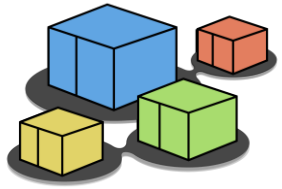


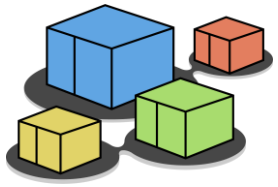
Kathará

kathara lab

static-routing

Version	1.2
Author(s)	L. Ariemma, G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini
E-mail	contact@kathara.org
Web	https://www.kathara.org/
Description	an example of configuration of static routes – kathara version of netkit lab static-routing vers. 2.2

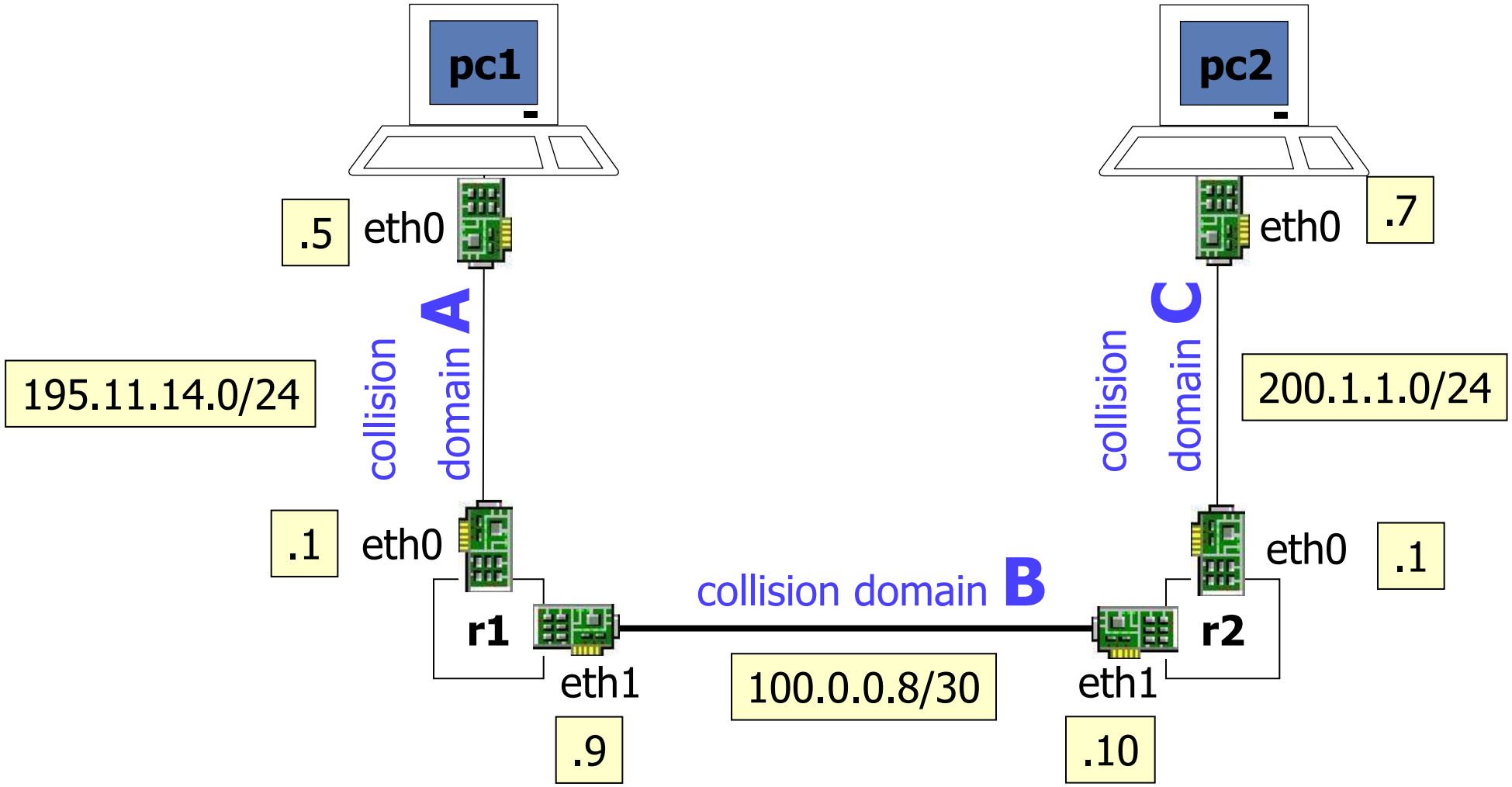
© Computer Networks Research Group
Roma Tre



