

ANEXO

Configuración de red en Linux

ÍNDICE

ANEXO: Configuración de red en Linux	4
A.1. Configuración de las tarjetas de red y Tabla de encaminamiento	5
A.1.1. Configuración básica	5
A.1.1.1. Visualización de la configuración actual	5
A.1.1.2. Modificación manual de la dirección IP de las Tarjetas de Red	6
A.1.1.3. Modificación Manual de la Tabla de Encaminamiento principal	11
A.1.1.4. Configuración persistente de las Tarjetas de Red: Ficheros de Configuración	13
A.1.1.5. Configuración persistente de la Tabla de Encaminamiento principal: Entradas Manuales	15
A.1.1.6. Des/Activación de las interfaces de red	17
A.1.1.7. Tablas de Encaminamiento Linux	19
A.1.1.8. Consulta/Modificación de parámetros de funcionamiento de las interfaces de red	20
A.1.1. Configuración de interfaces Wi-Fi	25
A.1.2. Asociación de varias IPs a una misma interfaz	30
A.1.3. Agrupación de interfaces de red (bonding)	33
A.1.4. Interfaces Ethernet virtuales sobre interfaces físicas	33
A.2. Tabla/Caché ARP	35

A.3. Resolución de nombres	36
A.4. Servicio DHCP	37
A.4.1. Servidor ISC DHCP	37
A.4.2. Cliente DHCP	37
A.4.3. Agente de Reenvío (Agent Relay) DHCP	38
A.5. Análisis de sockets	38

@ 2020

ANEXO: Configuración de red en Linux

En las distribuciones Linux actuales, los comandos clásicos de red "arp, ifconfig, route, netstat" (entre otros) se encuentran en estado obsoleto (desarrollo abandonado, funcionalidad no garantizada y eliminación próximamente), por lo que se desaconseja su uso. En su lugar, se recomienda emplear los comandos "ip" (Internet Protocol) y "ss" (Socket Statistics):

Paquete	Comando clásico (obsoleto)	Funcionalidad	Sustituto	Paquete
net-tools	arp	Gestionar caché ARP	ip neighbour	iproute2
	ifconfig netstat -i	Configurar tarjetas de red cableadas	ip link, ip address	
	iptunnel	Crear túneles IP	ip tunnel	
	ipmaddr netstat -g	Gestionar direcciones multi-cast	ip maddr	
	route netstat -r	Gestionar Tabla de encaminamiento principal (FIB)	ip route	
	netstat	Gestionar sockets locales	ss	
	nameif	Renombrar tarjetas de red	ip link ifrename	ifrename
	mii-tool	Controlar hardware y drivers de red	ethtool	ethtool
wireless-tools	iwconfig iwlist	Configurar tarjetas de red inalámbricas	iw	iw

INSTALACIÓN: Aunque suelen preinstalarse, si su sistema Linux no tuviese los comandos "ip" y "ss", puede instalarlos con "yum install iproute" (distribuciones de la rama Fedora) o con "apt-get install iproute2" (distribuciones de la rama Debian).

A.1. Configuración de las tarjetas de red y Tabla de encaminamiento

A.1.1. Configuración básica

A.1.1.1. Visualización de la configuración actual

Configuración actual de las tarjetas de red	Activas	<pre>ip address show up ip link show up ifconfig</pre>
	Todas	<pre>ip address [list show] ip addr [ls sh] ip a [l s] ip link [l s] ip l [l s] ifconfig -a cat /proc/net/dev</pre>
	Tarjeta concreta	<pre>ip address list [dev] ethX ip address show ethX ip link list ethX ip link show ethX ifconfig ethX</pre>
Des/Activación de las tarjetas de red	Activación (no configuran)	<pre>ip link set up eth0 ifconfig ethX up</pre>
	Desactivación (no desconfiguran)	<pre>ip link set down eth0 ifconfig ethX down</pre>
Tabla de encaminamiento principal actual	Con resolución inversa a nombres	<pre>ip -r route [ls sh] ip -r ro [ls sh] ip -r r [ls sh] route</pre>
	Sin resolución inversa (numérico)	<pre>ip route [ls sh] route -n cat /proc/net/route</pre>

NOTA: El comando “ip” (paquete iproute2) usa la sintaxis:

```
ip [opciones] objeto comando [parametros]
```

Tanto para el “objeto” (neighbour, link, address, route, ...) como para el “comando” (“list”, “show”, ...) se permite escribir la palabra completa o cualquier sub-prefijo de la misma. Por ejemplo, en lugar de “link” puede escribirse, indistintamente, “link”, “lin”, “li”, “l”. Normalmente no hay duplicidad, aunque en algunos casos sí; por ejemplo, para “ip link show” e “ip link set”, si se escribe “ip link s” habría ambigüedad, interpretándose como “show”. En algunos casos también existen alias abreviados, como por ejemplo “ip link list” e “ip link ls”. Asimismo, para cada “objeto” se define un “comando” por omisión; por ejemplo, son equivalentes “ip address list” e “ip address”.

Puede consultar las opciones del comando “ip” usando:

```
man ip
man ip-objeto
ip objeto help
```

```
https://wiki.linuxfoundation.org/networking/iproute2
https://www.policyrouting.org/iproute2-toc.html
https://baturin.org/docs/iproute2/
```

Si desea usar el comando “ip” con autocompletado en todos sus parámetros, puede instalar el paquete “bash-completion” (se autocarga al arrancar el sistema con el fichero “/etc/profile.d/bash-completion.sh”).

A.1.1.2. Modificación manual de la dirección IP de las Tarjetas de Red

- Configuración estática:

```
ip addr add      10.1.0.1/24 dev eth1 [brd +]
ifconfig eth1 10.1.0.1 netmask 255.255.255.0
```

De forma predeterminada (o con “brd +”), se configura automáticamente la dirección de difusión de la tarjeta calculada a partir de la IP y máscara indicados. Opcionalmente, puede indicarse una dirección de difusión distinta:

```
ip addr add      10.1.0.1/24 dev eth1 brd      IP_difusion
ifconfig eth1 10.1.0.1/24 broadcast IP_difusion
```

NOTA: los comandos:

```
ip link set up eth0  
ifconfig ethX up
```

sólo activan la interfaz (cambian a “up” su flag de estado), pero NO configuran ningún parámetro de la interfaz. De modo similar, los comandos:

```
ip link set down eth0  
ifconfig ethX down
```

desactivan la interfaz (cambian su flag de estado a “down”, haciendo que deje de recibir/enviar mensajes), pero no la desconfiguran (no eliminan la IP de la interfaz).

ADVERTENCIA: realmente, ambos comandos No son 100% equivalentes. El comando:

- “ip”: va asociando múltiples direcciones a la tarjeta de red (si ejecuta varios “ip a add...”, podrá ver con “ip a” las múltiples direcciones asignadas (para eliminarlas, debe usarse “ip a del...”). Asimismo, este comando sólo modifica la dirección IP y máscara de la interfaz, pero no su estado “up/down”.
- “ifconfig”: sólo opera sobre (modifica y muestra) una de las direcciones asignadas a la interfaz (pero realmente la interfaz usa todas las direcciones que muestra “ip a”; por ejemplo, puede comprobar como un “ping IP” a cualquiera de esas direcciones se envía a través de la interfaz “lo”). Además, a diferencia del equivalente “ip”, con este comando “ifconfig” automáticamente se cambia el estado de la interfaz a “up”.

El comando “ifup ethX” usa internamente el comando “ip addr add IP/nn dev ethX”. Consecuentemente, si se realizasen las operaciones:

```
# Se cambia IP en ifcfg-ethX a: IP1
ifup ethX
# Se cambia IP en ifcfg-ethX a: IP2
ifup ethX
# Se cambia IP en ifcfg-ethX a: IP3
ifup ethX
# Se cambia IP en ifcfg-ethX a: IP4
ifup ethX
```

al ejecutar:

- “ifconfig”: se muestra solamente “IP1”.
- “ip a”: se muestran las 4 IPs con las que realmente trabaja la interfaz ethX (IP1, IP2, IP3, IP4).

