



Università degli Studi Roma Tre
Dipartimento di Informatica e Automazione
Computer Networks Research Group

netkit lab

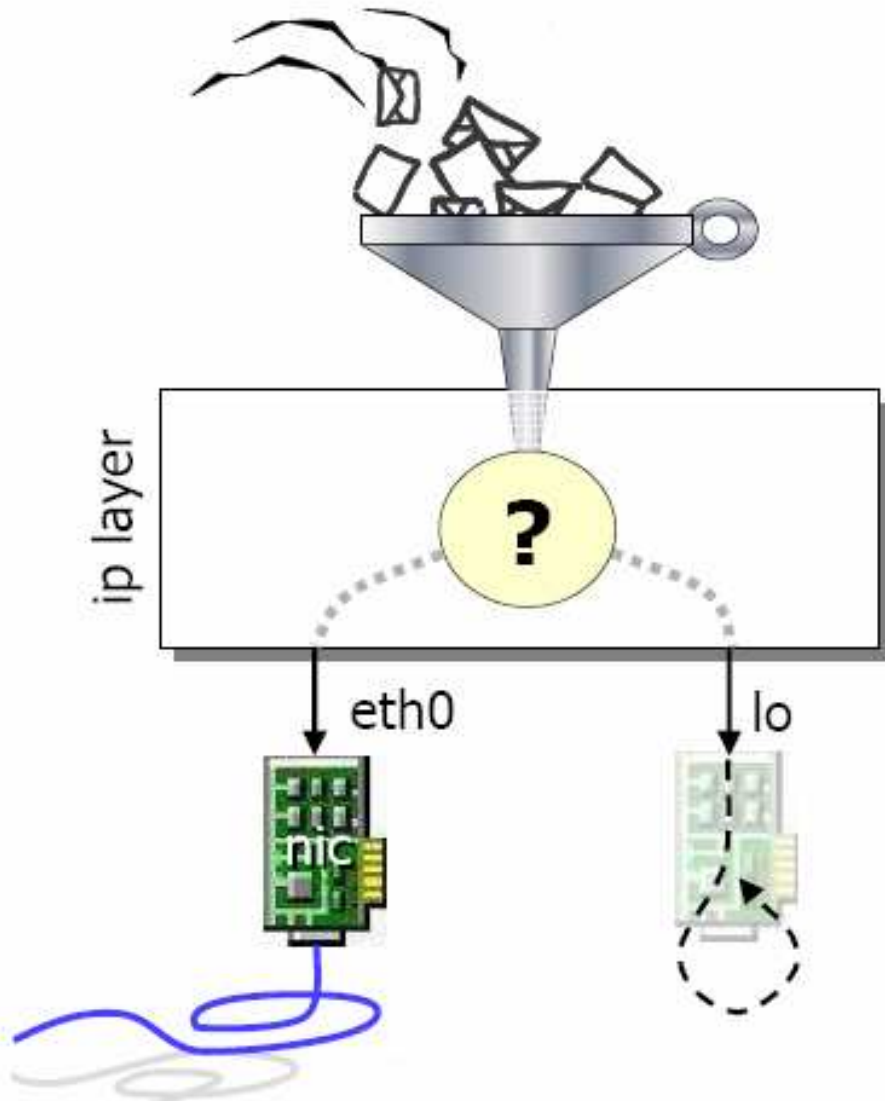
rip

Version	2.2
Author(s)	G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini, traducción J.M. San José
E-mail	contact@netkit.org
Web	http://www.netkit.org/
Description	Experiencias con el protocolo de encaminamiento vector distancia ripv2

copyright notice

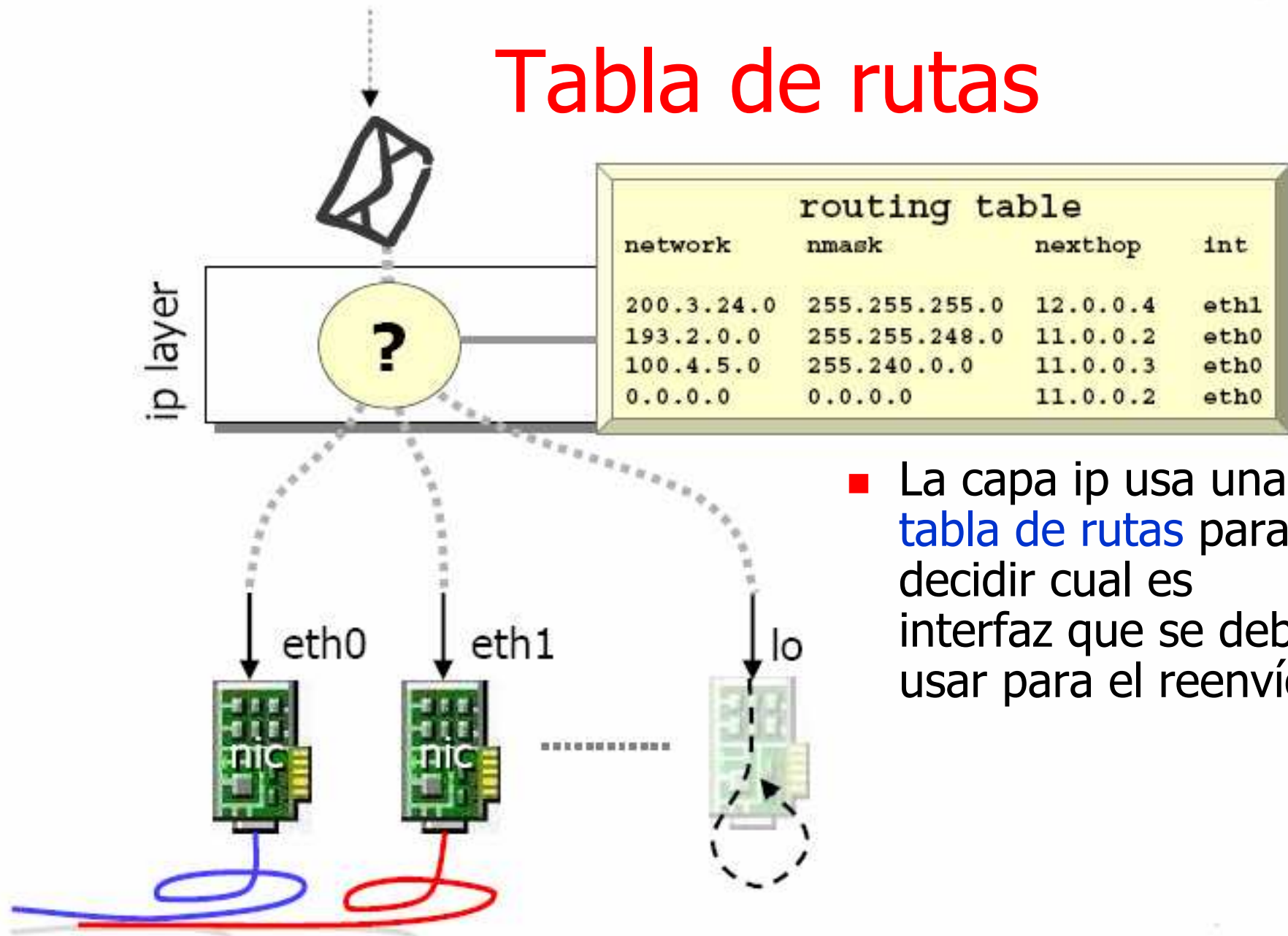
- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material “as is”, with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.

Los sistemas necesitan routing



- Cada sistema con una pila de red realiza un encaminamiento elemental
- Al menos, la pila de red puede ser usada para acceder a los servicios locales (p.e., XWindows)
- El sistema debe decidir cuando un paquete necesita ser enviado a la tarjeta del interfaz de red (NIC) y cuando se necesita ser rebotado al interfaz loopback (lo)

Tabla de rutas



- La capa ip usa una **tabla de rutas** para decidir cual es interfaz que se debe usar para el reenvío

The diagram illustrates the routing process. A packet enters a router, and the routing process asks "is it for me?". If yes, it is delivered. If no, it is forwarded to the next hop. The diagram shows the IP layer, network interfaces (eth0, eth1, lo), and network cards (NIC).

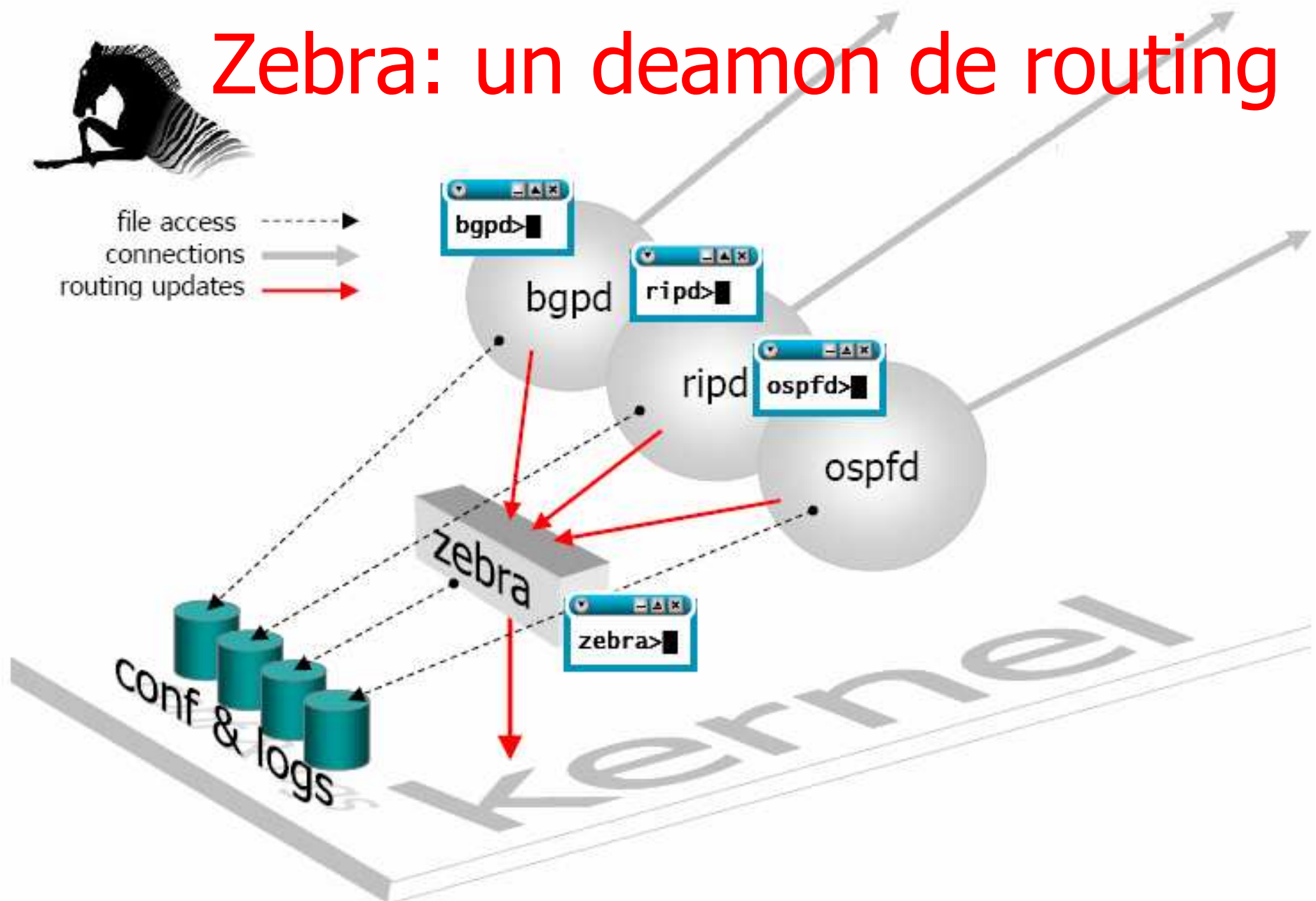
- © Computer Networks netkit
Research Group Roma Tre

Protocolos de routing

- Los protocolos de routing se usan para actualizar automáticamente las tablas de rutas
- Se clasifican en dos categorías principales:
 - Protocolos de routing de estado del enlace
 - Aproximación: envío de la información mínima a todo el mundo
 - Cada router reconstruye el grafo completo de red y calcula el camino más corto a todos los destinos
 - Ejemplos: IS-IS, OSPF
 - Protocolos de routing de vector distancia
 - Aproximación: envío de toda la información a unos pocos
 - Actualiza la información de routing basándose en lo que ha oído
 - Ejemplos: RIP, BGP
- En este laboratorio veremos un ejemplo de protocolo RIPv2 en las cajas zebra



Zebra: un daemon de routing



Revisando los ficheros de configuración de Zebra

▼ Máquina virtual

```
pc1:~# cd /etc/zebra/  
pc1:/etc/zebra# ls  
bgpd.conf          ospf6d.conf        ripd.conf          vtysh.conf  
bgpd.conf.sample  ospf6d.conf.sample ripd.conf.sample  vtysh.conf.sample  
bgpd.conf.sample2 ospfd.conf          ripngd.conf        zebra.conf  
daemons           ospfd.conf.sample  ripngd.conf.sample zebra.conf.sample  
pc1:/etc/zebra# █
```

- Cuando zebra se arranca, cada deamon comprueba estos ficheros para leer la configuración de arranque

Ejemplo de fichero deamons

▼ Máquina virtual

```
pc1:/etc/zebra# less daemons
# This file tells the zebra package
# which daemons to start.
# Entries are in the format: <daemon>=(yes|no|priority)
# where 'yes' is equivalent to infinitely low priority, and
# lower numbers mean higher priority. Read
# /usr/doc/zebra/README.Debian for details.
# Daemons are: bgpd zebra ospfd ospf6d ripd ripngd
zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=no
daemons (END)
```

El daemon principal de zebra arrancará

El daemon de RIP arrancará también

Ejemplo de fichero de configuración de Zebra (zebra.conf)

▼ Máquina virtual

```
pc1:/etc/zebra# less zebra.conf
```

```
! *- zebra *-
```

```
!
```

```
! zebra sample configuration file
```

```
!
```

```
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer
```

```
Exp $
```

```
!
```

```
hostname Router
```

```
password zebra
```

```
enable password zebra
```

```
!
```

```
! interface lo
```

El prompt del interfaz zebra

La contraseña para para conectarse al daemon

La contraseña administrativa

Ejemplo de fichero de configuración de ripd (ripd.conf)

▼ Máquina virtual

```
pc1:/etc/zebra# cat ripd.conf
```

```
!
```

```
hostname ripd
```

```
password root
```

```
enable password root
```

```
!
```

```
router rip
```

```
redistribute connected
```

```
network 100.1.0.0/16
```

```
!
```

```
log file /var/log/zebra/ripd.log
```

