

Unified MPLS配置示例

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景](#)

[体系结构](#)

[配置](#)

[验证](#)

[故障排除](#)

[相关信息](#)

简介

本文描述Unified多协议标签交换(MPLS)目的并且提供配置示例。

[先决条件](#)

[要求](#)

本文档没有任何特定的要求。

[使用的组件](#)

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

背景

unified MPLS目的是所有关于比例缩放。为了扩展MPLS网络，有不同类型的platforms和服务在网络的部分中，它有意义分割网络到不同的区域。一典型的设计引入有一个核心在有聚合的中心在侧的层级。为了扩展，那里可以在不同的内部网关协议(IGP)在核心与聚合。为了扩展，您不能分配从一个IGP的IGP前缀到其他。如果不分配从一个IGP的IGP前缀到另一个IGP，端到端标签交换路径

(LSP)不是可能的。

为了提供端到端MPLS的服务，您需要LSP端到端。目标是保持MPLS服务(MPLS VPN， MPLS L2VPN)，因为他们是，但是引入更加极大的可扩展性。为了执行此，请搬入某些IGP前缀边界网关协议(BGP) (服务商边缘路由器的环回前缀)，然后分配端到端的前缀。

体系结构

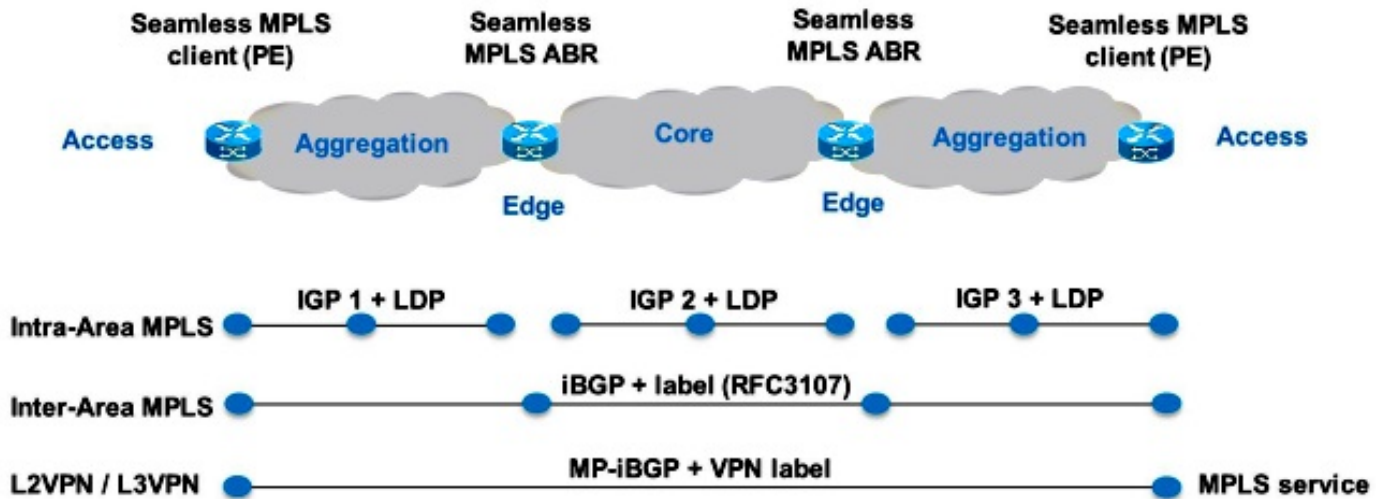


Figure 1

图1显示与三个不同的区域的网络：一个核心和两个聚合区域在侧。各个领域运行其自己的IGP，没有再分配在他们之间在区域边界路由器(ABR)。使用BGP是需要的为了提供一端到端MPLS LSP。BGP通告PE路由器的环回用在整体域间的一个标签，并且提供一端到端LSP。BGP部署在观点扫描器和ABR之间与RFC 3107，因此意味着BGP发送IPv4前缀+标签(AFI/SAFI 1/4)。

