

# MPLS TE

Version 1.5

2013-07-26

文档作者： 红茶三杯

文档备注： 文档将不定期更新，请关注 <http://ccietea.com> 以便获得最新版本的文档。

版权信息： 红茶三杯( <http://weibo.com/vinsoney> )原创技术文档，供广大网友学习交流使用，可随意转载，  
转载请保留原作者信息。

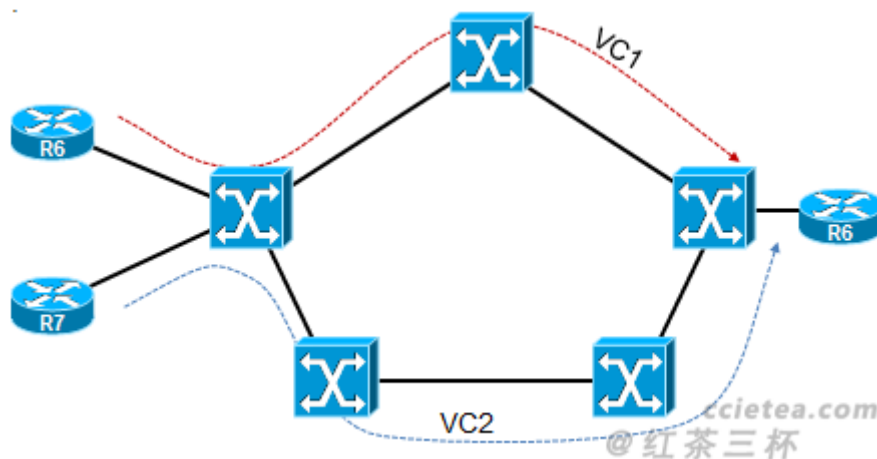
# 1 MPLS TE 概述

## 1.1 TE 概述

**流量工程 (Traffic engineering)** 我们可以笼统的理解为“驾驭流量穿越网络的能力”，我们能够通过 TE 的部署让流量以最优的方式在网络中从一个节点传输到另一个节点。

传统的 TE 解决方案有如下这么一些示例：

- TE with Layer3
  - 通过动态路由协议控制 metric 来影响路由选路
  - 源路由、策略路由、静态路由等等
- TE with Layer2 overlay
  - 例如帧中继环境下



## 1.2 MPLS TE

**传统 IP 路由的尴尬：**

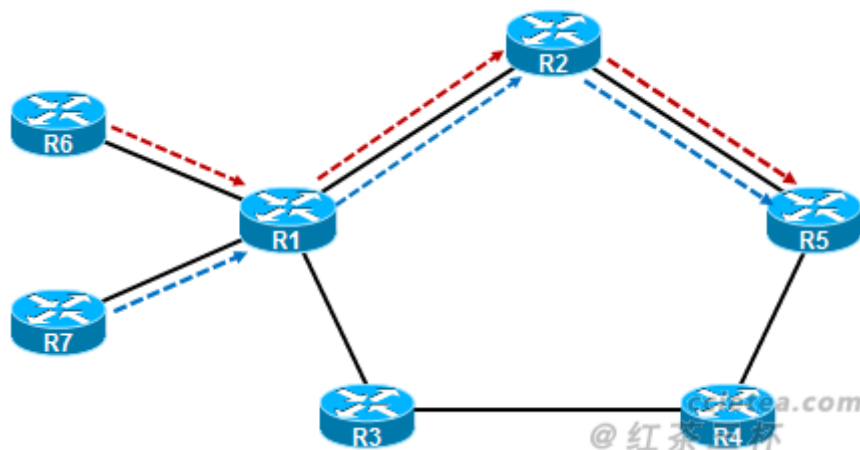
- IP 网络中的路由协议根据最小 metric 原则进行数据包的转发。
- IP 报文在每一跳路由器中的转发都是仅仅根据目的地址进行。
- 每一个被转发到某路由器的 IP 报文在经过该路由器之前及被转发之后都是相同的。

- IP 转发实例并不考虑链路的可用带宽和负载能力，有可能链路所分配的 metric 与实际的度量值并不一样。
- 这种 IP 报文的转发行为可能会导致某些网络中链路过度使用，而另一些链路使用概率却很小。
- 基于 IP 的流量工程是无连接的，不能实现显式路径（explicit routing）

**TE 可以提供一种解决方案，通过对流量，或者部分流量的控制来避免链路过载的情况发生。**

**下面我们来做个对比：**

## 1. IP 流量工程的状况

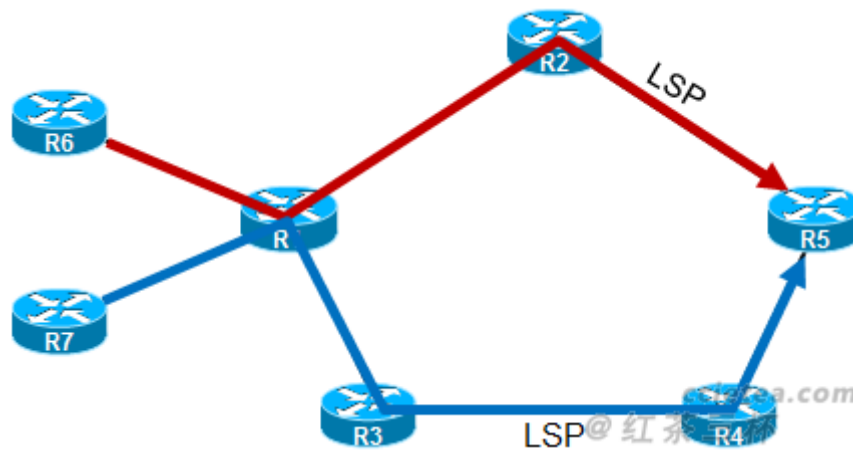


像上图，就非常容易出现问题流量走单边，R1-R3-R4-R5 链路却没有承载流量的情况，也许可以通过调整诸如 cost 值等手段使得下面的链路也能使用，但是这种方法有可能在调整的同时却给其他的流量带来问题。总的来说，IP 流量工程对流量的引导决策还是比较单一的，无非是像 ospf 协议通过操控 cost 或者利用协议本身的特性来操控路由，或者像 BGP 那样操控路径属性，又或者通过策略路由来操控。

## 2. MPLS TE 的状况

**MPLS TE 提供了如下解决方案：**

- MPLS TE 提高了流量在网络中扩散的效率，避免了链路的使用不充分和使用过度
- MPLS TE 考虑了配置（静态）在链路上的带宽
- MPLS TE 考虑了链路的属性参数（如延迟、抖动等）
- MPLS TE 可以通过自适应来改变链路的带宽和属性参数
- 使用流量工程的负载使用的是基于源的路由，而非基于目的 IP 地址的路由



MPLS TE 实现的机制中，一条 LSP 的首端路由器能够了解到网络的拓扑，并可以通过计算得出穿越网络到达该 LSP 尾端路由器的最优路由（不仅仅考虑 metric，还考虑带宽、链路属性等元素）。

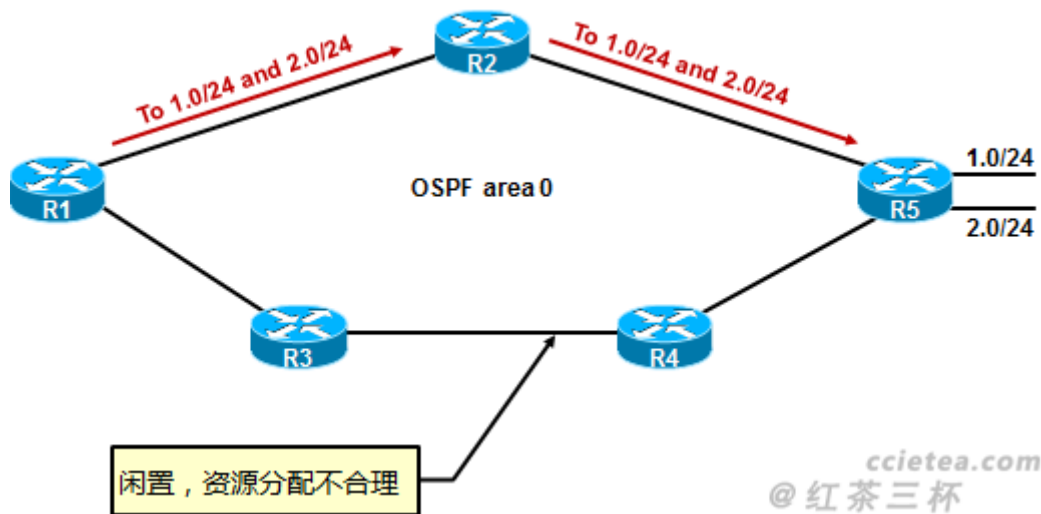
在运行了 MPLS 的网络中，可以将这两条路径设置为不同的 LSP，使用不同的标签。在 R1 上，R6 和 R7 发送给它的标签数据通过不同的入站标签值说明了数据是属于 R6 还是 R7。LSP 中端点 LSR 需要了解链路的带宽以及其他参数。所以 MPLS TE 端点（首、尾端 LSR）的路由协议必须是链路状态路由协议。通过链路状态协议，每个区域内的路由器都对本区域的拓扑有全面的了解。这样一来首端 LSR 就知道如何安排使用基于 MPLS 流量工程的 LSP 了。

这里可以使用基于源的路由。这条 LSP 被称为 MPLS TE 隧道。它是单向的。并且隧道的配置只需要在首端 LSR 上进行就可以了。

## 1.3 MPLS TE tunnel

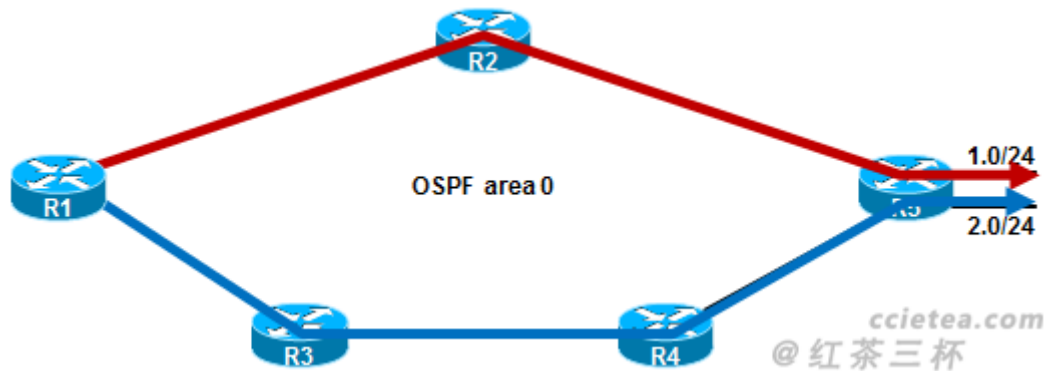
**MPLS TE tunnel 也叫做 Traffic trucks，用于克服 hop-by-hop IP 路由的缺陷**

- **TE tunnels are an aggregation of traffic flows of the same class (bandwidth , etc.) which are placed inside a common MPLS LSP**
- **TE tunnels are distinct from MPLS LSP through which it traverses:**
  - In operational contexts , a tunnel can be removed from one path and palced onto another.
- **Attributes are explicitly assigned to the TE tunnel through administration action.**
  - Its ingress and exgress LSRs
  - Bandwidth , latency , policy constrains
  - Class of data – The FEC which is mapped onto it



传统的 IP 路由在进行路由选择的时候，只会单纯的考虑 metric，例如如果上图中，我们运行的是 OSPF，那么最终 R1 去往 1.0/24 及 2.0/24 都会优选从 R2 到 R3 这条路径，因为 metric 小。那么这就直接造成了另一条可选路径的闲置以及主走路径的过度拥塞。

那么我们可以利用 TE tunnel 思维，在 R1 上建立两条 TE tunnel，如下图：



这样一来，我的网络资源就得到了更合理的利用，当 R1 去往 1.0/24 及 2.0/24 的时候，会将流量在 R1 上就送入不同的 tunnel。此刻 R1 是我 tunnel 的首端，R5 是尾端。Tunnel 可以设置带宽要求，以便我这条 tunnel 在网络中进行更加科学和合理的选路，而不仅仅是使用 OSPF metric 来进行选路。Tunnel 路径的计算是在 R1 也就是 tunnel 首端路由器上完成了，为了让首端路由器能够更加合理的进行 tunnel 路径的计算（不仅仅使用 metric，还能用可用带宽、链路属性等元素参与路径计算），我们就要求对网络中运行的链路状态路由协议进行扩展，以便泛洪更多路径计算所需的信息，扩展的 OSPF 和 IS-IS 都能完成此重任。

## 1.4 MPLS TE 组件

### 链路的限制

每条链路所能支持的最大流量以及链路所能使用的TE隧道

### TE信息分发

通过启用了MPLS TE的链路状态路由协议

### 路径计算CSPF

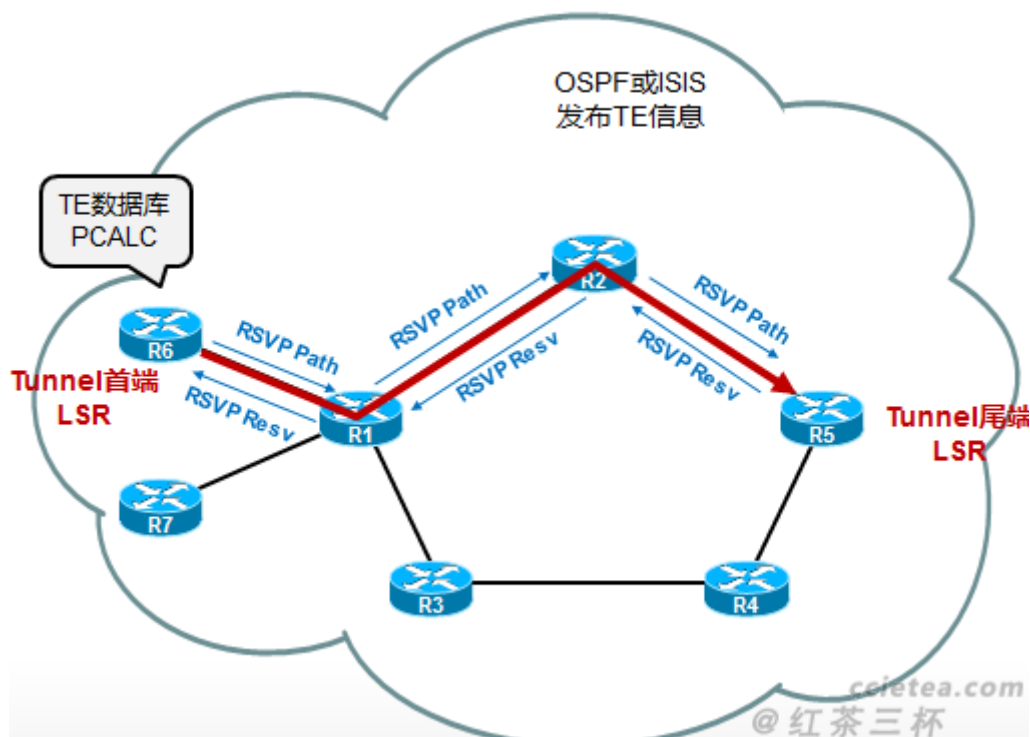
一种用来计算从前端LSR到尾端LSR的最优路径的算法

### RSVP

一种用来在穿越网络的TE隧道中发送信号的信令协议

### Static、AutoRoute、PBR

一种将流量转发至TE隧道的方法

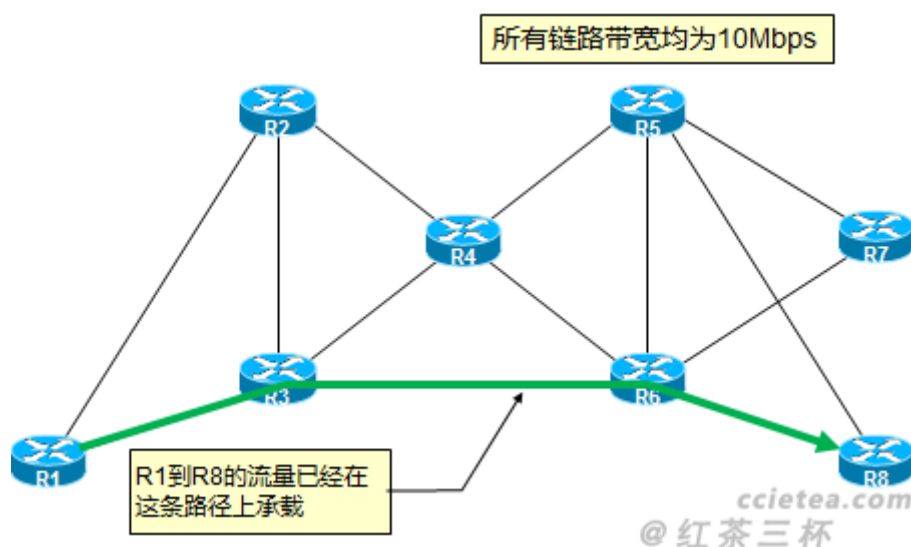


首先我们需要一种能通告各种网络资源（权重、IGP metric、可用带宽、链路属性等）的路由协议，这个路由协议必须是链路状态路由协议，而且必须支持 MPLS TE 扩展，我们有两个可选，一是 OSPF 另一个是 IS-IS。在 CISCO IOS 内部会根据“**针对 MPLS TE 进行扩展的链路状态路由协议**”发送的 TE 信息构建一个 **TE 数据库**，这个数据库包含了所有启用了 MPLS TE 的链路以及这些链路的特征或者参数。可用于 MPLS TE 的带宽以及链路属性等参数在网络中的所有链路都是可配置的。

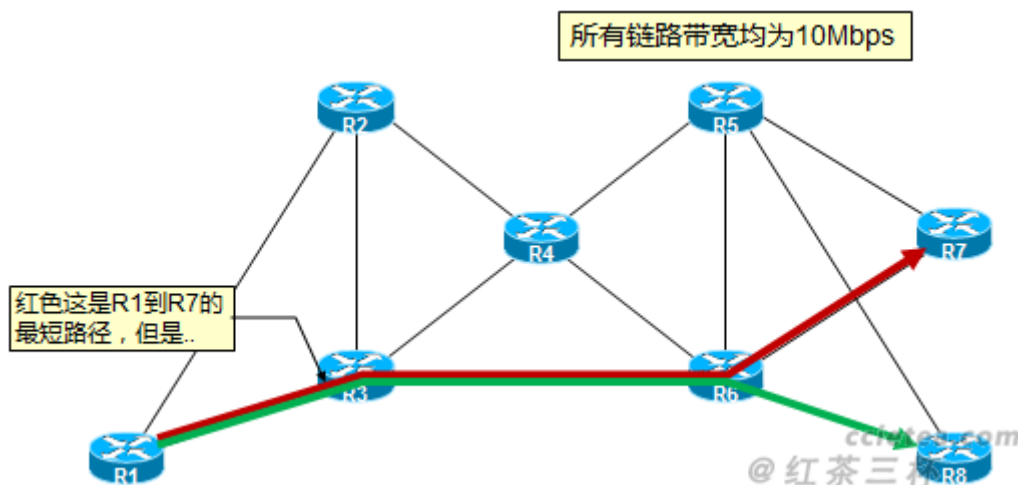
当你在台 LSR 上配置一条 **TE Tunnel** 的时候，该 LSR 就变成了这条 TE tunnel 的首端 LSR。接下来你就可以指定 TE Tunnel 的目的 LSR，以及它必须遵守的限制例如隧道的保证带宽等等，这些相关参数在首端 LSR 的 Tunnel 口中进行配置。接下去在首端 LSR 上，路径计算 PCALC 或者限制的 SPF（CSPF）算法可以根据上面我们提到的 MPLS TE 数据库结合 Tunnel 的要求计算出最短的、满足所有条件的、从首端 LSR 到尾端 LSR 的路径。完成这些工作后的输出，是一个从 tunnel 头端到尾端的一个 IP 地址序列，指引出**一条路径**。PCALC 和 CSPF 是用于 MPLS TE 的一种 SPF 算法的变体。

前面 CSPF 计算出来的一个结果（一条路径），会输送到另一个模块：**RSVP**。MPLS TE 的流量的靠标签来转发的，这个标签是靠 RSVP 分发的，另外，资源的预留也是通过 RSVP 来完成。LSP 的链路中 LSR 需要了解用于特定 LSP 的 TE 隧道的入站和出站标签。链路中的 LSR 只能通过在首端路由器和链路中的 LSR 之间的信令协议来学习标签。CISCO 使用 TE 扩展的 RSVP 来为 MPLS TE 隧道传递信令。RSVP 的扩展使得 RSVP 可以承载 MPLS 标签信息和其他 TE 特定的参数。本质上说，RSVP 尝试沿着从首端 LSR 到尾端 LSR 的路径传递信令，而这个路径是根据首端 LSR 的 TE 数据库计算得来的。RSVP 的另一个非常重要的功能是建立及维护资源预留（如带宽）。

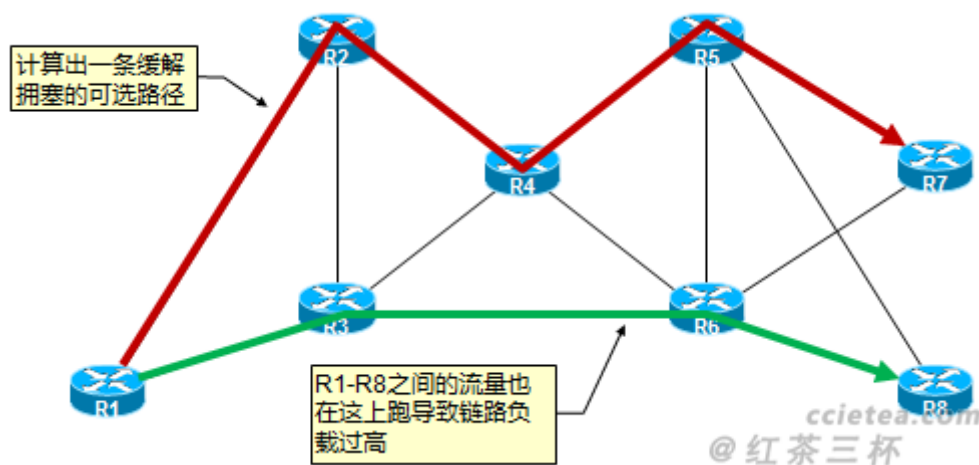
## • 工作过程简要介绍



初始环境如上图，R1 到 R8 的流量，已经在图中所示绿色的路径上被承载。

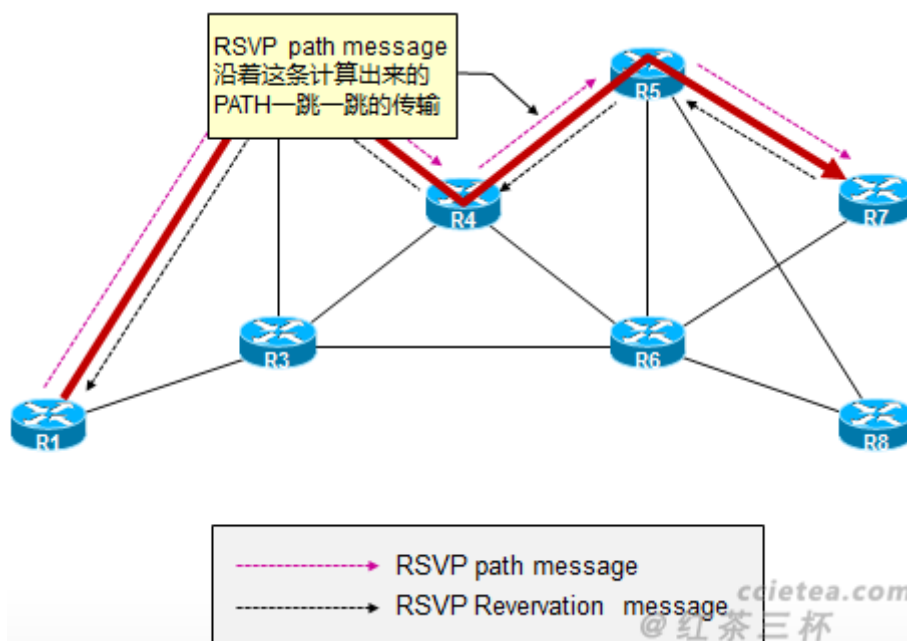


我们假设上述拓扑中，所有链路的带宽均为 10Mbps。那么运行 IGP 协议后，我们知道从 R1 到 R7，最短的路径就是 R1-R3-R6-R7，但是由于网络中 R1-R8 之间的流量已经在这个路径上运行了，并且占用了比较高的带宽，链路的负载过高了，传统的 IP 路由协议是没办法这么智能的考虑到这点，但是利用 MPLS TE，我们可以让 R1 到 R7 走一种更优的路径。。



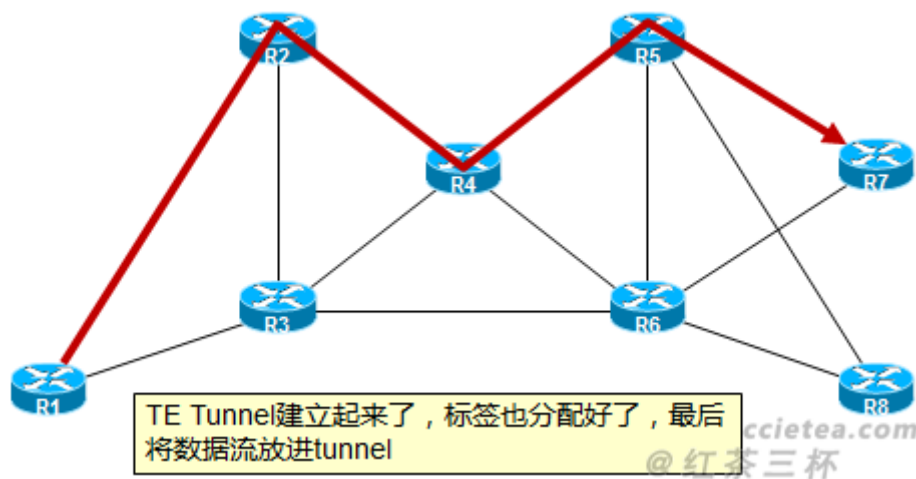
接下去，R1 通过已有的信息，计算出另一条到达 R7 的可选路径：R1-R2-R4-R5-R7，这条路径通过 CSPF 计算得出。上图中，红色的路径就是经过 CSPF 计算得出的。





计算出路径之后，需要通过 RSVP 来预留资源及分发标签，RSVP path message 沿着这条计算出来的 PATH 一跳一跳的传输并且请求资源及标签，然后 R7 会给应答同时将标签带回来，这个标签分发和资源预留的动作也是一跳一跳的进行。这样最终一条 LSP 就建立好了。

那么最后，就是将数据引入建立好的 tunnel 上。实际上，在 R1 上创建一条 TE Tunnel 后，在其本地一个 Tunnel 接口就会被创建。我们只要将流量放入 tunnel 口即可，方法还是有不少，我们在后续的内容中继续介绍。



## 2 MPLS TE 信息分发

### 2.1 概述

MPLS TE tunnel 的路径计算需要有相关的信息才能够进行。那么我们就需要使用一种链路状态路由协议来泛洪链路的信息到网络中所有运行了 TE 的路由器中。TE 路由器搜集这些信息后，建立自己的 MPLS TE Database。

**为了让网络设备进行更加智能化的路径计算，需要发布各种信息：**

- 链路状态信息            IGP 本身就支持
- TE Metric                相当于权重值的概念，注意这要和 IGP metric 区分开来
- 可用带宽                接口的可用带宽
- 隧道优先级
- 属性标记及亲和属性

**先做个简要的介绍：**

- TE metric 是一个用于构建与 IP 拓扑完全不同的 TE 拓扑的参数（虽然默认情况下与 IGP metric 相等）
- 最大带宽就是链路的全部带宽，在 CISCO IOS 中，这个值是匹配物理链路上的，或者是配置的带宽值
- 最大可预留带宽很明显是表示链路中可供 TE 使用的带宽，可以通过 ip rsvp bandwidth 命令配置
- 不可预留带宽是提供给 TE 后所剩下的带宽，也就是链路最大带宽减去目前被 TE 隧道所保留的带宽
- 属性标记是一个 32bits 的字段，这个在后面有详细的介绍

### 2.2 对 IGP 的要求

#### 2.2.1 针对 TE 的 OSPF 扩展

##### 1. 概述



- 在诸如 OSPF 这类 IGP 协议中，使用所配置的带宽或接口 cost 来计算最优路由
- MPLS TE 会将网络资源的实际使用情况纳入考虑，因此它需要更多的信息而不仅仅只是接口带宽或 cost。
- 由 tunnel 的头端进行 CSPF 计算，计算的结果是一串 IP（tunnel 的路径）
- 上述输出的结果供扩展的 RSVP 使用

## 2. 用于 MPLS TE 的 LSA

RFC2370 描述了 OSPF 的一种扩展，定义了 3 种新的 LSA。这些 LSA 称为**迟钝的 LSA (opaque LSA)**，他们的区别仅限于传播范围。这些 LSA 可以准确的将 MPLS TE 所需的信息提供给 OSPF：

- **类型 9**      泛洪范围只在本链路
- **类型 10**    泛洪范围只在本区域
- **类型 11**    泛洪范围是整个域（不会进入 stub 区域）

MPLS TE 使用类型 10 的 LSA 来为区域内的 MPLS TE 工作。

LS老化时间	option	9,10或11
Opaque type	选项ID	
通告的路由器		
LS序列号		
LS checksum	Length	
Opaque信息		
... ..		

ccietea.com  
@红茶三杯

ccietea.com  
@ 红茶三杯

```

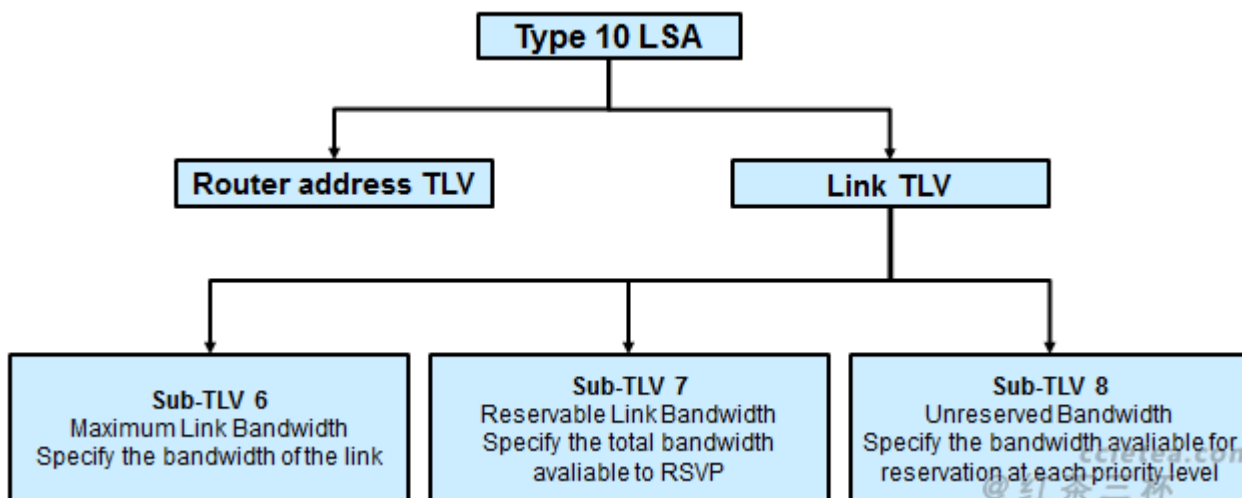
LS Type: Opaque LSA, Area-local scope
LS Age: 2 seconds
Do Not Age: False
+ Options: 0x20 (DC)
Link-State Advertisement Type: Opaque LSA, Area-local scope (10)
Link State ID Opaque Type: Traffic Engineering LSA (1)
Link State ID TE-LSA Reserved: 0
Link State ID TE-LSA Instance: 0
Advertising Router: 1.1.1.1 (1.1.1.1)
LS Sequence Number: 0x8000000a
LS Checksum: 0x5df4
Length: 132
+ MPLS Traffic Engineering LSA
+ Router Address: 1.1.1.1
  TLV Type: 1 - Router Address
  TLV Length: 4
  MPLS/TE Router ID: 1.1.1.1 (1.1.1.1)
+ Link Information
  TLV Type: 2 - Link Information
  TLV Length: 100
  + Link Type: 2 - Multi-access
  + Link ID: 10.1.12.2
  + Local Interface IP Address
  + Traffic Engineering Metric: 1
  + Maximum Bandwidth: 12500000 bytes/s (100000000 bits/s)
  + Maximum Reservable Bandwidth: 9375000 bytes/s (75000000 bits/s)
  + Unreserved Bandwidth
  + Resource Class/Color: 0x00000000
  + Unknown Link sub-TLV: 32770
  
```

type 10 Opaque LSA

Opaque信息

许多TLV, 使得OSPF可以通过一种伸缩性的方式运载数据

可以看到 Opaque 信息这里, 有许多 TLV ( 类型长度值 ), 这些 TLV 使得 OSPF 可以通过一种伸缩性的方式运载数据。这些 TLV 承载了特定的 MPLS TE 数据。一共有两种 TLV 分别是: **路由器地址 TLV** 和 **link TLV** ( 在上面抓包的结果中的 router address 和 link information 分别就是 )。



路由器地址 TLV 承载了用于 TE 的路由器 TE router ID。

而链路 TLV 则承载了一系列描述用于 MPLS TE 的单条链路的子 TLV，关于这些子 TLV 描述如下：

子TLV编号	名称	以8bits为单位的长度
1	链路类型	1
2	链路ID	4
3	本地接口IP地址	4
4	远程接口IP地址	4
5	流量工程度量值	4
6	最大带宽	4
7	最大预留带宽	4
8	未保留的带宽	32
9	Resource class/color	4

- 链路类型说明了链路是点到点还是多路访问链路
- 链路 ID 可以是邻居的 routerID，如果是多路访问链路，则为 DR 的接口地址
- 带宽参数是以 bytes 为单位
- 未保留带宽（不可预留带宽）参数有 32 个 8bits，其他的带宽参数只有 4 个 8bits，因为未保留带宽实际上是 8 个优先级每个优先级别又有 4 个字节。优先级的范围是 0-7。

### 3. 如何在 CISOC IOS 路由器上查看 opaque LSA-10 呢？

**R1#show ip ospf database opaque-area self-originate**

OSPF Router with ID (1.1.1.1) (Process ID 1)

Type-10 Opaque Link Area Link States (Area 0)

LS age: 29

Options: (No TOS-capability, DC)

LS Type: Opaque Area Link

Link State ID: 1.0.0.0

Opaque Type: 1

Opaque ID: 0

Advertising Router: 1.1.1.1

LS Seq Number: 80000001

Checksum: 0x6FEB

Length: 132

Fragment number : 0

MPLS TE router ID : 1.1.1.1

Link connected to Broadcast network

Link ID : 10.1.12.2

Interface Address : 10.1.12.1

Admin Metric : 1

**!! TE metric , 默认和下面的 IGP metric 相等**

Maximum bandwidth : 12500000

**!!单位是 Bytes , 乘以 8 也就是 100Mbps**

Maximum reservable bandwidth : 9375000

**!!最大可预留带宽是 75M , 也就是 100M 的 75%**

Number of Priority : 8

Priority 0 : 9375000      Priority 1 : 9375000

Priority 2 : 9375000      Priority 3 : 9375000

Priority 4 : 9375000      Priority 5 : 9375000

Priority 6 : 9375000      Priority 7 : 9375000

Affinity Bit : 0x0

IGP Metric : 1

**!!接口的 OSPF cost**

Number of Links : 1

#### 4. O 比特

DN	O	DC	L	N/P	MC	E	•
----	---	----	---	-----	----	---	---

O 比特被定义使用在 OSPF 的 option 字段中, 用来说明路由器是否有能力发送和接收 opaque LSA。

当我们部署 MPLS TE 且使用 OSPF 作为 TE 的时候, OSPF 就需要扩展以便支持 MPLS TE, 这时候就可以看到 Obit 位的置位了, 而且只能够在这种环境下的 DBD 报文中看到相应的 option 的 obit 置位, 而其他报文中的 option 里 obit 仍然为 0。

在 RFC5250 中对这里做了描述: A neighbor is opaque-capable if and only if it sets the O-bit in the Options field of its Database Description packets; the O-bit SHOULD NOT be set and MUST be ignored when

received in packets other than Database Description packets.

