

# kathara lab

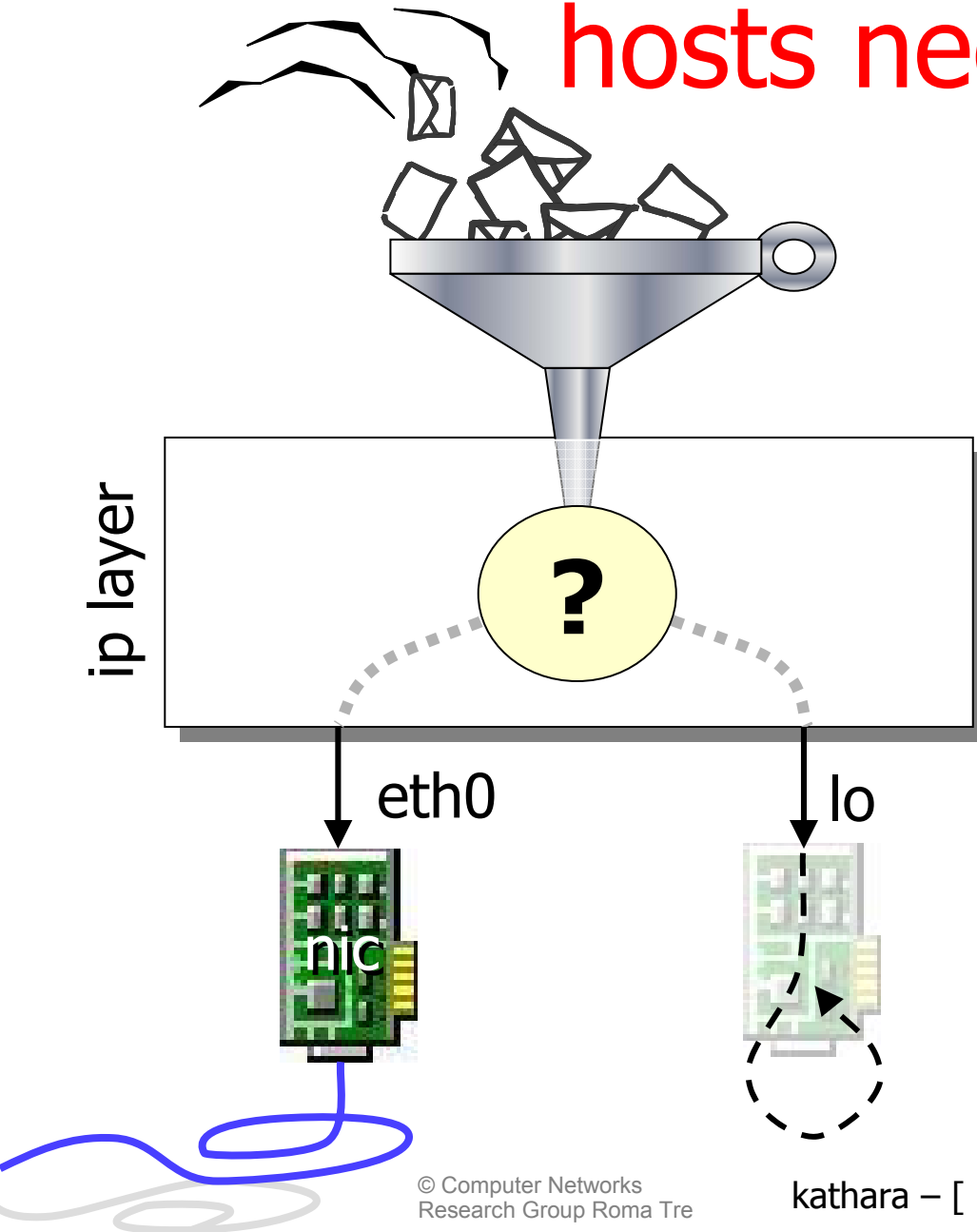
## zebra/quagga

<b>Version</b>	1.0
<b>Author(s)</b>	G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini
<b>E-mail</b>	contact@kathara.org
<b>Web</b>	<a href="http://www.kathara.org/">http://www.kathara.org/</a>
<b>Description</b>	experiences with zebra/quagga configurations and command line interface

# copyright notice

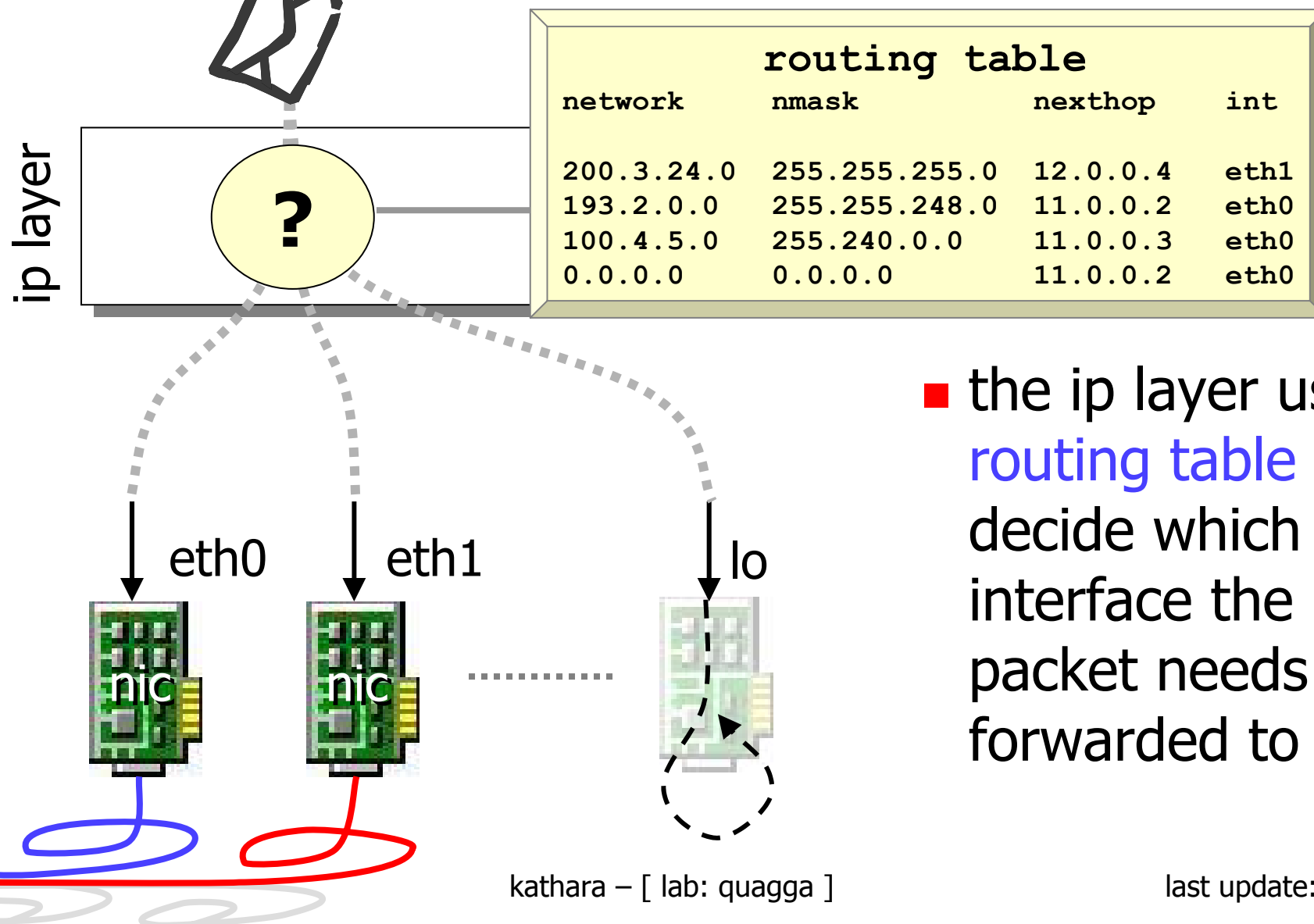
- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property 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.