因为网络的核心和聚合零件集成，并且提供端到端LSP， Unified MPLS解决方案也指“无缝的MPLS”。

这里，只没有使用新技术或协议MPLS、标签转发协议(LDP)、IGP和BGP。因为您不要分配PE路由器的环回前缀从网络的一部分的到另一个部分，您需要运载在BGP的前缀。内部边界网关协议(iBGP)用于一网络，因此前缀的下一跳地址是PE路由器的环回前缀，不由IGP在网络的其他部分中知道。这意味着下一跳地址不可能用于对IGP前缀的递归。窍门是做ABR路由器路由反射器(RR)和设置下一跳为自己，为反射的iBGP前缀。为了此能工作，一个新的瘤是需要的。

支持此体系结构的仅RR需要更新的软件。因为RR通告BGP前缀下一跳设置对他们自己，他们分配本地MPLS标签到BGP前缀。这意味着在数据层面，在这些端到端LSP转发的数据包有一个额外的MPLS标签在标签栈。RR在转发路径。

Note:在此体系结构，提供所有MPLS服务。例如， MPLS VPN或MPLS L2VPN提供在PE路由器之间。在数据层面的差异这些数据包的是他们当前有三个标签在标签栈，而他们有两个标签在标签栈，当未使用Unified MPLS。

有两个可能的情况：

- ABR不设置下一跳为ABR通告的(反射由BGP)前缀的自己到网络的聚合零件。因此，ABR需要重新分配ABR的环回前缀从核心IGP的到聚合IGP。如果这执行，仍有可扩展性。仅ABR环回前缀(从核心)需要通告到聚合零件，从PE路由器的不是环回前缀从远程聚合零件。
- ABR设置下一跳为ABR通告的(反射由BGP)前缀的自己到聚合零件。因此，ABR不需要重新分配ABR的环回前缀从核心IGP的到聚合IGP。

在两种情况下，ABR设置下一跳为从网络的聚合零件的ABR通告的(反射由BGP)前缀的自己到核心零件。如果这没有执行，ABR需要重新分配观点扫描器的环回前缀从聚合IGP的到核心IGP。如果这执行，没有可扩展性。

为了设置下一跳为反射的iBGP路由的自己，您必须配置**next-hop-self all**命令邻接的x.x.x.x。

配置

这是PE路由器和ABR的配置方案的2。

Note: topology在表2.显示。示例服务是**xconnect** (MPLS L2VPN)。在PE路由器和ABR之间，有**IPv4 +标签的BGP**。

PE1

```
interface Loopback0
 ip address 10.100.1.4 255.255.255.255

!
interface Ethernet1/0
 no ip address
 xconnect 10.100.1.5 100 encapsulation mpls
!
router ospf 2
 network 10.2.0.0 0.0.255.255 area 0
 network 10.100.1.4 0.0.0.0 area 0
!
router bgp 1
 bgp log-neighbor-changes
 network 10.100.1.4 mask 255.255.255.255
 neighbor 10.100.1.1 remote-as 1
 neighbor 10.100.1.1 update-source Loopback0
 neighbor 10.100.1.1 send-label
```

RR1

```
interface Loopback0
 ip address 10.100.1.1 255.255.255.255
router ospf 1
 network 10.1.0.0 0.0.255.255 area 0
 network 10.100.1.1 0.0.0.0 area 0
!
router ospf 2
 redistribute ospf 1 subnets match internal route-map ospf1-into-ospf2
 network 10.2.0.0 0.0.255.255 area 0
```

```
!  
router bgp 1  
  bgp log-neighbor-changes  
  neighbor 10.100.1.2 remote-as 1  
  neighbor 10.100.1.2 update-source Loopback0  
  neighbor 10.100.1.2 next-hop-self all  
  neighbor 10.100.1.2 send-label  
  neighbor 10.100.1.4 remote-as 1  
  neighbor 10.100.1.4 update-source Loopback0  
  neighbor 10.100.1.4 route-reflector-client  
  neighbor 10.100.1.4 next-hop-self all  
  neighbor 10.100.1.4 send-label  
  
ip prefix-list prefix-list-ospf1-into-ospf2 seq 5 permit 10.100.1.1/32  
  
route-map ospf1-into-ospf2 permit 10  
  match ip address prefix-list prefix-list-ospf1-into-ospf2
```

