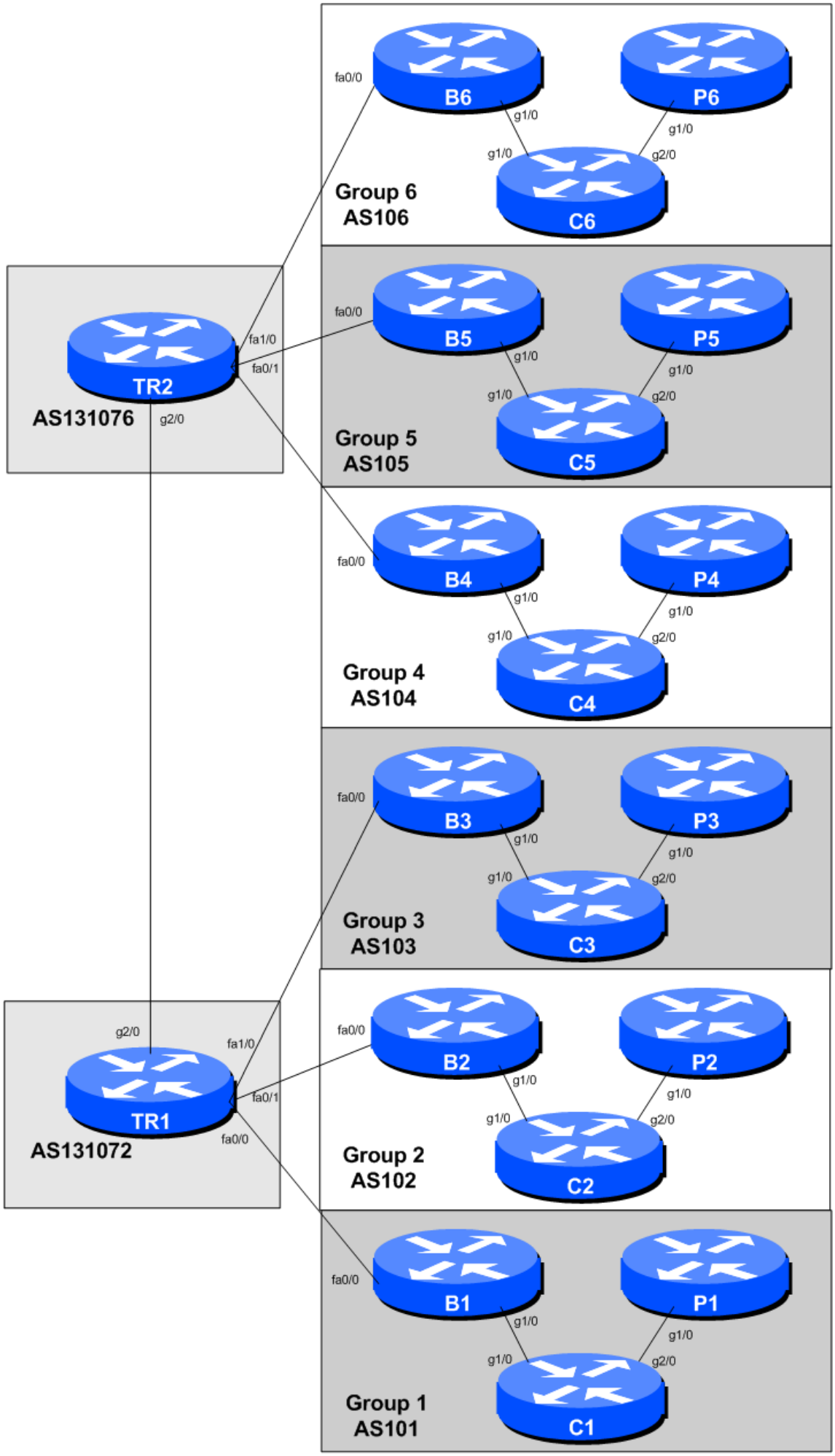
Peering & IXP Lab (Part 1: Transit)

The purpose of this lab is to investigate the differences and best practices for transit and peering configuration when an AS is connected both to a transit provider, a private peer, and an Internet Exchange Point.

Lab Topology

The initial lab topology sets up 6 autonomous systems, each with three routers. In each AS, one router is the border router (for connecting to transit providers), one router is the core router (representing the rest of the network operator's core network), and one router is a peering router (for connecting to private peers and IXPs).

The lab will start simply by configuring each autonomous system, and making sure that transit works via their transit provider. The address plan for the entire network is described in the Address Plan document. The initial lab topology is shown below.


Group 6
AS106
B6
P6
C6
fa0/0
g1/0
g2/0
Group 5
AS105
B5
P5
C5
Group 4
AS104
B4
P4
C4
Group 3
AS103
B3
P3
C3
Group 2
AS102
B2
P2
C2
Group 1
AS101
B1
P1
C1
TR2
AS131076
fa1/0
fa0/1
TR1
AS131072

Logistics

Each participant will be assigned to a group. Depending on the number of participants, either a single person or a group will be responsible for the configuration of a router. You may be asked to rotate and work on a different router so that you have the opportunity to understand the network from another point of view.

As you go through the exercises, you will see examples of configurations for one or more routers. **Make sure to take those examples and adapt them to your own router, network topology and addressing scheme. Use the diagrams to guide you.**

Refer to the Lab Access Instructions document for information about logging into the routers that have been assigned to you.

Address Space Allocation

Refer to the Address Plan document for information about the IP address plan for the network infrastructure for these labs.

Basic Router Configuration

The following configuration examples show the suggested/recommended configuration to be implemented on the routers in each group. Replace the **RX** in the examples with the router type (either B for Border or C for Core or P for Peering) and Group number as appropriate.

Name the router

```
Router> enable
Router# config terminal
Router(config)# hostname RX
```

Configure Authentication

```
aaa new-model
aaa authentication login default local
aaa authentication enable default enable
username nsrc secret lab-PW
enable secret lab-EN
service password-encryption
line vty 0 4
 transport preferred none
line console 0
 transport preferred none
```

Configure logging

```
no logging console
logging buffered 8192 debugging
```

Disable DNS resolution

```
no ip domain-lookup
```

Activate IPv6 routing

Turn on IPv6 Routing and activate IPv6 CEF (not on by default in Cisco IOS)

```
ipv6 unicast-routing
ipv6 cef
```

Disable source routing for IPv4 and IPv6

```
no ip source-route
no ipv6 source-route
```

Path MTU Discovery

Enable Path MTU Discovery on the router - this is not enabled by default for connections to the control plane (but it is enabled by default now for BGP).

```
ip tcp path-mtu-discovery
```

Exit configuration mode and save

```
end
write memory
```

Interface Configuration

Links to other Routers

Configure your interfaces according to the diagram

Notice that for the links to the Upstream we will use the Upstream's addresses, while for internal links we use internal addresses.

On CX:

```
interface GigabitEthernet1/0
 description P2P Link to BX
 ip address 100.68.X0.17 255.255.255.252
 no ip directed-broadcast
 no ip redirects
 no ip proxy-arp
 ipv6 address 2001:db8:X0:10::0/127
 ipv6 nd prefix default no-advertise
 ipv6 nd ra suppress
 no shutdown
!
```

On BX:

```
interface GigabitEthernet1/0
 description P2P Link to CX
 ip address 100.68.X0.18 255.255.255.252
 no ip directed-broadcast
 no ip redirects
 no ip proxy-arp
 ipv6 address 2001:db8:X0:10::1/127
 ipv6 nd prefix default no-advertise
 ipv6 nd ra suppress
 no shutdown
!
```

You will need to do something similar for the link from your Peering router to the Core router. Use the above configuration examples as hints.

Explanations for some of the commands used

no ip directed-broadcast

An IP directed broadcast is an IP packet whose destination address is a valid broadcast address for some IP subnet, but which originates from a node that is not itself part of that destination subnet.

