

MPLS control plane



肖宏辉 · 2 个月前

相关阅读：[MPLS基础](#)

上一篇主要讲了MPLS的data plane。上一篇介绍过的术语解释在这一篇会继续用。Data plane就是MPLS的转发过程，而MPLS的转发就是基于Label的转发，上一篇介绍了一个已经配置好的MPLS网络中，MPLS packet是如何在LER（或者PE）之间进行转发。这篇来看一下一个MPLS网络是如何生成的。先从几个问题开始：

- 传统的路由网络中，路由器是基于路由做转发。那路由是怎么到路由器上的？可以静态配置，也可以通过IGP，BGP来广播学习。对于MPLS网络，有同样的问题，Label信息是怎么到路由器上来的？
- 另一方面，上篇说过，MPLS转发与传统的路由转发区别在于，传统的路由转发是无

状态的，但是MPLS的转发实际上是需要了解下一跳路由器的状态（也就是Label，需要将下一跳路由器能识别的Label提前加入到MPLS packet中），下一跳路由器的Label信息是如何到当前路由器来的？

- 在上一篇的最后一个图例中，第一步是PE路由器之间建立LSP是怎么建立的？

Local Label的生成

上一篇提到了FEC（交换等价类），一个FEC中的IP packet会走相同的路径。在传统路由网络里面，路由的prefix与FEC相关；在MPLS网络里面，Label与FEC相关。那对于LSR来说，将prefix和Label做绑定，就是自然的选择。LSR会对路由器本身的所有直连的，静态的，IGP/BGP学习的路由，都随机分配一个本地的Label。这也就是LSR上Local Label的生成过程。也就是说MPLS Label与传统路由中的IP prefix相关联。MPLS网络里面，网络数据包仍然是沿着传统路由确定的路径前进，只是交换方式换成了Label。

Remote Label的传递

MPLS网络中，LSP路径上的两个相邻的路由器Ru和Rd。如果MPLS packet是从Ru发送到Rd。那么Ru称为upstream LSR，Rd称为downstream LSR。

为了能让downstream LSR能识别MPLS packet，数据包在发送给downstream LSR之前，必须先打上downstream LSR针对当前FEC生成的Local Label。那么upstream LSR必须事先知道downstream LSR的Local Label。或者说，downstream LSR必须事先告知upstream LSR。

这种告知是通过LEP（Label Exchange Protocol）完成的。MPLS架构中并没有指定一种LEP，实时上也有多种LEP，例如RFC3036定义的LDP，BGP的扩展MP-BGP，RSVP等。所有这些协议都定义了downstream LSR如何将自己的Local Label告知upstream LSR。对于upstream LSR来说，被告知的Label就是Remote Label。

LDP

前面说过，实际中有多种LEP，它们各有特点。这里主要介绍由RFC3036(注：已经被RFC5036替代)定义的LDP。LDP传递的是什么？是目的IP地址与Local Label的binding信息，或者称为prefix-label binding。经过这样的信息传递，其他的LSR才知道将MPLS packet

