.

Project 1c

By Sandeep Kadagathur Vadiraj, 8998-3394-34

Question 9 - (30 points) Configure your local-preference as well as the exportation rules to implement your customer/provider and peer/peer business relationships with neighboring ASes. You should implement (1) no-valley routing and (2) prefer-customer routing.

No-valley routing specifies that you should not advertise routes learned from one provider/peer to another provider/peer, and prefer-customer routing specifies that if you can reach a destination through multiple AS-paths, you should always prefer the paths that go through your customers first, then peers, then providers. If you have paths through multiple neighbors of the same class, prefer-customer does not specify which to choose.

Hint: To implement no-valley routing, we advise you to use BGP communities to keep track of where the routes have been learned, and set up exportation rules based on BGP communities. You can use other BGP attributes as well if you like. However, it is a requirement that all best routes that would not cause a valley should be announced. Similarly, all the routes being announced shouldn't result in a valley. Also, your solution should not depend on the actual prefixes announced by your neighbors. To test that your BGP policies work correctly, you can use the management VM to launch traceroute from another AS. For example, you can launch a traceroute from one of your peers towards another peer, and verify that your AS does not provider transit service between these two peers. In the report, describe (1) how you use BGP attributes to implement no-valley routing and prefer-customer routing, (2) the actual router configuration you made, (3) evidences to support that you have implemented these two policies correctly (4) explanation about how to interpret the evidence.

(1) No-valley: Used bgp community list to tag and block the tags on export to another peering AS.

    Prefer-customer: Used local-preferences to prefer customer as it is an outbound policy. Used the same route-map and applied the local preference with highest to lowest preference in order customers, peers and providers.

(2) Actual configuration: (the 1st and 2nd line commands are inside router bgp 5 command)

```
NEWY
neighbor 179.24.17.1 remote-as 4
 neighbor 179.24.17.1 route-map no_export_PP out
ip prefix-list 15_network seq 10 permit 15.0.0.0/8
ip prefix-list 15_network seq 15 permit 10.0.0.0/8
ip prefix-list 15_network seq 20 permit 11.0.0.0/8
ip prefix-list 15_network seq 25 permit 2.0.0.0/8
ip prefix-list 15_network seq 30 permit 14.0.0.0/8

route-map no_export_PP permit 10
```

```
 match ip address prefix-list 15_network
 set community no-export
!
route-map no_export_PP permit 20
!
```

Explanation:

BGP No-Export ensures prefixes to not get advertised outside of an AS. In the topology provided to us I am tier 2 and am connected to the two ASs in particular 4 and and 15. NEWY is connected to AS 4 and SEAT to AS5.

Using the prefix list I am grouping the network to /8 or all of the ASs subnets. Now I apply a prefix list in the route-map and set the community to no-export which will actually block all the outbound traffic from our AS. No-export is by default. Ref

http://www.nongnu.org/quagga/docs/docs-multi/BGP-Communities-Attribute.html

Similarly I applied the same configuration on peers and customers. So as to not become transit for any other ASs.

SEAT

```
neighbor 179.24.18.1 remote-as 15
 neighbor 179.24.18.1 route-map no_export_PP out
ip prefix-list 15_network seq 10 permit 4.0.0.0/8
ip prefix-list 15_network seq 15 permit 10.0.0.0/8
ip prefix-list 15_network seq 20 permit 11.0.0.0/8
ip prefix-list 15_network seq 25 permit 2.0.0.0/8
ip prefix-list 15_network seq 30 permit 14.0.0.0/8


route-map no_export_PP permit 10
 match ip address prefix-list 15_network
 set community no-export
!
route-map no_export_PP permit 20
!
```

WASH

```
neighbor 179.24.27.1 remote-as 15
 neighbor 179.24.27.1 route-map no_export_PP out
ip prefix-list 15_network seq 10 permit 4.0.0.0/8
ip prefix-list 15_network seq 15 permit 10.0.0.0/8
ip prefix-list 15_network seq 20 permit 15.0.0.0/8
ip prefix-list 15_network seq 25 permit 2.0.0.0/8
ip prefix-list 15_network seq 30 permit 14.0.0.0/8


route-map no_export_PP permit 10
 match ip address prefix-list 15_network
 set community no-export
!
route-map no_export_PP permit 20
!
```

SALT

```
neighbor 179.24.30.1 remote-as 15
 neighbor 179.24.30.1 route-map no_export_PP out
ip prefix-list 15_network seq 10 permit 4.0.0.0/8
ip prefix-list 15_network seq 15 permit 11.0.0.0/8
ip prefix-list 15_network seq 20 permit 15.0.0.0/8
ip prefix-list 15_network seq 25 permit 2.0.0.0/8
ip prefix-list 15_network seq 30 permit 14.0.0.0/8


route-map no_export_PP permit 10
 match ip address prefix-list 15_network
 set community no-export
!
route-map no_export_PP permit 20
!
```

KANS

```
neighbor 179.24.31.1 remote-as 15
 neighbor 179.24.31.1 route-map no_export_PP out
ip prefix-list 15_network seq 10 permit 4.0.0.0/8
ip prefix-list 15_network seq 15 permit 10.0.0.0/8
ip prefix-list 15_network seq 20 permit 11.0.0.0/8
ip prefix-list 15_network seq 25 permit 2.0.0.0/8
ip prefix-list 15_network seq 30 permit 15.0.0.0/8


route-map no_export_PP permit 10
 match ip address prefix-list 15_network
 set community no-export
!
route-map no_export_PP permit 20
!
```