Because directed broadcasts, and particularly Internet Control Message Protocol (ICMP) directed broadcasts, have been abused by malicious persons, we recommend disabling the *ip directed-broadcast* command on any intereface where directed broadcasts are not needed (probably all).

no ip proxy-arp

Proxy ARP is the technique in which one host, usually a router, answers ARP requests intended for another machine. By “faking” its identity, the router accepts responsibility for routing packets to the “real” destination. Proxy ARP can help machines on a subnet reach remote subnets without the need

to configure routing or a default gateway.

Disadvantages of proxy arp:

- It increases the impact of ARP spoofing, in which a machine claims to be another in order to intercept packets.
- It hides network misconfigurations in hosts
- Hosts will have larger ARP tables

no ip redirects

ICMP redirects can be sent to a host when the router knows that another router in the same subnet has a better path to a destination. If a hacker installs a router in the network that causes the legitimate router to learn these ilegitimate paths, the hacker's router will end up diverting legitimate traffic thanks to ICMP redirects. Thus, we recommend that you disable this feature in all your interfaces.

ipv6 nd ra suppress

IPv6 router advertisements are sent periodically by routers to inform hosts that the router is present, and to allow hosts to autoconfigure themselves using stateless autoconfiguration mechanisms. This is not necessary on point-to-point interfaces.

ipv6 nd prefix default no-advertise

This prevents the router from sending any prefixes as part of router advertisements, so the client will not auto-configure itself with a global IPv6 address. This is helpful for IOS versions where you cannot suppress solicited RA messages.

Connectivity Testing

Do some PING tests

```
C1# ping 100.68.10.18       <- B1
C1# ping 2001:DB8:10:10::1  <- B1
C1# ping 100.68.10.22       <- P1
C1# ping 2001:DB8:10:11::1  <- P1
```

and then verify the output of the following commands:

```
show arp             : Show ARP cache
show interface       : Show interface state and config
show ip interface    : Show interface IP state and config
show ipv6 neighbors  : Show IPv6 neighbours
show ipv6 interface  : Show interface state and config
```

```
show cdp neighbors   : Show neighbours seen via CDP
```

Save Configuration

Verify and save the configuration.

```
show running-config
write memory
```

Configuring IS-IS

Each IXP team will need to configure ISIS between the three routers in their AS. The core router should be straightforward to configure. It has one loopback interface, one interface connecting to the peering router, and one interface connecting to the border router. And even though we are using ethernet to connect the routers, these are only point-to-point links and IS-IS should be configured as such.

For the peering and border routers, ISIS should only be activated on the internal interface, with the loopback marked as passive. Note we do not configure the interface pointing to outside of our network as we will be using *next-hop-self* for our iBGP sessions.

Here is a configuration example for the core router in AS105:

```
key chain as105-key
 key 1
  key-string cisco
!
router isis as105
 net 49.0001.1000.6805.0002.00
 is-type level-2-only
 metric-style wide
 set-overload-bit on-startup wait-for-bgp
 authentication mode md5 level-2
 authentication key-chain lab-key level-2
 log-adjacency-changes all
 metric 100000
 passive-interface loopback0
!
 address-family ipv6
  multi-topology
  set-overload-bit on-startup wait-for-bgp
!
interface gigabitethernet 1/0
 description BackBone link to B5
 ip router isis as105
 isis metric 2
 isis network point-to-point
```

```
 ipv6 router isis as105
 isis ipv6 metric 2
!
interface gigabitethernet 2/0
 description BackBone link to P5
 ip router isis as105
 isis metric 2
 isis network point-to-point
 ipv6 router isis as105
 isis ipv6 metric 2
!
```

For the routers with connections outside the local autonomous system, we have to be very careful not to enable IS-IS on those external links. Nor do we need to carry those external link addresses in IS-IS. So do not enable IS-IS on an interface unless the router at the other end of that link is part of your own autonomous system.

Once IS-IS is working inside your autonomous system, check that you can reach all other routers inside your AS. Easiest way to test this it to ping the IPv4 and IPv6 loopback addresses from each router. Does it all work? If not, what could be wrong?

Configuring iBGP

The next step is to configure iBGP mesh between the three routers in each autonomous system. We'll use a route-reflector set up to handle this, as this is very common practice today, and full mesh iBGP does not scale, as was covered in the presentations.

We will make the core router the route reflector, as is standard practice. The border router will be a client, and the peering router will be another client. Before setting up the iBGP route reflector, we need to consider the following:

- border router connects to transit ISP, so can in theory receive the entire global routing table from the upstream (either as a default route, or as every individual announced prefix).
- peering router connects to private and public peers - the peering router should only hear routes that the peers should be able to receive. A common mistake is for peering routers in service provider backbones to carry the full Internet routing table, resulting in bandwidth hijack, misrouted traffic, and so on.

Configuring Core Router iBGP

First we set up the core router. We create two peer groups, one for the standard iBGP mesh, the other for use with peering routers. Here is an example:

```
router bgp 101
 bgp deterministic-med
 no bgp default ipv4-unicast
!
 address-family ipv4
```

```
  distance bgp 200 200 200
  neighbor ibgp-partial peer-group
  neighbor ibgp-partial description Local Routes only
  neighbor ibgp-partial remote-as 101
  neighbor ibgp-partial update-source loopback0
  neighbor ibgp-partial next-hop-self
  neighbor ibgp-partial password cisco
  neighbor ibgp-partial send-community
  neighbor ibgp-partial route-reflector-client
  neighbor ibgp-partial filter-list 10 out
  neighbor ibgp-full peer-group
  neighbor ibgp-full description Local Routes only
  neighbor ibgp-full remote-as 101
  neighbor ibgp-full update-source loopback0
  neighbor ibgp-full next-hop-self
  neighbor ibgp-full password cisco
  neighbor ibgp-full send-community
  neighbor ibgp-full route-reflector-client
  neighbor 100.68.10.1 peer-group ibgp-full
  neighbor 100.68.10.1 description iBGP with B1
  neighbor 100.68.10.3 peer-group ibgp-partial
  neighbor 100.68.10.3 description iBGP with P1
!
 address-family ipv6
 < do the same for IPv6 >
!
ip as-path access-list 10 permit ^$
```

Notice how we have set up one peer-group called ibgp-partial for use with peering routers - its only difference from the peer-group called ibgp-full is that it has one additional line only permit prefixes originated by the local AS to go to that router. So if the upstream provider sends a default route, or any prefixes from the global BGP table, they will now not make their way to the peering router. While we have used an AS-path filter here, we could also use BGP communities (much more scalable!).

Originating Prefixes

We will now originate our prefixes into iBGP. We will only do this on the core router (common practice is to originate prefixes on the core routers in a network operator's backbone, never on the peering or border routers). So, returning to the core router, we now add in network statements to cover our IPv4 and IPv6 address blocks. Here is an example for AS105:

```
router bgp 105
 address-family ipv4
  network 100.68.50.0 mask 255.255.255.0
 address-family ipv6
  network 2001:DB8:50::/48
!
ip route 100.68.50.0 255.255.255.0 Null0
ipv6 route 2001:DB8:50::/48 Null0
```

Don't forget the pull up routes for the aggregate - the network statement in Cisco IOS only tells BGP to put that address block into BGP if the match block is in the global RIB - and the simplest way to install it in the global RIB is to set up a static route pointing to the Null0 interface.

