Networking

Configuración de nuestros interfaces de red

Pablo Garaizar Sagarminaga
txipinet@txipinet.com

Muchas de las utilidades que usamos todos los días, como el correo electrónico o los navegadores web, utilizan protocolos de red para comunicarse. En este artículo veremos cómo configurar esos protocolos desde cero. Comprenderemos todo lo que rodea a una comunicación a través de los interfaces de red y aprenderemos a administrarlos.

Breve repaso a las redes TCP/IP

Antes de afrontar la configuración de red de nuestros equipos, vamos a desempolvar nuestras nociones sobre redes TCP/IP. Siempre que se habla de redes de ordenadores, se habla de protocolos. Un protocolo no es más que un acuerdo entre dos entidades para entenderse. Es decir, si yo le digo a un amigo que le dejo una llamada perdida cuando llegue a su portal, para que baje a abrirme, habremos establecido un protocolo de comunicación, por ejemplo. Pues bien, los ordenadores funcionan de una forma más o menos parecida. Cuando queremos establecer una conexión entre ordenadores mediante una red, hay muchos factores en juego: primero está el medio físico en el que se producirá la conexión (cable, ondas electromagnéticas, etc.), por otro lado está la tecnología que se utilizará (tarjetas de red Ethernet, modems, etc.), por otro los paquetes de datos que enviaremos, las direcciones o destinos dentro de una red... Dado que hay tantos elementos que intervienen en una comunicación, lo que se hace es establecer protocolos o normas para entidades del mismo nivel, es decir, se divide todo lo que interviene en la comunicación en diferentes capas. En el nivel más bajo de todas las capas estaría el nivel físico: el medio que se va a utilizar, los voltajes utilizados, etc. Sobre esa capa se construye la siguiente, en donde transformamos las señales electricas en datos propiamente dichos. Esta sería la capa de enlace de datos. Posteriormente, se van estableciendo una serie de capas intermedias, cada una con mayor refinamiento de la información que maneja, hasta llegar a la capa de aplicación, que es con la que interactuaremos para realizar nuestros trabajos.

Así, por ejemplo, si queremos mandar un correo electrónico a un amigo, utilizaremos la aplicación indicada para mandar un mensaje, en este caso nuestro mailer preferido (mutt, Kmail, mozilla...). El mailer enviará el mensaje al servidor, para que éste lo encamine hasta el buzón de nuestro amigo, pero en ese envío sucederán una serie de pasos que pueden parecernos transparentes en un principio. Primeramente se establece una conexión con el servidor, para ello la aplicación (el mailer) enviará a las capas más bajas de protocolos de red una petición al servidor de correo. Esa capa, aceptará la petición, y realizará otro encargo a una capa inferior, solicitando enviar un paquete de datos por la red. La capa inferior, a su vez, pedirá a la capa física enviar una serie de señales eléctricas por el medio, para hacer su cometido. Tal y como está enfocada la "pila de protocolos de red", cada "trabajo" de red complejo se divide en partes cada vez más sencillas hasta llegar a la capa física, que se encargará de la transmisión eléctrica. Es como si el jefe de una empresa de videojuegos mandara a su subdirector que hiciera un juego de acción. El subdirector iría a donde sus subordinados y les pediría un guión, unos gráficos, un motor de animaciones, etc. Los encargados de los gráficos irían a donde sus subordinados y les pedirían, la portada, los decorados, los personajes, etc. Éstos, a su vez, se repartirían en grupos y cada uno haría un trabajo más

concreto, y así sucesivamente. Es decir, una idea compleja, se divide en trabajos concretos y sencillos para hacerse, estructurándose en capas.

