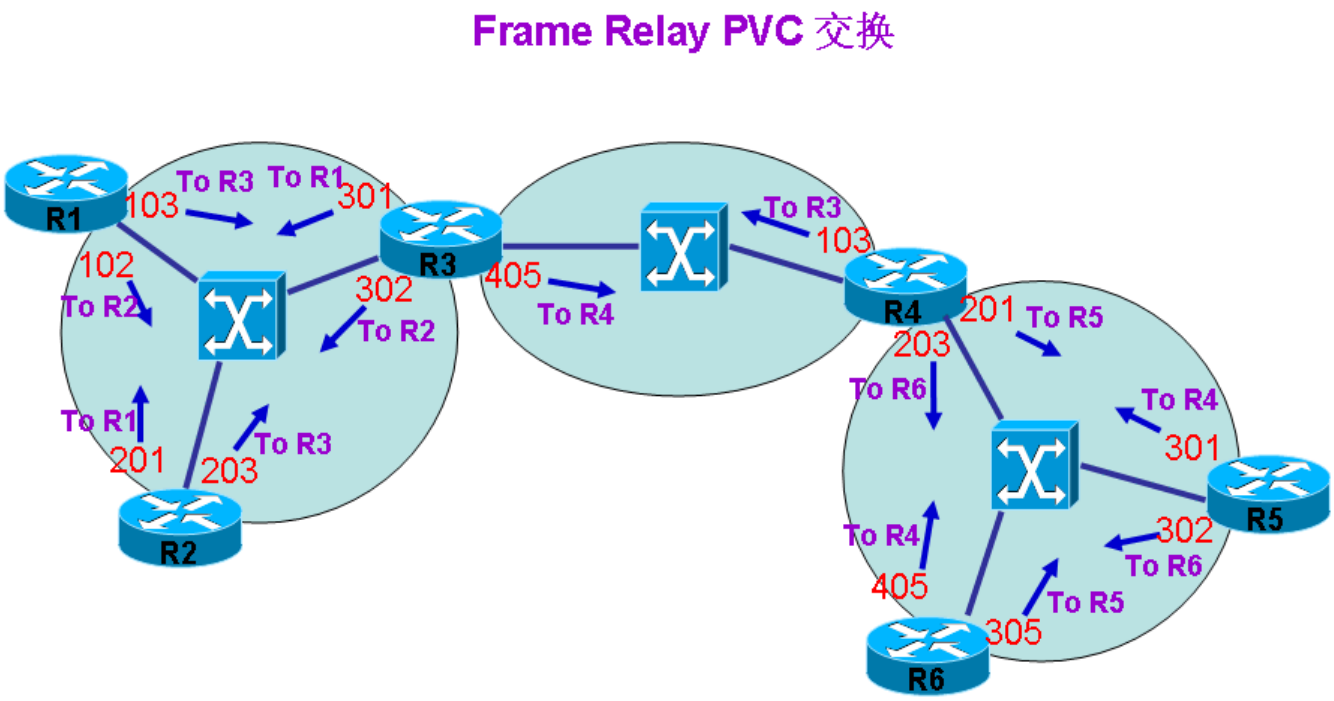
MPLS / MPLS_VPN

目录

MPLS

非 IP 包头交换过程

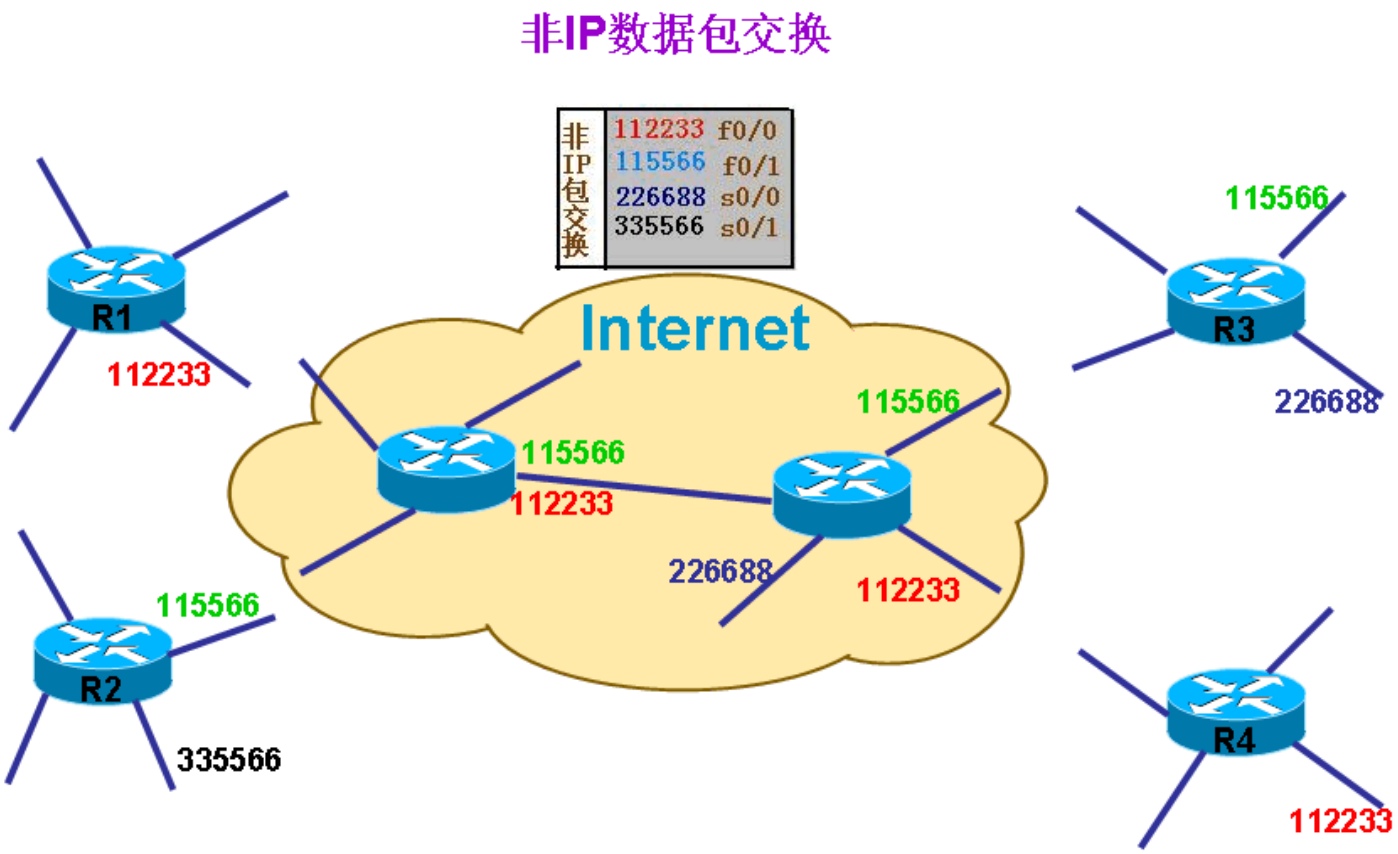
1.帧中继 PVC 交换方式



在我们现有的网络当中，IP 数据包网络占绝大部分，这样的 IP 数据包网络，在网络设备传递数据包时，是根据数据包的 IP 包头信息进行交换的，也就是网络设备根据包头中的目标 IP 地址，来决定从哪个接口转发出去。所以在数据包当中，指导设备正确转发数据包的就是 IP 地址信息，而 IP 地址只是数据包的一个标识而已。既然数据包的包头信息能够指导设备正确转发，那数据包的包头只要能够被设备正确接受，就能够做出正确的转发决策，正因为如此，网络就产生了其它不同于 IP 数据包交换方式，比如我们应当熟悉的帧中继网络（Frame Relay）。在帧中继网络中，很明显，帧中继设备（帧中继交换机）在决定数据包该从哪个接口被发出去时，查看的就是包头 PVC 号码，而不是 IP 地址，这个 PVC 号码，对于帧中继设备来说，就关系到这个数据包应该从哪个接口被转发出去。如上图所示，在帧中继交换机中，只关系数据包的 PVC 号码是多少，只要看到这个号码，就知道该从哪个接口出去，等

数据包到了下一台交换机之后，下一台交换机也做同样的操作，即查看数据包的 PVC 号码后就从相应接口发出去，但是不同数据包的 PVC 号码肯定应该是不一样的，因为同一个 PVC 号码，对于交换机来说，都应该从同一个接口出去。所以说一台交换机上的每个接口相关联的 PVC 号码都是不一样的。但是，这台交换机用过的 PVC 号码，到了下一台交换机之后，还是可以使用的，因为前面一台交换机根据某个 PVC 号码对数据包转发之后，自己再根据数据包的 PVC 进行转发，与前面是不冲突的，因为是各自关联好的。从这里也可以想象出，一个数据包经过一台帧中继交换机之后，到了下面一台，数据包的 PVC 是应该被设备进行重新改写才交换的，因为不可能一个 PVC 经过 N 台交换机还是一样的。所以可以得出一个理论就是，帧中继数据包的包头信息（即 PVC 号码）仅一跳有效，也就是本地有效，不同交换机之间，包头信息可重复，在这里用过的 PVC 号码，在别的交换机上也可能出现一样的，只要保证在单台交换机唯一就可以了，所以每次经过一个交换机之后，需要重新改写包头信息。

2. 非 IP 数字包头交换方式



下面来看一下既不是 IP 数据包交换，也不是帧中继交换的网络，那么这样的网络给数据包写上什么样的包头来指导设备正确转发呢？就写一个号码而已，我们暂且称它为非 IP 数字包头交换。在这样的网络中，设备看到包头中的这个数字，就知道该从哪个接口转发出去，每台都是一样的。这种网络跟帧中继交换相同的是设备也是查看一个号码，而不同的是，一个数据包写上一个号码之后，永远都不会被任何设备改写，直接传到目的地为止，可想而知，网络中任意两台主机之间，他们的号码必须是唯一的，因为每台交换机都要根据这个号码来做出转发，如果两个数据包的号码相同，那么所有的交换机都做同样的转发，结果就导致这些数据包被发到同一台设备。这样的数字网络，全球中，每两点之间仅有一个号码表示，第一个点的数据包头写上此号码，必定是发到第二个点，不可能发送到第三个点，因为第一个点和第三个点，会使用另外一个号码，所以此号码为两点唯一，网络设备中有每两点（即每个号码）的出口信息，收到任何一个包，都能不看 IP 地址而根据此号码选择从相应接口发出去，每台设备执行相同的过程，即可完成任意两点间的传输。此交换方式其实并没有在计算机网络中应用，但是我们使用的电话网络，就是这种交换方式，即任何两台电话之间打电话，号码唯一，不可能有相同号码，如果你拨打电话，别人也拨一个电话号码，你们拨的号码如果是一样的，那肯定就打到同一个人那里去了。所以要实现此交换方式，网络中所有设备需要计算出任意两个点之间的号码，每一个号码都是唯一的，不可重复，与到目的地的相应出口作对应，生成转发表。但是如果全球计算机网络使用这样的方式，那就是任何一台设备为任何一台主机计算路径时，都要所有全球的设备共同参与，如果不全部都参与，就可能和没参与的计算出重复号码，可想而知工作量之庞大。

3.交换方式总结

以上两种交换方式，都是在不看 IP 地址（IP 包头），只看号码的情况下，做出的交换选择。

可以仔细想一下，在使用帧中继交换时，因为一个 PVC 号码只要保证单台设备不重复就可以了，这个号码跟接口是关联着的，也就是说一个数据包写上的 PVC 号码，这个 PVC 号码的范围只要比交换机的接口多就行，比如范围是 1024，所以帧中继交换的包头，号码不是很庞大，也就是说包头并不是很长。而非 IP 包交换的网络中，因为每两点之间都要有独立的号码，所以如果网络中有 10 亿个点，那么这个号码的范围就应该比 10 亿还要大，所以非 IP 包交换，数据包头肯定要比帧中继的包头大。

但是从结论中，我们能不能说哪个好，哪个不好呢？当然不能，因为帧中继的包

头虽然比非 IP 的包头要小，但是每经过一台设备都要重新改写，也就是说帧中继网络中，设备都在不停地为每个数据包改写 PVC 号码，这也是巨大的工作量啊。而非 IP 包头虽然要大一些，但是这个号码写好之后，就永远不会再变了，只要中间的设备看到号码直接转发就行，不用改写了。

4.MPLS（多协议标签交换）

在使用 IP 包交换网络的时候，人们总是认为设备要根据 IP 地址查路由表做出转发决定，觉得这样很耗时，总想着寻找一种新的交换技术来代替 IP 包交换。最初就考虑使用数字号码的方式来代替 IP 地址，从上面介绍的交换方式中，由于第二种非 IP 包交换技术，需要在网络中对任意两点计算出一个全球唯一的号码，因为一个号码即代表了两个点之间的传输，如果其它的点之间的号码和别人出现重复，那么数据的走向也就会发生错误。当前没有研发出协议敢保证计算出任意两点的号码一定是唯一的。

在帧中继的交换中，只要保证每个网段（每台交换机）之间的 PVC 号码唯一即可，因为经过每个网段（每个交换机）号码都会重新修改。此交换方式也可避免设备检查数据包的 IP 地址来作做转发决定。（但也不要忘记，这种交换方式的弊端,在似乎节省了时间的同时，其实也浪费了许多时间。）

而当前人们认为效率比较高的 MPLS（多协议标签交换）方式，它的数据交换思想则倾向帧中继的交换方式，即认为设备在查看 IP 地址之后做出转发决定，会比较慢，会耗更多时间，则给数据包写上了额外的号码，根据此号码而不看 IP 地址，便能找出相应出口从而转发出去，这正是标签交换，而 MPLS 称此额外的号码为标签。在此可以看出，MPLS 的交换号码（标签）并不是全球唯一，只是每个网段唯一，或者说是每跳唯一，所以，经过一跳之后，此号码对下一跳设备毫无意义，经过一跳之后，此号码要修改成对下一跳有意义的号码，让其根据号码做出转发决定。由此可见，MPLS 的标签交换，是每跳都会改写标签，因为根据标签，便能够做出转发决定，所以省略了查看 IP 地址的过程，被人们认为比 IP 交换要快。

而 MPLS 根据自己的标签交换，需要给数据包先写上自己的标签，然后设备才能查看标签之后就转发，此标签是需要在原有的数据包的基础上加进去的，并没有将以前的包头删除，MPLS 的标签加在了第二层帧的帧头之后，但又在第三层数据包的包头之前，而 MPLS 不管是什么协议的数据包，不管以前的包头是什么，都能够在包中加入对自己有利的标签，所以称 MPLS 交换为多协议标签交换。

MPLS 优势：

1.（不正确的理由）：因为 IP 在路由当中，总是根据目的地址在路由表中查找目标网段，并且逐条匹配最优路径，速度慢。

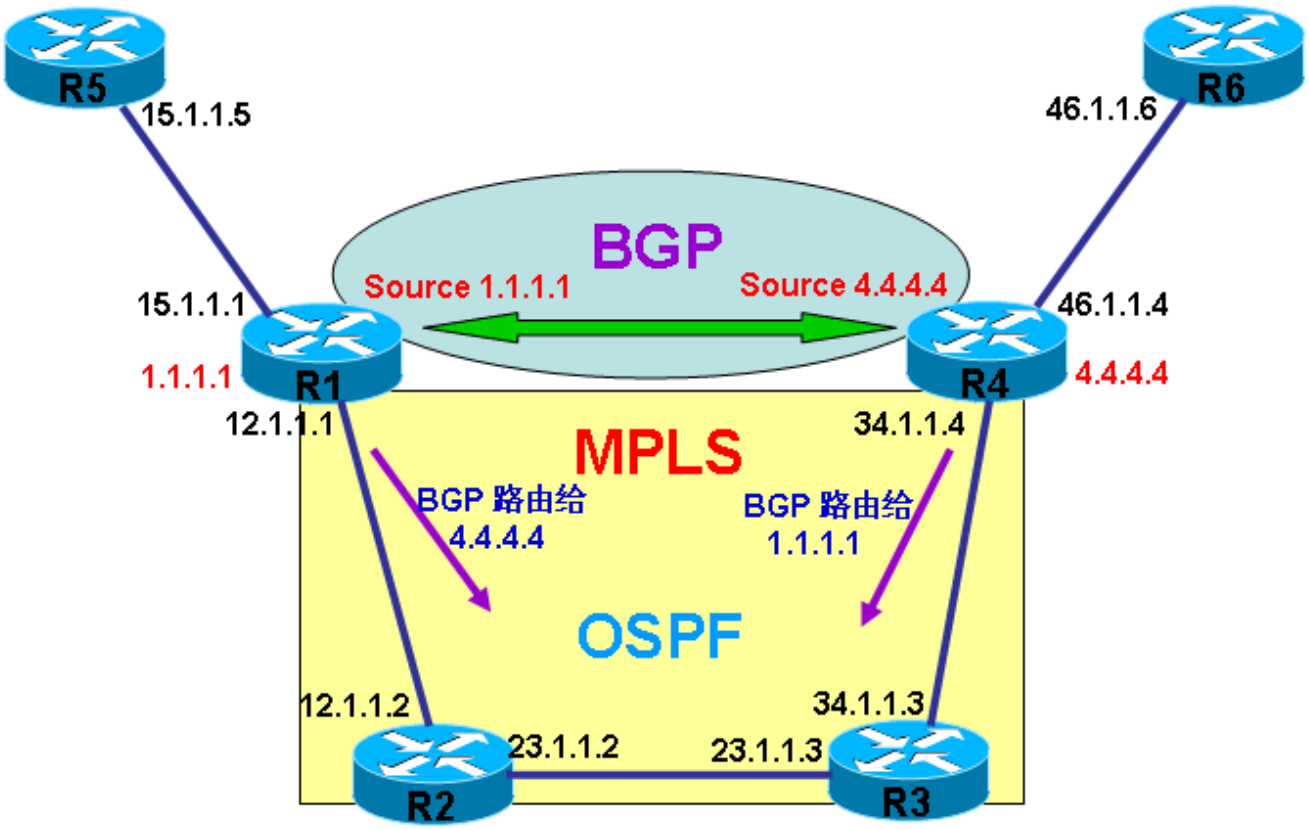
2. MPLS 只根据数据包顶部标签来查找并转发，速度快。

结论：由于现在设备采用 ASIC（专用集成电路）交换，所以速度并不慢，而 MPLS 借签了帧中继交换方式，在数据包每经过一台设备时，都要重新封装，所以 MPLS 在速度上，并不是优势。

但 MPLS 可以给数据包加上标签，以做流量控制，这是优势。MPLS 还可以承载各种协议，如 IPv4,IPv6, 以太网，HDLC，PPP，以及其它第二层帧。

注：MPLS 在骨干网中传输任意第二层帧的特征被称为 MPLS 的任意传输（AToM）。

MPLS 中的 BGP



BGP 在决定一个数据包该如何被转发出去，是通过查找 IP 地址在路由表中的下一跳，有时不能避免这下一跳不是跟自己直连的，但是 BGP 只要知道如何到达那个下一跳即可，所以 BGP 路由的下一跳，也许自己不清楚，但是只要 IGP 的路由能帮助自己到达下一跳就行。在大型的核心网络中，我们完全可以设计出网络这边的 BGP 路由器，它的下一跳在网络那边，那么如何到达网络那边的下一跳，中间就可以使用 IGP 去完成，只要中间的设备能够帮助 BGP 到达最终下一跳地址就足够了，所以这样的网络，需要 BGP 协议的只是网络的边缘路由器，而中间的路由器，只要做一件事，那就是帮 BGP 找下一跳，就不用启用 BGP 了，这就大大节省了系统资源。而 MPLS 的标签交换，就可以用在这样的网络中，来为 BGP 寻找下一跳，也就是 MPLS 只要为 BGP 路由的下一跳打上标签，能够帮助 BGP 找到下一跳，那么其它的问题，都不是问题，其它的路由，BGP 就能够自己完成。

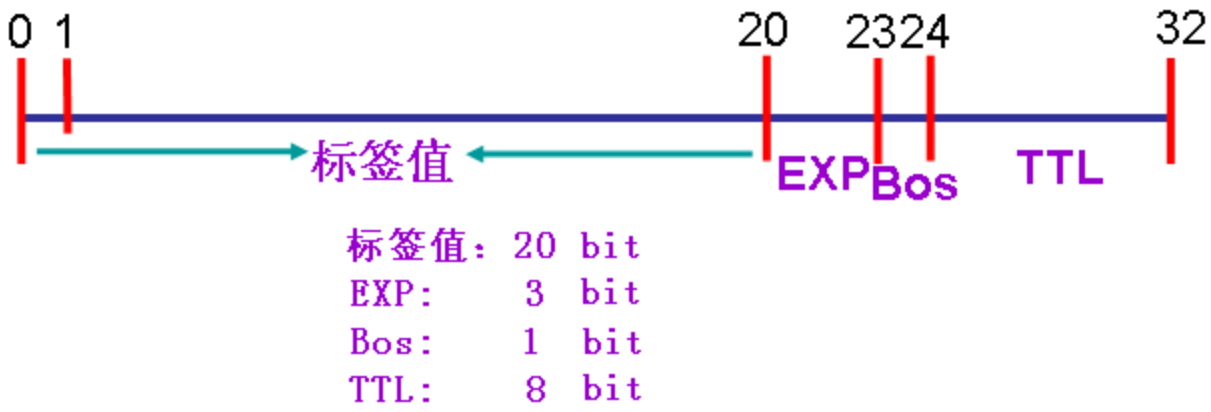
略： MPLS 流量工程，路径是由首端路由器指定的，所以又称为基于源的路由。

思科 MPLS 历史

最初 Cisco 在 IP 报文顶部加入标签时，称其为标记交换（tag switching），而标记，现在改叫标签了，为每个路由条目分配好标记并写进去，所以这就需要一张表来指导标记交换，称为转发信息库（TFIB），每一台标记交换路由器查看数据包入站的标记，并转为出站标记后发出去。

注：思科第一个支持标记交换的 IOS 就支持流量工程，就是资源预留协议（RSVP）

MPLS 标签



一个标签由 32 个 bit 组成

前 20 为标签值，范围从 0 到 2 的 20 次方减一，即 1048575。

其中前 16bit 不能随便定义，有特定含义，从 21 到 23bit 共 3 位试验用（EXP），用于 QOS。

第 24 比特是栈底 Bos 位，值为 0，如果是栈底，就为 1，标签栈中，标签数量没有限制。

从 25 到 32 共 8 个 bit 是 TTL

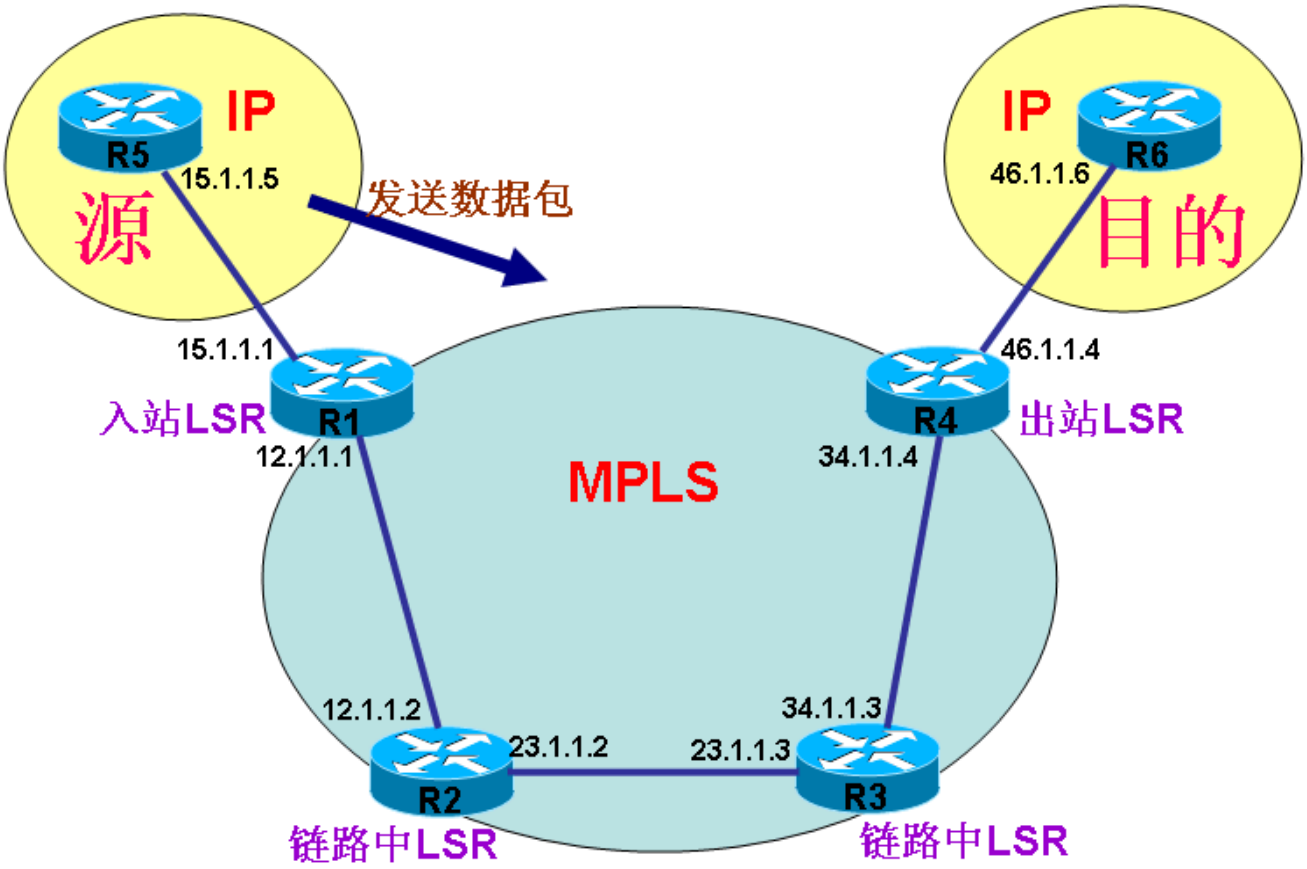
MPLS 标签栈

MPLS 路由器对数据包可能添加一个标签，也可能添加多个标签，这些标签集合起来叫做标签栈，第一个为顶部标签，最后一个为底部标签，中间数量可以无限，底部标签 BOS 总是 1，否则就是 0。而在数据包传输过程中，设备只根据第一个顶部标签来决定怎么转发。

有些情况是需要两个标签的，两个典型是 MPLS VPN 和 AToM。MPLS VPN 要用两个标签，是因为在骨干中传输时，用一个，等出了骨干，再用另外一个。

那么设备收到一个 MPLS 标签数据包时，又怎么知道这个数据包是要查 IP 路由表来决定转发呢，还是查标签表做出转发呢？那是因为标签是在第二层帧和第三层数据包之间，第二层会在数据链路层的协议字段写上新的值，以说明后面是一个带有 MPLS 标签的报文，所以设备能够做出正确的转发决策。

MPLS 设备类型



能够理解 MPLS 标签并根据标签转发数据包的路由器称为 LSR

共有以下 3 种 LSR：

入站 LSR：接收没有标签的数据包，打上标签并发出

出站 LSR：接收带有标签的数据包，移除标签，并发出，出站和入站 LSR 都是边缘 LSR，所以它们同时连接了 IP 网络和 MPLS 网络。

链路中 LSR：接收到带标签的数据包，对其进行操作，然后按正确的接口交换出去，所以链路中的 LSR 只进行标签转发。

LSR 操作过程

LSR 可以执行三种操作：提取，添加和交换

提取，即从标签栈的顶部移除一个或多个标签，移除全部标签是出站 LSR 必须做的。

添加，向报文添加标签，如果没有标签，就加新的，入站 LSR 必须做的。

交换，收到一个有标签的报文，用新的标签交换到顶部，再发出，是链路中的 LSR 做的。

注：在 MPLS VPN 中，入站和出站 LSR 就是提供商边缘 PE，链路中 LSR 就是 P，术语 PE 和 P 在没有运行 MPLS VPN 时，也可使用。