Configuring Peering Router iBGP

We now turn to the peering router, and will configure iBGP on that as well. We'll follow the same ideas as we used for the Core router, only the peering router is a route reflector client. Here is a configuration example:

```
router bgp 103
 bgp deterministic-med
 no bgp default ipv4-unicast
!
 address-family ipv4
  neighbor ibgp-rr peer-group
  neighbor ibgp-rr description iBGP with RR
  neighbor ibgp-rr remote-as 103
  neighbor ibgp-rr update-source loopback0
  neighbor ibgp-rr next-hop-self
  neighbor ibgp-rr password cisco
  neighbor ibgp-rr send-community
  neighbor 100.68.30.2 peer-group ibgp-rr
  neighbor 100.68.30.2 description iBGP with C3
!
 address-family ipv6
 < do the same for IPv6 >
!
```

Configuring Border Router iBGP

We now turn to the border router, and will configure iBGP on that as well. We'll follow the same ideas as we used for the Core router, only the Border router is a route reflector client. Here is a configuration example:

```
router bgp 105
 bgp deterministic-med
 no bgp default ipv4-unicast
!
 address-family ipv4
  neighbor ibgp-rr peer-group
  neighbor ibgp-rr description iBGP with RR
  neighbor ibgp-rr remote-as 105
  neighbor ibgp-rr update-source loopback0
  neighbor ibgp-rr next-hop-self
  neighbor ibgp-rr password cisco
  neighbor ibgp-rr send-community
  neighbor 100.68.50.2 peer-group ibgp-rr
```

```
  neighbor 100.68.50.2 description iBGP with C3
!
 address-family ipv6
 < do the same for IPv6 >
!
```

Notice that the peer-group is identical to the one used on the Peering Router.

Improving Routing Security

There are a few things we need to tidy up here before we continue with the lab.

Peering Router

The peering router is just that, a router that peers with other network operators. It does not provide any transit. The peers should only see the routes that you want them to see. We've made sure of this by putting in a route filter on the core router so that the peering router can only see locally originated routes. But it is also a good idea to null route the default route, as we will soon be distributing a default route around the AS using IS-IS. So on the peering router we now do:

```
ip route 0.0.0.0 0.0.0.0 null0
ipv6 route ::/0 null0
```

Now if any of the IXP participants point a default route to the local network, the traffic will simply be dumped in the Null interface of the peering router. Only traffic for specific destinations which available in the routing table on the IXP router will be forwarded to the rest of the network. This is a very important **network security** requirement.

Border Router

The border router connects to the upstream provider, and therefore gives us access to the whole Internet. The upstream provider will usually send us a default route by eBGP (yet to be set up). Once we hear this default route, how should it be propagated around the autonomous system?

It can be propagated using iBGP, but that tends to be non-optimal, certainly when trying to load balance between two or more transit providers, as the BGP best path is just that. If we distribute the default by the IGP instead, then at least the default route becomes the nearest exit, to the nearest border router. So we will now configure this - for example for AS102:

```
router isis as102
 default-information originate
 !
 address-family ipv6
  default-information originate
!
```

We now should be ready to proceed with the next part of the lab.

Configuring the link to the Transit Provider

The next step is to set up eBGP with the Transit Provider. The lab instructors will be running the routers for the two Transit Providers and will have already configured them.

Physical Link

Follow the guidelines in the IP Address Plan document to configure the link to the upstream. Make sure that you can ping the upstream's router using both IPv4 and IPv6 - if it doesn't work, investigate why.

Here is an example configuration for AS106:

```
interface fastethernet 0/0
 description Link to TR2
 ip address 100.122.1.10 255.255.255.252
 ipv6 address 2001:19:0:12::1/127
!
```

IS-IS

Do not configure IS-IS towards the upstream provider! They are not part of your autonomous system.

External BGP

We now configure eBGP with the upstream. Again, the configuration on the two transit routers will have already been completed by the instructors, so once configured, the eBGP session should just come up and work. Don't forget to filter what you hear from the upstream, and what you send to them. You should only accept a default route from them (they may send you more), and you should only send prefixes you originated!

Here is a configuration example for AS101:

```
ip prefix-list AS101-block permit 100.68.10.0/24
ipv6 prefix-list AS101-v6block permit 2001:DB8:10::/48
!
ip prefix-list default-route permit 0.0.0.0/0
ipv6 prefix-list default-v6route permit ::/0
!
router bgp 101
 address-family ipv4
  neighbor 100.121.1.1 remote-as 131072
```

```
  neighbor 100.121.1.1 description eBGP with TRANSIT 1
  neighbor 100.121.1.1 password cisco
  neighbor 100.121.1.1 prefix-list AS101-block out
  neighbor 100.121.1.1 prefix-list default-route in
!
 address-family ipv6
  neighbor 2001:18:0:10:: remote-as 131072
  neighbor 2001:18:0:10:: description eBGP with TRANSIT 1
  neighbor 2001:18:0:10:: password cisco
  neighbor 2001:18:0:10:: prefix-list AS101-v6block out
  neighbor 2001:18:0:10:: prefix-list default-v6route in
!
```

Once this has been configured, you should now see a default route coming from the upstream provider, and you should be able to see your aggregate being sent to your upstream. The commands to see what you are receiving are:

```
show ip bgp neigh 100.121.1.1 routes
show bgp ipv6 uni neigh 2001:18:0:10:: routes
```

and to show what you are sending:

```
show ip bgp neigh 100.121.1.1 advertised-routes
show bgp ipv6 uni neigh 2001:18:0:10:: advertised-routes
```

Confirmation

Check on the Core and Peering Router what you now see in the BGP table. Are there differences? Can you explain what they are, and why?

What does:

```
show ip bgp 0.0.0.0 0.0.0.0
show ip route 0.0.0.0 0.0.0.0
show ipv6 route ::/0
show bgp ipv6 uni ::/0
```

show you on the core router and on the peering router?

You will notice that the default route is being propagated by BGP throughout the AS.

- On the core router, it will complain of a **RIB failure** when you look at the BGP table, because IS-IS has a default route with a lower protocol distance.
- On the border router, it will complain of a **RIB failure** when you look at the BGP table, because there is a static default route to the Null interface.

While there is nothing wrong with a **RIB failure**, we can just remove the default from being propagated by the iBGP. To do this, we go back to the eBGP session, look for the default, tag it with the *no-advertise* community, and then the border router will no longer announce the default by iBGP.

Here is a configuration example for AS102:

```
route-map tag-default permit 10
 match ip address prefix-list default-route
 set community no-advertise
!
route-map tag default permit 20
!
route-map tag-v6default permit 10
 match ipv6 address prefix-list default-v6route
 set community no-advertise
!
route-map tag-v6default permit 20
!
router bgp 102
 address-family ipv4
  neighbor 100.121.1.5 route-map tag-default in
 address-family ipv6
  neighbor 2001:18:0:11:: route-map tag-v6default in
!
```

To confirm, has the default route now disappeared from the BGP table on the Core and Peering routers? If not, did you remember to do route-refresh after you applied the above configuration?

Testing

Once this has been completed, test the connectivity. Can you reach the other groups in the class? You should be able to ping all the IPv4 and IPv6 loopbacks across the whole classroom.

Can you see the Internet too? The lab has IPv4 connectivity to the Internet - check that this works by trying a few pings or trace routes to well known destinations (e.g. to 8.8.8.8).

