Dynamic routing: OSPF

Network Infrastructures labs

Marco Spaziani Brunella

SAPIENZA
UNIVERSITÀ DI ROMA

Lecture details

- **Readings:**
  - RFC 2328
- **Lecture outline:**
  - Dynamic routing
  - OSPF
  - Quagga & Zebra

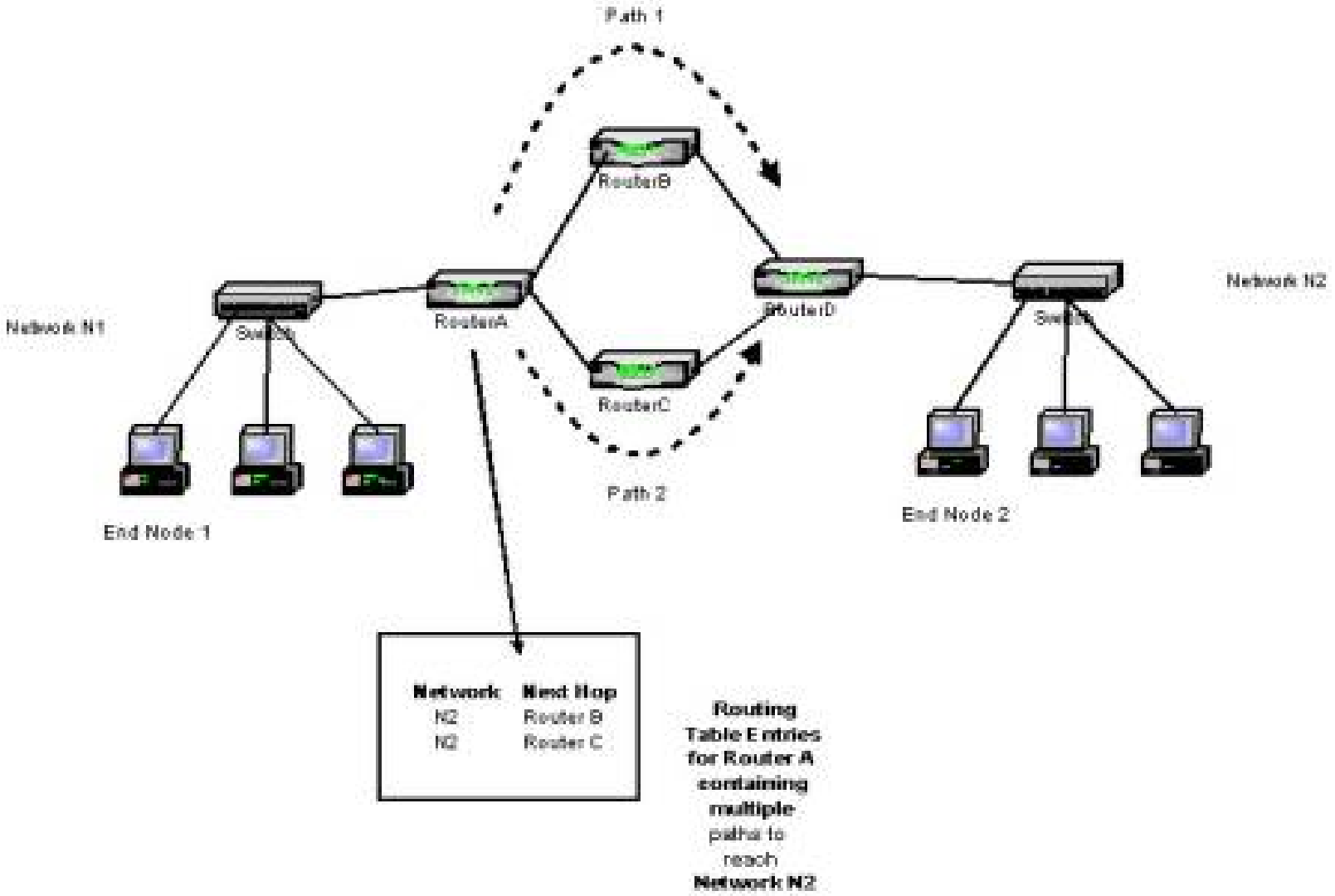
Where does static routing fails?