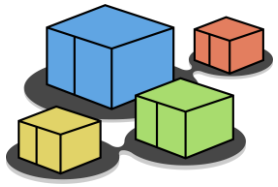
Copyright notice

- 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.

Step 1 – Network topology



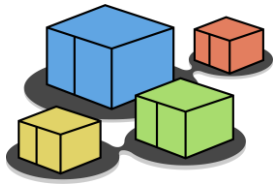
© Computer Networks Research Group
Roma Tre



Step 2 – The lab

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

© Computer Networks Research Group
Roma Tre



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

```
ip address add 195.11.14.5/24 dev eth0
```

r1.startup

```
ip address add 195.11.14.1/24 dev eth0
ip address add 100.0.0.9/30 dev eth1
```

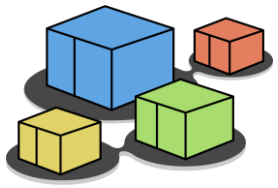
pc2.startup

```
ip address add 200.1.1.7/24 dev eth0
```

r2.startup

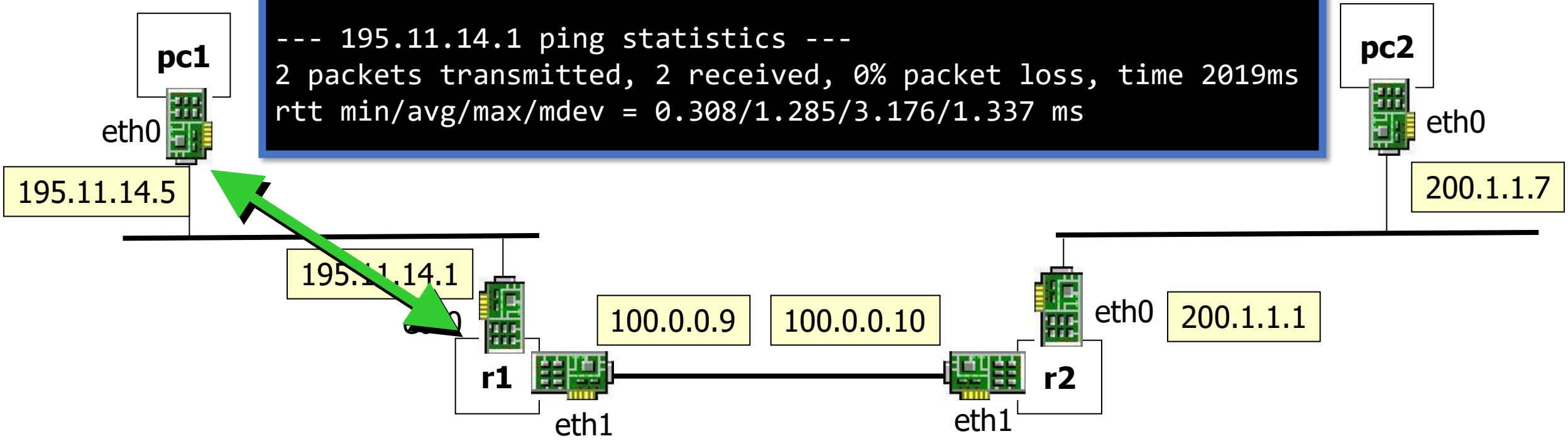
```
ip address add 200.1.1.1/24 dev eth0
ip address add 100.0.0.10/30 dev eth1
```