# hosts need routing

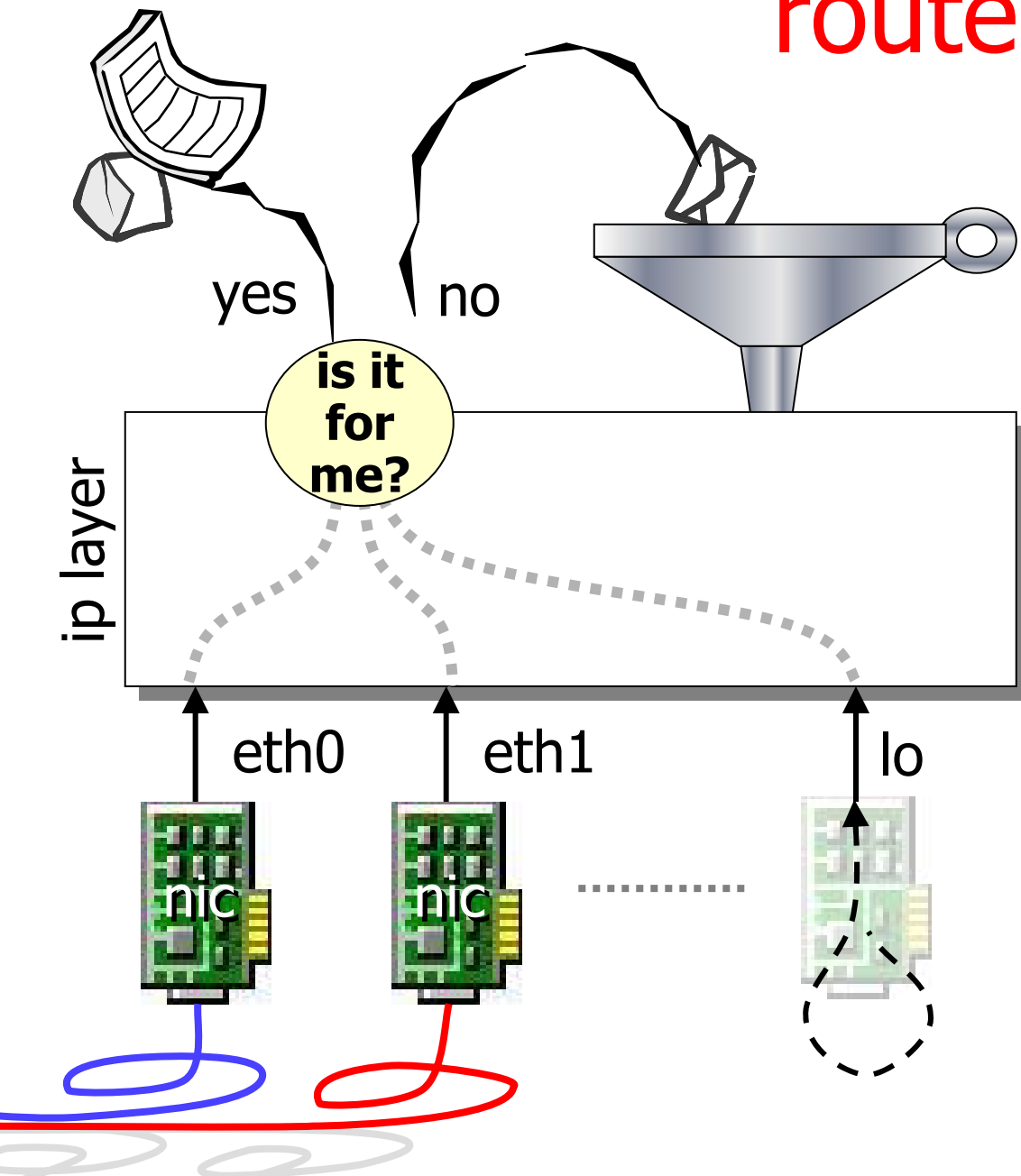


- each host with a network stack performs some elementary routing
- at the very least, the network stack may be used to access local services (e.g., Xorg)
- the host must decide when a packet needs to be sent to the network interface card (nic) and when it needs to be bounced to the loopback interface (lo)

# routing table



# routers



- a **router** (also called **gateway** or **intermediate-system**)

