kathara lab

bgp: transit as

Version	1.7.1 (compact)
Author(s)	Lorenzo Ariemma, Luca Cittadini, Giuseppe Di Battista, Valerio Gregori, Alessandro Oddi, Massimo Rimondini
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	possible architectures for a transit provider, bad interactions between igp and bgp routing protocols, configuration of tunnels – kathara version of a netkit lab

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.

scenario

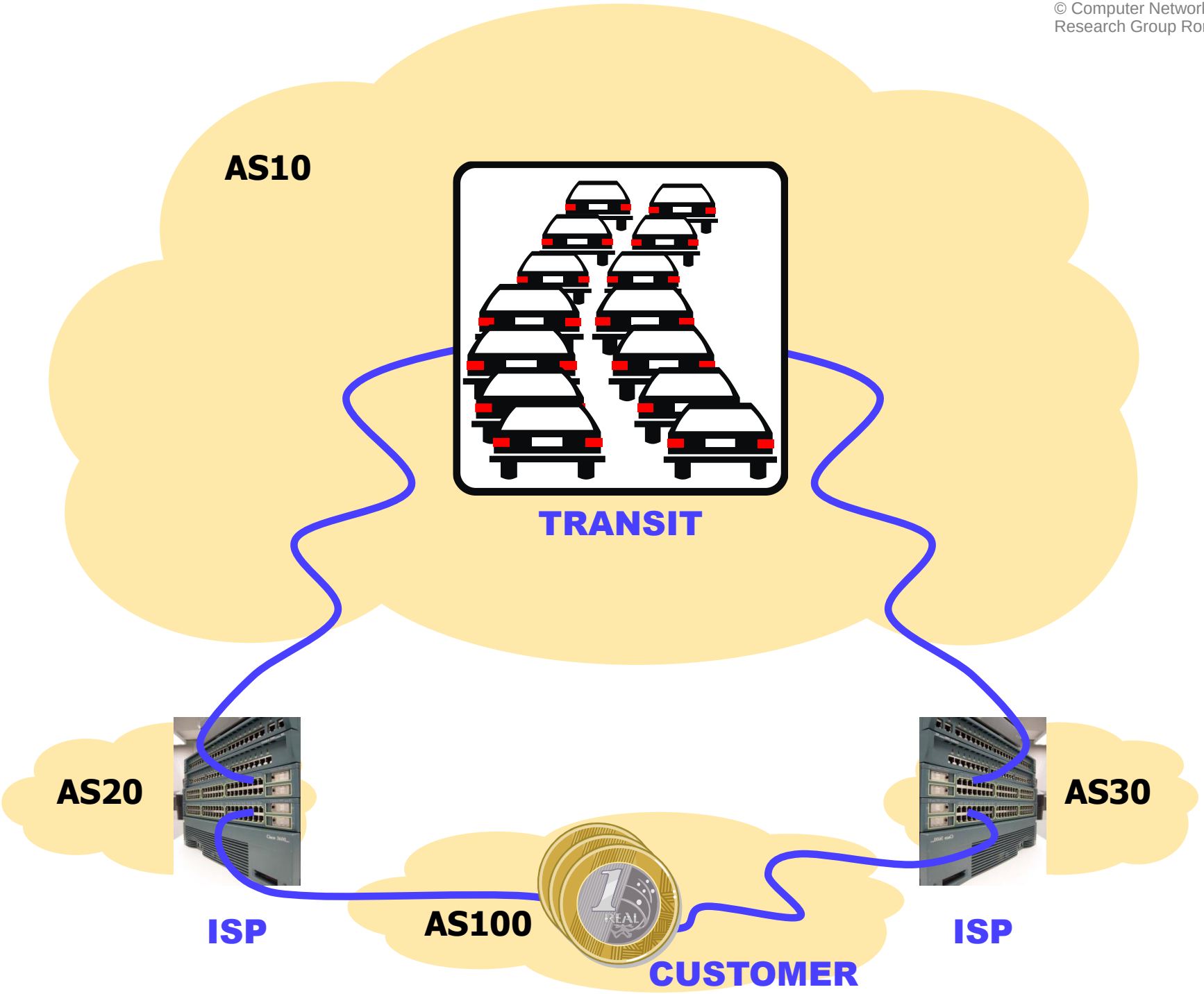
- a transit as
  - receives and propagates the full bgp routing table from/to its neighbors (customers, peers, providers)
  - receives and forwards traffic across its neighbors

transit as: degrees of freedom

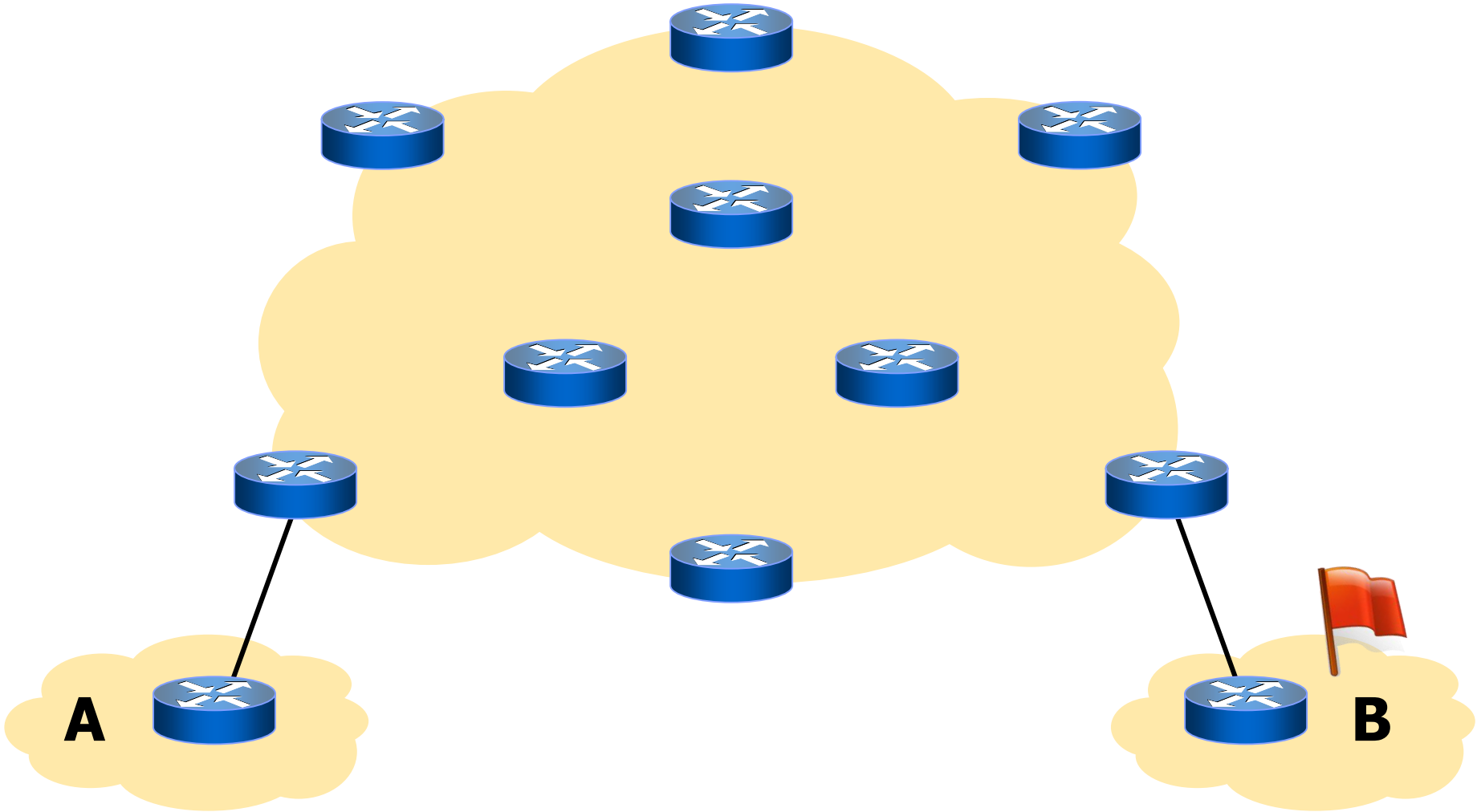
- internal routers must support traffic flows from/to neighboring ases
  - choice 1: redistribute bgp routes into the igp
    - overgrowth of igp routing tables 
    - update churn from bgp affects the igp 
  - choice 2: route traffic flowing through via an ad-hoc overlay
    - internal routers know about border routers only

