

# Tabla de encaminamiento, arp, ping y traceroute ARO/AI

GSyC  
Departamento de Teoría de la Señal y Comunicaciones y  
Sistemas Telemáticos y Computación  
URJC

Marzo de 2018



©2017-2018 GSyC - URJC.  
Algunos derechos reservados.  
Este trabajo se distribuye bajo la licencia  
Creative Commons Attribution Share-Alike  
disponible en <http://creativecommons.org/licenses/by-sa/3.0/es>

# Contenidos

- 1 Herramientas de configuración de la red: route
- 2 Configuración de rutas mediante ficheros de configuración
- 3 Herramientas de diagnóstico de red: arp, ping, traceroute

# Contenidos

- 1 Herramientas de configuración de la red: route
- 2 Configuración de rutas mediante ficheros de configuración
- 3 Herramientas de diagnóstico de red: arp, ping, traceroute

# Mostrar la tabla de encaminamiento

- La información de la tabla de encaminamiento de una máquina se puede obtener con la orden `route` o con `ip` o con `netstat`.

- Con `route`:

```
pc1:~# route
Kernel IP routing table
Destination Gateway Genmask      Flags   Metric  Ref  Use  Iface
11.0.0.0    *        255.255.255.0  U       0       0    0    eth0
```

- Con `ip`:

```
pc1:~# ip route show
11.0.0.0/24 dev eth0      proto kernel    scope   link      src 11.0.0.1
```

- Con `netstat`:

```
pc1:~# netstat -r
Kernel IP routing table
Destination Gateway Genmask      Flags   MSS  Window  irtt  Iface
11.0.0.0    *        255.255.255.0  U       0    0       0     eth0
```

## Configuración por defecto de la tabla de encaminamiento (I)

- Si una máquina no tiene asignada ninguna dirección IP, la tabla de encaminamiento estará vacía.
- Al asignar una dirección IP a una máquina, **automáticamente** se añade una entrada en la tabla de encaminamiento para que dicha máquina se pueda comunicar con las máquinas que están directamente conectadas a dicha subred.

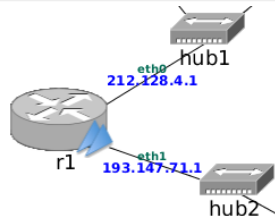


```
pc1:~# route
Kernel IP routing table
Destination  Gateway Genmask         Flags     Metric  Ref  Use Iface
212.128.4.0   *         255.255.255.0   U           0        0    0  eth0
```

\* : Es equivalente a 0.0.0.0

## Configuración por defecto de la tabla de encaminamiento (II)

- Si una máquina tiene asignadas varias direcciones IP, **automáticamente** tendrá configuradas en su tabla de encaminamiento tantas entradas como subredes a las que esté conectada directamente dicha máquina.



```
pc1:~# route
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
212.128.4.0	*	255.255.255.0	U	0	0	0	eth0
193.147.71.0	*	255.255.255.0	U	0	0	0	eth1

- \* : Es equivalente a 0.0.0.0
- En cada ruta de la tabla, la interfaz (iface) que aparece se refiere a la interfaz de la máquina en la que se ejecuta la orden (r1) por la que **saldrán** los paquetes que utilicen esa ruta.

# Añadir una ruta en la tabla de encaminamiento

- Con **route**:

- Ruta a una máquina:

```
route add -host <máquinaDestino> gw <gateway>
```

```
pc1:~# route add -host 12.0.0.1 gw 11.0.0.1
```

- Ruta a una subred

```
route add -net <subredDestino> netmask <máscara> gw <gateway>
```

```
pc1:~# route add -net 12.0.0.0 netmask 255.255.255.0 gw 11.0.0.1
```

- Ruta por defecto

```
route add default gw <gateway>
```

```
pc1:~# route add default gw 11.0.0.2
```

- Con **ip**:

- Ruta a una máquina o a una subred:

```
ip route add <dirIP/máscara> via <gateway>
```

```
pc1:~# ip route add 12.0.0.0/24 via 11.0.0.1
```

- Ruta por defecto `ip route add default via <gateway>`

```
pc1:~# ip route add default via 11.0.0.2
```

- Los cambios realizados con estas órdenes no se conservan al reiniciar la máquina.



# Borrar una ruta en la tabla de encaminamiento

- Con `route`:

- Ruta a una máquina:

```
route del -host <máquinaDestino>
```

```
pc1:~# route del -host 12.0.0.1
```

- Ruta a una subred

```
route del -net <subredDestino> netmask <máscara>
```

```
pc1:~# route del -net 12.0.0.0 netmask 255.255.255.0
```

- Ruta por defecto

```
route del default
```

```
pc1:~# route del default
```

- Con `ip`:

- Ruta a una máquina o a una subred:

```
ip route del <dirIP/máscara> via <gateway>
```

```
pc1:~# ip route del 12.0.0.0/24 via 11.0.0.1
```

- Ruta por defecto

```
ip route del default via <gateway>
```

```
pc1:~# ip route del default via 11.0.0.2
```

- Los cambios realizados con estas órdenes no se conservan al reiniciar la máquina.

# Contenidos

- 1 Herramientas de configuración de la red: route
- 2 Configuración de rutas mediante ficheros de configuración
- 3 Herramientas de diagnóstico de red: arp, ping, traceroute

# Fichero de configuración de red

- Los cambios en la configuración de red realizados en el terminal con `ifconfig/ip/route` no se mantienen si se apaga y se vuelve a encender la máquina.
- Al arrancar una máquina su configuración de red por defecto se lee de un fichero de configuración.
- Dependiendo de la distribución de Linux, la configuración de red puede estar en un fichero o conjunto de ficheros diferentes.
  - En Debian y derivados (como Ubuntu) la configuración de red está en el fichero `/etc/network/interfaces`

# Ruta por defecto en `/etc/network/interfaces`

- Ejemplo de configuración de red en el fichero `/etc/network/interfaces`:

<code>auto lo</code>	→ la interfaz <code>lo</code> se configurará automáticamente al activar la red
<code>iface lo inet loopback</code>	→ la interfaz <code>lo</code> tendrá la dirección predefinida para la interfaz de loopback ( <code>127.0.0.1</code> )
<code>auto eth0</code>	→ la interfaz <code>eth0</code> se configurará automáticamente al activar la red
<code>iface eth0 inet static</code>	→ la interfaz <code>eth0</code> tendrá una IP estática
<code>address 11.0.0.10</code>	→ dirección IP de <code>eth0</code>
<code>netmask 255.255.255.0</code>	→ máscara de la subred a la que pertenece <code>eth0</code>
<code>gateway 11.0.0.1</code>	→ ruta por defecto a través de <code>11.0.0.1</code> (opcional)

- Cuando se modifica este fichero es necesario reiniciar las interfaces de red para que la nueva configuración surta efecto, mediante la orden:  
`/etc/init.d/networking restart`
- Puedes ver otros ejemplos de configuración de interfaces de red con:  
`zless /usr/share/doc/ifupdown/examples/network-interfaces.gz`
- Puedes consultar el manual: `man interfaces`

# Configuración de rutas a través de /etc/network/interfaces: Ejemplo

- Fichero `/etc/network/interfaces` incluyendo rutas:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 11.0.0.10
    netmask 255.255.255.0
    up route add -net 12.0.0.0 netmask 255.255.255.0 gw 11.0.0.2
    up route add default gw 11.0.0.1
```

- Es equivalente poner:

```
up route add default gw 11.0.0.1
```

a poner:

```
gateway 11.0.0.1
```

- En la sección de una interfaz puede ponerse cualquier orden precedida por `up`: cuando se active esa interfaz se ejecutará la orden.
- También pueden ponerse órdenes precedidas por `down`: cuando se apague esa interfaz se ejecutará la orden.