LOSA

```
neighbor 179.24.32.1 remote-as 15
 neighbor 179.24.32.1 route-map no_export_PP out
ip prefix-list 15_network seq 10 permit 4.0.0.0/8
ip prefix-list 15_network seq 15 permit 10.0.0.0/8
ip prefix-list 15_network seq 20 permit 11.0.0.0/8
ip prefix-list 15_network seq 25 permit 15.0.0.0/8
ip prefix-list 15_network seq 30 permit 14.0.0.0/8


route-map no_export_PP permit 10
 match ip address prefix-list 15_network
 set community no-export
!
route-map no_export_PP permit 20
!
```

For prefer customer:

route-map LOCALPREF permit 10

set local-preference 300 ! For customer I changed it to 700 and for peers 300 and for providers 100

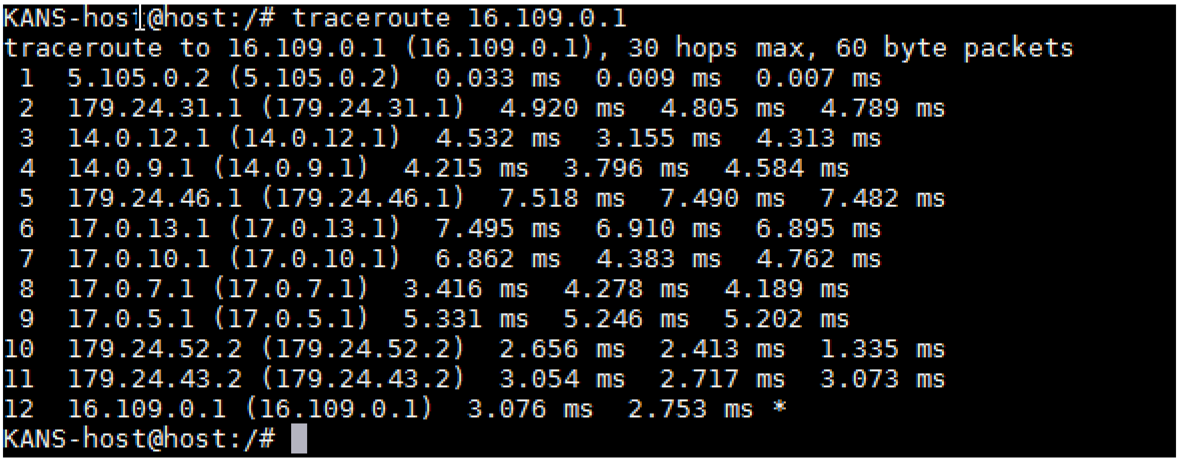
router bgp 5

neighbor 179.24.X.Y route-map LOCALPREF in

The above shows the configuration of typical local preference set on routers namely NEWY, SEAT, KANS, SALT, WASH, LOSA.

