

网 络 之 路

MPLS 技术专刊

Route to Network

策划：刘宇 陈旭盛 刘炜刚

主编：陆宇翔

编委：杜祥宇 文旭 徐庆伟
王慧升 王辉 朱皓
王乐 陆强 刘先楠
张雪莲 蔡金龙 王君菠
贾欣武



基础知识

- 1 MPLS发展简史

深入探讨

- 6 MPLS及LDP协议基础
19 MPLS LSP ping-traceroute
29 MPLS L3VPN基础
48 MPLS L3VPN多实例路由协议
56 MPLS L2VPN之VLL篇
71 VPLS技术简介
88 MPLS TE技术原理简介
107 MPLS TE技术重点协议及相关报文描述
124 分层PE技术简介

测试万法

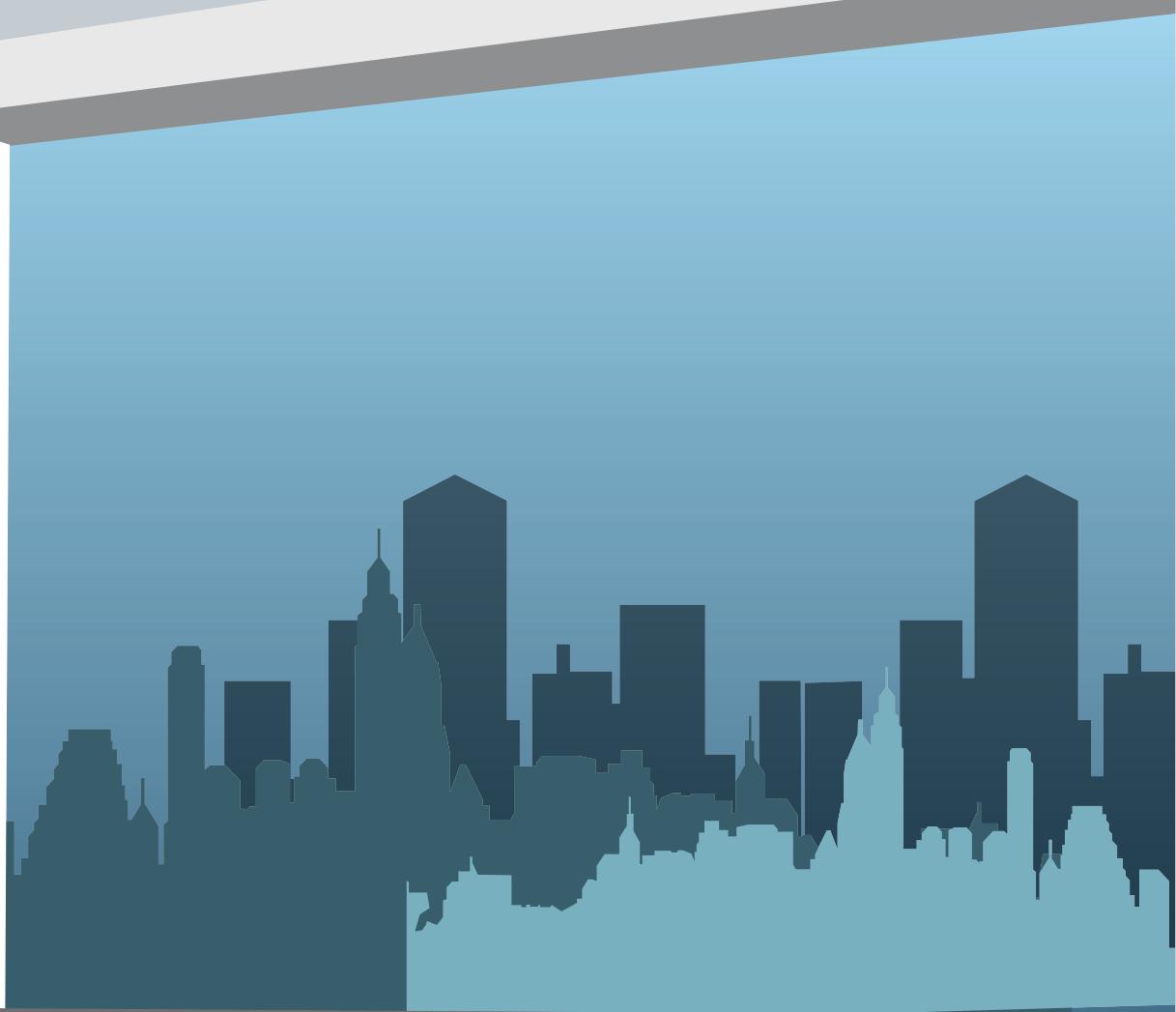
- 135 浅谈MPLS测试方法
151 MPLS测试仪器使用方法

组网应用

- 175 MPLS VPN组网应用分析

附录

- 186 缩略语



MPLS发展简史

邹旭东



Internet的发展 提出的挑战

90年代初，随着Internet的快速普及，由于当时硬件技术的限制，采用最长匹配算法、逐跳转发方式的路由器日益成为限制网络转发性能的一大瓶颈，快速路由器技术成为当时研究的一个热点。而与此同时，ATM技术因为采用定长标签，并且只需维护比路由表小得多的标签表，可以提供比IP路由方式高得多的转发性能。这在当时也导致了广泛的争论，到底ATM和IP，谁将成为下一代网络技术的基础：IP技术简单，但性能受到限制；ATM技术性能高，但其复杂的控制信令和高昂的部署成本让人望而却步。

很自然的，开始有人尝试把ATM和IP技术的优势结合起来，在保持IP技术简洁性的前提下，提供类似于ATM技术的高性能。很多厂商都进行了类似研究，其结果是各个厂商提出了各自的标签交换解决方案。

IP Switching

1996年春，美国加州一个名为IPSilon的小公司推出了一项具有震撼意义的技术，称为IP Switching。IP Swiching技术通过在ATM交换机上提供一个额外的IP路由引擎，较好地把ATM的高速转发能力和IP的简洁易部署特点结合起来。