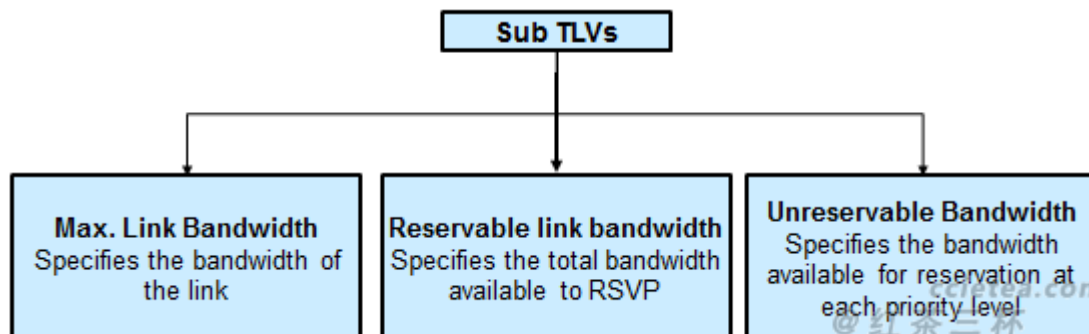
## 2.2.2 针对 TE 的 ISIS 扩展

### 1. 针对 TS 的扩展 IS-IS 新增了三种类型的 TLS :

- **TLV 22 : Extended intermediate-System Reachability**

Ability to send detail link information , including bandwidth . Replaces IS Neighbor TLV

TLV 22 可以包含 7 个子 TLVs , 其中三个包含用于 TE 的带宽信息



- **TLV134 : TE RouterID**

设备的 TE RouterID

- **TLV135 : Extended IP Rechability**

Replaces both IP Reachability TLVs ( 128 and 130 ) . Uses wide metrics

### 2. 查看报文

```

ISO 10589 ISIS InTRA Domain Routeing Information Exchange Protocol
  Intra Domain Routing Protocol Discriminator: ISIS (0x83)
  PDU Header Length: 27
  Version (==1): 1
  System ID Length: 0
  PDU Type          : L1 LSP (R:000)
  Version2 (==1): 1
  Reserved (==0): 0
  Max.AREAS: (0==3): 0
☐ ISO 10589 ISIS Link State Protocol Data Unit
  PDU length: 147
  Remaining lifetime: 1197
  LSP-ID: 0000.0000.0001.00-00
  Sequence number: 0x00000008
  + Checksum: 0x916f [correct]
  + Type block(0x01): Partition Repair:0, Attached bits:0, overload bit:0, IS type:1
  + Area address(es) (4)
  + Protocols supported (1)
  + Hostname (2)
☐ Traffic Engineering Router ID (4) TLV134
  Traffic Engineering Router ID: 1.1.1.1 (1.1.1.1)
  + IP Interface address(es) (4)
☐ Extended IP Reachability (17) TLV135
  + IPv4 prefix: 10.1.12.0/24, Metric: 10, Distribution: up, no sub-TLVs present
  + IPv4 prefix: 1.1.1.1/32, Metric: 10, Distribution: up, no sub-TLVs present
☐ Extended IS reachability (74) TLV22
☐ IS neighbor: 0000.0000.0002.01
  Metric: 10
  Administrative group(s):
  IPv4 interface address: 10.1.12.1
  Maximum link bandwidth : 100.00 Mbps
  Reservable link bandwidth: 75.00 Mbps
  + Unreserved bandwidth:
    Traffic engineering default metric: 1
  
```

## 2.2.3 IGP 泛洪

IGP 在下列情况发生时泛洪 TE 信息：

- 链路状态发生变化
- 配置变化
- 周期性泛洪
- 预留带宽发生变化（重大变化）
- 在隧道设置失败之后



下面将分别对上述的情况做分析：

1. 和常规的 IP 路由一样，接口的 up down 或者手工修改了针对 IGP 的接口参数的时候， OSPF 将泛洪 LSA 或者 ISIS 会泛洪 LSP。

2. OSPF 和 ISIS 也有周期性泛洪的机制

OSPF 默认的泛洪周期是 30 分钟，可以通过 timers pacing lsa-group 修改

ISIS 默认的泛洪周期是 15 分钟，可以通过 lsp-refresh-interval 修改

3. 带宽的变化

默认的变化阈值（带宽占用率）：

Up Thresholds            15 30 45 60 75 80 85 90 95 96 97 98 99 100

Down Thresholds:        100 99 98 97 96 95 90 85 80 75 60 45 30 15

```
router(config)#interface fast0/0
```

```
router(config-if)#mpls traffic-eng flooding thresholds up/down ?
```

修改带宽 UP/DOWN 阈值

```
router# show mpls traffic-eng link-management bandwidth-allocation
```

查看带宽的 Thresholds

**在通过 IGP 协议（OSPF 或 ISIS）泛洪相关信息后，所有的 TE 路由器都形成区域内统一的 TEDB  
TEDB 与 IGP 的链路状态数据库是独立的。**

## 2.3 信息详解

### 2.3.1 带宽信息

MPLS TE 需要通告的带宽信息有：

- 最大接口可用带宽：  
可使用默认或者用这条命令修改 Router(config-if)# bandwidth ?
- 最大可预留带宽信息：

Router(config-if)# ip rsvp bandwidth ?

这个带宽是给 TE tunnel 预留的，如果命令后面不加带宽值，则缺省为接口带宽的 75%

## 2.3.2 隧道优先级

### 1. 我们可以定义 TE tunnel 的优先级

- 优先级范围 0-7，越小越优先
- 隧道优先级有两种类型：
  - 建立优先级 setup priority
  - 保持优先级 hold priority
- 高优先级的隧道可以抢占低优先级隧道的资源。支持 0-7 共 8 个隧道优先级别
- 设置命令：

```
router(config-if)#tunnel mpls traffic-eng priority 6 6
```

第一个 6 是 setup priority，第二个 6 是 hold priority。

如果一个 tunnel1 的建立优先级，比 tunnel2 的保持优先级高，那么 tunnel1 就能抢占 tunnel2 不能将 tunnel 的建立优先级设置得比保持优先级高。

当然，我们在部署的时候，强烈建议同一个隧道的建立优先级和保持优先级要相等

### 2. 每个优先级都对应一个可用带宽

支持 0-7 共 8 个隧道优先级别

接口下根据各自优先级会列出当前的最大可用带宽

Show mpls traffic-eng topology ?

Show mpls traffic-eng link-management advertisement

Show mpls traffic-eng link-management bandwidth-allocation

### 3. 关于隧道的优先级及当前可用带宽的验证：

首先 tunnel 的配置如下：

```
interface Tunnel0
 ip unnumbered Loopback0
 tunnel destination 4.4.4.4
 tunnel mode mpls traffic-eng
```

**tunnel mpls traffic-eng priority 6 6**

tunnel mpls traffic-eng path-option 10 dynamic

假设本地物理出接口的带宽为 100M，那么 rsvp 默认的可预留带宽是 75%，这里就是 75M

所以我们得到如下的查看信息：

**R1#sh mpls traffic-eng topology 1.1.1.1 (1.1.1.1 是本地路由器)**

IGP Id: 1.1.1.1, MPLS TE Id:1.1.1.1 Router Node id 1

link[0]:Nbr IGP Id: 10.1.12.2, nbr\_node\_id:8, gen:47

frag\_id 0, Intf Address:10.1.12.1

TE metric:1, IGP metric:1, attribute\_flags:0x0

physical\_bw: 100000 (kbps), max\_reservable\_bw\_global: 75000 (kbps)

max\_reservable\_bw\_sub: 0 (kbps)

	Total Allocated	Global Pool	Sub Pool
	BW (kbps)	Reservable	Reservable
	-----	-----	-----
bw[0]:	0	75000	0
bw[1]:	0	75000	0
bw[2]:	0	75000	0
bw[3]:	0	75000	0
bw[4]:	0	75000	0
bw[5]:	0	75000	0
bw[6]:	0	75000	0
bw[7]:	0	75000	0

上面我们可以看到所有隧道优先级的可预留带宽。

现在我们来修改优先级为 6 的隧道的可预留带宽。

Interface Tunnel0

ip unnumbered Loopback0

tunnel destination 4.4.4.4

tunnel mode mpls traffic-eng

**tunnel mpls traffic-eng priority 6 6**

**tunnel mpls traffic-eng bandwidth 33000 !!改为 33M**

```
tunnel mpls traffic-eng path-option 10 dynamic
```

#### R1#sh ip rsvp interface

interface	allocated	i/f max	flow	max sub	max
Fa0/0	33M	75M	75M	0	!! 可以看到，这里 f0/0 口已经被分掉了 33M

#### R1#sh mpls traffic-eng topology 1.1.1.1

IGP Id: 1.1.1.1, MPLS TE Id:1.1.1.1 Router Node id 1

link[0]:Nbr IGP Id: 10.1.12.2, nbr\_node\_id:8, gen:51

frag\_id 0, Intf Address:10.1.12.1

TE metric:1, IGP metric:1, attribute\_flags:0x0

physical\_bw: 100000 (kbps), max\_reservable\_bw\_global: 75000 (kbps)

max\_reservable\_bw\_sub: 0 (kbps)

	Total Allocated BW (kbps)	Global Pool Reservable BW (kbps)	Sub Pool Reservable BW (kbps)
	-----	-----	-----
bw[0]:	0	75000	0
bw[1]:	0	75000	0
bw[2]:	0	75000	0
bw[3]:	0	75000	0
bw[4]:	0	75000	0
bw[5]:	0	75000	0
<b>bw[6]:</b>	<b>33000</b>	<b>42000</b>	<b>0</b>
<b>bw[7]:</b>	<b>0</b>	<b>42000</b>	<b>0</b>

可以看到，我们为优先级为 6 的隧道分配了 33M 的带宽，原先接口的总可预留带宽为 75M，分了 33M 出去，所以还有 42M，我们也可以看到，因为我们修改了优先级为 6 的隧道的可预留带宽，因此优先级比其低的隧道，可预留的带宽也变了。

### 2.3.3 属性标记和亲和属性

## 1. 概念简介

- **属性标记 attribute-flags**

用来描述物理链路的属性，由 32bits 组成，每一位都可以单独表示链路的一个属性（如是否加密等）或者管理型的策略。这 32bits 没有特定的语法，每一个 bit 都可以进行设置后者不去动他，网络设计者可以认为的根据需要为特定的 bit 指定特定的意义。

在物理接口上配置：

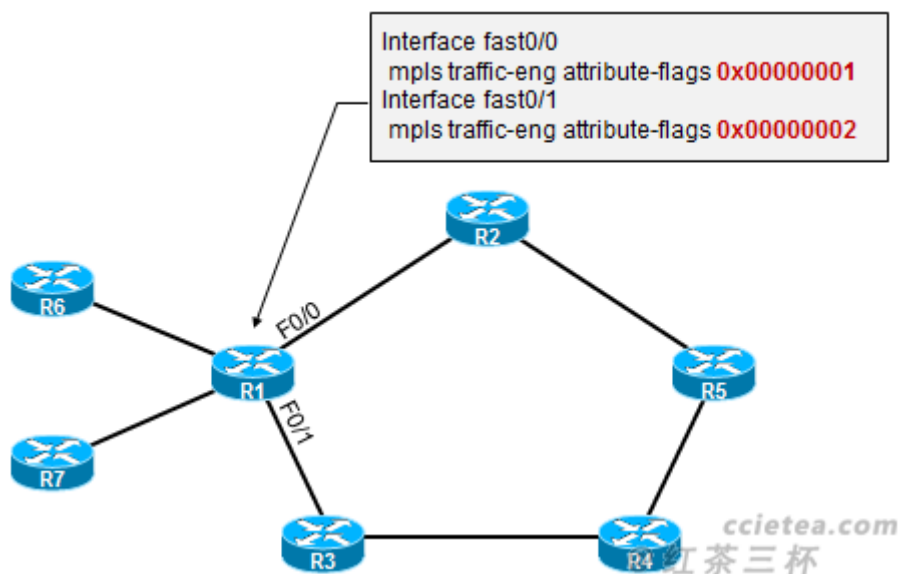
```
Router(config-if)# mpls traffic-eng attribute-flags ?
<0x0-0xFFFFFFFF> Attribute flags
```

- **亲和属性 Affinity Attributes 及掩码**

亲和属性是描述 MPLS TE tunnel 的属性，也由 32bits 组成，正好与管理上面所说的 attribute-flags 位数相同。与亲和属性同时被配置的还有一个掩码，也是 32bits，用来给属性标记和亲和属性做匹配动作。

默认是 0x0 / 0xFFFF（默认掩码是全 1，所以所有的 bits 都无所谓）

## 2. 概念详解



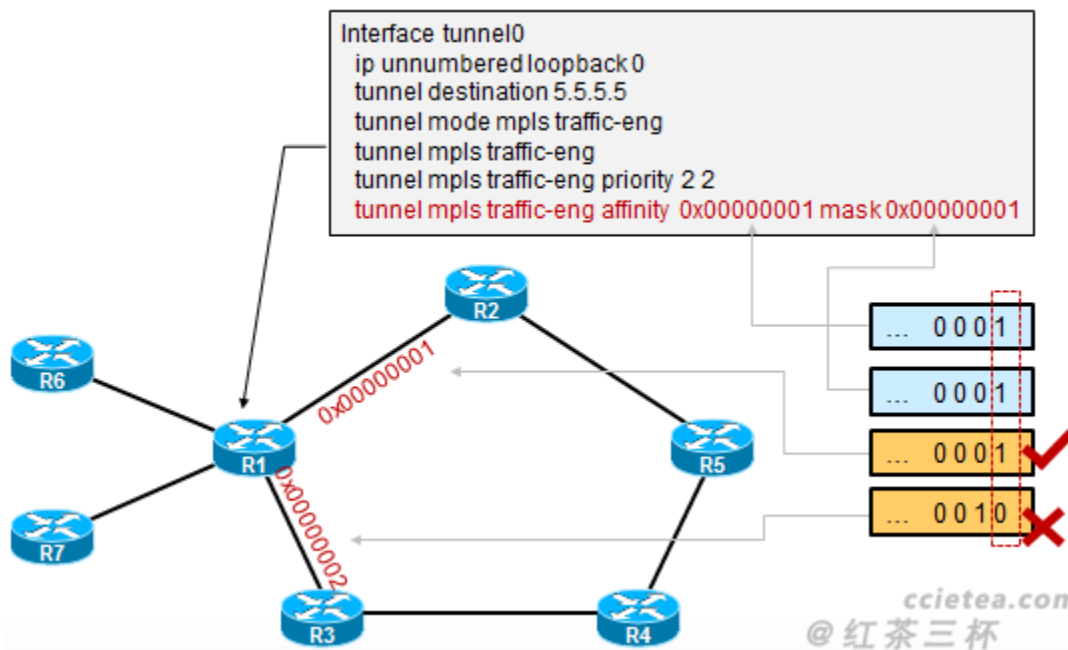
上图中，我们给 R1 的两个接口分别定义了属性标记，属性标记 attribute-flags 是一个 32bits 的字段，用 16 进制形式输入和体现。使用的命令如图中所示，命令输入后，将触发 R1 发送 Opaque LSA，并且通告其链路的这个管理属性，LSA 相关内容如下（注意，下面的报文截图与上述网络环境无关，仅仅为解释 attribute-flags 在报文中的体现）：

- ▣ MPLS Traffic Engineering LSA
  - ▣ Link Information
    - TLV Type: 2 - Link Information
    - TLV Length: 100
    - ⊕ Link Type: 2 - Multi-access
    - ⊕ Link ID: 10.1.23.2
    - ⊕ Local Interface IP Address
    - ⊕ Traffic Engineering Metric: 1
    - ⊕ Maximum Bandwidth: 12500000 bytes/s (100000000 bits/s)
    - ⊕ Maximum Reservable Bandwidth: 9375000 bytes/s (75000000 bits/s)
    - ⊕ Unreserved Bandwidth
    - ▣ Resource Class/Color: 0x00000002
      - TLV Type: 9: Resource Class/Color
      - TLV Length: 4
      - ▣ MPLS/TE Link Resource Class/Color: 0x00000002
        - Group 1
    - ⊕ Unknown Link sub-TLV: 32770

ccietea.com

@ 红茶三杯

接下去，我们在 R1 的 tunnel 中做如下配置：



我们定义了 R1 这个 TE tunnel 的亲和属性及掩码，那么这样一来，R1 在为这条 tunnel 计算路径的时候，R1 将 affinity 值 0x00000001 转换成二进制，与 R1 两个接口上配置好的这两个 attribute-flags 值进行——比对，由于我们掩码配置的是 0x00000001，将其也写成二进制，掩码为 1 的位，就是必须匹配的，因此仅有 R2-R2 这段链路可用。在实际的网络部署中，我们可以根据自己的需求灵活的定义链路属性每一个 bit 的含义，结合亲和属性就可以进行路径选路的把控和规避等等。

## 2.3.4 TE Metric : Administrative weight

### 1. 概念简介

- 缺省情况下用 TE metric 作为 TE tunnel 最短路径的计算依据

```
router(config-if)#mpls traffic-eng administrative-weight ?
```

- 可以用上述命令修改 TE metric ( 在物理接口上配置 )
- 缺省情况下 TE metric 等于 IGP metric ( 这句话的意思是 , 在没有使用上面的命令在物理接口中配置的情况下 , TE metric 等于 IGP metric )

- 可以修改作为 TE tunnel 最短路径的计算依据

```
router(config)#tunnel mpls traffic-eng path-selection metric ?
```

```
igp use IGP metric
```

```
te use TE metric 缺省是这个
```

上述命令 , 也可在 tunnel 口中配置

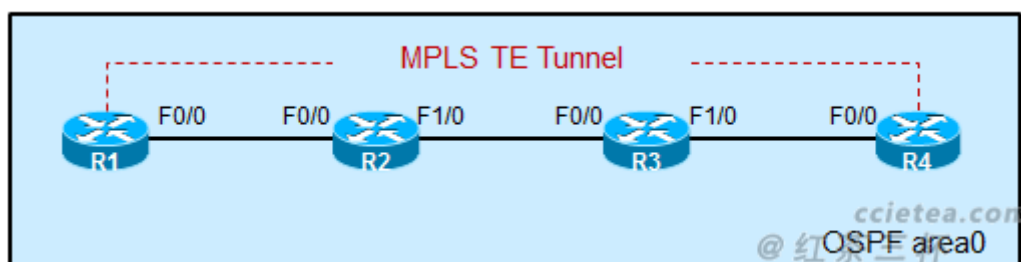
如果将 Path-selection metric 改为 igp , 那么 tunnel 的最短路径将忽略物理接口中配置的 administrator-weight , 而只采用 IGP metric 来计算 tunnel 的最短路径。

### 2. 验证及查看

```
router#show mpls traffic-eng topology
```

```
router#show mpls traffic-eng tunnels
```

使用上述命令查看 TE metric



例如在上述拓扑中 , R1 建立了一条到 R4 的 TE tunnel , 所有路由器都宣告自己的 loopback 进 OSPF。所有的接口 cost 都为 1。

```
R1#show mpls traffic-eng tunnels
```

```

Name: R1_t0                                (Tunnel0) Destination: 4.4.4.4
Status:
  Admin: up      Oper: up      Path: valid      Signalling: connected
  path option 10, type dynamic (Basis for Setup, path weight 3)

Config Parameters:
  Bandwidth: 33000      kbps (Global)  Priority: 6 6  Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: disabled  LockDown: disabled  Loadshare: 33000      bw-based
  auto-bw: disabled

InLabel  : -
OutLabel : FastEthernet0/0, 200
RSVP Signalling Info:
  Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 0, Tun_Instance 163
RSVP Path Info:
  My Address: 10.1.12.1
  Explicit Route: 10.1.12.2  10.1.23.2  10.1.23.3  10.1.34.3
                  10.1.34.  4 4.4.4.4
  Record Route:  NONE
  ... ..
  
```

可以看到 Path weight=3，这个 3 是怎么来的呢？因为 R1 向 R4 建立 tunnel，那么 R1 到达 R4（注意不是到 R4 的 loopback 口）的 OSPF metric=3，所以默认情况下 TE metric=3

当然，我们也可以修改 TE metric，例如在 R2 的 F1/0 口上：

```
R2(config-if)#mpls traffic-eng administrative-weight 100
```

那么 R1 上再去看 tunnel，他的 TE metric 就成了 102 了。

那么如果在 R1 上的 tunnel 口中，使用如下配置：

```
router(config)#tunnel mpls traffic-eng path-selection metric igp
```

那么 R1 将使用 IGP 的 metric，而忽略 R2 上物理接口配置的 adminnitation-weight，因此上面的命令生效后，再去 R1 上查看 tunnel 的 path weight，就又变成了 3。



## 3 路径计算

### 3.1 CSPF 算法概述

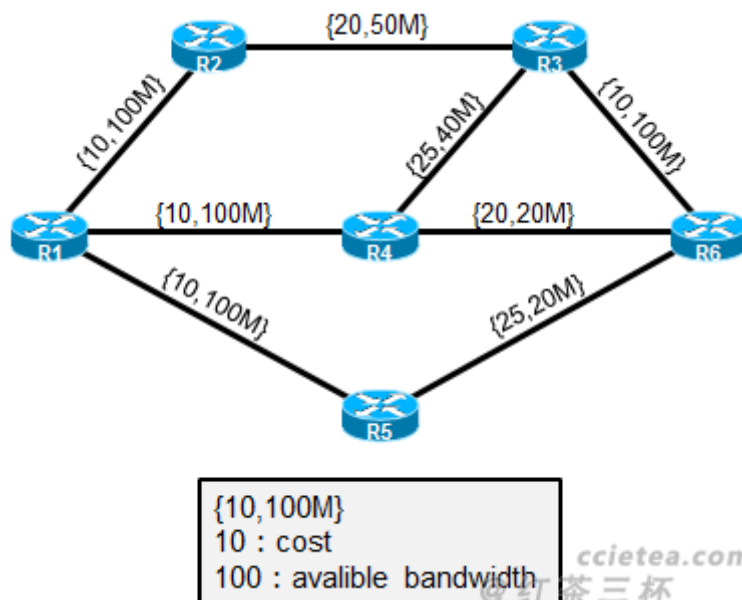
CSPF ( Constrained Shortest Path First ) 有约束条件的 SPF 算法

约束条件：

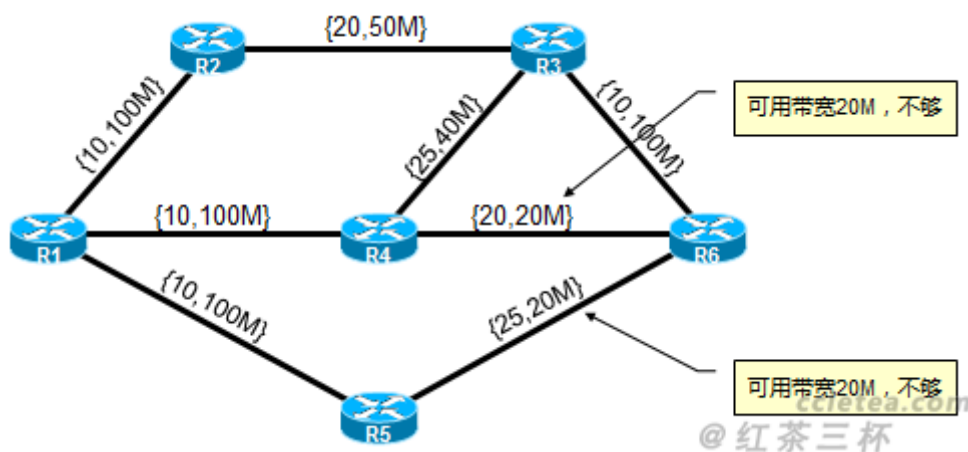
- TE Metric
- 可用带宽
- 链路属性

CSPF 算法的输出是在 tunnel 的两个端点之间的一个 IP 接口地址序列 ( 下一跳路由器地址 )。

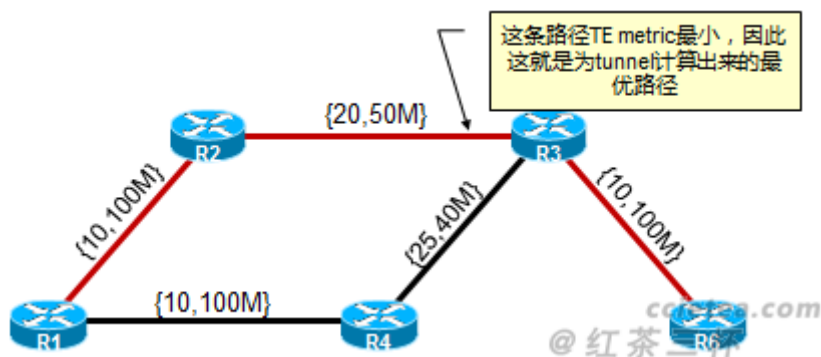
### 3.2 CSPF 算法工作机制



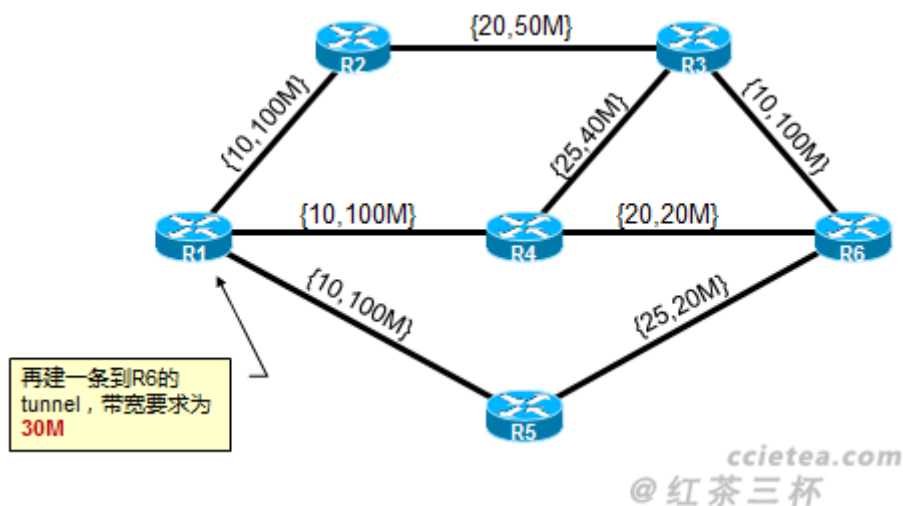
我们看上图，从 R1 到 R6 如果要建立一条 TE tunnel，也就是最优的路径，且带宽要求为 30Mbps，那么计算过程是怎样的呢？首先注意，得益于“针对 MPLS TE 扩展的 OSPF 或 IS-IS”在区域中的链路状态信息的泛洪，R1 此刻已经完成了自己 TEDB 的信息搜集，接下去 tunnel 路径的计算过程在 R1 上完成的。



R4-R6、R5-R6 之间的链路，带宽只有 20M，显然不符合要求。因此 R1 只会考虑如下链路：

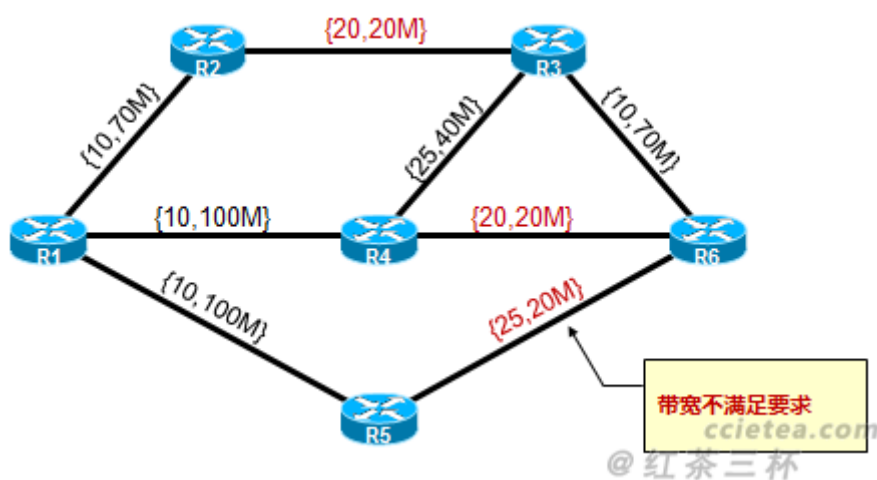
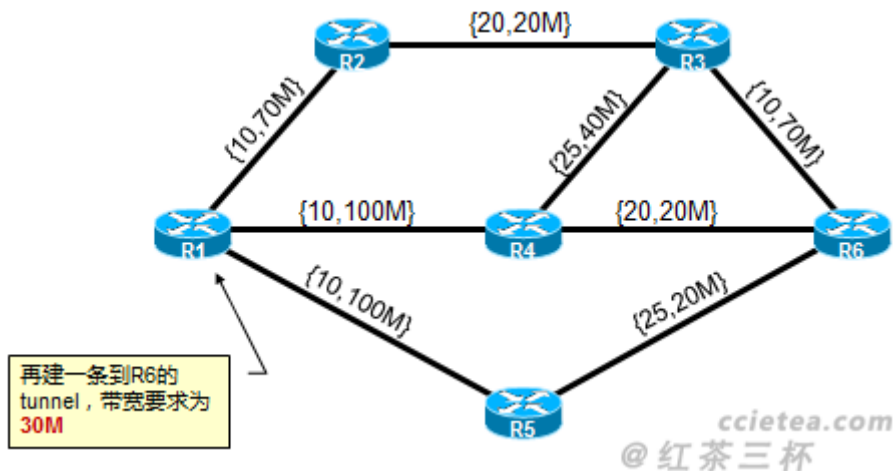


