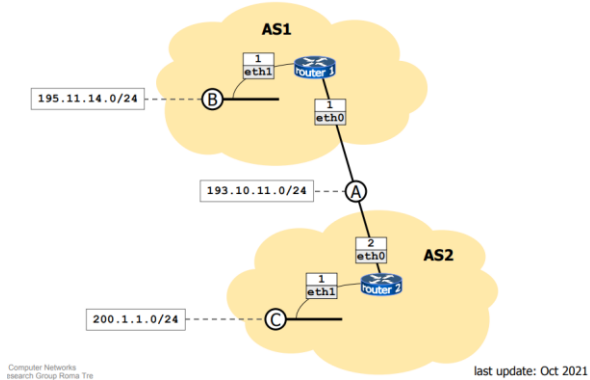


## BGP-SIMPLE PEERING

I router mettono in piedi la connessione, ma non hanno nulla da dirsi. Si informano della presenza ma non si dicono nulla di ciò che conoscono.

### a simple peering



- 1- **lab.conf** -> uguale a ospf e rip
- 2- **r1.startup** -> ip address add 193.10.11.1/24 dev eth0, collego indirizzi IP delle varie lan
- 3- **daemons** -> avrò acceso il demone di bgpd
- 4- **ip route** su router 1 -> non vede la rete 200.1.1.0/24 perché nessuno gliene ha parlato. Hanno solo instaurato la connessione

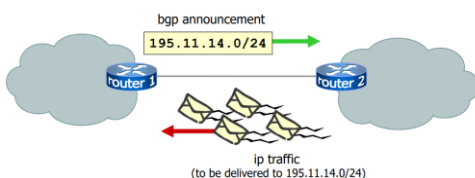
```
root@router1:/# ip route
193.10.11.0/24 dev eth0 proto kernel scope link src 193.10.11.1
195.11.14.0/24 dev eth1 proto kernel scope link src 195.11.14.1
```

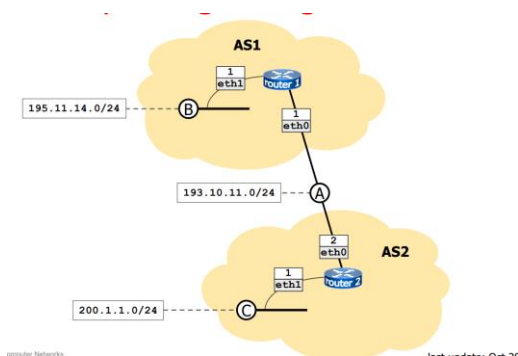
- 5- **sh ip bgp neighbors** -> mi fa vedere i vicini, dal punto di vista di bgp

## BGP-ANNOUNCING PREFIXES

Ogni router annuncia un prefisso, dicendo che tramite lui si può mandare traffico per raggiungere quel prefisso. Se non voglio mandare traffico tra due router non mando l'annuncio. Se colui che riceve l'annuncio non vuole rispondere, butta l'annuncio.

Il traffico fa il percorso inverso dell'annuncio.





## FRR CONF (router 1)

- 1- comando **network** -> per definire un prefisso da annunciare. Sto dicendo ai miei vicini che ho quella rotta. Se non lo metto bgp non funziona
- 2- **no bgp network import check** -> manda annunci anche se il prefisso non sta nella tabella di instradamento. Non serve quando il prefisso che devo annunciare è direttamente connesso in quel router e quindi sta nella tabella di instradamento. Se non voglio metterlo lo commento con **!**
- 3- **no bgp ebgp-requires policy** -> mi serve quando non ho dei filtri. Per filtri si intende che voglio far passare solo alcuni prefissi e non tutti. Non mi serve quando ho filtri. Se non voglio metterlo lo commento con **!**

```
nt\router1\etc\frr> cat frr.conf
!
! FRRouting configuration file
!
password zebra
enable password zebra
!
log file /var/log/frr/frr.log
!
! BGP CONFIGURATION
!
debug bgp keepalives
debug bgp updates in
debug bgp updates out
!
router bgp 1
no bgp ebgp-requires-policy
! no bgp network import-check (not needed for this lab)
neighbor 193.10.11.2 remote-as 2
neighbor 193.10.11.2 description Router 2 of AS2
!
network 195.11.14.0/24
```

- 4- **vttysh** -> **sh ip route** -> vedo non solo le rotte direttamente connesse, ma mi dice anche che tramite bgp ho appreso 200.1.1.0/24 con n.h 193.10.11.2 eth0

```
router1-frr# sh 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, F - PBR,
f - OpenFabric,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure

C>* 193.10.11.0/24 is directly connected, eth0, 00:00:11
C>* 195.11.14.0/24 is directly connected, eth1, 00:00:11
B>* 200.1.1.0/24 [20/0] via 193.10.11.2, eth0, weight 1, 00:00:09
```

- 5- **sh ip bgp summary** -> vedo lo stato della configurazione e le varie informazioni. Il peering è attivo e si scambiano messaggi

```
router1-frr# sh ip bgp summary

IPv4 Unicast Summary (VRF default):
BGP router identifier 195.11.14.1, local AS number 1 vrf-id 0
BGP table version 2
RIB entries 3, using 576 bytes of memory
Peers 1, using 20 KiB of memory

Neighbor    V    AS  MsgRcvd  MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd  PfxSnt Desc
193.10.11.2  4      2         6         6         2    0    0 00:01:06      1         2 Router 2 of AS2

Total number of neighbors 1
```