IP Switching的IP路由引擎运行标准的IP路由协议，可以执行普通的逐跳IP转发，它利用了ATM高效的转发平面，放弃了ATM复杂的控制平面协议，使用自己开发的较为简单的协议，包括标签绑定协

议（称为Ipsilon Flow Management Protocol, IFMP, RFC1953）和交换机管理协议（称为General Switch Management Protocol , GSMP, RFC1987），来完成设备间的ATM 通道的建立。IP Switching技术中转发通道的建立采用数据驱动的方式，当检测到一个业务流时，IP路由引擎会通过GSMP/IFMP协议和上游的邻接节点协商，为该业务分配一个VPI/VCI，并更新对应的ATM交换表；该业务流的路由通道上每个IP Switching设备均重复这个操作，就为这个业务流建立了一个端到端的ATM通道。这样，该业务流就由逐跳的IP转发方式，转变成为ATM转发方式了，可以获得更高的转发性能。

IP Switching技术的推出使得Ipsilon公司由一个默默无闻的小公司，一举成为IP通信界的明星，并刺激业界巨头，如CISCO、IBM纷纷推出更易于扩展和升级的三层交换解决方案，由此引发了路由交换技术的一次革命，并导致了MPLS技术的诞生。

Tag Switching

就在Ipsilon宣布他们的IP Switching技术不久，1996年秋天，Cisco公司宣布了自己的标签交换解决方案，称为Tag Switching（标记交换）技术。Tag Switching技术为路由表中每个目的地址分配一个短而定长的标签，并在网络中为这个目的地址建立一条标签转发通道，IP数据流被封装在标签中，转发时根据标签进行转发，因为CISCO认为，短而定长的标签的转发过程，比采用最长匹配方式的IP路由转发更高效，能提供更高的转发速率。

Tag Switching和IP Switching相比，在技术

上有较大差别， Tag Switching为出现在标记交换路由器路由表中的每个目的地址建立一条转发通道；而IP Switching则由数据流来驱动建立转发通道，转发通道只为该数据流使用。可以这样认为， Tag Switching是拓扑驱动的，而IP Switching是数据驱动的。