发到这个LSR时，应该事先打上什么Label。LDP由两部分组成：

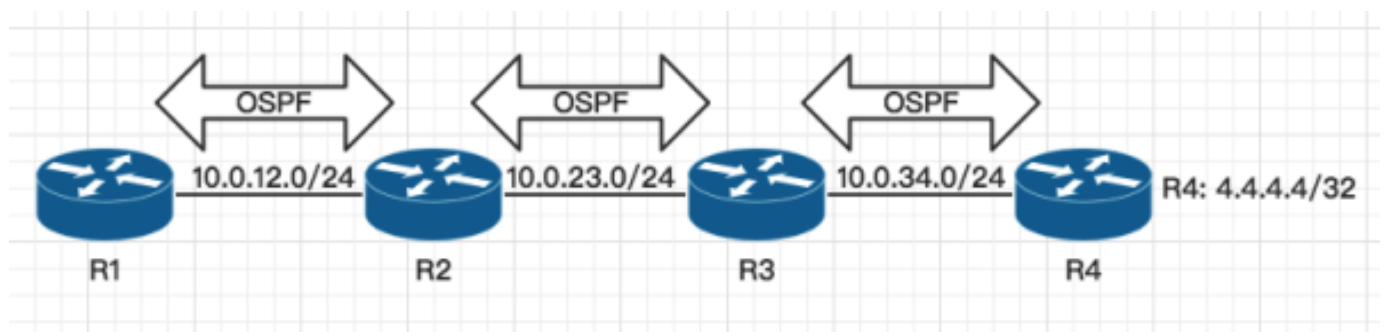
- Hello message：广播到224.0.0.2的UDP请求。工作在UDP646端口。包含了LDP的router-id，router-id必须是当前LSR的一个可用IP。router-id可以通过配置指定，如果未配置则使用最高值的loopback地址，如果没有loopback就用最高值的网卡地址。Cisco设备可以通过 `mpls ldp router-id <interface>` 指定对应网卡来获取IP作为router-id。
- TCP session：在LSR之间交换Label。工作在TCP646端口。TCP连接建立在Hello message所包含的router-id之间，所以要求router-id必须是一个可用的IP。

为什么选用TCP来作为Label交换的传输层？因为TCP是一个可靠的能传输大量数据的传输层协议，MP-BGP也是基于TCP的，因为MP-BGP是基于BGP，而BGP的传输层协议是TCP。所以，LDP是这样一个工作过程。首先通过UDP广播获取当前所有的LDP neighbor，之后与这些neighbor之间建立TCP session来交换prefix-label binding。这样，LSR就能知道所有的与自己相连的LSR的Label信息，并且存储为Remote Label在本地。这些步骤一旦配置好了都会自动执行。

LSP的建立

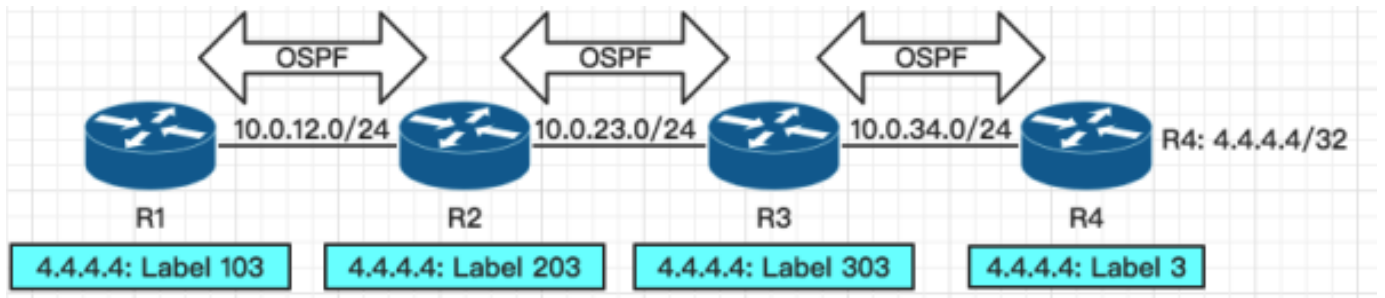
来看最后一个问题。看图说话，现在有4个路由器，路由器4上有一个直连路由4.4.4.4/32，需要建立从R1到R4针对4.4.4.4/32地址的LSP。

1.