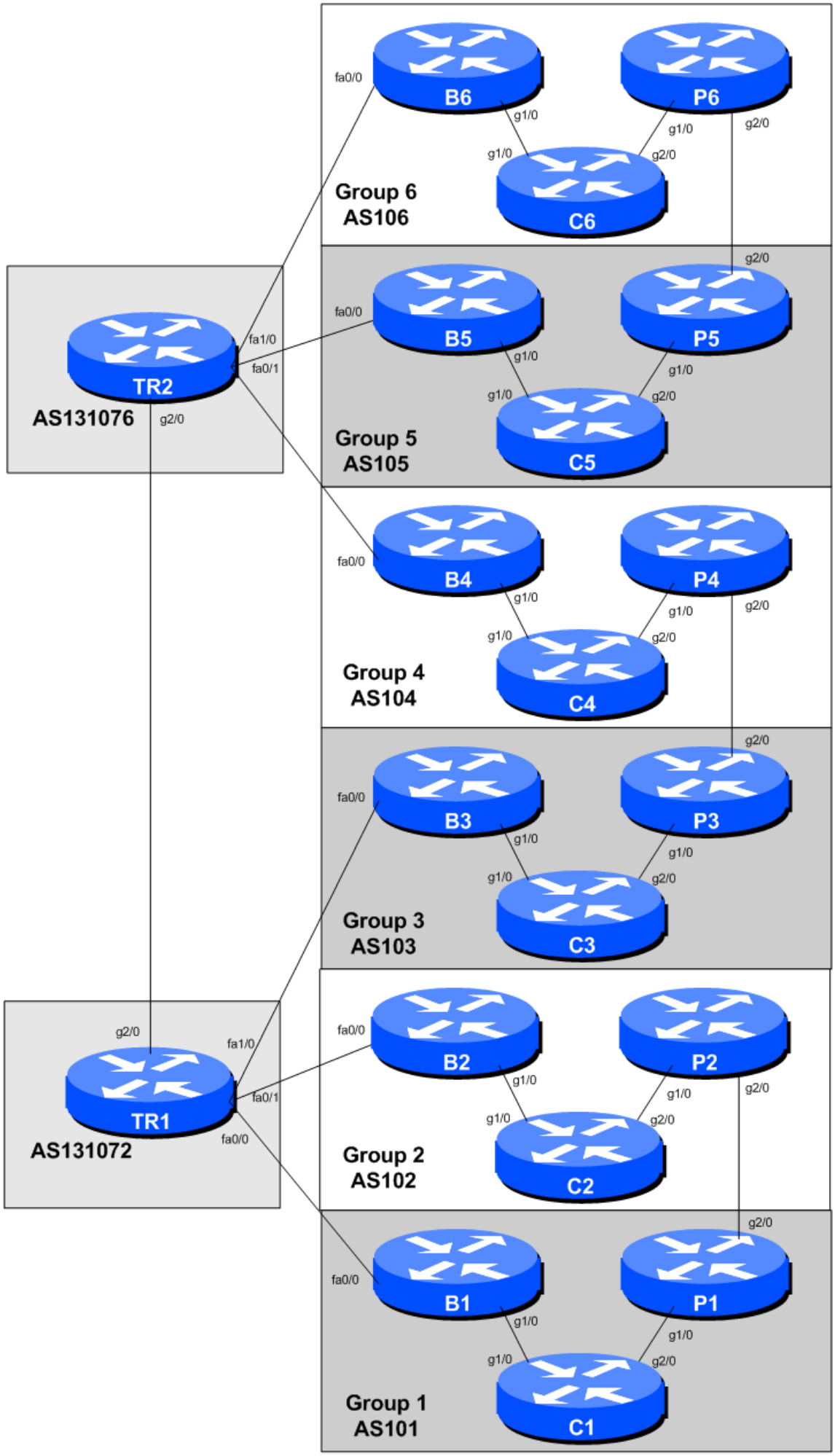
STOP AND WAIT HERE

Peering & IXP Lab (Part 2: Private Peering)

The purpose of this part of the lab is to set up private peering between adjacent autonomous systems. In our case, AS101 will private peer with AS102, AS103 with AS104, and AS105 with AS106. We will use what we have learned so far to ensure that the adjacent ASNs only hear the routes they are meant to hear - a true private peering.

Lab Topology

The lab topology has been enhanced according to the diagram below. We simply add in the peering links mentioned in the preamble.


B6
P6
C6
Group 6
AS106
B5
P5
C5
Group 5
AS105
TR2
AS131076
B4
P4
C4
Group 4
AS104
B3
P3
C3
Group 3
AS103
TR1
AS131072
B2
P2
C2
Group 2
AS102
B1
P1
C1
Group 1
AS101
fa0/0
fa1/0
fa0/1
g1/0
g2/0

Configuring the Private Peering Links

Each group should now configure the private peering links as shown in the diagram.

Physical Link

Agree on which addresses should be used for the point to point links. Typically one group will contribute the IPv4 /30 and IPv6 /127 on the link.

```
interface GigabitEthernet 2/0
 description Link to Group 2 Peering Router
 ip address 100.68.10.25 255.255.255.252
 ipv6 address 2001:DB8:10:12::/127
!
```

Once the interfaces have been configured make sure that the links can be pinged on both IPv4 and IPv6 endpoints.

Configuring IS-IS

Do not configure IS-IS towards your private peer! They are not part of your autonomous system.

Configuring eBGP

We now configure eBGP with the private peer. Don't forget to filter what you hear from the private peer, and what you send to them. You should only accept their address blocks from them (they may send you more by mistake!), and you should only send prefixes you originated!

Here is a configuration example for AS103 - note that we are reusing some configuration we have set up earlier:

```
ip prefix-list AS103-block permit 100.68.30.0/24
ipv6 prefix-list AS103-v6block permit 2001:DB8:30::/48
!
ip prefix-list AS104-block permit 100.68.40.0/24
ipv6 prefix-list AS104-v6block permit 2001:DB8:40::/48
!
router bgp 103
 address-family ipv4
  neighbor 100.68.30.26 remote-as 104
  neighbor 100.68.30.26 description eBGP with AS104
  neighbor 100.68.30.26 password cisco
  neighbor 100.68.30.26 prefix-list AS103-block out
  neighbor 100.68.30.26 prefix-list AS104-block in
```

```
!
 address-family ipv6
  neighbor 2001:DB8:30:12::1 remote-as 104
  neighbor 2001:DB8:30:12::1 description eBGP with AS104
  neighbor 2001:DB8:30:12::1 password cisco
  neighbor 2001:DB8:30:12::1 prefix-list AS103-v6block out
  neighbor 2001:DB8:30:12::1 prefix-list AS104-v6block in
!
```

Once this has been configured, you should now see your private peer originated routes coming from them, and you should be able to see your aggregate being sent to your private peer. The commands to see what you are receiving are:

```
show ip bgp neigh 100.68.30.26 routes
show bgp ipv6 uni neigh 2001:DB8:30:12::1 routes
```

and to show what you are sending:

```
show ip bgp neigh 100.68.30.26 advertised-routes
show bgp ipv6 uni neigh 2001:DB8:30:12::1 advertised-routes
```

Confirmation

Check on the Border, Core and Peering Router what you now see in the BGP table.

What is the best path to your private peer? What does trace route tell you?