Esto es un ejemplo de alguno de los motivos por los que se recomienda abandonar el uso del comando clásico “ifconfig”.

- Configuración DHCP (forzar la solicitud de una nueva configuración):

```
dhclient -r eth0; ip link set down eth0; dhclient eth0
```


ACLARACIÓN: “dhclient -r” provoca el envío al servidor DHCP de un mensaje DHCP Release, para liberar la licencia (lease) de la IP actualmente asignada (las licencias DHCP, y su estado, se encuentran en la carpeta /var/lib/dhcp/, tanto las recibidas por los clientes “dhclient.leases” como las asignadas por el servidor “dhcpd.leases”).

Por defecto, al desactivar una interfaz “ifdown ethX”, Linux no libera la licencia (no envía ningún “DHCP Release”), debiendo liberarla el servidor cuando expire su tiempo. Si se desea que el cliente solicite la liberación debe indicarse manualmente con este comando “dhclient -r”.

El comando “dhclient eth0” activa (up) la interfaz si estaba desactivada. Si la interfaz está activada y se desea que vuelva a solicitar configuración por DHCP, puede hacerse manualmente del siguiente modo:

- Desactivar la interfaz:

```
dhclient -r; ip link set down eth0; pkill -9 dhclient
```

- Reactivarla:

```
dhclient eth0
```

AYUDA: Al ejecutar el comando “ifup eth0”, si:

- El fichero “ifcfg-eth0” contiene el parámetro “BOOTPROTO=dhcp”, y
- Existe el fichero “/var/run/dhclient-eth0.pid” y contiene el PID de un proceso existente (sea el proceso “dhclient” o no)

entonces “ifup eth0” muestra el mensaje:

```
Determining IP information for eth0...dhclient(6394) is  
already running - exiting.
```

no realizando ninguna configuración sobre la interfaz (en otro caso, aunque la interfaz ya se encuentre UP y tenga IPs asignadas, sí aplica configuración sobre la misma).

Si al ejecutar los comandos:

```
ifdown ethX; ifup ethX
```

obtiene el mensaje:

```
Determining IP information for eth0...dhclient(6394) is  
already running - exiting.
```

Ello se debe a que la interfaz “ethX” está configurada para usar DHCP, y el comando “ifdown” no ha podido cerrar (intenta un cierre suave) el proceso “dhclient” (el comando “ifup”, al detectar que el proceso está activo, termina directamente su ejecución con ese mensaje). Para solucionarlo, basta forzar el cierre del proceso:

```
pkill -9 dhclient
```

y luego volver a ejecutar los comandos anteriores.

NOTA: El comando “ip” admite operaciones adicionales sobre la configuración de las interfaces. Por ejemplo:

- Eliminar una dirección IP de la interfaz (sin desactivarla):

```
ip addr del IP_ethX/mascara dev ethX
ip addr flush to IP_ethX
```

Para borrarlas todas: `ip address flush dev ethX`

- Modificar el nombre de una interfaz (para usar “ifup”, debe ajustar el nuevo nombre en los ficheros de configuración de las tarjetas de red):

```
ip link set down eth0
ip link set ethX name nombre
ip link set up nombre
```

A.1.1.3. Modificación Manual de la Tabla de Encaminamiento principal

Para añadir entradas manuales (todos admiten “metric valor”):

Entrada Directa			
ip route add	10.1.0.0/24		dev eth1
route add -net	10.1.0.0/24		dev eth1
route add -net	10.1.0.0 netmask 255.255.255.0		dev eth1

Indirecta (de pasarela) por defecto o genérica: para cualquier IP ("default" implica IP_subred="0.0.0.0" y máscara="0.0.0.0", "0.0.0.0/0", esto es, cualquier IP)			
ip route add default via	10.1.0.254		dev eth1
route add default gw	10.1.0.254		dev eth1

Entrada Indirecta (de pasarela) restringida a una subred (rango de IPs)			
ip route add	10.1.0.0/24	via 10.1.0.254	dev eth1
route add -net	10.1.0.0/24	gw 10.1.0.254	dev eth1

ADVIERTA:

- Sólo es posible añadir entradas en la tabla de encaminamiento asociadas a la interfaz “ethX”, si dicha interfaz se encuentra activada (UP). En caso contrario se mostrará el mensaje “RTNETLINK answers: Network is down”.
- Al eliminar la IP asignada a una interfaz (“ip a del IP dev ethX”), se autoeliminan todas las entradas de la tabla de encaminamiento asociadas a dicha interfaz para esa IP.
- A pesar de lo anterior, es posible añadir manualmente entradas directas (no indirectas) en la tabla de encaminamiento usando direcciones NO asignadas a una interfaz (activa). Ejemplo:

```
ip l set up eth1
ip a del 10.0.0.1/8          dev eth1
ip r add 10.0.0.0/8         dev eth1
```

- Conforme a lo anterior, sólo se pueden introducir entradas indirectas cuya pasarela tenga una IP recogida en alguna de las entradas directas (alcanzable a L2). De otro modo, se mostrará el mensaje “RTNETLINK answers: Network is unreachable”.

Para eliminar las entradas se usa la misma sintaxis, cambiando “add” por “del” (si sólo hay una pasarela, para borrarla basta “ip route del default” o “route del default”).

NOTA: A diferencia del comando “route” (que permite varias entradas de pasarela por defecto “default” por interfaz), el comando “ip” sólo admite una entrada “default” por métrica. Si se intenta añadir una segunda (haya sido añadida la primera manualmente o por DHCP), el comando “ip” mostrará el mensaje “RTNETLINK answers: File exists”, “Error: x.x.x.x is a duplicate” o “Error: x.x.x.x is a garbage”.

De igual modo, el comando “ip” tampoco permite introducir dos entradas directas o indirectas restringidas para, exactamente, el mismo rango de direcciones (misma IP de subred y máscara) y métrica, incluso aunque se indiquen distintas interfaces “ethX” (el comando “route” sí lo permite). Resumen:

Entradas	ip route	route
----------	----------	-------

Directa* + Indirecta restringida	“1” por <i>IP_subred/mask</i> y <i>métrica</i>	“1” por <i>ethX</i> , <i>IP_subred/mask</i> y <i>métrica</i>
Indirecta general**	“1” por <i>métrica</i>	“1” por <i>ethX</i> , <i>IP_pasarela</i> y <i>métrica</i>

* Al asociar la IP a las interfaces (“`ip address`”), sí pueden generarse “2” (o más) entradas directas para la misma *IP_subred/mask* (asignando a dichas interfaces una IP dentro de la misma subred).

** Si existen dos entradas indirectas generales para la misma interfaz con igual métrica, la añadida en segundo lugar nunca podrá ser usada (por eso el comando “`ip`” no la permite). Si tendrán utilidad ambas, en caso de poseer distinta métrica (sin forzado de interfaz de usará la de menor métrica; el forzar la interfaz puede obligar a usar la de métrica mayor).

