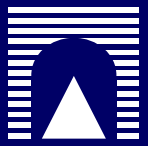


UNIVERSITY OF ROMA TRE
Department of Computer Science and Automation

Emulating Computer Networks with Netkit

Massimo Rimondini

Computer Networks Research Group
http://www.dia.uniroma3.it/~compunet

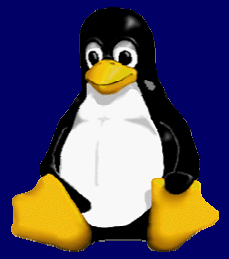
4th International Workshop on Internet Performance, Simulation, Monitoring, and Measurement

Copyright notice

- All the 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 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 a written agreement
- Authors assume no responsibility for the contents of this material, which may be subject to changes
- This copyright notice must always be redistributed together with the material, or its portions

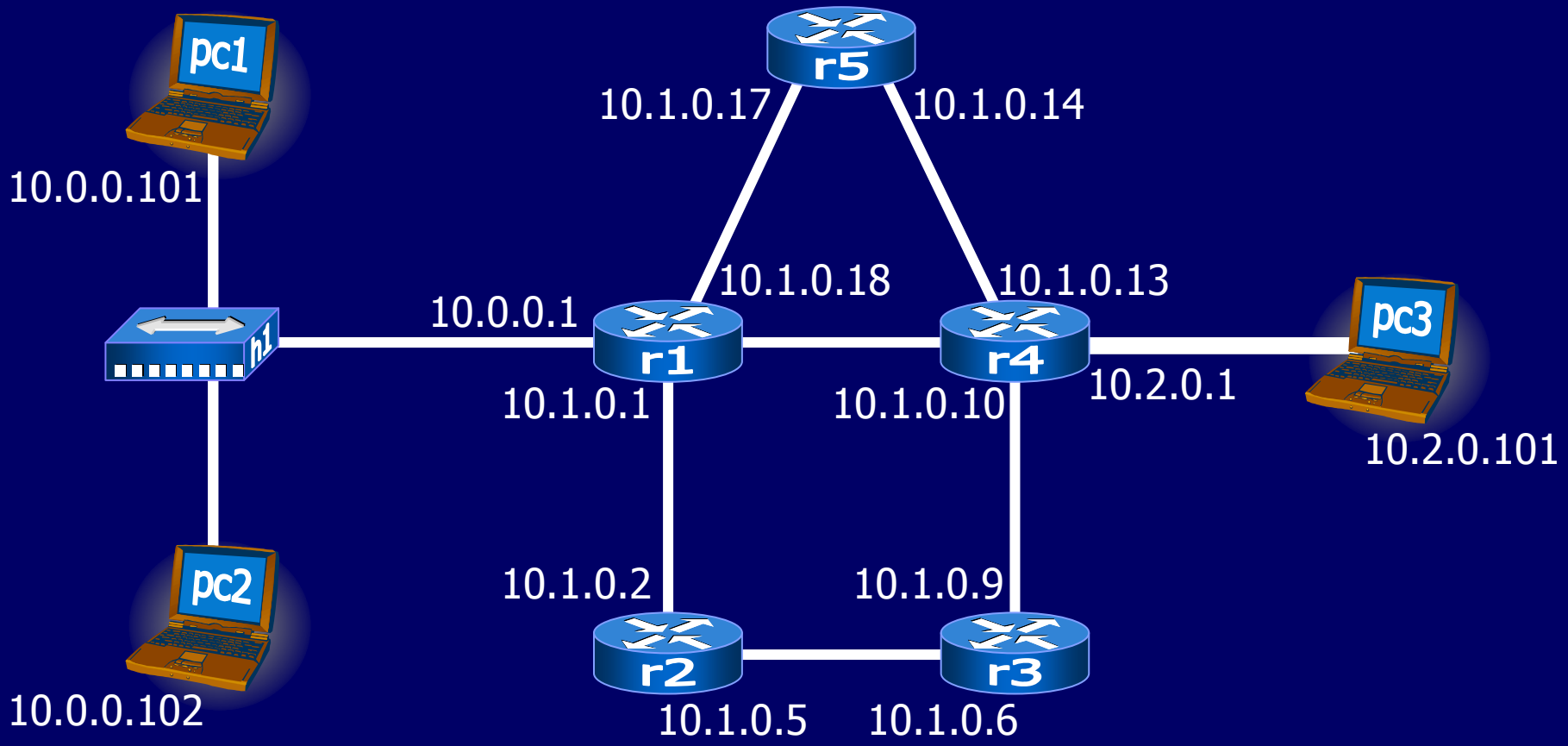
Prerequisites

- Very basic knowledge of the Linux OS



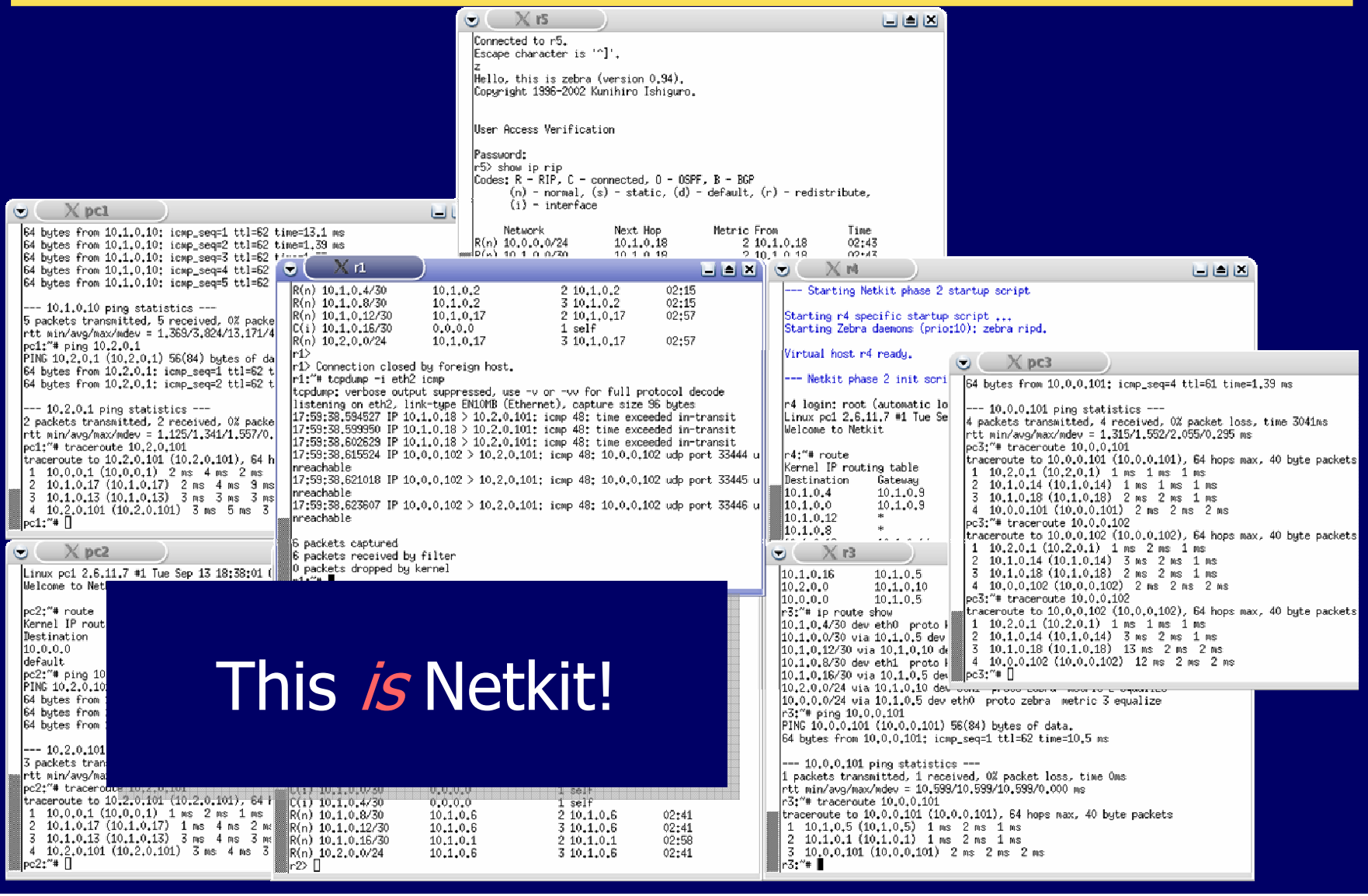
- Networking
  - iso-osi stack
  - routing protocols (rip, bgp)
  - basic network tools

What's this *Natkeet* thing?



This is *not* Netkit...

What's this *Natkeet* thing?



Yet another boring emulator toy?

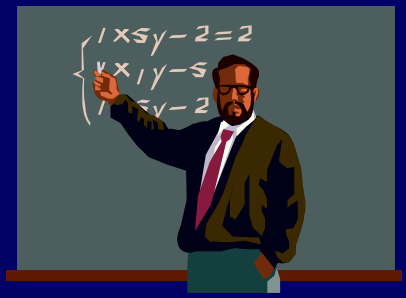
- From Wikipedia:
  - Bochs, portable open source x86 and AMD64 PCs emulator
  - FAUmachine
  - Microsoft Virtual PC
  - Microsoft Virtual Server
  - OPEN COLINUX - Linux inside Windows
  - Parallels
  - QEMU
  - SVISTA
  - TRANGO Real-time Embedded Hypervisor
  - twoOStwo
  - User-mode Linux
  - Virtuozzo
  - VM from IBM, apparently the first true virtual machine system and still in wide use today.
  - VMWare
  - Xen

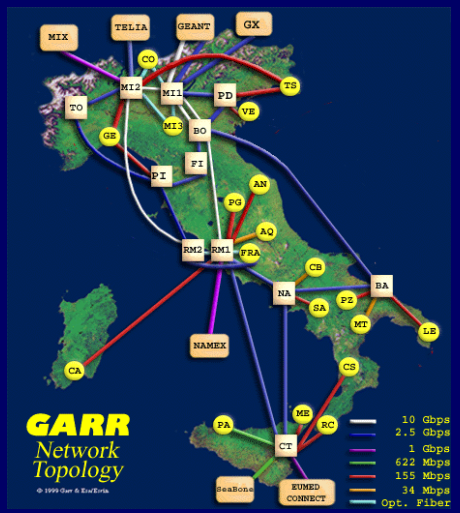
Yet another boring emulator toy?

Product	License	Pros	Cons
Xen	Open Source	Performance	Requires replacing the host kernel Requires porting the guest OS kernel
VMWare	Commercial	Near native performance	Complex configuration
Bochs	Open Source	Good support for many guest OSes	Performance
QEMU	(almost) Open Source	Emulates multiple architectures Comes with an accelerator module	Requires privileges to be installed Not lightweight
Plex86	Open Source	Lightweight	Development stalled

Success stories involving Netkit

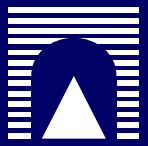
- Networking courses
  - routing protocols
  - application level services (dns, web, email, etc.)
- Emulation of the italian academic research network (GARR)
  - assignment of OSPF weights