© Computer Networks Research Group
Roma Tre



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=3.17 ms
64 bytes from 195.11.14.1: icmp_seq=2 ttl=64 time=0.371 ms

--- 195.11.14.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2019ms
rtt min/avg/max/mdev = 0.308/1.285/3.176/1.337 ms
```

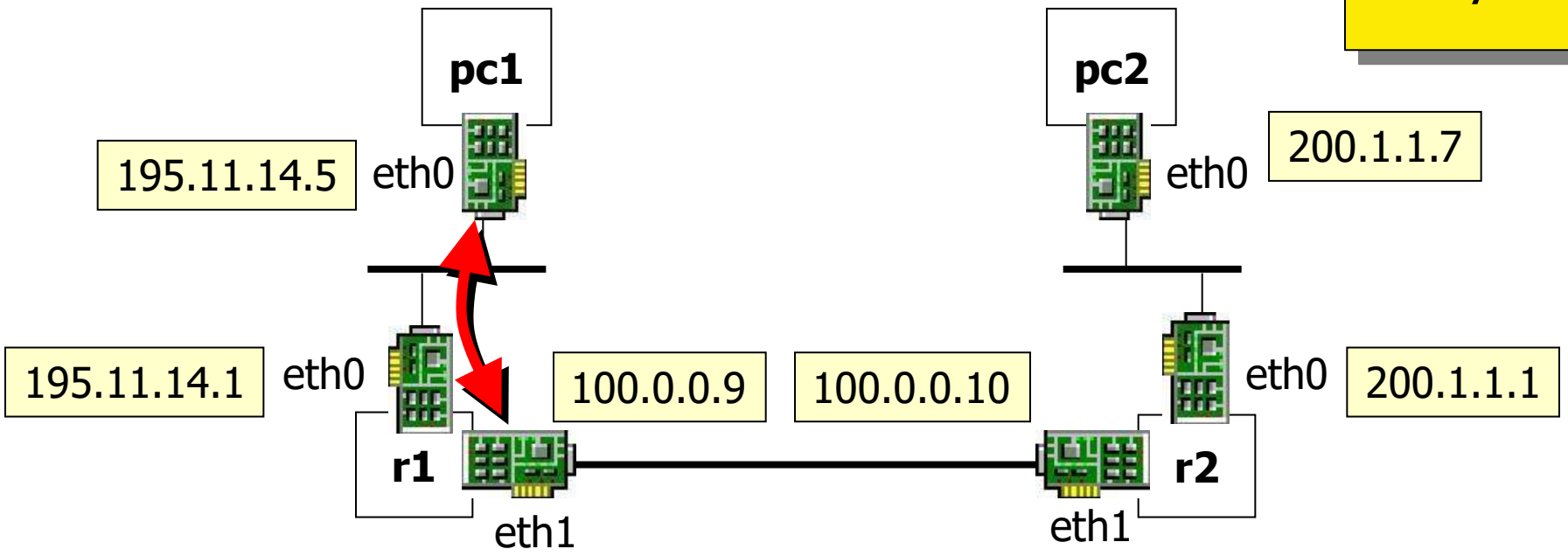


Step 3 – Testing connectivity

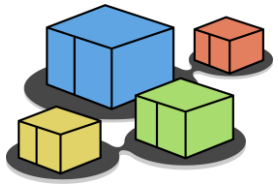
```
root@pc1:~$ ping 100.0.0.9
connect: Network is unreachable
```

interfaces on different domains cannot be reached

can you tell why?



© Computer Networks Research Group
Roma Tre



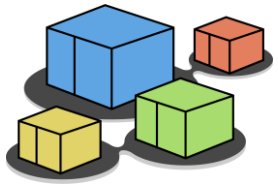
Step 3 – Inspecting routing tables

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