RR2

```
interface Loopback0  
  ip address 10.100.1.2 255.255.255.255  
  
router ospf 1  
  network 10.1.0.0 0.0.255.255 area 0  
  network 10.100.1.2 0.0.0.0 area 0  
!  
router ospf 3  
  redistribute ospf 1 subnets match internal route-map ospf1-into-ospf3  
  network 10.3.0.0 0.0.255.255 area 0  
!  
router bgp 1  
  bgp log-neighbor-changes  
  neighbor 10.100.1.1 remote-as 1  
  neighbor 10.100.1.1 update-source Loopback0  
  neighbor 10.100.1.1 next-hop-self all  
  neighbor 10.100.1.1 send-label  
  neighbor 10.100.1.5 remote-as 1  
  neighbor 10.100.1.5 update-source Loopback0  
  neighbor 10.100.1.5 route-reflector-client  
  neighbor 10.100.1.5 next-hop-self all  
  neighbor 10.100.1.5 send-label  
  
ip prefix-list prefix-list-ospf1-into-ospf3 seq 5 permit 10.100.1.2/32  
  
route-map ospf1-into-ospf3 permit 10  
  match ip address prefix-list prefix-list-ospf1-into-ospf3
```

PE2

```
interface Loopback0  
  ip address 10.100.1.5 255.255.255.255  
  
interface Ethernet1/0  
  no ip address  
  xconnect 10.100.1.4 100 encapsulation mpls  
  
router ospf 3  
  network 10.3.0.0 0.0.255.255 area 0  
  network 10.100.1.5 0.0.0.0 area 0  
  
router bgp 1
```

```

bgp log-neighbor-changes
network 10.100.1.5 mask 255.255.255.255
neighbor 10.100.1.2 remote-as 1
neighbor 10.100.1.2 update-source Loopback0
neighbor 10.100.1.2 send-label

```

Note:核心IGP (ospf 1)的再分配到聚合IGP里(ospf2或ospf 3)用route-map执行。此route-map允许RR的环回前缀重新分配到聚合IGP。对此的原因是RR的环回前缀直接地只通告到核心IGP (ospf 1)。然而，在聚合IGP必须也知道RR的环回前缀，因此在PE路由器的BGP能并列与RR的环回。