- has more than one network interface card
- feeds incoming ip packets (that are not for the router itself) back in the routing process
  - this operation is called **relaying** or **forwarding**

# routing protocols

- routing protocols are used to automatically update routing tables, relieving administrators from the need to do it manually
- routers (i.e., devices that run routing protocols) in netkit are virtual machines that run a specific piece of software that implements routing protocols



zebra/quagga

# about zebra/quagga

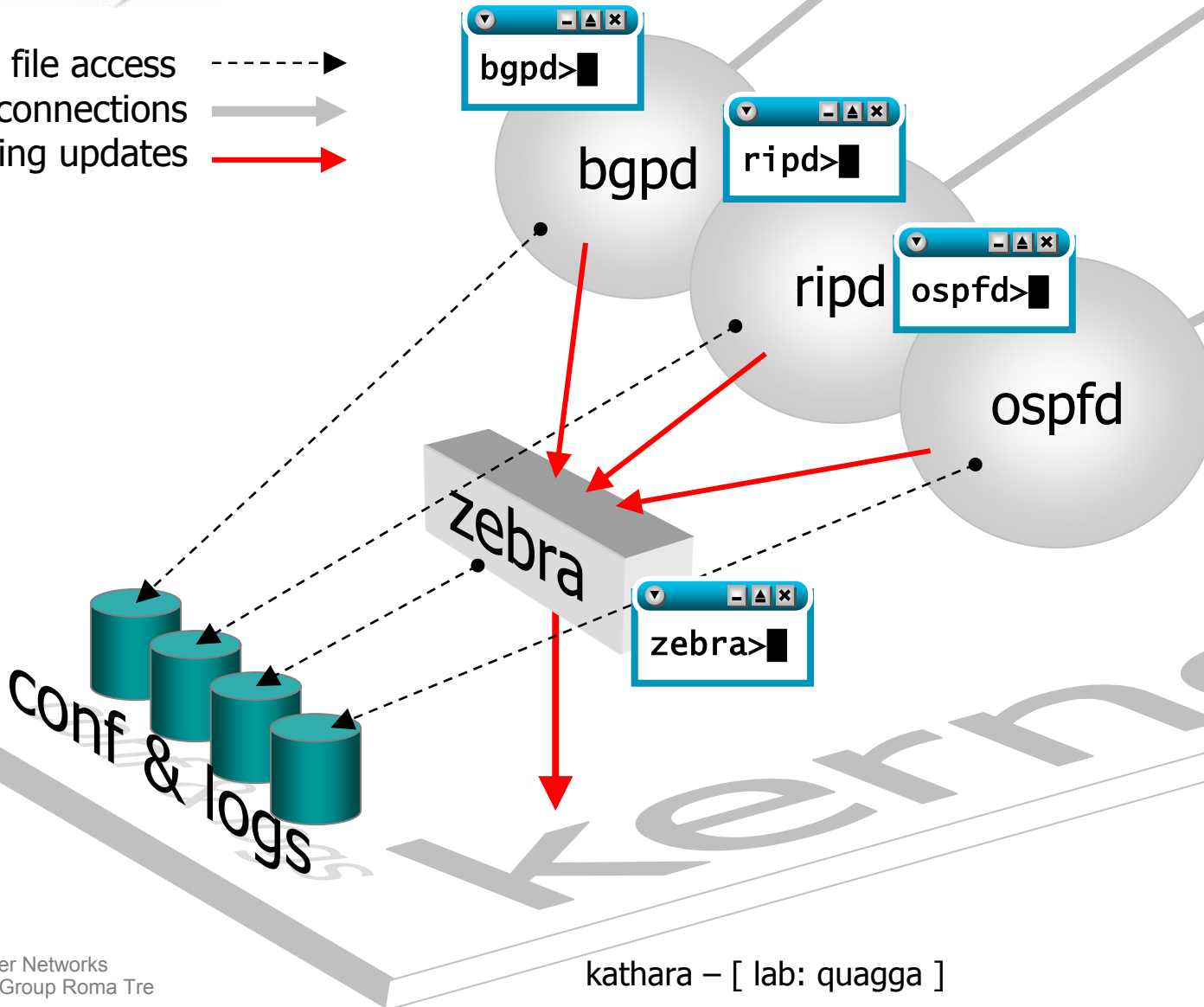
Quagga Routing Software Suite,  
GPL licensed IPv4/IPv6 routing software

- a software that implements several routing protocols
  - rip (v1 and v2)
  - ospf (v2 and v3)
  - is-is
  - bgp
- quagga: “a fork of GNU Zebra [that] aims to build a more involved community around Quagga than the current centralised model of GNU Zebra”
- zebra development stopped at release 0.95a
- quagga superseded zebra
  - in most cases, in netkit you can equivalently refer to “quagga” or “zebra”



# zebra: a routing daemon

file access      
connections      
routing updates    





# inspecting quagga configuration files



```
virtual machine
pc1:~# cd /etc/quagga/
pc1:/etc/quagga# ls
daemons      debian.conf   zebra.conf
pc1:/etc/quagga#
```

- when quagga is started, each daemon checks these files to read the starting configuration

# sample daemons file

## virtual machine

```
pc1:/etc/quagga# 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/quagga/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)
```