标签交换路径 LSP

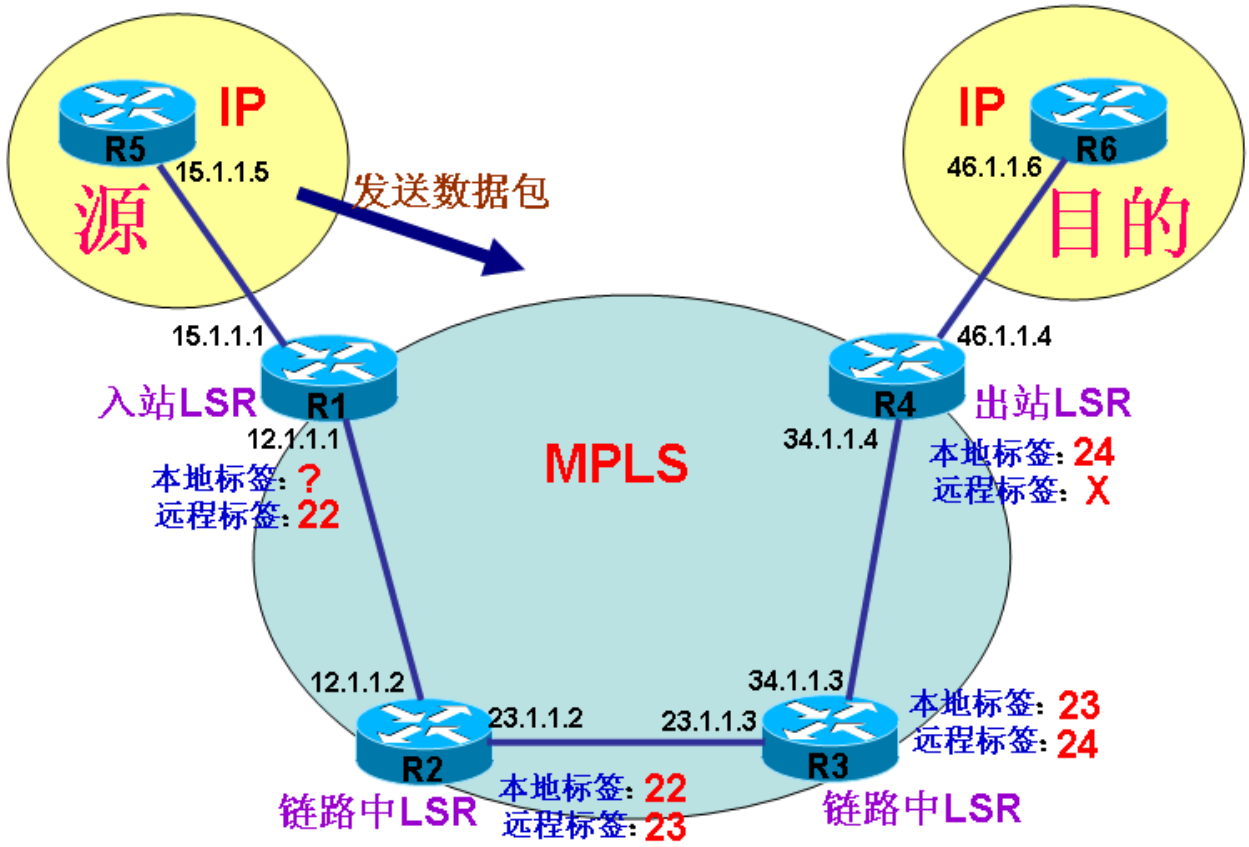
LSP 指的就是一个源到最终目的需要经过的路径，而在路径中，是要被修改多次标签，比如一个源到目的要分别被改为 20，50，35，68，那么也可以简单认为这两个点之间的 LSP 是 20-50-35-68。LSP 是 LSR 在 MPLS 网络中转发标签数据后的产生的，是标签报文穿越 MPLS 网络的路径。LSP 不需要记住，只需要知道是什么。LSP 中第一台 LSR 就是入站 LSR，最后一台就是出站 LSR，之间的就是链路中 LSR。

转发等价类（FEC）

FEC 可以认为是同一条路由，或者说是到达目标主机的路径是相同的，或者说是相同转发路径的数据流。同一个 FEC，所有标签都相同，并不是拥有相同标签的报文都是同一个 FEC，可能 EXP 值不同。报文属于哪个 FEC，由入站 LSR 决定。FEC 和 LSP 一样只需要知道是什么，不用记住。

注：一条 FEC 可以包含多个流，但不是一个流一个 FEC，比如一台主机在看新浪的网页，这是一个流，又在看新浪的视频，这又是一个流，这两个流在新浪发给远程主机时，走的路径应该是相同的，所以一个 FEC 有多个流，但是每个流并没有属于单独的 FEC。

MPLS 标签交换过程



虽然一条路由可以打了多标签，但是中间的 LSR 只根据最顶部的标签便可以做出转发。但是，每台 LSR 的转发表里都会为一条路由显示两个标签，一个是本地标签，一个是远程标签，要显示两个标签，是因为一台 LSR 收到数据包之后，就查看它的顶部标签，如果这个标签是某一条相应的本地标签，那么就从相应的接口发出去，同时在发出去的时候，就将数据包的顶部标签改为与这个本地标签对应的远程标签，这是每台 LSR 在传输时都必须改的，因为改了相应的远程标签，对下一台 LSR

做出正确转发是有帮助的。所以 LSR 自己对于某一条路由，别人要给数据打上什么标签，它才能够正确转发，它是要明确告诉邻居的，最后结果就是每一台 LSR 都会将这个能指导自己做过正确转发的标签发给邻居，邻居就认定这个发来的标签是远程标签，那么邻居在发出去数据包之前，都会改成这个远程标签，最后就能按正确路径转发了。

上图中，比如 R4 要看到数据包中有标签 24，就能够正确往 R6（目的地）的方向发，那么这个标签 24 对于 R4 来说就是本地标签，那么别人 LSR 把数据包发给 R4 之前，就得为它打上标签 24 才发出来，所以 R4 就把这个标签 24 告诉给它的邻居 R3，邻居 LSR 收到标签之后，就认为是远程标签，所以 R3 认为远程标签是 24，而自己要拿到一个数据后，是什么标签自己就会改成 24 发给 R4 呢，自己也会产生一个本地标签，可以看到，R3 产生的本地标签是 23，数据包中只要顶部标签是 23，就能够让 R3 把标签改成 24 后发给 R4，所以 R3 也把这个标签 23 发给了邻居 R2，邻居 R2 也同样为路由产生一个本地标签，是 22，也就是说 R2 收到数据包标签为 22 的，就改成 23，然后传给下一跳 R3，最后 R2 也把对自己有利的标签 22 发给了 R1，那么 R1 就知道发给 R2 之前，要把标签改成 22，要不然 R2 就会转发出错的。

而因为 R1 是入站 LSR，所以从外面发给它的数据包都是 IP 包，是不会有标签的，所以它是第一个向数据包加标签的 LSR，可以看出只要它一开始往数据包里加上标签 22 后发给 R2，最后就能一跳一跳地被发到 R6，因为中间 LSR 的标签大家都是协商好的。在数据包到达 R4 时，R4 就会将标签全部移除后发给 R6，因为是进入 IP 网络，没有必要再打标签发出去。要说明的是，该为一个路由条目产生什么样的本地标签，是 LSR 自己计算的，没有规则可言，从前面的传输过程可以看出，在 MPLS 网络中，LSR 每次收到数据包，都要将标签改成对下一跳有利的标签才转发出去，这个对下一跳有利的标签就是自己看到的远程标签，也称为出口标签，这个标签就是邻居告诉自己的，因为看到什么标签才能正确转发，邻居是知道的，所以告诉给其它邻居之后，才能保证最终路径的正确。

打标签

所有 LSR，根据路由表，将数据包打上标签，发出去，打的这个标签不能乱打，是有意义的，因为下一跳路由器要根据这个标签做出自己的决策，收到的路由器将顶部标签（上一跳路由器加的标签）去除，再加入出站标签，然后再下一跳路由器重复之前的动作，所以每一台路由器都需要对路由条目的标签达成共识，要不然这台路由器为数据包打上标签，给下一跳路由器，却不是按设想的接口发出去的，就

不能到达目标网络。因此，每一台 LSR 必须明确哪个出站标签来交换哪个入站标签，标签对于邻接 LSR 来说是本地有效，没有全局意义，这和帧中继的 PVC 是一个道理。所以，基于这些，需要有一种标签分发协议来为所有的 LSR 分发一个正确的标签，只有这样，LSR 才能根据标签将数据包正确地发到目标网络。

标签分发方式

标签分发有两种方式：

1．在已存在的 IP 路由协议中分发标签。

2．使用一种独立的协议来分发标签。

1.在现有的路由协议中分发

距离矢量路由协议，比如 EIGRP 很好做，直接绑在前缀上，因为这样的路由协议可以随意修改路由条目的内容。

链路状态路由协议，比如 OSPF 和 ISIS 就比较难，因为是发链路状态，而链路状态必须毫无更改地发给邻居，更改链路状态数据包是违背原理。

所以也就没有 IGP 做标签分发的工作，但是 BGP 却可以同时发前缀和标签（注：BGP 发标签是有条件限制的）。

2.标签分发协议

最终建议使用独立的分发协议，不影响路由协议，但需要在 LSR 上额外运行协议。

以下是可以分发标签的协议

标记分发协议（TDP）

标签分发协议（LDP）

资源预留协议（RSVP）

TDP 是思科专有的标记分发协议，已经被 LDP 所取代。

LDP 是 IETF 开发，功能广泛，所以只涉及 LDP，因为 TDP 已不用了。

RSVP 只用在 MPLS TE 中。

标签分发协议 LDP

对于 IP 路由表中每一条 IGP 前缀，每台 LSR 都会进行本地捆绑，也就是为路由条目加上一个标签，这个标签称为本地标签，到时收到一个数据包后，看到顶部标签如果是自己所拥有的本地标签，那么就根据这个标签从相应接口转发出去，所以邻居把数据包发给自己时，必须在数据包上写好自己对自己有利的标签，那么要怎样才能让邻居写上一个能让自己看了就正确转发的标签呢，这就得自己把这个标签告诉邻居，自己把本地标签发给邻居后，这个标签对于邻居来说就称为远程标签，每当邻居要发送数据包给自己的时候，就先把数据包的顶部标签改成远程标签，也就是改成之前发给邻居的标签，邻居改好标签后，把数据包发给我们，我们就能够做出正确转发了，那么邻居也会像我们一样，把自己的本地标签再发给它们的邻居，因为他要把数据正确发给我们，也是由它的邻居在数据包上写好标签告诉它的。所以在邻居和邻居之后，是要大家协商好每条路由该写上什么标签后发给邻居。LSR 把自己生成的本地标签，和邻居发过来的远程标签，不管是用的着的还是用不着的，统统都保存在一张标签表里，这个表称为 LIB 表（标签信息库）。LDP 就是用来发送标签的协议。

已经介绍过，MPLS 的标签，就像帧中继的 PVC 号码一样，只是每台设备唯一的，但是 MPLS 的标签，在设备上保持唯一还分两种，基于设备唯一和基于接口唯一。

如果基于设备，就是一条路由在一台设备上只有一个唯一的标签。

如果基于接口，就是一条路由在一台设备上，是每个接口都有一个唯一的标签。也就是说标签只要能保证在每个接口上是唯一的即可，在整台设备上可以重复多次。

但 LSR 可能有多个远程标签，因为可能有多个邻居 LSR。

路由协议 EIGRP 会将所有邻居发给自己的路由条目存放在拓朴表里，再从拓朴表里选中最优最好的放到路由表也供自己使用，当路由表中的条目失效后，再从拓朴表中拿出次优的使用。而 MPLS 标签交换的 LSR 也像 EIGRP 那样，会把所有邻居发来的标签都存放在 LIB 表里（就像 EIGRP 的拓朴表），然后从路由会条目的多个标签中选择一个最优的使用，这个选择方法可以通过 IGP 路由表，选到的下一跳是谁，那就用谁发来的标签，被选中的正在使用的标签，全部都是存放在 LFIB（标签转发信息库）表里的，就像 EIGRP 的路由表。

注：IOS 中，LDP 不会为 BGP 的 IPv4 前缀捆绑标签。

标签的选择都是根据 IGP 最优路径。

标签也可以不是 LDP 分发，比如 TE 中，由 RSVP 分发，在 MPLS VPN 中，由 BGP 分发。

因为 MPLS 的标签是加在数据包的二层帧头之上，三层包头之下，三层包头，被认为是上层协议，也就是有效负载,中间的 LSR 并不知道上层协议类型，因为标签不会写，但它也不需要知道，自己只会根据标签来做出转发决定；但是出站 LSR 需要知道，所以会为 FEC 分配一个本地标签，以用作报文的入站标签，这样就可以了解有效负载了。

LSR 在查看标签时，是要看基于接口还是设备，如果是基于接口，不能单看标签，还要看接口，如果是基于设备的那就只看标签。

注:在 IOS 中，所有的标签交换控制的 ATM（LC-ATM）接口都采用基于接口的模式，其它通通基于设备，也就是说 CCIE R&S 的考生，只需要关心基于设备的标签即可。

标签分发模式

LSR 在向邻居分发标签的时候，有三个需要注意的地方，这三个地方分为：

1．标签分发模式

2．标签保持模式

3．LSP 控制模式

1.标签分发模式：

是用来定义标签该什么时候发给邻居，分为两种方式：

（1）下游被动 DOD 模式

（2）下游主动 UD 模式

（1）在 DOD，即被动模式中，是 LSR 请求下游（路由表的下一跳）为某条路由分发标签，也就是说一台 LSR 并不知道某些路由自己该写上什么标签后发给下一跳，所以这时就去问邻居要，要来的自己就作为该路由的远程标签存放。

（2）在 UD，即主动模式中，LSR 不需要为路由请邻居请求标签，标签是邻居会主动发过来的，不用请求，所以在 UD 模式中，LSR 发现一条路由，就马上将自己的本地标签发送给邻居作为远程标签。

在此可以得出一个结论，在 DOD 中，LSR 会向路由条目的下一跳请求标签，所以 LIB 只显示一个远程标签。在 UD 中，因为可能有多个邻居发送标签过来，所以一条路由可以看到多个标签。

注：IOS 除了 LC-ATM，全部使用 UD 模式，也就是说有多个标签从邻居发来。

2.标签保存模式

用来定义 LSR 在将标签保存时，该保存多久，分两种方式：

（1）自由的标签保持模式（LLR）

（2）保守的标签保持模式（CLR）

（1）在 LLR 中，LSR 将所有的标签存放在 LIB 中，然后使用的放到 LFIB 中，不使用的也保存在 LIB 中，当路由变化时，马上从 LIB 中找到新的。

（2）在 CLR 中，LSR 将用到的标签放入 LFIB 之后，不会在 LIB 中保存任何标签。

注：IOS 中，除了 LC-ATM 接口，其它所有都使用 LLR，也就是能在 LIB 中看到标签。

3.LSP 控制模式

用来定义 LSR 什么时候应该为一条路由创建标签，创建出的这个标签就是自己的本地标签，发给邻居之后，邻居就称其为远程标签。分两种创建方式：

（1）独立于 LSP 的控制模式

（2）非独立于 LSP 的控制模式

（1）LSR 可以独立于其它 LSR 创建本地标签，称为独立模式，路由器在路由表中发现一个路由，就马上为该路由创建一个标签。

（2）在非独立模式时，LSR 只有意识到它是某 FEC 的出站 LSR 时，或者从下一跳收到某路由的标签时，才会为路由条目创建本地标签，然后发给邻居作为远程标签，邻居收到后，然后又会再创建了发给它的邻居。

由上可以看出，独立于 LSP 的模式在 LSP 中还没有让所有的 LSR 完成标签，有些 LSR 就开始标签转发，所有这些数据包有可能不能被正确转发，有可能被丢弃。而非独立的模式，只有从邻居收到标签了，开始自己的标签，所以自己使用标签转发后，下一跳邻居肯定是能接受的，不可能因为下一跳不认识标签而被丢弃 。

注:IOS 使用独立的 LSP 控制模式，也就是说发现一条 FEC，就马上创建标签，ATM 除外。

在标签转发中，LSR 查看标签前 20bit，并在 LFIB 中查找相应的值。

在 IP 转发中，查看 IP 地址，在 CEF 表中做出转发。

路由器可以查看第二层头部的协议字段，就知道是标签报文，然后查找 LFIB，则带标签出去，所以查 CEF 就等于是查 IP，则不带标签出去。

路由器要为路由条目打上标签，就必须有功能支持改写数据包包头，CEF 是唯一一种可以用于标签报文转发模式的，CEF 是可以改写数据包包头的，所以启用 MPLS 时，必须在路由器上开 CEF，否则无标签。

比如查看路由 10.1.1.0 在 CEF 中的操作，可以使用命令

show ip cef 10.1.1.0 查看这条路由的过程

MPLS 负载均衡

在 IPv4 存在多条相同 metric 出口时，标签的出站也会对应多个接口，出站的标签可以是相同的，也可以不同的；如果下一跳是同台设备，肯定相同，是不同设备会不同，也就独立分配。

当 LSR 中某条路由有多个下一跳，如果是有标签的和无标签的，无标签的不走，是考虑到 MPLS VPN 中数据会丢包，因为在 MPLS VPN 网络中，P 路由器是没有两边私有网络的 IP 路由的，所以无法路由，最终造成丢包。

MPLS 未知标签

通常 LSR 只接收和发送带标签的并且能理解的数据包，因为某些原因，标签没找到的话，IOS 是默认采用丢弃行为，如果找不到标签也传，也不能保证别人能传不丢弃，所以就自己开始丢弃。

MPLS 保留标签

MPLS 标签范围中，并不是所有的标签都是可以随便用的，有些是保留的，范围是 0-15，有特殊作用，0 是显式空（null）,3 是隐式空，1 是路由器报警标签，14 是 OAM 报警，其它还没定义。下面是某些保留标签的重要用途：

隐式空 3 标签

在 MPLS 网络中，P 路由器是完全按照标签交换的，而边缘路由器 PE 是同时连接了 MPLS 网络和 IP 网络，因此，一个数据包在 MPLS 网络中传输到 PE 路由器的时候，PE 路由器的工作是：结束标签交换过程，从而转入 IP 网络，而转入 IP 网络就要执行 IP 地址的查找。那么从此可以看出，一个标签数据包到达 PE 路由器之后，PE 路由器第一步开始根据数据包的标签去查找 LFIB 表，通过查找 LFIB 之后发现已经不再是标签交换了，第二步就马上转入查 IP 路由表，最终在 IP 路由表里查到了结果，从 IP 网络中发出去。很明显，PE 路由器既然最后不可能使用标签交换，而要使用 IP 交换的，又何必去查了 LFIB 表才知道结果呢。所以就考虑到一个方法，能不能让 PE 的上一跳路由器不要为数据包打标签，直接改成 IP 数据包就行了，这样上一跳

路由器没有写标签，那么 PE 在收到数据包之后，就能马上查 IP 路由表而做出转发。在这里，要让 PE 的上一跳路由器不要为数据打标签而直接改成 IP 数据包，这还得需要 PE 来告诉它才行。正常情况下，PE 路由器是告诉上一跳正常的标签，上一跳将这个标签变成远程标签，但现在，PE 路由器就不应该告诉上一跳正常的标签，它告诉的是隐式空（标签号为 3）标签，所以收到标签为 3 的 LSR，就不会在数据包发给下一跳时打上标签。这种终点使用隐式空标签来告诉上一跳不要打标签的行为叫倒数第二跳移除（PHP）行为，所以一台收到隐式空标签的 LSR，相应出口就不再是一个远程标签，而应该在 outgoing 显示为 pop。这种上一跳标签移除称为标签弹出。

注：在 IOS 中，PHP 这种行为是默认的，但只会为直连路由和聚合路由通告隐式空 3，但 3 不会明写。

显示空标签

显示空的功能是在隐式空的基础上的，IPv4 标签号为 0，IPv6 为 2。

因为标签中 EXP 用于 QOS，前一跳移除后，这些信息也没了，可能希望保留，所以是上一跳将标签变为 0，来告诉终点不用为 0 查找 LFIB，只看 EXP，所以只关心 QOS 效果，这样也省事。

路由器报警标签 1，只是需要特别注意，并且软件转发。

未保留的都可以用，20 个 bit，是 16-1048575,IOS 默认是 16-10 0000,IGP 够了，但 BGP 可能不够，可以查看和修改标签范围。

MPLS TTL 行为

在正常情况下，当数据包的 TTL 值减到 0 时，路由器会向源发 ICMP 类型 11 和代码 0(时间超时)的数据包，来告诉源主机目标超时不可达。所以 TTL 无论对于 IP 网络还是 MPLS 网络都是非常重要的。

在数据包从 IP 网络进入 MPLS 网络时，IP 刚进来，以前的 TTL 是多少，PE 减 1 后，写到标签的 TTL 位，在出 MPLS 网络时，PE 再看标签中的 TTL 是多少，肯定比 IP 原来的 TTL 值小，减 1 后写回去，如果 TTL 值比 IP 原来的 TTL 值还大，就不正常，就不写了。

标签到标签，添加和交换等操作，也是减 1 后再复制，中间 P 路由器只修改顶部标签中的 TTL，顶部以下的标签是不动的。

如果遇见一个数据包 TTL 值为 0，普通的 MPLS 网络就是沿原来的 LSP 回去，只有是 IPv4 和 IPv6 才会，其它的丢弃。

MPLS VPN 中 TTL 没有后，是由终点 PE 或 CE 发回的，因为 P 没有源主机的路由，所以无法发送超时 ICMP。

MPLS MTU

OSI 第三层网络层协议的包头是在第二层帧头之上的，也就是说在封装二层帧头的时候，是将数据内容和三层包头全部作为数据封装在里面的，对于二层来说，之前的数据最大是多少，就是由 MTU 来决定的，所以正常情况下 MTU 就是第三层数据和包头的最大尺寸，这时无需分段就能传输，如果比 MTU 大，就得分段后传输。但是 MPLS 的标签是在二层帧头之后的，所以二层帧头将标签的大小和三层包的内容累加到一起作为数据封装的，因为三层包的所有内容正好和 MTU 一样大，在此基础上加上 MPLS 标签的话，就肯定比额定的 MTU 要大，所以这时 MPLS 的标签数据是会被分段后传输的，如果不想被分段，就得更新 MTU 的大小。（一般 MPLS 数据包加最多两个标签，一个标签 4 字节，所以只要改成比正常 MTU 多 8 字节即可。）而改 MTU 还必须在 MPLS 网络中所有设备上进行更改，除非允许分段。

MPLS 最大接收单元（MRU）

此内容无须配置

查看： sh mpls forwarding-table 1.1.1.1 detail

MPLS 默认超过 MTU 的数据包是和 IP 数据包一样要分段传输的，分段就是 LSR 移除标签，对 IP 数据分好相应大小后，再将原来的标签加到每个包，如果 IP 包头设置了不分段（DF），LSR 就丢掉报文，然后返回一个需要分段的 ICMP（不分段位设置为类型 3，代码 4），然后沿来的 LSP 发回去。

MTU 路径发现

（自动执行）

查看从源到目的的最小 MTU，就是类似窗口的机制，数据包试着发出去，如果被丢了，就减小后再发，如果再丢就再减再发，直到能正常发到目的地为止。有时这个不太好用，因为 ICMP 不能返回，可能是防火墙挡住了。

标签分发

要让 IGP 全部都支持标签分发并且相互协同工作，不现实，所以新分发协议要独立于所有路由协议，那么就使用 LDP，但是 BGP 可以广泛使用，范围大，最后 BGP 就自己发标签。LDP 不为 BGP 的 IPv4 路由发送标签。

LDP 运行

LDP 运行时有四大功能：

（1）运行 LDP 的 LSR 发现

（2）会话的建立和维护

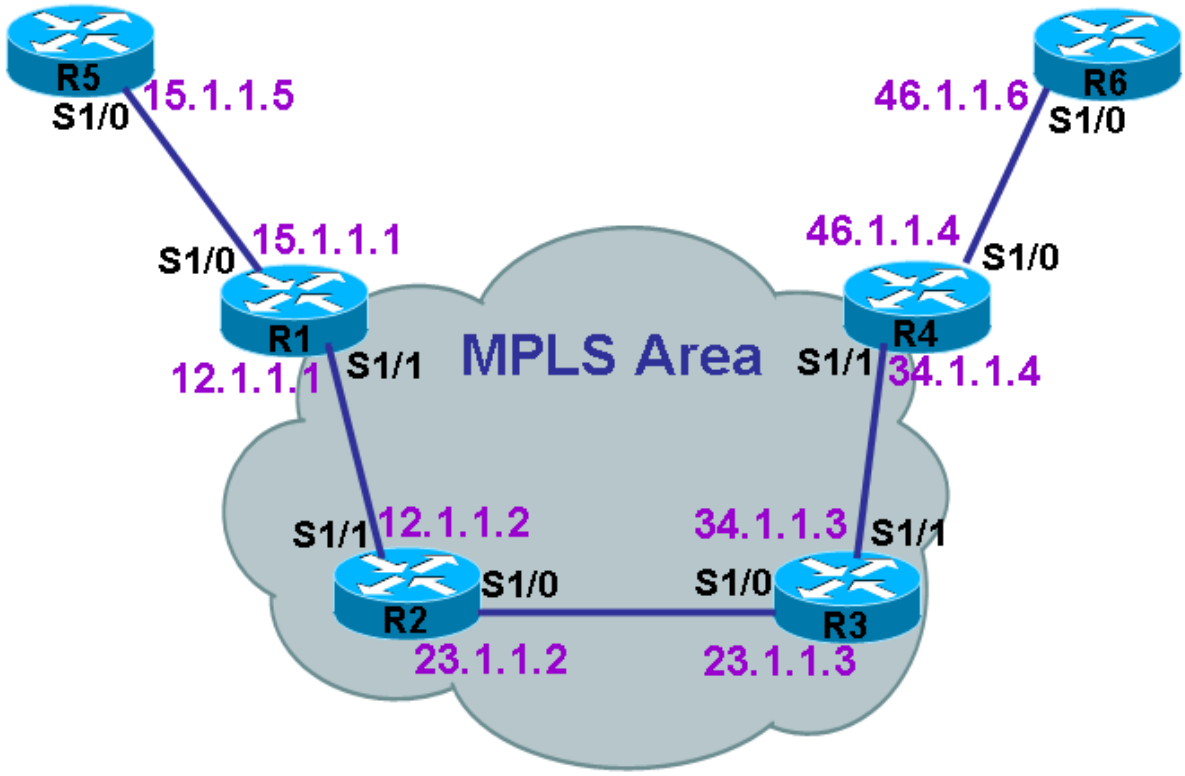
（3）标签映射通告

（4）使用通知来进行管理

LDP 是需要像 OSPF 那样建邻居的，使用 hello 包发现和维护邻居关系，LDP 会在启用了的接口上发送 hello 来找邻居，发送 hello 用 UDP 646，目的地为 224.0.0.2，hello 时间和保持时间分别是 5 秒 15 秒。这个 hello 包是不能跨网段传递的，而这个 hello 包被称为 LDP Link Hello。

LDP 在 LSR 之间除了建立邻居关系之外，还要建立 LDP 会话，建 LDP 会话就是用来交换标签的，使用的是 TCP 连接。而这个会话也只能和直连邻居建立，这样会话被称为 LDP sessions。LDP 会话的 hello 和超时分别是 60 秒和 180 秒。如果 LDP 邻居关系丢失，那么 LDP 会话也会断开。

配置 MPLS



说明：

以上图为拓朴，配置 MPLS，MPLS 网络区域的范围是 R1、R2、R3、R4，而 R1 和 R4 同时连接 MPLS 区域和 IP 网络，最终数据包在 MPLS 区域内传递时，我们将看到标签交换的效果。在开始配置之前，需要申明的是，每台路由器上都已配置 loobback0，地址分别为 X.X.X.X/32,其中 X 表示设备号码，比如 R2 的 loopback0 地址为 2.2.2.2/32,R5

的 loopback0 地址为 5.5.5.5/32；在所有设备中，已经启用 OSPF 协议，除了每台设备的接口 loopback0 没有放进 OSPF 进程以外，其它所有接口均在 OSPF 进程里通告。

1.查看和修改标签范围（可选配置）

（1）看默认标签数量:

R1#sh mpls label range

Downstream Generic label region: Min/Max label: 16/100000

R1#

说明：默认标签范围是 16 到 100000

（2）改标签范围：

R1(config)#mpls label range 16 1010000

R1#sh mpls label range

Downstream Generic label region: Min/Max label: 16/1010000

R1#

说明:已将标签范围改成：16 到 1010000

2.查看和修改 MTU （可选配置）

（1）查看路由器接口 MTU:

R1#show mpls int s1/1 detail

（2）修改路由器接口 MTU:

R1(config)#int s1/1

R1 (config-if)#mpls mtu 1508

或

R1 (config)#int s1/1

R1 (config-if)#mtu 2000

注:某些 IOS 下的接口不能设置 MTU，只能分段传输。

（3）修改交换机支持小巨型帧：

sw（config）#system jumbomtu 2000

sw(config)#system mtu 2000

3.全局开启 CEF （必须配置）

注：某些 IOS 版本已默认开启的，可跳过此步，请以自身 IOS 为准。

r1(config)#ip cef

4.配置 LDP （必须配置）

（1） 全局启用 LDP:

说明：如果全局启用 LDP，就将在此路由器的所有接口都开启 LDP，但也可以选择只在某接口开启。

r1(config)#mpls label protocol ldp

（2） 接口启用 LDP:

说明：如果路由器并非全部接口都需要开启 LDP，则只在相应接口开启。

r1(config)#int s1/1

r1(config-if)# mpls label protocol ldp

（3）在接口下开启发 hello 包找邻居：

r1(config)#int s1/1

r1(config-if)#mpls ip

说明：接口上配置 mpls ip 就算打开了

IOS 有时还在使用 tag-switching 来代替 mpls ip,但功能是一样的，这两个命令相等。

注：请按上述配置 LDP 的方法，在 MPLS 区域内的所有路由器所有相关接口开启 LDP 并发出 hello 包，以方便 LDP 邻居的建立。

附：按以上拓朴，总结出需要的配置为：

R1：

r1(config)#int s1/1

r1(config-if)# mpls label protocol ldp

r1(config-if)#mpls ip

R2：

R2(config)#mpls label protocol ldp

R2(config)#int s1/0

R2(config-if)#mpls ip

R2(config-if)#exit

R2(config)#int s1/1

R2(config-if)#mpls ip

R3：

R3(config)#mpls label protocol ldp

R3(config)#int s1/0

R3(config-if)#mpls ip

R3(config-if)#exit

R3(config)#int s1/1

R3(config-if)#mpls ip

R4：

R4(config)#int s1/1

R4(config-if)#mpls label protocol ldp

R4(config-if)#mpls ip

5.查看 LDP 简单信息

（1）可以查看哪些接口开启了 mpls：

r1#sh mpls interfaces

Interface IP Tunnel Operational

Serial1/1 Yes (ldp) No Yes

r1#

说明：可以看出，R1 相关接口 S1/1 已经运行在 LDP 下。（其它设备接口状态略过！）

（2）查看看 LDP 详情，包括包含 hello 时间，会话时间：

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 180 sec; keep alive interval: 60 sec

Discovery hello: holdtime: 15 sec; interval: 5 sec

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

r1#

说明：可以看到，默认 hello 和 hold 分别是 5s 和 15s，会话时间 hello 和 hold 分别是 60s 和 180s。

（3）修改时间机制（并不建议修改）：

注:两边保持时间不一样，选用小的一端，改了多个，也是用小的而不是最新的。

改 hello:

r1(config)#mpls ldp discovery hello interval 3

r1(config)#exi

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 180 sec; keep alive interval: 60 sec

Discovery hello: holdtime: 15 sec; interval: 3 sec

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

说明:可以看到 hello 时间被改成了 3s

r1#sh mpls ldp discovery detail

Local LDP Identifier:

12.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 3000 ms; Transport IP addr: 12.1.1.1

LDP Id: 2.2.2.2:0; no host route to transport addr

Src IP addr: 12.1.1.2; Transport IP addr: 12.1.1.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r1#

说明:也可以看到 hello 时间被改成了 3s

再改：

r1(config)#mpls ldp discovery hello interval 8

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 180 sec; keep alive interval: 60 sec

Discovery hello: holdtime: 15 sec; interval: 8 sec

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

r1#

说明:可以看到 hello 时间被改成了 8s

r1#sh mpls ldp discovery detail

Local LDP Identifier:

12.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 12.1.1.1

LDP Id: 2.2.2.2:0; no host route to transport addr

Src IP addr: 12.1.1.2; Transport IP addr: 12.1.1.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r1#

说明:在这可以看到 hello 时间还是选默认 5s，因为这个小。

改会话时间（不建议）：

r1(config)#mpls ldp holdtime 150

```
r1#sh adjacency detail

*Mar  1 01:32:03.063: %SYS-5-CONFIG_I: Configured from console by console

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 150 sec; keep alive interval: 50 sec

Discovery hello: holdtime: 15 sec; interval: 8 sec

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

r1#
```

说明可以看到 hold 时间被改成了 150s。

6.查看 LDP 邻居相关信息

(1)在 R1 上查看 LDP discovery 情况：

```
r1#sh mpls ldp discovery detail

 Local LDP Identifier:

    1.1.1.1:0
```

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0; no route to transport addr

Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

r1#

说明:Local LDP Identifier 是每台 LSR 都必须有的，这个 ID 用 6 个字节表示，前 4 个节字称为 Rotuer-ID，先选 loopback 地址最大的，然后是物理接口，它的选举方法和 OSPF Rotuer-ID 相同，后面 2 个字节是表示标签空间的，也就是标签是基于设备还是基于接口，如果是基于设备，就是 0，可以从上面看出，1.1.1.1:0 中，1.1.1.1 表示 R1 的 Rotuer-ID，而 0 表示标签是基于设备的。再看后面还有个 Transport IP ，而这个 IP 默认是选用 Rotuer-ID 的地址，这个地址在建邻居时非常重要，是会话的源地址，如果这个地址对方没有路由可达，那么就不可能建起邻居。所以一定要保证双方 Transport IP 是路由相通的。从上面结果中还可以看出，R1 已经收到了对方 R2 的 hello，对方 Transport IP 是 2.2.2.2，也就是对方的 loopback0 地址，而因为这个地址不在 OSPF 进程里，所以 R1 不能到达，也就不能建邻居，后面提示为“no route to transport addr”。

（2）在 R2 上查看 LDP discovery 情况：

r2#sh mpls ldp discovery detail

Local LDP Identifier:

2.2.2.2:0

Discovery Sources:

Interfaces:

```
    Serial1/0 (ldp): xmit/recv
      Enabled: Interface config
      Hello interval: 5000 ms; Transport IP addr: 2.2.2.2
      LDP Id: 3.3.3.3:0; no route to transport addr
        Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3
        Hold time: 15 sec; Proposed local/peer: 15/15 sec
    Serial1/1 (ldp): xmit/recv
      Enabled: Interface config
      Hello interval: 5000 ms; Transport IP addr: 2.2.2.2
      LDP Id: 1.1.1.1:0; no route to transport addr
        Src IP addr: 12.1.1.1; Transport IP addr: 1.1.1.1
        Hold time: 15 sec; Proposed local/peer: 15/15 sec
r2#
```

说明:可以看出，R2 的 Rotuer-ID 是 2.2.2.2，这个地址也就是 Transport IP，而 R1 的 Transport IP 是 1.1.1.1，从上面也看出这两个地址是路由上互不相通的，所以不可能建立 LDP 邻居。

(3)解决邻居建立问题：

说明:要解决邻居建立问题，就要让双方的 Transport IP 能够相通，而 Transport IP 就是选用 Rotuer-ID 的 IP 地址，可以修改 Rotuer-ID 为可路由的接口，即可解决 Transport IP 互通。

改 R1Router-Id 为 S1/1 接口地址

r1(config)#mpls ldp router-id serial 1/1 force　　force 说明立即生效

r1(config)#exi

查看结果：

r1#sh mpls ldp discovery detail

Local LDP Identifier:

12.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 12.1.1.1

LDP Id: 2.2.2.2:0; no route to transport addr

Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

r1#

说明:可以看到 Rotuer-ID 已经改成接口 S1/1 的地址 12.1.1.1，Transport IP 也随即变成了 12.1.1.1。

R1 的 Transport IP 对 R2 来说已经可达了，可是 R2 的 Transport IP 还是自己的 loopback 口地址，R1 不能到达，所以还是不能建邻居。Transport IP 默认选用 Rotuer-ID 的地址，但我们可以明确指定 Transport IP 为某个接口的地址，这次我们直接改 R2 的 Transport IP 地址为 S1/1 的接口地址，Rotuer-ID 不变：

先看没改之前：

```
r2#sh mpls ldp discovery detail
 Local LDP Identifier:
    2.2.2.2:0
    Discovery Sources:
    Interfaces:
      Serial1/0 (ldp): xmit/recv
        Enabled: Interface config
        Hello interval: 5000 ms; Transport IP addr: 2.2.2.2
        LDP Id: 3.3.3.3:0; no route to transport addr
          Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3
          Hold time: 15 sec; Proposed local/peer: 15/15 sec
      Serial1/1 (ldp): xmit/recv
        Enabled: Interface config
        Hello interval: 5000 ms; Transport IP addr: 2.2.2.2
        LDP Id: 12.1.1.1:0; no host route to transport addr
          Src IP addr: 12.1.1.1; Transport IP addr: 12.1.1.1
          Hold time: 15 sec; Proposed local/peer: 15/15 sec
          Reachable via 12.1.1.0/24
r2#conf t
```

现在修改：

r2(config)#int s1/1

r2(config-if)#mpls ldp discovery transport-address interface

r2(config-if)#exi

再看：

r2#sh mpls ldp discovery detail

Local LDP Identifier:

2.2.2.2:0

Discovery Sources:

Interfaces:

Serial1/0 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 2.2.2.2

LDP Id: 3.3.3.3:0; no route to transport addr

Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 12.1.1.2

LDP Id: 12.1.1.1:0; no host route to transport addr

```
        Src IP addr: 12.1.1.1; Transport IP addr: 12.1.1.1

        Hold time: 15 sec; Proposed local/peer: 15/15 sec

        Reachable via 12.1.1.0/24

r2#
```

说明:可以看到，s1/1 上 Transport IP 已经不再是 Router-ID 的地址，已经被改成本接口地址了。但是接口 S1/0 还是使用原来 Router-ID 的地址。

但是邻居还是不会有，因为 R2 上直接改接口为 Transport IP，是要重启进程才能生效的：

```
r2#cle mpls ldp neighbor *

r2#
```

再查看邻居：

```
r2#sh mpls ldp neighbor

   Peer LDP Ident: 12.1.1.1:0; Local LDP Ident 2.2.2.2:0

     TCP connection: 12.1.1.1.646 - 12.1.1.2.11155

     State: Oper; Msgs sent/rcvd: 9/9; Downstream

     Up time: 00:00:43

     LDP discovery sources:

      Serial1/1, Src IP addr: 12.1.1.1

     Addresses bound to peer LDP Ident:

      15.1.1.1       12.1.1.1       1.1.1.1

r2#
```

说明:可以看到，邻居已经有了，并且可以看出端口号是 646。

（4）再来关心 R2 和 R3 的邻居：

说明：因为 R2 现在连 R3 的接口 S1/0 的 Transport IP 还是使用 Router-ID 的地址 2.2.2.2，而 R3 也到不了 2.2.2.2，所以 R2 和 R3 之间的 LDP 邻居关系是建不起来的，我们还是像 R2 和 R1 建邻居那样，把 S1/0 接口的 Transport IP 改成使用本接口的地址。

```
r2(config)#int s1/0

r2(config-if)#mpls ldp discovery transport-address interface

r2(config-if)#exi
```

而 R3 的 Transport IP 还是使用自己的 Router-ID 地址 3.3.3.3，这个地址 R2 也是无法到达的，在 R3 上也可以通过将该地址放进 OSPF 进程来使 R2 能够 ping 通，从而建立 LDP 邻居。

```
r3(config)#router ospf 2

r3(config-router)#network 3.3.3.3 0.0.0.0 area 0

r3(config-router)#exi
```

在 R2 上查看建 LDP 邻居的源地址：

```
r2#sh mpls ldp discovery detail
 Local LDP Identifier:
    2.2.2.2:0
    Discovery Sources:
    Interfaces:
       Serial1/0 (ldp): xmit/recv
          Enabled: Interface config
          Hello interval: 5000 ms; Transport IP addr: 23.1.1.2
          LDP Id: 3.3.3.3:0
```

Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 3.3.3.3/32

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 12.1.1.2

LDP Id: 12.1.1.1:0; no host route to transport addr

Src IP addr: 12.1.1.1; Transport IP addr: 12.1.1.1

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r2#

说明:可以看到，R2 连 R3 的接口 S1/0 的 Transport IP 已经成功改成了 23.1.1.2。

但是 R2 还是不会有 R3 的邻居，所以按以前方法重置 LDP 进程，再看就会有了：

来查看重置后的邻居状态：

r2#sh mpls ldp neighbor

Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0

TCP connection: 3.3.3.3.646 - 23.1.1.2.61206

State: Oper; Msgs sent/rcvd: 10/9; Downstream

Up time: 00:00:09

LDP discovery sources:

Serial1/0, Src IP addr: 23.1.1.3

```
        Addresses bound to peer LDP Ident:
          23.1.1.3        34.1.1.3        3.3.3.3
      Peer LDP Ident: 12.1.1.1:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 12.1.1.1.646 - 12.1.1.2.44954
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:00:03
        LDP discovery sources:
          Serial1/1, Src IP addr: 12.1.1.1
        Addresses bound to peer LDP Ident:
          15.1.1.1        12.1.1.1        1.1.1.1
r2#
```

说明:在 R2 上可以看见和 R3 的 LDP 邻居关系已经建立。

（5）R3 跟 R4 的邻居关系：

说明：R3 的 Router-ID 已经通告进 OSPF 进程，所以 R4 也能到达了，那么 R4 也选择将 Loopback 地址放 loop 进 OSPF 进程来完成和 R3 的 LDP 邻居关系建立。

说明:最终保证所有 LDP 邻居建立（R1 和 R2 的邻居，R2 和 R3 的邻居，R3 和 R4 的邻居全部都有）。

7.查看标签交换相关信息

说明:先以 R4 的 loopback0 地址 4.4.4.4/32 这条路由为例，来看别的路由器对这条路由的标签状况。

（1）在 R4 上查看 LFIB，看路由 4.4.4.4 的情况：

```
r4#sh mpls forwarding-table

Local  Outgoing    Prefix          Bytes tag  Outgoing    Next Hop

tag    tag or VC   or Tunnel Id    switched   interface

16     Pop tag     23.1.1.0/24     0          Se1/1       34.1.1.3

17     16          12.1.1.0/24     0          Se1/1       34.1.1.3

18     18          15.1.1.0/24     0          Se1/1       34.1.1.3

19     Pop tag     3.3.3.3/32      0          Se1/1       34.1.1.3

r4#
```

说明:可以看出，R4 上对自己直连接口的标签并没有出现在 LFIB 表中，因此该路由不需要进行标签交换，属正常。

（2） 查看 R4 上 CEF 对 4.4.4.4 的处理情况：

注：所有路由的处理，即使是打标签，都要由 CEF 来处理。

```
r4#sh ip cef 4.4.4.4

4.4.4.4/32, version 15, epoch 0, connected, receive

  tag information set

    local tag: implicit-null

r4#
```

说明:可以看出，R4 的 CEF 对 4.4.4.4 这条路由打的本地标签是 implicit-null(隐式空标签，对于隐式空标签的解释，请参见 MPLS 正文内容)，本地路由发给邻居之后，就成为了邻居的远程标签，所以邻居到达 4.4.4.4 的路由标签都应该是隐式空标签。

（3）查看 R3 的 CEF 对 4.4.4.4 的处理情况：

```
r3#sh ip cef 4.4.4.4
```

4.4.4.4/32, version 19, epoch 0, cached adjacency 34.1.1.4

0 packets, 0 bytes

tag information set

local tag: 19

via 34.1.1.4, Serial1/1, 0 dependencies

next hop 34.1.1.4, Serial1/1

valid cached adjacency

tag rewrite with Se1/1, 34.1.1.4, tags imposed: {}

r3#

说明:很明显，4.4.4.4 在 R4 上的本地标签（implicit-null）发给 R3 之后，就成为了 R3 的远程标签，可以看到，因为 R4 发来时是隐式空，所以 R3 就不能为 4.4.4.4 打任何标签，所以最终结果的空的。而 R3 对于 4.4.4.4 这条路由是要生成自己的本地标签的，因为自己要对这条路由使用标签交换，可以看到 R3 自己给 4.4.4.4 打的本地标签是 19，那么这个标签发给别的邻居之后，就该变成远程标签 19。

（4）再查看 R3 的 LFIB 表：

r3#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	12.1.1.0/24	0	Se1/0	23.1.1.2
17	Pop tag	46.1.1.0/24	0	Se1/1	34.1.1.4
18	18	15.1.1.0/24	0	Se1/0	23.1.1.2
19	Pop tag	4.4.4.4/32	0	Se1/1	34.1.1.4

r3#

说明:可以看到 R3 对于路由条目 4.4.4.4 的本地标签是 19，这是要发给别人的，而出的标签是 Pop tag，也就是要移除标签，也就是当 R3 收到邻居发来一个数据包，如果查看结果为顶部标签是 19，那么自己就将该标签移除后，再从 S1/0 发出去。

（5） 查看 R3 的 FIB 表：

注：LSR 对于路由条目所有的标签都是保存在 FIB 表里的，用到的才放进 LFIB，所以在 LFIB 表里将可能看到路由的多个标签，但是只有一个标签是正被使用的。

```
r3#sh mpls ip binding
  2.2.2.2/32
      out label:    imp-null   lsr: 2.2.2.2:0
  3.3.3.3/32
      in label:     imp-null
      out label:    19         lsr: 2.2.2.2:0
      out label:    19         lsr: 4.4.4.4:0
  4.4.4.4/32
      in label:     19
      out label:    imp-null   lsr: 4.4.4.4:0       inuse
      out label:    20         lsr: 2.2.2.2:0
  12.1.1.0/24
      in label:     16
      out label:    imp-null   lsr: 2.2.2.2:0       inuse
      out label:    17         lsr: 4.4.4.4:0
  15.1.1.0/24
```

in label: 18

out label: 18 lsr: 2.2.2.2:0 inuse

out label: 18 lsr: 4.4.4.4:0

23.1.1.0/24

in label: imp-null

out label: imp-null lsr: 2.2.2.2:0

out label: 16 lsr: 4.4.4.4:0

34.1.1.0/24

in label: imp-null

out label: 16 lsr: 2.2.2.2:0

out label: imp-null lsr: 4.4.4.4:0

46.1.1.0/24

in label: 17

out label: 17 lsr: 2.2.2.2:0

out label: imp-null lsr: 4.4.4.4:0 inuse

r3#

说明:从以上结果看出，路由条目 4.4.4.4 的 in 标签，即本地标签是 19,而出的标签有两个，分别是 20 和 imp-null ，而 imp-null 后面有关键字 inuse，也就表示 imp-null 是正在使用中的。

（6） 查看 R2 的 LFIB:

r2#sh mpls forwarding-table

Local Outgoing Prefix Bytes tag Outgoing Next Hop

```
tag    tag or VC   or Tunnel Id      switched    interface

16     Pop tag     34.1.1.0/24       0           Se1/0       23.1.1.3

17     17          46.1.1.0/24       0           Se1/0       23.1.1.3

18     Pop tag     15.1.1.0/24       0           Se1/1       12.1.1.1

19     Pop tag     3.3.3.3/32        0           Se1/0       23.1.1.3

20     19          4.4.4.4/32        0           Se1/0       23.1.1.3

r2#
```

说明:正因为 R3 收到标签为 19 的数据包，才会移除标签后从 S1/1 发给 R4，所以 R2 肯定是应该将 R3 发来的本地标签 19 变成自己的远程标签 19，以上结果也显示，R2 对 4.4.4.4 打的出标签正是 19，而本地标签是 20，也就是说当 R2 收到一个顶部标签为 20 的数据包，就将标签换成 19 后从 S1/0 发出去，发出去正好就是发给了 R3，而 R3 根据自己的处理最终发到 R4。R2 给 4.4.4.4 打的本地标签 20，发给谁也就会变成谁的远程标签 20。

（7）查看 R2 的 CEF 对 4.4.4.4 的处理情况：

```
r2#sh ip cef 4.4.4.4

4.4.4.4/32, version 25, epoch 0, cached adjacency 23.1.1.3

0 packets, 0 bytes

  tag information set

    local tag: 20

    fast tag rewrite with Se1/0, 23.1.1.3, tags imposed: {19}

  via 23.1.1.3, Serial1/0, 0 dependencies

    next hop 23.1.1.3, Serial1/0

    valid cached adjacency
```

tag rewrite with Se1/0, 23.1.1.3, tags imposed: {19}

r2#

说明:可以看出，R2 将 4.4.4.4 的本地标签改成 20，将出的标签改成 19（19 正是 R3 上的本地标签）。所以和上面的理论全部对应。

（8）查看 R1 的 LFIB 表：

r1#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	16	34.1.1.0/24	0	Se1/1	12.1.1.2
17	Pop tag	23.1.1.0/24	0	Se1/1	12.1.1.2
18	17	46.1.1.0/24	0	Se1/1	12.1.1.2
19	19	3.3.3.3/32	0	Se1/1	12.1.1.2
20	20	4.4.4.4/32	0	Se1/1	12.1.1.2

r1#

说明:从上面结果中看出，正因为 R2 对 4.4.4.4 打的本地标签是 20，所以发给 R1，就变成 R1 的远程标签，即出口标签是 20 了，而 R1 给 4.4.4.4 打的本地标签也是 20，这是自己随意打上去的，从这个结果表明，当 R1 收到一个数据包标签为 20 的，那么就改成标签 20 从 S1/1 发出动。

（9）在 R1 上跟踪路由 4.4.4.4 的传输情况：

r1#traceroute 4.4.4.4

Type escape sequence to abort.

Tracing the route to 4.4.4.4

1 12.1.1.2 [MPLS: Label 20 Exp 0] 156 msec 220 msec 156 msec

2 23.1.1.3 [MPLS: Label 19 Exp 0] 168 msec 188 msec 196 msec

3 34.1.1.4 160 msec * 140 msec

r1#

说明:因为对于条目 4.4.4.4，R4 给 R3 的标签为隐式空，即不打标签，R3 给 R2 的标签为 19,所以 R2 会打上标签 19 后发给 R3，而 R2 给 R1 的标签是 20，所以 R1 应该打上标签 20 后发给 R2，那么在 traceroute 时，就看到了，给 12.1.1.2(R2)打上的标签是 20,然后发给 23.1.1.3（R3）时打的标签是 19,最后 R3 发给 34.1.1.4（R4）时是没有打标签就发出去了。

（10）到 IP 网络 R5 上去 traceroute 4.4.4.4:

r5#traceroute 4.4.4.4

Type escape sequence to abort.

Tracing the route to 4.4.4.4

1 15.1.1.1 136 msec 72 msec 84 msec

2 12.1.1.2 300 msec 368 msec 192 msec

3 23.1.1.3 252 msec 296 msec 148 msec

4 34.1.1.4 216 msec * 188 msec

r5#

说明:从结果中发现，IP 网络中发的数据在穿越 MPLS 区域时，标签的交换过程是看不见的，要想看到标签的交换过程，只有在 MPLS 网络中才能看见。

8.查看标签交换过程

说明:再以 6.6.6.6 这条路由条目为例，看一下它的标签如何。在边缘路由器 R4 上，会将 IP 网络的路由条目以隐式空标签发给邻居，但是默认只有自己直连的 IP 网络才会发隐式空标签，而 6.6.6.6 是在 R6 上的网段，不是 R4 的直连网段，所以对于 6.6.6.6，R4 不应该隐式空标签给邻居。

（1）查看 R4 上对 6.6.6.6 的标签情况：

```
r4#sh mpls ip binding
  3.3.3.3/32
      in label:     19
      out label:    imp-null   lsr: 3.3.3.3:0       inuse
  4.4.4.4/32
      in label:     imp-null
      out label:    19         lsr: 3.3.3.3:0
  6.6.6.6/32
      in label:     20
      out label:    20         lsr: 3.3.3.3:0
  12.1.1.0/24
      in label:     17
      out label:    16         lsr: 3.3.3.3:0       inuse
  15.1.1.0/24
      in label:     18
      out label:    18         lsr: 3.3.3.3:0       inuse
```

```
  23.1.1.0/24
      in label:     16
      out label:    imp-null   lsr: 3.3.3.3:0       inuse
  34.1.1.0/24
      in label:     imp-null
      out label:    imp-null   lsr: 3.3.3.3:0
  46.1.1.0/24
      in label:     imp-null
      out label:    17         lsr: 3.3.3.3:0
r4#
```

说明:从结果中看出，因为 4.4.4.4 是 R4 的直连接口，所以本地标签（in label）是空的，而 6.6.6.6 不是自己直连网络，所以不应该打隐式空标签，可以看到它的本地标签（in label）是 20。

（2）查看 R4CEF 里面的 6.6.6.6：

```
r4#sh ip cef 6.6.6.6
6.6.6.6/32, version 20, epoch 0, cached adjacency 46.1.1.6
0 packets, 0 bytes
  tag information set
    local tag: 20
  via 46.1.1.6, Serial1/0, 0 dependencies
    next hop 46.1.1.6, Serial1/0
    valid cached adjacency
```

tag rewrite with Se1/0, 46.1.1.6, tags imposed: {}

r4#

说明:也可以看出，6.6.6.6 的处理不同于 4.4.4.4，6.6.6.6 的本地标签确实真实存在，是 20.

（3）看到 R3 给 6.6.6.6 打的标签:

r3#sh ip cef 6.6.6.6

6.6.6.6/32, version 20, epoch 0, cached adjacency 34.1.1.4

0 packets, 0 bytes

tag information set

local tag: 20

fast tag rewrite with Se1/1, 34.1.1.4, tags imposed: {20}

via 34.1.1.4, Serial1/1, 0 dependencies

next hop 34.1.1.4, Serial1/1

valid cached adjacency

tag rewrite with Se1/1, 34.1.1.4, tags imposed: {20}

r3#

说明:R3 给 6.6.6.6 打本地标签是正常的,打了远程标签(出标签)是 20,也在情理之中,因为不需要弹出标签(移除标签)。

（4）再看 R3 的 LFIB 表：

r3#sh mpls forwarding-table

Local Outgoing Prefix Bytes tag Outgoing Next Hop

tag tag or VC or Tunnel Id switched interface

16 Pop tag 12.1.1.0/24 1672 Se1/0 23.1.1.2

17 Pop tag 46.1.1.0/24 0 Se1/1 34.1.1.4

18 18 15.1.1.0/24 540 Se1/0 23.1.1.2

19 Pop tag 4.4.4.4/32 1132 Se1/1 34.1.1.4

20 20 6.6.6.6/32 540 Se1/1 34.1.1.4

r3#

说明:R3 对 6.6.6.6 出标签是 20，说明处理也是正常的

（5）R2 上看 6.6.6.6 的处理：

r2#sh mpls forwarding-table

Local Outgoing Prefix Bytes tag Outgoing Next Hop

tag tag or VC or Tunnel Id switched interface

16 Pop tag 34.1.1.0/24 0 Se1/0 23.1.1.3

17 17 46.1.1.0/24 0 Se1/0 23.1.1.3

18 Pop tag 15.1.1.0/24 520 Se1/1 12.1.1.1

19 Pop tag 3.3.3.3/32 0 Se1/0 23.1.1.3

20 19 4.4.4.4/32 756 Se1/0 23.1.1.3

21 20 6.6.6.6/32 540 Se1/0 23.1.1.3

r2#

说明:R3 的本地标签 20，变成 R2 的出标签 20，正常，而 R2 给 6.6.6.6 打的本地标签 21，发给邻居将成为远程标签。

（6）再看 R1 对 6.6.6.6 的处理：

r1#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	16	34.1.1.0/24	0	Se1/1	12.1.1.2
17	Pop tag	23.1.1.0/24	0	Se1/1	12.1.1.2
18	17	46.1.1.0/24	0	Se1/1	12.1.1.2
19	19	3.3.3.3/32	0	Se1/1	12.1.1.2
20	20	4.4.4.4/32	0	Se1/1	12.1.1.2
21	21	6.6.6.6/32	0	Se1/1	12.1.1.2

r1#

说明:R1 将 R2 发来的标签 21 变成自己的出标签（远程标签）21，正常，自己的本地标签也正常。

（7）在 R1 上跟踪路由的标签交换过程：

r1#traceroute 6.6.6.6

Type escape sequence to abort.