6- **sh ip bgp** -> mi dice cosa vede bgp, con alcuni attributi (metric, local pref, path)

metric: più è alta e più sfavorisco un link, annunci in uscita

local pref: più è alta e più favorisco, annunci in ingresso

path: registra i vari AS attraversati per raggiungere una rotta

```
router1-frr# sh ip bgp
BGP table version is 2, local router ID is 195.11.14.1, vrf id 0
Default local pref 100, local AS 1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Next hop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*> 195.11.14.0/24  0.0.0.0              0         32768 i
*> 200.1.1.0/24    193.10.11.2          0          0 2 i

Displayed 2 routes and 2 total paths
```

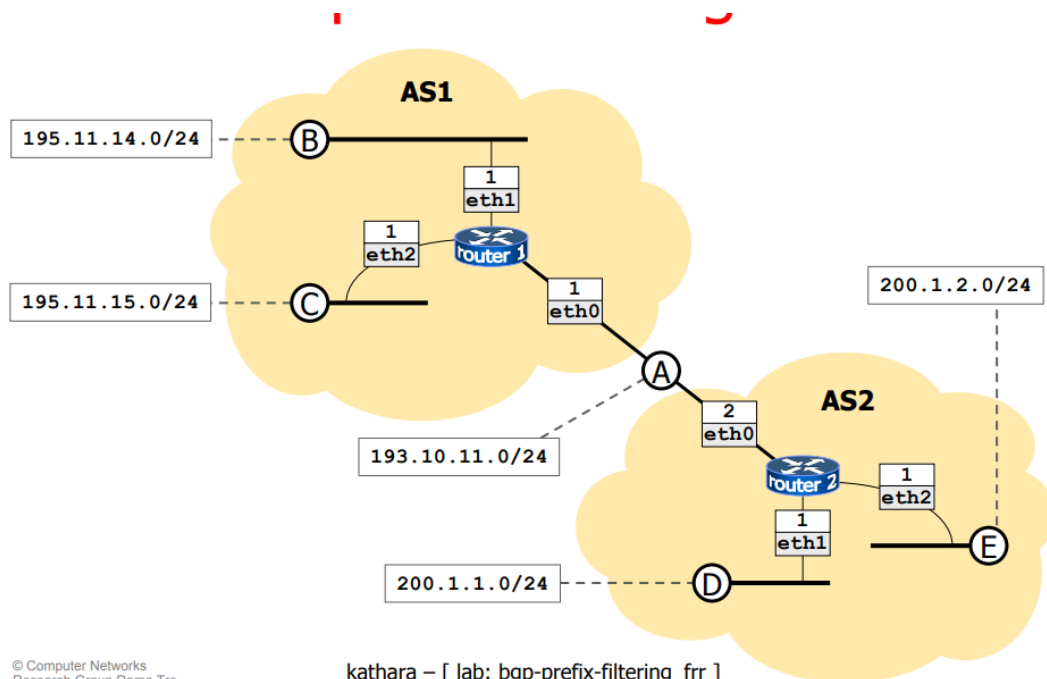
## BGP-PREFIX FILTERING

Decido di non voler annunciare tutto ma solamente alcuni, metto quindi dei filtri utilizzando due strumenti:

- prefix list: serve per filtrare prefissi, per questo specifico vicino non annunciare una specifica in uscita
- filter list: se negli annunci è contenuto questo AS number scarta l'annuncio

Posso anche modificare un annuncio con i suoi attributi in due modi:

- route map: ne definisco una per gli annunci in uscita (out-metric) e una per gli annunci in entrata (in-local pref)
- access list



## FRR CONF (router 1)

```
! BGP CONFIGURATION
!
debug bgp keepalives
debug bgp updates in
debug bgp updates out
!
router bgp 1
! no bgp ebgp-requires-policy (we have policy now!)
! no bgp network import-check (not needed for this lab)
neighbor 193.10.11.2 remote-as 2
neighbor 193.10.11.2 description Router 2 of AS2
neighbor 193.10.11.2 prefix-list partialOut out
neighbor 193.10.11.2 prefix-list partialIn in
!
network 195.11.14.0/24
network 195.11.15.0/24
!
!
ip prefix-list partialOut permit 195.11.14.0/24
!
ip prefix-list partialIn deny 200.1.1.0/24
ip prefix-list partialIn permit any
```

!no bgp ebgp-requires policies (lo commento perchè ho filtri)

!no bgp network import check (lo commento perchè ciò che annuncio è direttamente connesso quindi lo ho in tabella)

- 1- **neighbor 193.10.11.2 prefix-list partialOut out** -> sto definendo una prefix list nei confronti di 193.10.11.2 dicendo di annunciargli solamente 195.11.14.0/24  
**ip prefix-list partialOut permit 195.11.14.0/24**
- 2- **neighbor 193.10.11.2 prefix-list partialIn in** -> sto definendo una prefix list nei confronti di 193.10.11.2 dicendo di accettare tutto tranne 200.1.1.0/24.

**L'ORDINE CONTA, IO LEGGO IL PRIMO STATEMENT. SE METTO DENY ANY (DEFAULT) PER PRIMO BUTTA TUTTO.**

## FRR CONF (router 2)

```
! BGP CONFIGURATION
!
debug bgp keepalives
debug bgp updates in
debug bgp updates out
!
router bgp 2
no bgp ebgp-requires-policy
! no bgp network import-check (not needed for this lab)
neighbor 193.10.11.1 remote-as 1
neighbor 193.10.11.1 description Router 1 of AS1
!
network 200.1.1.0/24
network 200.1.2.0/24
```

no bgp ebgp-requires policies (lo metto perchè non ho filtri)

!no bgp network import check (lo commento perchè ciò che annuncio è direttamente connesso quindi lo ho in tabella)

