Construir un firmware OpenWRT backfire para TP-Link WR740n V4

19-04-2012

- Miguel Pérez
- Pablo Boronat

https://dev.openwrt.org/ticket/10492

En esta receta es para compilar OpenWRT para un TP-Link WR740n (Versión 4 pero sirve para las otras versiones de este modelo).

OpenWRT es una versión del sistema GNU/Linux especializada en dispositivos de red. Comunmente se ha usado en routers-wifi de bajo coste.

En la receta se incluye una versión del firmware ya compilada pero se ponen las intrucciones por si alguien quiere hacerse una versión con otros paquetes. Si se añaden otros paquetes hay que seleccionarlo bien porque a poco que pongamos nos saldrá una imagen que no cabrá en la memoria flash (4Megabytes en este modelo) que es donde se guarda el sistema operativo.

En la versión que dejamos compilada hemos incluido el cliente vpnc que es el que actualmente se usa para conectar con la red de la Universidad Jaume I (UJI). La idea es que con un router de este tipo (muy económico, alrededor de 20€) el túnel se abre desde el TP-Link y luego lo que conectemos con él ya tiene acceso directo a la red de la UJI y salida a internet pasando por la UJI (para la comunidad universitaria); por ejemplo las tablillas con Android no tienen un cliente VPN que funcione con el servidor de túneles de la UJI. En la receta se propone una configuración del router para hacer esto una vez ya se ha instalado OpenWRT.

Hemos añadido en el inicio de esta receta una guía rápida para pasar de un TP-Link WR740n con el software original a tenerlo con openwrt y configuración vpn uji en 5 pasos.

Guía rápida

Guía rápida de configuración. Si necesitas más información sobre cada paso la encontrarás más adelante en la receta.

Firmware para el tp-link TL-WR841N V8.*. (Hay que hacer unzip)

Configuración completa para el WR841 V8.*

Firmware para el tp-link TL-WR841N V9.*. (Hay que hacer unzip; está por comprobar)

1. Cambiar el firmware
    - Abrir interfaz web del router http://192.168.0.1 (admin/admin)
    - Ir a TP-Link —> System —> upgrade
    - Cargar el Firmware ya compilado (hay que hacer gunzip).
2. Activar root openwrt cambiando su password

- Abrir interfaz web del router http://192.168.1.1
- login → go to password configuration
- poner clave guifiadmin

3. Cargar la configuración de vpn
   - Abrir interfaz web del router http://192.168.1.1 (root/guifiadmin)
   - Ir a System→ Backup / Flash Firmware → Restore Backup
   - Cargar una de las siguientes configuraciones:
     - configuración completa
       - ip 192.168.1.10 en la red al exterior e ip 192.168.2.1 en la red interna
       - usuario del router login **root** y password **guifiadmin**
       - muy recomendable cambiar la contraseña: System → Administration
       - wifi configurada con SSID **guifiHome** y con clave WPA2 **guifiadmin**
       - muy recomendable cambiar la contraseña: Network → Wifi → Edit → Wireless Security
       - pre-configuración de vpn en /etc/rc.local
       - reinicio de vpn automático
     - configuración básica
       - ip 192.168.1.1 en la red al exterior e ip 192.168.2.1 en la red interna
       - pre-configuración de vpn en /etc/rc.local
       - reinicio de vpn automático
     - En las dos configuraciones el router espera tener la antena en la 192.168.1.20. Si has utilizado el unsolclic tu antena será la 192.168.1.1, en ese caso deberás modificar la ruta por defecto en Network→Interfaces→WAN→Ipv4 Gateway, y también para la ruta estática en Network→Static routes→Ipv4 Gateway.
4. Configurar vpn
   - editar tu usuario y contraseña en la configuración de vpn de System→startup (/etc/rc.local)
   - activar el arranque de vpnc en System → startup → vpnc Disabled/Enabled (/etc/init.d/vpnc enable)
5. Reinicio y comprobación
   - reiniciar el router para que se active la configuración vpn en System → Reboot → Perform reboot
   - si todo funciona se encenderá el candado y podrás navegar a través del vpn
   - si no funciona no habrá luz en el candado y tendrás que revisar los pasos en el resto de esta guía.