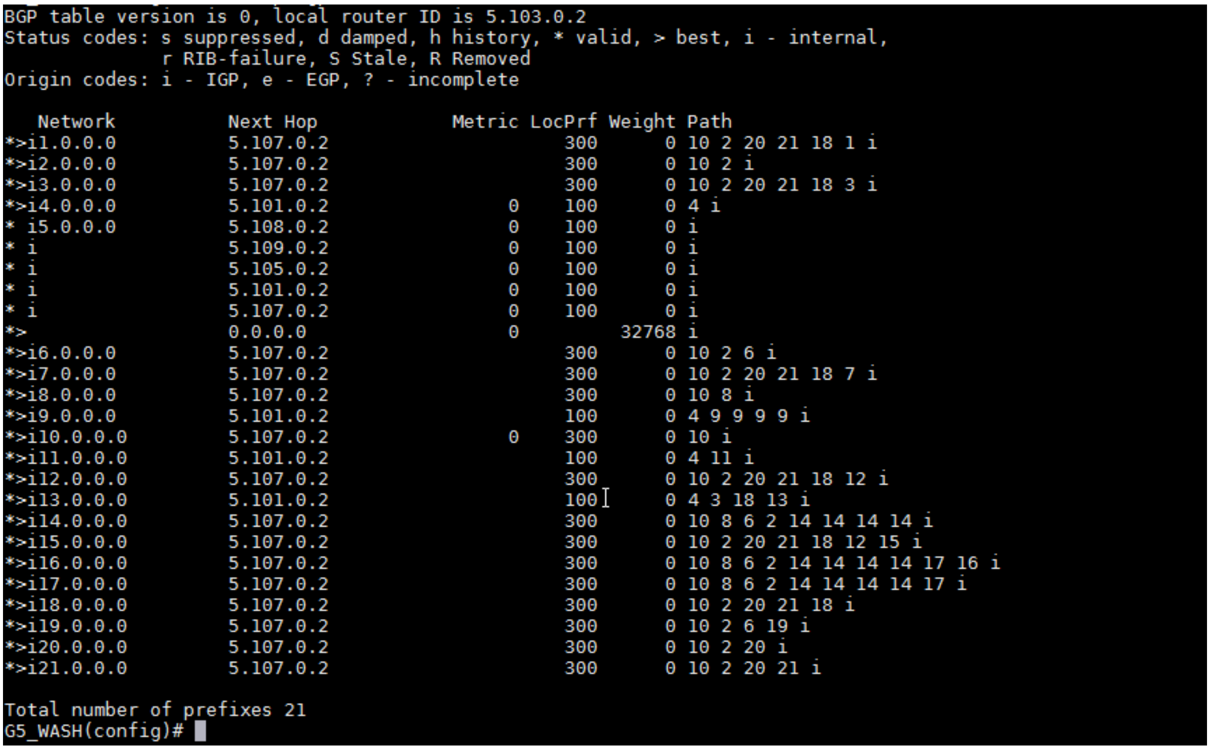
(3) Evidence

Prefer customer Kans advertised with local preference of 700



```
KANS-host@host:/# traceroute 16.109.0.1
traceroute to 16.109.0.1 (16.109.0.1), 30 hops max, 60 byte packets
 1  5.105.0.2 (5.105.0.2)  0.033 ms  0.009 ms  0.007 ms
 2  179.24.31.1 (179.24.31.1)  4.920 ms  4.805 ms  4.789 ms
 3  14.0.12.1 (14.0.12.1)  4.532 ms  3.155 ms  4.313 ms
 4  14.0.9.1 (14.0.9.1)  4.215 ms  3.796 ms  4.584 ms
 5  179.24.46.1 (179.24.46.1)  7.518 ms  7.490 ms  7.482 ms
 6  17.0.13.1 (17.0.13.1)  7.495 ms  6.910 ms  6.895 ms
 7  17.0.10.1 (17.0.10.1)  6.862 ms  4.383 ms  4.762 ms
 8  17.0.7.1 (17.0.7.1)  3.416 ms  4.278 ms  4.189 ms
 9  17.0.5.1 (17.0.5.1)  5.331 ms  5.246 ms  5.202 ms
10  179.24.52.2 (179.24.52.2)  2.656 ms  2.413 ms  1.335 ms
11  179.24.43.2 (179.24.43.2)  3.054 ms  2.717 ms  3.073 ms
12  16.109.0.1 (16.109.0.1)  3.076 ms  2.753 ms *
KANS-host@host:/#
```

Prefer peer: Show ip bgp table shows the peer advertised routers when I change the local preference on the customers.



```
BGP table version is 0, local router ID is 5.103.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i1.0.0.0          5.107.0.2                     300      0 10 2 20 21 18 1 i
*>i2.0.0.0          5.107.0.2                     300      0 10 2 i
*>i3.0.0.0          5.107.0.2                     300      0 10 2 20 21 18 3 i
*>i4.0.0.0          5.101.0.2                0    100      0 4 i
* i5.0.0.0          5.108.0.2                0    100      0 i
* i                 5.109.0.2                0    100      0 i
* i                 5.105.0.2                0    100      0 i
* i                 5.101.0.2                0    100      0 i
* i                 5.107.0.2                0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i6.0.0.0          5.107.0.2                     300      0 10 2 6 i
*>i7.0.0.0          5.107.0.2                     300      0 10 2 20 21 18 7 i
*>i8.0.0.0          5.107.0.2                     300      0 10 8 i
*>i9.0.0.0          5.101.0.2                     100      0 4 9 9 9 9 i
*>i10.0.0.0         5.107.0.2                0    300      0 10 i
*>i11.0.0.0         5.101.0.2                     100      0 4 11 i
*>i12.0.0.0         5.107.0.2                     300      0 10 2 20 21 18 12 i
*>i13.0.0.0         5.101.0.2                     100      0 4 3 18 13 i
*>i14.0.0.0         5.107.0.2                     300      0 10 8 6 2 14 14 14 14 i
*>i15.0.0.0         5.107.0.2                     300      0 10 2 20 21 18 12 15 i
*>i16.0.0.0         5.107.0.2                     300      0 10 8 6 2 14 14 14 14 17 16 i
*>i17.0.0.0         5.107.0.2                     300      0 10 8 6 2 14 14 14 14 17 i
*>i18.0.0.0         5.107.0.2                     300      0 10 2 20 21 18 i
*>i19.0.0.0         5.107.0.2                     300      0 10 2 6 19 i
*>i20.0.0.0         5.107.0.2                     300      0 10 2 20 i
*>i21.0.0.0         5.107.0.2                     300      0 10 2 20 21 i

Total number of prefixes 21
G5_WASH(config)#
```

(4) I chose these two examples because they give a clear example of the network status:
In the above traceroute notice how the exiting is happening at the 179.24.31.1 which is through the KANS ebgp and not through the provider.
In the bgp table you can see all the routes with local preference 300 instead of 700 which is customer.

Question 10 - (20 points) For this question, you will configure BGP policies in order to conduct traffic engineering on the outbound traffic. To optimize your resources, you want to split your outbound traffic between at least two of your peers (or providers). For tier-1 networks: For some prefixes that are reachable through more than one peers, force your routers to send part of the traffic to one of your peers, and another part of the traffic to another peer. Regardless of which router you use, traffic to the same destination IP address should go through the same peer. For other networks: For some prefixes that are reachable through more than one providers, force your routers to send part of the traffic to one of your providers, and another part of the traffic to another provider. Regardless of which router you use, traffic to the same destination IP address should go through the same provider. Hint: AS-path length can change outside the control of your AS, so the way you split your traffic between the two peers (or providers) should not be based on the AS-path length--it should work regardless of changes other ASes make. Launch traceroutes from your AS towards different destinations (reachable from both peer/providers) and verify that your configuration works well.

In your report, please describe (1) the prefixes you choose to conduct traffic engineering on, (2) how you plan to split traffic to these prefixes across peers/providers, (3) how you used BGP attributes to implement such outbound traffic engineering, (4) the actual router configuration you made, (5) evidences to support that you have implemented these two policies correctly, and (6) explanation about how to interpret the evidence. You should only use one single method to conduct outbound traffic engineering, even if you apply that method consistently to multiple prefixes.

(1) Prefix I chose to do the traffic engineering on:

I am splitting the traffic for half of my Tier 1 through NEWY and other half through my SEAT

So 1.0.0.0/8 till 8.0.0.0/8 is routed via NEWY and 9.0.0.0/8 till 15.0.0.0/8 is routed via SEAT.