Outline

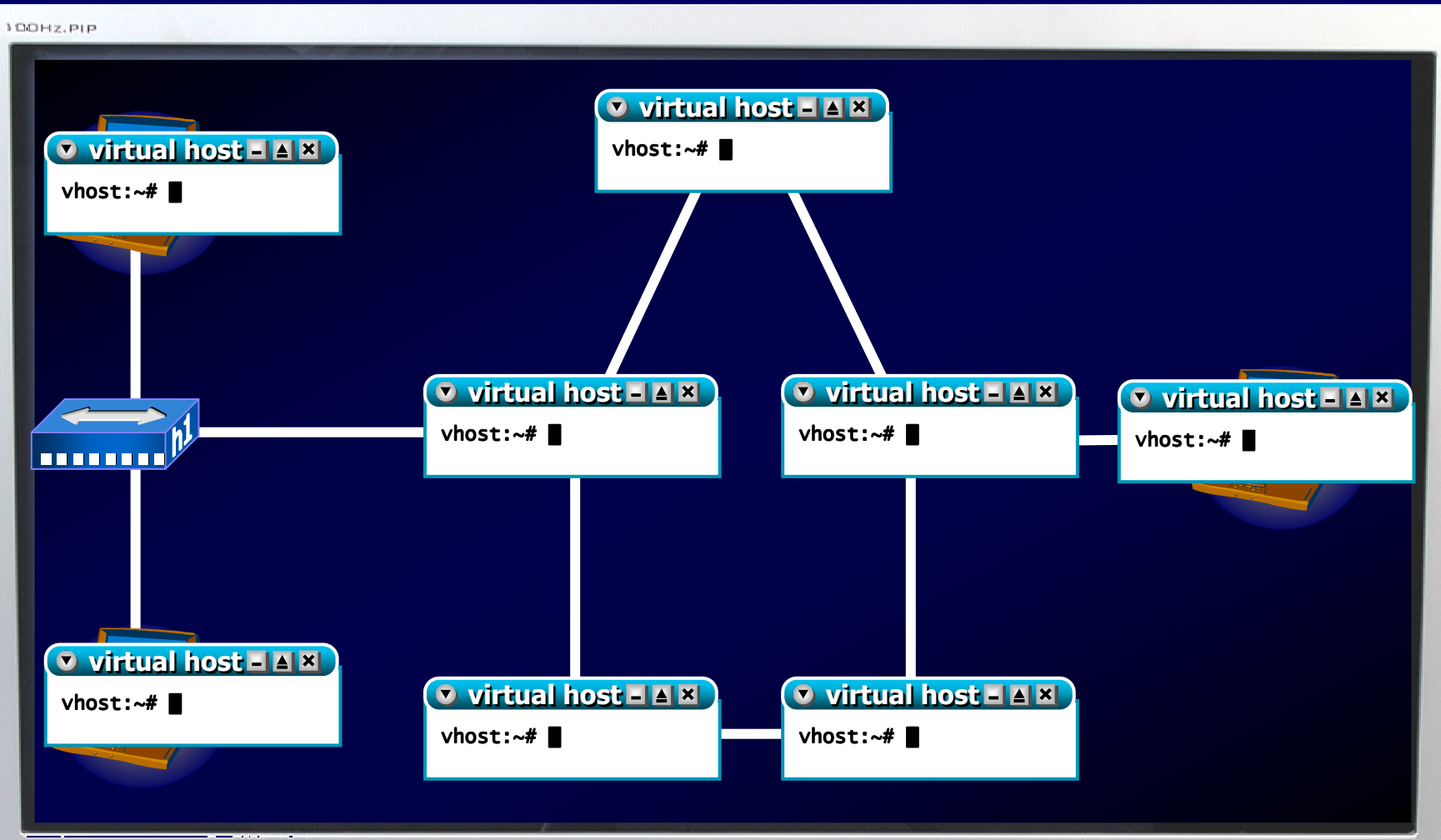
- Understanding Netkit
  - Architecture overview
- Setting up Netkit
  - Download and installation
- Using Netkit
  - Getting acquainted with Netkit commands
  - How to prepare Netkit labs
- Sample scenarios
  - A virtual network running BGP



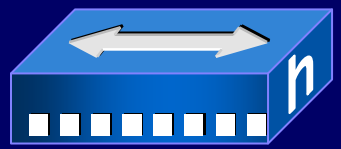
UNIVERSITY OF ROMA TRE
Department of Computer Science and Automation

Understanding Netkit

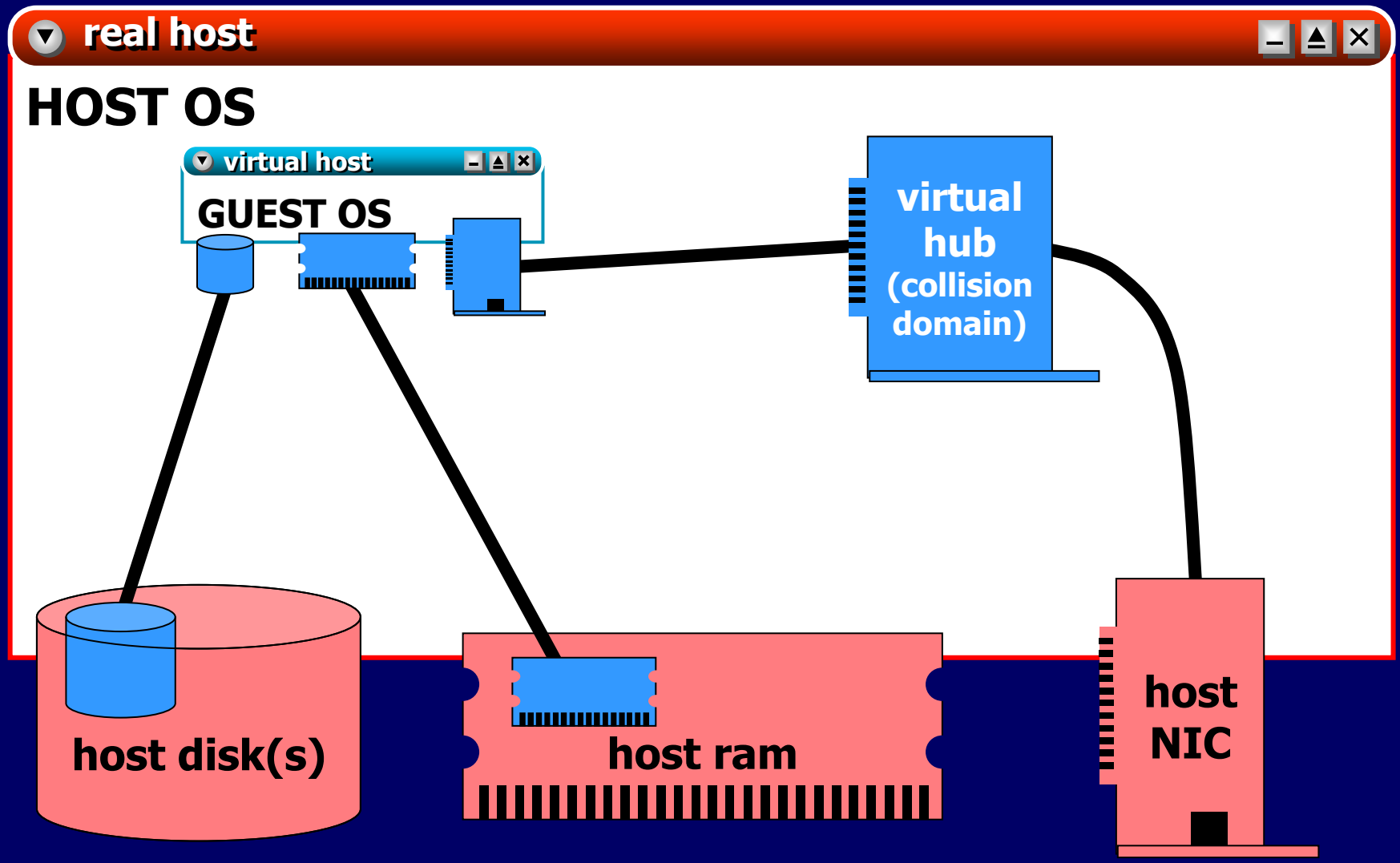
A Netkit network



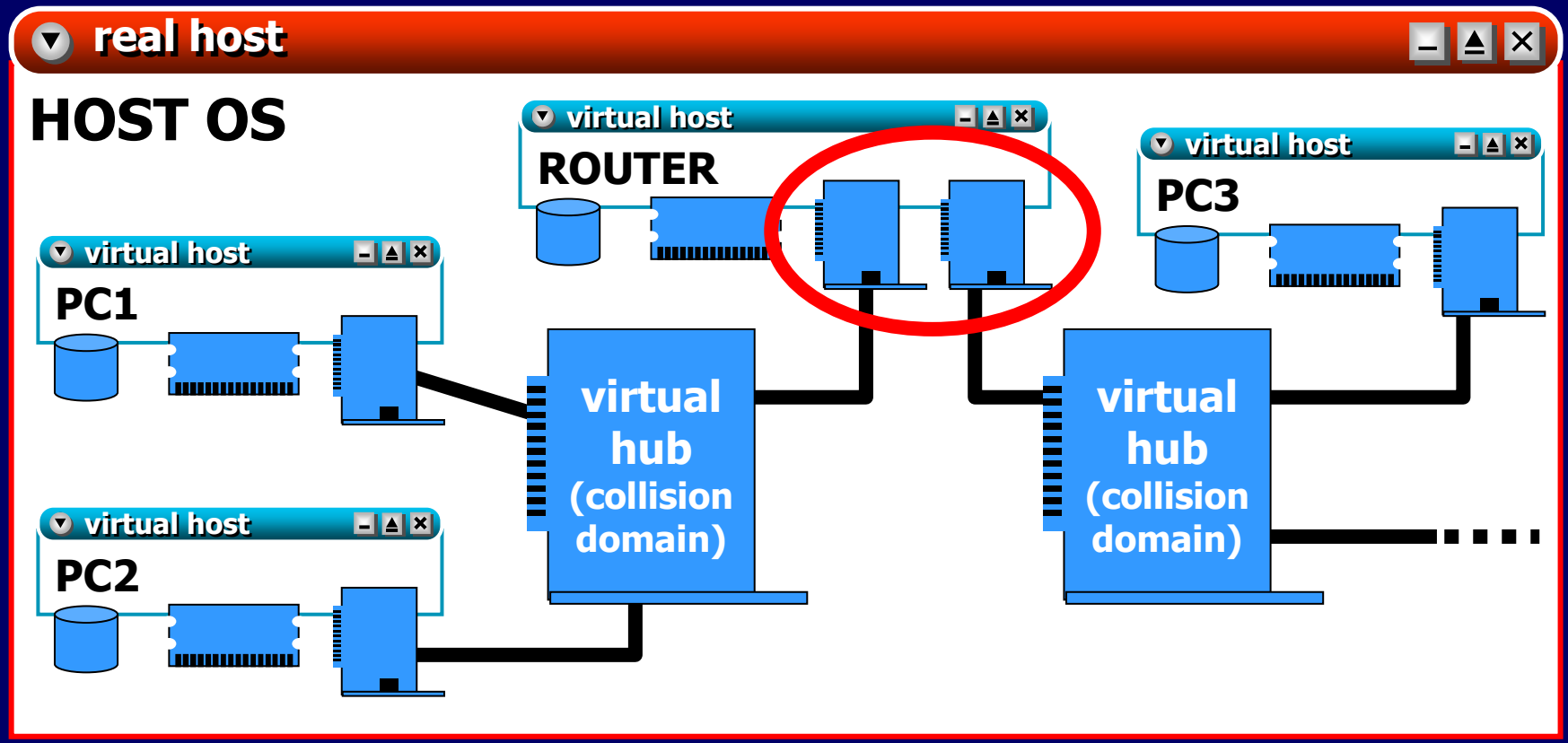
A Netkit network



A closer look at virtual machines



Interconnecting virtual machines

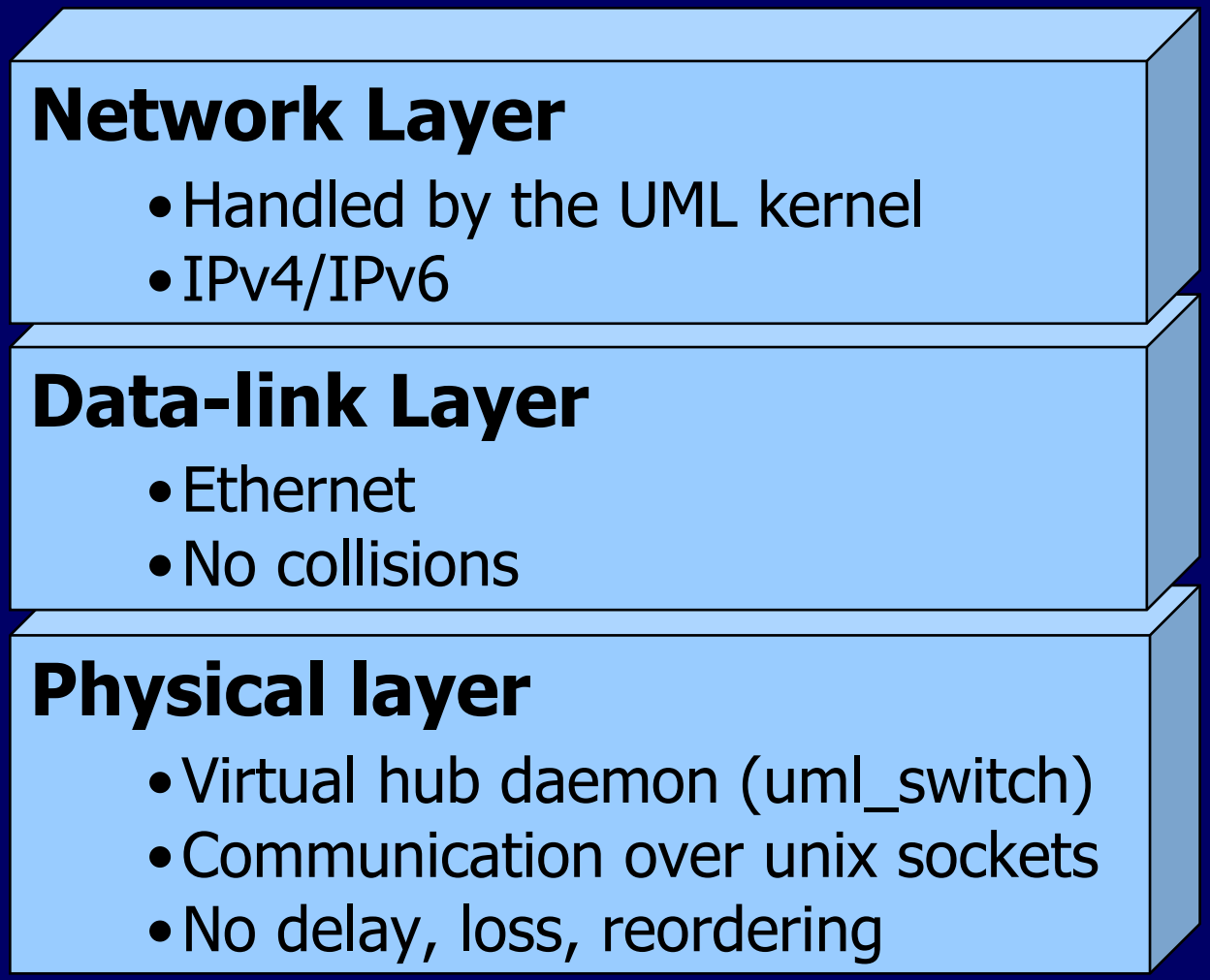


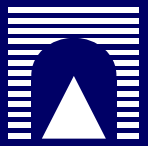
A few technicalities

- Virtual machines run a user-mode-linux kernel
  - Linux kernel compiled for running as a userspace process
  - Thin emulation layer ⇒ better performance
  - http://user-mode-linux.sourceforge.net/
- The guest OS is a Debian GNU/Linux unstable
  - Most popular networking tools are included
  - Copy-On-Write
    - "Damage-free" filesystem