# Contenidos

- 1 Herramientas de configuración de la red: route
- 2 Configuración de rutas mediante ficheros de configuración
- 3 Herramientas de diagnóstico de red: arp, ping, traceroute

# Herramientas de diagnóstico de red

- **Diagnóstico de red:** Monitorizar el estado de conectividad a la red de las máquinas
- Herramientas que veremos en este tema:
  - arp
  - ping
  - traceroute

## Cachés de ARP

- Para consultar la caché de ARP en una máquina se utiliza la orden `arp`:

```
pc2:~# arp -a  
? (11.0.0.1) at 0A:29:92:55:93:70 [ether] on eth0
```

- Para borrar la caché de ARP:
  - Pasados unos 10 minutos de la última vez que se consultó una entrada, esta entrada se borra
  - Si se apaga y enciende una interfaz de red, se borran todas las entradas aprendidas por esa interfaz:

```
pc2:~# ifconfig eth0 down  
pc2:~# ifconfig eth0 up
```

- Puede borrarse manualmente una entrada concreta con la orden `arp` mediante la opción `-d`:

```
pc2:~# arp -d 11.0.0.2
```



## Comprobar la conectividad entre dos dispositivos: ping

- La orden **ping** permite comprobar si se puede alcanzar una máquina, y el tiempo que se tarda en ir y volver a ella (*round-trip time*, RTT).
- Envía un paquete cada segundo. La máquina destino contestará a cada uno de ellos con un paquete de respuesta.
- Por defecto **ping** se ejecuta indefinidamente. Hay que utilizar Ctrt+C para interrumpirlo.
- Tiene muchas opciones, las más habituales son:
  - **-c <númeroPaquetes>**: número de paquetes a enviar en vez de ejecutarse indefinidamente. que se envían. que se envían.
  - **-t <TTL>**: TTL inicial de los paquetes que se envían (por defecto, 64).

## ping: Ejemplo

```
pc2:~# ping 11.0.0.1
PING 11.0.0.1 (11.0.0.1): 56(84) bytes of data
64 bytes from 11.0.0.1: icmp_seq=0 ttl=64 time=1.896 ms
64 bytes from 11.0.0.1: icmp_seq=1 ttl=64 time=2.110 ms
64 bytes from 11.0.0.1: icmp_seq=2 ttl=64 time=2.125 ms
^C
--- 11.0.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 1.896/2.044/2.125/0.105 ms
```

- Cuando se interrumpe el **ping**, aparece un resumen estadístico que contiene:
  - porcentaje de pérdidas
  - RTT (*round-trip time*, “tiempo de ronda”, es decir, tiempo en ir y volver al destino) mínimo, medio y máximo, y desviación media

## Comprobar la ruta desde un origen a un destino: traceroute

- Envía paquetes UDP con puerto destino 33435, variando el TTL.
- Comienza enviando 3 paquetes con TTL=1, cuando obtiene alguna respuesta (ICMP: time exceeded) aumenta a TTL=2 y así sucesivamente hasta que obtiene la respuesta UDP port unreachable
- Cada vez que se obtiene una respuesta se imprime información de la máquina que envió dicha respuesta y el RTT con dicha máquina.
- NOTA: Consulta los detalles completos del funcionamiento del traceroute en las transparencias de teoría.

# traceroute: Ejemplo

```
pc4:~# traceroute -n 11.0.0.1
traceroute to 11.0.0.1 (11.0.0.1), 64 hops max, 40 byte packets
 1 14.0.0.1 (14.0.0.1) 2.3 ms 3.3 ms 1.8 ms
 2 13.0.0.1 (13.0.0.1) 4.7 ms 5.6 ms 4.8 ms
 3 12.0.0.1 (12.0.0.1) 6.3 ms 8.3 ms 7.6 ms
 4 15.0.0.1 (15.0.0.1) 8.9 ms 10.5 ms 9.8 ms
 5 11.0.0.1 (11.0.0.1) 11.3 ms 10.3 ms 11.7 ms
```