更为混乱。

为了协调各方利益，形成一个统一的标准，1996年底，IETF成立了一个工作组，对集成路由和交换技术的标签解决方案进行标准化。到1997年初，这个工作组形成了IETF认可的章程，工作组的第一次会议在1997年4月召开。经过多次商讨，最终MPLS（Multiprotocol Label Switching）这个术语被确定下来，作为独立于各个厂家私有标准的一系列标准的名称。

IBM的ARIS

几乎在Cisco公司宣布他们的Tag Switching技术的同时，IBM公司提出了一个称为ARIS（Aggregate Route-Based IP Switching）的标签交换解决方案。ARIS与Cisco公司的Tag Switching技术比较接近，都是把标签和路由关联，是拓扑驱动型标签技术。ARIS也有一些自己的特色，比如它一开始是考虑了把ATM作为链路层的，并提出了VC合并技术，这一思想最终也融入了MPLS中。

MPLS工作组

在上面提到的三种标签交换技术之外，还有其他各种类似技术，如3COM FASTIP、Cascade Navigator等，均能提供支持IP的二层交换功能。当时的情形是，各厂商纷纷提出自己的标签交换技术，如果没有一个标准化工作组，将会出现更多的互不兼容的标签交换技术，从而使市场变得

MPLS的现在与未来

MPLS用短而定长的标签来封装网络层分组。MPLS从各种链路层（如PPP、ATM、帧中继、以太网等）得到链路层服务，又为网络层提供面向连接的服务。MPLS能从IP路由协议和控制协议中得到支持，同时，还支持基于策略的约束路由，它路由功能强大、灵活，可以满足各种新应用对网络的要求。MPLS技术起源于IPv4，但其核心技术可扩展到多种网络协议（IPv6、IPX等）。MPLS最初是为提高路由器的转发速度而提出的一个协议，不过，随着硬件技术的进步，采用ASIC和NP进行转发的高速路由器和三层交换机得到广泛应用，MPLS提高转发速度的初衷已经没有意义。但是，MPLS支持多层标签和面向连接的特点，使得其在VPN、流量工程（TE，Traffic Engineering）、QoS等方面得到广泛应用，并因为其良好的扩展性，使得在统一的MPLS/IP基础网络架构上为客户提供各类服务成为可能，从而使得MPLS日益成为大规模网络的基础技术。



MPLS VPN技术概述

MPLS VPN 是一种基于MPLS技术的VPN，是在路由和交换设备上应用MPLS技术实现的虚拟专用网络，可灵活满足多种业务需求：可以用在解决企业互连、政府相同/不同部门的互连，也可以用来提供各种新业务，如为IP电话业务专门开辟一个VPN以解决IP网络地址不足和QoS的问题，或者用MPLS VPN为IPv6提供开展业务的可能。

目前MPLS L3 VPN，即BGP/MPLS VPN技术比较成熟，已经形成标准。L2 MPLS VPN近段时间发展迅速技术不断发展成熟，虽然标准都处于草案阶段，但由于多厂家支持逐渐形成一些事实上标准。MPLS技术在其它方面的应用，比如MPLS流量工程、组播VPN技术还不成熟，形成的草案尚在不断变化中。

BGP / MPLS VPN

在MPLS/BGP VPN的模型中，网络由运营商的骨干网与用户的各个Site组成，所谓VPN就是对site集合的划分，一个VPN对应一个由若干site组成的集合。

基于BGP扩展实现的L3 MPLS VPN所包含的基本组件：

PE : Provider Edge Router, 骨干网边缘路由器，存储VRF (Virtual Routing Forwarding In-

stance)，处理VPN-IPv4路由，是MPLS三层VPN的主要实现者。

CE : Custom Edge Router, 用户网边缘路由器，发布用户网络路由。

P router : Provider Router, 骨干网核心路由器，负责MPLS转发。

VPN用户站点 (site) : 是VPN中的一个孤立的IP网络，一各site之间通过运营商骨干网实现连通。公司总部、分支机构都是site的具体例子。CE路由器通常是VPN Site中的一个路由器或交换设备，Site通过一个单独的物理端口或逻辑端口连接到PE设备上。