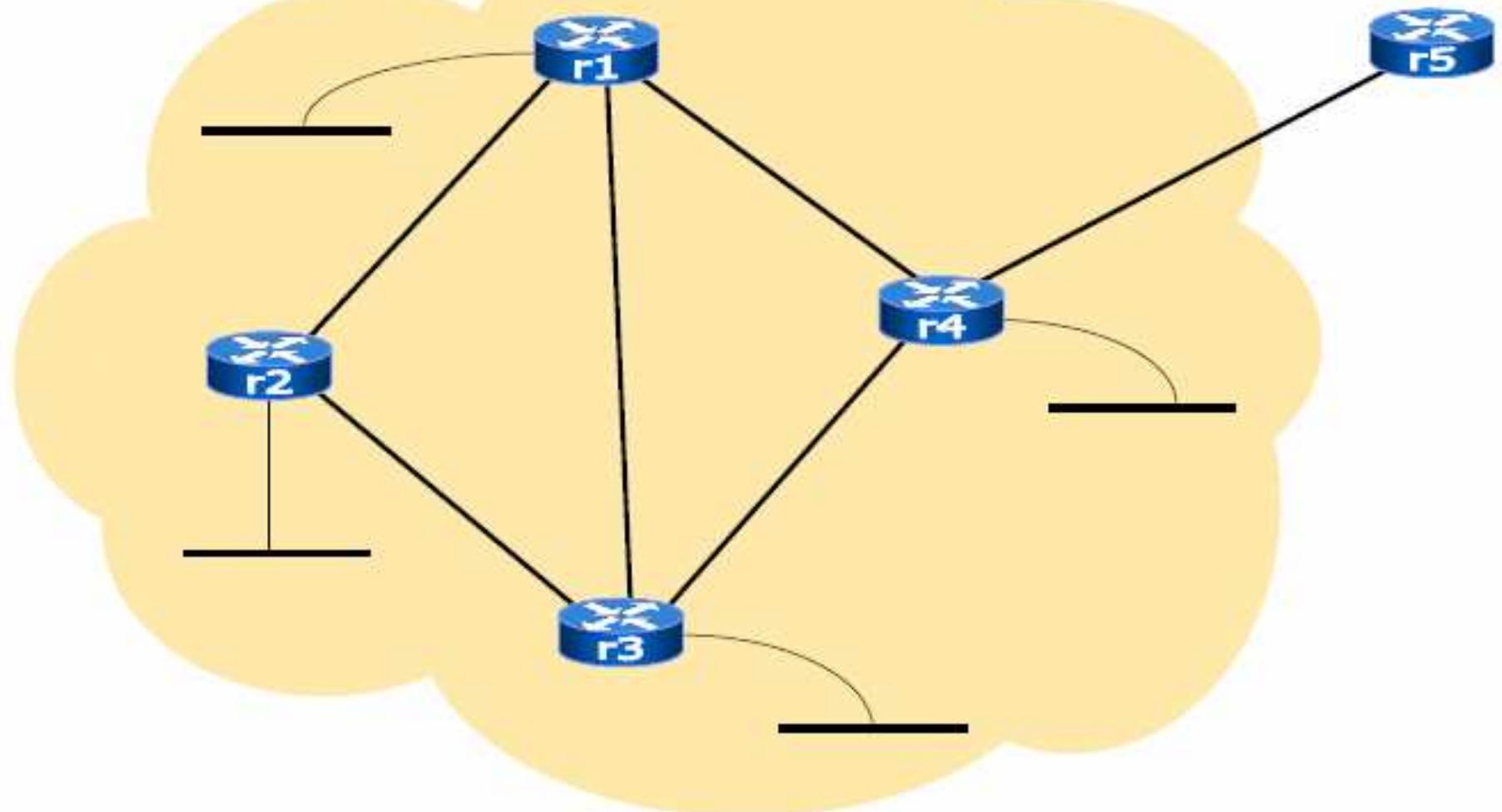
```
pc1:/etc/zebra# █
```

Habla RIP en un interfaz dado

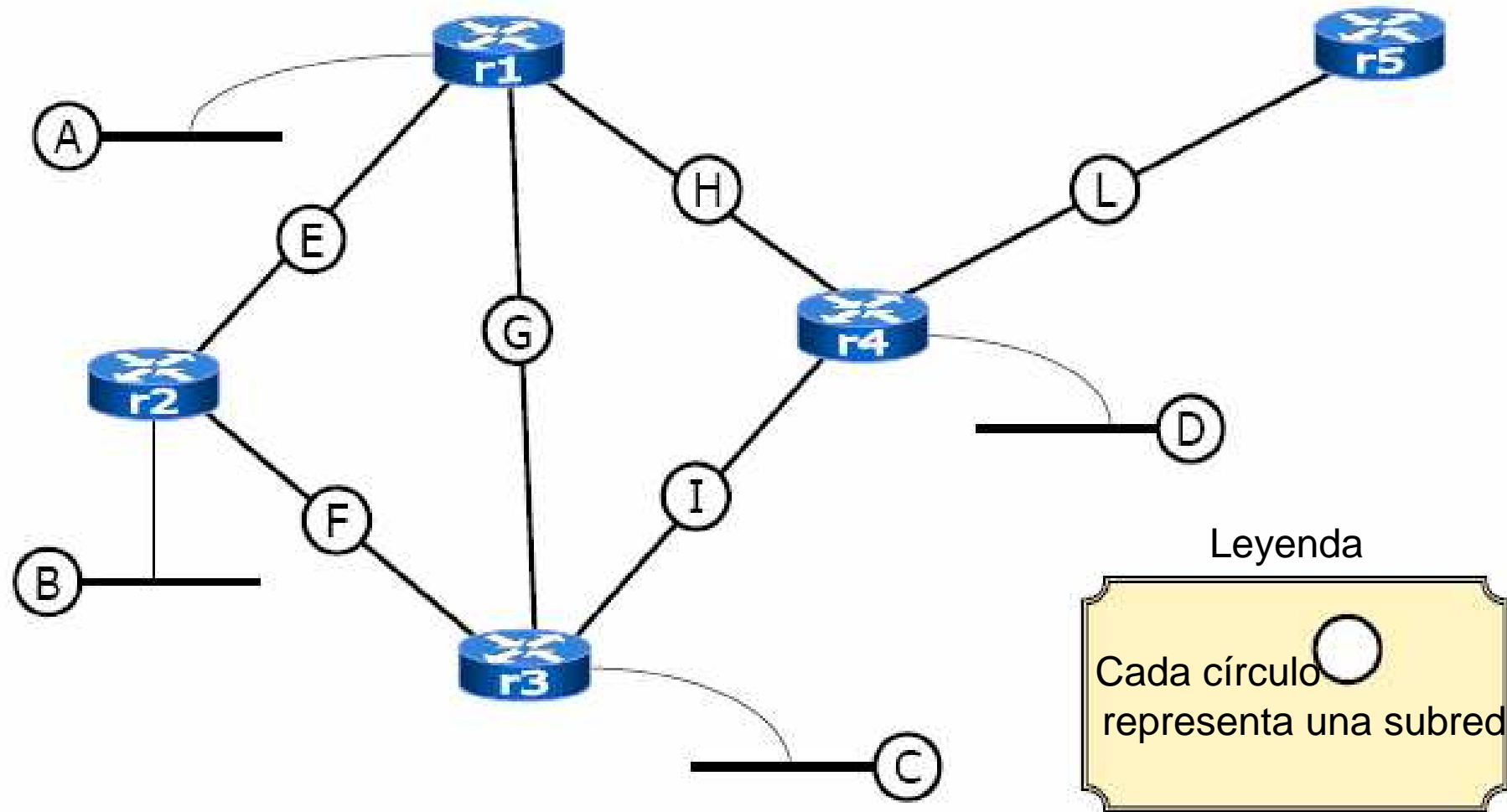
Redistribuye a los vecinos RIP información sobre todas las subredes directamente conectadas

Envía paquetes multicast de RIP a todos los interfaces que caigan en este prefijo

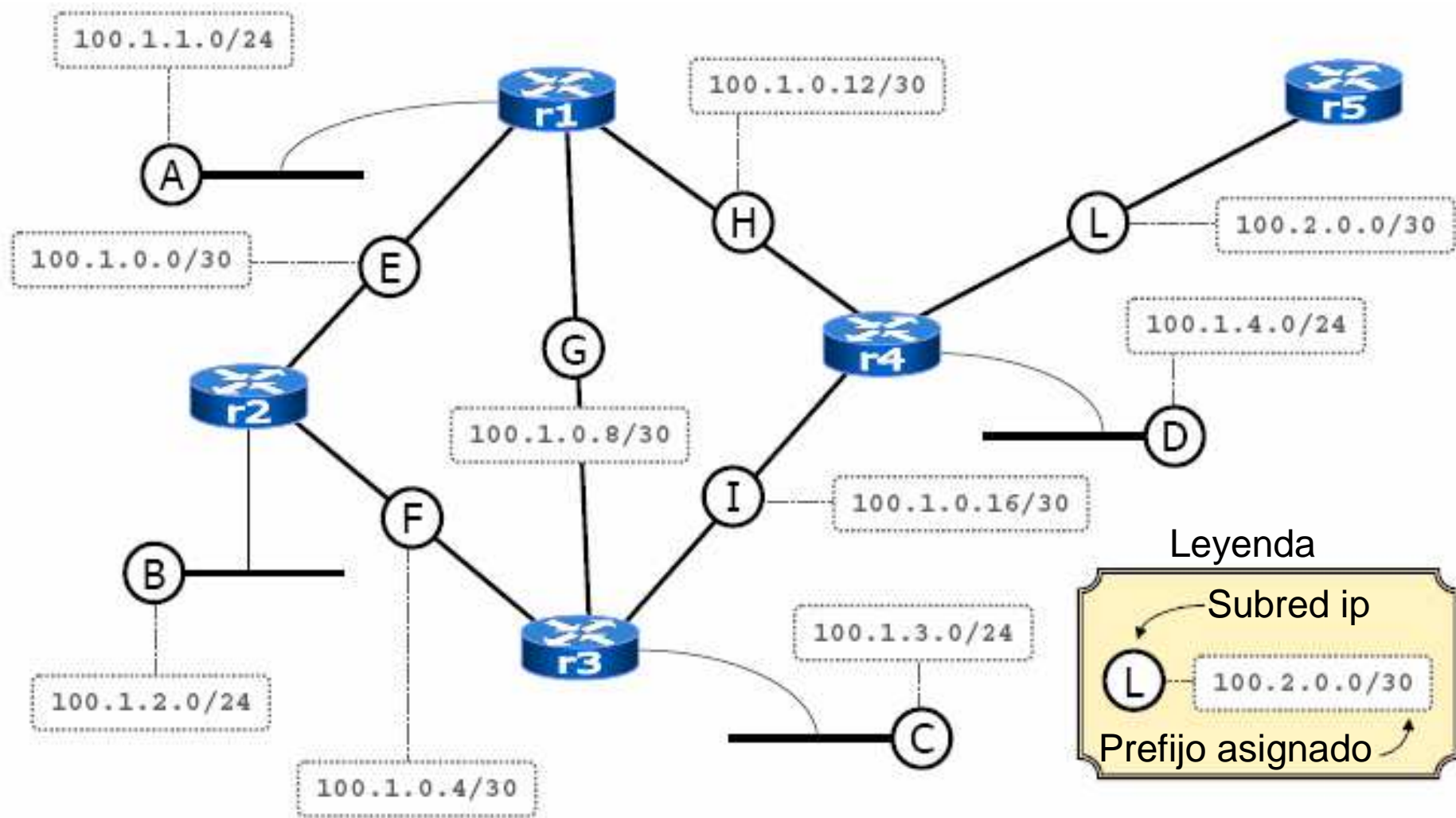
Una pequeña red conectada a Internet



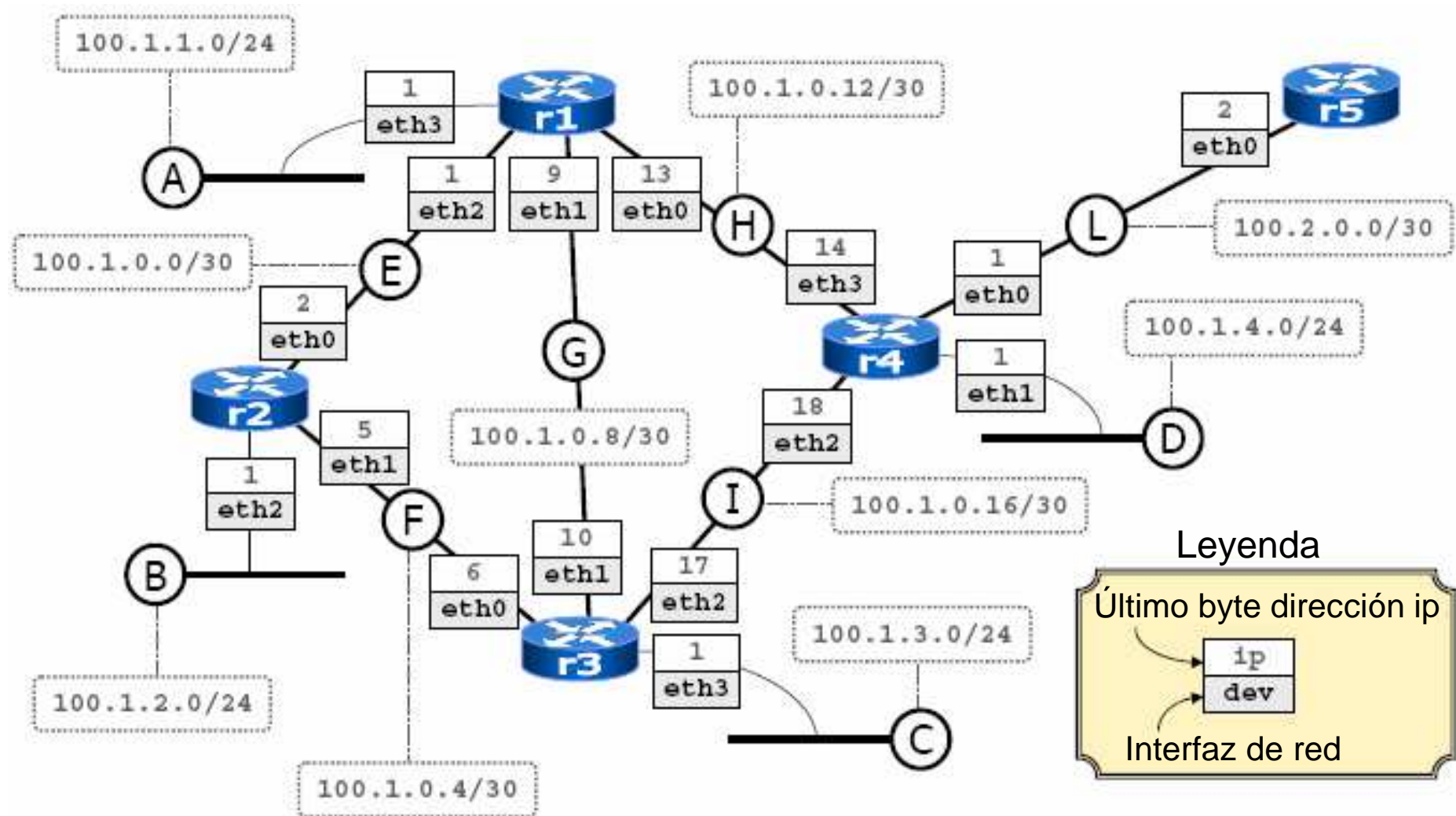
Las subredes ip relacionadas



Asignando direcciones ip a subredes



Asignando direcciones ip a interfaces



Lanzando el script del laboratorio



```
▼ Máquina anfitrión
user@localhost:~$ cd netkit-lab_rip
user@localhost:~/netkit-lab_rip$ lstart
```

- La configuración del laboratorio es la siguiente:
 - Cinco sistemas virtuales se crean y conectan a los dominios de colisión correctos (hubs virtuales)
 - Para cada sistema virtual
 - Los interfaces de red se configuran automáticamente
 - Los ficheros de configuración **/etc/zebra/daemons**, **/etc/zebra/zebra.conf**, and **/etc/zebra/ripd.conf** están actualizados.
 - El daemon de routing de zebra no se arranca automáticamente.

