
Unidad 9: Configuración de la Red

**VII Curso de Introducción a la Administración de
Servidores GNU/Linux
Extensión Universitaria. Universidad de Sevilla
Mayo 2010**

por Enrique Ostúa

Contenidos

- Acerca del “Network Manager”
- Información para la configuración
- Detección del Hardware
- Configuración IP:
 - interfaz de red
 - nombre del host y anfitrión
 - servicio de nombres
 - otros: dhcp, wireless, modems, ...
- Comprobación de la red
- Introducción a los servicios de red
- Instalando un servidor de DHCP

1. Network Manager

Al instalar ubuntu viene un gestor de red llamado **network-manager**

Este intenta mantener al equipo siempre conectado por cualquiera de los interfaces que tenga a su alcance.

Se ejecuta un applet en el panel superior que nos permite conectar a la red que queremos y cambiar algunas configuraciones.



Pero en servidores recomiendo no usarlo:

```
$ sudo apt-get remove network-manager
```

2. Información para la configuración

Recopilamos esta información:

- dirección IP de la máquina
- máscara de la subred
- dirección IP del gateway (salida de la subred)
- dirección IP del/los servidor(es) de DNS
- nombre y dominio de la máquina

Ejemplo:

Nuestro PC va a tener la IP 150.214.141.122, en una subred de máscara 255.255.255.0, con la puerta de enlace 150.214.141.1. El nombre será 'saturno' en el dominio dte.us.es. Como DNS tenemos 150.214.141.100 y 180.44.1.3 (este último está fuera de la subred).

3. Detección del Hardware

Hardware de red: *tarjetas Ethernet o WiFi*

Para poder configurar la red, primero se deben cargar los drivers del kernel, generalmente en forma de módulos.

En kernels de Linux recientes es un proceso automático, se cargan en la detección durante el proceso de arranque.

Detección del hardware

Mensajes del kernel:

```
$ dmesg      ["# dmesg -c" para borrarlo]
```

```
$ gnome-system-log      (fichero kern.log)
```

Identificación del Hardware:

```
$ lspci [-v]           # lshw
```

El administrador puede de forma manual cargar y descargar los módulos activos en el kernel:

```
$ lsmod
```

```
# insmod, modprobe, rmmod
```

4.1 Configuración IP: Interfaces de red

Cada dispositivo de red se llama **interfaz**.

Cada *ethernet* se numera *eth0*, *eth1*, ...

Los *wireless* depende: *wlanX*, *ocX*, *raX*, ...

Siempre hay un *loopback*, con nombre "*lo*"

Las conexiones vía módem son *ppp0*, *ppp1*...

La información de configuración de los interfaces de red, en Linux basados en Debian (como Ubuntu), se guarda en el directorio **/etc/network**

El script de arranque es `/etc/init.d/networking`

`/etc/network/interfaces (1)`

```
$ cat /etc/network/interfaces
# Auto loading
auto lo eth0 eth2
# Loopback interface
iface lo inet loopback
# Red privada del laboratorio
iface eth0 inet static
    address 10.1.15.121
    netmask 255.255.252.0
    network 10.1.12.0
    broadcast 10.1.15.255
```

/etc/network/interfaces (2)

```
# Subred de profesores (acceso a internet)
iface eth1 inet static
    address 150.214.141.195
    netmask 255.255.255.0
    network 150.214.141.0
    broadcast 150.214.141.255
    gateway 150.214.141.1
    pre-up /usr/local/sbin/chequeoseguridad.sh

iface eth2 inet dhcp
```

Up y down de un interfaz

El proceso de activar o desactivar un interfaz en caliente se denomina *levantar y tirar* el interfaz (*up y down [o shutdown]*).

El administrador puede forzar unos determinados comandos a la hora de levantar o tirar un interfaz de red, colocando scripts en los subdirectorios de /etc/network:

if-pre-up.d/

if-down.d/

if-up.d/

if-post-down.d/

ifup / ifdown / ifconfig

Comandos:

ifup eth1

ifdown ppp0

* Activa (ifup) y desactiva (ifdown) la configuración del interfaz correspondiente, según el fichero */etc/network/interfaces* y los scripts en los subdirectorios *if-pre-up.d*, *if-up.d*, *if-down.d*, *if-post-down.d*

ifconfig [-a]

* Muestra valores de la interfaz de red o configura valores de red en una interfaz de forma manual.

