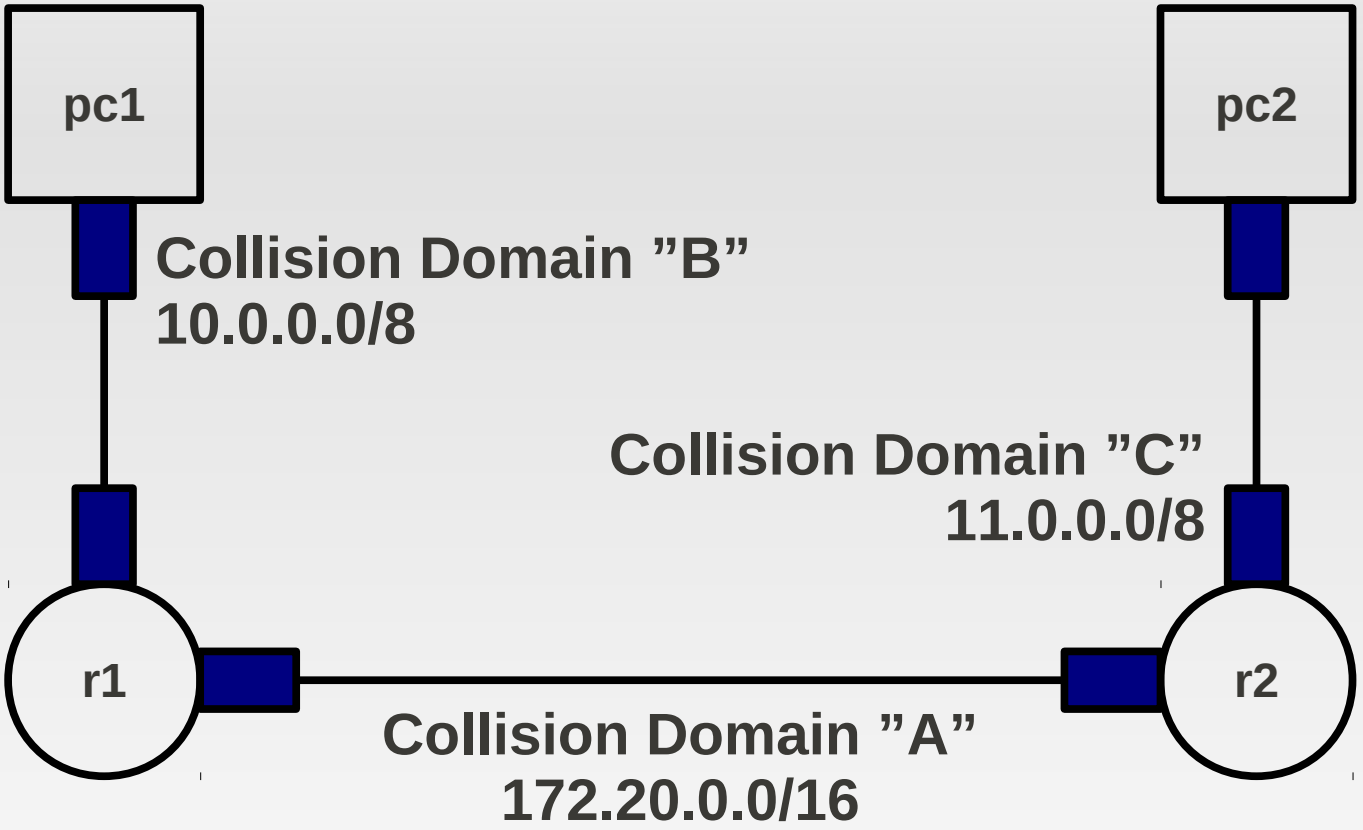
Static Routing

- Linux provides two different tools to configure statically the routes that define how IP packets are forwarded in the network
  - route
  - iproute2
- The command route permits to manipulate the main routing table used by the forwarding mechanism of the kernel to select the next hop

Single Path Routing



Single Path Routing

- Lab configuration (file lab.conf)

  r1[0]=”A” → eth0 of r1 on collision domain A

  r1[1]=”B” → eth1 of r2 on collision domain B

  r2[0]=”A” → eth0 of r2 on collision domain A

  r2[1]=”C” → eth1 of r2 on collision domain C

  pc1[0] = ”B” → eth0 of pc1 on collision domain B

  pc2[0] = ”C” → eth0 of pc2 on collision domain C

-

Single Path Routing

- To speed up the configuration of the lab, we define also the startup files (commands to be performed soon after the boot)
- pc1.startup