Most of the problems of static routing can be reconducted to:

- Scalability issues
- Reliability issues
- Heterogeneity issues

Scalability because as network increases in complexity, configuring every single router for every single path can be a tough task, in particular if you want to take into account the different path costs in a redundant network.
Reliability because if a link goes down, the network cannot *failover* a redundant link automatically.
Heterogeneity because the network can be composed of different router made by different vendors with different OSs, meaning that the sysadmin must be able to configure them all.

Redundant network example



Dynamic (adaptive) routing protocols 101

"One protocol to **rule** *them all, One protocol to* **find** *them, One protocol to* **bring** *them all, and in the network* **bind** *them"*

Rule: Once we have a routing protocol, we can abstract from how it is implemented and develop platform independent architectures, solving heterogeneity issues.
Find: A good routing protocol must discover failures and react properly, solving reliability issues.
Bring: the routing protocol must provide control messages to other nodes using the same protocol to exchange informations (e.g. subnets advertisement), solving scalability issues.
Bind: Last but not least, the routing protocol must guarantee connectivity all over the network nodes with the minimum amount of overhead.

Dynamic routing protocols available

The most used routing protocols are:

- RIP
- OSPF*
- IS-IS
- IGRP

Why and what is Open Shortest Path First

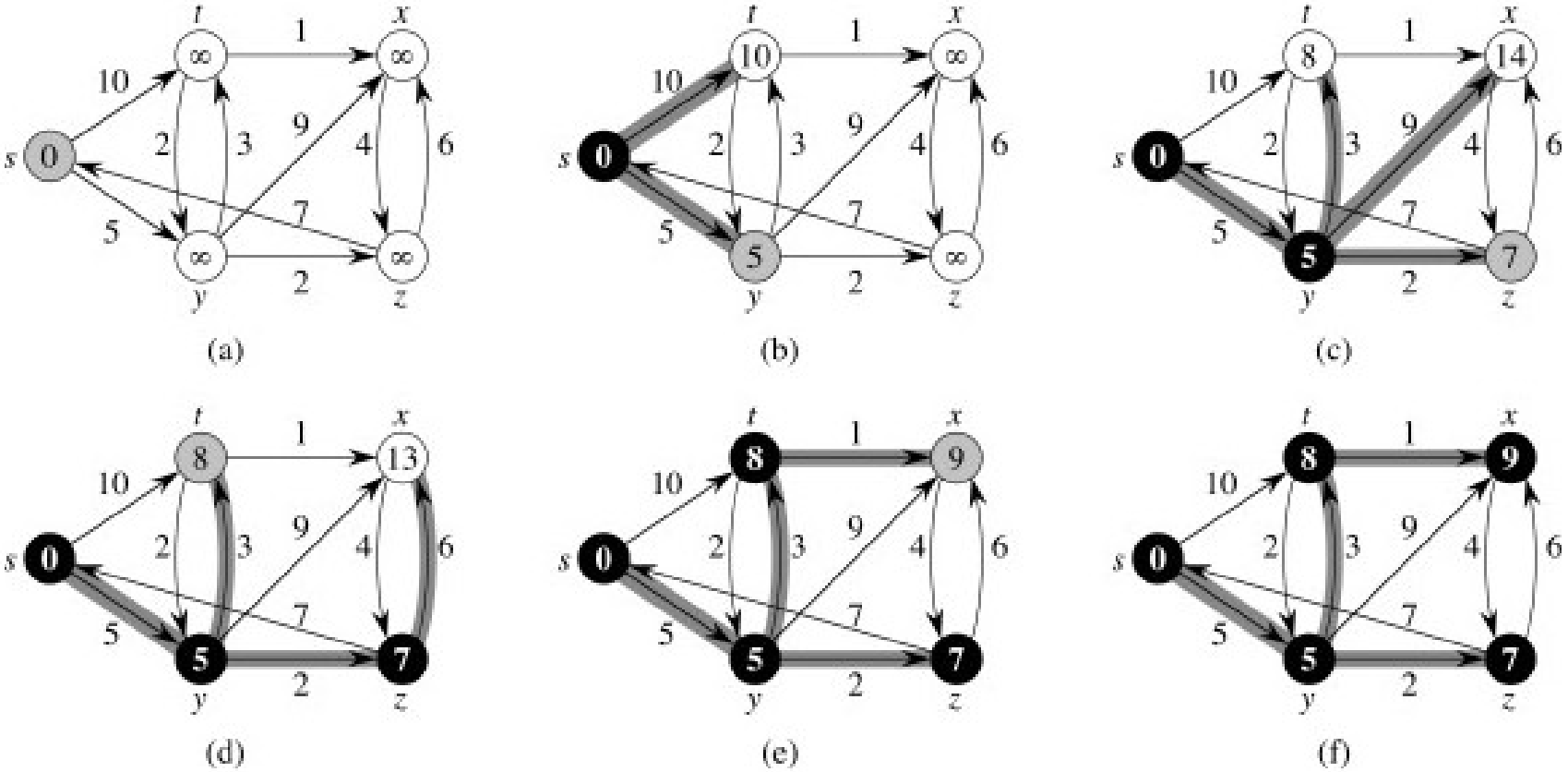
Is one of the most spread routing protocol and one of the fastest. It has a lot of parameters that can be configured, but it takes just few knowledge to understand it and to start to make it work properly.
OSPF falls in the category of the Interior Gateway Protocols, in particular in the sub-category of the Link State Protocols, meaning that every node in the network must know the entire network topology.