(2)(3)(4)So the plan is to split the traffic based on the IP address using access-list and redistribute it in the route-map

For SEAT

```
access-list 10 permit 1.0.0.0 247.255.255.255
route-map no_export_PP permit 15
 match ip address 10
 set weight 200
router bgp 5
neighbor 179.24.X.Y route-map no_export_PP out
```

In the similar way on NEWY

the access-list will be

```
access-list 10 permit 9.0.0.0 0.255.255.255
```

(5)

```
G5_SALT(config)# do traceroute 15.101.0.1

traceroute to 15.101.0.1 (15.101.0.1), 30 hops max, 60 byte packets

 1  5.0.9.1 (5.0.9.1)  0.030 ms  0.008 ms  0.006 ms

 2  5.0.6.1 (5.0.6.1)  0.020 ms  0.010 ms  0.010 ms

 3  5.0.1.1 (5.0.1.1)  0.021 ms  0.013 ms  0.011 ms

 4  179.24.17.1 (179.24.17.1)  2.485 ms  2.442 ms  2.439 ms

 5  4.0.10.1 (4.0.10.1)  2.373 ms  2.378 ms  2.353 ms

 6  4.0.7.1 (4.0.7.1)  2.378 ms  0.611 ms  0.481 ms

 7  4.0.5.1 (4.0.5.1)  0.668 ms  0.554 ms  1.860 ms

 8  4.0.4.1 (4.0.4.1)  2.785 ms  2.746 ms  2.741 ms

 9  15.0.2.1 (15.0.2.1)  2.743 ms  1.066 ms  0.890 ms

10  15.0.1.1 (15.0.1.1)  1.227 ms  0.840 ms  0.932 ms

11  15.101.0.1 (15.101.0.1)  1.351 ms  1.979 ms  1.860 ms
```

So we can see the exiting is happening on the 4th hop at NEWY

(6)

This shows that the access-list splitting for the NEWY happens as expected by our policy since the exiting is happening at our intended place.

Question 11 - (20 points) For this question, you will configure BGP policies in order to conduct traffic engineering on the inbound traffic destined to your own prefix. For tier-1 networks: Configure BGP policies such that the inbound traffic destined to your own network through peering links to come in priority from NEWY, then SEAT, then SALT or WASH. For other networks: Configure BGP policies such that the inbound traffic destined to your own network through provider links uses the provider connected to NEWY in priority.

You can use BGP attributes to express your route preference to other networks. However, do not setup any exportation rules to hide links. For example, only advertising paths in NEWY is not a proper solution. You can test your configuration by launching traceroutes using the management VM. Consider launching traceroutes from a AS which is reachable only via your peers (if you are a tier-1) or your providers (if you are not a Tier1) towards an IP belonging to your prefix, and see if the traceroute does use the correct peer or provider. For example, if you are AS 10, one way to test the configuration is by issuing traceroutes from AS 4, since it connects to both providers of AS 10, but not to AS 10 directly.

Hint: It is possible that other ASes implements outbound traffic engineering as required in question 10. In these cases, traceroutes from them may also be influenced by their outbound traffic engineering in addition to your inbound traffic engineering. However, keep in mind that other ASes will not conduct traffic engineering to their customers. Also, the ASes that are controlled by TAs (AS 4, AS 10, AS 18, AS 19, AS 20) will not implement any traffic engineering. You should always be able to find at least one AS to issue traceroutes from without worrying about its outbound traffic engineering policy. In your report, please describe

(1) how you used BGP attributes to implement such inbound traffic engineering,

(2) the actual router configuration you made,

(3) evidences to support that you have implemented these two policies correctly

(4) explanation about how to interpret the evidence. For (3), if you use traceroutes from different ASes to test your configurations, please make sure to include which ASes you choose, why did you choose them, and the traceroute results

Task 11:

1. BGP AS-path prepending was used to advertise SEAT router with more AS-path length than the others.
2. The commands and explanation:

Commands used on AS SEAT:

```
neighbor 179.24.18.1 remote-as 15
```

The above command adds the neighbor to a particular AS.

```
neighbor 179.24.18.1 route-map AS4_prefixes out
```