- R1 在剩余的两条路径的基础上，再去比较 TE metric (默认等于 IGP Metric)
- 最终选择上面的这条红色的路径，因为这条路径的 TE metric 更小

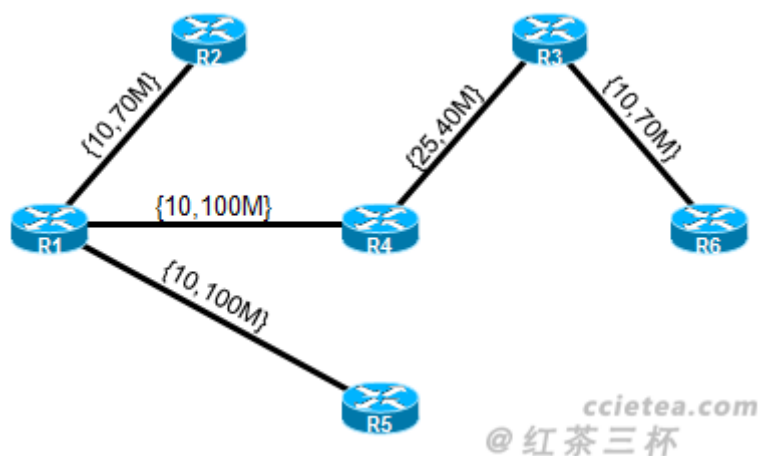


现在，我们要再建立一条 R1 到 R6 的 TE tunnel，带宽要求为 30M，注意，此刻网络中的可用带宽由于上面

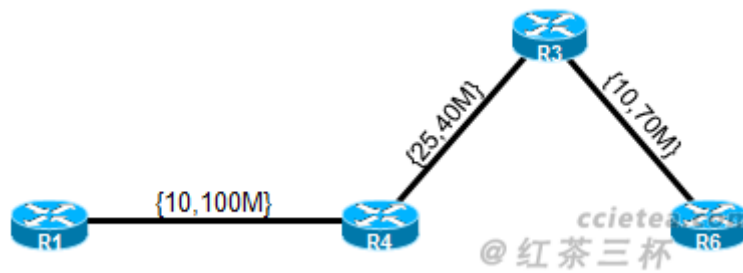
那条建立好的 tunnel 的存在已经发生了变化，此时此刻的网络环境如下：



- 带宽不满足要求的链路不考虑。那么经过筛选剩下如下链路：



- 最后，第二条 TE tunnel 的路径如下：



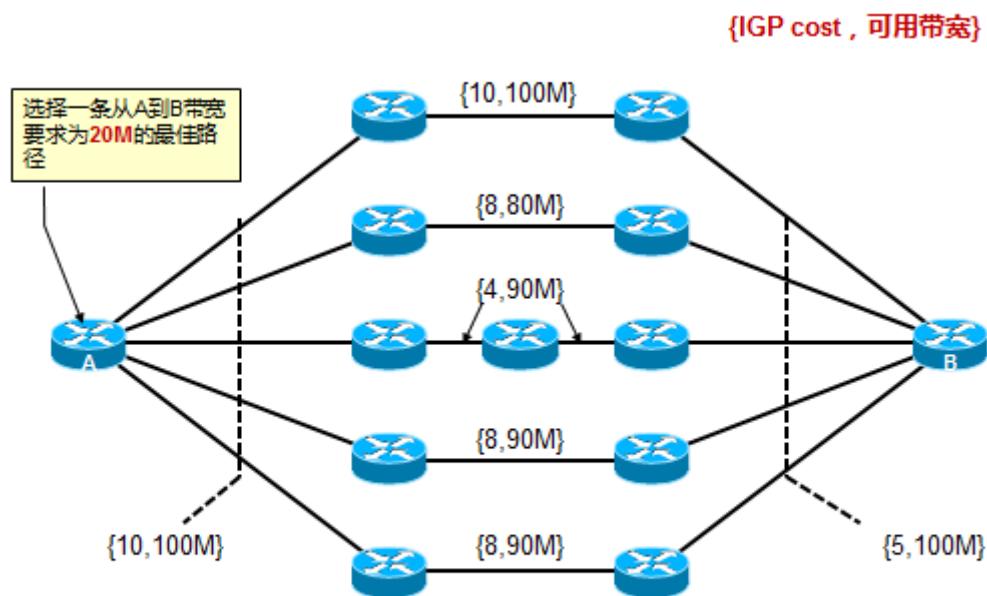
### 3.3 CSPF 最高仲裁

- 在标准的 SPF 算法中，到达同一个目的地可以有多条等价的路径存在，我们称之为 ECMP (Equal-cost Multipath)
- 但在 CSPF 算法中，对于一个目的地只能有一条路径。当存在多条满足基本条件的路径时（这个规则只有在 TE metric、带宽及相关属性等都无法决策的情况下才会进行）：

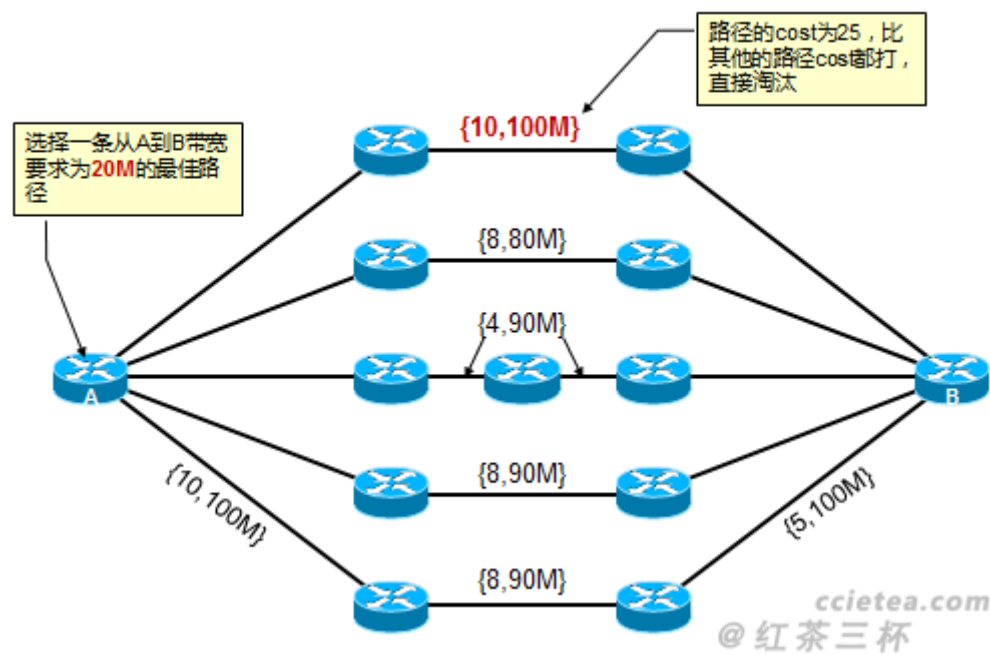
1. 选择 IGP cost 最小的路径
2. 选择有最大的“最小可用带宽”的路径
3. 选择最少跳数的路径
4. 如果还不能区分，则随机选择一条

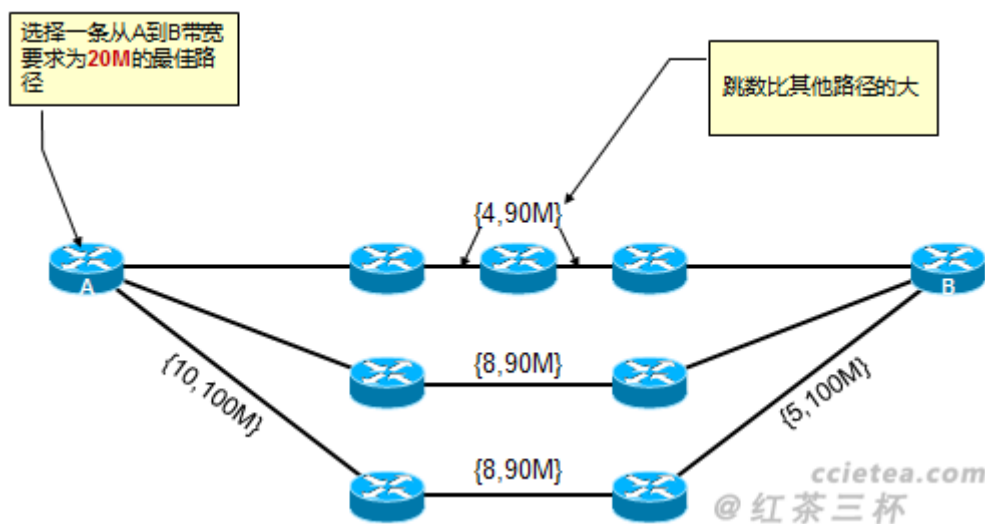
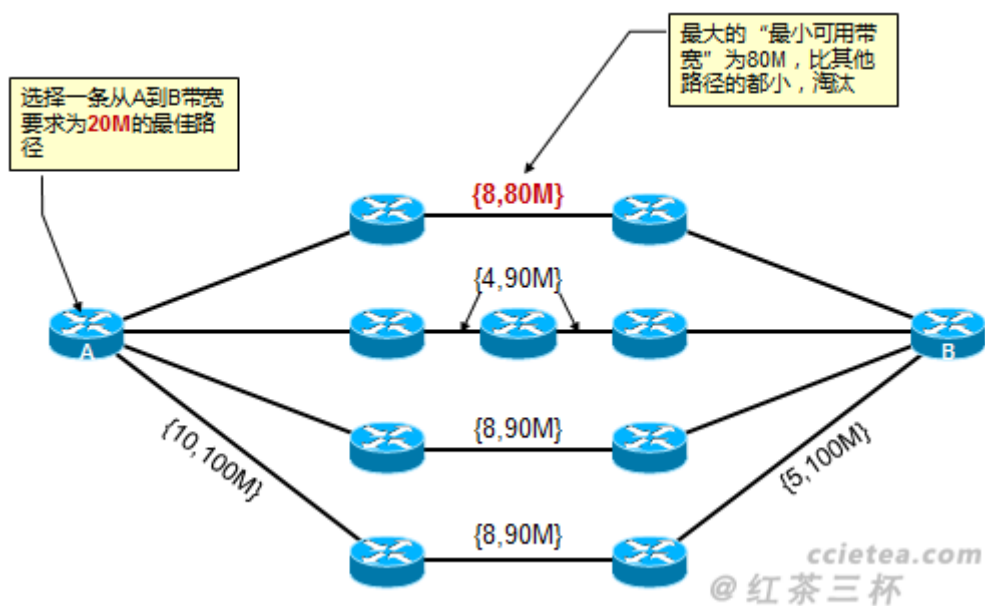
注意，这个所谓的最高仲裁，只有在多条路径的 TE metric 相等、可用带宽都满足要求，并且没有其他的链路属性的干扰的情况下，才会生效。最高仲裁中的第一条：选择 IGP cost 最小的路径，这个就是 IGP 的 cost，注意和 TE metric 区分，TE metric 在缺省的情况下的确和 IGP cost 一样，当然，我们也可以在物理接口上做修改。

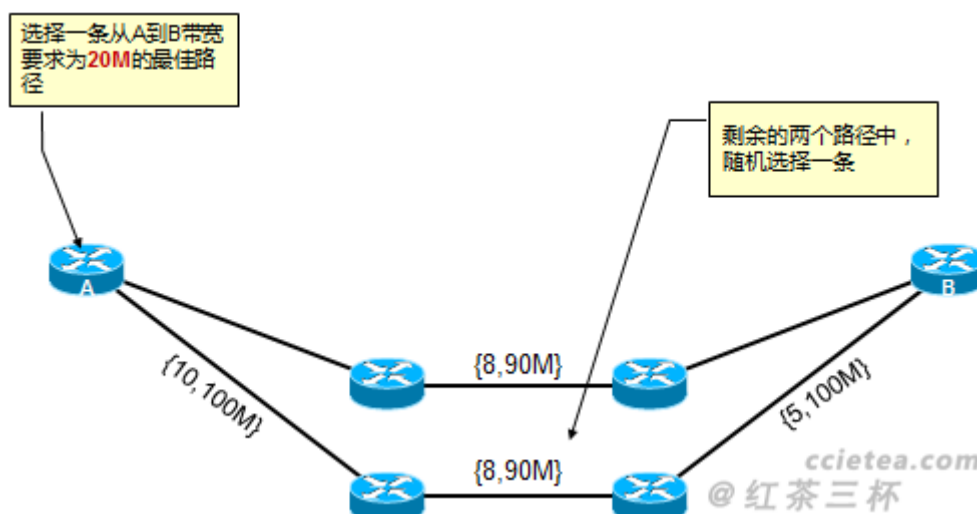
下面我们来看一个例子：



- 注：我们这个环境中所有路径的TE Metric均相等（如果TE metric不相等，就直接淘汰掉部分链路了，而不用进入到最高仲裁）



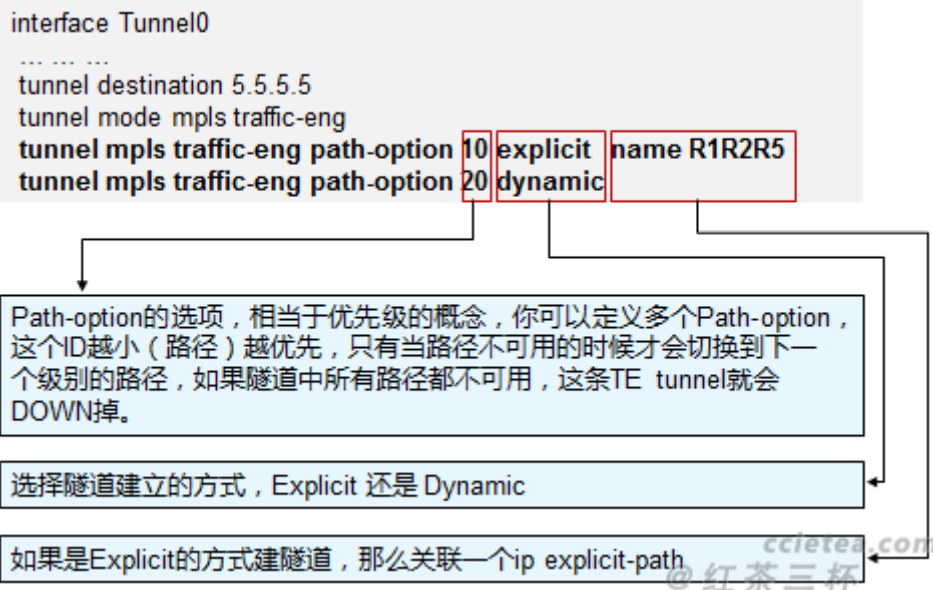




### 3.4 路径设置选项 path-option

#### 1. 可以通过两种方式设置 TE tunnel 的路径 ( Explicit 显式或 Dynamic 动态 )

- 在动态的设置中,让隧道首端路由器计算出 TE tunnel 能够以最优的方式穿越网络到达尾端的路径。我们只需要配置 tunnel 的 destination, 然后首端路由器通过 CSPF 算法自己去算路径。
- 在显式的设置方式中,我们必须要指定 TE tunnel 需要穿越的路由器, 你可以指定链路中路由器的 TE RouterID, 或者接口 IP 地址。



## 2. 关于 Explicit Path

- 当使用 Explicit 方式建立 tunnel 时，需关联一个 explicit path。
- Explicit Path 由一系列 IP 构成，一条 explicit path 上的 IP 可以是接口 IP，也可以是 MPLS TE 路由器 ID。
- 通过 exclude-address 命令来控制 LSP 不经过某节点（接口或路由器）。可以使用这个关键字搭配某个接口 IP 或者某台 MPLS TE 路由器 ID 从而让 tunnel 路径绕开某段链路或某台路由器。
- 定义一个 Explicit path 的命令如下：

```

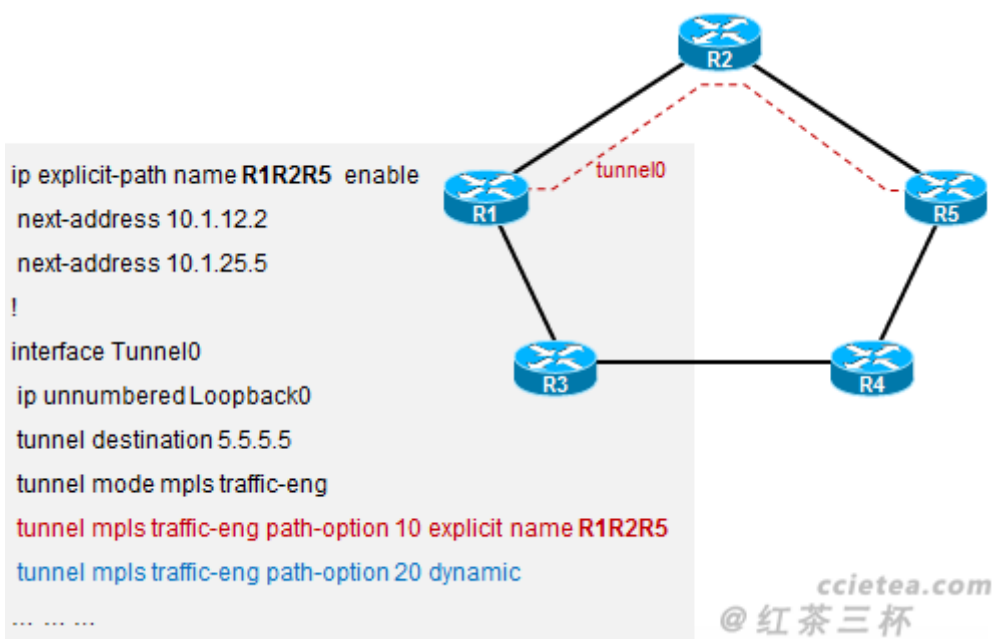
Router(config)#ip explicit-path name ccietea
Router(cfg-ip-expl-path)#next-address 10.1.12.2 !!注意输入顺序
Router(cfg-ip-expl-path)#next-address 10.1.25.5
Router(cfg-ip-expl-path)#
!
Router(config)#interface tunnel0
Router(config-if)#tunnel mpls traffic-eng path-option 10 explicit name ccietea !! 关联到 path-option
  
```

如上，注意输入的顺序。

在 cfg-ip-expl-path 模式内，还可以使用 list 关键字查看已经输入好的所有 next-address，使用 index x next-address 指定每一个已经输入的下一跳 IP 的索引编号。

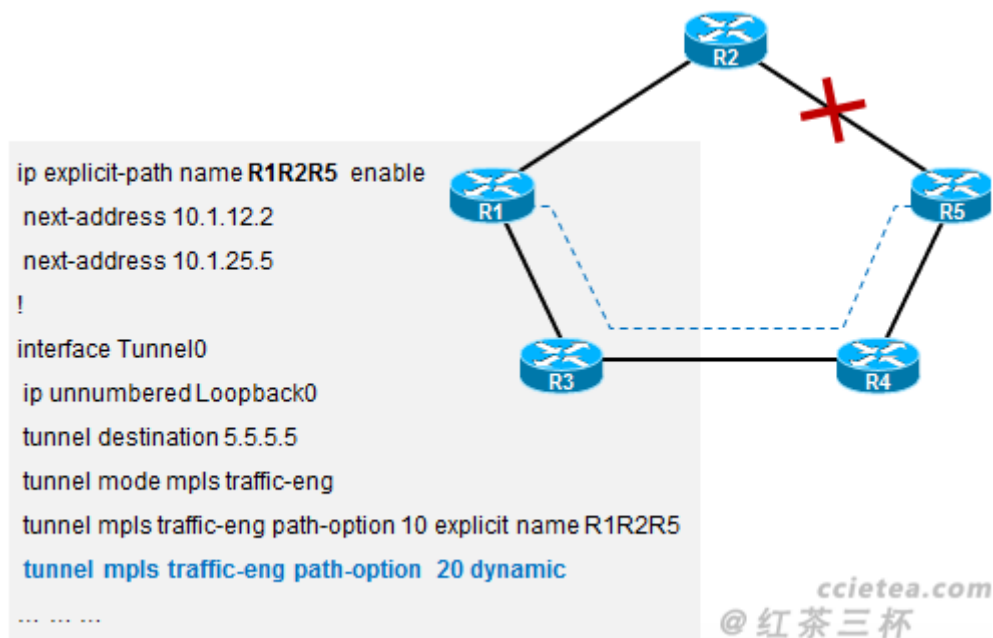
## 3. 实验验证





看上面这个拓扑，我们要在 R1 及 R5 之间建立一个 TE tunnel，在网络正常情况下，我们希望 tunnel 的路径是 R1-R2-R5，这是一个显式的指定路径。为了让 tunnel 的路径有备份可走，我们再定义一个 path-option，这个路径采用 dynamic 的设置方式，让 CSPF 自己去计算，那么配置如上。

当 R1-R2-R5 这段路径出现问题的时候，例如 R2-R5 之间的链路 DOWN 掉，那么 TE tunnel 会向下一个优先级的 Path-option 切换，也就是 dynamic 设置方式的这条，因此会切换到 R1-R3-R4-R5 上。



## 3.5 隧道优先级

如果网络中存在多条 Tunnel 为不同的流量服务,那么有可能这些 tunnel 的重要程度根据受实际的业务需求影响他们的重要程度也有所不同。那么如果我一跳更重要的 Tunnel 在一条更次要的 tunnel 之后配置的,就有可能导致更重要的这条 Tunnel 找不到最优化的、或者足够带宽的路径。我们可以通过**隧道优先级**的方式来规避上面的问题。

- **隧道优先级的范围 0-7, 越小越优先**

- **隧道优先级有两种类型:**

- 设置优先级 setup priority
- 保持优先级 hold priority

设置优先级和保持优先级都通过相应的数值来说明一条 TE tunnel 是否可以抢占另一条 TE tunnel。优先级越低重要程度就越高。

设置优先级说明某条 tunnel 的重要程度以至于可以抢占其他 tunnel;

保持优先级表明了某条 tunnel 的权重是多少,以便它可以在链路中被保留下来;

如果 tunnel 0 的设置优先级比 tunnel 1 的保持优先级低,那么 tunnel0 可以抢占 tunnel1。

- **设置命令**

```
router(config-if)#tunnel mpls traffic-eng priority 建立优先级 保持优先级
```

注意:不能将 tunnel 的建立优先级设置得比保持优先级高

## 4 RSVP

### 4.1 协议概述

#### 1. 相关 RFCs

RFC 2205	Resource Reservation Protocol(RSVP)
RFC 2210	The Use of RSVP with IETF Intergrated Services
RFC3209	RSVP 对 TE 的扩展

#### 2. 基本概念

在 TE 隧道中，需要一种信令协议来确保可以使用 TE 隧道所穿越的 LSR 接口的链路，并且可以逐跳地传播 TE 隧道中的流量所需要的标签。

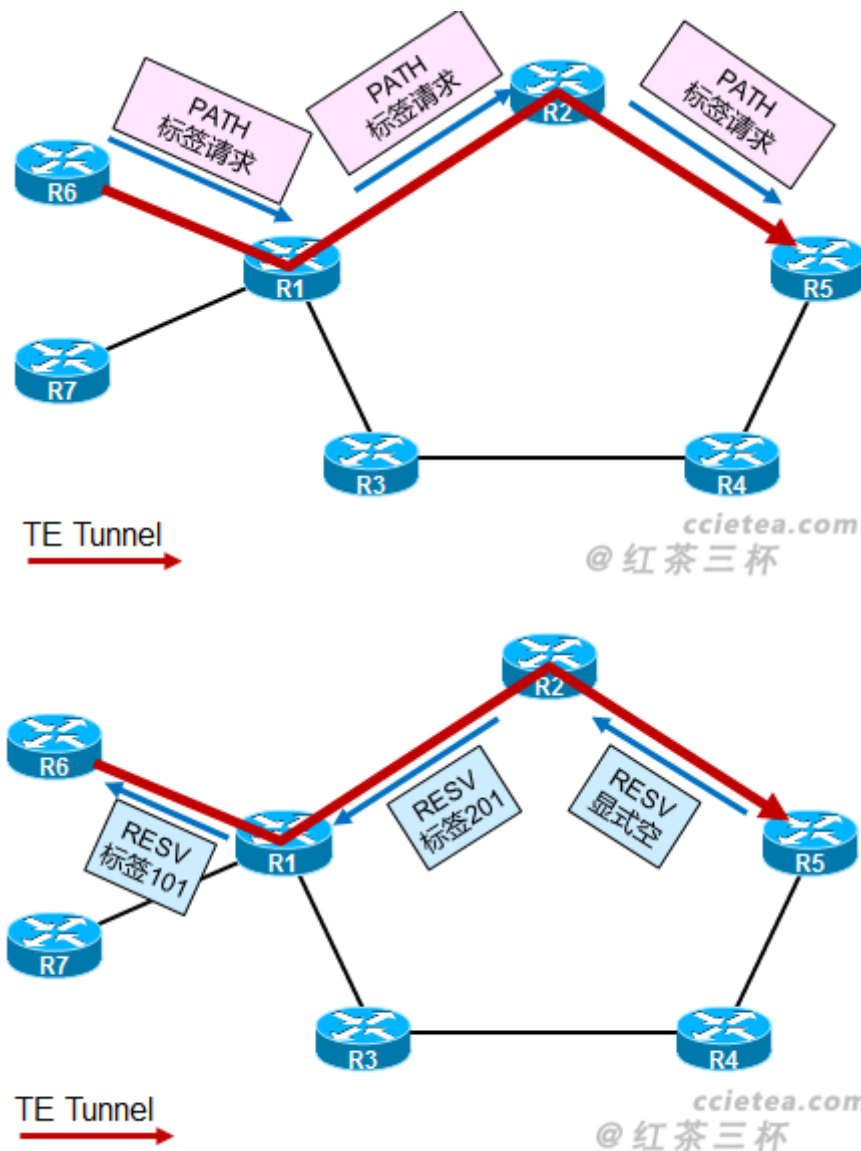
- TE 隧道的首端路由器根据 TE 数据库计算出 TE 隧道的最优路径，参考带宽和限制条件。另外最优路径还可以通过用户配置于隧道接口上的一个显式路径选项来定义。无论哪一种情况，首端路由器都会了解到 TE 隧道所使用的具体路径。
- RSVP 使用 PATH 和 RESV 消息来在一条路径中传递信令。TE 首端路由器发送 PATH 消息到尾端路由器，PATH 消息被一跳一跳的发送到尾端；而 RESV 消息则是通过相反的方向被发送给首端路由器。
- TE 隧道需要穿越的每一跳 LSR 都会被加入一个 ERO。这是一个接口 IP 地址的顺序列表，每一台 LSR 一个 IP 地址。
- PATH 消息是从首端路由器向后面的路由器进行传递的，那么下一跳路由器将从 ERO 中移除自己的 IP 地址，看到下一个 IP 地址是什么之后再将该 PATH 消息发送给下一跳。这个行为向后持续下去，直到该 TE 隧道的尾端路由器收到这个 PATH 消息。
- 在收到这个 PATH 消息后，尾端路由器将沿着 PATH 消息所使用的相同路径返回一个 RESV（保留）消息，只不过方向是逆向。RESV 消息里就包含 RSVP 为 Tunnel 分配的标签。

### 3. 所以，RSVP 的主要功能有：

- 路径的建立和维护
- 路径的拆除
- 错误通告

## 4.2 RSVP 与标签

RSVP 为 TE 隧道在路径中传递信令，但是它还承载 MPLS 标签分发的义务，使得报文可以沿着 TE 隧道的路径进行标签交换。

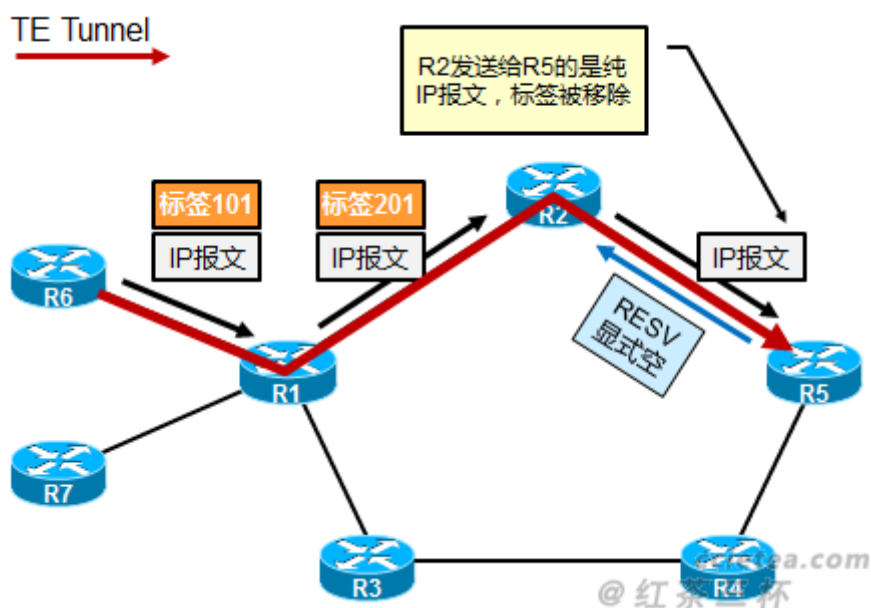
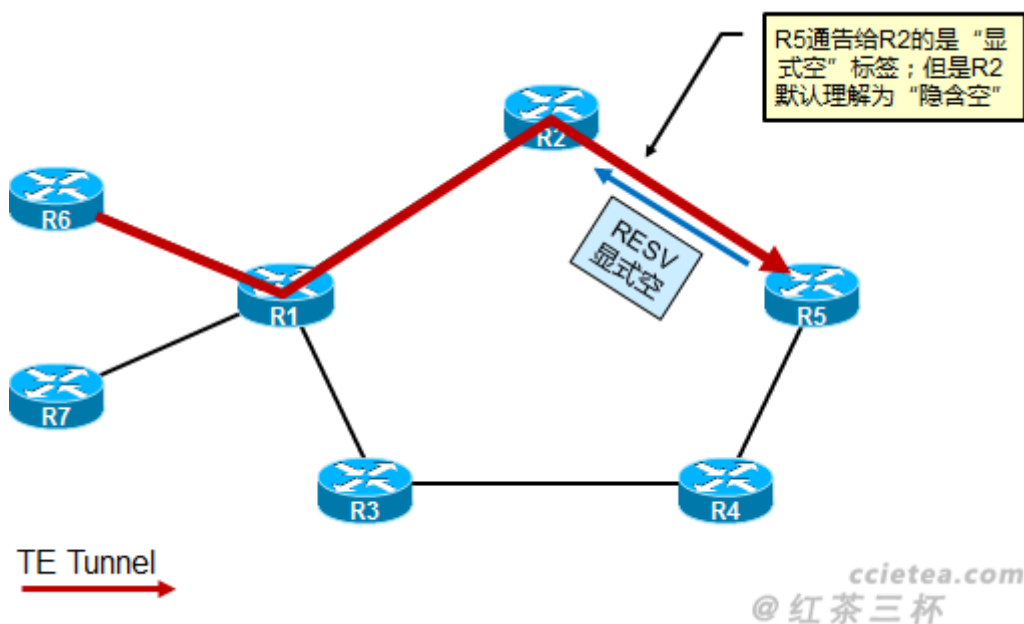


- 扩展的 OSPF 及 IS-IS 将 TE 所需的信息在网络中进行泛洪，那么 Tunnel 首端路由器就可以构建自己的 TE 数据库并且进行 CSPF 算法的计算，计算得出的结果是 Tunnel 的路径——一个 IP 地址的序列。
- 接下去 RSVP 的工作基于这个 IP 地址的序列继续工作。
- PATH 消息携带了一个标签请求对象，从 Tunnel 首端路由器开始一跳一跳的传递到了尾端路由器。
- 当尾端路由器接收到该标签请求对象的时候，它将为这一条 Tunnel LSP 分配一个标签，并且将该标签通过 RESV 消息中的标签对象通告给上游路由器（倒数第二跳）。这个标签是尾端路由器 LFIB 表中的入站标签。
- 倒数第二跳路由器从尾端路由器收到该标签后，将该标签作为这条 TE Tunnel 的出站标签，同时自己再为该隧道分配一个标签并将这个标签放置于 RESV 消息中的标签对象中发送给它自己的上游路由器。
- 这样的行为将持续下去直到 RESV 消息到达该 TE 隧道 LSP 的首端路由器。也就是说，在首端路由器发出请求之后，标签是从尾端路由器向首端路由器逐跳通告的。这说明 TE 隧道使用的是下游被动 DOD 的标签分发

模式。

### 注意事项：

- 尾端路由器给倒数第二跳路由器通告的标签是“显式空标签”，然而如果倒数第二跳路由器运行的是 CISCO IOS 的话，它将以隐式空标签的形式对此进行理解，也就是说，默认情况下，在 TE 隧道中，倒数第二条路由器发送给尾端路由器的报文是没有标签的（或者说顶部标签已被移除）。

