

kathara lab

bgp: stub-as with frr

Version	1.1
Author(s)	G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	architecture of a stub network; 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.

preconditions

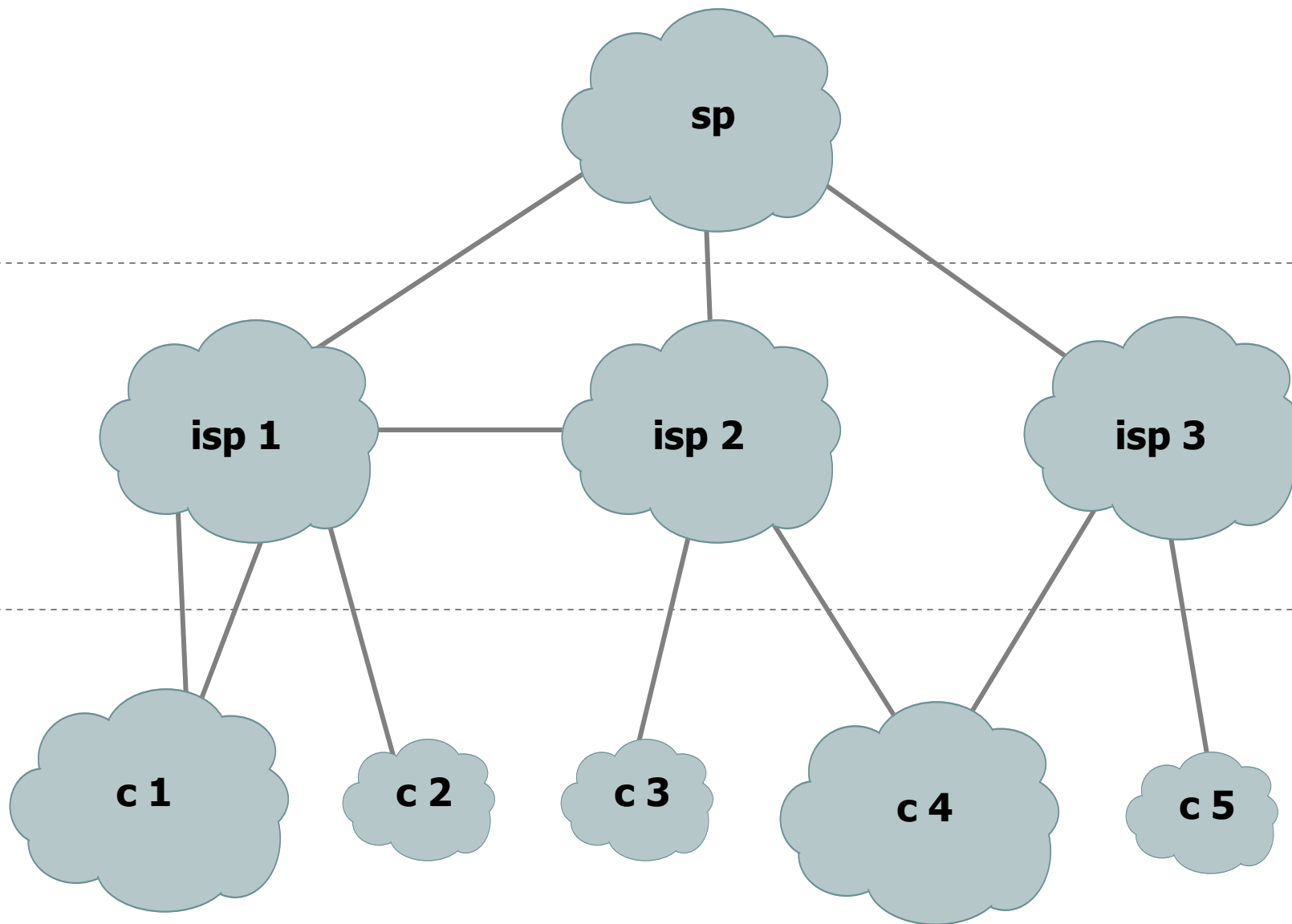
- for this lab we assume you have chosen “kathara/frr” as the default image of your Kathará installation
 - execute “kathara settings”
 - select “choose default image”
 - select “kathara/frr”
 - exit from the settings procedure