```
root@r1:~$ ip route
100.0.0.8/30 dev eth1 proto kernel scope link src 100.0.0.9
195.11.14.0/24 dev eth0 proto kernel scope link src 195.11.14.1
```

- 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!)

© Computer Networks Research Group
Roma Tre

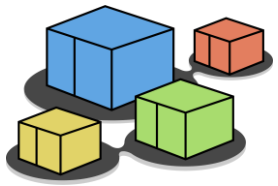


Step 4 – Default routes on PCs

- 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”

```
root@pc1:~$ ip route add default via 195.11.14.1 dev eth0
root@pc1:~$ ip route
default via 195.11.14.1 dev eth0
195.11.14.0/24 dev eth0 proto kernel scope link src 195.11.14.5
```

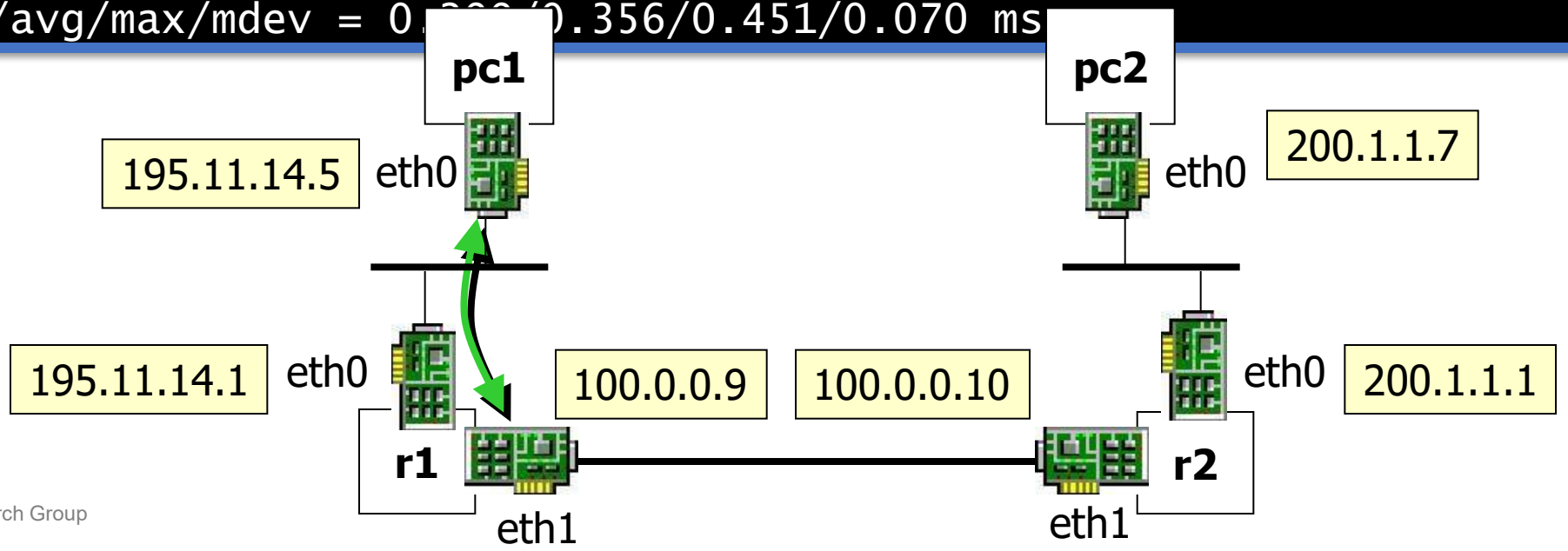
© Computer Networks Research Group
Roma Tre



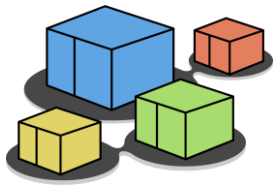
Step 4 – Default routes on PCs: test