Comprobando la conectividad

- A un destino directamente conectado.

▼ r4

```
r4:~# ping 100.1.0.13
```

```
PING 100.1.0.13 (100.1.0.13) 56(84) bytes of data.
```

```
64 bytes from 100.1.0.13: icmp_seq=1 ttl=64 time=1.23 ms
```

```
64 bytes from 100.1.0.13: icmp_seq=2 ttl=64 time=0.592 ms
```

```
64 bytes from 100.1.0.13: icmp_seq=3 ttl=64 time=0.393 ms
```

```
--- 100.1.0.13 ping statistics ---
```

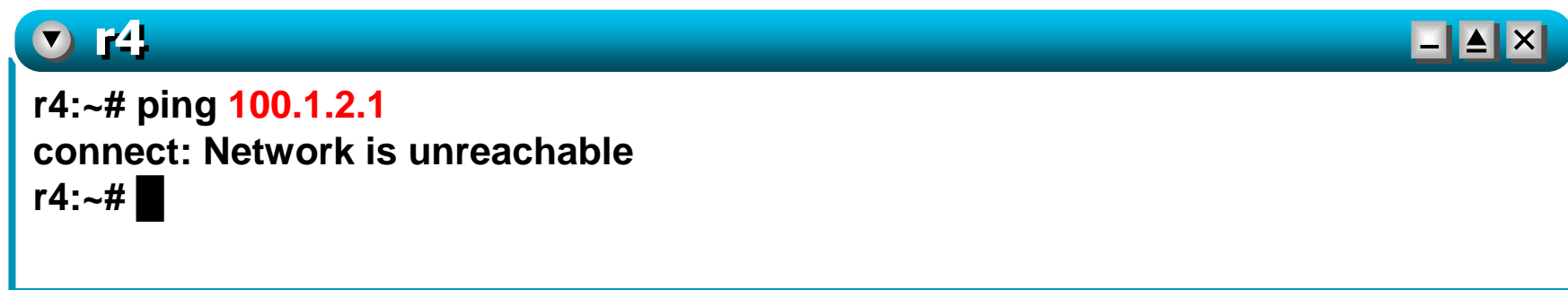
```
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
```

```
rtt min/avg/max/mdev = 0.393/0.741/1.238/0.360 ms
```

```
r4:~# █
```

Comprobando la conectividad

- A un destino remoto.



```
r4:~# ping 100.1.2.1
connect: Network is unreachable
r4:~#
```

A terminal window titled 'r4' with standard window controls (minimize, maximize, close) in the top right corner. The terminal shows a user at the 'r4' prompt typing 'ping 100.1.2.1'. The output is 'connect: Network is unreachable', indicating a connectivity failure. The prompt returns to 'r4:~#' with a cursor.

- ¿Qué está pasando?

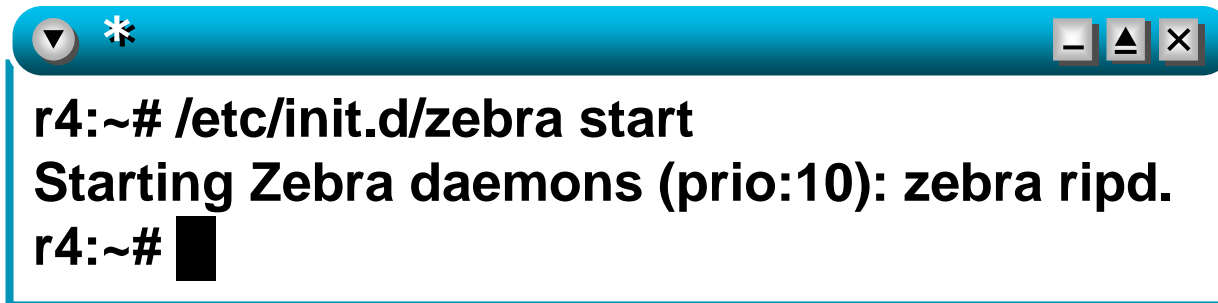
Examinando la tabla de rutas del kernel

```
r4:~# route
Kernel IP routing table
Destination      Gateway Genmask         Flags Metric Ref    Use     Iface
100.1.0.16       *      255.255.255.252  U        0      0      0      eth2
100.2.0.0        *      255.255.255.252  U        0      0      0      eth0
100.1.0.12       *      255.255.255.252  U        0      0      0      eth3
100.1.4.0        *      255.255.255.0    U        0      0      0      eth1
r4:~#
```

- Desde el momento en que no hay deamon de routing en ejecución sólo son conocidos los destinos directamente conectados

Arrancando los deamon de routing

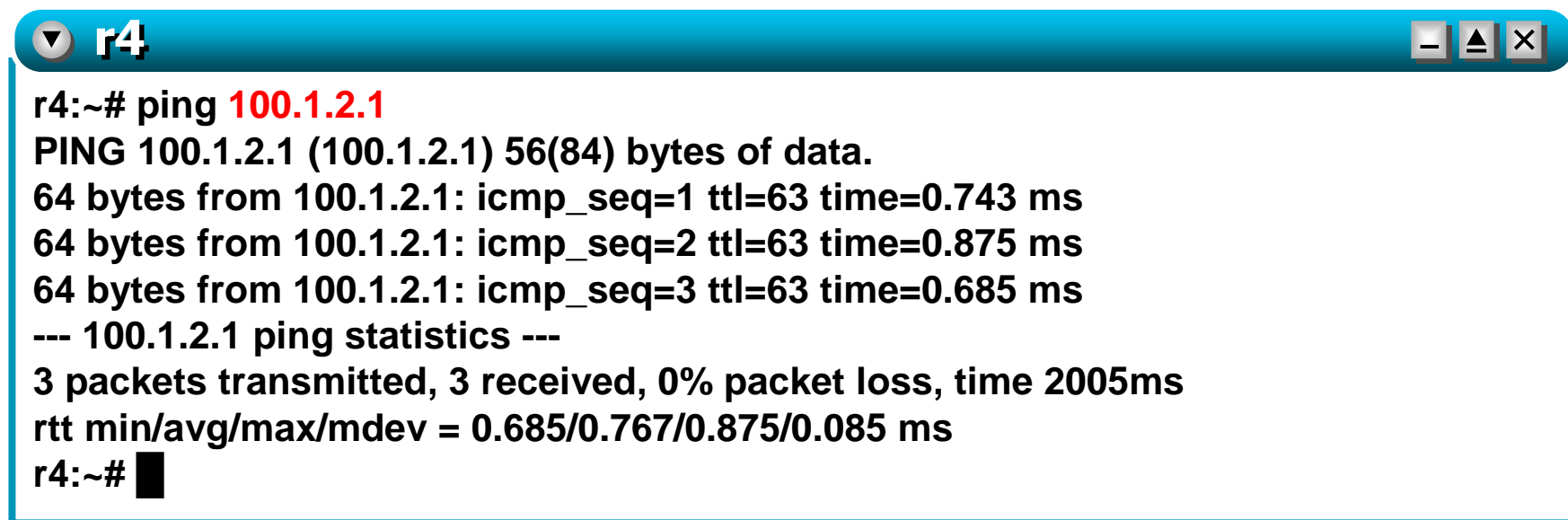
- Para cada router (también r5) se ejecuta el comando siguiente

A terminal window with a blue title bar containing a dropdown arrow, an asterisk, and window control buttons (minimize, maximize, close). The terminal text shows a user at the r4 prompt running the command to start Zebra daemons.

```
r4:~# /etc/init.d/zebra start
Starting Zebra daemons (prio:10): zebra ripd.
r4:~# █
```

Comprobando la conectividad de nuevo

- A un destino remoto



```
r4:~# ping 100.1.2.1
PING 100.1.2.1 (100.1.2.1) 56(84) bytes of data.
64 bytes from 100.1.2.1: icmp_seq=1 ttl=63 time=0.743 ms
64 bytes from 100.1.2.1: icmp_seq=2 ttl=63 time=0.875 ms
64 bytes from 100.1.2.1: icmp_seq=3 ttl=63 time=0.685 ms
--- 100.1.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.685/0.767/0.875/0.085 ms
r4:~#
```

- Después de un momento, todos los destinos remotos son alcanzables

Comprobando la tabla de rutas

- La tabla de rutas está ya actualizada

```
r4:~# route
Kernel IP routing table
Destination      Gateway    Genmask          Flags Metric Ref    Use    Iface
100.1.0.16       *          255.255.255.252  U      0      0      0      eth2
100.1.0.0        100.1.0.13 255.255.255.252  UG     2      0      0      eth3
100.1.0.4        100.1.0.17 255.255.255.252  UG     2      0      0      eth2
100.2.0.0        *          255.255.255.252  U      0      0      0      eth0
100.1.0.8        100.1.0.17 255.255.255.252  UG     2      0      0      eth2
100.1.0.12       *          255.255.255.252  U      0      0      0      eth3
100.1.4.0        *          255.255.255.0    U      0      0      0      eth1
100.1.2.0        100.1.0.17 255.255.255.0    UG     3      0      0      eth2
100.1.3.0        100.1.0.17 255.255.255.0    UG     2      0      0      eth2
100.1.1.0        100.1.0.13 255.255.255.0    UG     2      0      0      eth3
r4:~#
```

Una ojeada a los paquetes ripv2

■ Capturemos paquetes ripv2

r4

```
r4:~# tcpdump -i eth2 -v -n -s 1518
```

**Mostrar detalles del paquete
(activa la decodificación
completa de protocolos)**

**Captura paquetes
ethernet completos
(por defecto solo se
capturan los 68
primeros Bytes)**

**No resuelve
direcciones a nombres**

Una ojeada a los paquetes ripv2

■ Capturemos paquetes ripv2

```
r4:~# tcpdump -i eth2 -v -n -s 1518
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 1518
bytes
16:47:48.333986 IP (tos 0x0, ttl 1, id 0, offset 0, flags [DF], length: 152)
100.1.0.17.520 > 224.0.0.9.520: [udp sum ok]
    RIPv2, Response, length: 124, routes: 6
        AFI: IPv4: 100.1.0.0/30, tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4: 100.1.0.4/30, tag 0x0000, metric: 1, next-hop: self
        AFI: IPv4: 100.1.0.8/30, tag 0x0000, metric: 1, next-hop: self
        AFI: IPv4: 100.1.1.0/24, tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4: 100.1.2.0/24, tag 0x0000, metric: 2, next-hop: self
        AFI: IPv4: 100.1.3.0/24, tag 0x0000, metric: 1, next-hop: self
1 packets captured
1 packets received by filter
0 packets dropped by kernel
r4:~#
```


Un traceroute

▼ r4

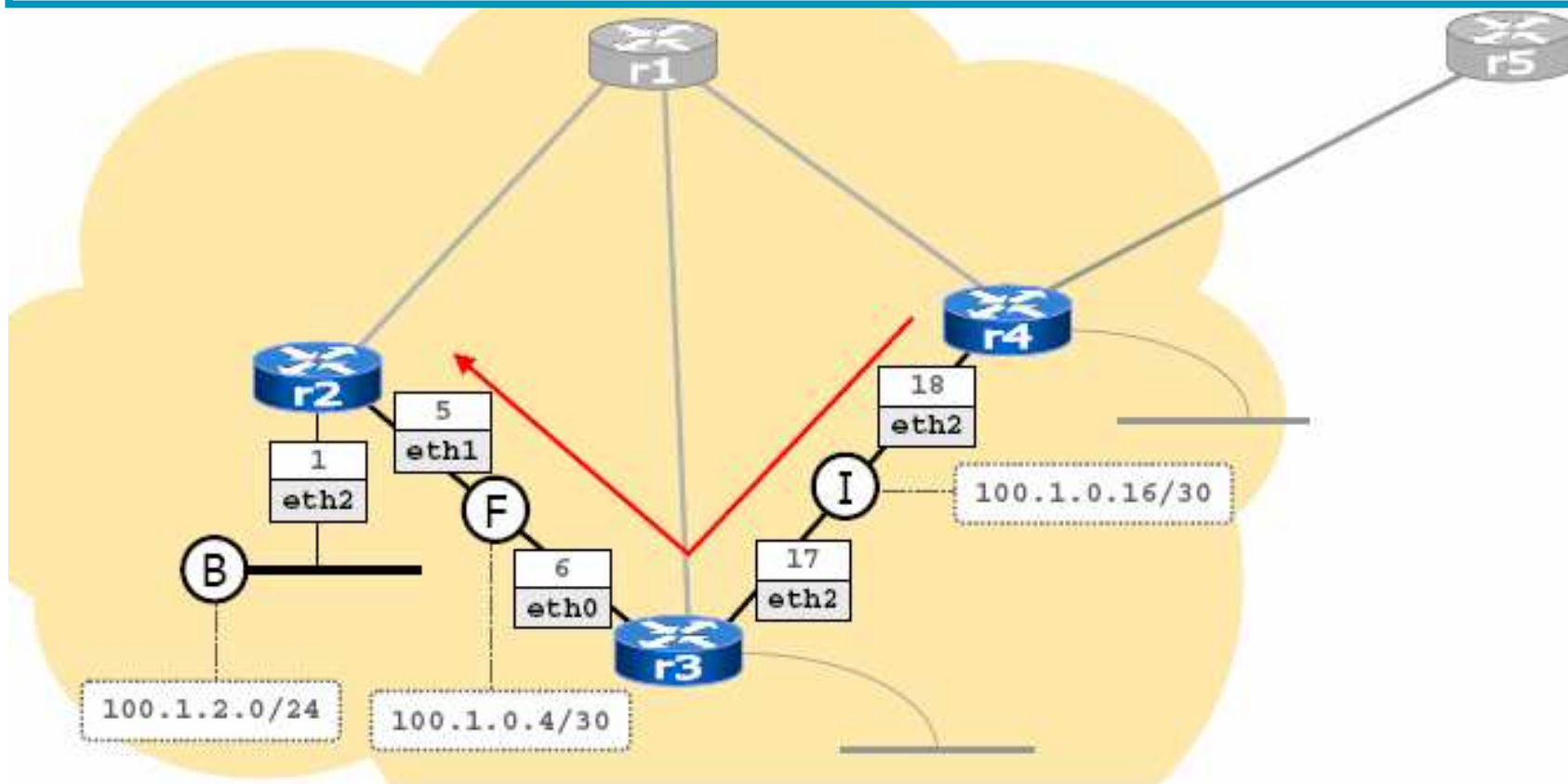
```
r4:~# traceroute 100.1.2.1
```

```
traceroute to 100.1.2.1 (100.1.2.1), 64 hops max, 40 byte packets
```