The above command adds the route-map on the outbound for a given neighbor

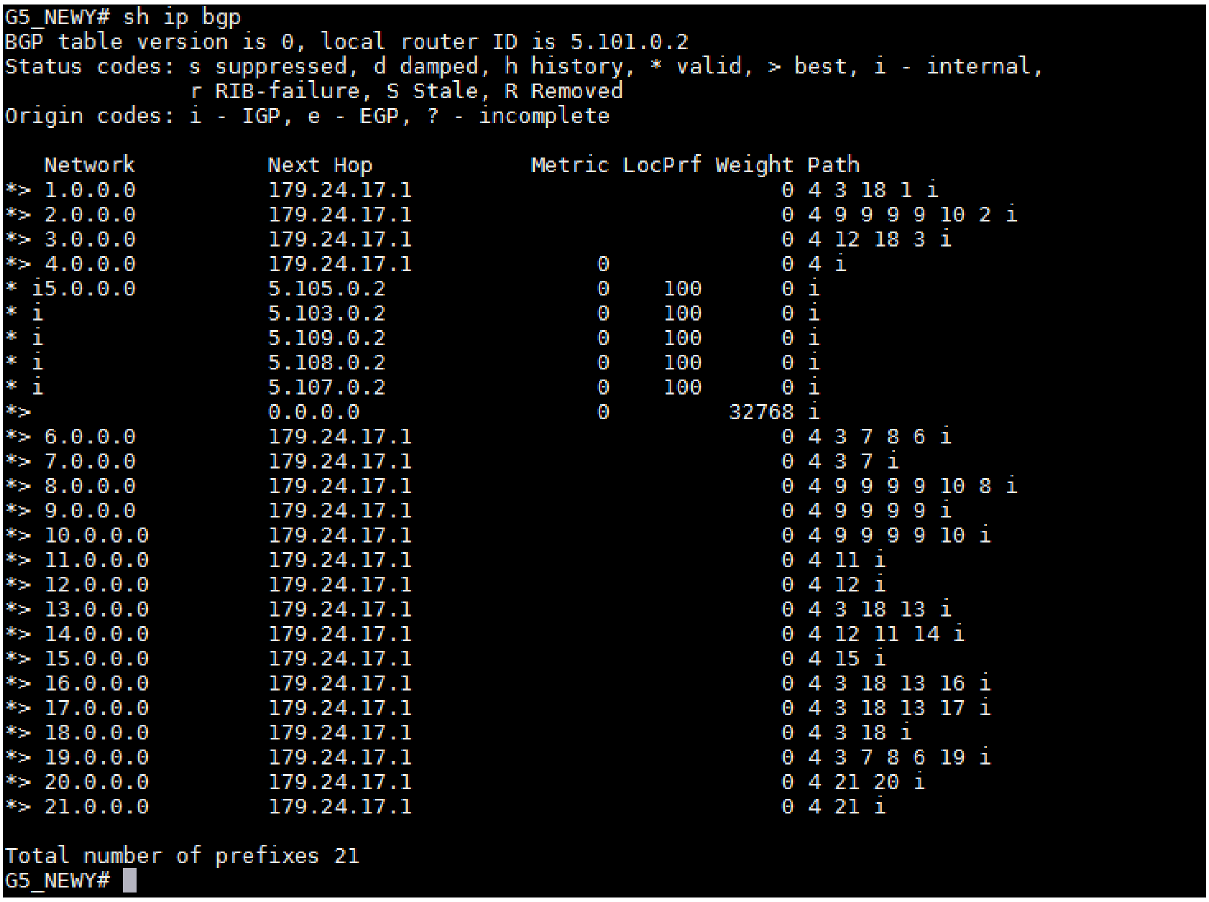
```
route-map AS4_prefixes permit 5
```

The first rule in the route map asks the router to advertise the pre-pended AS path to the neighbors, so that this route would be less preferred as compared to NEWY router. The appended AS path will appear as “5 5 5” one more than what is pre-pended in the bgp table.

```
 set as-path prepend 5 5
!
```

(3) Evidences:

Prefer NEWY to SEAT task. AS path is prepended on the SEAT router, this makes the router less preferred as compared to the NEWY and look at the routing table on NEWY. It originates all the prefixes. (Notice: I had to shut down my peering links and customer interfaces as the paths were coming in via those at higher local preference then the providers due to the prefer customer policy implemented)

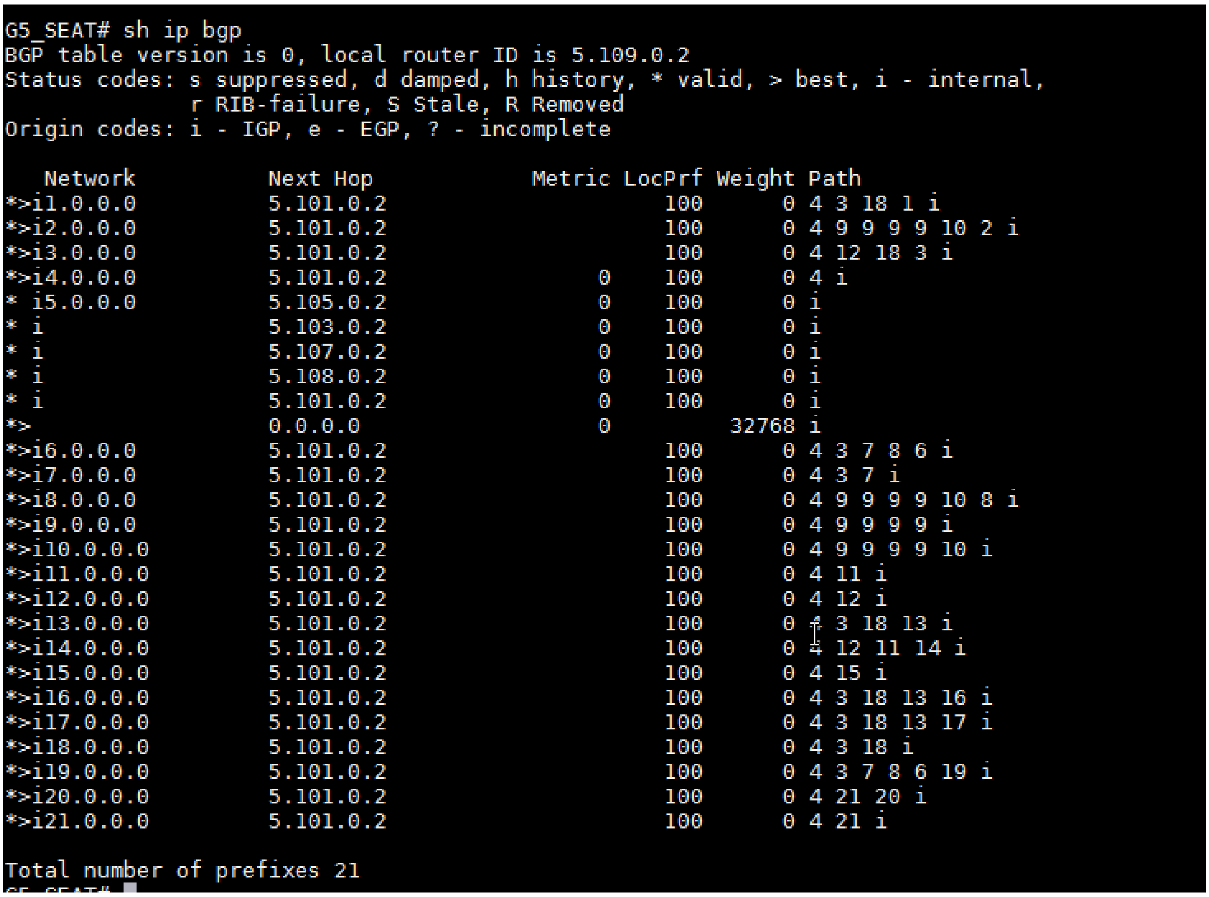


```
G5_NEWY# sh ip bgp
BGP table version is 0, local router ID is 5.101.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.0.0.0          179.24.17.1                            0 4 3 18 1 i
*> 2.0.0.0          179.24.17.1                            0 4 9 9 9 9 10 2 i
*> 3.0.0.0          179.24.17.1                            0 4 12 18 3 i
*> 4.0.0.0          179.24.17.1              0             0 4 i
* i5.0.0.0          5.105.0.2                0    100      0 i
* i                 5.103.0.2                0    100      0 i
* i                 5.109.0.2                0    100      0 i
* i                 5.108.0.2                0    100      0 i
* i                 5.107.0.2                0    100      0 i
*>                  0.0.0.0                  0         32768 i
*> 6.0.0.0          179.24.17.1                            0 4 3 7 8 6 i
*> 7.0.0.0          179.24.17.1                            0 4 3 7 i
*> 8.0.0.0          179.24.17.1                            0 4 9 9 9 9 10 8 i
*> 9.0.0.0          179.24.17.1                            0 4 9 9 9 9 i
*> 10.0.0.0         179.24.17.1                            0 4 9 9 9 9 10 i
*> 11.0.0.0         179.24.17.1                            0 4 11 i
*> 12.0.0.0         179.24.17.1                            0 4 12 i
*> 13.0.0.0         179.24.17.1                            0 4 3 18 13 i
*> 14.0.0.0         179.24.17.1                            0 4 12 11 14 i
*> 15.0.0.0         179.24.17.1                            0 4 15 i
*> 16.0.0.0         179.24.17.1                            0 4 3 18 13 16 i
*> 17.0.0.0         179.24.17.1                            0 4 3 18 13 17 i
*> 18.0.0.0         179.24.17.1                            0 4 3 18 i
*> 19.0.0.0         179.24.17.1                            0 4 3 7 8 6 19 i
*> 20.0.0.0         179.24.17.1                            0 4 21 20 i
*> 21.0.0.0         179.24.17.1                            0 4 21 i

Total number of prefixes 21
G5_NEWY# 
```

