Notice of Liability

Despite careful checking of content, we accept no liability for the content of external links. Content on linked sites is exclusively the responsibility of the respective website operator.

Links visited during the webinar

→ BGP Route Reflectors: https://www.rfc-editor.org/info/rfc4456
→ BGP Communities:
  • Large: http://www.rfc-editor.org/info/rfc8092 and http://www.rfc-editor.org/info/rfc8195
→ Regular expressions:
  • Concept: https://en.wikipedia.org/wiki/Regular_expression
  • In Cisco IOS:
    • https://www.cisco.com/c/en/us/td/docs/ios/12_2/termserv/configuration/guide/ftersv_c/tcfaaapr.html#wp1022889
→ Cisco Route-Maps
→ Tools for network emulation:
  • GNS3: https://www.gns3.com
  • Mikrotik router image: https://mikrotik.com/download (use Cloud Hosted Router)
Example Network
GNS3 is used to emulate the routers in our example network. We are using Cisco routers with IOS. Routers in AS64500 are named R1-R4 and are "properly" configured, while routers in the other Autonomous Systems are just there to do BGP to AS64500 and are not "properly" configured.
Cisco IOS Router Configurations - AS64500

R1

hostname R1
!
interface Loopback0
   ip address 192.168.1.1 255.255.255.255
   ip router isis
!
interface GigabitEthernet0/0
   description to R3
   ip address 192.168.2.1 255.255.255.252
   ip router isis
!
interface GigabitEthernet1/0
   description to R2
   ip address 192.168.2.5 255.255.255.252
   ip router isis
!
interface GigabitEthernet2/0
   description to AS64496
   ip address 10.96.1.2 255.255.255.252
!
   router isis
   net 49.0000.0000.0000.0001.00
!
   router bgp 64500
   neighbor internal peer-group
   neighbor internal remote-as 64500
   neighbor internal update-source Loopback0
   neighbor internal next-hop-self
   neighbor internal send-community both
   neighbor upstream peer-group
   neighbor upstream soft-reconfiguration inbound
   neighbor upstream route-map upstream-in in
   neighbor upstream route-map upstream-out out
   neighbor 10.96.1.1 remote-as 64496
   neighbor 10.96.1.1 peer-group upstream
   neighbor 192.168.1.2 peer-group internal
   neighbor 192.168.1.3 peer-group internal
!
   ip bgp-community new-format
   ip community-list expanded announce-to-customers permit 64500:4[1357][0-9][0-9][0-9]
   ip community-list expanded announce-to-peers permit 64500:4[2367][0-9][0-9][0-9]i
   ip community-list expanded announce-to-upstream permit 64500:4[4567][0-9][0-9][0-9]i
   ip community-list expanded delete-my-communities permit 64500:.*
!
We are using IS-IS as IGP. This is to enable it on this interface and to announce the interfaces address.

No IS-IS to Upstream

This number needs to be unique in your network.
The Big Picture

```bash
route-map upstream-in permit 100
  set metric 0
  set local-preference 10
  set comm-list delete-my-communities delete
  set community 64500:41000 additive
! route-map upstream-in deny 65000
! route-map upstream-out permit 100
  match community announce-to-upstream
  set metric 0
! route-map upstream-out deny 65000
! end

We set community 64500:41000 here
This matches against "announce-to-customers" on other routers.

R2

hostname R2
!
interface Loopback0
  ip address 192.168.1.2 255.255.255.255
  ip router isis
!
interface GigabitEthernet0/0
  description to R3
  ip address 192.168.2.9 255.255.255.252
  ip router isis
!
interface GigabitEthernet1/0
  description to R1
  ip address 192.168.2.6 255.255.255.252
  ip router isis
!
interface GigabitEthernet2/0
  description to AS64497
  ip address 10.97.1.2 255.255.255.252
!
interface GigabitEthernet3/0
  description to DE-CIX / AS517
  ip address 80.81.192.200 255.255.252.0
  ! router isis
  net 49.0000.0000.0000.0002.00
!
router bgp 64500
neighbor internal peer-group
neighbor internal remote-as 64500
neighbor internal update-source Loopback0
```
neighbor internal next-hop-self
neighbor internal send-community both
neighbor upstream peer-group
neighbor upstream soft-reconfiguration inbound
neighbor upstream route-map upstream-in in
neighbor upstream route-map upstream-out out
neighbor peering peer-group
neighbor peering soft-reconfiguration inbound
neighbor peering route-map peering-in in
neighbor peering route-map peering-out out
neighbor 10.97.1.1 remote-as 64497
neighbor 10.97.1.1 peer-group upstream
neighbor 80.81.193.17 remote-as 517
neighbor 80.81.193.17 peer-group peering
neighbor 192.168.1.1 peer-group internal
neighbor 192.168.1.3 peer-group internal

ip bgp-community new-format
ip community-list expanded announce-to-customers permit 64500:4[1357][0-9][0-9][0-9]
ip community-list expanded announce-to-peers permit 64500:4[2367][0-9][0-9][0-9]
ip community-list expanded announce-to-upstream permit 64500:4[4567][0-9][0-9][0-9]
ip community-list expanded delete-my-communities permit 64500:.*