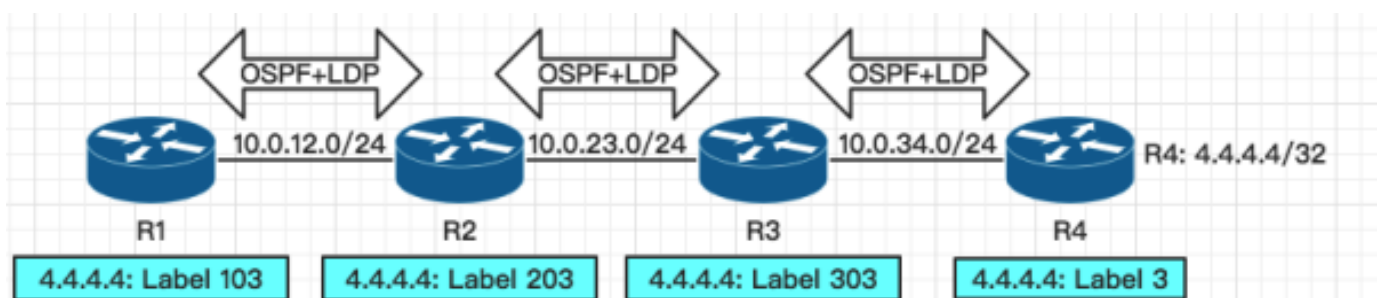
首先，配置好4个路由之间的OSPF连接，这样每个路由器都有一条从OSPF学习的目的地址是4.4.4.4/32的本地路由。这里可以是别的IGP，或者BGP协议，甚至静态配置路由。但是为了让之后的步骤能进行，每个路由器必须有道目的地址4.4.4.4/32的本地路由。

2.



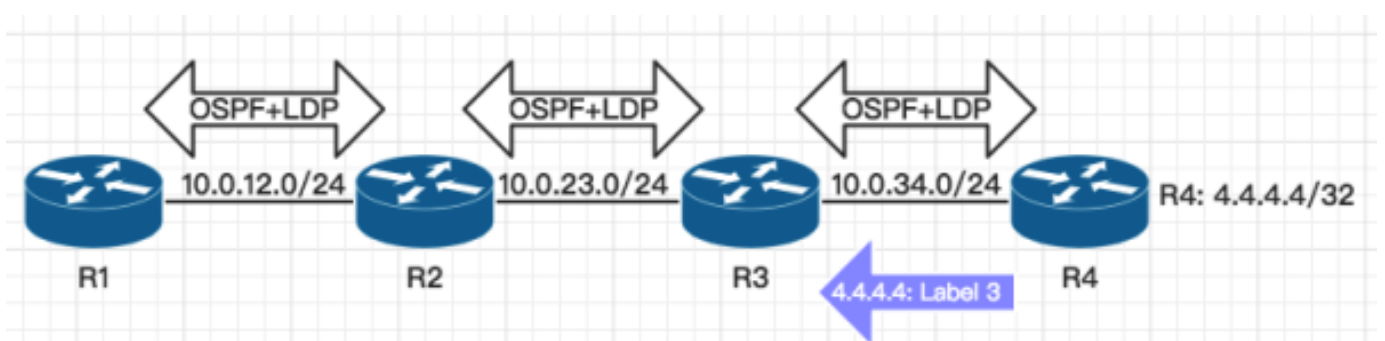
接下来，就像前面描述的一样，LSR会为每一条本地路由生成一个Local Label，因此每个路由器对4.4.4.4/32这条路由都有Local Label与之对应，也就是说每个LSR都有自己的prefix-label binding。

3.