```
root@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.[illegible]0.356/0.451/0.070 ms
```

the "backbone interface" of **`r1`** is reachable



© Computer Networks Research Group
Roma Tre

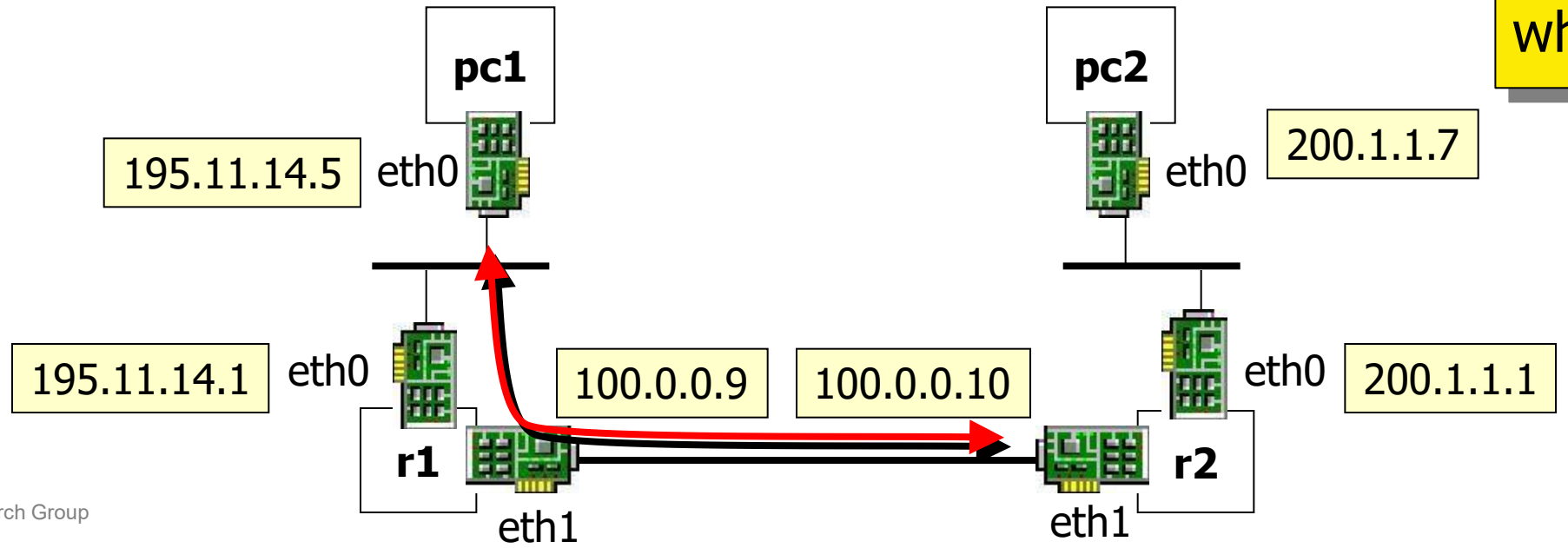


Step 4 – Default routes on PCs: test

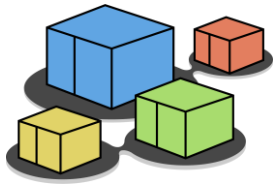
```
root@pc1:~$ ping 100.0.0.10
PING 100.0.0.10 (100.0.0.10) 56(84) bytes of data.

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

interfaces on **r2** seem unreachable!

can you tell why?



