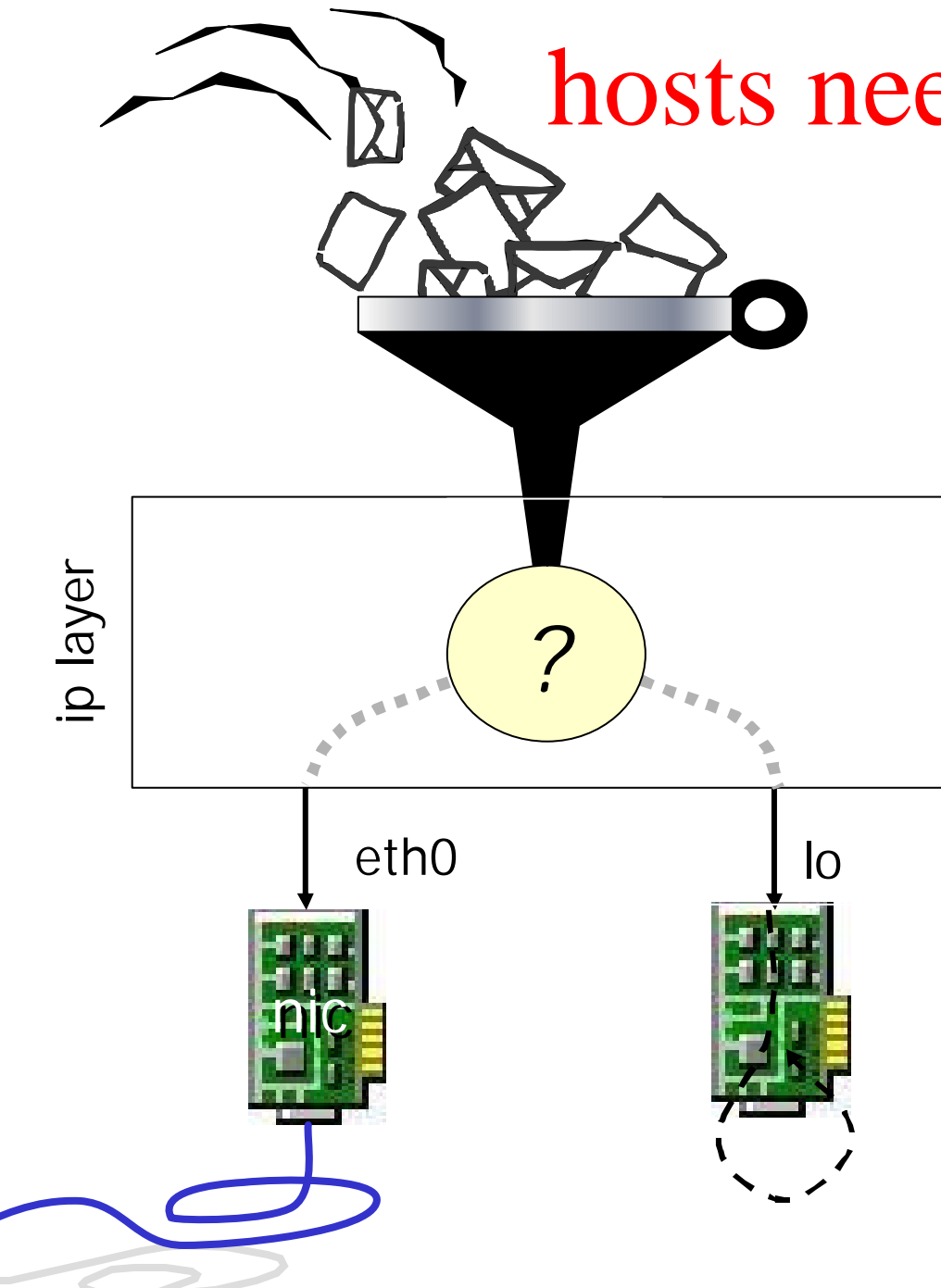


# lab – rip

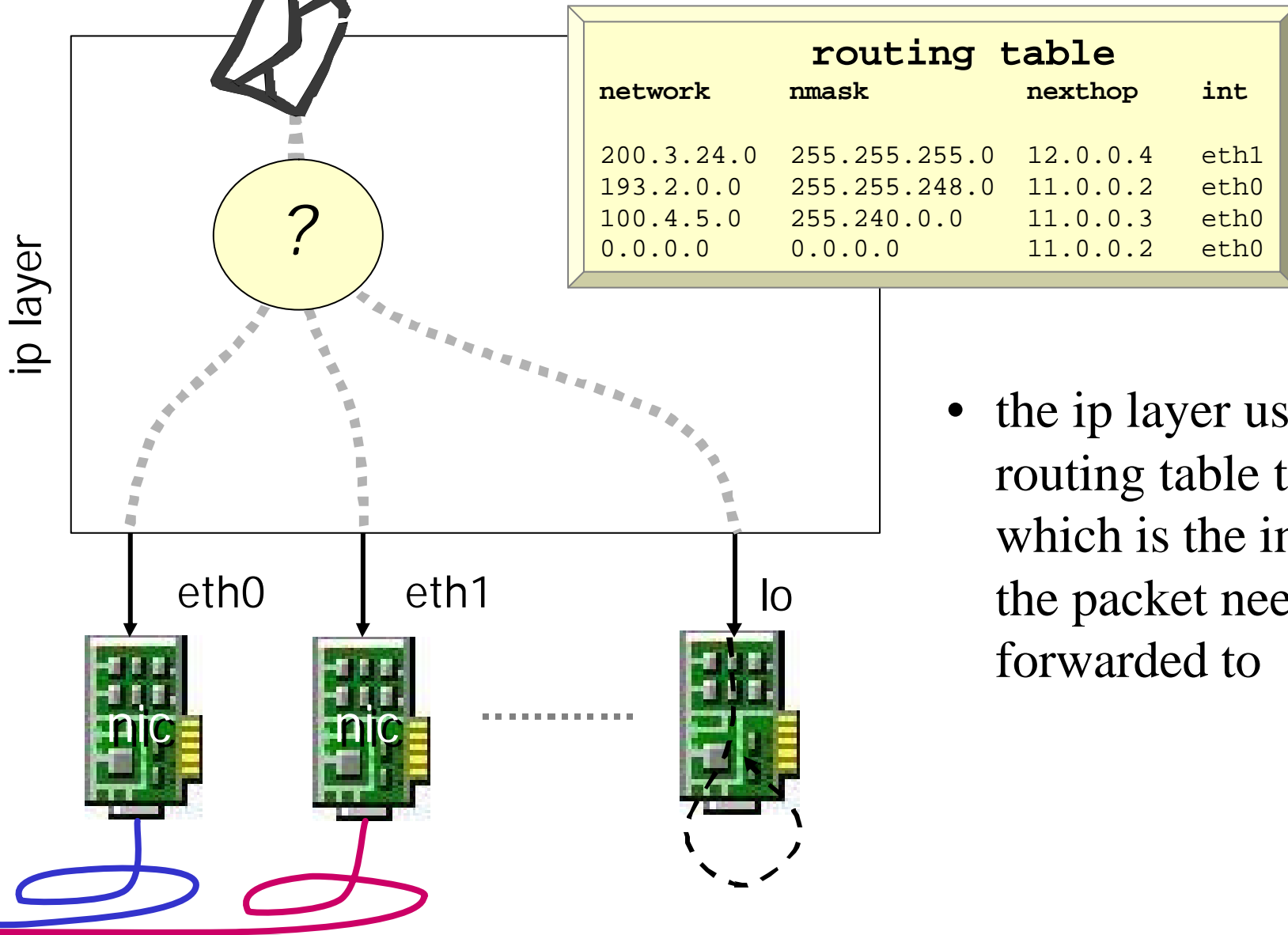
experiences with RIPv2 distance vector protocol