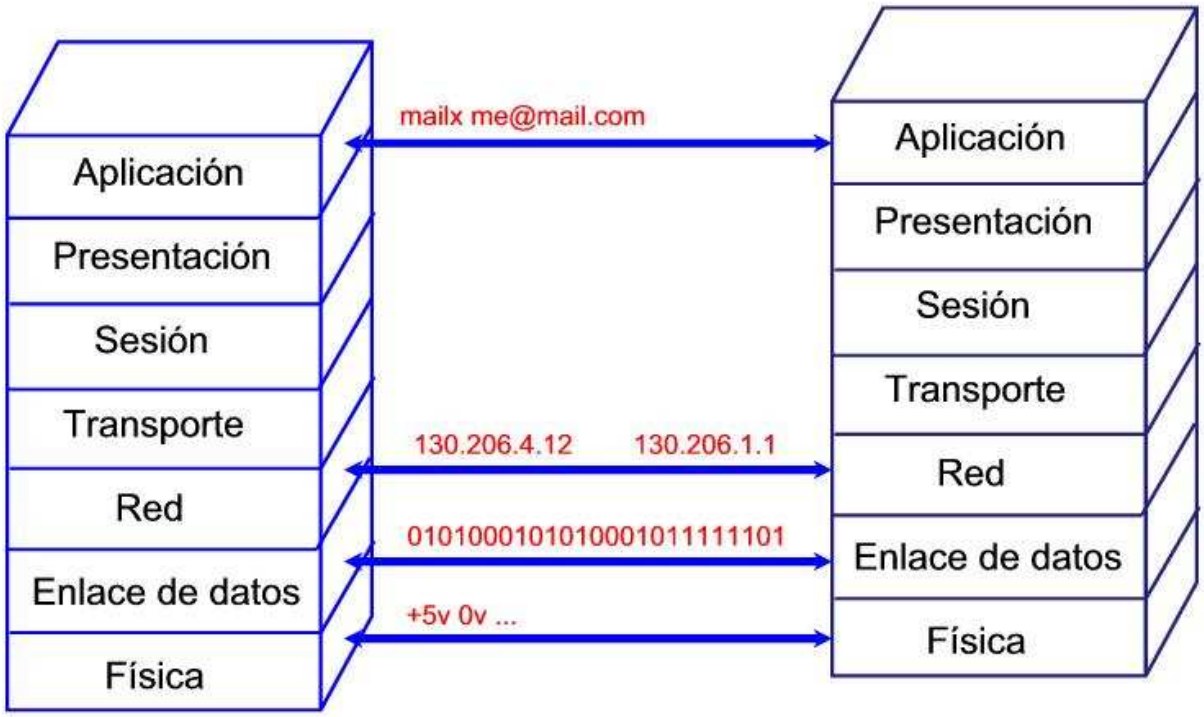


Figura 1. Comunicación mediante capas de protocolos de red.

En el caso específico que nos interesa, la pila de protocolos que utilizamos se denomina TCP/IP, porque dos de sus protocolos principales se llaman TCP (capa de transporte) e IP (capa de red). Cuando utilizamos estos protocolos, cada uno de los posibles destinos de una red necesita un nombre diferente o dirección IP. Al igual que sucede con los teléfonos, para diferenciar todos los ordenadores y dispositivos conectados a una red, se les asigna a cada uno un número diferente, y basta con "marcar" ese número para acceder a él. Actualmente esos números van desde el 0 al 4294967296, pero en lugar de utilizar símplemente el número, se emplea una notación más sencilla, separando el número en 4 dígitos del 0 al 255, por ejemplo: 128.244.34.12 ó 192.168.0.1. En un futuro no muy lejano, las direcciones IP cambiarán su formato, ya que el espacio de direcciones que ofrece la versión actual de IP (IPv4) se está agotando, por lo que habrá que ampliar su rango (al igual que ocurre en ciudades o provincias con mucha demanda de números de teléfono, que amplían la longitud de sus números de teléfono en una o varias cifras).

Siempre que queramos acceder a una red TCP/IP, deberemos tener una dirección IP que nos identifique. Está prohibido viajar sin matrícula por estas carreteras. En nuestras redes privadas, nuestras intranets o pequeñas LANs, la manera de establecer esas direcciones IP la marcamos nosotros mismos (o el administrador de red, en su caso). Es decir, dentro de nuestras organizaciones, somos nosotros los que ponemos los nombres. Esto es lo mismo que lo que sucede en una organización grande, con muchos teléfonos internos y una centralita. El número de extensión de cada teléfono, lo inventamos nosotros mismos, no la compañía telefónica. Cuando queremos salir a una red pública como pueda ser Internet, no podemos inventarnos nuestra dirección IP, deberemos seguir unas normas externas para poder circular por allí. Siguiendo el símil telefónico, si queremos un teléfono accesible por todo el mundo, deberemos solicitar un número válido a la empresa telefónica.

Hasta aquí todo claro: los ordenadores tienen unos números similares a los números de teléfono para identificarse, y cuando queremos comunicarnos con un destino en concreto, sólo tenemos que "marcar" su número, pero... ¿cuándo pedimos una página web a www.linux.org cómo sabe nuestra máquina qué número "marcar"? Buena pregunta, tiene que haber un "listín telefónico" IP, que nos diga que IP corresponde con una dirección específica. Estas "páginas amarillas" de las redes IP se denominan DNS (Domain Name System). Justo antes de hacer la conexión a www.linux.org, nuestro navegador le pregunta la dirección IP al DNS, y luego conecta via dirección IP con el servidor web www.linux.org.

Bueno, si hemos entendido los párrafos anteriores, tenemos unas nociones básicas de lo que son protocolos de comunicaciones, la pila de protocolos TCP/IP, direccionamiento IP, y resolución de nombres o DNS.

¿Cómo funciona esto en Linux?

Tal y como suele ser normal en Linux, el sistema proporciona una interfaz virtual para enmascarar u ocultar las complejidades del hardware de comunicaciones. Los usuarios y aplicaciones utilizan estas abstracciones o interfaces para enviar y recibir datos, despreocupándose de las particularidades físicas.