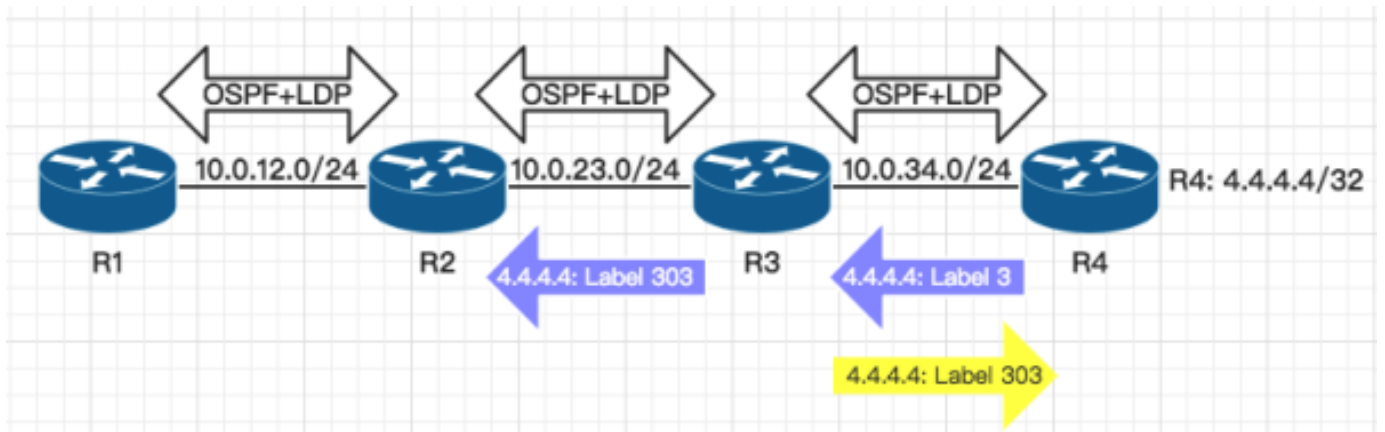
配置好4个路由器之间的LDP连接，LDP协议经过广播查找，获取LDP neighbor，建立LDP TCP session，接下来要将进行Label exchange。

4.



Label exchange并非是串行的过程，而是并行的同时在各个LSR之间发生。为了直观的描述，这里从R4开始说。R4将其Local Label向其唯一的LDP neighbor，也就是R3发送。

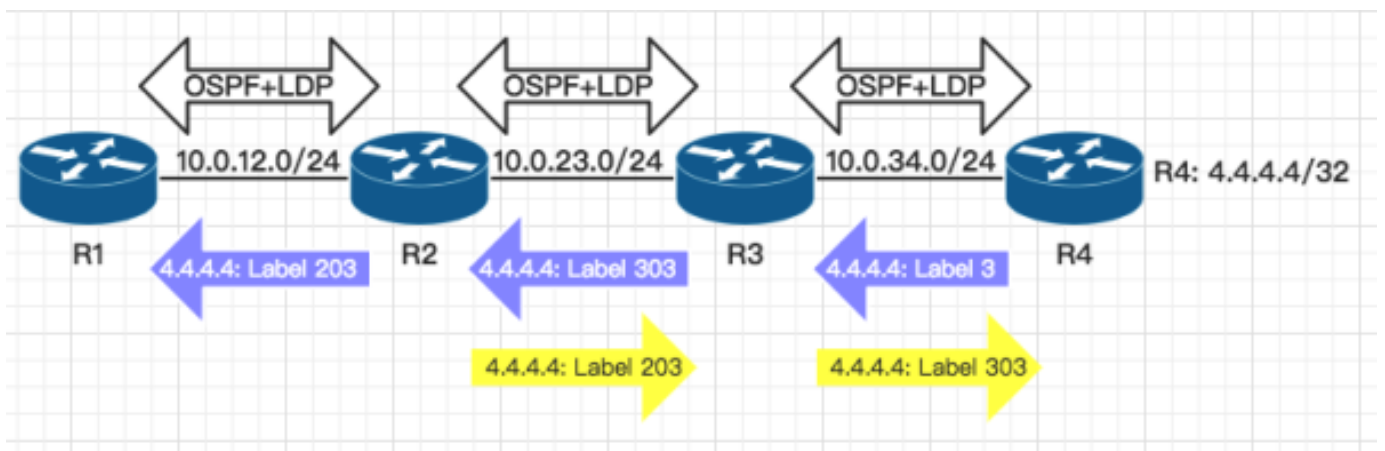
5.