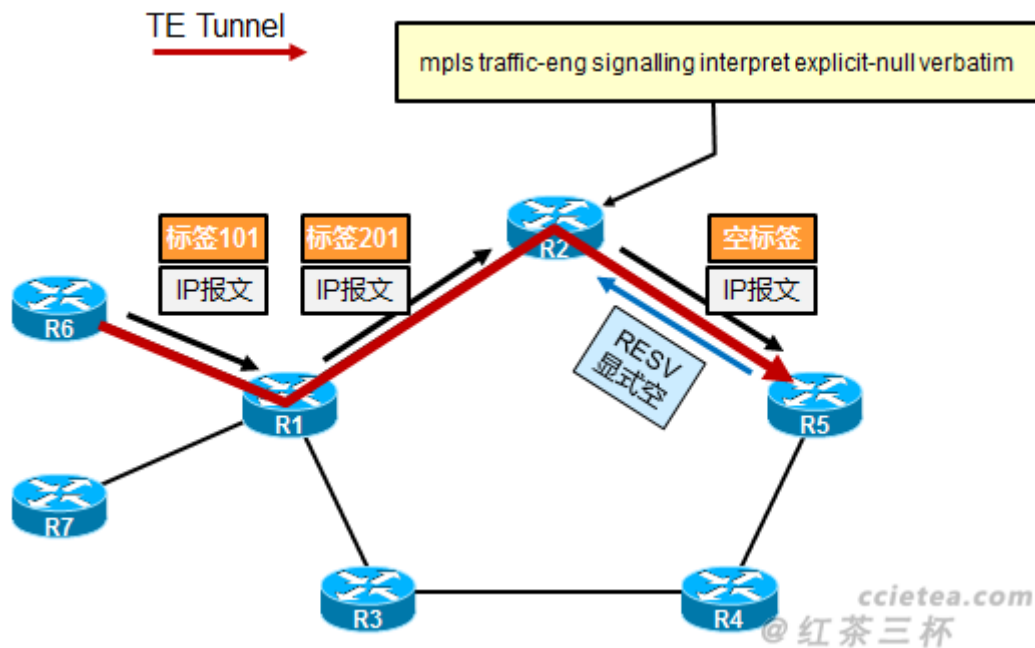


如果你希望倒数第二跳路由器将尾端路由器通告的“显式空标签”就理解为“显式空标签”，那么你可以

在倒数第二跳路由器（也就是上图中的 R2）上配置这条隐藏的命令：

```
mpls traffic-eng signalling interpret explicit-null verbatim
```

这条命令将使得倒数第二跳路由器在发送数据给尾端的时候，仍然带上标签，这个标签是空标签，verbatim 单词的中文翻译是“完全按照字面的”。这么做的目的，还是为了保留 EXP 字段，以便保留 QoS 信息，如下图：



另外，你可以在尾端路由器上配置如下命令：

```
mpls traffic-eng signalling advertise implicit-null [acl x]
```

这条命令将使得尾端路由器通告隐式空标签。（默认的动作通告显式空标签）

ACL 用来匹配向谁发送显式空标签。

- 因为 RSVP 已经能够完成标签分配的任务了，所以就不需要在接口上再激活标签分发协议（LDP）了。

## 4.3 消息类型

- RSVP 消息类型

- PATH 用来建立和维护、保留
- RESV 响应 PATH 消息，用来建立和维护 LSP 通道
- PATHTear 报文和 PATH 类似，只不过它是在需要拆除首端路由器创建的 TE tunnel LSP 的时候发送，例如 tunnel 接口被手工 shutdown。
- RESVTear 报文和 RESV 类似，只不过它是尾端路由器回应所收到的 PathTear 消息的时候发送
- PATHErr 实际上是发送给首端路由器的，发送该 error 消息的最主要原因是被 TE LSP 使用的链路发生故障，还可能是 LSR 收到带有伪造消息的 PATH 消息
- RESVErr 是发送给尾端路由器的

- RSVP 消息都有一个公共头部，后面跟随一个或多个 Objects

## 4.4 RSVP 对 LSP tunnel 的扩展

### 1. RSVP 对 LSP tunnel 的扩展

RFC3209 RSVP-TE Extensions to RSVP for LSP tunnels 中详细描述了 RSVP 协议对 MPLS TE 功能的扩展  
RSVP 对于 LSP tunnel 的扩展：

- 支持 DoD ( Downstream-on-Demand ) 方式的标签分配
- 支持为显式的 LSP tunnel 分配网络资源
- 改变属性 ( 路径、带宽 ) 时，采用 Make Before Break 的方式
- 支持记录 LSP tunnel 经过的每个站点，可以用来防止环路
- 支持对 LSP tunnel 进行诊断

### 2. RSVP-TE 扩展的新对象

对象名称	存在于报文	描述
Label_request	PATH	标签请求对象
Label	RESV	下游分配过来的标签
Explicit_route	PATH	这个不用说了吧？
Record_route	PATH/RESV	RRO

Session_Attribute	PATH	包含隧道的设置优先级、保持优先级和一些 flags 标志位
Sender_Template	PATH	包含隧道发送者地址、LSP ID 等信息
Filter_Spec	RESV	包含隧道发送者地址、LSP ID 等信息
Session	PATH/RESV	包含隧道终点地址、隧道 ID、扩展隧道 ID (就是隧道起点) 等信息
Flowspec	RESV	各种带宽信息
Sender_Tspec	PATH	主要包含 TE tunnel 的带宽请求

上述相同背景色的字段都是有对应关系的。

## 4.5 Record Route ( RRO )

激活 RRO 使用如下配置：

```
interface Tunnel0
ip unnumbered Loopback0
tunnel destination 5.5.5.5
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 60000
tunnel mpls traffic-eng path-option 10 explicit name R1R2R5
tunnel mpls traffic-eng path-option 20 dynamic
tunnel mpls traffic-eng record-route
```

激活 record route，可以使得 PATH、RESV 消息携带 record route 对象，记录下 tunnel 沿途的经过的 IP，可以起到一定的防环作用，当然还有其他重要的作用，在后面解释。

**R4#sh mpls tr tun**

```
LSP Tunnel R1_t0 is signalled, connection is up
InLabel : FastEthernet0/0, 400
OutLabel : FastEthernet1/0, implicit-null
RSVP Signalling Info:
Src 1.1.1.1, Dst 5.5.5.5, Tun_Id 0, Tun_Instance 26
```



#### RSVP Path Info:

My Address: 10.1.45.4

Explicit Route: 10.1.45.5 5.5.5.5

**Record Route: 10.1.34.3 10.1.13.1** **!!表示从 13.1 到 34.3 再到了本地 ( 34.4 )**

Tspec: ave rate=60000 kbits, burst=1000 bytes, peak rate=60000 kbits

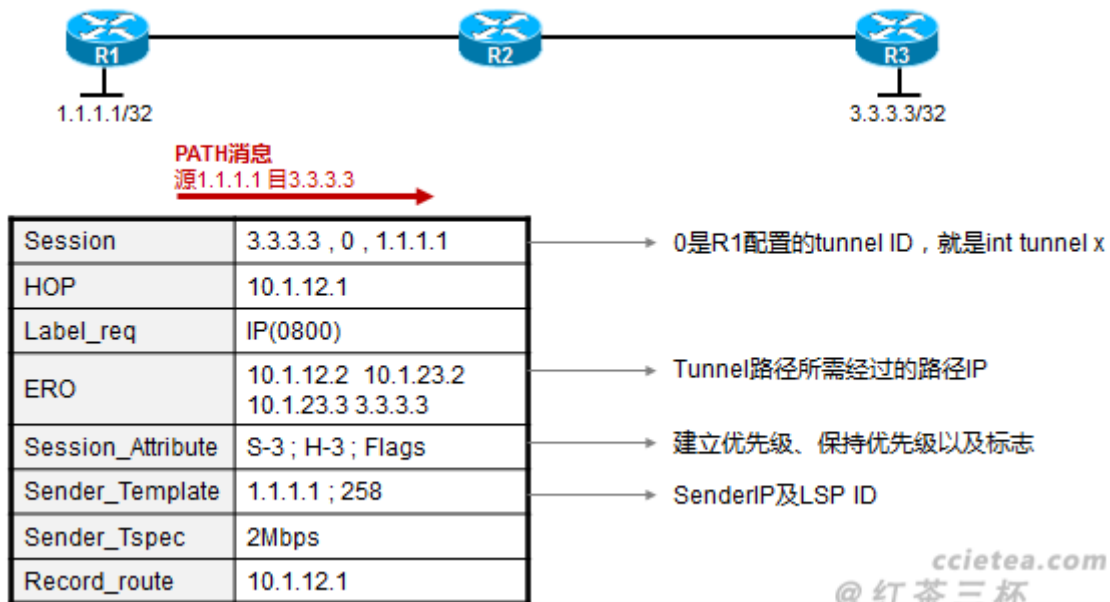
#### RSVP Resv Info:

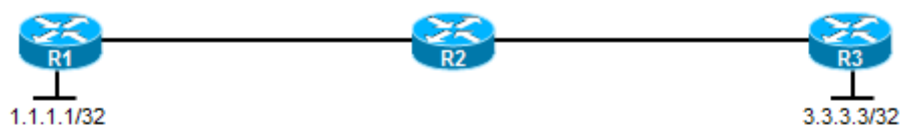
Record Route: NONE

Fspec: ave rate=60000 kbits, burst=1000 bytes, peak rate=60000 kbits

## 4.6 RSVP 路径建立过程

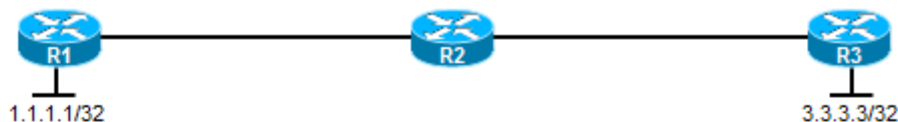
- TE LSP tunnel 都是由隧道的首端发起建立
- CSPF 计算出来路径
- RSVP-TE 按照上述路径请求建立 LSP tunnel
- LSP tunnel 都是单向的





PATH消息  
源1.1.1.1 目3.3.3.3

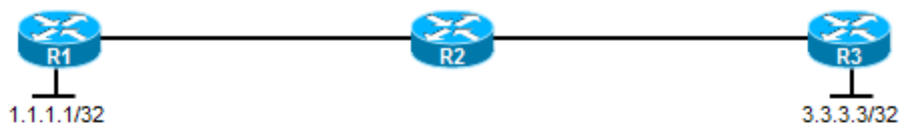
Session	3.3.3.3 , 0 , 1.1.1.1
HOP	10.1.23.2
Label_req	IP(0800)
ERO	10.1.23.3 3.3.3.3
Session_Attribute	S-3 ; H-3 ; Flag
Sender_Template	1.1.1.1 ; 258
Sender_Tspec	2Mbps
Record_route	10.1.23.2 10.1.12.1



PATH消息  
源10.1.23.3 目10.1.23.2

表示对Make Before Break的支持

Session	3.3.3.3 , 0 , 1.1.1.1
HOP	10.1.23.3
Style	Shared-Explicit
FlowSpec	2Mbps
FilterSpec	1.1.1.1 ; 258
Label	Pop (0)



PATH消息  
源10.1.23.3 目10.1.23.2

Session	3.3.3.3 , 0 , 1.1.1.1
HOP	10.1.12.2
Style	Shared-Explicit
FlowSpec	2Mbps
FilterSpec	1.1.1.1 ; 258
Label	200

## 4.7 RSVP 路径维护及拆除

### 1. 路径维护

- 每隔 30S 首端发送 PATH 消息给下游
- 下游邻居每隔 30S 发送 RESV 消息

### 2. 路径的拆除

如果首端要拆除 tunnel ,则沿 tunnel 路径发送 PATHtear ,收到的节点回应 RESVtear ,这样两点之间的 tunnel 就拆除了 , 然后继续往下拆除。



想象一下如果 AB 之间的链路承载了大量的 tunnel ,那么当这根链路 DOWN 掉的时候 ,将有可能在短时间内触发大量的 PATHtear 及 RESVtear ,那么如何优化呢 ?

可以使用 `router(config)# ip rsvp signalling rate-limit` 来限制信令消息的发送速率

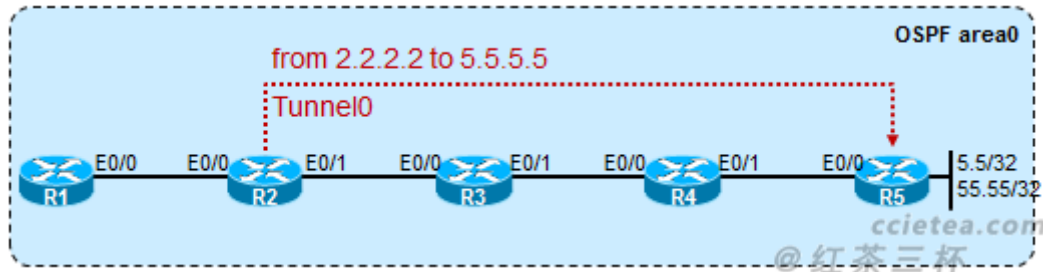
或者使用 `router(config-if)# hold-queue x in` 来限制接收速率

## 5 MPLS TE 流量转发

MPLS TE 报文转发是基于标签的 ,数据包将沿着预先建立好的 LSP 进行报文转发。那么 tunnel 建立好后 ,如何把流量引入 MPLS TE Tunnel 呢。流量只有上了 tunnel ,才会进入 LSP。有三种方法 :

- 静态路由 Static Route
- 自动路由 Auto Route
- 策略路由 Policy Route

## 5.1 Static route



假设我们现在已经建立好了一条首端为 R2、尾端为 R5 的 TE Tunnel，这将在 R2 本地路由表中产生一个 tunnel 接口。那么一种最简单的方法，就是将上 tunnel 的流量直接用静态路由引到 tunnel 接口。例如用下面的配置完成后，R1 及 R2 去往 55.55.55.0/24 的流量就会从 TE tunnel 走。

```
ip route 55.55.55.55 255.255.255.255 tunnel0
```

R1#traceroute 55.55.55.55

```
1 10.1.12.2 4 msec 0 msec 4 msec
2 10.1.23.3 [MPLS: Label 300 Exp 0] 0 msec 0 msec 0 msec
3 10.1.34.4 [MPLS: Label 400 Exp 0] 0 msec 12 msec 8 msec
4 10.1.45.5 8 msec 8 msec *
```

MPLS TE tunnel 的静态路由转发方式，支持路由递归。

使用静态路由的方式当然是最简单，也是最好把控的，但是扩展性太差，如果存在多条 tunnel 那么配置和维护就比较麻烦。

## 5.2 AutoRoute

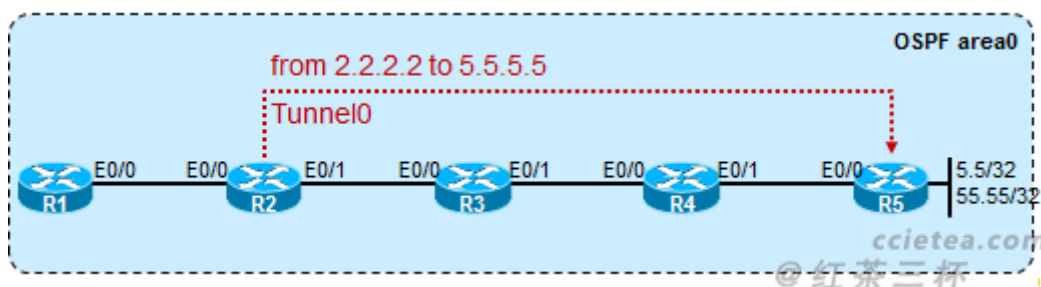
### 5.2.1 AutoRoute 概述

- MPLS TE tunnel 本身不支持 IGP 路由协议因为：
  - TE tunnel 是单向的
  - TEDB 已经拥有所有区域内的链路状态信息，所以不需要再使用 IGP 路由协议

- **Auto Route :**

- MPLS TE 的 AutoRoute 特性把 TE tunnel 作为一个直连链路参与 SPF 计算 (只是 R2 自己)
- 这里的 SPF 计算是普通的 SPF 计算, 而不是 CSPF
- AutoRoute 只会影响 TE 隧道的首端, 其他路由器并不知道 TE 隧道的存在, 不会影响他们的路由选路

## 5.2.2 AutoRoute 基本配置



**R1 的配置如下 :**

```
interface eth0/0
 ip address 10.1.12.1 255.255.255.0
interface loopback0
 ip address 1.1.1.1 255.255.255.255
!
Ip cef
mpls traffic-eng tunnels
mpls label range 100 199
!
router ospf 1
 router-id 1.1.1.1
 network 10.1.12.1 0.0.0.0 area 0
 mpls traffic-eng router-id loopback0
 mpls traffic-eng area 0
!
interface eth 0/0
```

```
mpls traffic-eng tunnels
ip rsvp bandwidth
```

### R2 的配置如下 (暂时没配置 tunnel):

```
interface eth0/0
  ip address 10.1.12.2 255.255.255.0
interface eth0/1
  ip address 10.1.23.2 255.255.255.0
interface loopback0
  ip address 2.2.2.2 255.255.255.255
!
Ip cef
mpls traffic-eng tunnels
mpls label range 200 299
!
router ospf 1
  router-id 2.2.2.2
  network 2.2.2.2 0.0.0.0 area 0
  network 10.1.12.2 0.0.0.0 area 0
  network 10.1.23.2 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/1
  mpls traffic-eng tunnels
  ip rsvp bandwidth
```

### R3 的配置如下 :

```
interface eth0/0
  ip address 10.1.23.3 255.255.255.0
interface eth0/1
```

```

ip address 10.1.34.3 255.255.255.0
interface loopback0
ip address 3.3.3.3 255.255.255.255
!
Ip cef
mpls traffic-eng tunnels
mpls label range 300 399
!
router ospf 1
router-id 3.3.3.3
network 10.1.23.3 0.0.0.0 area 0
network 10.1.34.3 0.0.0.0 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface eth 0/0
mpls traffic-eng tunnels
ip rsvp bandwidth
interface eth 0/1
mpls traffic-eng tunnels
ip rsvp bandwidth

```

#### R4 的配置如下：

```

interface eth0/0
ip address 10.1.34.4 255.255.255.0
interface eth0/1
ip address 10.1.45.4 255.255.255.0
interface loopback0
ip address 4.4.4.4 255.255.255.255
!
Ip cef
mpls traffic-eng tunnels
mpls label range 400 499

```

```
!
router ospf 1
  router-id 4.4.4.4
  network 10.1.34.4 0.0.0.0 area 0
  network 10.1.45.4 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/1
  mpls traffic-eng tunnels
  ip rsvp bandwidth
```

#### R5 的配置如下：

```
interface eth0/0
  ip address 10.1.45.5 255.255.255.0
interface loopback0
  ip address 5.5.5.5 255.255.255.255
interface loopback1
  ip address 55.55.55.55 255.255.255.255
!
ip cef
mpls traffic-eng tunnels
mpls label range 500 599
!
router ospf 1
  router-id 5.5.5.5
  network 10.1.45.5 0.0.0.0 area 0
  network 5.5.5.5 0.0.0.0 area 0
  network 55.55.55.55 area 0
  mpls traffic-eng router-id loopback0
```



```
mpls traffic-eng area 0
!
interface eth 0/0
 mpls traffic-eng tunnels
 ip rsvp bandwidth
```

那么到目前为止，基础配置都完成了。现在我们在 R2 上创建 TE Tunnel：

**R2 的配置增补如下：**

```
Interface tunnel0
 ip unnumbered loopback 0
 tunnel destination 5.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng path-option 10 dynamic
```

**tunnel mpls traffic-eng autoroute announce**

**!!激活 Tunnel 的 autoroute 特性**

完成上述配置后，相当于 R2 上多了一条到 R5 的单向直连链路。这条链路将直接参与 R2 本身的路由计算。总的结果就是，相当于 R2 上现在有了三个直连接口：eth0/0、eth0/1、tunnel0。

现在我们来验证一下。

**R2# show ip route**

```
C      2.2.2.2 is directly connected, Loopback0
O      5.5.5.5 [110/31] via 5.5.5.5, 00:00:01, Tunnel0
C      10.1.12.0/24 is directly connected, Ethernet0/0
C      10.1.23.0/24 is directly connected, Ethernet0/1
O      10.1.34.0/24 [110/20] via 10.1.23.3, 00:00:01, Ethernet0/1
O      10.1.45.0/24 [110/30] via 10.1.23.3, 00:00:01, Ethernet0/1
O      55.55.55.55 [110/31] via 5.5.5.5, 00:00:01, Tunnel0
```

我们看到 R2 上，关于 5.5.5.5 及 55.55.55.55 这两条在 R5 也就是 tunnel 尾端的“后方的”网络（的路由），出接口均为 tunnel0。这就意味着，当 R2 收到去往这两个网段的数据，直接放入 TE tunnel。相比于静态路由的放方式，autoroute 的优势就显示出来了。

**R1#traceroute 55.55.55.55**

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

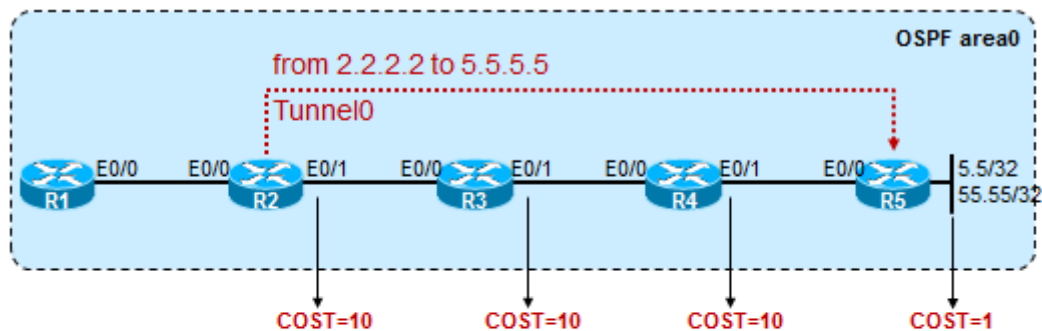
Tracing the route to 55.55.55.55

```
1 10.1.12.2 8 msec 4 msec 0 msec
2 10.1.23.3 [MPLS: Label 300 Exp 0] 4 msec 0 msec 4 msec
3 10.1.34.4 [MPLS: Label 400 Exp 0] 4 msec 0 msec 4 msec
4 10.1.45.5 0 msec * 0 msec
```

看到标签交换的过程了吧？证明数据的确是上了 TE tunnel 的。

### 5.2.3 AutoRoute metric

再进一步分析一下 metric 的问题。我们重点看 55.55.55.55，经过上面的基本配置，目前在 R2 的路由表中的 metric 为 31，计算方法当然很简单，如下：



R2# show ip route

```
O 5.5.55.5 [110/31] via 5.5.55.5, 00:00:01, Tunnel0
O 55.55.55.55 [110/31] via 5.5.55.5, 00:00:01, Tunnel0
```

ccietea.com  
@ 红茶三杯

注意，此刻如果希望通过修改 tunnel 接口的 ip ospf cost 从而来调整路由的 cost，这在针对 autoroute 特性计算得到的关联 tunnel 接口的路由是无效的，需使用 tunnel mpls traffic-eng autoroute metric 命令来配置。

#### • tunnel mpls traffic-eng autoroute metric 命令解析

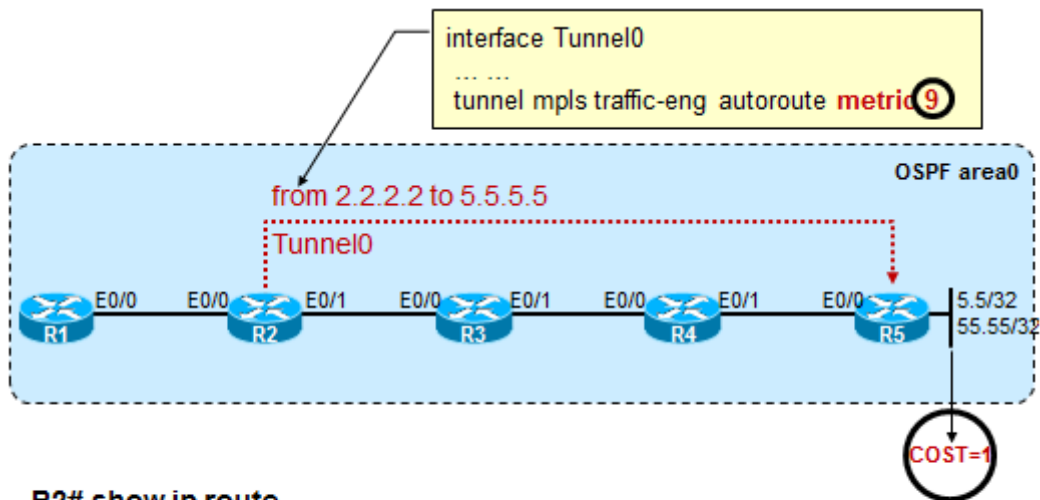
如果我们在 R2 的 TE tunnel 口中增加如下配置：

```
interface Tunnel0
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 5.5.55.5
```

```
tunnel mpls traffic-eng autoroute announce
```

```
tunnel mpls traffic-eng autoroute metric 9
```

```
tunnel mpls traffic-eng path-option 10 dynamic
```



R2# show ip route

```
O    5.5.5.5 [110/10] via 5.5.5.5, 00:00:01, Tunnel0
O    55.55.55.55 [110/10] via 5.5.5.5, 00:00:01, Tunnel0
```

ccietea.com  
@ 红茶三杯

增加这条命令后,依靠 autoroute 特性计算出来、关联 tunnel0 口的这两条 OSPF 路由 metric 就变成了 9+1 : 这里的 9 就是我们配置的 autoroute metric 9 ,这里的 1 就是 R5 的 Loopback 口 cost。注意, **这只会影响 R2 自身的路由选择。**

补充一下:上面这个环境中,loopback 接口 cost=1,其他所有物理接口的 cost=10。那么如果我们配置:

```
tunnel mpls traffic-eng autoroute metric 32
```

这将导致 R2 计算路由的时候,从 tunnel0 口到达 5 及 55 路由 metric 变成 32+1=33,大于从 e0/1 走的路由的 metric=31。于是 R2 的路由表发生了变化,5 及 55 网段的路由变成了:

```
O    5.5.5.5 [110/31] via 10.1.23.3, 00:02:29, Ethernet0/1
O    55.55.55.55 [110/31] via 10.1.23.3, 00:02:29, Ethernet0/1
```

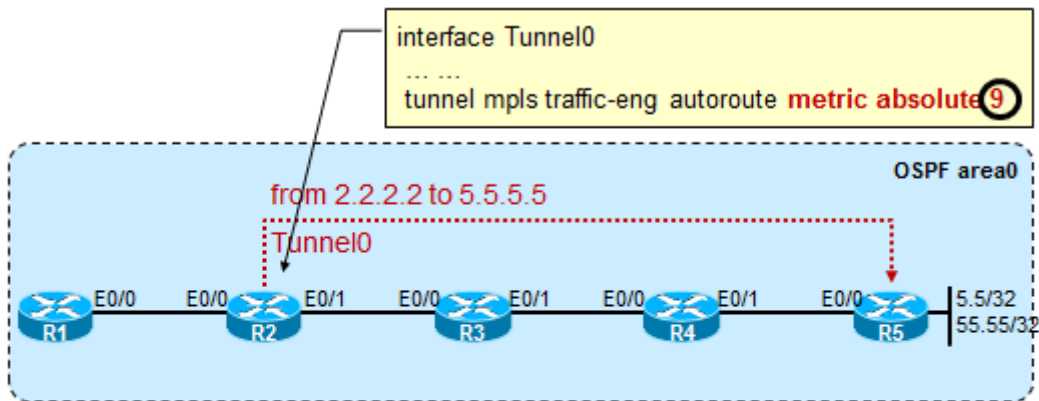
不再走 tunnel 了,此时再从 R2 去 traceroute 55.55.55.55,就不再看到标签了。

#### • tunnel mpls traffic-eng autoroute metric absolute 解析

如果 R2 的配置变更为:

```
interface Tunnel0
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
```

```
tunnel destination 5.5.5.5
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng autoroute metric absolute 9
tunnel mpls traffic-eng path-option 10 dynamic
```



R2# show ip route

```
O    5.5.5.5 [110/9] via 5.5.5.5, 00:00:01, Tunnel0
O    55.55.55.55 [110/9] via 5.5.5.5, 00:00:01, Tunnel0
```

ccietea.com  
@ 红茶三杯

配置这条命令，在 R2 上依赖 autoroute 特性计算出来的、关联 TE tunnel 口 OSPF 路由 metric 都变成了 9。这就是 absolute 关键字的意义，这里不会去做 cost 的累加。

补充一下：上面这个环境中，loopback 接口 cost=1，其他所有物理接口的 cost=10。那么如果我们配置：

**tunnel mpls traffic-eng autoroute metric absolute 32**

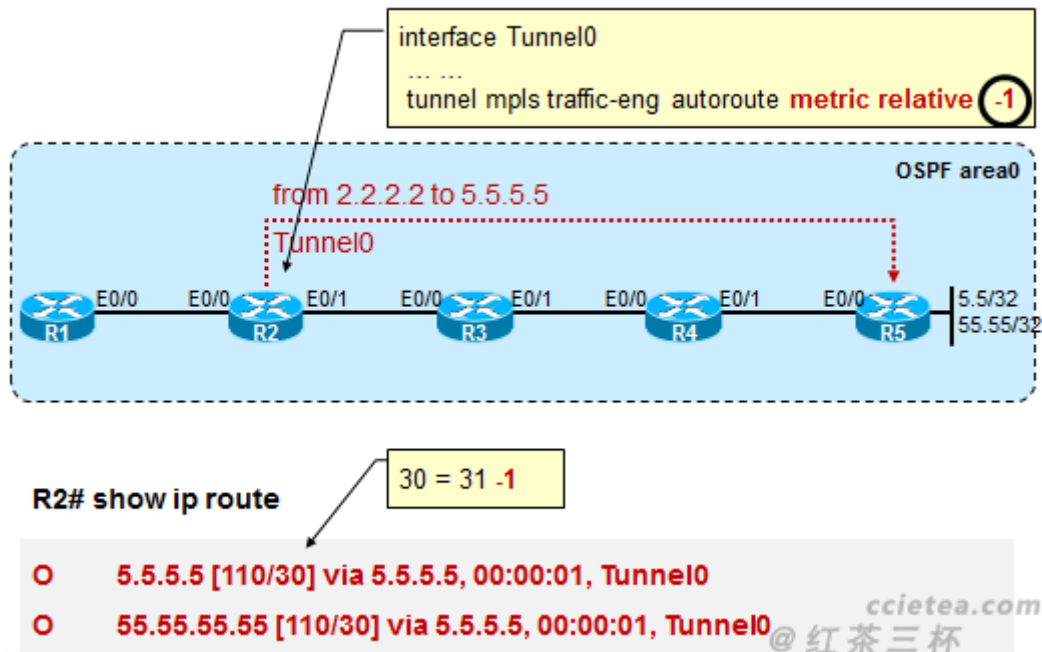
这将导致 R2 的路由发生变化，去往 5.5.5.5 及 55.55.55.55 的路由出接口变成了 e0/1。于是 R2 去往 55.55.55.55 及 5.5.5.5 将不再走 tunnel。

#### • tunnel mpls traffic-eng autoroute metric relative 解析

如果 R2 的配置变更如下：

```
interface Tunnel0
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 5.5.5.5
 tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng autoroute metric relative -1
```

```
tunnel mpls traffic-eng path-option 10 dynamic
```



这条 relative 的命令，可以配置个-10 到 10 的值。意思是在 IGP 度量值的基础上，加或者减去这个特定值。例如上图中，在不考虑 tunnel 的情况下，R2 上关于 5.5.5.5 和 55.55.55.55 的路由 metric=31。这个值怎么算的上面已经说过了。现在我们配置了这条带有 relative -1 关键字的命令后，依靠 autoroute 计算出来的、指向 tunnel0 口的 OSPF 路由，metric 就变成了 31-1=30。

小结一下，上面说了三条命令：

**tunnel mpls traffic-eng autoroute metric x**

首先这三条命令，都是对开启了 autoroute 特性的 TE tunnel 计算出来的路由得出的路由起作用的。上面这条命令，修改的是 tunnel 接口本身的 cost。

