Università degli Studi Roma Tre
Dipartimento di Ingegneria
Computer Networks Research Group

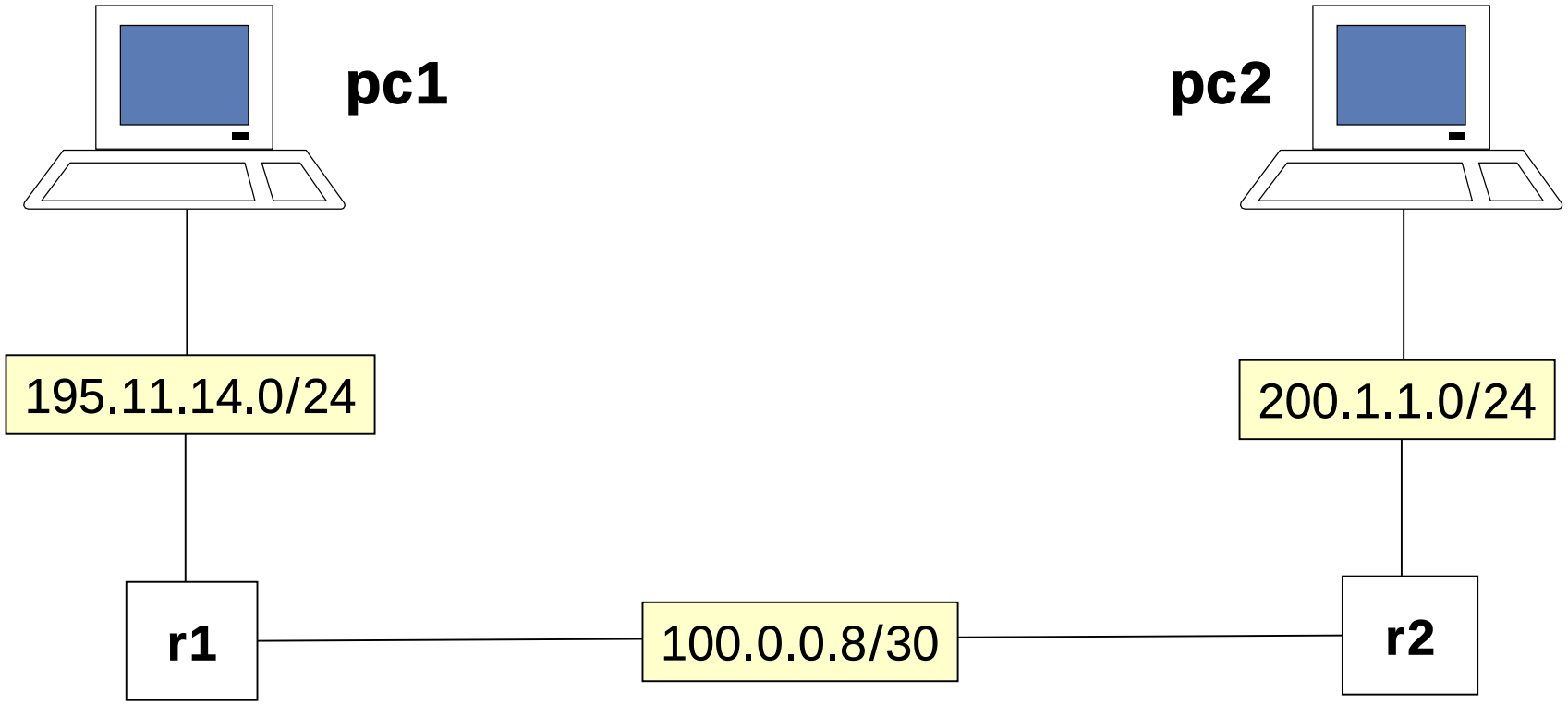
kathara lab

static-routing

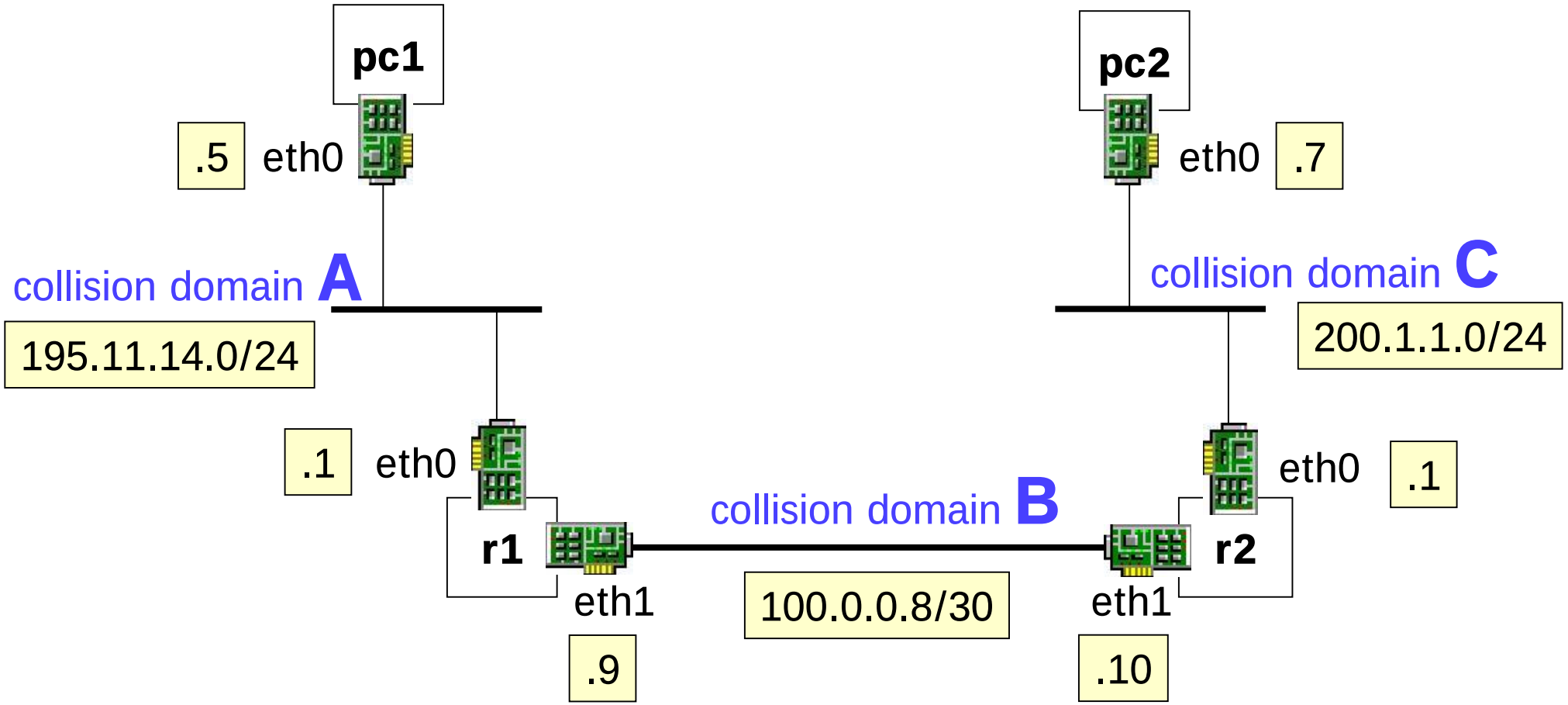
Version	1.0
Author(s)	G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	an example of configuration of static routes – kathara version of netkit lab static-routing vers. 2.2

Copyright notice

step 1 – network topology high level view



step 1 – network topology configuration details



step 2 – the lab

- lab directory hierarchy
  - lab.conf
  - pc1/
  - pc1.startup
  - pc2/
  - pc2.startup
  - r1/
  - r1.startup
  - r2/
  - r2.startup

step 2 – the lab

lab.conf

```
r1[0]=A
r1[1]=B

r2[0]=C
r2[1]=B

pc1[0]=A
pc2[0]=C
```

pc1.startup

```
ifconfig eth0 195.11.14.5 netmask 255.255.255.0 broadcast 195.11.14.255 up
#route add default gw 195.11.14.1 dev eth0
```

the routing table entries will be added manually

pc2.startup

```
ifconfig eth0 200.1.1.7 netmask 255.255.255.0 broadcast 200.1.1.255 up
#route add default gw 200.1.1.1 dev eth0
```

step 2 – the lab

r1.startup

```
ifconfig eth0 195.11.14.1 netmask 255.255.255.0 broadcast 195.11.14.255 up
ifconfig eth1 100.0.0.9 netmask 255.255.255.252 broadcast 100.0.0.11 up
#route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 dev eth1
```

r2.startup

```
ifconfig eth0 200.1.1.1 netmask 255.255.255.0 broadcast 200.1.1.255 up
ifconfig eth1 100.0.0.10 netmask 255.255.255.252 broadcast 100.0.0.11 up
#route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 dev eth1
```