目光转向R3，R3会将自身生成的对4.4.4.4/32的Local Label向其两个LDP neighbor发送。但是R3向R4发送的prefix-label binding并没有实际的意义，因为4.4.4.4/32是R4的直连路由，R4的Local Label优先级更高，因此，尽管R3向R4发送了prefix-label binding，但是不会实际写入NHLFE。

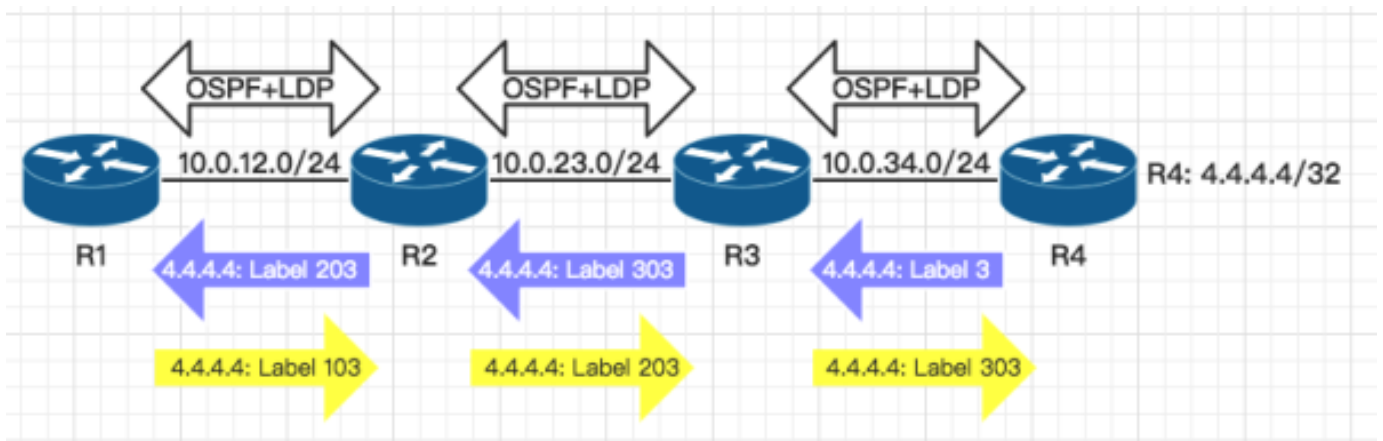
不过对于R2来说，R3发送的prefix-label binding是有意义的，会作为Remote Label写入R2的NHLFE。

6.



R2也会做同样的事情，但是同样的R2向R3发送的prefix-label binding并没有实际的意义，因为R3有更短的跳数到4.4.4.4/32的路由。从这里可以看出，与传统网络中的IGP，BGP类似，MPLS中的LDP，虽然会交换所有的Label信息，但是只有最优的才会写入相应的路由表中。而决定哪些Label会写入NHLFE，本质上还是由路由优先算法决定的。

7.



最后是R1，R1只有一个LDP neighbor，因此只会向R2发送没有意义的prefix-label binding。到此为止，R1到R4针对4.4.4.4/32的LSP已经建立。此时查看R2的MPLS转发表可以看到类似如下信息：