**tunnel mpls traffic-eng autoroute metric absolute y**

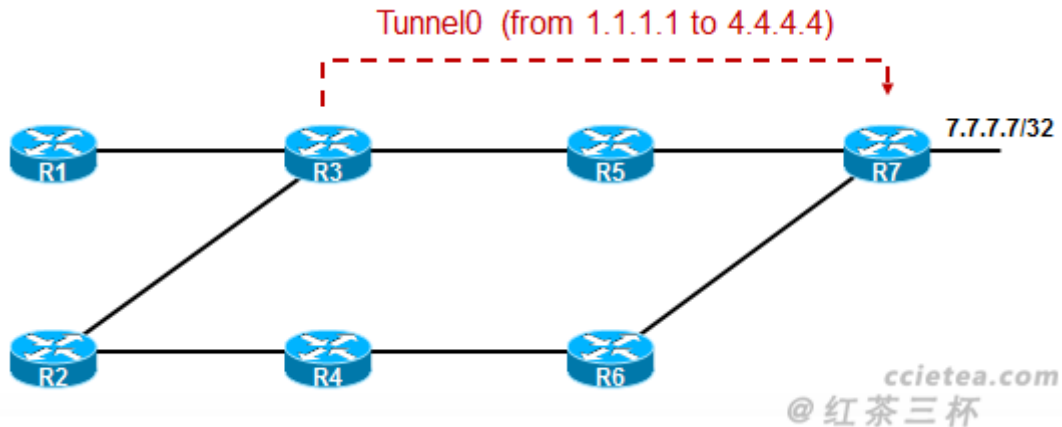
上面这条命令将直接把上述计算出来的路由的 cost 全改成 y

**tunnel mpls traffic-eng autoroute metric relative z**

上面这条命令将在路由的原始 cost 基础上，加上或者减去 z (z 的区间是-10 到 10)

## 5.2.4 AutoRoute 的缺陷

- Auto Route 在 MPLS TE tunnel 首端路由器上配置，只影响首端路由器的 OSPF 路由选路。
- 在实际的环境中，往往只有在少数的核心结点部署 MPLS TE，那么边缘结点无法感知 TE tunnel 的存在，只能按照传统的 IGP 选路。



例如上面的例子，tunnel0 是建立在 R3 到 R7 的，对于 R1 和 R2 而言，他们是不知道 tunnel 的存在的，当他们去往 7.7.7.7/32 的时候，只会进行 IGP 的本身的路由计算，那么 R2 就有可能走的是 R4-R6-R7 这条路径，但是我们可能更希望这些流量能够通过 tunnel 来传输，那么对于这个需求，autoroute 就无法实现了，需要借助 Forwarding adjacency。

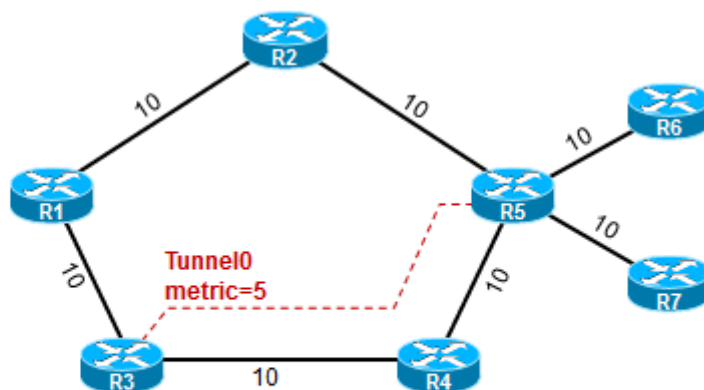
## 5.3 Forwarding Adjacency 转发邻接

### 5.3.1 特性概述

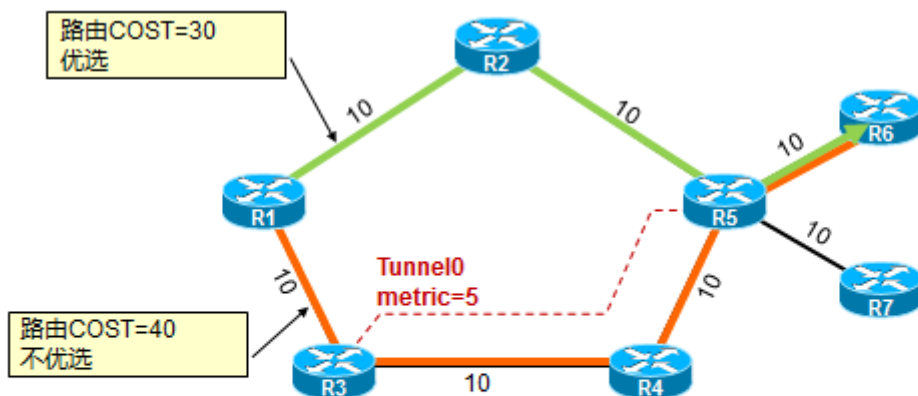
- 转发邻接特性把会使得 TE 路由器将 TE tunnel 作为虚路径（直连链路）在 IGP 路由协议区域内进行通告，这样所有的路由器都知道隧道的存在。这里的 IGP 协议可以是 OSPF 或 IS-IS
- 使用转发邻接特性时，隧道首端和尾端必须在同一个区域中，而且首尾两端的 Tunnel 接口都需要配置，也就是说这必须是一个双向隧道，并且两者的 tunnel 接口都要激活转发邻接

下面我们来看一个例子：

- **首先是无转发邻接的情况：**



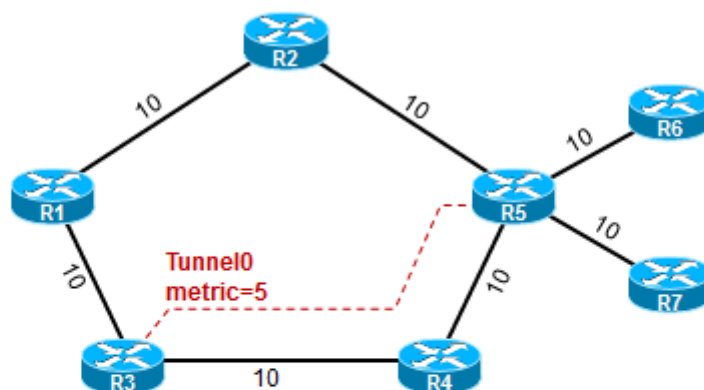
- R3及R5之间建立TE tunnel，注意这个tunnel是双向的



- TE隧道没有激活forwarding-adjacency时，R1到达R6的路径会选择R2作为下一跳，因为R1不知道tunnel0的存在，仍然以IGP metric看全局，从R2走的路由cost更小

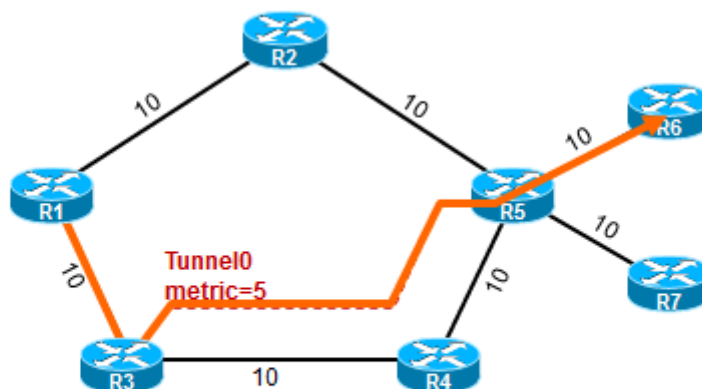
我们看到虽然 R3 到 R5 之间建立了 tunnel，但是这条 tunnel 对于 R1 来说是看不见的，那么在 R1 眼里，也就只有 OSPF 的那些链路 OSPF cost，因此 R1 前往 R6，肯定是优选上面那条链路，因为 cost 小。

- 接下去我们在 R3 和 R5 上激活转发邻接，那么 R3 和 R5 就会在区域内通告这条 tunnel（以直连链路的方式）。



- R3及R5在TE tunnel上激活forwarding-adjacency
- R3和R5会将tunnel的存在以直连链路的方式通过IGP协议（如OSPF或ISIS）告知给区域内其他路由器

ccietea.com  
@ 红茶三杯



- R1将选择R3作为到达R6的下一跳，R1上我们去看这条路径，它的metric=25

ccietea.com  
@ 红茶三杯

### 这里有几个注意事项：

1. R3 及 R5 都要建立 TE tunnel，也就是说，TE tunnel 必须是双向的
2. R3 及 R5 都要激活转发邻接特性
3. 在激活特性后，R3 和 R5 都会在区域内通告这条 tunnel（以直连链路的形式）的存在。相当于咱在拓扑环境中多了一条路径（虚路径）。



### 5.3.2 配置命令

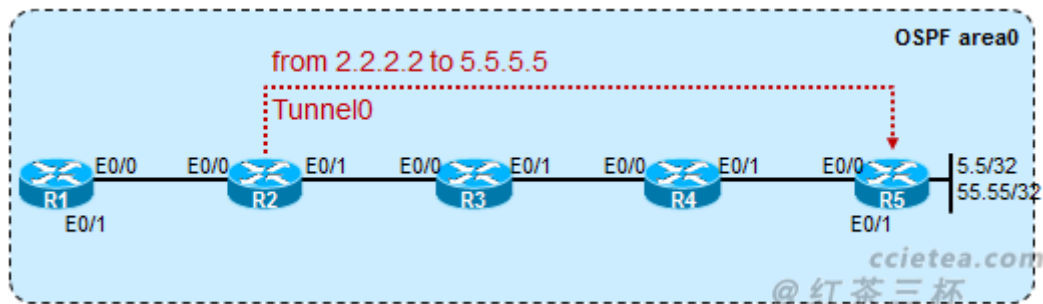
```
R3(config)#Interface tunnel 0
```

```
R3(config-if)#tunnel mpls traffic-eng forwarding-adjacency ?
```

holdtime      how long in msec to wait upon flooding a down forwarding      adjacency

tunnel 接口可以直接配置 igp metric，例如如果跑 ospf，直接在 tunnel 口中配置 ip ospf cost 即可

### 5.3.3 配置示例



**R1 的配置如下：**

```
interface eth0/0
  ip address 10.1.12.1 255.255.255.0
!
interface loopback0
  ip address 1.1.1.1 255.255.255.255
!
ip cef
mpls traffic-eng tunnels
mpls label range 100 199
!
router ospf 1
  router-id 1.1.1.1
  network 10.1.12.1 0.0.0.0 area 0
```

```
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface eth 0/0
mpls traffic-eng tunnels
ip rsvp bandwidth
```

**R2 的配置如下 (暂时没配置 tunnel):**

```
interface eth0/0
ip address 10.1.12.2 255.255.255.0
interface eth0/1
ip address 10.1.23.2 255.255.255.0
interface loopback0
ip address 2.2.2.2 255.255.255.255
!
Ip cef
mpls traffic-eng tunnels
mpls label range 200 299
!
router ospf 1
router-id 2.2.2.2
network 2.2.2.2 0.0.0.0 area 0
network 10.1.12.2 0.0.0.0 area 0
network 10.1.23.2 0.0.0.0 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface eth 0/0
mpls traffic-eng tunnels
ip rsvp bandwidth
interface eth 0/1
mpls traffic-eng tunnels
ip rsvp bandwidth
```

### R3 的配置如下：

```
interface eth0/0
  ip address 10.1.23.3 255.255.255.0
interface eth0/1
  ip address 10.1.34.3 255.255.255.0
interface loopback0
  ip address 3.3.3.3 255.255.255.255
!
Ip cef
mpls traffic-eng tunnels
mpls label range 300 399
!
router ospf 1
  router-id 3.3.3.3
  network 10.1.23.3 0.0.0.0 area 0
  network 10.1.34.3 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/1
  mpls traffic-eng tunnels
  ip rsvp bandwidth
```

### R4 的配置如下：

```
interface eth0/0
  ip address 10.1.34.4 255.255.255.0
interface eth0/1
  ip address 10.1.45.4 255.255.255.0
interface loopback0
  ip address 4.4.4.4 255.255.255.255
```

```
!
Ip cef
mpls traffic-eng tunnels
mpls label range 400 499
!
router ospf 1
  router-id 4.4.4.4
  network 10.1.34.4 0.0.0.0 area 0
  network 10.1.45.4 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/1
  mpls traffic-eng tunnels
  ip rsvp bandwidth
```

#### R5 的配置如下：

```
interface eth0/0
  ip address 10.1.45.5 255.255.255.0
interface loopback0
  ip address 5.5.5.5 255.255.255.255
interface loopback1
  ip address 55.55.55.55 255.255.255.255
!
Ip cef
mpls traffic-eng tunnels
mpls label range 500 599
!
router ospf 1
  router-id 5.5.5.5
```

```
network 10.1.45.5 0.0.0.0 area 0
network 5.5.5.5 0.0.0.0 area 0
network 55.55.55.55 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface eth 0/0
mpls traffic-eng tunnels
ip rsvp bandwidth
```

那么到目前为止，基础配置都完成了。现在我们在 R2 和 R5 上创建 TE Tunnel：

#### R2 的配置增补如下：

```
Interface tunnel0
ip unnumbered loopback 0
tunnel destination 5.5.5.5
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng path-option 10 dynamic
tunnel mpls traffic-eng forwarding-adjacency      !!激活 Tunnel 的 forwarding-adjacency
```

#### R5 的配置增补如下：

```
Interface tunnel0
ip unnumbered loopback 0
tunnel destination 2.2.2.2
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng path-option 10 dynamic
tunnel mpls traffic-eng forwarding-adjacency      !!激活 Tunnel 的 forwarding-adjacency
```

完成上述配置后，R2 及 R5 将在自己的 1 类 LSA 中通告这条 tunnel，就像一条直连链路一样。

#### R2#show ip ospf database router self-originate

```
OSPF Router with ID (2.2.2.2) (Process ID 1)
Router Link States (Area 0)
LS age: 43
Options: (No TOS-capability, DC)
LS Type: Router Links
```

Link State ID: 2.2.2.2  
 Advertising Router: 2.2.2.2  
 LS Seq Number: 80000011  
 Checksum: 0xAE93  
 Length: 72  
 Number of Links: 4

**Link connected to: another Router (point-to-point)**

**(Link ID) Neighboring Router ID: 5.5.5.5**

**(Link Data) Router Interface address: 0.0.0.13**

**Number of MTID metrics: 0**

**TOS 0 Metrics: 1000**

**!!默认 tunnel 口通告的 cost=1000**

Link connected to: a Stub Network

(Link ID) Network/subnet number: 2.2.2.2

(Link Data) Network Mask: 255.255.255.255

Number of MTID metrics: 0

TOS 0 Metrics: 1

Link connected to: a Transit Network

(Link ID) Designated Router address: 10.1.23.3

(Link Data) Router Interface address: 10.1.23.2

Number of MTID metrics: 0

TOS 0 Metrics: 10

Link connected to: a Transit Network

(Link ID) Designated Router address: 10.1.12.1

(Link Data) Router Interface address: 10.1.12.2

Number of MTID metrics: 0

TOS 0 Metrics: 10

R5 也是类似：

此时此刻对于 R2 来说，他有两条路径去往 5.5.5.5 和 55.55.55.55。

一条从 tunnel0 口走， $\text{cost}=1000+1=1001$

一条从 eth0/1 口走， $\text{cost}=10+10+10+1=31$

这时候 R2 去往 55.55.55.55，走的是传统 IP 数据，而不走上 LSP。

**R2#show ip route**

```
O          5.5.5.5 [110/31] via 10.1.23.3, 00:03:44, Ethernet0/1
O          55.55.55.55 [110/31] via 10.1.23.3, 00:03:44, Ethernet0/1
```

但是如果我们在 R1 上修改配置：

```
Interface tunnel0
Ip ospf cost 1
```

那么从 tunnel0 口， $\text{cost}=1+1=2$ ，更优，于是 R2 的路由表就变了：

**R2#show ip route**

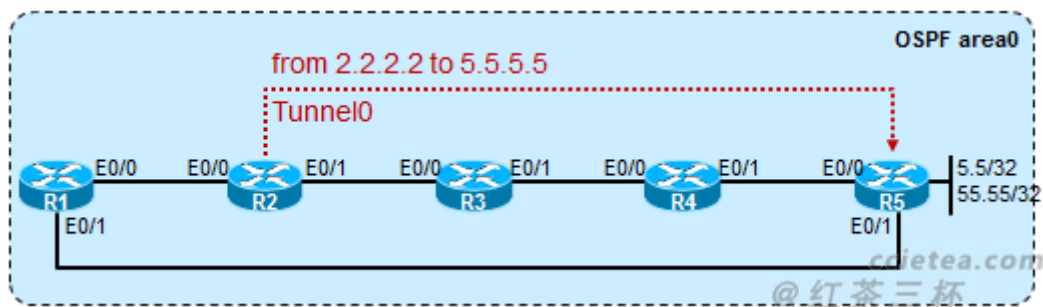
```
O          5.5.5.5 [110/2] via 0.0.0.0, 00:00:01, Tunnel0
O          55.55.55.55 [110/2] via 0.0.0.0, 00:00:01, Tunnel0
```

R2 优选从 tunnel 口走的路由，这时候 R2 去往 55.55.55.55 的流量就上了 tunnel，走 LSP 进行标签交换。

所以我们在 CISCO IOS 路由器上，激活转发邻接特性，除了让 TE tunnel 端点路由器将 tunnel 当成直连链路进行通告，同时在自身进行 SPF 算法计算的时候，也会一并考虑这条 tunnel。而且如果我们修改 tunnel 口的 IGP metric，这个 tunnel 的 metric（对于 OSPF 来说这个 tunnel 口默认  $\text{cost}=1000$ ）也会随着链路状态信息一并通告给其他的邻居，并且也影响其他路由器的选路。

这里要注意和 autoroute 区分开来。我们上面的配置中没有激活 autoroute 特性。autoroute 特性以及 autoroute metric 的配置都只影响 tunnel 端点路由器自身，而不会通告给其他邻居。但是转发邻接特性就不同了。而且两者修改 metric 的方式也不用，autoroute 需要搭配 autoroute 关键字来修改 metric 并且只影响配置者自身。而在转发邻接中，是直接在 tunnel 接口中修改 metric，例如使用 ISIS，那么就是 isis metric 命令，如果用 ospf，那就是 ip ospf cost。

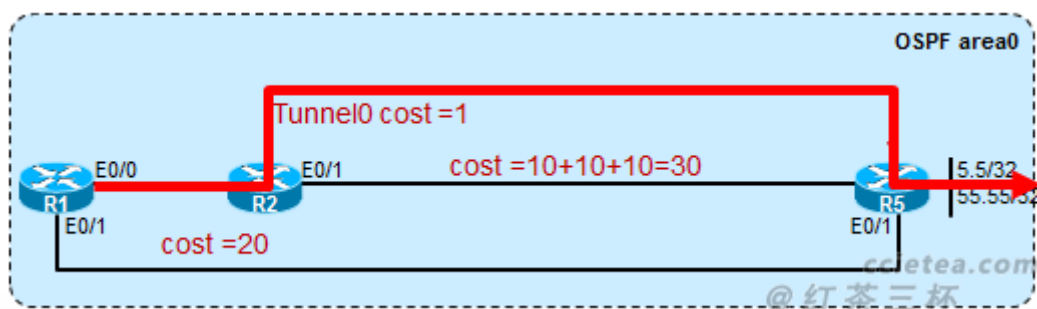
现在我们来考虑另一个场景，假设 R1 和 R5 之间增加一条物理连接，为了方便实验我直接在 R1-R5 之间拉了条线，大家权当这条新增的路径中有多台路由器，发挥下想象吧亲：



那么默认情况下，R1 去往 5.5.5.5 及 55.55.55.55 肯定是优选从 E0/1 直接到 R5。

但是，如果我们希望 R1 去往 5.5.5.5 及 55.55.55.55 流量上 TE tunnel 呢？

其实这就是个 metric 的把戏，只不过我们在考虑的时候，记得将 tunnel 这条虚路径一并考虑进去。



在本环境中，所有物理接口的 ospf cost=10，loopback 口的 cost=1

默认情况下 tunnel0 口的 cost=1000，我们可以将 tunnel 接口的 cost 修改为 1；再把 R1 的 e0/1 口的 cost 改为 20。

这样一来首先 R2 自己会优选从 tunnel 口到 R5，metric=1+1=2。而对于 R1 去 R5 有两个选择：

R1 从 R2 走然后上 tunnel，这条路径的 cost=10+1+1=12

R1 从 eth0/1 到 R5，这条路径的 cost=20+1=21

## 5.4 策略路由

- TE 的策略路由与传统的策略路由没什么区别，也不会改变路由表流量的转发基于配置的策略
- 只需把满足条件的流量的下一跳扔到 tunnel 接口即可



## 6 MPLS TE 配置及实验

### 6.1 配置命令

#### 1. 基本配置

```
router(config)#ip cef
```

这个不用说了吧？激活 CEF

```
router(config)#mpls traffic-eng tunnels
```

全局开启 MPLS TE tunnel 特性，相当于一个全局的开关

```
router(config-if)#mpls traffic-eng tunnels
```

激活接口上的 MPLS TE tunnel 特性

```
router(config-if)#ip rsvp bandwidth [total-flow-kbps] [single-flow-kbps]
```

激活接口上的 RSVP，并配置预留带宽（可选）

- 如果该命令后不接带宽参数，那么该接口的“总可预留带宽”默认为该接口物理带宽的 75%
- ip rsvp bandwidth 2000 这条命令将接口的总可预留带宽设置为 2M
- ip rsvp bandwidth 2000 500 后面的 500 是针对单个流的，限制在 500K

```
router(config-router)#mpls traffic-eng area x
```

在 OSPF 进程模式中配置，这条命令将激活特定区域的 MPLS TE

```
router(config-router)#mpls traffic-eng router-id {interface}
```

在 IGP（如 OSPF 或 ISIS）的路由进程下配置。这条命令设置 TE router 的 routerID

```
router(config)#interface tunnel x
```

创建一个 tunnel

```
router(config-if)#tunnel destination ip-addr
```

配置 tunnel 的目的地

```
router(config-if)#tunnel mode mpls traffic-eng
```

配置 tunnel 的封装模式为 MPLS TE

```
router(config-if)#tunnel mpls traffic-eng bandwidth BW
```

配置 tunnel 的 requested bandwidth

```
router(config-if)#tunnel mpls traffic-eng path-option x {dynamic | explicit {name path-name | path-number } } [lockdown]
```

配置 tunnel 使用静态指定的路径或通过 TE 拓扑数据库计算出来的动态路径

## 2. 3 种可能导致 TE tunnel 重新最优化 ( reoptimize ) 的触发因素。

- **周期性 reoptimize**

在 CISCO IOS 中，一条 TE tunnel 的 reoptimize 默认每 1 小时进行一次。

```
router(config)#mpls traffic-eng reoptimize timers frequency ?
```

如果该时间指定为 0，那么周期性 reoptimize 将会在路由器上的所有 TE tunnel 中被关闭。当然，你可以为单条 TE tunnel 关闭 reoptimize，命令如下：

```
Router(config-if)# mpls traffic-eng path-option x {dynamic | explicit name y } [lockdown]
```

使用 lockdown 关键字（注意上述配置是在 TE tunnel 口中配置）。

- **事件导致的 reoptimize**

缺省情况下，CISCO IOS 不会因为网络中的一条链路重新可以被一条 TE tunnel 所使用的时候而触发重新最优化，但是可以激活这个操作。要在一条链路在 MPLS TE 中变为可操作的时候启用重新最优化，使用下面的命令：

```
mpls traffic-eng reoptimize events link-up
```

经实验验证有效。可用 debug mpls traffic-eng tunnels events 及 debug mpls traffic-eng tunnels reoptimize 查看。

- **手工重新 reoptimize**

```
Router# mpls traffic-eng reoptimize
```

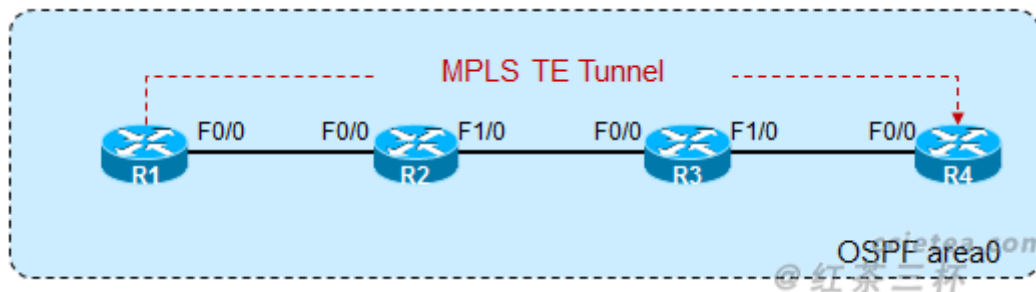
手工重新最优化

## 6.2 验证查看

```
debug ip rsvp dump-messages
```

可以看到出站 PATH 及入站 RESV 消息

## 6.3 基础实验 1 ( OSPF )



### 1. 实验环境

- 设备互联网段为 10.1.xy.0/24，其中 xy 为设备编号，x 小 y 大
- 所有设备的 loopback0 地址空间为 x.x.x.x/32，x 为设备编号
- 全网运行 OSPF，宣告直连和 loopback 接口
- 在 R1 上建立一个 TE Tunnel，源为自身的 loopback0，目的为 R4 的 4.4.4.4

### 2. 实验步骤

- 完成基本的 IP 配置（配置省略）
- 所有路由器运行 OSPF
- 全局激活 MPLS TE tunnel，并设定 MPLS 标签空间
- 为每台路由器的 OSPF 激活 MPLS TE 的扩展，并且手工设置用于 MPLS TE 的 RouterID
- 每台路由器的接口都激活 RSVP 及 MPLS TE tunnel 的支持
- 在 R1 上完成 MPLS TE 的配置
- 测试数据流的传输过程，观察现象

### 3. 配置命令

R2 的配置如下（所有配置都省去了接口 IP 的配置）：

```
ip cef
mpls traffic-eng tunnels
mpls label range 200 299
!
router ospf 1
router-id 2.2.2.2
network 2.2.2.2 0.0.0.0 area 0
```

```

network 10.1.12.2 0.0.0.0 area 0
network 10.1.23.2 0.0.0.0 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface fast 0/0
mpls traffic-eng tunnels
ip rsvp bandwidth
interface fast 1/0
mpls traffic-eng tunnels
ip rsvp bandwidth

```

**R3 的配置如下：**

```

ip cef
mpls traffic-eng tunnels
mpls label range 300 399
!
router ospf 1
  router-id 3.3.3.3
  network 3.3.3.3 0.0.0.0 area 0
  network 10.1.34.3 0.0.0.0 area 0
  network 10.1.23.3 0.0.0.0 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface fast 0/0
mpls traffic-eng tunnels
ip rsvp bandwidth
interface fast 1/0
mpls traffic-eng tunnels
ip rsvp bandwidth

```

**R4 的配置如下：**

```

ip cef

```

```

mpls traffic-eng tunnels
mpls label range 400 499
!
router ospf 1
  router-id 4.4.4.4
  network 4.4.4.4 0.0.0.0 area 0
  network 10.1.34.4 0.0.0.0 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface fast 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth

```

R1 的配置非常关键，我们重点来看一下：

**R1 的配置如下：**

```

ip cef
mpls traffic-eng tunnels
mpls label range 100 199
!
router ospf 1
  router-id 1.1.1.1
  network 1.1.1.1 0.0.0.0 area 0
  network 10.1.12.1 0.0.0.0 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface fast 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
!
Interface tunnel0 !! TE tunnel

```

```

ip unnumbered loopback 0
tunnel destination 4.4.4.4
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 7 7          !!tunnel 的建立和保持优先级
tunnel mpls traffic-eng bandwidth 20000      !!带宽要求为 20M
tunnel mpls traffic-eng path-option 10 dynamic  !!使用动态计算的方式建立 tunnel PATH

```

#### 4. 验证及查看

首先我们在 R1 上观察一下：

R1#show ip ospf database

```

Type-10 Opaque Link Area Link States (Area 0)
Link ID      ADV Router    Age      Seq#          Checksum Opaque ID
1.0.0.0      1.1.1.1       366      0x80000005  0x0067EF 0
1.0.0.0      2.2.2.2       697      0x80000001  0x0090AA 0
1.0.0.0      3.3.3.3       1230     0x8000001A  0x002ED2 0
1.0.0.0      4.4.4.4       947      0x80000018  0x008F6C 0
1.0.0.1      3.3.3.3       1230     0x80000018  0x001C01 1
1.0.0.2      3.3.3.3       1230     0x80000011  0x00F741 2

```

上面输出的就是在本环境中，各路由器在 area0 内泛洪的 type10 的 opaque-area LSA。

可以进一步详细的看一下，例如看 R1 自己产生 LSA10：

R1#show ip ospf database opaque-area self-originate

```

OSPF Router with ID (1.1.1.1) (Process ID 1)
      Type-10 Opaque Link Area Link States (Area 0)
LS age: 29
Options: (No TOS-capability, DC)
LS Type: Opaque Area Link
Link State ID: 1.0.0.0
Opaque Type: 1
Opaque ID: 0
Advertising Router: 1.1.1.1
LS Seq Number: 80000001

```

Checksum: 0x6FEB

Length: 132

Fragment number : 0

MPLS TE router ID : 1.1.1.1

Link connected to Broadcast network

Link ID : 10.1.12.2

**!!只有一个接口激活 MPLS TE**

Interface Address : 10.1.12.1

Admin Metric : 1

**!! TE metric , 默认和下面的 IGP metric 相等**

Maximum bandwidth : 12500000

**单位是 Bytes , 乘以 8 也就是 100Mbps**

Maximum reservable bandwidth : 9375000

**!!最大可预留带宽 是 75M**

Number of Priority : 8

Priority 0 : 9375000      Priority 1 : 9375000

Priority 2 : 9375000      Priority 3 : 9375000

Priority 4 : 9375000      Priority 5 : 9375000

Priority 6 : 9375000      Priority 7 : 9375000

Affinity Bit : 0x0

IGP Metric : 1

**!!接口的 OSPF cost**

Number of Links : 1

接下去在 R1 上看一下 Tunnel :

**R1#show mpls traffic-eng tunnels**

Name: R1\_t0 (Tunnel0) Destination: 4.4.4.4

Status:

Admin: up      Oper: up      Path: valid      Signalling: connected

path option 10, type dynamic (Basis for Setup, path weight 3)

Config Parameters:

**Bandwidth: 20000   kbps (Global)   Priority: 7   7   Affinity: 0x0/0xFFFF**

**Metric Type: TE (default)** **!!带宽要求为 20M , 隧道抢占优先级及保持优先级为 7**

AutoRoute: disabled LockDown: disabled Loadshare: 20000 bw-based

auto-bw: disabled

InLabel : -

**OutLabel : FastEthernet0/0, 201** **!! 出站标签 201 , 这是 R2 给的**

RSVP Signalling Info:

Src 1.1.1.1, Dst 4.4.4.4, Tun\_Id 0, Tun\_Instance 2

RSVP Path Info:

My Address: 10.1.12.1

**Explicit Route: 10.1.12.2 10.1.23.2 10.1.23.3 10.1.34.3**

**10.1.34.4 4.4.4.4**

**!!LSP 的路径**

Record Route: NONE

Tspec: ave rate=20000 kbits, burst=1000 bytes, peak rate=20000 kbits

RSVP Resv Info:

Record Route: NONE

Fspec: ave rate=20000 kbits, burst=1000 bytes, peak rate=20000 kbits

History:

Tunnel:

Time since created: 10 minutes, 5 seconds

Time since path change: 9 minutes, 49 seconds

Current LSP:

Uptime: 9 minutes, 33 seconds

Selection: reoptimization

Prior LSP:

ID: path option 10 [1]

Removal Trigger: configuration changed

接下去在 R2 上看一下 Tunnel :

**R2#show mpls traffic-eng tunnels**

LSP Tunnel R1\_t0 is signalled, connection is up

**InLabel : FastEthernet0/0, 201** **!!进站标签是 201**



**OutLabel : FastEthernet1/0, 301                      !!出站标签是 301**

RSVP Signalling Info:

Src 1.1.1.1, Dst 4.4.4.4, Tun\_Id 0, Tun\_Instance 2

RSVP Path Info:

My Address: 10.1.23.2

Explicit Route: 10.1.23.3 10.1.34.3 10.1.34.4 4.4.4.4

Record Route: NONE

Tspec: ave rate=20000 kbits, burst=1000 bytes, peak rate=20000 kbits

RSVP Resv Info:

Record Route: NONE

Fspec: ave rate=20000 kbits, burst=1000 bytes, peak rate=20000 kbits

**现在我们在 R1 上将其去往 4.4.4.4 的流量引到 tunnel，用一个最简单的方法：静态路由。**

```
R1(config)#ip route 4.4.4.4 255.255.255.255 tunnel 0
```

```
R1#show ip route
```

```
S          4.4.4.4 is directly connected, Tunnel0
```

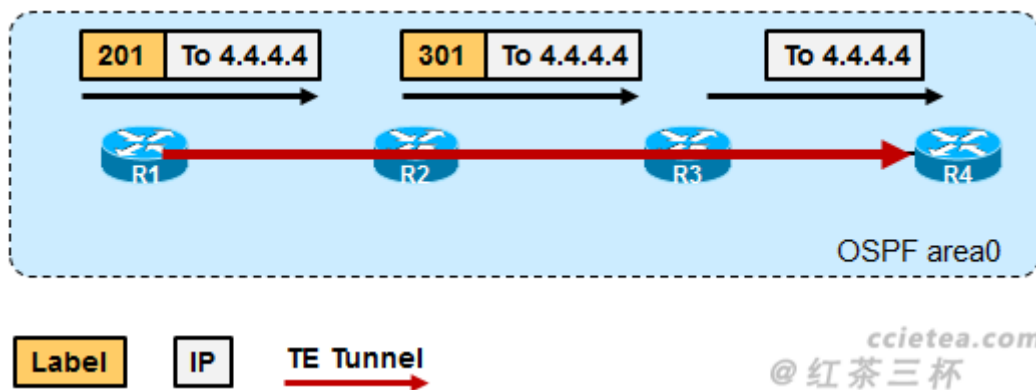
**接下去就可以在 R1 上去测试流量的转发了。**

```
R1#traceroute 4.4.4.4
```

Type escape sequence to abort.