NOTA3: El comando “`ip route`” admite operaciones adicionales sobre la tabla:

- “change”: modificar una entrada existente.
- “replace”: modificar (change) o añadir si no existe (add) una entrada.
- “flush” borrar entradas de las interfaces indicadas (*ethX*, *all*).
- “get”: consultar la interfaz de red y pasarela que, con la configuración actual de la tabla de encaminamiento e interfaces, se usará si se intenta acceder a determinada dirección IP. El parámetro opcional “oif” permite hacer la consulta forzando una interfaz de salida (igual “-I” en el comando “ping”). Ejemplo:

```
ip route get 8.8.8.8 oif eth0
```

A.1.1.4. Configuración persistente de las Tarjetas de Red: Ficheros de Configuración

A continuación se muestra, a modo de ejemplo, los ficheros para aplicar una configuración persistente (mantenida tras apagar el equipo o al desactivar y reactivar la interfaz de red) a la tarjetas de red, según el SO usado:

- a) Linux Debian (un fichero para todas las interfaces):

/etc/network/interfaces

auto lo	# Activar la interfaz en arranque
iface lo inet loopback	# Configuración local
 auto eth0	 # Activar la interfaz en arranque
iface eth0 inet dhcp	# Configuración dinámica DHCP
 # auto eth1	 # Activar la interfaz en arranque
iface eth1 inet static	# Configuración estática
address 10.1.0.1	# IP interfaz eth1
netmask 255.255.252.0	# Máscara
network 10.1.0.0	# IP de subred
broadcast 10.1.0.255	# IP de difusión
# gateway 10.1.0.254	# Pasarela/Encaminador
# dns-nameservers 8.8.8.8	# DNSs interfaz (a insertar en /etc/resolv.conf)
# post-up comando	# Comando a ejecutar tras arrancar la interfaz
# post-down comando	# Comando a ejecutar tras desactivar la interfaz

b) Linux CentOS (un fichero por interfaz):

/etc/sysconfig/network-scripts/ifcfg-lo

DEVICE=lo	# Dispositivo dinámico eth1
ONBOOT=yes	# Activar interfaz en arranque
BOOTPROTO=static	# Configuración estática
IPADDR=127.0.0.1	# IP interfaz eth1
NETMASK=255.0.0.0	# Máscara
NETWORK=127.0.0.0	# IP de subred
BROADCAST=127.255.255.255	# IP de difusión
NM_CONTROLLED=no	# No usar Network Manager

/etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0	# Dispositivo dinámico eth0
ONBOOT=yes	# Activar interfaz en arranque
BOOTPROTO=dhcp	# Configuración dinámica (DHCP)
NM_CONTROLLED=no	# No usar Network Manager

/etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1	# Dispositivo dinámico eth1
ONBOOT=no	# No activar interfaz en arranque
BOOTPROTO=static	# Configuración estática
IPADDR=10.1.0.1	# IP interfaz eth1
NETMASK=255.255.255.0	# Máscara
NETWORK=10.1.0.0	# IP de subred
BROADCAST=10.1.0.255	# IP de difusión
# GATEWAY=10.10.10	# Pasarela/Encaminador
# DNS=8.8.8.8	# DNSs interfaz(/etc/resolv.conf)
# MACADDR=MAC	# Dirección MAC "virtual" a usar
NM_CONTROLLED=no	# No usar Network Manager

REFERENCIAS: Ficheros de configuración de interfaces de red:

- Linux Debian (fichero "interfaces"):
<https://www.debian.org/doc/manuals/debian-reference/ch05.en.html>
- Linux CentOS (ficheros "ifcfg-XXX"):
https://www.centos.org/docs/5/html/Deployment_Guide-en-US/s1

**A.1.1.5. Configuración persistente de la Tabla de Encaminamiento principal:
Entradas Manuales**

A partir de la configuración de las interfaces de red, el sistema añade automáticamente, en la tabla de encaminamiento, una entrada “directa” por cada interfaz, y una única entrada “indirecta genérica” (la de la última interfaz activada -orden alfabético de su fichero de configuración- con parámetro “GATEWAY”). Para añadir más entradas de este tipo, o para añadir entradas “indirectas restringidas”, existen tres opciones:

- a) Adición Manual: ejecutando directamente el comando “route” o “ip”.
- b) Adición automática al arrancar el SO: ubicando los anteriores comandos “route” o “ip” en un shell-script invocado al arrancar el sistema, tal como:
 - Fichero clásico (SysVinit): /etc/rc.d/rc.local
 - Servicio Systemd: /etc/systemd/system/servicio.service
 - Regla “udev” (invocada al cargarse el controlador de la tarjeta de red en “/etc/udev/rules.d/70-persistent-net.rules”).
- c) Adición automática con la activación de las tarjetas de red:

- Debian: usando las entradas “post-up” y “post-down” del fichero de configuración de las interfaces de red para ejecutar los comandos “route” o “ip”:

/etc/network/interfaces

```
...
auto eth0
iface eth0 inet static
address 10.1.0.1
    netmask      255.255.252.0
    network      10.1.0.0
    broadcast    10.1.0.255
    # gateway    10.1.0.254
post-up    ip route add 10.2.0.0/24                dev eth1
post-up    ip route add default                    via 10.1.0.3 dev eth1
post-up    ip route add 10.3.0.0/30 via 10.1.0.5 dev eth1
```

- Fedora (no se dispone de un equivalente a “post-up” para ejecutar comandos): para añadir entradas manuales a la tabla de encaminamiento se dispone de los ficheros “route-ethX”, específicos para ello. Estos ficheros admiten dos formatos posibles:

- ▶ Formato “ficheros ifcfg”:

/etc/sysconfig/network-scripts/route-eth0

```
ADDRESS0=10.2.0.0      # IP_Subred (Directa)
NETMASK0=255.255.255.0 # Máscara

GATEWAY1=10.1.0.3      # Pasarela (Indirecta Genérica)

ADDRESS2=10.3.0.0      # IP_subred (Indirecta Restrignida)
NETMASK2=255.255.255.252 # Máscara
GATEWAY2=10.1.0.5      # Pasarela
```

- ▶ Formato “comando ip”:

/etc/sysconfig/network-scripts/route-eth0

```
10.2.0.0/24                dev eth1    # Directa
default                    via 10.1.0.3 dev eth1    # Indirecta Genérica
10.1.0.4/30 via 10.1.0.5 dev eth1    # Indirecta Restrignida
```


ENTRADAS INDIRECTAS RESTRINGIDAS POR DHCP

Al igual que con la configuración local, cuando una interfaz de red toma su “IP”, “máscara” y “pasarela” de un servidor DHCP, automáticamente se añaden en la tabla de encaminamiento la entrada directa e indirectas correspondientes.

Vía DHCP pueden añadirse entradas “directas” o “indirectas” adicionales, incluyendo entradas indirectas restringidas, usando las denominadas “Rutas estáticas sin clase” (RFC 3442). En el caso de los servidores ISC DHCP (archivo “dhcpd.conf”), para añadir más entradas “directas” e “indirectas restringidas” (sólo se recomienda una indirecta genérica por interfaz, introducida con “option routers 10.0.0.254”):

```
# Definir el tipo de datos a usar para
# enviar esta información de encaminamiento
option classless-routes code 121 = array of unsigned integer 8;

# Para Clientes DHCP Windows
option classless-routes-win code 249 =array of unsigned integer 8;

subnet 10.0.0.0 netmask 255.255.255.0
{
    option broadcast-address 10.0.0.255;

    option routers 10.0.0.254; # GW (default): 10.0.0.254

    # Definir entradas indirectas restringidas
    # GW (192.168.100.0/24): 10.0.0.2
    # GW (10.4.0.0/30): 10.0.0.2
    option classless-routes 24, 192,168,100, 10,0,0,2,
                           30, 10,4,0, 10,0,0,2;

    # Alternativamente:
    # option rfc3442-classless-static-routes ...
    # option rfc3442-static-routes ...

    # Para Clientes DHCP Windows:
    option classless-routes-win 24, 192,168,100, 10,0,0,2,
                                30, 10,4,0, 10,0,0,3;

    # Alternativamente:
    # option windows-classless-static-routes ...
    # option windows-static-routes ...
}
```

A.1.1.6. Des/Activación de las interfaces de red

- a) Para aplicar la configuración, por ejemplo, de la interfaz eth1, puede ejecutarse el comando (estos comandos requieren que existan los anteriores ficheros de configuración de la interfaz eth1):

- Desactivar y desconfigurar la interfaz: `ifdown eth1`
- Configurar y activar la interfaz: `ifup eth1`

NOTA: El script “`ifup eth1`” usa internamente el comando “`ip addr add IP/nn dev ethX`”. Consecuentemente, si el fichero “`ifcfg-ethX`” tiene configurada para la interfaz una dirección distinta a las que actualmente tiene asignadas (`ip a`), al ejecutar el comando “`ifup ethX`” se le asignara a la interfaz como otra dirección adicional la indicada en “`ifcfg-ethX`” (este funcionalidad la realiza el comando “`ifup`” independientemente de que la interfaz se encuentre activada o desactivada).

