.168.1.1
DD-WRT v23 SP1 std Date: 05/07/06 (c) 2006 NewMedia-NET GmbH
login: root
Password:
-----
DD-WRT build #23 SP1
some code portions OpenWRT and EWRT
additional thanks to Cesar Gonzales, Toxic,
Elektik, MBChris, Mbd, TheIndividual, Pierre Belanger
and all the wonderful supporters of this Project

http://www.dd-wrt.com
-----
Now DD-WRT UeryBusyBox v1.01 (2006.05.07-01:23+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.
~ # wl rate
rate is 54 Mbps
~ #
    
```

Fig. 2.3: Ejecución comando: *wl rate*.

y nos devuelve tal como esperamos la velocidad de nuestro router: 54 Mbps.

A continuación, hemos de averiguar de qué manera podemos recoger este valor, a través del correspondiente OID dentro de la MIB.

Uno de los primeros problemas fue ver cómo podíamos almacenar este valor, ya que al apagar el router este valor se perdía. En la documentación del firmware de DD-WRT se recomendaba guardar esta instrucción en un fichero dentro de una partición *jffs*, en concreto en:

- `/jffs/snmpd/wlrate`

Para ello teníamos que configurar el sistema *jffs*.

Journalized Flash File System (JFFS) se usa para almacenar programas de usuario y datos en la memoria flash.

Para habilitar en nuestro router, *jffs*, introdujimos los siguientes comandos:

- `nvramp set sys_enable_jffs2=1`
- `nvramp set sys_clean_jffs2=1`
- `nvramp set jffs_mounted=1`
- `nvramp commit`
- `reboot`

Pero el router WRT54GS no disponía de la suficiente memoria, por lo que no nos quedaba espacio para almacenar el valor del parámetro *rate*, ya que la partición *jffs* está totalmente ocupada con la versión standard del firmware de DD-WRT.

```
Now DD-WRT UeryBusyBox v1.01 (2006.05.07-01:23+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ # df
Filesystem            1k-blocks    Used Available Use% Mounted on
/dev/root              2880        2880      0 100% /
/dev/mtdblock/4       320         320      0 100% /jffs
~ #
```

Fig. 2.4: Porcentaje de ocupación particiones del router Linksys.

Otra posibilidad era instalar otras versiones del firmware como la versión *mini*: entonces sí que disponía de más espacio *jffs*, de hecho me indicaba que tenía ocupada un 75 % de la partición *jffs*, pero me di cuenta que esta versión del firmware que es una versión reducida, no contaba con el módulo Snmp que necesitaba para capturar los parámetros por lo que tuve que desinstalar e instalar de nuevo la anterior.

Así que consultando en varios foros vi como alternativa a la opción de utilizar *jffs*, crear unas variables en la NVRAM para asegurarme de la persistencia de este fichero.

A través de la variable `rc_startup` en la memoria flash también conocida como NVRAM variable, podemos conservar en cada reinicio del router los valores almacenados y asegurar que siempre que el router está en funcionamiento disponemos de estos ficheros.

```

~ #
~ # nvram get rc_startup
/bin/echo "#!/bin/sh
/usr/sbin/wl rate | /usr/bin/cut -b 9-10
" > /tmp/snmp.wlrate ; /bin/chmod 755 /tmp/snmp.wlrate
~ #

```

Fig. 2.5: Contenido variable de inicio: `rc_startup` para obtener el valor de la velocidad.

De forma que cuando arranca el router se ejecuta el comando `echo` que nos muestra la sentencia que nos proporcionará el valor de la velocidad, con el comando `cut` aislamos sólo el valor (54) y lo redireccionamos a un fichero que creamos cada vez que se inicia el router: `wlrate.snmp`

```

~ #
~ # cat /tmp/snmp.wlrate
#!/bin/sh
/usr/sbin/wl rate | /usr/bin/cut -b 9-10
~ #

```

Fig. 2.6: Contenido fichero: `snmp.wlrate`.

A través de la variable `snmpd_conf` creamos la entrada en la MIB del parámetro `wlrate` proporcionando los siguientes parámetros:

- `nvram set snmpd_conf="syslocation <your_location>`
- `syscontact <your_email>`
- `sysname <your_system_name>`
- `rocommunity <your_password>`
- `rwcommunity <your_other_password>`
- `exec wlrate /tmp/snmp.wlrate"`
- `nvram commit`
- `reboot`

Al ejecutar: `exec wlr rate /tmp/snmp.wlr rate` asignamos el alias `wlr rate` al resultado de la instrucción previamente guardada durante la ejecución de la variable `rc_startup` en el fichero que acabamos de crear (`snmp.wlr rate`).

```

~ #
~ # nvram get snmpd_conf
syslocation micasa
syscontact luis
sysname luis
rocommunity public
rwcommunity private
exec wlr rate /tmp/snmp.wlr rate
~ # _

```

Fig. 2.7: Contenido variable: `snmp_conf`

Si ahora nos dirigimos a la entrada de la MIB:

- 1.3.6.1.4.1.2021.8.1

observamos que:

- En 1.3.6.1.4.1.2021.8.1.1.1 se ha creado un nuevo índice con el valor 1 donde figura la primera entrada a la tabla de parámetros.
- En 1.3.6.1.4.1.2021.8.1.1.2 aparece el alias.
- En 1.3.6.1.4.1.2021.8.1.3.1 el fichero donde se encuentra la instrucción que vía SNMP se ha ejecutado.
- En 1.3.6.1.4.1.2021.8.1.100.1 un código de finalización de la instrucción, y finalmente:
- En 1.3.6.1.4.1.2021.8.1.101.1 se muestra el resultado de la instrucción en este caso (54) el valor de la velocidad que buscamos.

```

C:\>
C:\>snmpwalk -c public -v2c -On 192.168.1.1 1.3.6.1.4.1.2021.8.1
.1.3.6.1.4.1.2021.8.1.1.1 = INTEGER: 1
.1.3.6.1.4.1.2021.8.1.2.1 = STRING: wlr rate
.1.3.6.1.4.1.2021.8.1.3.1 = STRING: /tmp/snmp.wlr rate
.1.3.6.1.4.1.2021.8.1.100.1 = INTEGER: 0
.1.3.6.1.4.1.2021.8.1.101.1 = STRING: 54
.1.3.6.1.4.1.2021.8.1.102.1 = INTEGER: 0
.1.3.6.1.4.1.2021.8.1.103.1 = STRING:
C:\>_

```

Fig. 2.8: Contenido entrada índice MIB.

A partir de este momento ya podemos guardar los parámetros de radio que necesitamos y que podremos proporcionar a JFFNMS para su monitorización.

### 3.2 Implementando el parámetro: canal ( *wl channel* )

Para implementar el parámetro canal disponemos del comando *wl channel*:

*channel*

- Set the channel:
- valid channels for 802.11b/g (2.4GHz band) are 1 through 14
- valid channels for 802.11a (5 GHz band) are:
  - 36, 40, 44, 48, 52, 56, 60, 64,
  - 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140,
  - 149, 153, 157, 161,
  - 184, 188, 192, 196, 200, 204, 208, 212, 216

Fig. 2.9: Sintaxis comando: *wl channel*

```

~ #
~ # wl channel
No scan in progress.
current mac channel      11
target channel 11
~ #

```

Fig. 2.10: Ejecución comando: *wl channel*

Para poder aislar el valor del canal (11) utilizaremos la instrucción:

- `/usr/sbin/wl channel | /usr/bin/tail -c 3`

```

/ #
/ # /usr/sbin/wl channel | /usr/bin/tail -c 3
11
/ # _

```

Fig. 2.11: Contenido fichero: *snmp.wlchannel*.

Esta vez en lugar de añadir el parámetro desde la consola del router, hemos decidido utilizar de forma alternativa la interfaz gráfica que ofrece el panel de control de DD\_WRT para la administración del router:

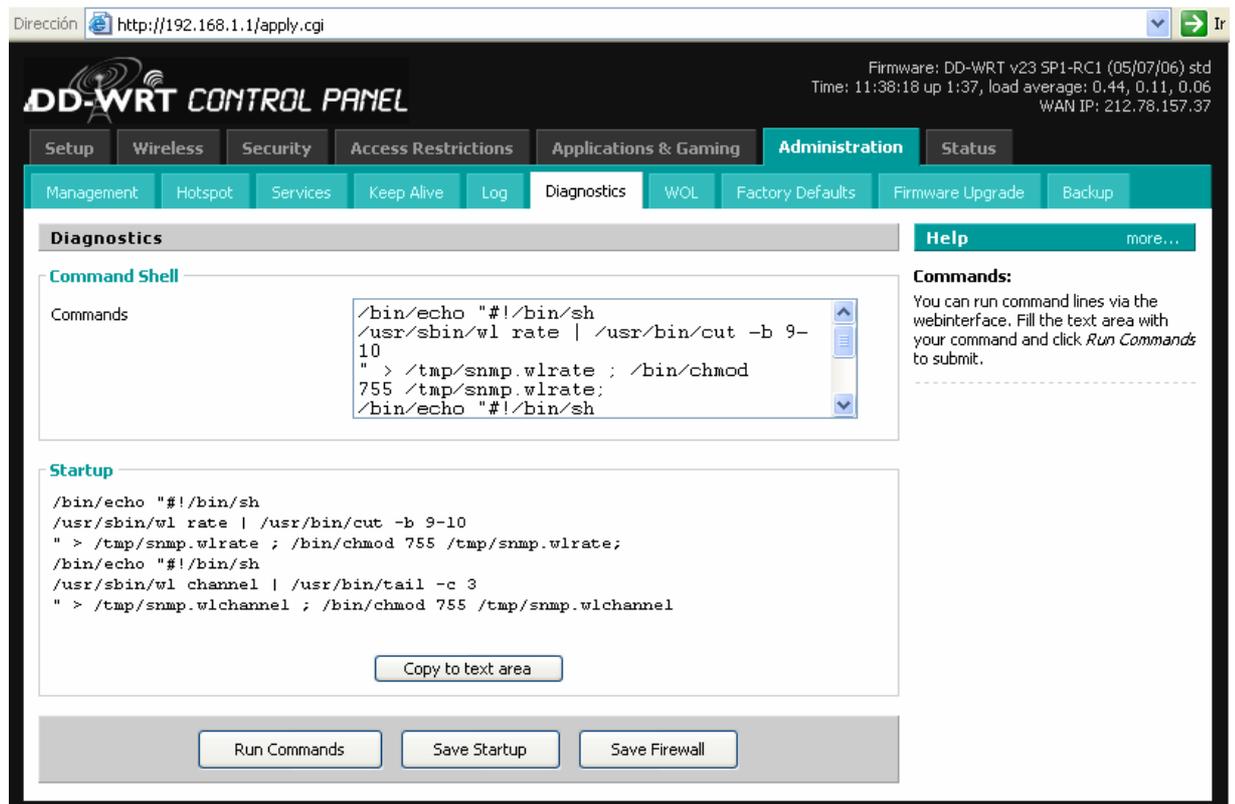


Fig. 2.12: Panel de control del firmware DD-DRT para Linksys Wrt54GS.

Añadimos la ejecución de la variable canal para crear la segunda entrada en el índice de la MIB, esta vez sí desde la consola del router:

```

DD-WRT v23 SP1 std Date: 05/07/06 (c) 2006 NewMedia-NET GmbH

login: root
Password:
-----

                DD-WRT build #23 SP1
                some code portions OpenWRT and EWRT
                additional thanks to Cesar Gonzales, Toxic,
                Elektik, MBChris, Nbd, TheIndividual, Pierre Belanger
                and all the wonderful supporters of this Project

                http://www.dd-wrt.com
-----

Now DD-WRT VeryBusyBox v1.01 (2006.05.07-01:23+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ # nvram set snmpd_conf="syslocation micasa
> syscontact luis
> sysname luis
> rocommunity public
> rwcommunity private
> exec .1.3.6.1.4.1.2021.50.wlrate /tmp/snmp.wlrate
> exec .1.3.6.1.4.1.2021.51.wlchannel /tmp/snmp.wlchannel
> "
~ # nvram
usage: nvram [get name] [set name=value] [unset name] [show]
~ # nvram commit
nvram_commit(): start
nvram_commit(): end
~ # reboot
-----

C:\>snmpwalk -c public -v 2c -On 192.168.1.1 1.3.6.1.4.1.2021.50
.1.3.6.1.4.1.2021.50.1.1 = INTEGER: 1
.1.3.6.1.4.1.2021.50.2.1 = STRING: "wlrate"
.1.3.6.1.4.1.2021.50.3.1 = STRING: "/tmp/snmp.wlrate"
.1.3.6.1.4.1.2021.50.100.1 = INTEGER: 0
.1.3.6.1.4.1.2021.50.101.1 = STRING: "54"
.1.3.6.1.4.1.2021.50.102.1 = INTEGER: 0

C:\>snmpwalk -c public -v 2c -On 192.168.1.1 1.3.6.1.4.1.2021.51
.1.3.6.1.4.1.2021.51.1.1 = INTEGER: 1
.1.3.6.1.4.1.2021.51.2.1 = STRING: "wlchannel"
.1.3.6.1.4.1.2021.51.3.1 = STRING: "/tmp/snmp.wlchannel"
.1.3.6.1.4.1.2021.51.100.1 = INTEGER: 0
.1.3.6.1.4.1.2021.51.101.1 = STRING: "11"
.1.3.6.1.4.1.2021.51.102.1 = INTEGER: 0

C:\>

```

Fig. 2.13: Creación y verificación entrada del parámetro *wl channel* en la MIB.

Observaciones: generalmente al ejecutar:

```
exec NOMBRE PROGRAMA [ARGS ...]
```

```
#
```

```
# NOMBRE: Un nombre genérico: ( Ej.: wlrate)
```

```
# PROGRAMA: El programa a ejecutar incluyendo el path (Ej.: /tmp/snmp.wlrate)
```