Existen diferentes interfaces de red, en función de su uso y la tecnología empleada. Veámos las más utilizadas comunmente:

- **lo**: localhost o interfaz de loopback. Se utiliza para referirnos a nosotros mismos. Esto es, cuando queramos establecer una conexión de red a nosotros mismos sólo tenemos que conectar con este interfaz y todo lo demás se realizará normalmente. Por convenio tiene la dirección IP 127.0.0.1.
- **ppp**: peer-to-peer protocol, o protocolo de conexión punto a punto. Cuando hacemos una conexión mediante un modem a nuestro proveedor de Internet, establecemos un enlace punto a punto con él. Éste será el interfaz de red por el que típicamente se transmitirán los datos que deseemos cuando estemos conectándonos a una red por modem.
- **eth0**: interfaz Ethernet. El número 0 indica que es el primer interfaz de red instalado, si tuviéramos varios interfaces de red Ethernet se llamarían eth1, eth2... Cuando nos conectemos a una LAN (Local Area Network) que utilice Ethernet (la mayoría de los casos), nuestras comunicaciones se transmitirán por este interfaz.

Existen otros interfaces de red en nuestros sistemas Linux (dummy, slip...), si bien quedan fuera del alcance de este artículo.

Hace unos años la configuración de ppp en linux era un trabajo bastante costoso. A la inherente complejidad de las primeras versiones de las distribuciones de linux, se unía mi corta experiencia en Internet, por lo que cuando conseguí realizar una simple llamada a mi nodo local de la extinta infovía, sentí la misma satisfacción que si hubiese encontrado vida extraterrestre. Hoy en día esto ha cambiado radicalmente, existen infinidad de asistentes para la configuración de la conexión mediante ppp (kppp, reciéntemente explicado en +Linux) y todo se reduce a introducir nuestros datos de conexión y realizar la llamada. Por ello no vamos a ahondar más en este tema, sólo deciros que wvdial es un asistente muy recomendable si utilizáis este tipo de conexiones.

La configuración de un interfaz de red Ethernet en Linux es algo que todos deberíamos conocer ya que muchos de los sistemas Linux están en un entorno LAN, bien como servidores, bien como estaciones de trabajo. El primer factor que tendremos que tomar en cuenta es el modelo de tarjeta Ethernet que disponemos. Existen muchas distribuciones que nos ahorran conocer esto, pero vamos a ponernos en el peor de los casos: no disponemos de ningún asistente para instalar la tarjeta y no la hemos comprado nosotros mismos. A pesar de que esto pueda parecer un verdadero reto, con un par de trucos podremos salir de este embrollo fácilmente.

Lo primero en lo que deberemos fijarnos es en si es una tarjeta EISA o PCI. Si nuestro sistema Linux es un ordenador medianamente nuevo, casi sin lugar a dudas se tratará de una tarjeta PCI, pero si estamos en un 386, 486 o primeras versiones de Pentium, hay bastantes posibilidades de que sea una EISA. Lo mejor de todo es quitar la tapa del ordenador y mirar a dónde está conectada la tarjeta: si es un conector alargado, de unos 13 cms y típicamente negro, se tratará de un slot EISA; si por el contrario es un conector más pequeño, de unos 8 cms de largo y normalmente blanco, es un slot PCI.

Una vez hecha esta comprobación, deberemos cargar el módulo que dé soporte a nuestra tarjeta. Si se trata de una tarjeta EISA, intentaremos cargar el módulo para tarjetas "NE2000 compatibles" ("NE2000/NE1000 support", tened cuidado de no elegir NE2000 pci). En el caso de ser una tarjeta PCI, lo normal es que dispongan del chip 8029 si son de 10 Mbps o el 8139 si son de 100 Mbps.

Para probar el módulo y ver si el sistema reconoce la tarjeta haremos uso del comando "modprobe". Con él probaremos si el módulo es adecuado. Veámos cómo sería la sintáxis para cargar el módulo (desde una shell de root) necesario para una tarjeta pci con el chip 8139 (100 Mbps):

```
# modprobe rtl8139
8139too Fast Ethernet driver 0.9.18a
PCI: Found IRQ 11 for device 00:0c.0
PCI: Sharing IRQ 11 with 01:00.0
eth0: RealTek RTL8139 Fast Ethernet at 0xda801000, 00:c0:26:a0:7f:fc, IRQ 11
eth0: Identified 8139 chip type 'RTL-8139B'
```

De esta manera hemos cargado el módulo para la tarjeta PCI y el sistema nos ha devuelto un mensaje confirmando que ha encontrado la tarjeta en la IRQ 11. Si el sistema nos indica algún error, deberemos asegurarnos de que ese es el módulo adecuado para nuestra tarjeta o probar otro similar. Para una tarjeta a 10 Mbps, en lugar de cargar ese módulo, cargaríamos normalmente el módulo "ne2k-pci".

Vamos a ver ahora un ejemplo para tarjetas EISA. En principio el proceso es muy similar, si bien habitualmente hay que especificar a mano la IRQ y la IO Address que utilizará la tarjeta. Para la gran mayoría de las tarjetas de red EISA, podremos utilizar los valores io=0x300 y irq=3, ó io=0x340 y irq=5 (consultad el manual de la tarjeta si podéis). Veámos cómo sería:

```
# modprobe ne io=0x300 irq=3
NE*000 ethercard probe at 0x300: 00 c0 26 80 15 93
eth0: NE2000 found at 0x300, using IRQ 3.
```

Ya tenemos instalado el módulo de nuestra tarjeta, vamos a asegurarnos que lo hemos cargado correctamente. Para ello utilizamos el comando "lsmod" que muestra un listado de los módulos cargados:

```
# lsmod
Module                  Size  Used by
serial                 19564   1  (autoclean)
ne                      6272   1
8390                    6040   0  [ne]
unix                   10212  89  (autoclean)
```

En esta lista podemos observar como el módulo "ne" ha sido cargado correctamente.