El script “`ifdown ethX`” usa internamente el comando “`ip addr flush dev eth0 scope global`” para desconfigurar completamente los valores de la interfaz.

Para todas las interfaces sería:

- Debian: `service networking restart`
- CentOS: `service network restart`

- b) La activación de las interfaces de red puede hacerse:

- Sólo activar (no configura, no requiere que exista el fichero de configuración de la interfaz):

```
ip link set eth1 up
ifconfig eth1 up
```

- Activar y configurar (leyendo fichero de configuración de red):

```
ifup eth1
```

- c) De forma similar, la desactivación de las interfaces de red puede hacerse:

- Sólo desactivar:

```
ip link set eth1 down
ifconfig eth1 down
```

- Desactivar y “limpiar” datos como IP interfaz (requiere la existencia del fichero de configuración de red):

```
ifdown eth1
```

- d) Tanto “ifup” como “ifdown” activan o desactivan aquello que indique el fichero de configuración de la tarjeta de red en el momento en que son invocados. Por ello, el proceso recomendable de uso es:

1º Desactivar la interfaz: `ifdown ethX`

2º Reconfigurarla: modificar `ifcfg-ethX` (centOS) o `interfaces` (Debian).

3º Reactivar la interfaz con la nueva configuración: `ifup ethX`

Si no, pueden producirse efectos extraños. Por ejemplo:

0º Configuración con `BOOTPROTO=dhcp`

1º Activación de interfaz: `ifup ethX` => Se crea proceso cliente DHCP `dhclient`.

2º Modificación configuración a `BOOTPROTO=static`

3º Desactivación interfaz: `ifdown ethX` => NO elimina proceso `dhclient` (porque la configuración indica en este momento “static”).

4º Reactivación de interfaz: `ifup ethX` => La tarjeta se activa con la configuración manual configurada, pero el client `dhclient` sigue activo. Resultado: cada cierto tiempo (el indicado en la configuración del cliente DHCP), `dhclient` modificará automáticamente los valores de la tarjeta por DHCP.

A.1.1.7. Tablas de Encaminamiento Linux

En Linux, hay 4 tipos tablas de encaminamiento, consultables a través del comando “ip”

- a) Local: aplica el encaminamiento a las direcciones locales y las direcciones de subred y difusión:

```
ip rule show
```

```
ip route show table local
```

- b) Principal (la habitual; el comando “route” sólo muestra ésta)

```
ip route show [table main]
route -n
```

- c) Default: normalmente vacía, se usa para operaciones de post-procesamiento cuando el paquete no ha producido “match” con las reglas anteriores.

```
ip route show table default
```

- d) Tablas personalizadas

La prioridad entre las mismas es configurable, a través de:

```
cat /etc/iproute2/rt_tables
```

REFERENCIAS:

<http://www.policyrouting.org/PolicyRoutingBook/ONLINE/TOC.html>
<http://linux-ip.net/html/routing-tables.html>

A.1.1.8. Consulta/Modificación de parámetros de funcionamiento de las interfaces de red

- a) Flags de una tarjeta de red (impresos con “ip a”, “ip l” con el formato “ethX: <FLAG1, FLAGS, ...>”), documentados en “man 7 netdevice”:

Flag	Significado
ALLMULTI	Recibe todos los paquetes multicast
AUTOMEDIA	Selección automática del medio activada
BROADCAST	Dirección de difusión configurada con ese uso (brd)
DEBUG	Flag interno de depuración activado
DORMANT	Interfaz “dormida” (no preparada para enviar paquetes, en espera de algún evento)
DYNAMIC	Dirección de red perdida al desactivar (DOWN) la interfaz
ECHO	Paquetes de echo enviados

Flag	Significado
LOOPBACK	Interfaz de “bucle local”
LOWER_UP	Actividad L1 en el cable (cable conectado y con alimentación)
MASTER	Tarjeta maestro en un bonding
MULTICAST	Soporte multicast activado
NO_CARRIER	Son Cable de red o cable de red sin alimentación
NOARP	Protocolo ARP desactivado
NOTRAILERS	No usar encapsulación “trailer” (RFC 893)
POINTOPOINT	Interfaz punto a punto
PORTSEL	Tipo de medio (10 Base T, ...) configurable
PROMISC	Modo promiscuo activado
RUNNING	Recursos necesarios reservados
SLAVE	Tarjeta secundaria en un bonding
UP	Activada (envío/recibe tráfico)

La des/activación de aquellas características (flags) que son configurables en la interfaz (algunos no son configuraciones si no estados; dentro de los configurables, dependerá de si el firmware de la interfaz permite configurarlo o no) se realizará en general con:

```
ip link set ethX      flag off
ifconfig ethX [+/-] flag
```

Ejemplos

Modo promiscuo (flag PROMISC)	ip link set ethX promisc off/on ifconfig ethX [-]promisc
Protocolo ARP (flag NOARP)	ip link set ethX arp off/on ifconfig ethX [-]arp
Broadcast (flag BROADCAST)	ip link set ethX broadcast off/on ifconfig ethX [-]broadcast

ADVERTENCIA: Al desactivar una tarjeta de red (`ip link set down dev eth1`), desaparecerán los flags “UP” (recibe tráfico) y “LOWER_UP” (alimentación física) de la tarjeta (`ip link show eth1`). Sin embargo, según el sistema Linux (es un comportamiento determinado a nivel de controlador, pudiendo variar según la versión de éste que use el SO), la tarjeta se:

- Mantiene físicamente alimentada (leds encendidos), a pesar de no mostrar el flag “POWER_UP”: provocando que en el conmutador el puerto se muestre como “Up”.
- Si cortan la alimentación de la tarjeta (como corresponde a la desaparición del flag “POWER_UP”): provocando que en el conmutador el puerto se muestre como “Down”.

- b) Comprobar que una tarjeta de red ha sido detectada (`lspci` también muestra el controlador de la tarjeta):

```
lspci
lspci -nn | grep Network
dmesg | grep Network
```

- c) Consultar el módulo (y así el fabricante y tarjeta de red concreta) asociado a una interfaz de red `ethX`:

- Opción 1 (cambie “ethX”):

```
ls -l /sys/class/net/ethX/device/driver/module
```

- Opción 2 (paquete “lshw”):

```
lshw -C network
```

- Iluminación (blinkig) de los LEDs de la tarjeta:

```
ethtool -p ethX
```

SUGERENCIA: Esta funcionalidad resulta especialmente útil para identificar el dispositivo físico asociado a cada interfaz de red. No obstante, sólo funcionará si esta está soportada por la tarjeta de red (algunos modelos de tarjetas físicas o las tarjetas de red virtuales no la soportan). Si no está soportada, se mostrará el mensaje:

```
Cannot identify NIC: Operation not supported
```

- d) Consultar el estado del cable de red (conectado o no) de una tarjeta de red: estado “yes/no”, respectivamente, en la línea “Link detected:” del comando

```
ethtool ethX | grep -i "Link detected"
```

Alternativamente pueden usarse los flags “NO-CARRIER” y “LOWER_UP” de la tarjeta de red, que aportan la misma información.