```
# ARGS: argumentos opcionales que pueden ser pasados al programa
```

Por omisión crea una entrada con el índice= 1 en la tabla externa de la MIB en el OID:

.1.3.6.1.4.1.2021.8.1

y después, por cada ejecución de *exec*, debería crearse con el índice incrementado en una unidad:

.1.3.6.1.4.1.2021.8.2

sin embargo, como sólo creaba la primera entrada del índice, hemos tenido que indicarle explícitamente dónde debe crear las sucesivas entradas.

En nuestro caso se lo indicamos de la siguiente manera:

.1.3.6.1.4.1.2021.50: Wlrate

.1.3.6.1.4.1.2021.51: Wlchannel

# ESTUDIO DE LAS CARACTERÍSTICAS Y FUNCIONAMIENTO DE JFFNMS

## 1. DESCRIPCIÓN DE JFFNSM

Las siglas JFFNMS corresponden a Just For Fun Network Management System.

- JFFNMS es un sistema de gestión y monitorización de red diseñado para monitorizar una red IP.
- Es un proyecto desarrollado por Javier Szyszlican.
- Se encuentra en constante desarrollo.
- Su versión mas reciente es la 0.8.2.
- Se trata de software libre y esta bajo licencia GPL.
- Permite monitorizar una red IP mediante SNMP, Syslog y Tacacs+
- Puede ser utilizado para monitorizar cualquier dispositivo SNMP, servidor, router, puerto TCP o cualquier elemento que se desee siempre que se programe una extensión adecuada a dicho elemento para JFFNMS.
- También dispone de características orientadas al manejo de dispositivos Cisco.
- JFFNMS esta escrito en PHP y funciona en entornos GNU/Linux, FreeBSD y Windows 2000/XP.
- La Consola de Eventos muestra todos los tipos de eventos de manera ordenada en el mismo Display.
- JFFNMS genera gráficas para todos los dispositivos de la red tráfico de red, utilización de CPU, errores, etc.
- Tiene soporte de base de datos ( MySQL o PostgreSQL ), integra logs de Syslog y autenticación e informes de Tacacs+.
- JFFNMS es muy modular y extensible lo que significa que se pueden programar extensiones en caso de que no se disponga de soporte para los elementos específicos de la red.

- JFFNMS se basa en las tecnologías: Apache, Cron, MySQL, PHP, RRDTool y SNMP.
- Dispone de un Mapa de Estado que permite visualizar la red de una manera sencilla.
- La lista de dispositivos, o “Tipos de Interfaz” que JFFNMS puede monitorizar es extensa, y gracias a sus desarrolladores y usuarios está creciendo.
- La comunidad de usuarios de JFFNMS posee una lista de distribución que compensa la carencia de documentación sobre la herramienta.

## 2. FUNCIONAMIENTO

JFFNMS permite monitorizar dispositivos a través de la utilización del protocolo Snmp. La idea es definir una jerarquía de objetos interrelacionados a través de una serie de identificadores , y de unos procesos de polling basados en el planificador del sistema que se ejecutan periódicamente y que tiene por objeto actualizar la información.

Así pues, primero identificamos una red, a continuación los dispositivos que están conectados a ella. En el siguiente nivel las interfaces de cada dispositivo, es decir, todos aquellos elementos que pueden ser monitorizados dentro de un dispositivo: el adaptador de red, la memoria, el almacenamiento, etc.

El proceso consiste primero en definir y configurar los dispositivos a ser monitorizados y de qué manera. Todo ello, a partir de su OID y posición dentro de la MIB; en localizar los valores de las variables que representan estas interfaces y posteriormente vía el protocolo Snmp con las funciones *snmpget* o *snmpwalk*, recoger estos valores.

Una vez localizada la información, *jffnms*, permite, representar la información, determinar el estado de los dispositivos, si están up o down, realizar gráficas de rendimientos, mostrar alarmas cuando se alcanzan ciertos umbrales, generar SLA's de dispositivos e informes.

# AMPLIACIÓN JFFNMS PARA MONITORIZAR LOS PARÁMETROS DE FUNCIONAMIENTO WIFI DEL PUNTO DE ACCESO LINKSYS WRT54

## 1. INTRODUCCION

Este capítulo se divide en dos partes, en la primera explico detalladamente como se definen los objetos en Jffnms y describo como localizo desde Jffnms los parámetros almacenados en la MIB del router. Me centro en los dos primeros parámetros con los que inicie el proyecto que fueron en este orden: velocidad (wl rate), y canal (wl channel). En la segunda parte continúo con la implementación de los parámetros nivel de señal (wl rssi) y clientes asociados (wl associat).

## 2. CONFIGURACIÓN E IMPLEMENTACION DE PARÁMETROS: VELOCIDAD (RATE) Y CANAL (CHANNEL)

### 2.1 ZONA

Primero creamos una nueva Zona desde el menú de administración sobre la opción Hosts and Interfaces / Zones y pulsamos Add

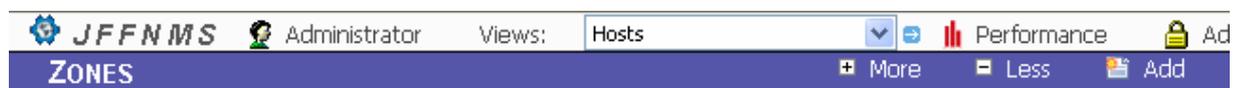


Fig. 4.1: Acceso a menú: Zonas.

Introducimos:

- Nombre de una zona: Zona1
- Nombre abreviado: ZN1
- Imagen: un fichero con extensión .png



Fig. 4: Alta de una nueva zona.

Visibility: Show para que se muestre en pantalla y  
 Network Discovery Seeds CIDR: Red/máscara para localizar los hosts.

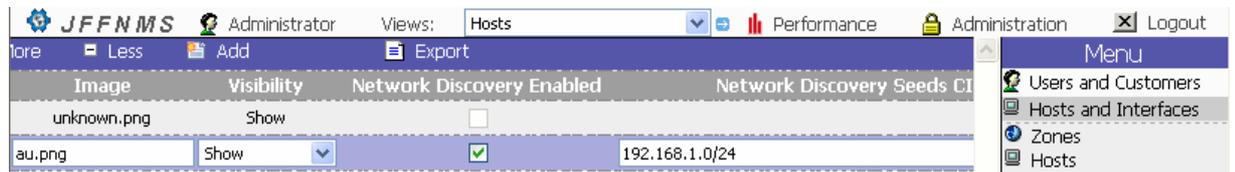


Fig. 4.2: Alta de una nueva zona (1/2).

Introducimos:

- SNMP Communities CSV: public

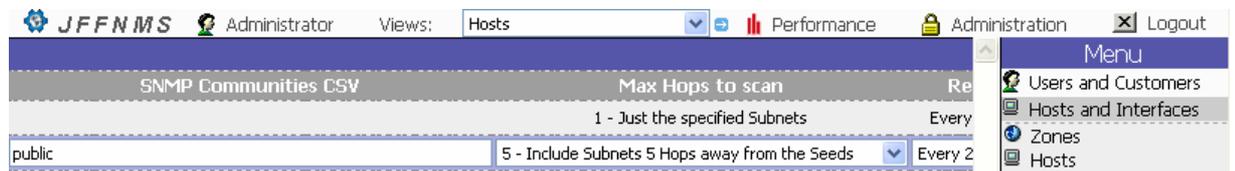


Fig. 4.3: Alta de una nueva zona (2/2).

## 2.2 HOST

Seleccionamos sobre el menú: Hosts y pulsamos Add:



Fig. 4.4: Acceso a menú: Host.

Aparece la pantalla de introducción de datos para un nuevo host:

- Name: Wrt54Gs
- IP address: 192.168.1.1.
- R/O Community: SNMPv1 / public

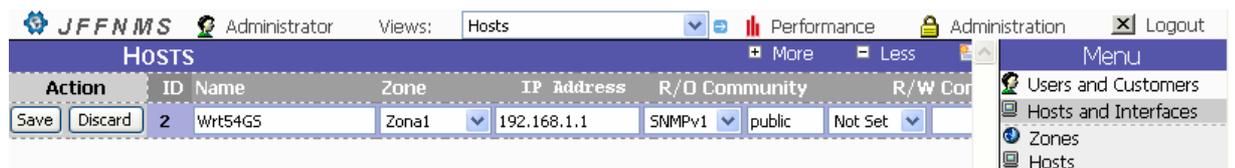


Fig. 4.5: Alta de un nuevo Host.

## 2.3 INTERFACE

Desde Menu / Internal Configuration / Polling & Discovery / interfaces Type, pulsamos Add e introducimos:

- Description: Radio
- Marcamos los campos AD Enabled y Validate in AD
- Discovery function (Internal): radio

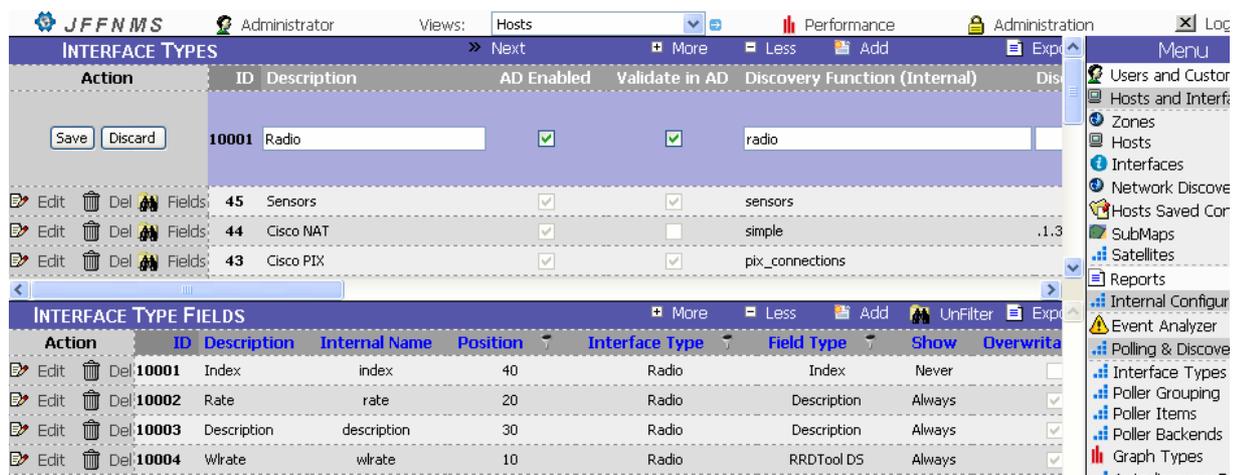


Fig. 4.6: Alta de un nueva Interfaz.

De esta forma configuramos jffnms para que detecte la nueva interfaz (radio) donde aparecerá nuestro primer parámetro rate. Para ello debemos desde la opción Fields crear los campos necesarios:

- Index: de tipo Index no es necesario mostrarlo sólo se utiliza a efectos internos del programa
- Rate: de tipo description pues sí se deberá mostrar en la interfaz cuando ésta sea detectada.
- Channel: de tipo description.
- Rssi: de tipo description.
- Assoclist: de tipo description.
- Description: de tipo description donde podemos indicar una descripción de la interfaz.

- Wlrate: de tipo RRDToolDS para poder capturar vía polling su valor y poderlo representar gráficamente. Como valor por defecto introducimos: Gauge Min:0 Max:1000.
- Wlchannel: de tipo RRDToolDS para poder capturar vía polling su valor y poderlo representar gráficamente. Como valor por defecto Gauge Min:0 Max:1000.
- Wlrsi de tipo RRDToolDS para poder capturar vía polling su valor y poderlo representar gráficamente. Como valor por defecto Gauge Min:0 Max:1000.
- Wlassoclist: de tipo RRDToolDS para poder capturar vía polling su valor.

| Position | Interface Type | Field Type  | Show   | Overwritable                        | Tracked                             | Default Value                |
|----------|----------------|-------------|--------|-------------------------------------|-------------------------------------|------------------------------|
| 40       | Radio          | Index       | Never  | <input type="checkbox"/>            | <input type="checkbox"/>            |                              |
| 20       | Radio          | Description | Always | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |                              |
| 30       | Radio          | Description | Always | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |                              |
| 10       | Radio          | RRDTool DS  | Always | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | Type: Gauge Min: 0 Max: 1000 |

Fig. 4.7: Alta de los campos de la nueva Interfaz.

Nota: en *position* indicamos valores 10,20,30,40 que sirven para indicar la prioridad. Marcamos *Owerwritable* y *Tracked* en *Rate* y *wrate* para que pueda actualizarse su valor en caso necesario

Regresamos a la pantalla de configuración de *Host* y activamos *Manual Discovery* dentro del desplegable *Action*

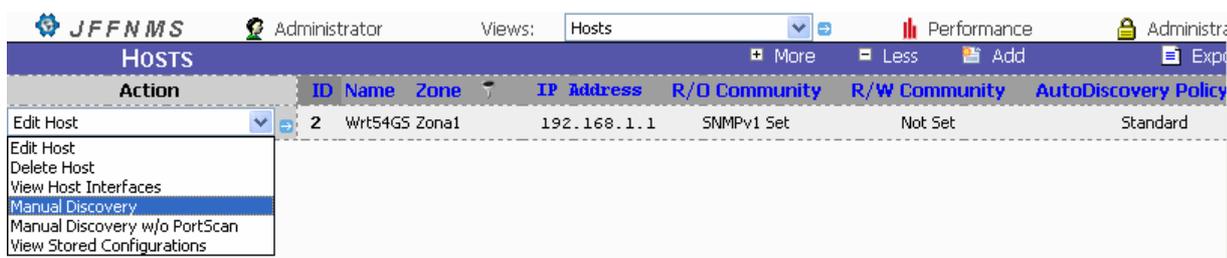


Fig. 4.8: Activación Manual Discovery.

A continuación nos aparecerán las interfaces detectadas para nuestro Host

| JFFNMS Administrator Views: Interfaces Performance Administrati |                                |                 |         |                                |               |             |
|---|--------------------------------|-----------------|---------|--------------------------------|---------------|-------------|
| HOSTS   |                                |                 |         |                                |               |             |
| Action  | ID                             | Name            | Zone    | IP Address                     | R/O Community |             |
| Manual Discovery w/o PortScan                                   | 2                              | Wrt54GS         | Zona1   | 192.168.1.1                    | SNMPv1 Set    |             |
| Storage   |                                |                 |         |                                |               |             |
| Add/Edit  | Index                          | Name            | Status  | Disk Type                      | Size (Bytes)  | Description |
| Edit  | 101                            | Real Memory     | up      | Ram                            | 14409728      |             |
| Edit  | 102                            | Swap Space      | down    | VirtualMemory                  |               |             |
| Edit  | 103                            | Memory Buffers  | down    | Other                          |               |             |
| SQL Query [Manual Add]  |                                |                 |         |                                |               |             |
| Radio   |                                |                 |         |                                |               |             |
| Add/Edit  | Index                          | Name            | Status  | Assoclist                      | Rate          | Channel     |
| Edit  | 1                              | Radio_interface | up      | Hostname: wks01 MAC: 00:0e:35: | 54            | 11          |
| Status  | Assoclist                      | Rate            | Channel | Description                    | Rssi          |             |
| up  | Hostname: wks01 MAC: 00:0e:35: | 54              | 11      | Radio_desc                     | -37           |             |

Fig. 4.9: Descubrimiento de la nueva Interfaz de Radio.

| JFFNMS Administrator Views: Hosts All Interfaces Performance Administrati |                                  |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
|---|----------------------------------|---------------------------------|---------------------------------|------------------------------|---------------------------|-----------------------------|--------------------------------|-----------------------------|--------------------------|----------------------------|------------------------------|
| Root Map Submaps: None Customers: All Options: DHTML                      |                                  |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| TCP Ports   | Port 22 Unknown                  | Port 23 Unknown telnet          | Port 53 Unknown domain          | Port 80 Unknown http         | Port 22 1 Unknown         | Port 23 1 Unknown telnet    | Port 53 1 Unknown domain       | Port 80 1 Unknown http      |                          |                            |                              |
| Card wlan   | vlan1 Unknown 212.78.157:        |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Card br   | b0 Unknown 192.168.1.1           |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Card lo   | lo Unknown 127.0.0.1             |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Card teql   | teq0 Unknown                     |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Card eth  | eth0 Unknown                     |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Card imq  | imq0 Unknown                     | imq1 Unknown                    |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Card eth  | eth1 Unknown                     |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Card wlan   | vlan0 Unknown                    |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Storage   | Real M Unknown Ram 14409728      | Swap S Unknown VirtualMem       | Memory B Unknown Other          |                              |                           |                             |                                |                             |                          |                            |                              |
| Linux/Uni   | CPU Unknown Linux 2.4.33         |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |
| Applioati   | bdflush Unknown Application      | cron Unknown Application        | dnsmasq Unknown Application     | dropbear Unknown Application | httpd Unknown Application | init Unknown Application    | j2_god_mtd Unknown Application | keventd Unknown Application | kl0 Unknown Application  | kswapd Unknown Application | kupdated Unknown Application |
|   | mtdblockd Unknown Application    | process_mor Unknown Application | resetbutton Unknown Application | sh Unknown Application       | snmpd Unknown Application | telnetd Unknown Application | udhcpc Unknown Application     | udhcpd Unknown Application  | upnp Unknown Application | wland Unknown Application  |                              |
| Radio   | Radio_Interf Unknown 54 11 Radio |                                 |                                 |                              |                           |                             |                                |                             |                          |                            |                              |

Fig. 4.10: Representación gráfica de todas las interfaces del access point WRT54GS.



Fig. 4.11: Detalle representación gráfica de la interfaz de parámetros de radio del acces point WRT54GS.

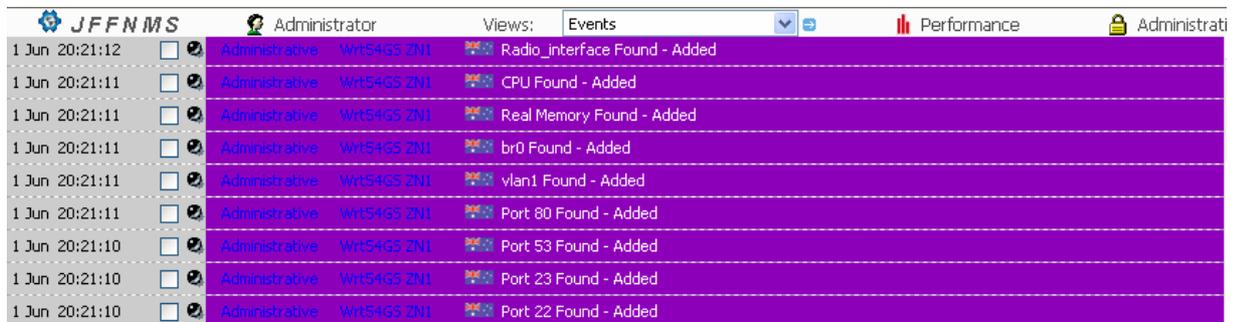


Fig. 4.12: Registro de la incorporación de la nueva interfaz descubierta.

## 2.4 LOCALIZACIÓN PARÁMETROS DE LA INTERFACE

Creación de un plugin para recoger los parámetro de radio el fichero se llama radio.inc.php y debe cumplir las siguientes características

- Ubicación en: `jffnms/engine/discovery`
- Es un requisito indispensable que el formato del fichero sea: `nombre_de_la_interfaz.inc.php`
- La función interna debe llamarse: `discovery_nombre_de_la_interfaz`