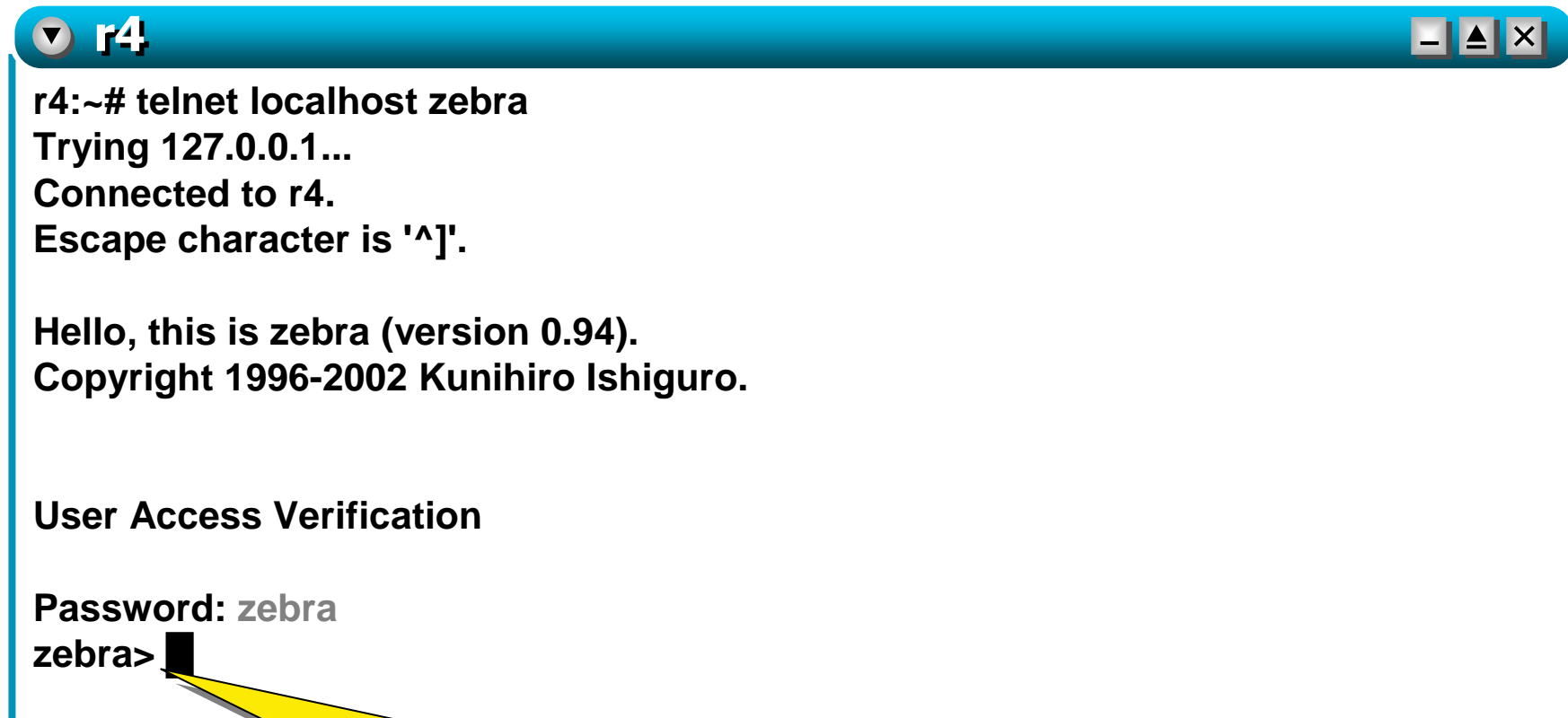
```
1 100.1.0.17 (100.1.0.17) 10 ms 3 ms 1 ms
```

```
2 100.1.2.1 (100.1.2.1) 15 ms 1 ms 1 ms
```

```
r4:~# █
```



Conectando con el deamon principal de zebra



```
r4:~# telnet localhost zebra
Trying 127.0.0.1...
Connected to r4.
Escape character is '^]'.

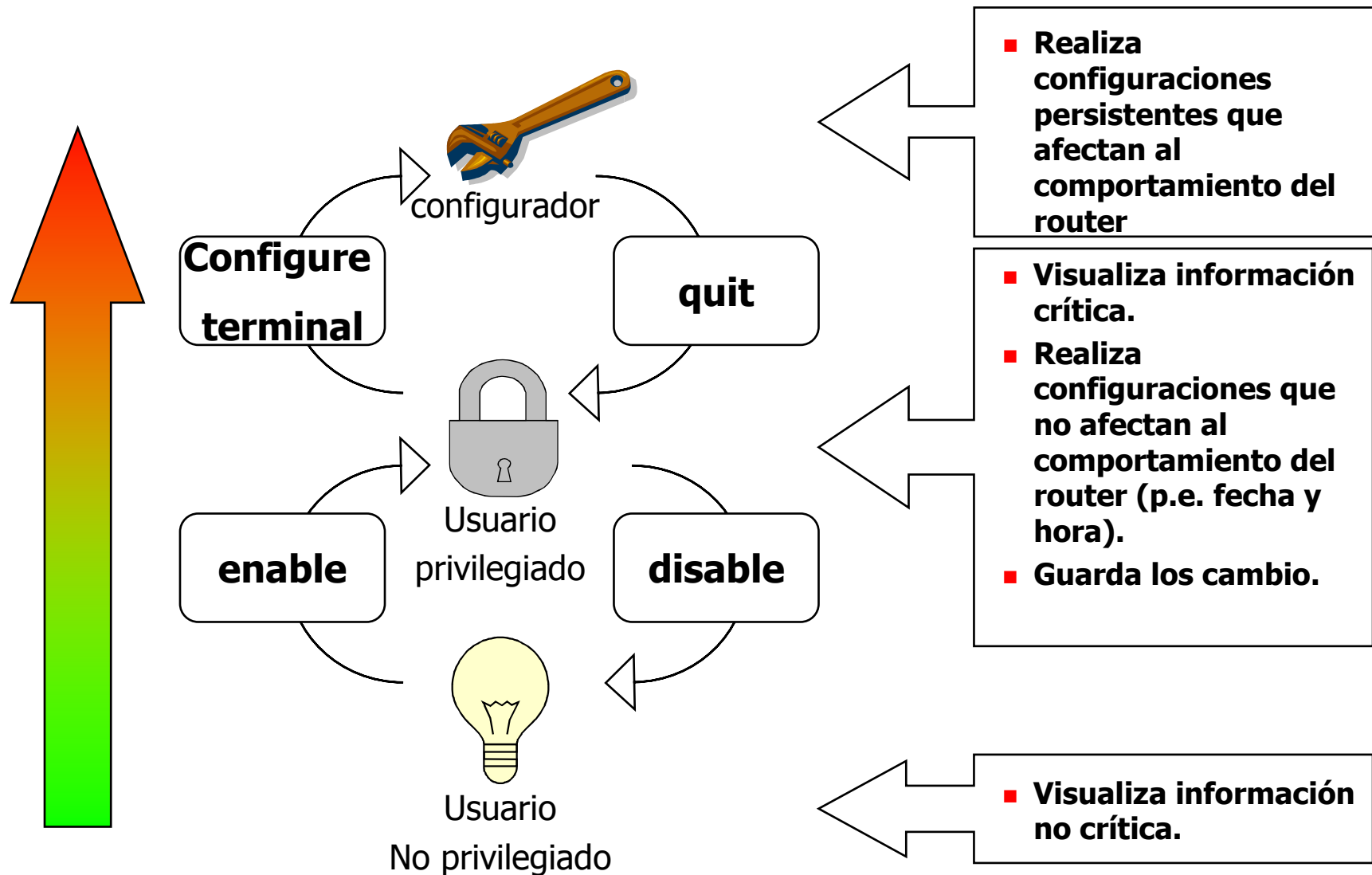
Hello, this is zebra (version 0.94).
Copyright 1996-2002 Kunihiro Ishiguro.

User Access Verification

Password: zebra
zebra>
```

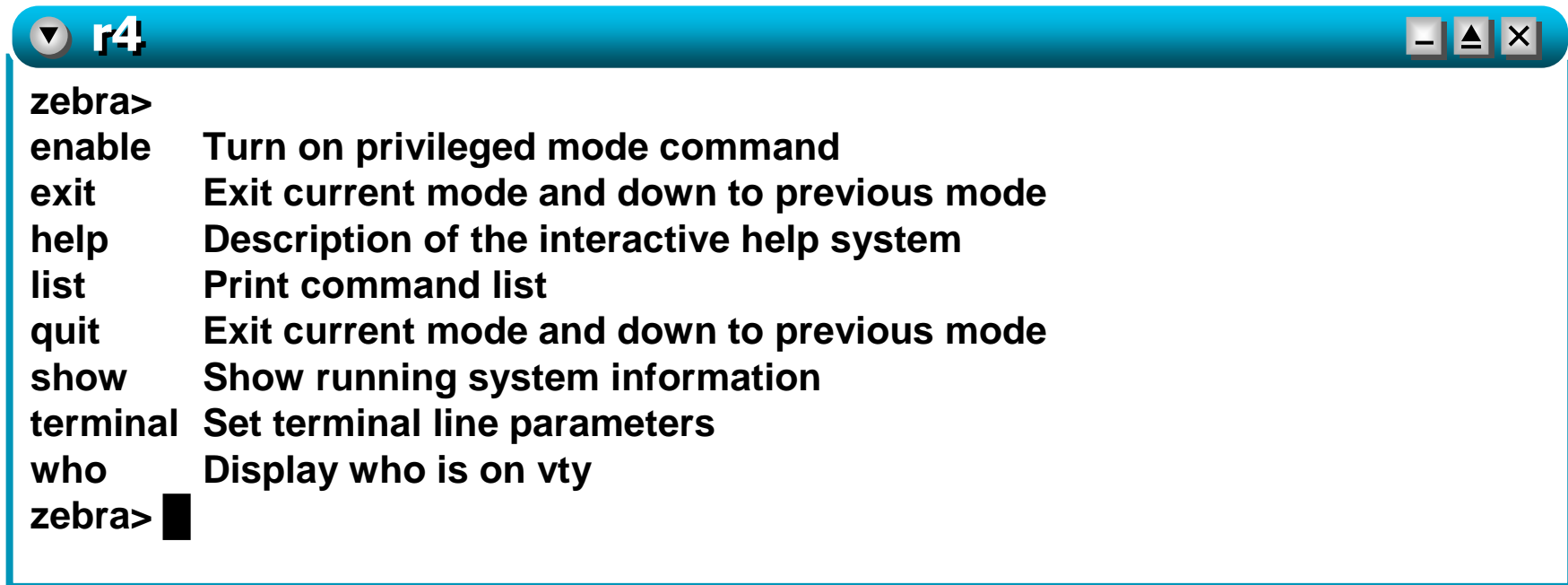
Somos usuarios no privilegiados

Privilegios en un router



Comandos disponible

- Pulsar “?” en la línea de comandos ...

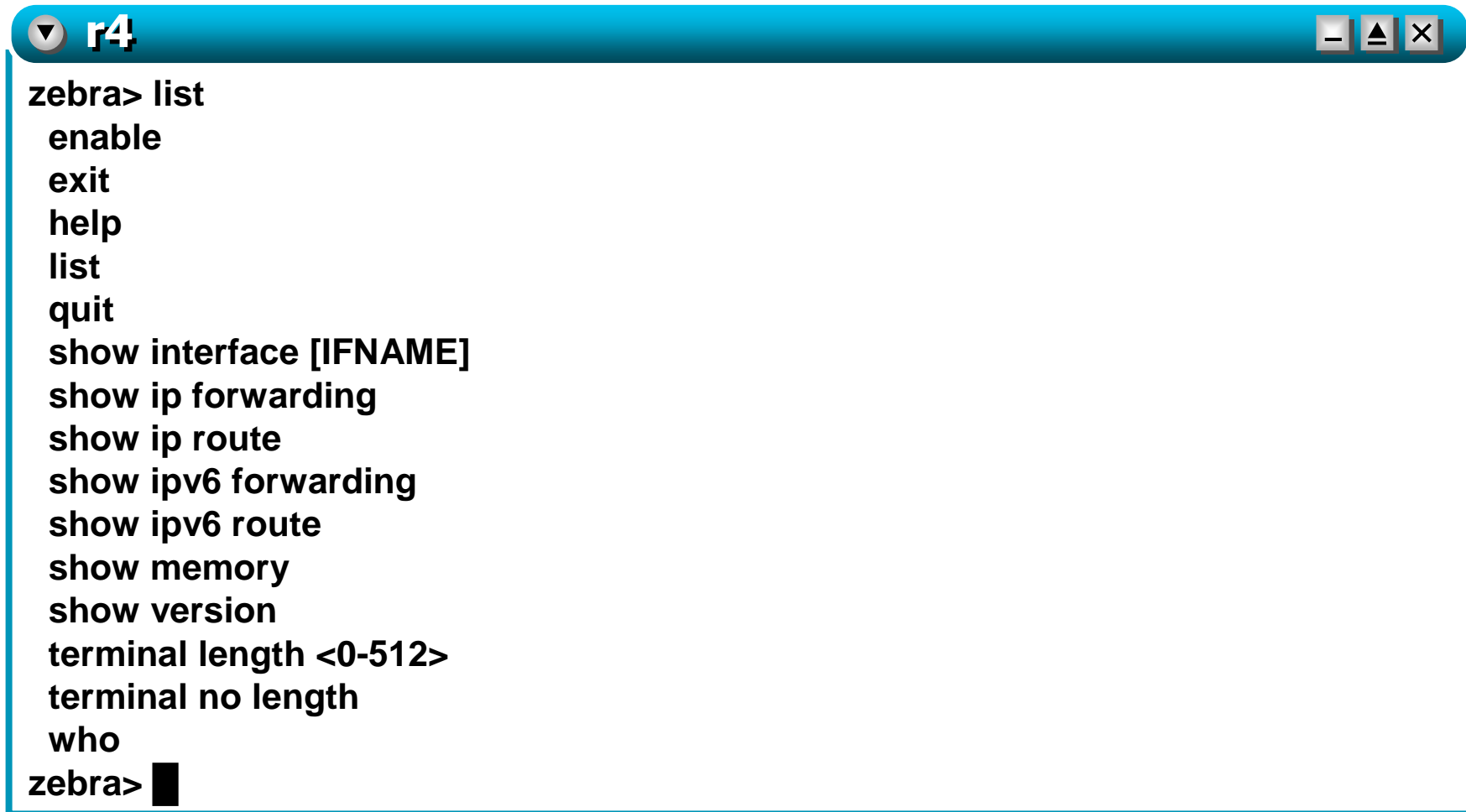


```
zebra>  
enable    Turn on privileged mode command  
exit      Exit current mode and down to previous mode  
help      Description of the interactive help system  
list      Print command list  
quit      Exit current mode and down to previous mode  
show      Show running system information  
terminal  Set terminal line parameters  
who       Display who is on vty  
zebra> █
```

■ ... 0 ...