- 3- **ip route** su router 1 -> vedo le due direttamente connesse, lan di peering e una sola di quelle di AS2 perchè una l'ho negata.

```
root@router1:~# ip route
193.10.11.0/24 dev eth0 proto kernel scope link src 193.10.11.1
195.11.14.0/24 dev eth1 proto kernel scope link src 195.11.14.1
195.11.15.0/24 dev eth2 proto kernel scope link src 195.11.15.1
200.1.2.0/24 nhid 8 via 193.10.11.2 dev eth0 proto bgp metric 20
```

- 4- **sh ip bgp** -> cosa vede bgp, ne vedo solo tre per via dei filtri (quelle dir. connesse le vedo sempre)

```
root@router1:~# vtysh
Hello, this is FRRouting (version 9.0.1).
Copyright 1996-2005 Kunihiko Ishiguro, et al.

router1-frr# sh ip bgp
BGP table version is 3, local router ID is 195.11.15.1, vrf id 0
Default local pref 100, local AS 1
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Next hop codes: 0NNN next hop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*> 195.11.14.0/24  0.0.0.0           0             32768 i
*> 195.11.15.0/24  0.0.0.0           0             32768 i
*> 200.1.2.0/24   193.10.11.2       0              0 2 i

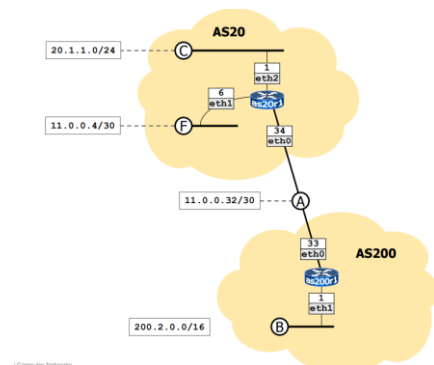
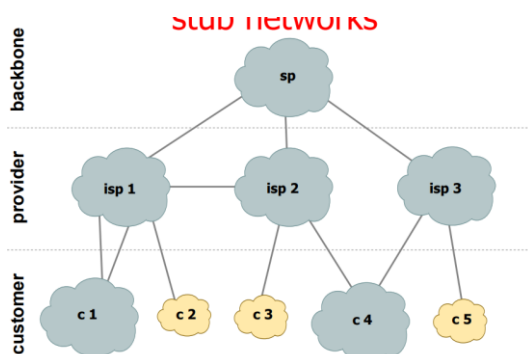
Displayed 3 routes and 3 total paths
```

## BGP ATTRIBUTES

- **prefix e as path**: ogni volta che annuncio un prefisso ad un differente AS tramite e-bgp, l'AS number viene inserito in testa all'attributo. Il router quindi sa che percorso ha fatto l'annuncio. Ha due ruoli: evitare i loop (se un AS riceve lo stesso annuncio in cui c'era già il suo numero lo scarta), applicare una metrica per definire il cammino più breve (se un AS riceve due annunci, sceglie quello con as path più breve)
- **next hop**: indica il router che manda l'annuncio.  
Nel momento in cui i peering avvengono su una LAN, il next hop rimane quello di partenza, perché non ha senso cambiarlo se tanto poi il traffico arriverà lì.  
I router interni fanno "recursive lookup" ovvero verificano se il prefisso dell'annuncio che ricevono via bgp è raggiungibile. Se non è raggiungibile, l'annuncio viene rifiutato. In quel caso il next hop cambia solo quando l'annuncio va verso un diverso AS.
- **origin**: dice da dove parte l'annuncio
- **aggregator**: mi dice quale AS ha costruito l'annuncio e quale router ha inserito quel prefisso nella rete

## BGP-STUB AS

Consideriamo il caso in cui un customer ha un unico link verso un isp. Si creerà quindi un singolo peering dove il customer annuncia la rotta e accetta la default dall'isp. Bisogna quindi generare dei filtri per inviare e trasmettere solo alcuni prefissi.



### FRR CONF (as200r1)

```
! FRRouting configuration file
!
password zebra
enable password zebra
!
log file /var/log/frr/frr.log
!
! BGP CONFIGURATION
!
debug bgp keepalives
debug bgp updates in
debug bgp updates out
!
router bgp 200
no bgp ebgp-requires-policy
! no bgp network import-check
neighbor 11.0.0.34 remote-as 20
neighbor 11.0.0.34 description Router as20r1
!
network 200.2.0.0/16
```

## FRR CONF (as20r1)

```
! FRRouting configuration file
!
password zebra
enable password zebra
!
log file /var/log/frr/frr.log
!
! BGP CONFIGURATION
!
debug bgp keepalives
debug bgp updates in
debug bgp updates out
!
router bgp 20
! no bgp ebgp-requires-policy
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
```

- 1- **neighbor 11.0.0.33 default originate** -> mando la default da isp a 11.0.0.33
- 2- **neighbor 11.0.0.33 prefix-list customerIn in** -> ricevi da as200 solo 200.2.0.0/16, il resto nega tutto
- 3- **neighbor 11.0.0.33 prefix-list defaultOut out** -> manda a as200 solo la default
- 4- vado su router 200 **ip route** -> ho la default che mi arriva da sopra, /16 dove sto io e 11.0.0.32/30 direttamente connessa

```
default nhid 6 via 11.0.0.34 dev eth0 proto bgp metric 20
11.0.0.32/30 dev eth0 proto kernel scope link src 11.0.0.33
200.2.0.0/16 dev eth1 proto kernel scope link src 200.2.0.1
```

- 5- vado su router 20 **ip route** -> ho /24 e /30 dir. connesse, /30 del peering, /16 da sotto

```
11.0.0.4/30 dev eth1 proto kernel scope link src 11.0.0.6
11.0.0.32/30 dev eth0 proto kernel scope link src 11.0.0.34
20.1.1.0/24 dev eth2 proto kernel scope link src 20.1.1.1
200.2.0.0/16 nhid 8 via 11.0.0.33 dev eth0 proto bgp metric 20
```