  ifconfig eth0 10.0.0.101 netmask 255.0.0.0 up

  route add default gw 10.0.1.2 dev eth0
- pc2.startup

  ifconfig eth0 11.0.0.102 netmask 255.0.0.0 up

  route add default gw 11.0.2.2 dev eth0

Single Path Routing

- r1.startup

  ifconfig eth0 172.20.1.1 netmask 255.255.0.0 up

  ifconfig eth1 10.0.1.2 netmask 255.0.0.0 up

  route add -net 11.0.0.0/8 gw 172.20.2.1 dev eth0

- r2.startup

  ifconfig eth0 172.20.2.1 netmask 255.255.0.0 up

  ifconfig eth1 10.0.2.2 netmask 255.0.0.0 up

  route add -net 10.0.0.0/8 gw 172.20.1.1 dev eth0

Single Path Routing

- By default, 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!)
- The **default** route provides the next hop router used to forward the packets, when **no other** entry matches the packet IP destination (either host or network)

Single Path Routing

- Test the connectivity using ICMP messages
- From pc1 console, perform a ping to pc2

  ping 11.0.0.102
- From r1 and r2 console capture the traffic

  tcpdump -i eth0 -n -e
- Change the interface used to overhear the traffic (from eth0 and eth1) for looking at the arp message exchange
  - The arp command permit to modify the arp cache

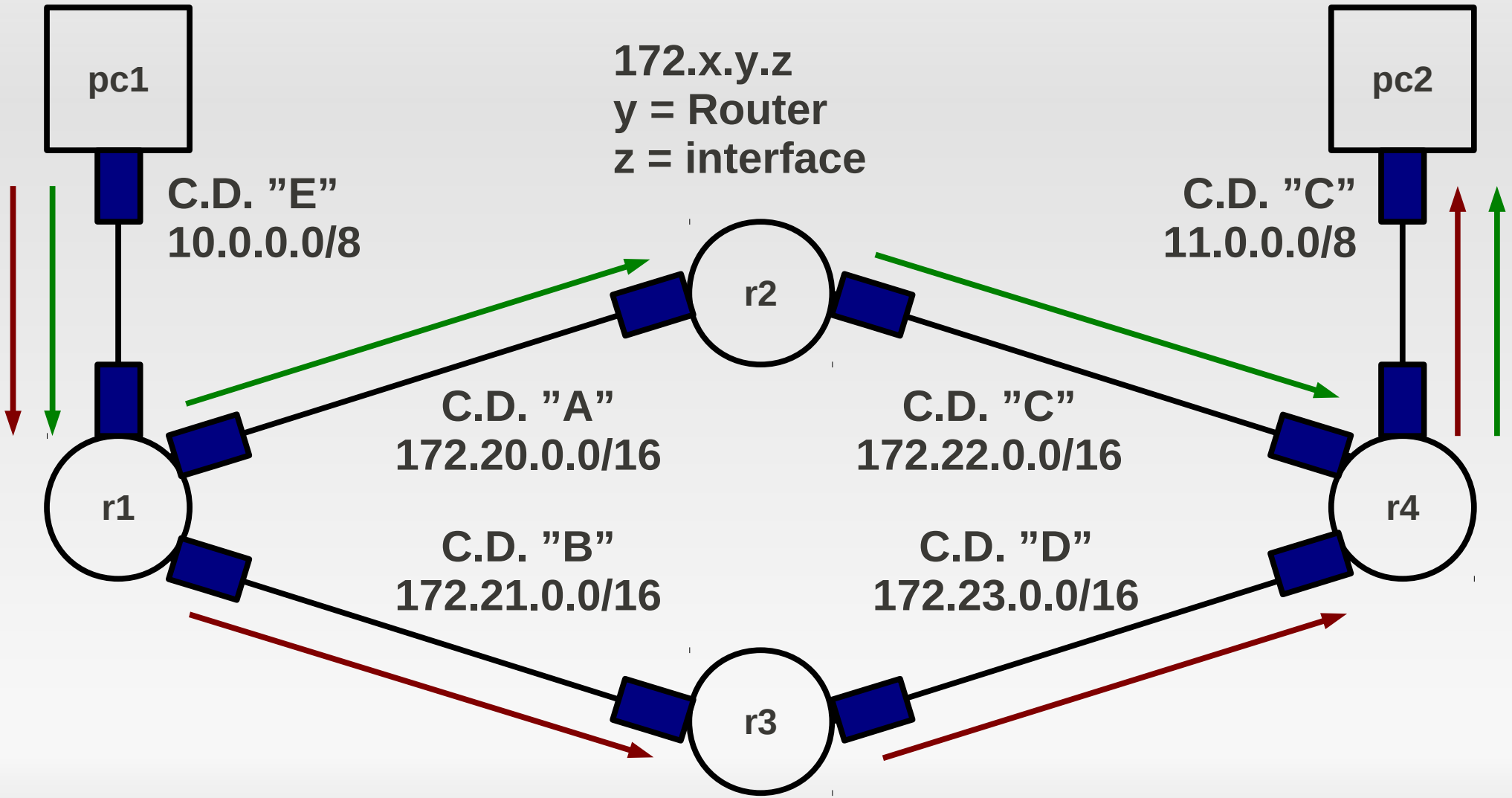
Multi-Paths Routing

- When more paths are available to reach the same destination network, it is better off configuring the router to exploit all network paths
  - Higher availability/reliability
  - Higher Performance (paths bonding)
  - Load balancing
- One of the most widespread tools used to configure multi-path routing is iproute2