! route-map peering-out permit 100
  match community announce-to-peers
  set metric 0
  set comm-list delete-my-communities delete
route-map peering-out deny 65000
!
route-map upstream-in permit 100
  set local-preference 10
  set comm-list delete-my-communities delete
  set community 64500:41000 additive
route-map upstream-in deny 65000
!
route-map peering-in permit 100
  set local-preference 1000
  set comm-list delete-my-communities delete
  set community 64500:41000 additive
route-map peering-in deny 65000
!
route-map upstream-out permit 100
  match community announce-to-upstream
  set metric 0
  set comm-list delete-my-communities delete
route-map upstream-out deny 65000
!
end

All communities starting with "64500:" get deleted when sending and receiving prefixes
**R3 (at start)**

```
hostname R3
!
interface GigabitEthernet0/0
description to R1
  ip address 192.168.2.2 255.255.255.252
!
interface GigabitEthernet1/0
description to R2
  ip address 192.168.2.10 255.255.255.252
!
interface GigabitEthernet2/0
description to R4
  ip address 192.168.2.13 255.255.255.252
!
interface GigabitEthernet3/0
description to AS64511
  ip address 192.168.3.1 255.255.255.252
!
ip bgp-community new-format
!
end
```
ip bgp-community new-format
!
ip route 192.168.200.0 255.255.255.0 Null0 tag 40000
!
route-map static-to-bgp permit 100
  match tag 41000
  match source-protocol static
  set community 64500:41000
route-map static-to-bgp permit 200
  match tag 42000
  match source-protocol static
  set community 64500:42000
route-map static-to-bgp permit 300
  match tag 43000
  match source-protocol static
  set community 64500:43000
route-map static-to-bgp permit 400
  match tag 44000
  match source-protocol static
  set community 64500:44000
route-map static-to-bgp permit 500
  match tag 45000
  match source-protocol static
  set community 64500:45000
route-map static-to-bgp permit 600
  match tag 46000
  match source-protocol static
  set community 64500:46000
route-map static-to-bgp permit 700
  match tag 47000
  match source-protocol static
  set community 64500:47000
route-map static-to-bgp permit 1000
  match tag 40000
  match source-protocol static
  set community no-export
route-map static-to-bgp deny 10000
end

Static prefix to be redistributed into BGP and not announced to anybody.
Router Configs: Neighbor Networks

Neighbor networks use minimal configuration only.

AS64511 - "Customer" Network

```
hostname AS64511
!
interface GigabitEthernet0/0
description to AS64500 R3
 ip address 192.168.3.2 255.255.255.252
!
routing bgp 64511
 network 172.16.1.0 mask 255.255.255.0
neighbor upstream peer-group
neighbor upstream send-community both
neighbor upstream route-map upstream-in in
neighbor upstream route-map upstream-out out
neighbor 192.168.3.1 remote-as 64500
neighbor 192.168.3.1 peer-group upstream
!
ip bgp-community new-format
!
ip route 172.16.1.0 255.255.255.0 Null0
!
route-map upstream-in permit 100
!
route-map upstream-out permit 100
 set community 286:33 64500:1 64500:1234
!
end
```

Here we set some BGP communities to test AS64500's incoming filters.
AS64496 - "Upstream Provider"

hostname AS64496
  !
  interface GigabitEthernet0/0
  description to AS64500 R1
  ip address 10.96.1.1 255.255.255.252
  !
  interface GigabitEthernet1/0
  description to AS517
  ip address 10.96.1.5 255.255.255.252
  !
  router bgp 64496
  network 10.96.0.0 mask 255.255.0.0
  neighbor customer peer-group
  neighbor customer send-community both
  neighbor customer route-map customer-in in
  neighbor customer route-map customer-out out
  neighbor 10.96.1.2 remote-as 64500
  neighbor 10.96.1.2 peer-group customer
  neighbor 10.96.1.6 remote-as 517
  neighbor 10.96.1.6 peer-group customer
  !
  ip route 10.96.0.0 255.255.0.0 Null0
  !
  route-map customer-in permit 100
  !
  route-map customer-out permit 100
  !
  end