Let us check if it is true on the other end at SEAT



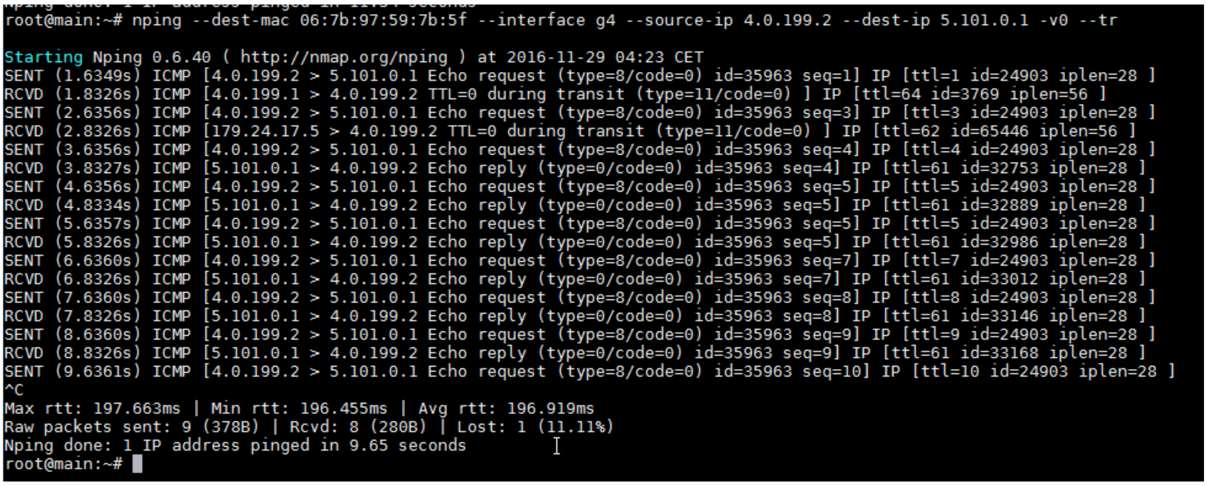
```
G5_SEAT# sh ip bgp
BGP table version is 0, local router ID is 5.109.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i1.0.0.0          5.101.0.2                     100      0 4 3 18 1 i
*>i2.0.0.0          5.101.0.2                     100      0 4 9 9 9 9 10 2 i
*>i3.0.0.0          5.101.0.2                     100      0 4 12 18 3 i
*>i4.0.0.0          5.101.0.2                0    100      0 4 i
* i5.0.0.0          5.105.0.2                0    100      0 i
* i                 5.103.0.2                0    100      0 i
* i                 5.107.0.2                0    100      0 i
* i                 5.108.0.2                0    100      0 i
* i                 5.101.0.2                0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i6.0.0.0          5.101.0.2                     100      0 4 3 7 8 6 i
*>i7.0.0.0          5.101.0.2                     100      0 4 3 7 i
*>i8.0.0.0          5.101.0.2                     100      0 4 9 9 9 9 10 8 i
*>i9.0.0.0          5.101.0.2                     100      0 4 9 9 9 9 i
*>i10.0.0.0         5.101.0.2                     100      0 4 9 9 9 9 10 i
*>i11.0.0.0         5.101.0.2                     100      0 4 11 i
*>i12.0.0.0         5.101.0.2                     100      0 4 12 i
*>i13.0.0.0         5.101.0.2                     100      0 4 3 18 13 i
*>i14.0.0.0         5.101.0.2                     100      0 4 12 11 14 i
*>i15.0.0.0         5.101.0.2                     100      0 4 15 i
*>i16.0.0.0         5.101.0.2                     100      0 4 3 18 13 16 i
*>i17.0.0.0         5.101.0.2                     100      0 4 3 18 13 17 i
*>i18.0.0.0         5.101.0.2                     100      0 4 3 18 i
*>i19.0.0.0         5.101.0.2                     100      0 4 3 7 8 6 19 i
*>i20.0.0.0         5.101.0.2                     100      0 4 21 20 i
*>i21.0.0.0         5.101.0.2                     100      0 4 21 i

Total number of prefixes 21
```

We can see from SEAT that next hop is 5.101.0.2 which is the IP address of the NEWY confirming our assertion that NEWY path is preferred by all the ASs due to shorter AS path as all other BGP attributes such as Local Preference and Weight are the same.

