

Redes

Netconfig

Las computadoras no son muy interesantes por sí mismas. Claro, puedes instalar juegos en ellos, pero eso los convierte en consolas de entretenimiento glorificadas. Hoy en día, las computadoras necesitan poder hablar entre sí; necesitan estar en red. Ya sea que esté instalando una red de negocios con cientos o miles de computadoras o simplemente esté configurando una sola PC para el acceso a Internet, Slackware es simple y fácil. Este capítulo debería enseñarle cómo configurar redes cableadas típicas. La configuración inalámbrica común se tratará en detalle en la siguiente sección, pero gran parte de lo que lea aquí también será aplicable allí.

Existen muchas formas diferentes de configurar su computadora para conectarse a una red o a Internet, pero se dividen en dos categorías principales: estática y dinámica. Las direcciones estáticas son sólidas; se establecen con el entendimiento de que no se cambiarán, al menos no en el corto plazo. Las direcciones dinámicas son fluidas; el supuesto es que la dirección cambiará en algún momento en el futuro. Normalmente, cualquier tipo de servidor de red requiere una dirección estática simplemente para que otras máquinas sepan dónde contactarla cuando necesiten servicios. Las direcciones dinámicas tienden a usarse para estaciones de trabajo, clientes de Internet y cualquier máquina que no requiera una dirección estática por ningún motivo. Las direcciones dinámicas son más flexibles, pero presentan complicaciones propias.

Puede encontrar muchos tipos diferentes de protocolos de red, pero la mayoría de las personas solo tendrán que lidiar con el Protocolo de Internet (IP). Por esa razón, nos centraremos exclusivamente en IP en este libro.

Configuración manual

Bien, entonces instaló Slackware, configuró un escritorio, pero no puede conectarse a Internet o a la LAN de su empresa (red de área local), ¿qué hace? Afortunadamente, la respuesta a esa pregunta es simple. Slackware incluye una serie de herramientas para configurar su conexión de red. Lo primero que veremos es el muy poderoso **ifconfig** (8), que se usa para configurar o modificar la configuración del hardware más común para la conexión a redes: una tarjeta de interfaz de red (NIC o Ethernet). **ifconfig** es una herramienta increíblemente poderosa capaz de hacer mucho más que configurar direcciones IP. Para una introducción completa, debe leer su página de manual. Por ahora, solo lo usaremos para mostrar y cambiar las direcciones de red de algunos controladores Ethernet.

```
darkstar:~# ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:699 errors:0 dropped:0 overruns:0 frame:0
            TX packets:699 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:39518 (38.5 KiB)  TX bytes:39518 (38.5 KiB)
```

```
wlan0      Link encap:Ethernet  HWaddr 00:1c:b3:ba:ad:4c
            inet addr:192.168.1.198  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::21c:b3ff:feba:ad4c/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:1630677 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1183224 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1627370207 (1.5 GiB)  TX bytes:163308463 (155.7 MiB)

wmaster0   Link encap:UNSPEC  HWaddr 00-1C-B3-BA-
AD-4C-00-00-00-00-00-00-00-00-00-00-00-00-00-00
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Como puede ver claramente aquí, cuando se ejecuta sin ningún argumento, **ifconfig** mostrará toda la información que tiene en todas las tarjetas Ethernet (y tarjetas Ethernet inalámbricas) presentes en su sistema. Lo anterior representa una conexión inalámbrica típica desde mi computadora portátil, así que no tenga miedo si lo que ve en su sistema no coincide. Sin embargo, si no ve ninguna interfaz ethX o wlanX, es posible que la interfaz esté inactiva. Para mostrar todas las NIC presentes actualmente si son “up” o “down”, simplemente pase el argumento -a.