choice 1

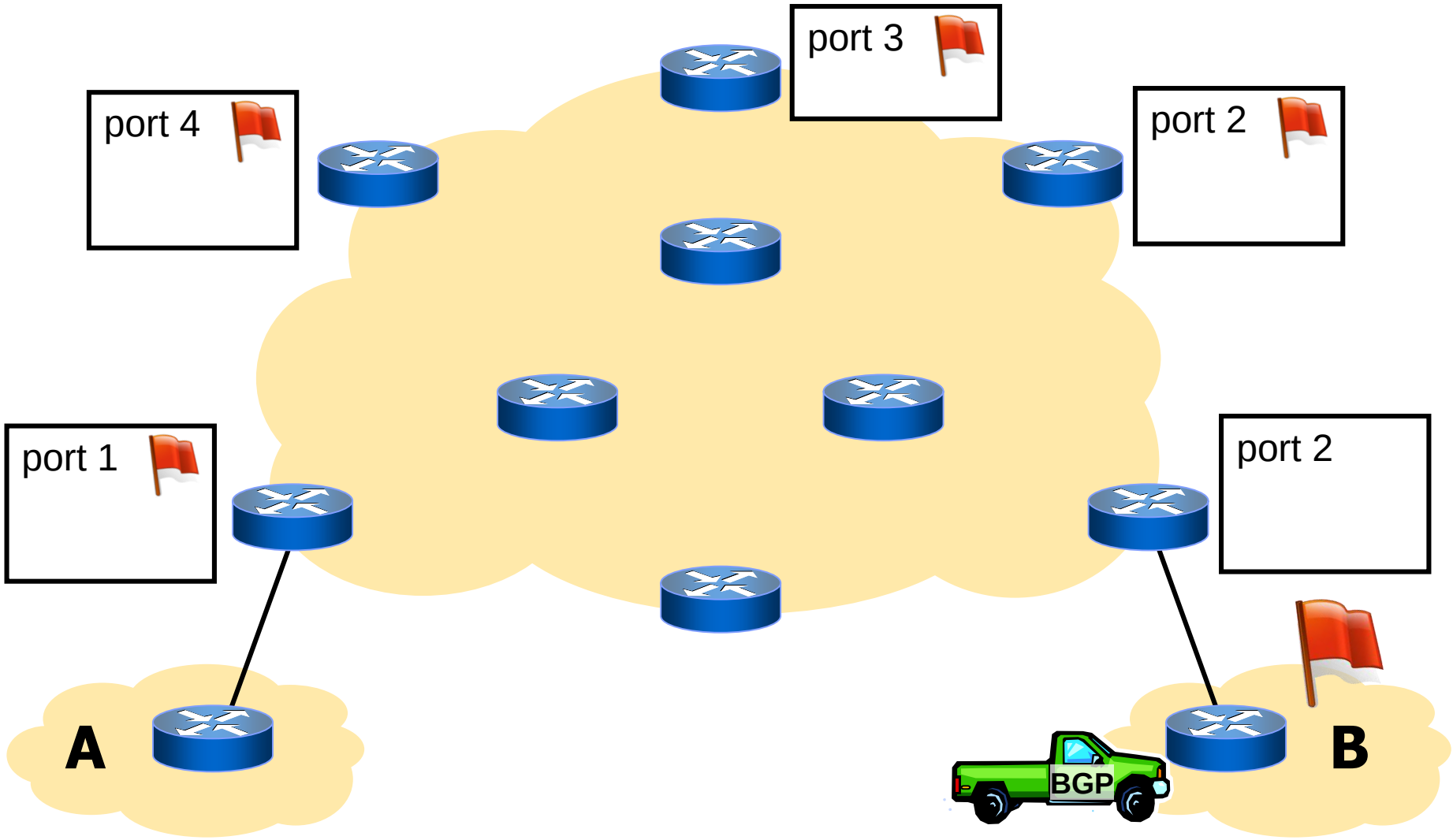
redistribution



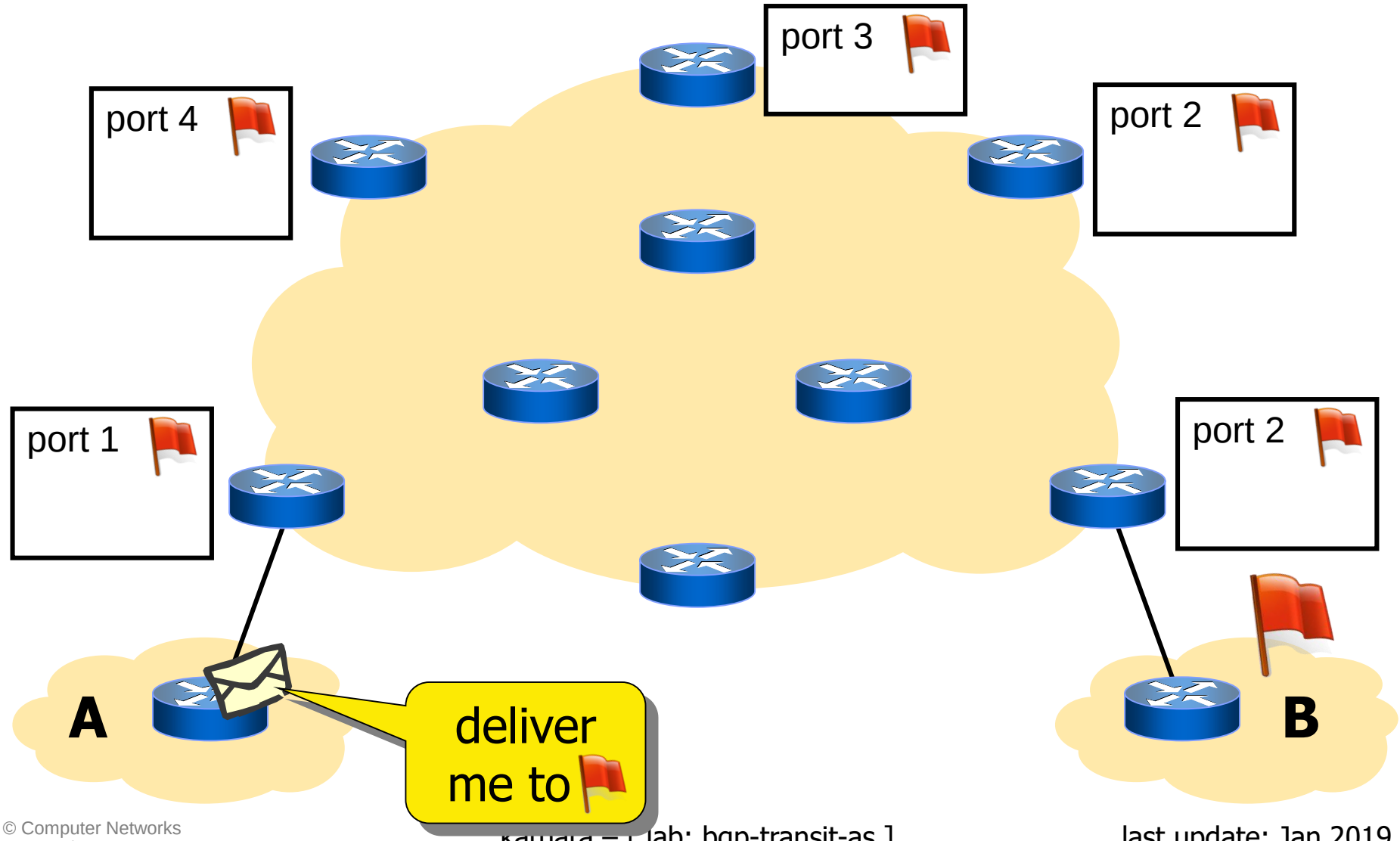
a reason to redistribute



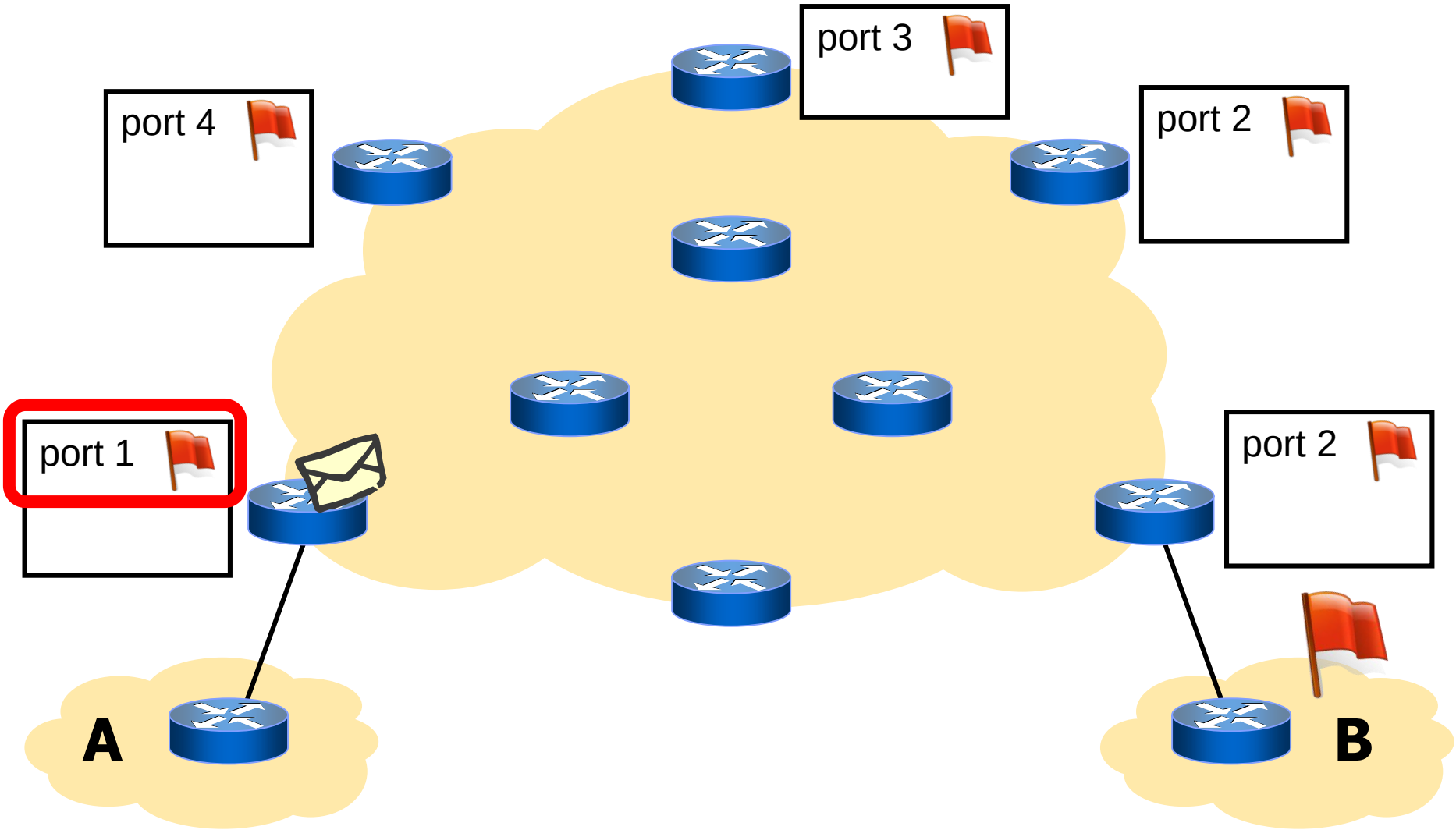
a reason to redistribute



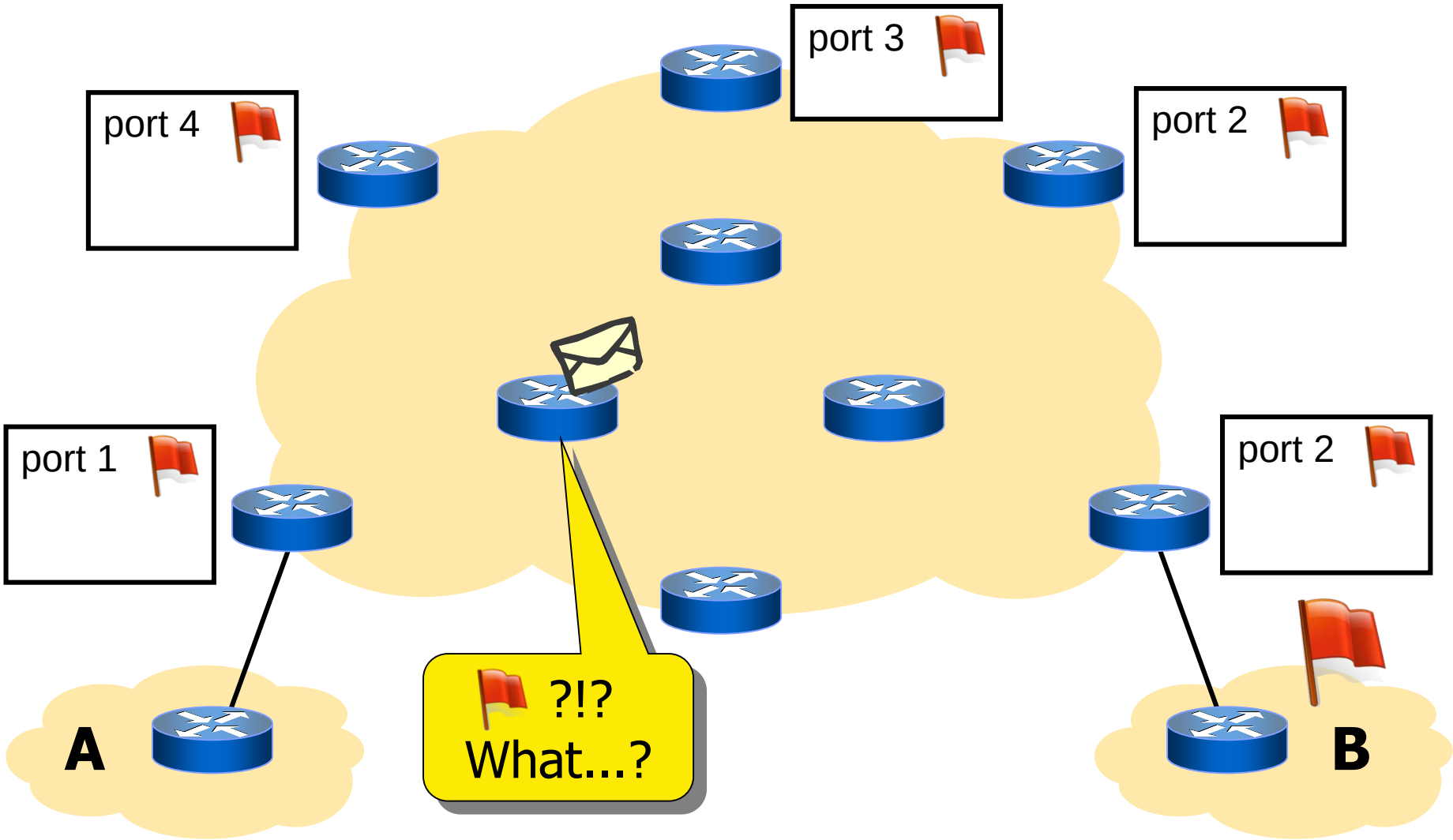
a reason to redistribute



a reason to redistribute



a reason to redistribute

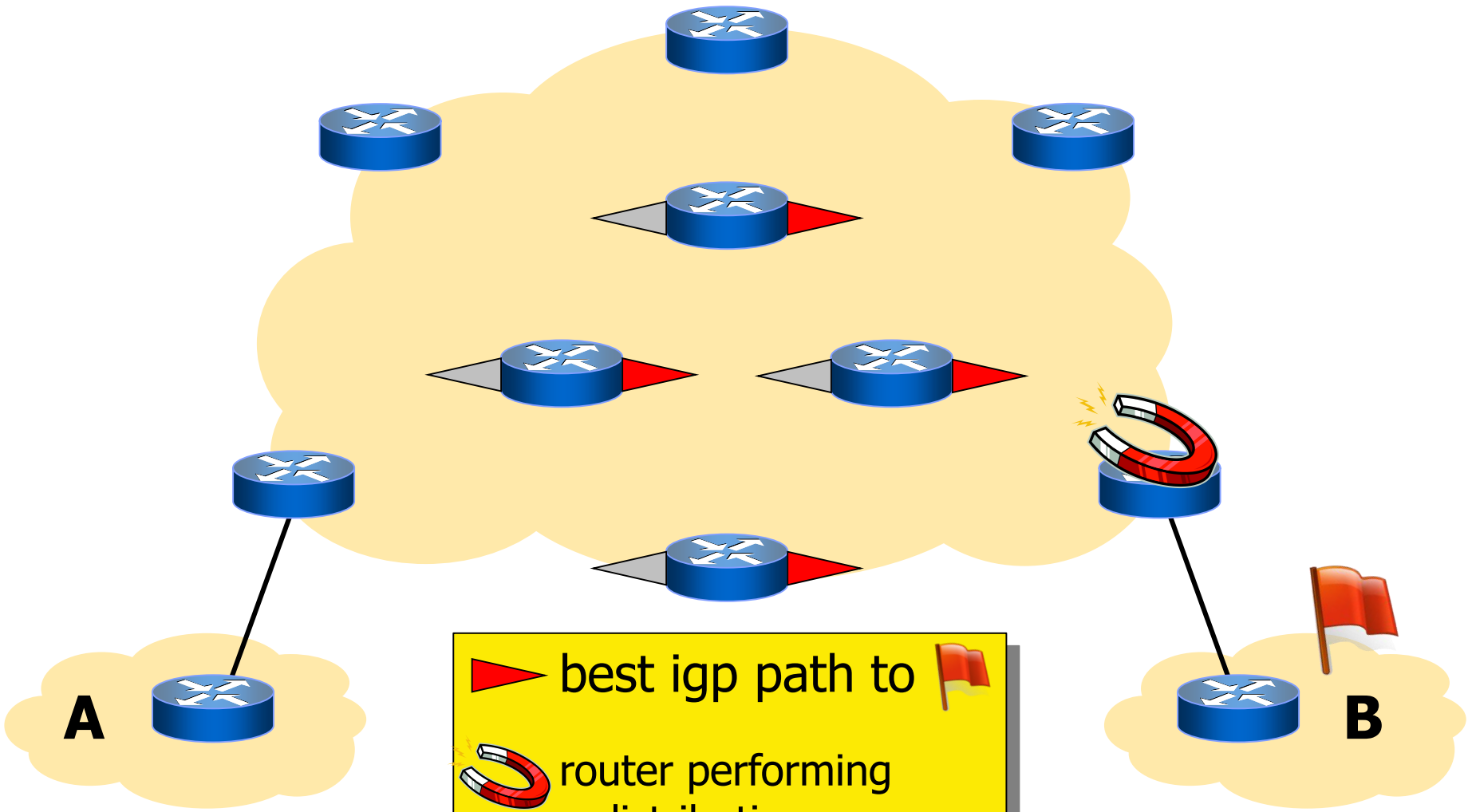


how to redistribute bgp → rip

zebra rip configuration file

```
router rip
    network eth1
    redistribute connected
    redistribute bgp
```

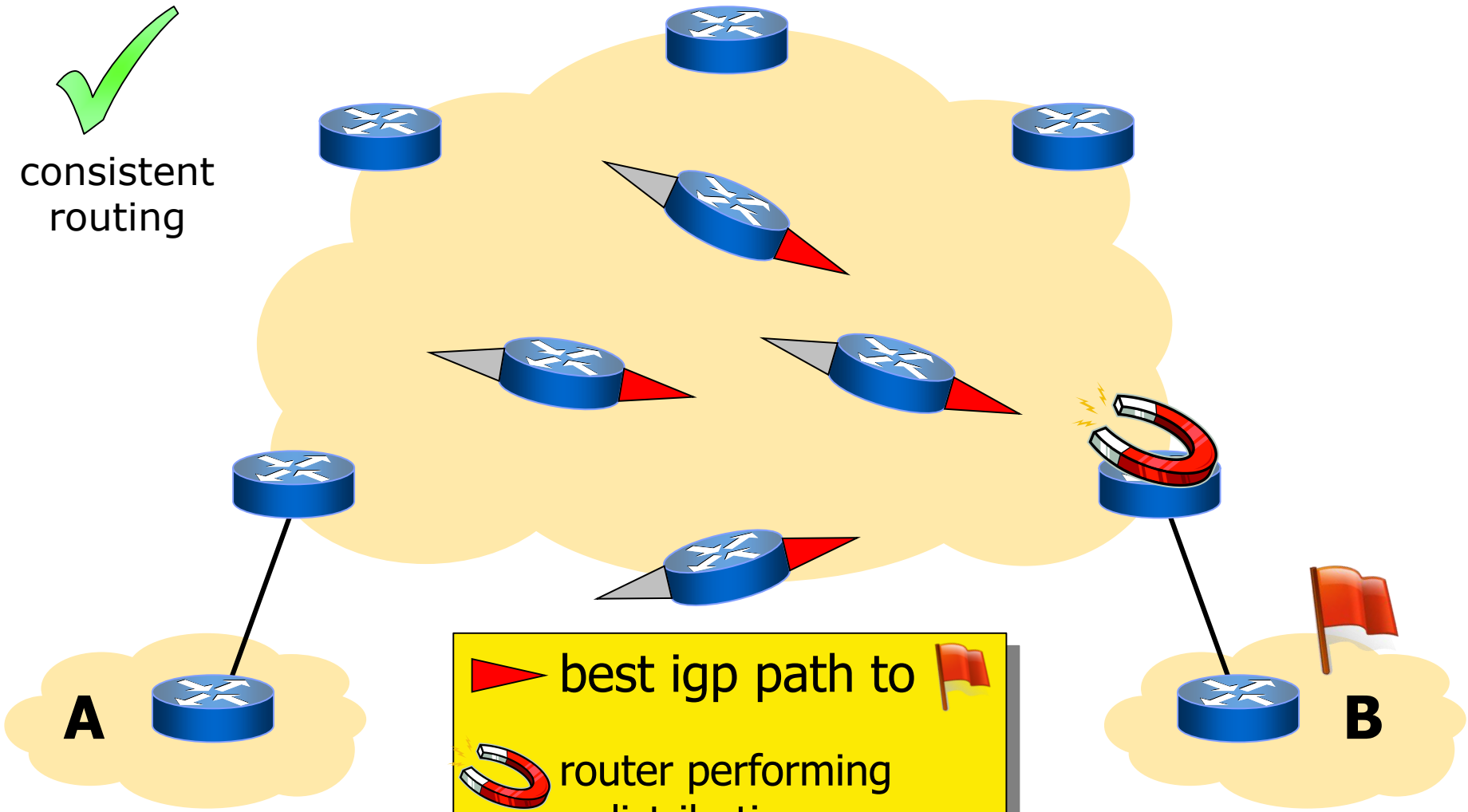
beware of redistributing ibgp!



last update: Jan 2019

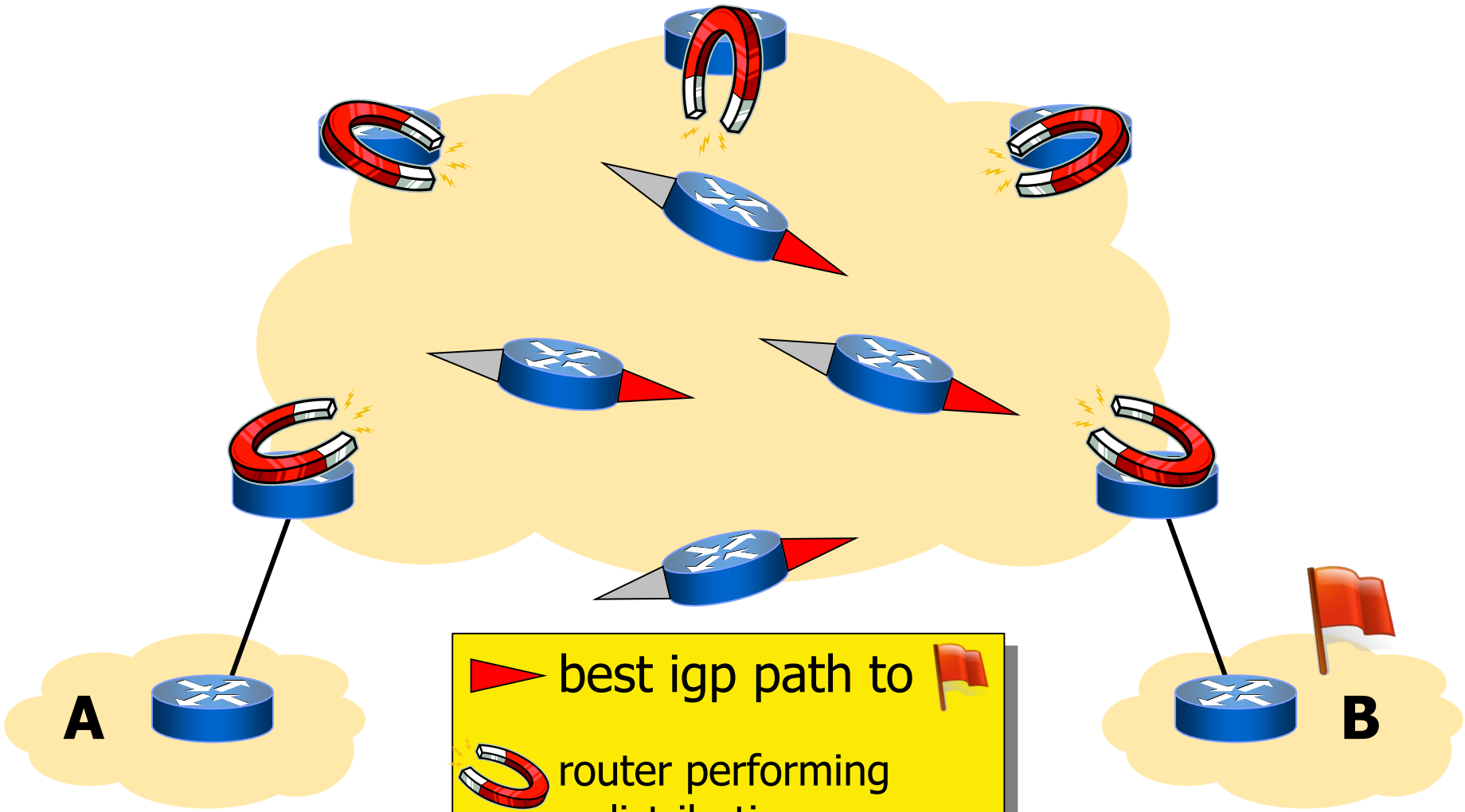
beware of redistributing ibgp!

consistent routing



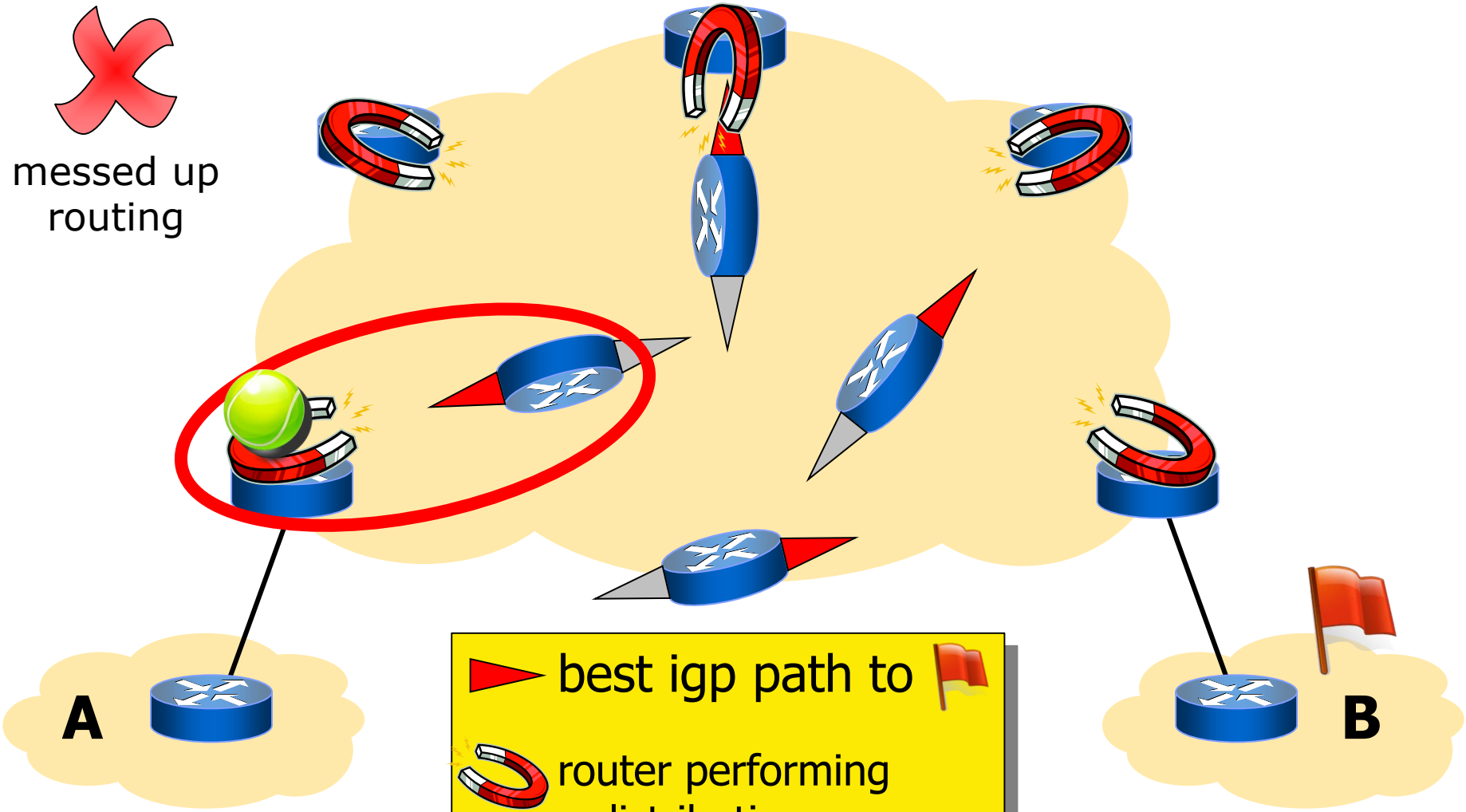
last update: Jan 2019

beware of redistributing ibgp!



last update: Jan 2019

beware of redistributing ibgp!

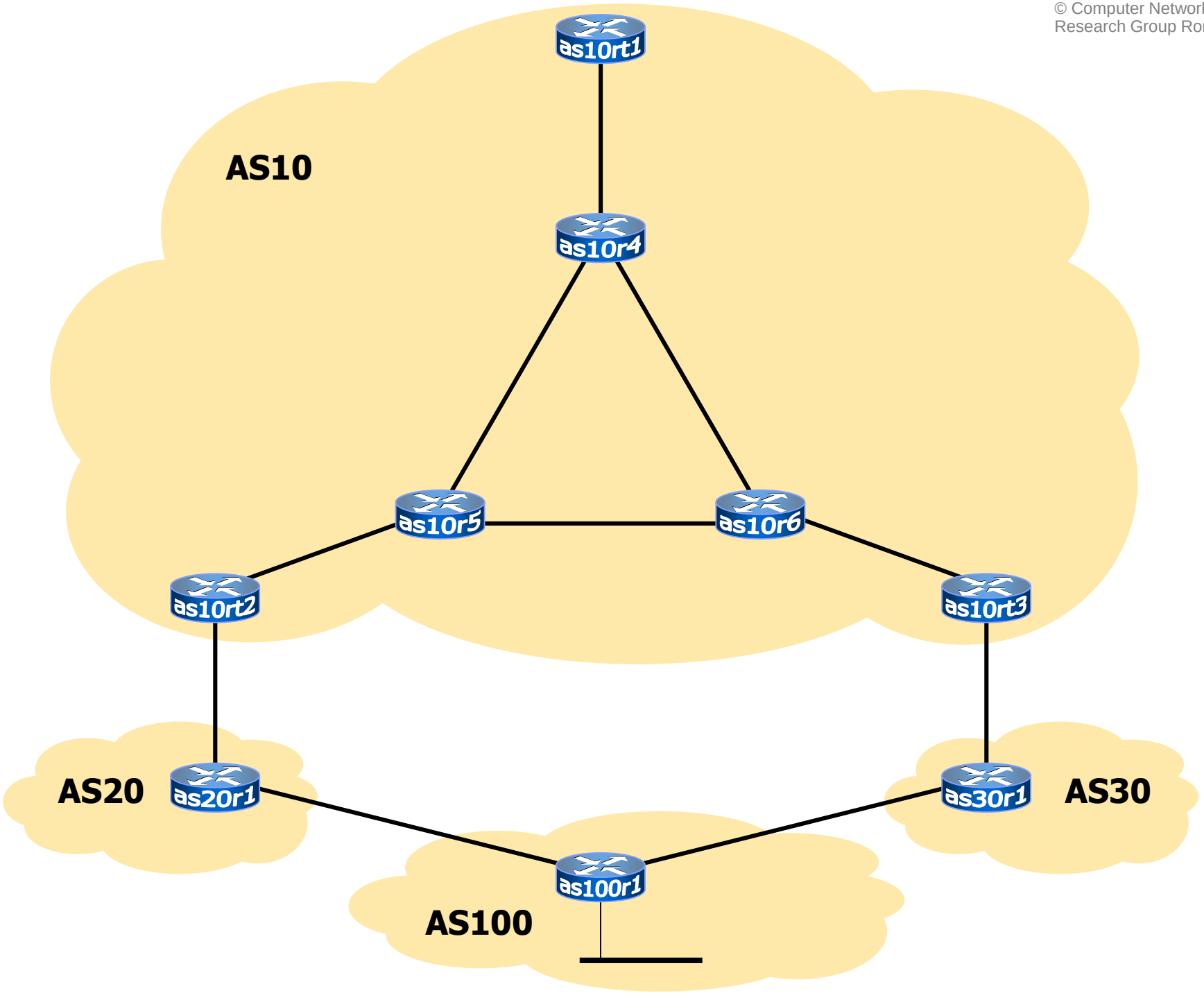


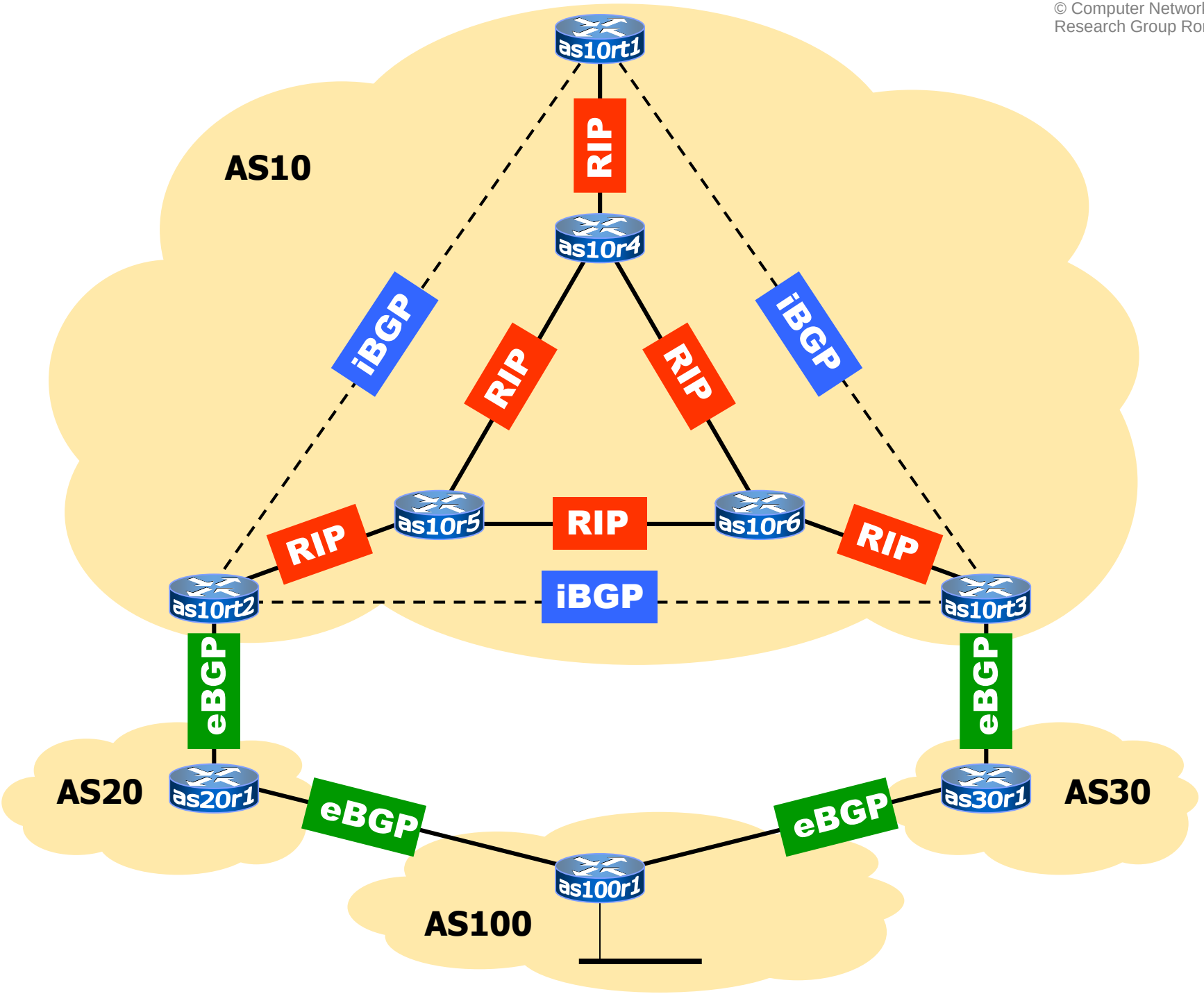
last update: Jan 2019

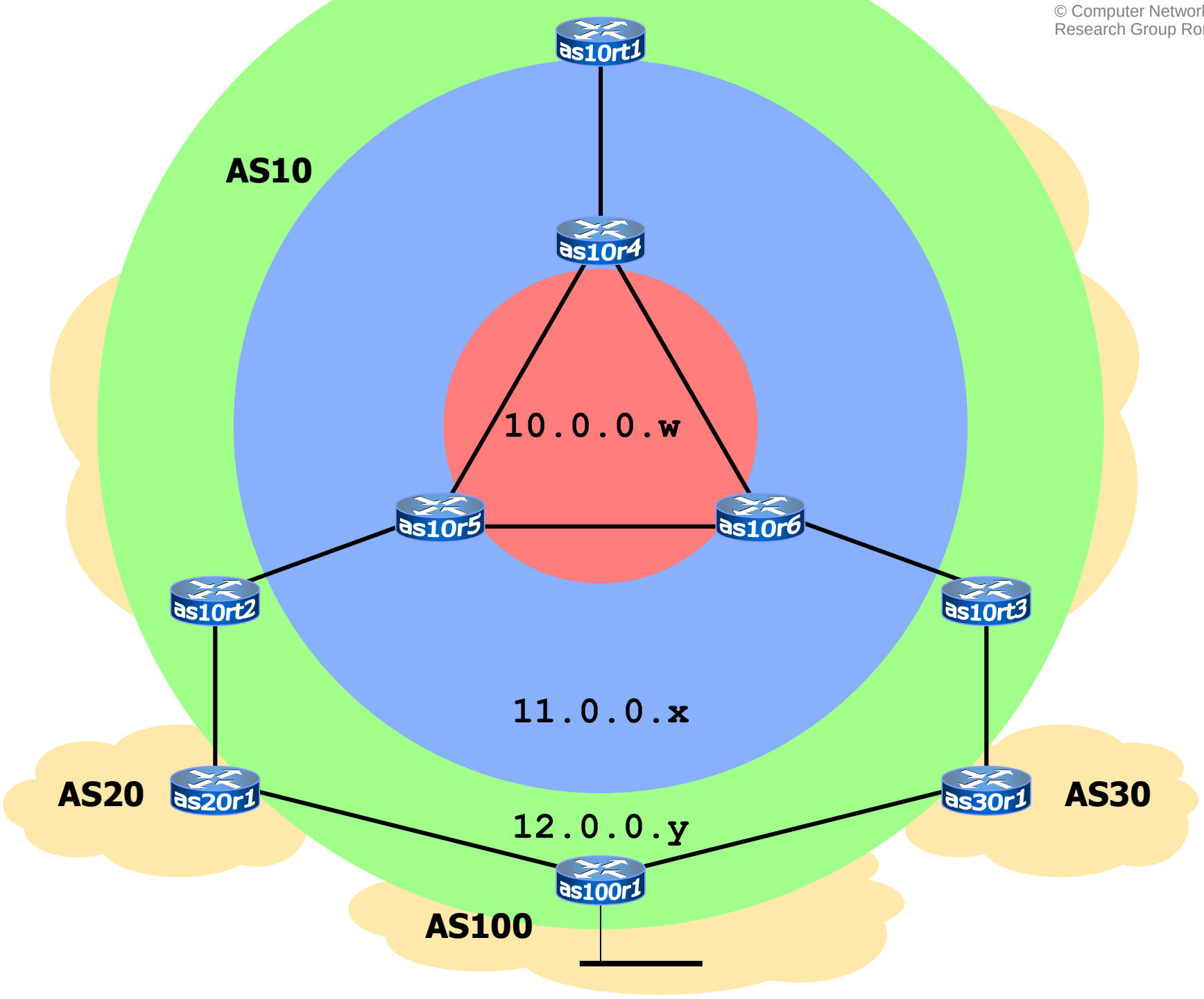
beware of redistributing ibgp!