Construir el firmware

Fix Me!: Deprecated

```
mkdir OPENWRT-TPLINK
cd OPENWRT-TPLINK
svn co svn://svn.openwrt.org/openwrt/trunk/
```

https://dev.openwrt.org/ticket/10492

```
cd trunk
./scripts/feeds update -a
./scripts/feeds install -a
```

Luego seleccionamos los paquetes:

```
make menuconfig
  Target System Atheros AR7xxx/AR9xxx
  Target Profile TP-LINK TL-WR740N/ND
  Luci ---> Collections ---> luci-ssl
  Network ---> VPN ---> vpnc
```

Luego lanzamos la compilación, que le cuesta un buen rato.

```
make V=99
```

Luego, el firmware es .../OPENWRT-TPLINK/trunk/bin/ar71xxx/openwrt-ar71xx-generic-tl-wr740n-v4-squashfs-factory.bin

...

Versión ya compilada para descargar:

Firmware ya compilado (hay que hacer gunzip). (ATTITUDE ADJUSTMENT (Bleeding Edge, r31342))

Cargar el firmware

Para cargarlo se puede hacer desde el sistema de TP-Link —> System —> upgrade

El TP-Link de fábrica viene con la 192.168.0.1 en el switch y la cuenta admin con password admin.

Se carga el firmware y se reinicia el router.

Ahora la IP es 192.168.1.1/24. Hay que entrar por telnet con el login root sin password. Se pone el password de root (orden passwd root) y salimos con exit. Entonces se desactiva el telnet y se activa ssh.

Luci debe funcionar en https://192.168.1.1 (Luci es el entorno gráfico de OpenWRT).

Ponerse en le ordenador La ip 192.168.0.10/24 (bueno, una IP que permita conectar con el TP-Link a través de una de las bocas amarillas del switch ethernet).

Hacer:

```
atftp -p -l openwrt-ar71xx-generic-tl-wr740n-v4-squashfs-factory.bin --verbose --trace 192.168.0.1
```

o con tftp:

```
tftp 192.168.0.1
           tftp> bin
           tftp> tra
           tftp> put openwrt-ar71xx-generic-tl-wr740n-v4-squashfs-factory.bin
```

- En ese momento, apagar y encender el TP-Link, en pocos segundos se iniciará la transferencia del fichero y el TP-Link se reiniciará tras unos minutos.

- Ponerse en el ordenador una ip en el rango 192.168.1.0/24 y ya debemos poder conectar con él como se ha dicho anteriormente.

Configurar vpnc

La configuración para que funcione abriendo un túnel con la UJI.

En /etc/vpnc/default.conf:

```
IPSec gateway vpn-server.uji.es
IPSec ID UJI
IPSec secret 12345678
#IKE Authmode hybrid
Xauth username AQUI_TU_USUARIO
Xauth password AQUI_TU_CONTRASEÑA
```

Ya podemos probar el túnel ejecutando vpnc en el terminal (vpnc-disconnect para parar el túnel). Para poder probar el túnel debemos estar conectados a guifi.net o a la freenet o debemos poder hacer ping a una de las IPs del vpn-server.uji.es .

Ahora que ya funciona el túnel debemos hacer que lo inicie en el arranque y que periódicamente pruebe si está funcionando por si hay que volverlo a levantar.

- En el interfaz web (en el navegador web poner la IP 192.168.1.1) System —> Startup
- En la parte de bajo hay cuadro (**Local Startup**) donde podemos escribir instrucciones para el arranque. Podemos dejarlo de la siguiente forma:

```
# Put your custom commands here that should be executed once
# the system init finished. By default this file does nothing.

cat > /etc/vpnc/default.conf <<EOF
IPSec gateway vpn-server.uji.es
IPSec ID UJI
IPSec secret 12345678
Xauth username AQUI_TU_USUARIO
Xauth password AQUI_TU_CONTRASEÑA
EOF

exit 0
```

- Ahora hay que reiniciar el bicho (System —> Reboot —> Perform reboot)

Para los pasos siguientes nos hemos basado en:
https://forum.openwrt.org/viewtopic.php?id=31853 y
http://wiki.openwrt.org/oldwiki/vpn.client.vpnc