Cables & Interfaces

- Virtual machines can be equipped with an arbitrary number of network interfaces





UNIVERSITY OF ROMA TRE
Department of Computer Science and Automation

Setting up Netkit

System requirements

Host	i386 Linux host
CPU	>600 MHz • lower freqs impact performance
RAM	~10MB for each virtual machine • depends on the amount of emulated memory
Hard disk space	~650 MB + ~1-20 MB for each virtual machine • depends on vm usage
Others	ext2/ext3 filesystems preferred

A tiresome procedure ☺

- Just three steps

1. Download @ http://www.netkit.org
   - `netkit-2.4.tar.bz2`
   - `netkit-filesystem-F2.2.tar.bz2`
   - `netkit-kernel-K2.2.tar.bz2`

2. **`cd`** to a directory of your choice and unpack the files

host machine

```
foo@host:~$ cd netkit
foo@host:~/netkit$ tar xjf netkit-2.4.tar.bz2
foo@host:~/netkit$ tar xjf netkit-filesystem-F2.2.tar.bz2
foo@host:~/netkit$ tar xjf netkit-kernel-K2.2.tar.bz2
foo@host:~$
```

A tiresome procedure ☺

3. Set some environment variables
   - set **NETKIT_HOME** to the path where you installed Netkit
   - set **PATH** to the string "**$PATH:$NETKIT_HOME/bin**"
   - set **MANPATH** to the string "**:$NETKIT_HOME/man**"

   Example (using bash):

host machine

```
foo@host:~/netkit$ export NETKIT_HOME=/home/foo/netkit/netkit2
foo@host:~/netkit$ export PATH=$PATH:$NETKIT_HOME/bin
foo@host:~/netkit$ export MANPATH=:$NETKIT_HOME/man
foo@host:~/netkit$
```

That's all folks!

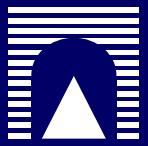
Test your installation

- `cd` to the Netkit directory
- Run `check_configuration.sh`

host machine

```
foo@host:~/netkit$ cd netkit2
foo@host:~/netkit/netkit2$ ./check_configuration.sh
......
[ READY ] Congratulations! Your Netkit setup is now complete!
          Enjoy Netkit!
foo@host:~/netkit/netkit2$
```

- Test failures are accompanied by a short description of the fix

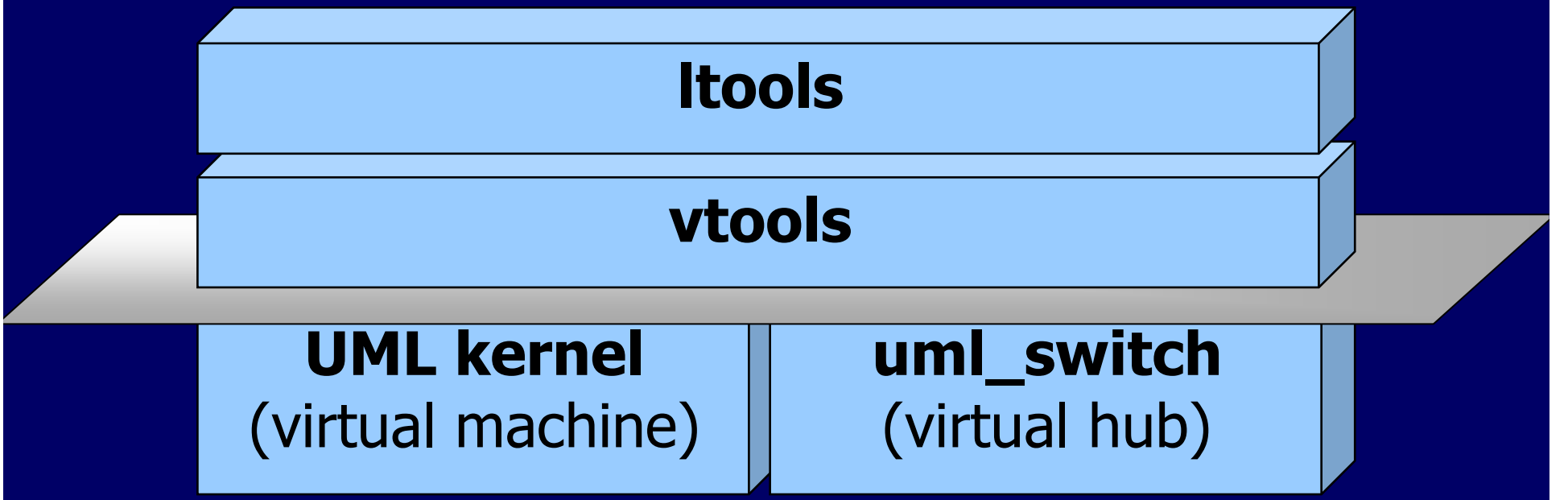


UNIVERSITY OF ROMA TRE
Department of Computer Science and Automation

Using Netkit
(at last…)

A toolkit made up of tools

- Virtual machines can be controlled by using two interfaces



Vtools

vstart	Start a virtual machine with a given configuration
vconfig	Attach a network interface to a running virtual machine
vlist	List running virtual machines
vhalt	Gracefully shut down a virtual machine
vcrash	Kill a virtual machine
vclean	Panic button 

Booting a virtual machine

`vstart [options] MACHINE_NAME`

- Most common options:
  - **`--ethN=collision_domain`**
    - Interfaces on the same collision domain can exchange traffic
  - **`-M memory_amount`**
  - **`-p`**
    - Just show what would be done

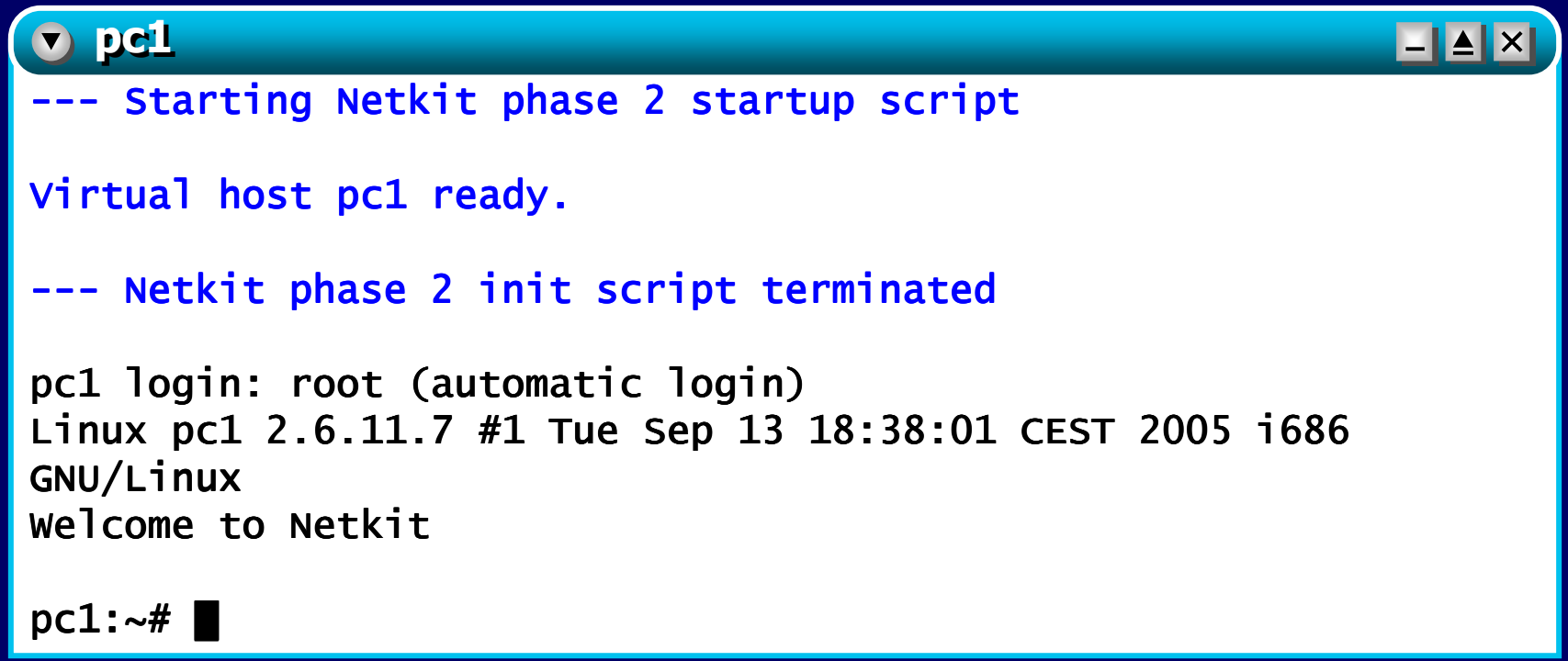
host machine

```
foo@host:~$ vstart --eth0=CD_A --eth1=CD_B -M 256 pc1

============= Starting virtual machine "pc1" =============
   Kernel:     /home/max/netkit2/kernel/netkit-kernel
   Modules:    /home/max/netkit2/kernel/modules
......
```

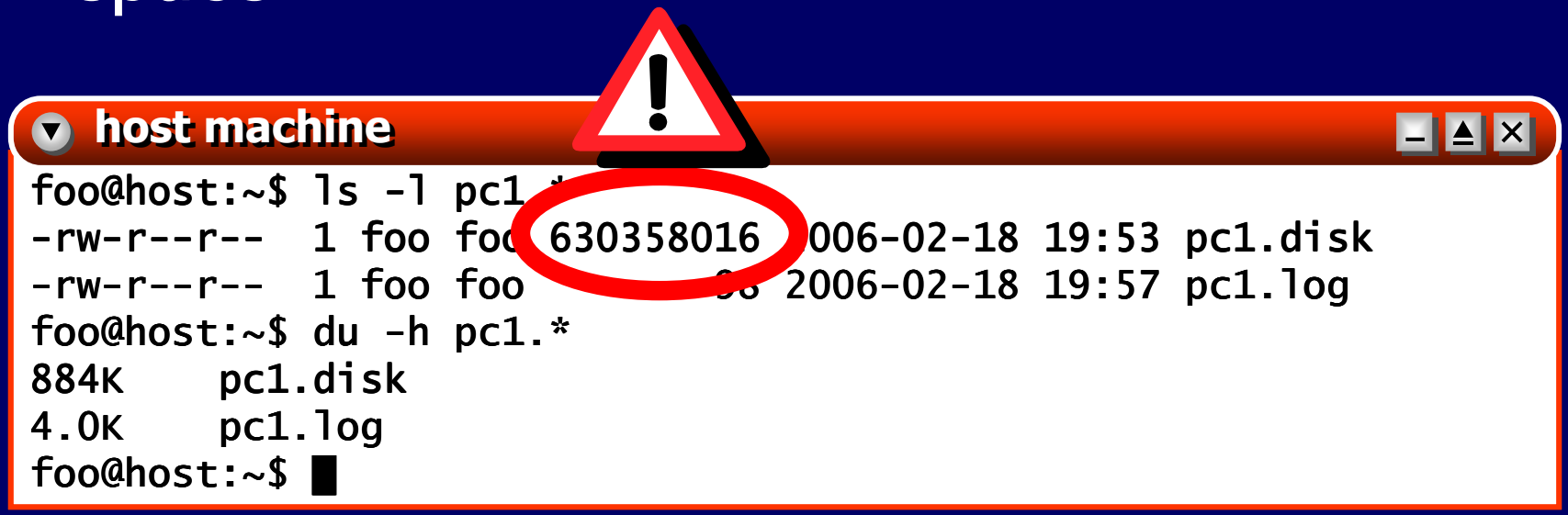
Look: it's booting...