Ej: ifconfig -a, ifconfig eth0, ifconfig eth0 10.1.4.5

4.2 Nombre de host

El nombre del propio equipo se guarda en */etc/hostname*

El dominio se guarda en */etc/hosts*

Comando **hostname**:

\$ hostname -> saturno (nombre)

\$ hostname -d -> dte.us.es (dominio)

\$ hostname -f -> saturno.dte.us.es (completo)

hostname <nuevo_nombre>

-> cambia el nombre (se debe reiniciar)

Anfitriones (“hosts”)

En el fichero `/etc/hosts` se almacenan pares de direcciones IP y nombres de 'equipos locales' (también llamados “anfitriones”).

Se resuelven directamente, sin usar el DNS.

También se describe ahí el *localhost* (127.0.0.1, la IP del *loopback*) y también el dominio local del equipo.

Ejemplo: `$ cat /etc/hosts`

```
127.0.0.1      localhost    localhost.localdomain
150.214.141.122 saturno      saturno.dte.us.es
150.214.141.140 neptuno
```

4.3 Servicio de nombres (DNS)

DNS: Domain Name System

En `/etc/resolv.conf` se configuran los servidores de DNS

También se puede incluir un dominio local o un orden para la búsqueda de dominios.

Ejemplo-1: `$ cat /etc/resolv.conf`

```
domain dte.us.es
nameserver 150.214.186.69
```

Ejemplo-2: `$ cat /etc/resolv.conf`

```
search dte.us.es etsii.us.es
Nameserver 150.214.186.69
```

4.4 Otros: Tabla de Rutas

Con el comando **route** se puede consultar o modificar la tabla de rutas.

Ejemplo: `$ route -n`

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	If
150.214.141.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
10.1.12.0	0.0.0.0	255.255.252.0	U	0	0	0	eth0
default	150.214.141.1	0.0.0.0	UG	0	0	0	eth1

Salvo en redes complejas, los cambios se hacen de forma automática al ejecutar los comandos **ifup** e **ifdown**.

Protocolo DHCP

DHCP: Dynamic Host Configuration Protocol

Si hay un servidor DHCP en nuestra red bastará poner en `/etc/network/interfaces`:

```
iface eth0 inet dhcp
```

Al levantarse el interfaz se hace un broadcast pidiendo al servidor DHCP de la red una IP, el cual responderá con la que se le asigna, y entonces se configura el interfaz.

Ej: al levantar eth0 el servidor DHCP responde que debe configurarse la IP 150.214.141.44, netmask 255.255.255.0, gateway 150.214.141.1 y el DNS 150.214.141.2

“dhclient” solicita la IP al DHCP de la red.

Wireless

Lo fundamental es el paquete *wireless-tools*

Comandos específicos: **iwconfig** (como ifconfig), **iwlist** (lista redes, APs, ...)

```
# iwlist wlan0 scan
# iwconfig wlan0 essid Oficina
# dhclient wlan0
```

Una vez probado, se debe añadir en `/etc/network/interfaces` la configuración:

```
iface wlan0 inet dhcp
    wireless-essid Oficina
    wireless-mode ad-hoc
```

`$ man wireless`

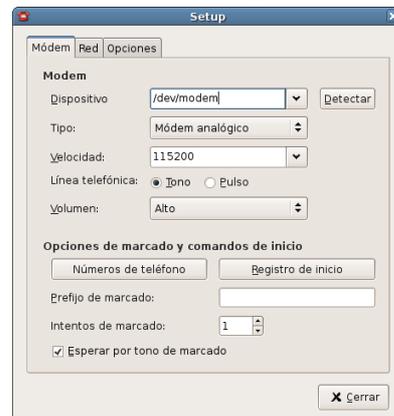
Módem RTC

El protocolo es PPP (*Point to Point Protocol*)

Paquete **pppconfig** / **gnome-ppp** (gráfica)

Comandos: `$ pon` `$ poff` `# pppconfig`

Al conectar se configura la IP por DHCP.