- 6- **sh ip bgp** -> non le ho tutte per via dei filtri che ho inserito. Vedo la 0/0 perché avevo comando network, 20.1.1.0/24 per stesso motivo, 200.2.0.0/24 la prendo da sotto con as path 200 i

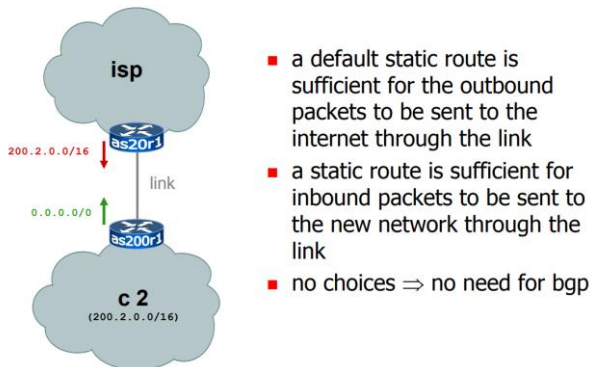
```
as20r1-frr# sh ip bgp
BGP table version is 3, local router ID is 20.1.1.1, vrf id 0
Default local pref 100, local AS 20
Status codes: s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   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
```

## BGP-STUB AS STATIC

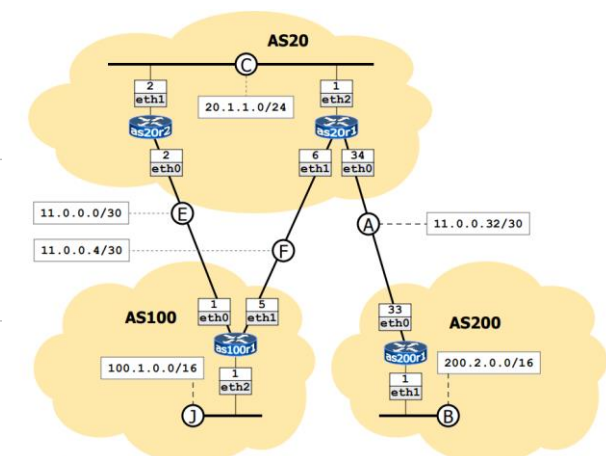
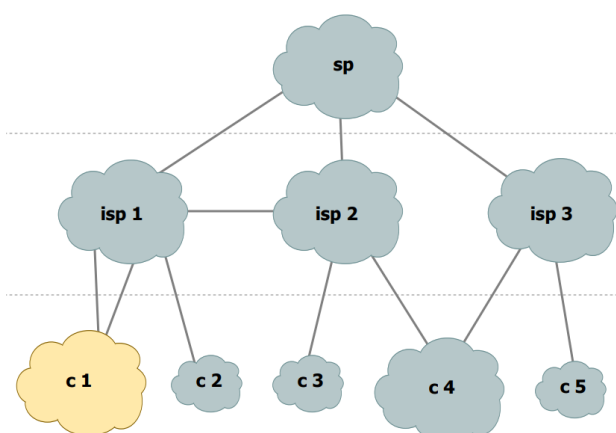
Quando ho una situazione di questo tipo:



Non serve fare bgp, se non ho nessun tipo di comportamento “intelligente”.

## BGP-MULTI HOMED STUB

Consideriamo il caso in cui un customer abbia due link verso un unico isp. Isp decide se filtrare gli annunci o preferirli o sfavorirli. Userò quindi un link come primario e uno come secondario definendo alcune metriche nella configurazione. Quello che non voglio è che l'isp sfrutti il customer per fare traffico che serve a lui. Per gli annunci uscenti dal customer, inserirò un valore di metrica, che di default è zero, nel link secondario (più è alto e più sfavorisco quel link). Per gli annunci entranti nel customer, specifico un valore di local pref, che di default è 100, nel link secondario (più è basso e meno lo preferisco).



## FRR CONF (router as100r1)

```
router bgp 100
! no bgp ebgp-requires-policy (not needed for this lab)
! no bgp network import-check (not needed for this lab)
!
! NEIGHBORS
!
neighbor 11.0.0.2 remote-as 20
neighbor 11.0.0.2 description Router as20r2 (primary)
neighbor 11.0.0.6 remote-as 20
neighbor 11.0.0.6 description Router as20r1 (backup)
!
! NETWORKS
!
network 100.1.0.0/16
!
! POLICIES
!
neighbor 11.0.0.2 prefix-list mineOutOnly out
neighbor 11.0.0.2 prefix-list defaultIn in
neighbor 11.0.0.6 prefix-list mineOutOnly out
neighbor 11.0.0.6 route-map metricOut out
neighbor 11.0.0.6 prefix-list defaultIn in
neighbor 11.0.0.6 route-map localPrefIn in
!
access-list myAggregate permit 100.1.0.0/16
!
ip prefix-list mineOutOnly permit 100.1.0.0/16
ip prefix-list defaultIn permit 0.0.0.0/0
!
route-map metricOut permit 10
match ip address myAggregate
set metric 10
!
route-map localPrefIn permit 10
set local-preference 90
```

- 1- **neighbor 11.0.0.2 prefix-list mineOutOnly out** -> dico che posso annunciare solo la 100.1.0.0/16
- 2- **neighbor 11.0.0.2 prefix-list defaultIn in** -> ricevo solamente la 0.0.0.0/0
- 3- faccio lo stesso per il neighbor 11.0.0.6 (altro link) -> aggiungo route map per modificare attributi  
**neighbor 11.0.0.6 route-map metricOut out** -> se fai match con 100.1.0.0/16( definisci access list)  
setta metrica a 10 (sfavorisci quel collegamento)  
**neighbor 11.0.0.6 route-map localPrefIn in** -> setta la local pref a 90 (sfavorisci quell collegamento)