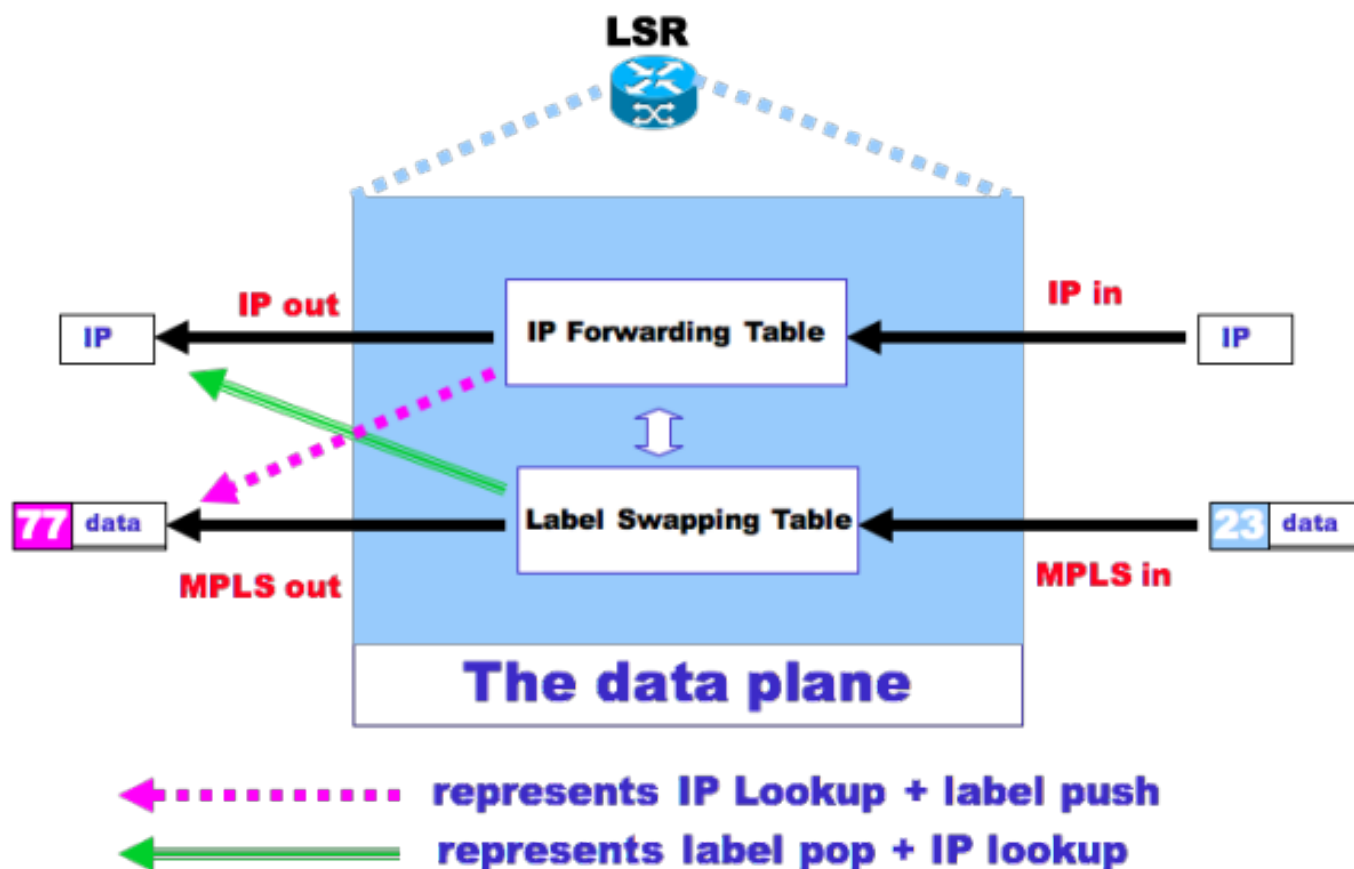
```
Local Outgoing Prefix Next Hop
203 303 4.4.4.4/32 10.0.23.3
```

这条NHLFE表明如果收到了Label是203的MPLS packet，会SWAP其Label到303，再发送到R3。其他路由器的MPLS转发表有类似条目。具体的转发流程上一篇的最后一部分描述过，这里就不再重复了。

LSP的建立有两点需要注意：

- LSP的建立，是从终点向起点建立，也就是与IP packet的走向相反。
- LSP是单向的，如果需要建立双向的通讯，必须有两个LSP。

对于R1，因为只有一个LSR，也就是R2与之相连，实际上R1不会收到Label是103的MPLS packet，没有LSR会为其生成Label 103。R1一般是ingress LER，因此R1通常会收到IP packet。作为LER，R1会识别其目的地址，并为IP packet打上Label 203，再发送到R2。对于LSR来说，不管是基于Label的转发，还是传统的路由转发，都需要传统路由表和MPLS路由表配合才能完成。例如这里说的R1。