用户接入MPLS VPN的方式是每个site提供一个或多个CE，同骨干网的PE连接。在PE上为这个site配置VRF，将连结PE-CE的物理接口、逻辑接口、甚至L2TP/IPSec隧道绑定到VRF上。

BGP扩展实现的MPLS VPN扩展了BGPNLRI中的IPv4地址，在其前增加了一个8字节的RD (Route Distinguisher)。RD用来区分不同VPN的IPV4地址。VPN的成员关系是通过VPN-IPV4路由所携带的Route Target属性来获得的，每个VRF配置了一些策略，规定一个VRF可以接收携带何种Route Target的路由信息，向外发布路由时携带什么Route Target属性，每个PE根据这些策略，确定接收到的哪些路由可以引入某个VRF中，并进行路由计算生成VRF相关的路由表。

PE-CE之间要交换路由信息，可以通过静态路由，也可以通过RIP、OSPF、BGP、IS-IS等动态路由协议。PE-CE之间采用静态路由的好处是可以减少CE设备可能会因为管理不善等原因，造成对骨干网BGP路由产生震荡，影响骨干网的稳定性。

MPLS/BGP VPN提供了灵活的地址管理。由于采用了单独的路由表，允许每个VPN使用单独的地址空间，称为VPN-IPv4地址空间，RD加上IPv4地址就构成了VPN-IPv4地址。很多采用私有地址的用户不必再进行地址转换NAT，NAT只有在

两个有冲突地址的用户需要建立Extranet进行通信时才需要。

在MPLS/BGP VPN中属于同一的VPN的两个site之间转发报文使用两层标签来解决，在入口PE上为报文打上两层标签：第一层（外层）标签在骨干网内部进行交换，代表了从PE到对端PE的一条隧道，VPN报文打上这层标签就可以沿着LSP到达对端PE；第二层（内层）标签，指示了报文应该到达哪个site，或者更具体一些到达哪一个CE。这样报文到达PE时剥掉了外层标签，这时，根据内层标签就可以找到转发的接口。

MPLS L2VPN

MPLS L2VPN 提供基于MPLS网络的二层VPN服务。使用基于MPLS的L2VPN解决方案，运营商可以在统一的MPLS基础网络架构上，提供基于不同媒介（包括ATM、FR、VLAN、Ethernet、PPP等）的二层VPN服务。同时这个MPLS网络仍然可以提供通常的IP、三层VPN、流量工程和QoS等其他服务，极大地节省网络建设的投资。

简单来说，MPLS L2VPN就是在MPLS网络上透明传递用户的二层数据，从用户的角度来看这个MPLS网络就是一个二层的交换网络，通过这个网络可以在不同站点之间建立二层的连接。

对于MPLS二层VPN，网络运营商负责给二层VPN用户提供二层的连通性，不需要参与VPN用户的路由计算，在提供全连接的二层VPN时和传统的二层VPN一样（如ATM PVC提供的VPN），存在N方问题。每个VPN的CE到其它的CE都需要在CE与PE之间分配一条连接。对于PE设备来说，在一个

VPN有N个Site的时候，CE-PE必须有N-1个物理或逻辑端口连接。

在MPLS L2VPN中，CE、PE、P的概念与BGP/MPLS VPN一样，原理也很相似：它也是利用标签栈来实现用户报文在MPLS网络中的透明传送，外层标记（称为tunnel标记）用于将报文从一个PE传递到另一个PE，内层标记（在MPLS L2VPN中称为VC标记）用于区分不同的VPN中的不同连接，接收方的PE根据VC标记决定将报文传递给哪个CE。

由于MPLS L2VPN中PE设备不参与用户的路由处理，因此它的可扩展性比L3VPN要好得多。MPLS L2VPN的可扩展性只与PE能连接的VPN用户数目相关，但是作为代价L2VPN的灵活性要差一些，无法实现Extranet。