- e) Consultar las características (velocidades soportadas, full duplex, ...) de una tarjeta de red “ethX”:

```
ethtool ethX
```

- f) Velocidad de trabajo:

- Consulta:

```
ethtool ethX | grep -i Speed
mii-tool ethX
dmesg          | grep -i duplex
```

- Modificación puntual (se pierde al reiniciar la interfaz) de la velocidad:

```
ethtool -s ethX speed 10/100/1000 duplex full
```

- Modificación por configuración (se mantiene al reiniciar la interfaz):

Fichero ifcfg-ethX (CentOS)	ETHTOOL="speed 100 duplex full"
Fichero interfaces (Debian)	post-up ethtool -s ethX speed 100 duplex full

NOTA: color (habitual) del led de las tarjetas de red según la velocidad configurada

Mbps	10	100	1000
Color	Apagado	Verde	Amarillo/Naranja

g) Modificar la unidad de transferencia máxima (MTU) de una interfaz:

```
ip link set mtu valor dev ethX
ifconfig eth0 mtu valor
```

h) Modificar la longitud de la cola de transmisión de una interfaz:

```
ip link set txqueuelen valor dev ethX
```

i) Dirección MAC de la interfaz:

- Consultar la dirección MAC actualmente usada (MAC funcional):

```
ip a ls ethX
ifconfig ethX
```

- Modificar la dirección MAC funcional:

- ▶ Modificación puntual (se pierde al reiniciar la interfaz):

```
ip link set ethX address MAC
ifconfig ethX hw ether MAC
macchanger -m MAC ethX
```

- ▶ Modificación por configuración (se mantiene al reiniciar la interfaz):

Fichero ifcfg-ethX (CentOS)	MACADDR=MAC
Fichero interfaces (Debian)	hwaddress ether MAC

- Consultar la dirección MAC permanente (real) de una interfaz (si la MAC funcional ha sido cambiada con “macchanger” o similar, “ifconfig” muestra esa MAC funcional, pero este comando seguirá mostrando la MAC física real):

```
ethtool -P ethX
```

- Restaurar la dirección MAC real (permanente) de la interfaz:

```
ip link set ethX address $(ethtool -P ethX | cut -d " " -f 3)
macchanger -p ethX
```


j) Encendido remoto (WOL):

- Consulta (debe aparecer el flag “g”: “Wake-on: g”):

```
ethtool ethX | grep -i Wake-on
```

- Activación (“soap” indica una posible clave):

```
ethtool -s ethX wol g [sopass cc:ll:aa:vv:ee:xx]
```

- Desactivación:

```
ethtool -s ethX wol d
```

- Modificación permanente:

Fichero ifcfg-ethX (CentOS)	ETHTOOL_OPTS="wol g"
Fichero interfaces (Debian)	post-up ethtool -s ethX wol g

A.1.1. Configuración de interfaces Wi-Fi

Para la configuración de las tarjetas de red inalámbricas se recomienda usar el comando “iw” (paquete “iw”), que reemplaza al comando clásico “iwconfig” (paquete “wireless-tools”).

INSTALACIÓN:		
Comando “iw”	CentOS	yum install iw
	Debian	apt-get install iw
Comandos clásicos (iwconfig, ...)	CentOS	yum install wireless-tools
	Debian	apt-get install wireless-tools
Soporte WPA	CentOS	yum install wpa_supplicant
	Debian	apt-get install wpasupplicant

CONTROLADORES: para poder usar una tarjeta inalámbrica es necesario (igual que con cualquier otro hardware) cargar el controlador (driver) de la misma. Los sistemas Linux actuales disponen de la posibilidad de autodetectar gran cantidad de chipsets Wi-Fi (siempre que estén marcados en el kernel con el que se haya arrancado). Si no fuese el caso, deberá instalar el controlador manualmente, revisando los módulos disponibles en el kernel, instalando el módulo del chipset desde repositorios (Ralink “apt-cache search ralink”, Atheros “apt-cache search madwifi”, Intel “apt-cache search firmware | grep -i intel | grep -i firmware”, reutilizar driver Windows usando “Ndiswrapper”, ...) o acudiendo a la web del fabricante de la tarjeta.

La configuración de las interfaces Wi-Fi presenta dos partes:

- Parámetros de red (IP, máscara, tabla de encaminamiento, ARP, ...): se configura con los comandos vistos anteriormente (estos son válidos para interfaces cableadas o inalámbricas).
- Parámetros específicos de interfaces Wi-Fi: se detalla en este apartado.

A continuación se resumen los principales comandos útiles para configurar los parámetros específicos de las interfaces Wi-Fi (man iwconfig, man iwlist):

a) Detección de redes Wi-Fi:

Tasas de Tx soportadas por una interfaz	<code>iwlist wlanX bitrate</code>
---	-----------------------------------

b) Análisis de las características de una interfaz Wi-Fi (man iwlist):

```
iwlist wlanX parametro
```

Soporte de	Sistemas encriptación	Canales/Frecuencias	Tasa Tx	...
parametro	keys / enc	freq / channel	rate	...

c) Configuración manual temporal de los parámetros de la interfaz (man iwconfig):

```
iwconfig interface [essid X] [nwid N] [mode M] [freq F]
[channel C][sens S ] [ap MAC_AP ][nick NN ] [rate R]
[rts RT] [frag FT] [txpower T] [enc E] [key K]
[power P] [retry R] [modu M] [commit]
```

ssid	ESSID (Nombre de red)
key	Clave (<i>solo WEP</i>): “clave hexadecimal” o “s:clave ASCII”
mode	Modo: Ad-Hoc (sin Access Point), Managed (red con APs), Master (el nodo es un AP), Repeater (reenvía entre dos nodos wifi), Secondary (master/repeater de backup), Monitor (no conectado, monitoriza pasivamente los paquetes) or Auto.
channel	Canal (admite “auto”).
freq	Frecuencia en Hz (o con múltiplos “k”, “M”, “G”).
ap	MAC del AP con el que registrarse forzosamente (admite “auto”)
rate/bit	Tasa de bit (en b/s). Admite “auto”.
txpower	Potencia de Tx en dBm (por omisión) o en “mW”. Admite “on” (activar), “off” (desactivar) la interfaz radio y “auto”.
...	...

Ejemplos básicos:

Interfaces inalámbricas del equipo detectadas y sus propiedades (ESSID, Mode, Potencia, ...)	<code>iwconfig [wlanX]</code>
Configurar ESSID y clave WEP	<code>iwconfig wlan0 essid ESSID key WIRELESS_KEY</code>
Configurar tasa de Tx	<code>iwconfig wlanX rate 24M</code>
Tasas de Tx soportadas	<code>iwlist wlanX bitrate</code>

Para conectarse a redes con WPA/2 se requiere “wpa_supplicant” (la forma de generar el fichero “wpa_supplicant.conf” se indica más abajo en el punto de configuración permanente mediante ficheros):

```
wpa_supplicant -D[driver] -i[wlanX] -c[/path/to/config]
```

Ejemplo:

```
wpa_supplicant -B -iwlanX \
-c/etc/wpa_supplicant/wpa_supplicant.conf -Dwext
```

d) Configuración temporal interactiva con “wpa_client” (man wpa_client): ejemplos

```
wpa_client -iwlanX
> scan
> scan_results
> add_network
> set_network 0 ssid "ESSID"
> set_network 0 psk "passphrase"
> enable_network 0
> reconnect
> status
> quit
```

Pueden aplicarse desde el terminal con:

```
wpa_cli -iwlanX comando_wpa-client
```

- e) Configuración persistente (mantenida tras apagar el equipo o al desactivar y reactivar la interfaz de red) por configuración: los parámetros de red se configuran como se indicó anteriormente. Estos ejemplos se centran en los parámetros específicos de Wi-Fi.