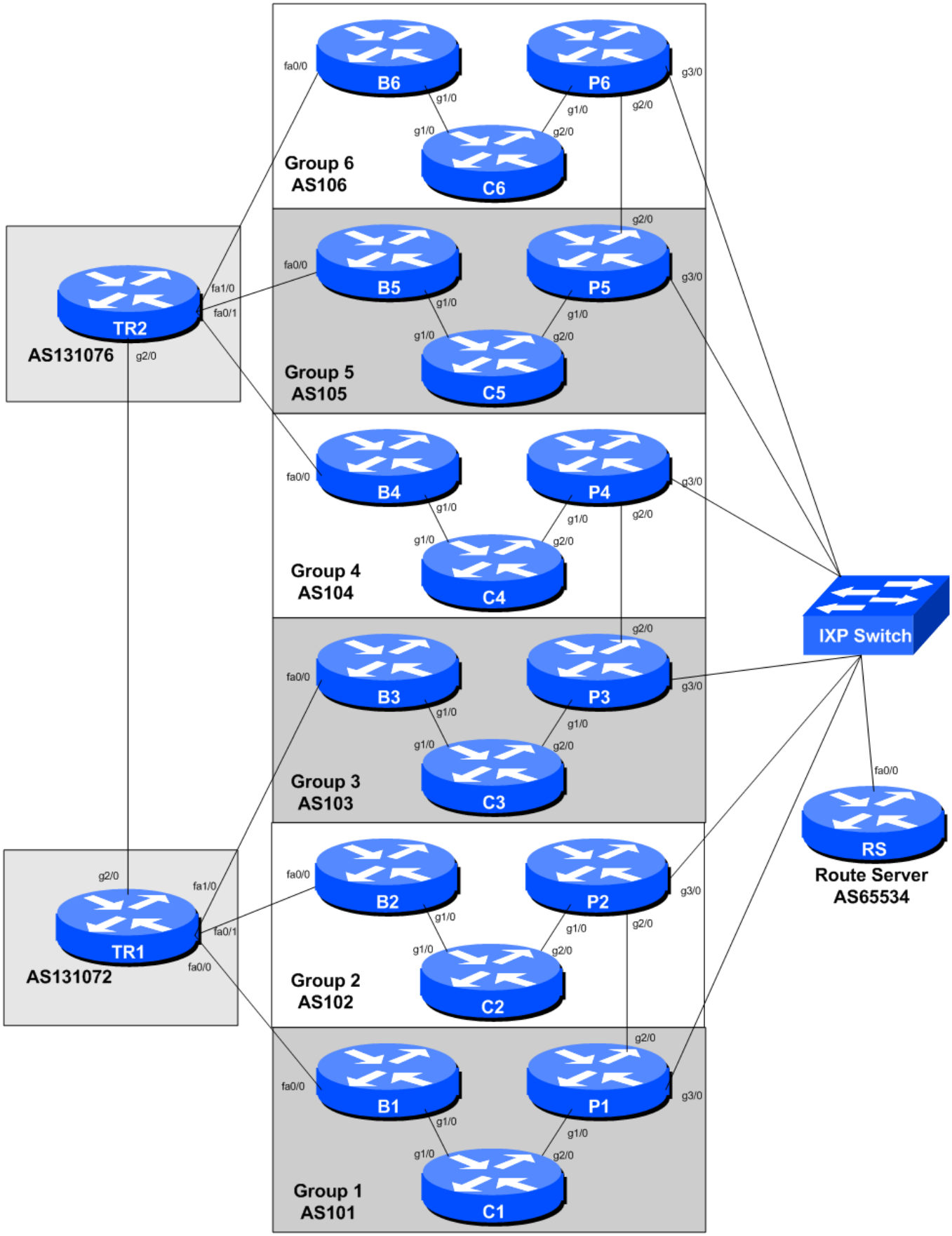
Hopefully you will see that the best path to your private peer will be via the private peering link. And the routes to the rest of the class will be via your upstream provider's default route. If this is not the case, you will need to start doing some troubleshooting!

Peering & IXP Lab (Part 3: IXP)

The purpose of this part of the lab is introduce an Internet Exchange Point into our lab. IXPs are a very important if not critical component of today's Internet architecture, and it is vitally important to ensure the correct configuration so that network operators gain maximum advantage from their participation at an IXP.

Lab Topology

The lab topology has been further enhanced according to the diagram below. An Internet Exchange Point and its router server are now installed.



Configuring the IXP links

Each group should now configure their link to the IXP according to the above diagram.

Physical Link

Consult the Address Plan document for the address space used by the IXP. Following the document, configure the interface on the router accordingly.

```
interface GigabitEthernet 3/0
 description Link to IXP
 ip address 100.127.1.5 255.255.255.0
 ipv6 address 2001:DB8:FFFF:1::5/64
!
```

Note the subnet masks - this time the ethernet is **NOT** a point-to-point link but a shared LAN media. Once the interfaces have been configured see if you can ping any of the other groups on their IXP addresses (both IPv4 and IPv6). Are you able to ping the Route Server too?

Configuring IS-IS

Do not configure IS-IS towards any IXP peer! They are not part of your autonomous system.

However, so that traceroutes across the IXP do not break, we might wish to carry the IXP LAN address block within our IS-IS (**not** iBGP). To do this, we simply mark the IXP facing interface as passive in the IS-IS configuration. Here is an example for AS106:

```
router isis as106
 passive-interface GigabitEthernet 3/0
```

If you recall from the IS-IS presentation, this will tell IS-IS to announce the subnet attached to this interface.

Now all routers in your AS will see the IXP LAN address - check from your Core and Border routers, just to make sure.

Configuring eBGP

We now configure eBGP with the Exchange Point's Route Server (we might add in bi-lateral BGP peering later, but for now we will just peer with the Route Server).

The Route Server sits in AS 65534 - this is a private AS, and is not visible on the public Internet. In fact, we don't want this AS to be visible inside our own AS either, and that's one of the unique features of a Route Server - it does not add its AS into the AS path when distributing prefixes to its eBGP neighbours.

DO NOT forget to filter what you hear from the Route Server, and what you send to the Route Server. You should only accept their address blocks from the other IXP participants (they may send you more by mistake!), and you should only send prefixes you originated!

Here is a configuration example for AS104 - note that we are reusing some configuration we have set up earlier:

```
ip prefix-list AS104-block permit 100.68.40.0/24
ipv6 prefix-list AS104-v6block permit 2001:DB8:40::/48
!
ip prefix-list IXP-RS permit 100.68.10.0/24
ip prefix-list IXP-RS permit 100.68.20.0/24
ip prefix-list IXP-RS permit 100.68.30.0/24
ip prefix-list IXP-RS permit 100.68.40.0/24
ip prefix-list IXP-RS permit 100.68.50.0/24
ip prefix-list IXP-RS permit 100.68.60.0/24
!
ipv6 prefix-list IXP-v6RS permit 2001:DB8:10::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:20::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:30::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:40::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:50::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:60::/48
!
router bgp 104
 address-family ipv4
  neighbor 100.127.1.254 remote-as 65534
  neighbor 100.127.1.254 description eBGP with IXP RS
  neighbor 100.127.1.254 password ixp-rs
  neighbor 100.127.1.254 prefix-list AS104-block out
  neighbor 100.127.1.254 prefix-list IXP-RS in
!
 address-family ipv6
  neighbor 2001:DB8:FFFF:1::FE remote-as 65534
  neighbor 2001:DB8:FFFF:1::FE description eBGP with IXP RS
  neighbor 2001:DB8:FFFF:1::FE password ixp-rs
  neighbor 2001:DB8:FFFF:1::FE prefix-list AS104-v6block out
  neighbor 2001:DB8:FFFF:1::FE prefix-list IXP-v6RS in
!
```

Once this has been configured, has the BGP session with the Route Server established? If not, why not? What do the router logs tell you?

You will notice from the logs that the router is complaining about a BGP peer AS not being in the announced AS path - this is Cisco IOS protecting against improper BGP announcements, as according to the BGP RFC, the AS PATH of the neighbouring AS must appear as the adjacent AS in the AS PATH. And if you recall from early in the notes, that was a special feature of the Route Server: its AS does not appear in the path.

So we need to turn this safety check off in IOS:

```
router bgp 105
 no bgp enforce-first-as
```

and once this has been done you will now see that the eBGP session with the Route Server will have been established.

What do you now see in the BGP table?

What about the routes between you and your private peer that you set up earlier? Which is the best path now? Through the IXP, or over the private peering link?

Private Peering link versus IXP Peering

We are now going to deal with the issue where we see two paths between us and our private peer. One is via our private peering link, the other is via our peering with them across the IXP.

In day to day Internet operations, network operators prioritise links according to the value they attach to them - the list goes something like this:

Type of Link	Priority (local preference)
Private Peer	200
IXP Bi-lateral	180
IXP RS	170
Customer	120
(default)	100
Local Transit	70
Global Transit	50

Obviously there will be many variations on this theme, but the principle remains the same. Peering links have no operational cost, so are highly preferred over links which have an operational cost (transit). Private peering links are preferred over IXP links as the former is brokered directly with the partner, while the IXP links are via a third party infrastructure. It is not physically possible to peer privately with every operator, and this is the function that the IXP then provides (as was covered in the course presentations).

We will now attach local preference to the routes we hear from our private peer and from the IXP Route Server, according to the table above.