Módem ADSL/Cable

El protocolo suele ser PPPoE (ó PPPoA)
(*PPP Over Ethernet / ATM*)

Paquete **ppoeconf**

Básicamente funciona como un módem RTC, necesita validar usuario y contraseña (propio del PPP) y se asigna IP, DNS, *gateway*...

Comandos: **pppoe-discovery** y **pppoeconfig**

Router ADSL/Cable

El router es el dispositivo de red que se conecta a Internet, vía *PPPoE/PPPoA*, y el resto de la LAN le usa a él como *gateway*.

Así, en nuestro equipo sólo debemos configurar la red correctamente, poniendo el router como *gateway por defecto*.

Generalmente disponen de servidor DHCP por lo que ni siquiera tenemos que preocuparnos de elegir las direcciones de la subred privada.

GUI para Configuración de Red

En *Sistema* → *Administración* → *Red*

(paquete “gnome-network-admin”)



5. Comprobación de la red

Una vez todo configurado, vamos a comprobar que estamos en la red y tenemos conectividad con otros equipos e internet.

Varias comprobaciones:

estado de las interfaces

conectividad y rutado

servicios de nombres

También disponemos de una herramienta gráfica para estas tareas (aunque no viene instalada por defecto)

Estado de las interfaces

```
$ ifconfig -a
```

eth0

```
Link encap:Ethernet HWaddr 00:4F:4E:05:FA:35
inet addr:150.214.141.122 Bcast:150.214.141.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:9373307 errors:1801 dropped:993 overruns:80 frame:0
TX packets:8026804 errors:3583 dropped:0 overruns:0 carrier:7166
collisions:224583 txqueuelen:100
RX bytes:1764525259 (1682.7 Mb) TX bytes:3841778389 (3663.8 Mb)
Interrupt:9 Base address:0x4000
```

lo

```
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1 ...
```

Conectividad entre equipos

La forma más sencilla, enviando un *ping* a otro equipo y esperando la respuesta.

Comprobaremos el *gateway*, *broadcast* y otros equipos de la red y de fuera.

```
$ ping <dirección_IP>|<nombre_maq>
```

Envía paquetes de forma sucesiva a una IP/nombre y mide el tiempo que tarda esta en responder.

Ejemplo: \$ ping 150.214.141.1

```
PING 150.214.141.1 (150.214.141.1): 56 data bytes
64 bytes from 150.214.141.1: icmp_seq=0 ttl=255 time=2.3 ms
64 bytes from 150.214.141.1: icmp_seq=1 ttl=255 time=3.9 ms
64 bytes from 150.214.141.1: icmp_seq=2 ttl=255 time=1.9 ms
....
```

Comprobando el Rutado

Aunque con **ping** podemos comprobar si llegamos a otras redes, **traceroute** nos da mucha información adicional sobre las rutas, sobre todo en redes WAN (o internet).

Ejemplo: `$ traceroute www.microsoft.com`

```
1 150.214.141.2 (gw-141.us.es) 1.051 ms 0.568 ms 0.292 ms
2 193.147.173.174 (193.147.173.174) 0.389 ms 0.304 ms 0.389 ms
3 GE0-1-0.EB-Sevilla0.red.rediris.es (130.206.194.1) 0.359 ms 0.527 ms 0.391 ms
4 * AND.SO4-1-0.EB-IRIS2.red.rediris.es (130.206.240.17) 11.586 ms 11.564 ms
5 213.242.71.145 (213.242.71.145) 11.765 ms 11.895 ms 11.752 ms
6 ae-0-51.mpls1.Madrid1.Level3.net (213.242.70.1) 11.990 ms 12.091 ms 12.265 ms
7 ae-1-0.bbr2.London1.Level3.net (212.187.128.57) 90.061 ms 39.008 ms 39.070 ms
8 ae-0-0.mp1.Seattle1.Level3.net (209.247.9.121) 174.553 ms 173.833 ms 202.173 ms
9 ge-2-0-0-56.gar1.Seattle1.Level3.net (4.68.105.169) 173.826 ms 173.734 ms 173.736 ms
10 65.59.235.6 (65.59.235.6) 194.535 ms 194.134 ms 195.023 ms
11 gig3-1.tuk-76cb-1a.ntwk.msn.net (207.46.42.1) 194.232 ms 194.554 ms 194.230 ms
12 pos1-0.iusxixcpxc1202.ntwk.msn.net (207.46.36.146) 194.353 ms * 195.174 ms
13 pos1-0.tke-12ix-1b.ntwk.msn.net (207.46.155.5) 194.234 ms 194.026 ms 194.881 ms
14 po13.tuk-65ns-mcs-1b.ntwk.msn.net (207.46.224.217) 194.103 ms * 194.731 ms
15 ***
```