Tracing the route to 6.6.6.6

1 12.1.1.2 [MPLS: Label 21 Exp 0] 356 msec 284 msec 204 msec

2 23.1.1.3 [MPLS: Label 20 Exp 0] 388 msec 196 msec 192 msec

3 34.1.1.4 [MPLS: Label 20 Exp 0] 152 msec 268 msec 260 msec

4 46.1.1.6 244 msec * 172 msec

r1#

说明:因为 R4 对于路由条目 6.6.6.6 没有采用隐式空标签，而给 R3 发了标签 20，而 R3 给 R2 也发了标签 20，R2 给 R1 发了标签 21，所以结果中显示数据包发由 R1 发给 R2（12.1.1.2）时打的标签正是 21，R2 发给 R3 时，打的标签是 20，重要的是 R3 在发给 R4 时，并没有移除标签，而是打了标签 20 发出去的。

（8）在 IP 网络 R5 上看 6.6.6.6 的标签交换情况：

r5#traceroute 6.6.6.6

Type escape sequence to abort.

Tracing the route to 6.6.6.6

1 15.1.1.1 180 msec 92 msec 60 msec

2 12.1.1.2 308 msec 256 msec 264 msec

3 23.1.1.3 368 msec 200 msec 200 msec

4 34.1.1.4 232 msec 252 msec 144 msec

5 46.1.1.6 248 msec * 408 msec

r5#

说明：IP 网络的 R5 没有看到标签交换情况，所以正常。

9.查看数据包交换数量

说明:可以在 LSR 上查看某条路由经过标签交换的数据包数量:

（1）在 R3 上查看 4.4.4.4 有多少个数据包经过了标签交换:

```
r3#sh mpls forwarding-table 4.4.4.4 detail
Local  Outgoing    Prefix          Bytes tag   Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched    interface
19     Pop tag     4.4.4.4/32      1744        Se1/1       34.1.1.4
       MAC/Encaps=4/4, MRU=1504, Tag Stack{}
       4CA18847
       No output feature configured
    Per-packet load-sharing
r3
```

（2）在 R3 上查看 6.6.6.6 有多少个数据包经过了标签交换:

```
r3#sh mpls forwarding-table 6.6.6.6 detail
Local  Outgoing    Prefix          Bytes tag   Outgoing    Next Hop
tag    tag or VC   or Tunnel Id      switched    interface
20     20          6.6.6.6/32      2028        Se1/1       34.1.1.4
       MAC/Encaps=4/8, MRU=1500, Tag Stack{20}
       4CA18847 00014000
       No output feature configured
    Per-packet load-sharing
```

10.路由条目的标签限制

说明:在某些时候，并不希望 LSR 对相应的条目打上标签，那么就可以在 LSR 限制相应路由条目的标签通告或接收。

（1） 限制标签接收：

说明:在 R1 上测试限制接收 6.6.6.6 的标签：

```
r1(config)#access-list 6 permit 6.6.6.6

r1(config)#mpls ldp neighbor 12.1.1.2 labels accept 6
```

说明:命令意思为只从邻居 12.1.1.2 那里接收 ACL 6 允许的路由条目的标签，其它的不接收。

（2）查看配置标签限制后的 LFIB：

```
r1#sh mpls forwarding-table

Local  Outgoing    Prefix          Bytes tag  Outgoing    Next Hop

tag    tag or VC   or Tunnel Id    switched   interface

16     Untagged    34.1.1.0/24     0          Se1/1       12.1.1.2

17     Untagged    23.1.1.0/24     0          Se1/1       12.1.1.2

18     Untagged    46.1.1.0/24     0          Se1/1       12.1.1.2

19     Untagged    3.3.3.3/32      0          Se1/1       12.1.1.2

20     Untagged    4.4.4.4/32      0          Se1/1       12.1.1.2

21     21          6.6.6.6/32      0          Se1/1       12.1.1.2

r1#
```

说明:可以看到，除了 6.6.6.6 打上了标签 21 以外，其它路由条目全部都没有标签，说明配置正确。

（3）限制标签发送：

说明:可以在 LSR 限制从邻居处接收标签，同样也可以限制将什么样的路由发送标签给邻居。

下面测试在 R3 上限制只发送 6.6.6.6 的标签给 R2，其它统统不发标签：

先看 R2 上在没有做限制时的标签情况：

r2#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	34.1.1.0/24	0	Se1/0	23.1.1.3
17	17	46.1.1.0/24	0	Se1/0	23.1.1.3
18	Pop tag	15.1.1.0/24	3340	Se1/1	12.1.1.1
19	Pop tag	3.3.3.3/32	0	Se1/0	23.1.1.3
20	19	4.4.4.4/32	972	Se1/0	23.1.1.3
21	20	6.6.6.6/32	1188	Se1/0	23.1.1.3

说明:可以看到 6.6.6.6 和其它路由都有标签。

（4）在 R3 上配置只发 6.6.6.6 的标签给 R2：

r3(config)#access-list 2 permit 2.2.2.2　　此地址必须为对方的 Router-ID

r3(config)#access-list 6 permit 6.6.6.6　　匹配路由条目

ldp advertise-labels 此命令必须配，用以禁止正常标签传送，否则所有限制无效

r3(config)#mpls ldp advertise-labels for 6 to 2 对相应邻居做相应限制

r3(config)#exi

（5）再看 R2 上 6.6.6.6 的标签情况：

r2#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	34.1.1.0/24	0	Se1/0	23.1.1.3
17	Untagged	46.1.1.0/24	0	Se1/0	23.1.1.3
18	Untagged	15.1.1.0/24	3340	Se1/1	12.1.1.1
19	Untagged	3.3.3.3/32	0	Se1/0	23.1.1.3
20	Untagged	4.4.4.4/32	972	Se1/0	23.1.1.3
21	20	6.6.6.6/32	1188	Se1/0	23.1.1.3

说明:可以看到，除了 6.6.6.6 有标签，其它都没有标签，说明配置生效。

（6）在 R3 上也可以看配置的效果：

r3#sh mpls ldp bindings advertisement-acls

Advertisement spec:

Prefix acl = 6; Peer acl = 2

tib entry: 2.2.2.2/32, rev 13

```
    tib entry: 3.3.3.3/32, rev 4
    tib entry: 4.4.4.4/32, rev 15
    tib entry: 6.6.6.6/32, rev 18
        Advert acl(s): Prefix acl 6; Peer acl 2
    tib entry: 12.1.1.0/24, rev 8
    tib entry: 15.1.1.0/24, rev 12
    tib entry: 23.1.1.0/24, rev 6
    tib entry: 34.1.1.0/24, rev 2
    tib entry: 46.1.1.0/24, rev 10
r3#
```

LDP 邻居认证

说明:邻居之间可以配置相应密码，如果密码不同，则邻居无法建立。

（1）配置 R2 对 R1 使用密码 cisco，如果 R1 无密码，则邻居失败:

```
r2(config)#mpls ldp neighbor 12.1.1.1 password 0 cisco
r2(config)#
```

说明:指定邻居时，后面应该为邻居的 Router-ID,0 表示在内存中显示时不加密。其它邻居失败的效果略过，请自行配置查看。

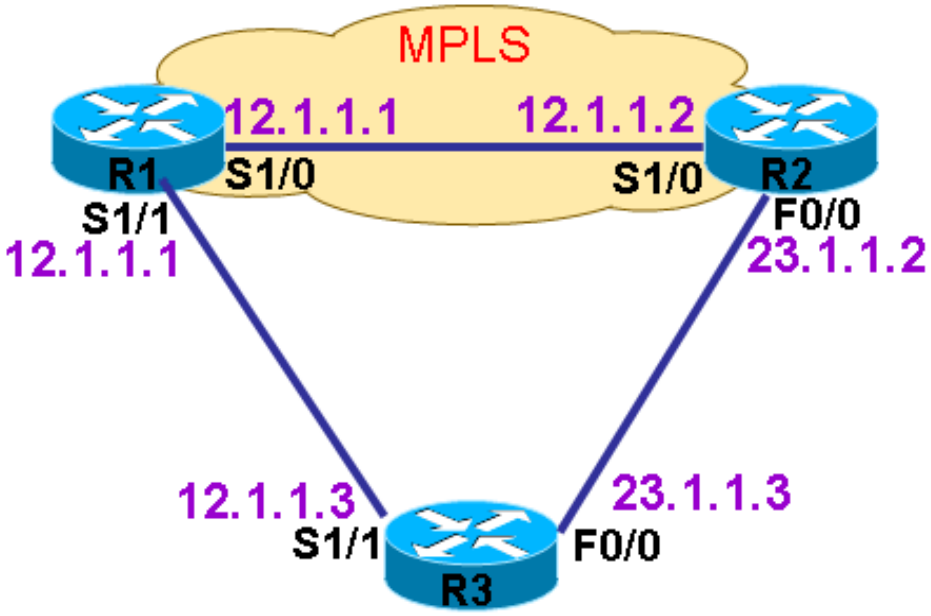
LDP 会话保护

在讲 LDP 时说过，LDP 在建邻居时，需要用到 hello 包，在直连接口上发送 hello 包出去，这个 hello 包是不能跨网段传递的，而这个 hello 包被称为 LDP Link Hello，如果对方有邻居回应了这个包，那么就建立 LDP 会话，称为 LDP sessions，LDP 会话建立后就可以传递标签，这是直连邻居。但是邻居也可以远程建立，也就是说不直连，那么这样的 hello 称为 LDP Targeted Hello，而远程建立的会话就叫 targeted session。一般情况下，两台直连 LSR 建立会话之后，如果链路断掉了，那么会话也就断掉了，所以所有标签等到会话再次建立后再次重新计算。但是当两台 LSR 之间链路断掉之后，如果他们之间还有备用链路的话，完全可以事先在备用链路上建个远程会话 targeted session，这样的话，即使两台 LSR 之间直连链路断了，也可以因为还有远程会话而不用清空所有标签，等到直连链路恢复后，再切换回来。这就是开启会话保护之后，通过建立远程会话来保护标签表的作用，但要做这样的保护，LSR 之间必须得有备用链路，否则无效。

配置保护时，指定在多少时间内，会话不要断开，流量继续发送，也可以为软重置提供保护。配置时应该双方路由器都配置会话保护，或者一方配了，另一方至少要能回应 Targeted Hello。

上面就是利用会话保护的功能来为邻居之间在备用链路上创建远程会话（targeted session），远程会话可以保护邻居断掉之间不会马上断开，因为存在备用链路。这样的远程会话可以通过会话保护的功能来创建，除此之外，还有一种创建方法，那就是手工创建远程会话，下面分别来介绍这两种方法的配置。

配置



说明:R1 的 loopback0 为 1.1.1.1/32，R2 的 loopback0 为 2.2.2.2/32，R3 的 loopback0 为 3.3.3.3/32，OSPF 在所有设备上开启，并且所有 loopback 均放入 OSPF 进程。LDP 只在 R1 和 R2 之间的 S1/0 开启。从图中可以发现，当 R1 和 R2 之间的直连链路断掉以后，LDP 会话也会断开的，但是 R1 和 R2 明明还可以通过 R3 建立远程会话连接。

1.配置会话保护

（1）在 R1 上开启会话保护：

R1(config)# mpls ldp session protection

（2）在 R2 上开启会话保护：

R2(config)# mpls ldp session protection

（3）也可以通过 ACL 指定邻居和时间，如：

router(config)#access 1 per 1.1.1.1

router(config)#mpls ldp session protection for 1 duration 90s

2.查看会话保护效果

（1）先看一下保护之前的邻居状态：

r1#sh mpls ld neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.23261 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 9/9; Downstream

Up time: 00:00:04

LDP discovery sources:

Serial1/0, Src IP addr: 12.1.1.2

Addresses bound to peer LDP Ident:

23.1.1.2 12.1.1.2 2.2.2.2

说明:可以看出没有任何远程会话的信息。

（2）查看开了会话保护的邻居状态：

r1#sh mpls ld neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.39052 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 11/11; Downstream

Up time: 00:02:27

LDP discovery sources:

```
        Serial1/0, Src IP addr: 12.1.1.2
        Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive
      Addresses bound to peer LDP Ident:
        23.1.1.2        12.1.1.2        2.2.2.2
r1#
```

说明:可以看出和 R2 之间存在远程会话信息。

（3） 再看 discovery 信息：

```
r1#sh mpls ld discovery detail
 Local LDP Identifier:
    1.1.1.1:0
    Discovery Sources:
    Interfaces:
      Serial1/0 (ldp): xmit/recv
          Enabled: Interface config
          Hello interval: 5000 ms; Transport IP addr: 1.1.1.1
          LDP Id: 2.2.2.2:0
            Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2
            Hold time: 15 sec; Proposed local/peer: 15/15 sec
            Reachable via 2.2.2.2/32
      Serial1/1 (ldp): xmit
          Enabled: Interface config
```

```
        Hello interval: 5000 ms; Transport IP addr: 1.1.1.1
    Targeted Hellos:
      1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv
        Hello interval: 10000 ms; Transport IP addr: 1.1.1.1
        LDP Id: 2.2.2.2:0
          Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2
          Hold time: 90 sec; Proposed local/peer: 90/90 sec
          Reachable via 2.2.2.2/32
r1#
```

说明:同样能看到远程会话信息。

（4） 断开 R1 和 R2 之间的直连链路测试:

```
r1#sh mpls ld discovery detail
 Local LDP Identifier:
    1.1.1.1:0
    Discovery Sources:
    Interfaces:
      Serial1/1 (ldp): xmit
        Enabled: Interface config
        Hello interval: 5000 ms; Transport IP addr: 1.1.1.1
    Targeted Hellos:
```

```
    1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv
      Hello interval: 10000 ms; Transport IP addr: 1.1.1.1
      LDP Id: 2.2.2.2:0
       Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2
       Hold time: 90 sec; Proposed local/peer: 90/90 sec
       Reachable via 2.2.2.2/32
r1#


r1#sh mpls ld neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
      TCP connection: 2.2.2.2.39052 - 1.1.1.1.646
      State: Oper; Msgs sent/rcvd: 15/13; Downstream
      Up time: 00:03:27
      LDP discovery sources:
       Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive
      Addresses bound to peer LDP Ident:
       23.1.1.2        12.1.1.2        2.2.2.2
r1#
```

说明:可以看见，R1 和 R2 之间的直连链路断开后，LDP 会话还在。

3.手工配置远程会话

说明:之前是通过会话保护的功能产生的远程会话，下面通过手工创建远程会话。

（1） R1 上创建远程会话：

说明:需要指定对方 Router-ID 地址

r1(config)#mpls ldp neighbor 2.2.2.2 targeted ldp　不指定 LDP，默认为 TDP

（2）R2 上创建远程会话：

R2(config)#mpls ldp neighbor 1.1.1.1 targeted ldp

（3）查看效果：

先看建立之前的状态：

```
r1#sh mpls ld neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 2.2.2.2.23261 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 9/9; Downstream
        Up time: 00:00:04
        LDP discovery sources:
          Serial1/0, Src IP addr: 12.1.1.2
        Addresses bound to peer LDP Ident:
          23.1.1.2        12.1.1.2        2.2.2.2
r1#
```

说明:可以看出没有任何远程会话的信息。

再看建立之后的：

```
r1#sh mpls ldp neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
      TCP connection: 2.2.2.2.23261 - 1.1.1.1.646
      State: Oper; Msgs sent/rcvd: 10/10; Downstream
      Up time: 00:01:23
      LDP discovery sources:
        Serial1/0, Src IP addr: 12.1.1.2
        Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive
      Addresses bound to peer LDP Ident:
        23.1.1.2        12.1.1.2        2.2.2.2
r1#


r1#sh mpls ld discovery detail
 Local LDP Identifier:
    1.1.1.1:0
    Discovery Sources:
    Interfaces:
      Serial1/0 (ldp): xmit/recv
        Enabled: Interface config
```

```
        Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

        LDP Id: 2.2.2.2:0

          Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2

          Hold time: 15 sec; Proposed local/peer: 15/15 sec

          Reachable via 2.2.2.2/32

      Serial1/1 (ldp): xmit

        Enabled: Interface config

        Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

    Targeted Hellos:

      1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv

        Hello interval: 10000 ms; Transport IP addr: 1.1.1.1

        LDP Id: 2.2.2.2:0

          Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2

          Hold time: 90 sec; Proposed local/peer: 90/90 sec

          Reachable via 2.2.2.2/32

r1#
```

说明:可以看出和 R2 之间存在远程会话信息。

（4）断开 R1 和 R2 之间的直连链路测试:

```
r1#sh mpls ld discovery detail

 Local LDP Identifier:
```

```
    1.1.1.1:0
    Discovery Sources:
    Interfaces:
      Serial1/1 (ldp): xmit
        Enabled: Interface config
        Hello interval: 5000 ms; Transport IP addr: 1.1.1.1
    Targeted Hellos:
      1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv
        Hello interval: 10000 ms; Transport IP addr: 1.1.1.1
        LDP Id: 2.2.2.2:0
          Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2
          Hold time: 90 sec; Proposed local/peer: 90/90 sec
          Reachable via 2.2.2.2/32
r1#


r1#sh mpls ldp neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
      TCP connection: 2.2.2.2.23261 - 1.1.1.1.646
      State: Oper; Msgs sent/rcvd: 15/13; Downstream
      Up time: 00:02:50
      LDP discovery sources:
```

```
    Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive

    Addresses bound to peer LDP Ident:

      23.1.1.2        12.1.1.2        2.2.2.2

r1#
```

说明:可以看见，R1 和 R2 之间的直连链路断开后，LDP 会话还在。

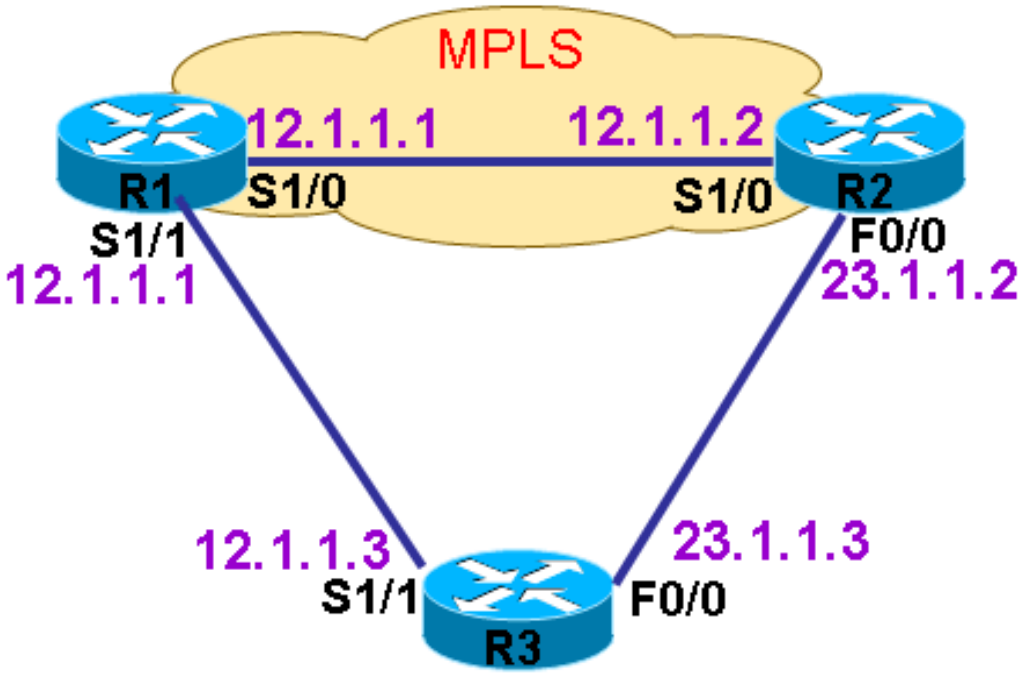
IGP 和 LDP 同步

概述

在某些情况下，当 LDP 邻居还没有建立或者邻居丢失而没有为路由发送标签时，如果这时 IGP 邻居已经建立并学到路由条目后，就会开始 IP 交换，那么后来当 LDP 正常以后，可能会出现丢包的情况，那么就利用 IGP 和 LDP 同步的特性，来限制 LSR，只有当 IGP 和 LDP 都认为某条链路该转发，才转发流量。

IGP 和 LDP 同步的特性只能在接口下配置过 mpls ip 时可以使用，并且目前只支持 OSPF 和 LDP 的同步。当配置好同步以后，相应的接口下，OSPF 邻居在 LDP 邻居还没有建立之前，自己是不会建立邻居关系的，但是可以配置一个最大等待时候，称为 holddown 时候，默认是没有配的，如果 OSPF 在这个 holddown 时间过后，LDP 邻居还没有建立起来，那么自己还是会建立邻居关系，但是，当 OSPF 在 LDP 没有建立邻居关系时，自己从邻居收到的路由条目将会被打上 Metric 65536，这个值是很大的。

配置



说明:OSPF 在 R1 和 R2 之间开启，还在 R1 和 R3 之间开启，而 LDP 只在 R1 和 R2 之间的接口开启。并且 R1 的 loopback0 为 1.1.1.1/32，R2 的 loopback0 为 2.2.2.2/32，R3 的 loopback0 为 3.3.3.3/32，所有 loopback 均放入 OSPF 进程。

1.配置 IGP 和 LDP 的同步

说明:在 R1 上配置 IGP 和 LDP 的同步

（1）在 OSPF 进程下开启 IGP 和 LDP 的同步：

注：同步只在接口下配置了 mpls ip 才会生效。同步在进程下开启之后，所以接口生效，也可以基于相应接口关闭。

```
R1(config)#router os 2

R1(config-router)#mpls ldp sync
```

2.查看配置

```
r1#sh ip ospf mpls ldp interface
Serial1/1
  Process ID 2, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Not required
  Holddown timer is disabled
  Interface is up
Serial1/0
  Process ID 2, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Required
  Holddown timer is configured : 10000 msecs
  Holddown timer is not running
  Interface is up and sending maximum metric
Loopback0
  Process ID 2, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Not required
  Holddown timer is disabled
  Interface is up
```

r1#

说明:从结果中看出，只要接口 S1/0 同步才生效，因为只有此接口配了 mpls ip，但是接口下并没有 Holddown。

r1(config)#mpls ldp igp sync holddown 10000

3.配置 Holddown

说明：配置了同步以后，OSPF 在 LDP 没有建立邻居之前，自己是不会建立邻居关系的，但是如果超过了 Holddown 所限制的时间，即使 LDP 邻居关系还没有建立，OSPF 还是会强行建立自己的邻居关系的。

4.查看同步的效果

说明:最终的效果为，在 R2 还没有配 LDP 之前，也就是 R1 的 LDP 邻居关系不能建立，在 Holddown 时间之前，OSPF 邻居关系也没有，但是过了这个时间，LDP 邻居还没有的话，OSPF 还是会建立邻居的，但是 R1 将 R2 发过来的路由条目的 Metric 值设置成 65536，然后发给邻居。

（1）在 R1 上查看 R2 发来的 OSPF 路由：

到 R3 上去看 R2 发给 R1 的路由，然后 R1 再发给自己后 Metric 是 65536，正常情况下不是

r1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

```
      ia - IS-IS inter area, * - candidate default, U - per-user static route

      o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


    1.0.0.0/32 is subnetted, 1 subnets

C       1.1.1.1 is directly connected, Loopback0

    2.0.0.0/32 is subnetted, 1 subnets

O       2.2.2.2 [110/65536] via 12.1.1.2, 00:00:55, Serial1/0

    3.0.0.0/32 is subnetted, 1 subnets

O       3.3.3.3 [110/65] via 13.1.1.3, 00:00:55, Serial1/1

    23.0.0.0/24 is subnetted, 1 subnets

O       23.1.1.0 [110/74] via 13.1.1.3, 00:00:55, Serial1/1

    12.0.0.0/24 is subnetted, 1 subnets

C       12.1.1.0 is directly connected, Serial1/0

    13.0.0.0/24 is subnetted, 1 subnets

C       13.1.1.0 is directly connected, Serial1/1

r1#
```

说明:可以看到，R1 上对于 R2 发来的路由条目 2.2.2.2，Metric 值是 65546。

（2）在 R3 上查看路由：

```
r3#sh ip route
```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

O 1.1.1.1 [110/65] via 13.1.1.1, 00:00:08, Serial1/1

2.0.0.0/32 is subnetted, 1 subnets

O 2.2.2.2 [110/65600] via 13.1.1.1, 00:00:08, Serial1/1

3.0.0.0/32 is subnetted, 1 subnets

C 3.3.3.3 is directly connected, Loopback0

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/128] via 13.1.1.1, 00:00:08, Serial1/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/1

r3#

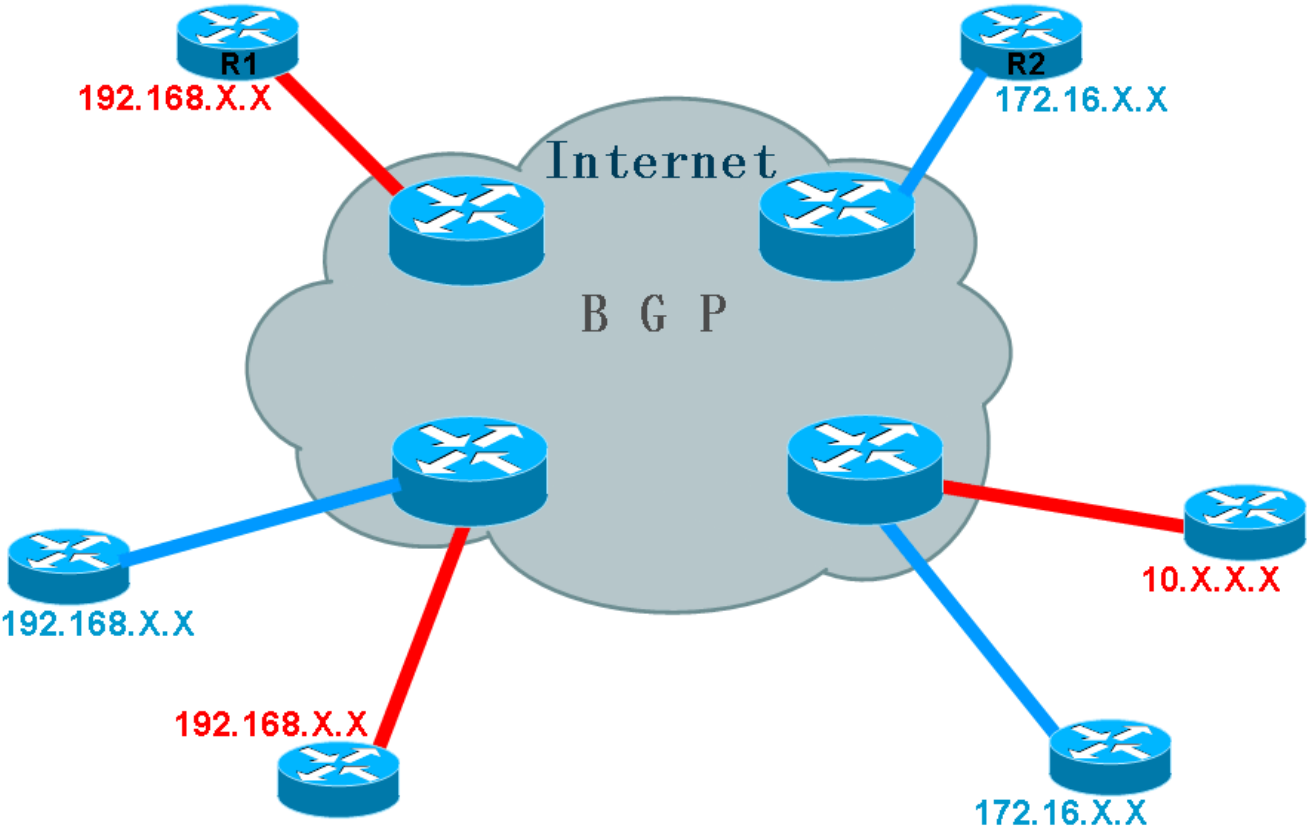
说明:在 R3 上看到 R1 发来的路由 2.2.2.2 的 Metric 是设置了 65536，再加上自己出口的 64，结果为 65600。