a small internet

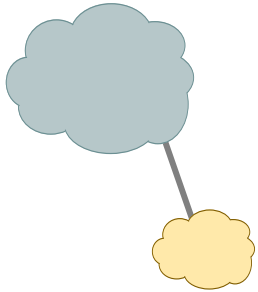
backbone

provider

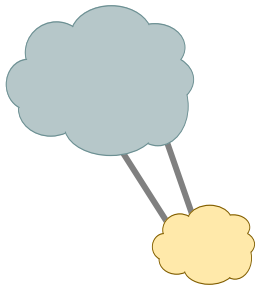
customer



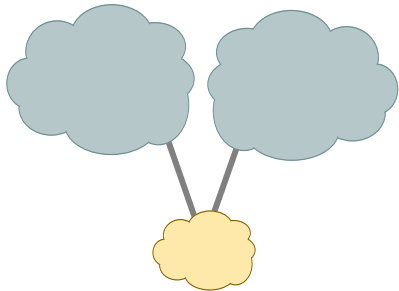
customer classification



- stub networks
 - one link to a single isp



- multi-homed stub network
 - two or more links to the same isp
 - purposes: backup or load sharing



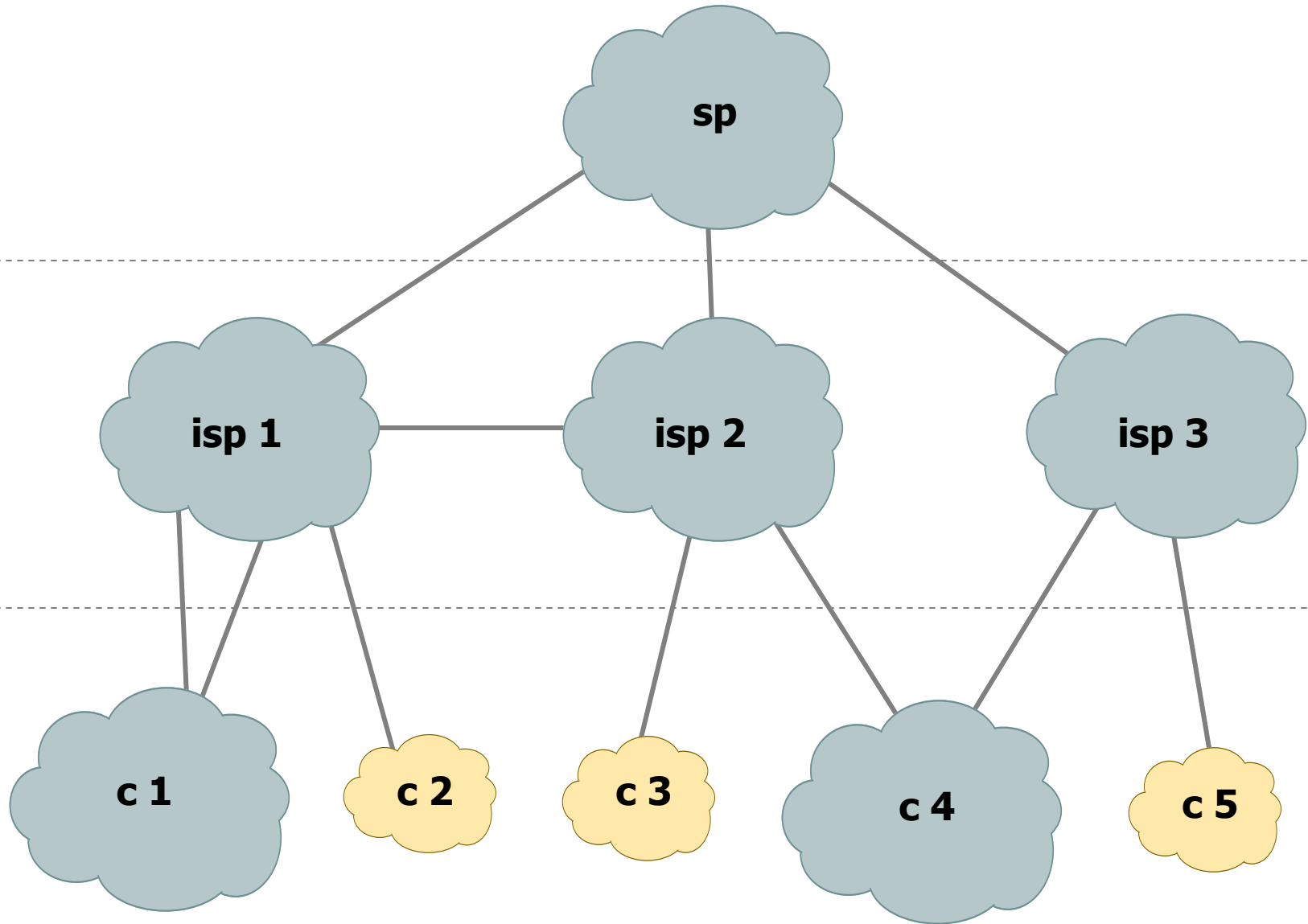
- multi-homed network
 - two or more links to different isps
 - purposes: backup or load sharing