```

<?
function discovery_radio($ip,$rocommunity,$hostid,$param) {

    $radio_devices = array();

    if ($ip && $hostid && $rocommunity)
        $deviceIndex = snmp_walk($ip,
$rocommunity, ".1.3.6.1.4.1.2021.50.1.1");
        $deviceWlrate = snmp_get($ip, $rocommunity,
".1.3.6.1.4.1.2021.50.101.1");
        $deviceWlchannel = snmp_get($ip, $rocommunity,
".1.3.6.1.4.1.2021.51.101.1");

        $devInfo = array();
        $devInfo["rate"] = strval($deviceWlrate);
        $devInfo["channel"] = strval($deviceWlchannel);
        $devInfo["interface"] = "Radio_interface";
        $devInfo["description"] = "Radio_desc";
        $devInfo["admin"] = "up";
        $devInfo["oper"] = "up";

        $radio_devices[1] = $devInfo;
    return $radio_devices;
}
?>

```

Fig. 4.13: Código para el plugin del descubrimiento de la interfaz de radio.

La función recibe 4 parámetros:

- \$ip: ip del host.
- \$rocommunity= public
- \$hostid: código que identifica el host dentro de la tabla de hosts
- \$param (opcional)

Devuelve un array completando los campos de la interfaz recogiendo los valores de los OID's designados.

| Action   | ID    | Description  | Name (Match RRD Struct DS) | Poller Command (file) | Parameters            |
|----------|-------|--------------|----------------------------|-----------------------|-----------------------|
| Edit Del | 10004 | Wl assoclist | wlassoclist                | snmp_counter          | .1.3.6.1.4.1.2021.53. |
| Edit Del | 10003 | Wl rssi      | wlrssi                     | snmp_counter          | .1.3.6.1.4.1.2021.52. |
| Edit Del | 10002 | Wl channel   | wlchannel                  | snmp_counter          | .1.3.6.1.4.1.2021.51. |
| Edit Del | 10001 | Wl rate      | wlrate                     | snmp_counter          | .1.3.6.1.4.1.2021.50. |

Fig. 4.14: Configuración pollers para recoger los parámetros de la MIB del router.

| Action        | ID    | Description             | Interface Type        |
|---------------|-------|-------------------------|-----------------------|
| Edit Del View | 10001 | Radio Device            | Radio                 |
| Edit Del View | 46    | Sensors                 | Sensors               |
| Edit Del View | 45    | Cisco NAT               | Cisco NAT             |
| Edit Del View | 44    | PIX Connection Stat     | Cisco PIX             |
| Edit Del View | 43    | IPTable Chain           | IPTables Chains       |
| Edit Del View | 42    | UPS Lines               | UPS Lines             |
| Edit Del View | 41    | UPS                     | UPS                   |
| Edit Del View | 40    | Windows Informant Disks | Windows Logical Disks |

| Action   | ID    | Position | Poller      | Backend         |
|----------|-------|----------|-------------|-----------------|
| Edit Del | 10001 | 10       | Wl rate     | Temporal Buffer |
| Edit Del | 10006 | 11       | Wl channel  | Temporal Buffer |
| Edit Del | 10007 | 12       | Wl rssi     | Temporal Buffer |
| Edit Del | 10010 | 14       | Wl rssi_abs | Temporal Buffer |
| Edit Del | 10009 | 80       | No Poller   | RRDTool All DSs |

Fig. 4.15: Configuración del tipo de poller.

Observaciones: es fundamental completar en el apartado POLLER GROUP - POLLERS/BACKENDRELATION la última línea indicando un tipo de Poller: No Poller y en el campo Backend el valor RRDTool All DSs, ya que de lo contrario el procedimiento poller no genera el fichero: interface-idhost-1.rrd necesario para actualizar los valores de las variables y poder ser representadas gráficamente.

## 2.5 POLLER

Tenemos que modificar POLLER GROUP editando el nuevo ID y cambiando los campos:

- Poller: de No Poller a Wl radio
- Backend: de RRDToolAllDss a Temporal Buffer

Completamos el campo Discovery Default Poller con el valor: Radio Device para enlazar la interfaz de Radio con el Poller anteriormente creado.

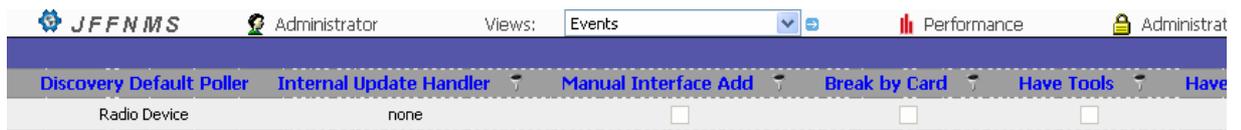


Fig. 4.16: Configuración del identificador de pollers en la interfaz de radio.

## 2.6 GRÁFICOS

Creación de un plugin para recoger los parámetro de radio el fichero se llama radio.inc.php y debe cumplir las siguientes características

- Ubicación en: \jffnms\engine\graphs
- El formato del fichero es: nombre.inc.php
- Requisito de la aplicación Jffnms es que la función interna debe llamarse: graph\_nombredegrafico

En el proyecto hemos definido dos gráficos:

- Gráfico1: para medir el nivel de señal ( graph\_rssi ).
- Gráfico2: para medir velocidad máxima de la conexión ( graph\_rate ).

```
<?
function graph_rssi ($data) {

    $opts_DEF = rrdtool_get_def($data,array('wlrssi'));
    $opts_GRAPH = array(

        "CDEF:rssi_p=wlrssi",
        "LINE3:rssi_p#FF0000:'Total RSSI \:  '",
        "GPRINT:rssi_p:LAST:%6.0lf %sdBm\\n",

    );

    $opts_header[] = "--vertical-label='Used Rssi Abs'";

    return array ($opts_header,
        @array_merge($opts_DEF,$opts_GRAPH));
    }
?>
```

Fig 4.17. : Gráfico 1: Nivel de señal: Rssi

```

<?
function graph_rate ($data) {

    $opts_DEF = rrdtool_get_def($data,array('wrate'));

    $opts_GRAPH = array(
        "LINE3:wrate#FF0000:'Total Rate:  ",
        "GPRINT:wrate:MAX:'%6.0lf %sKBps\\n'",
    );

    $opts_header[] = "--vertical-label='Used Rate'";