当前MPLS L2VPN 还没有形成正式的标准，IETF 的PPVPN (Provider-provisioned Virtual Private Network) 工作组制订了多个框架草案，其中最主要的两种称为Martini草案和Kompella草案。Martini草案是通过LDP扩展实现MPLS L2VPN，而Kompella草案则是通过MP-BGP扩展实现，两种二层VPN采用的封装协议都是 Draft-martini-l2circuit-encap-mpls。

MPLS及LDP协议基础

贾欣武



MPLS基础

MPLS的产生及现状

MPLS概念的最初提出是为了提高转发效率。因为当时IP转发大多靠软件进行，在转发的每一跳都要进行至少一次最长匹配查找，操作复杂导致转发速度比较慢。有些厂商借鉴ATM的转发方式来简化IP转发过程，由此产生了一种结合IP和ATM的优势于一身的新技术—MPLS。在当时的条件下这可以说是一个很大的创举，其优势也是显而易见的，但后来IP转发领域有很多新技术产生，如硬件转发与网络处理器的出现，导致MPLS的速度优势体现不出来，纯MPLS转发在实际应用中几乎没有用武之地。

但MPLS是一个很有“潜力”的技术，可灵活扩展。很多新的应用依靠纯IP转发实现起来有很大的难度，但用MPLS再结合其它技术就可以实现，如：BGP/MPLS VPN、流量工程等技术就是对MPLS灵活扩展的结果。当前，MPLS越来越受重视，成为当今网络技术的热点，还有一些新的应用需求也正在利用MPLS来实现。

MPLS相关概念

1. Label

即标签，在帧模式链路上，Label位于二层头与IP报文之间，一个Label头的结构如下：

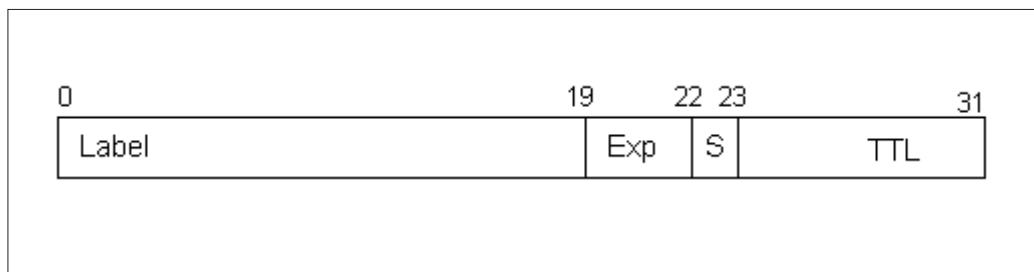


图 1 标签格式



Label : 标签值，长度为20bit，标签值是标签转发表的关键索引

Exp : 用于QoS，长度为3bit，作用与Ethernet802.1p值相似

S : 栈底标志，长度为1bit，如果有多个Label时，在栈底的Label的S位置“1”，其它为“0”，只有一个Label时S位置“1”

TTL : 存活时间，8bit，与IP报文中的TTL值相似，这个值从IP报文头的TTL域拷贝过来，每进行一次Label交换时，外层Label的TTL值就减“1”

需要注意的是一个MPLS报文可以有多个Label，靠近二层头的Label为栈顶Label，靠近IP报文的Label为栈底Label，LSR执行Label交换时总是基于栈顶Label。有多个Label时，每个Label都包括以上完整的32bit，并不是其它的Label只包括20bit的Label值，如下图所示：