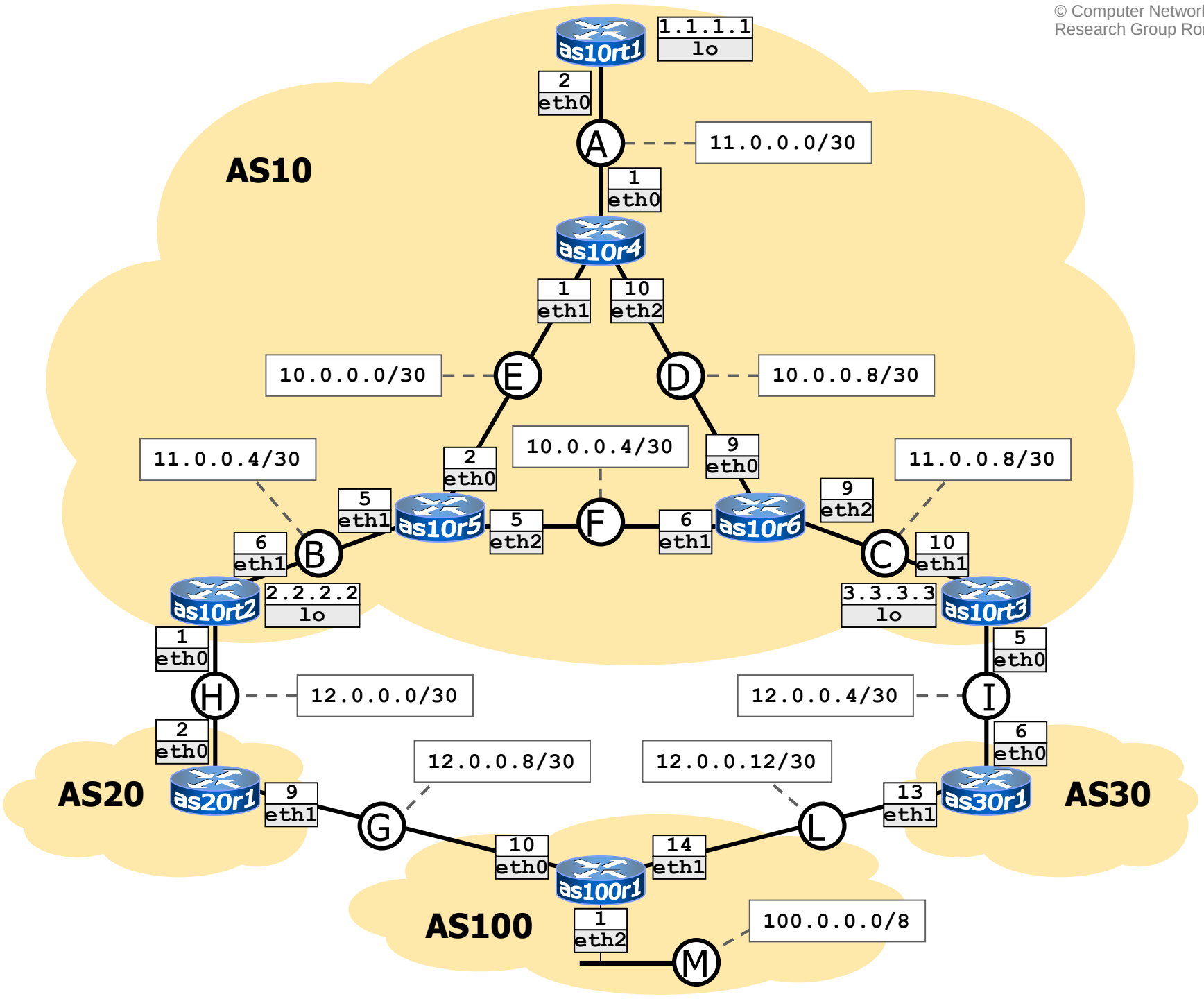
- cisco (and juniper) say:

> *By default, iBGP redistribution into IGP is disabled. To enable redistribution of iBGP routes into IGP, issue the* **`bgp redistribute-internal`** *command. Precautions should be taken to redistribute specific routes using route maps into IGP. Note: Redistributing internal Border Gateway Protocol (iBGP) routes into an Interior Gateway Protocol may cause routing loops within the Autonomous System (AS). This is not recommended. Route filters should be set to control the information which is imported into the IGP.*
>
> *https://supportforums.cisco.com/document/8621/unable-redistribute-ibgp-learnt-routes-igp-such-eigrp-ospf-and*









transit as: bgp peerings

- bgp peerings are established on loopback interfaces
  - improved resiliency
    - the peering stays up even if all the router's physical interfaces are down
- an extra loopback for each border router of as10
  - `ifconfig lo:1 2.2.2.2 netmask 255.255.255.255 up`
    - `lo:1` is an ip alias used for the peerings
    - the usual loopback address, `lo`, is still available

beware: using `ifconfig lo:1 2.2.2.2/32` sets up a `/0` netmask instead(!)

- a default route would unexpectedly be announced when loopback interfaces are redistributed in an igp

transit as: bgp peerings

- be careful when configuring peerings on the loopbacks
  - bgp complains if the source address of OPEN messages from a neighbor does not match the neighbor's address configured in the peering (in this case, the loopback address)
  - bgp messages come out of a physical interface, whose address is different from the loopback's
  - need to force the source address of bgp messages
    - **`update-source`**
- cisco says:

  You only have to use the **`update-source`** *command when someone is peering to your loopback address*

transit as: bgp peerings

zebra bgp configuration file

```
router bgp 10
   network 10.0.0.0/8
   network 12.0.0.0/30
   neighbor 1.1.1.1 remote-as 10
   neighbor 1.1.1.1 update-source 2.2.2.2
   neighbor 1.1.1.1 description as10rt1(iBGP)
   neighbor 3.3.3.3 remote-as 10
   neighbor 3.3.3.3 update-source 2.2.2.2
   neighbor 3.3.3.3 description as10rt3(iBGP)
```

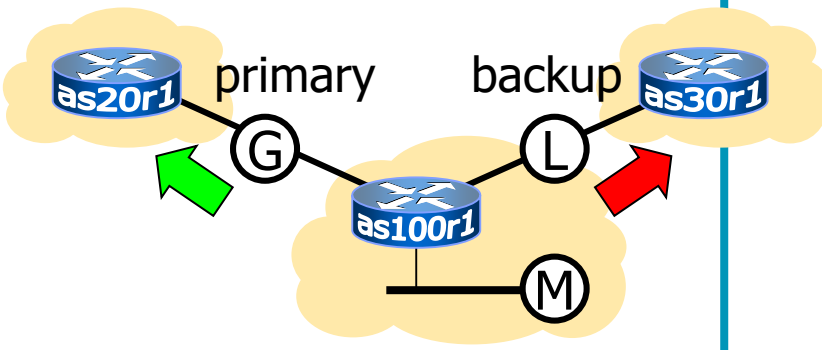
- note
  - **`update-source`** accepts an ip address or an interface name
  - zebra does not allow to set the **`update-source`** to an alias interface (e.g., **`lo:1`**)

transit as: some other flavouring

as100r1

```
as100r1:~# less /etc/zebra/bgpd.conf
hostname as100r1-bgpd
password zebra
!
ip prefix-list mineOut permit 100.0.0.0/8
!
route-map lowerPreference permit 10
   set local-preference 10
!
router bgp 100
   network 100.0.0.0/8
   neighbor 12.0.0.9 remote-as 20
   neighbor 12.0.0.9 description as20r1
   neighbor 12.0.0.9 prefix-list mineOut out
   neighbor 12.0.0.13 remote-as 30
   neighbor 12.0.0.13 description as30r1
   neighbor 12.0.0.13 prefix-list mineOut out
   neighbor 12.0.0.13 route-map lowerPreference in
```

/etc/zebra/bgpd.conf

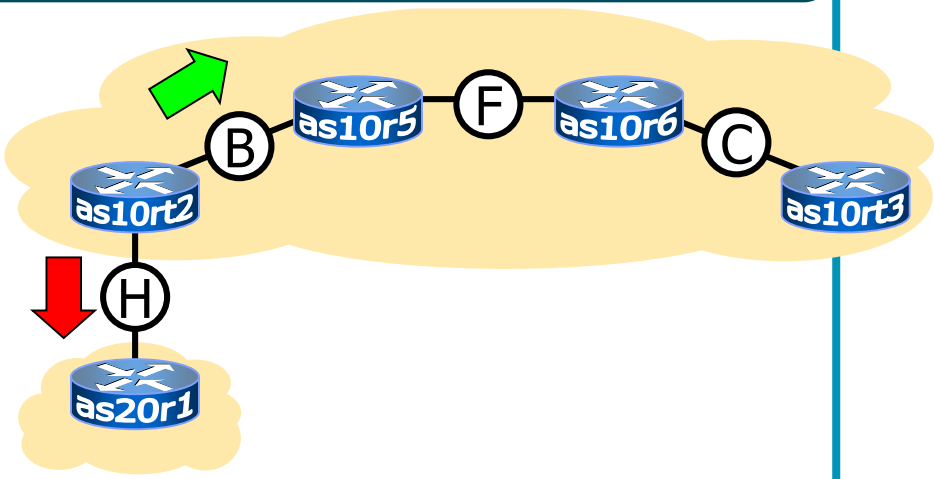


the customer prefers using link G

transit as: some other flavouring

as10rt2

```
as10rt2:~# less /etc/zebra/bgpd.conf
hostname as10rt2-bgpd
password zebra
...
!
route-map dePref permit 10
   set local-preference 10
!
router bgp 10
   network 10.0.0.0/8
   network 12.0.0.0/30
   neighbor 1.1.1.1 remote-as 10
   neighbor 1.1.1.1 update-source lo
   neighbor 1.1.1.1 description as10rt1(iBGP)
   neighbor 3.3.3.3 remote-as 10
   neighbor 3.3.3.3 update-source lo
   neighbor 3.3.3.3 description as10rt3(iBGP)
   neighbor 12.0.0.2 remote-as 20
   neighbor 12.0.0.2 description as20r1(eBGP)
   neighbor 12.0.0.2 route-map dePref in
   neighbor 12.0.0.2 prefix-list noDefault in
/etc/zebra/bgpd.conf
```

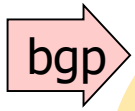


as10rt2 prefers using the egress router **as10rt3**

Zebra redistributes iBGP!

- AS10 receives announcements of 100.0.0.0/8 from AS20 and AS30 and prefers the one from AS30
- The behaviour is not fully deterministic: there is a race between BGP and RIP
  - Case 1: RIP wins the race
  - Case 2: BGP wins the race

case 1: **rip** wins the race



as10rt2

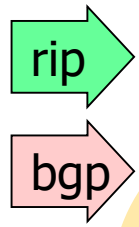
```
as10rt2:~# telnet localhost zebra
...
Router> show ip route 100.0.0.0/8
Routing entry for 100.0.0.0/8
  Known via "bgp", distance 200, metr
  Last update 00:00:35 ago
    12.0.0.6

Routing entry for 100.0.0.0/8
  Known via "rip", distance 120, metric 4, best
  Last update 00:01:16 ago
  * 11.0.0.5, via eth1

Router>
```

as10rt2 has learned both alternatives but prefers using rip information, no Loop

case 2: **bgp** wins the race



case 2: **bgp** wins the race

Beware of redistributing ibgp!

As10rt2

```
Router> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

R>* 1.1.1.1/32 [120/4] via 11.0.0.5, eth1, 00:02:07
C>* 2.2.2.2/32 is directly connected, lo
R>* 3.3.3.3/32 [120/4] via 11.0.0.5, eth1, 00:02:07
R>* 10.0.0.0/30 [120/2] via 11.0.0.5, eth1, 00:02:12
R>* 10.0.0.4/30 [120/2] via 11.0.0.5, eth1, 00:02:12
R>* 10.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:02:12
R>* 11.0.0.0/30 [120/3] via 11.0.0.5, eth1, 00:02:12
C>* 11.0.0.4/30 is directly connected, eth1
R>* 11.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:02:12
C>* 12.0.0.0/30 is directly connected, eth0
B   12.0.0.4/30 [200/0] via 3.3.3.3, 00:02:04
R>* 12.0.0.4/30 [120/4] via 11.0.0.5, eth1, 00:02:07
B>* 12.0.0.8/30 [20/0] via 12.0.0.2, eth0, 00:02:05
B>  12.0.0.12/30 [200/0] via 12.0.0.6 (recursive), 00:02:04 via 11.0.0.5, eth1, 00:02:04
B>  100.0.0.0/8 [200/0] via 12.0.0.6 (recursive), 00:02:04 via 11.0.0.5, eth1, 00:02:04
C>* 127.0.0.0/8 is directly connected, lo
```

To reach 100.0.0.0/8 go to as10r5

beware of redistributing ibgp!

As10r5

```
Router> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

R>* 1.1.1.1/32 [120/3] via 10.0.0.1, eth0, 00:02:38
R>* 2.2.2.2/32 [120/2] via 11.0.0.6, eth1, 00:02:42
R>* 3.3.3.3/32 [120/3] via 10.0.0.6, eth2, 00:02:42
C>* 10.0.0.0/30 is directly connected, eth0
C>* 10.0.0.4/30 is directly connected, eth2
R>* 10.0.0.8/30 [120/2] via 10.0.0.1, eth0, 00:02:44
R>* 11.0.0.0/30 [120/2] via 10.0.0.1, eth0, 00:02:44
C>* 11.0.0.4/30 is directly connected, eth1
R>* 11.0.0.8/30 [120/2] via 10.0.0.6, eth2, 00:02:43
R>* 12.0.0.0/30 [120/2] via 11.0.0.6, eth1, 00:02:42
R>* 12.0.0.4/30 [120/3] via 10.0.0.6, eth2, 00:02:42
R>* 12.0.0.8/30 [120/2] via 11.0.0.6, eth1, 00:02:34
R>* 12.0.0.12/30 [120/2] via 11.0.0.6, eth1, 00:02:34
R>* 100.0.0.0/8 [120/2] via 11.0.0.6, eth1, 00:02:34
C>* 127.0.0.0/8 is directly connected, lo
```

To reach 100.0.0.0/8 go to as10rt2
THERE IS A LOOP!

how to redistribute **e**bgp → rip

- Unfortunately, zebra redistributes both ebgp and ibgp
- **`route-map`**s can be applied on redistributed routes
  - for example based on recognizing ebgp next-hops

how to redistribute
ebgp → rip

As10rt2 zebra rip configuration file

```
ip prefix-list myNeighbors permit 12.0.0.0/30 le 32
route-map eBGP permit 10
   match ip next-hop prefix-list myNeighbors
router rip
   network eth1
   redistribute connected
   redistribute bgp route-map eBGP
```

As10rt1 zebra rip configuration file

```
route-map eBGP deny 10
router rip
   network eth0
   redistribute connected
   redistribute bgp route-map eBGP
```

how to redistribute bgp → rip

Match only updates with next-hop in the prefix-list myNeighbors

```
ip p      -list myNeighbors permit 12.0.0.0/30 le 32
route-ma  eBGP permit 10
   match ip next-hop prefix-list my
router rip
   network eth1
   redistribute connected
   redistribute bgp route-map eBGP
```

match all the more specifics of the 12.0.0.0/30 network (next-hops are single ip addresses)
m

As10rt1 zebra rip configuration file

```
route-map eBGP deny 10
router rip
   network eth0
   redistribute connected
   redistribute bgp route-map eBGP
```

redistributing ibgp: Playing with route-map

As10rt2

```
Router> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

R>* 1.1.1.1/32 [120/4] via 11.0.0.5, eth1, 00:23:55
C>* 2.2.2.2/32 is directly connected, lo
R>* 3.3.3.3/32 [120/4] via 11.0.0.5, eth1, 00:23:55
R>* 10.0.0.0/30 [120/2] via 11.0.0.5, eth1, 00:23:56
R>* 10.0.0.4/30 [120/2] via 11.0.0.5, eth1, 00:23:56
R>* 10.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:23:56
R>* 11.0.0.0/30 [120/3] via 11.0.0.5, eth1, 00:23:56
C>* 11.0.0.4/30 is directly connected, eth1
R>* 11.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:23:55
C>* 12.0.0.0/30 is directly connected, eth0
B   12.0.0.4/30 [200/0] via 3.3.3.3, 00:23:49
R>* 12.0.0.4/30 [120/4] via 11.0.0.5, eth1, 00:23:55
B>* 12.0.0.8/30 [20/0] via 12.0.0.2, eth0, 00:23:48
R>* 12.0.0.12/30 [120/4] via 11.0.0.5, eth1, 00:23:40
B   12.0.0.12/30 [200/0] via 12.0.0.6 (recursive), 00:23:
                         via 11.0.0.5, eth1, 00:23:44
R>* 100.0.0.0/8 [120/4] via 11.0.0.5, eth1, 00:23:40
B   100.0.0.0/8 [200/0] via 12.0.0.6 (recursive), 00:23:44
                        via 11.0.0.5, eth1, 00:23:44
C>* 127.0.0.0/8 is directly connected, lo
```

To reach 100.0.0.0/8 go to as10r5

beware of redistributing ibgp!