MPLS_VPN

概述

从前面可以看出，如果在英特网上大范围部署 MPLS 标签传输网络，这并没有给网络的的速度带来多少优势，而 MPLS 除了能够实现流量工程以外，还有一个很多人都认为很有优势的功能，那就是实现 VPN，具体 VPN 的效果和 VPN 的作用，详细过程可以参见本站“IPSec VPN”主题部分。在理解 MPLS_VPN 之前，应对 VPN 的功能有正确的认识。正因为用户的网络都是私有网络地址，所以这些用户之间的私有网络传输，并不能在 IP 公网上面很好的传递，如果一旦用户的私有网络在公网上传递时，将会有无数个相同的私有网络，这将给 IP 网络带来麻烦。而又因为核心网部署 MPLS 网络之后，这样的网络可以不检查数据包的 IP 地址而进行传输，所以能让用户的私有网络在公网上传输的目的得以实现。不要忘了 IPSec VPN 同样可以做到这样的效果，而 MPLS_VPN 还有一个特点就是可以轻松实现多用户之间的全互联网络，但是我个人认为，这并不能称为 MPLS_VPN 的优势，因为用户要实现 VPN 传输，也许想要解决的，不仅仅是用私网地址通信这么单纯的目的，而数据的加密和安全性，也是应该被重点关心的，而这些，MPLS_VPN 要做到是有难度的。不仅是这样，要完成两个远程用户网络之间的 MPLS_VPN，必须得保护这两个远程网络之间所经过的所有核心网都要支持 MPLS，而且核心网和用户网之间必须相互配合，不能出一点差错，否则也会给 MPLS_VPN 带来麻烦，也就是说，当任何一方对自己来说如果比较难以控制，那么要实施和排错 MPLS_VPN 将会存在相当的难度。而 IPSec VPN，可以将所有问题统统解决在用户手里，也就是说实施和排错 IPSec VPN 时，用户将掌握所有控制权利，将问题解决在自己手中。所以上述两种 VPN，用户可以根据自己的情况来作出选择，需要提醒的是，IPSec VPN 存在着多种 VPN 架构，需要清楚地理解其每种架构的实现方法和作用。

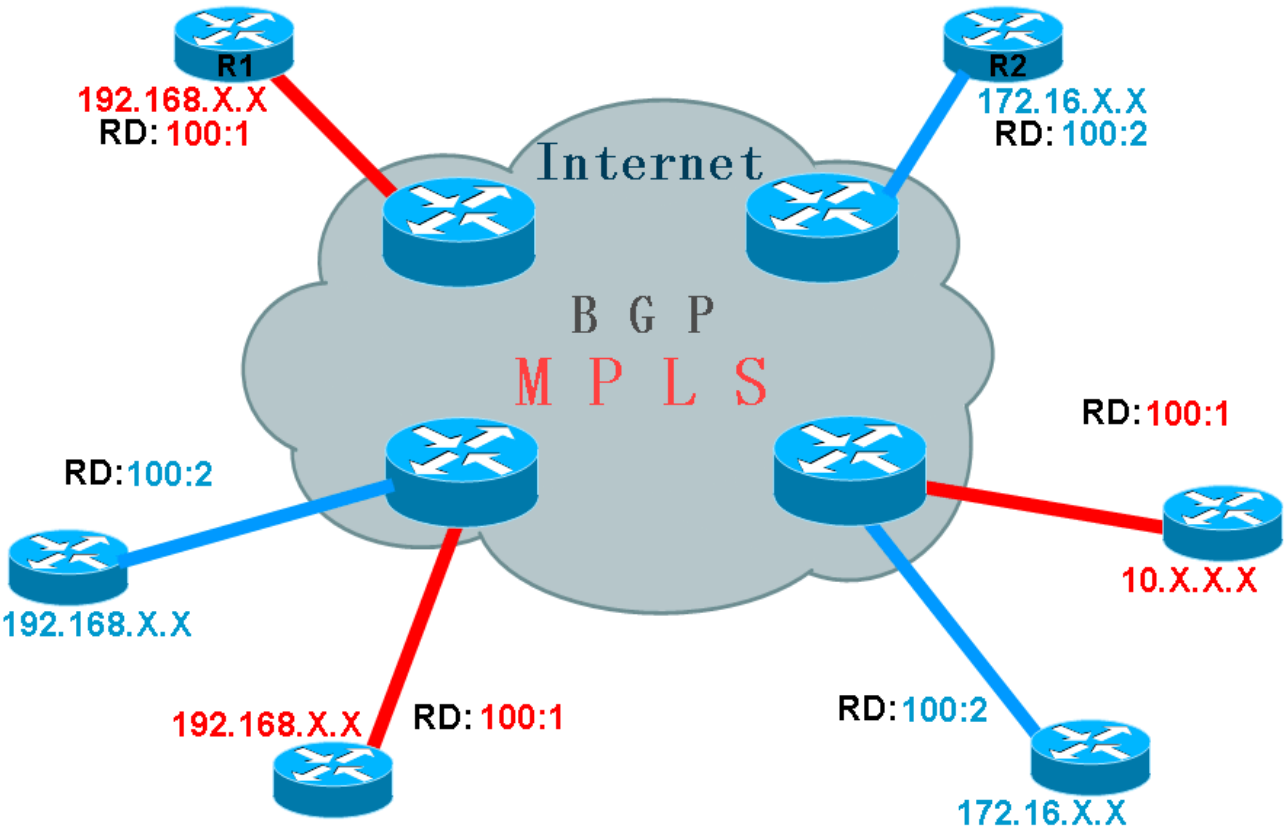
下面来详细讲解 MPLS_VPN 的实现方式。



如上图所示，在 Internet 的边缘，连接着多个用户网络，这些用户网络使用的 IP 网段都是私有网段，如：192.168.0.0，172.16.0.0，10.0.0.0，而这些网段是无法在 Internet 上进行传递的，因为如果用户的私有网段被放入 Internet 时，将意味着有无数个用户网段会发生地址冲突，而最终导致 Internet 路由器无法区分这些网段谁是谁了。既然这样，用户之间要直接通过对方的私有地址进行访问，正常情况下是不可能的。那么要帮助用户将这些私有网段在 Internet 上进行传递，除非使用一些技术能够让这些私有网络在 Internet 上进行传递时，是唯一的，是互不干扰的。这样的技术其实已经存在，那就是在 Internet 中运行 MPLS，因为在 MPLS 网络中，LSR 是不看 IP 地址进行数据转发的，所以用户的私有网络在进入 MPLS 网络时，无论他们是私有的还是重叠都的，对于 MPLS 来说，这些都不是问题，只要他们的标签是正确的，就能被 MPLS 正确转发。

RD（路由区分符）

根据以上所说，在 Internet 核心内部运行 MPLS 网络之后，用户的私有网络可以在 MPLS 里面自由传递，但是用户要想到达另外的用户网络，终究也是要出 MPLS 网络的，当用户的网络在 MPLS 中到达 Internet 边缘路由器时，如果这些路由器还是无法区分用户的私有网络，那么通信的问题依旧存在。如果要消除这个大问题，那么就必须让 Internet 边缘路由器能够正确拥有用户的私有 IP 网络，并且要正确的区分他们，只有这样，用户之间的私有网络通信才能成为可能。（请看下图）



如上图所示，要让 Internet 边缘路由器拥有用户的私有网络，那么唯一的方法就是在边缘路由器与边缘路由器之间运行 BGP，因为在 MPLS 中只要保证 BGP 之间的邻居地址可达就行，这个是可以轻易做到的，而当 BGP 在将用户的数据转入核心 MPLS 网络时，因为 MPLS 核心网络是只看标签而忽视用户私有网络的 IP 地址的，所以这不会存在任何问题，

虽然这样看来问题已经解决了，但是，还有一个最大的问题，那就是虽然 Internet

核心部分已经可以正确传递用户的私有网络了，Internet 边缘路由器也成功的拥有了用户的私有网络地址，可是，用户的网络，毕竟是私有网段，即使 BGP 拥有了用户的私有网段，那么当存在着多个用户的私有网段时，他们的地址势必会重叠和冲突，就如上图所示，有多个用户的网段都是 192.168.0.0 或者都是 172.16.0.0，这样的情况出现后，BGP 也是无法区别用户的网络谁是谁了。要让 BGP 正确区分哪条私有网段是属于哪个用户，就还得给用户这些网段加上额外的标记才行，只有这样，才能让 BGP 正确区别各个用户网段并且允许他们重叠。

既然要给用户的私有网段再加上额外的标记让 BGP 能够正确区分他们，就得考虑这些标记是否能被正确支持。当考虑使用额外标记时，对于 BGP 来说，这已经不算问题，因为 BGP 可以视任何标记为外部属性，既然自己不能理解，也能很好的传递出去。

由于上述原因，便在用户的私有网段进入 BGP 时，就被加上了额外的标记以区分它们，这样的标记被称为路由区分符（RD）。所有的用户私有网络在被 BGP 传递时，都加入了 RD，BGP 要支持这些 RD 的传递，就是多协议的 BGP（MP-BGP），所以 MP-BGP 在实现 MPLS_VPN 时，是必不可少的。

原来用户的网段地址长度都是 32 bit，而 RD 长度是 64 bit，当用户的地址被加上 RD 之后，就变成了 96 bit，这样的地址被称为 vpnv4。

RD 的格式

RD 分为两种格式：

ASN:nn（常用）和 IP-address:nn

ASN 代表 BGP AS 号码，nn 代表数字，数字可以随便定义，只要合理即可，但这个数字，对于一台路由器上的不同用户，肯定是不同的。

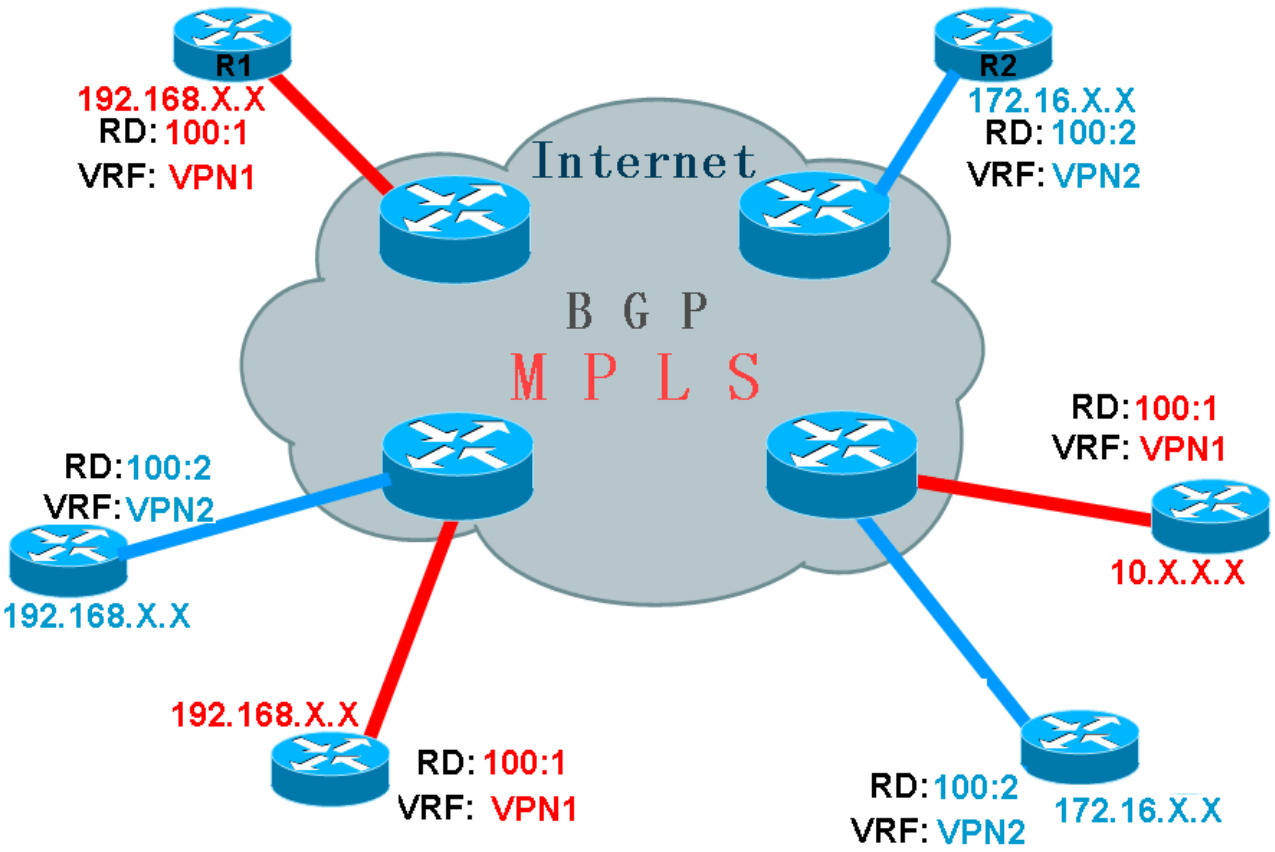
比如一个用户的网段是 10.1.1.0/24，RD 是 100:1，那么用户的 vpnv4 为 100:1:10.1.1.0/24

VRF（虚拟路由表）

到此为止，还并不能让用户网络之间直接通过内网地址访问，也就是还不能实

现 VPN 功能。

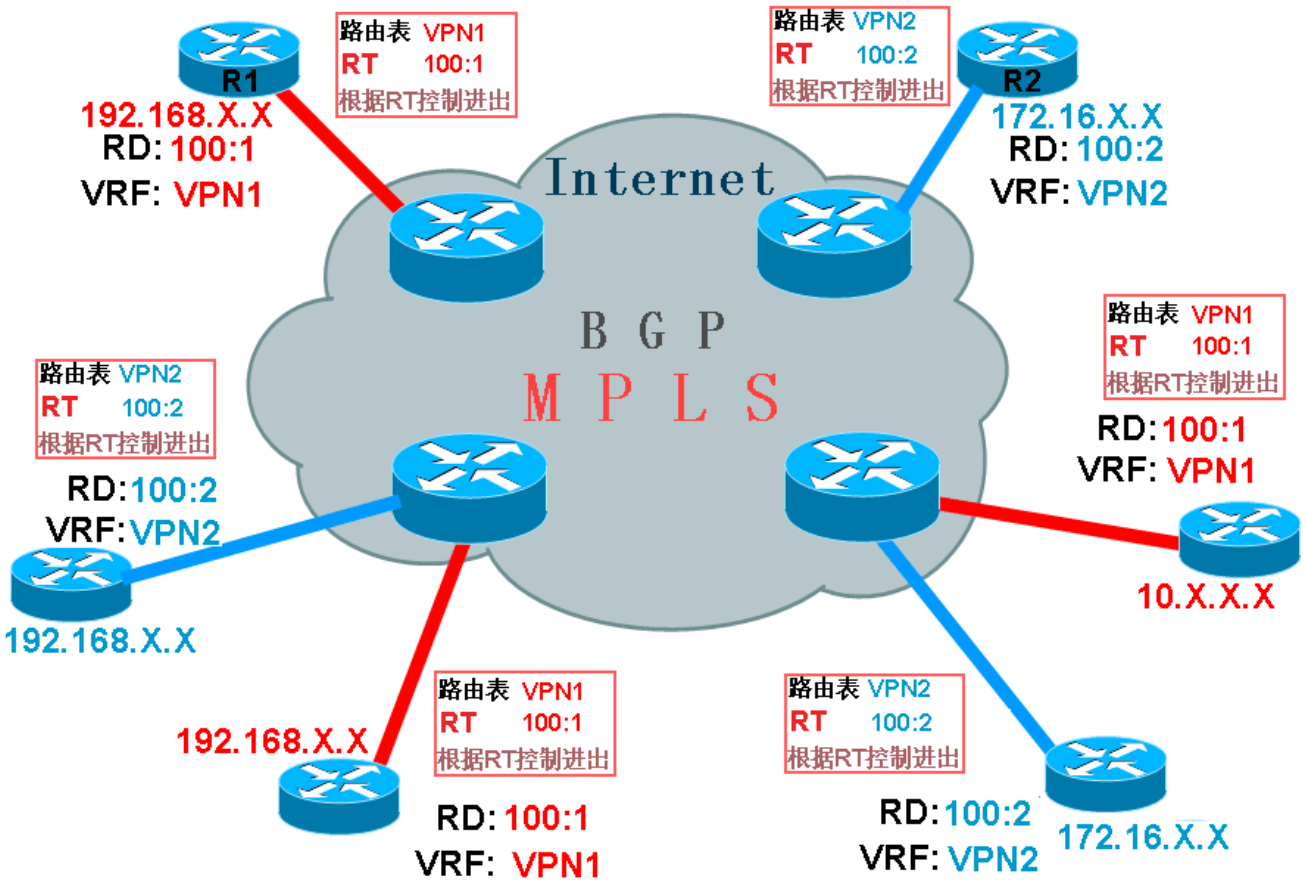
问题在于，当两个用户都连到同一台 Internet 边缘路由器时，比如他们的网段都是 192.168.0.0/24，虽然运行了 MP-BGP 的边缘路由器能够区分它们是两条不同的路由条目，如下图所示：



当左下角两个用户连到同一台边缘路由器时，因为两个用户的网段都是 192.168.0.0，既然 BGP 能够知道它们是两条网段，可是当远程有一个用户，比如左上角的用户（左上角红色网络 R1）要访问左下角红色网络时，它发出数据包的目的地址为 192.168.0.0，边缘路由器又如何能够正确将该数据包发给红色网络的 192.168.0.0，而不会错发给蓝色网络的 192.168.0.0 呢？这个问题值得思考。出现这样的问题，正因为边缘路由器中存在着两条 192.168.0.0 的网段，所以路由器无法区分数据包到底该发给谁，要解决这个问题，很明显，只要让路由器的路由表中只存在一条 192.168.0.0 的网段即可。那么又怎样才能让路由器中只存在一条不会重叠的私有网段呢，那就是在路由器上创建多个路由表，而这个路由表就只包含需要通信的用户网络，比如上图中只包含左上角红色网络和左下角红色网络，那么这样一来，当左上角红色网络 R1 要发数据包给左下角红色网络时，边缘路由器因为路由表中

只存在一条 192.168.0.0，所以就再也不会错把数据发给蓝色网络了。路由器上为需要通信的用户之间创建单独的路由表，这样的路由表被称为虚拟路由表（VRF），如果一台边缘路由器连接着多个不同用户，那么它将创建多个虚拟路由表，这个路由表和普通的路由表没有任何区别，只不过它只用来为 VPN 用户传递数据的，而与虚拟路由表相对的正常路由表被称为全局路由表。

RT（路由对象）

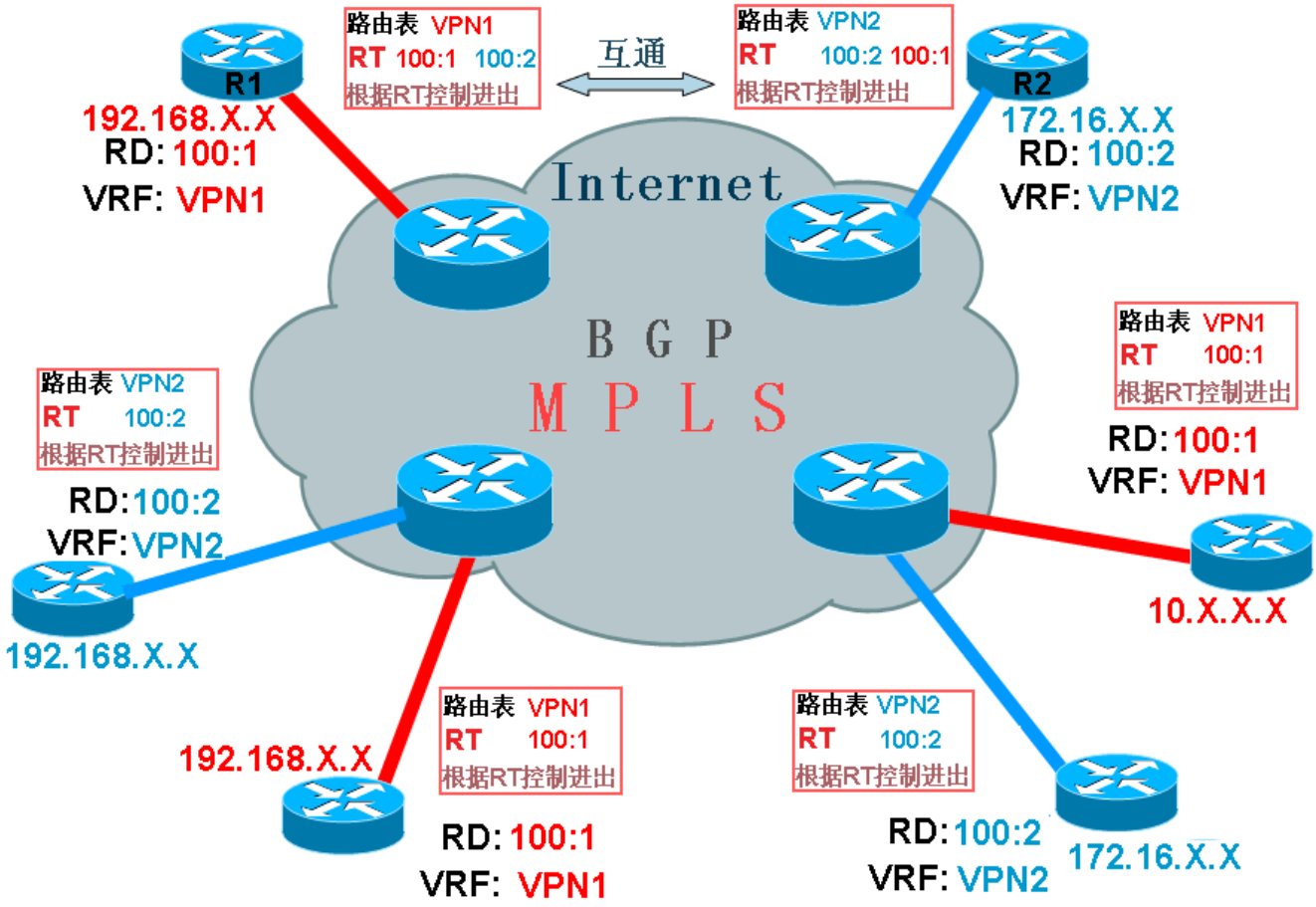


如上图所示，如果在刚才左下角的边缘路由器上为需要通信的红色网络创建出 VRF 之后，如 VRF 名字为 VPN1，那么又如何保证只有需要通信的红色网络的网段会进入路由表，而其它不相关的网络（如蓝色网络）不会错误地被放进路由表呢？这些，都必须有所控制，否则用户之间的网络还是混乱的。

可以回想一下，之前我们说过，用户的网络在进入 BGP 之间，都是会被标记 RD

的，既然我们需要让红色的网络之间可以通信，让蓝色的网络不能进入 VPN1，那么我们完全可以在给路由标记 RD 时，就将红色的网络标记为相同的 RD，比如 100：1，而将蓝色网络的 RD 标记为 100：2，这样一来，边缘路由器就可以正确匹配它们了。因为我们已经为红色的网络创建了 VRF 为 VPN1，而红色网络的 RD 都为 100：1，蓝色网络的 RD 都为 100：2，我们就可以完全控制只让 RD 为 100:1 的路由条目能够进入 VPN1，这样就能达到我们所有的目的，并且不会混淆不同用户的网络。这样控制路由表只能进入什么样的路由或者只能出去什么样的路由，被称为 RT（路由对象）。那么如果要让蓝色的网络之间可以相互通信，就可以为他们单独创建 VRF，如 VPN2，并且他们的 RD 是 100：2，最终就可以控制 RT 只让 RD 为 100：2 的路由条目能够进入 VPN2。通过这一切，就可以实现红色网络之间拥有单独的路由表而互通，让蓝色的网络也拥有自己单独的路由表而互通了。

上面所说的控制什么样的 RD 值能够进入什么样的 VRF 路由表，是 RT（路由对象）来控制的，比如一个 VRF 为 CCIE，用户的网段分别有 RD 为 100：1 的，也有 100：2 的和 100：3 的，我们只想让 RD 为 100：3 的路由条目能够进入 CCIE 的话，那么我们就可以设置 CCIE 的 RT 为 100：3。这样做之后，只有 RD 为 100：3 的路由条目才能进入 VRF CCIE，从而和其它 RD 为 100：1 和 100：2 的网络隔离开来。RT 是 BGP 扩展团体属性，分为输入和输出两种，可以简单理解为什么样的 RD 值可以进入该 VRF 或者该 VRF 什么样 RD 值的路由将被导出。如果路由的 RD 值和 RT 所允许的值不匹配，将不能进入或者出去的，这样不同 RD 的用户也就不能通信，想要通信，就配置为相同的 RD。但是一个 VRF 中，可以配置多个 RT，也就是说一个 VRF 可以让多个不同 RD 值的路由进入或出去，最终结果为，如果要让两个不同 RD 的用户要通信，就分别为两个用户各自的 VRF 同时配置两个用户的 RD，这样大家的 VRF 中都有互相的路由条目，也就可以互通了。如下图所示：



从上图中显示，红色网络中的 R1 和蓝色网络中的 R2，默认情况下，他们是不能通信的，因为红色网络的 VRF 只允许 RD 为 100：1 的路由条目进入，而蓝色网络的 VRF 只允许 RD 为 100：2 的路由条目进入，但是配置为大家的 VRF 都同时允许 RD 为 100：1 和 100：2 进入，最后两个 VRF 中都有了对方的路由，也就实现了互通。

一般情况下，要通信的用户之间，RD 都是配置为一样的，这样的通信称为内部通信，如果不同 RD 之间的网络也要通信，就可以分别在相互的 VRF 中配置 RT 允许双方的 RD 值都能进入和出去，也能够通信，这样的通信就称为外部通信。所以用户之间是否能通信，完全是由 RT 来控制的。RD 是让 BGP 来区分用户路由条目的，VRF 是让路由器来区分用户网络的。RT 是用来控制 VRF 的路由进入和出去的。

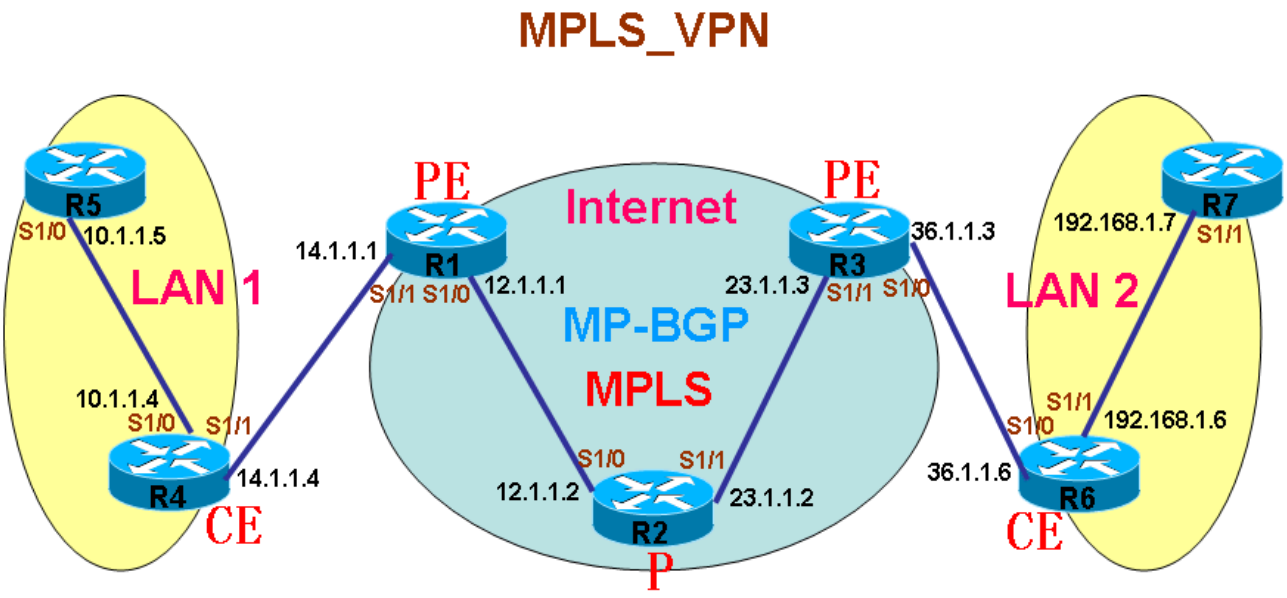
ISP 中存在 P 和 PE，其中 PE 直接和用户相连，P 则没有，但 P 和 PE 必须已经运行 MPLS。

用户中存在 CE 和 C，CE 是直接和 ISP 相连的路由器，但都不需要运行 MPLS。

CE 和 PE 直接交互，所以必须运行路由协议，或者是写静态路由，CE 只有一个对等 PE，但也可多宿主，也就是和多 PE 相连，

VPN，让 P 完全不用知道 VPN，就不用负担 VPN 的路由信息了，则使用 MPLS 来实现，为用户的私有网络打上标签，然后 P 也不用运行 BGP。但用户的路由必须在 PE 上出现，PE 是必须了解用户私有网络的。