Comprobando el DNS

`$ host <nombre|nombre.completo>`

`$ host <nombre.completo> <IP.del.servidor.DNS>`

Realiza una consulta al servidor de DNS para resolver el nombre de host facilitado.

Ejemplo:

```
$ host www.ubuntulinux.com
```

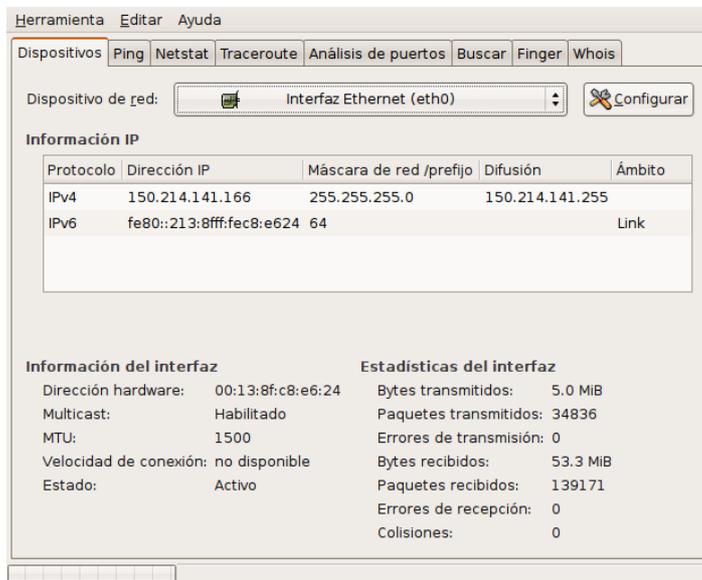
```
www.ubuntulinux.com has address 82.211.81.166
```

`$ host <direccion.IP>`

Resolución inversa de la dirección IP.

Herramienta Gráfica

En Sistema → Admón → Herramientas de Red



Paquete "gnome-nettool"

6. Introducción a los servicios de red

Cada equipo de la red TCP/IP puede ofrecer el acceso a unos servicios al resto de equipos. Cada servicio lleva asociado un puerto.

El servicio se presta a través de un programa servidor específico (denominado de forma genérica "demonio" o *daemon*) o a través de *inetd*, un servidor genérico para lanzar servicios bajo demanda.

Generalmente los distintos *daemons* (incluido *inetd*) se controlan de forma similar.

Control de los servidores (I)

Las configuraciones se guardan en /etc/<servidor> o similar

Ejemplo: El servidor web Apache2 guarda su configuración en /etc/apache2

El script de control es /etc/init.d/<servidor>

start: lanza el servicio

stop: detiene el servicio

restart: lo detiene y vuelve a lanzarlo

reload: recarga la configuración

Ejemplo: # /etc/init.d/apache2 stop

Control de los servidores (y II)

Recientemente se ha unificado el gestor de servicios:

```
# service <servidor> <comando>
```

Ejemplos:

```
# service cron restart
```

```
# service apache2 status
```

```
# service --status-all
```

El servidor INETD

Se controla como un *daemon* más, pero el propio `inetd` facilita el acceso a múltiples servicios, configurados en `/etc/inetd.conf`

Ejemplo: `$ cat /etc/inetd.conf`

```
#:MAIL:
smtp stream tcp nowait mail /usr/sbin/exim exim -bs
#:FTP:
ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd
```

Los servicios estándar están en el fichero `/etc/services`

Hoy día no se usa mucho, generalmente sólo cuando no hay *daemon* propio.