Una vez aquí, podemos olvidarnos de todas las complicaciones a nivel de hardware. Centrémonos ahora en la configuración de nuestra dirección IP, máscaras de red, servidores DNS, etc. Para proporcionar una dirección IP a nuestro interfaz de red (eth0), utilizaremos el comando "ifconfig":

```
# ifconfig eth0 192.168.0.3
```

Así habremos asignado al interfaz eth0, la IP 192.168.0.3. Si no definimos la máscara de red, ifconfig le asignará la que es propia a esa familia de direcciones IP (en este caso 255.255.255.0, por ser 192.168.0.3 de clase C). Si nuestro administrador de redes ha establecido subredes y debemos usar una máscara distinta a la estándar, podremos indicarlo en el comando ifconfig, por ejemplo:

```
# ifconfig eth0 192.168.0.3 netmask 255.255.244.0
```

Nuestro interfaz de red ya tiene IP y podrá acceder a todos los equipos de su misma red, pero si quiere acceder a equipos de redes diferentes, necesitará establecer una dirección de pasarela (gateway) que nos comunique con otras redes. Para fijar nuestra dirección de pasarela, haremos uso del comando "route":

```
# route add default gw 192.168.0.1
```

Es decir, añadimos una ruta diciendo que el gateway o pasarela por defecto será la IP 192.168.0.1. Todas las peticiones a IPs que no estén en nuestra red, las haremos a través de esa pasarela.

Después de configurar nuestra IP y la pasarela por defecto, estaremos en condiciones de llegar a cualquier IP a la que nos de acceso dicha pasarela. En este caso, los ejemplos de este artículo me han servido para configurar uno de mis ordenadores de mi pequeña LAN doméstica. A todos les he asignado una IP inventada por mí (en este caso 192.168.0.*), y mi router ADSL tiene como IP interna 192.168.0.1. Así,

toda petición que no sea a otro ordenador de mi casa, se pedirá al gateway por defecto (192.168.0.1), y él (el router ADSL) lo encaminará hacia fuera (Internet). Para probar si llegamos desde nuestra máquina a otra máquina a través de la red, se suele utilizar el comando "ping":

```
# ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2): 56 data bytes
64 bytes from 192.168.0.2: icmp_seq=0 ttl=255 time=0.7 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=255 time=0.4 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=255 time=0.4 ms

--- 192.168.0.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.5/0.7 ms
```

Este comando lanzará un paquete ICMP de 64 bytes contínuamente al destino indicado, y verá si el destino nos responde, indicando el tiempo transcurrido (time=0.7 ms). Si no lo detenemos con Control+C, estará indefinidamente lanzando paquetes ICMP y esperando su respuesta.

Lo único que nos falta para terminar de configurar correctamente nuestra conexión Ethernet es establecer quién será nuestro servidor de nombres DNS. Para ello es necesario editar dos ficheros:

- /etc/host.conf
- /etc/resolv.conf

El primero de ellos indica cuál será la fuente de las correspondencias dirección URL <-> dirección IP, ya que esta traducción puede hacerse preguntando a un servidor DNS o utilizando un fichero (/etc/hosts) que tenga esas correspondencias. Normalmente lo que se hace es utilizar ambas cosas, servidor y fichero, y poner en el fichero sólo las direcciones locales que no se preguntan al DNS. En el fichero /etc/resolv.conf indicaremos las direcciones IP de nuestros servidores DNS. Veámos unos ejemplos muy simples de lo que podría ir en cada uno de estos ficheros:

```
# cat /etc/hosts
127.0.0.1       localhost
# cat /etc/host.conf
order hosts, bind
# cat /etc/resolv.conf
nameserver 194.179.1.100
nameserver 194.179.1.101
```

En el primer ficheros (/etc/hosts) únicamente tenemos la correspondencia entre nombre y dirección IP para 127.0.0.1, es decir, localhost. El resto se preguntará al servidor o servidores DNS. En el segundo fichero (/etc/host.conf) indicamos cuál será el órden para preguntar acerca de direcciones IP. En nuestro caso primero miraremos el fichero hosts (/etc/hosts) y posteriormente preguntaremos a los servidores DNS (bind). En el tercer fichero indicamos qué servidores DNS utilizaremos. El órden influye, por lo que 194.179.1.100 será nuestro servidor DNS primario, y 194.179.1.101 el secundario.