```
darkstar:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:19:e3:45:90:44
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:122780 errors:0 dropped:0 overruns:0 frame:0
            TX packets:124347 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:60495452 (57.6 MiB)  TX bytes:17185220 (16.3 MiB)
            Interrupt:16

lo        Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:699 errors:0 dropped:0 overruns:0 frame:0
            TX packets:699 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:39518 (38.5 KiB)  TX bytes:39518 (38.5 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:1c:b3:ba:ad:4c
            inet addr:192.168.1.198  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::21c:b3ff:feba:ad4c/4 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:1630677 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1183224 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1627370207 (1.5 GiB)  TX bytes:163308463 (155.7 MiB)
```

```
wmaster0 Link encap:UNSPEC HWaddr 00-1C-B3-BA-
AD-4C-00-00-00-00-00-00-00-00-00-00
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Observe que la interfaz eth0 ahora se encuentra entre las devoluciones. **ifconfig** también puede cambiar la configuración actual en una NIC. Por lo general, necesitaría cambiar la dirección IP y la máscara de subred, pero puede cambiar prácticamente cualquier parámetro.

```
darkstar:~# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
darkstar:~# ifconfig eth0
eth0      Link encap:Ethernet HWaddr 00:19:e3:45:90:44
          inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:122780 errors:0 dropped:0 overruns:0 frame:0
          TX packets:124347 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:60495452 (57.6 MiB) TX bytes:17185220 (16.3 MiB)
          Interrupt:16
```

Si observa detenidamente, notará que la interfaz ahora tiene la dirección IP 192.168.1.1 y una máscara de subred 255.255.255.0. Ahora hemos configurado los conceptos básicos para la conexión a nuestra red, pero aún necesitamos configurar una puerta de enlace predeterminada y nuestros servidores DNS. Para hacer eso, necesitaremos ver algunas herramientas más.

A continuación, en nuestra parada a través de la red, está el igualmente poderoso **route** (8). Esta herramienta es responsable de modificar la tabla de enrutamiento del kernel de Linux que afecta a toda la transmisión de datos en una red. Las tablas de enrutamiento pueden volverse inmensamente complejas o pueden ser sencillas y directas. La mayoría de los usuarios solo necesitarán configurar una puerta de enlace predeterminada, por lo que le mostraremos cómo hacerlo aquí. Si, por algún motivo, necesita una tabla de enrutamiento más compleja, le recomendamos que lea la página de manual completa para **route**, así como que consulte otras fuentes. Por ahora, echemos un vistazo a nuestra tabla de enrutamiento inmediatamente después de configurar eth0.

```
darkstar:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use
Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
loopback * 255.0.0.0 U 0 0 0 lo
```

No explicaré todo aquí, pero la información general debería ser fácil de recopilar si está familiarizado con las redes. Los campos Destino y Genmask especifican un rango de direcciones IP para que coincidan. Si se define una puerta de enlace, la información en forma de paquetes se enviará a ese host para su reenvío. También especificamos una interfaz en el campo final que la información debe atravesar. En este momento, solo podemos comunicarnos con computadoras con direcciones entre 192.168.1.0 y 192.168.1.255 y nosotros mismos a través de la interfaz de bucle de retorno, un tipo de NIC virtual que se usa para enrutar información desde esta computadora hacia sí misma. Para poder

llegar al resto del mundo, necesitaremos configurar una puerta de enlace predeterminada.

```
darkstar:~# route add default gw 192.168.1.254
darkstar:~# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use
Iface
192.168.1.0      *               255.255.255.0    U        0      0      0 eth0
loopback         *               255.0.0.0        U        0      0      0 lo
default          192.168.1.254   0.0.0.0          UG       0      0      0 eth0
```

Inmediatamente debe notar la adición de una ruta predeterminada. Esto especifica qué enrutador se debe usar para alcanzar cualquier dirección que no esté especificada en otra parte de nuestra tabla de enrutamiento. Ahora, cuando intentamos conectarnos, por ejemplo, 64.57.102.34, la información se enviará a 192.168.1.254, que es responsable de entregar los datos por nosotros.

Desafortunadamente, todavía no hemos terminado. Necesitamos alguna forma de convertir nombres de dominio como slackware.com en direcciones IP que la computadora pueda usar. Para eso, necesitamos hacer uso de un servidor DNS.