the zebra main daemon will be started

the rip daemon will be started too

# sample zebra configuration file (zebra.conf)

virtual machine

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

the prompt of the zebra interface

the password to connect to the daemon

administrative password

# sample ripd configuration file (`ripd.conf`)

virtual machine

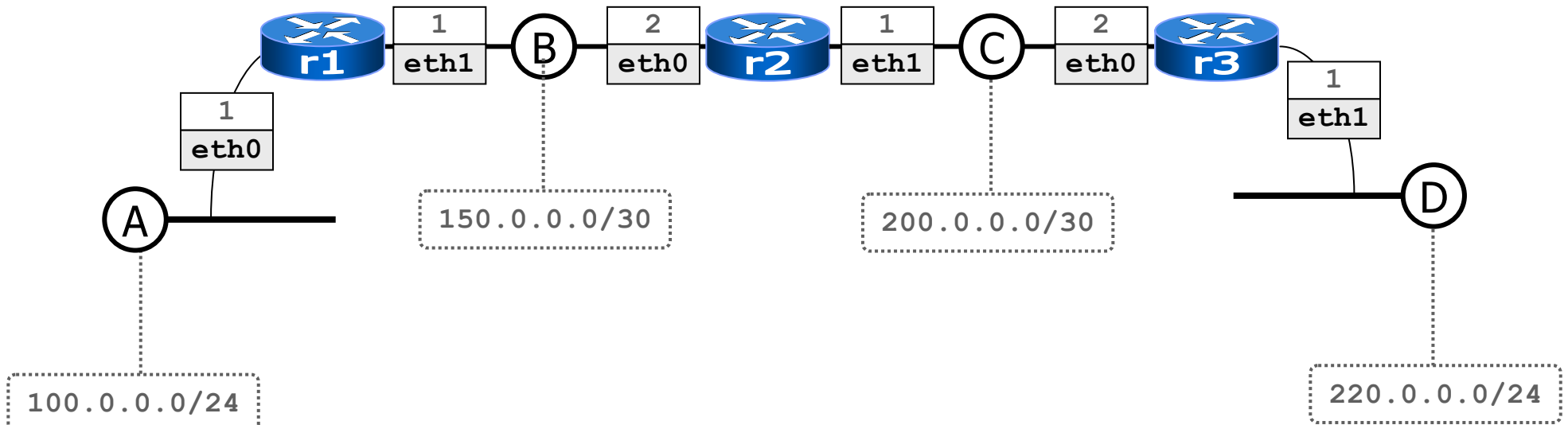
```
pc1:/etc/quagga# cat ripd.conf
!  
hostname ripd  
password zebra  
enable password zebra  
!  
router rip  
redistribute connected  
network 100.1.0.0/16  
!  
log file /var/log/zebra/ripd.log  
pc1:/etc/quagga#
```

talk rip on some interface

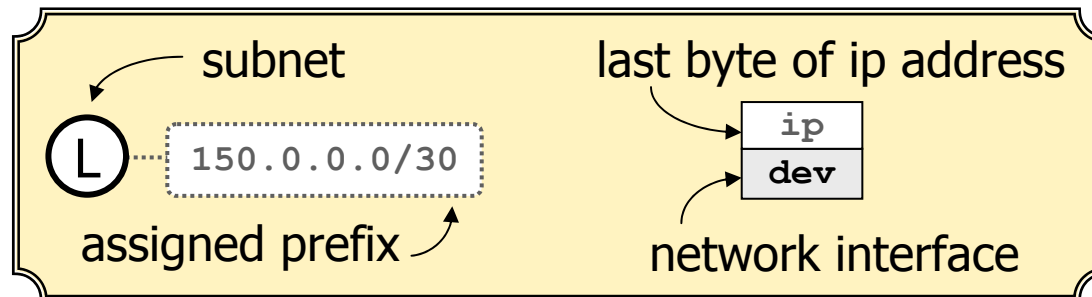
redistribute to rip neighbors  
information about all  
directly connected subnets