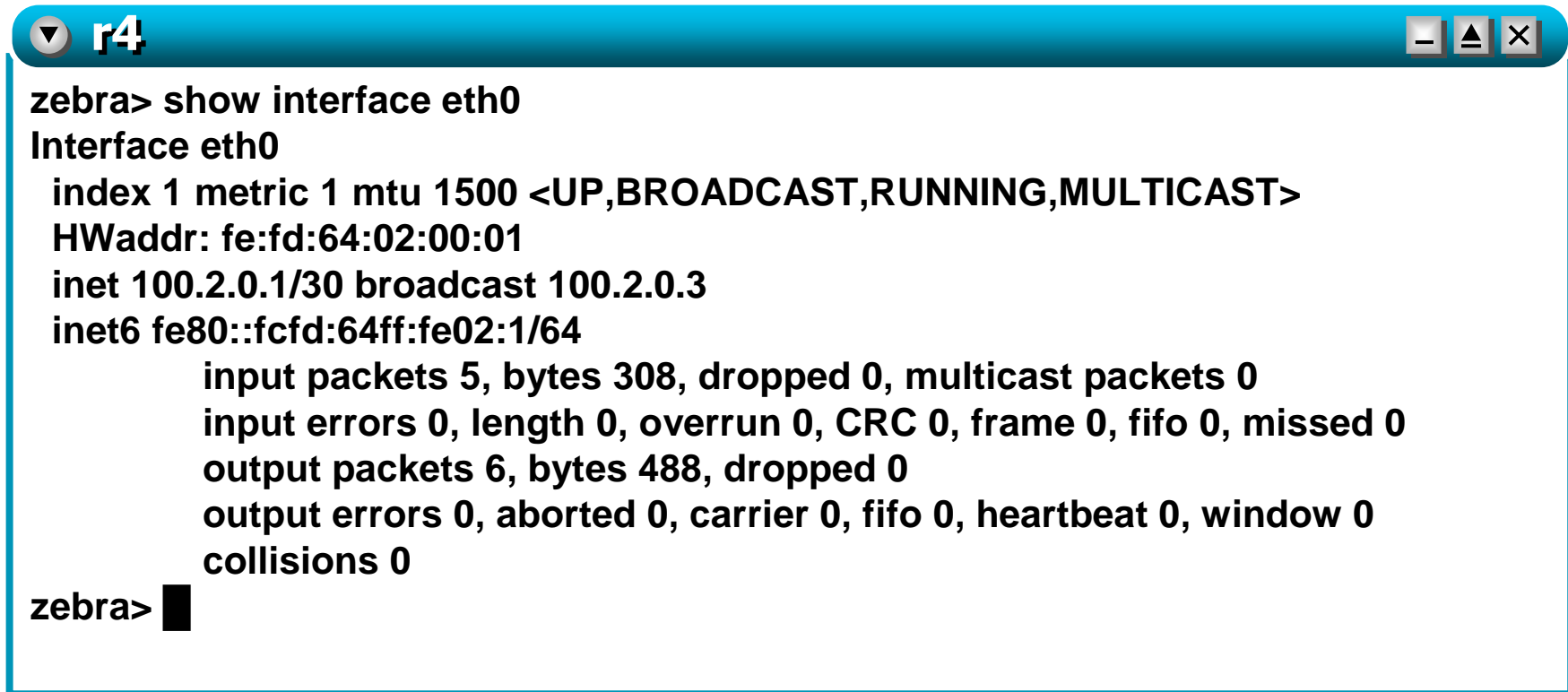
Comandos disponibles

- ... teclee "list" (se muestra un extracto de la salida)



```
▼ r4
zebra> list
enable
exit
help
list
quit
show interface [IFNAME]
show ip forwarding
show ip route
show ipv6 forwarding
show ipv6 route
show memory
show version
terminal length <0-512>
terminal no length
who
zebra> █
```

Revisando los interfaces



```
zebra> show interface eth0
Interface eth0
index 1 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
HWaddr: fe:fd:64:02:00:01
inet 100.2.0.1/30 broadcast 100.2.0.3
inet6 fe80::fcfd:64ff:fe02:1/64
    input packets 5, bytes 308, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 6, bytes 488, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 0
zebra> █
```

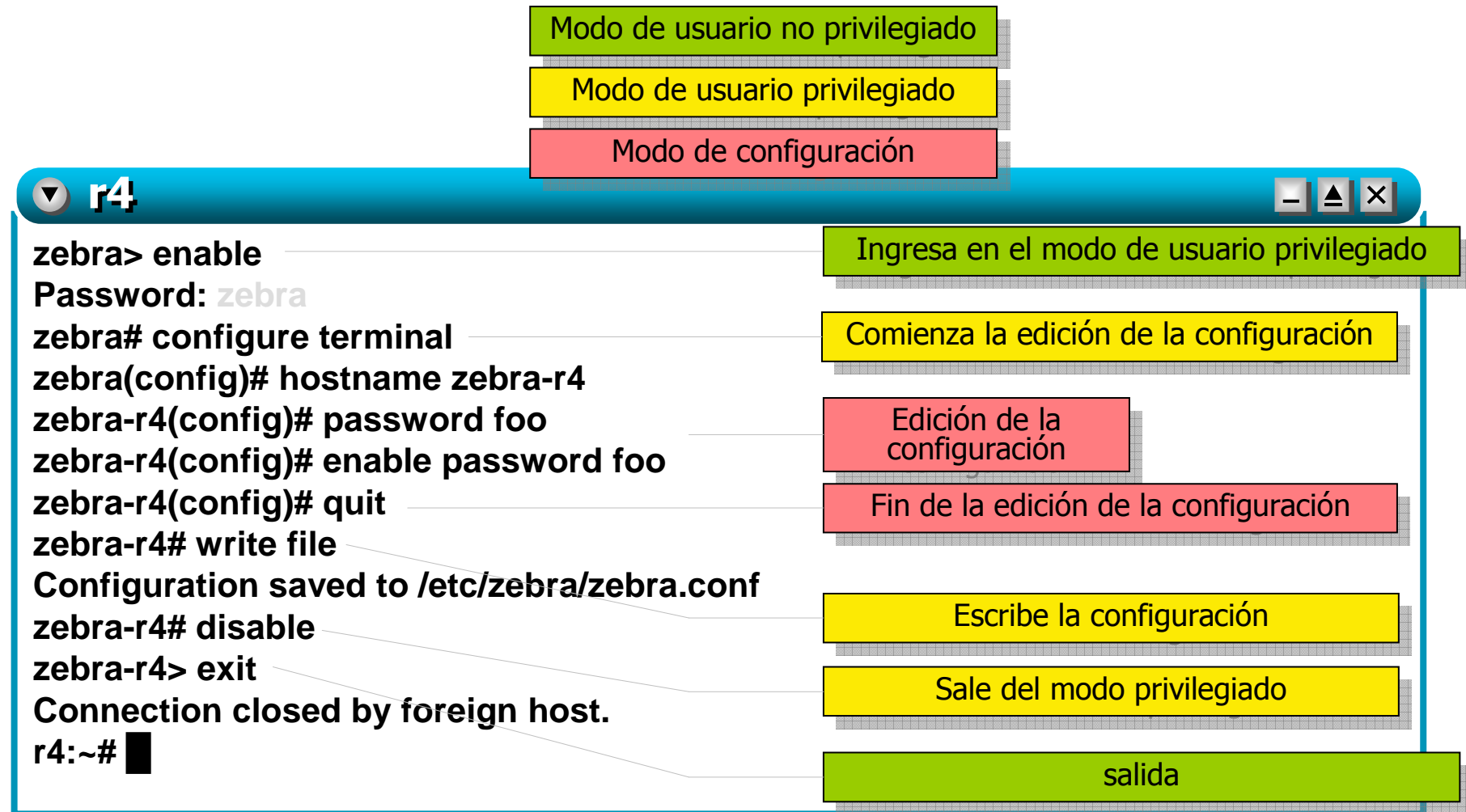
- Esto corresponde aproximadamente a usar `ifconfig` en la línea de comandos

Revisando la tabla de rutas de Zebra

```
▼ r4
zebra> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
      B - BGP, > - selected route, * - FIB route
R>* 100.1.0.0/30 [120/2] via 100.1.0.13, eth3, 01:28:42
R>* 100.1.0.4/30 [120/2] via 100.1.0.17, eth2, 01:28:52
R>* 100.1.0.8/30 [120/2] via 100.1.0.17, eth2, 01:28:52
C>* 100.1.0.12/30 is directly connected, eth3
C>* 100.1.0.16/30 is directly connected, eth2
R>* 100.1.1.0/24 [120/2] via 100.1.0.13, eth3, 01:28:42
R>* 100.1.2.0/24 [120/3] via 100.1.0.17, eth2, 01:28:47
R>* 100.1.3.0/24 [120/2] via 100.1.0.17, eth2, 01:28:52
C>* 100.1.4.0/24 is directly connected, eth1
C>* 100.2.0.0/30 is directly connected, eth0
C>* 127.0.0.0/8 is directly connected, lo
zebra> █
```

- La entradas FIB de esta tabla (marcadas con ">") son inyectadas en la tabla de rutas del kernel.