stub networks

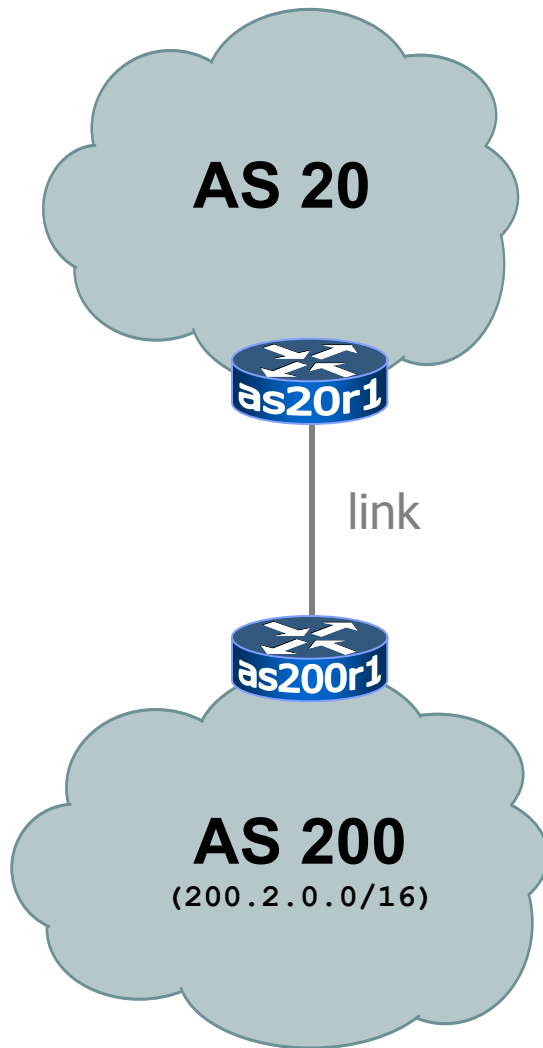
backbone

provider

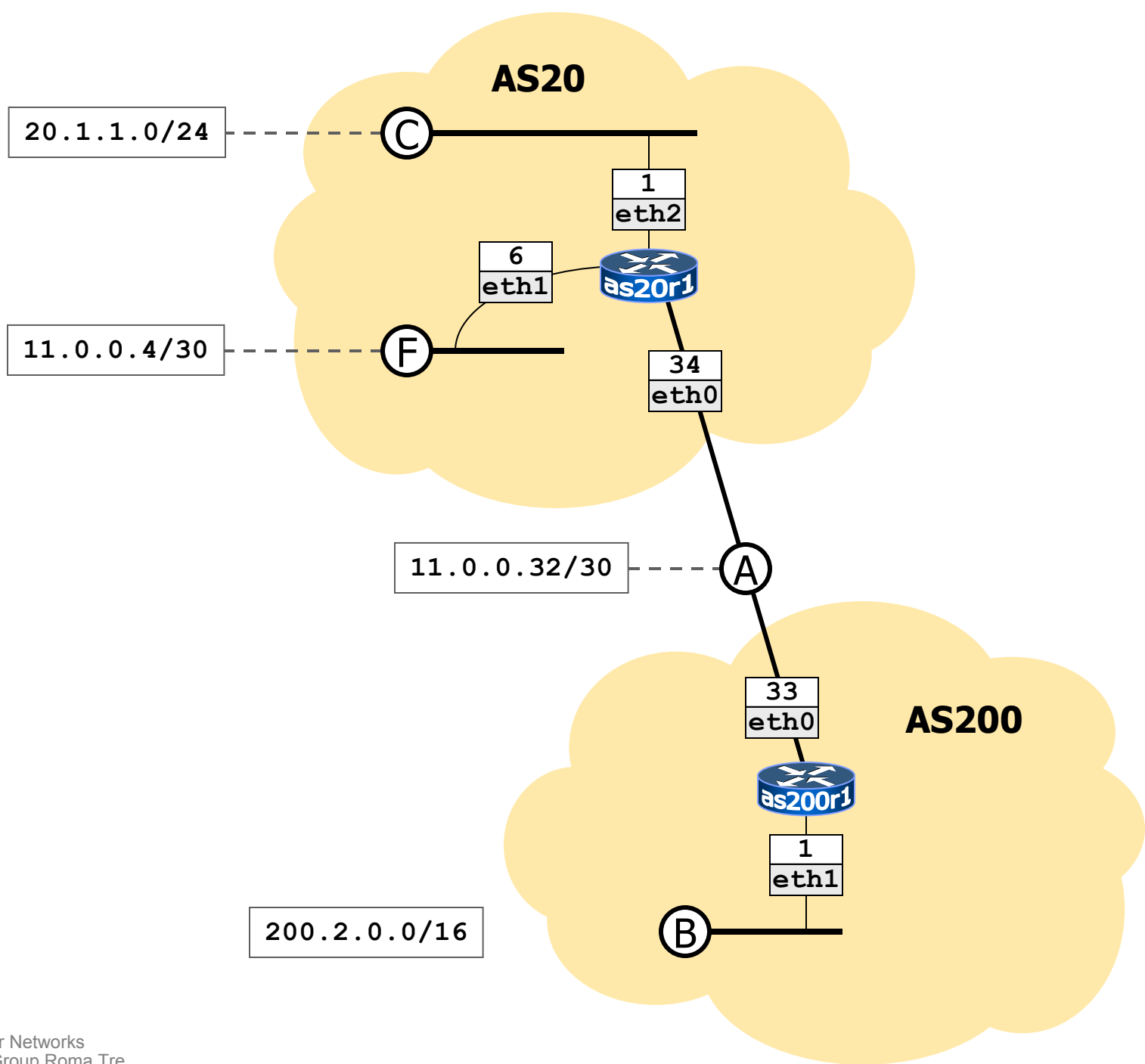
customer



stub network architecture



- one of the customer routers is chosen to be the default gateway
- the router is attached to a single router of the isp with a link (possibly more than one)
- a single peering in which as200 announces its route and accepts the default is enough



router as200r1 configuration



— frr configuration file —

```
! router as200r1 (customer side)
!
router bgp 200
no bgp ebgp-requires-policy
network 200.2.0.0/16
neighbor 11.0.0.34 remote-as 20
neighbor 11.0.0.34 description Router as20r1
```

router as20r1 configuration



— frr configuration file —

```
! router as20r1 (isp side)
router bgp 20
no bgp network import-check
neighbor 11.0.0.33 remote-as 200
neighbor 11.0.0.33 description Router as200r1
neighbor 11.0.0.33 default-originate
neighbor 11.0.0.33 prefix-list customerIn in
neighbor 11.0.0.33 prefix-list defaultOut out
!
network 20.1.1.0/24
network 0.0.0.0/0
!
ip prefix-list customerIn permit 200.2.0.0/16
ip prefix-list defaultOut permit 0.0.0.0/0
```

about **network 0.0.0.0/0**



- using **network 0.0.0.0/0** is enough to
 - place a default route in the local bgp routing table
 - as with any other prefix, the network command does not injected the default into the kernel
 - announce it
 - provided that you have it in the kernel or that you have a "**no bgp network import-check**" configuration
 - provided that you have set filters on the outgoing announces to the neighbors or that you have a "**no bgp ebgp-requires-policy**" configuration