send rip multicast  
packets to  
interfaces falling  
into this prefix

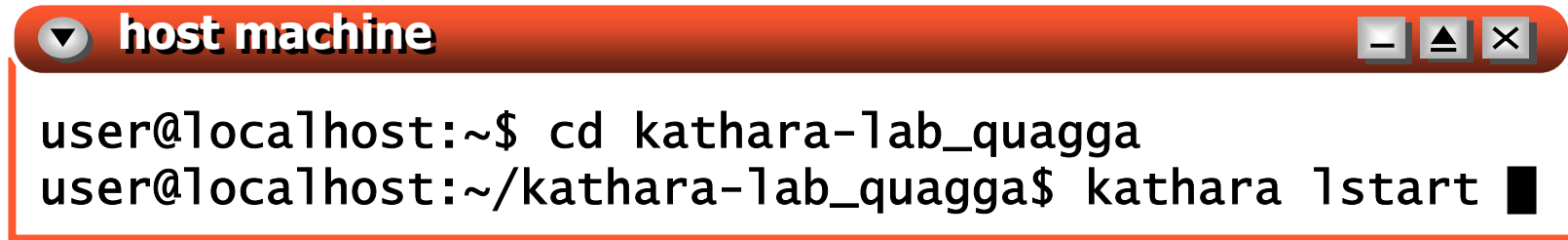
# a simple topology



## Legend



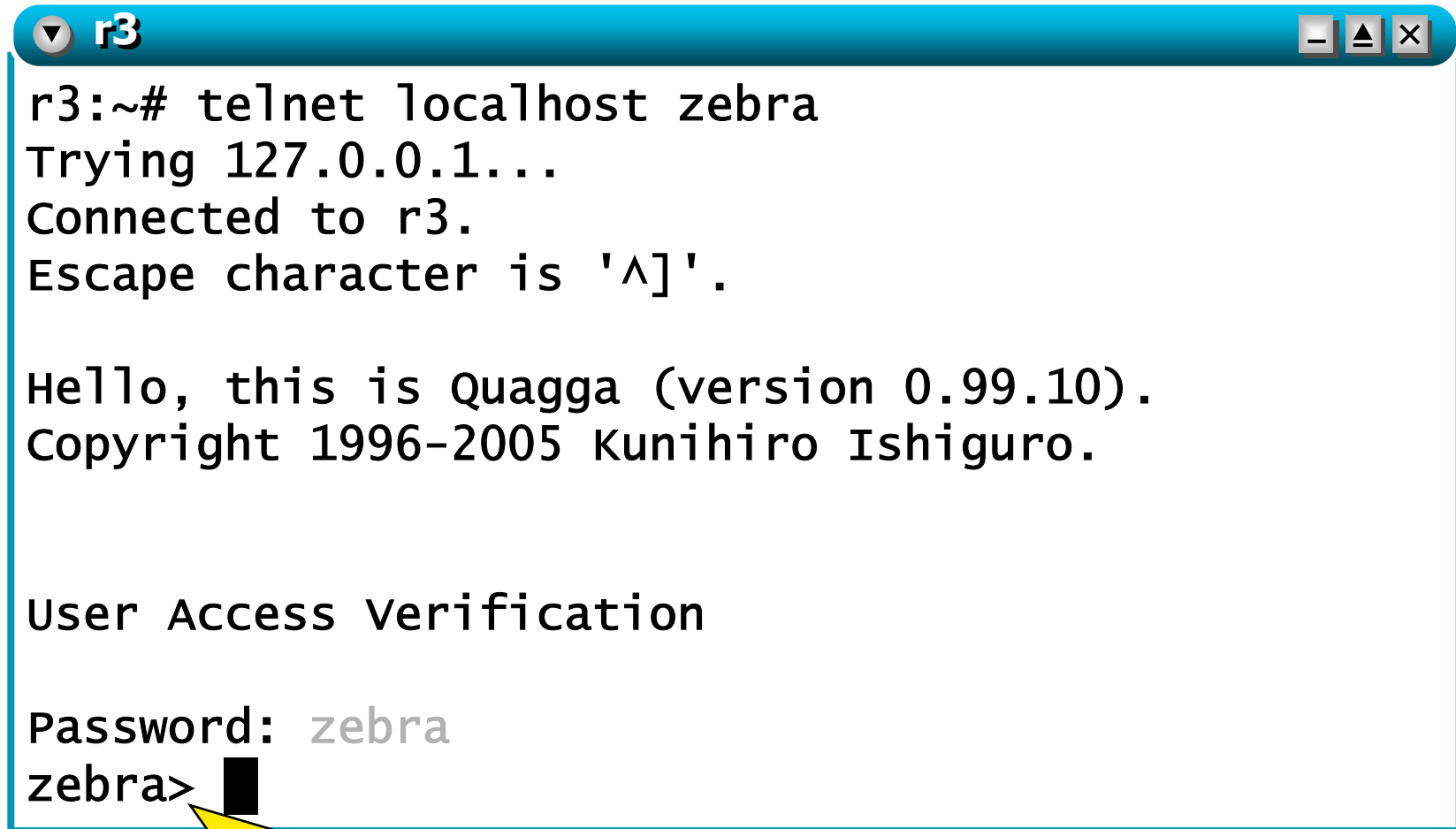
# launching the lab script



```
host machine
user@localhost:~$ cd kathara-lab_quagga
user@localhost:~/kathara-lab_quagga$ kathara 1start
```

- the lab configuration is such that
  - three virtual hosts are created and connected to the right collision domains (virtual hubs)
  - for each virtual host
    - network interfaces are automatically configured
    - quagga configuration files are updated
  - the zebra routing daemon is automatically started

# connecting to the main zebra daemon



```
r3:~# telnet localhost zebra
Trying 127.0.0.1...
Connected to r3.
Escape character is '^]'.

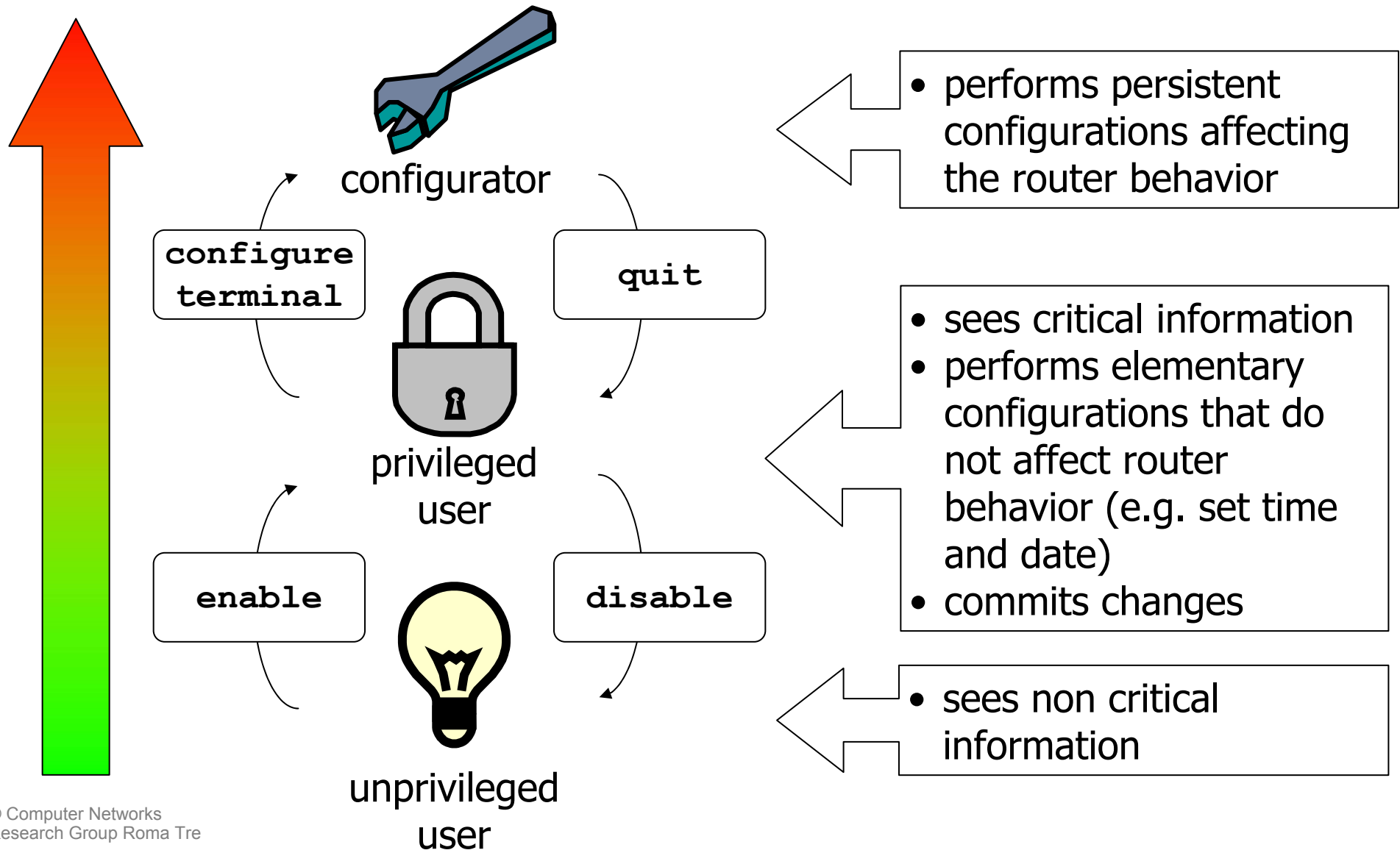
Hello, this is Quagga (version 0.99.10).
Copyright 1996-2005 Kunihiro Ishiguro.

User Access Verification

Password: zebra
zebra>
```