Cambiando la configuración de Zebra



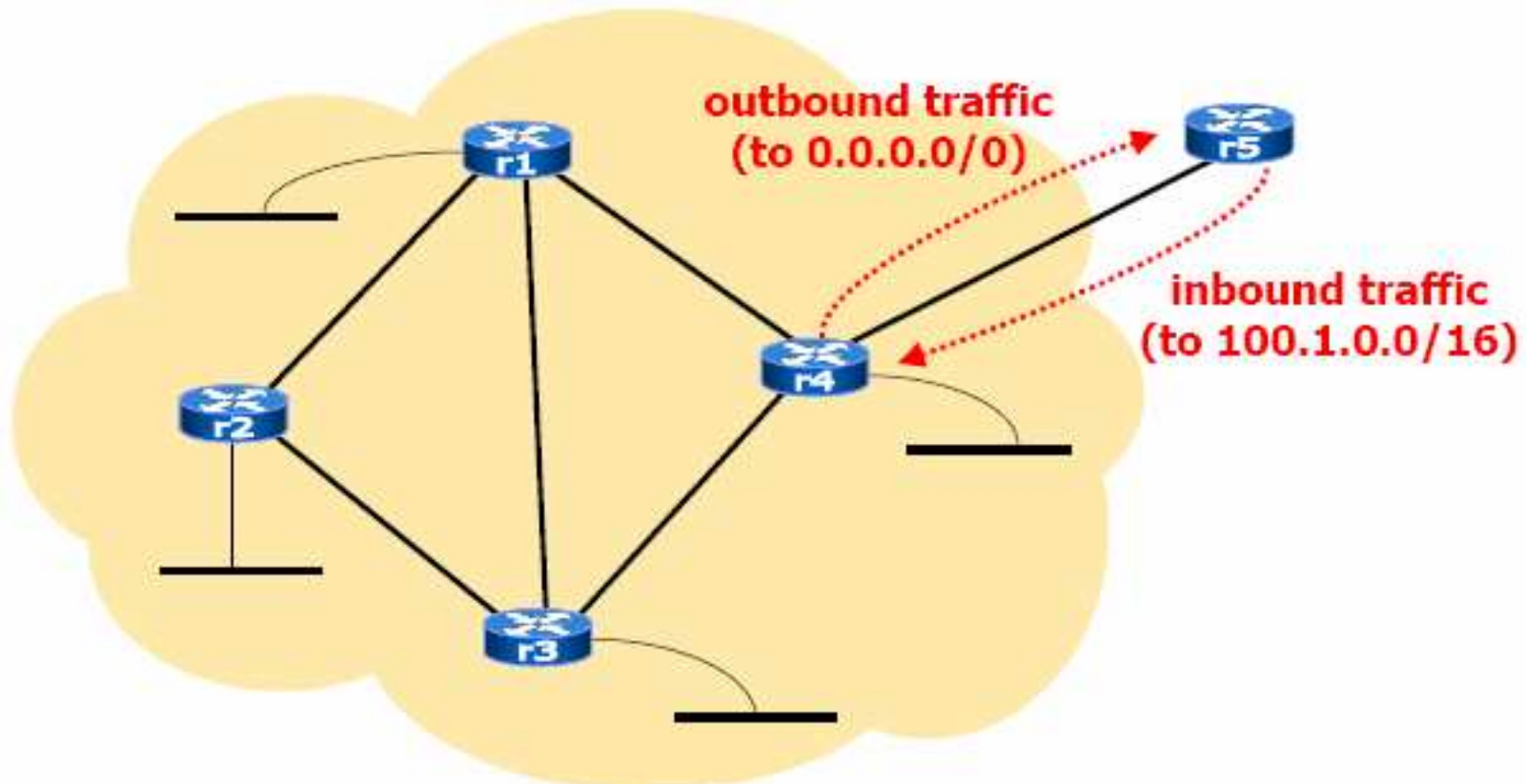
Revisando la tabla de rutas de RIP

```
▼ r4
r4:~# telnet localhost ripd
.....
Password: zebra
ripd> show ip rip
Codes: R - RIP, C - connected, O - OSPF, B - BGP
       (n) - normal, (s) - static, (d) - default, (r) - redistribute,
       (i) - interface

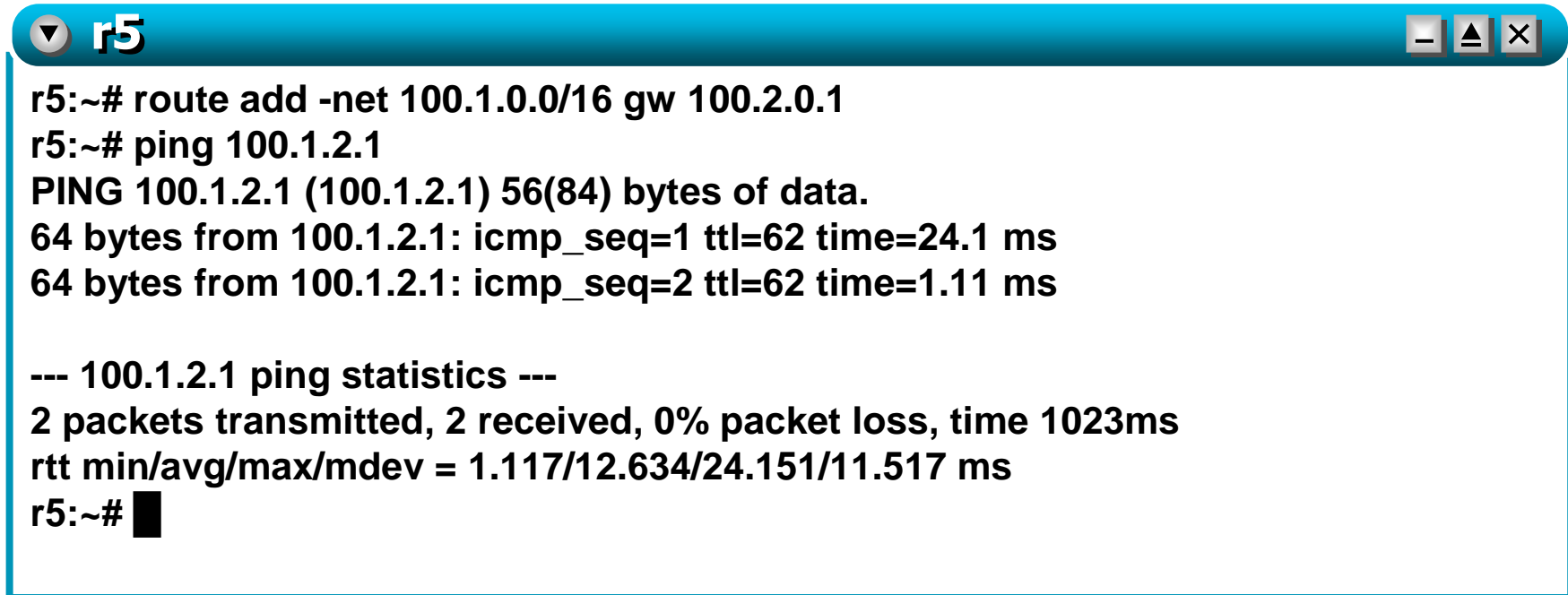
      Network Next Hop      Metric From      Time
R(n) 100.1.0.0/30 100.1.0.13      2 100.1.0.13     02:43
R(n) 100.1.0.4/30 100.1.0.17      2 100.1.0.17     02:46
R(n) 100.1.0.8/30 100.1.0.17      2 100.1.0.17     02:46
C(i) 100.1.0.12/30 0.0.0.0         1 self
C(i) 100.1.0.16/30 0.0.0.0         1 self
R(n) 100.1.1.0/24 100.1.0.13      2 100.1.0.13     02:43
R(n) 100.1.2.0/24 100.1.0.17      3 100.1.0.17     02:46
R(n) 100.1.3.0/24 100.1.0.17      2 100.1.0.17     02:46
C(i) 100.1.4.0/24 0.0.0.0         1 self
C(r) 100.2.0.0/30 0.0.0.0         1 self
ripd> █
```

Routing estático

- Nuestra red es una **red Stub** (p.e. sólo tiene una conexión a un router externo, r5); entonces, las rutas estáticas son suficientes para conectarse a Internet.



Añadiendo una ruta estática hacia r5



```
r5:~# route add -net 100.1.0.0/16 gw 100.2.0.1
r5:~# ping 100.1.2.1
PING 100.1.2.1 (100.1.2.1) 56(84) bytes of data.
64 bytes from 100.1.2.1: icmp_seq=1 ttl=62 time=24.1 ms
64 bytes from 100.1.2.1: icmp_seq=2 ttl=62 time=1.11 ms

--- 100.1.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1023ms
rtt min/avg/max/mdev = 1.117/12.634/24.151/11.517 ms
r5:~#
```

Comprobando la conectividad

▼ r5

```
r5:~# traceroute 100.1.2.1
```

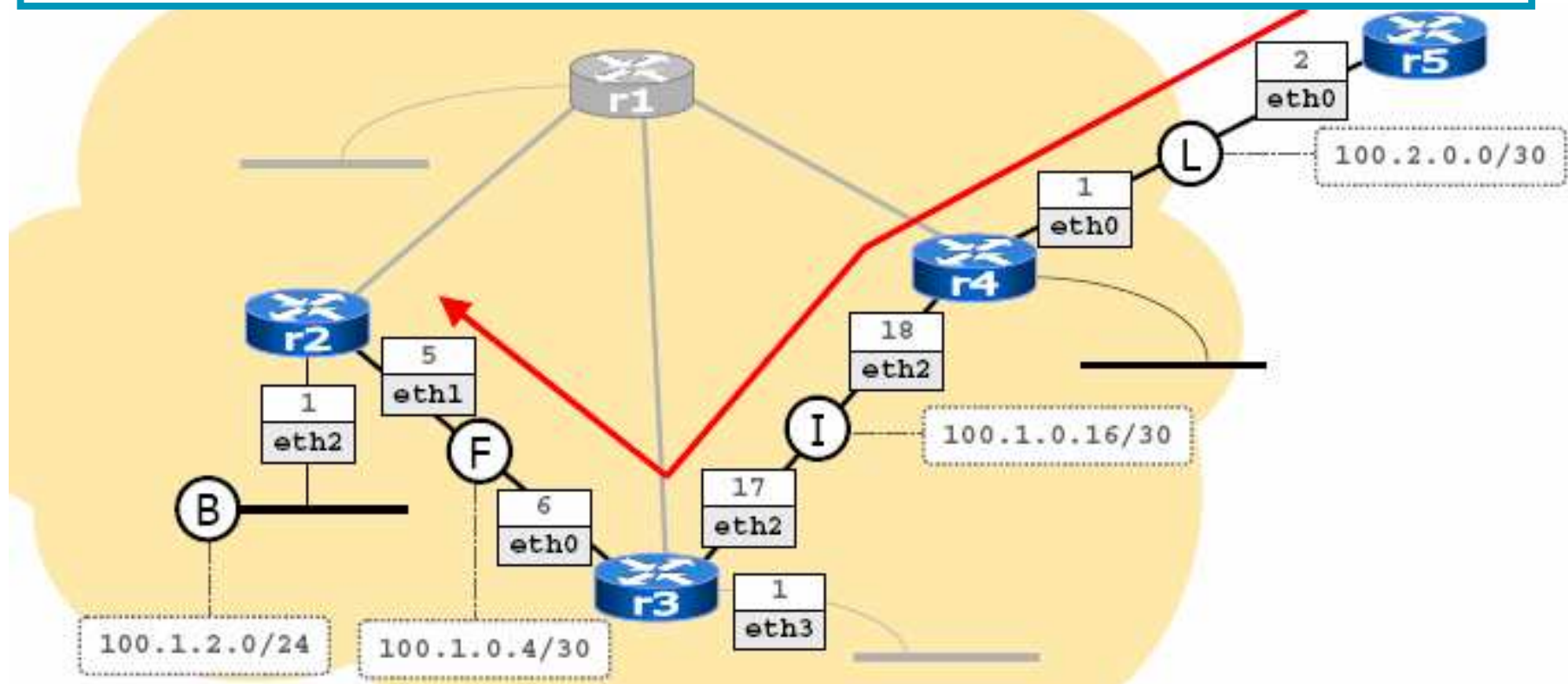
```
traceroute to 100.1.2.1 (100.1.2.1), 64 hops max, 40 byte packets
```

```
1 100.2.0.1 (100.2.0.1) 75 ms 1 ms 2 ms
```

```
2 100.1.0.17 (100.1.0.17) 7 ms 1 ms 1 ms
```

```
3 100.1.2.1 (100.1.2.1) 24 ms 3 ms 1 ms
```

```
r5:~# █
```



Configurando r4

■ Paso 1: configurando la ruta por defecto

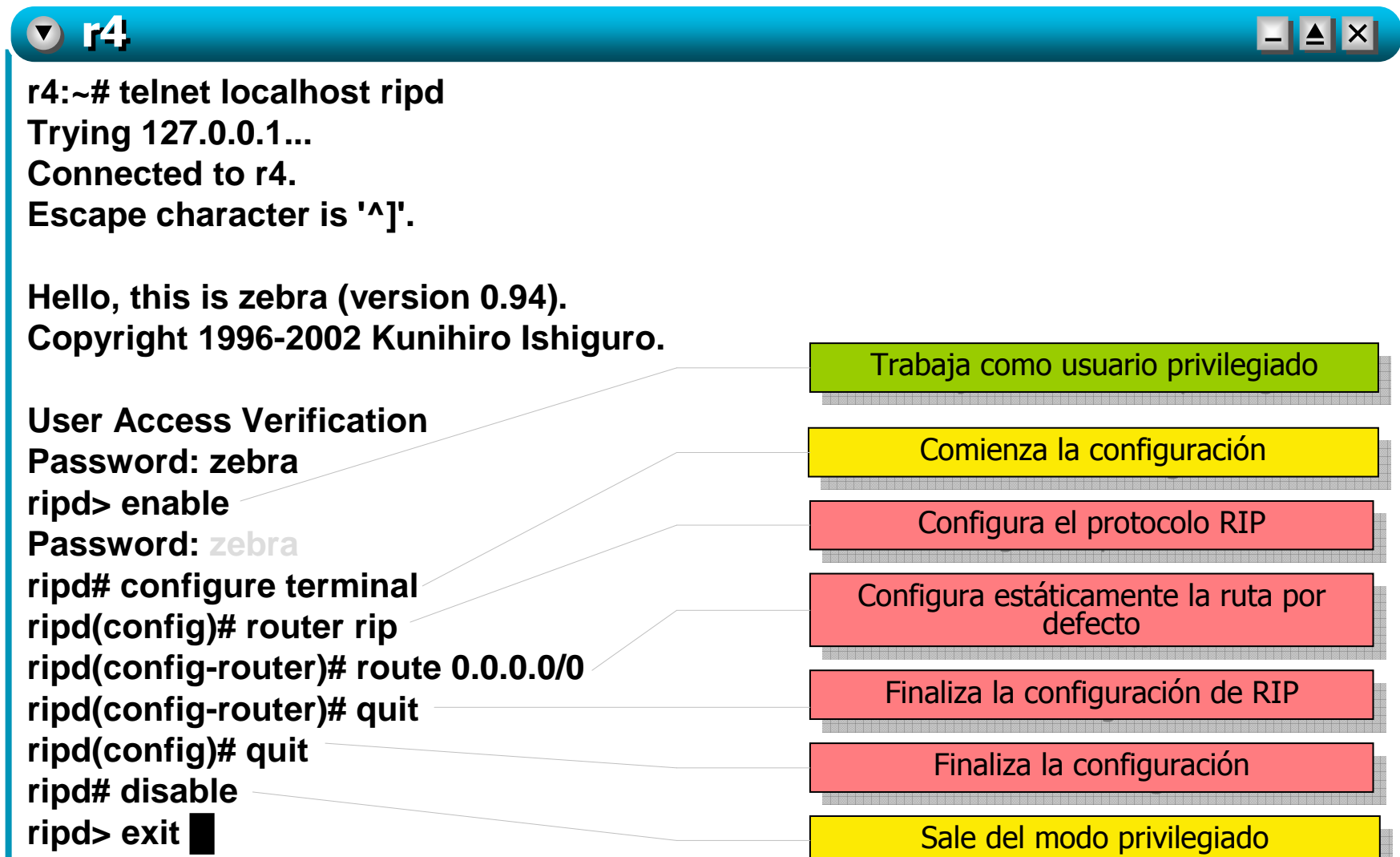
```
▼ r4
r4:~# route add default gw 100.2.0.2
r4:~# route
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
100.1.0.16	*	255.255.255.252	U	0	0	0	eth2
100.1.0.0	100.1.0.13	255.255.255.252	UG	2	0	0	eth3
100.1.0.4	100.1.0.17	255.255.255.252	UG	2	0	0	eth2
100.2.0.0	*	255.255.255.252	U	0	0	0	eth0
100.1.0.8	100.1.0.17	255.255.255.252	UG	2	0	0	eth2
100.1.0.12	*	255.255.255.252	U	0	0	0	eth3
100.1.4.0	*	255.255.255.0	U	0	0	0	eth1
100.1.2.0	100.1.0.17	255.255.255.0	UG	3	0	0	eth2
100.1.3.0	100.1.0.17	255.255.255.0	UG	2	0	0	eth2
100.1.1.0	100.1.0.13	255.255.255.0	UG	2	0	0	eth3
default	100.2.0.2	0.0.0.0	UG	0	0	0	eth0

```
r4:~# █
```

Configurando r4

■ Paso 2: propagando la ruta por defecto en RIP



```
r4:~# telnet localhost ripd
Trying 127.0.0.1...
Connected to r4.
Escape character is '^]'.

Hello, this is zebra (version 0.94).
Copyright 1996-2002 Kunihiro Ishiguro.

