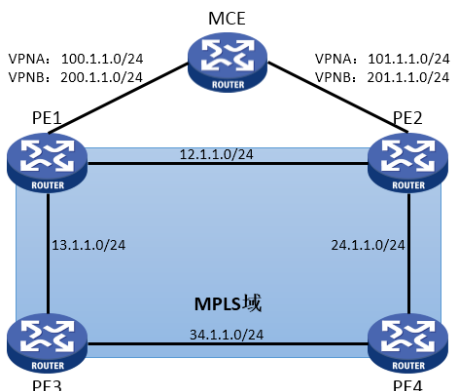


基于VPN实例的MPLS BGP VPN选路配置案例

MPLS L3VPN 秦军 2015-07-28 发表

PE1与CE之间、PE2与CE之间同时交互VPNA和VPNB两个VPN实例的路由，并通过MP-BGP传递给远端的PE3和PE4。由于PE1、PE2之间的互连链路开销较大，对于PE3而言，由于VPN路由MED值的不同，从PE1学习的VPN路由会优先通过PE1转发流量，而通过PE2学习的VPN路由会优先通过PE4转发给PE2。在实际应用中，客户希望不同的VPN流量走不同的转发链路，例如：PE3所有去往远端VPNA的流量通过PE1转发，所有去往远端VPNB的流量通过PE4转发。



VPNA的RT属性为1:1，VPNB的RT属性为2:1。

无

无

1. 基础配置

配置IGP路由协议和MPLS BGP VPN，确保公网PE之间路由互通，以及CE与PE1、PE2之间私网路由互通，PE3和PE4通过MP-BGP可以分别从PE1、PE2学习到VPN路由：

```
display bgp routing-table vpnv4
```

```
BGP local router ID is 3.3.3.3
Status codes: * - valid, > - best, d - dampened, h - history,
s - suppressed, S - stale, i - internal, e - external
Origin: i - IGP, e - EGP, ? - incomplete
```

```
Total number of routes from all PEs: 8
```

```
Route distinguisher: 1:1(VPNA)
```

```
Total number of routes: 5
```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
* > 3.1.1.1/32	127.0.0.1	0		32768	?
* > i 100.1.1.0/24	1.1.1.1	0	100	0	?
* i 2.2.2.2	3	100	0	?	
* > i 101.1.1.0/24	2.2.2.2	0	100	0	?
* i 1.1.1.1	3	100	0	?	

```
Route distinguisher: 2:1(VPNB)
```

```
Total number of routes: 5
```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
* > 3.1.1.2/32	127.0.0.1	0		32768	?
* > i 200.1.1.0	1.1.1.1	0	100	0	?
* i 2.2.2.2	3	100	0	?	
* > i 201.1.1.0	2.2.2.2	0	100	0	?
* i 1.1.1.1	3	100	0	?	

对于同一条VPN路由，既会从PE1接收，也会从PE2接收。对比从PE1、PE2接收的路由，它们有一个共同的属性，就是扩展属性中的RT属性。对于同一个VPN中的路由，RT属性都相同。因此，这里选择使用RT属性作为路由的分类依据，对于携带不同RT属性的路由应用不同的策略，以便作为BGP路由的选路依据。

```
display bgp routing-table vpnv4 100.1.1.0 24
```

```
BGP local router ID: 3.3.3.3  
Local AS number: 1
```

```
Route distinguisher: 1:1(VPNA)  
Total number of routes: 2  
Paths: 2 available, 1 best
```

```
BGP routing table information of 100.1.1.0/24:  
From      : 1.1.1.1 (1.1.1.1)  
Rely nexthop  : 13.1.1.1  
Original nexthop: 1.1.1.1  
OutLabel    : 1279  
Ext-Community :  
AS-path     : (null)  
Origin      : incomplete  
Attribute value : MED 0, localpref 100, pref-val 0  
State       : valid, internal, best  
IP precedence : N/A  
QoS local ID : N/A  
Traffic index : N/A
```

```
From      : 2.2.2.2 (2.2.2.2)  
Rely nexthop  : 34.1.1.2  
Original nexthop: 2.2.2.2  
OutLabel    : 1148  
Ext-Community : ,  
AS-path     : (null)  
Origin      : incomplete  
Attribute value : MED 3, localpref 100, pref-val 0  
State       : valid, internal  
IP precedence : N/A  
QoS local ID : N/A  
Traffic index : N/A
```

2.配置扩展团体属性列表

为了对不同VPN的路由进行匹配，这里使用扩展团体属性列表，用于匹配VPN路由的RT属性：

```
#  
ip extcommunity-list 1 permit rt 1:1 //匹配VPNA路由的RT值  
ip extcommunity-list 2 permit rt 2:1 //匹配VPNB路由的RT值  
#
```

3.配置路由策略

路由策略用于修改路由的local-preference属性。对于从PE1接收的VPNA的路由，local-preference属性修改为150，VPNB的路由，local-preference属性修改为50；对于从PE2接收的VPNA的路由，local-preference属性修改为50，VPNB的路由，local-preference属性修改为150；

```
#  
route-policy PEER1 permit node 10  
if-match extcommunity 1  
apply local-preference 150  
#  
route-policy PEER1 permit node 20  
if-match extcommunity 2  
apply local-preference 50  
#  
route-policy PEER1 permit node 30  
#  
route-policy PEER2 permit node 10  
if-match extcommunity 1  
apply local-preference 50  
#  
route-policy PEER2 permit node 20  
if-match extcommunity 2  
apply local-preference 150  
#  
route-policy PEER2 permit node 30  
#
```

4.应用路由策略

为相应的BGP VPNv4对等体设置接收路由策略。

```
bgp 1  
#  
address-family vpnv4  
peer 1.1.1.1 enable  
peer 1.1.1.1 route-policy PEER1 import  
peer 2.2.2.2 enable  
peer 2.2.2.2 route-policy PEER2 import  
#
```

查看VPNv4路由，确认local-preference属性是否修改成功：

dis bgp routing-table vpnv4

BGP local router ID is 3.3.3.3

Status codes: * - valid, > - best, d - dampened, h - history,
s - suppressed, S - stale, i - internal, e - external
Origin: i - IGP, e - EGP, ? - incomplete

Total number of routes from all PEs: 8

Route distinguisher: 1:1(VPNA)

Total number of routes: 5

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
* > 3.1.1.1/32	127.0.0.1	0		32768	?
* >i 100.1.1.0/24	1.1.1.1	0	150	0	?
* i 2.2.2.2	3	50	0	?	
* >i 101.1.1.0/24	1.1.1.1	3	150	0	?
* i 2.2.2.2	0	50	0	?	

Route distinguisher: 2:1(VPNB)

Total number of routes: 5

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
* > 3.1.1.2/32	127.0.0.1	0		32768	?
* >i 200.1.1.0	2.2.2.2	3	150	0	?
* i 1.1.1.1	0	50	0	?	
* >i 201.1.1.0	2.2.2.2	0	150	0	?
* i 1.1.1.1	3	50	0	?	

无