

使用路由策略实现BGP迭代时不迭代默认路由

路由策略

BGP

彭旭

2022-08-26 发表

问题描述

现场设备有一条非直连下一跳BGP路由，去往目的地址10.196.1.1，下一跳为192.168.128.1，192.168.128.1不是直连下一跳。

现场希望配置一条去往192.168.128.1（下一跳地址）的明细路由结合BFD检测，当这个路由失效的时候，去往10.196.1.1（目的地址）会匹配上现象另外一条10.196.1.0/24的网段路由走另外一边。

但是由于目前设备上有一条默认路由，所以当去往192.168.128.1（下一跳地址）的明细路由失效的时候，去往10.196.1.1（目的地址）的BGP路由还是不会失效，会迭代到默认路由走10.196.45.245。

去往目的地址10.196.1.1的路由，下一跳是192.168.128.1，还有一条10.196.1.0的网段路由，下一跳是192.168.127.1，但是由于设备有一条0.0.0.0的默认路由，导致当去往192.168.128.1的明细路由失效时，去往10.196.1.1的路由还是会迭代到默认路由走192.168.128.1，导致路由路径错误。

```
[H3C]disp ip routing-table vpn-instance pcx-ran 192.168.128.1
```

Summary count : 1

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/0	Static	60	0	10.196.45.245	Vlan3297

```
[H3C]disp ip routing-table vpn-instance pcx-ran 10.196.1.1
```

Summary count : 3

Destination/Mask	Proto	Pre	Cost	NextHop	Interface
0.0.0.0/0	Static	60	0	10.196.45.245	Vlan3297
10.196.1.0/24	BGP	255	0	192.168.127.1	Vlan3207
10.196.1.1/32	BGP	255	0	192.168.128.1	Vlan3297

解决方法

以上问题，可以通过设置路由策略让BGP路由不迭代默认路由，具体操作如下：

```
route-policy 1 deny node 1===路由策略拒绝迭代默认路由
if-match ip address prefix-list 1
#
route-policy 1 permit node 2===路由策略允许迭代其他路由
if-match ip address prefix-list 2
#
ip prefix-list 1 index 10 permit 0.0.0.0 0
ip prefix-list 2 index 10 permit 0.0.0.0 0 less-equal 32
```

最后在调用路由策略：

```
rib
#
address-family ipv4
    protocol bgp nexthop recursive-lookup route-policy 1
```

但是这种方法实现效果是：所有BGP的路由均不迭代默认路由，无法实现针对某条特定的路由设置让其不迭代默认路由。当前无法实现控制单条路由的迭代，只能是所有BGP路由不迭代默认+指定peer的路由可迭代默认，如下：

```
rib
#
address-family ipv4
    protocol bgp nexthop recursive-lookup route-policy 1

#
rib
#
address-family ipv4
    protocol bgp nexthop recursive-lookup route-policy rp1  #迭代策略，禁止BGP迭代默认路由，对所有bgp路由生效
#
bgp 100
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface LoopBack0
peer 1.1.1.1 nexthop-recursive-policy disable  #可以对单个BGP邻居来的路由，限制不使用迭代策略，那这个邻居仍然可以迭代到默认路由
策略禁止迭代默认
#
ip prefix-list 1 index 10 permit 0.0.0.0 0
#
route-policy rp1 deny node 0
if-match ip address prefix-list 1
#
route-policy rp1 permit node 1
```