- The virtual machine terminal window automatically pops up



Where VMs write things

- Copy-On-Write: every change to the model filesystem is written to **`pc1.disk`**
- Sparse files: zeros do not consume disk space



Getting rid of a virtual machine

- Gracefully shut down from the inside...

pc1

```
pc1:~# halt

Broadcast message from root (vc/0) (Sat Feb 18 19:46:13 2006):

The system is going down for system halt NOW!
INIT: Switching to runlevel: 0
INIT: Sending processes the TERM signal
```

- ...or from the outside...

host machine

```
foo@host:~$ vhalt pc1
Halting virtual machine "pc1" (PID 8598) owned by foo [...    ]
foo@host:~$
```

Getting rid of a virtual machine

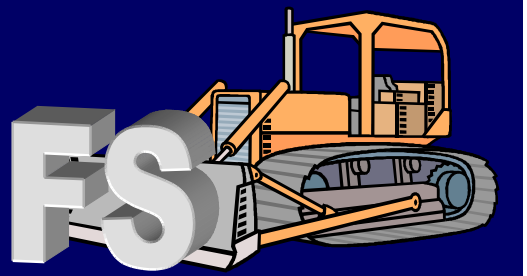
- ...or harshly unplug its power cord

host machine

```
foo@host:~$ vcrash pc1

========== Crashing virtual machine "pc1" (PID 9741) =========
Virtual machine owner: foo
Virtual machine mconsole socket:
/home/foo/.netkit/mconsole/pc1/mconsole
Crashing... done.
foo@host:~$
```

- Quick
- Screws virtual machine filesystem
- Remember: Copy-On-Write



A Netkit lab...

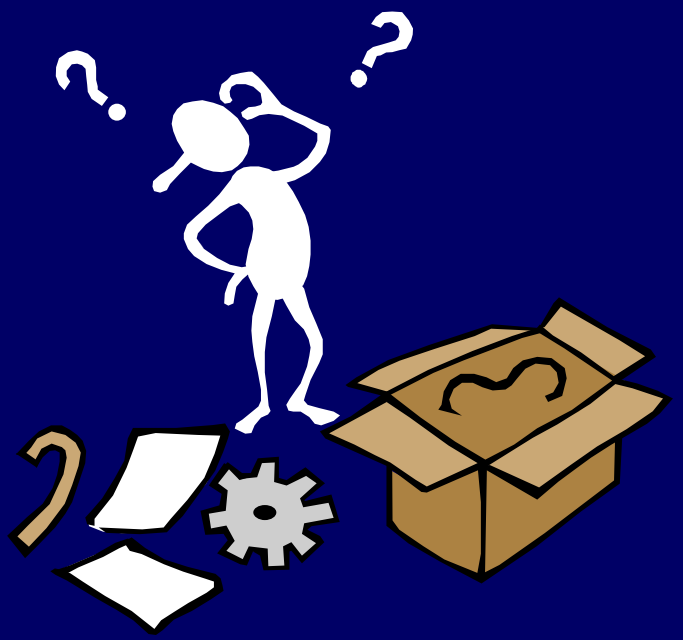
- ...*is* a set of pre-configured virtual machines that can be launched/stopped all together
- ...*consists* of a hierarchy of directories
- ...*allows* to set up complex network experiences
- ...*can be controlled* by using the ltools

Ltools

lstart	Start a Netkit lab or just some of its machines
ltest	Start a Netkit lab in test mode
lhalt	Gracefully shut down (some of) the virtual machines of a lab
lcrash	Kill (some of) the virtual machines of a lab
linfo	Display info about a lab without starting it; sketch the network topology
lclean	Remove temporary files (no panic!)

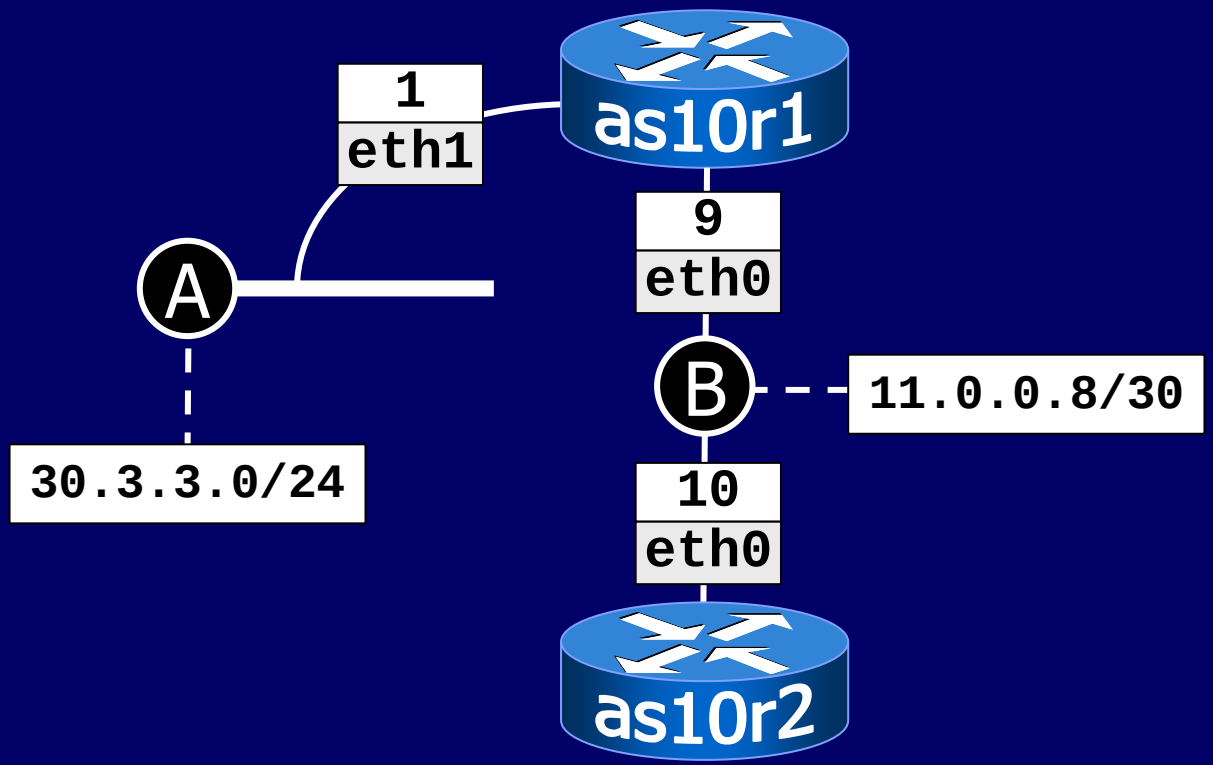
Mastering your own Netkit Lab

- How to prepare a lab:
  1. define topology
  2. assign addresses
  3. configure network services
  4. ...



My Own Lab, scene 1, take 1
Network topology

- Sketch your planned topology before implementing it



LEGEND

B — collision domain name

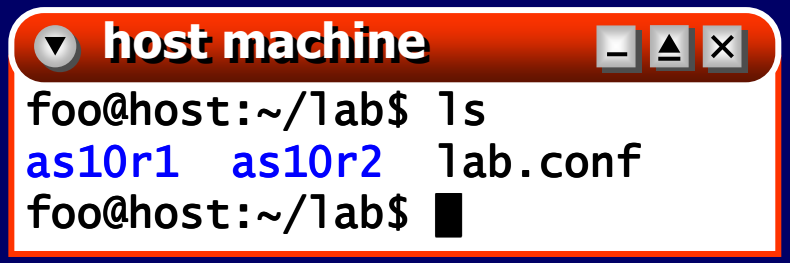
11.0.0.8/30 — network address

9 / eth0 — IP, last byte / Interface

A — *AN

My Own Lab, scene 1, take 2
Network topology

- A lab consists of a hierarchy of directories
- Each (even empty) directory represents a virtual machine



- A lab consisting of two virtual machines (**`as10r1`**, **`as10r2`**)
- Check with **`linfo`**

- Link-level connections are described inside the file **`lab.conf`** (in the lab root)

My Own Lab, scene 1, take 3
Network topology

- **`lab.conf`** syntax
  - **`vm[if]=cd`**
    - **`vm`**: virtual machine name (e.g., **`as10r1`**)
    - **`if`**: interface number (e.g., **`0`**)
    - **`cd`**: collision domain name (arbitrary string)
  - **`vm[opt]=val`**
    - **`opt`**: the name of a vstart option (e.g., **`mem`**)
    - **`val`**: a value for that option
  - Other optional items
    - Informational: **`LAB_DESCRIPTION`**, **`LAB_VERSION`**, **`LAB_AUTHOR`**, **`LAB_EMAIL`**, **`LAB_WEB`**
    - Explicit list of virtual machines (**`machines`**)

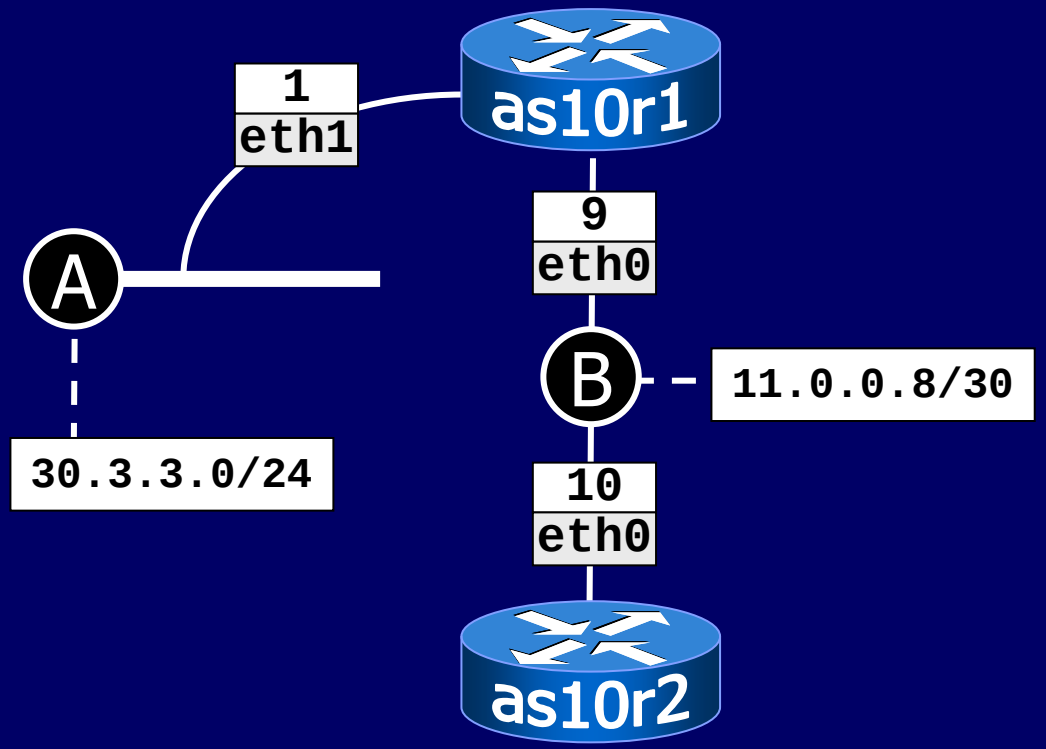
My Own Lab, scene 1, take 4
Network topology

- Sample `lab.conf`

host machine

```
as10r1[0]=B
as10r1[1]=A

as10r2[0]=B
lab.conf
```



My Own Lab, scene 2
Assigning addresses, etc.

- On startup, virtual machine **vm** runs
  - **shared.startup**
  - **vm.startup**
- Interfaces may be configured inside **vm.startup**

host machine

```
ifconfig eth0 11.0.0.9 netmask 255.255.255.252 broadcast 11.0.0.11 up
/etc/init.d/zebra start
as10r1.startup
```

- Network services may be started up inside **vm.startup** too…

My Own Lab, scene 3
Configuring network services

- On startup, Netkit copies the directory hierarchy in **vm/** (on the host) to **/** on virtual machine **vm**

host machine

```
foo@host:~/lab$ find .
.
./as10r1
./as10r1/root
./as10r1/root/this_is_a_file
./as10r2
./lab.conf
foo@host:~/lab$
```

as10r1

```
as10r1:~# pwd
/root
as10r1:~# ls
this_is_a_file
as10r1:~#
```

- Useful to alter routing software configuration files

Ignition!