we are  
unprivileged users

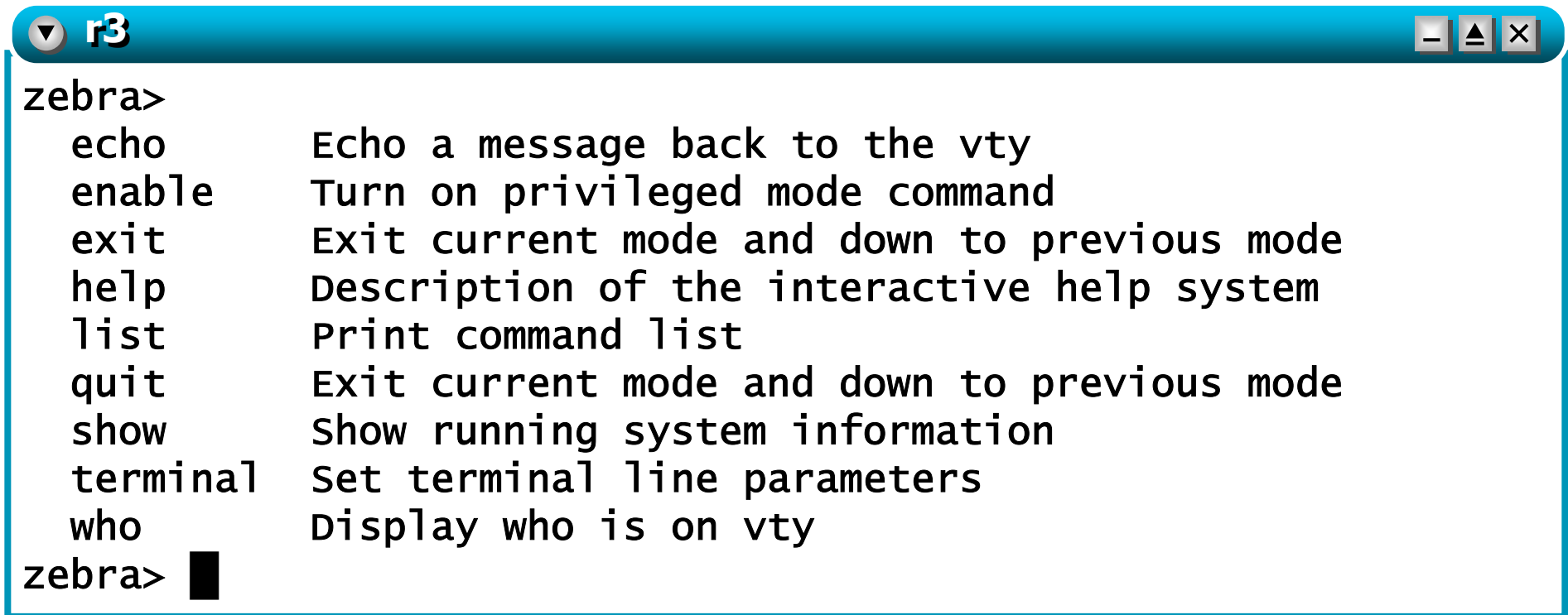
# privileges on a router





# available commands

- press '?' at the command prompt...



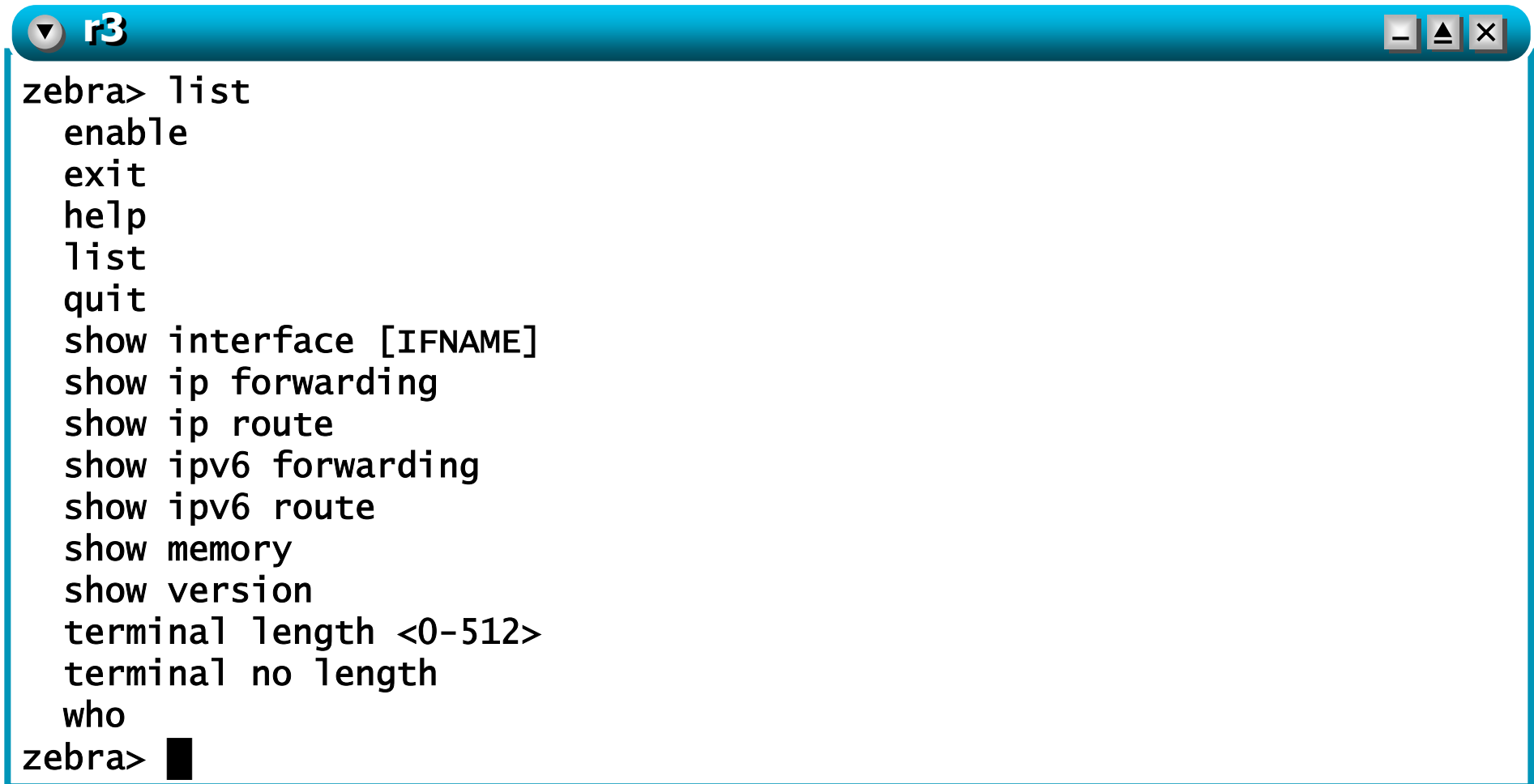
A terminal window titled 'r3' with standard window controls (minimize, maximize, close). The terminal shows the 'zebra>' prompt followed by a list of available commands and their descriptions. The commands are: echo, enable, exit, help, list, quit, show, terminal, and who. Each command is followed by a brief description of its function. The terminal ends with the 'zebra>' prompt and a cursor.

```
zebra>  
  echo      Echo a message back to the vty  
  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> █
```