从这里就可以看出，MPLS虽然是一套新的转发机制，但是很多地方都与传统的路由类似，很多地方需要借助传统路由才能完成。这不是一个真正的新世界。

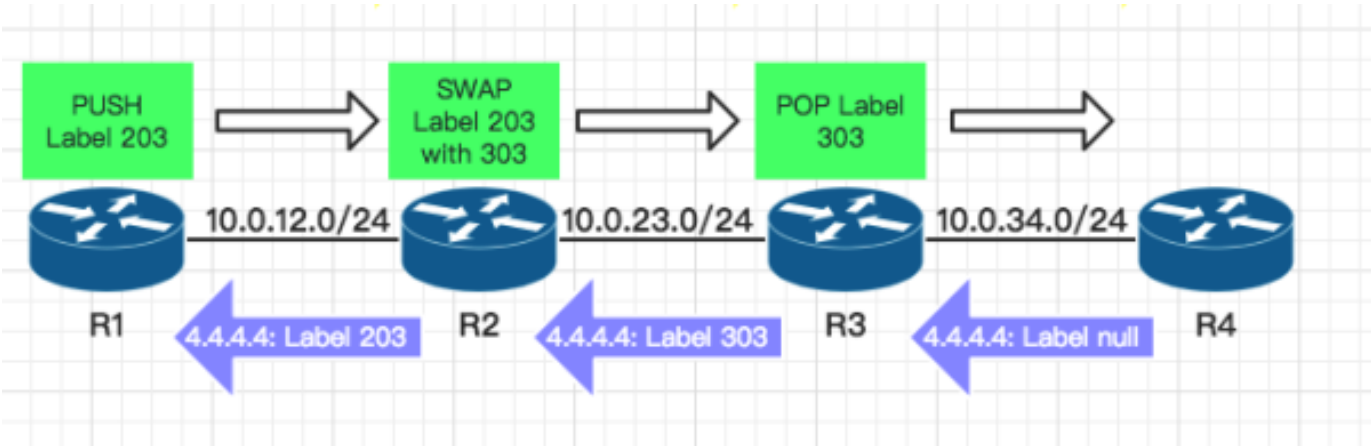
Penultimate Hop Popping

最后来看上面的R4，当R4收到了Label是3的MPLS packet，它需要做两件事：

1. 将Label 3从MPLS packet中去除，这样就获得了一个IP packet
2. 在本地的IP 路由中查找4.4.4.4/32，发现这是直连的路由，将IP packet发送出去。

这里实际上有可以改进空间的。如果R4对4.4.4.4/32生成的Local Label是一个特殊的Label，而R3在收到R4的这条特殊的Label时，就知道，当操作Label 303的MPLS packet时，只需要直接POP MPLS Label，然后转发到R4。这样R4收到的直接就是IP packet，而R4只需要做IP路由的查找就可以将IP packet转发出去。

这条特殊的Label一般称为implicit null。Penultimate Hop Popping的过程如下图所示：



在这个过程中，除了R4，各个路由器需要做的操作数不变。而R4需要做的操作变少了，不再需要对MPLS packet进行处理。

这就是Penultimate Hop Poping，Penultimate是倒数第二个的意思。Penultimate Hop Poping的意思就是在LSP的倒数第二个路由器，完成MPLS Label的POP操作，LSP的最后一个路由器只需要做IP路由转发就行。在实际的LSP中，Penultimate Hop Poping 常用来提高MPLS的工作效率。

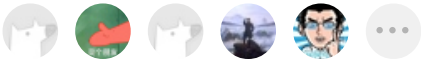
至此，不仅对MPLS的control plane描述完了，而且也描述了MPLS的unicast过程。前一篇说过，MPLS是一个很大的话题，unicast是其中的一个方面，以后有机会讲讲MPLS L3 VPN。

计算机网络


Multiprotocol Label Switching(MPLS)

☆ 收藏 ↗ 分享 ⚠ 举报

 14



4 条评论



写下你的评论...

事大夫