© Computer Networks Research Group
Roma Tre



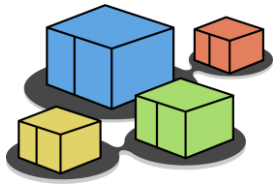
Step 4 – Let's inspect the network

- Do echo request packets reach **`r2`**?
- While pinging from **`pc1`**, sniff interface eth1 of **`r2`**

```
root@r2:~$ tcpdump -tenni eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
16:06:58.977851 arp who-has 100.0.0.10 tell 100.0.0.9
16:06:59.088906 arp reply 100.0.0.10 is-at fe:fd:64:00:00:0a
16:06:59.089990 IP 195.11.14.5 > 100.0.0.10: icmp 64: echo request seq 1
16:06:59.989368 IP 195.11.14.5 > 100.0.0.10: icmp 64: echo request seq 2
16:07:01.001888 IP 195.11.14.5 > 100.0.0.10: icmp 64: echo request seq 3

5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

echo requests are arriving!

© Computer Networks Research Group
Roma Tre

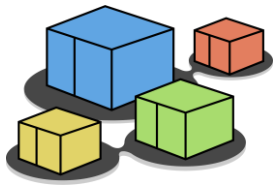


Step 4 – `r2`'s routing table

- **`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`**

```
root@r2:~$ ip route
100.0.0.8/30 dev eth1 proto kernel scope link src 100.0.0.10
200.1.1.0/24 dev eth0 proto kernel scope link src 200.1.1.1
```

© Computer Networks Research Group
Roma Tre



Step 5 – Configuring a static route

```
root@r2:~$ ip route add 195.11.14.0/24 via 100.0.0.9 dev eth1
```

network 195.11.14.0...

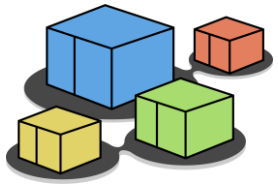
...with netmask 24...

...is reachable via 100.0.0.9...

...on interface eth1

```
root@r2:~$ ip route
100.0.0.8/30 dev eth1 proto kernel scope link src 100.0.0.10
195.11.14.0/24 via 100.0.0.9 dev eth1
200.1.1.0/24 dev eth0 proto kernel scope link src 200.1.1.1
```

© Computer Networks Research Group
Roma Tre

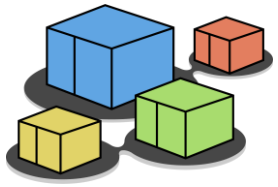


Step 5 – Configuring a static route

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

```
root@r1:~$ ip route add 200.1.1.0/24 via 100.0.0.10 dev eth1
root@r1:~$ ip route
100.0.0.8/30 dev eth1 proto kernel scope link src 100.0.0.9
195.11.14.0/24 dev eth0 proto kernel scope link src 195.11.14.1
200.1.1.0/24 via 100.0.0.10 dev eth1
```

© Computer Networks Research Group
Roma Tre

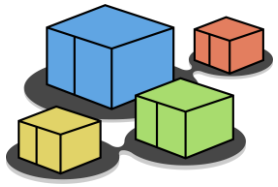


Step 5 – Testing static routes

- The PCs can now reach each other

```
root@pc1:~$ ping 200.1.1.7
PING 200.1.1.7 (200.1.1.7) 56(84) bytes of data.
64 bytes from 200.1.1.7: icmp_seq=1 ttl=62 time=111 ms
64 bytes from 200.1.1.7: icmp_seq=2 ttl=62 time=1.05 ms
64 bytes from 200.1.1.7: icmp_seq=3 ttl=62 time=0.820 ms

--- 200.1.1.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2042ms
rtt min/avg/max/mdev = 0.820/37.779/111.467/52.105 ms
```

© Computer Networks Research Group
Roma Tre



Proposed exercises

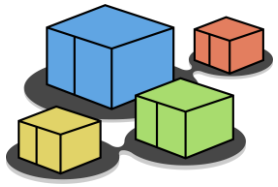
- The default route can be statically configured by using

```
ip route add default via 195.11.14.1 dev eth0
```

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

```
ip route add __/__ via __ dev __
```

© Computer Networks Research Group
Roma Tre



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

© Computer Networks Research Group
Roma Tre