Ya está, si todo ha ido bien, nuestro navegador encontrará las páginas que solicitemos, podremos conectarnos a servidores ftp, pop3, etc. Para comprobarlo, vamos a ver si llegamos con un ping a www.linux.org, por ejemplo:

```
# ping www.linux.org
PING www.linux.org (198.182.196.56): 56 data bytes
64 bytes from 198.182.196.56: icmp_seq=0 ttl=234 time=261.1 ms
64 bytes from 198.182.196.56: icmp_seq=1 ttl=234 time=257.9 ms
64 bytes from 198.182.196.56: icmp_seq=2 ttl=234 time=255.0 ms
64 bytes from 198.182.196.56: icmp_seq=3 ttl=234 time=258.5 ms
64 bytes from 198.182.196.56: icmp_seq=4 ttl=234 time=257.1 ms

--- www.linux.org ping statistics ---
6 packets transmitted, 5 packets received, 16% packet loss
round-trip min/avg/max = 255.0/257.9/261.1 ms
```

C'est voilà! Pero... quizá alguien se esté preguntando si es necesario todo esto cada vez que queramos configurar nuestro acceso a la red. Obviamente la respuesta a esa pregunta es NO. Veámos que alternativas tenemos para configurar esto una sola vez y no preocuparnos de ello después:

- editar los ficheros de sistema necesarios (por ejemplo, en debian: /etc/network/interfaces).

- utilizar alguna herramienta de configuración de todo el sistema con linuxconf.
- utilizar algún asistente para la configuración de Ethernet

En función de nuestro sistema, nos convendrá utilizar un método u otro. Si nuestra máquina Linux es un servidor con pocas cosas instaladas, lo mejor será editar a mano los ficheros correspondientes, para evitar instalar asistentes y librerías innecesarias. Si por el contrario estamos ante una estación de trabajo Linux con KDE, un entorno gráfico y amigable y nos sentimos cómodos así, podemos evitarnos trastear con ficheros de texto y utilizar los asistentes para la configuración.

Antes de cualquiera de las anteriores opciones, algo estrictamente necesario es instalar de forma permanente el módulo que hemos cargado previamente para instalar nuestra tarjeta Ethernet. Con modprobe lo instalamos sólo como prueba y la próxima vez que arranquemos el sistema no se cargará esa configuración. Para evitar esto, deberemos solicitar que se cargue el módulo durante el proceso de arranque. Una vez más tenemos las dos opciones típicas:

- modificar los ficheros de configuración.
- utilizar un asistente para configurar los módulos.

Si optamos por la primera opción, deberemos modificar el fichero /etc/modules.conf. Es bastante sencillo, pero por comodidad yo prefiero utilizar "modconf", un asistente para seleccionar los módulos que se cargarán con el arranque del sistema. Al ejecutar modconf, se nos muestran los posibles módulos que podemos cargar ordenados por categorías basta con indicar cuál queremos que se cargue o se descargue y salir, para que el propio asistente modifique convenientemente el fichero /etc/modules.conf.