- To start up a lab:

```
host machine
foo@host:~$ cd lab
foo@host:~/lab$ lstart
```

or

```
host machine
foo@host:~$ lstart -d lab
```

- To crash a lab (use `lhalt` to halt):

```
host machine
foo@host:~$ cd lab
foo@host:~/lab$ lcrash
```

or

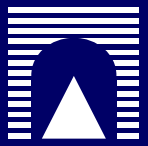
```
host machine
foo@host:~$ lcrash -d lab
```

- To get rid of temporary files (`.disk`, `.log`):

```
host machine
foo@host:~$ cd lab
foo@host:~/lab$ lclean
```

or

```
host machine
foo@host:~$ lclean -d lab
```

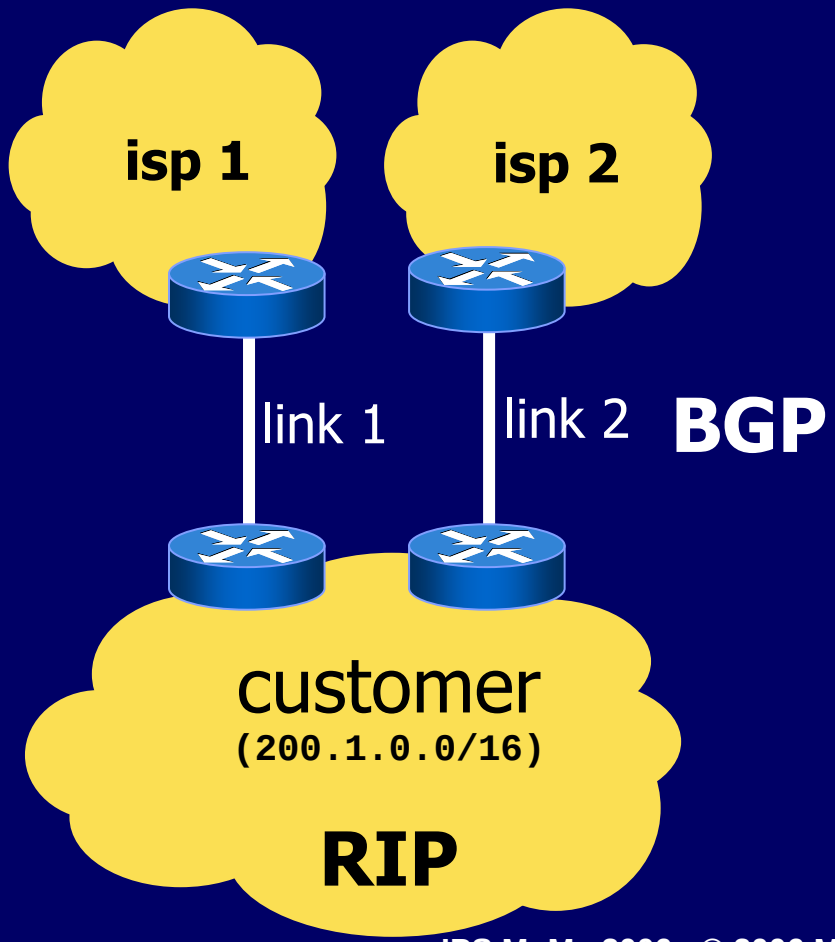


UNIVERSITY OF ROMA TRE
Department of Computer Science and Automation

Playing Around with Netkit

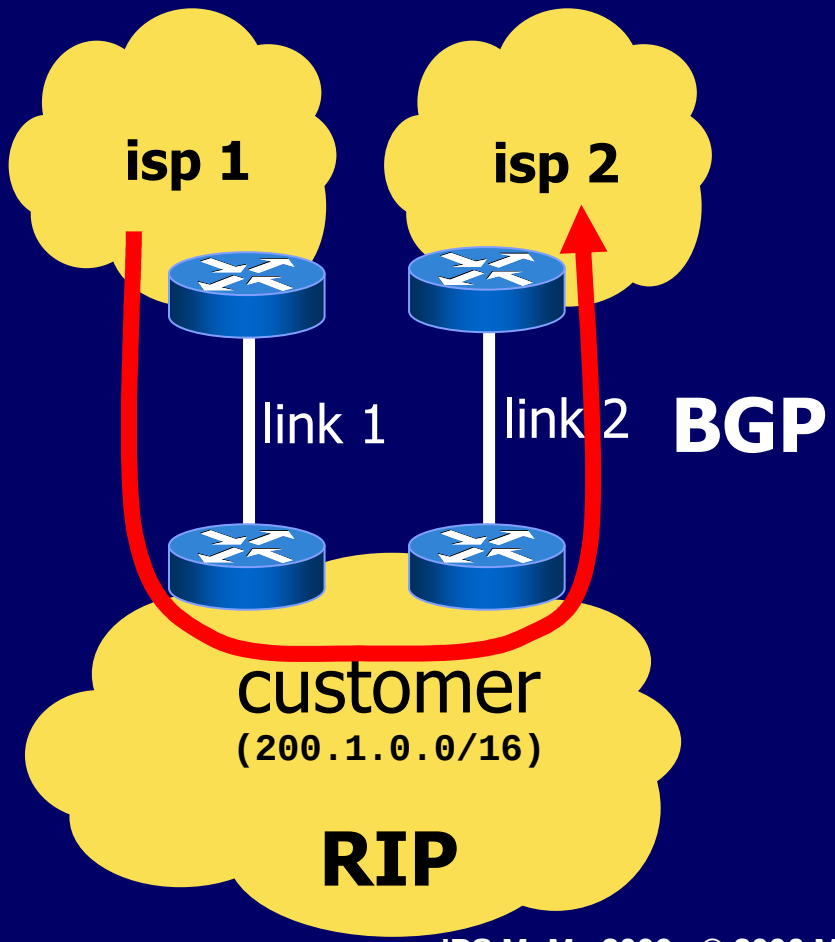
A sample lab

- A multihomed network



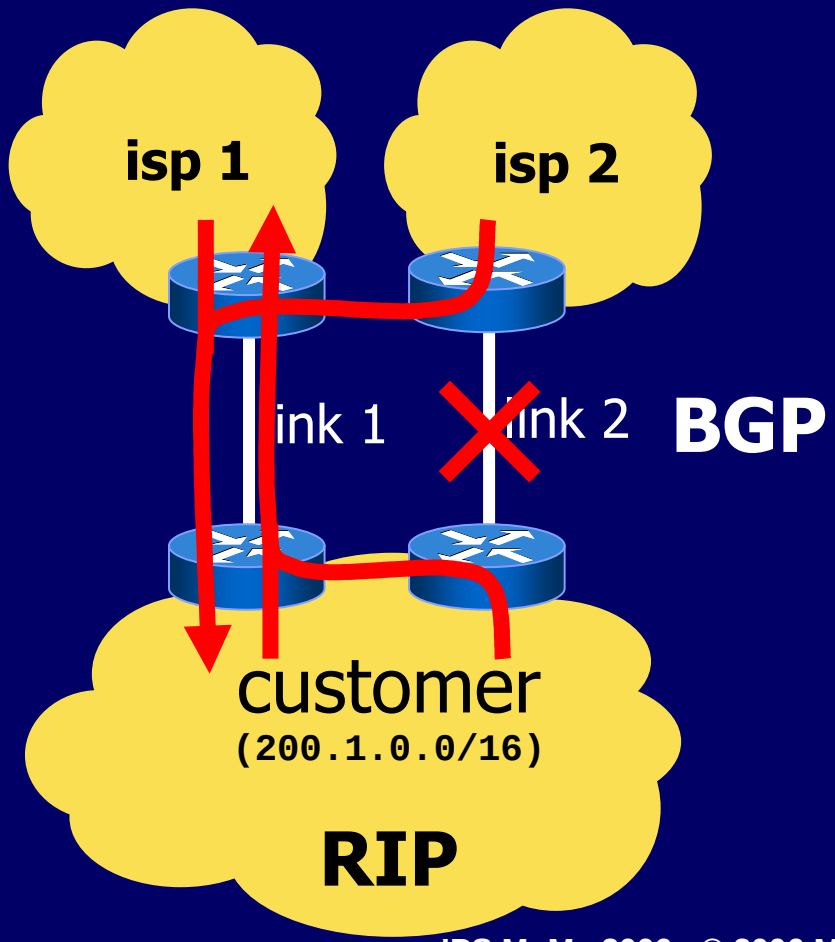
A sample lab

- A multihomed network, designed to:
  - Prohibit transit traffic



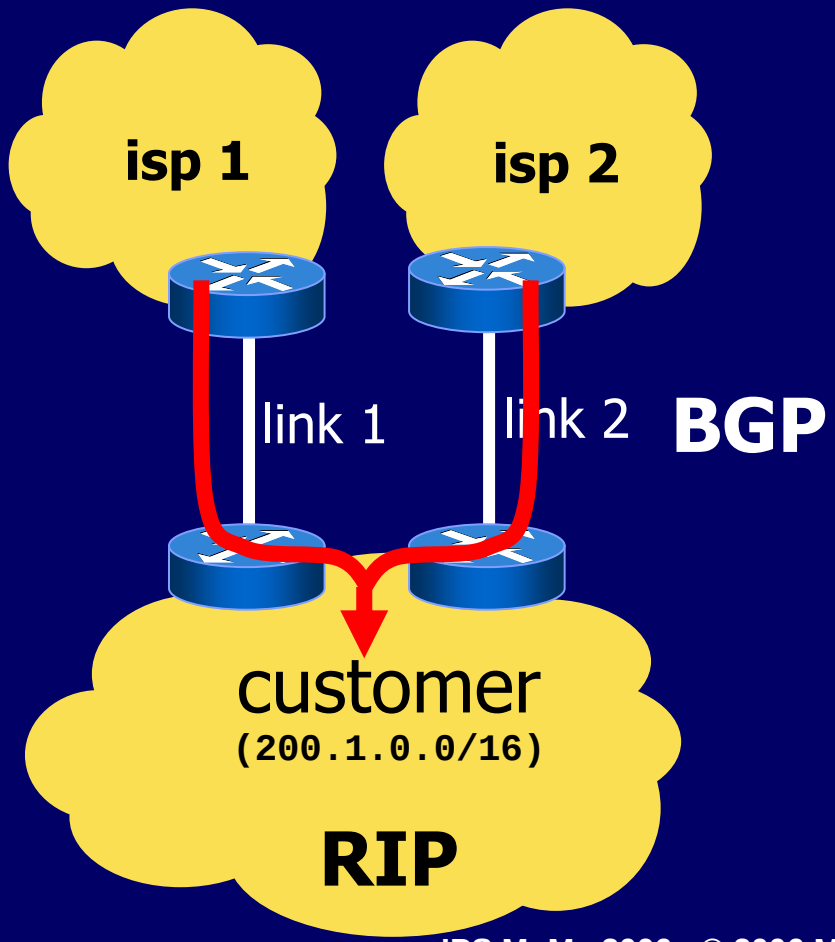
A sample lab

- A multihomed network, designed to:
  - Prohibit transit traffic
  - Be fault tolerant



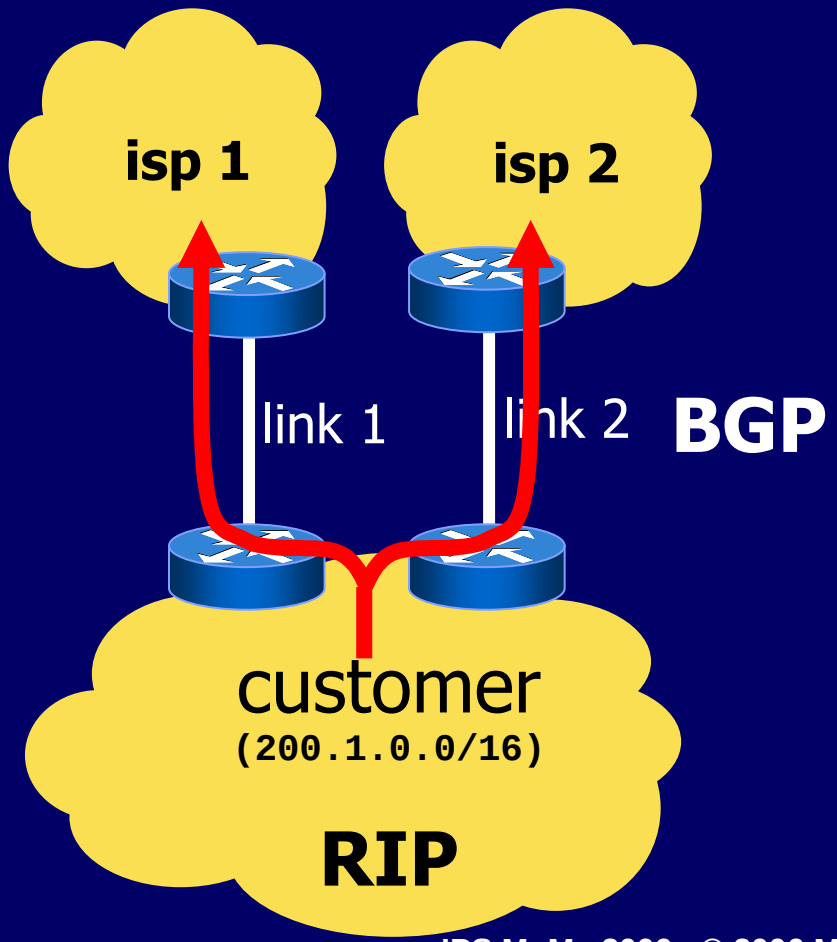
A sample lab

- A multihomed network, designed to:
  - Prohibit transit traffic
  - Be fault tolerant
  - Perform loadsharing
    - inbound: by announcing /17s