- Linux Debian:

/etc/network/interfaces

auto eth0	
iface eth0 inet dhcp	
wireless-essid essid	# ESSID de nuestra Wi-F
wireless-key clave	# WEP: Clave de acceso a la Wi-Fi
 # auto eth1	# Igual que interfaces cableadas
iface eth1 inet dhcp	
address 10.1.0.1	
...	
wpa-ssid valor_ESSID	# WPA: ESSID
wpa-proto tipo	# WPA: WPA o RSN (WPA2)
wpa-key-mgmt WPA-PSK	# WPA: Seguridad PSK
wpa-psk valor_PSK	# WPA: PSK hexadecimal
wpa-pairwise TKIP	# WPA: TKIP (WPA1) o
wpa-group TKIP	# o CCMP (WPA2)
wpa-driver wext	# WPA: Driver
# wpa-conf /etc/wpa_supplicant	# WPA: Fichero configuración
# /wpa_supplicant.conf	# (alternativamente)

- Linux CentOS:

/etc/sysconfig/network-scripts/ifcfg-wlan0

```

DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp           # Encriptacion WEP
TYPE=Wireless             # Tarjeta inalámbrica
ESSID=ssid                # ESSID
KEY=clave_WEP             # Clave WEP
CHANNEL=canal             # Canal
MODE=Modo                 # Modo
WPA=no                    # WEP (por omision)
NM_CONTROLLED=no

```

/etc/sysconfig/network-scripts/ifcfg-wlan1

```

DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Wireless             # Encriptacion WPA
WPA=yes                   # Tipo WPA (WPA_SUPPLICANT)
NM_CONTROLLED=no

```

/etc/sysconfig/wpa_supplicant

```

INTERFACES="-ieth1 -iwlan0" # Interfaces a configurar con
                              # wpa_supplicant. Formato:
                              # -iINTERFAZ
DRIVERS="-Dwext"            # Driver
OTHER_ARGS="-f /tmp/wpa.lo- # Fichero de logs
g"

```

Para ambos Linux, en el caso de WPA, el PSK se obtiene a partir del “WPA passphrase” y el “ESSID” con el comando:

```
wpa_passphrase ssid
```

Este comando vuelca por pantalla (puede usarse redirección “>”) el fichero “wpa_supplicant.conf” con los parámetros básicos y el PSK.

```
/etc/wpa_supplicant/wpa_supplicant.conf
```

```
ctrl_interface=/var/run/wpa_supplicant
eapol_version=1
ap_scan=1
fast_reauth=1

network={
    ssid="valor_ESSID"
    scan_ssid=1
    proto=RSN          # WPA=WPA1, RSN=WPA2/IEEE 802.11i
    key_mgmt=WPA-PSK
    pairwise=CCMP TKIP
    group=CCMP TKIP WEP104 WEP40
    psk=valor_PSK
    priority=5
}
```

En general, para recargar los cambios realizados en “wpa_supplicant.conf” (debe arrancarse como servicio al iniciar el sistema):

```
service wpa_supplicant restart
service network restart
```

REFERENCIAS: Puede consultar más información de los parámetros disponibles con “man wireless”, “man wpa_supplicant” y “man wpa_supplicant.conf”.

También puede encontrar información de otros comandos disponibles en el paquete “wireless-tools” (iw, iwevent, iwgetid, iwpriv, iwspy) en:

https://en.wikipedia.org/wiki/Wireless_tools_for_Linux
<https://wiki.centos.org/HowTos/Laptops/WpaSupplicant>

A.1.2. Asociación de varias IPs a una misma interfaz

Una misma interfaz Ethernet (NIC) puede poseer varias direcciones IP (usando todas ellas la misma MAC), afectando consecuentemente a la tabla de encaminamiento (en cuanto a las capturas de tráfico, se trata de una única interfaz). Para asociar varias IPs a una interfaz física “ethX”:

- a) Asociación manual temporal (hasta reiniciar el servicio “network”):

- Comando “ifconfig”: para cada nueva IP requiere que crear una nueva “interfaz” con el nombre “ethX:n” (siendo “n” un entero):

```
ifconfig ethX:0 IP0
ifconfig ethX:1 IP1
...
```

NOTA: Hay que darle nuevos nombres porque si se escribiese:

- “ifconfig ethX IP0”: ifconfig simplemente cambiaría la IP de ethX.
- “ifconfig nombre IP”: ifconfig no sabría con que interfaz física asociar esa IP.

- Comando “ip”: no necesita dar nuevos nombres. Simplemente, asocia varias IPs a esa interfaz.

```
ip addr add IP0/m dev ethX [label nombre0]
ip addr add IP1/m dev ethX [label nombre1]
...
```

Si se desea que el comando “ifconfig” imprima estas direcciones, se les debe asignar una etiqueta con formato “Linux-2.0 net aliases” ethX.Z”.

- b) Configuración persistente (mantenida tras apagar el equipo o al desactivar y reactivar la interfaz de red) por configuración: sólo admiten el funcionamiento de “ifconfig” basado en el formato “ethX:n”:

- Linux Debian:

/etc/network/interfaces

```
auto ethX:0
iface ethX:0 inet static
    address IP0
    ...

auto ethX:1
iface ethX:1 inet static
    address IP1
    ...
```

Primera IP

Otra IP

- Linux CentOS:

```
/etc/sysconfig/network-scripts/ifcfg-ethX:0
```

```
DEVICE=ethX:0
ONBOOT=yes
BOOTPROTO=static
APADDR=IP0
...
```

```
# Primera IP
```

```
/etc/sysconfig/network-scripts/ifcfg-ethX:1
```

```
DEVICE=ethX:1
ONBOOT=yes
BOOTPROTO=static
APADDR=IP1
...
```

```
# Otra IP
```

c) Listado de direcciones IP de una interfaz ethX:

- “ip addr list” (no aparecen en “ip link list”): muestra todas las IPs creadas (con ifconfig e ip). Para las creadas con ifconfig les aparecerá el nombre “ethX:n” (para las creadas con “ip”, simplemente “ethX”). Aparece una única entrada por interfaz física, y dentro las distintas IPs asignadas.

```
ip addr list [ethX]
```

- “ifconfig”: sólo muestra las interfaces que posee una nombre/etiqueta con formato “Linux-2.0 net aliases” ethX.Z (creadas con “ifconfig” o con el comando “ip” usando un “label” de ese formato “ethx:Z”). NO muestra las IPs asignadas con el comando “ip” a las que no se le haya aplicado ese label (motivo por lo que el que **se desaconseja el uso de ifconfig**, al no imprimir todas las IPs realmente asignadas a la interfaz). Imprime una entrada por la interfaz física y otra por cada IP añadida.

```
ifconfig
ifconfig ethX      # Sólo la IP nativa de interfaz física
ifconfig ethX:n    # Sólo la IP asociada a ethX:n
```

d) Eliminación de una IP:

```
ip addr del IP0/m dev ethX      # Admite "ethX:n", pero no
                                # es necesario el ":n"
ifconfig ethX:n down
```

Para borrar TODAS las IPs (incluyendo la nativa de la interfaz física):

```
ip address flush dev ethX
```

NOTA: Al desactivar (down) la interfaz física “ethx” dejan de funcionar todas las IPs, pero no se eliminan. Al reactivarla (up) siguen apareciendo dichas IPs.

A.1.3. Agrupación de interfaces de red (bonding)

Un “bonding” o “teaming” (agrupación de tarjetas para balanceo de carga) permite configurar varias interfaces con una misma dirección IP, trabajando a nivel de red como una única tarjeta (a nivel L2 cada una usa su dirección MAC) con mayor tasa de transferencia.

A.1.4. Interfaces Ethernet virtuales sobre interfaces físicas

Las interfaces Ethernet virtuales (“pseudo-ethernet” o “macvlan”) se comportan como varias interfaces físicas (cada una con su MAC) conectadas a un conmutador (y éste al cable físico de red que entra en la tarjeta física). Así, se tiene un “conmutador virtual” al cual están conectadas la interfaz física y las distintas interfaces Ethernet virtuales, cada una de ellas con su MAC y su IP.

Para crear una (o más) interfaz virtual con direcciones “IP” y “MAC” asociada a la interfaz física ethX (la configuración de IP, MAC... y activación se hace de igual modo que en las interfaces físicas):

- a) Creación (y eliminación) manual temporal (hasta reiniciar el servicio “network”): sólo es posible usando el comando “ip”.

```
ip link add      link ethX      nombre_interfaz_virtual      \  
                address MAC    type macvlan    [mode bridge]  
  
ip addr add IP/nn dev nombre_interfaz_virtual
```

Para que empiece a operar, debe activarse la interfaz (y su correspondiente desactivación con “down”):

```
ip link set nombre_interfaz_virtual up
```

NOTA: “ip” no admite cualquier dirección MAC para las interfaces. Comprueba que el comienzo sea un tipo de dirección MAC admitido (por ejemplo, no puede comenzar por “01:”).