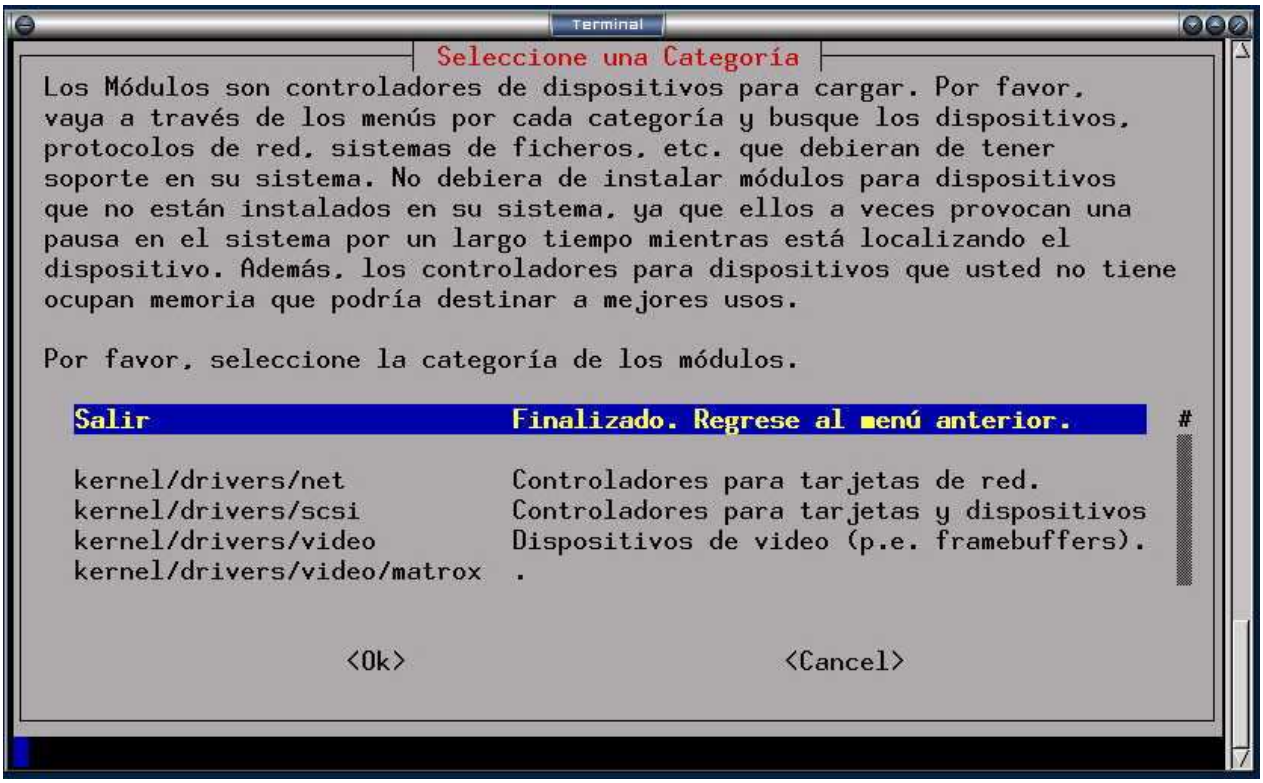


Figura 2. modconf, un asistente para la gestión de los módulos del kernel.

Después de asegurarnos que el módulo correcto correspondiente a nuestra tarjeta de red se cargará durante el arranque del sistema, vamos a editar los ficheros necesarios para que todo esté configurado desde el arranque. En función de la distribución que utilicemos habrá que modificar unos ficheros u otros. Normalmente todos están en /etc/ y tienen nombres como networks, networking, etc. En el caso concreto de debian, por ejemplo, todo está almacenado en /etc/network/interfaces:

```
argon:~ cat /etc/network/interfaces
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
# automatically added when upgrading
auto lo
iface lo inet loopback

# The internet interface
# automatically added when upgrading
auto eth0
iface eth0 inet static
        address 192.168.0.3
        network 192.168.0.0
        netmask 255.255.255.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
```

Su contenido es bastante sencillo de explicar: configuramos dos interfaces, lo y eth0. El interfaz de

loopback (lo) no necesita ningún tipo de parametrización, mientras que el interfaz de red (eth0) necesita una serie de parámetros, como su dirección IP (address), red a la que pertenece (network), máscara de subred (netmask), dirección de broadcast (broadcast) y puerta de enlace (gateway). Si todo esto está bien configurado podremos utilizar los scripts "ifup" y "ifdown" para habilitar o deshabilitar un interfaz en concreto:

```
argon:~# ifdown eth0
argon:~# ifup eth0
eth0: Setting half-duplex based on auto-negotiated partner ability 0000.
```

Esto puede parecer un poco confuso, por lo que quizá prefiramos utilizar un asistente algo más elaborado. Si nuestro sistema tiene Linuxconf instalado, podremos configurar los interfaces de red en su apartado "Networking", tal y como muestra la figura:

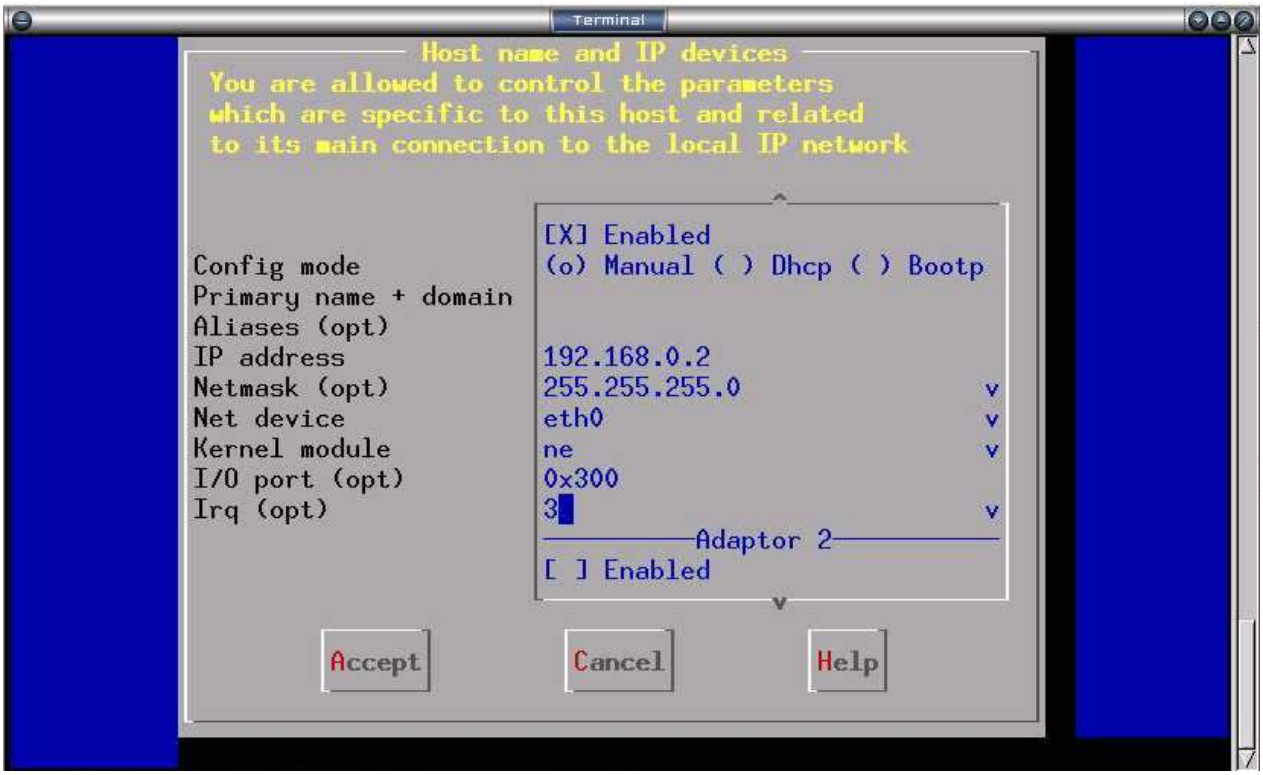


Figura 3. Linuxconf permite configurar de forma sencilla los interfaces de red.

Puede darse el caso de que nuestro sistema no funcione bajo la supervisión de un asistente de configuración centralizado como pueda ser Linuxconf, y quizá prefiramos utilizar asistentes específicos para la configuración de red. Con ello nos evitaremos editar los ficheros del sistema o perdernos entre las configuraciones de todos los elementos de nuestra máquina Linux.

Un ejemplo muy sencillo de lo que podría ser un asistente de este tipo es "etherconf", que mediante unas simples preguntas nos ayuda a configurar nuestros interfaces de red:

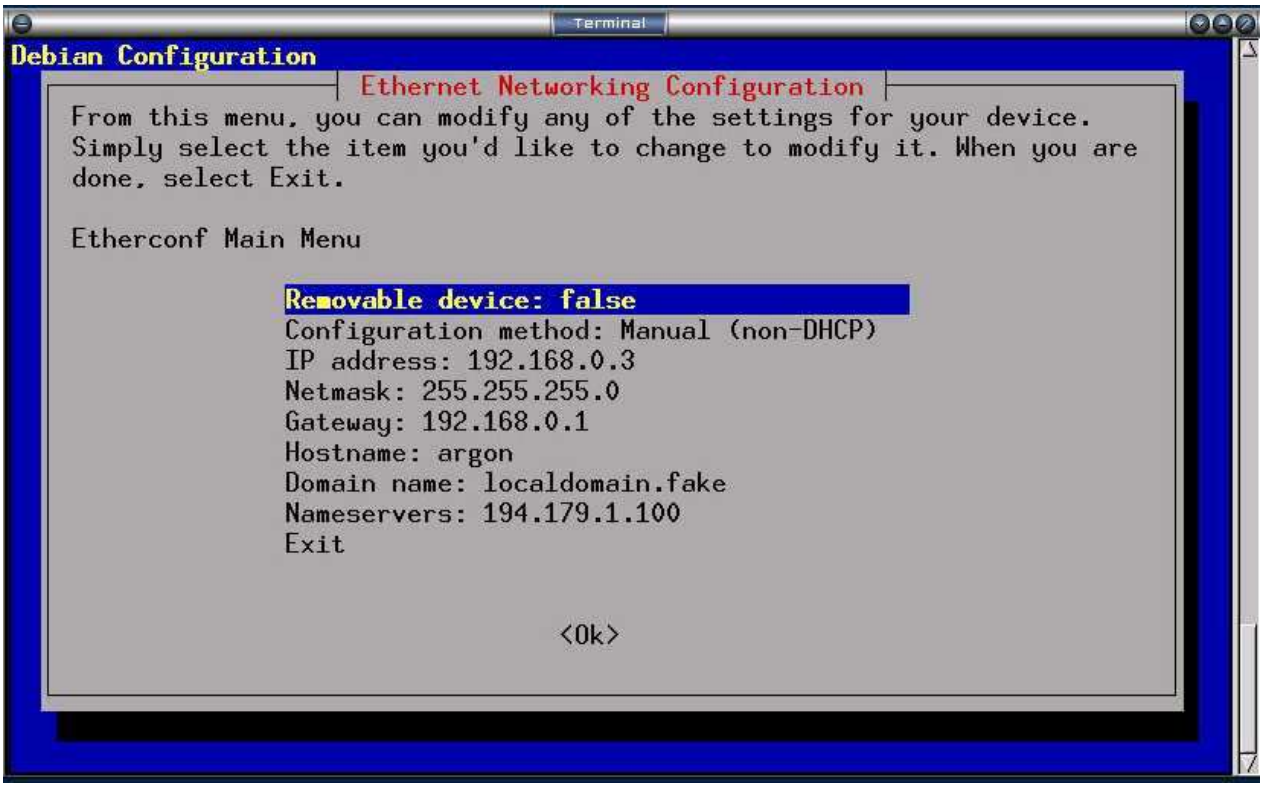
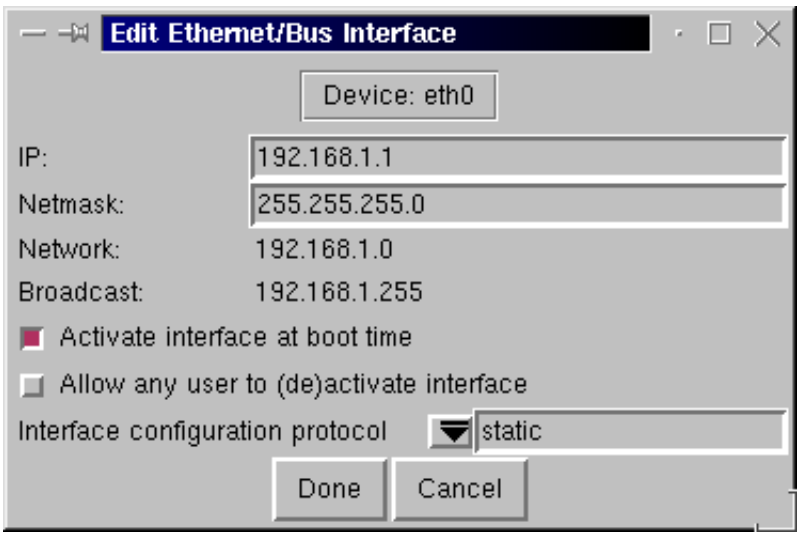