Tracing the route to 4.4.4.4

```
 1 10.1.12.2 [MPLS: Label 201 Exp 0] 132 msec 184 msec 88 msec
 2 10.1.23.3 [MPLS: Label 301 Exp 0] 124 msec 96 msec 104 msec
 3 10.1.34.4 160 msec * 140 msec
```



继续来 show 一下：

### R2#show mpls traffic-eng link-management advertisements

Flooding Status: ready

Configured Areas: 1

IGP Area[1] ID:: ospf area 0

System Information::

Flooding Protocol: OSPF

Header Information::

IGP System ID: 2.2.2.2

MPLS TE Router ID: 2.2.2.2

Flooded Links: 2

**Link ID:: 0**

**!! R2 有两条直连链路参与 MPLS TE，这是第一条，也就是 F0/0 口**

Link IP Address: 10.1.12.2

IGP Neighbor: ID 10.1.12.2

TE metric: 1

IGP metric: 1

Physical Bandwidth: 100000 kbits/sec

**!!接口的物理带宽**

Res. Global BW: 75000 kbits/sec

**!!接口的最大可预留带宽**

Res. Sub BW: 0 kbits/sec

Downstream::

Global Pool Sub Pool **!!接口为各个优先级 tunnel 的带宽池**

Reservable Bandwidth[0]: 75000 0 kbits/sec

Reservable Bandwidth[1]: 75000 0 kbits/sec

```

Reservable Bandwidth[2]:      75000      0 kbits/sec
Reservable Bandwidth[3]:      75000      0 kbits/sec
Reservable Bandwidth[4]:      75000      0 kbits/sec
Reservable Bandwidth[5]:      75000      0 kbits/sec
Reservable Bandwidth[6]:      75000      0 kbits/sec
Reservable Bandwidth[7]:      75000      0 kbits/sec

```

Attribute Flags: 0x00000000

**Link ID:: 1** **!! R2 有两条直连链路参与 MPLS TE , 这是第二条**

```

Link IP Address:      10.1.23.2
IGP Neighbor:         ID 10.1.23.2
TE metric:            1
IGP metric:           1
Physical Bandwidth:   100000 kbits/sec
Res. Global BW:       75000 kbits/sec
Res. Sub BW:          0 kbits/sec
Downstream::

```

	Global Pool	Sub Pool
	-----	-----
Reservable Bandwidth[0]:	75000	0 kbits/sec
Reservable Bandwidth[1]:	75000	0 kbits/sec
Reservable Bandwidth[2]:	75000	0 kbits/sec
Reservable Bandwidth[3]:	75000	0 kbits/sec
Reservable Bandwidth[4]:	75000	0 kbits/sec
Reservable Bandwidth[5]:	75000	0 kbits/sec
Reservable Bandwidth[6]:	75000	0 kbits/sec
Reservable Bandwidth[7]:	55000	0 kbits/sec

Attribute Flags: 0x00000000

**现在我们可以试着修改 R1 的 TE tunnel 配置：**

Interface tunnel0

tunnel mpls traffic-eng priority 5 5

**!!将 TE tunnel 的建立优先级和保持优先级改为 5**

然后再去 R2 上看一下：

## R2#sh mpls tr link-management advertisements

```

Flooding Status:      ready
Configured Areas:     1
IGP Area[1] ID::     ospf area 0

System Information::
  Flooding Protocol:   OSPF
Header Information::
  IGP System ID:       2.2.2.2
  MPLS TE Router ID:   2.2.2.2
  Flooded Links:       2
Link ID:: 0
  Link IP Address:     10.1.12.2
  IGP Neighbor:        ID 10.1.12.2
  TE metric:           1
  IGP metric:          1
  Physical Bandwidth:  100000 kbits/sec
  Res. Global BW:      75000 kbits/sec
  Res. Sub BW:         0 kbits/sec
  Downstream::
    Global Pool  Sub Pool
    -----
    Reservable Bandwidth[0]: 75000 0 kbits/sec
    Reservable Bandwidth[1]: 75000 0 kbits/sec
    Reservable Bandwidth[2]: 75000 0 kbits/sec
    Reservable Bandwidth[3]: 75000 0 kbits/sec
    Reservable Bandwidth[4]: 75000 0 kbits/sec
    Reservable Bandwidth[5]: 75000 0 kbits/sec
    Reservable Bandwidth[6]: 75000 0 kbits/sec
    Reservable Bandwidth[7]: 75000 0 kbits/sec
  Attribute Flags:      0x00000000
Link ID:: 1
  Link IP Address:     10.1.23.2
  
```

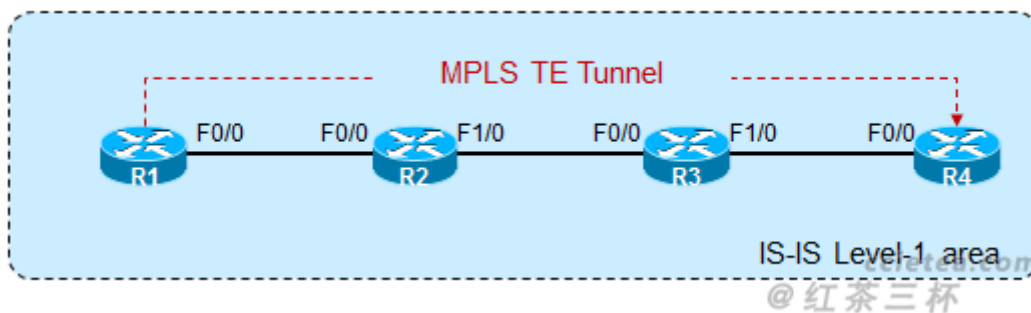
IGP Neighbor: ID 10.1.23.2  
TE metric: 1  
IGP metric: 1  
Physical Bandwidth: 100000 kbits/sec  
Res. Global BW: 75000 kbits/sec  
Res. Sub BW: 0 kbits/sec  
Downstream::

	Global Pool	Sub Pool
	-----	-----
Reservable Bandwidth[0]:	75000	0 kbits/sec
Reservable Bandwidth[1]:	75000	0 kbits/sec
Reservable Bandwidth[2]:	75000	0 kbits/sec
Reservable Bandwidth[3]:	75000	0 kbits/sec
Reservable Bandwidth[4]:	75000	0 kbits/sec
<b>Reservable Bandwidth[5]:</b>	<b>55000</b>	<b>0 kbits/sec</b>
<b>Reservable Bandwidth[6]:</b>	<b>55000</b>	<b>0 kbits/sec</b>
<b>Reservable Bandwidth[7]:</b>	<b>55000</b>	<b>0 kbits/sec</b>

Attribute Flags: 0x00000000

我们看到，低于优先级 5 的 tunnel，可用带宽跟着变成了 75-20=55M。

## 6.4 基础实验 2 ( IS-IS )



## 1. 实验环境

- 设备互联网段为 10.1.xy.0/24，其中 xy 为设备编号，x 小 y 大
- 所有设备的 loopback0 地址空间为 x.x.x.x/32，x 为设备编号
- 全网运行 IS-IS，宣告直连和 loopback 接口
- 在 R1 上建立一个 TE Tunnel，源为自身的 loopback0，目的为 R4 的 4.4.4.4

## 2. 实验步骤

- 完成基本的 IP 配置（配置省略）
- 所有路由器运行 IS-IS
- 全局激活 MPLS TE tunnel，并设定 MPLS 标签空间
- 为每台路由器的 OSPF 激活 MPLS TE 的扩展，并且手工设置用于 MPLS TE 的 RouterID
- 每台路由器的接口都激活 RSVP 及 MPLS TE tunnel 的支持
- 在 R1 上完成 MPLS TE tunnel 的配置
- 测试数据流的传输过程，观察现象

## 3. 配置命令

R2 的配置如下（所有配置都省去了接口 IP 的配置）：

```

ip cef
mpls traffic-eng tunnels
mpls label range 200 299
!
router isis
    net 49.0001.0000.0000.0002.00
    is-type level-1                !!!IS 的类型为 level1
    metric-style wide              !!metric 必须改为 wide 的
    mpls traffic-eng router-id loopback 0    !!设置 MPLS TE routerID
    mpls traffic-eng level-1          !!激活 level1 的 MPLS TE
!
interface fast0/0
    ip router isis
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface fast1/0
    
```

```
ip router isis
mpls traffic-eng tunnels
ip rsvp bandwidth
```

R3 的配置如下：

```
Ip cef
mpls traffic-eng tunnels
mpls label range 300 399
!
router isis
net 49.0001.0000.0000.0003.00
is-type level-1
metric-style wide
mpls traffic-eng router-id loopback 0
mpls traffic-eng level-1
!
interface fast0/0
ip router isis
mpls traffic-eng tunnels
ip rsvp bandwidth
interface fast1/0
ip router isis
mpls traffic-eng tunnels
ip rsvp bandwidth
```

R4 的配置如下：

```
Ip cef
mpls traffic-eng tunnels
mpls label range 400 499
!
router isis
net 49.0001.0000.0000.0004.00
is-type level-1
metric-style wide
```

```

mpls traffic-eng router-id loopback 0
mpls traffic-eng level-1
!
interface fast0/0
    ip router isis
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface loopback0
    ip router isis

```

最后我们来完成 R1 的配置：

R1 的配置如下：

```

ip cef
mpls traffic-eng tunnels
mpls label range 100 199
!
router isis
    net 49.0001.0000.0000.0001.00
    is-type level-1
    metric-style wide
    mpls traffic-eng router-id loopback 0
    mpls traffic-eng level-1
!
interface fast0/0
    ip router isis
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface loopback0
    ip router isis
!
Interface tunnel0                                !! TE tunnel
    ip unnumbered loopback 0

```



```

tunnel destination 4.4.4.4
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng priority 7 7           !!tunnel 的建立和保持优先级
tunnel mpls traffic-eng bandwidth 20000       !!带宽要求为 20M
tunnel mpls traffic-eng path-option 10 dynamic !!使用动态计算的方式建立 tunnel PATH

```

完成配置后，Tunnel 就起来了。

#### R1#show isis database verbose R1.00-00

IS-IS Level-1 LSP R1.00-00

LSPID	LSP Seq Num	LSP Checksum	LSP Holdtime	ATT/P/OL
R1.00-00	* 0x0000000C	0x8973	772	0/0/0

Area Address: 49.0001  
 NLPID: 0xCC  
 Hostname: R1  
 Router ID: 1.1.1.1  
 IP Address: 1.1.1.1  
 Metric: 10 IP 10.1.12.0/24  
 Metric: 10 IP 1.1.1.1/32  
 Metric: 10 IS-Extended R2.01  
 Affinity: 0x00000000  
 Interface IP Address: 10.1.12.1  
 Physical BW: 100000 kbits/sec  
 Reservable Global Pool BW: 75000 kbits/sec  
 Global Pool BW Unreserved:

[0]: 75000 kbits/sec,	[1]: 75000 kbits/sec
[2]: 75000 kbits/sec,	[3]: 75000 kbits/sec
[4]: 75000 kbits/sec,	[5]: 75000 kbits/sec
[6]: 75000 kbits/sec,	<b>[7]: 55000 kbits/sec</b>

Admin. Weight: 1

## R1#show mpls traffic-eng tunnels

```

Name: R1_t0                                (Tunnel0) Destination: 4.4.4.4
Status:
  Admin: up          Oper: up          Path: valid          Signalling: connected

  path option 10, type dynamic (Basis for Setup, path weight 3)

Config Parameters:
  Bandwidth: 20000    kbps (Global)  Priority: 7   7   Affinity: 0x0/0xFFFF
  Metric Type: TE (default)
  AutoRoute: disabled LockDown: disabled Loadshare: 20000    bw-based
  auto-bw: disabled

InLabel  : -
OutLabel : FastEthernet0/0, 200
RSVP Signalling Info:
  Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 0, Tun_Instance 39
RSVP Path Info:
  My Address: 10.1.12.1
Explicit Route: 10.1.12.2 10.1.23.2 10.1.23.3 10.1.34.3
10.1.34.4 4.4.4.4
  Record Route: NONE
  Tspec: ave rate=20000 kbits, burst=1000 bytes, peak rate=20000 kbits
RSVP Resv Info:
  Record Route: NONE
  Fspec: ave rate=20000 kbits, burst=1000 bytes, peak rate=20000 kbits
History:
  Tunnel:
    Time since created: 5 hours, 7 minutes
    Time since path change: 12 seconds
  Current LSP:
    Uptime: 12 seconds
  
```

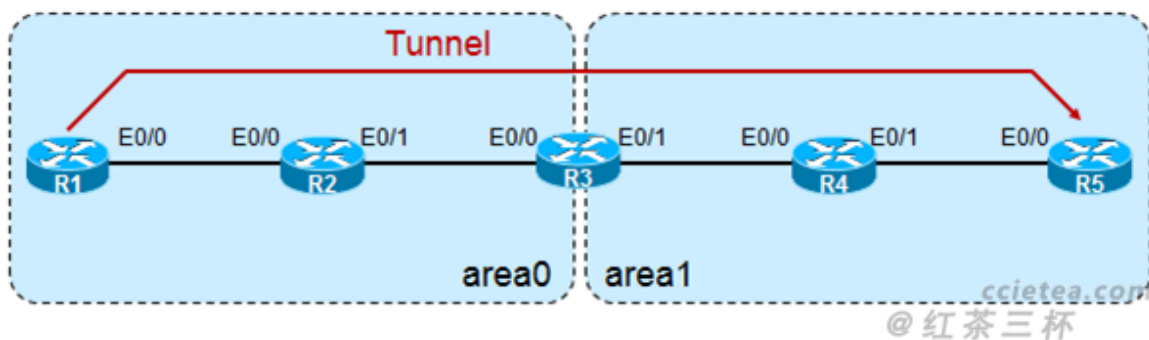
Selection: reoptimization

Prior LSP:

ID: path option 10 [36]

Removal Trigger: configuration changed

## 6.5 跨区域 TE Tunnel ( OSPF )



### 1. 实验环境

- 设备互联网段为 10.1.xy.0/24，其中 xy 为设备编号，x 小 y 大
- 所有设备的 loopback0 地址空间为 x.x.x.x/32，x 为设备编号
- 全网运行 OSPF，宣告直连和 loopback 接口
- R3 为 ABR，连接 area0 及 area1
- 在 R1 上建立一个 TE Tunnel，源为自身的 loopback0，目的为 R4 的 4.4.4.4

### 2. 实验步骤

- 完成基本的 IP 配置（配置省略）
- 所有路由器运行 OSPF
- 全局激活 MPLS TE tunnel，并设定 MPLS 标签空间
- 为每台路由器的 OSPF 激活 MPLS TE 的扩展，并且手工设置用于 MPLS TE 的 RouterID
- 每台路由器的接口都激活 RSVP 及 MPLS TE tunnel 的支持
- 在 R1 上完成 MPLS TE 的配置
- 测试数据流的传输过程，观察现象

### 3. 配置命令

R2 的配置如下 ( 所有配置都省去了接口 IP 的配置 ):

```

ip cef
mpls traffic-eng tunnels
mpls label range 200 299
!
router ospf 1
  router-id 2.2.2.2
  network 10.1.12.2 0.0.0.0 area 0
  network 10.1.23.2 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth0 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth0 1/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth

```

R3 的配置如下 ( 注意 , area0 和 area1 都要激活 MPLS TE ):

```

ip cef
mpls traffic-eng tunnels
mpls label range 300 399
!
router ospf 1
  router-id 3.3.3.3
  network 10.1.34.3 0.0.0.0 area 1
  network 10.1.23.2 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0

```

```

mpls traffic-eng area 1                                !!注意 area0 和 area1 都要激活 MPLS TE
!
interface eth0 0/0
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface eth0 1/0
    mpls traffic-eng tunnels
    ip rsvp bandwidth

```

#### R4 的配置如下

```

Ip cef
mpls traffic-eng tunnels
mpls label range 400 499
!
router ospf 1
    router-id 4.4.4.4
    network 10.1.34.4 0.0.0.0 area 1
    network 10.1.45.4 0.0.0.0 area 1
    mpls traffic-eng router-id loopback0
    mpls traffic-eng area 1
!
interface eth0 0/0
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface eth0 1/0
    mpls traffic-eng tunnels
    ip rsvp bandwidth

```

#### R5 的配置如下

```

Ip cef
mpls traffic-eng tunnels
mpls label range 500 599
!

```

```
router ospf 1
  router-id 5.5.5.5
  network 5.5.5.5 0.0.0.0 area 1
  network 10.1.45.5 0.0.0.0 area 1
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 1
!
interface eth0 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
```

接下去在 R1 上配置：

#### Ip explicit R1toR5

```
next-address 10.1.12.2           !!本区域内直接用 strict 下一跳
next-address 10.1.23.3
next-address loose 10.1.34.4      !!跨区域的话必须用 loose 下一跳
next-address loose 10.1.45.5     !!这条就可要可不要了，你懂的
!
```

#### interface Tunnel0

```
ip unnumbered Loopback0
tunnel destination 5.5.5.5
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 50000
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng path-option 10 explicit name R1toR5
```

#### R1#show mpls traffic-eng tunnels

```
Name: R1_t0                      (Tunnel0) Destination: 5.5.5.5
Status:
  Admin: up      Oper: up      Path: valid      Signalling: connected
```

path option 10, type explicit test (Basis for Setup, path weight 20)

#### Config Parameters:

Bandwidth: 0 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF

Metric Type: TE (default)

AutoRoute: disabled LockDown: disabled Loadshare: 0 bw-based

auto-bw: disabled

#### Active Path Option Parameters:

State: explicit path option 10 is active

BandwidthOverride: disabled LockDown: disabled Verbatim: disabled

InLabel : -

OutLabel : Ethernet0/0, 201

#### RSVP Signalling Info:

Src 1.1.1.1, Dst 5.5.5.5, Tun\_Id 0, Tun\_Instance 9

#### RSVP Path Info:

My Address: 10.1.12.1

Explicit Route: **10.1.12.2 10.1.23.2 10.1.23.3 10.1.34.4\***  
**10.1.45.5\***

Record Route:

Tspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

#### RSVP Resv Info:

Record Route: 10.1.23.2 10.1.34.3 10.1.45.4 10.1.45.5

Fspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

#### History:

##### Tunnel:

Time since created: 11 minutes, 42 seconds

Time since path change: 10 seconds

Number of LSP IDs (Tun\_Instances) used: 9

##### Current LSP:

Uptime: 10 seconds

Selection: reoptimization

Prior LSP:

ID: path option 10 [8]

Removal Trigger: configuration changed

## 7 MPLS TE 路径优化

### 7.1 TE tunnel reoptimize

- **周期性 reoptimize**

在 CISCO IOS 中，一条 TE tunnel 的 reoptimize 默认每 1 小时进行一次。

```
router(config)#mpls traffic-eng reoptimize timers frequency ?
```

如果该时间指定为 0，那么周期性 reoptimize 将会在路由器上的所有 TE tunnel 中被关闭。当然，你可以为单条 TE tunnel 关闭 reoptimize，命令如下：

```
Router(config-if)# mpls traffic-eng path-option x {dynamic | explicit name y } [lockdown]
```

使用 lockdown 关键字（注意上述配置是在 TE tunnel 口中配置）。

- **事件导致的 reoptimize**

缺省情况下，CISCO IOS 不会因为网络中的一条链路重新可以被一条 TE tunnel 所使用的时候而触发重新最优化（例如可用带宽发生变化，这个变化被 IGP 协议泛洪出来了），但是可以激活这个操作。要在一条链路在 MPLS TE 中变为可操作的时候启用重新最优化，使用下面的命令：

```
mpls traffic-eng reoptimize events link-up
```

经实验验证有效。

可用 debug mpls traffic-eng tunnels events 及 debug mpls traffic-eng tunnels reoptimize 查看。

- **手工重新 reoptimize**

```
Router# mpls traffic-eng reoptimize
```

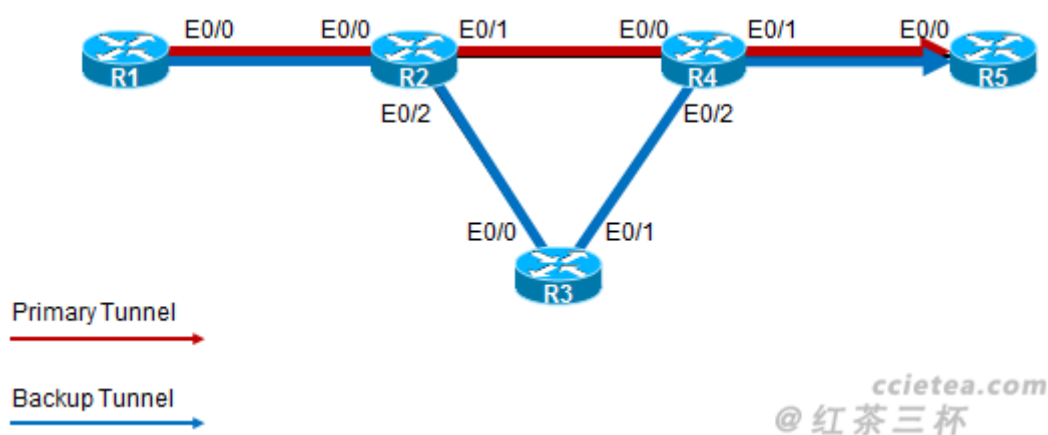
手工重新最优化



## 7.2 Path Protection

### 7.2.1 基本概念

### 7.2.2 基础实验



#### 1. 实验环境

- 设备互联网段为 10.1.xy.0/24，其中 xy 为设备编号，x 小 y 大
- 所有设备的 loopback0 地址空间为 x.x.x.x/32，x 为设备编号
- 全网运行 OSPF，宣告直连和 loopback 接口
- 在 R1 上建立一个 TE Tunnel，路径为 R1-R2-R4-R5，再创建另一个备份 Tunnel 路径为 R1-R2-R3-R4-R5，这条路径作为前者的备份 Tunnel。

#### 2. 设备配置

R1 的配置如下（暂时不配置 tunnel）：

```
ip cef
mpls traffic-eng tunnels
mpls label range 100 199
!
router ospf 1
```

```

router-id 1.1.1.1
network 1.1.1.1 0.0.0.0 area 0
network 10.1.12.1 0.0.0.0 area 0
mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface eth 0/0
mpls traffic-eng tunnels
ip rsvp bandwidth

```

R2 的配置如下

```

ip cef
mpls traffic-eng tunnels
mpls label range 200 299
!
router ospf 1
  router-id 2.2.2.2
  network 10.1.12.2 0.0.0.0 area 0
  network 10.1.23.2 0.0.0.0 area 0
  network 10.1.24.2 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/1
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/2
  mpls traffic-eng tunnels
  ip rsvp bandwidth

```

R3 的配置如下

```

ip cef
mpls traffic-eng tunnels
mpls label range 300 399
!
router ospf 1
  router-id 3.3.3.3
  network 10.1.23.3 0.0.0.0 area 0
  network 10.1.34.3 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/1
  mpls traffic-eng tunnels
  ip rsvp bandwidth

```

R4 的配置如下

```

ip cef
mpls traffic-eng tunnels
mpls label range 400 499
!
router ospf 1
  router-id 4.4.4.4
  network 10.1.45.4 0.0.0.0 area 0
  network 10.1.34.3 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth

```

```
interface eth 0/1
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface eth 0/2
    mpls traffic-eng tunnels
    ip rsvp bandwidth
```

R5 的配置如下

```
ip cef
mpls traffic-eng tunnels
mpls label range 500 599
!
router ospf 1
    router-id 5.5.5.5
    network 10.1.45.5 0.0.0.0 area 0
    network 5.5.5.5 0.0.0.0 area 0
    mpls traffic-eng router-id loopback0
    mpls traffic-eng area 0
!
interface eth 0/0
    mpls traffic-eng tunnels
    ip rsvp bandwidth
```

现在在 R1 上建立 tunnel :

```
ip explicit-path name R2R4 enable                !! Primary tunnel
    next-address 10.1.12.2
    next-address 10.1.24.4
!
ip explicit-path name R2R3R4 enable              !! Backup tunnel
    next-address 10.1.12.2
    next-address 10.1.23.3
    next-address 10.1.34.4
!
```

```
interface Tunnel0
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 5.5.5.5
 tunnel mpls traffic-eng path-option 10 explicit name R2R4
 tunnel mpls traffic-eng path-option protect 10 explicit name R2R3R4
!
ip route 5.5.5.5 255.255.255.255 Tunnel0
```

注意，这里主 tunnel 强烈建议是 explicit 的，如果这里主 LSP 是 dynamic 的，那么当主 LSP DOWN 掉后（例如 shutdown 掉 R2 的 E0/1 口），将切换到备份 LSP 上来。但是如果这时候 R2 的 E0/1 口恢复了，经过在 CISCO IOS 路由器上测试，发现流量无法切回主 LSP。用 explicit path 则没这个问题。

#### R1#show mpls traffic-eng tunnels protection

```
R1_t0
LSP Head, Tunnel0, Admin: up, Oper: up
Src 1.1.1.1, Dest 5.5.5.5, Instance 30
Fast Reroute Protection: None
Path Protection: 2 Common Link(s), 2 Common Node(s)
Link Sharing Detail:
  P2P Links:          0
  Multiaccess Links:  2
    Both interfaces:   2
    1 interface:       0
    0 interfaces:      0 (only media is shared)

Primary lsp path:10.1.12.1 10.1.12.2
10.1.24.2 10.1.24.4
10.1.45.4 10.1.45.5
5.5.5.5

Protect lsp path:10.1.12.1 10.1.12.2
10.1.23.2 10.1.23.3
10.1.34.3 10.1.34.4
10.1.45.4 10.1.45.5
```

### 5.5.5.5

Path Protect Parameters:

Bandwidth: 0 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF

Metric Type: TE (default)

InLabel : -

OutLabel : Ethernet0/0, 204

RSVP Signalling Info:

Src 1.1.1.1, Dst 5.5.5.5, Tun\_Id 0, Tun\_Instance 31

RSVP Path Info:

My Address: 10.1.12.1

Explicit Route: 10.1.12.2 10.1.23.2 10.1.23.3 10.1.34.3

10.1.34.4 10.1.45.4 10.1.45.5 5.5.5.5

Record Route: NONE

Tspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

RSVP Resv Info:

Record Route: NONE

Fspec: ave rate=0 kbits, burst=1000 bytes, peak rate=0 kbits

## 7.3 Fast Reroute

### 7.3.1 FRR 基本概念

- 如果 link 或 node 出现故障，流量会因为故障而被重新进行路由，如果重新路由需要花上几秒钟的话，由于链路的高性能会导致有大量指向该故障点的流量都会被丢弃，对于某些业务来说可能会导致业务中断。
- **Fast Reroute (FRR)** 是一种 link 或 node 保护机制。允许在 link 或 node 出现问题的时候，头端路由器 rerouting，并且使用一条临时的、预先建立好的路径绕过故障点。

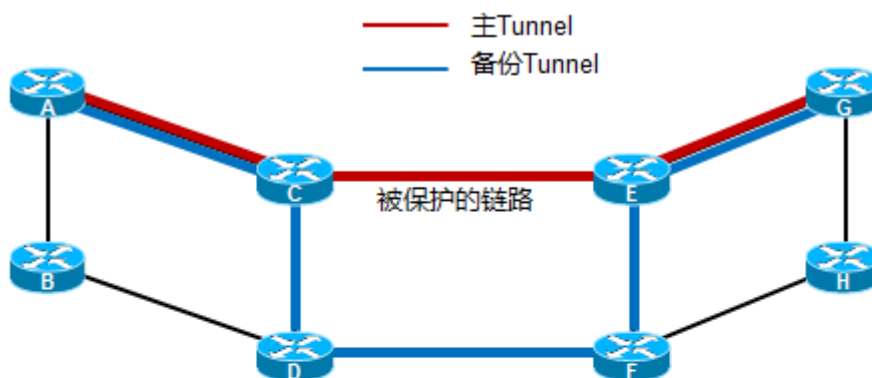
按保护对象可分为：

- Link Protection (链路保护)

- Node Protection ( 节点保护 )

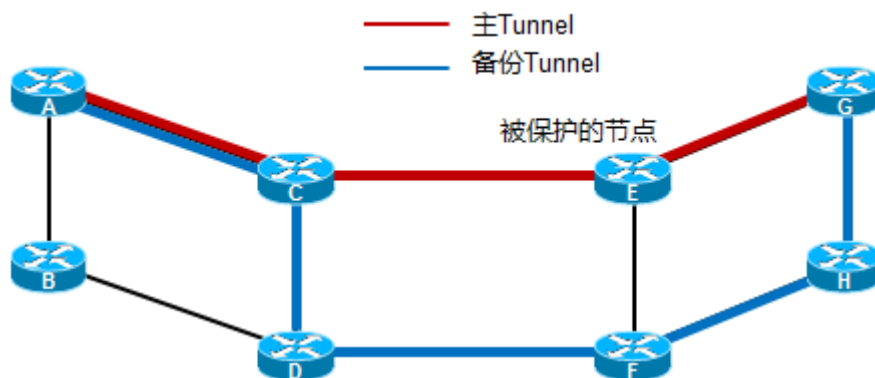
## 7.3.2 术语

- PLR ( Point of local repair ) : 本地修理点，这是备份隧道的首端
- MP ( Merge Point ) : 汇合点，这是备份隧道的尾端，也可以理解为主隧道和备份隧道交接点
- NHop ( Next Hop Router ) : 下一跳路由器，PLR 的下一跳路由器
- NNHop ( Next-Next Hop Router ) : 下下一跳路由器，PLR 下一跳路由器的下一跳



- C是备份Tunnel的PLR ( Point of local repair )
- E是备份Tunnel的MP ( Merge Point )
- E是C的NHop ( Next-hop )
- G是C的NNHop ( Next-Next-Hop )

ccietea.com  
@ 红茶三杯

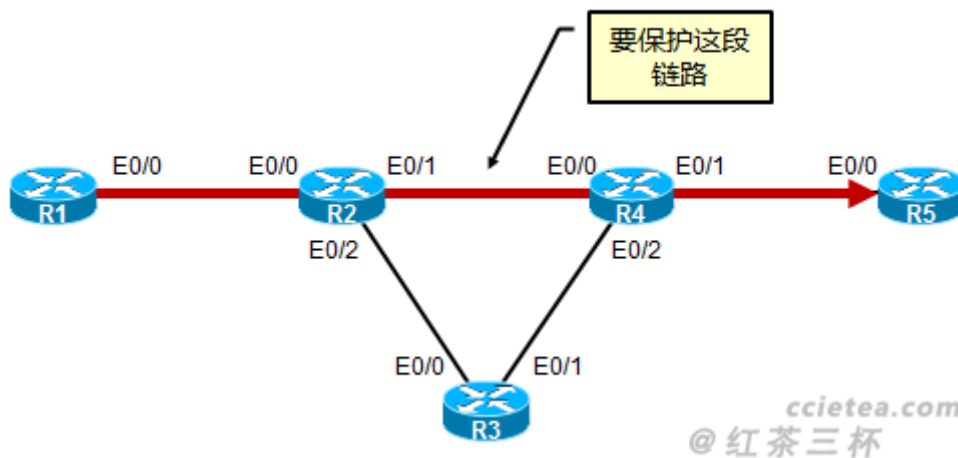


- C是备份Tunnel的PLR ( Point of local repair )
- G是备份Tunnel的MP ( Merge Point )
- E是C的NHop ( Next-hop )
- G是C的NNHop ( Next-Next-Hop )

ccietea.com

@ 红茶三杯

### 7.3.3 基础实验 ( LinkProtection )



ccietea.com

@ 红茶三杯

#### 1. 环境描述

- 设备互联网段为 10.1.xy.0/24，其中 xy 为设备编号，x 小 y 大
- 所有设备的 loopback0 地址空间为 x.x.x.x/32，x 为设备编号
- 全网运行 OSPF，宣告直连和 loopback 接口
- 在 R1 上建立一个 TE Tunnel，源为自身的 loopback0，目的为 R5 的 5.5.5.5



## 2. 实验步骤

- 完成基本的 IP 配置（配置省略）
- 所有路由器运行 OSPF
- 全局激活 MPLS TE tunnel，并设定 MPLS 标签空间
- 为每台路由器的 OSPF 激活 MPLS TE 的扩展，并且手工设置用于 MPLS TE 的 RouterID
- 每台路由器的接口都激活 RSVP 及 MPLS TE tunnel 的支持
- 在 R1 上完成 MPLS TE 的配置
- Shutdown 掉 R2 的 e0/1 口，在 R1 上抓取 debug 信息
- 在 R1 上激活 fast-reroute 特性，再观察现象
- 在 R2 上配置路径保护，再观察现象

## 3. 设备配置

R1 的配置如下：

```

ip cef
mpls traffic-eng tunnels
mpls label range 100 199
!
router ospf 1
  router-id 1.1.1.1
  network 1.1.1.1 0.0.0.0 area 0
  network 10.1.12.1 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
!
ip explicit-path name R2R4R5 enable
  next-address 10.1.12.2
  next-address 10.1.24.4
  next-address 10.1.45.5
!
  