NOTA: “ip” admite varios tipos de interfaces virtuales además de “macvlan”, tales como:

- “veth”: crea interfaces virtuales conectadas entre sí por un conmutador virtual, SIN vinculación con ninguna interfaz física.
- “vlan”: crea una interfaz VLAN IEEE 802.1q asociada a una interfaz física “ethX”, con una determinada etiqueta. Cuando se reciben de la interfaz física mensajes con dicha etiqueta, se entregan a esta interfaz, y al enviar se etiquetan con dicha etiqueta.

```
ip link add name nombreA type veth peer name nombreB
```

```
ip link add name ${VLAN interface name} \
    link ethX type vlan id ${tag}
```

```
ip link add name eth0.110 link eth0 type vlan id 110
```

- b) Configuración persistente (mantenida tras apagar el equipo o al desactivar y reactivar la interfaz de red) por configuración: sólo admiten que el nombre de la interfaz virtual sea el nombre de la interfaz física asociada seguido de “:Z” (siendo “Z” un entero) y con la misma MAC que la interfaz física:

- Linux Debian:

/etc/network/interfaces

```
auto nombre
iface nombre inet dhcp
    pre-up    ip link add nombre link ethX type macvlan mode bridge
    post-down ip link del nombre link ethX type macvlan mode bridge
    ...
```

- Linux CentOS: no está directamente soportado. Si se añaden los ficheros (documentación accesible en “<https://github.com/larsks/ini-scripts-macvlan>”):

```
/etc/sysconfig/network-scripts/ifup-macvlan
https://raw.githubusercontent.com/larsks/ini-scripts-macvlan/master/ifup-macvlan
```

```
/etc/sysconfig/network-scripts/ifdown-macvlan
https://raw.githubusercontent.com/larsks/ini-scripts-macvlan/master/ifdown-macvlan
```

entonces sería posible usar:

/etc/sysconfig/network-scripts/ifcfg-nombre

```

DEVICE="nombre"
DEVICETYPE="macvlan"
TYPE="macvtap"
BOOTPROTO="dhcp"
ONBOOT="yes"
MACADDR="AB:CD:EF:GH:IJ:KL"
MACVLAN_PARENT="eth0"
...
NM_CONTROLLED="no"

```

- c) Listado de interfaces Ethernet virtuales: sólo las muestra el comando “ip”, con el formato “nombre_interfaz_virtual@ethX”:

```

ip addr ls [nombre_interfaz_virtual]
ip link ls [nombre_interfaz_virtual]

```

- d) Eliminación de interfaces virtuales:

```
ip link del dev nombre_interfaz_virtual
```

NOTA: Al desactivar (down) la interfaz física “ethx” dejan de funcionar todas sus interfaces virtuales asociadas,, pero no se eliminan. Al reactivarla (up) siguen apareciendo dichas interfaces virtuales.

A.2. Tabla/Caché ARP

Imprimir Tabla ARP	Con resolución inversa a nombres	ip -r neighbour ip -r neigh ip -r n arp
	Sin resolución inversa	ip neighbour ip neigh ip n arp -n cat /proc/net/arp
Imprimir sólo entradas con cierta IP		arp [-n] IP [-i ethX] ip n show IP [dev ethX]
Añadir una entrada manual		ip n add IP lladdr MAC dev ethX [nud estado] arp -s IP MAC -i ethX arp -s IP -D -i ethX #Usa MAC #de ethX

Cargar entradas manuales del fichero /etc/ethers (Lista: MAC IP)		<code>arp -f /etc/ethers</code>
Cambiar la IP de una entrada*		<code>ip n change IP lladdr MAC dev ethX [nud estado]</code>
Borrar MAC de entrada manual/automática (se autoborra conforme al ciclo de vida de la tabla)		<code>ip n del IP dev ethX arp -d IP -i ethX</code>
Borrar las entradas <i>dinámicas</i>	Sólo la MAC (NO la entrada)	<code>ip [-s -s] neigh flush all/ethX</code>
	Entrada Completa	<code>ip link set ethX arp off; ip link set ethX arp on</code>
Imprimir el estado de las entradas		<code>ip neigh show ip -s neigh</code>
Cambiar el estado		<code>ip n change IP dev ethX nud estado</code>
Des/Activar protocolo ARP en interfaz		<code>ip link set ethX arp on/off ifconfig ethX [-larp]</code>

* **NOTA:** La orden “flush” elimina la dirección MAC de la entrada, no la entrada en sí misma. La entrada se autoborra con el siguiente proceso:

1º Al aprender una MAC (ARP Request en envío o recepción) aparece en la tabla en estado (inicialmente “DELAY”) “REACHABLE”.

2º Tras un tiempo sin usarse (consultarse localmente) esa entrada, pasa a estado “STALE” (antes de usarla, se envía ARP Request para comprobar si sigue siendo válida). Dicho tiempo es de “60 s” habitualmente, estando configurado en el parámetro del kernel en tiempo de ejecución (man 7 arp):

```
/proc/sys/net/ipv4/neigh/default/gc_stale_time
```

3º Si sigue sin usarse la entrada, o se le aplica “flush”, se elimina la MAC, pasando la entrada a estado “FAILED” (o “INCOMPLETE”).

4º Si no se actualiza la MAC de la entrada (ARP Request o Reply), se elimina completamente la entrada.

A.3. Resolución de nombres

Los servidores DNS usados por el sistema se controlan desde el fichero “/etc/resolv.conf”. Un posible ejemplo de este fichero es el siguiente (Linux sólo usa los tres primeros “nameserver”; consulte “man resolv.conf” para mas información):

/etc/resolv.conf

```
domain ait.us.es
search ait.us.es
nameserver 172.16.17.254
nameserver 8.8.8.8
nameserver 150.214.186.69
nameserver 150.214.130.15
nameserver 130.206.1.2
nameserver 8.8.8.4
```

A.4. Servicio DHCP

Cliente y Servidor arrancados como servicios del sistema.

A.4.1. Servidor ISC DHCP

Manuales	dhcpd(8), dhcpd.conf(5), dhcpd.leases(5), dhclient-script(8), dhcp-options(5), dhcp-eval(5)
Configuración del servidor	/etc/dhcp/dhcpd.conf
Fichero de licencias del servidor	/var/lib/dhcp/dhcpd.leases
Fichero de logs (por omisión)	/var/log/messages

A.4.2. Cliente DHCP

Manuales	dhclient(8), dhclient.conf(5), dhclient.leases(5), dhcp-options(5)
Configuración del cliente	dhclient [-cf /etc/dhcp/dhclient.conf]

Fichero de licencias del cliente	<code>dhclient [-lf /var/lib/dhcp/dhclient.leases]</code>
Solicitar (DHCP DISCOVER) o Renovar (DHCP REQUEST) una licencia por ethX	<code>dhclient ethX</code>
Liberar la licencia actual (DHCP RELEASE)	<code>dhclient -r ethX</code>
Forzar activación del flag (bit) de broadcast (DISCOVER y REQUEST)	<code>dhclient -B ethX</code>

A.4.3. Agente de Reenvío (Agent Relay) DHCP

Paquete	Fedora	<code>yum install dhcp</code>
	Debian	<code>apt-get install isc-dhcp-relay</code>
Manuales		<code>dhcrelay (8)</code>
Fichero de configuración	Fedora	<code>/etc/sysconfig/dhcrelay</code>
	Debian	<code>/etc/default/isc-dhcp-relay</code>
Arranque	Fedora	<code>service dhcrelay restart</code>
	Debian	<code>service isc-dhcp-relay restart</code>