- ...or...

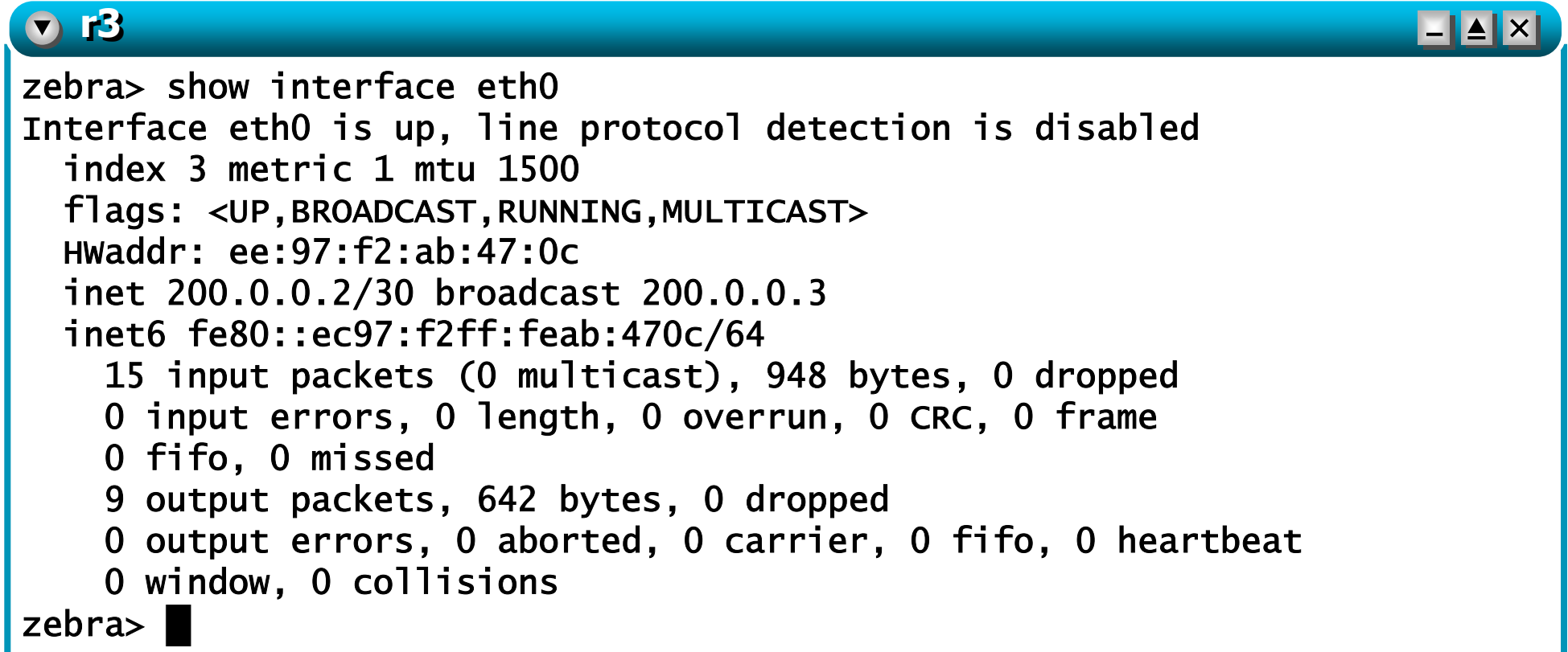
# available commands

- ...type 'list' (an excerpt of the output follows)



```
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> █
```