```

```
interface Tunnel0
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 5.5.5.5
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 2000
 tunnel mpls traffic-eng path-option 10 explicit name R2R4R5
 no routing dynamic
```

R2 的配置如下：

```
ip cef
 mpls traffic-eng tunnels
 mpls label range 200 299
!
router ospf 1
 router-id 2.2.2.2
 network 2.2.2.2 0.0.0.0 area0
 network 10.1.12.2 0.0.0.0 area 0
 network 10.1.23.2 0.0.0.0 area 0
 network 10.1.24.2 0.0.0.0 area 0
 mpls traffic-eng router-id loopback0
 mpls traffic-eng area 0
!
interface eth 0/0
 mpls traffic-eng tunnels
 ip rsvp bandwidth
interface eth 0/1
 mpls traffic-eng tunnels
 ip rsvp bandwidth
interface eth 0/2
 mpls traffic-eng tunnels
```

**ip rsvp bandwidth**

R3 的配置如下：

```

ip cef
mpls traffic-eng tunnels
mpls label range 300 399
!
router ospf 1
  router-id 3.3.3.3
  network 10.1.23.3 0.0.0.0 area 0
  network 10.1.34.3 0.0.0.0 area 0
  mpls traffic-eng router-id loopback0
  mpls traffic-eng area 0
!
interface eth 0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth
interface eth 0/1
  mpls traffic-eng tunnels
  ip rsvp bandwidth

```

R4 的配置如下：

```

ip cef
mpls traffic-eng tunnels
mpls label range 400 499
!
router ospf 1
  router-id 4.4.4.4
  network 4.4.4.4 0.0.0.0 area 0
  network 10.1.34.4 0.0.0.0 area 0
  network 10.1.24.4 0.0.0.0 area 0
  network 10.1.45.4 0.0.0.0 area 0

```

```

mpls traffic-eng router-id loopback0
mpls traffic-eng area 0
!
interface eth 0/0
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface eth 0/1
    mpls traffic-eng tunnels
    ip rsvp bandwidth
interface eth 0/2
    mpls traffic-eng tunnels
    ip rsvp bandwidth

```

R5 的配置如下：

```

ip cef
mpls traffic-eng tunnels
mpls label range 500 599
!
router ospf 1
    router-id 5.5.5.5
    network 5.5.5.5 0.0.0.0 area 0
    network 10.1.45.5 0.0.0.0 area 0
    mpls traffic-eng router-id loopback0
    mpls traffic-eng area 0
!
interface eth 0/0
    mpls traffic-eng tunnels
    ip rsvp bandwidth

```

#### 4. 现象观察

初始化配置如上，在现在的环境中，R1 并没有激活 fasterroute，而且 R2 上也没有去部署保护性的 tunnel。首先在 R1 上 debug ip rsvp dump-messages

然后 shutdown R2 的 e0/1 口：

R1#deb ip rsvp dump-messages

```
*Aug 18 04:37:06.239: Incoming PathError:    !!R2 发给 R1 的 patherror 消息
*Aug 18 04:37:06.239:   version:1 flags:0000 cksum:7B25 ttl:255 reserved:0 length:132
*Aug 18 04:37:06.239:   SESSION                               type 7 length 16:
*Aug 18 04:37:06.239:   Tun Dest:   5.5.5.5  Tun ID: 0  Ext Tun ID: 1.1.1.1
*Aug 18 04:37:06.239:   ERROR_SPEC                               type 1 length 12:
*Aug 18 04:37:06.239:   Error Node: 10.1.12.2
*Aug 18 04:37:06.239:   Error Code: 24 (Routing Problem)
*Aug 18 04:37:06.239:   Error Value: 0x5  (No route available toward destination)
*Aug 18 04:37:06.239:   Flags: 0x0
*Aug 18 04:37:06.239:   SENDER_TEMPLATE                         type 7 length 12:
*Aug 18 04:37:06.239:   Tun Sender: 1.1.1.1  LSP ID: 20
*Aug 18 04:37:06.239:   SENDER_TSPEC                           type 2 length 36:
*Aug 18 04:37:06.239:   version=0, length in words=7
*Aug 18 04:37:06.239:   Token bucket fragment (service_id=1, length=6 words
*Aug 18 04:37:06.239:     parameter id=127, flags=0, parameter length=5
*Aug 18 04:37:06.239:     average rate=250000 bytes/sec, burst depth=1000 bytes
*Aug 18 04:37:06.239:     peak rate   =250000 bytes/sec
*Aug 18 04:37:06.239:     min unit=0 bytes, max pkt size=2147483647 bytes
*Aug 18 04:37:06.239:   ADSPEC                                 type 2 length 48:
*Aug 18 04:37:06.239:   version=0  length in words=10
*Aug 18 04:37:06.239:   General Parameters  break bit=0  service length=8
*Aug 18 04:37:06.239:                                   IS Hops:1
*Aug 18 04:37:06.239:                                   Minimum Path Bandwidth (bytes/sec):1250000
*Aug 18 04:37:06.239:                                   Path Latency (microseconds):0
*Aug 18 04:37:06.239:                                   Path MTU:1500
*Aug 18 04:37:06.239:   Controlled Load Service  break bit=0  service length=0
*Aug 18 04:37:06.239:
*Aug 18 04:37:06.239: Outgoing PathTear:    !!拆除 tunnel
*Aug 18 04:37:06.239:   version:1 flags:0000 cksum:DEB3 ttl:255 reserved:0 length:132
*Aug 18 04:37:06.239:   SESSION                               type 7 length 16:
```

```
*Aug 18 04:37:06.239: Tun Dest: 5.5.5.5 Tun ID: 0 Ext Tun ID: 1.1.1.1
*Aug 18 04:37:06.239: HOP type 1 length 12:
*Aug 18 04:37:06.239: Hop Addr: 10.1.12.1 LIH: 0x02000403
*Aug 18 04:37:06.239: SENDER_TEMPLATE type 7 length 12:
*Aug 18 04:37:06.239: Tun Sender: 1.1.1.1 LSP ID: 20
.....
*Aug 18 04:37:06.239:
*Aug 18 04:37:06.243: Incoming ResvTear:
*Aug 18 04:37:06.243: version:1 flags:0000 cksum:FFAA ttl:255 reserved:0 length:92
*Aug 18 04:37:06.243: SESSION type 7 length 16:
*Aug 18 04:37:06.243: Tun Dest: 5.5.5.5 Tun ID: 0 Ext Tun ID: 1.1.1.1
*Aug 18 04:37:06.243: HOP type 1 length 12:
*Aug 18 04:37:06.243: Hop Addr: 10.1.12.2 LIH: 0x02000403
*Aug 18 04:37:06.243: STYLE type 1 length 8 :
*Aug 18 04:37:06.243: Shared-Explicit (SE)
*Aug 18 04:37:06.243: FLOWSPEC type 2 length 36:
*Aug 18 04:37:06.243: version = 0 length in words = 7
*Aug 18 04:37:06.243: service id = 5, service length = 6
*Aug 18 04:37:06.243: tspec parameter id = 127, flags = 0,length = 5
*Aug 18 04:37:06.243: average rate = 250000 bytes/sec, burst depth = 1000 bytes
*Aug 18 04:37:06.243: peak rate = 250000 bytes/sec
*Aug 18 04:37:06.243: min unit = 0 bytes,max pkt size = 1500 bytes
*Aug 18 04:37:06.243: FILTER_SPEC type 7 length 12:
*Aug 18 04:37:06.243: Tun Sender: 1.1.1.1, LSP ID: 20
*Aug 18 04:37:06.243:
R1#
*Aug 18 04:37:24.735: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to
down
```

现在激活 R1 的 fast-reroute 特性，修改 R1 的 Tunnel 口配置如下：

```
interface Tunnel0
 ip unnumbered Loopback0
```

```
tunnel mode mpls traffic-eng
tunnel destination 5.5.5.5
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 2000
tunnel mpls traffic-eng path-option 10 explicit name R2R4R5
tunnel mpls traffic-eng fast-reroute
```

### R1 上 debug 信息如下：

```
*Aug 18 09:06:02.699: Outgoing Path:    !! R1 发送出去的 path 消息发生了变化
*Aug 18 09:06:02.699:  version:1 flags:0000 cksum:5878 ttl:255 reserved:0 length:216
*Aug 18 09:06:02.699:  SESSION                      type 7 length 16:
*Aug 18 09:06:02.699:  Tun Dest:    5.5.5.5  Tun ID: 0  Ext Tun ID: 1.1.1.1
*Aug 18 09:06:02.699:  HOP                      type 1 length 12:
*Aug 18 09:06:02.699:  Hop Addr: 10.1.12.1 LIH: 0x20000403
*Aug 18 09:06:02.699:  TIME_VALUES              type 1 length 8 :
*Aug 18 09:06:02.699:  Refresh Period (msec): 30000
*Aug 18 09:06:02.699:  EXPLICIT_ROUTE          type 1 length 52:
*Aug 18 09:06:02.699:  10.1.12.2 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:02.699:  10.1.24.2 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:02.699:  10.1.24.4 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:02.699:  10.1.45.4 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:02.699:  10.1.45.5 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:02.699:  5.5.5.5 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:02.699:  LABEL_REQUEST           type 1 length 8 :
*Aug 18 09:06:02.699:  Layer 3 protocol ID: 2048
*Aug 18 09:06:02.699:  SESSION_ATTRIBUTE      type 7 length 16:
*Aug 18 09:06:02.699:  Setup Prio: 7, Holding Prio: 7
*Aug 18 09:06:02.699:  Flags: (0x7) Local Prot desired, Label Recording, SE Style    !! Flag 位里有体现了
*Aug 18 09:06:02.699:  Session Name: R1_t0
*Aug 18 09:06:02.699:  SENDER_TEMPLATE        type 7 length 12:
```

```
*Aug 18 09:06:02.699: Tun Sender: 1.1.1.1 LSP ID: 247
*Aug 18 09:06:02.699: SENDER_TSPEC          type 2 length 36:
*Aug 18 09:06:02.699: version=0, length in words=7
*Aug 18 09:06:02.699: Token bucket fragment (service_id=1, length=6 words
*Aug 18 09:06:02.699:     parameter id=127, flags=0, parameter length=5
*Aug 18 09:06:02.699:     average rate=250000 bytes/sec, burst depth=1000 bytes
*Aug 18 09:06:02.699:     peak rate    =250000 bytes/sec
*Aug 18 09:06:02.699:     min unit=0 bytes, max pkt size=2147483647 bytes
*Aug 18 09:06:02.699: ADSPEC          type 2 length 48:
*Aug 18 09:06:02.699: version=0 length in words=10
*Aug 18 09:06:02.699: General Parameters break bit=0 service length=8
*Aug 18 09:06:02.699:                               IS Hops:1
*Aug 18 09:06:02.699:                               Minimum Path Bandwidth (bytes/sec):1250000
*Aug 18 09:06:02.699:                               Path Latency (microseconds):0
*Aug 18 09:06:02.699:                               Path MTU:1500
*Aug 18 09:06:02.699: Controlled Load Service break bit=0 service length=0
*Aug 18 09:06:02.699:
*Aug 18 09:06:07.919: Incoming PathError:
*Aug 18 09:06:07.919: version:1 flags:0000 cksum:7A42 ttl:255 reserved:0 length:132
*Aug 18 09:06:07.919: SESSION          type 7 length 16:
*Aug 18 09:06:07.919: Tun Dest: 5.5.5.5 Tun ID: 0 Ext Tun ID: 1.1.1.1
*Aug 18 09:06:07.919: ERROR_SPEC       type 1 length 12:
*Aug 18 09:06:07.919: Error Node: 10.1.12.2
*Aug 18 09:06:07.919: Error Code: 24 (Routing Problem)
*Aug 18 09:06:07.919: Error Value: 0x5 (No route available toward destination)
*Aug 18 09:06:07.919: Flags: 0x0
*Aug 18 09:06:07.919: SENDER_TEMPLATE  type 7 length 12:
*Aug 18 09:06:07.919: Tun Sender: 1.1.1.1 LSP ID: 247
*Aug 18 09:06:07.919: SENDER_TSPEC     type 2 length 36:
*Aug 18 09:06:07.919: version=0, length in words=7
*Aug 18 09:06:07.919: Token bucket fragment (service_id=1, length=6 words
*Aug 18 09:06:07.919:     parameter id=127, flags=0, parameter length=5
```



```
*Aug 18 09:06:07.919:      average rate=250000 bytes/sec, burst depth=1000 bytes
*Aug 18 09:06:07.919:      peak rate      =250000 bytes/sec
*Aug 18 09:06:07.919:      min unit=0 bytes, max pkt size=2147483647 bytes
*Aug 18 09:06:07.919: ADSPEC                      type 2 length 48:
*Aug 18 09:06:07.919: version=0  length in words=10
*Aug 18 09:06:07.919: General Parameters  break bit=0  service length=8
*Aug 18 09:06:07.919:                      IS Hops:1
*Aug 18 09:06:07.919:                      Minimum Path Bandwidth (bytes/sec):1250000
*Aug 18 09:06:07.919:                      Path Latency (microseconds):0
*Aug 18 09:06:07.919:                      Path MTU:1500
*Aug 18 09:06:07.919: Controlled Load Service  break bit=0  service length=0
*Aug 18 09:06:07.919:
*Aug 18 09:06:07.919: Incoming PROXY_PATH:
*Aug 18 09:06:07.919:  version:1 flags:0000 cksum:0000 ttl:255 reserved:0 length:232
*Aug 18 09:06:07.919: SESSION                      type 7 length 16:
*Aug 18 09:06:07.919:  Tun Dest:  5.5.5.5  Tun ID: 0  Ext Tun ID: 1.1.1.1
*Aug 18 09:06:07.919: HOP                          type 1 length 12:
*Aug 18 09:06:07.919:  Hop Addr: 127.0.0.1 LIH: 0x00000000
*Aug 18 09:06:07.919: TIME_VALUES                  type 1 length 8 :
*Aug 18 09:06:07.919:  Refresh Period (msec): 30000
*Aug 18 09:06:07.919: EXPLICIT_ROUTE              type 1 length 68:
*Aug 18 09:06:07.919:  1.1.1.1 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919:  10.1.12.1 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919:  10.1.12.2 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919:  10.1.24.2 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919:  10.1.24.4 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919:  10.1.45.4 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919:  10.1.45.5 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919:  5.5.5.5 (Strict IPv4 Prefix, 8 bytes, /32)
*Aug 18 09:06:07.919: SESSION_ATTRIBUTE          type 7 length 16:
*Aug 18 09:06:07.919:  Setup Prio: 7, Holding Prio: 7
*Aug 18 09:06:07.919:  Flags: (0x7) Local Prot desired, Label Recording, SE Style
```

```
*Aug 18 09:06:07.919: Session Name: R1_t0
*Aug 18 09:06:07.919: SENDER_TEMPLATE      type 7 length 12:
*Aug 18 09:06:07.919: Tun Sender: 1.1.1.1 LSP ID: 248
*Aug 18 09:06:07.919: SENDER_TSPEC      type 2 length 36:
*Aug 18 09:06:07.919: version=0, length in words=7
*Aug 18 09:06:07.919: Token bucket fragment (service_id=1, length=6 words
*Aug 18 09:06:07.919: parameter id=127, flags=0, parameter length=5
*Aug 18 09:06:07.919: average rate=250000 bytes/sec, burst depth=1000 bytes
*Aug 18 09:06:07.919: peak rate    =250000 bytes/sec
*Aug 18 09:06:07.919: min unit=0 bytes, max pkt size=2147483647 bytes
*Aug 18 09:06:07.919: ADSPEC            type 2 length 48:
*Aug 18 09:06:07.919: version=0 length in words=10
*Aug 18 09:06:07.919: General Parameters break bit=0 service length=8
*Aug 18 09:06:07.919: IS Hops:0
*Aug 18 09:06:07.919: Minimum Path Bandwidth (bytes/sec):2147483647
*Aug 18 09:06:07.919: Path Latency (microseconds):0
*Aug 18 09:06:07.919: Path MTU:4294967295
*Aug 18 09:06:07.919: Controlled Load Service break bit=0 service length=0
*Aug 18 09:06:07.919: LABEL_REQUEST     type 1 length 8 :
*Aug 18 09:06:07.919: Layer 3 protocol ID: 2048
*Aug 18 09:06:07.919:
*Aug 18 09:06:07.919: Outgoing PathTear:
*Aug 18 09:06:07.919: version:1 flags:0000 cksum:BFD0 ttl:255 reserved:0 length:132
*Aug 18 09:06:07.919: SESSION           type 7 length 16:
*Aug 18 09:06:07.919: Tun Dest: 5.5.5.5 Tun ID: 0 Ext Tun ID: 1.1.1.1
*Aug 18 09:06:07.919: HOP               type 1 length 12:
*Aug 18 09:06:07.919: Hop Addr: 10.1.12.1 LIH: 0x20000403
*Aug 18 09:06:07.919: SENDER_TEMPLATE   type 7 length 12:
*Aug 18 09:06:07.919: Tun Sender: 1.1.1.1 LSP ID: 247
*Aug 18 09:06:07.919: SENDER_TSPEC      type 2 length 36:
*Aug 18 09:06:07.919: version=0, length in words=7
*Aug 18 09:06:07.919: Token bucket fragment (service_id=1, length=6 words
```

```
*Aug 18 09:06:07.919:      parameter id=127, flags=0, parameter length=5
*Aug 18 09:06:07.919:      average rate=250000 bytes/sec, burst depth=1000 bytes
*Aug 18 09:06:07.919:      peak rate      =250000 bytes/sec
*Aug 18 09:06:07.919:      min unit=0 bytes, max pkt size=2147483647 bytes
*Aug 18 09:06:07.919: ADSPEC                      type 2 length 48:
*Aug 18 09:06:07.919: version=0  length in words=10
*Aug 18 09:06:07.919: General Parameters  break bit=0  service length=8
*Aug 18 09:06:07.919:                               IS Hops:0
*Aug 18 09:06:07.919:                               Minimum Path Bandwidth (bytes/sec):2147483647
*Aug 18 09:06:07.919:                               Path Latency (microseconds):0
*Aug 18 09:06:07.919:                               Path MTU:4294967295
*Aug 18 09:06:07.919: Controlled Load Service  break bit=0  service length=0
*Aug 18 09:06:07.919:
```

**在 R2 配置备份链路，R2 上增加配置如下：**

```
ip explicit-path name R3R4 enable
  next-address 10.1.23.3
  next-address 10.1.34.4
!
interface Tunnel0
  ip unnumbered Loopback0
  tunnel mode mpls traffic-eng
  tunnel destination 4.4.4.4
  tunnel mpls traffic-eng path-option 10 explicit name R3R4
!
Interface eth0/1
  mpls traffic-eng backup-path tunnel0
```

**从 R1 的 debug 信息可以看出：**

```
*Aug 18 11:26:02.546: Incoming Resv:    !! R2 发给 R1 的 resv 消息
*Aug 18 11:26:02.546:   version:1 flags:0000 cksum:CA2B ttl:255 reserved:0 length:160
*Aug 18 11:26:02.546: SESSION                      type 7 length 16:
```

```
*Aug 18 11:26:02.546: Tun Dest: 5.5.5.5 Tun ID: 0 Ext Tun ID: 1.1.1.1
*Aug 18 11:26:02.546: HOP type 1 length 12:
*Aug 18 11:26:02.546: Hop Addr: 10.1.12.2 LIH: 0x0D000408
*Aug 18 11:26:02.546: TIME_VALUES type 1 length 8 :
*Aug 18 11:26:02.546: Refresh Period (msec): 30000
*Aug 18 11:26:02.546: STYLE type 1 length 8 :
*Aug 18 11:26:02.546: Shared-Explicit (SE)
*Aug 18 11:26:02.546: FLOWSPEC type 2 length 36:
*Aug 18 11:26:02.546: version = 0 length in words = 7
*Aug 18 11:26:02.546: service id = 5, service length = 6
*Aug 18 11:26:02.546: tspec parameter id = 127, flags = 0,length = 5
*Aug 18 11:26:02.546: average rate = 250000 bytes/sec, burst depth = 1000 bytes
*Aug 18 11:26:02.546: peak rate = 250000 bytes/sec
*Aug 18 11:26:02.546: min unit = 0 bytes,max pkt size = 1500 bytes
*Aug 18 11:26:02.546: FILTER_SPEC type 7 length 12:
*Aug 18 11:26:02.546: Tun Sender: 1.1.1.1, LSP ID: 521
*Aug 18 11:26:02.546: LABEL type 1 length 8 :
*Aug 18 11:26:02.546: Labels: 200
*Aug 18 11:26:02.546: RECORD_ROUTE type 1 length 52:
*Aug 18 11:26:02.546: 2.2.2.2/32, Flags:0x21 (Local Prot Avail/to NHOP, Node-id)
!! 这里也发生了变化
*Aug 18 11:26:02.546: Label subobject: Flags 0x1, C-Type 1, Label 200
*Aug 18 11:26:02.546: 4.4.4.4/32, Flags:0x20 (No Local Protection, Node-id)
*Aug 18 11:26:02.546: Label subobject: Flags 0x1, C-Type 1, Label 403
*Aug 18 11:26:02.546: 5.5.5.5/32, Flags:0x20 (No Local Protection, Node-id)
*Aug 18 11:26:02.546: Label subobject: Flags 0x1, C-Type 1, Label 0
*Aug 18 11:26:02.546:
```

上面是正常情况下的 debug 信息。

#### R1#show mpls traffic-eng tunnels

```
Name: R1_t0 (Tunnel0) Destination: 5.5.5.5
Status:
```

Admin: up      Oper: up      Path: valid      Signalling: connected  
path option 10, type explicit R2R4R5 (Basis for Setup, path weight 30)

#### Config Parameters:

Bandwidth: 2000      kbps (Global)      Priority: 7    7      Affinity: 0x0/0xFFFF  
Metric Type: TE (default)  
AutoRoute: enabled      LockDown: disabled      Loadshare: 2000      bw-based  
auto-bw: disabled

#### Active Path Option Parameters:

State: explicit path option 10 is active  
BandwidthOverride: disabled      LockDown: disabled      Verbatim: disabled

InLabel : -

OutLabel : Ethernet0/0, 200

#### RSVP Signalling Info:

Src 1.1.1.1, Dst 5.5.5.5, Tun\_Id 0, Tun\_Instance 521

#### RSVP Path Info:

My Address: 10.1.12.1  
Explicit Route: 10.1.12.2 10.1.24.2 10.1.24.4 10.1.45.4  
10.1.45.5 5.5.5.5

Record    Route:    NONE

Tspec: ave rate=2000 kbits, burst=1000 bytes, peak rate=2000 kbits

#### RSVP Resv Info:

Record    Route:    **2.2.2.2(200) 4.4.4.4(403)**  
**5.5.5.5(0)**

**!! 发生了变化了，在 head 路由器这，会记录沿路的标签。**

Fspec: ave rate=2000 kbits, burst=1000 bytes, peak rate=2000 kbits

#### History:

##### Tunnel:

Time since created: 5 hours, 38 minutes  
Time since path change: 10 minutes, 34 seconds  
Number of LSP IDs (Tun\_Instances) used: 521

```
Current LSP:
  Uptime: 5 minutes, 53 seconds
Prior LSP:
  ID: path option 10 [248]
  Removal Trigger: re-route path verification failed
R1#
```

现在，将 R2 的 e0/1 口 shutdown 掉。

R1 上的 debug 信息

```
*Aug 18 11:31:44.598: Incoming PathError:      !! R2 发给 R1 的 Patherror 消息
*Aug 18 11:31:44.598:   version:1 flags:0000 cksum:7931 ttl:255 reserved:0 length:132
*Aug 18 11:31:44.598:  SESSION                               type 7 length 16:
*Aug 18 11:31:44.598:   Tun Dest:   5.5.5.5  Tun ID: 0  Ext Tun ID: 1.1.1.1
*Aug 18 11:31:44.598:  ERROR_SPEC                               type 1 length 12:
*Aug 18 11:31:44.598:   Error Node: 10.1.12.2
*Aug 18 11:31:44.598:   Error Code: 25 (Notify)
*Aug 18 11:31:44.598:  Error Value: 0x3  (Tunnel locally repaired)
                        !!密切注意这里
*Aug 18 11:31:44.598:   Flags: 0x0
*Aug 18 11:31:44.598:  SENDER_TEMPLATE                           type 7 length 12:
*Aug 18 11:31:44.598:   Tun Sender: 1.1.1.1  LSP ID: 521
*Aug 18 11:31:44.598:  SENDER_TSPEC                               type 2 length 36:
*Aug 18 11:31:44.598:   version=0, length in words=7
*Aug 18 11:31:44.598:   Token bucket fragment (service_id=1, length=6 words
*Aug 18 11:31:44.598:     parameter id=127, flags=0, parameter length=5
*Aug 18 11:31:44.598:     average rate=250000 bytes/sec, burst depth=1000 bytes
*Aug 18 11:31:44.598:     peak rate   =250000 bytes/sec
*Aug 18 11:31:44.598:     min unit=0 bytes, max pkt size=2147483647 bytes
*Aug 18 11:31:44.598:  ADSPEC                                    type 2 length 48:
*Aug 18 11:31:44.598:   version=0  length in words=10
*Aug 18 11:31:44.598:  General Parameters  break bit=0  service length=8
```

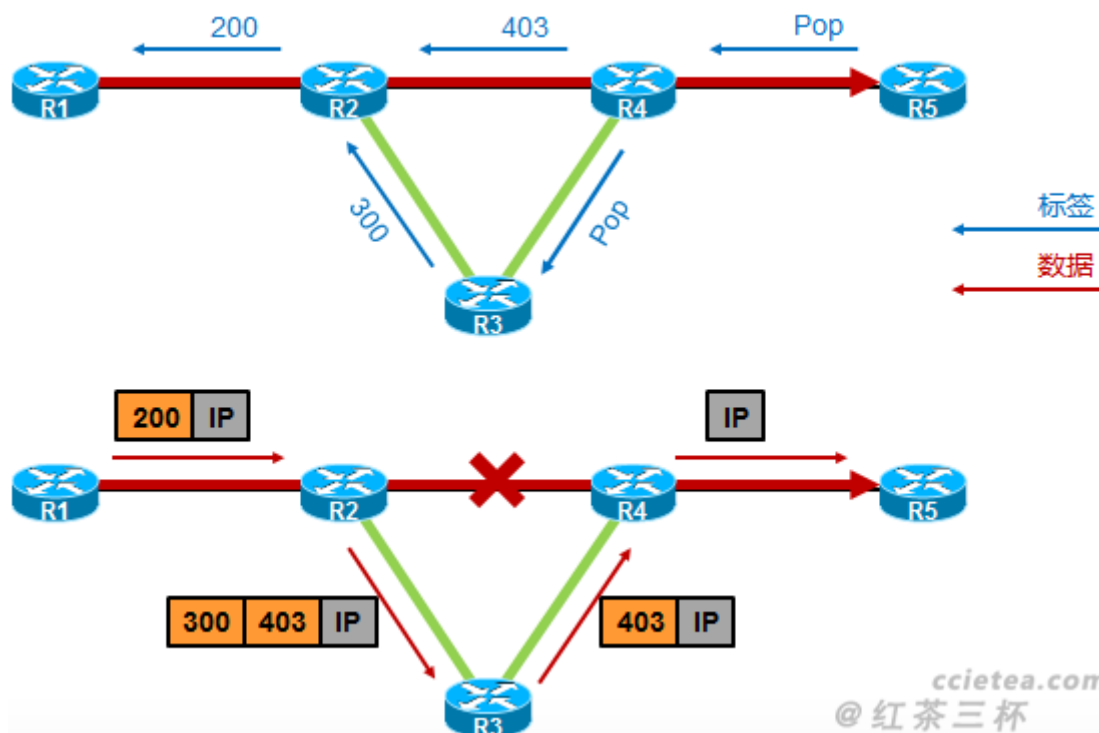
```
*Aug 18 11:31:44.598: IS Hops:1
*Aug 18 11:31:44.598: Minimum Path Bandwidth (bytes/sec):1250000
*Aug 18 11:31:44.598: Path Latency (microseconds):0
*Aug 18 11:31:44.598: Path MTU:1500
*Aug 18 11:31:44.598: Controlled Load Service break bit=0 service length=0
*Aug 18 11:31:44.598:
```

### R1# traceroute 5.5.5.5

Type escape sequence to abort.