验证

参见图2为了验证控制层面操作。

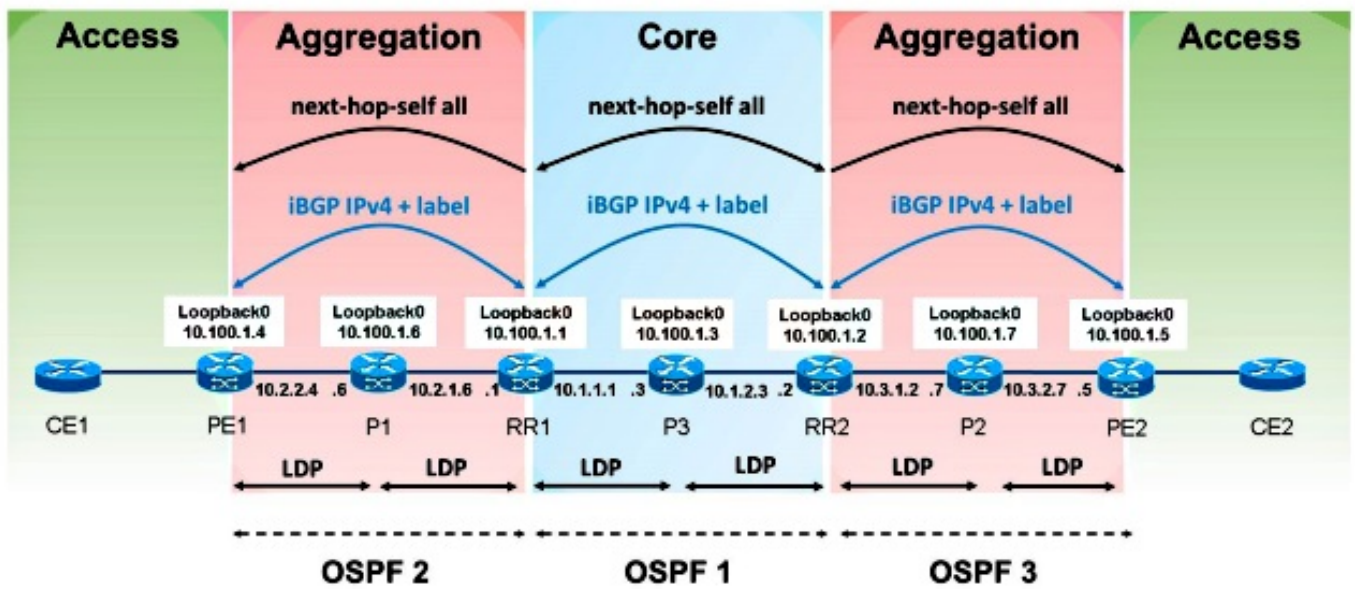


Figure 2

参见图3为了验证MPLS标签广告。

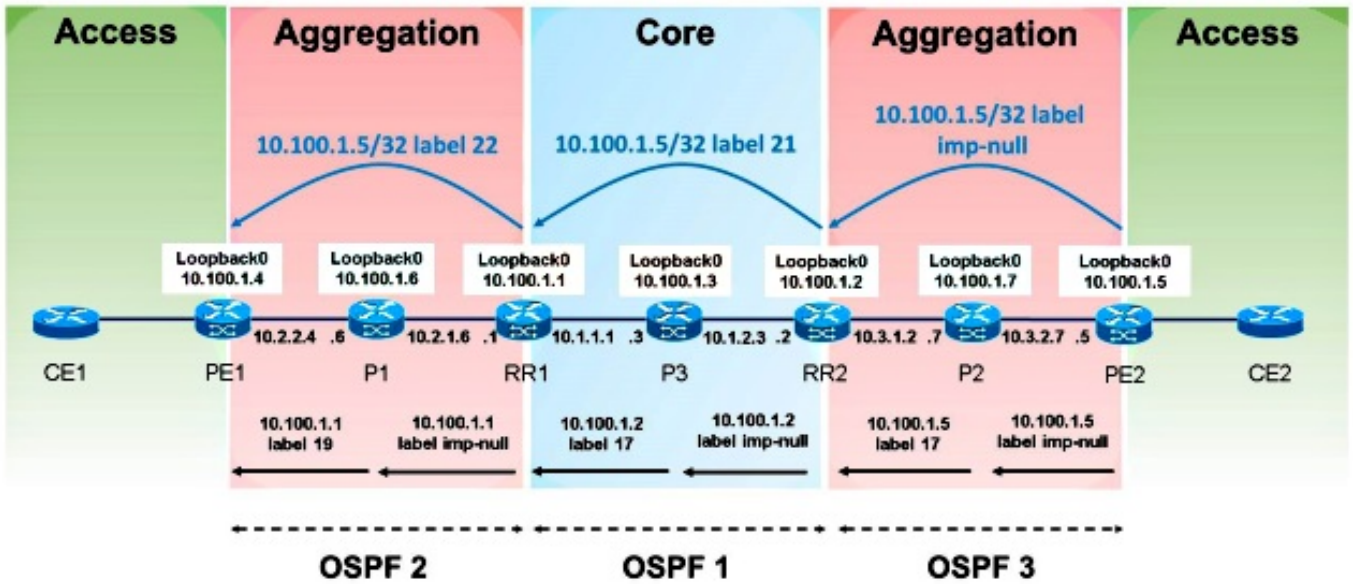


Figure 3

参见图4为了验证信息包转发。

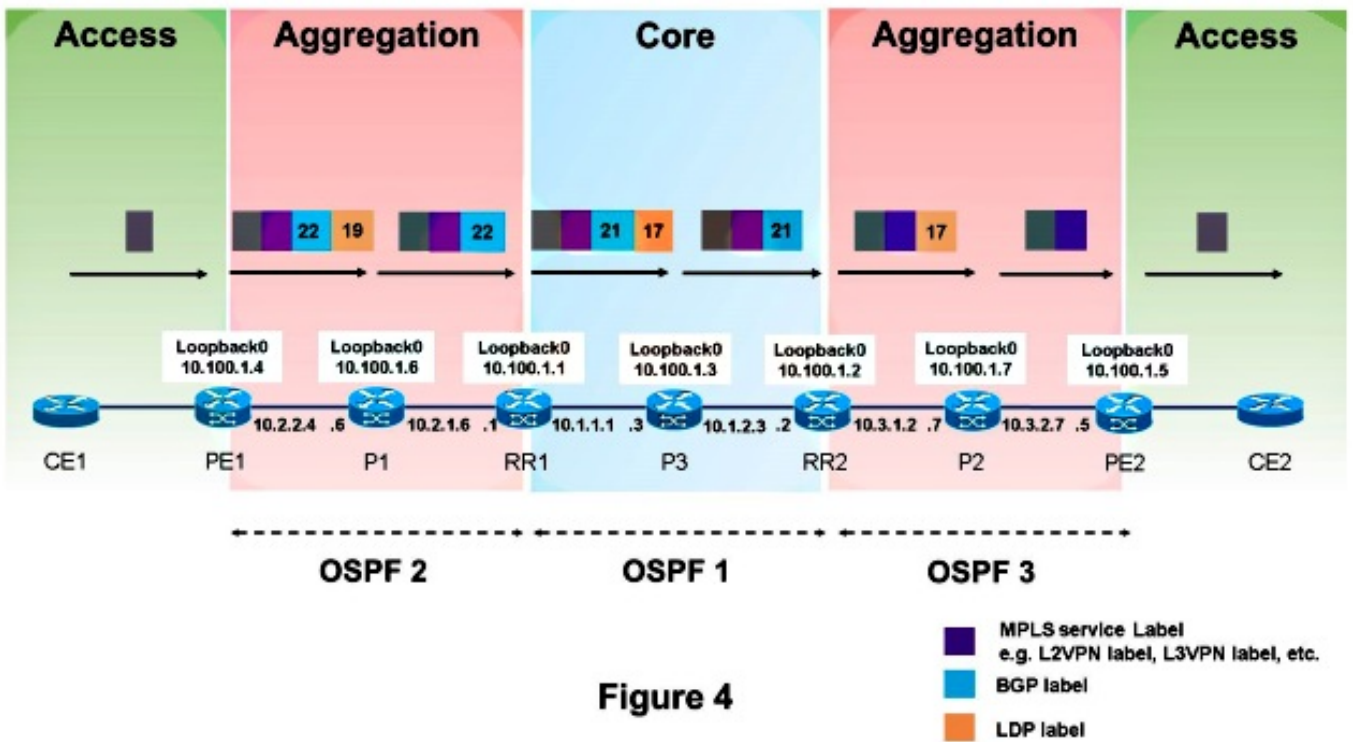


Figure 4

这是数据包如何从PE1转发到PE2。PE2环回前缀是10.100.1.5/32，因此前缀是利益。

```
PE1#show ip route 10.100.1.5
```

```
Routing entry for 10.100.1.5/32
  Known via "bgp 1", distance 200, metric 0, type internal
  Last update from 10.100.1.1 00:11:12 ago
  Routing Descriptor Blocks:
    * 10.100.1.1, from 10.100.1.1, 00:11:12 ago
```

Route metric is 0, traffic share count is 1
AS Hops 0
MPLS label: 22

PE1#show ip cef 10.100.1.5
10.100.1.5/32
nexthop 10.2.2.6 Ethernet0/0 **label 19 22**

PE1#show ip cef 10.100.1.5 detail
10.100.1.5/32, epoch 0, flags rib defined all labels
1 RR source [no flags]
recursive via 10.100.1.1 label 22
nexthop 10.2.2.6 Ethernet0/0 **label 19**

PE1#show bgp ipv4 unicast labels

Network	Next Hop	In label/Out label
10.100.1.4/32	0.0.0.0	imp-null/nolabel
10.100.1.5/32	10.100.1.1	nolabel/22