# hosts need routing

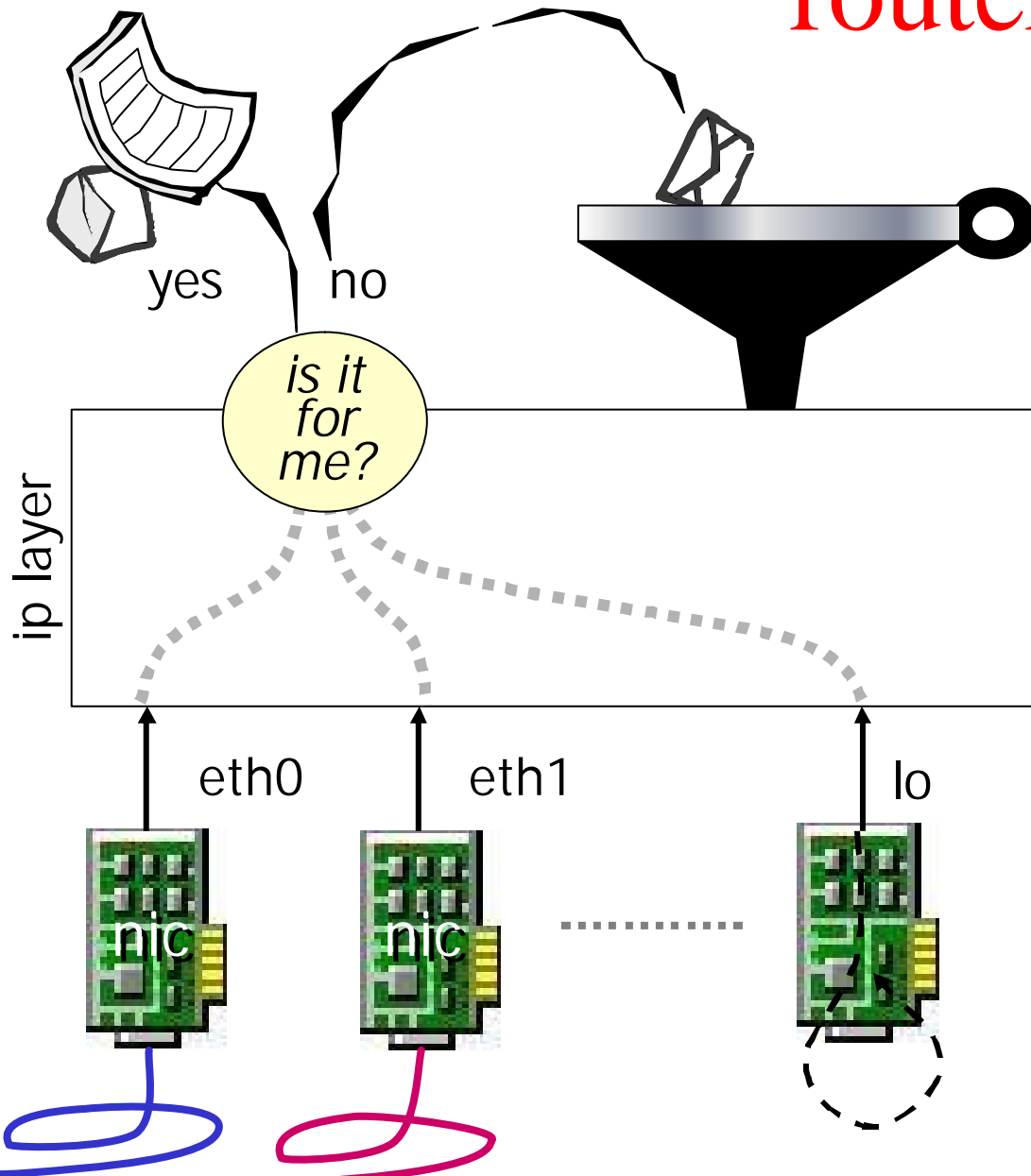


- each host with a network stack performs some elementary routing
- the network stack may be used to access local services (e.g., XWindows)
- the host must decide when a packet needs to be sent to the network interface card (nic) and when it needs to be bounced to the loopback interface (lo)

# routing table



# routers

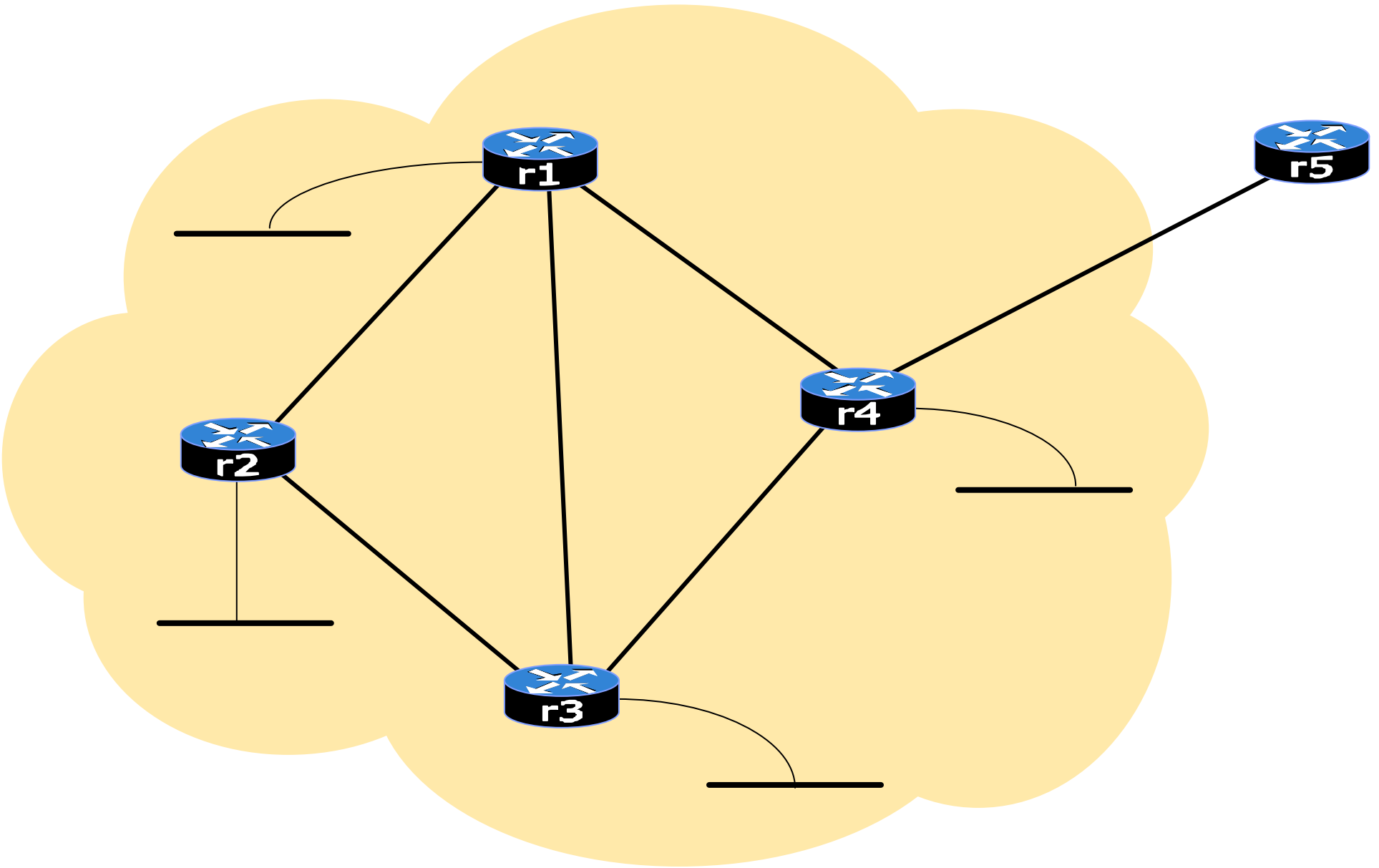


- a router (also called: *gateway* or *intermediate-system*):
  - has more than one network interface card
  - feeds incoming ip packets (that are not for the router itself) back in the routing process
    - this operation is called *relaying* or *forwarding*

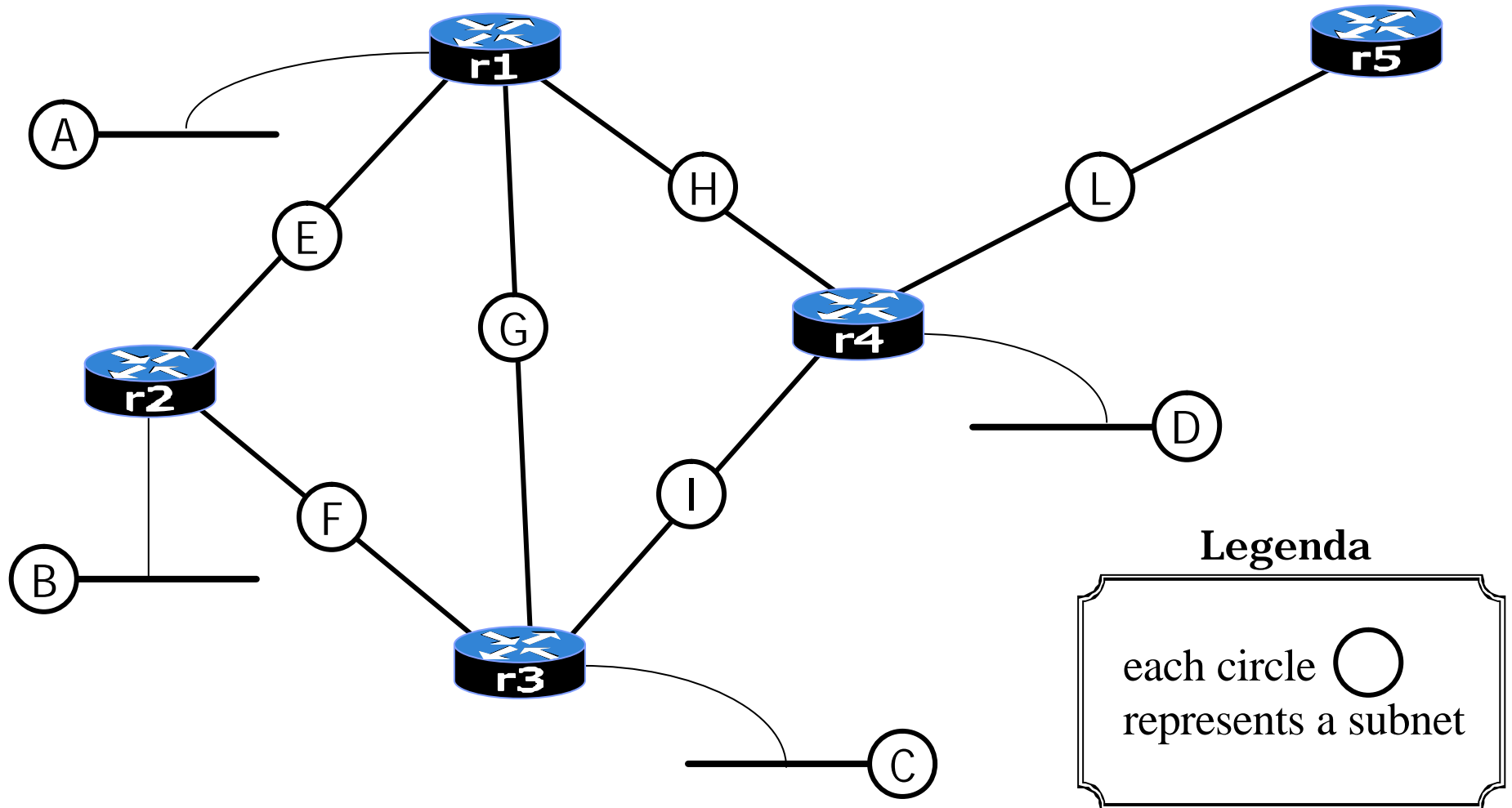
# routing protocols

- routing protocols are used to automatically update the routing tables
- they fall into two main categories:
  - link-state routing protocols
    - approach: send the minimum information to everyone
    - each router reconstructs the whole network graph and computes a shortest path tree to all destinations
    - examples: is-is, ospf
  - distance-vector routing protocols
    - approach: send all your information to a few
    - update your routing information based on what you hear
    - examples: rip
- in this lab we will see an example of RIPv2 protocol on zebra boxes