To do this we will create two route-maps, one for the private peer, the other for the RS peering. Here is an example for AS101:

```
route-map private-peer-in permit 10
 set local-preference 200
!
route-map IXP-RS-peer-in permit 10
 set local-preference 170
!
router bgp 101
 address-family ipv4
  neighbor 100.68.10.26 route-map private-peer-in in
  neighbor 100.127.1.254 route-map IXP-RS-peer-in in
!
 address-family ipv6
```

```
  neighbor 2001:DB8:10:12::1 route-map private-peer-in in
  neighbor 2001:DB8:FFFF:1::FE route-map IXP-RS-peer-in in
!
```

Once this is configured, do a route-refresh inbound on the two peering, and now you should now see the local preferences attached to the routes from the IXP and from the private peer. What has happened now?

Confirmation

Check on the Border, Core and Peering Router what you now see in the BGP table.

What is the best path to your private peer? What does trace route tell you?

Hopefully you will see that the best path to your private peer will be via the private peering link. And the routes to the rest of the class will be via the Internet Exchange Point. The only traffic going via the Upstream Provider now will be traffic out to the Internet itself. If this is not the case, you will need to start doing some troubleshooting!

Peering with Route Server or Peering Directly with IXP peers?

The final part of this workshop lab is to investigate how to set up peering directly with IXP Peers. There are three types of peering policies adopted by network operators today:

Peering Policy	Description
Open	Network Operator will peer with allcomers, no questions asked. At an IXP this means they will peer with the Route Server.
Selective	Network Operator will usually peer with most operators, but enters a conversation with the peering partner first before establishing the link. At an IXP this means they will set up a direct peering across the IX fabric.
Restricted	Network Operator will choose who they peer with, under very stringent conditions. They rarely show up at an IXP, and if they do, peering will be directly across the IX fabric.

We have set up our peering at the moment to assume that all groups have an *Open* peering policy. But what if they had a *Selective* policy instead? How do we configure that?

Bi-lateral peering across IX Fabric

What we will do now is modify our eBGP at the IXP so that we also include a direct eBGP session with our IXP peers. We'll set this up to supplement the Route Server (or we could simply remove the Route Server peering once we have peered with all members of the IXP).

Here is a configuration example for AS102, peering with AS103 (again noting that we are re-using configuration created earlier on):

```
ip prefix-list AS102-block permit 100.68.20.0/24
ipv6 prefix-list AS102-v6block permit 2001:DB8:20::/48
!
ip prefix-list AS103-block permit 100.68.30.0/24
ipv6 prefix-list AS103-v6block permit 2001:DB8:30::/48
!
route-map IXP-bilateral-in permit 10
 set local-preference 170
!
router bgp 102
 address-family ipv4
  neighbor 100.127.1.3 remote-as 103
  neighbor 100.127.1.3 description eBGP with AS103
  neighbor 100.127.1.3 password cisco
  neighbor 100.127.1.3 prefix-list AS102-block out
  neighbor 100.127.1.3 prefix-list AS103-block in
  neighnor 100.127.1.3 route-map IXP-bilateral-in in
!
 address-family ipv6
  neighbor 2001:DB8:FFFF:1::3 remote-as 103
  neighbor 2001:DB8:FFFF:1::3 description eBGP with AS103
  neighbor 2001:DB8:FFFF:1::3 password cisco
  neighbor 2001:DB8:FFFF:1::3 prefix-list AS102-v6block out
  neighbor 2001:DB8:FFFF:1::3 prefix-list AS103-v6block in
  neighbor 2001:DB8:FFFF:1::3 route-map IXP-bilateral-in in
!
```

What do you see now?

You should see two paths to your IXP peers - they are almost indistinguishable apart from the router-id of the neighbouring router - one will be of the route-server, the other will be of the direct eBGP peer.

Repeat the above for all the members of the IXP.

Once you are peering with all of the members of the IXP, you can remove your peering with the Route Server if you wish:

- if you have decided that your group has a Selective Peering Policy, you may not want to peer with the Route Server, as then you won't be able to select the new peers (unless the Route Server operator offers a more sophisticated configuration than we have here in the lab)
- Many operators bi-lateral peer across the IXP but retain the Route Server peering as back as well - for redundancy purposes.

Peering & IXP Lab (Part 4: Communities)

The purpose of this part of the lab is introduce BGP Communities into our lab. We are doing this for

ease of management of the various prefixes we are carrying in our iBGP.

Lab Topology

The lab topology is unchanged from the topology used in the previous section.

BGP Communities

We will be updating our BGP configurations to use BGP Communities. Following the presentation, we specifically want to tag prefixes we learn from our private peers, our IXP peers, and our own prefixes we introduce into our iBGP. This way we can manage what we announce to all external networks simply by applying community filters, removing the need to maintain prefix-lists, and therefore simplifying the management of the network configuration.

Community Policy

The communities we are going to adopt are as follows:

Prefix Type	Community	Description
Our aggregate	*AS*:1000	Our assigned address block(s)
Subnets of our aggregate	*AS*:1001	Any subnets we carve out of our address block(s)
Customer independent addresses	*AS*:1005	Any address space customers are assigned independently of what they receive from us
Private Peer addresses	*AS*:1100	Any addresses our private peers send to us
IXP Peer addresses	*AS*:1200	Any addresses our IXP peers send to us

Each group will now set up these community definitions on all three of their routers. Here is an example for AS102:

```
ip community-list 1 permit 102:1000
ip community-list 2 permit 102:1001
ip community-list 3 permit 102:1005
ip community-list 4 permit 102:1100
ip community-list 5 permit 102:1200
```

We also need to tell Cisco IOS to use the industry format for BGP communities (as described in RFC1998) rather than the specification standard (as described in RFC1997). The latter represents communities as a 32-bit integer; the former represents communities as two 16-bit integers separate by a colon.

```
ip bgp-community new-format
```

Setting Community on our prefixes

We introduced our aggregate for our AS on the Core router using the BGP network statement. We are now going to augment this network statement so that the community is set on the prefix we introduce. Here is an example:

```
route-map set-aggregate-community permit 10
 set community 103:1000
!
router bgp 103
 address-family ipv4
  network 100.68.30.0 mask 255.255.255.0 route-map set-aggregate-community
 address-family ipv6
  network 2001:DB8:30::/48 route-map set-aggregate-community
!
```

Once this is done, check the BGP table. Run the:

```
show ip bgp community
```

and

```
show bgp ipv6 unicast community
```

commands. What do you see now?

All being well, you should see your IPv4 and IPv6 aggregates displayed in the output. This command shows you all prefixes which have a BGP community set on them. Then show the actual BGP table entry - you will now see an extra line in the output, indicating which community has been set. It should be in the format *AS*:1000.

Setting Community prefix learned from the private peer

We now set the community on the prefix we learn from our private peer. As with the previous example, we look for the prefix they are sending us, and tag it with the appropriate community. Here is an example for AS104 - note that we are adding to the existing configuration (which includes the *private-peer-in* route-map and the inbound prefix-list for the private peer):

```
route-map private-peer-in permit 10
 set community 104:1100
!
router bgp 104
 address-family ipv4
  neighbor 100.68.30.25 route-map private-peer-in in
!
 address-family ipv6
  neighbor 2001:DB8:30:12:: route-map private-peer-in in
!
```

Once the route-map and the BGP neighbour configuration has been updated, remember to do the inbound route-refresh. Cisco IOS does not do route-refreshes automatically.

Check the communities listed in the BGP tables as you did earlier. Do you now see the private peer prefix with a community set on it?