We announce one prefix and let through whatever AS517 announces
AS64497 - "Upstream Provider"

hostname AS64497
interface GigabitEthernet0/0
description to AS64500 R2
ip address 10.97.1.1 255.255.255.252
interface GigabitEthernet1/0
description to AS517
ip address 10.97.1.5 255.255.255.252
router bgp 64497
network 10.97.0.0 mask 255.255.0.0
neighbor customer peer-group
neighbor customer send-community both
neighbor customer route-map customer-in in
neighbor customer route-map customer-out out
neighbor 10.97.1.2 remote-as 64500
neighbor 10.97.1.2 peer-group customer
neighbor 10.97.1.6 remote-as 517
neighbor 10.97.1.6 peer-group customer
ip bgp-community new-format
ip route 10.97.0.0 255.255.0.0 Null0
route-map customer-in permit 100
route-map customer-out permit 100
end

We announce one prefix and let through whatever AS517 announces
AS517 - "some provider who peers"

hostname AS517

interface GigabitEthernet0/0
description to AS64496
ip address 10.96.1.6 255.255.255.252
!
interface GigabitEthernet1/0
description to AS64497
ip address 10.97.1.6 255.255.255.252
!
interface GigabitEthernet2/0
description to DE-CIX
ip address 80.81.193.17 255.255.252.0
!
routers bgp 517
network 172.17.1.0 mask 255.255.255.0
network 172.17.2.0 mask 255.255.255.0
neighbor upstream peer-group
neighbor upstream send-community both
neighbor upstream route-map upstream-in in
neighbor upstream route-map upstream-out out
neighbor peering peer-group
neighbor peering send-community both
neighbor peering route-map peering-in in
neighbor peering route-map peering-out out
neighbor 10.96.1.5 remote-as 64496
neighbor 10.96.1.5 peer-group upstream
neighbor 10.97.1.5 remote-as 64497
neighbor 10.97.1.5 peer-group upstream
neighbor 80.81.192.200 remote-as 64500
neighbor 80.81.192.200 peer-group peering
!
ip route 172.17.1.0 255.255.255.0 Null0
ip route 172.17.2.0 255.255.255.0 Null0
!
ip prefix-list via-peering seq 10 permit 172.17.1.0/24
!
ip prefix-list via-upstream seq 5 permit 172.17.1.0/24
ip prefix-list via-upstream seq 10 permit 172.17.2.0/24
!
routemap peering-in permit 100
route-map peering-out permit 100
match ip address prefix-list via-peering
!
routemap upstream-in permit 100
route-map upstream-out permit 100
match ip address prefix-list via-upstream
!
end
Experiments
During the webinar on router R3 the following experiments are performed.

Experiment 1: Setup IS-IS as IGP
Here we add a Loopback interface and configure IS-IS on all internal interfaces (although it is not really necessary on interface gig 2/0 to R4).

```
int Loopback0
 ip address 192.168.1.3 255.255.255.255
 ip router isis
!
int gig 0/0
 ip router isis
!
int gig 1/0
 ip router isis
!
int gig 2/0
 ip router isis
!
router isis
 net 49.0000.0000.0000.0003.00
!
```

Experiment 2a: Setup fully meshed iBGP to R1 and R2

```
router bgp 64500
 neighbor internal peer-group
 neighbor internal remote-as 64500
 neighbor internal next-hop-self
 neighbor internal send-community both
 neighbor internal update-source loopback 0
!
neighbor 192.168.1.1 peer-group internal
neighbor 192.168.1.2 peer-group internal
```
Experiment 2b: Setup R4 as route reflector client

```
router bgp 64500
  neighbor internal-rr peer-group
  neighbor internal-rr remote-as 64500
  neighbor internal-rr route-reflector-client
  neighbor internal-rr next-hop-self all
  neighbor internal-rr send-community both
!
neighbor 192.168.2.14 peer-group internal-rr
```