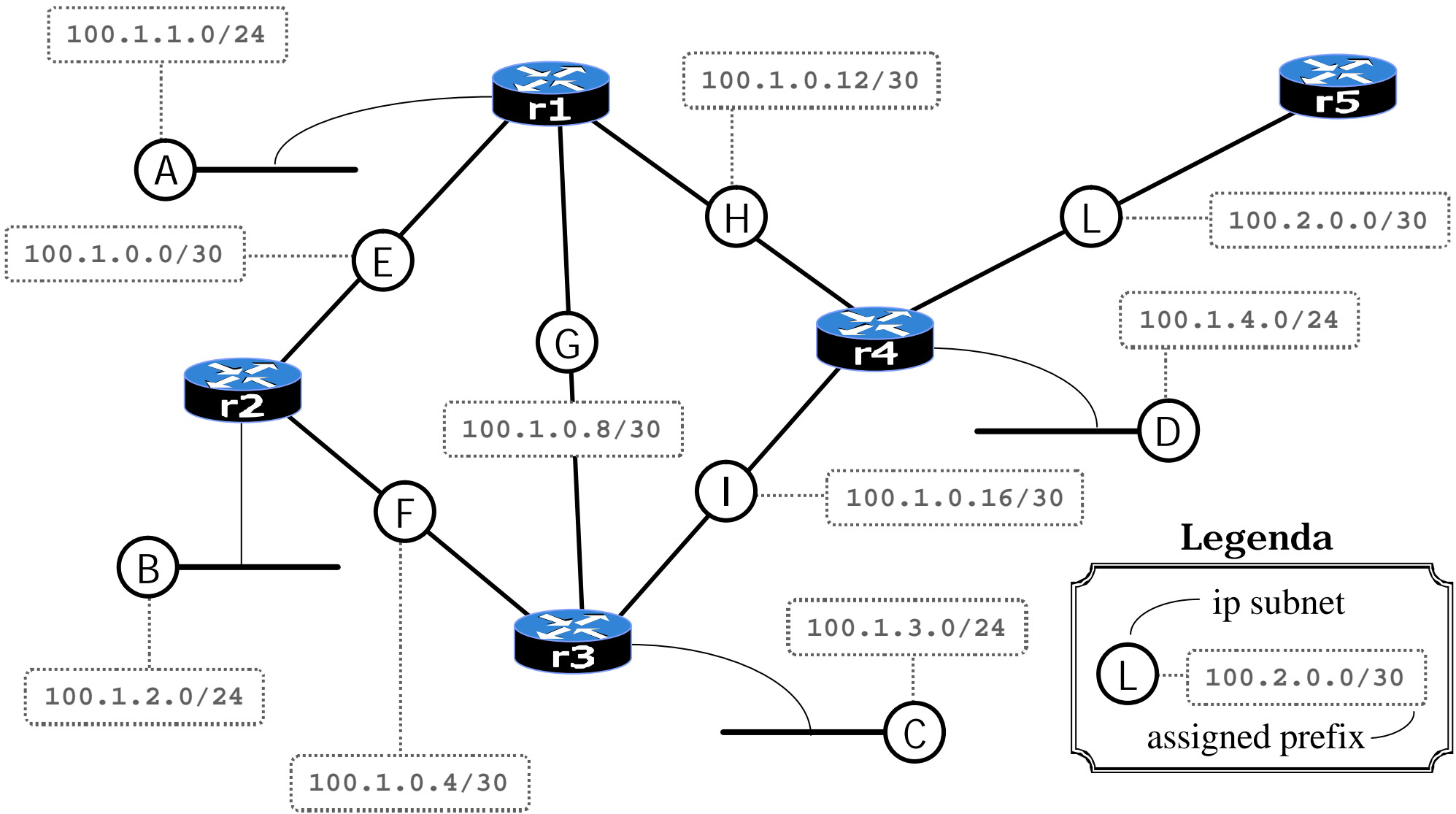
# a small network connected to the Internet