OSPF for dummies

Every router running OSPF advertise its presence to neighbors routers and "floods" the network with particular messages called Link State Packets which contains information about the link (in terms of metrics and state) to which the node is connected to.

Every router constructs the network topology (namely a graph) from the LSP received and, with the Dijkstra algorithm, builds the "best path" for every single host. This will be the router's routing table.

Dijkstra algorithm



OSPF in Linux

In Linux, OSPF is included in a broader suite called *quagga*. The daemon wich implements OSPF is called *zebra*. In order to have OSP up and running, we must first configure the zebra daemon and then the OSPVv2 daemon. Example configurations can be:

```
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
isisd=no
ldpd=no
~
```

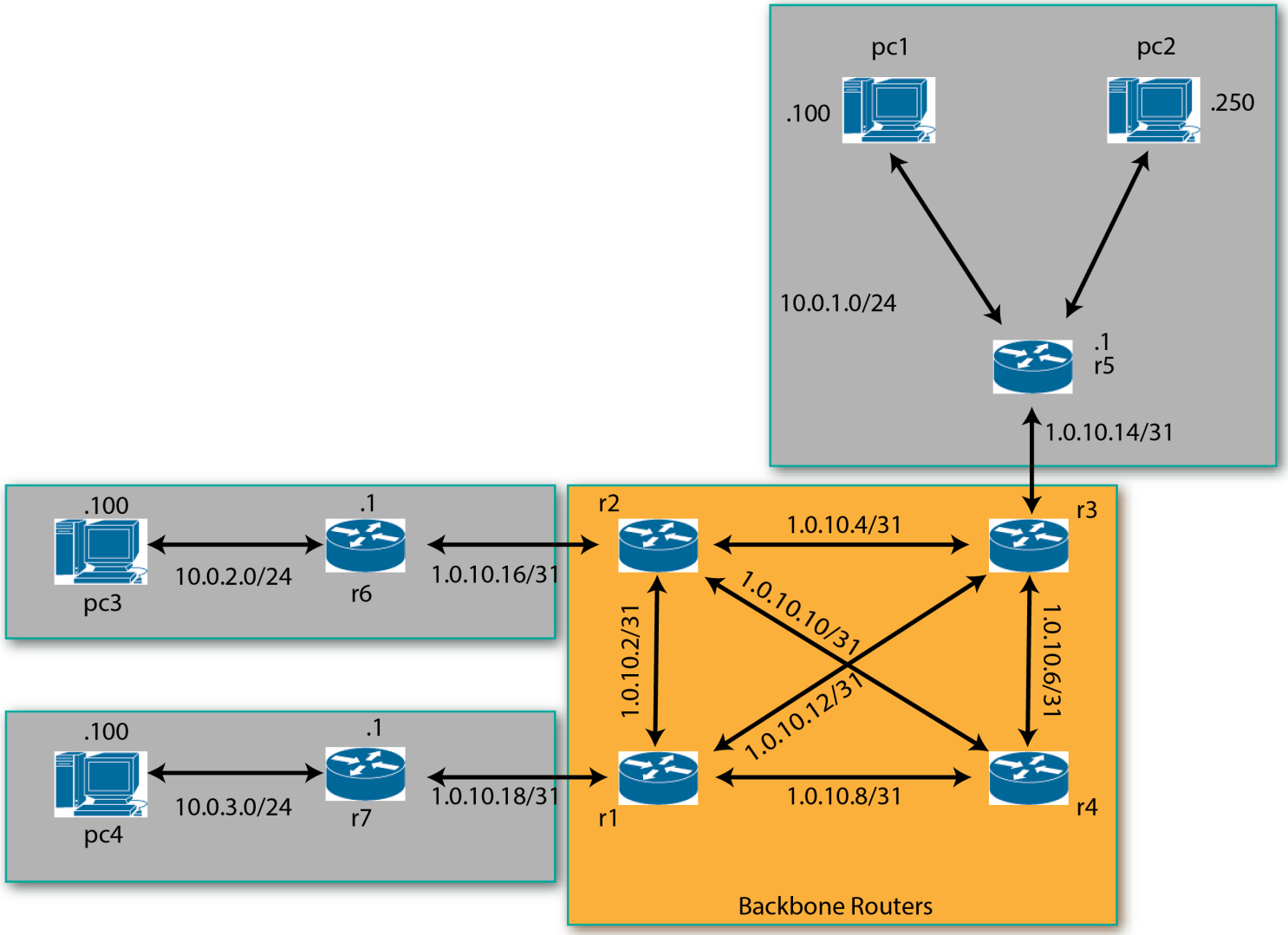
/etc/quagga/zebra.conf

```
! -*- zebra -*-
!
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.1.1.1 2002/12/13 20:15:30 paul Exp $
!
hostname r1
password zebra
enable password zebra

interface eth0
ip address 1.0.10.2/31
link-detect

interface eth1
ip address 1.0.10.9/31
link-detect

interface eth2
ip address 1.0.10.13/31
link-detect

interface eth3
ip address 1.0.10.19/31
link-detect
```