As10r5

```
Router> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

R>* 1.1.1.1/32 [120/3] via 10.0.0.1, eth0, 00:22:24
R>* 2.2.2.2/32 [120/2] via 11.0.0.6, eth1, 00:22:24
R>* 3.3.3.3/32 [120/3] via 10.0.0.6, eth2, 00:22:23
C>* 10.0.0.0/30 is directly connected, eth0
C>* 10.0.0.4/30 is directly connected, eth2
R>* 10.0.0.8/30 [120/2] via 10.0.0.1, eth0, 00:22:24
R>* 11.0.0.0/30 [120/2] via 10.0.0.1, eth0, 00:22:24
C>* 11.0.0.4/30 is directly connected, eth1
R>* 11.0.0.8/30 [120/2] via 10.0.0.6, eth2, 00:22:23
R>* 12.0.0.0/30 [120/2] via 11.0.0.6, eth1, 00:22:24
R>* 12.0.0.4/30 [120/3] via 10.0.0.6, eth2, 00:22:23
R>* 12.0.0.8/30 [120/2] via 11.0.0.6, eth1, 00:22:16
R>* 12.0.0.12/30 [120/3] via 10.0.0.6, eth2, 00:22:08
R>* 100.0.0.0/8 [120/3] via 10.0.0.6, eth2, 00:22:08
C>* 127.0.0.0/8 is directly connected, lo
Router>
```

To reach 100.0.0.0/8 go to as10r6

beware of redistributing *bgp

as10r6

```
Router> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

R>* 1.1.1.1/32 [120/3] via 10.0.0.10, eth0, 00:08:33
R>* 2.2.2.2/32 [120/3] via 10.0.0.5, eth1, 00:08:32
R>* 3.3.3.3/32 [120/2] via 11.0.0.10, eth2, 00:08:33
R>* 10.0.0.0/30 [120/2] via 10.0.0.10, eth0, 00:08:33
C>* 10.0.0.4/30 is directly connected, eth1
C>* 10.0.0.8/30 is directly connected, eth0
R>* 11.0.0.0/30 [120/2] via 10.0.0.10, eth0, 00
R>* 11.0.0.4/30 [120/2] via 10.0.0.5, eth1, 00:
C>* 11.0.0.8/30 is directly connected, eth2
R>* 12.0.0.0/30 [120/3] via 10.0.0.5, eth1, 00:
R>* 12.0.0.4/30 [120/2] via 11.0.0.10, eth2, 00
R>* 12.0.0.8/30 [120/3] via 10.0.0.5, eth1, 00:
R>* 12.0.0.12/30 [120/2] via 11.0.0.10, eth2, 0
R>* 100.0.0.0/8 [120/2] via 11.0.0.10, eth2, 00:
C>* 127.0.0.0/8 is directly connected, lo
```

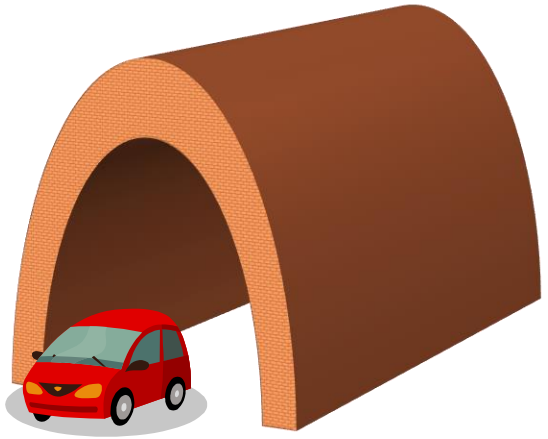
routing tables of the internal routers become unnecessarily large

choice 2

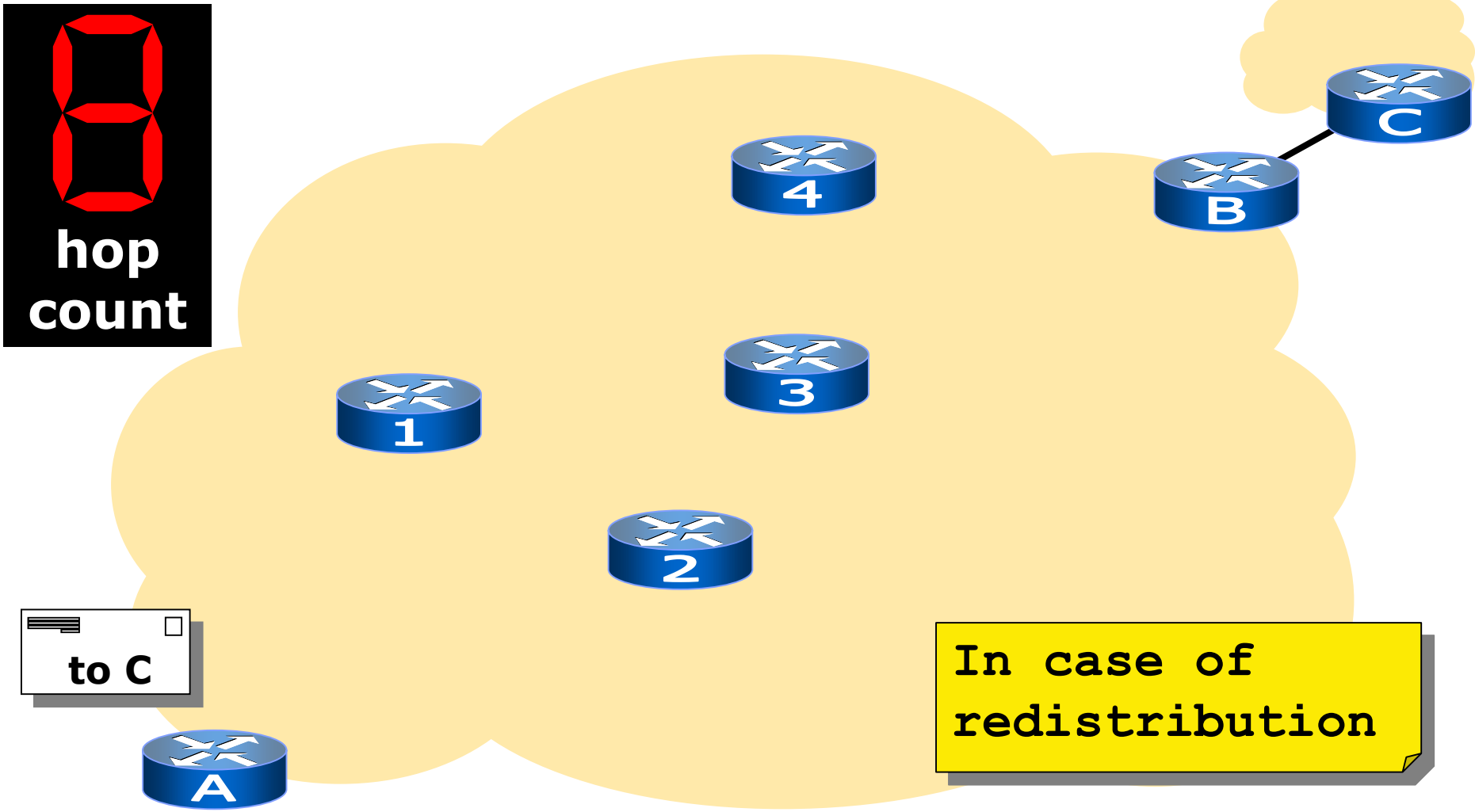
overlay

overlay

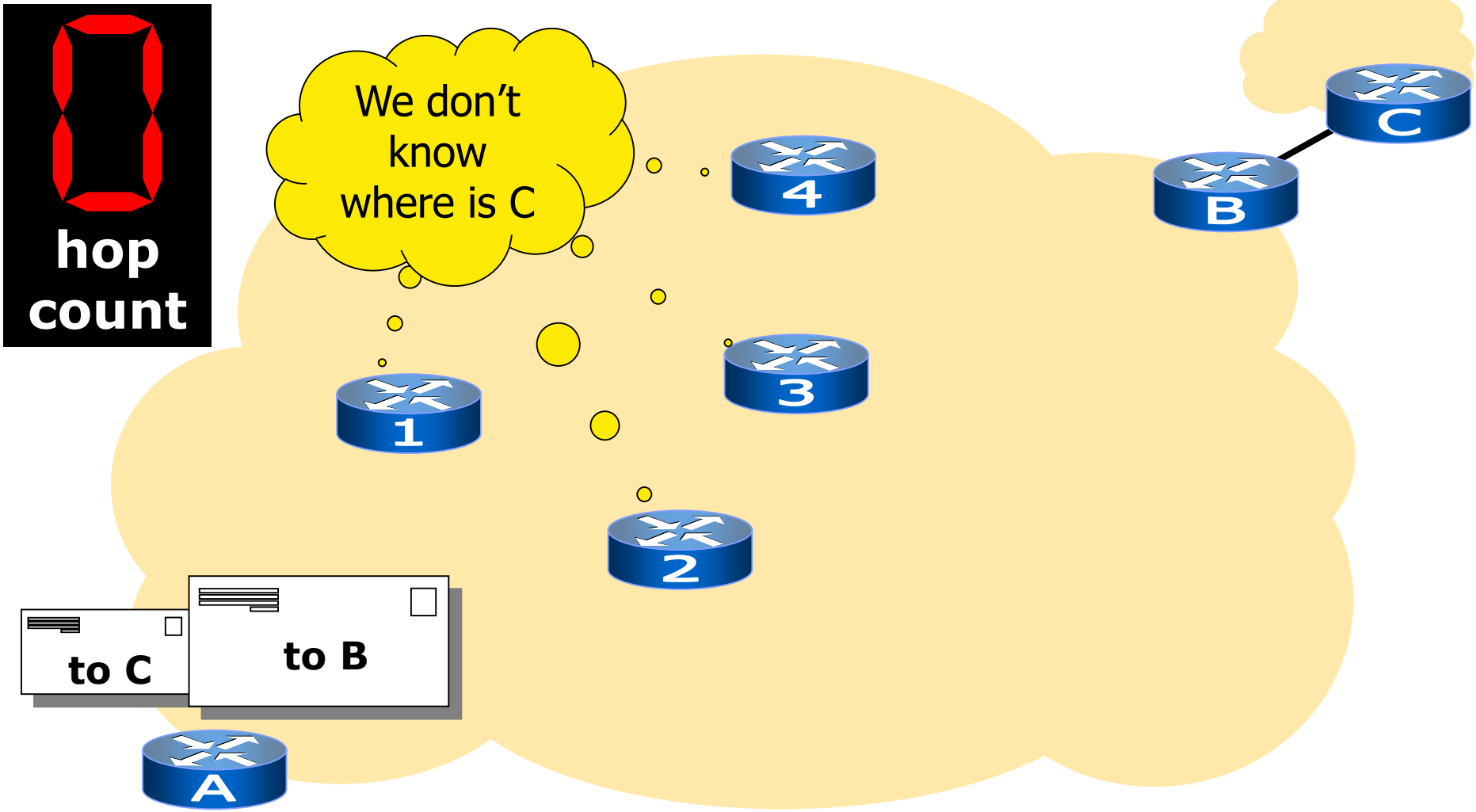
- ebgp is not redistributed into the igp
  - smaller routing tables
  - less igp churn
- ebgp next hops are reached via a direct link (tunnel)