Crear el fichero /etc/init.d/vpnc :

```
#!/bin/sh /etc/rc.common
START=75
STOP=10

start() {
        mkdir -p -m777 /var/run/vpnc
        vpnc --non-inter /etc/vpnc/default.conf
}

stop() {
        PID_F=/var/run/vpnc/pid
        if [ -f $PID_F ]; then
           PID=$(cat $PID_F)
           kill $PID
           while [ -d /proc/$PID ];
           do
                sleep 1
           done
        fi
}
```

—Por si el túnel se cuelga, conviene tener un script que prueba ping a dos IPs a las que debería llegar cuando el túnel funciona. /usr/local/vpn-keepalive :

```
#!/bin/sh
#

#no hacer nada si el vpn está deshabilitado, para evitar bloqueo
de password por reintentos
[ ! -f /etc/rc.d/S75vpnc ] && exit 0


# Restart VPNC if both of the specified hosts on the command line
are unavailable
if ! [ $(ping -q -c 1 ${1} 2>&1 | grep "1 packets received" | sed
"s/.*\(1\) packets received.*/\1/") ] ||
   ! [ $(ping -q -c 1 ${2} 2>&1 | grep "1 packets received" | sed
"s/.*\(1\) packets received.*/\1/") ]; then
    echo Not alive $1 or $2, restarting VPNC
    /etc/init.d/vpnc stop
    /etc/init.d/network restart
    sleep 5
    /etc/init.d/vpnc start
else
echo Alive $1 or $2
fi
```

Una alternativa que realiza hasta 3 pings (o n) a las IPs dadas y comprueba si se han perdido todos:

```
#!/bin/sh
#

#no hacer nada si el vpn est.. deshabilitado, para evitar bloqueo
```

```
de password por reinten
 [ ! -f /etc/rc.d/S75vpnc ] && exit 0


 # Restart VPNC if both of the specified hosts on the command line
are unavailable
if [ $(ping -q -c 3 ${1} 2>&1 | grep "100% packet loss" | sed
"s/.*\(100%\) packet loss.*
   [ $(ping -q -c 3 ${2} 2>&1 | grep "100% packet loss" | sed
"s/.*\(100%\) packet loss.*

        echo Not alive $1 or $2, restarting VPNC
        /etc/init.d/vpnc stop
#        /etc/init.d/network restart
        sleep 5
        /etc/init.d/vpnc start
else
        echo Alive $1 or $2
fi
```

El código anterior se pone crontab para ejecutarse cada 2 minutos. Si falla un ping se volverá a ejecutar el vpnc para abrir el túnel. Deben ponerse dos IPs a las que se llega sólo si el túnel está abierto. Por ejemplo 8.8.8.8. Para ponerlo en crontab la siguiente orden abre el editor para incluirlo en crontab (los asteriscos son minutos horas días semanas meses): crontab -e

```
*/2 * * * *  /usr/local/vpn-keepalive 192.168.0.1 192.168.0.10 &
```

Otra versión de vpn-keepalive, en la que se comprueba si existe la interfaz del túnel (tun0) en vez de usar ping:

```
#!/bin/sh
#
# Restart VPNC if both of the specified hosts on the command line
are unavailab

# Do nothing if vpn is not set.
[ ! -f /etc/rc.d/S75vpnc ] && exit 0   # vpn deshabilitado

# Do nothing if vpn-keepalive is already running.
[  -f /var/run/vpn-keepalive ] && exit 0  # vpn-keepalive
ejecutandose

# We are running vpn-keepalive.
touch /var/run/vpn-keepalive


#if ! [ $(ping -q -c 1 ${1} 2>&1 | grep "1 packets received" | sed
"s/.*\(1\) p
#   ! [ $(ping -q -c 1 ${2} 2>&1 | grep "1 packets received" | sed
"s/.*\(1\) p
#       echo Not alive $1 or $2, restarting VPNC
if ! [ $(route -n | grep "tun0" | awk '/tun0/ {++x} END {print
x}') ]; then
```

```
    echo Tunnel not alive, restarting VPNC
    /etc/init.d/vpnc stop
    /etc/init.d/network restart
        sleep 5

    /etc/init.d/vpnc start

else

    echo vpn seems alive

fi


# vpn-keepalive is not running anymore.
rm /var/run/vpn-keepalive
```

Cargar configuración

Puedes cargar esta configuración para saltarte algunos de los siguientes pasos.