MP-BGP



如上图所示，用户局域网 LAN1 连接 Internet，用户局域网 LAN2 也连接 Internet，当 LAN1 要和 LAN2 直接通过内网地址进行相互访问时，只要之间实现 MPLS_VPN 即可，在实施 MPLS_VPN 时，Internet 中之前是运行着 MPLS 的，我们已经知道，在 MPLS 中，中间的 LSR 称为 P，而 MPLS 中连接着用户网络的边缘 LSR 称为 PE，用户连接 PE 的路由器称为 CE，而用户自己内部的路由器可以称为是 C。

用户的网段到达 PE 时，因为要让 BGP 区分不同的用户网段，所以 PE 会为它加上 RD，加上了 RD 的网段称为 vpnv4，这时 PE 会将 vpnv4 的路由放入 MP-BGP，然后 MP-BGP 再将 vpnv4 从 MPLS 网络中通告给对端的 BGP 邻居，但是要保证 vpnv4

到达对端 BGP 邻居之后，对方还能够认识这这些 vpnv4 的 RD，那么 BGP 就必须为 vpnv4 打上相应的标签，对方 BGP 邻居看了这个标签，就能根据 RD 将其放入相应的 VRF，从而将数据包根据这些 VRF 转发给相应的 CE。正因为 BGP 要为 vpnv4 加入额外的标签，所以要使用 MP-BGP。

在 MPLS 中，数据包可以被打上多个标签，而 LSR 在转发时，只看顶部的标签，也就是说在数据包的多个标签中，只有顶部这一个标签会被使用和修改，所以 MP-BGP 在给 vpnv4 加入标签时，只要加在顶部标签的下面，这样数据包在 MPLS 中传输时，就不会被修改了，所以对端 BGP 邻居能够正确识别数据包的相应 RD，这也是为什么 MPLS_VPN 中数据包需要使用两个标签的理由。

MP-BGP 规则

普通的 BGP 通过额外配置 address-family 之后，就可以实现 MP-BGP 的功能，普通的 BGP 只是能够传递普通 IPv4 的路由，所以这样也会有个默认 address-family，称为 ipv4，但是因为 IP 分为单播和多播，所以这种默认的 address-family 就是 ipv4 unicast，如果要让 BGP 支持 ipv4 多播，那 address-family 就是 ipv4 multicast。

如果要支持 IPv6 的单播和多播，那么 address-family 就分别为 ipv6 unicast 和 ipv6 multicast。

在这里，以上的都不是我们要用到的 address-family，因为我们要传递的即不是 ipv4 单播，也不是 ipv4 多播，更不是 ipv6，我们要传递的是 vpnv4，所以就要开启 MP-BGP 支持 vpnv4，要支持 vpnv4，也需要创建相应的 vpnv4 的 address-family。并且需要创建相应的 VRF，这样相应的 vpn4 就和相应的 VRF 相关联起来。所有多协议的 BGP 在运行之前，应该保证普通的 BGP 邻居是正常的。MP-BGP 在为 vpnv4 通告标签时，并且所有信息要当作团体属性带出去，要手工指定。

注：IOS 缺省为每个 vpnv4 分配一个唯一的 MPLS 标签。

PE-CE 路由协议

不同用户网络之间要通过 MPLS_VPN 进行通信，就需要 PE 上有用户网络的路由信息，PE 再将这些路由在 MP-BGP 中通告给对端 MP-BGP 邻居。在这一切开始之前，PE 就应该获得用户的内部路由信息，而让 PE 获得这些路由信息的方法，可以使用动态路由或静态路由，如果要使用动态路由，那么 PE 和 CE 之间就必须启用某些路由协议，否则如果 PE 上没有用户的私有网络，那么远程用户之间也就不可能通信了。在 PE 有了用户的路由协议之后，必须将这些路由信息重分布进 MP-BGP，再由 MP-BGP 通告给对端邻居，从而到达远程用户，但是如果 CE 没有远程用户的私有网段信息，也是不能通信的，所以在 PE 上，MP-BGP 的路由也同样要发给 CE，也可以通过将 MP-BGP 的路由重分布进 PE 和 CE 间的路由协议，虽然普通 BGP 不允许将自己的路由重分布进 IGP，但 MP-BGP 不受此限制。

还需要说明的是，PE 和 CE 相连的某个接口，是属于某一个 VRF 的，所以从该接口进来的 CE 路由，都属于该 VRF。

PE 和 CE 之间支持的协议有

静态路由

RIPv2

OSPF

EIGRP

IS-IS

协议配置方法

在配置这些协议时，因为协议需要同时运行在 PE 和 CE 上，所以配置的步骤也分

为 PE 和 CE 两步。

1.静态路由

（1）PE 上直接对某个 VRF 写静态路由：

r4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.4

（2）在 PE 上将静态路由重分布进 MP-BGP:

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute static

r1(config-router-af)#exi

(3)CE 上要有默认路由指向 PE:

r4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.1

2.RIPv2

注：RIP 路由 metric 值超过 15，路由无效，而外部路由重分布进 RIP 时，默认不指定 metric 的情况下，值为无穷大，所以默认将不显示在 RIP 中，所以请指明 metric 或配置 default-metric

（1）PE 上配置 RIP:

r1(config)#router rip

r1(config-router)#version 2

r1(config-router)#no auto-summary

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#no auto-summary

r1(config-router-af)#network 14.0.0.0

r1(config-router-af)#redistribute bgp 100 metric 1

（2）将 RIP 重分布进 MP-BGP:

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute rip

（3）CE 上正常配置 RIP 即可:

r4(config)#router rip

r4(config-router)#version 2

r4(config-router)#no auto-summary

r4(config-router)#network 14.0.0.0

r4(config-router)#network 10.0.0.0

3.OSPF

说明:OSPF 还具有 Sham-lin（伪装链路的功能，此功能将不作任何讲述，如果 CCIE R&S v4.0 考试有该要求，本篇将立即加入详细讲解和配置过程!）

（1）在 PE 上配置 OSPF:

r3(config)#router ospf 100 vrf vpn1

r3(config-router)#router-id 36.1.1.3

r3(config-router)#network 36.1.1.3 0.0.0.0 area 0

r3(config-router)#redistribute bgp 100 subnets

（2）将 OSPP 重分布进 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#redistribute ospf 100

（3）CE 正常配置 OSPF:

r6(config)#router ospf 100

r6(config-router)#router-id 6.6.6.6

r6(config-router)#network 36.1.1.6 0.0.0.0 area 0

r6(config-router)#network 192.168.1.6 0.0.0.0 area 0

r6(config-router)#exit

r6(config)#

4.EIGRP

（1）在 PE 上配置 EIGRP:

r3(config)#router eigrp 1

r3(config-router)#no auto-summary

r3(config-router)#address-family ipv4 vrf vpn2

r3(config-router-af)#no auto-summary

r3(config-router-af)#network 83.1.1.3 0.0.0.0

r3(config-router-af)#autonomous-system 1

r3(config-router-af)#redistribute bgp 100 metric 10000 100 255 1 1500

（2）将 EIGRP 重分布进 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn2

r3(config-router-af)#redistribute eigrp 1

（3）CE 正常配置 EIGRP:

r8(config)#router eigrp 1

r8(config-router)#no auto-summary

r8(config-router)#network 172.16.1.8 0.0.0.0

r8(config-router)#network 83.1.1.8 0.0.0.0

IS-IS 暂且不作说明

5.EBGP

IOS 只支持 CE 和 PE 之间运行 EBGP

（1）PE 上配置 EBGP:

r1(config)#router bg 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#neighbor 14.1.1.4 remote-as 200

r1(config-router-af)#neighbor 14.1.1.4 activate

（2）和 CE 和 EBGP 默认会重分布进 MP-BGP:

（3）CE 上配置 EBGP:

r4(config)#router bgp 200

r4(config-router)#neighbor 14.1.1.1 remote

r4(config-router)#neighbor 14.1.1.1 remote-as 100

r4(config-router)#network 10.1.1.0 mask 255.255.255.0

r4(config-router)#

如果所有客户网络 AS 都是一样的，CE 会拒绝，因为看到与自己相同的 AS，但服务提供商可以比较，如果是一样的，就修改成自己的发过去，

服务提供商配置为：

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

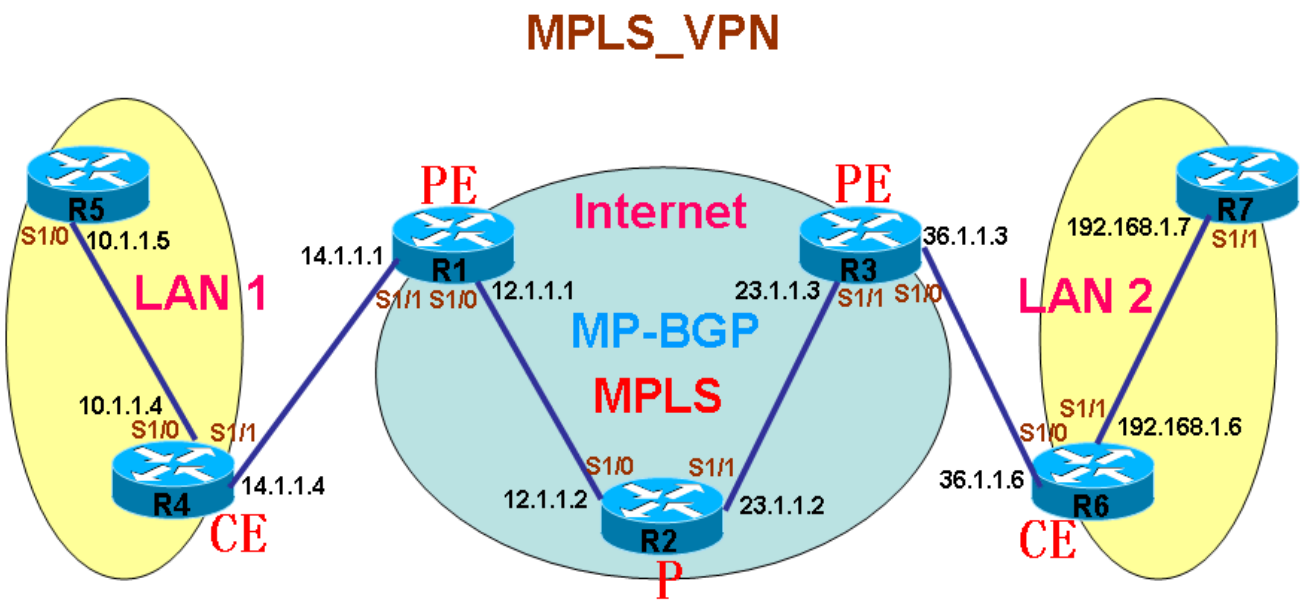
r1(config-router-af)#neighbor 14.1.1.4 as-override

PE 给 CE 再回来，hub-spoke，就配：nei allowas-in 1-10

配置 MPLS_VPN

概述

以下面一张图为拓朴，详细介绍配置 MPLS_VPN，其中 R1、R2、R3 上均已配置 loopback0，地址分别为 1.1.1.1/32，2.2.2.2/32，3.3.3.3/32，并且已经配置好 OSPF，让 MPLS 区域内所有直连接口和 loopback 口互通。



1.配置 MPLS

说明:因为 MP-BGP 是配置 MPLS_VPN 的必须协议，在开始配置 MPLS_VPN 之前，应该先将 MPLS 区域的相关接口实现标签交换，如各自的直连口，loopback 口均可看见已经通过标签进行交换。

（1） 配置 MPLS 区域略过，详细步骤请参见之前 MPLS 配置。

（2） 配置 MPLS_VPN 时，需要在边缘路由器 R1 和 R3 之间使用 MP-BGP 协议，我们使用路由器的 loopback 口作为源地址来建立邻居，首先测试双方 loopback 口已经实现标签交换。

r1#traceroute 3.3.3.3

Type escape sequence to abort.

Tracing the route to 3.3.3.3

1 12.1.1.2 [MPLS: Label 17 Exp 0] 104 msec 120 msec 140 msec

2 23.1.1.3 72 msec * 112 msec

r1#

说明:从结果可以看出，R1 到 R3 的 loopback0 已经实现标签交换。

r3#traceroute 1.1.1.1

Type escape sequence to abort.

Tracing the route to 1.1.1.1

1 23.1.1.2 [MPLS: Label 16 Exp 0] 100 msec 140 msec 196 msec

2 12.1.1.1 160 msec * 224 msec

r3#

说明:从结果可以看出，R3 到 R1 的 loopback0 已经实现标签交换。

2.配置普通 BGP

说明:在 R1 和 R3 之间配置普通 BGP，因为在配置 MP-BGP 之前，需要保证正常的 BGP 邻居是正常连通的

（1）在 R1 上配置普通 BGP：

```
r1(config)#router bgp 100
r1(config-router)#neighbor 3.3.3.3 remote-as 100
r1(config-router)#neighbor 3.3.3.3 update-source loopback 0
```

（2）在 R2 上配置普通 BGP：

```
r3(config)#router bgp 100
r3(config-router)#neighbor 1.1.1.1 remote-as 100
r3(config-router)#neighbor 1.1.1.1 update-source loopback 0
```

（3）在 R1 上确认与 R3 的普通 BGP 邻居关系已建立：

```
r1#sh ip bgp summary
BGP router identifier 1.1.1.1, local AS number 100
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3         4   100       3       3        1    0    0 00:00:51        0
r1#
```

说明:可以看到 R1 与 R3 已建立正常 BGP 邻居关系。

（4）在 R3 上确认与 R1 的普通 BGP 邻居关系已建立：

```
r3#sh ip bgp summary

BGP router identifier 3.3.3.3, local AS number 100

BGP table version is 1, main routing table version 1


Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd

1.1.1.1         4   100       4       4        1    0    0 00:01:07        0

r3#
```

说明:可以看到 R3 与 R1 已建立正常 BGP 邻居关系。

3.在 PE 上创建 VRF

说明:在 PE 上为用户创建相应的 VRF，并且指定 RD 值，需要通信的两个用户网络之间，VRF 和 RD 值保持一致。

（1）在 R1 上创建 VRF，并指定 RD 值：

```
r1(config)#ip vrf vpn1

r1(config-vrf)#rd 100:1
```

（2）在 R3 上创建 VRF，并指定 RD 值：

```
r3(config)#ip vrf vpn1

r3(config-vrf)#rd 100:1
```

4.在 PE 上将连 CE 的接口划入 VRF

说明:在 PE 上将相应的 CE 接口划入相应的 VRF，以后从该接口进入的用户数据包，则属于相应的 VRF，该用户的数据只能根据该 VRF 路由表作出转发决策。

（1）在 R1 上将连 CE R4 的接口 s1/1 划入 VRF:

```
r1(config)#int s1/1

r1(config-if)#ip vrf forwarding vpn1

% Interface Serial1/1 IP address 14.1.1.1 removed due to enabling VRF vpn1

r1(config-if)# ip add 14.1.1.1 255.255.255.0
```

说明:当一个正常的接口被划入 VRF 之后，接口上的地址会消失，所以需要重新配置一次该接口的 IP 地址。

(2) 在 R3 上将连 CE R6 的接口 s1/0 划入 VRF:

```
r3(config)#int s1/0

r3(config-if)#ip vrf forwarding vpn1

% Interface Serial1/0 IP address 36.1.1.3 removed due to enabling VRF vpn1

r3(config-if)#ip add 36.1.1.3 255.255.255.0
```

5.在 PE 上查看 VRF 的路由表情况

说明:从用户发到 PE 的数据包，PE 只能根据该用户的 VRF 路由表作出转发决策，也就是说，如果两个要通信的用户网络，如果各自的内网路由没有出现在 PE 的 VRF 路由表里，那么他们将不能通信，

（1）在 R1 上查看 VRF vpn1 的路由表:

```
r1#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

       E1 - OSPF external type 1, E2 - OSPF external type 2

       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

       ia - IS-IS inter area, * - candidate default, U - per-user static route

       o - ODR, P - periodic downloaded static route



Gateway of last resort is not set



     14.0.0.0/24 is subnetted, 1 subnets

C       14.1.1.0 is directly connected, Serial1/1

r1#
```

说明:可以看见，PE 在将连 CE R4 的接口 s1/1 划入 VRF vpn1 之后，该接口就进入 VRF vpn1 的路由表。并且要说明的是，该接口就不会再出现在全局路由表里。

（2） 查看 PE R1 的全局路由表，已经不会再有连 CE 接口 s1/1 的路由了：

```
r1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
        E1 - OSPF external type 1, E2 - OSPF external type 2

        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

        ia - IS-IS inter area, * - candidate default, U - per-user static route

        o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     1.0.0.0/32 is subnetted, 1 subnets

C       1.1.1.1 is directly connected, Loopback0

     2.0.0.0/32 is subnetted, 1 subnets

O       2.2.2.2 [110/65] via 12.1.1.2, 00:14:25, Serial1/0

     3.0.0.0/32 is subnetted, 1 subnets

O       3.3.3.3 [110/129] via 12.1.1.2, 00:14:25, Serial1/0

     23.0.0.0/24 is subnetted, 1 subnets

O       23.1.1.0 [110/128] via 12.1.1.2, 00:14:25, Serial1/0

     12.0.0.0/24 is subnetted, 1 subnets

C       12.1.1.0 is directly connected, Serial1/0

r1#
```

6.创建 MP-BGP

说明:通过上面在 PE 上查看 VRF 路由表发现，VRF 路由表中并没有双方用户的路由，所以，必须创建 MP-BGP，来为双方用户网络传递路由信息。

（6 **(1)在 PE R1 上创建 MP-BGP：**

r1(config)#router bgp 100

r1(config-router)#address-family vpnv4

r1(config-router-af)#neighbor 3.3.3.3 activate

r1(config-router-af)#neighbor 3.3.3.3 send-community both

说明:因为要传递 vpnv4 的路由，所以创建的 address-family 为 vpnv4，并且将正常的 BGP 邻居在 vpnv4 里面激活，而且还需要将这些 BGP 的扩展属性手工强行发给对端，否则对方收到的路由信息不会携带扩展属性，也就无法正常区分用户的路由信息。

（2） 在 PE R3 上创建 MP-BGP：

r3(config)#router bgp 100

r3(config-router)#address-family vpnv4

r3(config-router-af)#neighbor 1.1.1.1 activate

r3(config-router-af)#neighbor 1.1.1.1 send-community both

（3） 查看 MP-BGP 邻居：

r1#sh ip bgp all summary

For address family: IPv4 Unicast

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
3.3.3.3	4	100	23	23	1	0	0	00:03:18	0

For address family: VPNv4 Unicast

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
3.3.3.3	4	100	23	23	1	0	0	00:03:18	0

r1#

说明:可以看到，R1 上已经和 R3 建立普通 BGP 邻居关系，同时也建立 MP-BGP 邻居关系。

7.查看 MP-BGP 的 VRF 路由

r1#sh ip bgp vpnv4 all

r1#

说明:在 PE 上只是已经创建了 MP-BGP，并没有为 BGP 创建 VRF，所以无法看到 BGP 中的 VRF 路由表信息，所以还需要手工创建 VRF 路由表。

8.为 MP-BGP 创建 VRF

说明:MP-BGP 在收到用户的路由信息后，必须将其放入相应的 VRF 路由表，但是这个 VRF 表是要手工创建的，并且和该用户相关联的 VRF 名字保持一致。

（1）在 R1 上为 MP-BGP 创建 VRF vpn1:

```
r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1
```

（2）在 R3 上为 MP-BGP 创建 VRF vpn1:

```
r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1
```

（3）查看已创建的 VRF 路由表的路由条目:

```
r1#sh ip bgp vpnv4 all

r1#

r1#sh ip bgp vpnv4 vrf vpn1

r1#
```

说明:可以看到，BGP VRF 路由表中并没有用户的路由信息。

9.配置 RT 控制 VRF 路由信息

说明:因为 MP-BGP 的 VRF 路由表能让什么样的路由进入，是靠 RT 来控制的，所以要想让用户的路由被 MP-BGP 传递，就必须为 VRF 配置相应的 RT，只有 RT 允许的 RD 路由，才能进入和出去 VRF 表。

（1）在 R1 上为 VRF 配置相应的 RT：

r1(config)#ip vrf vpn1

r1(config-vrf)#route-target both 100:1

说明：VRF vpn1 允许 RD 为 100:1 的路由进入和出去。

（2）在 R3 上为 VRF 配置相应的 RT：

r3(config)#ip vrf vpn1

r3(config-vrf)#route-target both 100:1

10.配置 PE-CE 的路由协议

说明:虽然 MP-BGP 的 VRF 已经允许相应的用户路由进入，但是在 PE 上，此时并不能获知用户的路由信息，所以 MP-BGP 的 VRF 路由表中，依然为空，要想让 MP-BGP 的 VRF 路由表能够导入相应的用户路由，那就必须和用户 CE 之前启用路由协议，以获得对方的路由信息，从而导入 MP-BGP 的 VRF 表。

（1）在 PE R1 上配置 RIP:

说明:PE-CE 路由协议 RIP 只支持版本 2

r1(config)#router rip

r1(config-router)#version 2

r1(config-router)#no auto-summary

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#no auto-summary

r1(config-router-af)#network 14.0.0.0

r1(config-router-af)#redistribute bgp 100 metric 1

注：发布路由都是在 address-family 中进行的，并且请关闭自动汇总功能，且将 MP-BGP 的路由重分布进 RIP，否则对方 CE 将无法得知远程用户的路由信息。

（2）在 CE R4 上配置 RIP:

说明:在 CE 上的路由协议和正常路由配置没任何区别，配置完协议后，将可以从 PE 上收到路由信息。

```
r4(config)#router rip
r4(config-router)#version 2
r4(config-router)#no auto-summary
r4(config-router)#network 14.0.0.0
r4(config-router)#network 10.0.0.0
```

（3）在 PE R3 上配置 OSPF:

说明:同样也要将 MP-BGP 的路由重分布进 OSPF，以便传递给 CE 端。

```
r3(config)#router ospf 100 vrf vpn1
r3(config-router)#router-id 36.1.1.3
r3(config-router)#network 36.1.1.3 0.0.0.0 area 0
r3(config-router)#redistribute bgp 100 subnets
```

（4） 在 CE R6 上配置 OSPP:

说明:CE 上的路由协议 OSPF 与正常 OSPF 配置无区别。

```
r6(config)#router ospf 100
r6(config-router)#router-id 6.6.6.6
```

r6(config-router)#network 36.1.1.6 0.0.0.0 area 0

r6(config-router)#network 192.168.1.6 0.0.0.0 area 0

r6(config-router)#exit

r6(config)#

11.在 PE 上查看 VRF 路由

说明:已经在 PE 和 CE 配置路由协议，所以 PE 上应该已经获得用户的内部路由信息。

(1) 在 R1 上查看是否得到 CE R4 的内部路由信息 10.1.1.0/24:

r1#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

R 10.1.1.0 [120/1] via 14.1.1.4, 00:00:27, Serial1/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, Serial1/1

r1#

说明:PE R1 已经成功通过路由协议 RIP 获得用户的内部路由信息。

(2) 查看 MP-BGP 的 VRF 路由表中是否已将用户的内部路由信息导入：

r1#sh ip bgp vpnv4 all

r1#

说明:MP-BGP 的 VRF 路由表还没有得到用户的内部路由信息，因为 PE 和 CE 之间运行的是 IGP 协议，所以要想让 IGP 学到的路由进入 MP-BGP 的 VRF 路由表，必须手工重分布。

12.将路由重分布进 MP-BGP

说明:PE-CE 之间在运行 IGP 时，无法自动导入 MP-BGP，所以手工重分布。

（1）在 R1 上将 RIP 路由导入 MP-BGP:

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute rip

（2）在 R3 上将 OSPP 路由导入 MP-BGP:

```
r3(config)#router bgp 100
r3(config-router)#address-family ipv4 vrf vpn1
r3(config-router-af)#redistribute ospf 100
```

13.查看 MP-BGP 路由

说明:MP-BGP 已经和 IGP 之间实现重分布，查看双方路由是否获得。

（1）查看 R1 上 MP-BGP 的 VRF 路由表:

```
r1#sh ip bgp vpnv4 all
BGP table version is 9, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf vpn1)
*> 10.1.1.0/24      14.1.1.4                 1         32768 ?
*> 14.1.1.0/24      0.0.0.0                  0         32768 ?
*>i36.1.1.0/24      3.3.3.3                  0    100      0 ?
*>i192.168.1.6/32   3.3.3.3                  65   100      0 ?
```

```
r1#
```

说明:已经拥有双方用户网络的内部路由信息。

（2）查看 R3 上 MP-BGP 的 VRF 路由表:

```
r3#sh ip bgp vpnv4 all
BGP table version is 9, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf vpn1)
*>i10.1.1.0/24     1.1.1.1                  1    100      0 ?
*>i14.1.1.0/24     1.1.1.1                  0    100      0 ?
*> 36.1.1.
0/24       0.0.0.0                  0         32768 ?
*> 192.168.1.6/32  36.1.1.6              65        32768 ?
r3#
```

14.查看 VRF 路由

说明:MP-BGP 中已经拥有双方用户的网络信息，再看 VRF 中是否全部拥有。

（1）在 R1 上查看 VRF 表：

```
r1#show ip route vrf vpn1

Routing Table: vpn1
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     36.0.0.0/24 is subnetted, 1 subnets
B       36.1.1.0 [200/0] via 3.3.3.3, 00:02:50
     10.0.0.0/24 is subnetted, 1 subnets
R       10.1.1.0 [120/1] via 14.1.1.4, 00:00:04, Serial1/1
     192.168.1.0/32 is subnetted, 1 subnets
B       192.168.1.6 [200/65] via 3.3.3.3, 00:02:50
     14.0.0.0/24 is subnetted, 1 subnets
```

C 14.1.1.0 is directly connected, Serial1/1

r1#

说明:PE R1 上的 VRF 已经拥有双方用户的网络信息，再看 VRF 中是否全部拥有。

（2）在 R3 上查看 VRF 表：

r3#show ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

36.0.0.0/24 is subnetted, 1 subnets

C 36.1.1.0 is directly connected, Serial1/0

10.0.0.0/24 is subnetted, 1 subnets

B 10.1.1.0 [200/1] via 1.1.1.1, 00:04:57

192.168.1.0/32 is subnetted, 1 subnets

O 192.168.1.6 [110/65] via 36.1.1.6, 00:04:01, Serial1/0

14.0.0.0/24 is subnetted, 1 subnets

B 14.1.1.0 [200/0] via 1.1.1.1, 00:04:57

r3#

说明:PE R1 上的 VRF 已经拥有双方用户的网络信息，再看 VRF 中是否全部拥有。

15.查看 CE 路由

说明:因为 PE 上已经拥有双方用户的路由信息，并且 PE 和 CE 之间也运行路由协议，所以这些路由应该出现在 CE 的路由表中，从而双方用户可以实现通信。

（1）查看 CE R4 上的路由表：

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

36.0.0.0/24 is subnetted, 1 subnets

R 36.1.1.0 [120/1] via 14.1.1.1, 00:00:26, Serial1/1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Serial1/0

192.168.1.0/32 is subnetted, 1 subnets

R 192.168.1.6 [120/1] via 14.1.1.1, 00:00:26, Serial1/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, Serial1/1

r4#

r4#

说明:可以看到，CE R4 上已经拥有所有用户的内部路由。

（2）查看 CE R6 上的路由表：

r6#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

```
      ia - IS-IS inter area, * - candidate default, U - per-user static route

      o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


    36.0.0.0/24 is subnetted, 1 subnets

C       36.1.1.0 is directly connected, Serial1/0

    6.0.0.0/32 is subnetted, 1 subnets

C       6.6.6.6 is directly connected, Loopback0

    10.0.0.0/24 is subnetted, 1 subnets

O E2    10.1.1.0 [110/1] via 36.1.1.3, 00:00:17, Serial1/0

C    192.168.1.0/24 is directly connected, Loopback192

    14.0.0.0/24 is subnetted, 1 subnets

O E2    14.1.1.0 [110/1] via 36.1.1.3, 00:00:17, Serial1/0

r6#
```

说明:可以看到，CE R6 上已经拥有所有用户的内部路由。

16.测试用户之间通信

说明:因为用户 CE 之间已经拥有双方的内部路由信息，所以应该能够通信

（1）R5 应该具有默认路由将所有数据从 CE R4 上发出去：

```
r5#sh ip route
```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.4 to network 0.0.0.0

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Serial1/0

S* 0.0.0.0/0 [1/0] via 10.1.1.4

r5#

说明:R5 已经拥有指向 CE R4 的默认路由。

（2）LAN 1 的 R5 上测试到 LAN 2 的网段 192.168.1.6 的连通性：

r5#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 188/242/288 ms

r5#

说明:R5 上已经成功穿越 MPLS 网络将数据包发送给远程用户网络，说明 MPLS_VPN 成功。

（3）跟踪路由:

r5#traceroute 192.168.1.6

Type escape sequence to abort.