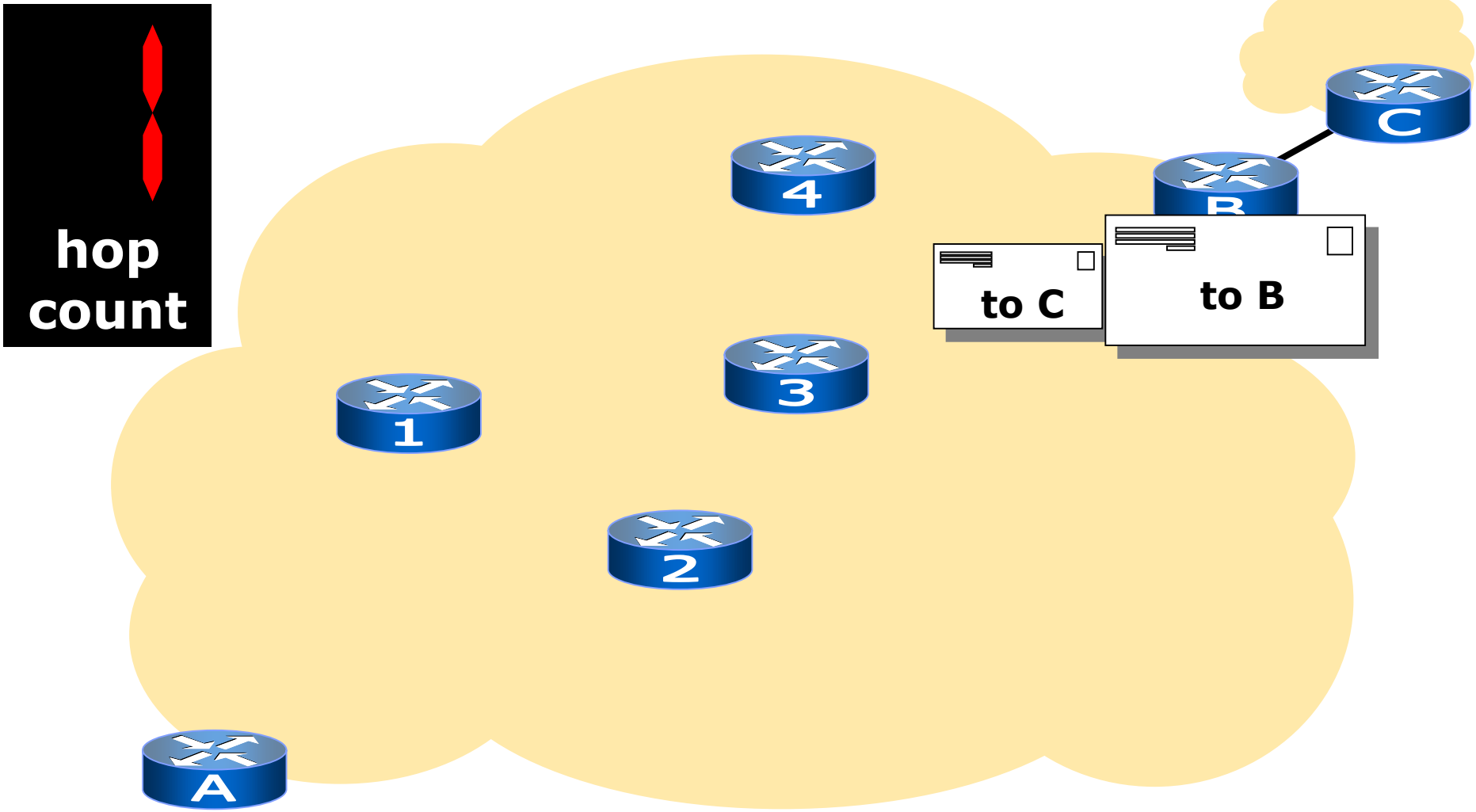
getting through the tunnel



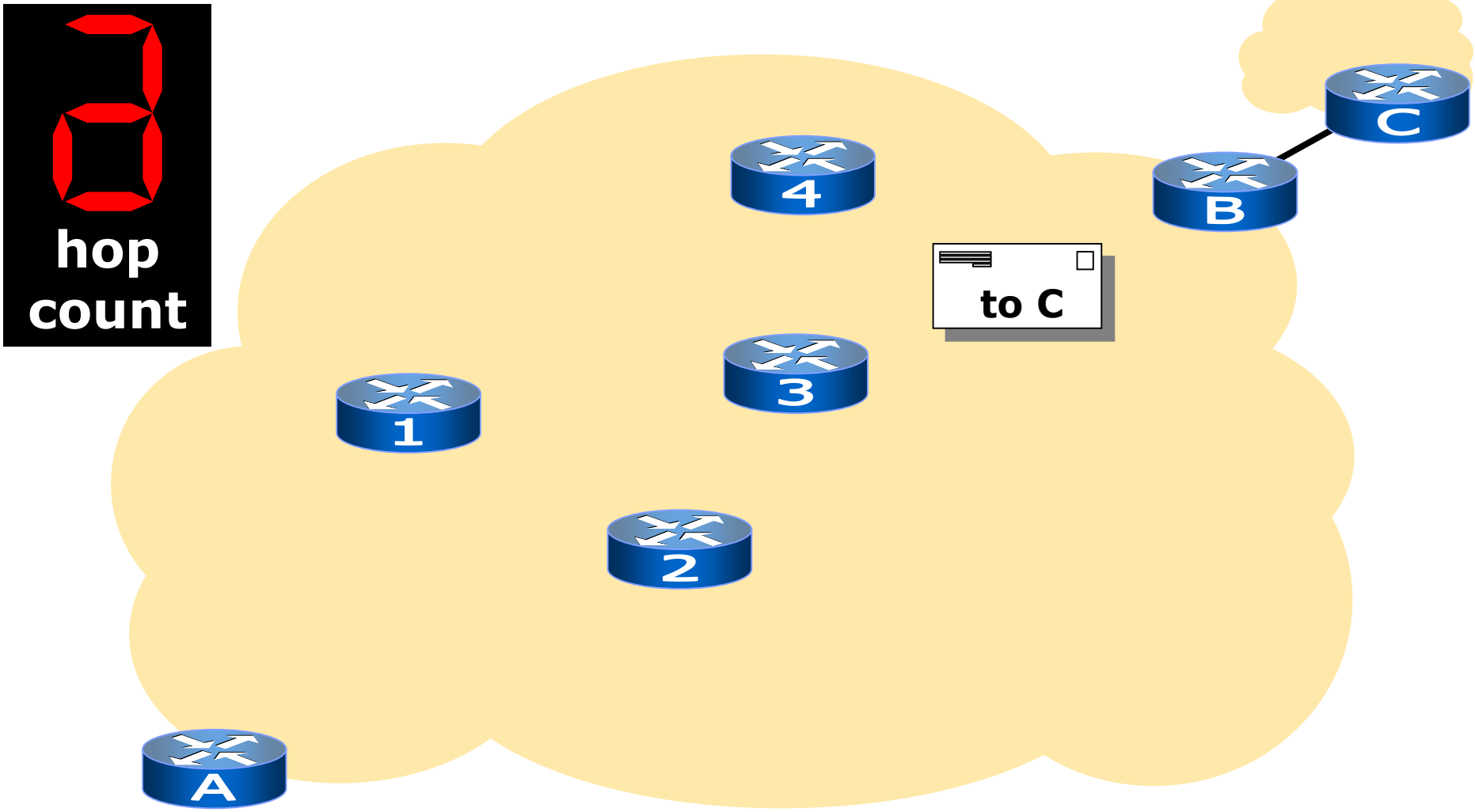
getting through the tunnel



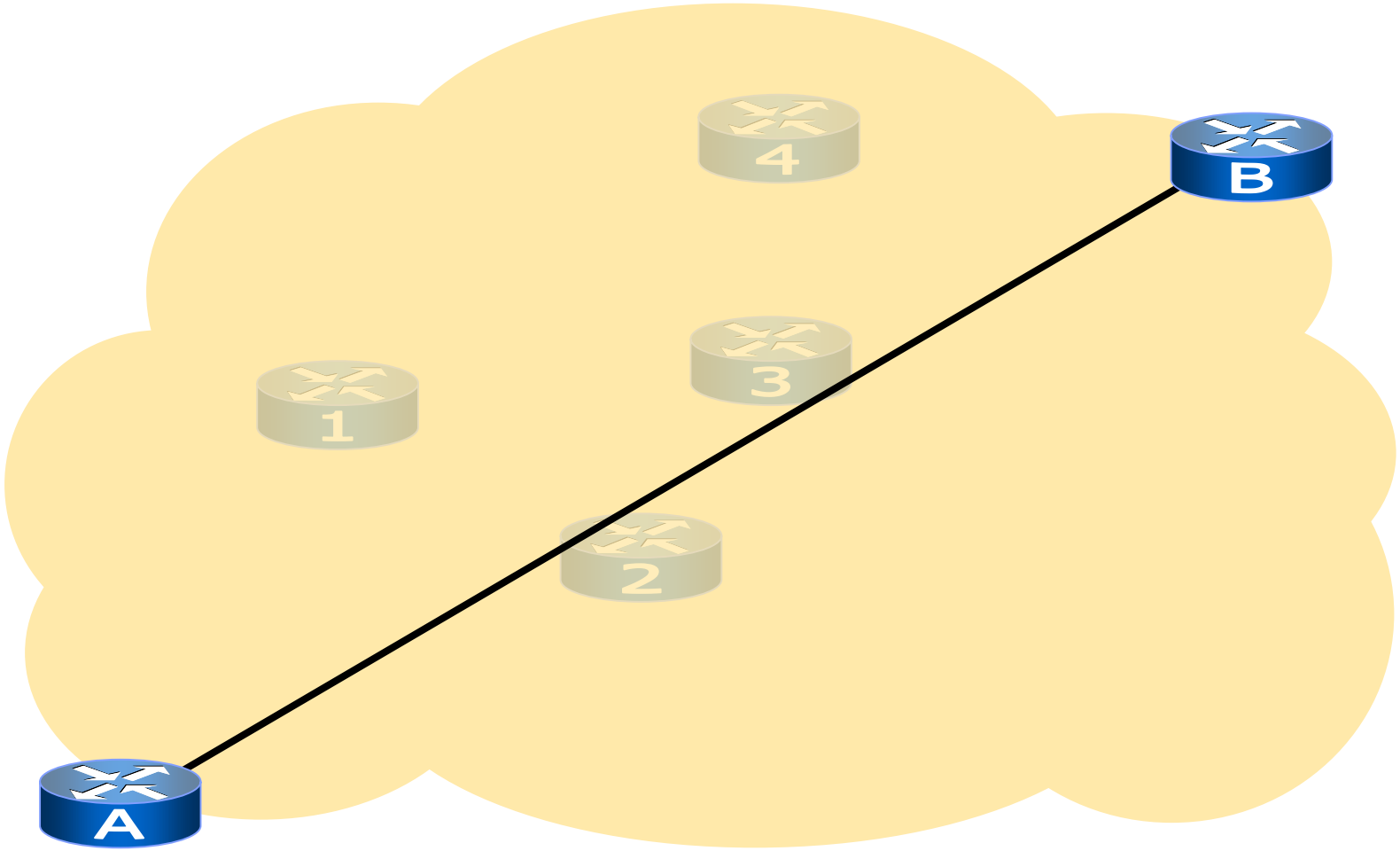
getting through the tunnel



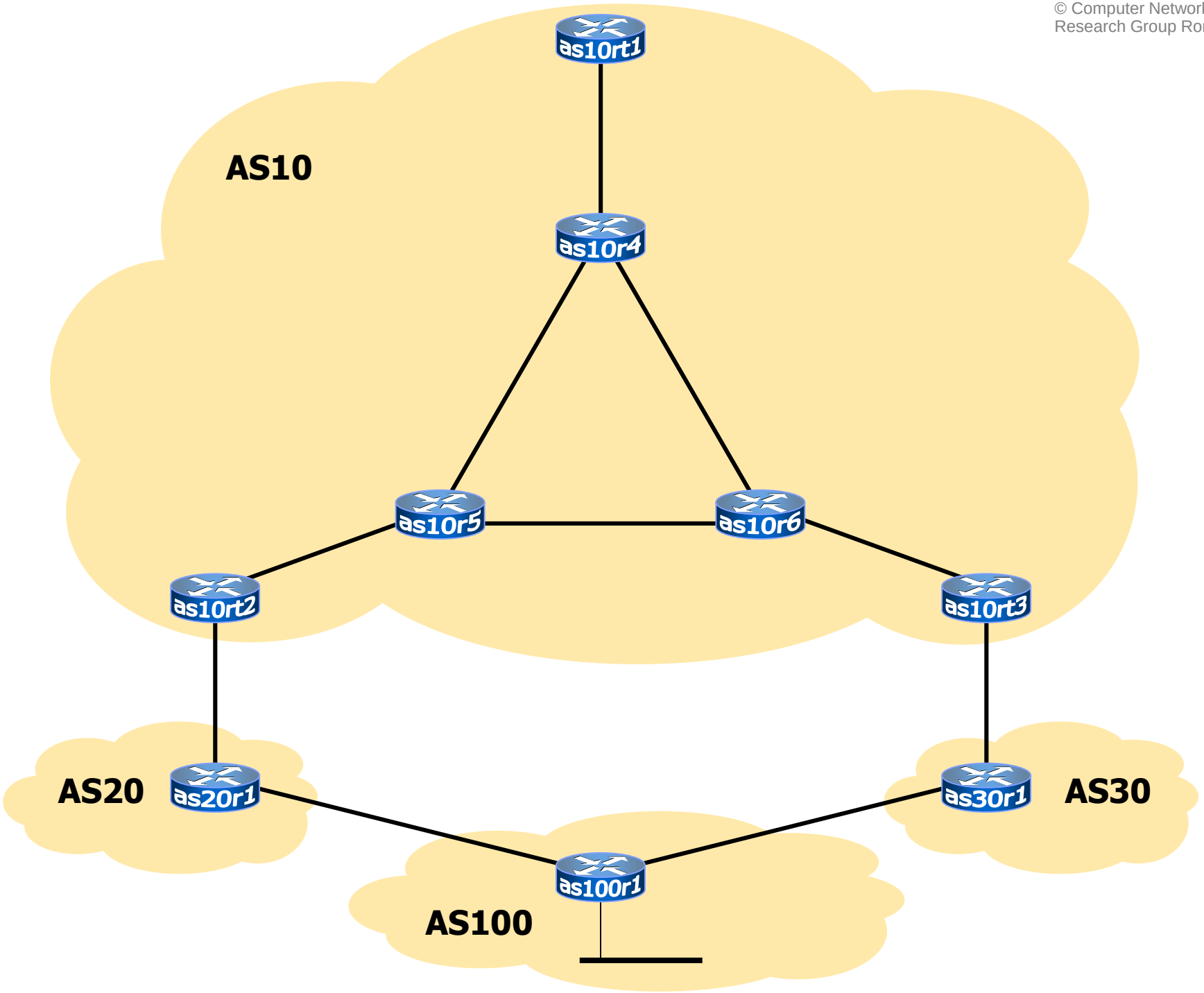
getting through the tunnel

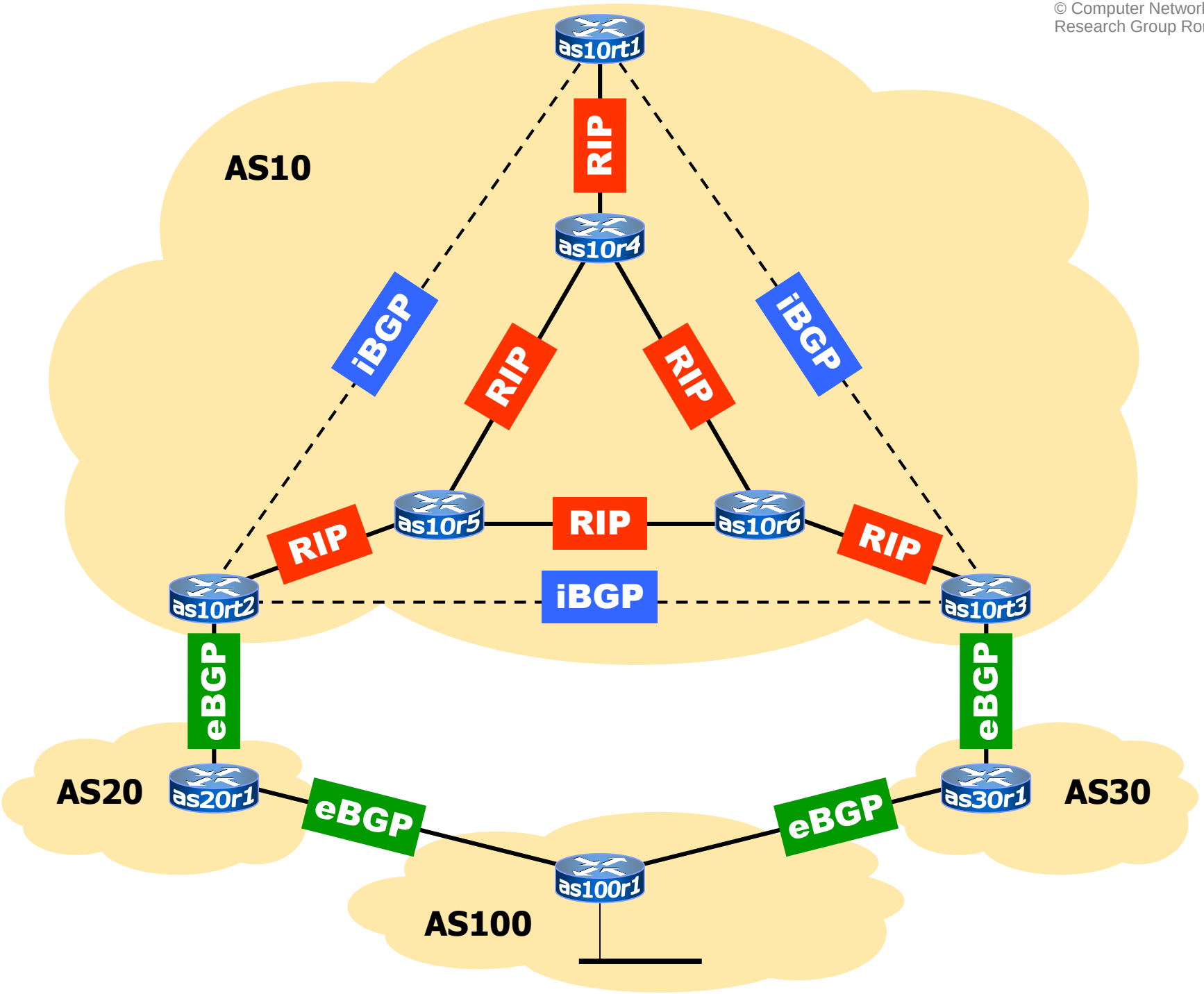


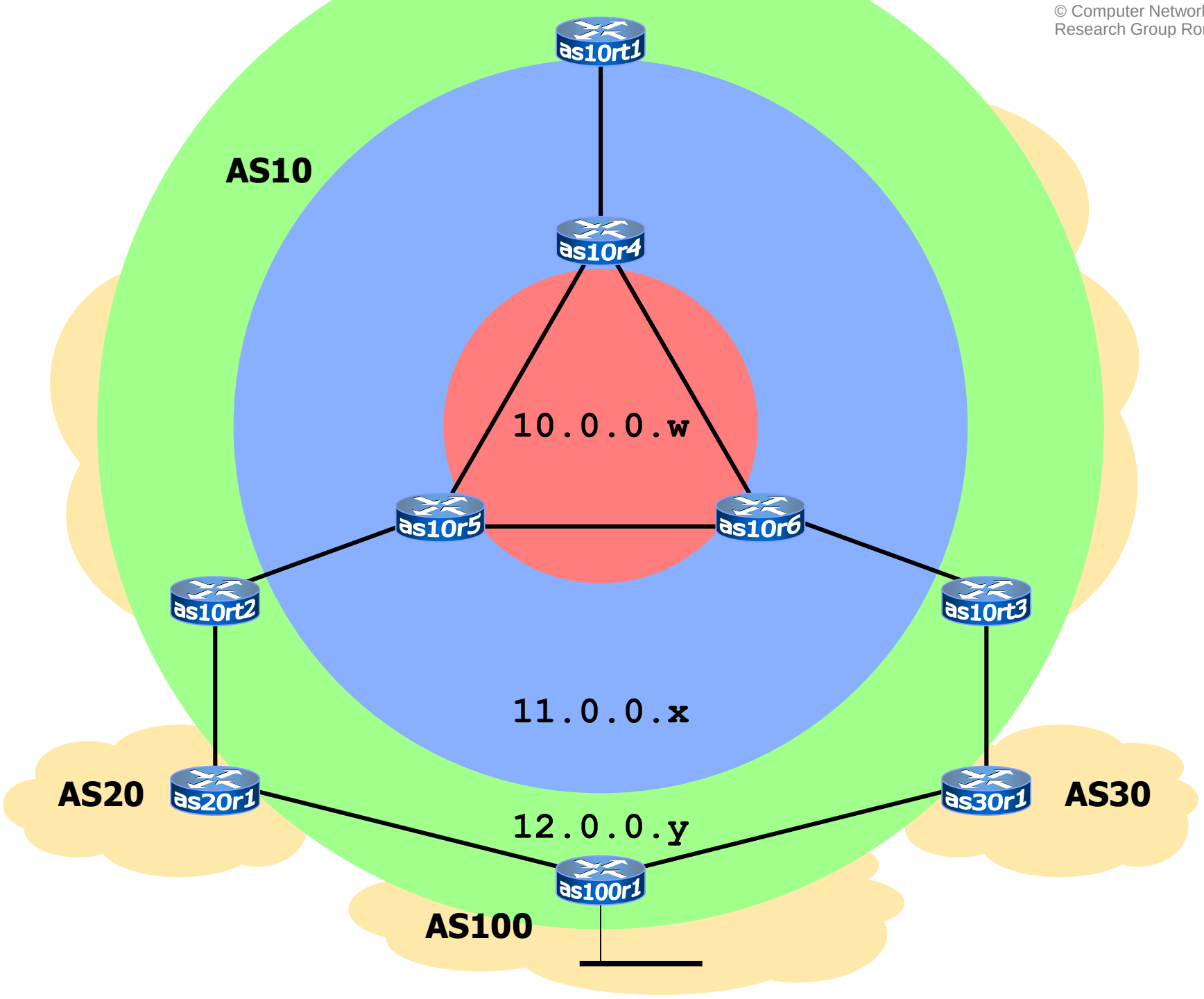
getting through the tunnel

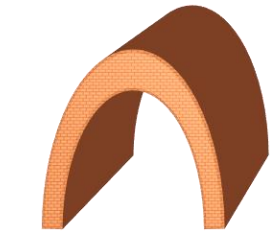


network topology









13.0.0.z

AS10

as10rt1

as10r4

10.0.0.w

as10r5

as10r6

as10rt2

as10rt3

11.0.0.x

AS20

as20r1

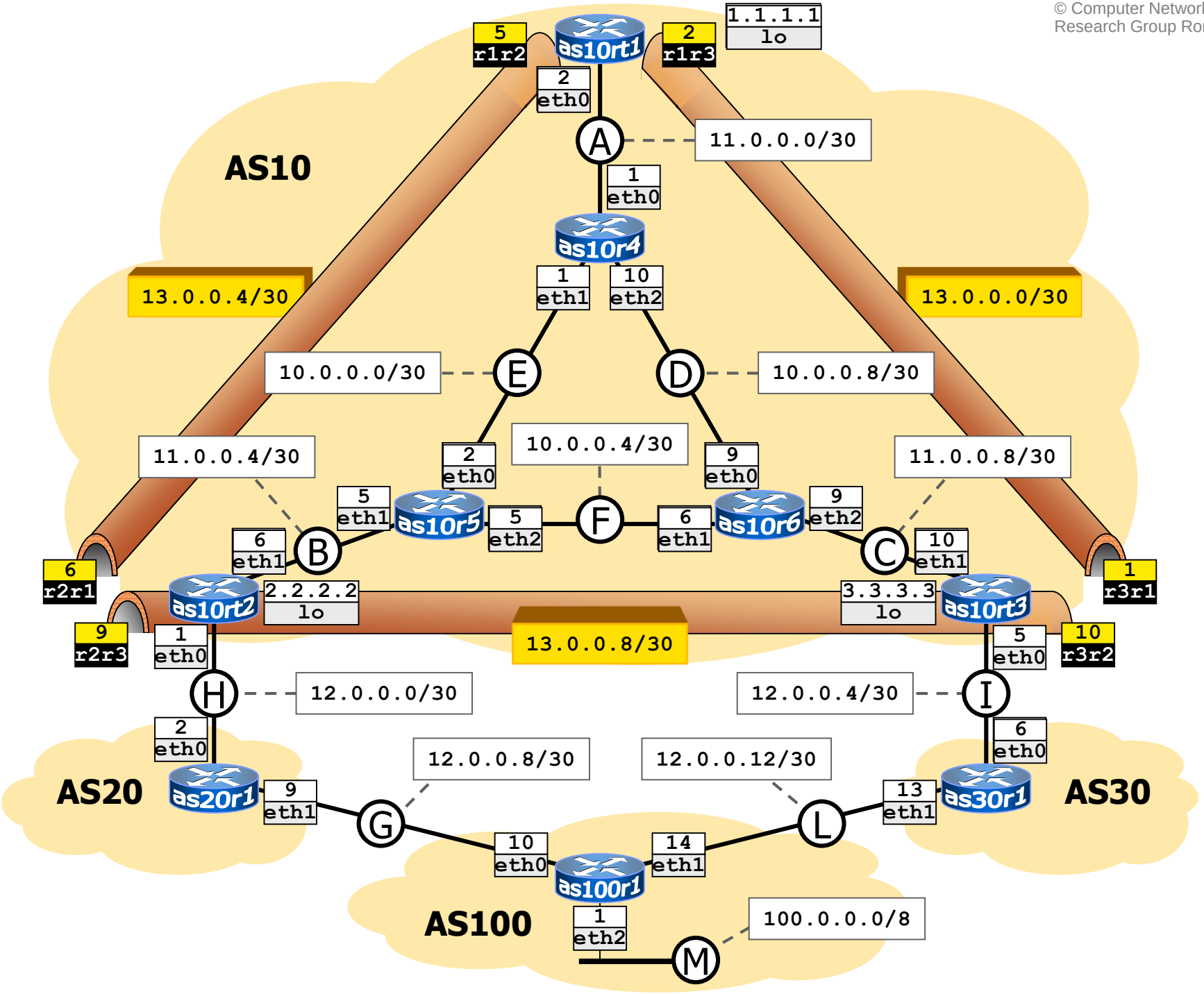
as30r1

AS30

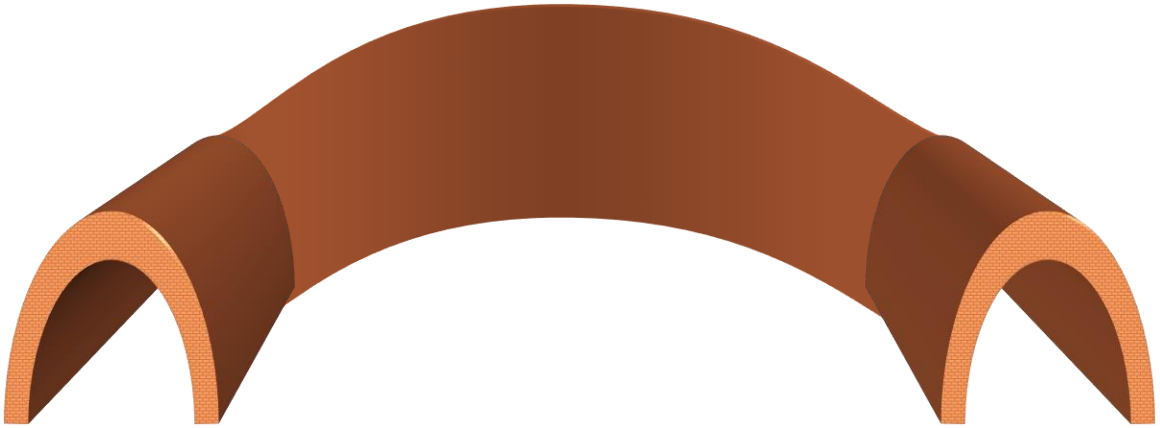
12.0.0.y

as100r1

AS100



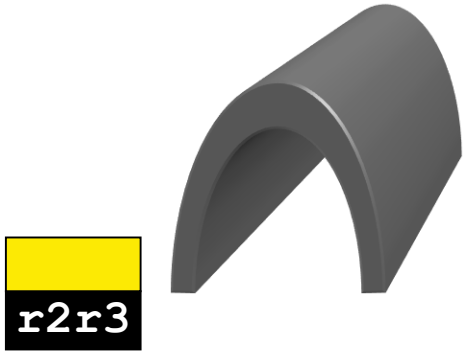
configuring a tunnel



as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
ip tunnel add r2r1 mode ipip remote 1.1.1.1 local 2.2.2.2 ttl 10
ip link set r2r1 multicast on
ip addr add dev r2r1 13.0.0.6 peer 13.0.0.5
ifconfig r2r1 up
```

configuring a tunnel

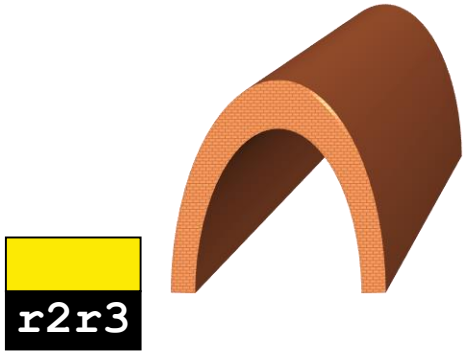


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.0 peer 13.0.0.10
ifconfig r2r3 up
ip tunnel add r2r1 mode ipip remote 1.1.1.1 local 2.2.2.2 ttl 10
ip link set r2r1 multicast on
ip addr add dev r2r1 13.0.0.5 peer 13.0.0.0
ifconfig r2r1 up
```

endpoint name (appears as a virtual interface on the router

configuring a tunnel

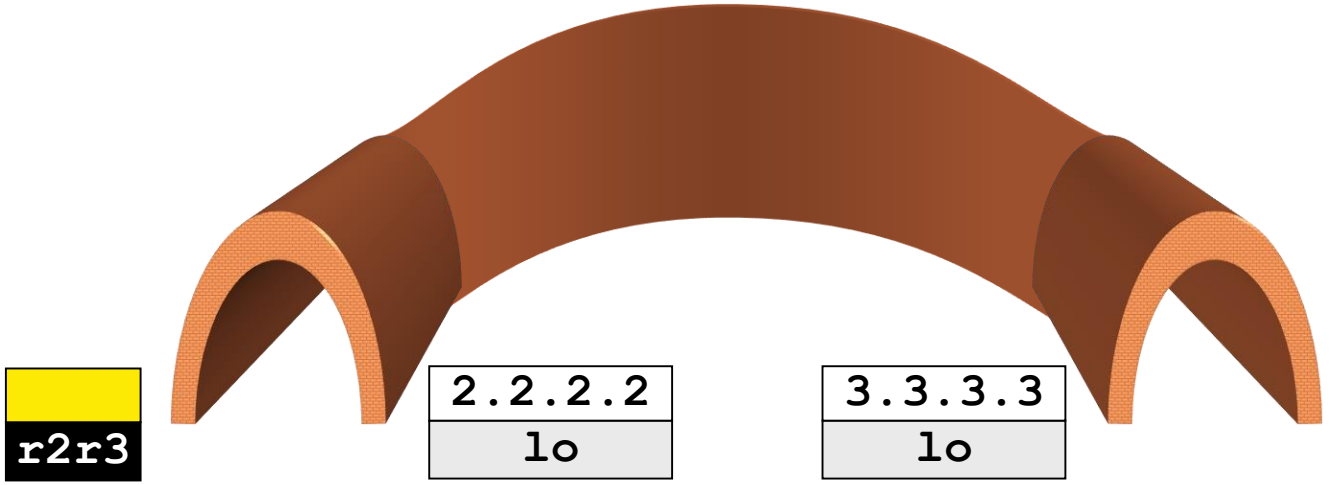


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 mult[illegible]cast on
ip addr add dev [illegible] 10
ifconfig r2r3 [illegible]
ip tunnel add [illegible] local 2.2.2.2 ttl 10
ip link set r2r[illegible]
ip addr add dev r2r1 13.0.0.6 peer 13.0.0.5
ifconfig r2r1 up
```

encapsulation type
(IP in IP)

configuring a tunnel

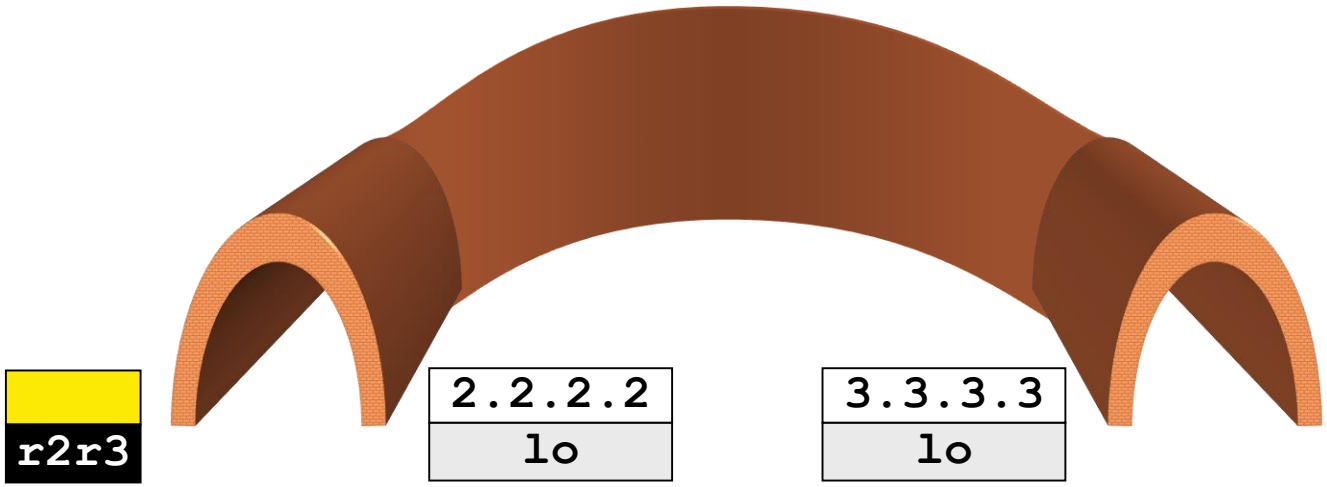


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 [illegible]0
ifconfig r2r3 up
ip tunnel add r2r1 mode ipip r[illegible] local 2.2.2.2 ttl 10
ip link set r2r1 multicast on
ip addr add dev r2r1 13.0.0.6 peer 13.0.0.5
ifconfig r2r1 up
```

tunnel endpoints

configuring a tunnel

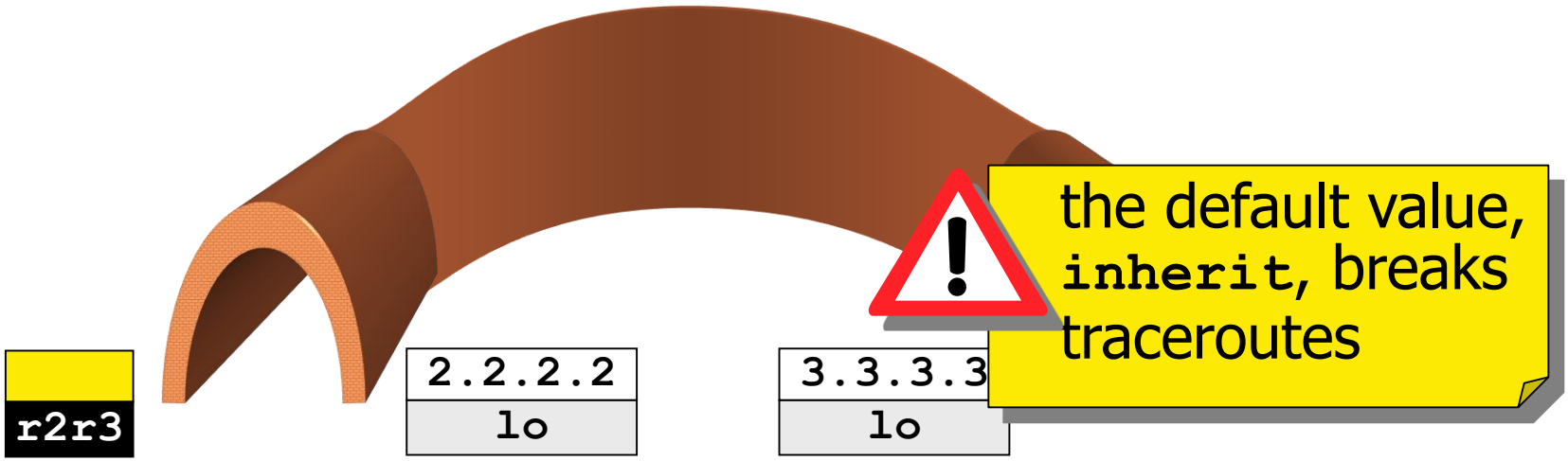


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
ip tunnel add r2r1 mode i
ip link set r2r1 multicas
ip addr add dev r2r1 13.0
ifconfig r2r1 up
```

ttl of the external envelope when a packet is encapsulated
(will expectedly travel through at most 4 hops, so any value >=4 should be fine)

configuring a tunnel

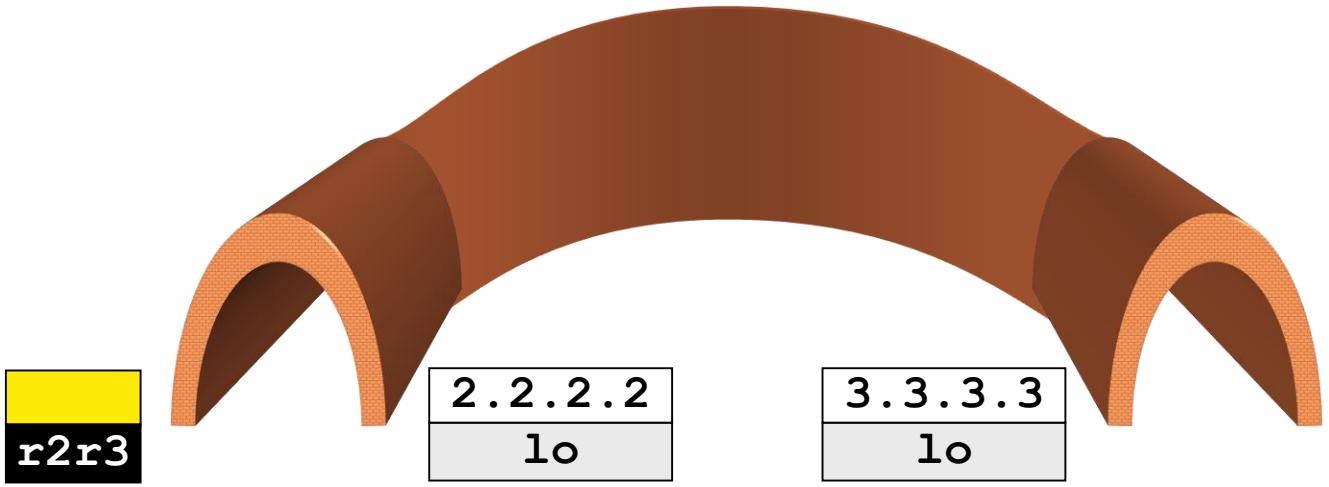


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.0 peer 13.0.0.10
ifconfig r2r3 up
ip tunnel add r2r1 mode i
ip link set r2r1 multicas
ip addr add dev r2r1 13.0
ifconfig r2r1 up
```