La carga de la configuración se realiza desde System→ Backup / Flash Firmware → Restore Backup.

Esta configuración deja preparado:

- cron cada 5 minutos de vpn-keepalive a www.google.com y www.google.es
- red 192.168.2.0/24 para red interna (el tp-link será 192.168.2.1 en las bocas amarillas y la wifi)
- red 192.168.1.0/24 para red al exterior (boca WAN, azul) (ruta por defecto a la nanostation 192.168.1.20) asigna al TP-Link la 192.168.1.1/24. Esto puede dar problemas si la nanostation se configura con el unsolclic. En ese caso la nanostation tendrá la 192.168.1.1 en la ethernet y lo mejor sería poner que el TP-Link configure la WAN por DHCP.
- servidor de nombres, zona horaria y servidor de tiempos configurados
- ruta para llegar a guifi.net (10.0.0.0/8)
- pre-configuración de vpn en /etc/rc.local
- configuración de leds, se usa el candado para indicar vpn activo
- cambio en las opciones de backup para incluir todo lo anterior

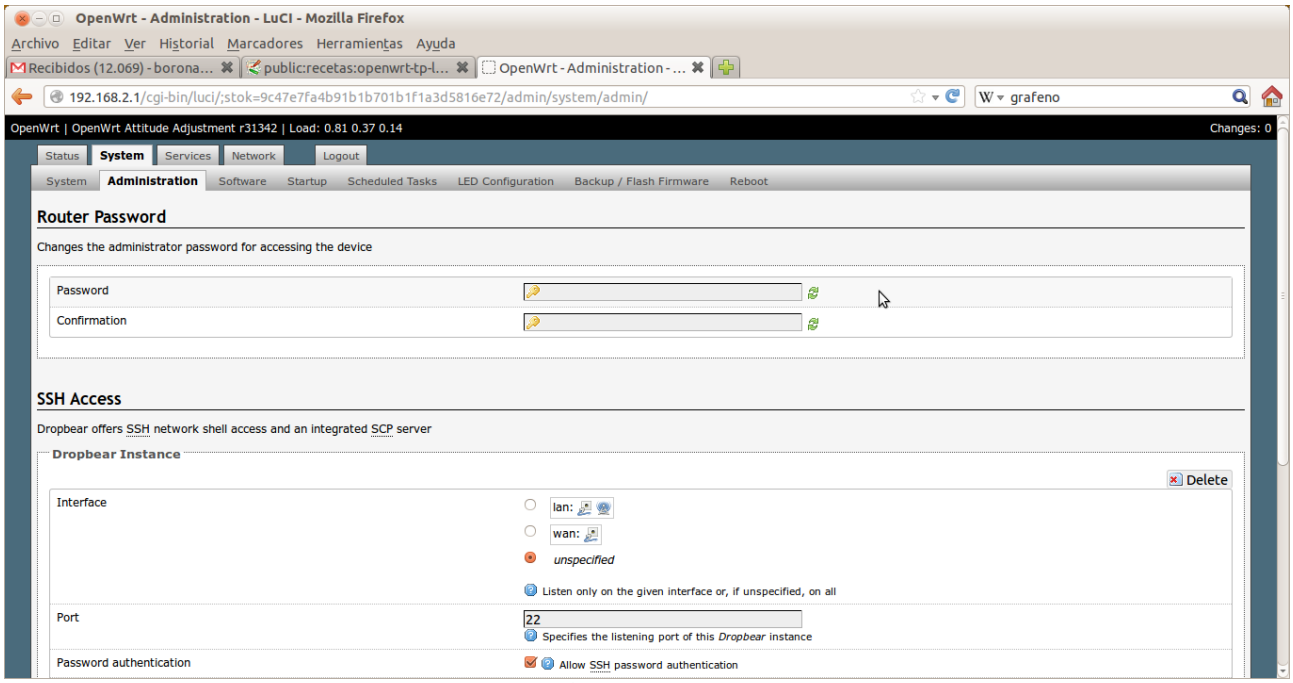
Así que sólo queda:

- editar tu usuario y contraseña en la configuración de vpn de System→startup (/etc/rc.local)
- activar el arranque de vpnc en System → startup → vpnc Disabled/Enabled (/etc/init.d/vpnc enable)
- configurar la wifi. Con esta configuración estará deshabilitada. Configurarla en Network → Wifi

Versión revisada del fichero de configuración

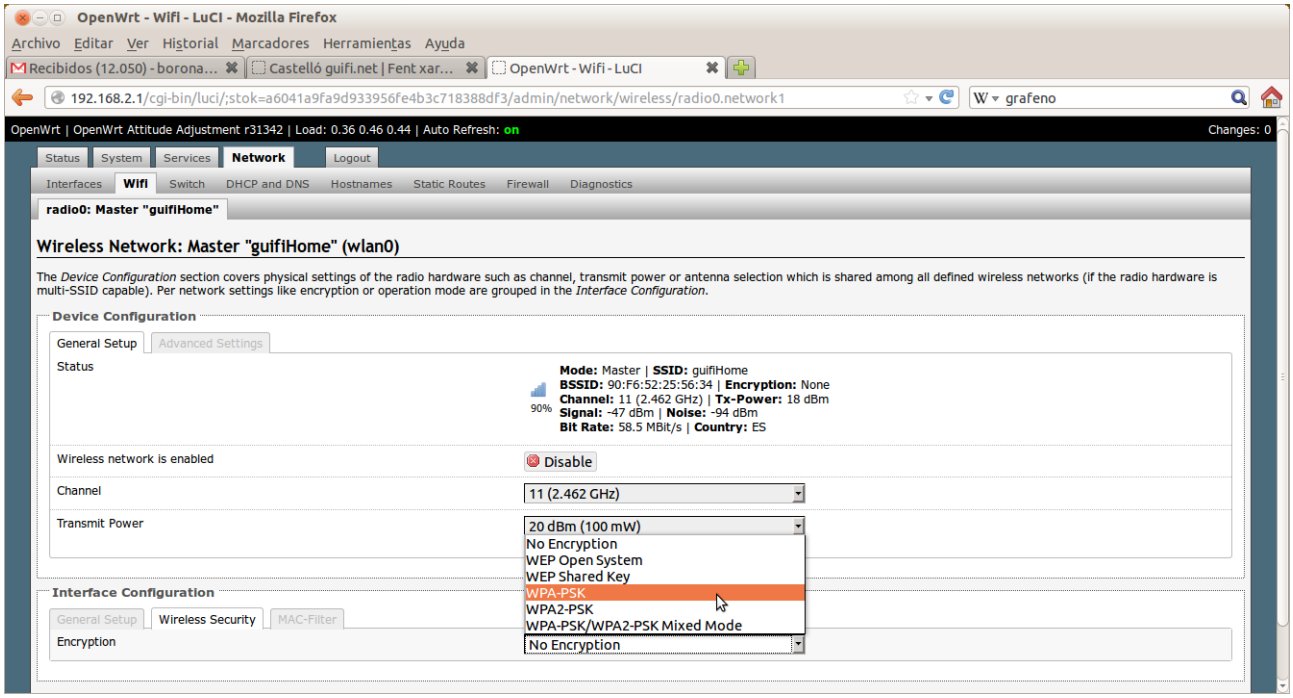
Aquí está la En esta versión parece que se resuelve la configuración de la wifi, la nueva versión error wifi corregido. Faltan detalles por pulir, pero es funcional. Incluye los cambios siguientes:

- Para configurar el router vía web o por terminal con ssh (192.168.2.1), la cuenta es login: **root** y password **guifiadmin** . Esta es la cuenta para configurar el router TP-link. Para cambiar la contraseña ir a System → Administration. Poner repetir dos veces la contraseña y en la parte de bajo darle a *Safe & Apply* (ver captura):



Cambiar la contraseña de la cuenta root del router.