# the involved ip subnets

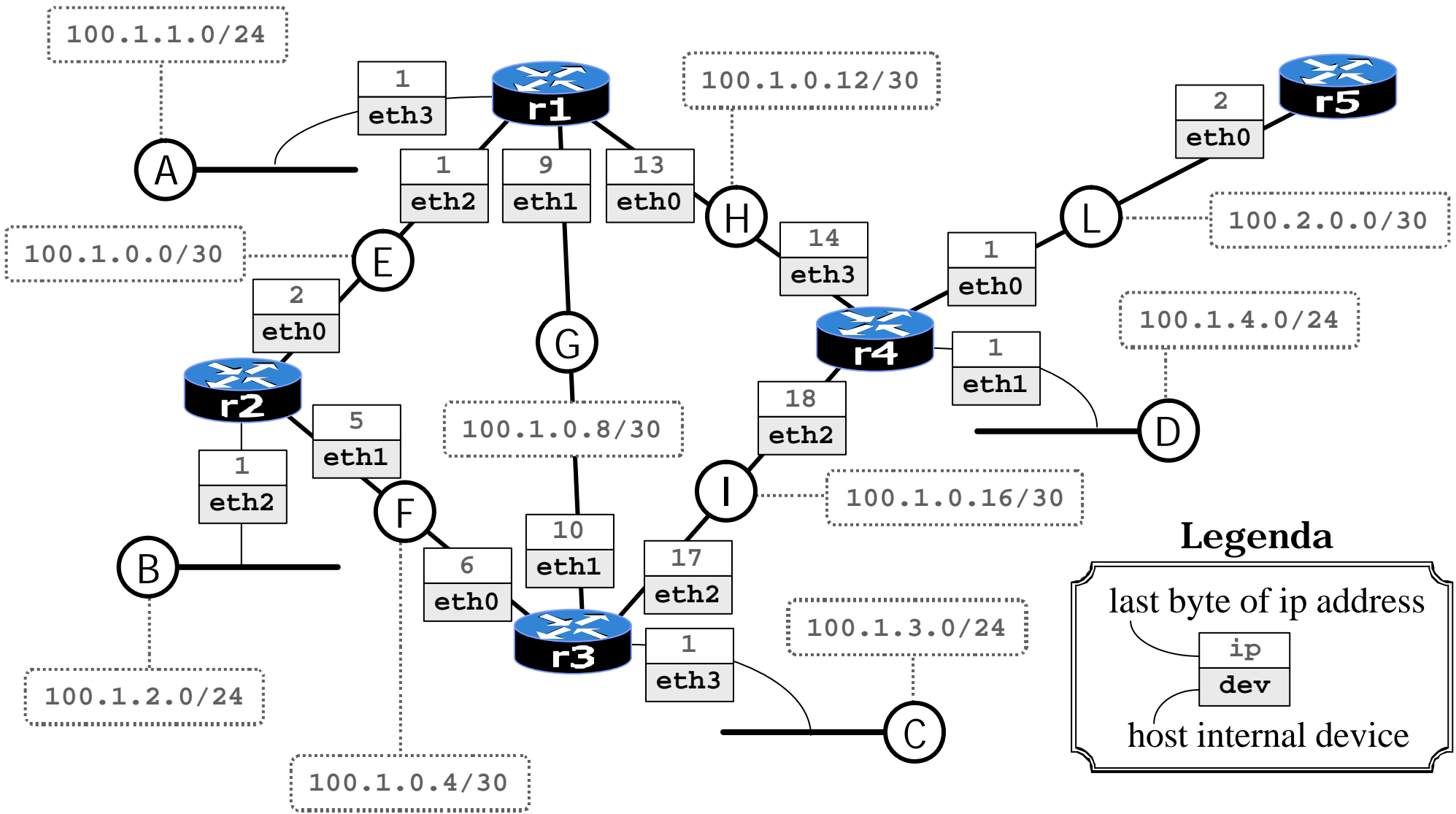


# assigning ip numbers to subnets





# assigning ip numbers to interfaces



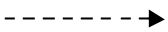


# launching the lab script

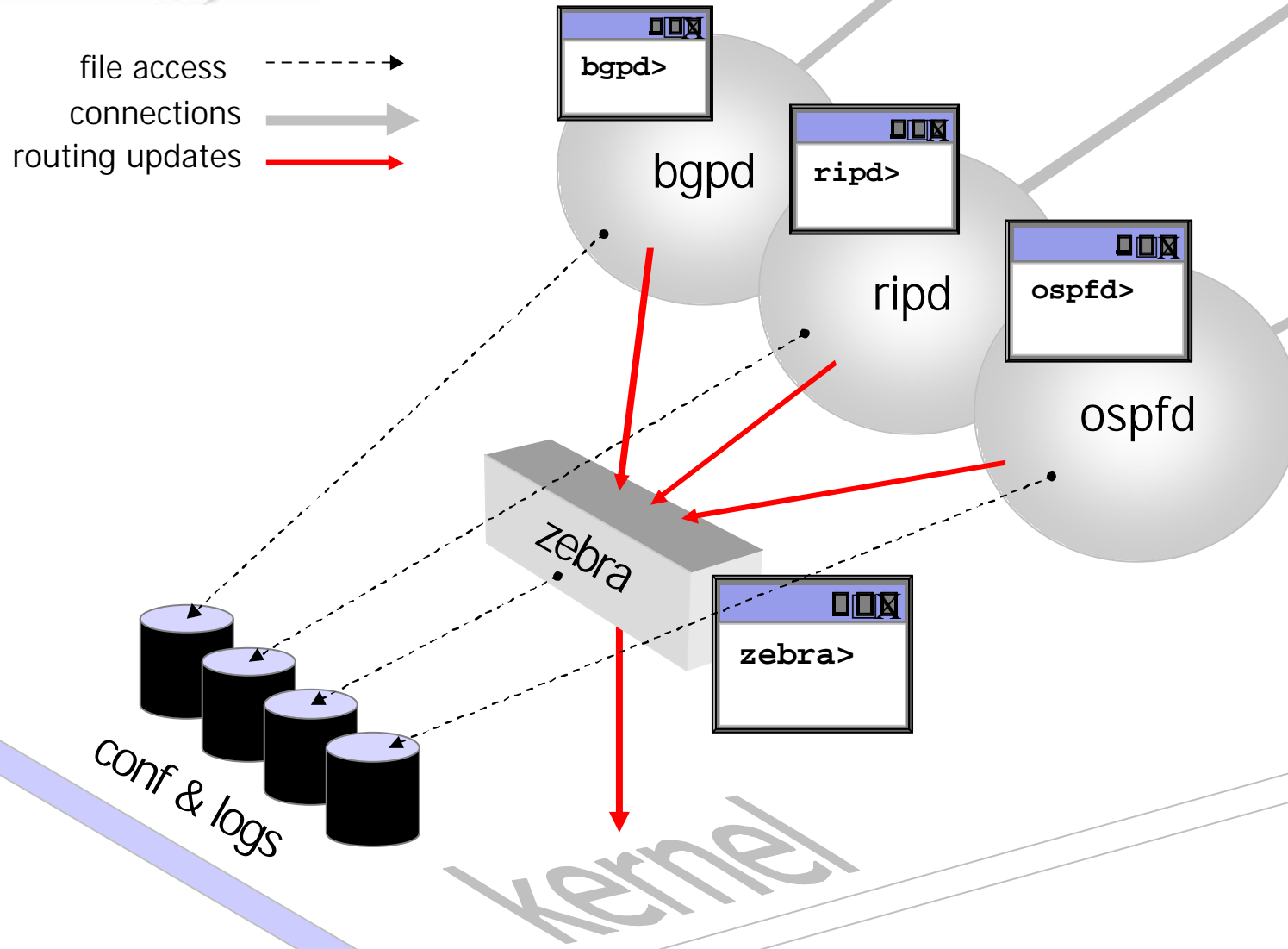
```
user@localhost:~$ lab-rip start
```

the `lab-rip` script:

- creates the five virtual hosts connecting them to the right collision domains (virtual hubs)
- for each virtual host:
  - configures its interfaces
  - updates zebra configuration file `/etc/zebra/daemons`
  - updates zebra configuration file `/etc/zebra/zebra.conf`
  - updates zebra configuration file `/etc/zebra/ripd.conf`
- it does not start zebra on the machines

# zebra is a routing daemon

file access   
connections   
routing updates 



# inspecting zebra configuration files

```
r4:~# cd /etc/zebra
```

```
r4:/etc/zebra# ls
```

```
bgpd.conf  ospf6d.conf  ripd.conf  vtysh.conf  
daemons    ospfd.conf  ripngd.conf  zebra.conf
```

```
r4:/etc/zebra#
```

when zebra will be started each daemon will check these files to read the starting configuration

# inspecting the daemons file

```
r4:/etc/zebra# more 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
r4:/etc/zebra#
```

zebra main daemon will be started

rip daemon will be started too

# examining zebra configuration file

```
r4:/etc/zebra# more zebra.conf
```

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

```
!
```

```
! zebra configuration file
```

```
!
```

```
hostname zebrad
```

the prompt of zebra interface

```
password root
```

the password to connect to the daemon

```
enable password root
```

the superuser password

```
!
```

```
! Static default route sample.
```

```
!
```

```
!ip route 0.0.0.0/0 203.181.89.241
```

```
!
```

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

```
r4:/etc/zebra#
```

# examining ripd configuration file

```
r4:/etc/zebra# more ripd.conf
```

```
!
```

```
hostname ripd
```

```
password root
```

```
enable password root
```

talk rip on some interface

```
!
```

```
router rip
```

redistribute to rip neighbors  
all directly connected subnets

```
redistribute connected
```

```
network 100.1.0.0/16
```

send rip multicast  
packets to interfaces  
falling into this prefix

```
!
```

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

```
r4:/etc/zebra#
```

# checking connectivity

towards a directly connected destination

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

```
PING 100.1.0.13 (100.1.0.13): 56 data bytes
```

```
64 bytes from 100.1.0.13: icmp_seq=0 ttl=255 time=101.5 ms
```

```
64 bytes from 100.1.0.13: icmp_seq=1 ttl=255 time=5.1 ms
```

```
64 bytes from 100.1.0.13: icmp_seq=2 ttl=255 time=4.9 ms
```

```
64 bytes from 100.1.0.13: icmp_seq=3 ttl=255 time=4.9 ms
```

```
64 bytes from 100.1.0.13: icmp_seq=4 ttl=255 time=4.9 ms
```

```
64 bytes from 100.1.0.13: icmp_seq=5 ttl=255 time=4.9 ms
```

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

```
6 packets transmitted, 6 packets received, 0% packet loss
```

```
round-trip min/avg/max = 4.9/21.0/101.5 ms
```

```
r4:~#
```



# checking connectivity

towards a remote destination

```
r4:~# ping 100.1.2.1
```

```
PING 100.1.2.1 (100.1.2.1): 56 data bytes
```

```
ping: sendto: Network is unreachable
```

```
ping: wrote 100.1.2.1 64 chars, ret=-1
```

```
ping: sendto: Network is unreachable
```

```
ping: wrote 100.1.2.1 64 chars, ret=-1
```

```
ping: sendto: Network is unreachable
```

```
ping: wrote 100.1.2.1 64 chars, ret=-1
```

```
--- 100.1.2.1 ping statistics ---
```

```
3 packets transmitted, 0 packets received, 100% packet loss
```

```
r4:~#
```

# examining the kernel routing table

```
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:~#
```

only directly connected destinations  
are known by the “router”  
(no routing daemon is running)

# starting the daemons on all routers

```
r4:~# /etc/init.d/zebra start
```

```
Starting Zebra daemons (prio:10): zebra ripd.
```

```
r4:~#
```

(after a while) all remote destinations are reachable

```
r4:~# ping 100.1.2.1
```

```
PING 100.1.2.1 (100.1.2.1): 56 data bytes
```

```
64 bytes from 100.1.2.1: icmp_seq=0 ttl=254 time=12.2 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=1 ttl=254 time=8.5 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=2 ttl=254 time=8.4 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=3 ttl=254 time=8.1 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=4 ttl=254 time=8.2 ms
```

```
--- 100.1.2.1 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip min/avg/max = 8.1/9.0/12.2 ms
```

```
r4:~#
```

# sniffing RIPv2 traffic

```
r4:~# tcpdump -i eth2 -nn -s 1518
```

```
device eth2 entered promiscuous mode
```

```
tcpdump: listening on eth2
```

```
21:18:04.111625 100.1.0.18.520 > 224.0.0.9.520:
```

```
RIPv2-resp [items 7]:
```

```
{100.1.0.0/255.255.255.252}(2) {100.1.0.8/255.255.255.252}(2)
```

```
{100.1.0.12/255.255.255.252}(1) {100.1.1.0/255.255.255.0}(2)
```

```
{100.1.2.0/255.255.255.0}(3) {100.1.4.0/255.255.255.0}(1)
```

```
{100.2.0.0/255.255.255.252}(1) (DF) [ttl 1]
```

```
21:18:18.150705 100.1.0.17.520 > 224.0.0.9.520:
```

```
RIPv2-resp [items 7]:
```

```
{100.1.0.0/255.255.255.252}(2) {100.1.0.4/255.255.255.252}(1)
```

```
{100.1.0.8/255.255.255.252}(1) {100.1.0.12/255.255.255.252}(2)
```

```
{100.1.1.0/255.255.255.0}(2) {100.1.2.0/255.255.255.0}(2)
```

```
{100.1.3.0/255.255.255.0}(1) (DF) [ttl 1]
```

```
r4:~#
```

# the routing table is now updated

```
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	0	0	0	eth3
100.2.0.0	*	255.255.255.252	U	0	0	0	eth0
100.1.0.4	100.1.0.17	255.255.255.252	UG	0	0	0	eth2
100.1.0.8	100.1.0.13	255.255.255.252	UG	0	0	0	eth3
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	100.1.0.13	255.255.255.0	UG	0	0	0	eth3
100.1.3.0	100.1.0.17	255.255.255.0	UG	0	0	0	eth2
100.1.1.0	100.1.0.13	255.255.255.0	UG	0	0	0	eth3