Also, to verify the AS path pre-pending happens I shutdown the NEWY router this makes all the traffic to be originated only by the SEAT router (Notice: that all the peers and customers have implemented no-valley policies or if it is TA controlled ASs at that time those bgp interfaces were shut down). So, my neighbouring AS15 shared his bgp table show casing my appended AS path. Here's the appended AS path:

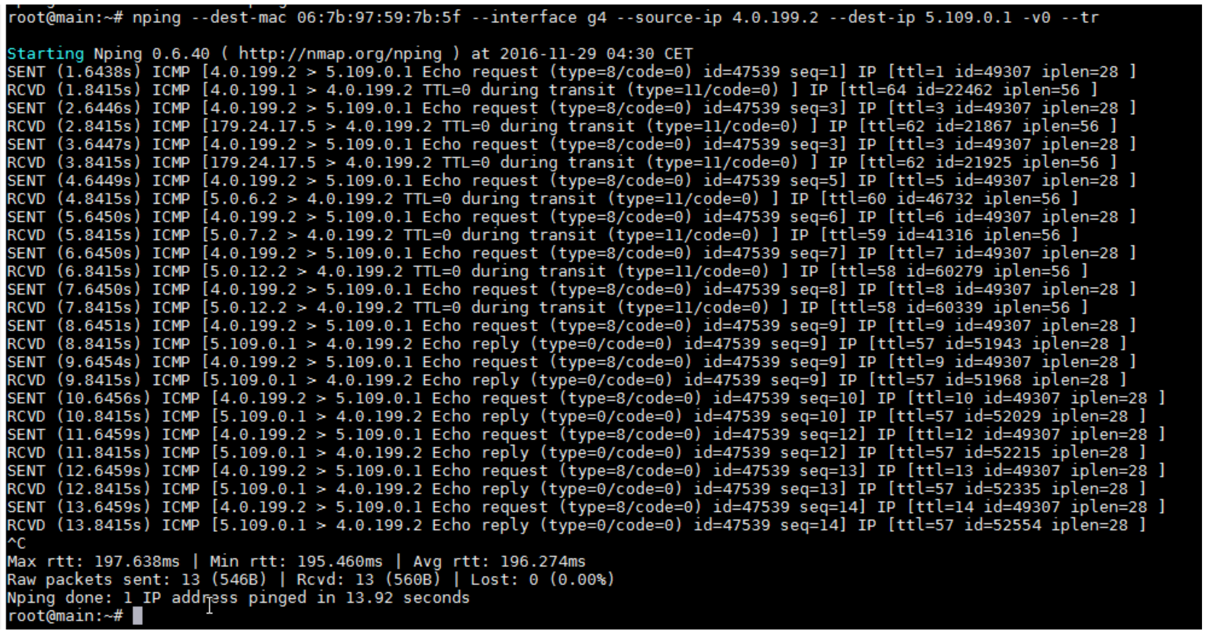


```
root@main:~# nping --dest-mac 06:7b:97:59:7b:5f --interface g4 --source-ip 4.0.199.2 --dest-ip 5.101.0.1 -v0 --tr

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2016-11-29 04:23 CET
SENT (1.6349s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=1] IP [ttl=1 id=24903 iplen=28 ]
RCVD (1.8326s) ICMP [4.0.199.1 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=64 id=3769 iplen=56 ]
SENT (2.6356s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=3] IP [ttl=3 id=24903 iplen=28 ]
RCVD (2.8326s) ICMP [179.24.17.5 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=65446 iplen=56 ]
SENT (3.6356s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=4] IP [ttl=4 id=24903 iplen=28 ]
RCVD (3.8327s) ICMP [5.101.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=35963 seq=4] IP [ttl=61 id=32753 iplen=28 ]
SENT (4.6356s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=5] IP [ttl=5 id=24903 iplen=28 ]
RCVD (4.8334s) ICMP [5.101.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=35963 seq=5] IP [ttl=61 id=32889 iplen=28 ]
SENT (5.6357s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=5] IP [ttl=5 id=24903 iplen=28 ]
RCVD (5.8326s) ICMP [5.101.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=35963 seq=5] IP [ttl=61 id=32986 iplen=28 ]
SENT (6.6360s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=7] IP [ttl=7 id=24903 iplen=28 ]
RCVD (6.8326s) ICMP [5.101.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=35963 seq=7] IP [ttl=61 id=33012 iplen=28 ]
SENT (7.6360s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=8] IP [ttl=8 id=24903 iplen=28 ]
RCVD (7.8326s) ICMP [5.101.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=35963 seq=8] IP [ttl=61 id=33146 iplen=28 ]
SENT (8.6360s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=9] IP [ttl=9 id=24903 iplen=28 ]
RCVD (8.8326s) ICMP [5.101.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=35963 seq=9] IP [ttl=61 id=33168 iplen=28 ]
SENT (9.6361s) ICMP [4.0.199.2 > 5.101.0.1 Echo request (type=8/code=0) id=35963 seq=10] IP [ttl=10 id=24903 iplen=28 ]
^C
Max rtt: 197.663ms | Min rtt: 196.455ms | Avg rtt: 196.919ms
Raw packets sent: 9 (378B) | Rcvd: 8 (280B) | Lost: 1 (11.11%)
Nping done: 1 IP address pinged in 9.65 seconds
root@main:~#
```

WKT traceroute works by sending the ICMP with decreasing values of TTL. Interpreting the traceroute is as follows: We are trying to reach the 5.101.0.1 from the 4.0.199.2 IP address.