P1#show mpls forwarding-table labels 19 detail

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
19	Pop Label	10.100.1.1/32	603468	Et1/0	10.2.1.1

MAC/Encaps=14/14, MRU=1504, Label Stack{
AABBCC000101AABBCC0006018847
No output feature configured

RR1#show mpls forwarding-table labels 22 detail

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
22	21	10.100.1.5/32	575278	Et0/0	10.1.1.3

MAC/Encaps=14/22, MRU=1496, **Label Stack{17 21}**
AABBCC000300AABBCC0001008847 0001100000015000
No output feature configured

RR1#show bgp ipv4 unicast labels

Network	Next Hop	In label/Out label
10.100.1.4/32	10.100.1.4	19/imp-null
10.100.1.5/32	10.100.1.2	22/21

P3#show mpls forwarding-table labels 17 detail

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
17	Pop Label	10.100.1.2/32	664306	Et1/0	10.1.2.2

MAC/Encaps=14/14, MRU=1504, Label Stack{
AABBCC000201AABBCC0003018847
No output feature configured

RR2#show mpls forwarding-table labels 21 detail

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
21	17	10.100.1.5/32	615958	Et0/0	10.3.1.7

MAC/Encaps=14/18, MRU=1500, **Label Stack{17}**
AABBCC000700AABBCC0002008847 00011000
No output feature configured

```
RR2#show bgp ipv4 unicast labels
```

```
Network          Next Hop          In label/Out label
10.100.1.4/32    10.100.1.1       22/19
10.100.1.5/32    10.100.1.5       21/imp-null
```

```
P2#show mpls forwarding-table labels 17 detail
```

```
Local      Outgoing  Prefix           Bytes Label  Outgoing  Next Hop
Label      Label     or Tunnel Id     Switched     interface
17         Pop Label 10.100.1.5/32    639957       Et1/0      10.3.2.5
MAC/Encaps=14/14, MRU=1504, Label Stack{}
AABBCC000500AABBCC0007018847
No output feature configured
```

```
PE1#trace
```

```
Protocol [ip]:
```

```
Target IP address: 10.100.1.5
```

```
Source address: 10.100.1.4
```

```
DSCP Value [0]:
```

```
Numeric display [n]:
```

```
Timeout in seconds [3]:
```

```
Probe count [3]:
```

```
Minimum Time to Live [1]:
```

```
Maximum Time to Live [30]:
```

```
Port Number [33434]:
```

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

```
Type escape sequence to abort.
```

```
Tracing the route to 10.100.1.5
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
 1 10.2.2.6 [MPLS: Labels 19/22 Exp 0] 3 msec 3 msec 3 msec
 2 10.2.1.1 [MPLS: Label 22 Exp 0] 3 msec 3 msec 3 msec
 3 10.1.1.3 [MPLS: Labels 17/21 Exp 0] 3 msec 3 msec 2 msec
 4 10.1.2.2 [MPLS: Label 21 Exp 0] 2 msec 3 msec 2 msec
 5 * * *
 6 10.3.2.5 4 msec * 4 msec
```

Note:跳5显示？5***？。这是因为路由器P2没有源IP地址的10.100.1.4 (PE1)一个路由traceroute。因此，路由器P2不能送回互联网控制消息协议(ICMP)错误消息到PE1。因为问题的Unified MPLS是没有所有PE路由器环回前缀在出现的一个聚合零件中在其他聚合零件的IGP，这是正常。路由器P2不尝试转发与原始标签栈的ICMP错误信息。这是因为原始标签栈只有一个标签。如果数据包的此原始标签栈有两个或多个标签，ICMP错误信息沿LSP转发，并且能有上一步traceroute的来源。如果原始标签栈只有一个标签，生成ICMP错误信息的路由器尝试路由查找并且设法路由它与使用路由表(没有使用原始标签栈)。

```
P2#show ip route 10.100.1.4
```

```
% Subnet not in table
```

故障排除

目前没有针对此配置的故障排除信息。

相关信息

- [无缝的MPLS体系结构](#)
- [技术支持和文档 - Cisco Systems](#)