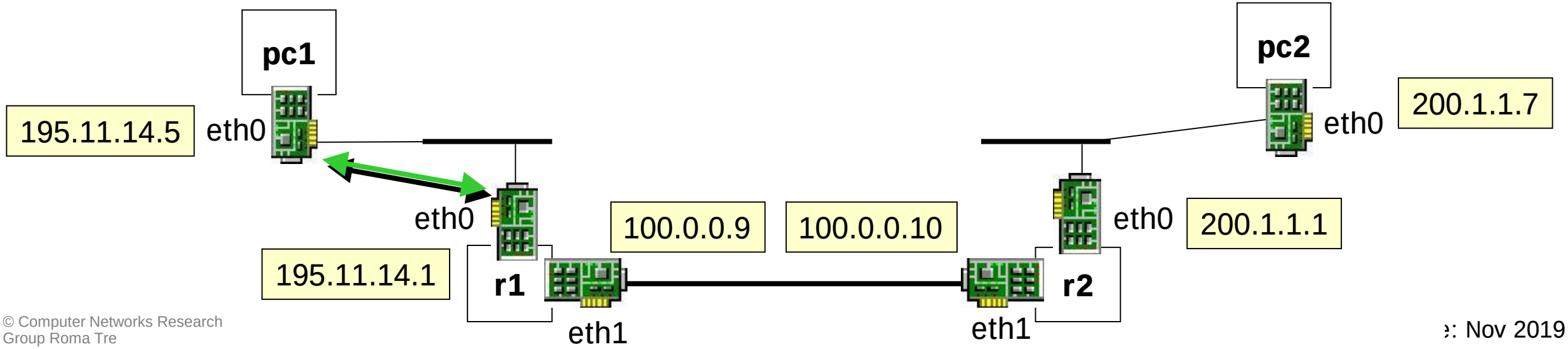
the routing table entries will be added manually

step 3 – testing connectivity

```
root@pc1:~$ ping 195.11.14.1
PING 195.11.14.1 (195.11.14.1) 56(84) bytes of data.
64 bytes from 195.11.14.1: icmp_seq=1 ttl=64 time=3.1
64 bytes from 195.11.14.1: icmp_seq=2 ttl=64 time=0.3

--- 195.11.14.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, ti
rtt min/avg/max/mdev = 0.308/1.285/3.176/1.337 ms
```

interfaces on the same domain can reach each other

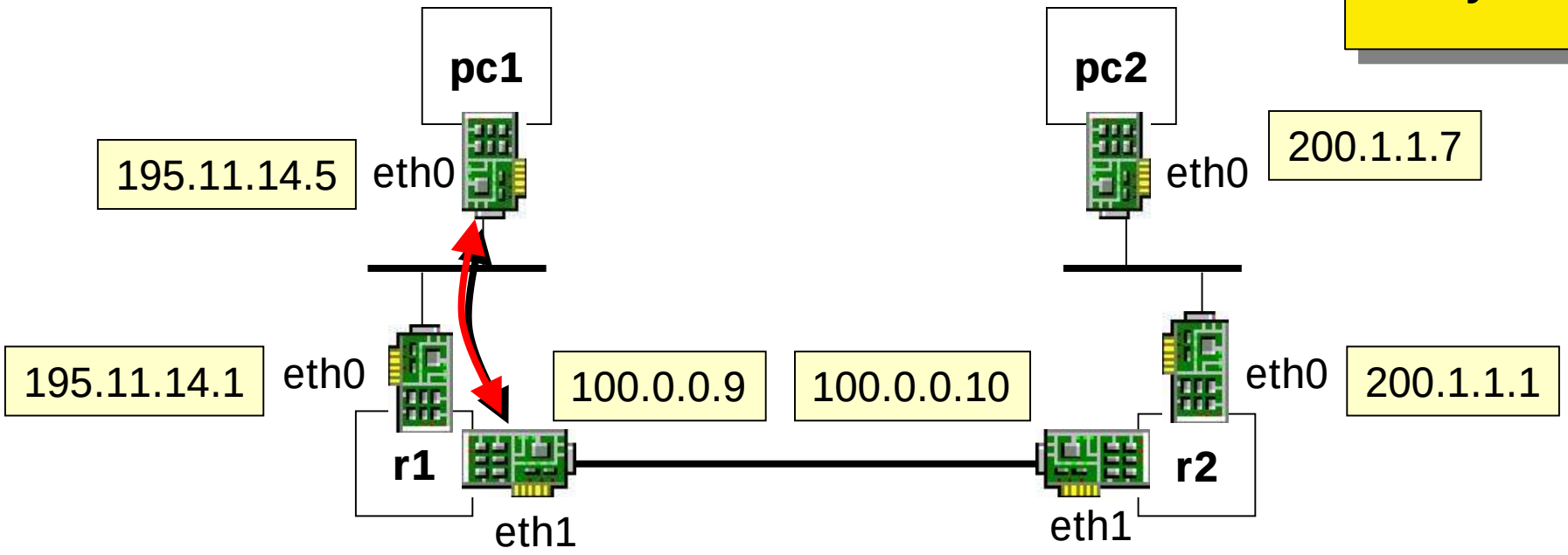


step 3 – testing connectivity

```
pc1:~# ping 100.0.0.9
connect: Network is unreachable
pc1:~#
```