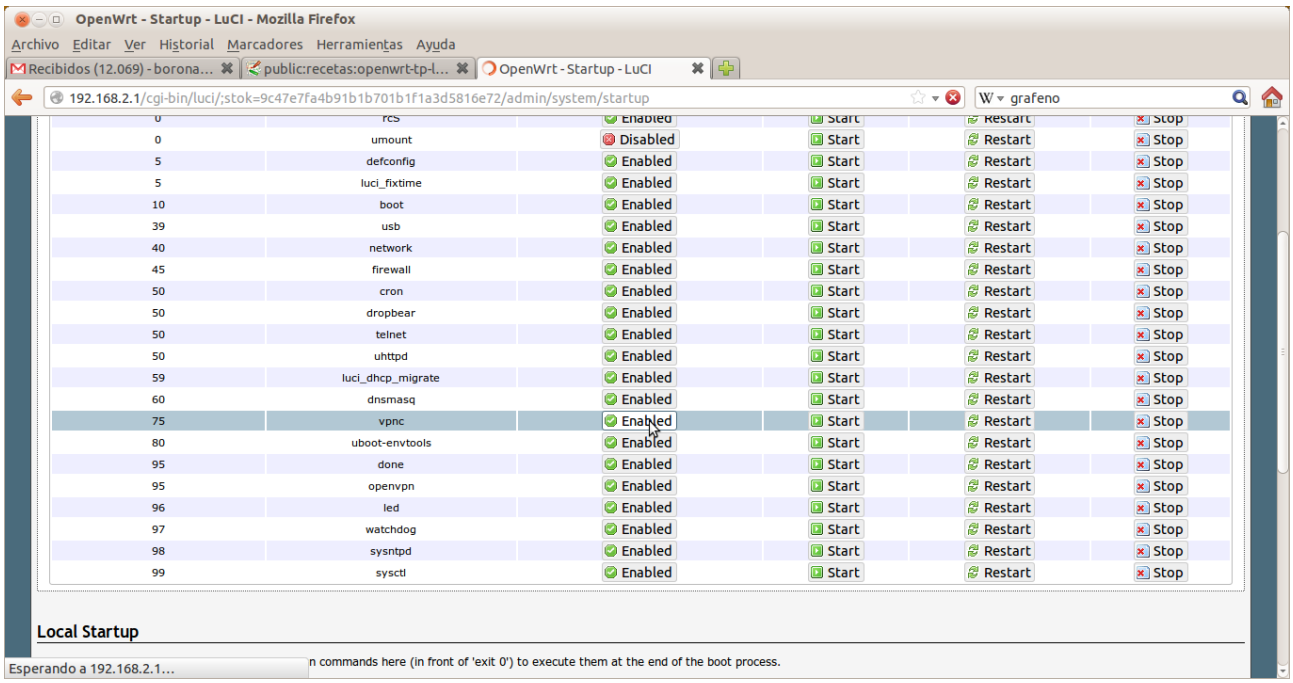
- Usa la versión de vpn-keepalive que comprueba si existe tun0. Además se reinicia la configuración de red porque en la versión anterior el túnel no se reiniciaba bien.
- La comprobación con vpn-keepalive se ejecuta cada 2 minutos.
- Se deja la wifi configurada con SSID guifiHome y con una clave WPA2 (clave guifiadmin). En la página web del router (192.168.2.1) Pestaña Network → Pestaña Wifi, editar. En *Interface Configuration* se puede activar o cambiar seguridad inalámbrica (ver captura). Al menos hay que cambiar la clave:



Poner clave WPA2 para el acceso wifi.

- IP 192.168.1.10/24 para red al exterior (boca WAN, azul). Ruta por defecto al router del nodo cliente. Si es una ubiquiti, por defecto debe ser 192.168.1.20. OJO, si se ha configurado con el unsoloclic, la ruta por defecto debería ser 192.168.1.1. Para modificarlo: Network → Interfaces, editar WAN y cambiar el *IPv4 gateway* y también para la ruta estática en Network → Static routes *Ipv4 Gateway*.
- Hay corregidos errores en la configuración del DNS para poder conectar a nombres dentro de la propia guifi.net.

→Local Startup→initscriptsDisabledEnabled



Activar vpnc en el arranque.