Tracing the route to 192.168.1.6

1 10.1.1.4 64 msec 124 msec 48 msec

2 14.1.1.1 112 msec 92 msec 116 msec

3 12.1.1.2 260 msec 280 msec 236 msec

4 36.1.1.3 168 msec 196 msec 204 msec

5 36.1.1.6 268 msec * 248 msec

r5#

说明:R5 上已经成功穿越 MPLS 网络将数据包发送给远程用户网络，说明 MPLS_VPN 成功。

17.PE 到 CE 的通信

说明:因为 PE 将连 CE 的接口放入 VRF 之后，该接口就从全局路由表中消失，所以和 CE 的通信，由 VRF 路由表决定。

（1）查看 PE 的到 CE 的路由:

r1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

C 1.1.1.1 is directly connected, Loopback0

2.0.0.0/32 is subnetted, 1 subnets

O 2.2.2.2 [110/65] via 12.1.1.2, 00:51:02, Serial1/0

3.0.0.0/32 is subnetted, 1 subnets

O 3.3.3.3 [110/129] via 12.1.1.2, 00:51:02, Serial1/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/128] via 12.1.1.2, 00:51:02, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

r1#

说明:可以看到连 CE 的接口已经从全局路由表中消失。

（2）测试 ping:

r1#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r1#

说明:因为正常的 ping 是走的全局路由表，而全局路由表中没有到 CE 的接口，所以不通。

（3）使用 VRF 路由表:

r1#ping vrf vpn1 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 48/67/112 ms

r1#

说明:通过指定到 CE 的数据走 VRF 路由表，所以最后通信成功。

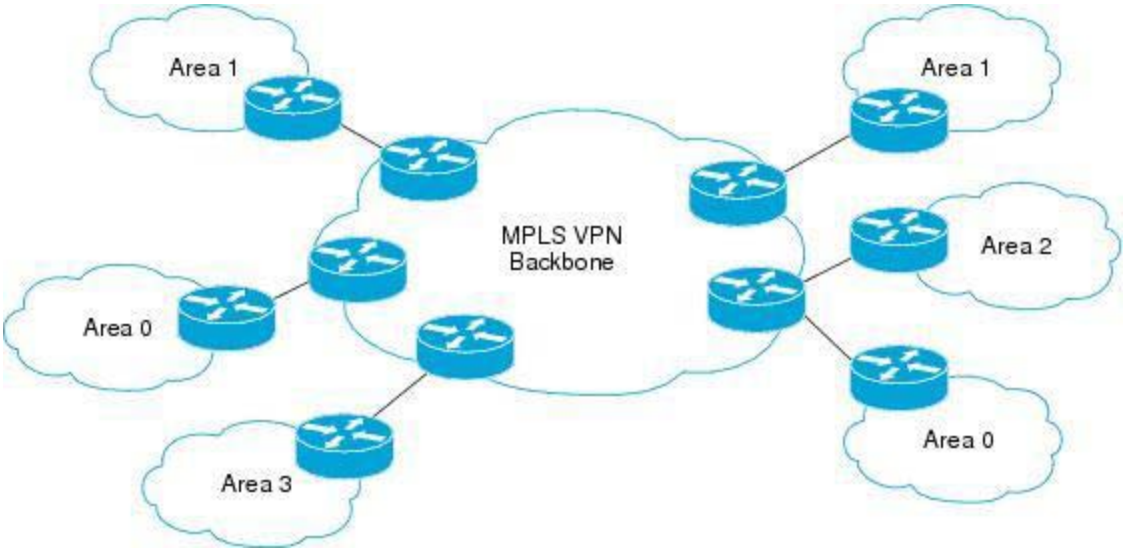
点此查看配置实例所有设备的 show running-config

OSPF Sham-Link

在 PE-CE 路由协议为 OSPF 时，当 OSPF 和 MP-BGP 之间互相重分布时，请不要改变 metric 值，因为这里的 metric 值是受保护的，如果您改了，可能造成路由环路，请小心配置。

当 PE 和 CE 使用 OSPF 时，PE 从 CE 学到路由后，都放在相应 VRF 中，然后传递给对端 PE，再由 PE 转发到远程 CE。

在 OSPF 中，当同时从 intra-area（同一个区域）和 inter-area(不同区域)学到相同路由时，intra-area 中的路由总是被优先选中。

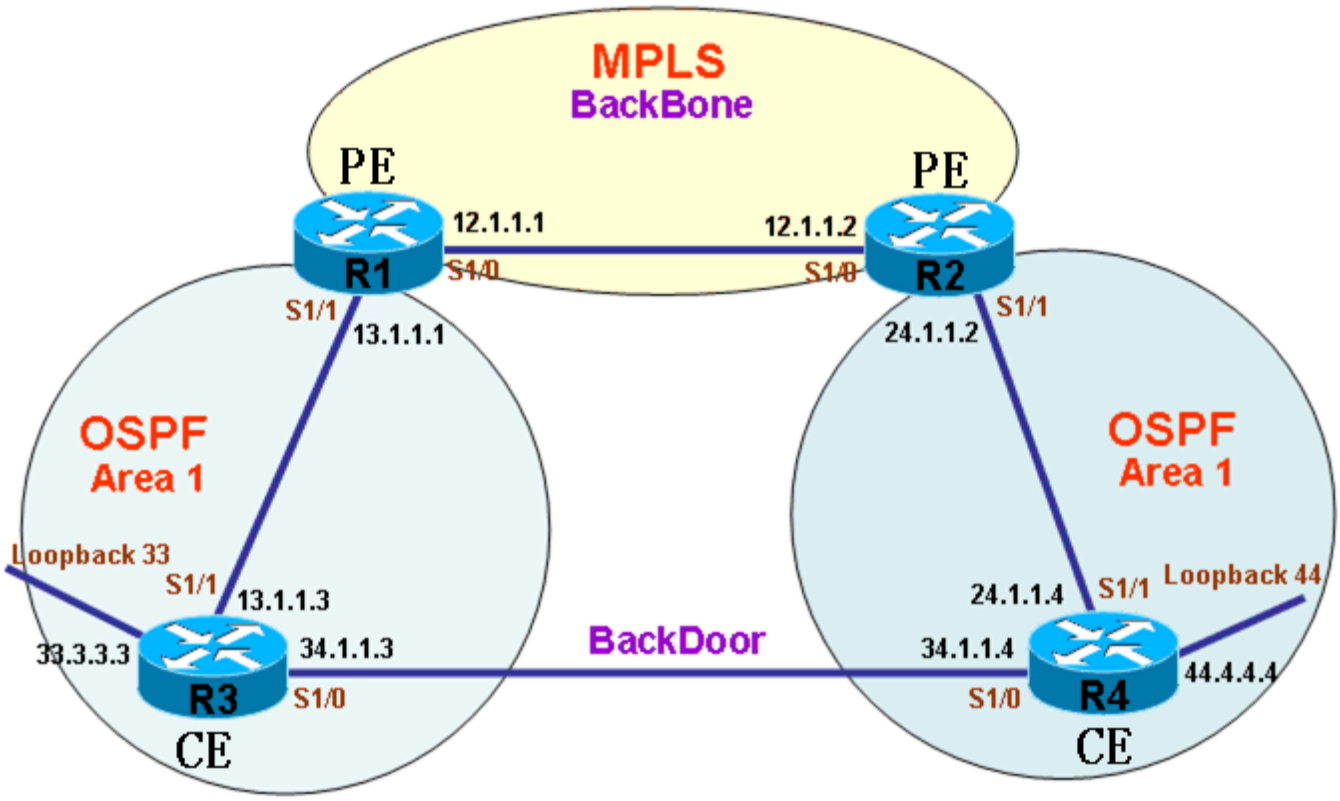


在上图中，有多个 MPLS VPN 场点同时连接到 MPLS VPN 骨干网络，PE-CE 路由协

议使用 OSPF，OSPF 区域分别有 0、1、2、3,当 MPLS VPN 场点之间互相再连接链路后，这样的链路被称为后门链路(BackDoor)，只要 OSPF 路由从这些后门链路上更新，那么势必会造成 PE 和 CE 路由器将 MPLS VPN 场点之间的流量优先选择从后门链路发送，而放弃从 MPLS VPN 骨干网络发送，原因为：

＊MPLS VPN 骨干网络中为 BGP，且为 IBGP，管理距离为 200，高于 OSPF 的值。

＊PE 和 CE 之间使用 OSPF，从同区域学习到的路由优先于其它区域路由。



根据上图中得出，当 CE 路由器 R3 从与 R4 的后门链路学习到 OSPF 路由 44.4.4.4 之后，R3 必定会优先选择从后门链路到达 44.4.4.4，因为从后门链路学习到的路由为同区域的，都为区域 1。不仅如此，PE 路由器 R1 也同样会选择从后门链路到达 44.4.4.4 而放弃从 MPLS VPN 骨干网络中传输。如果 CE 之间的后门链路仅仅是作为备份使用，那么将其选作主用链路，是不理智的。

如果要让 PE 路由器和 CE 路由器都优先选择从 MPLS VPN 骨干网络到达远程 VPN 场点，解决方法为，在 PE 路由器之间也创建一个 OSPF 区域，这样一来，从后门链路到达远程场点是 OSPF 路径，从 MPLS VPN 骨干网络中到达远程网络同样也是 OSPF

路径，所以就能轻松地控制 PE 和 CE 路由器到达远程场点的路径。

在 PE 之间创建额外的 OSPF 区域的方法是创建 OSPF Sham-Link，PE 之间的 Sham-Link 相当于一条逻辑的链路，但是这条链路也属于某个 OSPF 区域，供 PE 路由器选路使用。只要 PE 路由器想要从 MPLS VPN 骨干网络到达远程网络，就等于从 Sham-Link 上到达远程网络。

在 PE 上创建 OSPF Sham-Link 需要注意的是：

＊OSPF Sham-Link 也算是一条 OSPF 链路，这条链路可以指定为任何区域。

＊当 PE 到 CE 再从后门链路到远程网络都属于同一个 OSPF 区域，即学习到的路由为 intra-area 路由时，必须将 Sham-Link 也指定为同一区域，因为如果 Sham-Link 属于另一区域，那么就表示从 MPLS VPN 骨干网络到达远程网络是 inter-area 路由，而 inter-area 路由是不可能优先于 intra-area 路由的，所以也就不可能让到达远程网络的路径从 Sham-Link 走。

＊各个 PE 到 CE，以及 Sham-Link 都可以属于不同的区域，即使这些链路属于不同的区域，也可以控制 PE 和 CE 的选路，选路规则就是 intra-area 路由和 inter-area 路由的区别。所以当出事各类区域时，请自己控制好选路。

配置 Sham-Link 时，需要满足以下条件：

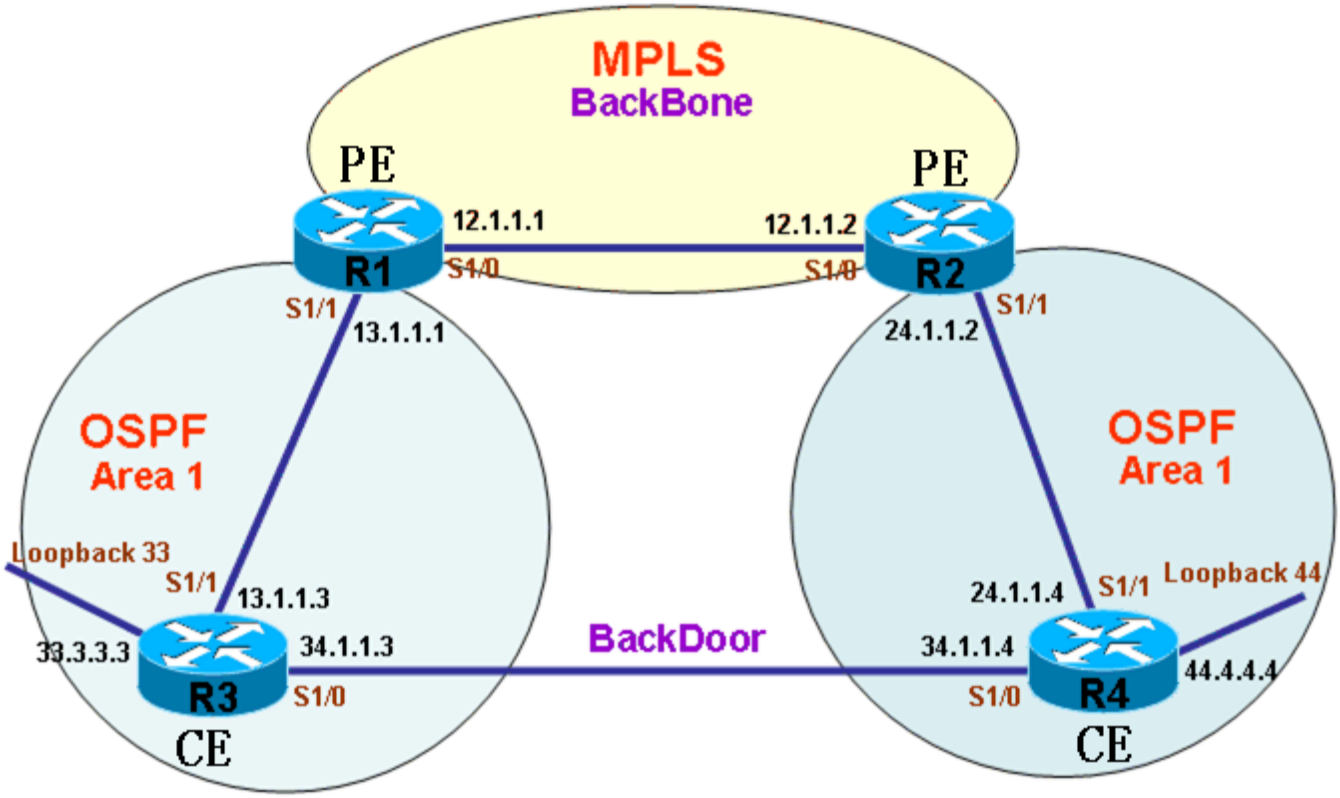
＊在 PE 上单独创建/32 位的地址，在 PE 之间使用这个地址来建立 Sham-Link。

＊这个 / 32 位地址的接口必须放入相应的 VRF。

＊这个 / 32 位地址必须在 BGP 里发布，而不能在 OSPF 里发布。

注：Sham-Link 是有 COST 值的，PE 穿越 MPLS VPN 骨干网络的 OSPF COST 值，就是 Sham-Link 的 COST 值，如果没有后门链路，Sham-Link 就不需要创建了..

配置 OSPF Sham-Link



说明:在上图中，CE 路由器 R3 和 R4 上分别有地址 33.3.3.3 和 44.4.4.4，配置 Sham-Link 使 PE 路由器到达远程场点地址从 MPLS VPN 骨干网络中传输。

注:Sham-Link 重要配置为第 7 步

1. 配置 R1 与 R2 的路由协议为 RIP

(1)在 R1 上配置 RIP

r1(config)#router rip

r1(config-router)#ver 2

r1(config-router)#no au

r1(config-router)#network 1.0.0.0 (loopback0)

r1(config-router)#network 12.0.0.0

(2)在 R1 上配置 RIP

r2(config)#router rip

r2(config-router)#ver 2

r2(config-router)#no au

r2(config-router)#network 2.0.0.0 (loopback0)

r2(config-router)#network 12.0.0.0

2. 在 R1 与 R2 上配置 MPLS

(1)在 R1 上配置 MPLS

r1(config)#int s1/0

r1(config-if)#mpls ip

r1(config)#mpls ldp router-id loopback 0 force

(2)在 R2 上配置 MPLS

r2(config)#int s1/0

r2(config-if)#mpls ip

r2(config)#mpls ld router-id loopback 0 force

3. 配置 MPLS VPN

(1)在 R1 上配置 MPLS VPN

r1(config)#ip vrf vpn

r1(config-vrf)#rd 100:1

r1(config-vrf)#route-target both 100:1

r1(config)#int s1/1

r1(config-if)#ip vrf forwarding vpn

(2)在 R2 上配置 MPLS VPN

r2(config)#ip vrf vpn

r2(config-vrf)#rd 100:1

r2(config-vrf)#route-target both 100:1

r2(config)#int s1/1

r2(config-if)#ip vrf forwarding vpn

4. 配置 OSPF

(1)在 R1 上配置 OSPF

r1(config)#router ospf 2 vrf vpn

r1(config-router)#router-id 1.1.1.1

r1(config-router)#network 13.1.1.1 0.0.0.0 a 1

(2)在 R2 上配置 OSPF

r2(config)#router ospf 2 vrf vpn

r2(config-router)#router-id 2.2.2.2

r2(config-router)#network 24.1.1.2 0.0.0.0 a 1

(3)在 R3 上配置 OSPF

r3(config)#router os 2

r3(config-router)#router-id 3.3.3.3

r3(config-router)#network 13.1.1.3 0.0.0.0 a 1

r3(config-router)#network 34.1.1.3 0.0.0.0 a 1

r3(config-router)#network 33.3.3.3 0.0.0.0 a 1

(4)在 R4 上配置 OSPF

r4(config)#router os 2

r4(config-router)#router-id 4.4.4.4

r4(config-router)#network 24.1.1.4 0.0.0.0 a 1

r4(config-router)#network 34.1.1.4 0.0.0.0 a 1

r4(config-router)#network 44.4.4.4 0.0.0.0 a 1

5. 配置 MP-BGP

(1)在 R1 上配置 MP-BGP

r1(config)#router bgp 100

r1(config-router)#no au

r1(config-router)#no sy

r1(config-router)#bg router-id 1.1.1.1

r1(config-router)#neighbor 2.2.2.2 remote-as 100

r1(config-router)#neighbor 2.2.2.2 up loopback 0

r1(config-router)#address-family vpnv4

r1(config-router-af)#neighbor 2.2.2.2 activate

r1(config-router-af)#neighbor 2.2.2.2 send-community both

r1(config-router-af)#exit

r1(config-router)#address-family ipv4 vrf vpn

r1(config-router-af)#redistribute ospf 2

r1(config)#router os 2 vrf vpn

r1(config-router)#re bgp 100 subnets

(2)在 R2 上配置 MP-BGP

r2(config)#router bgp 100

r2(config-router)#no au

r2(config-router)#no sy

r2(config-router)#bg router-id 2.2.2.2

r2(config-router)#nei 1.1.1.1 remote 100

r2(config-router)#neighbor 1.1.1.1 up loopback 0

r2(config-router)#address-family vpnv4

r2(config-router-af)#neighbor 1.1.1.1 activate

r2(config-router-af)#neighbor 1.1.1.1 send-community both

r2(config-router-af)#exit

r2(config-router)#address-family ipv4 vrf vpn

r2(config-router-af)#re ospf 2

r2(config)#router ospf 2 vrf vpn

r2(config-router)#re bgp 100 subnets

6.查看路由

(1)查看 PE 路由器 R1 的路由

```
r1#sh ip bgp vpnv4 all

BGP table version is 11, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete


   Network          Next Hop            Metric LocPrf Weight Path

Route Distinguisher: 100:1 (default for vrf vpn)

* i13.1.1.0/24      2.2.2.2                  192     100      0 ?

*>                  0.0.0.0                    0          32768 ?
```

```
* i24.1.1.0/24       2.2.2.2                  0    100      0 ?
*>                   13.1.1.3               192           32768 ?
* i33.3.3.3/32       2.2.2.2                129    100      0 ?
*>                   13.1.1.3                65           32768 ?
* i34.1.1.0/24       2.2.2.2                128    100      0 ?
*>                   13.1.1.3               128           32768 ?
* i44.4.4.4/32       2.2.2.2                 65    100      0 ?
*>                   13.1.1.3               129           32768 ?
r1#
```

说明:PE 路由器 R1 到达 33.3.3.3 和 44.4.4.4 都从 CE 路由器 R3 走。

(2)查看 PE 路由器 R1 的路由

```
r1#sh ip route vrf vpn


Routing Table: vpn
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
O       34.1.1.0 [110/128] via 13.1.1.3, 00:01:38, Serial1/1
     33.0.0.0/32 is subnetted, 1 subnets
O       33.3.3.3 [110/65] via 13.1.1.3, 00:01:39, Serial1/1
     24.0.0.0/24 is subnetted, 1 subnets
O       24.1.1.0 [110/192] via 13.1.1.3, 00:01:39, Serial1/1
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, Serial1/1
     44.0.0.0/32 is subnetted, 1 subnets
O       44.4.4.4 [110/129] via 13.1.1.3, 00:01:39, Serial1/1
r1#
```

说明:PE 路由器 R1 到达 33.3.3.3 和 44.4.4.4 都从 CE 路由器 R3 走。

7.在 PE 路由器之间创建 Sham-Link

(1)在 PE 路由器上创建/32 位 loopback 地址

R1:

```
r1(config)#int loopback 100
r1(config-if)#ip vrf forwarding vpn
```

r1(config-if)#ip add 100.1.1.1 255.255.255.255

说明:创建的/32 位地址必须放入 VRF。

R2:

r2(config)#int loopback 100

r2(config-if)#ip vrf forwarding vpn

r2(config-if)#ip add 100.1.1.2 255.255.255.255

说明:创建的/32 位地址必须放入 VRF。

(2)将/32 位地址在 MP-BGP 里发布

R1:

r1(config)#router bg 100

r1(config-router)#address-family ipv4 vrf vpn

r1(config-router-af)#network 100.1.1.1 mask 255.255.255.255

R2:

r2(config)#router bg 100

r2(config-router)#address-family ipv4 vrf vpn

r2(config-router-af)#network 100.1.1.2 mask 255.255.255.255

(3) 创建 Sham-Link

R1:

r1(config)#router ospf 2 vrf vpn

r1(config-router)#area 1 sham-link 100.1.1.1 100.1.1.2 cost 10

说明:创建 Sham-Link 时，要指定源地址和目的地址，并且指明 COST 值。

R2:

```
r2(config)#router ospf 2 vrf vpn
r2(config-router)#area 1 sham-link 100.1.1.2 100.1.1.1 cost 10
```

说明:创建 Sham-Link 时，要指定源地址和目的地址，并且指明 COST 值。

8. 查看结果

(1)在 PE 路由器 R1 上查看 Sham-Link

```
r1#sh ip ospf sham-links
Sham Link OSPF_SL0 to address 100.1.1.2 is up
Area 1 source address 100.1.1.1
  Run as demand circuit
  DoNotAge LSA allowed. Cost of using 10 State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Hello due in 00:00:04
    Adjacency State FULL (Hello suppressed)
    Index 2/2, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
r1#
```

说明: Sham-Link 建立成功。

(2)在 PE 路由器 R1 上查看 MP-BGP 的路由

```
r1#show ip bgp vpnv4 all

BGP table version is 21, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete


   Network          Next Hop            Metric LocPrf Weight Path

Route Distinguisher: 100:1 (default for vrf vpn)

*> 13.1.1.0/24      0.0.0.0                  0         32768 ?

r>i24.1.1.0/24      2.2.2.2                  0    100      0 ?

*> 33.3.3.3/32      13.1.1.3                65         32768 ?

* i34.1.1.0/24      2.2.2.2                128    100      0 ?

*>                  13.1.1.3                128         32768 ?

r>i44.4.4.4/32      2.2.2.2                  65    100      0 ?

*> 100.1.1.1/32     0.0.0.0                  0         32768 i

*>i100.1.1.2/32     2.2.2.2                  0    100      0 i

r1#
```

说明:到达远程场点 44.4.4.4 的路径选择从 MPLS VPN 骨干网络中走。

(3)在 PE 路由器 R1 上查看 VRF 的路由

```
r1#sh ip route vrf vpn

Routing Table: vpn
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

       E1 - OSPF external type 1, E2 - OSPF external type 2

       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

       ia - IS-IS inter area, * - candidate default, U - per-user static
route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     34.0.0.0/24 is subnetted, 1 subnets

O       34.1.1.0 [110/128] via 13.1.1.3, 00:03:02, Serial1/1

     100.0.0.0/32 is subnetted, 2 subnets

C       100.1.1.1 is directly connected, Loopback100

B       100.1.1.2 [200/0] via 2.2.2.2, 00:03:19

     33.0.0.0/32 is subnetted, 1 subnets

O       33.3.3.3 [110/65] via 13.1.1.3, 00:03:02, Serial1/1

     24.0.0.0/24 is subnetted, 1 subnets

O       24.1.1.0 [110/74] via 2.2.2.2, 00:03:03

     13.0.0.0/24 is subnetted, 1 subnets

C       13.1.1.0 is directly connected, Serial1/1
```

```
     44.0.0.0/32 is subnetted, 1 subnets
O       44.4.4.4 [110/75] via 2.2.2.2, 00:03:04
r1#
```

说明:到达远程场点 44.4.4.4 的路径选择从 MPLS VPN 骨干网络中走。

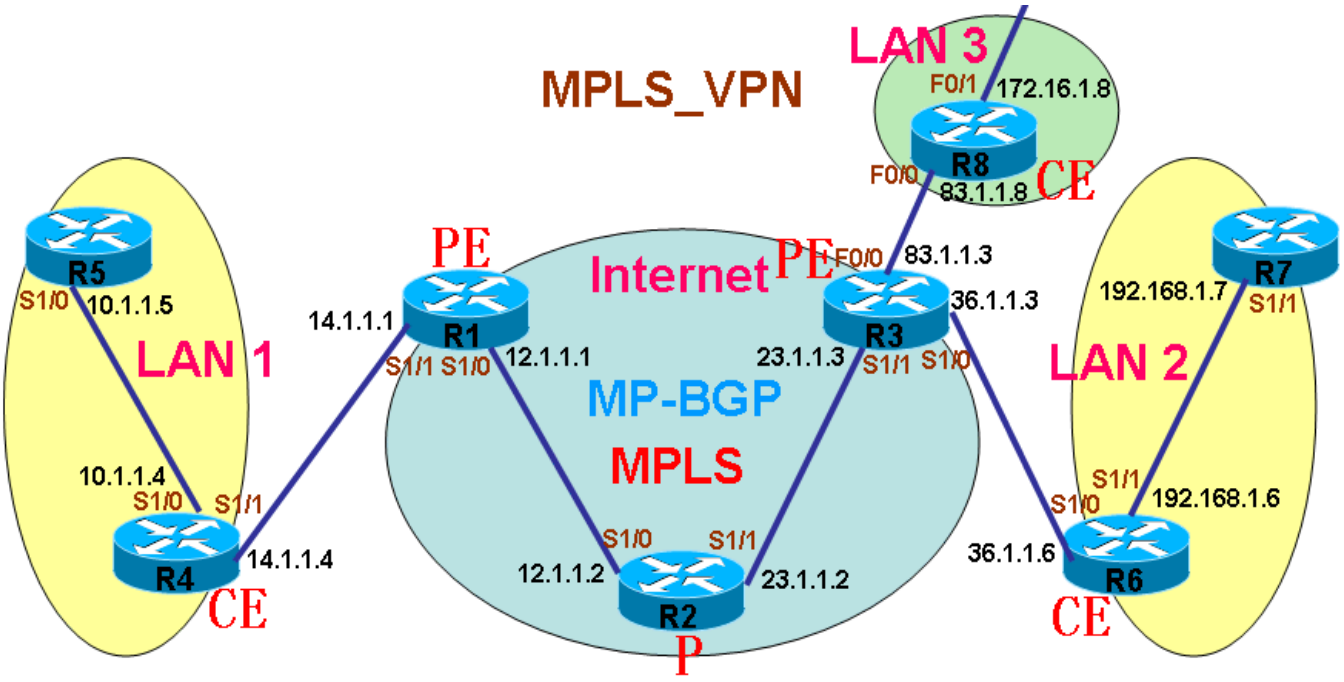
注：因为 sham-link 是要 COST 值的，相当于物理接口，要调整 CE 的选路，请调整 sham-link 和各接口 COST 值来完成。

外部通信

概述

以上是同一 VPN 的两个场点之间的通信，因为这两个场点之间 RD 是一样的，属于同一 VPN，所以这样的通信很好实现，被称为内部通信，但是如果不同 VPN 之间，即不同 RD 的 VPN 之间要通信，就要靠配置更多的 RT 来实现，这样的通信称为外部通信。

以下图为例，在原有两个 LAN 的基础上，加入 LAN 3，原有两个 LAN 之间的 RD 为 100：1，所以通信没有障碍，但是 LAN 3 的 RD 为 100：2，所以需要配置 RT 允许双方路由的导入导出，才能实现不同 LAN 的通信。