```
r4:~#
```

# a traceroute to 100.1.2.1

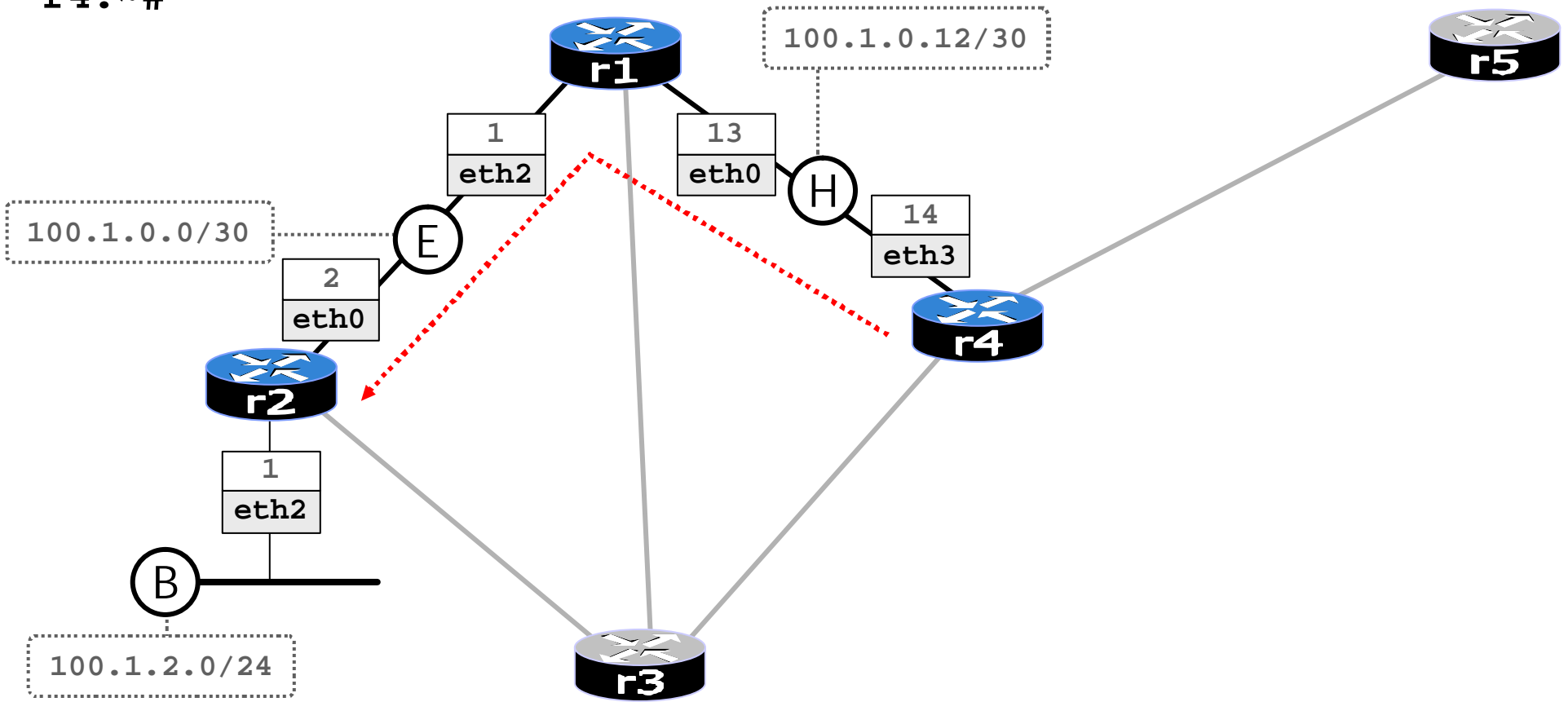
```
r4:~# traceroute 100.1.2.1
```

```
traceroute to 100.1.2.1 (100.1.2.1), 30 hops max, 40 byte packets
```

```
 1 100.1.0.13 (100.1.0.13) 10 ms 6 ms 5 ms
```

```
 2 100.1.2.1 (100.1.2.1) 11 ms 9 ms 8 ms
```

```
r4:~#
```



# connecting to the main zebra daemon

```
r4:~# telnet 127.0.0.1 zebra
```

```
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
```

```
Escape character is '^]'.  
  

```


```
Hello, this is zebra (version 0.84b)
```

```
Copyright 1996-2000 Kunihiro Ishiguro
```

```
User Access Verification
```

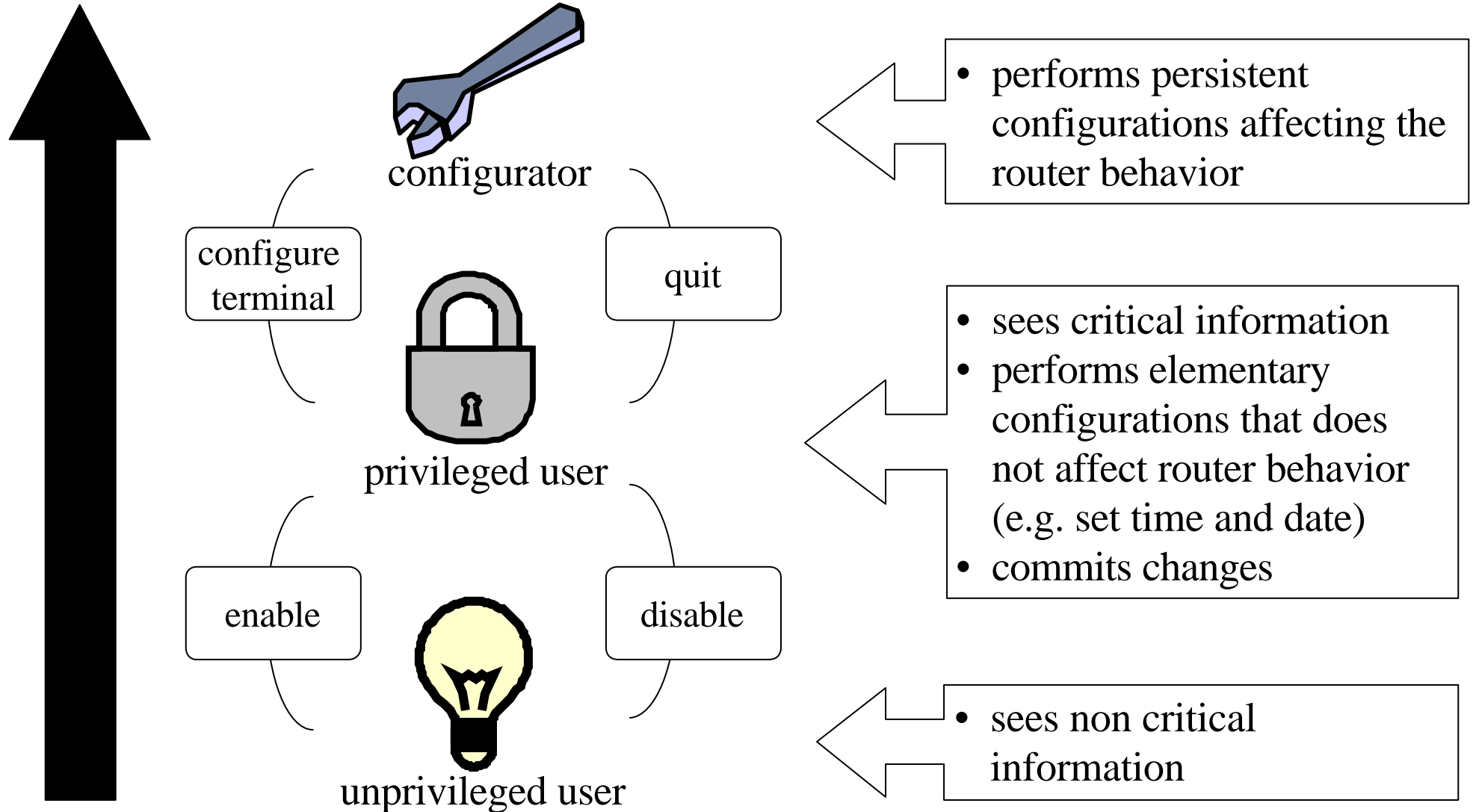
```
Password:
```

```
zebrad>
```



you are unprivileged user now

# privileges on a router





# available commands

zebrad> list

list all available commands

enable

exit

help

list

no terminal length [<0-512>]

quit

show debugging zebra

show interface [IFNAME]

show ip forwarding

show ip route [IPV4\_ADDRESS]

show ipv6 forwarding

show ipv6 route [IPV6\_ADDRESS]

show memory

show table

show version

terminal length <0-512>

who

zebrad>

# inspecting interfaces

```
zebrad> show interface eth0
```

```
Interface eth0
```

```
index 4 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,  
ALLMULTI,MULTICAST>
```

```
HWaddr: fe:fd:64:02:00:01
```

```
inet 100.2.0.1/30 broadcast 100.2.0.3
```

```
inet6 fe80::fcfd:64ff:fe02:1/10
```

```
input packets 4, bytes 232, dropped 0, multicast packets 0
```

```
input errors 0, length 0, overrun 0, CRC 0, frame 0,  
fifo 0, missed 0
```

```
output packets 4, bytes 288, dropped 0
```

```
output errors 0, aborted 0, carrier 0, fifo 0,  
heartbeat 0, window 0
```

```
collisions 0
```

```
zebrad>
```

# inspecting the zebra routing table

```
zebrad> show ip route
```

```
Codes: K - kernel route, C - connected, S - static, R - RIP,  
O - OSPF, B - BGP, * - FIB route.
```

```
R* 100.1.0.0/30          eth3 (7) 100.1.0.13  
R* 100.1.0.4/30         eth2 (6) 100.1.0.17  
R* 100.1.0.8/30        eth3 (7) 100.1.0.13  
C* 100.1.0.12/30       eth3 (7) direct  
C* 100.1.0.16/30       eth2 (6) direct  
R* 100.1.1.0/24        eth3 (7) 100.1.0.13  
R* 100.1.2.0/24        eth3 (7) 100.1.0.13  
R* 100.1.3.0/24        eth2 (6) 100.1.0.17  
C* 100.1.4.0/24        eth1 (5) direct  
C* 100.2.0.0/30        eth0 (4) direct  
C* 127.0.0.0/8         lo (1) direct
```