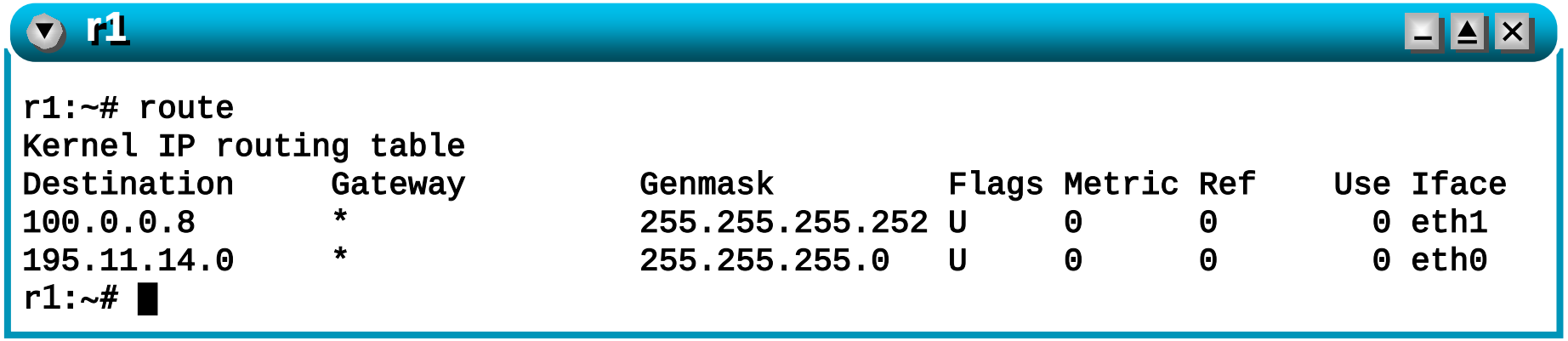
interfaces on different domains cannot be reached

can you tell why?



step 3 – inspecting routing tables

- both routers and pcs don’t know how to reach networks that are not directly connected to them



- directly connected networks are automatically inserted into the routing table when the corresponding interface is brought up
- this is a common behavior of all ip devices (even real-world routers!)

step 4 – default routes on pcs

- to fix the problem we could specify the default route on the pcs: “through this gateway (ip number) you can reach all the other networks”

pc1

```
pc1:~# route add default gw 195.11.14.1
pc1:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
195.11.14.0     *               255.255.255.0   U     0      0        0 eth0
default         195.11.14.1     0.0.0.0         UG    0      0        0 eth0
pc1:~#
```