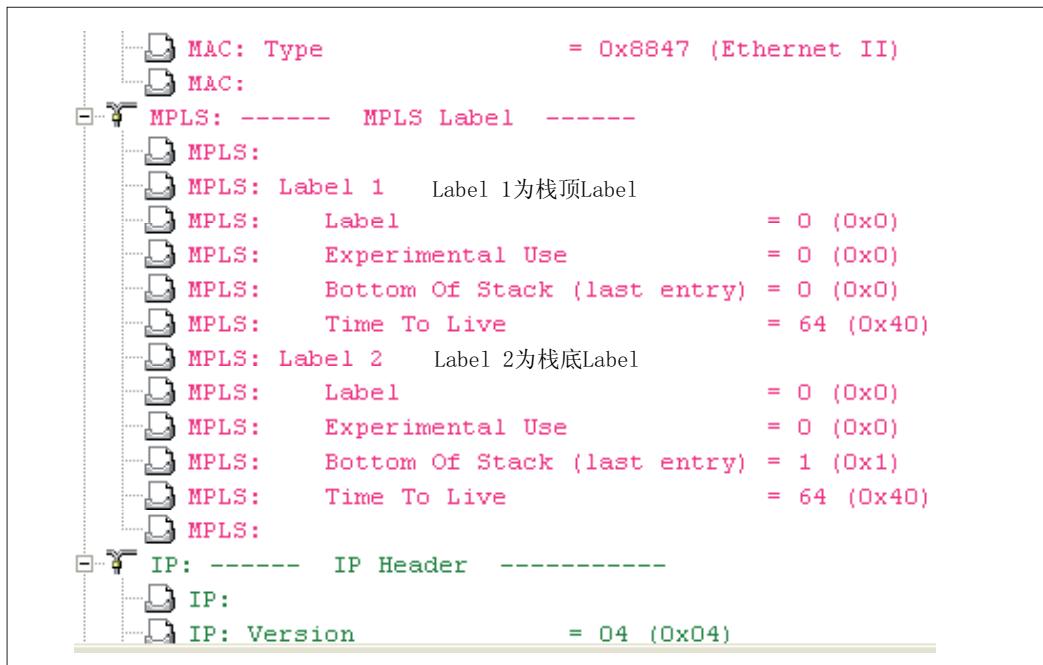


图 2 多层标签

2.LSR (Label Switching Router)

具有标记分发能力和标记交换能力的路由器。

3.LER (Label Edge Router)

标记边缘路由器，处在MPLS网络的边缘，负责将进入MPLS网络的报文或帧对应到具体的FEC并打上Label，变成MPLS帧转发；将离开MPLS网络的帧去掉Label还原成原来的报文或帧再查找相应的转发表转发。

4.FEC (Forwarding Equivalence Class)

LSR认为具有相同转发处理方式的报文，使用同一个标签来标记这些报文。如：匹配相同目的IP前缀

的多个IP报文可属于一个FEC，由于这些报文在做IP转发时是相同的转发处理方式及路径，所以标记这些报文的时候用同一个标签。

5.PUSH（加标签）

在第一跳 Ingress LER 上在报文的二层头和三层头之间插入Label，或者中间LSR 在MPLS 报文的标签栈顶增加新的Label。

6.POP（弹出标签）

在最后一跳 Egress LER 上将报文中的Label全部去掉，还原成IP报文，或者中间LSR 去掉栈顶标签减少标签栈层次。

7.SWAP（交换标签）

在转发的过程中根据标签转发表中的LSP 替换报文中栈顶Label的过程。

8.LSP（Label Switched Path）

标记转发路径，也就是转发MPLS 报文的路径。

FEC 和标签交换路径（LSP），进而标记报文。而在MPLS 网络核心的LSR 采用基于标签的第二层交换，工作相对较简单。从这里就可以看出MPLS 的好处，虽然处在MPLS 网络边缘的LER 工作较复杂，但处在核心的LSR 只需要像FR 或ATM 交换机那样执行二层交换就可以了，根本不需要最长匹配和多次查找。

典型的MPLS 转发过程如下：

Step 1：所有LSR 启用传统路由协议（OSPF、IS-IS等），在LSR 中建立IP路由表

Step 2：由LDP 结合IP路由表来建立LSP

Step 3：Ingress LER 接收IP包，分析IP包头并对应到FEC，然后给IP包加上标记，根据标签转发表中的LSP 将已标记的报文送到相应的出接口。

Step 4：LSR 收到带有标记的报文，将只分析标记头，不关注标记头之上的部分，根据Label头查找LSP，替换Label，送到相应的出接口

……………(中途转发过程与Step 4类似)

Step n-1：倒数第二跳 LSR 收到带有标记的报文，查找标记转发表，发现对应的出口标签为隐式空标签或显示空标签，弹出标签，发送IP报文到最后一跳 LSR

Step n：在最后一跳Egress LER上执行三层路由功能，根据报文的目的IP地址转发

MPLS转发方式

MPLS 技术综合了第二层交换和第三层路由的功能，将第二层的快速交换和第三层的路由有机地结合起来。MPLS 网络边缘的LER 主要完成以下工作：三层路由、分析IP包头用于决定对应的