## FRR CONF (router as20r1)

```
router bgp 20
! no bgp ebgp-requires-policy (not needed for this lab)
! no bgp network import-check (not needed for this lab)
!
! NEIGHBORS
!
neighbor 11.0.0.33 remote-as 200
neighbor 11.0.0.33 description Router as200r1
neighbor 11.0.0.5 remote-as 100
neighbor 11.0.0.5 description Router as100r1
neighbor 20.1.1.2 remote-as 20
neighbor 20.1.1.2 description Router as20r2 (iBGP)
!no neighbor 20.1.1.2 next-hop-self
!
! NETWORKS
!
network 20.1.1.0/24
network 11.0.0.4/30
network 11.0.0.32/30
network 0.0.0.0/0
!
! POLICIES
!
neighbor 11.0.0.33 default-originate
neighbor 11.0.0.33 prefix-list as200In in
neighbor 11.0.0.33 prefix-list defaultOut out
neighbor 11.0.0.5 default-originate
neighbor 11.0.0.5 prefix-list as100In in
neighbor 11.0.0.5 prefix-list defaultOut out
!
ip prefix-list as200In permit 200.2.0.0/16
ip prefix-list as100In permit 100.1.0.0/16
ip prefix-list defaultOut permit 0.0.0.0/0
```

E' necessario specificare i vicini. In questo caso ho un peering tra due router che appartengono allo stesso AS.

neighbor 20.1.1.2 .... (iBGP)



- 1- vado in as100r1, **ip route** -> vede la default che gliela manda as20r2 con link primario, le due lan di peering, e la propria dir. connessa
- 2- **sh ip bgp 0.0.0.0/0** -> mi dice cosa vede bgp riguardo alla default. La ricevo da due link, ma sceglie come best 11.0.0.2 perché localPref maggiore

```
as100r1-frr# sh ip bgp 0.0.0.0/0
BGP routing table entry for 0.0.0.0/0, version 2
Paths: (2 available, best #2, table default)
Not advertised to any peer
20, (stale)
  11.0.0.6 from 11.0.0.6 (20.1.1.1)
    Origin IGP, metric 0, localpref 90, valid, external
    Last update: Wed Nov 22 08:43:01 2023
20
  11.0.0.2 from 11.0.0.2 (20.1.1.2)
    Origin IGP, metric 0, valid, external, best (Local Pref)
    Last update: Wed Nov 22 08:43:01 2023
```

- 3- vado su as20r1 -> **sh ip bgp** -> la default la prendo nel mio AS, prendo la 11.0.0.0/30 tramite il mio vicino, 11.0.0.4/30 dir. connesso, 11.0.0.32/30 dir. connesso, 20.1.1.0/24 la prendo con iBGP tramite vicino e tramite me stesso, stessa cosa per 100.1.0.0/16 ma scelgo da link primario, 200.2.0.0/16 la prendo da sotto. Possiamo vedere che quando ricevo annunci via ibgp, il n.h rimane quello di partenza-esterno (n.h di as100r1).

```
as20r1-frr# sh ip bgp
BGP table version is 7, 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
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
   * 0.0.0.0/0      0.0.0.0          0         32768 i
   *>i11.0.0.0/30    20.1.1.2          0         100    0 i
   *> 11.0.0.4/30    0.0.0.0          0         32768 i
   *> 11.0.0.32/30   0.0.0.0          0         32768 i
   * i20.1.1.0/24    20.1.1.2          0         100    0 i
   *>                0.0.0.0          0         32768 i
   *>i100.1.0.0/16   11.0.0.1          0         100    0 100 i
   * 11.0.0.5        11.0.0.5         10         0 100 i
   *> 200.2.0.0/16   11.0.0.33         0         0 200 i
```

- 4- vado su as200r1 -> **sh ip bgp** -> vedo la default che me la manda as20r1 (as path 20 i), 200.2.0.0/16 che è dir. connessa. Le altre non le vedo perché a as200r1 non gli arriva nulla per via dei filtri.

```
as200r1-frr# sh ip bgp
BGP table version is 2, local router ID is 200.2.0.1, vrf id 0
Default local pref 100, local AS 200
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
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
   *> 0.0.0.0/0      11.0.0.34         0         0 20 i
   *> 200.2.0.0/16    0.0.0.0          0         32768 i

Displayed 2 routes and 2 total paths
```

- 5- vado su as20r2 -> **sh ip bgp** -> default la inietto io, 11.0.0.0/30 dir. connessa, 11.0.0.4/30 me la passa il vicino con ibgp, 11.0.0.32/30 me la passa il vicino con ibgp, 20.1.1.0/24 la ricevo con il vicino e dir. connessa (scelgo questa), 100.1.0.0/16 la prendo con il link primario, 200.2.0.0/16 me la passa il vicino con ibgp.

```
as20r2-frr# sh ip bgp
BGP table version is 7, local router ID is 20.1.1.2, 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
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
  -----
0.0.0.0/0         0.0.0.0          0         32768 i
*> 11.0.0.0/30     0.0.0.0          0         32768 i
*>i11.0.0.4/30     20.1.1.1         0        100      0 i
*>i11.0.0.32/30    20.1.1.1         0        100      0 i
* i20.1.1.0/24     20.1.1.1         0        100      0 i
*>                 0.0.0.0          0         32768 i
*> 100.1.0.0/16    11.0.0.1         0          0 100 i
*>i200.2.0.0/16    11.0.0.33        0        100      0 200 i

Displayed 7 routes and 8 total paths
```