about **network 0.0.0.0/0**



- it makes sense using **network 0.0.0.0/0** at the top of the isp hierarchy to originate the default route and to propagate it to all downstream providers
 - the default will be propagated with the full AS-path
- **network 0.0.0.0/0** should not be used at intermediate levels of the hierarchy
 - otherwise, routers would prefer the locally originated default route and remove the one offered by their upstream from the forwarding table

about default originate



- using **default-originate** for a specific neighbor
 - does not place a default route in the local bgp routing table
 - nor in the local kernel routing table
 - announces the default route to that neighbor
 - the prefix seems to be originated from the announcing AS as if it was declared with **network 0.0.0.0/0** even if it is not
 - **default-originate** overwrites the possible present **network 0.0.0.0/0** declaration

default-originate and route-maps



- a default route originated with **network 0.0.0.0/0** is handled like any other route
 - **route-maps** used with a specific neighbor are applied to the default route as well
- a default route originated with **default-originate** is processed by a different **route-map**:

—command syntax—

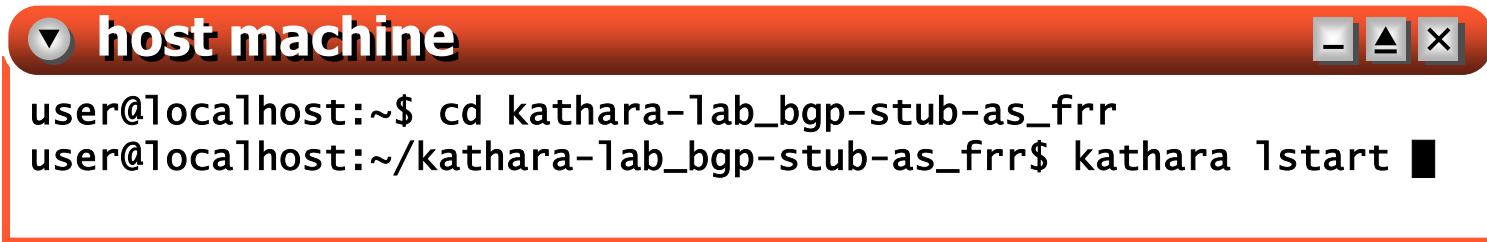
```
neighbor <neighbor-ip> default-originate route-map  
    <r-map-name> in
```

—command syntax—

```
neighbor <neighbor-ip> default-originate route-map  
    <r-map-name> out
```

stub as: lab

- start the lab



```
▼ host machine
user@localhost:~$ cd kathara-lab_bgp-stub-as_frr
user@localhost:~/kathara-lab_bgp-stub-as_frr$ kathara 1start
```

- check the frr configuration file



```
▼ as20r1
as20r1:~# less /etc/frr/frr.conf
```

- check the frr log file



```
▼ as20r1
as20r1:~# less /var/log/frr/frr.log
```

stub as: lab

- check the routing table of as20r1

```
as20r1
root@as20r1:/# vtysh

Hello, this is FRRouting (version 7.5.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

as20r1-frr# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

C>* 11.0.0.4/30 is directly connected, eth1, 00:14:46
C>* 11.0.0.32/30 is directly connected, eth0, 00:14:46
C>* 20.1.1.0/24 is directly connected, eth2, 00:14:46
B>* 200.2.0.0/16 [20/0] via 11.0.0.33, eth0, weight 1, 00:14:43
as20r1-frr#
```


stub as: lab

■ check the bgp status

▼ as20r1



```
as20r1-frr# show ip bgp neighbors
BGP neighbor is 11.0.0.33, remote AS 200, local AS 20, external link
  Description: Router as200r1
  Hostname: as200r1
  . . . . .
as20r1-frr# show ip bgp
BGP table version is 3, local router ID is 20.1.1.1, vrf id 0
Default local pref 100, local AS 20
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
                i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	0.0.0.0/0	0.0.0.0	0		32768	i
*>	20.1.1.0/24	0.0.0.0	0		32768	i
*>	200.2.0.0/16	11.0.0.33	0		0	200 i

```
Displayed 3 routes and 3 total paths
as20r1-frr#
```

stub as: lab

- perform several pings on the routers
- terminate the lab

▼ **host machine**

```
user@localhost:~/kathara-lab_bgp-stub-as_frr$ kathara 1clean █
```