Afortunadamente, configurar su computadora para usar un servidor DNS externo (o incluso interno) es muy fácil. Deberá usar su editor de texto favorito y abrir el archivo `/etc/resolv.conf`. No me pregunte qué pasó con la **e**. En mi computadora, `resolv.conf` se ve así.

```
# /etc/resolv.conf
search lizella.net
nameserver 192.168.1.254
```

Muchos usuarios no necesitarán la línea de búsqueda. Esto se utiliza para asignar nombres de host a nombres de dominio. Básicamente, si intento conectarme a “*barnowl*”, la computadora sabe que debe buscar “*barnowl.lizella.net*” gracias a esta línea de búsqueda. Estamos interesados principalmente en la línea del servidor de nombres. Esto le dice a Slackware a qué servidores de nombres de dominio (DNS) se deben conectar. En términos generales, estos siempre deben especificarse por dirección IP. Si sabe qué servidores DNS debe usar, puede agregarlos uno por uno a las líneas de servidores de nombres individuales. De hecho, no conozco ningún límite práctico para la cantidad de servidores de nombres que se pueden especificar en `resolv.conf`, así que agregue tantos como desee. Una vez hecho esto, debe poder comunicarse con otros hosts a través de su nombre de dominio completo.

Pero Alan! ¡Eso es mucho trabajo duro! No quiero hacer esto una y otra vez por docenas o incluso cientos de máquinas. Tiene toda la razón, y es por eso que personas más inteligentes que usted y yo crearon DHCP. DHCP significa Dynamic Host Control Protocol y es un método para configurar automáticamente computadoras con direcciones IP únicas, máscaras de red, puertas de enlace y servidores DNS. La mayoría de las veces, querrá usar DHCP. La mayoría de los enrutadores inalámbricos, DSL o módems de cable, incluso los firewalls, tienen servidores DHCP para que su vida sea mucho más fácil. Slackware incluye dos herramientas principales para conectarse a un servidor DHCP existente e incluso puede actuar como un servidor DHCP para otras computadoras. Por ahora, sin embargo, solo vamos a ver a los clientes DHCP.

Primero en nuestra lista es ***dhcpcd*** (8), parte de las utilidades DHCP de ISC. Suponiendo que su computadora está físicamente conectada a su red, y que tiene un servidor DHCP operativo en esa red, puede configurar su NIC de una sola vez.

```
darkstar:~/ dhcpd eth0
```

Si todo salió según lo planeado, su NIC debería estar correctamente configurada, y debería poder comunicarse con otras computadoras de su red y con Internet en general. Si por alguna razón, **dhcpd** falla, puede intentar **dhclient** (8). **dhclient** es una alternativa a **dhcpd** y funciona básicamente de la misma manera.

```
darkstar:~/ dhclient eth0
Listening on LPF/eth0/00:1c:b3:ba:ad:4c
Sending on   LPF/eth0/00:1c:b3:ba:ad:4c
Sending on   Socket/fallback
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.254
bound to 192.168.1.198 -- renewal in 8547 seconds.
```

¿Por qué Slackware incluye dos clientes DHCP? A veces, un servidor DHCP en particular puede estar roto y no responder bien a **dhcpd** o **dhclient**. En esos casos, puede recurrir al otro cliente DHCP con la esperanza de obtener una respuesta válida del servidor. Tradicionalmente, Slackware utiliza **dhcpd**, y esto funciona en la mayoría de los casos, pero puede ser necesario en algún momento que usted use **dhclient** en su lugar. Ambos son excelentes clientes DHCP, así que use el que prefiera.

Configuración automática con rc.inet1.conf

La configuración manual de las interfaces es una habilidad importante para tener, pero puede volverse tediosa. Nadie quiere configurar manualmente su conexión a Internet cada vez que se inicia el sistema. Más importante aún, es posible que no siempre tenga acceso físico a la máquina cuando arranca. Slackware facilita la configuración automática de las tarjetas Ethernet (e inalámbricas) al iniciar el sistema con `/etc/rc.d/rc.inet1.conf`. Por ahora, nos centraremos en las redes Ethernet tradicionales por cable; el siguiente capítulo discutirá varias opciones inalámbricas.