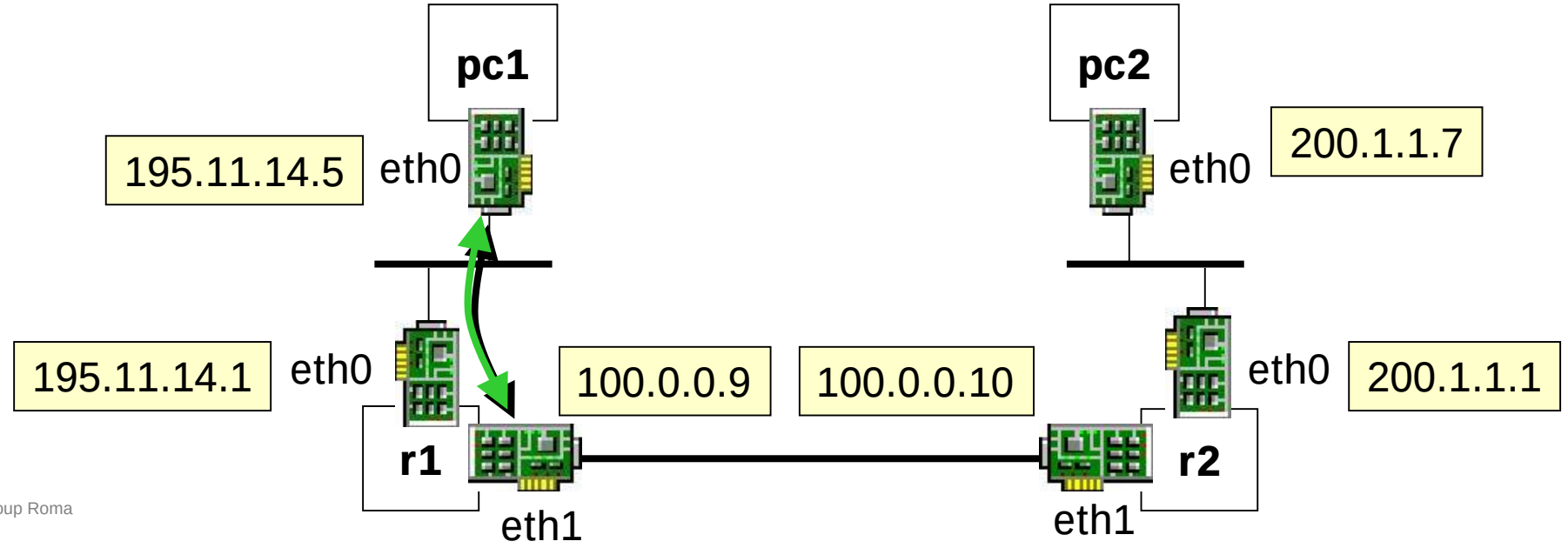
step 4 – default routes on pcs: test

pc1

```
pc1:~# ping 100.0.0.9
PING 100.0.0.9 (100.0.0.9) 56(84) bytes of data.
64 bytes from 100.0.0.9: icmp_seq=1 ttl=64 time=0.451 ms
64 bytes from 100.0.0.9: icmp_seq=2 ttl=64 time=0.299 ms
64 bytes from 100.0.0.9: icmp_seq=3 ttl=64 time=0.320 ms

--- 100.0.0.9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.299/0.356/0.451/0.070 ms
pc1:~#
```

the “backbone interface” of **r1** is reachable



step 4 – default routes on pcs: test

pc1

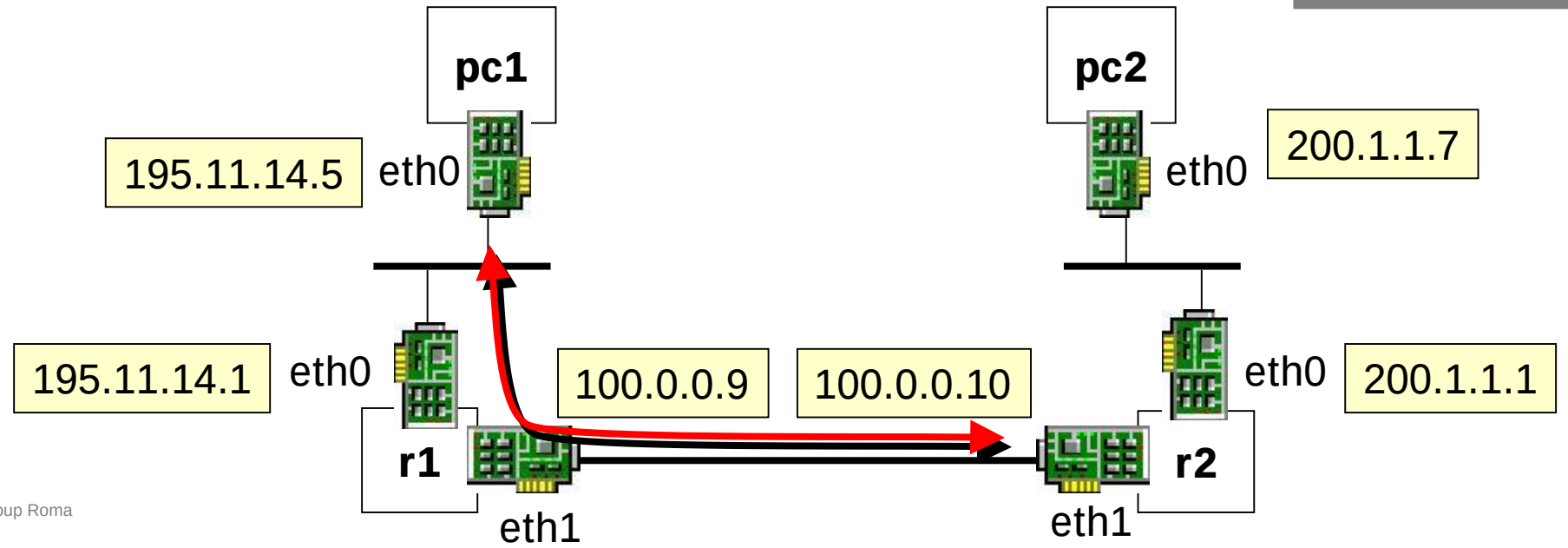
```
pc1:~# ping 100.0.0.10
PING 100.0.0.10 (100.0.0.10) 56(84) bytes of data.

--- 100.0.0.10 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6105ms

pc1:~#
```