1.在 PE 为 LAN 上创建 VRF

（1） 在 PE R3 上创建 VRF，并配置不同的 RD，为 100：2

r3(config)#ip vrf vpn2

r3(config-vrf)#rd 100:2

r3(config-vrf)#route-target both 100:2

r3(config-vrf)#exit

r3(config)#int f0/0

r3(config-if)#ip vrf forwarding vpn2

% Interface FastEthernet0/0 IP address 83.1.1.3 removed due to enabling VRF vpn2

r3(config-if)#ip add 83.1.1.3 255.255.255.0

2.配置 PE-CE 路由协议

说明:配置 PE 连接 LAN 3 的路由协议，在配置 PE-CE 路由协议时，因为有多个 IGP 协议可供选择，之前我们已经举例过 RIP 和 OSPF 的协议，下面我们再使用 EIGRP 的 PE-CE 协议。

（1）在 PE R3 上配置 EIGRP:

```
r3(config)#router eigrp 1

r3(config-router)#no auto-summary

r3(config-router)#address-family ipv4 vrf vpn2

r3(config-router-af)#no auto-summary

r3(config-router-af)#network 83.1.1.3 0.0.0.0

r3(config-router-af)#autonomous-system 1

r3(config-router-af)#redistribute bgp 100 metric 10000 100 255 1 1500
```

说明:在 PE 上配置 EIGRP 时，需要在 address-family 里面再次指定对方邻居的 AS 号，此 AS 号为对方正确配置的 AS 号码，并且将 MP-BGP 的路由重分布进 EIGR。

3.配置 EIGRP 重分布进 BGP

（1）在 R3 上配置 EIGRP 重分布进 BGP:

```
r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn2
```

```
r3(config-router-af)#redistribute eigrp 1
```

4.查看 MP-BGP 中的 VRF 路由表

```
r3#sh ip bgp vpnv4 all

BGP table version is 25, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

              r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete


   Network          Next Hop            Metric LocPrf Weight Path

Route Distinguisher: 100:1 (default for vrf vpn1)

*>i10.1.1.0/24      1.1.1.1                  1    100      0 ?

*>i14.1.1.0/24      1.1.1.1                  0    100      0 ?

*> 36.1.1.0/24      0.0.0.0                  0         32768 ?

*> 192.168.1.6/32   36.1.1.6                65         32768 ?

Route Distinguisher: 100:2 (default for vrf vpn2)

*> 83.1.1.0/24      0.0.0.0                  0         32768 ?

*> 172.16.1.0/24    83.1.1.8            409600         32768 ?

r3#
```

说明:从 VRF 表中可以看出，LAN 3 的 VRF 和其它两个 LAN 的 VRF 并不同步，所以

不可能实现通信。

5.配置 RT 允许双方 VRF 进入

说明:因为双方 VRF 的允许的 RD 不一样，所以路由表无法彼此进入，因此可以配置 RT 允许相互进出。

（1） 在 PE R3 上配置 RT 同时允许双方 RD:

```
r3(config)#ip vrf vpn1
r3(config-vrf)#route-target both 100:2

r3(config)#ip vrf vpn2
r3(config-vrf)#route-target both 100:1
```

（2）在 PE R1 上配置 RT 同时允许双方 RD:

```
r1(config)#ip vrf vpn1
r1(config-vrf)#route-target both 100:2
```

6.查看双方 VRF 路由表

说明:因为 RT 已经允许双方路由进出相互的 VRF，所以双方 VRF 路由表中应该有相互的路由信息。

（1） 在 R3 上查看 MP-BGP VRF 路由表:

```
r3#sh ip bgp vpnv4 all

BGP table version is 37, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

              r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete


   Network          Next Hop            Metric LocPrf Weight Path

Route Distinguisher: 100:1 (default for vrf vpn1)

*>i10.1.1.0/24      1.1.1.1                  1    100      0 ?

*>i14.1.1.0/24      1.1.1.1                  0    100      0 ?

*> 36.1.1.0/24      0.0.0.0                  0         32768 ?

*> 83.1.1.0/24      0.0.0.0                  0         32768 ?

*> 172.16.1.0/24    83.1.1.8            409600         32768 ?

*> 192.168.1.6/32   36.1.1.6                65         32768 ?

Route Distinguisher: 100:2 (default for vrf vpn2)

*>i10.1.1.0/24      1.1.1.1                  1    100      0 ?

*>i14.1.1.0/24      1.1.1.1                  0    100      0 ?

*> 36.1.1.0/24      0.0.0.0                  0         32768 ?

*> 83.1.1.0/24      0.0.0.0                  0         32768 ?

*> 172.16.1.0/24    83.1.1.8            409600         32768 ?

*> 192.168.1.6/32   36.1.1.6                65         32768 ?
```

r3#

说明:双方的 VRF 已经同步。

（2）在 CE R4 上查看路由表:

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

83.0.0.0/24 is subnetted, 1 subnets

R 83.1.1.0 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

36.0.0.0/24 is subnetted, 1 subnets

R 36.1.1.0 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

172.16.0.0/24 is subnetted, 1 subnets

R 172.16.1.0 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Serial1/0

192.168.1.0/32 is subnetted, 1 subnets

R 192.168.1.6 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, Serial1/1

r4#

说明:双方的 VRF 已经同步。

7.测试两个 LAN 的连通性

说明:因为双方的 VRF 路由表已经相同，所以应该可以通信

（1）在 CE 路由器 R5 上测试到 LAN3 172.16.1.8 的连通性：

R5ping

r5#ping 172.16.1.8

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.8, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 188/241/340 ms

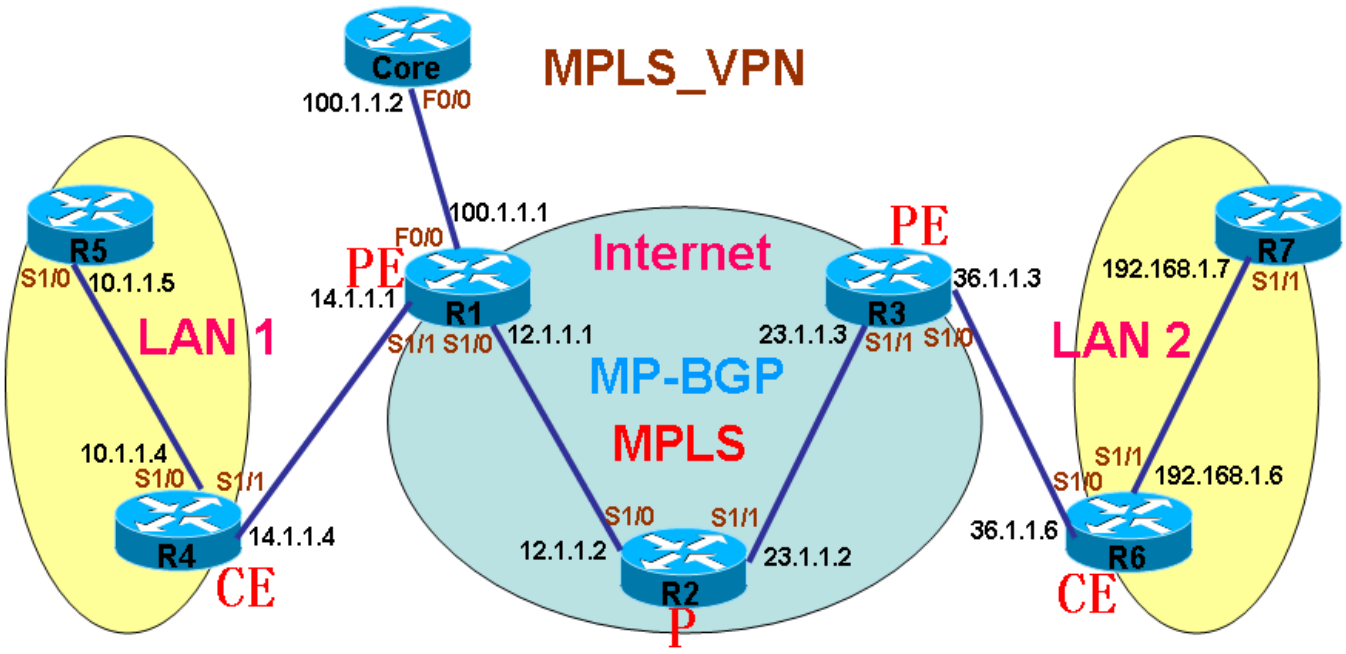
r5#

说明:成功实现不同 LAN 之间的外部通信。

点此查看配置实例所有设备的 show running-config

CE 接入英特网

概述



如上图所示，PE 路由器 R1 与 Internet 核心路由器 Core 相连，因为 PE 路由器 R1 连 CE 路由器 R4 的接口已经被划入 VRF vpn1 中，所以从 CE R4 过来的数据包，PE 都是根据 VRF vpn1 的路由表来作出转发决定的，当 CE 要将数据包发往 Internet 的时候，PE 是不会转发的，因为在 VRF vpn1 中，除了对方 LAN 的路由信息之外，并无到 Internet 核心的路由，所以要让 CE 到 Internet 核心的数据包被 PE 转发，那么就必须在 PE 上为 VRF vpn1 配置到 Internet 的路由，这样，CE 才能直接访问 Internet。

除了在 VRF 中静态为 CE 添加路由之外，还有一种方法就是，在 PE 和 CE 之间额外创建 tunnel，当 CE 到 Internet 的数据包发往 Tunnel 时，PE 就将该数据包发往 Internet。

最后一种让 CE 能够访问 Internet 的方法就是，CE 将自己的路由发往对端 LAN，让对端的 LAN 作转发处理。

配置

1.配置 VRF 静态路由

说明:直接为用户的 VRF 写到 Internet 的默认路由，让 CE 的 VRF 拥有到 Internet 的默认路由之后，就可以访问 Internet。

（1）在 PE 上为 VRF 配置默认路由:

r1(config)#ip route vrf vpn1 0.0.0.0 0.0.0.0 100.1.1.2 global

说明:为 VRF vpn1 添加默认路由指向下一跳 100.1.1.2

（2）在 PE 上还需手工指定到 CE 内部网络的路由:

r1(config)#ip route 10.1.1.0 255.255.255.0 serial 1/1 14.1.1.4

（3）最后 CE 上也要写默认路由到 PE:

R4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.1

（4）测试 CE 到 Internet 核心的连通性：

r5#ping 100.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 100.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 188/241/340 ms

r5#

说明:CE 到 Internet 核心的通信正常。

2.在 PE-CE 间配置 Tunnel

（1）在 PE 上配置到 CE 和 Tunnel：

r1(config)#int tunnel 0

r1(config-if)#ip address 50.1.1.1 255.255.255.0

r1(config-if)#tunnel source 14.1.1.1

r1(config-if)#tunnel destination 14.1.1.4

r1(config-if)#tunnel vrf vpn1

r1(config-if)#exit

说明:配置了到 CE 的 Tunnel 之后，还必须将该 Tunnel 放到 VRF 中。

（2）配置到 CE 内部网络的路由都走 tunnel:

r1(config)#ip route 10.1.1.0 255.255.255.0 tunnel 0

（3）在 CE 上配置到 PE 的 tunnel:

r4(config)#int tunnel 0

r4(config-if)#ip address 50.1.1.4 255.255.255.0

r4(config-if)#tunnel source 14.1.1.4

r4(config-if)#tunnel destination 14.1.1.1

r4(config-if)#exit

(4)指定 CE 所有的路由都从 Tunnel 发给 PE：

r4(config)#ip route 0.0.0.0 0.0.0.0 tunnel 0

（5）测试 CE 到 Internet 核心的连通性：

r5#ping 100.1.1.2

第三种方法将所有数据发到自己核心总部再转出去，略。

更多 PE-CE 路由协议

说明:之前说过 PE-CE 的路由协议有多种，我们已经举例过 RIP，OSPF，EIGRP，下面来举例静态写 VRF，还有就是 EBGP。

1.静态写 VRF 路由

说明:PE 的 VRF 中如果没有到 CE 的内部路由，将无法为不同 LAN 之间提供数据包转发，所以 PE 必须获得 CE 的内部网络路由信息，可以使用动态路由协议，也可以静态写 VRF 路由。

（1）在 PE R1 上写静态的 VRF 路由指向 CE：

r1(config)#ip route vrf vpn1 10.1.1.0 255.255.255.0 14.1.1.4

（2）在 PE R3 上写静态的 VRF 路由指向 CE：

r3(config)#ip route vrf vpn1 192.168.1.0 255.255.255.0 36.1.1.6

（3）在 PE R1 上将静态路由重分布进 MP-BGP:

说明:PE 上的 VRF 中已经拥有到 CE 内部网络的路由后，如果这个信息不发给远程 LAN，那么双方 LAN 之间也不能通信，所以可以将此静态路由重分布进 MP-BGP，从而发给远程 LAN。

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute static

r1(config-router-af)#exi

（4）在 PE R3 上将静态路由重分布进 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#redistribute static

(5)CE R4 也要写默认路由指向 PE：

r4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.1

（6）CE R6 也要写默认路由指向 PE：

r6(config)#ip route 0.0.0.0 0.0.0.0 36.1.1.3

(7)测试双方 LAN 的连通性：

r5#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 148/236/320 ms

r5#

说明:通过 PE 和 CE 之间写静态路由，双方 LAN 之间通信正常。

2.PE-CE 之间运行 EBGP

说明:PE 和 CE 之间只能运行 EBGP，因为当 PE 学到 CE 的路由后，要发给远程 LAN，所以此路由需要导入 MP-BGP，但是这种重分布在 EBGP 和 MP-BGP 之间会自动执行，无需额外配置。

（1）在 PE R1 上配置到 CE 的 EBGP：

r1(config)#router bg 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#neighbor 14.1.1.4 remote-as 200

r1(config-router-af)#neighbor 14.1.1.4 activate

（2）在 PE R3 上配置到 CE 的 EBGP：

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#neighbor 36.1.1.6 remote-as 300

r3(config-router-af)#neighbor 36.1.1.6 activate

r3(config-router-af)#

（3）在 CE R4 上配置到 PE 的 EBGP：

r4(config)#router bgp 200

r4(config-router)#neighbor 14.1.1.1 remote

r4(config-router)#neighbor 14.1.1.1 remote-as 100

r4(config-router)#network 10.1.1.0 mask 255.255.255.0

（4）在 CE R6 上配置到 PE 的 EBGP：

r6(config)#router bgp 200

r6(config-router)#neighbor 36.1.1.3 remote-as 100

r6(config-router)#network 192.168.1.0 mask 255.255.255.0

（5）解决 BGP 路由问题：

说明:因为在 LAN1 和 LAN2 之间配置的 BGP AS 号码都为 200，而本端 BGP 在收到路由时，如果发现路由携带的 AS 和自己的 AS 相同，则丢弃该路由，所以双方的 CE 都不会有对方的路由，所以这个问题要在 PE 的 BGP 上配置允许 AS 重叠。

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#neighbor 14.1.1.4 as-override

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#neighbor 36.1.1.6 as-override

（6）测试连通性：

r5#ping 192.168.1.6

Type escape sequence to abort.

```
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 148/236/320 ms

r5#
```

说明:通过 PE 上额外的配置，CE 之间的 BGP AS 号码重叠问题被解决，双方 LAN 之间实现通信。

Multi-VRF CE / VRF-Lite

概述

当在 PE 上将连接 CE 的接口放入 VRF 表之后，那么从此接口过来的所有 CE 数据都属于该 VRF，都根据该 VRF 来做出转发决定，所以从同一个接口发来的用户数据，都执行相同的路由查询。如果一个很大的公司,这个公司有两个分公司通过 Internet 相连,并且每个分公司都存在多个部门，那么当在正常实施 MPLS_VPN 时，要么让两个公公司的所有部门都能通信，因为他们走的是同一个 VRF 路由表，要么都不能通信。

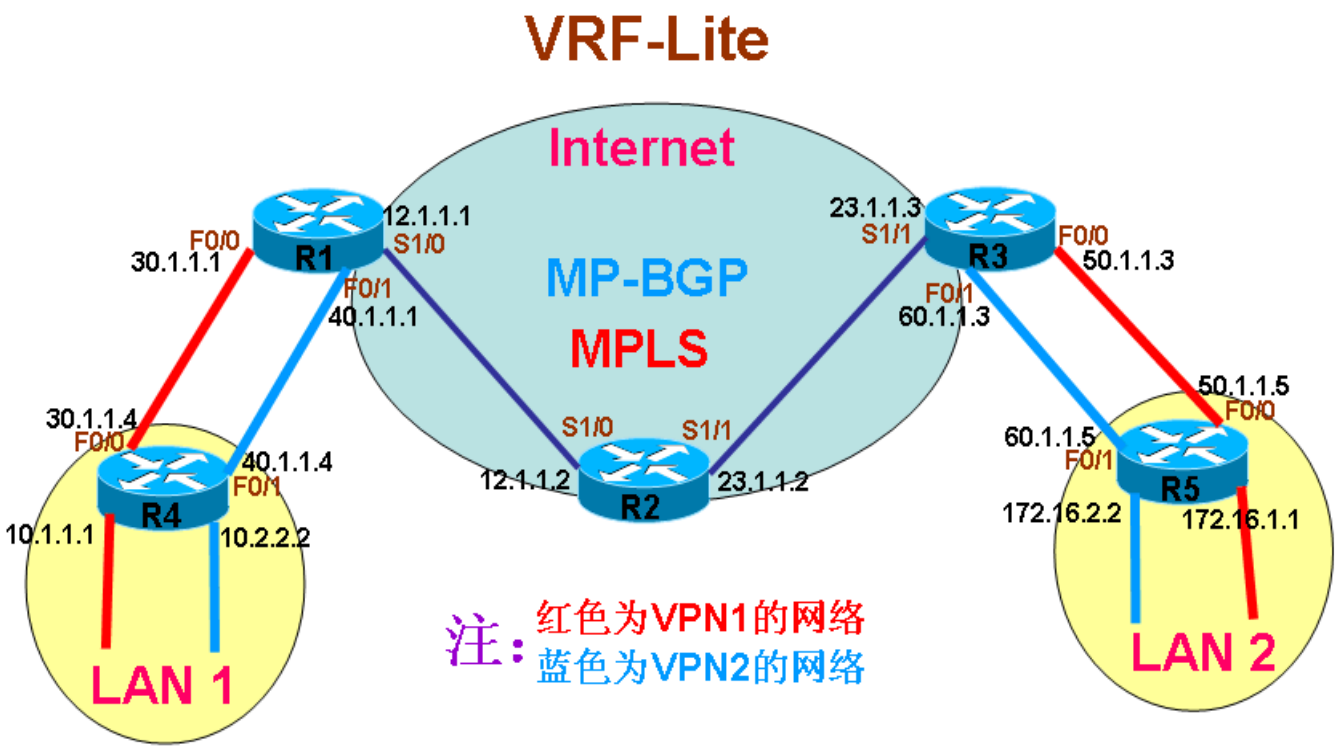
当某些时候因为业务要求，如果要让两个分公司的不同部门之间不能通信，按照 MPLS_VPN 的理论，就应该为要通信的部门单独创建 VRF，再为不要通信的部门之间单独创建 VRF，只有这样，才能将不同部门的数据隔离开来。

正因为 CE 连到 PE 的接口所有数据都属于同一个 VRF，所以当 CE 和 PE 之间只有一根链路相连时，要创建多个 VRF，这是不能实现的。但是，如果 PE 和 CE 之间是以太网链路，是可以支持子接口划分的，也就是可以通过为子接口划分不同的 VLAN 来实现。除了子接口之外，最好的办法就是在 PE 和 CE 间拉多条线路。

如果只通过在 PE 上将不同链路划入不同的 VRF，也可以达到隔离部门的需求。但是还有个方法就是，还可以直接在 CE 上面就将连接不同部门的接口直接划入不同的 VRF。因为在平时，要将接口划入 VRF，是在 PE 上做的，而 CE 是不了解 VRF

的，也就是说 MPLS_VPN 对于 CE 端来说是透明的，那么现在要让 CE 也能够认识 VRF，能够成功为不同接口划不同的 VRF，这就需要扩展 CE 具有 PE 的功能，这种功能被称为 Multi-VRF CE 或 VRF-Lite，在 CE 上为不同接口划 VRF 的同时，PE 上的 VRF 将继续保留不可删除，所以 PE 的功能将不作变动。而在 CE 将路由传递给 PE 时，这中间运行的协议是 OSPF，并且还要扩展 OSPF 的 vrf-lite 功能。

下面以下图为例，来详细讲解 VRF-Lite 的配置



说明:红色部分网络为需要通信的部门，但和蓝色网络的部门作隔离，其中 R1、R2、R3 上均已配置 loopback0，地址分别为 1.1.1.1/32，2.2.2.2/32，3.3.3.3/32，并且已经配置好 OSPF，让 MPLS 区域内所有直连接口和 loopback 口互通。下面以配置红色网络部门通信为例，蓝色网络部门通信与红色部门相似，请自行参考配置。

1.将 MPLS 区域网络配通

说明:此步略过，详细步骤请参见之前部分。

2.配置 MP-BGP

说明:配置 R1 和 R3 之间的 MP-BGP

（1） 在 R1 上配置 MP-BGP:

```
r1(config)#router bgp 100
r1(config-router)#neighbor 3.3.3.3 remote-as 100
r1(config-router)#neighbor 3.3.3.3 update-source loopback 0
r1(config-router)#address-family vpnv4
r1(config-router-af)#neighbor 3.3.3.3 activate
r1(config-router-af)#neighbor 3.3.3.3 send-community both
```

（2）在 R1 上配置 MP-BGP:

```
r3(config)#router bgp 100
r3(config-router)#neighbor 1.1.1.1 remote-as 100
r3(config-router)#neighbor 1.1.1.1 update-source loopback 0
r3(config-router)#address-family vpnv4
r3(config-router-af)#neighbor 1.1.1.1 activate
r3(config-router-af)#neighbor 1.1.1.1 send-community both
```

3.在 CE 上为不同部门创建不同 VRF

说明:在 CE 上为不同部门创建不同 VRF，并配置好 RD 和 RT

（1）在 R4 上为相应接口创建 VRF：

r4(config)#ip vrf vpn1

r4(config-vrf)#rd 100:1

r4(config-vrf)#route-target both 100:1

（2）在 R5 上为相应接口创建 VRF：

r5(config)#ip vrf vpn1

r5(config-vrf)#rd 100:1

r5(config-vrf)#route-target both 100:1

4.将相应部门的接口划入相应 VRF

（1） 在 R4 上将相应接口划入相应 VRF:

r4(config)#int loopback 10

r4(config-if)#ip vrf forwarding vpn1

r4(config-if)#ip add 10.1.1.1 255.255.255.0

r4(config)#int f0/0

r4(config-if)#ip vrf forwarding vpn1

r4(config-if)#ip add 30.1.1.4 255.255.255.0

（2）在 R5 上将相应接口划入相应 VRF:

r5(config)#int loopback 172

r5(config-if)#ip vrf forwarding vpn1

r5(config-if)#ip add 172.16.1.1 255.255.255.0

r5(config)#int f0/0

r5(config-if)#ip vrf forwarding vpn1

r5(config-if)#ip add 50.1.1.5 255.255.255.0

5.配置 PE-CE 间的 OSPF

（1） 在 CE R4 上启动 OSPF，并将相应路由放进 OSPF 进程，传送给 PE:

r4(config)#router ospf 100 vrf vpn1

r4(config-router)#capability vrf-lite

r4(config-router)#network 10.1.1.1 0.0.0.0 a 0

r4(config-router)#network 30.1.1.4 0.0.0.0 a 0

注：命令 capability vrf-lite 为扩展 OSPF，必须输入，否则 OSPF 无法接收路由。

（2）在 CE R5 上启动 OSPF，并将相应路由放进 OSPF 进程，传送给 PE:

r5(config)#router ospf 100 vrf vpn1

r5(config-router)#capability vrf-lite

r5(config-router)#network 172.16.1.1 0.0.0.0 a 0

r5(config-router)#network 50.1.1.5 0.0.0.0 a 0

注：命令 capability vrf-lite 为扩展 OSPF，必须输入，否则 OSPF 无法接收路由。

6.在 PE 上创建 VRF

（1）在 PE R1 上为 CE 的不同部门也创建不同的 VRF，并将连 CE 的相应线路划入相应 VRF:

```
r1(config)#ip vrf vpn1

r1(config-vrf)#rd 100:1

r1(config-vrf)#route-target both 100:1



r1(config)#int f0/0

r1(config-if)#ip vrf forwarding vpn1

r1(config-if)#ip add 30.1.1.1 255.255.255.0
```

(2)在 PE R3 上为 CE 的不同部门也创建不同的 VRF，并将连 CE 的相应线路划入相应 VRF:

```
r3(config)#ip vrf vpn1

r3(config-vrf)#rd 100:1

r3(config-vrf)#route-target both 100:1



r3(config)#int f0/0

r3(config-if)#ip vrf forwarding vpn1

r3(config-if)#ip add 50.1.1.3 255.255.255.0
```

7.在 PE 上启动 OSPF

（1）在 PE R1 上也启用 OSPF，用来接收 CE 发来的各部门的路由，以便让这些部门相通：

r1(config)#router ospf 100 vrf vpn1

r1(config-router)#network 30.1.1.1 0.0.0.0 a 0

(2)在 PE R1 上也启用 OSPF，用来接收 CE 发来的各部门的路由，以便让这些部门相通：

r3(config)#router ospf 100 vrf vpn1

r3(config-router)#network 50.1.1.3 0.0.0.0 a 0

8.在 MP-BGP 和 OSPF 间重分布

说明:将 MP-BGP 和 OSPF 相互重分布，最终让相应部门实现通信。

（1） 在 PE R1 上做 MP-BGP 和 OSPF 的双向重分布：

r1(config)#router ospf 100 vrf vpn1

r1(config-router)#redistribute bgp 100 subnets

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute ospf 100 vrf vpn1

（2）在 PE R3 上做 MP-BGP 和 OSPF 的双向重分布：

r3(config)#router ospf 100 vrf vpn1

r3(config-router)#redistribute bgp 100 subnets

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#redistribute ospf 100 vrf vpn1

9.查看 CE 各自 VRF 的路由

（1）在 CE R4 上查看 VRF vpn1 的路由:

r4#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

50.0.0.0/24 is subnetted, 1 subnets

O IA 50.1.1.0 [110/11] via 30.1.1.1, 00:06:00, FastEthernet0/0

172.16.0.0/24 is subnetted, 1 subnets

O IA 172.16.1.0 [110/21] via 30.1.1.1, 00:06:00, FastEthernet0/0

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Loopback10

30.0.0.0/24 is subnetted, 1 subnets

C 30.1.1.0 is directly connected, FastEthernet0/0

r4#

说明:CE R4 的 VRF vpn1 已经拥有双方部门的路由，应该可以实现部门间通信。

（2）在 CE R5 上查看 VRF vpn1 的路由:

r5#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

```
      ia - IS-IS inter area, * - candidate default, U - per-user static route

      o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


    50.0.0.0/24 is subnetted, 1 subnets
C       50.1.1.0 is directly connected, FastEthernet0/0
    172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Loopback172
    10.0.0.0/24 is subnetted, 1 subnets
O IA    10.1.1.0 [110/21] via 50.1.1.3, 00:05:07, FastEthernet0/0
    30.0.0.0/24 is subnetted, 1 subnets
O IA    30.1.1.0 [110/11] via 50.1.1.3, 00:05:07, FastEthernet0/0
r5#
```

说明:CE R4 的 VRF vpn1 已经拥有双方部门的路由，应该可以实现部门间通信。

10.测试连通性

说明:双方 CE 在相应 VRF 都拥有双方部门的路由信息，因此可以实现部门间通信。

（1）在 CE R4 上 ping 远程部门地址做测试:

说明:在上图中，测试时，因为数据从 CE 发出，默认不是走 VRF 表，所以发送 ICMP 时，除了指定源地址外，还要指定数据走相应的 VRF 路由表，否则测试不正确。

```
r4#ping vrf vpn1 172.16.1.1 source loopback 10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 96/116/164 ms
r4#
```

说明:通过配置 VRF Lite，实现了远程公司之间的部门通信，而不在同一 VRF 的不同部门，通信将被隔离。

点此查看配置实例所有设备的 show running-config

以上配置实例为红色网络部门间的通信，只要配置蓝色部门为不同的 VRF 和不同的 RD，通信将可实现隔离功能，蓝色网络的通信，请自行参照红色网络的配置。