    return array ($opts_header,
        @array_merge($opts_DEF,$opts_GRAPH));
    }
?>

```

Fig 4.18 : Gráfico 2: Velocidad: Rate

Desde la pantalla Graph Types del menú de administración definimos la altura y anchura del grafico y la asociamos a un identificador de gráficos que a su vez esta asociado a la interfaz de parámetros de radio.

| ID   | Description | Interface Type | Allow Aggregation        | Graph 1 | Width 1 | Height 1 | Graph 2 | Width 2 |
|------|-------------|----------------|--------------------------|---------|---------|----------|---------|---------|
| 0001 | Wrt54gs Wl  | Radio          | <input type="checkbox"/> | rssl    | 275     | 150      | rate    | 250     |

Fig4.19: Definición de características del gráfico asociado a la interfaz.

Creamos el nuevo tipo de grafico y luego enlazamos la interfaz Radio con el gráfico creado:

| Have Graph                          | Default Graph | Default SLA | RRDTool Structure RRA          | RRDTool Resolution | RRDTool Step |
|-------------------------------------|---------------|-------------|--------------------------------|--------------------|--------------|
| <input checked="" type="checkbox"/> | Wrt54gs Wl    | Wrt54gs     | RRA:AVERAGE:0.5:1:<resolution> | 103680             | 300          |

Fig 4.20 : Asociación entre el nuevo gráfico y la interfaz de radio.

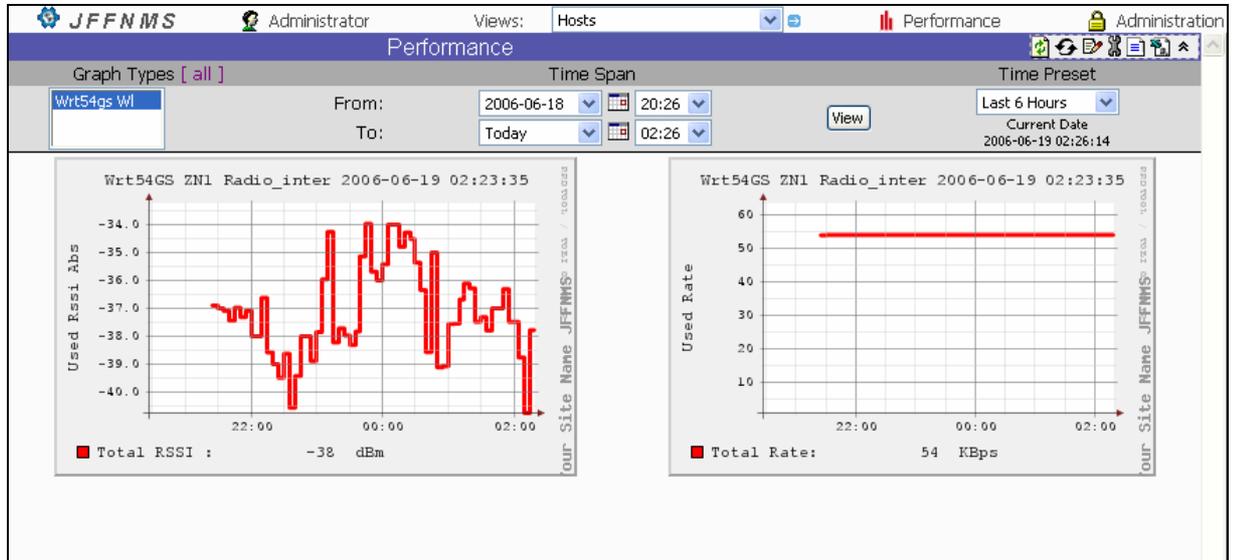


Fig.4.21 : Representación gráfica de los parámetros *Rssi* y *Rate*

## 2.7 SLA Y ALARMAS

SLA es una opción que presenta Jffnms para generar alarmas cuando se sobrepasa de un determinado nivel en parámetro de la interfaz. Se establece una o varias condiciones en las que el valor medido se compara con un valor prefijado. Según el periodo de tiempo especificado que por omisión es de 30 minutos se verifica la condición y si se cumple genera de nuevo el aviso.

En el proyecto definimos un SLA en el que si se sobrepasa la velocidad de conexión en más de 50Kbps, obtenemos un aviso.

The screenshot shows the JFFNMS interface with the 'SLAs' section active. It displays a table of SLA conditions and a configuration form for the selected condition (ID 10001).

| Action   | ID    | Description             | Interface Type         | Event Type | Alarm State | Event Text         | Threshold % (for Reports) |
|----------|-------|-------------------------|------------------------|------------|-------------|--------------------|---------------------------|
| Del View | 10001 | Wrt54gs                 | Radio                  | SLA        | alert       | Evento Radio       | 100                       |
| Del View | 12    | APC UPS                 | APC                    | SLA        | alert       | APC UPS            | 100                       |
| Del View | 11    | Windows CPU             | Windows System Info    | SLA        | alert       |                    | 100                       |
| Del View | 10    | Linux/Unix CPU          | Linux/Unix System Info | SLA        | alert       |                    | 100                       |
| Del View | 9     | Storage                 | Storage                | SLA        | alert       | Storage            | 100                       |
| Del View | 8     | Smokeping Host          | Smokeping Host         | SLA        | alert       | Smokeping:         | 100                       |
| Del View | 7     | Cisco Router            | Cisco System Info      | SLA        | alert       | Router:            | 100                       |
| Del View | 6     | Main Satellite Link     | Physical Interfaces    | SLA        | alert       | Main Sat Link:     | 100                       |
| Del View | 5     | Main Fiber Link         | Physical Interfaces    | SLA        | alert       | Main Link:         | 100                       |
| Del View | 4     | Customer Satellite Link | Physical Interfaces    | SLA        | alert       | Customer Sat Link: | 75                        |

| Action       | ID    | SLA     | Position | Condition      | Show                                |
|--------------|-------|---------|----------|----------------|-------------------------------------|
| Save Discard | 10001 | Wrt54gs | 10       | Rate > 50 Kbps | <input checked="" type="checkbox"/> |

Fig 4.22 : Configuración de SLA para la interfaz de radio.

The screenshot shows a notification bar for a generated event:

1 Jun 20:06:02 SLA Wrt54GS ZN1 Radio\_interface Evento Radio Rate > 50 Kbps: 54 kbps (Unknown Customer 54 11 Radio\_desc)

Fig 4.23 : Evento generado al cumplirse la condición.

### 3. AMPLIACIÓN DE PARÁMETROS: NIVEL DE SEÑAL (RSSI) Y CLIENTES ASOCIADOS (ASSOCLIST)

#### 3.1 NIVEL DE SEÑAL (RSSI)

Nivel de señal (Rssi): el nivel de señal presente en el dispositivo tiene un umbral (threshold) de potencia mínimo que la señal debe alcanzar para lograr una determinada tasa de transferencia (bitrate).

Para implementar el parámetro nivel de señal disponemos del comando *wl rssi*:

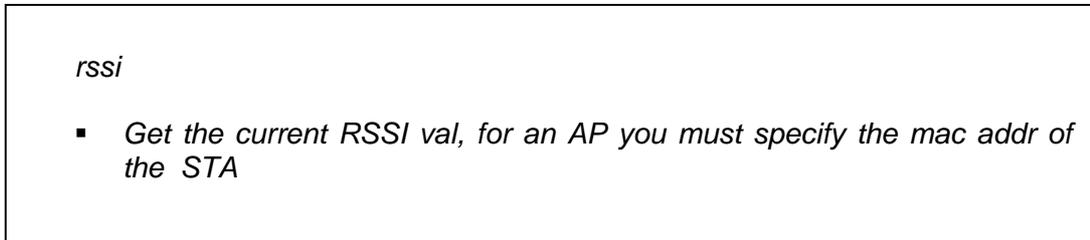


Fig. 4.24: Sintaxis comando: *wl rssi*.

La ejecución del comando que nos proporciona el nivel de señal es la siguiente:

```
~ # /usr/sbin/wl rssi 00-0E-35-01-0C-67
rssi is -40
~ # _
```

Fig. 4.25: Ejecución comando: *wl rssi*.

Para poder aislar el valor del nivel de señal (-45) utilizaremos la instrucción

- `/usr/sbin/wl rssi 00-0E-35-01-0C-67 | /usr/bin/tail -c 4`

```
~ # /usr/sbin/wl rssi 00-0E-35-01-0C-67 | /usr/bin/tail -c 4;
-45
~ #
```

Fig. 4.26: Contenido fichero: *snmp.wlrssi*.

Observaciones: cuando el router opera en modo AP (Acces Point) es necesario proporcionar al comando la dirección MAC de la estación cliente.

Incorporamos (en negrita) la creación del fichero `snmp.wlrssi` en la variable de inicio:

```

/ # nvram get rc_startup
/bin/echo "#!/bin/sh
/usr/sbin/wl rate | /usr/bin/cut -b 9-10
" > /tmp/snmp.wlrate ; /bin/chmod 755 /tmp/snmp.wlrate;
/bin/echo "#!/bin/sh
/usr/sbin/wl channel | /usr/bin/tail -c 3
" > /tmp/snmp.wlchannel ; /bin/chmod 755 /tmp/snmp.wlchannel
/bin/echo "#!/bin/sh
/usr/sbin/wl rssi 00-0E-35-01-0C-67 | /usr/bin/tail -c 4
" > /tmp/snmp.wlrssi ; /bin/chmod 755 /tmp/snmp.wlrssi
/bin/echo "#!/bin/sh

```

Fig. 4.27: Contenido variable: `rc_startup` tras la ampliación del parámetro `wl rssi`.

La instrucción que añadimos en el fichero: `snmpd_conf` para generar la entrada en el índice de la MIB es:

- `exec .1.3.6.1.4.1.2021.52 wlrssi /tmp/snmp.wlrssi`

Incorporamos (en negrita) en la variable `snmpd_conf`, la ejecución del fichero con la instrucción: `snmp.wlrssi`:

```

/ # nvram set snmpd_conf="syslocation micasa
> syscontact luis
> sysname luis
> rocommunity public
> rwcommunity private
> exec .1.3.6.1.4.1.2021.50 wlrate /tmp/snmp.wlrate
> exec .1.3.6.1.4.1.2021.51 wlchannel /tmp/snmp.wlchannel
> exec .1.3.6.1.4.1.2021.52 wlrssi /tmp/snmp.wlrssi"
/ # nvram commit
nvram_commit(): start
nvram_commit(): end
/ # reboot

```

Fig. 4.28: Contenido variable: `snmpd_conf` tras la ampliación del parámetro `wl rssi`.

Comprobamos con:

- `snmpwalk -c public -v2c -On 192.168.1.1 1.3.6.1.4.1.2021.52`

```
C:\>snmpwalk -c public -v2c -On 192.168.1.1 1.3.6.1.4.1.2021.52
.1.3.6.1.4.1.2021.52.1.1 = INTEGER: 1
.1.3.6.1.4.1.2021.52.2.1 = STRING: "wlrssi"
.1.3.6.1.4.1.2021.52.3.1 = STRING: "/tmp/snmp.wlrssi"
.1.3.6.1.4.1.2021.52.100.1 = INTEGER: 0
.1.3.6.1.4.1.2021.52.101.1 = STRING: "-40"
.1.3.6.1.4.1.2021.52.102.1 = INTEGER: 0
```

Fig. 4.29: Comprobación de la entrada del parámetro nivel de señal en la MIB del router.

### 3.2 CLIENTES ASOCIADOS (ASSOCLIST)

Cientes asociados (Assoclist): cada uno de los equipos conectados a una WLAN.

Para implementar este comando utilizaremos un script basado en las variables *assoclist* y en *dumpleases*.

Con el comando *assoclist* obtenemos todos los equipos asociados al access point, pero como sólo nos muestra sus direcciones MAC, añadiremos al script el comando *dumpleases* que nos proporciona el Hostname y la dirección IP de las MAC Adress.

*assoclist*

- *AP only: Get the list of associated MAC addresses*

Fig. 4.30 : Sintaxis comando: *wl assoclist*

```

~ # wl assoclist
assoclist 00:0E:35:01:0C:67
~ # dumpleases
Hostname      Mac Address      IP-Address      Expires in
wks01         00:0e:35:01:0c:67 192.168.1.100   8 minutes, 31 seconds
wks01         00:c0:9f:35:be:da 192.168.1.101   8 minutes, 38 seconds
~ # _

```

Fig. 4.31 : Ejecución de los comandos *wl assoclist* y *dumpleases*