Experiment 3: Setup eBGP to customer AS64511

```
router bgp 64500
  neighbor customer peer-group
  neighbor customer send-community both
  neighbor customer soft-reconfiguration inbound
  neighbor customer route-map customer-in in
  neighbor customer route-map customer-out out
  neighbor 192.168.3.2 peer-group customer
!
ip community-list expanded delete-incoming permit 64500:[0-35-9][0-9]*
route-map customer-in permit 100
  set metric 0
  set local-preference 10000
  set comm-list delete-incoming delete
  continue
!
ip community-list expanded announce-community-set permit 64500:4[0-9][0-9][0-9]
route-map customer-in permit 200
  match community announce-community-set
!
route-map customer-in permit 300
  set community 64500:47000 additive
!
route-map customer-in deny 65000
!
ip community-list expanded announce-to-customers permit 64500:4[1357][0-9][0-9][0-9]
route-map customer-out permit 100
  match community announce-to-customers
  set metric 0
  set comm-list announce-to-customers delete
!
route-map customer-out deny 65000
!```
Experiment 4: Adding our own prefixes to BGP

ip route 192.168.0.0 255.255.0.0 Null0 tag 47000

router bgp 64500
    redistribute static metric 0 route-map static-to-bgp

route-map static-to-bgp permit 100
    match tag 41000
    match source-protocol static
    set community 64500:41000

route-map static-to-bgp permit 200
    match tag 42000
    match source-protocol static
    set community 64500:42000

route-map static-to-bgp permit 300
    match tag 43000
    match source-protocol static
    set community 64500:43000

route-map static-to-bgp permit 400
    match tag 44000
    match source-protocol static
    set community 64500:44000

route-map static-to-bgp permit 500
    match tag 45000
    match source-protocol static
    set community 64500:45000

route-map static-to-bgp permit 600
    match tag 46000
    match source-protocol static
    set community 64500:46000

route-map static-to-bgp permit 700
    match tag 47000
    match source-protocol static
    set community 64500:47000

route-map static-to-bgp permit 1000
    match tag 40000
    match source-protocol static
    set community no-export

route-map static-to-bgp deny 10000
Router R3 (final, after all configs are added)

hostname R3
interface Loopback0
  ip address 192.168.1.3 255.255.255.255
  ip router isis
!
interface GigabitEthernet0/0
  description to R1
  ip address 192.168.2.2 255.255.255.252
  ip router isis
!
interface GigabitEthernet1/0
  description to R2
  ip address 192.168.2.10 255.255.255.252
  ip router isis
!
interface GigabitEthernet2/0
  description to R4
  ip address 192.168.2.13 255.255.255.252
  ip router isis
!
interface GigabitEthernet3/0
  description to AS64511
  ip address 192.168.3.1 255.255.255.252
!
routing isis
  net 49.0000.0000.0000.0003.00
!
routing bgp 64500
  redistribute static metric 0 route-map static-to-bgp
  neighbor internal peer-group
  neighbor internal remote-as 64500
  neighbor internal update-source Loopback0
  neighbor internal next-hop-self
  neighbor internal send-community both
  neighbor internal-rr peer-group
  neighbor internal-rr remote-as 64500
  neighbor internal-rr route-reflector-client
  neighbor internal-rr next-hop-self all
  neighbor internal-rr send-community both
  neighbor customer peer-group
  neighbor customer send-community both
  neighbor customer soft-reconfiguration inbound
  neighbor customer route-map customer-in in
  neighbor customer route-map customer-out out
  neighbor 192.168.1.1 peer-group internal
  neighbor 192.168.1.2 peer-group internal
  neighbor 192.168.2.14 peer-group internal-rr
  neighbor 192.168.3.2 remote-as 64511
  neighbor 192.168.3.2 peer-group customer

Here we redistribute static routes into BGP

Only difference in config between a route-reflector client and a standard iBGP neighbor
ip bgp-community new-format
ip community-list expanded delete-incoming permit 64500:[0-35-9][0-9]*
ip community-list expanded announce-community-set permit 64500:[4][0-9][0-9][0-9][0-9]
ip community-list expanded announce-to-customers permit 64500:[4][1357][0-9][0-9][0-9]

ip route 192.168.0.0 255.255.0.0 Null0 tag 47000
ip route 192.168.7.0 255.255.255.0 Null0 tag 40000

route-map customer-in permit 100
  continue
  set metric 0
  set local-preference 10000
  set comm-list delete-incoming delete
  route-map customer-in deny 200
    match community announce-community-set
  route-map customer-in permit 300
    set community 64500:47000 additive
  route-map customer-in deny 65000

route-map customer-out permit 100
  match community announce-to-customers
  set metric 0
  set comm-list announce-to-customers delete
  route-map customer-out deny 65000