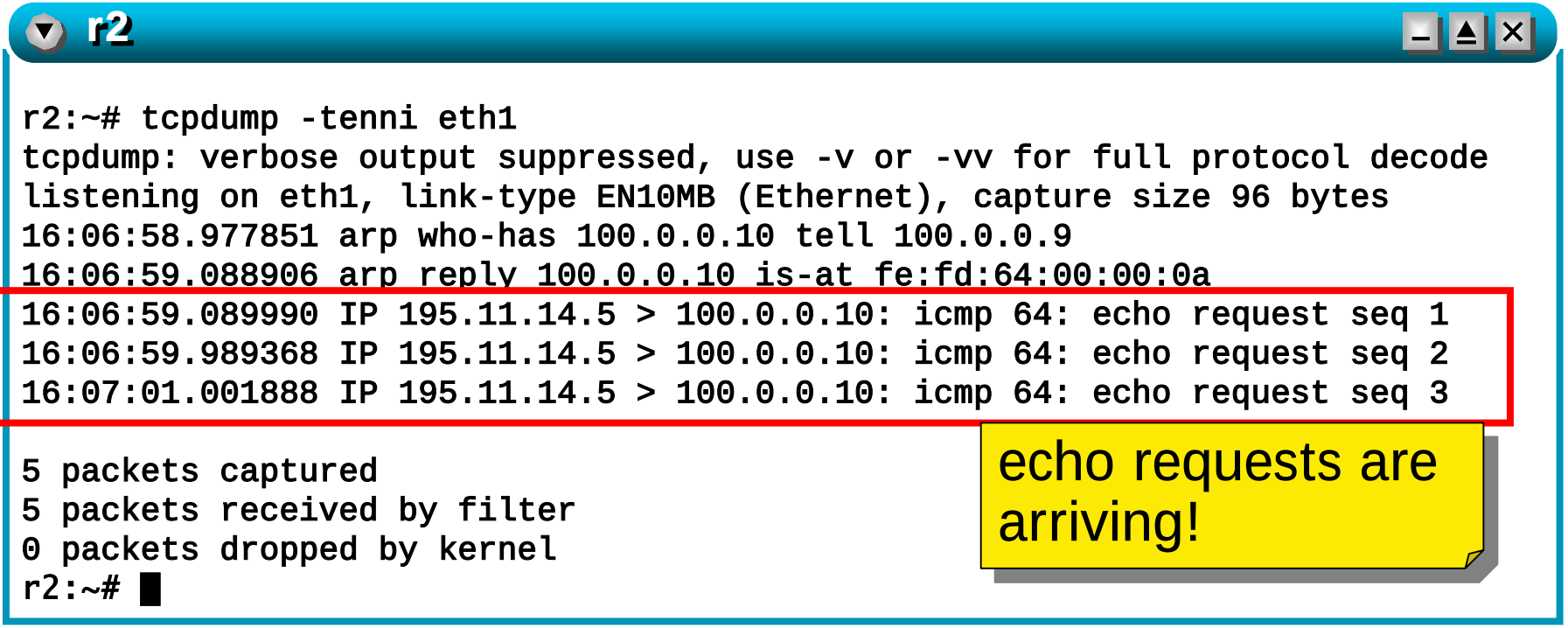
interfaces on **r2** seem unreachable!

can you tell why?



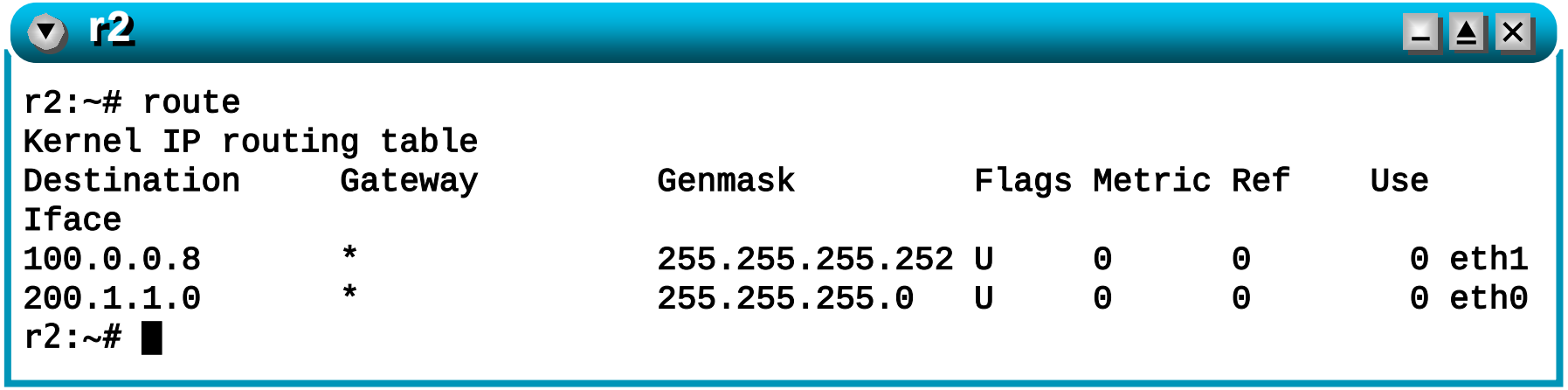
step 4 – let's inspect the network

- do echo request packets reach **`r2`**?
- let's check…
  - while pinging from **`pc1`** 100.0.0.10 sniff on interface **`eth1`** of **`r2`**