```
zebrad>
```

# modifying the zebra configuration

```
zebrad> enable
```

want to be privileged user

```
Password:
```

```
zebrad# configure terminal
```

want to edit the configuration

```
zebrad(config)# hostname r4-zebrad
```

```
r4-zebrad(config)# password foo
```

```
r4-zebrad(config)# enable password foo
```

changing the  
configuration

```
r4-zebrad(config)# quit
```

```
r4-zebrad# write file
```

committing changes to the conf file

```
Configuration saved to /etc/zebra/zebra.conf
```

```
r4-zebrad# disable
```

no more root privileges needed

```
r4-zebrad> exit
```

```
Connection closed by foreign host.
```

```
r4:~#>
```

back to the shell prompt

# inspecting rip routing table

```
ripd> show ip rip
```

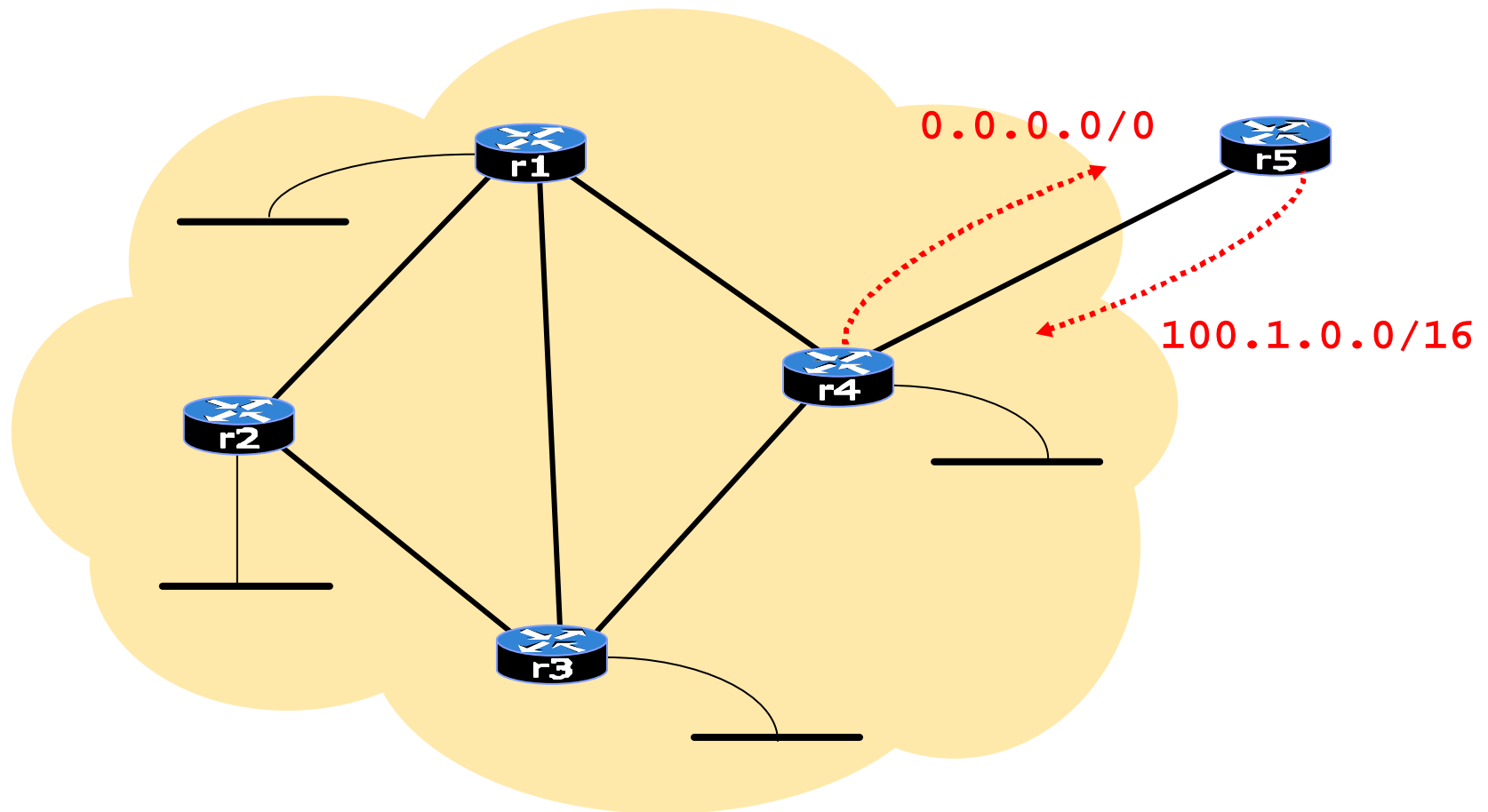
Codes: R - RIP C - connected

	Network	Next Hop	Metric	From	Time	
R	100.1.0.0/30	100.1.0.13	2	100.1.0.13	02:45	
R	100.1.0.4/30	100.1.0.17	2	100.1.0.17	03:00	
R	100.1.0.8/30	100.1.0.13	2	100.1.0.13	02:45	
C	100.1.0.12/30		1			1
C	100.1.0.16/30		1			1
R	100.1.1.0/24	100.1.0.13	2	100.1.0.13	02:45	
R	100.1.2.0/24	100.1.0.13	3	100.1.0.13	02:45	
R	100.1.3.0/24	100.1.0.17	2	100.1.0.17	03:00	
C	100.1.4.0/24		1			1
C	100.2.0.0/30		1			1

```
ripd>
```

# static routing

since our network is a *stub network* (i.e., it has one attachment only to an external router), some static routes will suffice for connecting it to the Internet



# adding a static route to 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 data bytes
```

```
64 bytes from 100.1.2.1: icmp_seq=0 ttl=253 time=173.7 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=1 ttl=253 time=28.4 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=2 ttl=253 time=26.1 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=3 ttl=253 time=23.1 ms
```

```
64 bytes from 100.1.2.1: icmp_seq=4 ttl=253 time=20.9 ms
```

```
--- 100.1.2.1 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip min/avg/max = 20.9/54.4/173.7 ms
```

```
r5:~#
```

# checking connectivity

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

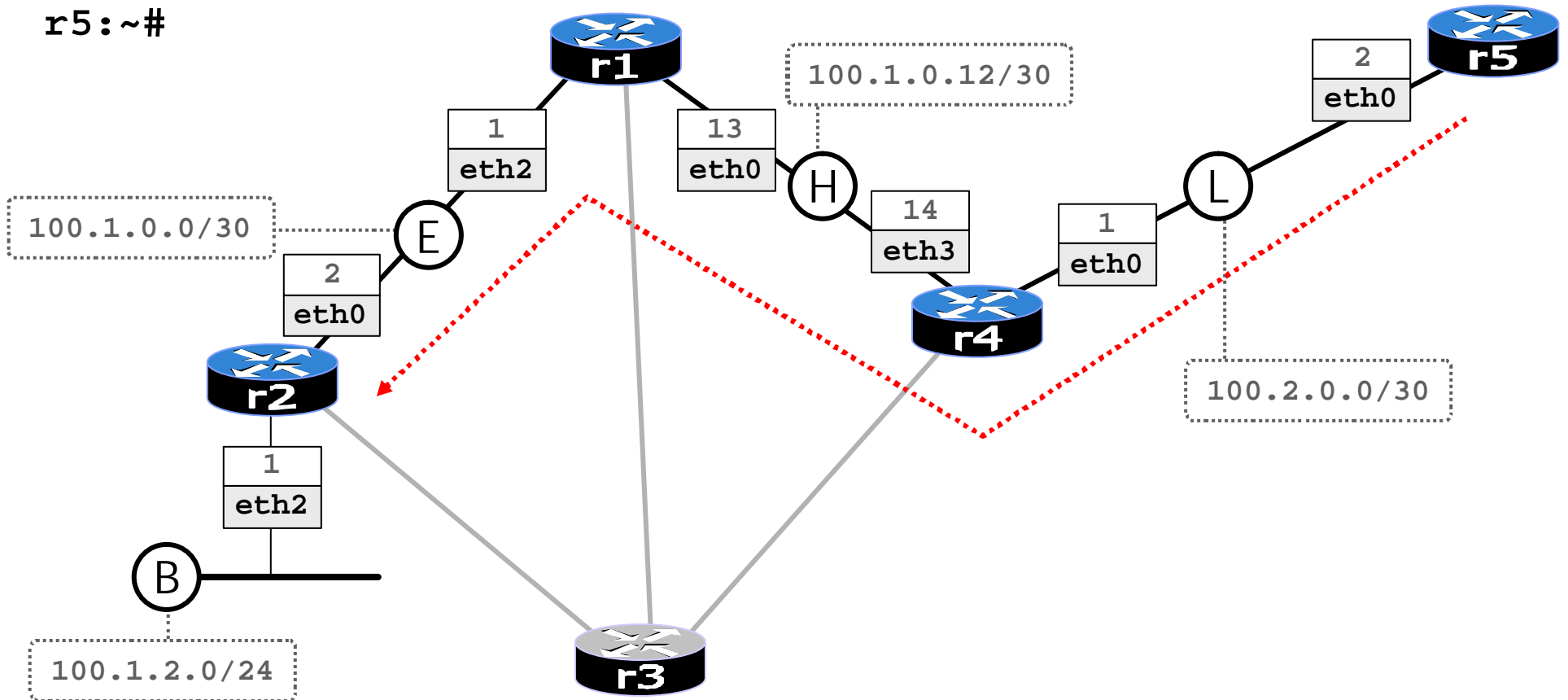
```
traceroute to 100.1.2.1 (100.1.2.1), 30 hops max, 40 byte packets
```

```
1 100.2.0.1 (100.2.0.1) 23 ms 19 ms 20 ms
```

```
2 100.1.0.13 (100.1.0.13) 26 ms 27 ms 23 ms
```

```
3 100.1.2.1 (100.1.2.1) 27 ms 30 ms 28 ms
```

```
r5:~#
```





