
Introducción a las redes TCP/IP en Linux

Diseño y Administración de Sistemas y Redes

Juan Céspedes <cespedes@gsync.es>



Curso 2005–2006



Subsistema de red

- Los subsistemas más importantes del kernel de *Linux* son: gestión de procesos, gestión de memoria, sistemas de ficheros, dispositivos, y el **subsistema de red**.
- Existen varias llamadas al sistema específicas para realizar conexiones y comunicaciones usando la red, como son `socket()`, `connect()`, `send()`, `recv()`, etc.
- Para que funcione el subsistema de red, es necesario que se haya configurado previamente, para que el sistema sea capaz de usar los dispositivos de red de que se dispongan, asociados a *direcciones* y *rutas*.

Dispositivos de red

- Para poder acceder a una red, es necesario que haya *hardware* específico para conectarse a ella.
- Este *hardware* pueden ser tarjetas *ethernet*, *token-ring*, tarjetas *wireless*, infrarrojos, *bluetooth*, etc.
- Cuando el *kernel* reconoce una tarjeta de red, le da un nombre para poder acceder a ella desde el sistema (“**ethN**” para Ethernet), pero *NO* existe ningún dispositivo (fichero del **/dev**) para poder acceder a ella.

Dispositivos de red (2)

- Se puede ver las tarjetas de red existentes mirando en los *buses* del sistema: PCI (**lspci**), USB (**lsusb**), etc.
- Cuando el *kernel* identifica y reconoce una tarjeta de red, muestra un mensaje por pantalla:

```
e100: eth0: e100_probe: addr 0xd0205000, irq 9, MAC addr 08:00:46:F3:F9:22
```

- Se puede consultar la lista de tarjetas de red existentes y reconocidas por el *kernel* en el fichero **/proc/net/dev**.
- Para consultar estadísticas, configuración, etc. de cada tarjeta de red: “**ifconfig**” o “**netstat -i**”.

ifconfig

- Se puede usar para ver la configuración de las tarjetas de red, o para modificar configuración de alguna de ellas.
- Sin argumentos, muestra todas las tarjetas *activas*.
- Con el nombre de una tarjeta de red, muestra esa tarjeta.
- Con la opción “-a”, muestra *todas* las tarjetas de red.

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:46:F3:F9:22
          inet addr:192.168.64.10  Bcast:192.168.62.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:46ff:fef3:f922/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:178320 errors:4212 dropped:4323 overruns:0 frame:4212
          TX packets:146240 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:116171954 (110.7 MiB)  TX bytes:21182840 (20.2 MiB)
```

ifconfig (2)

- La salida de `ifconfig`, por cada tarjeta de red, se divide en 4 partes:
 - Tipo de tarjeta y dirección *hardware*.
 - Configuración: dirección IP, *netmask*, *broadcast*, configuración IPv6.
 - *Flags*: UP (tarjeta activa), BROADCAST(*broadcast* en uso), MULTICAST (*multicast* en uso), PROMISC (modo promiscuo, etc).
 - Estadísticas: paquetes y bytes recibidos y enviados, errores, colisiones, etc.

Dirección IP, *netmask*, *broadcast*

- **inet addr**: Dirección IP local. Dos usos:
 - Todos los paquetes que tengan esa dirección IP de *destino* se entregan localmente a la máquina.
 - Los paquetes que se originen localmente y salgan por esa tarjeta de red, tienen esta dirección IP como *origen*.
- **broadcast**: Dirección IP para alcanzar a *toda la red*. Todos los paquetes que se envíen a esta dirección llegarán a todas las máquinas (incluida la máquina local).
- **netmask**: Indica cuál es el rango de direcciones IP que pertenecen a la red local en la que estamos situados.
Primera dirección de la red: (ip addr AND netmask)
Última dirección de la red: (ip addr OR (NOT netmask))

Dispositivo local (1o)

- No todos los dispositivos de red se corresponden con tarjetas físicas.
- Dispositivo “1o”: utilizado para establecer conexiones de red entre dos procesos de la misma máquina.
- Todo el tráfico enviado hacia el interfaz “1o” es recibido de nuevo por ese interfaz.
- Habitualmente se utiliza la dirección “127.0.0.1” para indicar la máquina local, o “localhost”, y se asigna esta dirección al interfaz “1o”, pero en algunos casos podría tener otra.

Tabla de rutas

- La configuración de cada tarjeta de red solo sirve para identificar qué paquetes van *hacia la máquina*, pero no qué hacer con los paquetes que van hacia fuera.
- La *tabla de rutas* indica qué hacer con cada paquete que no vaya dirigido a la máquina local, según su dirección IP destino.
- La tabla de rutas nos la proporciona el *kernel* en el fichero “`/proc/net/route`”, y se puede consultar mediante las órdenes “`route`” o “`netstat -r`”.