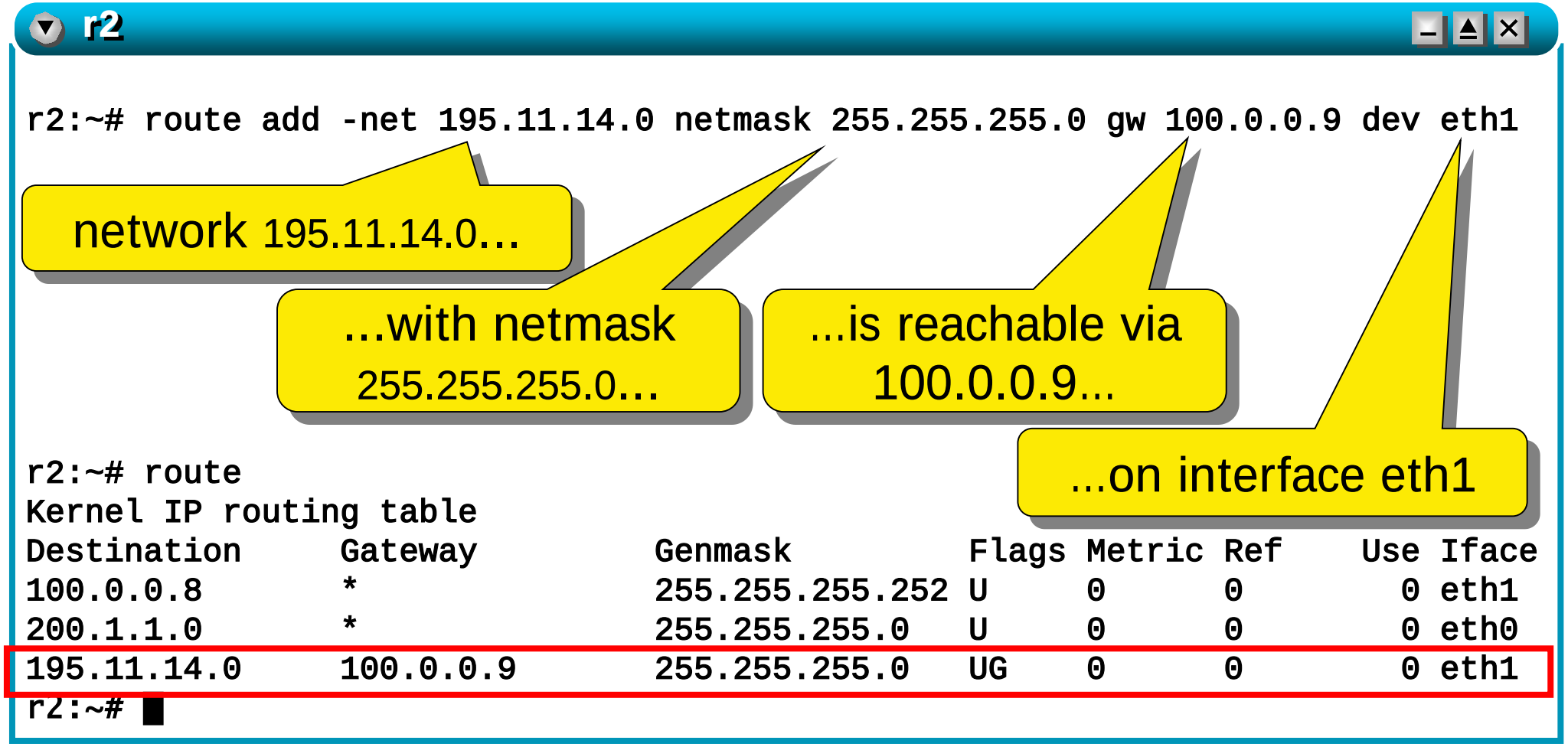


step 4 – **r2**'s routing table

- **pc1**'s address is 195.11.14.5
- **r2** does not know how to reach such an address.
- echo requests arrive to **r2** but **r2** does not know where echo replies should be forwarded!
- somebody should teach **r2** how to reach **pc1**
- we may insert a static route into the routing table of **r2**