ttl of the external envelope when a packet is encapsulated
(will expectedly travel through at most 4 hops, so any value >=4 should be fine)

configuring a tunnel

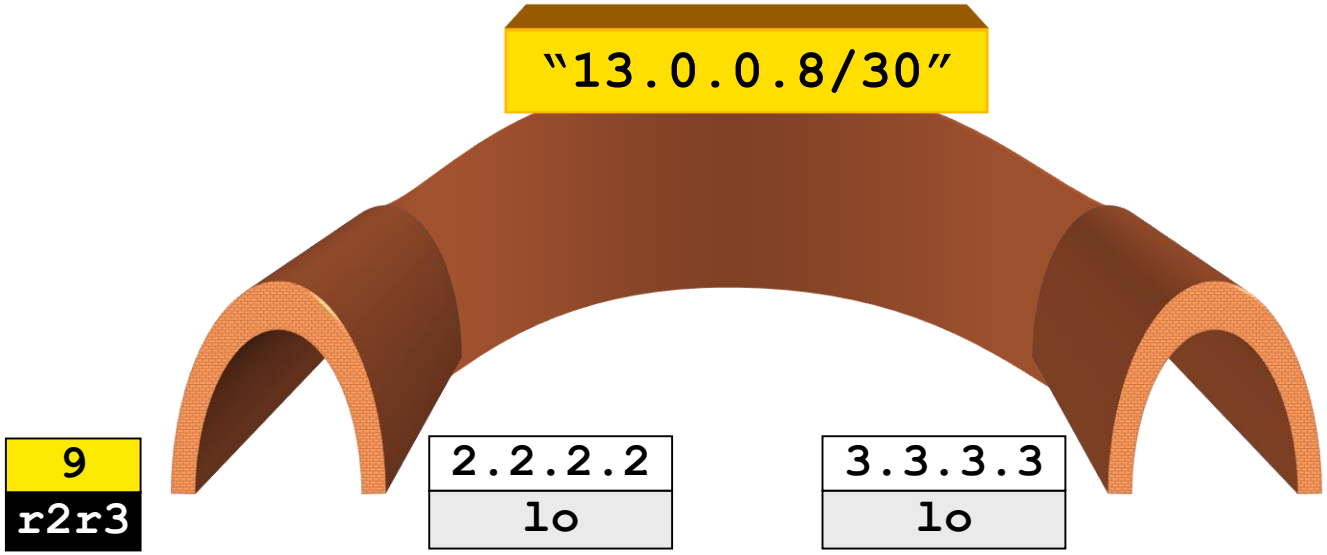


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 pee
ifconfig r2r3 up
ip tunnel add r2r1 mode ipip remote                    .2 ttl 10
ip link set r2r1 multicast on
ip addr add dev r2r1 13.0.0.6 peer 13.0.0.5
ifconfig r2r1 up
```

rip uses multicast packets

configuring a tunnel

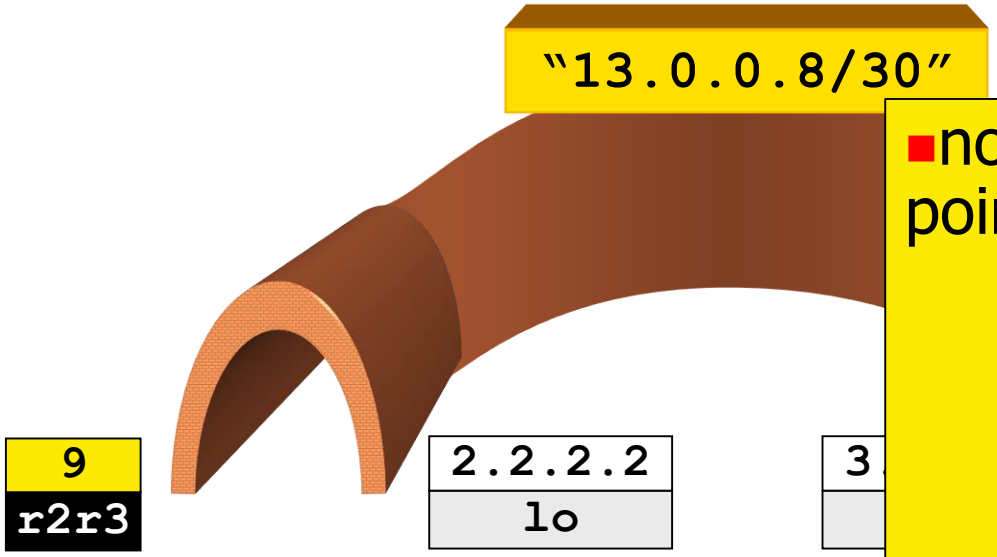


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
```

assign an ip address to the tunnel interface

configuring a tunnel



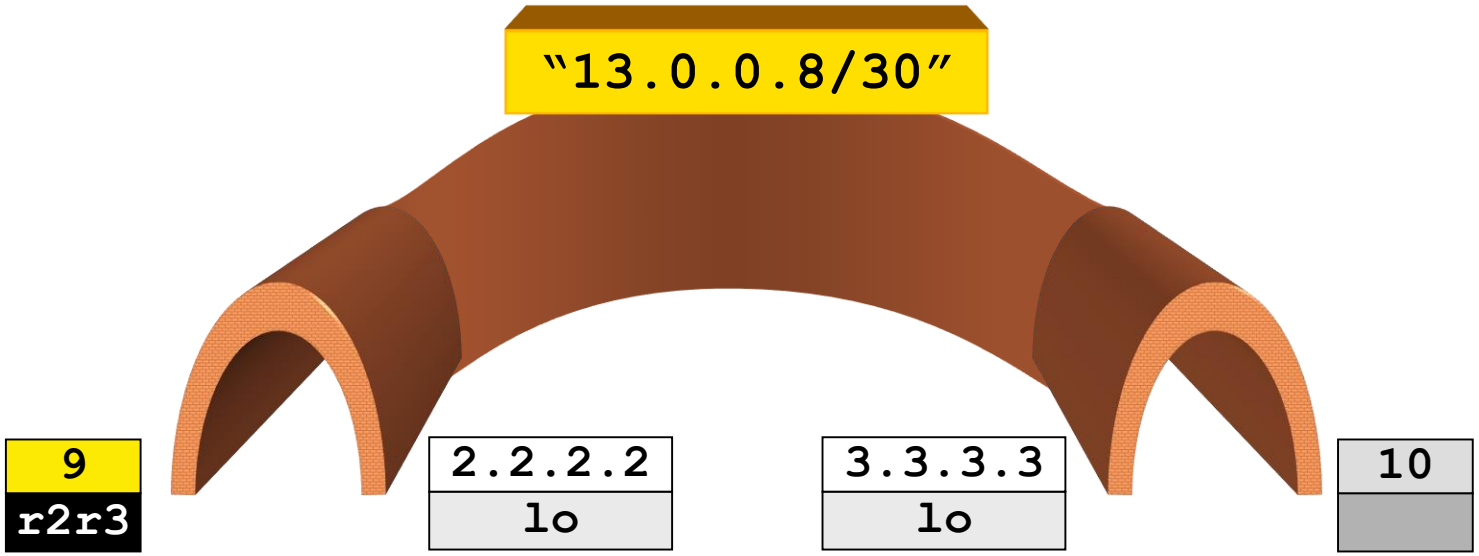
- note: this is a pointopoint interface
  - speaking of "network" is senseless
  - we do it nevertheless to simplify the graphical layout

as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
ip tunnel add r2r1 mode ipip remote 1.1.1.1 local 2.2.2.2 ttl 10
ip lin
ip add                                    13.0.0.5
ifconf
```

assign an ip address to the tunnel interface

configuring a tunnel

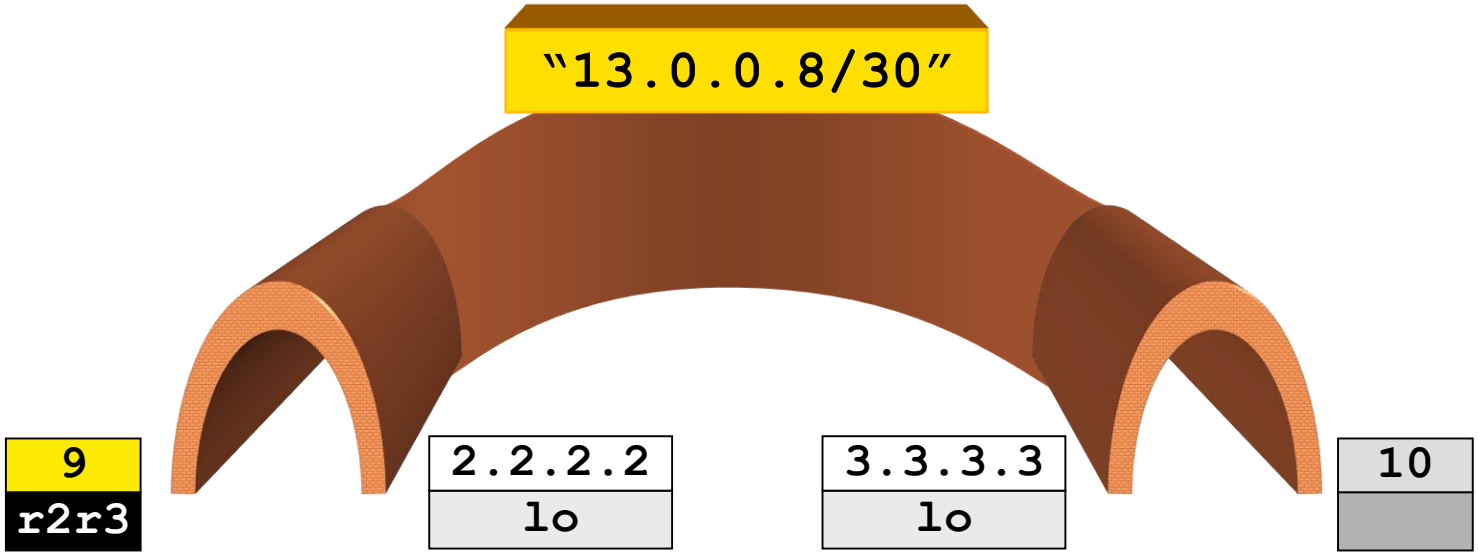


as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
ip tun                                  1.1.1 local 2.2.2.2 ttl 10
ip lin
ip add                                  0.0.5
ifconf
```

for a pointopoint interface we should set the address of the other endpoint

configuring a tunnel



as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
```

for a pointopoint interface we should set the address of the other endpoint

automatically inserts an entry in the routing table for `13.0.0.10`

configuring a tunnel

note: failure to set the peer's address prevents rip from recognizing packets coming from the tunnel (possibly because it cannot match the sender's address with any of the local interface's subnets)

```
2007/10/30 11:27:25 RIP: RECV packet from
   13.0.0.10 port 520 on unknown
2007/10/30 11:27:25 RIP: packe comes from
   unknown interface
```

9
r2r3

interface name expected here

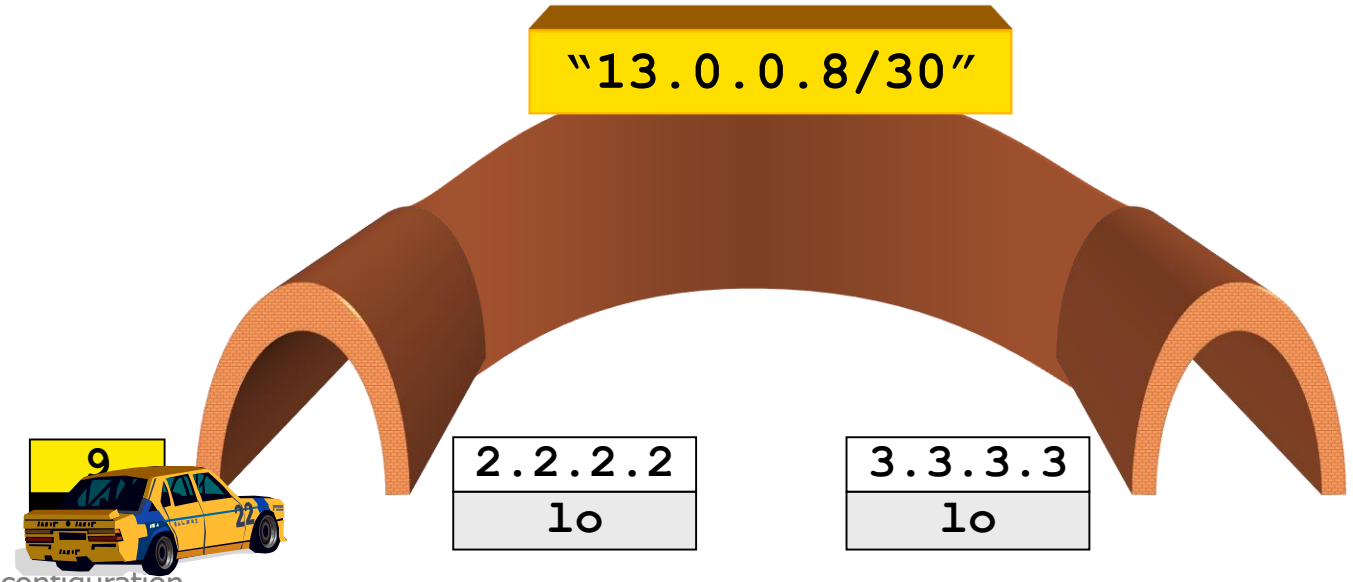
as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
```

for a pointopoint interface we should set the address of the other endpoint

automatically inserts an entry in the routing table for `13.0.0.10`

configuring a tunnel



as10rt2 configuration

```
ip tunnel add r2r3 mode ipip remote 3.3.3.3 local 2.2.2.2 ttl 10
ip link set r2r3 multicast on
ip addr add dev r2r3 13.0.0.9 peer 13.0.0.10
ifconfig r2r3 up
ip tunnel add r2r1 mode ipip remote 1.1.1.1 local 2.2.2.2 ttl 10
ip link set r2r1 mult
ip addr add dev r2r1 13
ifconfig r2r1 up
```

switch the tunnel interface on

tunnels and routing

as10rt2 ripd configuration

```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-list externalNetworks out r2r1
 distribute-list externalNetworks out r2r3
 distribute-list internalNetworks out eth1
 route 0.0.0.0/0
!
access-list externalNetworks permit 12.0.0.0/30
access-list externalNetworks deny any
access-list internalNetworks deny 13.0.0.0/24
access-list internalNetworks deny 12.0.0.0/24
access-list internalNetworks permit any
```

tunnels and routing

as10rt2 ripd configuration

```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-list externalNetworks out r2r1
 distribute-list externalNetworks out r2r3
 distribute-list internalNetworks out eth1
 route 0.0.0.0/0
!
access-list externalNetworks permit 12.0.0.0/30
access-list externalNetworks deny any
access-list internalNetworks deny 13.0.0.0/24
access-list internalNetworks deny 12.0.0.0/24
access-list internalNetworks permit any
```

tunnels and routing

as10rt2 ripd configuration

```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-list externalNe
 distribute-list externalNe
 distribute-list internalNet
 route 0.0.0.0/0
!
access-list externalNetworks permit 12.0.0.0/30
access-list externalNetworks deny any
access-list internalNetworks deny 13.0.0.0/24
access-list internalNetworks deny 12.0.0.0/24
access-list internalNetworks permit any
```

distribute also `lo:1`'s address
(but not `lo`'s address because it is within the reserved `127.0.0.0/8`)

tunnels and routing

as10rt2 ripd configuration

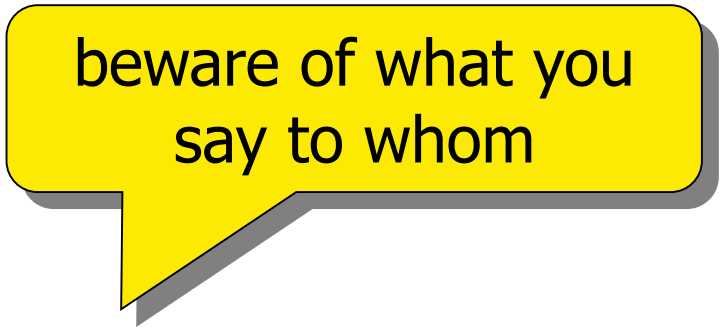
```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-list externalNe
 distribute-list externalNe
 distribute-list internal
 route 0.0.0.0/0
!
access-list externalNetworks
access-list externalNetworks deny any
access-list internalNetworks deny 13.0.0.0/24
access-list internalNetworks deny 12.0.0.0/24
access-list internalNetworks permit any
```

We must propagate a statically configured default route *inside* the transit as, because the routes learned via BGP are not redistribute inside.

tunnels and routing

as10rt2 ripd configuration

```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-list externalNetworks out r2r1
 distribute-list externalNetworks out r2r3
 distribute-list internalNetworks out eth1
 route 0.0.0.0/0
!
access-list externalNetworks permit 12.0.0.0/30
access-list externalNetworks deny any
access-list internalNetworks deny 13.0.0.0/24
access-list internalNetworks deny 12.0.0.0/24
access-list internalNetworks permit any
```



tunnels and routing

as10rt2 ripd configuration

```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-l
 distribute-l
 distribute-l
 route 0.0.0.0/0
!
access-list externalNetworks permit 12.0.0.0/30
access-list externalNetworks deny any
access-list internalNetworks deny 13.0.0.0/24
access-list internalNetworks deny 12.0.0.0/24
access-list internalNetworks permit any
```

ebgp next hops (in this case **as20r1**) are announced inside the tunnel

tunnels and routing

as10rt2 ripd configuration

```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-l
 distribute-l
 distribute-l
 route 0.0.0.
!
access-list ex                                  .0/30
access-list externa          orks deny any
access-list internal         works deny 13.0.0.0/24
access-list internalNetworks deny 12.0.0.0/24
access-list internalNetworks permit any
```

ebgp next hops and tunnel addresses (in this case **as20r1**) are *not* announced outside the tunnel

tunnels and routing

as10rt2 ripd configuration

```
router rip
 redistribute connected
 network eth1
 network r2r3
 network r2r1
 distribute-list externalNetworks out r2r1
 distribute-list externalNetworks out r2r3
 distribute-list internalNetworks out eth1
 route 0.0.0.0/0
!
access-list externalNetworks per
access-list externalNetworks den
access-list internalNetworks den
access-list internalNetworks den
access-list internalNetworks per
```

note: the same routing behavior could be obtained using static routes

tunnels and routing

- check the zebra routing table on **as10rt3**

as10rt2

```
Router> show ip route
Codes: …

R>* 1.1.1.1/32 [120/4] via 11.0.0.5, eth1, 00:08:43
C>* 2.2.2.2/32 is directly connected, lo
R>* 3.3.3.3/32 [120/4] via 11.0.0.5, eth1, 00:08
R>* 10.0.0.0/30 [120/2] via 11.0.0.5, eth1, 00:0
R>* 10.0.0.4/30 [120/2] via 11.0.0.5, eth1, 00:0
R>* 10.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:0
R>* 11.0.0.0/30 [120/3] via 11.0.0.5, eth1, 00:08:44
C>* 11.0.0.4/30 is directly connected, eth1
R>* 11.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:08:4
C>* 12.0.0.0/30 is directly connected, eth0
R>* 12.0.0.4/30 [120/2] via 13.0.0.10, r2r3, 00:08:10
B   12.0.0.4/30 [200/0] via 3.3.3.3 (recursive), 00:08:38
                        via 11.0.0.5, eth1, 00:08:38
B>* 12.0.0.8/30 [20/0] via 12.0.0.2, eth0, 00:08:36
B>  12.0.0.12/30 [200/0] via 12.0.0.6 (recursive), 00:07:45
  *                        via 13.0.0.10, r2r3, 00:07:45
C>* 13.0.0.5/32 is directly connected, r2r1
C>* 13.0.0.10/32 is directly connected, r2r3
B>  100.0.0.0/8 [200/0] via 12.0.0.6 (recursive), 00:07:45
  *                     via 13.0.0.10, r2r3, 00:07:45
C>* 127.0.0.0/8 is directly connected, lo
```

destinations routed through the tunnel

tunnels and routing

as10r6

```
as10r6-ripd> show ip rip
Codes: R - RIP, C - connected, O - OSPF, B - BGP
       (n) - normal, (s) - static, (d) - default, (r) - redistribute,
       (i) - interface

     Network            Next Hop         Metric From            Time
R(n) 0.0.0.0/0          11.0.0.10                1             8
R(n) 1.1.1.1/32         10.0.0.10                1             9
R(n) 2.2.2.2/32         10.0.0.5                 1             9
R(n) 3.3.3.3/32         11.0.0.10                1             8
R(n) 10.0.0.0/30        10.0.0.5                 1             9
C(i) 10.0.0.4/30        0.0.0.0                  0
C(i) 10.0.0.8/30        0.0.0.0                  0
R(n) 11.0.0.0/30        10.0.0.10                1             9
R(n) 11.0.0.4/30        10.0.0.5                 1             9
C(i) 11.0.0.8/30        0.0.0.0                  0
as10r6-ripd>
```

internal routers only know about internal destinations

(the default route is only there to offer Internet access, if required)