Figura 4. etherconf, un asistente para la configuración de interfaces de red.

Existen otras opciones mucho más atractivas para los usuarios de entornos gráficos como "netcfg", o los asistentes propios de nuestro gestor de ventanas:



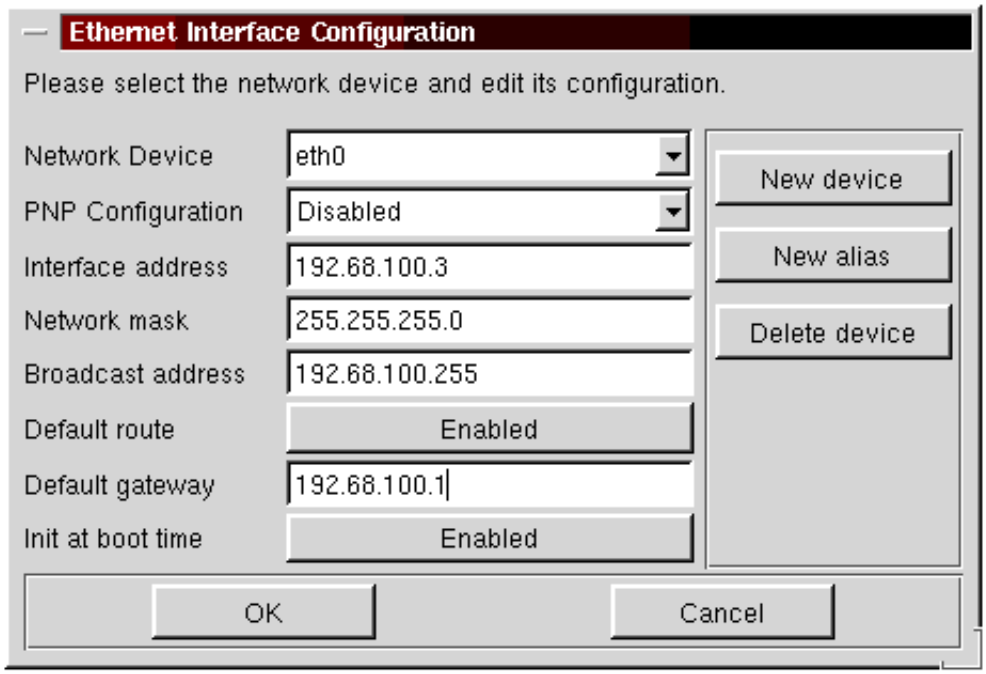


Figura 5. Diferentes opciones para la configuración de los interfaces de red mediante asistentes.

Conclusión

A lo largo de este texto hemos ido viendo los pasos necesarios para configurar correctamente los interfaces de red dentro de nuestros sistemas Linux y hemos ahondado en todas las posibles soluciones que pueden darse a este respecto. No he pretendido ser exhaustivo en cuanto a detalles y a pesar de que todos los comandos han sido explicados de forma somera, siempre es conveniente echar un vistazo a las páginas de manual de cada uno de ellos (man ifconfig, man route...). Espero que os haya gustado y os saque de más de un apuro en vuestras andanzas como administradores de red ;-)

Para saber más...

- Ethernet HOWTO: http://www.linuxdoc.org/HOWTO/Ethernet-HOWTO.html
- Ethernet home: http://wwwhost.ots.utexas.edu/ethernet/ethernet-home.html
- Configuración en SuSE: http://www.linuxheadquarters.com/howto/networking/yast_ethernet.shtml
- Configuración con netcfg en RedHat: http://www.linuxheadquarters.com/howto/networking/netcfg_ethernet.shtml
- Configuración con Linuxconf en RedHat: http://www.linuxheadquarters.com/howto/networking/linuxconf_ethernet.shtml

---

Este documento ha sido escrito por un miembro de e-GHOST, y su contenido es libre de ser reproducido en otros medios bajo las condiciones de la Licencia FDL (GNU Free Documentation License).