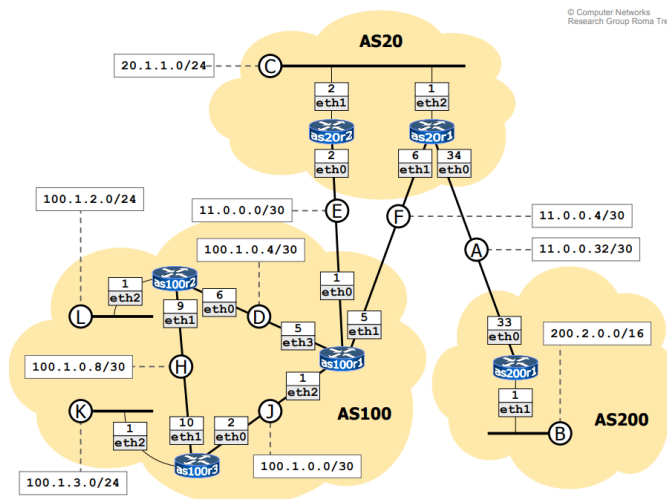
- 6- **traceroute da as100 a 200.2.0.1** -> si vede che il link primario funziona

```
root@as100r1:/# traceroute 200.2.0.1
traceroute to 200.2.0.1 (200.2.0.1), 30 hops max, 60 byte packets
 1  11.0.0.2 (11.0.0.2)  1.632 ms  1.508 ms  1.466 ms
 2  20.1.1.1 (20.1.1.1)  1.374 ms  1.221 ms  1.154 ms
 3  200.2.0.1 (200.2.0.1)  1.102 ms  1.014 ms  0.854 ms
```

## BGP-MULTI HOMED STUB LARGE

In questo tipo di lab, metto un'intera rete in cui distribuisco bgp e non solo un router.

Il router di frontiera di quell'area avrà una configurazione rip e una configurazione bgp.



FRR CONF (router di frontiera as100r1)

```
! RIP CONFIGURATION
!
router rip
 redistribute bgp
 network 100.1.0.0/16
!
! BGP CONFIGURATION
!
debug bgp keepalives
debug bgp updates in
debug bgp updates out
!
router bgp 100
! no bgp ebgp-requires-policy (not needed for this lab)
no bgp network import-check
!
! NEIGHBORS
!
neighbor 11.0.0.2 remote-as 20
neighbor 11.0.0.2 description Router as20r2 (primary)
neighbor 11.0.0.6 remote-as 20
neighbor 11.0.0.6 description Router as20r1 (backup)
!
! NETWORKS
!
network 100.1.0.0/16
```

```
! POLICIES
!
neighbor 11.0.0.2 prefix-list mineOutOnly out
neighbor 11.0.0.2 prefix-list defaultIn in
neighbor 11.0.0.6 prefix-list mineOutOnly out
neighbor 11.0.0.6 route-map metricOut out
neighbor 11.0.0.6 prefix-list defaultIn in
neighbor 11.0.0.6 route-map localPrefIn in
!
access-list myAggregate permit 100.1.0.0/16
!
ip prefix-list mineOutOnly permit 100.1.0.0/16
ip prefix-list defaultIn permit 0.0.0.0/0
!
route-map metricOut permit 10
match ip address myAggregate
set metric 10
!
route-map localPrefIn permit 10
set local-preference 90
```

- 1- nella configurazione rip ci metto un **redistribute bgp** per iniettare le rotte bgp e farle conoscere anche ai vicini che sono router rip
- 2- vado su as100r3 e faccio **sh ip route** -> la default proviene da rip perché lui non parla bgp e il router di frontiera gli trasforma bgp in rip e lui vede comunque rip, due sono dir. connesse e due me le da il vicino rip

```
Hello, this is FRRouting (version 9.0.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

as100r3-frr# sh 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, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

R>* 0.0.0.0/0 [120/2] via 100.1.0.1, eth0, weight 1, 00:00:32
C>* 100.1.0.0/30 is directly connected, eth0, 00:00:40
R>* 100.1.0.4/30 [120/2] via 100.1.0.9, eth1, weight 1, 00:00:36
C>* 100.1.0.8/30 is directly connected, eth1, 00:00:40
R>* 100.1.2.0/24 [120/2] via 100.1.0.9, eth1, weight 1, 00:00:36
C>* 100.1.3.0/24 is directly connected, eth2, 00:00:40
```

- 3- vado su as100r1 e faccio **sh ip bgp** -> la default la ottengo dal link primario come bgp perché parla entrambi e la riceve come gliela mandano e poi c'è la dir. connessa

```
as100r1-frr# sh ip bgp
BGP table version is 2, local router ID is 100.1.0.5, vrf id 0
Default local pref 100, local AS 100
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
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
S  0.0.0.0/0         11.0.0.6             0      90      0 20 i
*> 11.0.0.2          0                    0           0 20 i
*> 100.1.0.0/16      0.0.0.0              0          32768 i

Displayed 2 routes and 3 total paths
```