LDP协议初步

LDP 协议在[RFC 3036] 中详细定义，LDP 的协议报文除Hello报文基于UDP 外，其它报文都是在TCP 之上，端口号为646。当发生传输丢包时，能够利用TCP 协议提供错误指示，实现快速响应和恢复。与BGP 相似，这种基于TCP 的可靠连接使得协议状态机较为简单。





报文格式

1. LDP PDU 头部

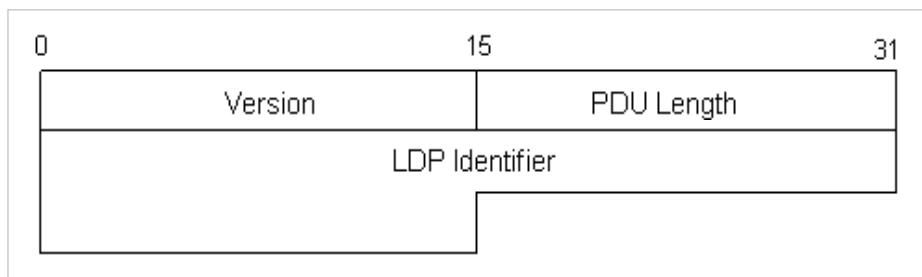


图 3 LDP PDU 头部

版本号 : 16bit, 目前LDP只有一个版本, 版本号始终为1 ;

PDU长度 : 为16bit, 值为LDP PDU头部以后的数据部分的长度, 不包括LDP PDU头部 ;

LDP Id : 长度为48bit, 前32bit为LSR-ID, 后16bit为标记空间标志, 全局空间为“0”, 局部接口空间为“1”。如 : 收到的LDP PDU中的LDP-ID为192.168.1.2:0, 表示对方的LSR-ID为192.168.1.2, 标签空间为全局空间。

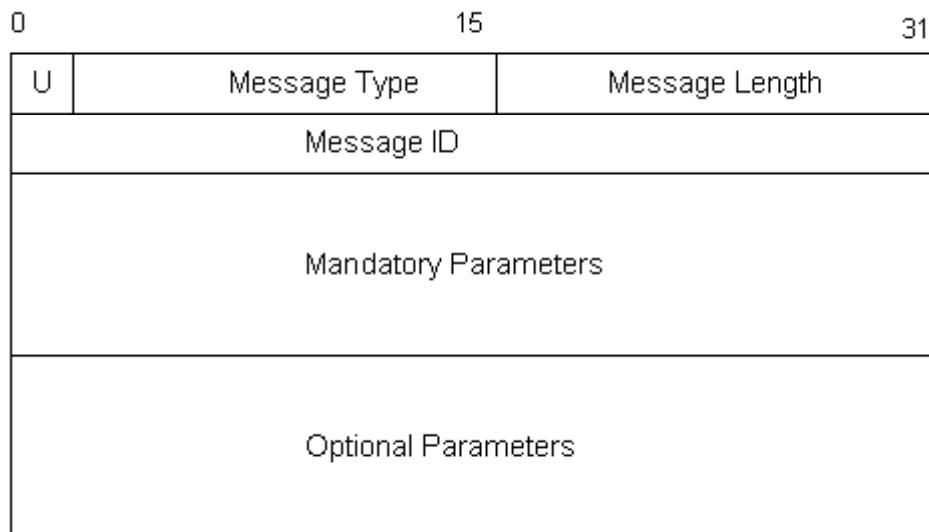


图 4 LDP消息格式

U : 这一位总是为“0”，代表可识别的消息，为“1”代表不可识别的消息；
 类型域：协议根据这个域识别不同的消息；
 长度域：指示出长度域之后的数据部分的长度；
 消息ID：用来唯一地标识这个消息，如果消息为Notification，则ID与导致产生Notification的消息ID相关联。