```

~ # while [ 1 ];
> do
> wl assoclist | awk '{print$2}' > /tmp/assocLIST
> # echo "<meta http-equiv='refresh' content='10'><b>Hostnames
and IPs of WLAN
clients</b> (last update: $(date))<p>" > /tmp/www/wlan.html
> while read assocLINE
> do
> dumpleases | grep -i $assocLINE | awk '{print "Hostname: " $1,
"MAC: " $2
, "IP: " $3}'
> # echo "<br>";
> done < /tmp/assocLIST # >> /tmp/www/wlan.html
> sleep 10;
> done;

Hostname: wks01 MAC: 00:0e:35:01:0c:67 IP: 192.168.1.101

```

Fig. 4.32: Obtención de Hostname, dirección mac y dirección IP de los clientes asociados.

Ampliamos `rc_startup` para introducir el parámetro clientes asociados:

```

/ # nvram get rc_startup
/bin/echo "#!/bin/sh
/usr/sbin/wl rate | /usr/bin/cut -b 9-10
" > /tmp/snmp.wlrate ; /bin/chmod 755 /tmp/snmp.wlrate;

/bin/echo "#!/bin/sh
/usr/sbin/wl channel | /usr/bin/tail -c 3
" > /tmp/snmp.wlchannel ; /bin/chmod 755 /tmp/snmp.wlchannel;

/bin/echo "#!/bin/sh
/usr/sbin/wl rssi 00-0E-35-01-0C-67 | /usr/bin/tail -c 4
" > /tmp/snmp.wlrssi ; /bin/chmod 755 /tmp/snmp.wlrssi;

/bin/echo "#!/bin/sh
# mkdir -p /tmp/www
/usr/sbin/wl assoclist | awk '{print\$2}' > /tmp/assocLIST
# echo "<meta http-equiv=\"refresh\"
content=\"10\"><b>Hostnames and IPs of WLAN clients</b> (last
update: $(date))<p>\" > /tmp/www/wlan.html
while read assocLINE
do
    /usr/sbin/dumpleases | /bin/grep -i \$assocLINE | awk '{print
\"Hostname: \" \$1, \"MAC: \" \$2, \"IP: \" \$3}'
# echo \"<br>\";
done < /tmp/assocLIST;
" > /tmp/snmp.wlassoclist ; /bin/chmod 755 /tmp/snmp.wlassoclist;
/ #

```

Fig 4.33: Obtención de Hostname, mac e IP de los clientes asociados.

Nota: tenemos que añadir el carácter `\` antes de los parámetros: `\$1`, `\$2`, `\$3` y `\$assocLINE`. Además tenemos que indicar explícitamente el path completo del comando: `grep`, sino no se genera el fichero `snmp.wlassoclist`.

```

/ # nvram set snmpd_conf="mylocation micasa
> syscontact luis
> sysname luis
> rocommunity public
> rwcommunity private
> exec .1.3.6.1.4.1.2021.50 wlrates /tmp/snmp.wlrates
> exec .1.3.6.1.4.1.2021.51 wlchannel /tmp/snmp.wlchannel
> exec .1.3.6.1.4.1.2021.52 wlrssi /tmp/snmp.wlrssi
> exec .1.3.6.1.4.1.2021.53 wlrssi_abs /tmp/snmp.wlrssi_abs
> exec .1.3.6.1.4.1.2021.54 wlassoclist /tmp/snmp.wlassoclist
"
/ # nvram commit
nvram_commit(): start
nvram_commit(): end
/ # reboot

```

Fig. 4.34: Implementación variable snmpd\_conf con la ampliación de wlassoclist

Comprobamos con:

- snmpwalk -c public -v2c -On 192.168.1.1 1.3.6.1.4.1.2021.54

```

C:\>snmpwalk -c public -v2c -On 192.168.1.1 1.3.6.1.4.1.2021.54
.1.3.6.1.4.1.2021.54.1.1 = INTEGER: 1
.1.3.6.1.4.1.2021.54.2.1 = STRING: "wlassoclist"
.1.3.6.1.4.1.2021.54.3.1 = STRING: "/tmp/snmp.wlassoclist"
.1.3.6.1.4.1.2021.54.100.1 = INTEGER: 0
.1.3.6.1.4.1.2021.54.101.1 = STRING: "Hostname: wks01 MAC: 00:0e:35:01:0c:67 IP:
192.168.1.100"
.1.3.6.1.4.1.2021.54.102.1 = INTEGER: 0
C:\>_

```

Fig. 4.35: Comprobación creación entrada en la MIB de clientes asociados.

Observaciones: las pruebas han sido efectuadas teniendo en cuenta que había un solo cliente asociado (con el que realizo las pruebas), no obstante para comprobar si el procedimiento sería igualmente válido si hubiera más de uno, he simulado una segunda conexión esta vez teniendo en cuenta la segunda interfaz de mi portátil, filtrando el comando assoclist y ejecutando solamente el comando *dumpleases* con lo cual he podido comprobar que se generaban correctamente dos entradas en la MIB como resultado de la ejecución del comando:

```

C:\>snmpwalk -c public -v2c -On 192.168.1.1 1.3.6.1.4.1.2021.53

.1.3.6.1.4.1.2021.53.1.1 = INTEGER: 1
.1.3.6.1.4.1.2021.53.2.1 = STRING: "wlassoclist"
.1.3.6.1.4.1.2021.53.3.1 = STRING: "/tmp/snmp.wlassoclist "
.1.3.6.1.4.1.2021.53.100.1 = INTEGER: 0
.1.3.6.1.4.1.2021.53.101.1 = STRING: "Hostname      Mac Address      IP-Address
Expires in"

.1.3.6.1.4.1.2021.53.101.2 = STRING: "              00:00:00:00:00:00 192.168.1.100  4
minutes, 28 seconds"

.1.3.6.1.4.1.2021.53.101.3 = STRING: "wks01          00:0e:35:01:0c:67 192.168.1.101  9
minutes, 45 seconds"

.1.3.6.1.4.1.2021.53.101.4 = STRING: "wks01          00:c0:9f:35:be:da 192.168.1.102  8
minutes, 1 seconds"

```

Fig. 4.36: Comprobación creación múltiple de clientes asociados en la MIB del router.

### 3.3 COMPLETANDO LA AMPLIACIÓN EN JFFNMS

Una vez finalizada la fase de incorporación de los nuevos parámetros en el router procedemos a ampliar el plugin para el descubrimiento de nuevas interfaces de tipo radio incorporando la detección de los nuevos parámetros: *rssi* y *assoclist*. Para ello modificamos el fichero *radio.inc.php*, ubicado en `jffnms/engine/discovery`:

```

function discovery_radio($ip,$rocommunity,$hostid,$param) {

    $radio_devices = array();