```
# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.64.0 * 255.255.255.0 U 0 0 0 eth0
default 192.168.64.1 0.0.0.0 UG 0 0 0 eth0
```

Tabla de rutas (2)

Cada ruta contiene 4 partes importantes:

- **Destino:** dirección IP base del rango que queremos indicar en esta ruta.
- **Máscara:** máscara asociada a la dirección destino para formar la red.
- **Gateway:** Dirección IP a la que mandar los paquetes que correspondan con esta ruta.
- **Interface:** Interfaz de red por el que enviar los paquetes de esta ruta.

Si el *gateway* no está especificado, indica que el paquete ha de enviarse localmente a la red a la que pertenece el interfaz.

Tabla de rutas (3)

- La máscara (*netmask*) asociada a una tabla de rutas ha de ser siempre una dirección con una serie de bits a “1” seguido de una serie de bits a “0”.
- El número de “1”s de la máscara indica lo genérica que es la red:
 - ningún “1” indicaría que esta es “toda la red”, también especificada como “default gateway”.
 - Todos los bits a “1” indica una ruta a “host”, esto es, hacia una única máquina y no una red de varias máquinas.
- Cuando un paquete tiene que rutarse y compararse con todas las rutas de la tabla, en caso de que coincida con varias, se usa siempre la ruta *más específica*, esto es, la que más bits tiene a “1”.

Envío de paquetes a una red local

- Cada tipo de tarjeta de red tendrá sus métodos para enviar paquetes locales.
- Lo más habitual es utilizar redes *ethernet* o compatibles, en las que se utiliza el protocolo ARP:
 - “ARP REQUEST” para averiguar qué dirección *hardware* corresponde con una determinada dirección IP.
 - “ARP REPLY” para contestar a una petición previa de “ARP REQUEST”.
- Las tablas ARP se almacenan en el fichero `/proc/net/arp`, y se consultan mediante la orden “`arp`”.
- La orden “`arping`” se puede usar para enviar paquetes “ARP REQUEST” y ver los “ARP REPLY” de respuesta.

Conexiones UDP y TCP

- En la práctica totalidad de comunicaciones entre dos máquinas se utiliza los protocolos UDP o TCP.
- El estado de las *conexiones* entre la máquina local y otras máquinas remotas usando UDP y TCP nos lo proporciona el *kernel* a través de los ficheros “/proc/net/{udp,tcp}”, y se pueden consultar con la orden “**netstat**”:
 - **netstat -u**: Muestra las conexiones UDP establecidas,
 - **netstat -t**: Muestra las conexiones TCP establecidas.
 - **netstat -l**: Muestra las conexiones en estado de espera.
 - **netstat -a**: Muestra todas las conexiones.
 - **netstat -n**: Muestra direcciones IP en lugar de nombres de máquina.

Resolución de nombres

- Normalmente, se utilizan nombres de máquinas en lugar de direcciones IP para establecer comunicaciones
- La traducción entre nombres y direcciones NO la realiza el *kernel*, sino las funciones de la *libc*. El *kernel* no sabe nada de nombres de máquina.
- Todas las aplicaciones utilizan funciones de la *libc*, como pueden ser `gethostbyname()` y `gethostbyaddr()`. Estas funciones leen el fichero “/etc/nsswitch.conf”, y actúan de distinta manera según el contenido de éste.

/etc/nsswitch.conf

- Base de datos para configuración de determinados servicios: passwd, group, hosts...
- Para cada servicio hay una línea en este fichero, con el nombre de dicho servicio, el carácter “:”, y la lista de maneras de acceder a ese servicio, separadas por espacio. Se prueban todas las maneras presentes en el fichero, hasta que una de ellas funciona.
- Ejemplo de /etc/nsswitch.conf:

```
passwd: compat
group:  compat
shadow: compat
```

```
hosts:  files dns
```

/etc/hosts

- Si en el fichero “/etc/nsswitch.conf” aparece una línea con “hosts: files”, se consulta el fichero “/etc/hosts” para realizar la conversión entre nombres de máquina y direcciones IP (y viceversa).
- Formato del fichero /etc/hosts: una línea por dirección IP, en la que aparezca:
dirección-IP nombre-máquina [alias1 alias2...]
- Ejemplo:

```
127.0.0.1    localhost
193.147.71.64 gsync.escet.urjc.es gsync
193.147.71.90 orion.cespedes.org orion mipc
```


/etc/resolv.conf

- Si en el fichero “/etc/nsswitch.conf” aparece una línea con “**hosts: dns**”, indica que se han de hacer las consultas de resolución de nombres a un servidor de DNS. En el fichero “/etc/resolv.conf” se indica qué servidores de DNS se deben consultar.
- Formato del /etc/resolv.conf: líneas con campos separados por espacios. El primero de los campos es una opción de configuración del DNS, y puede ser “**nameserver**”, “**domain**”, “**search**”, y algunas otras.
- La opción más relevante es “**nameserver**”. A continuación hay que indicar la dirección IP de un servidor de DNS al que realizar las peticiones. Si queremos usar varios servidores de DNS, hay que tener una línea “**nameserver**” por cada uno de ellos.

Cómo examinar el tráfico de una red

Si queremos saber qué está pasando por la red, para depurar algún problema o por cualquier otro motivo, podemos usar:

- **tcpdump**: Permite capturar todos los paquetes o los de un determinado tipo, de un interfaz de red en concreto, mostrando por pantalla las cabeceras o guardando cada paquete en un fichero para examinarlo posteriormente. Muy potente.
- **iptraf**: Genera estadísticas del tráfico recibido por uno o varios interfaces de red, a pantalla completa y con colores.
- **ethereal**: Programa interactivo y gráfico para examinar el tráfico. Permite hacer todo lo que hace **tcpdump**, pero de manera gráfica, mucho más fácil e intuitiva.

Herramientas de diagnóstico

- **netstat**: Muestra diversas configuraciones y estadísticas de la red: interfaces (“-i”), rutas (“-r”), conexiones TCP (“-t”), UDP (“-u”), etc.
- **ping**: envía paquetes ICMP ECHO_REQUEST, recibe ICMP ECHO_REPLY, habitualmente usado para comprobar la conectividad entre dos máquinas.
- **traceroute**, **mtr**: Envían paquetes con el campo TTL creciente, a partir de TTL=1, con el objeto de mostrar el camino que siguen los paquetes para llegar hasta su objetivo.
- **telnet**: Establece una conexión TCP con una máquina y un puerto dado.
- **host**, **dig**: Realizan consultas a un servidor DNS.

Resolución de problemas

- ¿Tarjeta de red instalada correctamente en el equipo? (conectada al cable de red, *link* de conexión, **lspci**, **lsusb**...).
- ¿Tarjeta de red reconocida por el *kernel*? (**dmesg**, contenido del `/proc/net/dev`, “**ifconfig -a**”).
- ¿Configurada correctamente? (**ifconfig**, “**route -n**”).
- ¿Recibe tráfico, muestra errores? (**ifconfig**).
- ¿Conectividad con la red local? (**arping**, **ping** al *gateway*).
- ¿Conectividad en Internet? (**ping**, **traceroute**, **mtr**, con direcciones IP en lugar de nombres).
- ¿DNS configurado correctamente? (`/etc/nsswitch.conf`, `/etc/resolv.conf`, **ping**, **host**, **dig**).