# adding the default route to r4

```
r4:~# route add default gw 100.2.0.2
```

adding default to kernel

```
r4:~# telnet 127.0.0.1 ripd
```

```
Trying 127.0.0.1...
```

```
Connected to 127.0.0.1.
```

```
Escape character is '^]'.  
  
Hello, this is zebra (version 0.84b)  
Copyright 1996-2000 Kunihiro Ishiguro
```

```
User Access Verification
```

```
Password:
```

```
ripd> enable
```

acquiring privileges

```
Password:
```

```
ripd# configure terminal
```

changing ripd configuration

```
ripd(config)# route 0.0.0.0/0
```

originating the default route

```
ripd(config)# quit
```

```
ripd# disable
```

```
ripd> exit
```

```
Connection closed by foreign host.
```

```
r4:~#
```

# checking connectivity

```
r1:~# ping 193.204.161.1
```

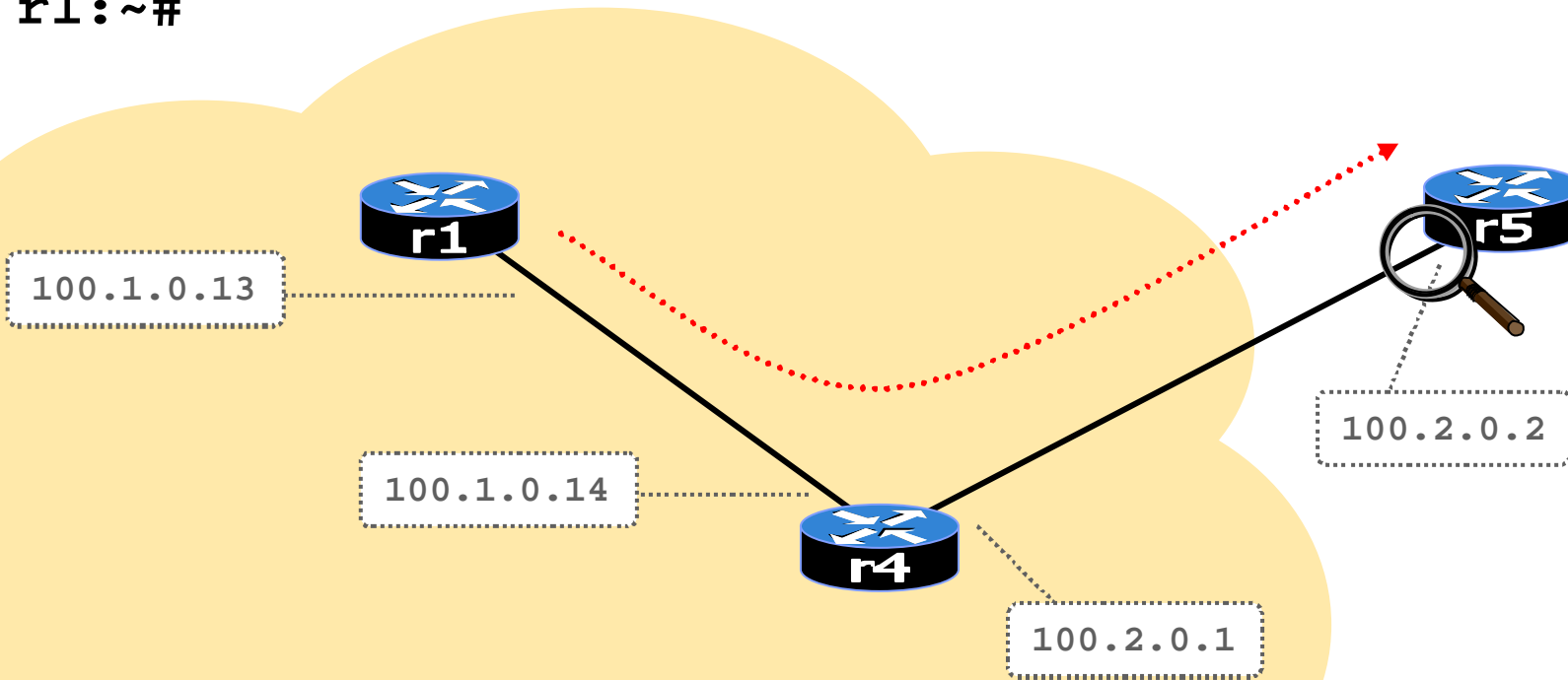
towards any unknown destination

```
PING 193.204.161.1 (193.204.161.1): 56 data bytes
```

```
--- 193.204.161.1 ping statistics ---
```

```
4 packets transmitted, 0 packets received, 100% packet loss
```

```
r1:~#
```



# router r5 receives the packets

```
r5:~# tcpdump -i eth0 -nn -s 1518
```

```
device eth0 entered promiscuous mode
```

```
tcpdump: listening on eth0
```

```
21:54:09.420907 arp who-has 100.2.0.2 tell 100.2.0.1
```

```
21:54:09.420907 arp reply 100.2.0.2 is-at fe:fd:64:2:0:2
```

```
21:54:09.472740 100.1.0.13 > 193.204.161.1:
```

```
icmp: echo request (DF)
```

```
21:54:09.472740 100.2.0.2 > 100.1.0.13:
```

```
icmp: net 193.204.161.1 unreachable [tos 0xc0]
```

```
21:54:10.521383 100.1.0.13 > 193.204.161.1:
```

```
icmp: echo request (DF)
```

```
21:54:10.521383 100.2.0.2 > 100.1.0.13:
```

```
icmp: net 193.204.161.1 unreachable [tos 0xc0]
```

```
21:54:14.472520 arp who-has 100.2.0.1 tell 100.2.0.2
```

```
21:54:14.472520 arp reply 100.2.0.1 is-at fe:fd:64:2:0:1
```

```
21:54:15.522365 100.1.0.13 > 193.204.161.1:
```

```
icmp: echo request (DF)
```

# the default route is injected into the net

```
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	0	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	0	0	0	eth0
100.1.0.4	100.1.0.10	255.255.255.252	UG	0	0	0	eth1
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	0	0	0	eth0
100.1.2.0	100.1.0.2	255.255.255.0	UG	0	0	0	eth2
100.1.3.0	100.1.0.10	255.255.255.0	UG	0	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	0	0	0	eth0

```
r1:~#
```

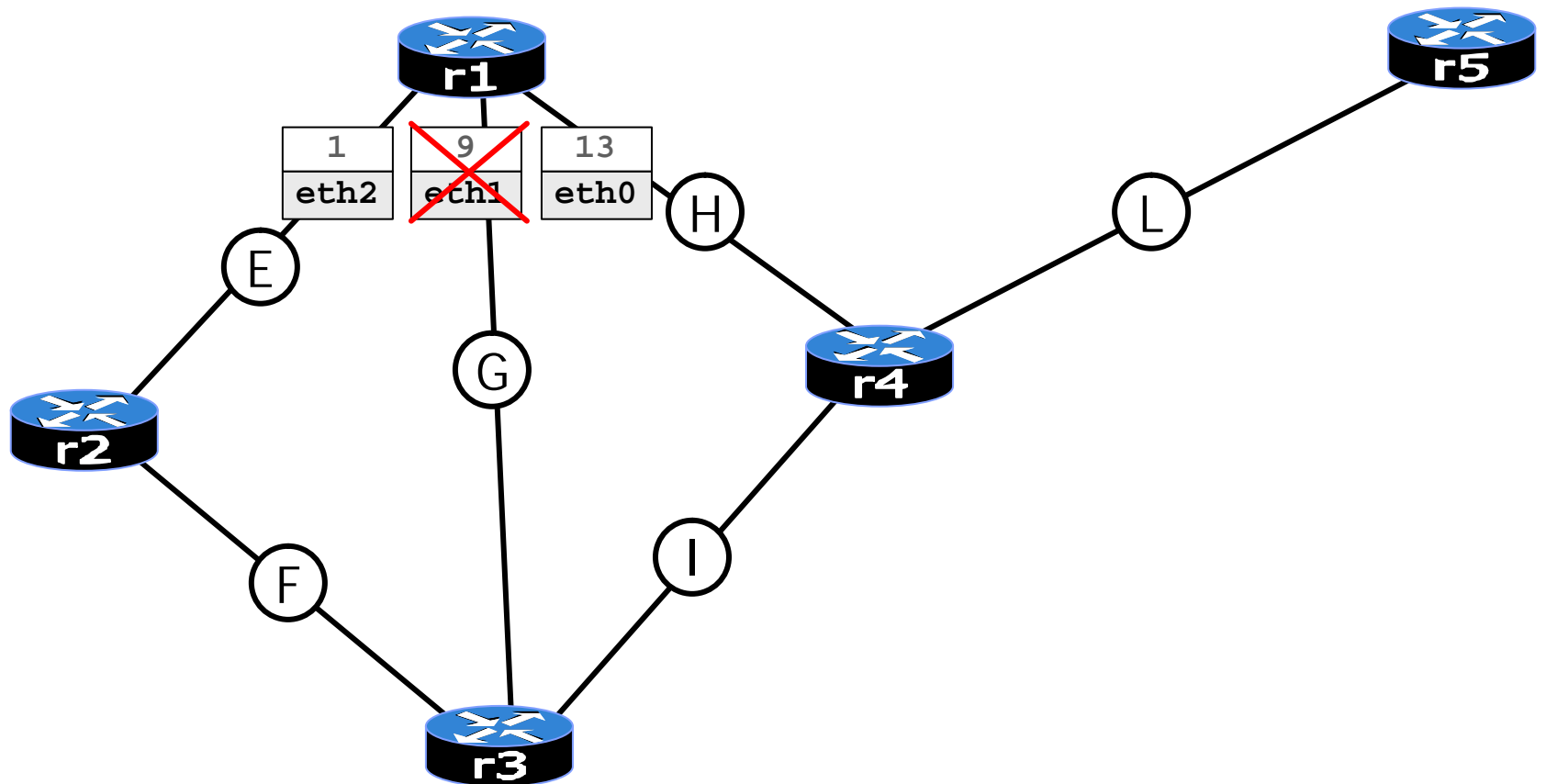
# shutting down an interface

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