    if ($ip && $hostid && $rocommunity)
        $deviceIndex = snmp_walk($ip,
$rocommunity, ".1.3.6.1.4.1.2021.50.1.1");
        $deviceWlrate = snmp_get($ip, $rocommunity,
".1.3.6.1.4.1.2021.50.101.1");
        $deviceWlchannel = snmp_get($ip, $rocommunity,
".1.3.6.1.4.1.2021.51.101.1");
        $deviceWlrssi = snmp_get($ip, $rocommunity,
".1.3.6.1.4.1.2021.52.101.1");
        $deviceWlrssi_abs = snmp_get($ip, $rocommunity,
".1.3.6.1.4.1.2021.53.101.1");
        $deviceWlassoclist = snmp_get($ip, $rocommunity,
".1.3.6.1.4.1.2021.54.101.1");
        $devInfo = array();
        $devInfo["rate"] = strval($deviceWlrate);
        $devInfo["channel"] = strval($deviceWlchannel);
        $devInfo["rssi"] = strval($deviceWlrssi);
        $devInfo["rssi_abs"] = strval($deviceWlrssi_abs);
        $devInfo["assoclist"] = strval($deviceWlassoclist);
        $devInfo["interface"] = "Radio_interface";
        $devInfo["description"] = "Radio_desc";
        $devInfo["admin"] = "up";
        $devInfo["oper"] = "up";
        $radio_devices[1] = $devInfo;
        return $radio_devices;
    }
}

```

Fig. 4.37: Contenido de *discovery\_radio* tras la ampliación de los parámetros *rssi* y *assoclist*.

Una vez creada la interfaz, si queremos más adelante actualizarla incorporando nuevas variables, no es suficiente con crear las variables y esperar que el proceso de polling las genere en la interfaz, ya que solamente se recogerían los valores pero no los mostraría en la pantalla de interfaces, sino que será necesario previamente borrar la interfaz, para ello:

- Desde el menú de *Administración* opción *Hosts*, activamos la opción: *View Hosts Interfaces*.
- Marcamos la interfaz a actualizar y con la opción *Del* la borramos.

| INTERFACES                          |    |             |                 |  |                               |
|-------------------------------------|----|-------------|-----------------|--|-------------------------------|
| Applications                        | ID | Host        | Interface Name  | Process Name   | Description                   |
| <input type="checkbox"/>            | 59 | Wrt54GS ZN1 | telnetd         | telnetd  | Application /usr/sbin/telnetd |
| <input type="checkbox"/>            | 60 | Wrt54GS ZN1 | udhcpd          | udhcpd   | Application udhcpd            |
| <input type="checkbox"/>            | 61 | Wrt54GS ZN1 | udhcpd          | udhcpd   | Application udhcpd            |
| <input type="checkbox"/>            | 62 | Wrt54GS ZN1 | upnp            | upnp   | Application upnp              |
| <input type="checkbox"/>            | 63 | Wrt54GS ZN1 | wland           | wland  | Application /sbin/wland       |
| Radio                               | ID | Host        | Interface Name  | Assoclist  | Rate                          |
| <input checked="" type="checkbox"/> | 68 | Wrt54GS ZN1 | Radio_interface | Hostname: wks01 MAC: 00:0e:35:01:0c:67 IP: 192.168.1.100 | 54                            |

Fig. 4.38: Eliminación de la interfaz de radio obsoleta.

Ejecutar desde el Menú de Administración, opción Hosts and Interfaces, submenú Hosts, seleccionar en el desplegable, la opción: Manual discovery w/o ports. De esta forma volvemos a volver a crear la interfaz con todos los campos completos

| HOSTS                         |    |         |       |             |               |               |                      |
|-------------------------------|----|---------|-------|-------------|---------------|---------------|----------------------|
| Action                        | ID | Name    | Zone  | IP Address  | R/O Community | R/W Community | AutoDiscovery Policy |
| Manual Discovery w/o PortScan | 2  | Wrt54GS | Zona1 | 192.168.1.1 | SNMPv1 Set    | Not Set       | Standard             |

Fig. 4.39: Localización de la nueva interfaz de radio.

Se nos presenta en negrita la nueva interfaz hallada, y la posibilidad de añadirla al resto de interfaces del Host, para ello: marcamos la interfaz en Add/Edit

|      |     |                |      |               |          |
|------|-----|----------------|------|---------------|----------|
| Edit | 101 | Real Memory    | up   | Ram           | 14409728 |
| Edit | 102 | Swap Space     | down | VirtualMemory |          |
| Edit | 103 | Memory Buffers | down | Other         |          |

| Radio                               |       |                        |           |   |           |    |
|-------------------------------------|-------|------------------------|-----------|---|-----------|----|
| Add/Edit                            | Index | Name                   | Status    | Assoclist                                     | Rate      | Ch |
| <input checked="" type="checkbox"/> | 1     | <b>Radio_interface</b> | <b>up</b> | <b>Hostname: wks01 MAC: 00:0e:35:01:0c:67</b> | <b>54</b> |    |

Fig 4.40: Detección de la nueva interfaz de radio.

y pulsando Add Marked Interfaces se nos añade de nuevo la interfaz con la totalidad de sus campos.

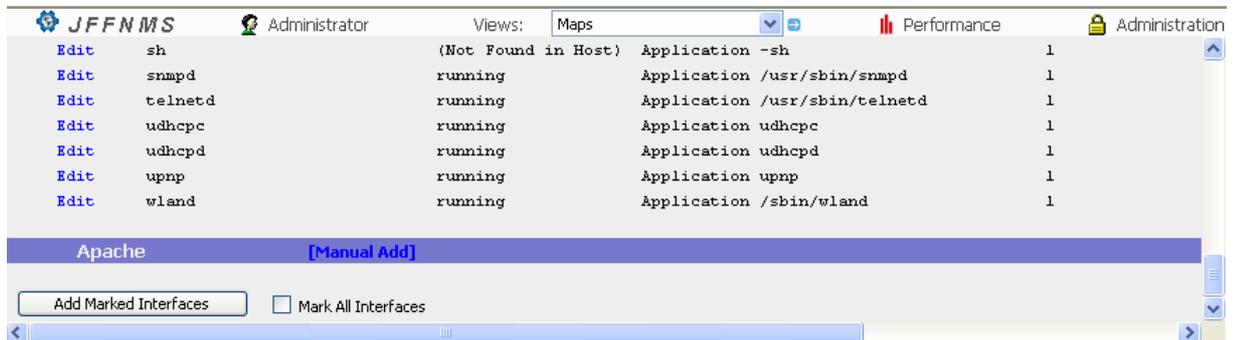


Fig. 4.41: Confirmación para añadir la nueva interfaz de radio.

En la siguiente pantalla podemos observar que se han incorporado junto a los campos Rate y Channel, las variables: RSSI que indican la potencia de la señal -44 dBm y Assolist, que indica la lista de clientes asociados a este Access Point, indicando nombre del equipo (Hostname), dirección Mac (MAC) y su dirección IP (IP).

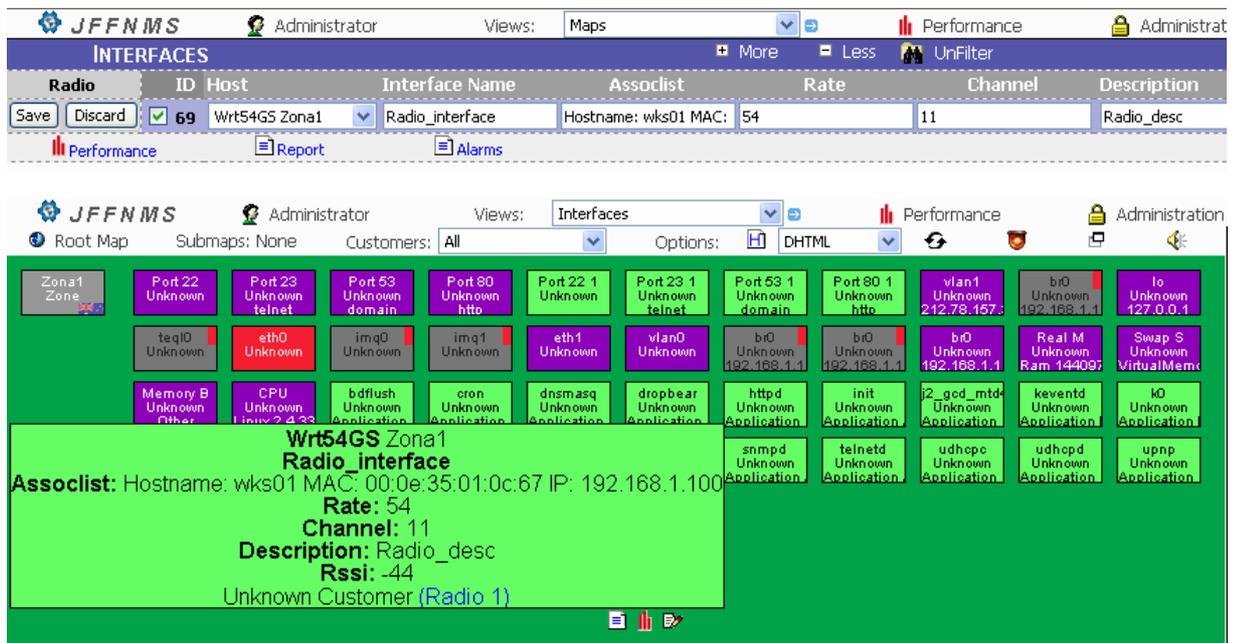


Fig. 4.32: Detalle de la interfaz de radio recogiendo todos los parámetros de radio detectados en Wrt54GS.

## RESUMEN Y CONCLUSIONES

- Para monitorizar los parámetros de radio del router acces point Linksys Wrt54GS desde la aplicación Jffnms, hemos tenido que seleccionar primero, el firmware del router más adecuado para localizar los parámetros de radio.
- A continuación hemos tenido que averiguar como podemos recoger estos parámetros y almacenarlos para su posterior tratamiento, para ello hemos estudiado las funciones que proporciona el protocolo snmp, y estudiar en qué OID's de la MIB podemos guardar estos valores.
- Un vez averiguada esta información hemos procedido a la realización de una serie de scripts para ser ejecutados por la variable de inicio del router.
- Una vez finalizada la fase de estudio e implementación en el acces point, procedimos al estudio de la aplicación Jffnms para conocer sus funcionalidades y ver de qué manera podíamos ampliar la aplicación con una nueva interfaz de parámetros de radio.
- De forma que tras estudiar como se interrelacionan los diferentes objetos en la aplicación, primero creamos un nuevo host (router wrt54gs), una nueva interfaz que llamamos radio, un script php para recoger los parámetros de la MIB del router. Una serie de objetos (wlrates, wlchannel, wlrssi, wlassoclis) de las cuales definimos los OID's donde Jffnms puede recoger sus valores.
- Una vez cargados estos valores en la variables definidas, podemos representarlas en la pantalla principal de interfaces, mostrando sus valores.
- A continuación, definimos un SLA para comprobar que podemos recibir una alarma cuando se sobrepasa de un determinado umbral, en concreto definimos, que cuando se superen los 50 Kbps en la velocidad de conexión, recibamos un aviso.
- Por último implementamos también unos gráficos para poder observar la variación del nivel de señal y de la velocidad máxima de conexión.

- Ha sido una experiencia muy interesante, más en su investigación que en la propia implementación, ya que a pesar de localizar suficiente documentación en Internet, muchos de los scripts propuestos para desarrollar funciones similares no eran del todo aplicables aunque sí servían como guía y entonces tenía que investigar la forma de adaptarlos. La implementación en el caso de JFFNMS, una vez hallada la manera de ampliar las funcionalidades para la detección de los parámetros de radio, supone un gran número de operaciones de inserción en numerosas tablas, y quizás como aspecto de mejora podría diseñarse algún script que parametrizara todas estas operaciones de forma automática, para evitar posibles errores en alguno de los pasos.

## BIBLIOGRAFÍA

1. Firmware Tracker  
<http://www.linksysinfo.org/modules.php?name=Content&pa=showpage&pid=35>
2. SP1: <http://www.dd-wrt.com/dd-wrtv2/index.php>
3. Actualización Firmware DD-WRT (Anexo1)  
[http://www.sincables.cl/index.php?option=com\\_content&task=view&id=38&Itemid=1](http://www.sincables.cl/index.php?option=com_content&task=view&id=38&Itemid=1)
4. DD-WRT Wiki: <http://wrt-wiki.bsr-clan.de/index.php?title=SNMP> (rate)
5. Berliner Quakeforen: <http://forum.bsr-clan.de/viewtopic.php?p=11432> (rate)
6. Configuración SNMPD: [http://www.snmpview.de/snmpd\\_conf.php](http://www.snmpview.de/snmpd_conf.php)
7. JFFNMS: <http://www.jffnms.org>
8. Teoría parámetros <http://www.paramowifix.net/antenas/calculoenlacewlan.html>
9. Ejemplo Script monitorización  
[http://www.dd-wrt.com/wiki/index.php?title=Script\\_Examples](http://www.dd-wrt.com/wiki/index.php?title=Script_Examples)
10. Ejemplo de snmpd [http://www.snmpview.de/snmpd\\_conf.php](http://www.snmpview.de/snmpd_conf.php)

## ANEXOS

### 1. ACTUALIZACIÓN FIRMWARE DD-WRT

Lo primero que necesitamos es disponer de una computadora con puerto de red. Después, una conexión a Internet de banda ancha (ya sea DSL, Cable o similar). Si se cumplen estas condiciones, debes comprar un Router WRT54G o WRT54GS de Linksys o algun router de la lista the "Support Hardware" citada a continuación:

WRT54G Version 1.0/1.1, 2.0, 2.2, 3.0, 3.1, 4.0.

WRT54GL Version 1.0. y 1.1

WRT54GS Version 1.0, 1.1, 2.0, 2.1, 3.0, 4.0.

ASUS WL500G - Deluxe.

Motorola WR850G / WR850GP.

Siemens SE505.

Buffalo WBR-G54 (no probado 100%).

Estos router pueden ser comprados directamente desde Internet en

[CiudadWireless](http://www.ciudadwireless.com) (<http://www.ciudadwireless.com>), en [Risk](http://www.riskinformatica.com)

[Informática](http://www.riskinformatica.com) (<http://www.riskinformatica.com>) o en un sitio basado en subastas como

[ebay](http://www.ebay.es) (<http://www.ebay.es>) . Luego, descarga el firmware libre DD-WRT DESDE

<http://www.dd-wrt.com> y sigue las instrucciones en [Como flashear tu WRT54G/GS o instrucciones especificas para marca/modelo](#) para instalar el firmware en tu router.

[\[edit\]](#)

**Para flashear el Firmware debes de establecer una conexión por medio de un cable UTP.**

*Flashear vía wireless es inestable y puede dejar colgado o "[bricked](#)" tu router en caso de una caída del servicio eléctrico u otra eventualidad.*

*No necesitaras conexión a Internet para realizar los pasos del 2 al 5*

#### 1) Descargar el firmware

- Descargue la última versión o una versión beta [AQUI](http://dd-wrt.gruftie.com/dd-wrtv2/index.php?link=downloads) (<http://dd-wrt.gruftie.com/dd-wrtv2/index.php?link=downloads>) o vaya a la página de DD-WRT en <http://www.dd-wrt.com/>

- Descomprima el archivo

#### 2) Acceda a la Interfaz Web

*Nota: para los usuarios de un firmware que no es original de linksys : no uses tftp! cambia el firmware al oficial de linksys, y continúe entonces.*

- Con el PC conectado a uno de los 4 puertos LAN del router abre el navegador web y teclea la IP del router (por defecto es 192.168.1.1).

- Te pedirá usuario y contraseña (el usuario no es necesario, la contraseña por defecto es *admin*).

- Después de teclear la contraseña debes estar en la [Web Interface](#) de tu WRT54G/S *sino recuerdas la contraseña, realiza el paso 3.*

### 3) Restaurar los ajustes de fabrica

*Puedes mantener pulsado el botón reset en la pared trasera del router durante 30 segundos. Esto borrara la clave a admin.*

- Entra al interfase web y pulsar la pestaña de "Administration".
- Pulsa la pestaña inferior de "Factory Defaults".
- Selecciona "Yes".
- Pulsa el botón "Save Settings".
- En la nueva pagina, pulsar "continue".

### 4) Actualizar el Firmware

*No apagues el ordenador, tampoco cierres el navegador web, ni apagues el router durante el proceso*

- Pulsa la pestaña "Administration"
- Pulsa la pestaña inferior "Firmware Upgrade".
- Pulsa el botón "Browse" y selecciona el archivo DD-WRT .bin ya descomprimido en el paso 1.
- El router tardará varios minutos en subir el archivo, flashear el firmware, y reiniciarse.
- Se abrirá una nueva pagina, espera de 2-5 minutos antes de pulsar "Continue".
- Si se ha flasheado bien, te encontraras en la interface web DD-WRT y el nombre del Router será DD-WRT.

### 5) Cambiar el idioma

-Una vez actualizado el firmware pueden que aparezcan caracteres raros y no puedas leer algunas cosas, no preocuparse, es debido a que el idioma por defecto no es el adecuado, para cambiarlo en la pestaña de "Administration" (la penultima). 'Language Selection' Elige 'Spanish' Y posteriormente guarda los cambios pulsando el primer botón al final de la página 'Save changes'.

### 6) A disfrutar

-Es extremadamente aconsejable que active la opción "Boot Wait" en la pestaña de "Administration". Esto te ayudara a recuperar el router en un futuro en caso de algún problema con el flasheo.

### 7) Restaurar los ajustes de fabrica OTRA VEZ

- Repita el paso 3 de nuevo

### 8) ?? Uh-oh ??

- Si el router falla al reinicio (la luz de POWER no deja de flashear, no hay interfase web, etc) necesitas [Recover from a Bad Flash](#).

## 2. INSTALACIÓN JFFNMS

Windows 2000/XP Install Procedure  
=====

Note: This will not work for Windows 95/98/Me, it's only for Windows 2000/XP and maybe NT 4.0

The installation should be done as and Administrator User.

This guide is for Apache2 with PHP5. If you want to use older versions,  
read the JFFNMS 0.7.9 INSTALL.win32.txt file.

Installation:  
-----

Install Apache:

Download Apache2 from:  
<http://www.apache.org/dist/httpd/binaries/win32/>  
Get the latest release "no\_ssl.msi" package, like  
"apache\_2.0.52-win32-x86-no\_ssl.msi".  
Complete Install, Click on Next until Finish.  
You should see Apache running in your taskbar.  
Test it going browsing to <http://localhost/> or  
<http://IP.OF.YOUR.BOX/>

Install MySQL:

Download from: <http://www.mysql.com/downloads/mysql/4.1.html>  
Scroll down until you find MySQL Essentials for Windows.  
  
Select the "Complete Install", Click on Next until Finish.  
  
When you go into the configuration phase, select "Server Machine",  
and then "Non-Transaccional Database Only".  
  
Then set a new root password.  
  
All other configuration leave as default.

Install PHP:

Download PHP 5 from: <http://www.php.net/downloads.php>  
Get the "Windows Binaries ZIP package" with all the modules  
(First One).

Extract the archive to c:\php

Execute the following:

```
c:\> copy c:\php\php.ini-dist c:\windows\php.ini
c:\> copy c:\php\libmysql.dll c:\windows\
```

Create Folder c:\usr

Copy c:\php\extra\libs to c:\usr\libs

Install JFFNMS:

Download the .zip file (I'm sure you already have done this) and extract it to c:\jffnms

Create the User 'jffnms' in the Windows User Manager with any password you choose.

Remember to CHECK 'Password Never Expires' and UNCHECK 'Must Change password at next logon' for this user.

Give the 'jffnms' user FULL CONTROL over the c:\jffnms folder (and its sub-folders)

Install RRDTOOL:

Download from:

<http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/?M=D>

Get the rrdtool-1.0.49-win32-bin.zip (or newer) file.

Extract the archive to a temporary location.

Copy the rrdtool.exe file from inside the archive to c:\jffnms

Install NMAP: (If you are running Windows XP please read the NMAP page carefully, it may not work)

Download it from:

[http://www.insecure.org/nmap/nmap\\_download.html](http://www.insecure.org/nmap/nmap_download.html)

Get the nmap-3.81-win32.zip file (or newer)

Extract the nmap.exe file from the archive to c:\jffnms\

Install WinPCAP for NMAP:

Download it from: <http://winpcap.polito.it/>

Choose the Auto-Installer.

Complete Install, Click on Next until Finish.

Configuration:

-----

Configure Apache for PHP5:

Add this to the bottom of your c:\program files\apache group\Apache2\conf\httpd.conf file:

```
LoadModule php5_module c:/php/php5apache2.dll
AddType application/x-httpd-php .php

NameVirtualHost *

<VirtualHost *>
    ServerAdmin webmaster@your-domain.com
    DocumentRoot c:\jffnms\htdocs
    ServerName nms.yourdomain.com
    DirectoryIndex index.php
    LimitRequestLine 20000
</VirtualHost>
```

\* Make sure you modify the above information to match your needed settings

Configure PHP:

Make sure you have this on your php.ini file:

```
register_globals = On
register_argc_argv = On
allow_url_fopen = On

extension_dir = c:\php\ext

(Uncomment the following):
extension=php_gd2.dll
extension=php_snmp.dll
extension=php_mysql.dll
extension=php_sockets.dll
```

SMTP = your.mail.server.com (set it to your mail server IP or DNS name)

Now Restart the Apache Service.

if you don't change these, JFFNMS will not work.

Configure MySQL:

To create the JFFNMS DB run the following:

Check the path to mysql.exe

```
c:\> c:\Program Files\MySQL\MySQL Server 4.1\bin\mysql.exe -u
root -p<Root MySQL Password>
```

```
mysql>CREATE DATABASE jffnms;
mysql>GRANT ALL PRIVILEGES ON jffnms.* TO jffnms@localhost
IDENTIFIED BY 'jffnms';
mysql>FLUSH PRIVILEGES;
mysql>quit;
```

And:

```
c:\> c:\Program Files\MySQL\MySQL Server 4.1\bin\mysql.exe -
ujffnms -pjffnms jffnms < c:\jffnms\docs\jffnms-0.8.x.mysql
```

Replace x with the version you are installing.

Configure JFFNMS:

```
-----
I M P O R T A N T
-----
```

Go to: <http://yourserver/>

Verify and change (if needed) the options.

Don't mind the errors on diff, neato, smsclient, fping, ntpq, PgSQL and SOAP.

Save the configuration and change the values until everything needed is Green.

Everything else SHOULD be OK.

Scheduled Tasks:

```
-----
      If you did not install jffnms in c:\jffnms you will
have to change the
      contents of the .bat files (path).
```

Start/Settings/Control Panel/Scheduled Tasks/Add  
Scheduled Task.

- 1) Next
- 2) Click Browse
- 3) Browse to c:\jffnms\engine\windows and select the first .bat file.
- 4) Select Daily, then Next.
- 5) Change the Start time to 12:00 AM
- 6) Select Every 1 Days then Next.
- 7) Enter Username: jffnms and Password: (the one you choosed before), then Next.
- 8) Click the 'Open Avanced...' checkbox, then click Finish

- 9) Change "Start In" to c:\jffnms\engine
- 10) Click the Schedule tab
- 11) For autodiscovery\_interfaces, consolidate, poller & rrd\_analizer:  
Click on the Advanced Button.  
Check Repeat Task. In the Every Field enter the respective times:  
autodiscovery\_interfaces every 30 minutes.  
consolidate Every 1 Minute.  
poller Every 4 minutes.  
rrd\_analizer every 30 minutes.  
Duration: 24 Hours.
- 12) Click on OK.
- 13) Click on OK.

\*\* REPEAT THESE STEPS FOR EACH .BATCH FILE IN THE  
C:\JFFNMS\ENGINE\WINDOWS PATH!

Now... try to access <http://yourserver/> or  
<http://yourserver/jffnms/> (if you took this way)  
Remember to add the last slash / to the URL.  
The Default Username and password is "admin".

If you want to read more, please read the INSTALL file and skip  
the Unix Parts.

=====

If you need anything, contact us at: [jffnms-users@lists.sourceforge.net](mailto:jffnms-users@lists.sourceforge.net)

Thank You for trying 'Just For Fun' Network Management System

Don't forget to Donate to the project if you find it useful for  
your work.

=====

Javier Szyszlican aka SzYsZ  
Buenos Aires, Argentina  
[javier@jffnms.org](mailto:javier@jffnms.org) (please try the mailing list first)  
ICQ #397319