route-map static-to-bgp permit 100
  match tag 41000
  match source-protocol static
  set community 64500:41000
route-map static-to-bgp permit 200
  match tag 42000
  match source-protocol static
  set community 64500:42000
route-map static-to-bgp permit 300
  match tag 43000
  match source-protocol static
  set community 64500:43000
route-map static-to-bgp permit 400
  match tag 44000
  match source-protocol static
  set community 64500:44000
route-map static-to-bgp permit 500
  match tag 45000
  match source-protocol static
  set community 64500:45000
route-map static-to-bgp permit 600
  match tag 46000
  match source-protocol static
  set community 64500:46000
route-map static-to-bgp permit 700
  match tag 47000

This prefix goes into BGP but is not announced externally.

This matches the "tag" of the static routes.

Static prefix to be redistributed into BGP and announced to everybody.
match source-protocol static
set community 64500:47000
route-map static-to-bgp permit 1000
  match tag 40000
  match source-protocol static
  set community no-export
route-map static-to-bgp deny 10000
!
end

Static routes with no "tag" do not get into BGP
Mikrotik Router Configurations - AS64500
Many ISPs use Mikrotik routers. As Mikrotik offers a free "Cloud Hosted Router" software image (with some limitations) I also generated a config file for Mikrotik. Note that Mikrotik does not support IS-IS, so IGP in this case is OSPF.
R1

/system identity
set name=R1

/interface bridge
add name=loopback0

/interface ethernet
set [ find default-name=ether1 ] comment="to R3"
set [ find default-name=ether2 ] comment="to R2"
set [ find default-name=ether3 ] comment="to AS64496"

/ip address
add address=192.168.2.1/30 interface=ether1 network=192.168.2.0
add address=192.168.2.5/30 interface=ether2 network=192.168.2.4
add address=10.96.1.2/30 interface=ether3 network=10.96.1.0
add address=192.168.1.1 interface=loopback0 network=192.168.1.1

/routing ospf instance
set [ find default=yes ] redistribute-connected=as-type-2 router-id=192.168.1.1

/routing ospf network
add area=backbone network=192.168.2.0/24

/routing bgp instance
set default as=64500 router-id=192.168.1.1

/routing bgp peer
add name=R2 nexthop-choice=force-self remote-address=192.168.1.2 remote-as=64500 update-source=loopback0
add name=R3 nexthop-choice=force-self remote-address=192.168.1.3 remote-as=64500 update-source=loopback0
add in-filter=upstream-in name=AS64496 out-filter=upstream-out remote-address=10.96.1.1 remote-as=64496

/routing filter
add chain=upstream-in set-bgp-communities=64500:41000 set-bgp-local-pref=10 set-bgp-med=0
add action=accept bgp-communities=64500:42000 chain=upstream-out set-bgp-med=0
add action=accept bgp-communities=64500:45000 chain=upstream-out set-bgp-med=0
add action=accept bgp-communities=64500:47000 chain=upstream-out set-bgp-med=0
add action=discard chain=upstream-out

Mikrotik does not support regular expressions here
So we need one rule per possible Community
R3 (complete)

/system identity
set name=R3

/interface bridge
add name=loopback0

/interface ethernet
set [ find default-name=ether1 ] comment="to R1"
set [ find default-name=ether2 ] comment="to R2"
set [ find default-name=ether3 ] comment="to R4"
set [ find default-name=ether4 ] comment="to AS64511"

/ip address
add address=192.168.2.2/30 interface=ether1 network=192.168.2.0
add address=192.168.2.10/30 interface=ether2 network=192.168.2.8
add address=192.168.2.13/30 interface=ether3 network=192.168.2.12
add address=192.168.3.1/30 interface=ether4 network=192.168.3.0
add address=192.168.1.3/32 interface=loopback0 network=192.168.1.3

/routing ospf instance
set [ find default=yes ] redistribute-connected=as-type-2 router-id=192.168.1.3
/routing ospf network
add area=backbone network=192.168.2.0/24

/routing filter
add action=accept bgp-communities=64500:40000 chain=bgp-out
add action=accept bgp-communities=64500:41000 chain=bgp-out
add action=accept bgp-communities=64500:42000 chain=bgp-out
add action=accept bgp-communities=64500:43000 chain=bgp-out
add action=accept bgp-communities=64500:44000 chain=bgp-out
add action=accept bgp-communities=64500:45000 chain=bgp-out
add action=accept bgp-communities=64500:46000 chain=bgp-out
add action=accept bgp-communities=64500:47000 chain=bgp-out
add action=discard chain=bgp-out
/routing bgp instance
set default as=64500 out-filter=bgp-out redistribute-static=yes router-id=192.168.1.3

/routing bgp peer
add name=R1 nexthop-choice=force-self remote-address=192.168.1.1 remote-as=64500 update-source=loopback0
add name=R2 nexthop-choice=force-self remote-address=192.168.1.2 remote-as=64500 update-source=loopback0