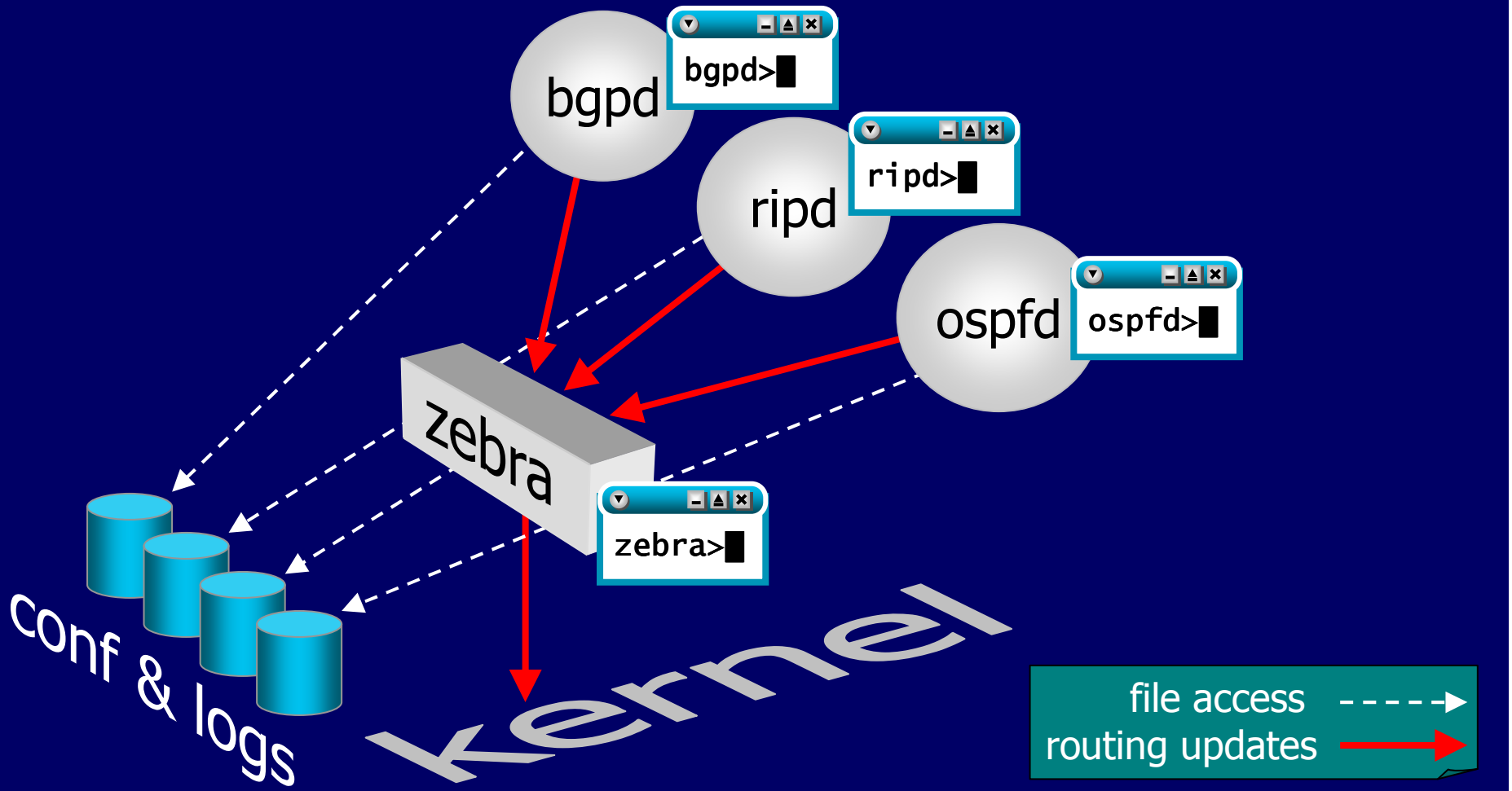
A sample lab

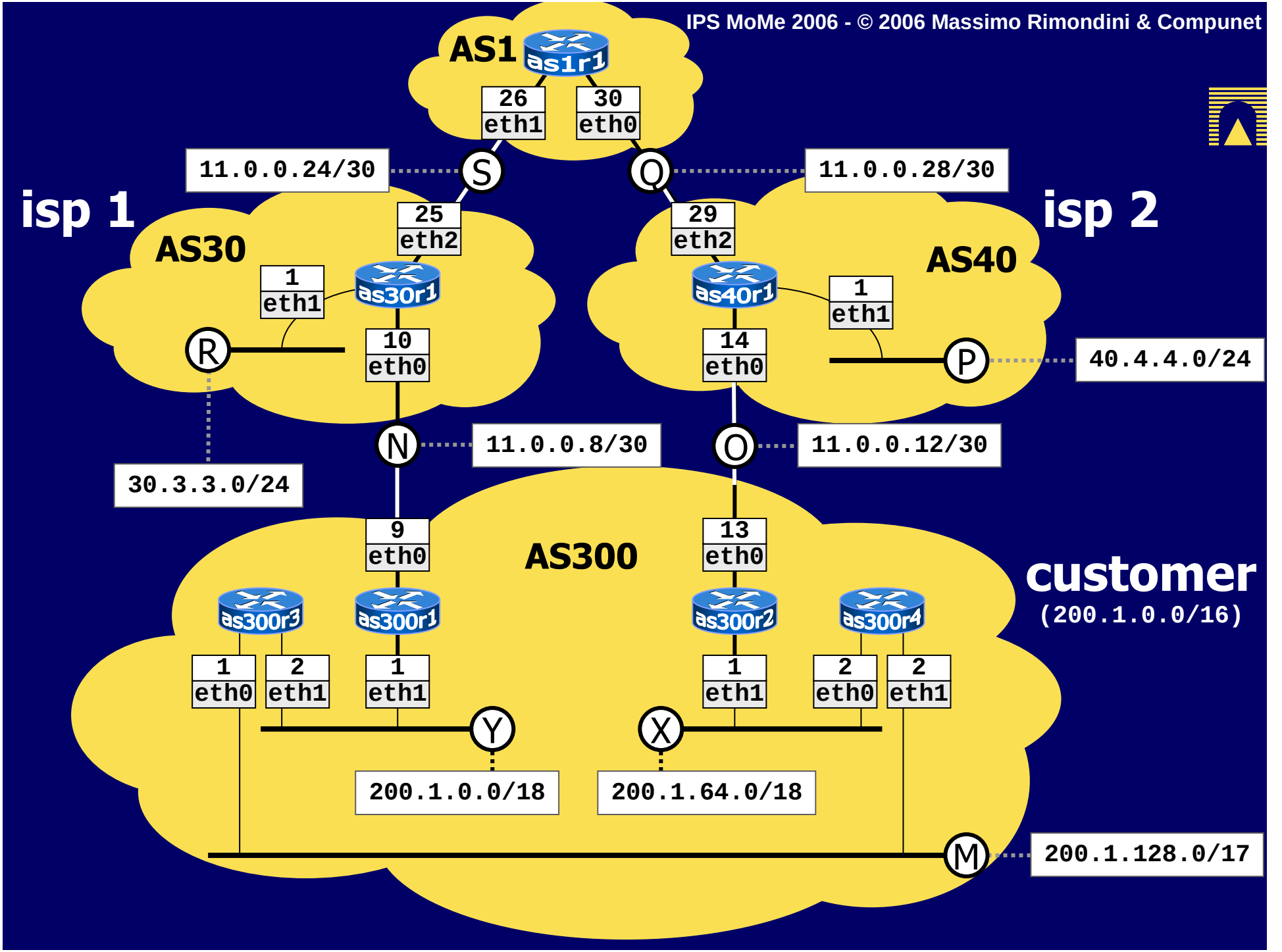
- A multihomed network, designed to:
  - Prohibit transit traffic
  - Be fault tolerant
  - Perform loadsharing
    - inbound: by announcing /17s
    - outbound: by nearest exit