User Access Verification
Password: zebra
ripd> enable
Password: zebra
ripd# configure terminal
ripd(config)# router rip
ripd(config-router)# route 0.0.0.0/0
ripd(config-router)# quit
ripd(config)# quit
ripd# disable
ripd> exit
```

Trabaja como usuario privilegiado

Comienza la configuración

Configura el protocolo RIP

Configura estáticamente la ruta por defecto

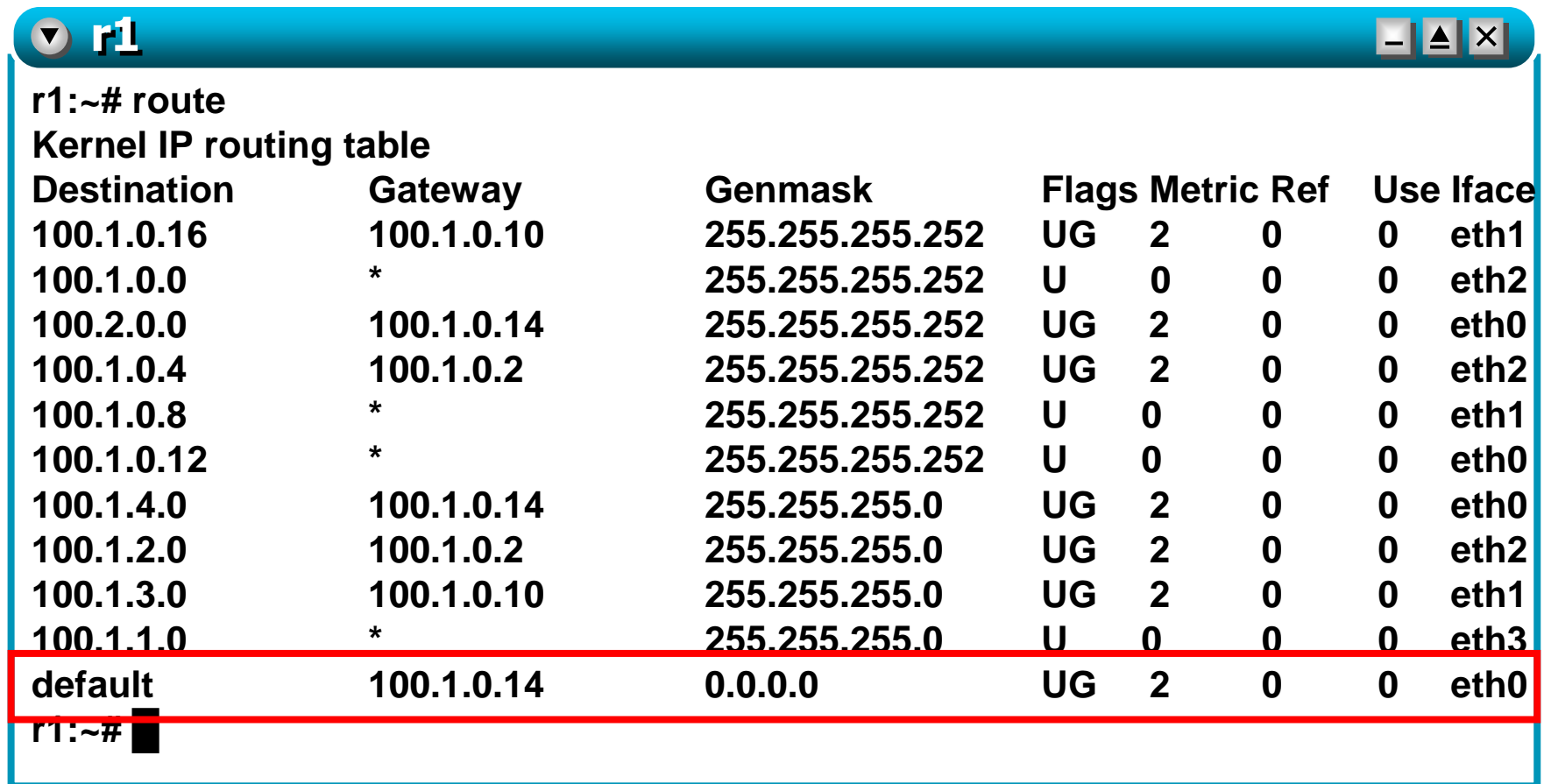
Finaliza la configuración de RIP

Finaliza la configuración

Sale del modo privilegiado

La ruta por defecto

- Después de un momento, la ruta por defecto ha sido inyectada (vía RIP) en la red



```
r1:~# route
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
100.1.0.16	100.1.0.10	255.255.255.252	UG	2	0	0	eth1
100.1.0.0	*	255.255.255.252	U	0	0	0	eth2
100.2.0.0	100.1.0.14	255.255.255.252	UG	2	0	0	eth0
100.1.0.4	100.1.0.2	255.255.255.252	UG	2	0	0	eth2
100.1.0.8	*	255.255.255.252	U	0	0	0	eth1
100.1.0.12	*	255.255.255.252	U	0	0	0	eth0
100.1.4.0	100.1.0.14	255.255.255.0	UG	2	0	0	eth0
100.1.2.0	100.1.0.2	255.255.255.0	UG	2	0	0	eth2
100.1.3.0	100.1.0.10	255.255.255.0	UG	2	0	0	eth1
100.1.1.0	*	255.255.255.0	U	0	0	0	eth3
default	100.1.0.14	0.0.0.0	UG	2	0	0	eth0

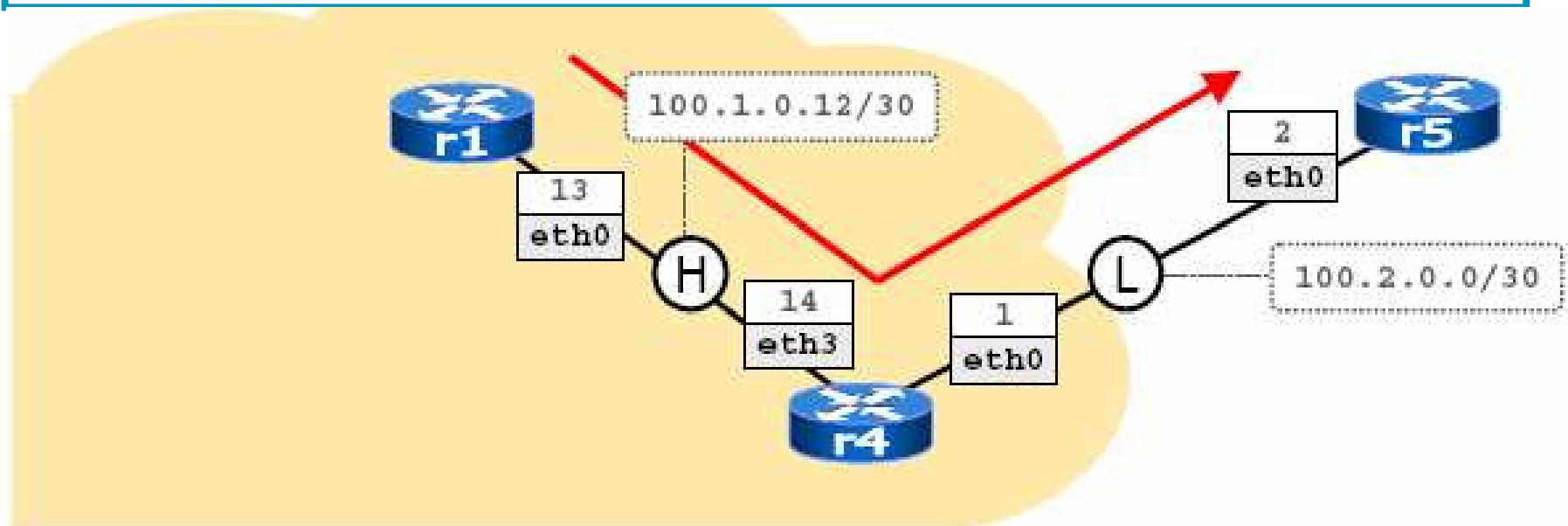
```
r1:~#
```

Comprobando la conectividad

▼ r1

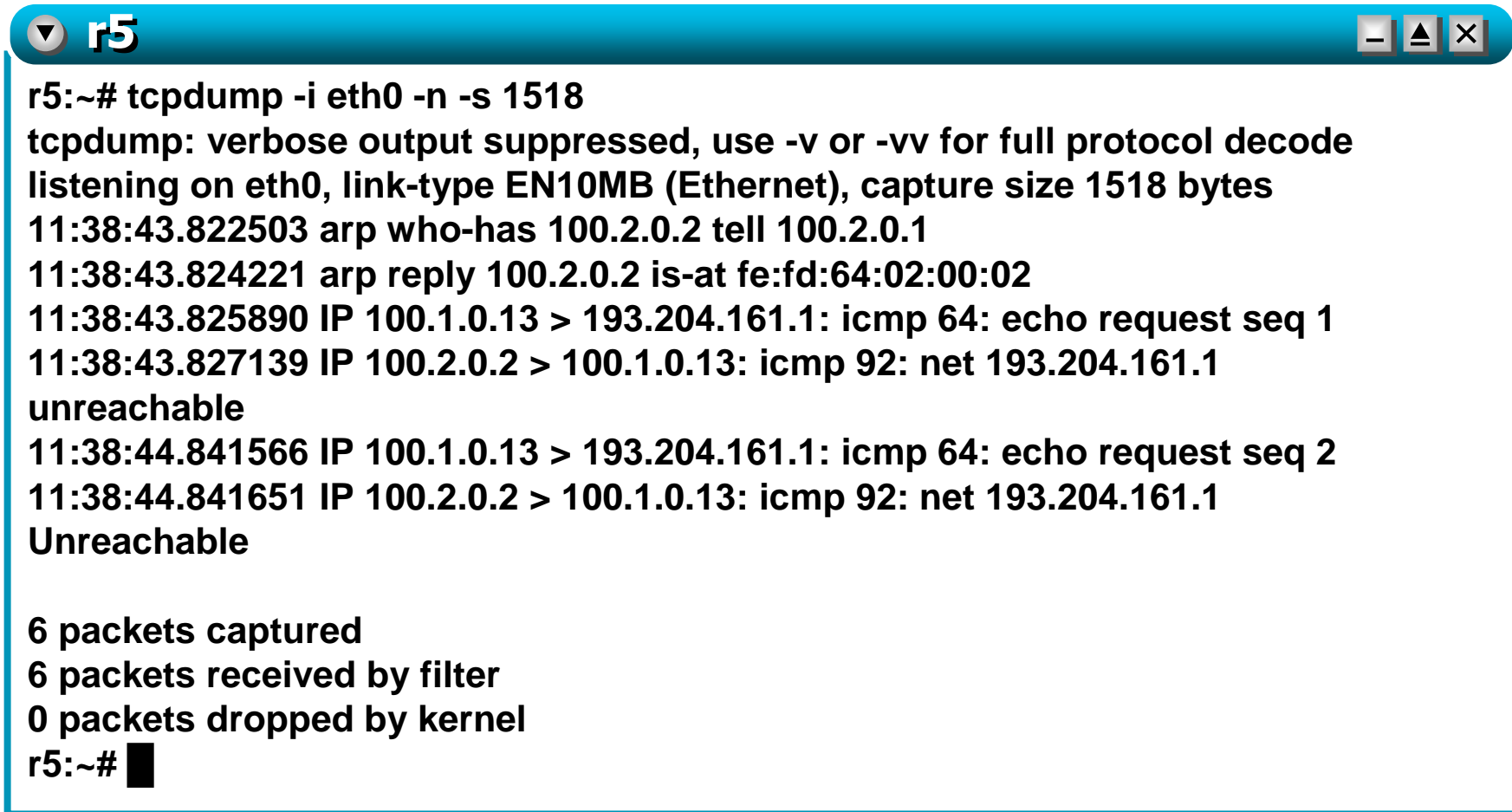
Cualquier destino (incluso no existente)

```
r1:~# ping 193.204.161.1
PING 193.204.161.1 (193.204.161.1) 56(84) bytes of data.
From 100.2.0.2 icmp_seq=1 Destination Net Unreachable
From 100.2.0.2 icmp_seq=2 Destination Net Unreachable
--- 193.204.161.1 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss,
time 999ms
r1:~# █
```



Comprobando la conectividad

- r5 está recibiendo paquetes de echo request



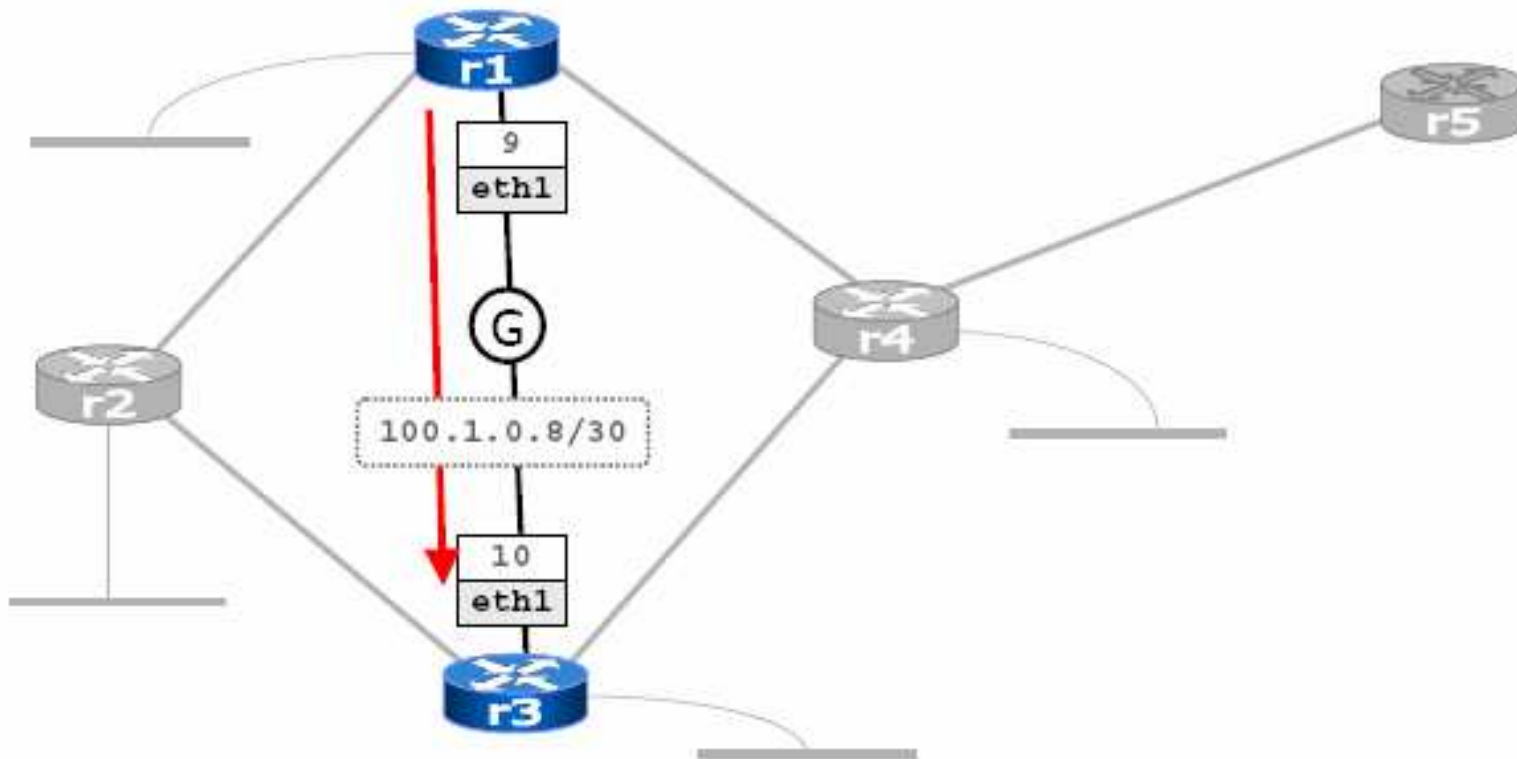
```
r5:~# tcpdump -i eth0 -n -s 1518
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 1518 bytes
11:38:43.822503 arp who-has 100.2.0.2 tell 100.2.0.1
11:38:43.824221 arp reply 100.2.0.2 is-at fe:fd:64:02:00:02
11:38:43.825890 IP 100.1.0.13 > 193.204.161.1: icmp 64: echo request seq 1
11:38:43.827139 IP 100.2.0.2 > 100.1.0.13: icmp 92: net 193.204.161.1
unreachable
11:38:44.841566 IP 100.1.0.13 > 193.204.161.1: icmp 64: echo request seq 2
11:38:44.841651 IP 100.2.0.2 > 100.1.0.13: icmp 92: net 193.204.161.1
Unreachable

6 packets captured
6 packets received by filter
0 packets dropped by kernel
r5:~#
```