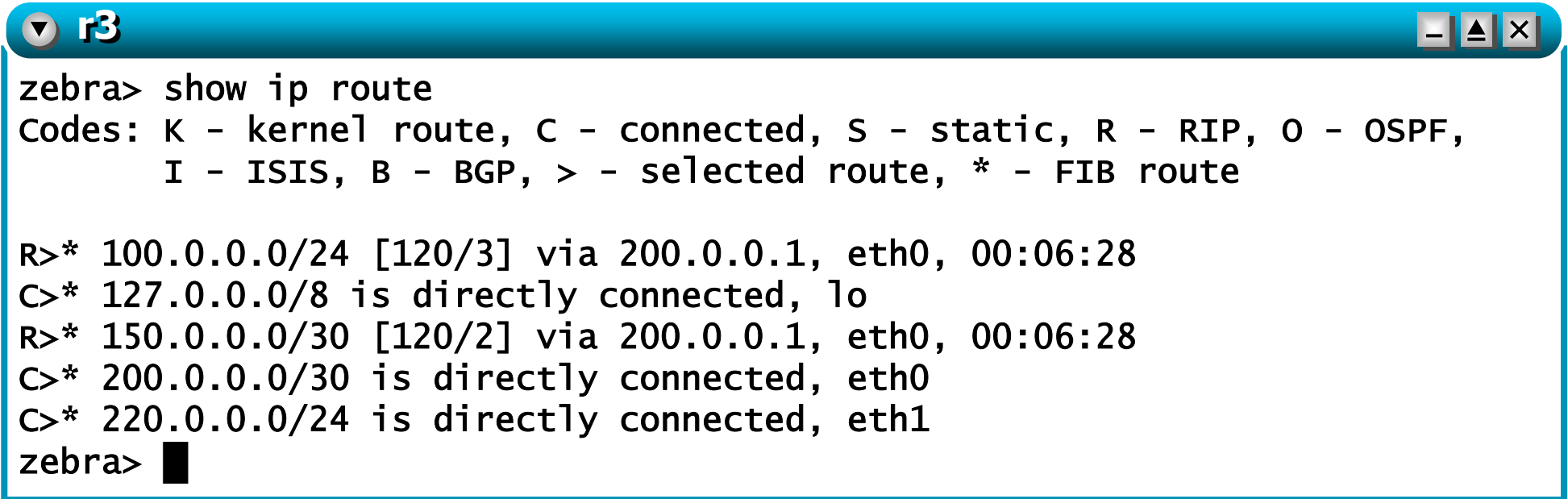
# inspecting interfaces



```
zebra> show interface eth0
Interface eth0 is up, line protocol detection is disabled
  index 3 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: ee:97:f2:ab:47:0c
  inet 200.0.0.2/30 broadcast 200.0.0.3
  inet6 fe80::ec97:f2ff:feab:470c/64
    15 input packets (0 multicast), 948 bytes, 0 dropped
    0 input errors, 0 length, 0 overrun, 0 CRC, 0 frame
    0 fifo, 0 missed
    9 output packets, 642 bytes, 0 dropped
    0 output errors, 0 aborted, 0 carrier, 0 fifo, 0 heartbeat
    0 window, 0 collisions
zebra> █
```

- this roughly corresponds to using `ifconfig` at the shell prompt

# inspecting the zebra routing table



```
▼ r3
zebra> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
      I - ISIS, B - BGP, > - selected route, * - FIB route

R>* 100.0.0.0/24 [120/3] via 200.0.0.1, eth0, 00:06:28
C>* 127.0.0.0/8 is directly connected, lo
R>* 150.0.0.0/30 [120/2] via 200.0.0.1, eth0, 00:06:28
C>* 200.0.0.0/30 is directly connected, eth0
C>* 220.0.0.0/24 is directly connected, eth1
zebra> █
```

- FIB entries from this table (marked with a '>') are injected into the kernel routing table

# altering zebra configuration

unprivileged user mode

privileged user mode

configurator mode

▼ r3

— ▲ ×

```
zebra> enable
Password: zebra
zebra# configure terminal
zebra(config)# hostname zebra-r3
zebra-r3(config)# password foo
zebra-r3(config)# enable password foo
zebra-r3(config)# quit
zebra-r3# write file
Configuration saved to /etc/zebra/zebra.conf
zebra-r3# disable
zebra-r3> exit
Connection closed by foreign host.
r3:~#
```

enter privileged user mode

start editing configuration

edit  
configuration

stop editing configuration

write changes to file

exit privileged user mode

exit

# inspecting the rip routing table

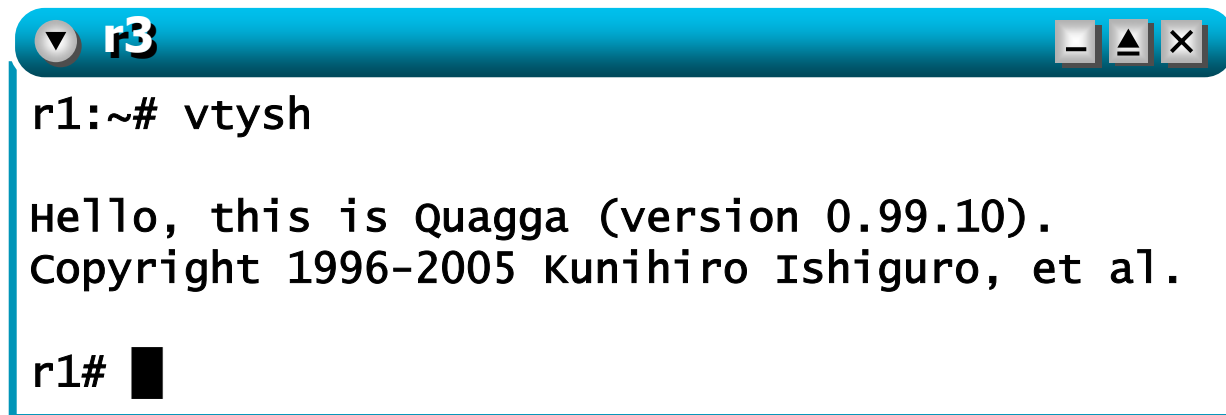
```
r3:~# telnet localhost ripd
.....
Password: zebra
ripd> show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
```

	Network	Next Hop	Metric	From	Tag	Time
R(n)	100.0.0.0/24	200.0.0.1	3	200.0.0.1	0	02:43
R(n)	150.0.0.0/30	200.0.0.1	2	200.0.0.1	0	02:43
C(i)	200.0.0.0/30	0.0.0.0	1	self	0	
C(i)	220.0.0.0/24	0.0.0.0	1	self	0	

```
ripd> █
```

# a one-fits-all shell

- instead of having to connect to each single daemon, users can interact with quagga by using a built-in shell, called **vttysh**



```
r3
r1:~# vtysh

Hello, this is Quagga (version 0.99.10).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r1#
```

- the user is not prompted for a password
- all the commands from the single routing daemons (including quagga itself) are available in this shell