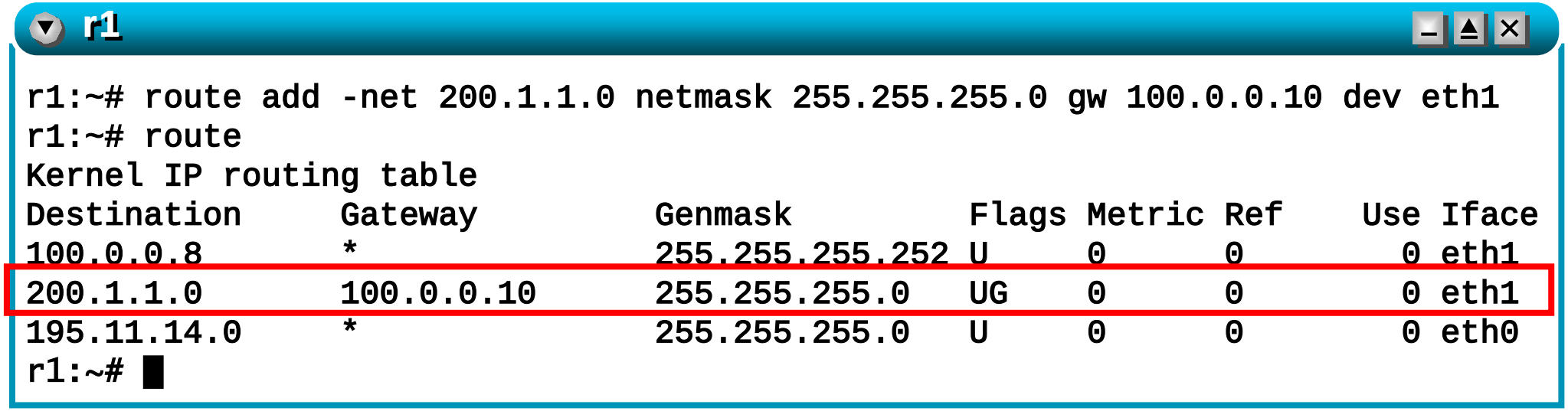


step 5 – configuring a static route



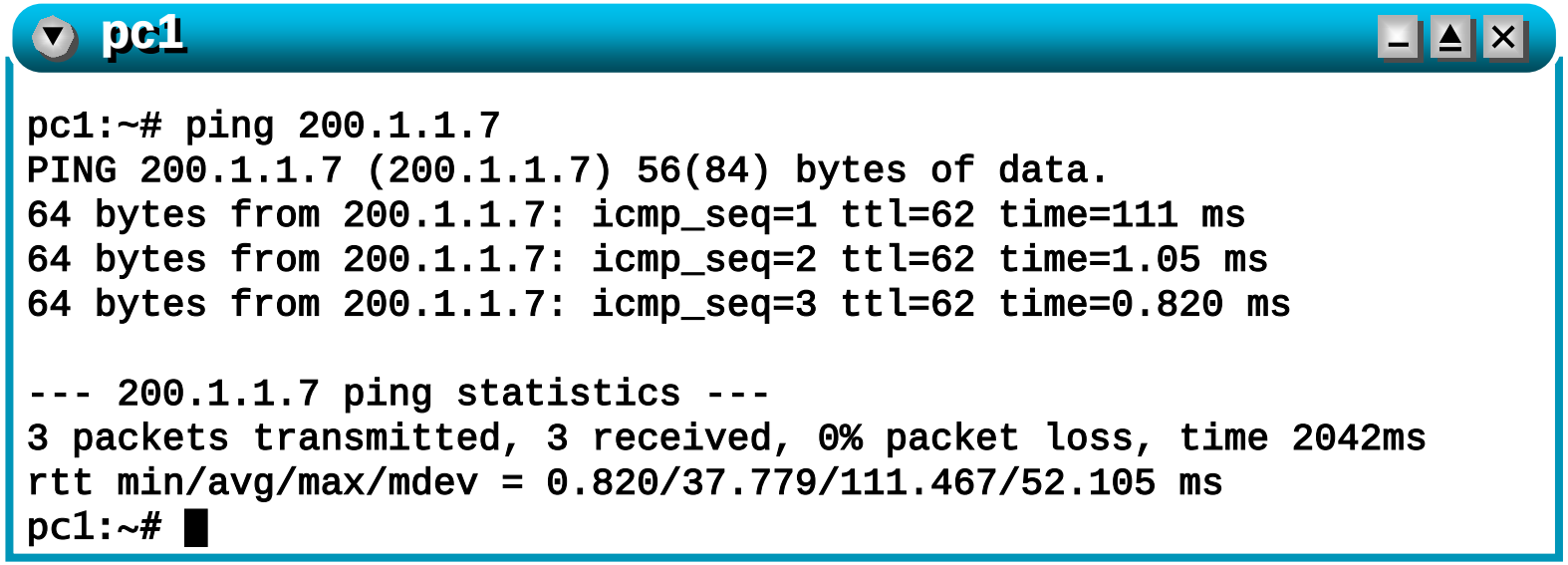
step 5 – configuring a static route

- a similar configuration should be deployed on **`r1`**



step 5 – testing static routes

- the PCs can now reach each other



proposed exercises

- the default route can be statically configured by using

```
route add default gw 195.11.14.1 dev eth0
```

- can you give a command to configure a static route that is equivalent to the default route?

```
route add -net __ netmask __ gw __ dev __
```

proposed exercises

- not all the routing tables contain a default route
- the network of this lab is so simple that routers **`r1`** and **`r2`** can be also configured to exclusively use default routes
- try such a configuration and test it