- 4- vado su as100r1 e faccio **sh ip rip** -> la default la ottengo via bgp da sopra, poi ho le lan che ottengo via rip e quelle dir. connesse

```
as100r1-frr# sh ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
         (n) - normal, (s) - static, (d) - default, (r) - redistribute,
         (i) - interface

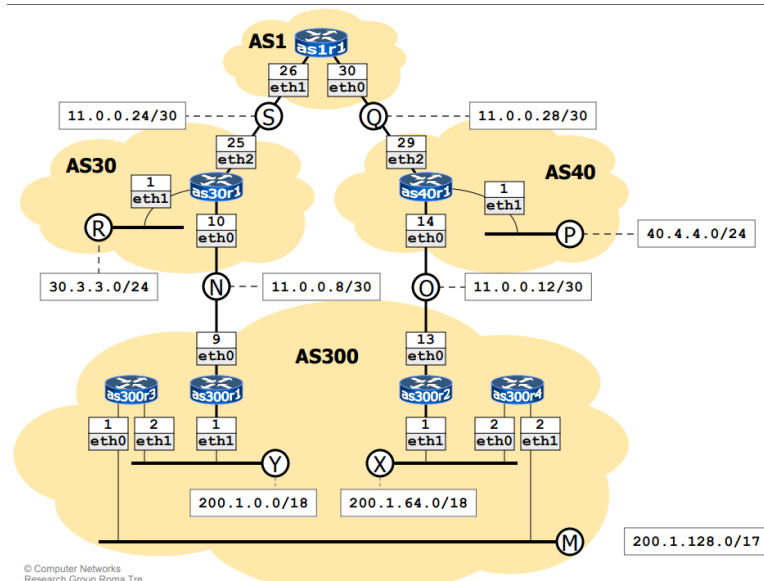
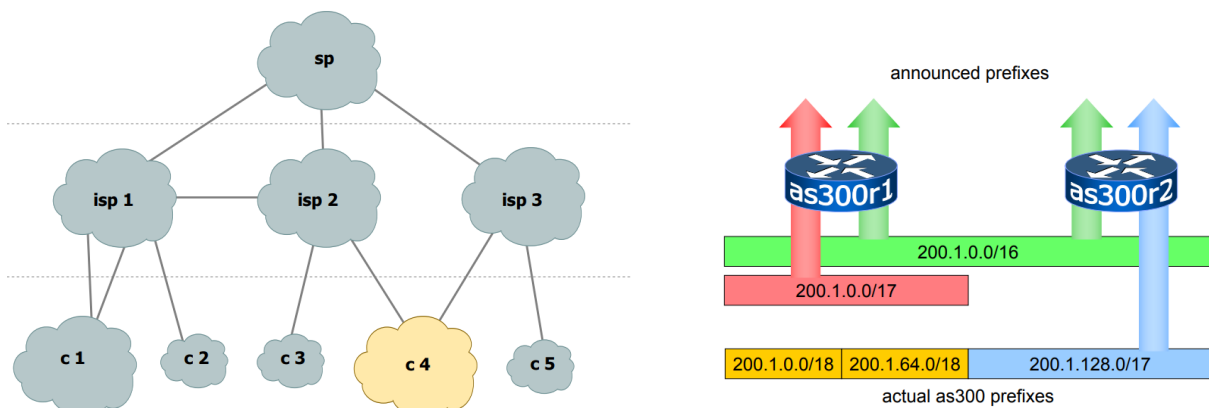
   Network          Next Hop          Metric From          Tag Time
B(r) 0.0.0.0/0       11.0.0.2             1 self            0
C(i) 100.1.0.0/30     0.0.0.0              1 self            0
C(i) 100.1.0.4/30     0.0.0.0              1 self            0
R(n) 100.1.0.8/30     100.1.0.6            2 100.1.0.6        0 02:52
R(n) 100.1.2.0/24     100.1.0.6            2 100.1.0.6        0 02:52
R(n) 100.1.3.0/24     100.1.0.2            2 100.1.0.2        0 02:46
```

## BGP-MULTI HOMED

In questo lab vediamo il caso in cui un customer ha due link con due provider diversi.

L'importante è fare load balancing sui due link, in cui metà del traffico passa su uno e metà del traffico passa sull'altro. Per gli annunci in uscita e in ingresso posso passare indifferentemente o su link 1 o su link 2. Non voglio però essere attraversato da traffico che serve a uso personale per isp.

Con bgp scelgo quale prefisso annunciare: es /16 la divido in due /17. Queste le annuncio su entrambi i link verso i provider. Coloro che vogliono raggiungere la prima porzione sulla barra di indirizzamento scelgono la prima /17; gli altri la seconda /17. Nei router sopra a isp1 e isp2, nella tabella di instradamento avrò due righe ulteriori per le due /17 aggiunte. Mandando un pacchetto alla prima /17 farà matching con la prima riga, un pacchetto verso la seconda /17 farà matching con la seconda. Se non sono soddisfatto del bilanciamento suddivido ulteriormente il prefisso. Lo scopo di questo è consentire che se un link è guasto l'altro continua a funzionare. Se link 1 è guasto, coloro che raggiungono la prima /17 e non possono fare matching con quella, faranno matching con la seconda su link 2. Oppure faranno matching con la /16 che è annunciata ovunque.



## FRR CONF (router di frontiera as300r1)

```
! RIP CONFIGURATION
!
router rip
redistribute bgp
network 200.1.0.0/16
!
! BGP CONFIGURATION
!
debug bgp keepalives
debug bgp updates in
debug bgp updates out
!
router bgp 300
! no bgp ebgp-requires-policy (not needed for this router)
no bgp network import-check
!
! NEIGHBORS
!
neighbor 11.0.0.10 remote-as 30
neighbor 11.0.0.10 description Router as30r1
!
! NETWORKS
!
network 200.1.0.0/16
network 200.1.0.0/17
!
! POLICIES
!
neighbor 11.0.0.10 prefix-list mineOutOnly out
neighbor 11.0.0.10 prefix-list defaultIn in
!
ip prefix-list mineOutOnly permit 200.1.0.0/16
ip prefix-list mineOutOnly permit 200.1.0.0/17
ip prefix-list defaultIn permit 0.0.0.0/0
```