```
traceroute to 100.1.0.10 (100.1.0.10), 30 hops max, 40 byte packets
```

```
 1  100.1.0.10 (100.1.0.10)  45 ms  12 ms  5 ms
```

```
r1:~# ifconfig eth1 down
```



# trying the default route

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

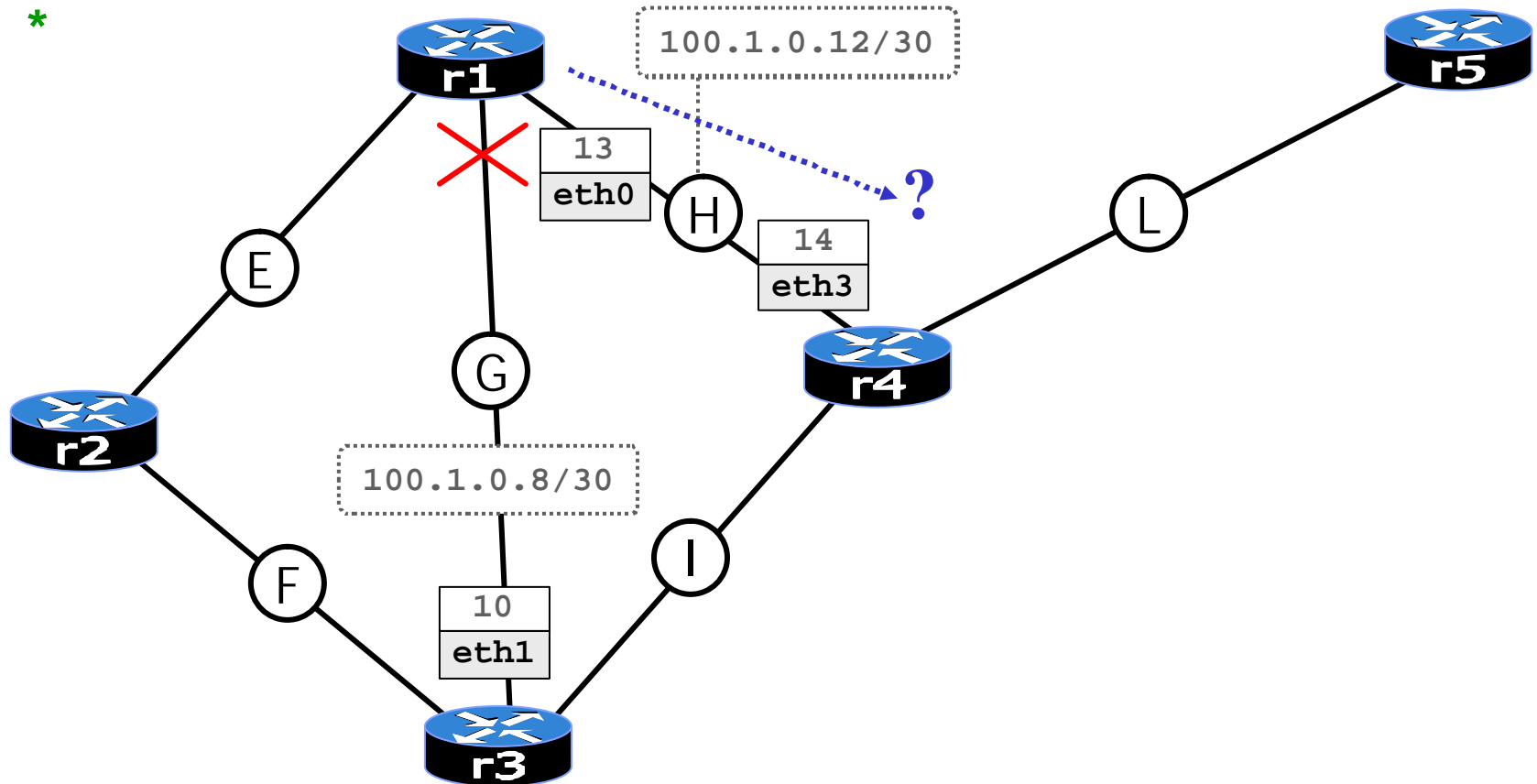
```
traceroute to 100.1.0.10 (100.1.0.10), 30 hops max, 40 byte packets
```

```
1  100.1.0.14 (100.1.0.14)  40 ms  6 ms  4 ms
```

```
2  * * *
```

```
3  * * *
```

```
r1:~#
```



# finding an alternative way

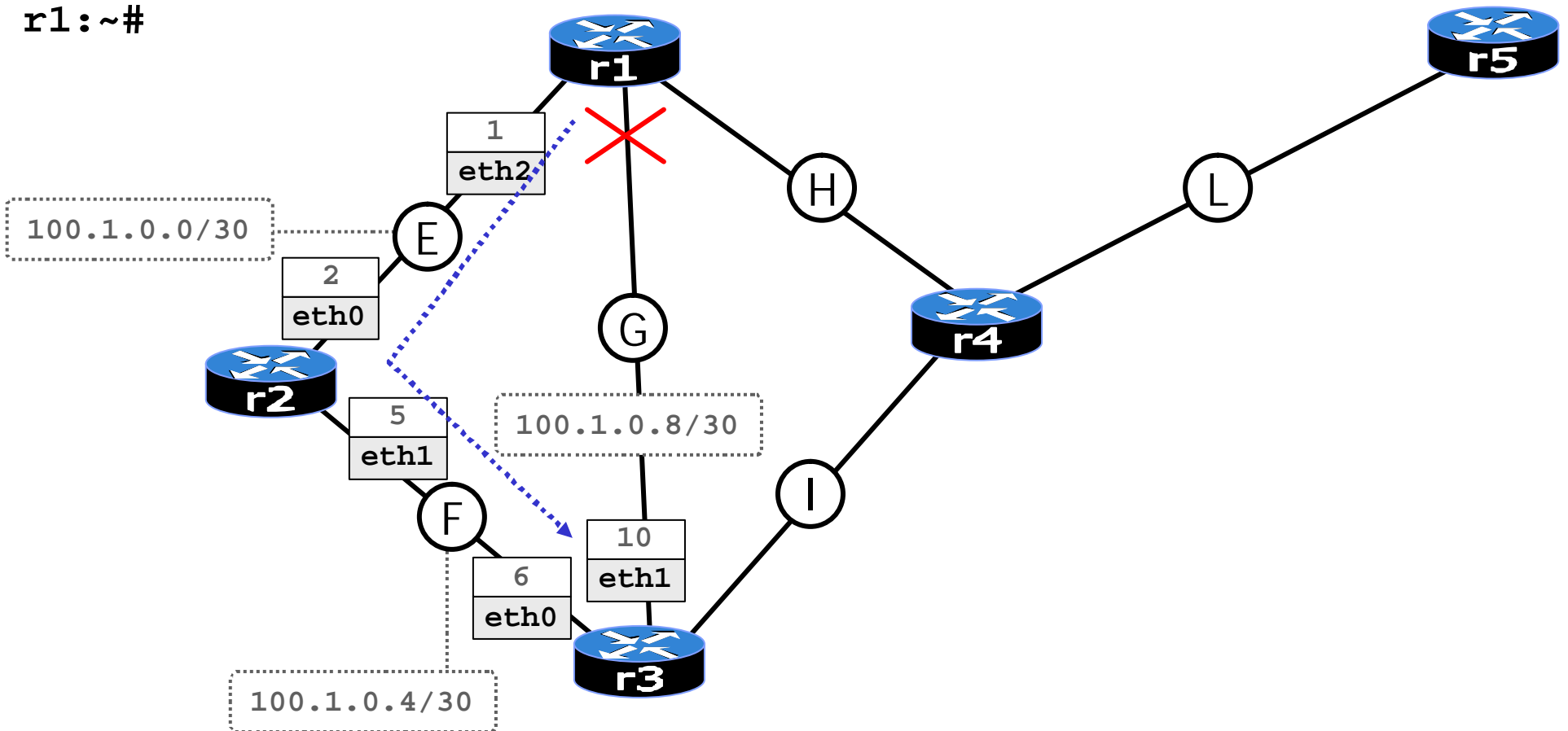
```
r1:~# traceroute 100.1.0.10
```

```
traceroute to 100.1.0.10 (100.1.0.10), 30 hops max, 40 byte packets
```

```
1  100.1.0.2 (100.1.0.2)  6 ms  6 ms  4 ms
```

```
2  100.1.0.10 (100.1.0.10)  8 ms  8 ms  6 ms
```

```
r1:~#
```



# the routing table of **r1** is updated

```
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	0	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	0	0	0	eth0
100.1.0.4	100.1.0.2	255.255.255.252	UG	0	0	0	eth2
100.1.0.8	100.1.0.2	255.255.255.252	UG	0	0	0	eth2
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	0	0	0	eth0
100.1.2.0	100.1.0.2	255.255.255.0	UG	0	0	0	eth2
100.1.3.0	100.1.0.2	255.255.255.0	UG	0	0	0	eth2
100.1.1.0	*	255.255.255.0	U	0	0	0	eth3
default	100.1.0.14	0.0.0.0	UG	0	0	0	eth0

```
r1:~#
```