Formato del fichero de configuración:

```

SERVERS="IP1 IP2"      # IP de Servidores DHCP a los
que                  # redirigir (se reenvía a todos)

INTERFACES="eth1 eth2" # Interfaces en las que trabajar
                     # (para recibir o reenviar)

OPTIONS=" "           # Opciones adicionales

```

A.5. Análisis de sockets

El comando clásico “netstat” se encuentra en estado obsoleto, en favor del comando “ss” (Socket Statistics, `man 8 ss`). Dicho comando presenta una sintaxis similar a “netstat”, con algunos aspectos adicionales como la posibilidad de aplicar filtros de búsqueda.

```
ss [opciones] [ filtro ]
```

En su salida estándar, el comando “ss” muestra las siguientes seis columnas:

- “netid”: indica el tipo de socket y el protocolo de transporte.

netid	Comunicación	Tipo Socket	Protocolo de transporte
p_raw	Red	Paquete	Raw
p_dgr			Datagrama
raw		IP	Raw
tcp			TCP
udp			UDP
sctp			SCTP
u_dgr	Local	UNIX	Datagrama
u_str			Flujo (stream)
u_seq			Secuencia
nl		Netlink	PIDs

SOCKETS: Puede consultar la información de los distintos tipos de sockets en “man 7 socket, man 7 raw, man 7 udp, man 7 tcp”. De forma resumida, “ss” muestra los sockets de tipo:

- Paquete (AF_PACKET: man 7 packet): comunicación de red a nivel L2 Ethernet. Entregan (al proceso asociado al socket) el paquete Ethernet completo (incluyendo la cabecera, tipo “Raw”) o sólo los datos (tipo “Datagrama”).
- IP (AF_INET/6: man 7 ip, man 7 ipv6): comunicación de red sobre protocolo IP. Entregan (al proceso asociado al socket) el datagrama “IP” (sockets Raw, con o sin cabecera) o el mensaje de transporte “TCP”, “UDP” o “SCTP” (con o sin cabecera).
- UNIX (AF_UNIX, AF_LOCAL: man 7 unix): comunicación local entre procesos, usando ficheros virtuales. Tipo Datagrama (orientado a datagrama, como UDP), Flujo (orientado a flujo, como TCP) y Secuencia (orientado a conexión pero no a flujo, como SCTP).
- Netlink (AF_NETLINK: man 7 netlink): comunicación local, principalmente con el kernel, usando PIDs (RFC 3549).

Dado que “ss” imprime al final los sockets IP, es habitual ejecutarlo combinado con el comando “tail” para que estos se impriman en primer lugar (o “less” o “more” para que se imprima pantalla a pantalla):

```
ss -l | tail -10
```

- “State”: estado del socket.
- “Recv-Q” y “Send-Q”: datos en las colas de recepción y transmisión.
- “Local Address:Port” y “Peer Address:Port”: direcciones origen y destino del socket.

Usos habituales del comando “ss”:

Lista total de sockets abiertos	netstat -a	ss -a
Sockets “de escucha” (servicios abiertos)	netstat -l	ss -l

Sockets de conexión (conexiones establecidas; omite los sockets en estado “listen, time-wait, syn_rcv, close”) de cualquier tipo	netstat	ss
Sockets de red de “sólo de conexión”, “sólo de escucha” o ambos, de tipo: TCP (t), UDP (u), RAW (w) y SCTP (S)	netstat -tuWS	ss -tuWS
	netstat -ltuWS	ss -ltuWS
	netstat -atuWS	ss -atuWS
Sockets Unix	netstat -x	ss -x
Sólo sockets IPv4 (4) o IPv6 (6)	netstat -a4 netstat -a6	ss -a4 ss -a6
Con Información del proceso asociado (p)	netstat -ap	ss -ap
Memoria usada por el socket (m)	netstat -lm	ss -lm
Información TCP interna (i) o timers (o)	netstat -lti netstat -to	ss -lti ss -to
Imprimir en formato numérico “n” (sin resolución a nombres)	netstat -an	ss -an
Resolver IP a nombres (por omisión, sólo se resuelven los puertos)	netstat -ar	ss -ar
Resumen de sockets abiertos	netstat -s	ss -s

Además de estos comandos, existen otros relacionados con los sockets. Por ejemplo, para obtener el proceso que tiene abierto el puerto “80” pueden usarse los siguientes comandos:

```
lsof      -i          :80
netstat -anp | grep :80
ss        -anp | grep :80
```

De modo similar, para obtener los puertos abiertos por un determinado proceso (con PID) puede usarse:

```
lsof      -i -P -a -p PID
lsof      -i      | grep PID
lsof      -i      | grep nombreProceso
ss -anp   | grep PID
ss -anp   | grep nombreProceso
```

El comando “netstat” permite aplicar filtros basados en el estado del socket. “ss” permite, además, filtros más avanzados, basados en el número del puerto o el valor de la dirección IP. Para filtrar por puerto o IP, se usa una sintaxis parecida a la de los filtros de captura “libpcap”

(mayoritariamente, los símbolos de los operadores matemáticos deben escaparse “\op” en el intérprete de comandos):

Operadores	Lógicos	and (&, && - operador por omisión), or (,), not (!)
	Aritméticos (para puertos sólo)	lt (<), gt(>), eq (=, ==), ge (>=), ne (!=)
	Agrupación	(,)

Operandos (datos del socket)	Estado*	state ESTADO_ALIAS exclude ESTADO_ALIAS
	Dirección (y puerto opcionalmente)** NOTA: Origen = Columna “Local” Destino = Columna “Peer”	src dirección[:puerto] dst dirección[:puerto] NOTA: Sin “:puerto” o con “:*” o “:” indica cualquier puerto
	Puerto (sólo)	sport operador [:]puerto dport operador [:]puerto
	Puertos asociados sistema local	autobound

* Para filtrar por estado, se admiten los Identificadores TCP “established, syn-sent, syn-recv, fin-wait-1, fin-wait-2, time-wait, closed, close-wait, last-ack, listen, closing” y los siguientes “ALIAS”:

Alias	Estados
all	Todos
connected	Todos salvo “listen” y “closed”
synchronized	Igual que “connected”, menos “syn-sent”
bucket	“time-wait” y “syn-recv”
big	Opuesto a “bucket”

** El formato de las direcciones depende de su tipo:

Tipo dirección	Formato (=Columnas “Local/Peer Address:Port”)
inet	IPv4. Admite “prefijo” y “prefijo/máscara” (INET, CIDR) (Ejemplos: “10.0.0.1”, “10.0.0”, 10.0.0/25”)

inet6	Igual “inet”, encerrando la dirección entre corchetes para evitar confusión con el puerto (Ejemplos: “[::]”, “[::0]”, “[::0]:123”)
packet	link:ID_protocolo_L2:Indice_interfaz (Ejemplo: “src link:*.2”)
unix	unix:cadena (Ejemplo: “src unix:private/”)
netlink	canal:pid (Ejemplo: “src 0:635”)

Ejemplos de filtros:

Sockets en estado “established”	netstat -t state established ss -t state established
Socket con el puerto “22”	ss -t 'sport = 22' ss -t sport \= 22 ss -t sport eq 22 ss -t sport eq ssh
Sockets con puerto inferior al “80”	ss -ltn sport \< 80 ss -ltn sport lt 80 ss -ltn sport lt http
Sockets con puerto menor o igual que el “80”	ss -ltn sport \<= 80 ss -ltn sport le 80 ss -ltn sport le http
Sockets con puerto superior a “1023”	ss -ltn sport \> 1023 ss -ltn sport gt 1023
Procesos locales conectados a nuestro servidor X	ss -x src "/tmp/.X11-unix/*"
Sockets de conexión TCP en estado “FIN-WAIT-1” con puerto origen “80” o “443” para la red “193.233.7/24”	ss -t state fin-wait-1 \ sport = :http or sport = :https \) dst 193.233.7/24 ss -t state fin-wait-1 and \ sport = :http or sport = :https \) and dst 193.233.7/24