- 1- come neighbor ho solamente as30r1 perché ho un link verso il provider
- 2- annuncio sia /16 che /17 e permetto solo quelle
- 3- ricevo solo 0.0.0.0/0 con prefix-list
- 4- vado su as30r1 e faccio **sh ip route** -> la default viene appresa tramite bgp e la prendo da isp, ho le due /30 dir.connesse, /24 dir. connessa, apprendo /17 e /16 da bgp (/17 sta prima perché è più specifica)

```
as30r1-frr# sh 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, F - PBR,
        f - OpenFabric,
        > - selected route, * - FIB route, q - queued, r - rejected, b - backup
        t - trapped, o - offload failure

B>* 0.0.0.0/0 [20/0] via 11.0.0.26, eth2, weight 1, 00:00:24
C>* 11.0.0.8/30 is directly connected, eth0, 00:00:27
C>* 11.0.0.24/30 is directly connected, eth2, 00:00:27
C>* 30.3.3.0/24 is directly connected, eth1, 00:00:27
B>* 200.1.0.0/16 [20/0] via 11.0.0.9, eth0, weight 1, 00:00:23
B>* 200.1.0.0/17 [20/0] via 11.0.0.9, eth0, weight 1, 00:00:23
as30r1-frr# exit
root@as30r1:/# ip route
default nhid 8 via 11.0.0.26 dev eth2 proto bgp metric 20
11.0.0.8/30 dev eth0 proto kernel scope link src 11.0.0.10
11.0.0.24/30 dev eth2 proto kernel scope link src 11.0.0.25
30.3.3.0/24 dev eth1 proto kernel scope link src 30.3.3.1
200.1.0.0/17 nhid 10 via 11.0.0.9 dev eth0 proto bgp metric 20
200.1.0.0/16 nhid 10 via 11.0.0.9 dev eth0 proto bgp metric 20
```

- 5- vado su as300r2 e faccio **sh ip route** -> ho la default appresa da bgp, /30 dir. connessa, /18 che apprendo tramite rip dal vicino, /18 dir. Connessa, /17 appresa tramite rip dal vicino.

```
as300r2-frr# sh 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, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

B>* 0.0.0.0/0 [20/0] via 11.0.0.14, eth0, weight 1, 00:02:55
C>* 11.0.0.12/30 is directly connected, eth0, 00:02:59
R>* 200.1.0.0/18 [120/3] via 200.1.64.2, eth1, weight 1, 00:02:56
C>* 200.1.64.0/18 is directly connected, eth1, 00:02:59
R>* 200.1.128.0/17 [120/2] via 200.1.64.2, eth1, weight 1, 00:02:56
```

- 6- **sh ip rip** -> vedo la default non come rip ma come bgp

```
as300r2-frr# sh ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
         (n) - normal, (s) - static, (d) - default, (r) - redistribute,
         (i) - interface

      Network      Next Hop      Metric From      Tag Time
B(r) 0.0.0.0/0     11.0.0.14          1 self           0
R(n) 200.1.0.0/18  200.1.64.2         3 200.1.64.2     0 02:48
C(i) 200.1.64.0/18 0.0.0.0            1 self           0
R(n) 200.1.128.0/17 200.1.64.2         2 200.1.64.2     0 02:48
```

- 7- vado su as300r3 e faccio **sh ip route** -> apprende la default da rip perché è interno e non sa che è stata iniettata da bgp

```
as300r3-frr# sh 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, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

R>* 0.0.0.0/0 [120/2] via 200.1.0.1, eth1, weight 1, 00:10:51
C>* 200.1.0.0/18 is directly connected, eth1, 00:10:56
R>* 200.1.64.0/18 [120/2] via 200.1.128.2, eth0, weight 1, 00:10:54
C>* 200.1.128.0/17 is directly connected, eth0, 00:10:56
```

- 8- vado su as1r1 e faccio **sh ip bgp** -> vedo la default e la inietto io, due /30 direttamente connesse, 11.0.0.8/30 che arriva da AS30, 11.0.0.12/30 che arriva da AS40, 30.3.3.0/24 che arriva da AS30, 40.4.4.0/24 che arriva da AS40, due /16 sui due link dei due provider con diversi as path, 200.1.0.0/17 e 200.1.128.0/17 dei due link

```
as1r1-frr# sh ip bgp
BGP table version is 10, local router ID is 11.0.0.30, vrf id 0
Default local pref 100, local AS 1
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
RPKI validation codes: V valid, I invalid, N Not found

      Network      Next Hop      Metric LocPrf Weight Path
*> 0.0.0.0/0        0.0.0.0          0           32768 i
*> 11.0.0.8/30      11.0.0.25         0           0 30 i
*> 11.0.0.12/30     11.0.0.29         0           0 40 i
*> 11.0.0.24/30     0.0.0.0           0           32768 i
*> 11.0.0.28/30     0.0.0.0           0           32768 i
*> 30.3.3.0/24      11.0.0.25         0           0 30 i
*> 40.4.4.0/24      11.0.0.29         0           0 40 i
* 200.1.0.0/16      11.0.0.25         0           0 30 300 i
*>                  11.0.0.29         0           0 40 300 i
*> 200.1.0.0/17     11.0.0.25         0           0 30 300 i
*> 200.1.128.0/17   11.0.0.29         0           0 40 300 i

Displayed 10 routes and 11 total paths
```

9- traceroute da as1r1 a 200.1.128.2

```
root@as1r1:/# traceroute 200.1.128.2
traceroute to 200.1.128.2 (200.1.128.2), 30 hops max, 60 byte packets
 1  11.0.0.29 (11.0.0.29)  0.811 ms  0.623 ms  0.573 ms
 2  11.0.0.13 (11.0.0.13)  0.559 ms  0.508 ms  0.492 ms
 3  200.1.128.2 (200.1.128.2)  0.477 ms  0.408 ms  0.382 ms
```