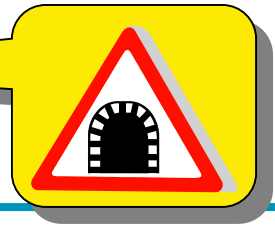
tunnels and routing

- **as10rt2** prefers the egress point **as10rt3**

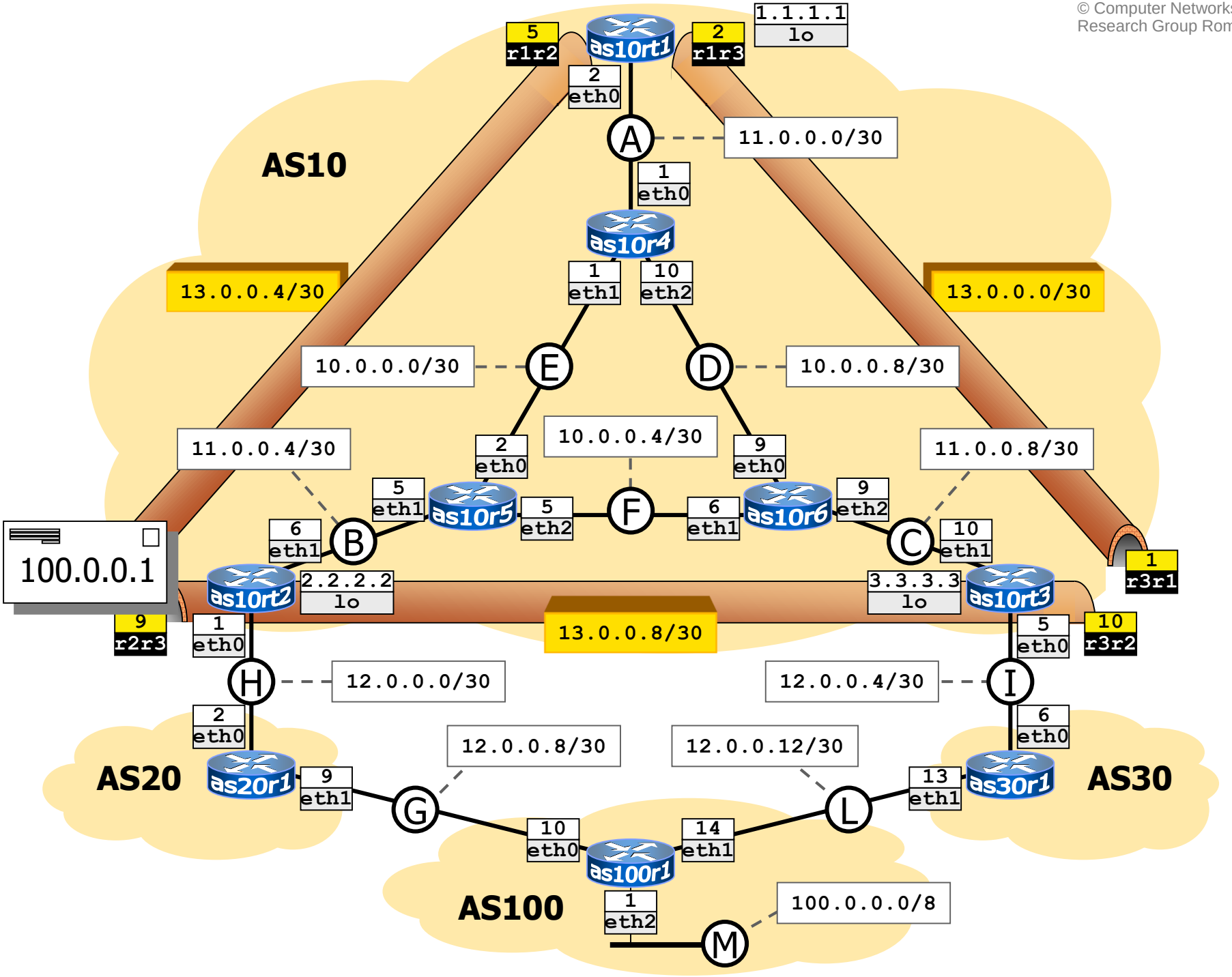
If we don't do this, the traceroute will fail because the ip 11.0.0.6 is choosen as source and it is not reachable outside the AS.

as10rt2

```
as10rt2:~# traceroute -s 12.0.0.1 100.0.0.1
traceroute to 100.0.0.1 (100.0.0.1) from 12.0.0.1, 64 hops max, 40
byte packets
 1  13.0.0.10 (13.0.0.10)  3 ms  3 ms  2 ms
 2  12.0.0.6 (12.0.0.6)  2 ms  4 ms  5 ms
 3  100.0.0.1 (100.0.0.1)  2 ms  2 ms  2 ms
```



- now **as10rt3** is directly reached via the tunnel



tunnels and routing

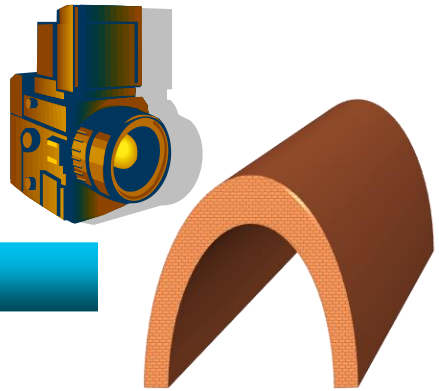
- check the zebra routing table on **as10rt3**

as10rt2

```
Router> show ip route
Codes:  K - kernel route, C - connected, S - static, R - RIP,
        O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel
 R>* 1.1.1.1/32 [120/4] via 11.0.0.5, eth1, 00:08:43
 C>* 2.2.2.2/32 is directly connected, lo
 R>* 3.3.3.3/32 [120/4] via 11.0.0.5, eth1, 00:08:43
 R>* 10.0.0.0/30 [120/2] via 11.0.0.5, eth1, 00:08:44
 R>* 10.0.0.4/30 [120/2] via 11.0.0.5, eth1, 00:08:44
 R>* 10.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:08:44
 R>* 11.0.0.0/30 [120/3] via 11.0.0.5, eth1, 00:08:44
 C>* 11.0.0.4/30 is directly connected, eth1
 R>* 11.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:08:44
 C>* 12.0.0.0/30 is directly connected, eth0
 R>* 12.0.0.4/30 [120/2] via 13.0.0.10, r2r3, 00:08:10
 B   12.0.0.4/30 [200/0] via 3.3.3.3 (recursive), 00:08:38
                         via 11.0.0.5, eth1, 00:08:38
 B>* 12.0.0.8/30 [20/0] via 12.0.0.2, eth0, 00:08:36
 B>  12.0.0.12/30 [200/0] via 12.0.0.6 (recursive), 00:07:45
   *                        via 13.0.0.10, r2r3, 00:07:45
 C>* 13.0.0.5/32 is directly connected, r2r1
 C>* 13.0.0.10/32 is directly connected, r2r3
 B>  100.0.0.0/8 [200/0] via 12.0.0.6 (recursive), 00:07:45
   *                       via 13.0.0.10, r2r3, 00:07:45
 C>* 127.0.0.0/8 is directly connected, lo
```

tunnels and routing

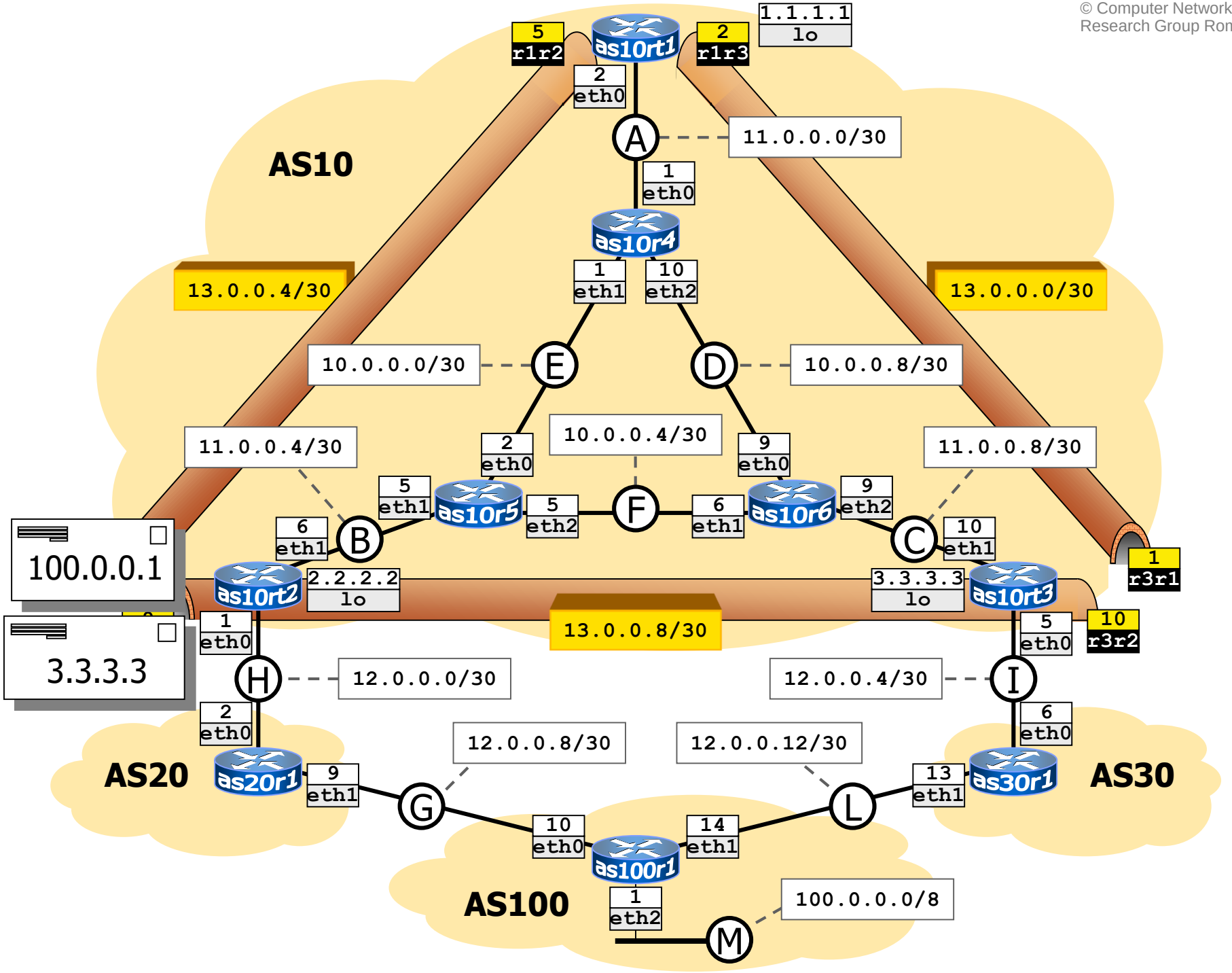
- a look outside the tunnel



as10rt2

```
as10rt2:~# ip tunnel show r2r3
r2r3: ip/ip  remote 3.3.3.3  local 2.2.2.2  ttl 10
as10rt2:~# ip link show eth0
1: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether fe:fd:0c:00:00:01 brd ff:ff:ff:ff:ff:ff
as10rt2:~# ip link show r2r3
7: r2r3@NONE: <POINTOPOINT,MULTICAST,NOARP,
    link/ipip 2.2.2.2 peer 3.3.3.3
as10rt2:~#
```

the tunnel is active



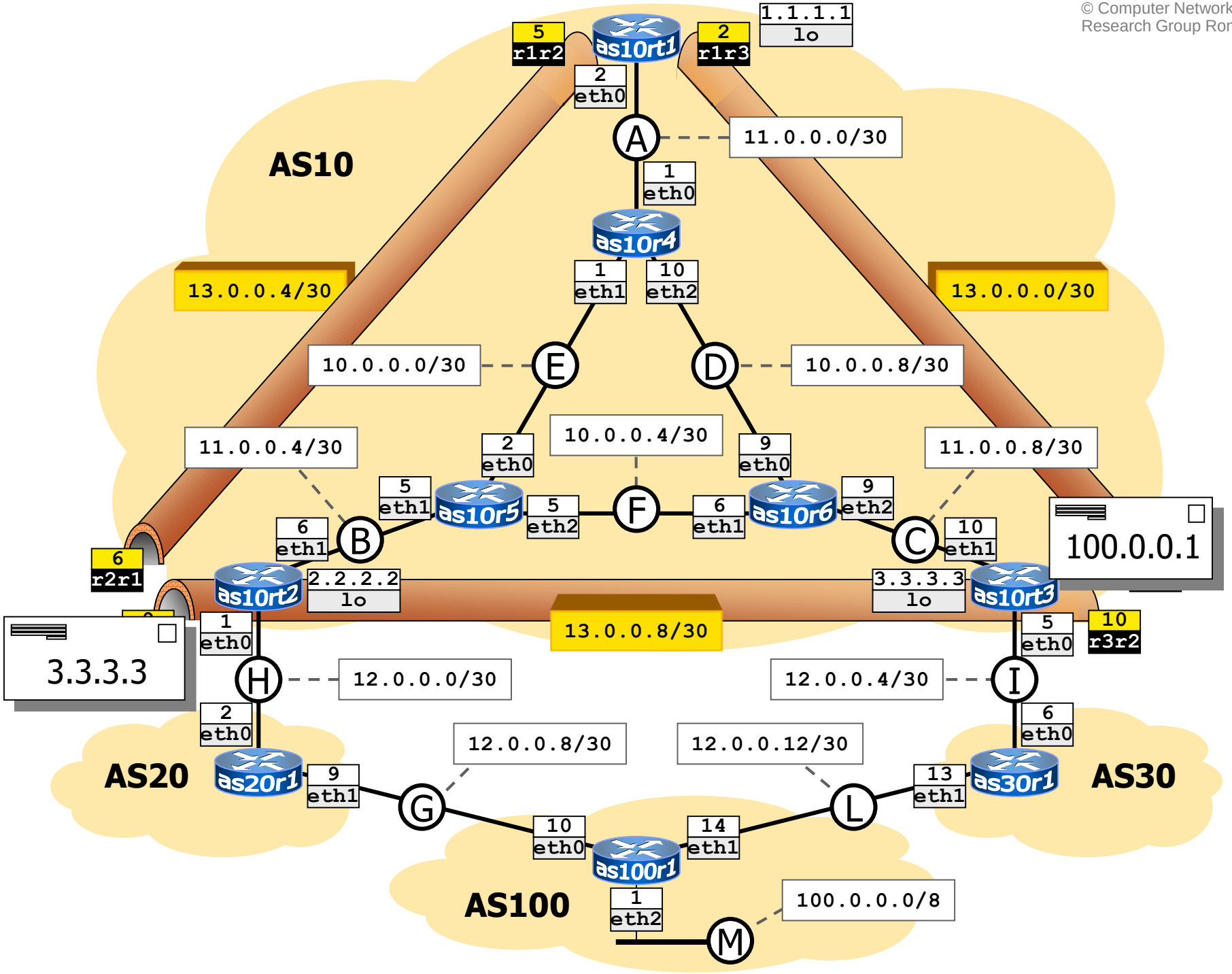
tunnels and routing

- check the zebra routing table on **as10rt3**

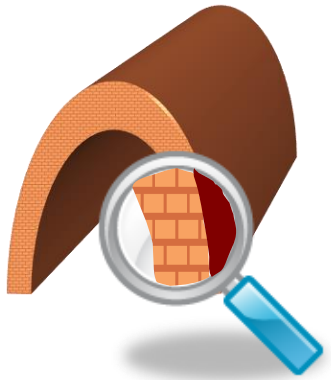
as10rt2

```
Router> show ip route
Codes: …

 R>* 1.1.1.1/32 [120/4] via 11.0.0.5, eth1, 00:08:43
 C>* 2.2.2.2/32 is directly connected, lo
 R>* 3.3.3.3/32 [120/4] via 11.0.0.5, eth1, 00:08:43
 R>* 10.0.0.0/30 [120/2] via 11.0.0.5, eth1, 00:08:44
 R>* 10.0.0.4/30 [120/2] via 11.0.0.5, eth1, 00:08:44
 R>* 10.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:08:44
 R>* 11.0.0.0/30 [120/3] via 11.0.0.5, eth1, 00:08:44
 C>* 11.0.0.4/30 is directly connected, eth1
 R>* 11.0.0.8/30 [120/3] via 11.0.0.5, eth1, 00:08:44
 C>* 12.0.0.0/30 is directly connected, eth0
 R>* 12.0.0.4/30 [120/2] via 13.0.0.10, r2r3, 00:08:10
 B   12.0.0.4/30 [200/0] via 3.3.3.3 (recursive), 00:08:38
                         via 11.0.0.5, eth1, 00:08:38
 B>* 12.0.0.8/30 [20/0] via 12.0.0.2, eth0, 00:08:36
 B>  12.0.0.12/30 [200/0] via 12.0.0.6 (recursive), 00:07:45
   *                       via 13.0.0.10, r2r3, 00:07:45
 C>* 13.0.0.5/32 is directly connected, r2r1
 C>* 13.0.0.10/32 is directly connected, r2r3
 B>  100.0.0.0/8 [200/0] via 12.0.0.6 (recursive), 00:07:45
   *                     via 13.0.0.10, r2r3, 00:07:45
 C>* 127.0.0.0/8 is directly connected, lo
```



tunnels and routing



- a look inside the tunnel

as10rt2

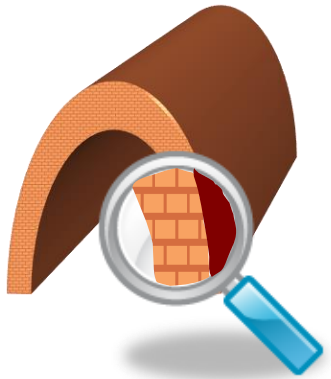
```
as10rt2:~# traceroute -s 12.0.0.1 100.0.0.1
traceroute to 100.0.0.1 (100.0.0.1) from 12.0.0.1,
         64 hops max, 40 byte packets
 1  13.0.0.10 (13.0.0.10)  3 ms  3 ms  2 ms
 2  12.0.0.6 (12.0.0.6)  2 ms  4 ms  5 ms
 3  100.0.0.1 (100.0.0.1)  2 ms  2 ms  2 ms
```

as10r5

```
as10r5:~# tcpdump -tenni eth1
tcpdump: verbose output suppressed, use -v or -
decode
listening on eth1, link-type EN10MB (Ethernet),
bytes
02:42:ac:eb:00:03 > 02:42:ac:eb:00:02, ethertype IPv4 (0x0800), length
94: 2.2.2.2 > 3.3.3.3: 12.0.0.1.55427 > 100.0.0.1.33434: UDP, length
32 (ipip-proto-4)
02:42:ac:eb:00:02 > 02:42:ac:eb:00:03, ethertype IPv4 (0x0800), length
122: 3.3.3.3 > 2.2.2.2: 13.0.0.10 > 12.0.0.1: ICMP time exceeded in-
transit, length 68 (ipip-proto-4)
```

inner ip addresses correspond to the real source and destination

tunnels and routing



- a look inside the tunnel

as10rt2

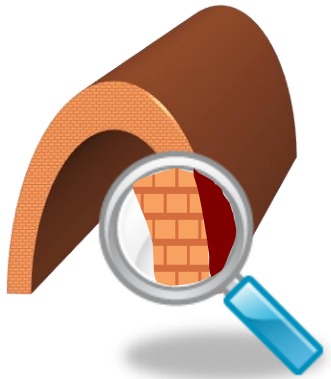
```
as10rt2:~# traceroute -s 12.0.0.1 100.0.0.1
traceroute to 100.0.0.1 (100.0.0.1) from 12.0.0.1,
         64 hops max, 40 byte packets
 1  13.0.0.10 (13.0.0.10)  3 ms  3 ms  2 ms
 2  12.0.0.6 (12.0.0.6)  2 ms  4 ms  5 ms
 3  100.0.0.1 (100.0.0.1)  2 ms  2 ms  2 ms
```

as10r5

```
as10r5:~# tcpdump -tenni eth1
tcpdump: verbose output suppressed, use -v or -vv f
decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144
bytes
02:42:ac:eb:00:03 > 02:42:ac:eb:00:02, ethertype IPv4 (0x0800), length
94: 2.2.2.2 > 3.3.3.3: 12.0.0.1.55427 > 100.0.0.1.33434: UDP, length
32 (ipip-proto-4)
02:42:ac:eb:00:02 > 02:42:ac:eb:00:03, ethertype IPv4 (0x0800), length
122: 3.3.3.3 > 2.2.2.2: 13.0.0.10 > 12.0.0.1: ICMP time exceeded in-
transit, length 68 (ipip-proto-4)
```

Packets are encapsulated

tunnels and routing



- a look inside the tunnel

as10rt2

```
as10rt2:~# traceroute -s 12.0.0.1 100.0.0.1
traceroute to 100.0.0.1 (100.0.0.1) from 12.0.0.1,
         64 hops max, 40 byte packets
 1  13.0.0.10 (13.0.0.10)  3 ms  3 ms  2 ms
 2  12.0.0.6 (12.0.0.6)  2 ms  4 ms  5 ms
 3  100.0.0.1 (100.0.0.1)  2 ms  2 ms  2 ms
```

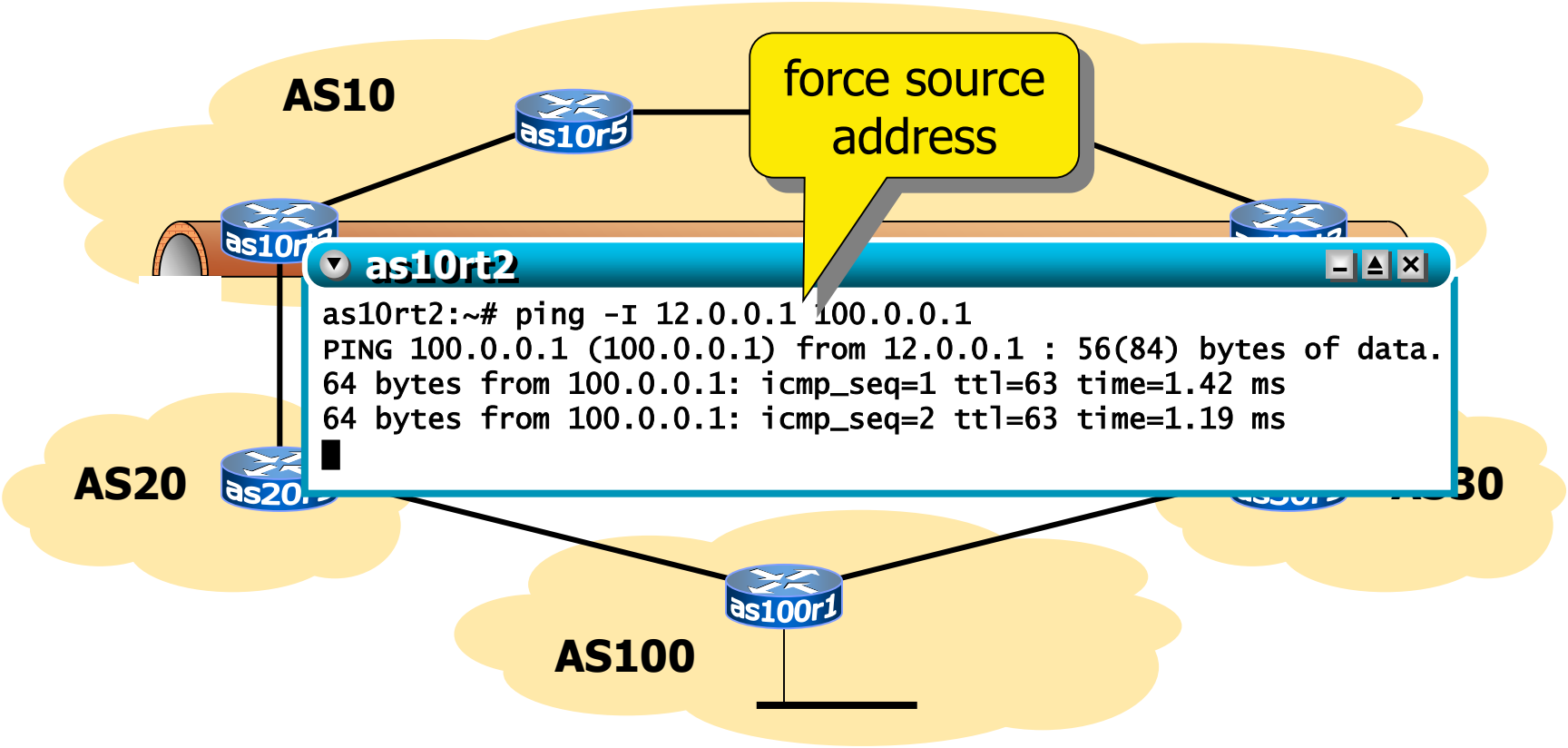
as10r5

```
as10r5:~# tcpdump -tenni eth1
tcpdump: verbose output suppressed, use -v or -
decode
listening on eth1, link-type EN10MB (Ethernet),
bytes
02:42:ac:eb:00:03 > 02:42:ac:eb:00:02, ethertype IPv4 (0x0800), length
94: 2.2.2.2 > 3.3.3.3: 12.0.0.1.55427 > 100.0.0.1.33434: UDP, length
32 (ipip-proto-4)
02:42:ac:eb:00:02 > 02:42:ac:eb:00:03, ethertype IPv4 (0x0800), length
122: 3.3.3.3 > 2.2.2.2: 13.0.0.10 > 12.0.0.1: ICMP time exceeded in-
transit, length 68 (ipip-proto-4)
```