/etc/quagga/ospfd.conf

```
hostname r1
password zebra

interface eth0
ospf hello-interval 2

interface eth1
ospf hello-interval 2

interface eth2
ospf hello-interval 2

interface eth3
ospf hello-interval 2

router ospf
network 1.0.10.18/31 area 0.0.0.0
network 1.0.10.2/31 area 0.0.0.0
network 1.0.10.8/31 area 0.0.0.0
network 1.0.10.12/31 area 0.0.0.0
```

lab_ospf


pc1
pc2
.100
.250
10.0.1.0/24
.1
r5
1.0.10.14/31
.100
.1
pc3
10.0.2.0/24
r6
1.0.10.16/31
r2
r3
1.0.10.4/31
1.0.10.10/31
1.0.10.2/31
1.0.10.6/31
1.0.10.12/31
.100
.1
pc4
10.0.3.0/24
r7
1.0.10.18/31
r1
1.0.10.8/31
r4
Backbone Routers

Link cost

OSPF daemon allows us to specify cost for each physical link on a router. The cost is a number greater than zero that has no physical meaning, it's just relative to the other links cost: a path with a lower cost respect to another will win the race and become part of a router's routing table.

```
hostname r2
password zebra

interface eth0
ospf hello-interval 2
ospf cost 10

interface eth1
ospf hello-interval 2
ospf cost 10

interface eth2
ospf hello-interval 2
ospf cost 10

interface eth3
ospf hello-interval 2
ospf cost 10

router ospf
network 1.0.10.2/31 area 0.0.0.0
network 1.0.10.4/31 area 0.0.0.0
network 1.0.10.10/31 area 0.0.0.0
```

Area

Areas are used to logically divide different routing domains. The routers belonging to different areas do not know the entire network topology of the other area, but use the Area Border Router as gateway. They are identified by octets, just like ip addresses. The area 0.0.0.0 is called the backbone area. We can identify (for our purposes) 3 kind of routers:

- Internal Router; all interfaces belong to the same area
- Area Border Router; connects one or more area to the backbone
- Backbone router; has at least one interface on the backbone

In lab_ospf we have:
Int. Routers: r4, r5, r6, r7
ABR: r1, r2, r3
Backbone Routers: r1, r2, r3, r4

Exercise

Try to use areas on lab_ospf, assigning the following areas:
0.0.0.0 to the backbone routers (already done)
1.1.1.1 to the subnet between r7-r1
2.2.2.2 to the subnet between r6-r2
3.3.3.3 to the subnet between r5-r3