1. As expected the management VM g4 interface IP address becomes its first hop IP address which is 4.0.199.1
2. Then, we get a reply from the 179.24.17.5 which is the ebgp peering interface IP address of AS5 and the IP address is 179.24.17.5
3. Then we get the reply from the destination 5.101.0.1 which is directly connected interface of the same router.



```
root@main:~# nping --dest-mac 06:7b:97:59:7b:5f --interface g4 --source-ip 4.0.199.2 --dest-ip 5.109.0.1 -v0 --tr

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2016-11-29 04:30 CET
SENT (1.6438s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=1] IP [ttl=1 id=49307 iplen=28 ]
RCVD (1.8415s) ICMP [4.0.199.1 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=64 id=22462 iplen=56 ]
SENT (2.6446s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=3] IP [ttl=3 id=49307 iplen=28 ]
RCVD (2.8415s) ICMP [179.24.17.5 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=21867 iplen=56 ]
SENT (3.6447s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=3] IP [ttl=3 id=49307 iplen=28 ]
RCVD (3.8415s) ICMP [179.24.17.5 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=62 id=21925 iplen=56 ]
SENT (4.6449s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=5] IP [ttl=5 id=49307 iplen=28 ]
RCVD (4.8415s) ICMP [5.0.6.2 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=60 id=46732 iplen=56 ]
SENT (5.6450s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=6] IP [ttl=6 id=49307 iplen=28 ]
RCVD (5.8415s) ICMP [5.0.7.2 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=59 id=41316 iplen=56 ]
SENT (6.6450s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=7] IP [ttl=7 id=49307 iplen=28 ]
RCVD (6.8415s) ICMP [5.0.12.2 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=58 id=60279 iplen=56 ]
SENT (7.6450s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=8] IP [ttl=8 id=49307 iplen=28 ]
RCVD (7.8415s) ICMP [5.0.12.2 > 4.0.199.2 TTL=0 during transit (type=11/code=0) ] IP [ttl=58 id=60339 iplen=56 ]
SENT (8.6451s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=9] IP [ttl=9 id=49307 iplen=28 ]
RCVD (8.8415s) ICMP [5.109.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=47539 seq=9] IP [ttl=57 id=51943 iplen=28 ]
SENT (9.6454s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=9] IP [ttl=9 id=49307 iplen=28 ]
RCVD (9.8415s) ICMP [5.109.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=47539 seq=9] IP [ttl=57 id=51968 iplen=28 ]
SENT (10.6456s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=10] IP [ttl=10 id=49307 iplen=28 ]
RCVD (10.8415s) ICMP [5.109.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=47539 seq=10] IP [ttl=57 id=52029 iplen=28 ]
SENT (11.6459s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=12] IP [ttl=12 id=49307 iplen=28 ]
RCVD (11.8415s) ICMP [5.109.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=47539 seq=12] IP [ttl=57 id=52215 iplen=28 ]
SENT (12.6459s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=13] IP [ttl=13 id=49307 iplen=28 ]
RCVD (12.8415s) ICMP [5.109.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=47539 seq=13] IP [ttl=57 id=52335 iplen=28 ]
SENT (13.6459s) ICMP [4.0.199.2 > 5.109.0.1 Echo request (type=8/code=0) id=47539 seq=14] IP [ttl=14 id=49307 iplen=28 ]
RCVD (13.8415s) ICMP [5.109.0.1 > 4.0.199.2 Echo reply (type=0/code=0) id=47539 seq=14] IP [ttl=57 id=52554 iplen=28 ]
^C
Max rtt: 197.638ms | Min rtt: 195.460ms | Avg rtt: 196.274ms
Raw packets sent: 13 (546B) | Rcvd: 13 (560B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 13.92 seconds
root@main:~#
```

The same thing is true even when I traceroute the SEAT router which can be routed directly through the tier 1. The path also doesn’t take any direct path to the SEAT, but goes only via the NEWY which has been by now advertised with higher preference.

Interpreting the traceroute is as follows: We are trying to reach the 5.101.0.1 from the 4.0.199.2 IP address.

1. As expected the management VM g4 interface IP address becomes its first hop IP address which is 4.0.199.1
2. Then, we get a reply from the 179.24.17.5 which is the ebgp peering interface IP address of AS5 and the IP address is 179.24.17.5
3. The next hop is the 5.0.6.2 which is the KANS. The routing is happening internally due to ibgp configured in the 1a
4. Then the next hop is 5.0.7.2 indicating the load balancing occurring internally in the AS
5. Then we get a reply from the 5.0.15.2 which is the SEAT router.
6. Then we get the reply from the destination 5.109.0.1 which is directly connected interface of the same SEAT router.

(4) Reason for choosing the evidences

- Even if the router has direct peer connection and the path is from another router within AS it is clear indication that there is differential treatment for a particular router.
- The traceroute gives the clear entry and exit path into the AS hence it was chosen. I chose AS4 and AS5 for nping traceroute that is because it is actually ebgp connected to SEAT and must prefer SEAT if this policy has no effect. But clearly nping traceroute command indicates otherwise.

We will set the inbound AS paths with the community of 15:100.

```
route-map AS4_prefixes permit 10

 set community 15:100

!
```

Again match the AS paths with the community of 4:100 and deny the traffic.

```
route-map AS4_prefixes deny 20

 match community 4:100
```