Configuración de la red y otros detalles

Ya por último faltan los detalles de configuración del router. Lo normal sería:

- Asignar (por ejemplo) la 192.168.10.0/24 a un pruente entre la wifi y el switch LAN (y quitar la 192.168.1.1/24 porque la usaremos entre el nodo guifi y el TP-Link
- Al puente anterior configurarle un DHCP para dar configuración a los cacharros que conectemos.
- Configurar nuestro punto de acceso wifi.
- La WAN (boca azul) conectarla con la LAN del POE (normalmente de una nanostation) y pedir configuración por DHCP.
- Cuando se abra el túnel VPN nos asignará la puerta por defecto a través del túnel. Si también queremos tener acceso a guifi.net, añadir una ruta a la 10.0.0.0/8 a través de la IP del nodo guifi (la nanostation) en la red privada (la 192.168.1.1 si hemos pasado el unsolclic o la 192.168.1.20 que es la que viene de Ubiquiti de fábrica).
- Configurar el cortafuegos.

Failsafe mode en Openwrt

Volver a los valores por defecto de la configuración

Si perdemos el acceso al router (olvidamos el password, ponemos mal alguna regla del firewall, etc), se puede volver a dejar el openwrt en su estado inicial, para ello hay que seguir estos pasos (de http://www.gargoyle-router.com/phpbb/viewtopic.php?f=8&t=2175):

- Desconectar el cable de red del router
- Conectar el PC con alguno de los puertos LAN del router.
- Configurar el PC con la IP estática 192.168.1.2
- PConectar el cable del router
- Esperar a que el LED “SYS” empiece a parpadear (tiene forma de estrella en el TL-W740N).
- Pulsar el botón “QSS” (en la parte posterior del router en el TL-W740N) - el LED “SYS” empezará a parpadear con una mayor frecuencia.
- Conectarse al router utilizando `telnet` a la dirección 192.168.1.1 - Se accederá a un shell de root del router sin necesidad de autenticarse
- Con la orden `firstboot` toda la configuración volverá a su estado inicial (de fábrica).

Actualizar el firmware

Si lo que se quiere es actualizar el firmware, hay que ejecutar los pasos anteriores, salvo el último (`firstboot`) y una vez dentro del router transferir el firmware al router y actualizarlo.

Utilizando Netcat si se dispone de suficiente memoria RAM en el router se puede transferir y actualizar a la vez.

En las instrucciones de openwrt dice textualmente **This method is** NOT **recommended!** aunque lo hemos probado repetidas veces en los TP-Link WDR3600.

En el ordenador ejecutar

```
nc -q0 192.168.1.1 1234 < openwrt-ar71xx-tl-wr1043nd-v1-squashfs-sysupgrade.bin
```

En el router

```
nc -l -p 1234 | mtd write - firmware
```

Si no se dispone de suficiente memoria RAM, el siguiente método es más seguro. Primero se transfiere el firmware.

En el ordenador ejecutar

```
cat [specified firmware].bin | pv -b | nc -l 3333
```

En el router

```
nc 192.168.1.111 3333 > /tmp/[specified firmware].bin
```

Donde 3333 es el puerto que se ha elegido (puede ser cualquier otro) y 192.168.1.111 es la IP que se le ha puesto al PC. `[specified firmware].bin` es el nombre del fichero que contiene el firmware que se quiere transferiri. La orden 'pv -b' es opcional y sirve para visualizar el avance del proceso.

Ahora se puede actualizar el firmware con `sysupgrade` o `mtd`:

```
  sysupgrade -v /tmp/[specified firmware].bin
```

o

```
  mtd -r write /tmp/[specified firmware].bin firmware
```

También se puede transferir el firmware `wget`, en el ordenador hay que poner el firmware en el directorio de apache (o del servidor web que se esté utilizando) y en el router ejecutar

```
wget http://192.168.1.111/[specified firmware].bin
```

Enlaces

En el wiki de openwrt está explicado el modo failsafe: http://wiki.openwrt.org/doc/howto/generic.failsafe y como actualizar el firmware http://wiki.openwrt.org/doc/howto/generic.sysupgrade