Tracing the route to 5.5.5.5

```
1 10.1.12.2 [MPLS: Label 200 Exp 0] 4 msec 0 msec 0 msec
2 10.1.23.3 [MPLS: Labels 300/403 Exp 0] 0 msec 4 msec 0 msec
3 10.1.34.4 [MPLS: Label 403 Exp 0] 0 msec 0 msec 4 msec
4 10.1.45.5 0 msec
```



由于我们在 R2 上部署了 link 保护，保护的是 R2-R4 之间的链路，因此当 shutdown 掉 R2 的 E0/1 口后，R2 发送给 R1 的 Patherror 消息中，提示 R1 并不需要去拆掉 tunnel。所以 R1 的 tunnel 不会拆除。并且在 R2 这里，标签包被压入了新的一层标签，外层标签是 R3 分配的，内层标签是 R4 分配的。

这时候再在 R2 上看一下：

**R2#show mpls traffic-eng fast-reroute database**

Headend frr information:

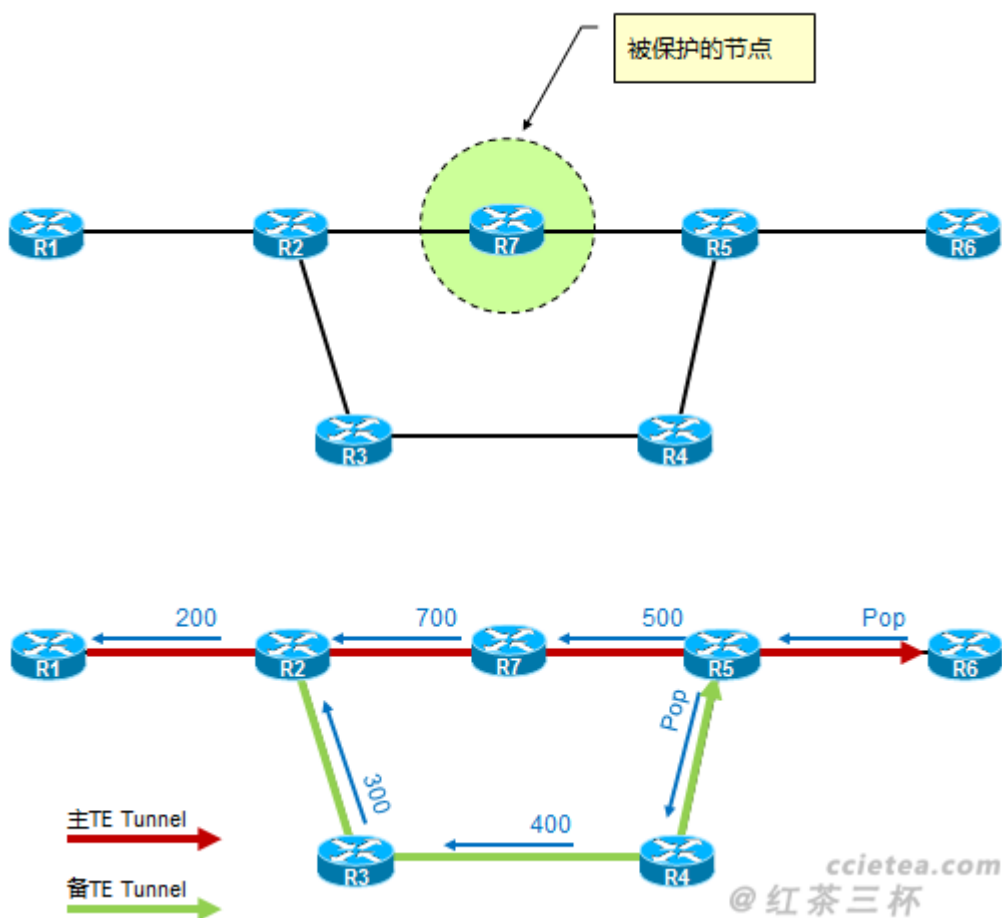
Protected tunnel	In-label	Out intf/label	FRR intf/label	Status
------------------	----------	----------------	----------------	--------

LSP midpoint frr information:

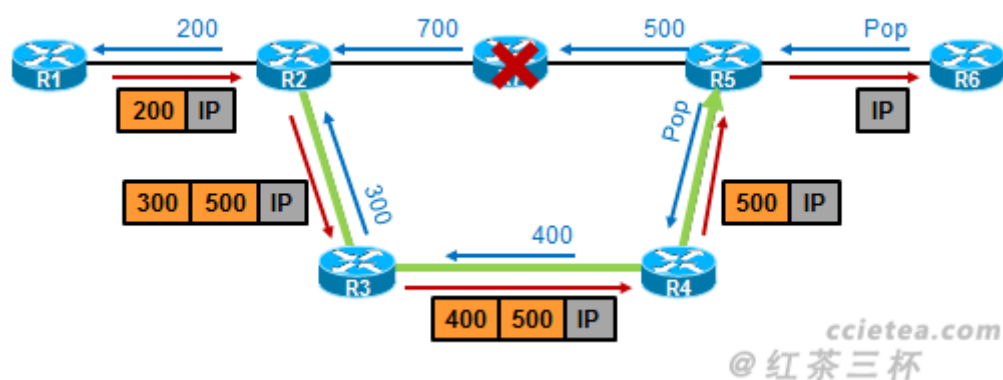
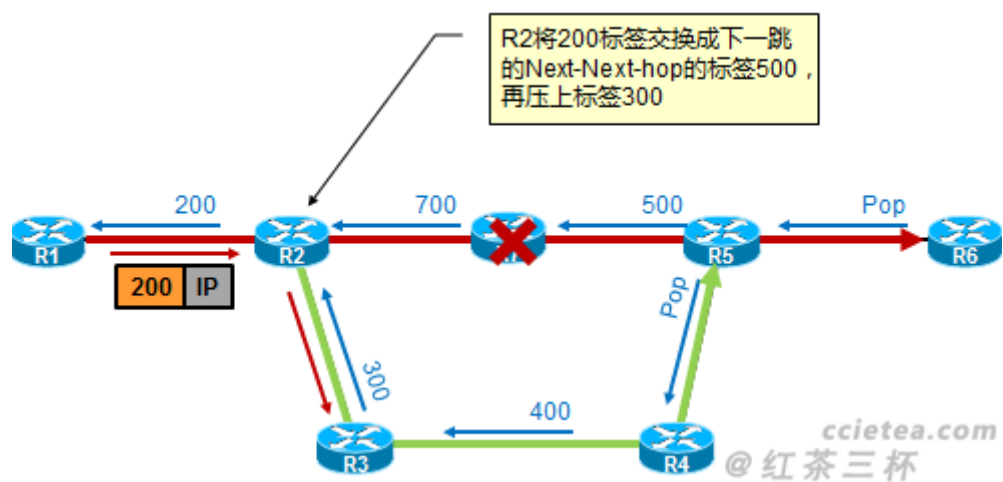
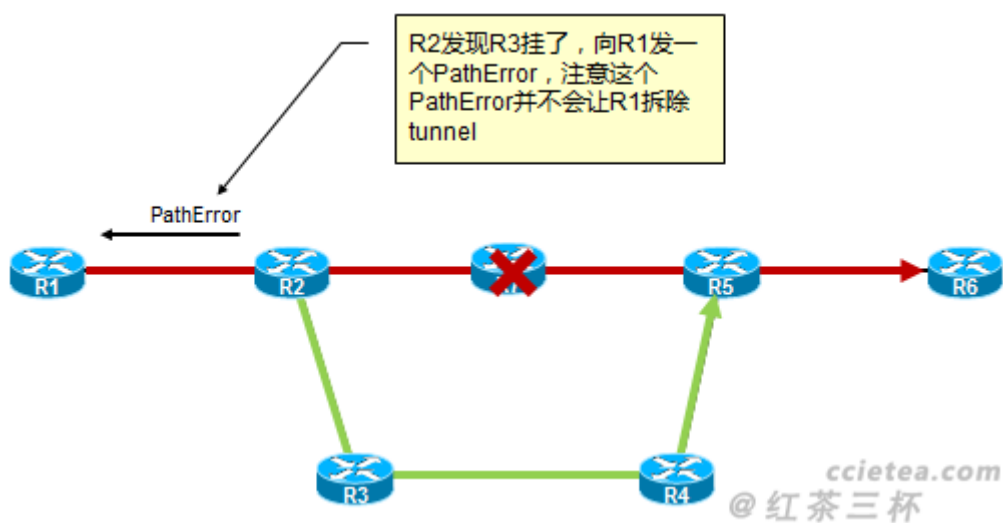
LSP identifier	In-label	Out intf/label	FRR intf/label	Status
1.1.1.1 0 [521]	200	Et0/1:403	Tu0:403	<b>active</b>

这里状态为 active。当主链路没有失效时，status 显示为 ready

### 7.3.4 基础实验 ( NodeProtection )







## 7.4 AutoBandwidth

Traffic Engineering (TE) automatic bandwidth feature adjusts the bandwidth allocation for TE tunnels based on their measured traffic load:

- It periodically changes tunnel bandwidth(BW) reservation based on traffic out tunnel.
- 基于每个 tunnel 测量平均 output 速率
- The allocated bandwidth is periodically adjusted to be the largest sample for the tunnel since the last adjustment

```
mpls traffic-eng auto-bw timers frequency 300
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel mode mpls traffic-eng
 tunnel destination 5.5.5.5
 tunnel mpls traffic-eng path-option 10 dynamic
 tunnel mpls traffic-eng bandwidth 2500
 tunnel mpls traffic-eng auto-bw frequency 3600 max-bw 3000 min-bw 1000
```

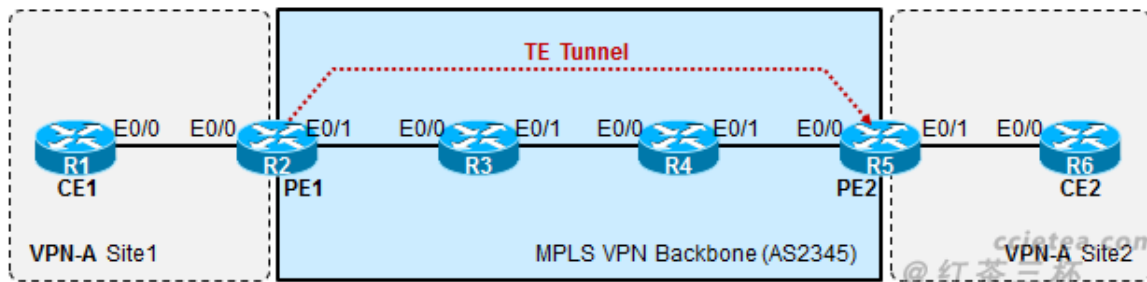
初始情况下tunnel的保证带宽为2.5M

每300S采样一次，取3600S内采样的带宽最大值。

无论采样的结果如何，带宽最小值保证为1M，最大为3M

## 8 MPLS TE and MPLS VPN

### 8.1 实验 1



#### 1. 实验环境

- 设备互联网段为 10.1.xy.0/24，其中 xy 为设备编号，x 小 y 大
- 所有设备的 loopback0 地址空间为 x.x.x.x/32，x 为设备编号
- R2、R3、R4、R5 为 MPLS VPN Backbone 路由器，运行的 Backbone IGP 协议是 OSPF，进程号使用 100。R2 及 R5 之间建立 MP-iBGP 连接。
- PE-CE 间运行的 VRF IGP 为 OSPF，使用进程号 1
- MPLS VPN Backbone 内进行针对 MPLS TE 的 OSPF 扩展，并在 R1 上建立一个 TE Tunnel，源为自身的 loopback0，目的为 R5 的 5.5.5.5

#### 2. 实验配置

R1 的配置如下：

```
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface Ethernet0/0
 ip address 10.1.12.1 255.255.255.0
!
router ospf 1
 router-id 1.1.1.1
 network 1.1.1.1 0.0.0.0 area 0
```

```
network 10.1.12.1 0.0.0.0 area 0
```

#### R6 的配置如下：

```
interface Loopback0
 ip address 6.6.6.6 255.255.255.255
!
interface Ethernet0/0
 ip address 10.1.56.6 255.255.255.0
!
router ospf 1
 router-id 6.6.6.6
 network 6.6.6.6 0.0.0.0 area 0
 network 10.1.56.6 0.0.0.0 area 0
```

#### R2-PE1 的配置如下：

```
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
mpls traffic-eng tunnels
mpls label range 200 299
mpls ldp router-id Loopback0
!
ip vrf VPN-A
 rd 2345:1
 route-target export 2345:2
 route-target import 2345:5
!
interface Ethernet0/0
 ip vrf forwarding VPN-A
 ip address 10.1.12.2 255.255.255.0
!
interface Ethernet0/1
 ip address 10.1.23.2 255.255.255.0
```

```

mpls traffic-eng tunnels
mpls ip
ip rsvp bandwidth
!
router ospf 1 vrf VPN-A
 redistribute bgp 2345 subnets
 network 10.1.12.2 0.0.0.0 area 0
!
router ospf 100
router-id 2.2.2.2
 network 2.2.2.2 0.0.0.0 area 0
 network 10.1.23.2 0.0.0.0 area 0
mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
router bgp 2345
 bgp router-id 2.2.2.2
 no bgp default ipv4-unicast
 neighbor 5.5.5.5 remote-as 2345
 neighbor 5.5.5.5 update-source Loopback0
!
address-family vpnv4
 neighbor 5.5.5.5 activate
 neighbor 5.5.5.5 send-community extended
exit-address-family
!
address-family ipv4 vrf VPN-A
 no synchronization
 redistribute ospf 1 vrf VPN-A match internal external 1 external 2
exit-address-family
!
interface Tunnel0

```

```
ip unnumbered Loopback0
tunnel mode mpls traffic-eng
tunnel destination 5.5.5.5
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 2000
tunnel mpls traffic-eng path-option 10 dynamic
```

### R3 的配置如下：

```
mpls traffic-eng tunnels
mpls label range 300 399
mpls ldp router-id Loopback0
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface Ethernet0/0
 ip address 10.1.23.3 255.255.255.0
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface Ethernet0/1
 ip address 10.1.34.3 255.255.255.0
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 3.3.3.3
```

```
network 3.3.3.3 0.0.0.0 area 0
network 10.1.23.3 0.0.0.0 area 0
network 10.1.34.3 0.0.0.0 area 0
```

#### R4 的配置如下：

```
mpls traffic-eng tunnels
mpls label range 400 499
mpls ldp router-id Loopback0
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!
interface Ethernet0/0
 ip address 10.1.34.4 255.255.255.0
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface Ethernet0/1
 ip address 10.1.45.4 255.255.255.0
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
router ospf 100
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 router-id 4.4.4.4
 network 4.4.4.4 0.0.0.0 area 0
 network 10.1.34.4 0.0.0.0 area 0
 network 10.1.45.4 0.0.0.0 area 0
```

### R5-PE2 的配置如下：

```
interface Loopback0
 ip address 5.5.5.5 255.255.255.255
!
mpls traffic-eng tunnels
mpls label range 500 599
mpls ldp router-id Loopback0
!
ip vrf VPN-A
 rd 2345:6
 route-target export 2345:5
 route-target import 2345:2
!
interface Ethernet0/0
 ip address 10.1.45.5 255.255.255.0
 mpls traffic-eng tunnels
 mpls ip
 ip rsvp bandwidth
!
interface Ethernet0/1
 ip vrf forwarding VPN-A
 ip address 10.1.56.5 255.255.255.0
!
!
router ospf 1 vrf VPN-A
 redistribute bgp 2345 subnets
 network 10.1.56.5 0.0.0.0 area 0
!
router ospf 100
 router-id 5.5.5.5
 network 5.5.5.5 0.0.0.0 area 0
 network 10.1.45.5 0.0.0.0 area 0
```



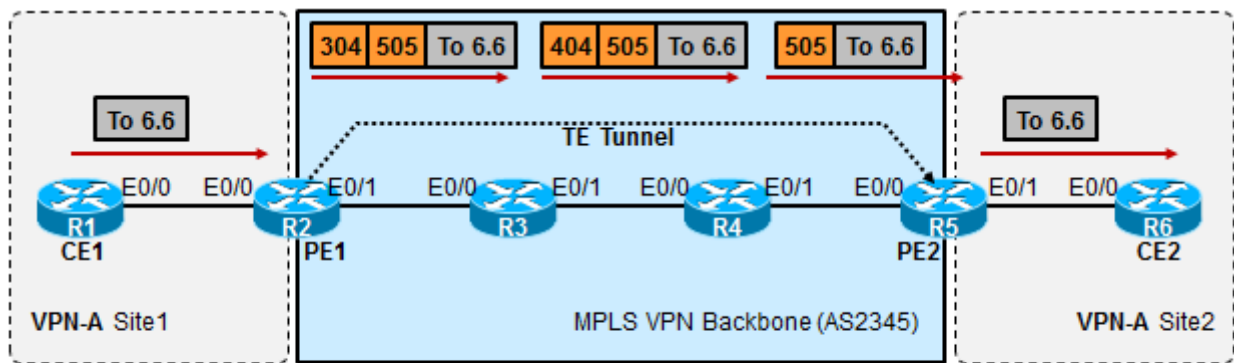
```
mpls traffic-eng router-id Loopback0
mpls traffic-eng area 0
!
router bgp 2345
  bgp router-id 5.5.5.5
  no bgp default ipv4-unicast
  neighbor 2.2.2.2 remote-as 2345
  neighbor 2.2.2.2 update-source Loopback0
  !
address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community extended
exit-address-family
!
address-family ipv4 vrf VPN-A
  no synchronization
  redistribute ospf 1 vrf VPN-A match internal external 1 external 2
exit-address-family
```

### 3. 现象解析

**我们在 R1 上去 traceroute 6.6.6.6**

**R1#traceroute 6.6.6.6**

```
Type escape sequence to abort.
Tracing the route to 6.6.6.6
 1 10.1.12.2 4 msec 4 msec 0 msec
 2 10.1.23.3 [MPLS: Labels 304/505 Exp 0] 4 msec 4 msec 4 msec
 3 10.1.34.4 [MPLS: Labels 404/505 Exp 0] 0 msec 0 msec 0 msec
 4 10.1.56.5 [MPLS: Label 505 Exp 0] 8 msec 0 msec 0 msec
 5 10.1.56.6 4 msec * 4 msec
```



ccietea.com  
@ 红茶三杯

首先 R2 的路由表：

R2-PE1#show ip route

```
O          5.5.5.5 [110/31] via 5.5.5.5, 00:06:39, Tunnel0
```

我们看到 Tunnel 已经起来了，而且 R2 上去往 5.5.5.5 的路由，出接口为 tunnel，上了 TE tunnel 了。

R2-PE1#show ip bgp vpnv4 all labels

Network	Next Hop	In label/Out label
.....		
Route Distinguisher: 2345:6		
6.6.6.6/32	5.5.5.5	<b>no label/505</b>
10.1.56.0/24	5.5.5.5	no label/506

我们看到 R5-PE2 为 VPN 客户路由 6.6.6.6 分配了 505 的标签。

所以 R2 收到去往 6.6.6.6 的数据包，先压入 505 的内层 VPN 标签。

接着需要需要顶层的标签，这就看下一跳，下一跳是 5.5.5.5，那么肯定要找 5.5.5.5 的标签。由于从路由表我们已经看到，此刻 R2 去往 5.5.5.5 实际上是进了 TE tunnel，那么：

R2-PE1#show mpls forwarding-table detail

Local	Outgoing	Prefix	Bytes Label	Outgoing	Next Hop
Label	Label	or Tunnel Id	Switched	interface	
.....					
204	Pop Label	<b>5.5.5.5/32</b>	0	Tu0	point2point
MAC/Encaps=14/18, MRU=1500, <b>Label Stack{304}</b> , via Et0/1					
0E00003017000E00003016108847 00130000					

No output feature configured

所以这里 R2 将标签包送入 tunnel 时，给压入了外层的标签为 304，而这个 304 很明显是 R3 分配的，而且是 RSVP 为这条 TE Tunnel 分配的标签。

上面的标签，还有另一个方法可以看到，就是直接在 R2 上看 TE Tunnel：

#### R2-PE1#show mpls traffic-eng tunnels

```
Name: R2-PE1_t0 (Tunnel0) Destination: 5.5.5.5
Status:
  Admin: up      Oper: up      Path: valid      Signalling: connected
  path option 10, type dynamic (Basis for Setup, path weight 30)
.....
InLabel  : -
OutLabel : Ethernet0/1, 304
RSVP Signalling Info:
  Src 2.2.2.2, Dst 5.5.5.5, Tun_Id 0, Tun_Instance 5
RSVP Path Info:
  My Address: 10.1.23.2
  Explicit Route: 10.1.23.3 10.1.34.3 10.1.34.4 10.1.45.4
```

接下去看 R3：

#### R3#show mpls forwarding-table

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Switched	Outgoing interface	Next Hop
300	Pop Label	2.2.2.2/32	13223	Et0/0	10.1.23.2
301	Pop Label	4.4.4.4/32	0	Et0/1	10.1.34.4
302	Pop Label	10.1.45.0/24	0	Et0/1	10.1.34.4
303	403	5.5.5.5/32	6621	Et0/1	10.1.34.4
<b>304</b>	<b>404</b>	<b>2.2.2.2 0 [5]</b>	<b>5135</b>	<b>Et0/1</b>	<b>10.1.34.4</b>

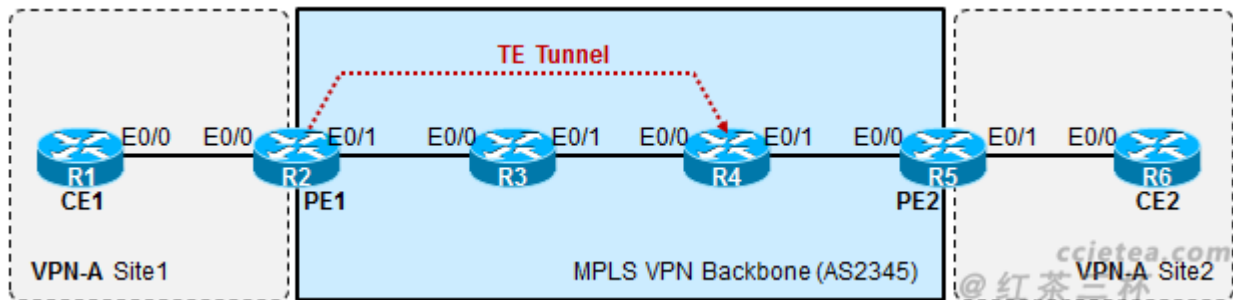
R3 将 304 替换成 404。注意，这里 304 和 404 都是 RSVP 分配的标签，因此我们在查看标签的时候还是要关注 tunnel。上面的表项 2.2.2.2 0[5]，注意不要搞错，不是 2.2.2.2 这个前缀对应的标签，而是为这条 tunnel 对应的标签。

再接下去就不用解释了吧？到了 R4，R4 将顶层标签弹出，然后将包转发给 R5。

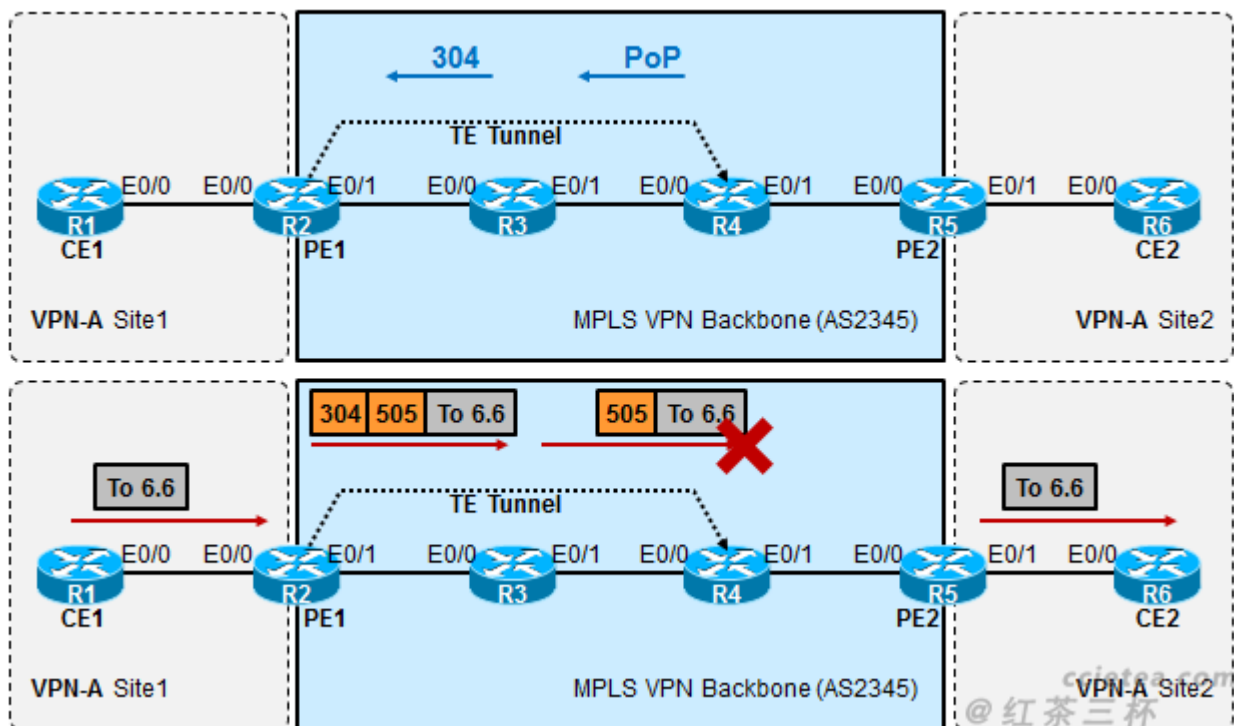
因此，在 MPLS VPN 环境中，如果我们在 MPLS VPN Backbone 内在 R1 到 R5 建立一条 TE Tunnel，并不

会影响 VPN 的流量。

## 8.2 实验 2



这一次我们做的测试，环境就跟实验 1 不同了，TE Tunnel 的尾端现在是到了 R4 的 4.4.4.4。



在这个环境中，由于 Tunnel 尾端在 R4，那么 R4 通过 RSVP 为这条 tunnel 分配的标签就是 POP。

这样一来当 R2 转发标签数据时，数据压入两层标签，外层是 TE Tunnel 的标签 304，内层是 VPNv4 标签 505，这个标签包到了 R3 后，R3 将顶层的标签弹出，然后将数据转发给 R4，但是到了 R4 它就傻逼了，因为 505 标签是由 R5 产生的，R4 压根不认识这 505 标签，于是就丢包了。

**解决方案：在 R2 及 R4 之间建立 targeted LDP 邻居关系**

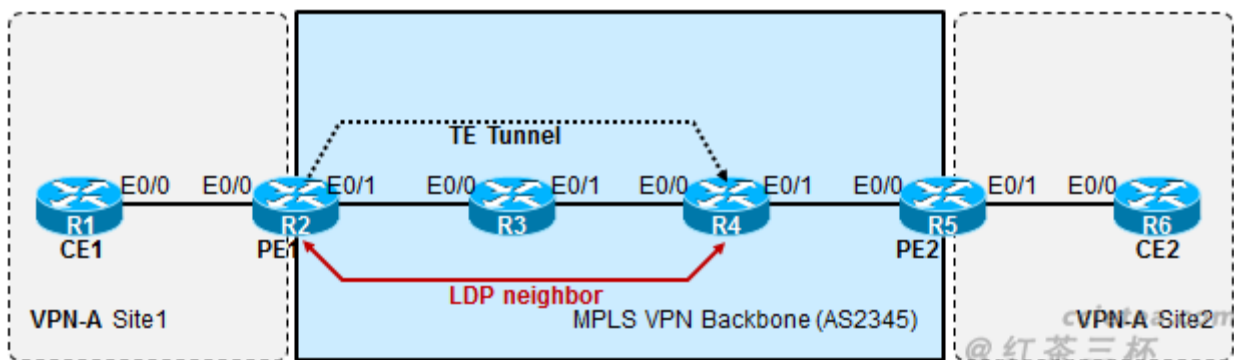
**R2 的 tunnel 配置修改如下：**

```
interface Tunnel0
ip unnumbered Loopback0
mpls ip
tunnel mode mpls traffic-eng
tunnel destination 4.4.4.4
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 7 7
tunnel mpls traffic-eng bandwidth 2000
tunnel mpls traffic-eng path-option 10 dynamic
```

上述配置完成后，R2 就会通过已经建立好的、到达 R4 的 tunnel 去发 LDP。注意 Tunnel 一定要激活 MPLS IP

**R4 的配置修改如下：**

```
mpls ldp neighbor 2.2.2.2 targeted ldp
```



**在 R2 上看一下 LDP 邻居：**

**R2-PE1#show mpls ldp neighbor**

```
Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
TCP connection: 3.3.3.3.62244 - 2.2.2.2.646
State: Oper; Msgs sent/rcvd: 112/112; Downstream
Up time: 01:29:44
LDP discovery sources:
  Ethernet0/1, Src IP addr: 10.1.23.3
Addresses bound to peer LDP Ident:
```

```

10.1.23.3      10.1.34.3      3.3.3.3
Peer LDP Ident: 4.4.4.4:0; Local LDP Ident 2.2.2.2:0
TCP connection: 4.4.4.4.24940 - 2.2.2.2.646
State: Oper; Msgs sent/rcvd: 19/19; Downstream
Up time: 00:07:56
LDP discovery sources:
Targeted Hello 2.2.2.2 -> 4.4.4.4, active
Addresses bound to peer LDP Ident:
10.1.34.4      10.1.45.4      4.4.4.4
  
```

已经和 R4 建立起来了 targeted LDP 邻居关系。

接着我们再去 R1 上测试一下：

R1#traceroute 6.6.6.6

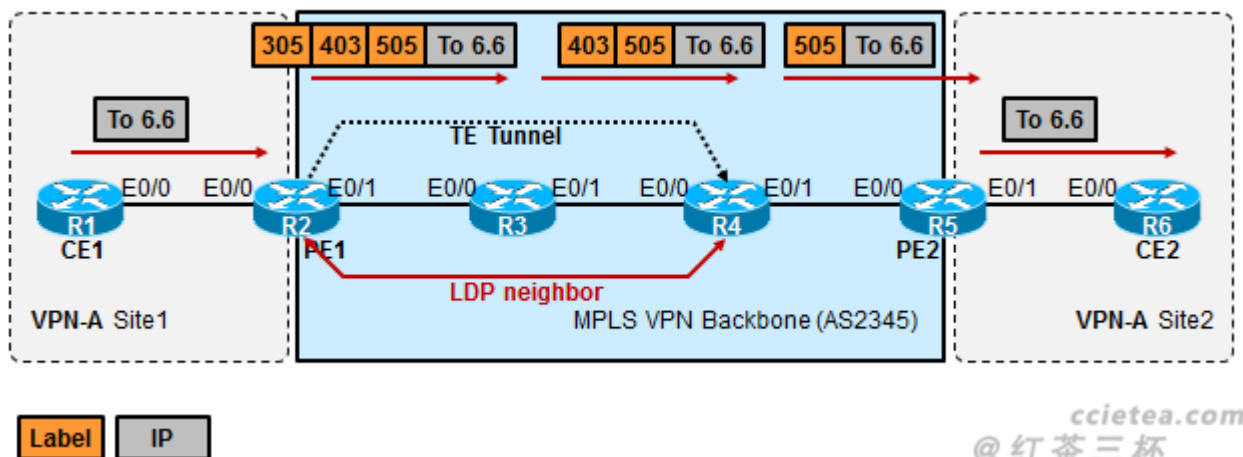
Type escape sequence to abort.

Tracing the route to 6.6.6.6

```

 1 10.1.12.2 12 msec 4 msec 4 msec
 2 10.1.23.3 [MPLS: Labels 305/403/505 Exp 0] 8 msec 4 msec 4 msec
 3 10.1.34.4 [MPLS: Labels 403/505 Exp 0] 0 msec 4 msec 0 msec
 4 10.1.56.5 [MPLS: Label 505 Exp 0] 4 msec 0 msec 0 msec
 5 10.1.56.6 8 msec * 0 msec
  
```

我们看到数据的转发层面是这样的：



那么来详细分析一下：

首先是 R2 上

由于去往 6.6.6.6 的 IP 数据包到了 R2，IP 数据包内层下压入一个 VPN 的标签 505，这个没问题。

接着我们继续，R2 上关于 6.6.6.6 的 VPNv4 路由是由 5.5.5.5 也就是 R5 传递过来的。

而 R2 的路由表此刻是这样的：

R2-PE1#show ip route

```
O          5.5.5.5 [110/31] via 4.4.4.4, 00:23:47, Tunnel0
```

走的是 tunnel。

因此：

R2-PE1#show mpls forwarding-table detail

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
204	403	5.5.5.5/32	0	Tu0	point2point
MAC/Encaps=14/22, MRU=1496, <b>Label Stack{305 403}</b> , via Et0/1					
0E00003017000E00003016108847 0013100000193000					
No output feature configured					

我们看到，这里是有两层标签。所以 R2 将原始的 IP 包先压入一层 VPN 标签 505，在压入 305 403 这两层标签。这里指的是注意的是 这里的是标签 403 是由 R4 分配为前缀 5.5.5.5/32 所分配的 LDP 标签，这是通过 targeted LDP 连接传递给 R2 的。而 305 这是 R3 通过 RSVP 为 TE Tunnel 所分配的标签。

下面可以验证一下：

R2-PE1# show mpls forwarding-table

Local Label	Outgoing Label	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
204 [T] <b>403</b>	<b>5.5.5.5/32</b>	0	Tu0	point2point	

我们在 R2 的 LFIB 表里看到了 R4 分配给前缀 5.5.5.5/32 的 LDP 标签 403

最终这个压入了 403 和 505 的标签包要被送入 TE Tunnel，因此：

R2-PE1#show mpls traffic-eng tunnels

```
Name: R2-PE1_t0 (Tunnel0) Destination: 4.4.4.4
Status:
Admin: up      Oper: up      Path: valid    Signalling: connected
```

path option 10, type dynamic (Basis for Setup, path weight 20)

#### Config Parameters:

Bandwidth: 2000 kbps (Global) Priority: 7 7 Affinity: 0x0/0xFFFF

Metric Type: TE (default)

AutoRoute: enabled LockDown: disabled Loadshare: 2000 bw-based

auto-bw: disabled

#### Active Path Option Parameters:

State: dynamic path option 10 is active

BandwidthOverride: disabled LockDown: disabled Verbatim: disabled

InLabel : -

**OutLabel : Ethernet0/1, 305**

#### RSVP Signalling Info:

Src 2.2.2.2, Dst 4.4.4.4, Tun\_Id 0, Tun\_Instance 6

#### RSVP Path Info:

My Address: 10.1.23.2

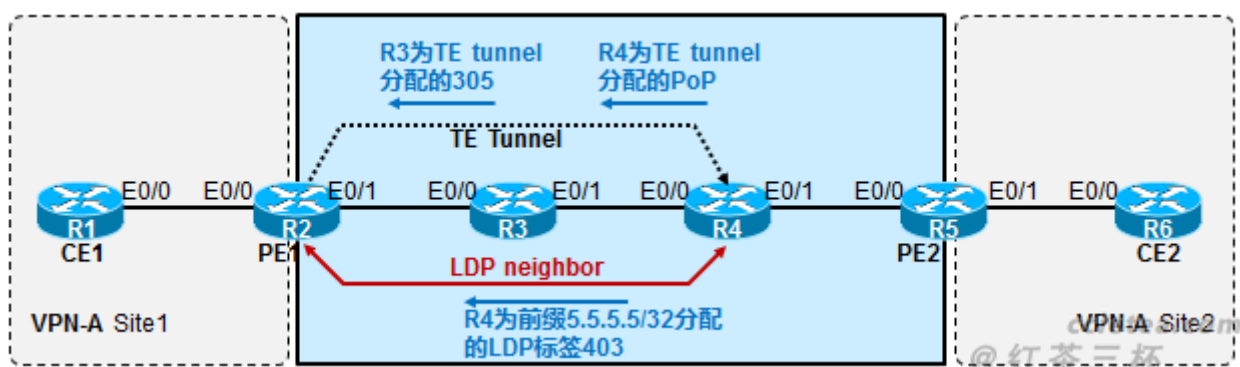
Explicit Route: 10.1.23.3 10.1.34.3 10.1.34.4 4.4.4.4

Record Route: NONE

Tspec: ave rate=2000 kbits, burst=1000 bytes, peak rate=2000 kbits

.....

标签的分发是这样的：





## 9 参考书籍

《MPLS 技术架构》