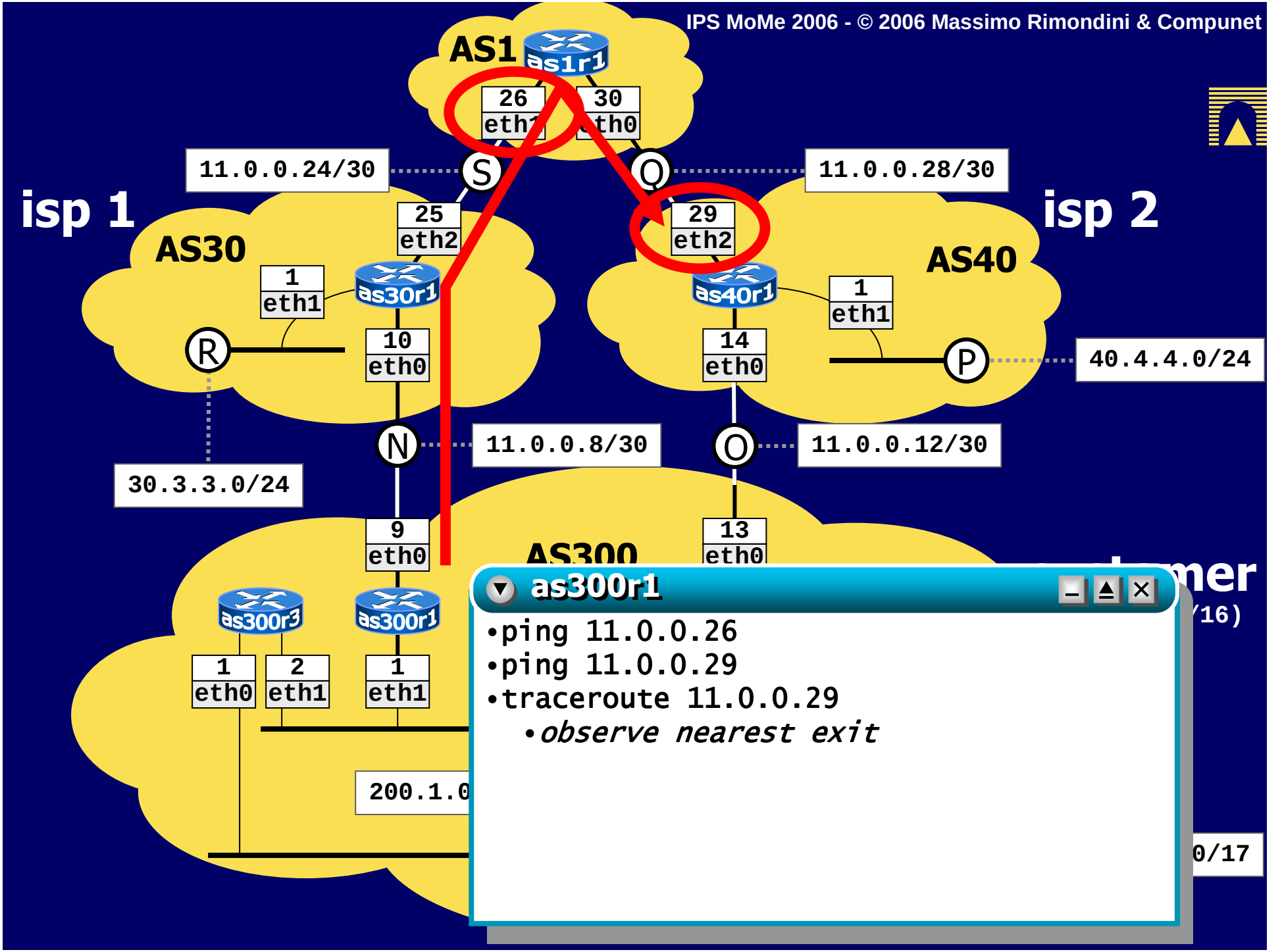


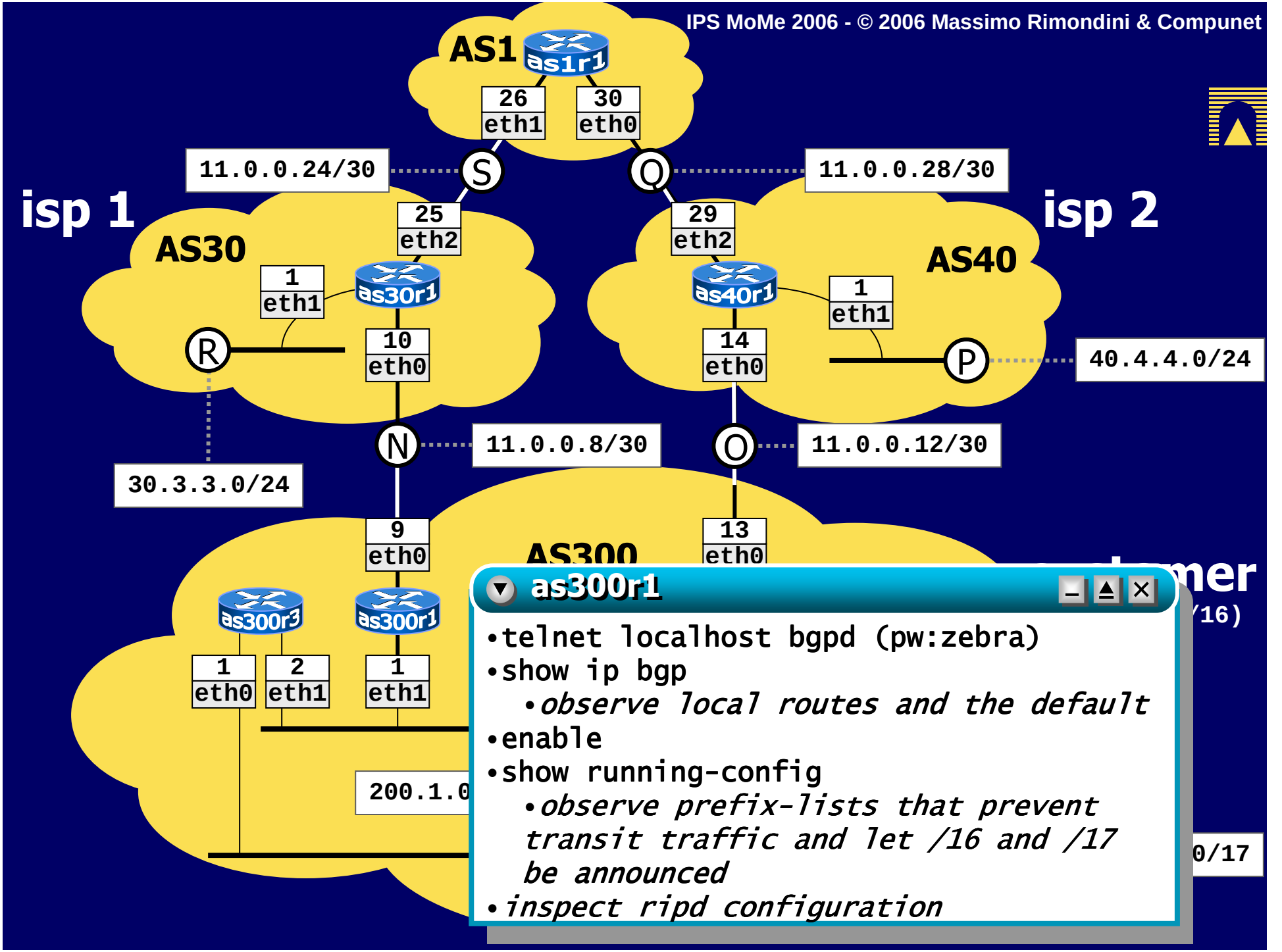
Making routers route traffic

- Routing is performed by the *zebra* software



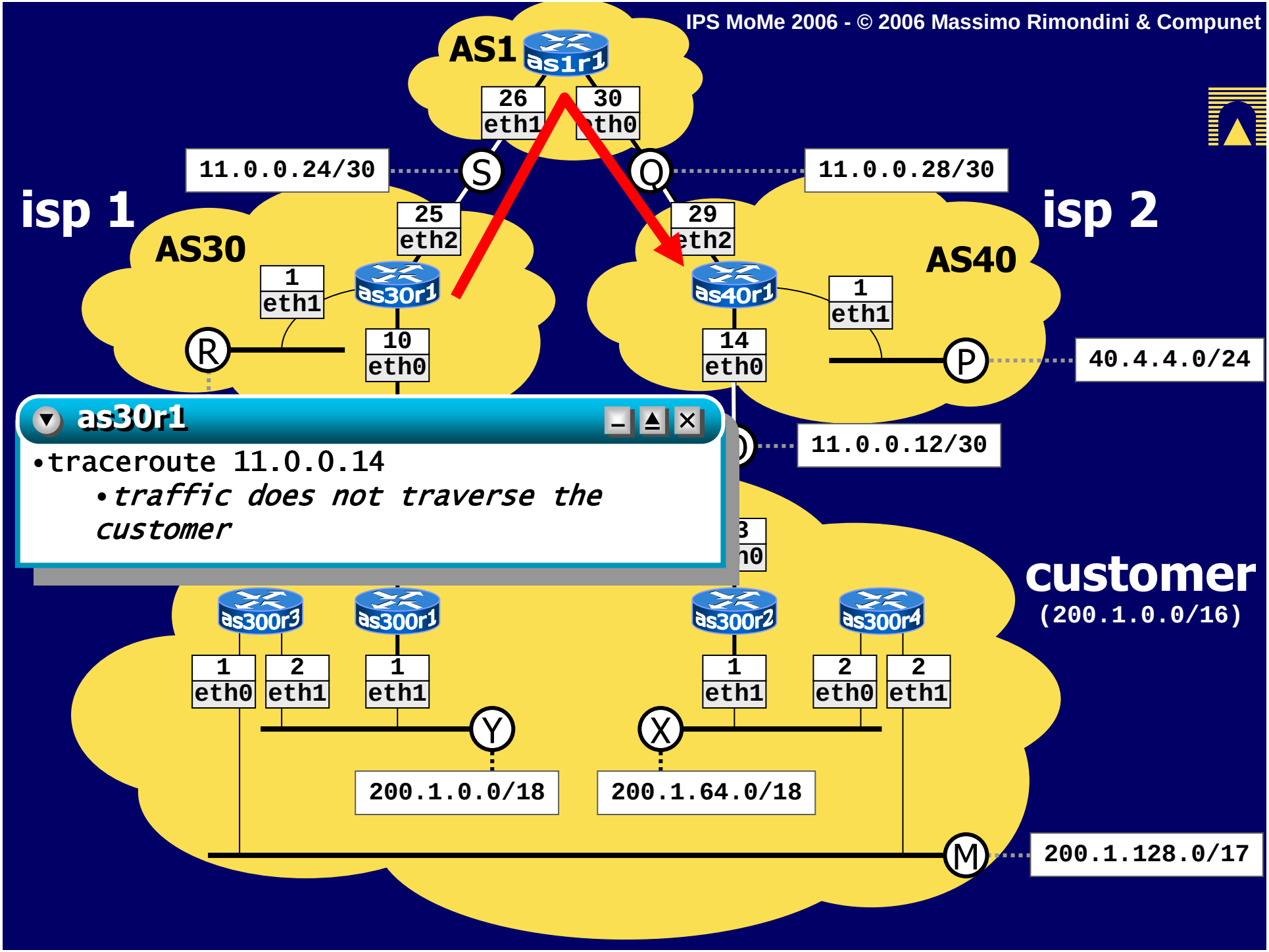


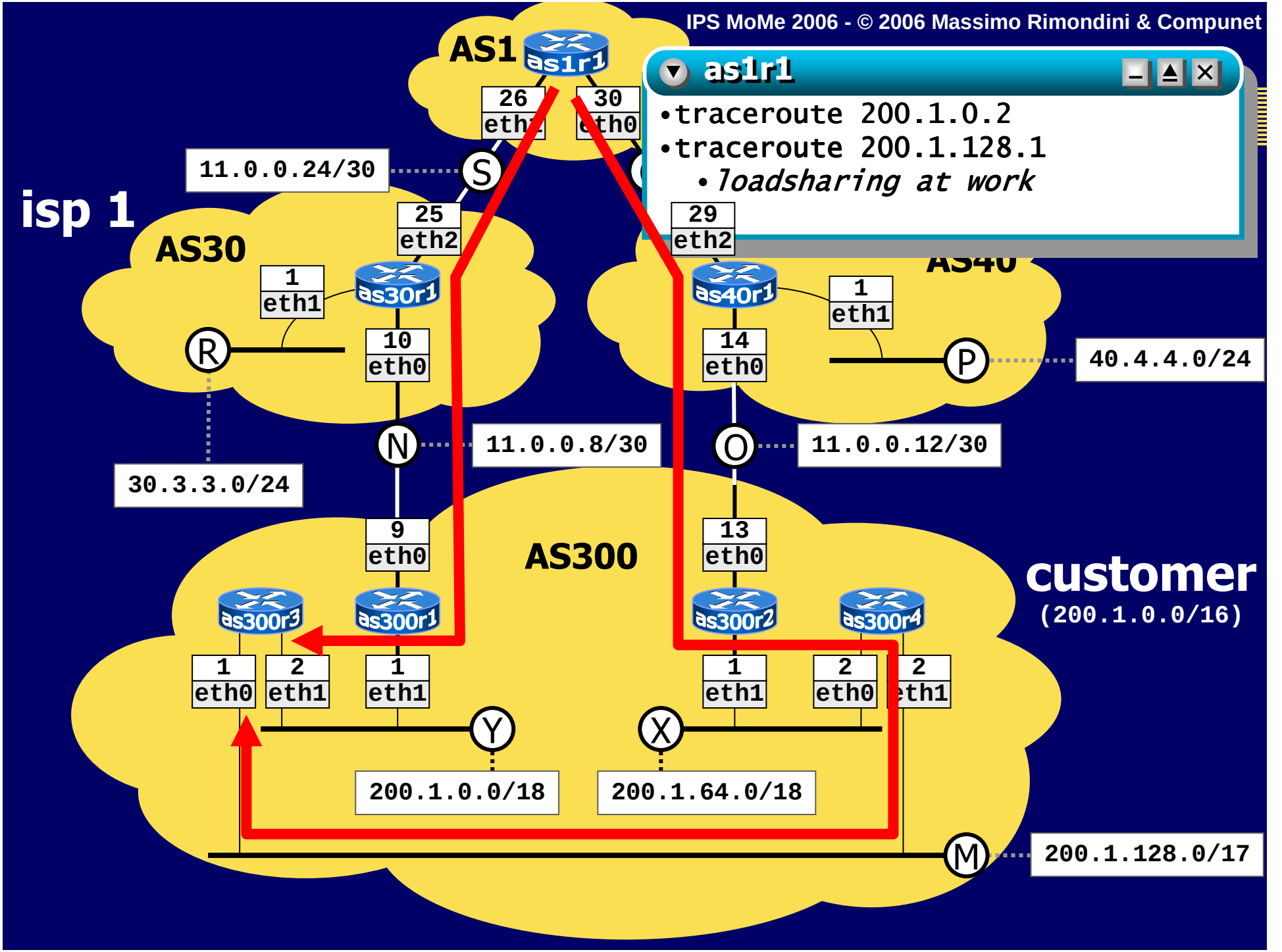

AS1
as1r1
26
eth1
30
eth0
11.0.0.24/30
S
Q
11.0.0.28/30
isp 1
isp 2
AS30
AS40
25
eth2
29
eth2
1
eth1
as30r1
as40r1
1
eth1
R
10
eth0
14
eth0
P
40.4.4.0/24
N
11.0.0.8/30
O
11.0.0.12/30
30.3.3.0/24
9
eth0
13
eth0
AS300
as300r3
as300r1
1
eth0
2
eth1
1
eth1
200.1.0
as300r1
•ping 11.0.0.26
•ping 11.0.0.29
•traceroute 11.0.0.29
•observe nearest exit