`rc.inet1.conf` es un archivo de configuración increíblemente poderoso, capaz de configurar la mayoría de sus tarjetas de red automáticamente cuando se inicia Slackware. El archivo está lleno de comentarios útiles, pero también hay una página de manual que analiza más a fondo su uso. Para comenzar, vamos a ver algunas de las opciones utilizadas en una de mis máquinas personales.

```
# Config information for eth0:
IPADDR[0]="192.168.1.250"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]=" "
DHCP_HOSTNAME[0]=" "
# Some lines omitted.
GATEWAY="192.168.1.254"
```

Esto representa la mayor parte de la información necesaria para configurar una dirección IP estática en un solo controlador Ethernet. **netconfig** usualmente completará estos valores para un solo dispositivo Ethernet para usted. Si tiene varias tarjetas de red en su máquina y necesita que todas se activen automáticamente en el momento del arranque, entonces deberá editar o agregar entradas adicionales a este archivo de la misma manera que antes. Primero, déjeme repasar algunos de los conceptos básicos.

Como ya habrá adivinado, IPADDR [n] es la dirección de protocolo de Internet para la tarjeta de interfaz de red **n**. Normalmente, **n** se corresponde para *eth0*, *eth1*, y así sucesivamente, pero no siempre es así. Puede especificar estos valores para pertenecer a un controlador de red diferente con la variable IFNAME [n], pero lo reservaremos para [redes inalámbricas](#), ya que se refiere más comúnmente a los controladores de red inalámbrica.

Del mismo modo, NETMASK [n] es la máscara de subred que se utiliza para el controlador de red. Si estas líneas se dejan vacías, las direcciones IP estáticas no se asignarán automáticamente a este controlador de red. La variable **USE_DHCP [n]** le dice a Slackware (naturalmente) que use DHCP para configurar la interfaz. **DHCP_HOSTNAME [n]** rara vez se usa, pero algunos servidores DHCP pueden requerirlo. En ese caso, se debe establecer en un nombre de host válido. Finalmente, llegamos a la variable GATEWAY. En realidad, se establece más bajo en el archivo de lo que aparece en mi ejemplo y controla la puerta de enlace predeterminada que se usará. Quizás se esté preguntando por qué no hay una variable **GATEWAY [n]**. La respuesta a eso está en cómo funciona el Protocolo de Internet. No voy a entrar en una discusión en profundidad sobre ese tema, pero basta con decir que solo hay una ruta predeterminada que una computadora puede usar, sin importar cuántas interfaces estén conectadas a ella.

Si necesita utilizar direcciones IP estáticas, deberá obtener una dirección IP estática única y la máscara de subred para la interfaz, así como la dirección de la puerta de enlace predeterminada, e ingresarlas aquí. No hay lugar para ingresar información de DNS en *rc.inet1.conf*, por lo que los servidores DNS deberán colocarse manualmente en *resolv.conf* como se explica en [Configuración manual](#). Por supuesto, si usa **netconfig**, el programa lo manejará por usted. Ahora echemos un vistazo a otra interfaz en mi computadora.

```
# Config information for eth1:
IPADDR[1]=" "
NETMASK[1]=" "
USE_DHCP[1]="yes"
DHCP_HOSTNAME[1]=" "
```

Aquí le digo a Slackware que configure eth1 utilizando DHCP. No necesito configurar las variables **IPADDR [1]** o **NETMASK [1]** cuando se usa DHCP (de hecho, si están configuradas, se ignorarán). Slackware se pondrá en contacto con un servidor DHCP tan pronto como la máquina comience a arrancar.

Navegación de capítulos

Capítulo anterior: [Emacs](#)

Capítulo siguiente: [Redes inalámbricas](#)

Fuentes

- Fuente original: <http://www.slackbook.org/beta>

- Escrito originalmente por Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traducido por: [Victor](#) 2019/02/03 12:20 (UTC)

[slackbook](#), [networking](#), [netconfig](#), [dhcpcd](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

<https://docs.slackware.com/es:slackbook:network>

Last update: **2019/03/02 22:20 (UTC)**