outer ip addresses correspond to the tunnel endpoints

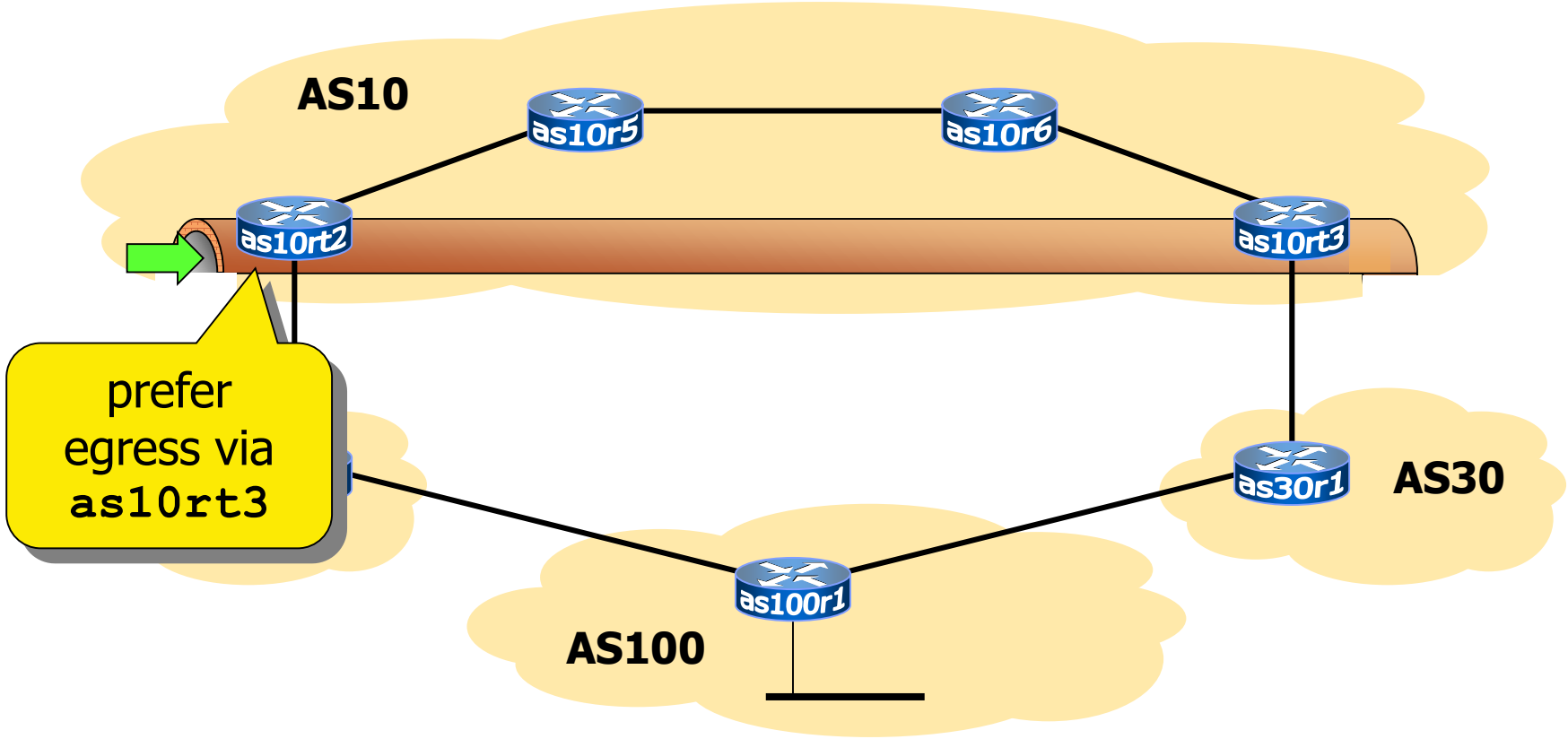
transit as: asymmetric routing

- bgp routing policies make routing asymmetric



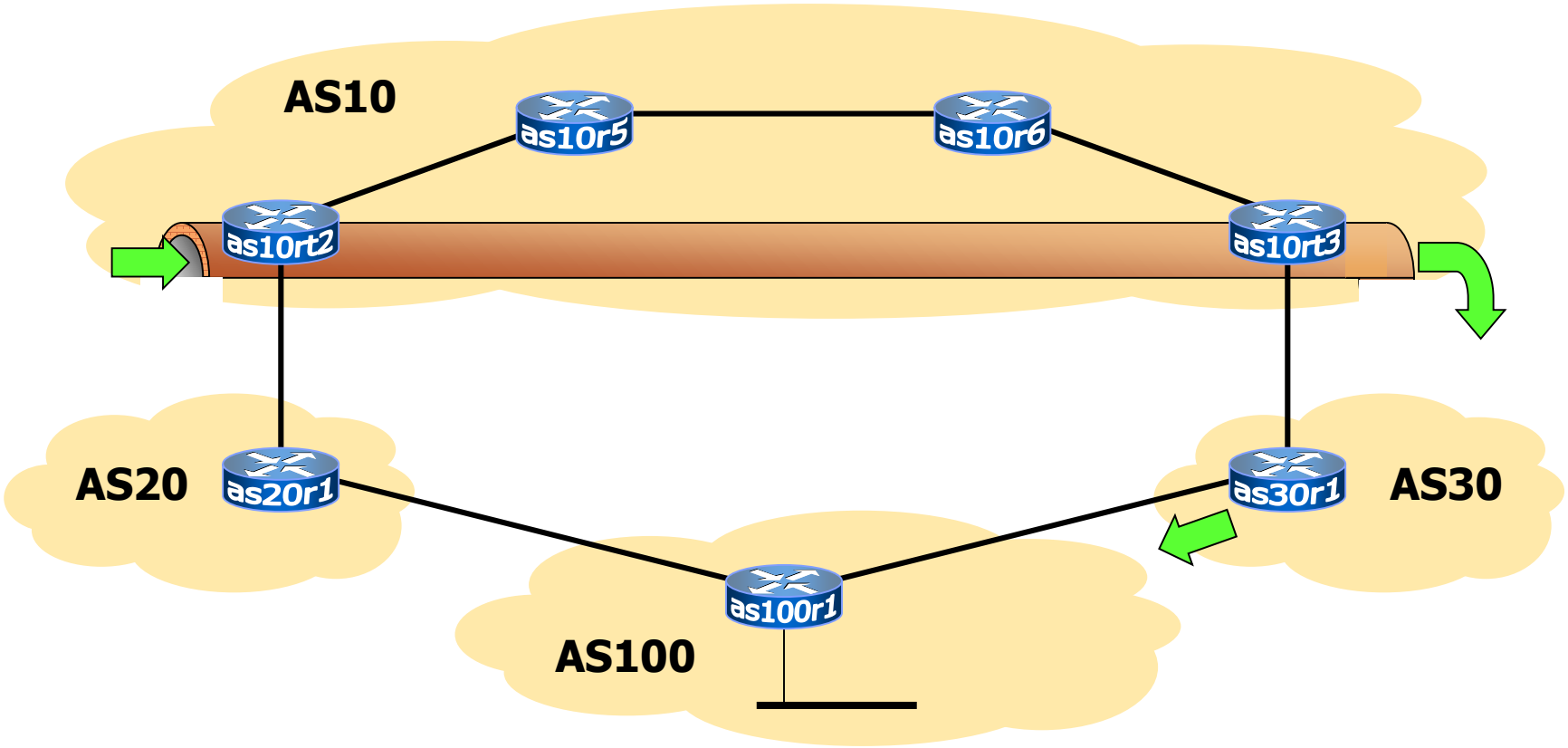
transit as: asymmetric routing

- bgp routing policies make routing asymmetric



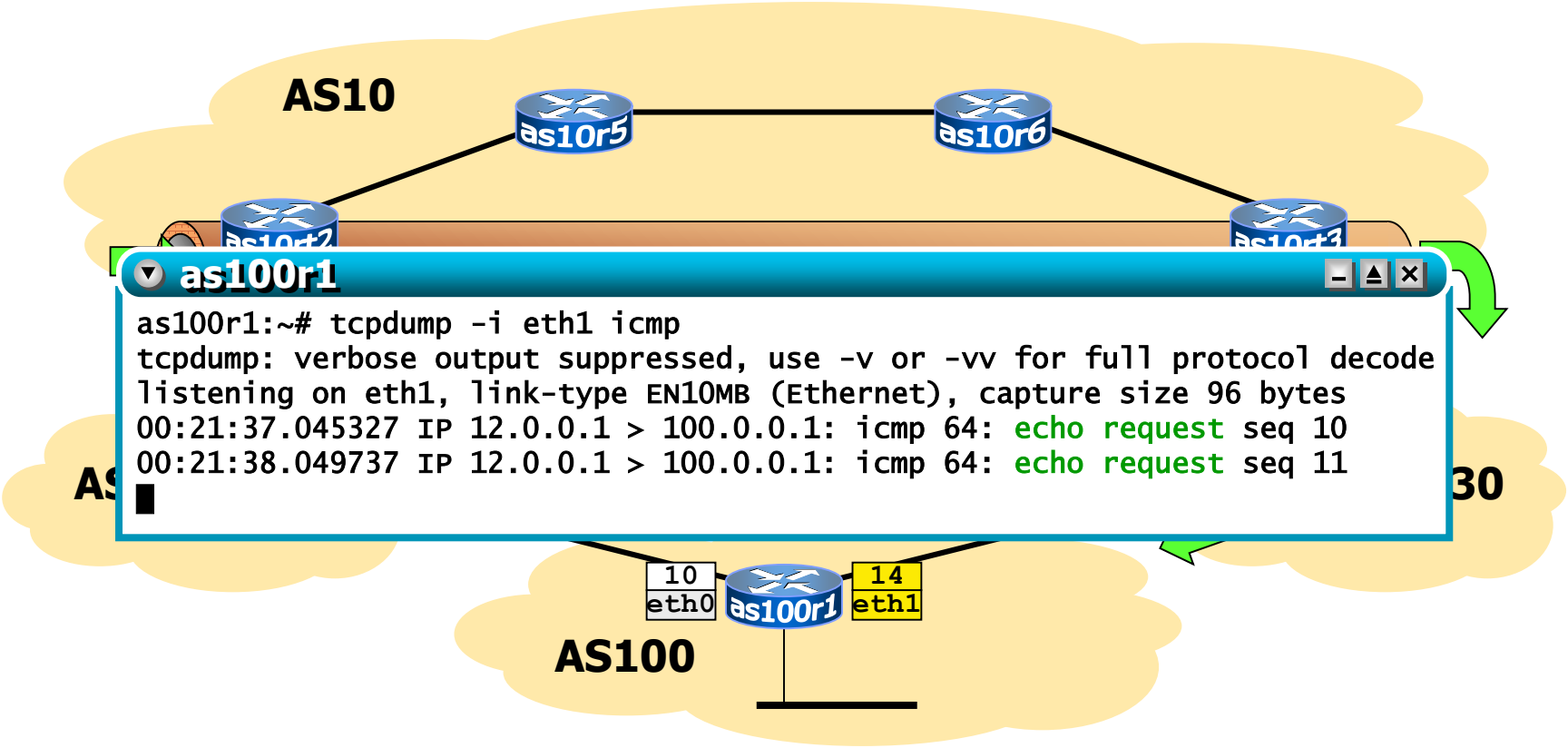
transit as: asymmetric routing

- bgp routing policies make routing asymmetric



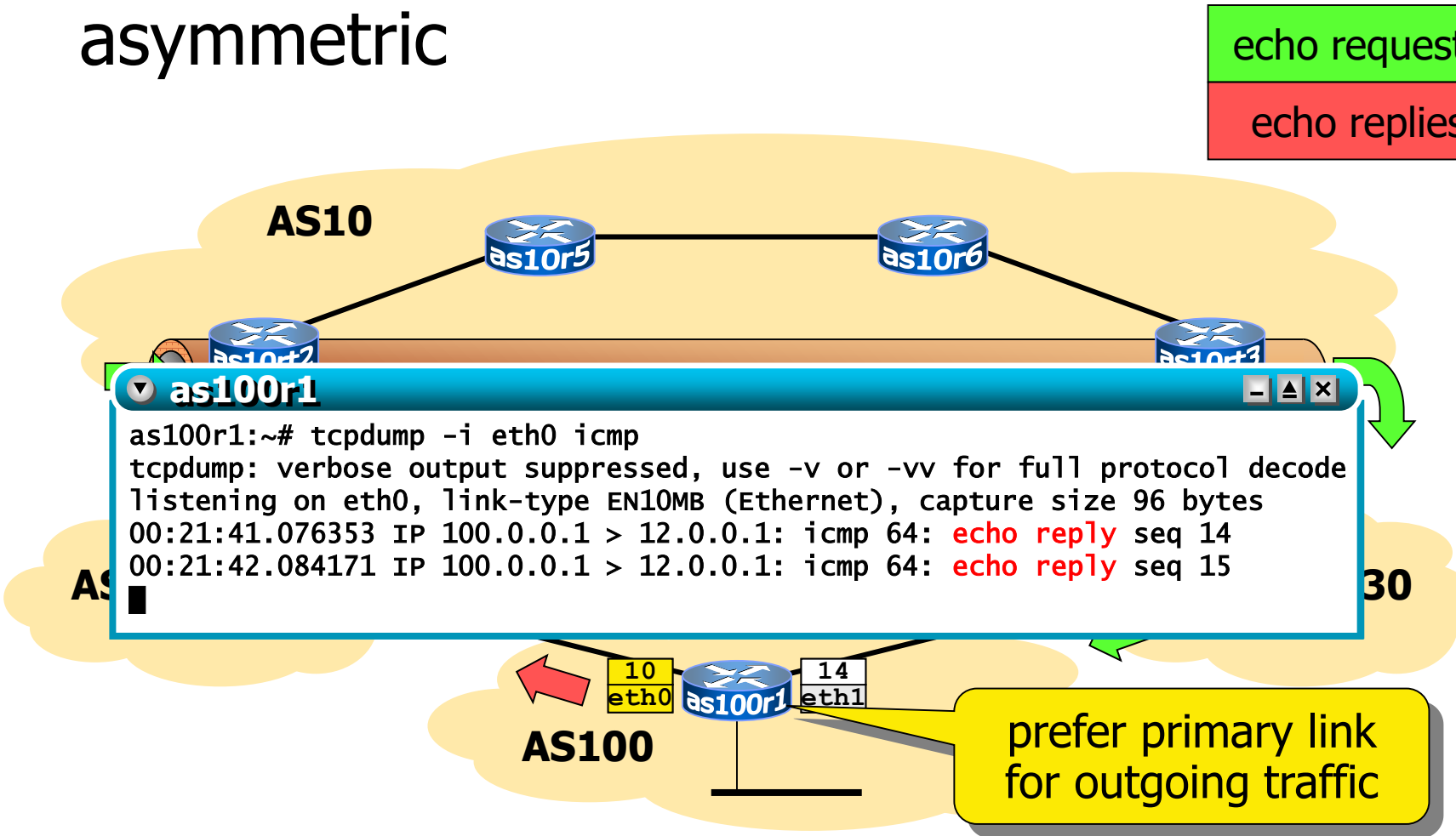
transit as: asymmetric routing

- bgp routing policies make routing asymmetric



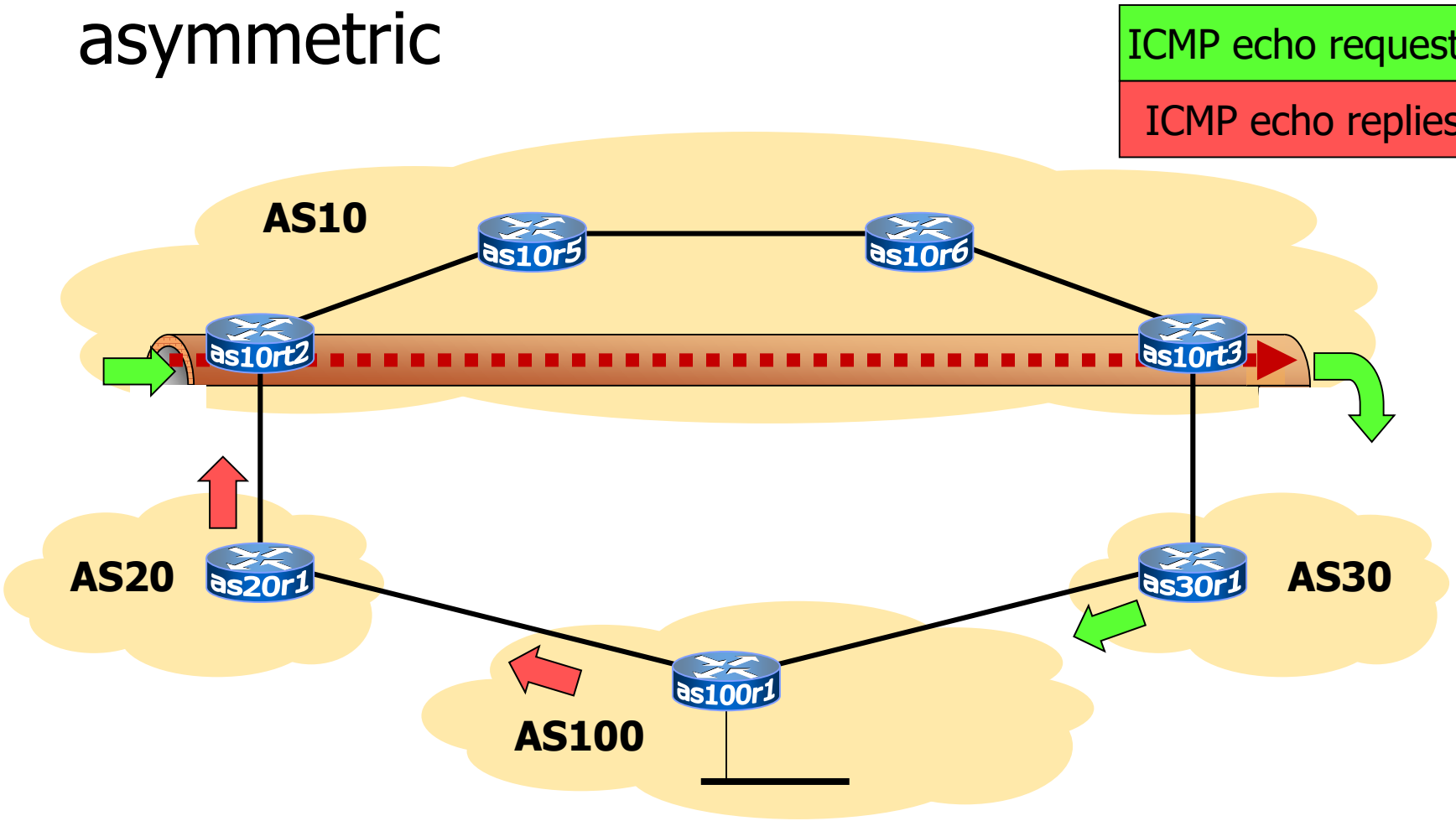
transit as: asymmetric routing

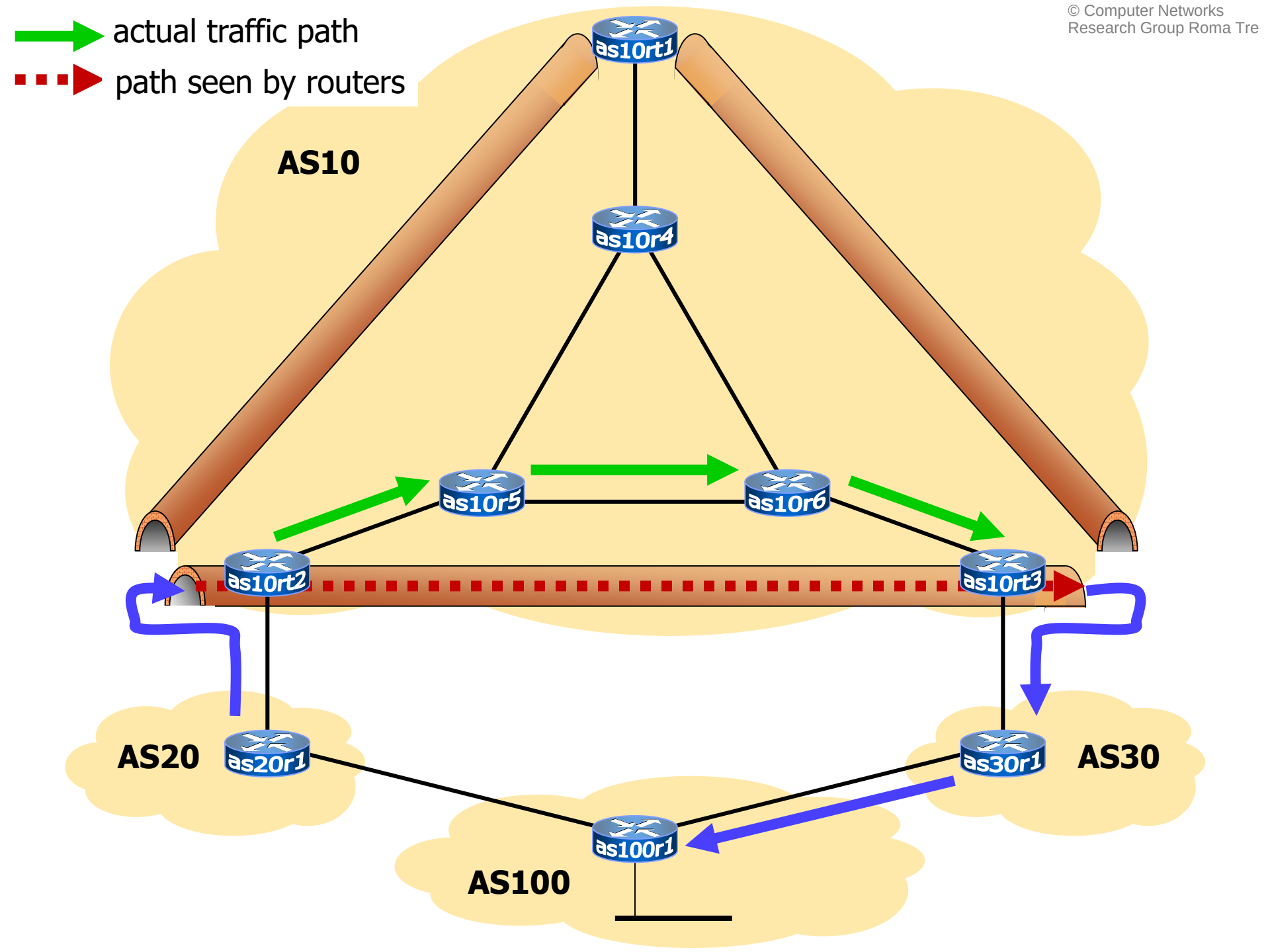
- bgp routing policies make routing asymmetric



transit as: asymmetric routing

- bgp routing policies make routing asymmetric





conclusions

- an overlay network is better
  - smaller routing tables on internal routers
  - less churn
  - predictable interplay between igp and egp
- sample implementation: tunnels
  - directed to the egress points
- observations
  - bgp peerings could be established on the tunnel interfaces
  - tunnels are as robust as the underlying igp