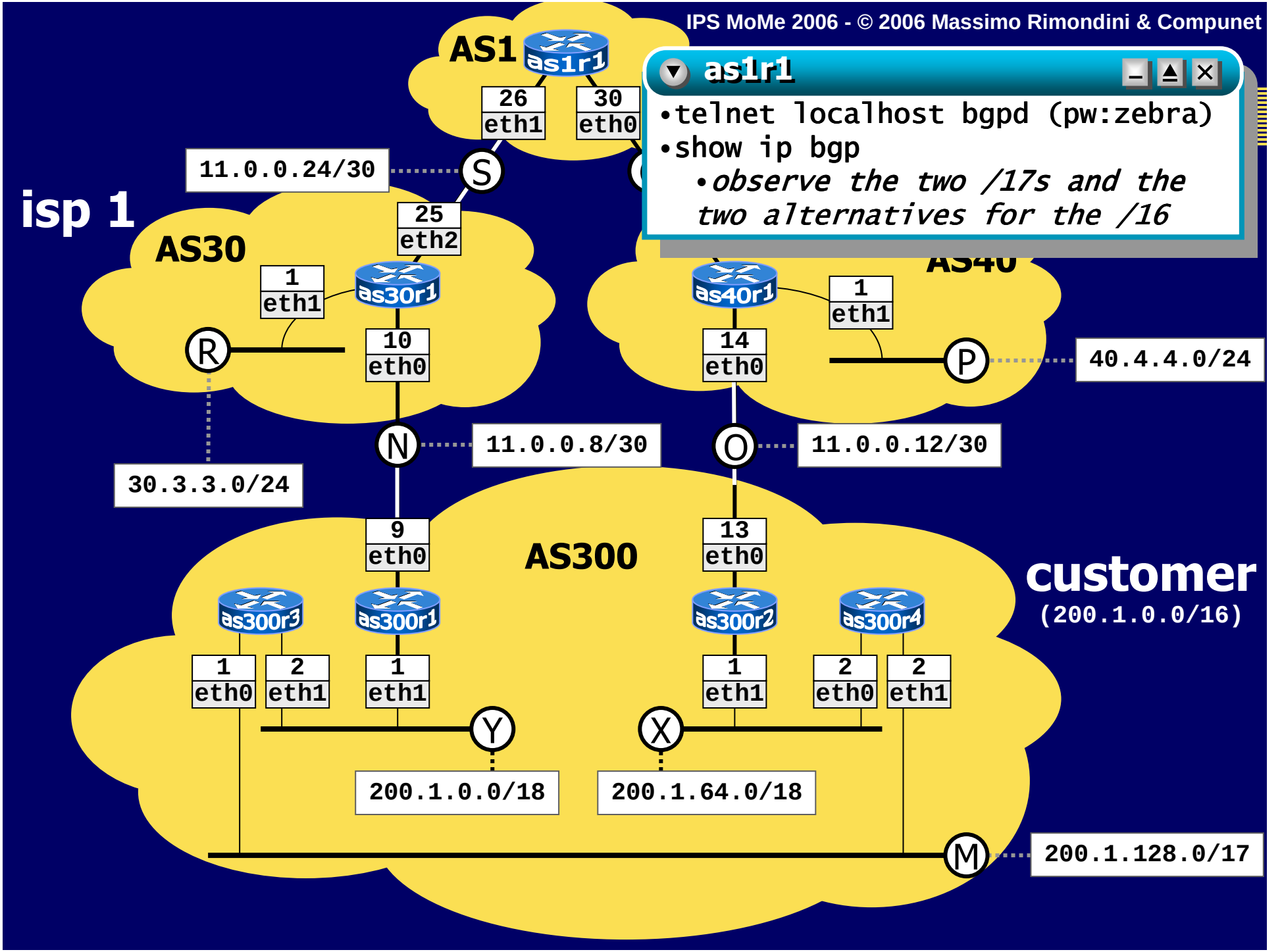


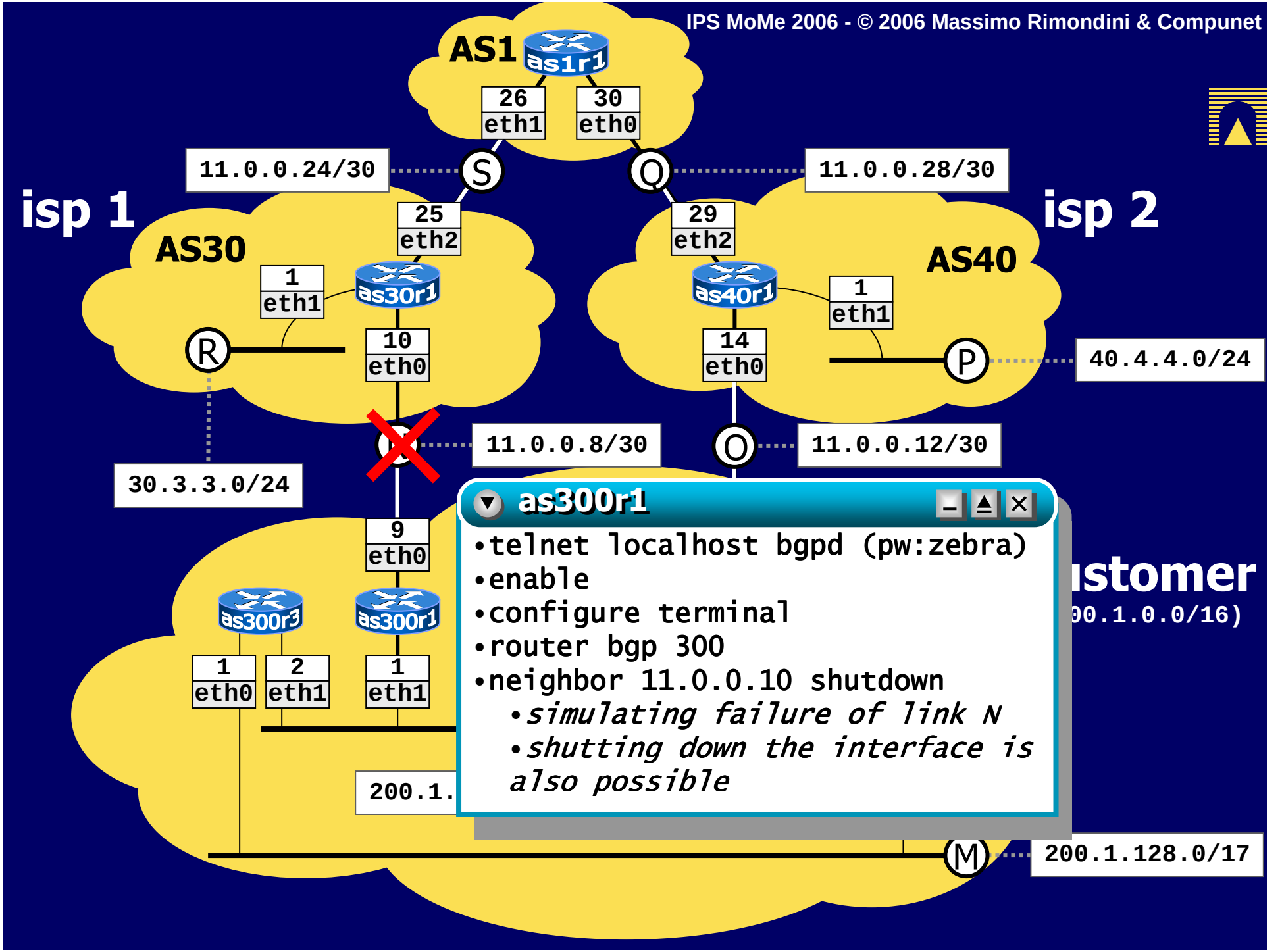
as300r1

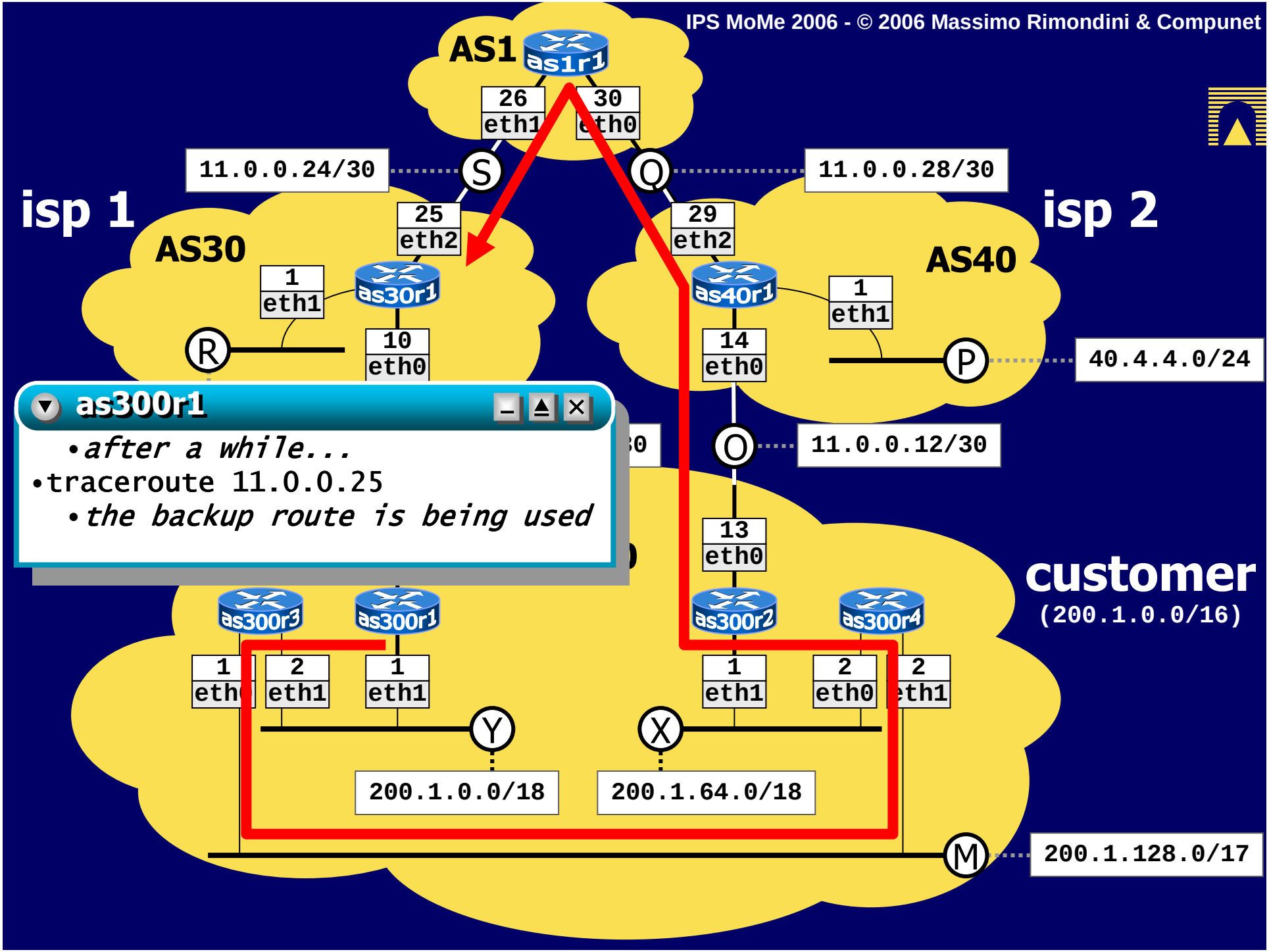
- telnet localhost bgpd (pw:zebra)
- show ip bgp
  - *observe local routes and the default*
- enable
- show running-config
  - *observe prefix-lists that prevent transit traffic and let /16 and /17 be announced*
- *inspect ripd configuration*


AS1
as1r1
26
eth1
30
eth0
11.0.0.24/30
S
Q
11.0.0.28/30
isp 1
isp 2
AS30
AS40
25
eth2
29
eth2
1
eth1
as30r1
as40r1
1
eth1
R
10
eth0
14
eth0
P
40.4.4.0/24
11.0.0.12/30
as30r1
•traceroute 11.0.0.14
•traffic does not traverse the customer
customer
(200.1.0.0/16)
as300r3
as300r1
as300r2
as300r4
1
eth0
2
eth1
1
eth1
1
eth1
2
eth0
2
eth1
Y
X
200.1.0.0/18
200.1.64.0/18
M
200.1.128.0/17


IPS MoMe 2006 - © 2006 Massimo Rimondini & Compunet
AS1
as1r1
26
eth1
30
eth0
as1r1
•traceroute 200.1.0.2
•traceroute 200.1.128.1
•loadsharing at work
11.0.0.24/30
S
isp 1
AS30
25
eth2
29
eth2
AS40
1
eth1
as30r1
as40r1
1
eth1
R
10
eth0
14
eth0
P
40.4.4.0/24
N
11.0.0.8/30
O
11.0.0.12/30
30.3.3.0/24
9
eth0
13
eth0
AS300
customer
(200.1.0.0/16)
as300r3
as300r1
as300r2
as300r4
1
eth0
2
eth1
1
eth1
1
eth1
2
eth0
2
eth1
Y
X
200.1.0.0/18
200.1.64.0/18
M
200.1.128.0/17

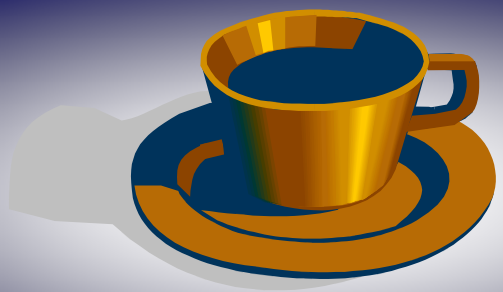

IPS MoMe 2006 - © 2006 Massimo Rimondini & Compunet
AS1
as1r1
26
eth1
30
eth0
as1r1
•telnet localhost bgpd (pw:zebra)
•show ip bgp
•observe the two /17s and the two alternatives for the /16
11.0.0.24/30
S
isp 1
AS30
25
eth2
1
eth1
as30r1
R
10
eth0
30.3.3.0/24
N
11.0.0.8/30
as40r1
1
eth1
14
eth0
P
40.4.4.0/24
O
11.0.0.12/30
9
eth0
AS300
13
eth0
customer
(200.1.0.0/16)
as300r3
as300r1
as300r2
as300r4
1
eth0
2
eth1
1
eth1
1
eth1
2
eth0
2
eth1
Y
X
200.1.0.0/18
200.1.64.0/18
M
200.1.128.0/17


AS1
as1r1
26
eth1
30
eth0
11.0.0.24/30
S
Q
11.0.0.28/30
isp 1
isp 2
AS30
AS40
25
eth2
29
eth2
1
eth1
as30r1
as40r1
1
eth1
R
10
eth0
14
eth0
P
40.4.4.0/24
11.0.0.8/30
O
11.0.0.12/30
30.3.3.0/24
9
eth0
as300r1
•telnet localhost bgpd (pw:zebra)
•enable
•configure terminal
•router bgp 300
•neighbor 11.0.0.10 shutdown
•simulating failure of link N
•shutting down the interface is also possible
as300r3
as300r1
1
eth0
2
eth1
1
eth1
200.1.
M
200.1.128.0/17
stomer
00.1.0.0/16)



Wanna play?

- Visit http://www.netkit.org
  - Other ready-to-use labs
  - Lecture slides
  - Netkit updates
  - Other resources (NetML)
- Forthcoming lab topology:



Thank you!