Puertos en servicio

Se pueden listar los puertos TCP/IP dónde hay un *daemon* 'escuchando' (esperando servicio), con ayuda del comando **netstat**

Ejemplo: `# netstat --inet -l -p`

Proto	RQ	SQ	Local	Foreign	State	PID/Program name
tcp	0	0	*:ftp	*:*	LISTEN	6864/pure-ftpd
tcp	0	0	localhost:smtp	*:*	LISTEN	6852/exim3

Para probar los puertos de otros equipos se usará un *análisis de puertos*, como **nmap**

Estas funciones también están en la utilidad gráfica ya vista, 'herramientas de red'

Análisis de puertos

\$ nmap saturno

Starting nmap 3.81 (<http://www.insecure.org/nmap/>) at 2006-01-31 14:49 CET

Interesting ports on saturno.dte.us.es (150.214.141.122):

(The 1650 ports scanned but not shown below are in state: closed)

PORT STATE SERVICE

22/tcp open ssh

25/tcp open smtp

80/tcp open http

81/tcp open hosts2-ns

111/tcp open rpcbind

139/tcp open netbios-ssn

143/tcp open imap

443/tcp open https

444/tcp open snpp

445/tcp open microsoft-ds

929/tcp open unknown

993/tcp open imaps

995/tcp open pop3s

Nmap finished: 1 IP address (1 host up) scanned in 0.205 seconds

netcat

netcat es una herramienta versátil para hacer pruebas con puertos

Permite 'escuchar' en un puerto determinado o bien conectarse a un servicio establecido, interactuando con el usuario.

```
$ netcat -l -p 5555          (escucha en TCP/5555)
$ netcat localhost 5555     (se conecta al puerto 5555)
$ netcat correo 25         (se conecta al servidor SMTP)
220 correo ESMTP Exim 4.50 Thu, 02 Feb 2006 13:38:30 +0100
help
214-Commands supported:
214 AUTH STARTTLS HELO EHLO MAIL RCPT DATA NOOP QUIT RSET HELP
quit
221 correo closing connection
```

6. Instalación Servidor DHCP

Instalar “dhcp3-server” (y “dhcpd”)

Script de inicio: /etc/init.d/dhcp3-server

Configuración: /etc/dhcp3/dhcpd.conf

Los “scopes” son las subredes en las que está el DHCP, y dentro del scope hay que definir los rangos de IP's compartidas (“shared Ips”).

También se puede añadir una configuración particular para algunos equipos, para que el servidor les asigne siempre la misma IP, por ejemplo a partir de la dirección MAC.

Configuración Servidor DHCP

cat /etc/dhcp3/dhcpd.conf

```
option domain-name "dte.us.es";
option domain-name-servers 10.0.1.70 212.59.120.33;
option routers 10.0.1.138;
subnet 10.0.1.0 netmask 255.255.255.0 {      # RED COMPLETA
    interface eth0;
    range 10.0.1.1 10.0.1.10;      # RANGO de IP's a ASIGNAR
}
host saturno {          # CONFIGURACIÓN DE UN EQUIPO CONCRETO
    hardware ethernet 00:9F:F5:0E:5A:30; # DIRECCIÓN MAC
    fixed-address 10.0.1.50;          # DIRECCIÓN IP FIJA
}
```

cat /etc/default/dhcpd3-server

```
INTERFACES="eth0"      # Para que se lance el demonio escuchando en eth0
```

GUI de configuración

Paquete “gadmin-dhcpd”

Information: isc-dhcpd-V3.0.5 Status: Deactivated

Scopes: **Single hosts** | Leases | Verify

Network card	Network address	Subnet mask	Loadbalancing server
eth0	192.168.1.0	255.255.255.0	

Shared IP-address ranges

Range from: [] to: []

Range from: 192.168.1.1 to: 192.168.1.10

Scope settings

Network card: eth0

Network address: 192.168.1.0

Subnet mask: 255.255.255.0

Client settings

Default lease time: 6000

Max lease time: 7200