Tirando un interfaz

▼ r1

```
r1:~# traceroute 100.1.0.10
traceroute to 100.1.0.10 (100.1.0.10), 64 hops max, 40 byte packets
1 100.1.0.10 (100.1.0.10) 24 ms 1 ms 1 ms
r1:~# ifconfig eth1 down
```



Tirando un interfaz

▼ r1

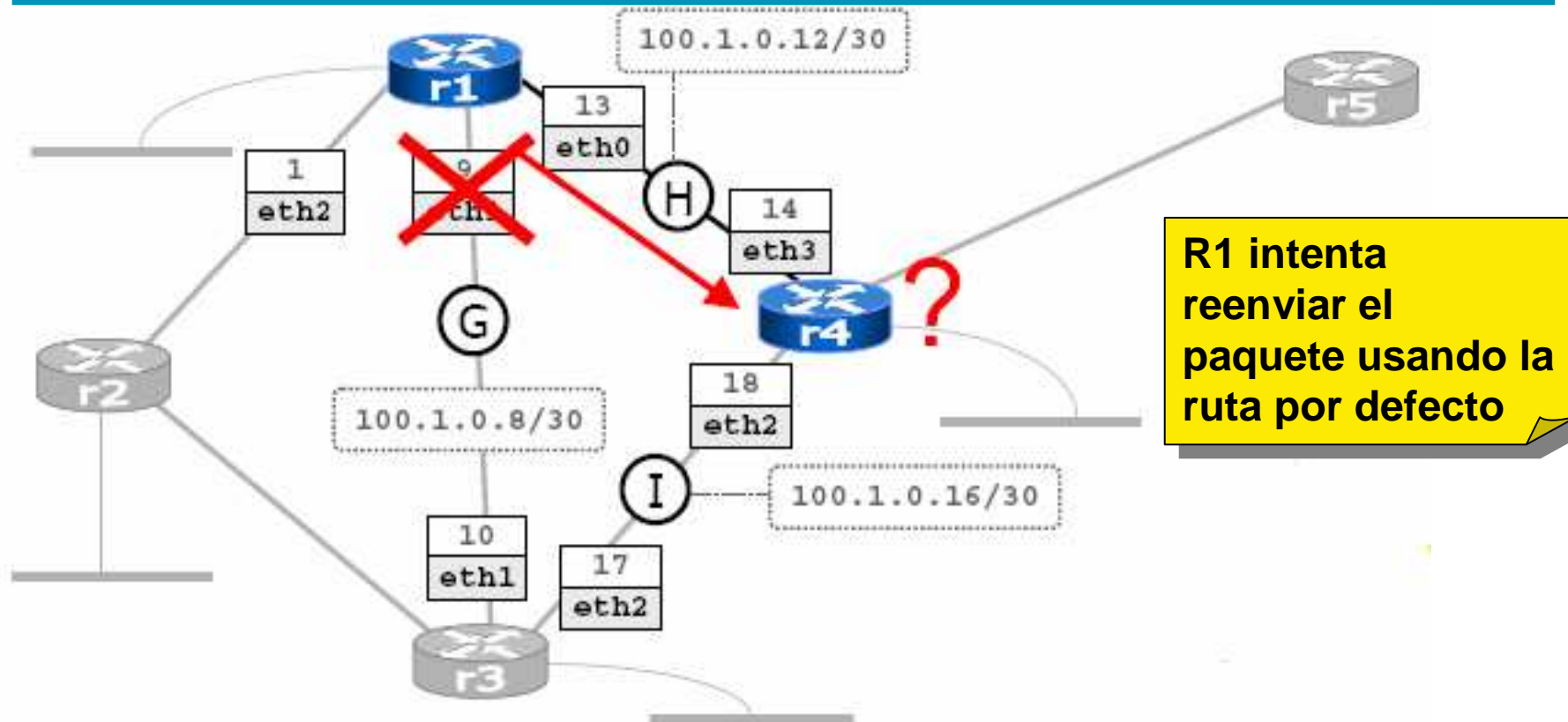
```
r1:~# traceroute 100.1.0.10
```

```
traceroute to 100.1.0.10 (100.1.0.10), 64 hops max, 40 byte packets
```

```
1 100.1.0.14 (100.1.0.14) 1 ms 1 ms 1 ms
```

```
2 * * *
```

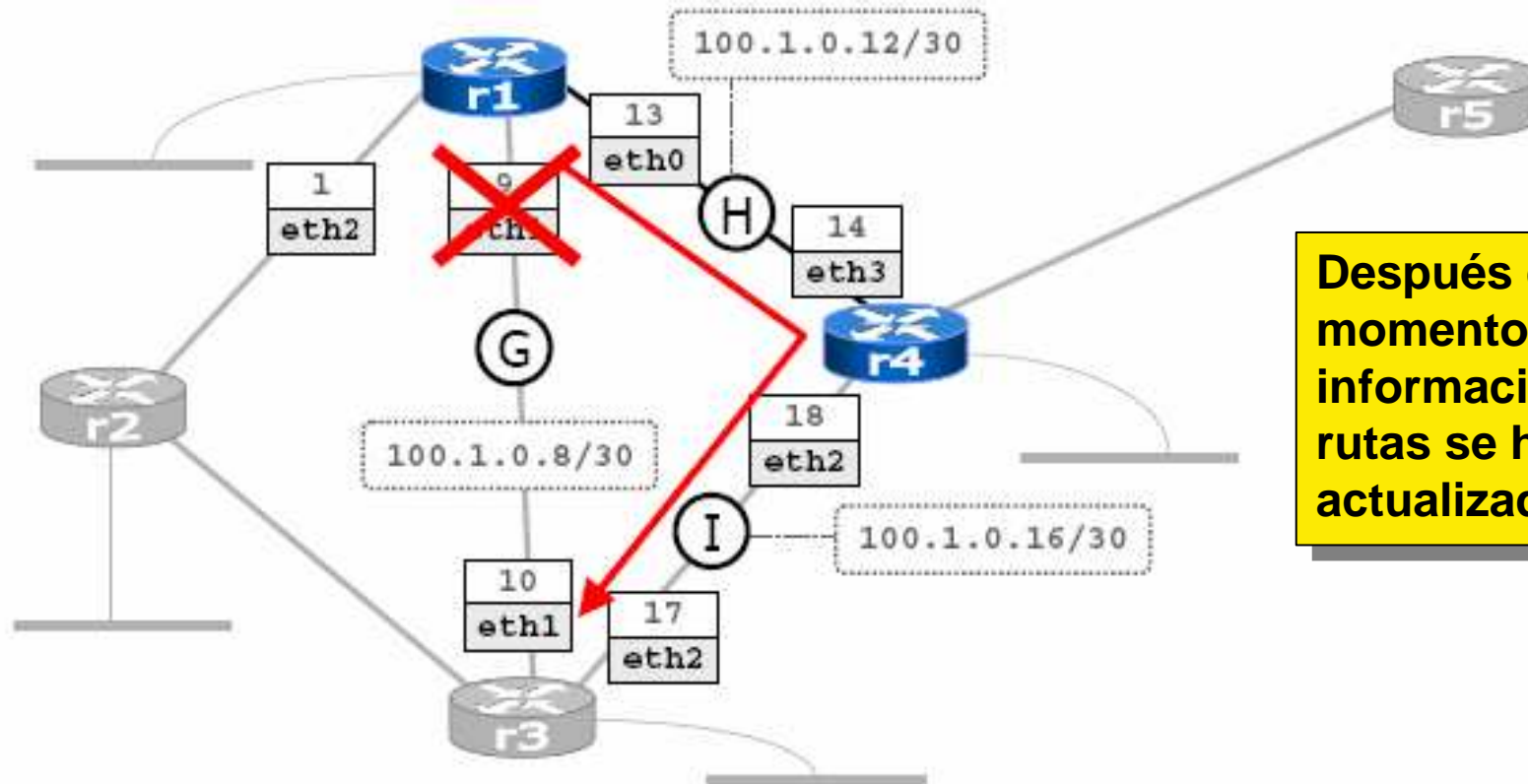
```
3 * * * ■
```



Tirando un interfaz

▼ r1

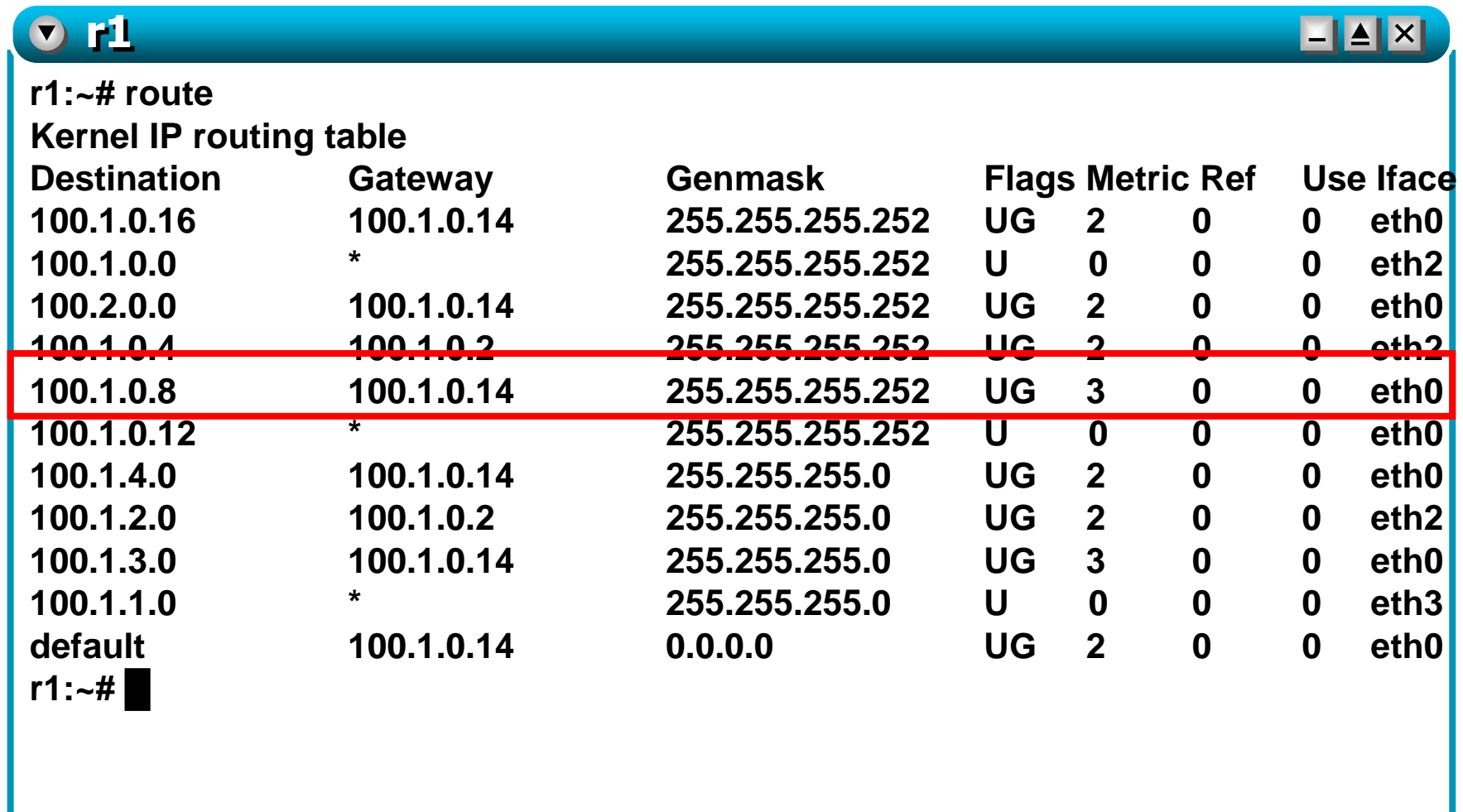
```
r1:~# traceroute 100.1.0.10
traceroute to 100.1.0.10 (100.1.0.10), 64 hops max, 40 byte packets
1 100.1.0.14 (100.1.0.14) 1 ms 1 ms 1 ms
2 100.1.0.10 (100.1.0.10) 5 ms 2 ms 1 ms
r1:~# █
```



Después de un momento, la información de rutas se ha actualizado

Tirando un interfaz

- La tabla de rutas de r1 s ha actualizado



r1:~# route

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
100.1.0.16	100.1.0.14	255.255.255.252	UG	2	0	0	eth0
100.1.0.0	*	255.255.255.252	U	0	0	0	eth2
100.2.0.0	100.1.0.14	255.255.255.252	UG	2	0	0	eth0
100.1.0.4	100.1.0.2	255.255.255.252	UG	2	0	0	eth2
100.1.0.8	100.1.0.14	255.255.255.252	UG	3	0	0	eth0
100.1.0.12	*	255.255.255.252	U	0	0	0	eth0
100.1.4.0	100.1.0.14	255.255.255.0	UG	2	0	0	eth0
100.1.2.0	100.1.0.2	255.255.255.0	UG	2	0	0	eth2
100.1.3.0	100.1.0.14	255.255.255.0	UG	3	0	0	eth0
100.1.1.0	*	255.255.255.0	U	0	0	0	eth3
default	100.1.0.14	0.0.0.0	UG	2	0	0	eth0

r1:~#