Setting Community prefix learned from the IXP peers

We now set the community on the prefix we learn from our IXP peers via the Route Server (or bi-laterals peers if you have chosen not to peer with the Route Server). As with the previous example, we look for the prefix they are sending us, and tag it with the appropriate community. Here is an example for AS105 - note again that we are adding to the existing configuration (which includes the *ixp-peer-in* route-map and the inbound prefix-list for the IXP peer):

```
route-map IXP-peer-in permit 10
 set community 105:1200
!
router bgp 105
 address-family ipv4
  neighbor 100.127.1.254 route-map IXP-peer-in in
!
 address-family ipv6
  neighbor 2001:DB8:FFFF:1::FE route-map IXP-peer-in in
!
```

Once the route-map and the BGP neighbour configuration has been updated, remember to do the inbound route-refresh.

Check the communities listed in the BGP tables as you did earlier. Do you now see the IXP peers' prefixes with a community set on them?

Outbound BGP Announcements Filtered by Community

The final step in this exercise is to update our outbound BGP filters to use BGP communities rather than the prefix-lists we set up earlier. Our inbound filters will remain using prefix-lists, as we want to control what our neighbours send to us. However, for outbound announcements, prefix-lists do not scale, as every time we need to announce a new prefix, we have to update the filters - which gets very tedious to maintain as the network grows larger.

The current BGP configuration should be looking something like this example shown for AS102's border router:

```
ip community-list 1 permit 102:1000
ip community-list 2 permit 102:1001
ip community-list 3 permit 102:1005
ip community-list 4 permit 102:1100
ip community-list 5 permit 102:1200
!
```

```
ip prefix-list AS102-block permit 100.68.20.0/24
ipv6 prefix-list AS102-v6block permit 2001:DB8:20::/48
!
ip prefix-list default-route permit 0.0.0.0/0
ipv6 prefix-list default-v6route permit ::/0
!
router bgp 102
 address-family ipv4
  neighbor 100.121.1.5 remote-as 131072
  neighbor 100.121.1.5 description eBGP with TRANSIT 1
  neighbor 100.121.1.5 password cisco
  neighbor 100.121.1.5 prefix-list AS102-block out
  neighbor 100.121.1.5 prefix-list default-route in
!
 address-family ipv6
  neighbor 2001:18:0:11:: remote-as 131072
  neighbor 2001:18:0:11:: description eBGP with TRANSIT 1
  neighbor 2001:18:0:11:: password cisco
  neighbor 2001:18:0:11:: prefix-list AS102-v6block out
  neighbor 2001:18:0:11:: prefix-list default-v6route in
!
```

and peering router:

```
ip community-list 1 permit 102:1000
ip community-list 2 permit 102:1001
ip community-list 3 permit 102:1005
ip community-list 4 permit 102:1100
ip community-list 5 permit 102:1200
!
ip prefix-list AS102-block permit 100.68.20.0/24
ipv6 prefix-list AS102-v6block permit 2001:DB8:20::/48
!
ip prefix-list AS101-block permit 100.68.10.0/24
ipv6 prefix-list AS101-v6block permit 2001:DB8:10::/48
!
ip prefix-list IXP-RS permit 100.68.10.0/24
ip prefix-list IXP-RS permit 100.68.20.0/24
ip prefix-list IXP-RS permit 100.68.30.0/24
ip prefix-list IXP-RS permit 100.68.40.0/24
ip prefix-list IXP-RS permit 100.68.50.0/24
ip prefix-list IXP-RS permit 100.68.60.0/24
!
ipv6 prefix-list IXP-v6RS permit 2001:DB8:10::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:20::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:30::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:40::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:50::/48
ipv6 prefix-list IXP-v6RS permit 2001:DB8:60::/48
!
route-map IXP-peer permit 10
```

```
 set local-preference 180
 set community 102:1200
!
route-map private-peer permit 10
 set local-preference 200
 set community 102:1100
!
router bgp 102
 address-family ipv4
  neighbor 100.68.10.25 remote-as 101
  neighbor 100.68.10.25 description eBGP with AS101
  neighbor 100.68.10.25 password cisco
  neighbor 100.68.10.25 prefix-list AS102-block out
  neighbor 100.68.10.25 prefix-list AS101-block in
  neighbor 100.68.10.25 route-map private-peer-in in
  neighbor 100.127.1.254 remote-as 65534
  neighbor 100.127.1.254 description eBGP with IXP RS
  neighbor 100.127.1.254 password ixp-rs
  neighbor 100.127.1.254 prefix-list AS102-block out
  neighbor 100.127.1.254 prefix-list IXP-RS in
  neighbor 100.127.1.254 route-map IXP-peer-in in
!
 address-family ipv6
  neighbor 2001:DB8:10:12:: remote-as 101
  neighbor 2001:DB8:10:12:: description eBGP with AS101
  neighbor 2001:DB8:10:12:: password cisco
  neighbor 2001:DB8:10:12:: prefix-list AS102-v6block out
  neighbor 2001:DB8:10:12:: prefix-list AS101-v6block in
  neighbor 2001:DB8:10:12:: route-map private-peer-in in
  neighbor 2001:DB8:FFFF:1::FE remote-as 65534
  neighbor 2001:DB8:FFFF:1::FE description eBGP with IXP RS
  neighbor 2001:DB8:FFFF:1::FE password ixp-rs
  neighbor 2001:DB8:FFFF:1::FE prefix-list AS102-v6block out
  neighbor 2001:DB8:FFFF:1::FE prefix-list IXP-v6RS in
  neighbor 2001:DB8:FFFF:1::FE route-map IXP-peer-in in
!
```

Our goal now is to replace the outbound prefix lists with a route-map which matches the community settings.

In this example for AS102, we want to replace the prefix-lists called “AS102-block” and “AS102-v6block”. Each group should now replace their equivalent prefix-list:

```
ip prefix-list AS102-block permit 100.68.20.0/24
ipv6 prefix-list AS102-v6block permit 2001:DB8:20::/48
```

with route-maps for each of our three types of BGP neighbours. For our private peer we have:

```
route-map private-peer-out permit 10
 match community 1
```

For our IXP peers we have:

```
route-map IXP-peer-out permit 10
 match community 1
```

For our upstream we have:

```
route-map upstream-out permit 10
 match community 1
```

and then replace the relevant entries in the BGP configuration on the border router:

```
no ip prefix-list AS102-block permit 100.68.20.0/24
no ipv6 prefix-list AS102-v6block permit 2001:DB8:20::/48
!
router bgp 102
 address-family ipv4
  no neighbor 100.121.1.5 prefix-list AS102-block out
  neighbor 100.121.1.5 route-map upstream-out out
!
 address-family ipv6
  no neighbor 2001:18:0:11:: prefix-list AS102-v6block out
  neighbor 2001:18:0:11:: route-map upstream-out out
!
```

and the peering router:

```
no ip prefix-list AS102-block permit 100.68.20.0/24
no ipv6 prefix-list AS102-v6block permit 2001:DB8:20::/48
!
router bgp 102
 address-family ipv4
  no neighbor 100.68.10.25 prefix-list AS102-block out
  neighbor 100.68.10.25 route-map private-peer-out out
  no neighbor 100.127.1.254 prefix-list AS102-block out
  neighbor 100.127.1.254 route-map IXP-peer-out out
!
 address-family ipv6
  no neighbor 2001:DB8:10:12:: prefix-list AS102-v6block out
  neighbor 2001:DB8:10:12:: route-map private-peer-out out
  no neighbor 2001:DB8:FFFF:1::FE prefix-list AS102-v6block out
  neighbor 2001:DB8:FFFF:1::FE route-map IXP-peer-out out
!
```

Note that we used the same route-map for both IPv4 and IPv6 BGP Peerings. This is because the route-map has no address family specific configuration in it - it is simply matching a BGP attribute that applies to both IPv4 and IPv6 address families.

Implement the route-refresh and then observe the changes to the BGP table - there should not be any change at all! (Note that Cisco IOS does not send communities to eBGP neighbours unless that feature is turned on - we don’t need to do this here, as we are not using communities to signal anything to

our neighbour networks - we are simply using them for our internal operational convenience.)