Brief Introduction to iproute2

- Nowadays, the iproute2 has replaced tools like ifconfig, route, arp
- iproute2 is more powerful and flexible than its predecessors
- The routes can be defined considering
  - MAC address
  - Source IP address
  - Port, Type of Service
- We can configure the router to exploit several paths towards a destination (multi-path routing)

Multi Paths Routing



ubuntu

Multi-Paths Routing

- Lab configuration (file lab.conf)

  r1[0]=”A”

  r1[1]=”B”

  r1[2]=”E”

  r2[0]=”A”

  r2[1]=”C”

  r3[0]=”C”

  r3[1]=”D”

Multi-Paths Routing

- Lab configuration (file lab.conf) … con't

  r4[0]=”C”

  r4[1]=”D”

  r4[2]=”F”

  pc1[0]=”E”

  pc2[0]=”F”

Multi-Paths Routing

- Startup files containig the commands performed after the boot
- pc1.startup

  ifconfig eth0 10.0.0.101 netmask 255.0.0.0 up

  ifconfig eth0:1 10.0.0.100 netmask 255.0.0.0 up

  route add default gw 10.0.1.3 dev eth0
- pc2.startup

  ifconfig eth0 11.0.0.102 netmask 255.0.0.0 up

  ifconfig eth0:1 11.0.0.103 netmask 255.0.0.0 up

  route add default gw 11.0.4.3 dev eth0

Multi-Paths Routing

- r2.startup

  ifconfig eth0 172.20.2.1 netmask 255.255.0.0 up

  ifconfig eth1 172.22.2.2 netmask 255.255.0.0 up

  route add -net 10.0.0.0/8 gw 172.20.1.1

  route add -net 11.0.0.0/8 gw 172.22.4.1

- r3.startup

  ifconfig eth0 172.21.3.1 netmask 255.255.0.0 up

  ifconfig eth1 172.23.3.2 netmask 255.255.0.0 up

  route add -net 10.0.0.0/8 gw 172.21.1.2

  route add -net 11.0.0.0/8 gw 172.23.4.2

Multi-Paths Routing

- r1.startup

```
ifconfig eth0 172.20.1.1 netmask 255.255.0.0 up
ifconfig eth1 172.21.1.2 netmask 255.255.0.0 up
ifconfig eth2 10.0.1.3 netmask 255.0.0.0 up
sysctl -w net.ipv4.conf.all.rp_filter=0
sysctl -w net.ipv4.conf.all.log_martians=1
ip route add table main default nexthop via 172.20.2.1 \\
                    weight 1 nexthop via 172.21.3.1 weight 1
```

Multi-Paths Routing

- r4.startup

```
ifconfig eth0 172.22.4.1 netmask 255.255.0.0 up
ifconfig eth1 172.23.4.2 netmask 255.255.0.0 up
ifconfig eth2 11.0.4.3 netmask 255.0.0.0 up
sysctl -w net.ipv4.conf.all.rp_filter=0
sysctl -w net.ipv4.conf.all.log_martians=1
ip route add table main default nexthop via 172.22.2.2 \\
                    weight 1 nexthop via 172.23.3.2 weight 1
```

Linux Forwarding Mechanism

- Given a multipath route, data traffic matching the route entry could be distributed between the next hops
  - Per flow: the next hop is selected for each unique combination of source and destination IP addresses.
  - Per connection: the next hop is selected every time a new connection is started.
  - Per packet: the next hop is selected for each packet.
- When there is no support for Multipath caching, Linux always distributes traffic between the different next hops of a Multipath route on a per-flow basis proportionally to the weights of the flows

Linux Forwarding Mechanism

- When multipath caching is enabled, traffic is distributed differently depending on ”the place” where it originates:
  - Locally generated traffic: traffic is distributed on a per-connection basis.
  - Ingress traffic that needs to be forwarded: traffic is distributed as if there was no support for multipath caching: the first matching cached route is always used.
- Define an alias for the interfaces of both PCs to force the selection of the two available paths

Interface Alias

- Use the command ifconfig to define an alias for a network interface

  ifconfig eth0:1 10.0.0.102 up

- Virtual interface:
  - eth0:1 is the alias of the virtual interface
  - 10.0.0.102 is the IP address assigned to this interface