Now in future if we need to add any prefixes to our address announcements, we simply tag them with the correct community (ASN:1000) and that prefix will be automatically announced to all peers.

NOTE: In the real operational Internet it is quite unlikely that anyone would use the identical outbound route-map for both transit, public and private peers which is why we gave each one different names here. It is more than likely that traffic engineering needs have to be considered, and different subnets being sent to private and public peers, depending on the needs of customers. Which would mean the use of a route-map per peer (the more common case).

Conclusion

This lab has shown the differences between peering and transit, and the value that operators place on each type. Transit is last resort as it has significant operational cost. Peering is highly desirable, and this lab has shown the differences between private peering and IXP peering configurations.

Appendix 1 - Route Server Configuration

This appendix shows the configuration of the route server used for this workshop. It is Cisco IOS based - most route servers today run either on BIRD or a modified version of Quagga.

```
interface FastEthernet0/0
 description IXP LAN
 ip address 100.127.1.254 255.255.255.0
 ipv6 address 2001:DB8:FFFF:1::FE/64
!
router bgp 65534
 bgp log-neighbor-changes
 bgp deterministic-med
 no bgp default ipv4-unicast
 neighbor ixp-peers peer-group
 neighbor ixp-peers password ixp-rs
 neighbor v6ixp-peers peer-group
 neighbor v6ixp-peers password ixp-rs
 neighbor 100.127.1.1 remote-as 101
 neighbor 100.127.1.1 peer-group ixp-peers
 neighbor 100.127.1.1 description AS101 peer
 neighbor 100.127.1.2 remote-as 102
 neighbor 100.127.1.2 peer-group ixp-peers
 neighbor 100.127.1.2 description AS102 peer
 neighbor 100.127.1.3 remote-as 103
```

```
 neighbor 100.127.1.3 peer-group ixp-peers
 neighbor 100.127.1.3 description AS103 peer
 neighbor 100.127.1.4 remote-as 104
 neighbor 100.127.1.4 peer-group ixp-peers
 neighbor 100.127.1.4 description AS104 peer
 neighbor 100.127.1.5 remote-as 105
 neighbor 100.127.1.5 peer-group ixp-peers
 neighbor 100.127.1.5 description AS105 peer
 neighbor 100.127.1.6 remote-as 106
 neighbor 100.127.1.6 peer-group ixp-peers
 neighbor 100.127.1.6 description AS106 peer
 neighbor 2001:DB8:FFFF:1::1 remote-as 101
 neighbor 2001:DB8:FFFF:1::1 peer-group v6ixp-peers
 neighbor 2001:DB8:FFFF:1::1 description AS101 peer
 neighbor 2001:DB8:FFFF:1::2 remote-as 102
 neighbor 2001:DB8:FFFF:1::2 peer-group v6ixp-peers
 neighbor 2001:DB8:FFFF:1::2 description AS102 peer
 neighbor 2001:DB8:FFFF:1::3 remote-as 103
 neighbor 2001:DB8:FFFF:1::3 peer-group v6ixp-peers
 neighbor 2001:DB8:FFFF:1::3 description AS103 peer
 neighbor 2001:DB8:FFFF:1::4 remote-as 104
 neighbor 2001:DB8:FFFF:1::4 peer-group v6ixp-peers
 neighbor 2001:DB8:FFFF:1::4 description AS104 peer
 neighbor 2001:DB8:FFFF:1::5 remote-as 105
 neighbor 2001:DB8:FFFF:1::5 peer-group v6ixp-peers
 neighbor 2001:DB8:FFFF:1::5 description AS105 peer
 neighbor 2001:DB8:FFFF:1::6 remote-as 106
 neighbor 2001:DB8:FFFF:1::6 peer-group v6ixp-peers
 neighbor 2001:DB8:FFFF:1::6 description AS106 peer
!
 address-family ipv4
  neighbor ixp-peers route-server-client
  neighbor 100.127.1.1 activate
  neighbor 100.127.1.2 activate
  neighbor 100.127.1.3 activate
  neighbor 100.127.1.4 activate
  neighbor 100.127.1.5 activate
  neighbor 100.127.1.6 activate
  distance bgp 200 200 200
 exit-address-family
!
 address-family ipv6
  neighbor v6ixp-peers route-server-client
  neighbor 2001:DB8:FFFF:1::1 activate
  neighbor 2001:DB8:FFFF:1::2 activate
  neighbor 2001:DB8:FFFF:1::3 activate
  neighbor 2001:DB8:FFFF:1::4 activate
  neighbor 2001:DB8:FFFF:1::5 activate
  neighbor 2001:DB8:FFFF:1::6 activate
  distance bgp 200 200 200
 exit-address-family
```

```
!
ip route 0.0.0.0 0.0.0.0 Null0
ipv6 route ::/0 Null0
```

Appendix 2 - Transit Router Configuration

This appendix shows the configuration used for the TR1 router in this lab. The TR2 router has a very similar configuration.

```
interface FastEthernet0/0
 description Link to AS101
 ip address 100.121.1.1 255.255.255.252
 ipv6 address 2001:18:0:10::/127
!
interface FastEthernet0/1
 description Link to AS102
 ip address 100.121.1.5 255.255.255.252
 ipv6 address 2001:18:0:11::/127
!
interface FastEthernet1/0
 description Link to AS103
 ip address 100.121.1.9 255.255.255.252
 ipv6 address 2001:18:0:12::/127
!
interface GigabitEthernet2/0
 description Link to TR2 (and to the world)
 ip address 100.121.0.1 255.255.255.252 secondary
 ip address 10.10.0.201 255.255.255.0
 ipv6 address 2001:18::/127
!
router bgp 131072
 bgp log-neighbor-changes
 bgp deterministic-med
 no bgp default ipv4-unicast
 neighbor 2001:18::1 remote-as 131076
 neighbor 2001:18::1 description eBGP with TR2
 neighbor 2001:18::1 password 7 01100F175804
 neighbor 2001:18:0:10::1 remote-as 101
 neighbor 2001:18:0:10::1 password 7 045802150C2E
 neighbor 2001:18:0:11::1 remote-as 102
 neighbor 2001:18:0:11::1 password 7 02050D480809
 neighbor 2001:18:0:12::1 remote-as 103
 neighbor 2001:18:0:12::1 password 7 13061E010803
 neighbor 100.121.0.2 remote-as 131076
 neighbor 100.121.0.2 description eBGP with TR2
 neighbor 100.121.0.2 password 7 1511021F0725
 neighbor 100.121.1.2 remote-as 101
 neighbor 100.121.1.2 password 7 05080F1C2243
```

```
 neighbor 100.121.1.6 remote-as 102
 neighbor 100.121.1.6 password 7 13061E010803
 neighbor 100.121.1.10 remote-as 103
 neighbor 100.121.1.10 password 7 00071A150754
 !
 address-family ipv4
  network 100.121.0.0 mask 255.255.0.0
  neighbor 100.121.0.2 activate
  neighbor 100.121.1.2 activate
  neighbor 100.121.1.2 default-originate
  neighbor 100.121.1.6 activate
  neighbor 100.121.1.6 default-originate
  neighbor 100.121.1.10 activate
  neighbor 100.121.1.10 default-originate
  distance bgp 200 200 200
 exit-address-family
 !
 address-family ipv6
  network 2001:18::/32
  neighbor 2001:18::1 activate
  neighbor 2001:18:0:10::1 activate
  neighbor 2001:18:0:10::1 default-originate
  neighbor 2001:18:0:11::1 activate
  neighbor 2001:18:0:11::1 default-originate
  neighbor 2001:18:0:12::1 activate
  neighbor 2001:18:0:12::1 default-originate
  distance bgp 200 200 200
 exit-address-family
!
ip route 0.0.0.0 0.0.0.0 10.10.0.254
ip route 100.121.0.0 255.255.0.0 Null0
!
ipv6 route 2001:18::/32 Null0
```