消息种类

邻居发现消息：在启用LDP协议的接口上周期性发送该消息

- Hello消息

会话建立和维护消息：用来建立和维护LDP会话

- Initialization消息
- KeepAlive消息

标签分发消息：用来请求、通告及撤销标签绑定

- Address message
- Address Withdraw message
- Label request message
- Label mapping message
- Label withdraw message



- Label release message
- Label abort request message
- 错误通知消息：用来提示LDP对等体在会话过程中的重要事件
- Notification消息

LDP相关概念

1. 标签空间

可分为全局标签空间和接口标签空间，全局标签空间表示LSR为特定目的地的FEC产生唯一的Label，接口标签空间表示LSR在每个接口上为特定目的地的FEC产生唯一的Label。在帧模式的链路上为全局标签空间，在信元模式的链路上为接口标签空间。LDP报文中的LDP-ID域中指示出标签空间值。

2. 上游和下游LSR

如图所示，对于LSR-C来说，FEC 172.16.1.0/24的上游路由器是LSR-B，对于LSR-B来说，FEC 172.16.1.0/24的上游路由器是LSR-A，下游路由器是LSR-C.

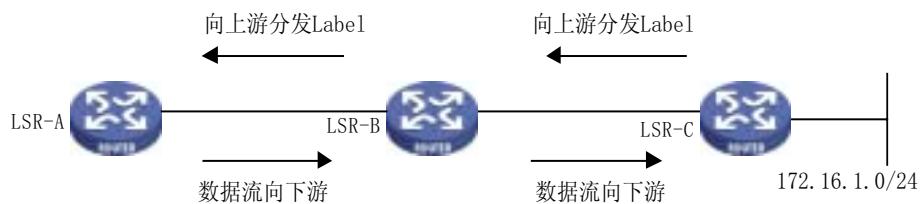


图 5 上/下游定义

3. 倒数第二跳弹出 (Penultimate Hop Popping)

在实际应用当中（如MPLS VPN），对于Egress LSR在弹出最外层Label后还需要进行其它较复杂的三层工作。而事实上最外层标签的作用在MPLS VPN的应用中只是为了将报文送到Egress LSR。因此，在倒数第二跳LSR已知报文下一跳的情况下，可以将最外层的标签弹出后转发到最后一跳LER，而不必进行标签替换。这样使得最后一跳LSR的工作相对简单了一些。因此在 [RFC 3032] 中规定，最后一跳LSR发给倒数第二跳LSR的标签为隐式空标签“3”。据此，收到标签“3”的上游LSR知道自己是该FEC的倒数第二跳，就知道自己在用该LSP转发Label报文时，应执行倒数第二跳弹出。

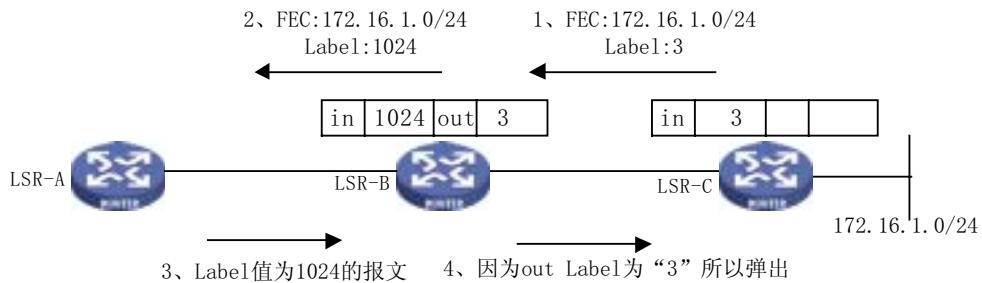


图 6 倒数第二跳弹出

标签分发方式

1. DU (Downstream Unsolicited)

下游LSR如果工作在DU方式（下游主动分发）下将根据某一触发策略向上游LDP邻居主动分发标签。下图中LSR-C标签分发触发策略是为直连32位掩码的路由分配标签，因此LSR-C通过Label mapping message向上游LDP邻居主动通告自己的直连路由172.16.1.1/32的标签，VRP系统缺省工作在DU方式。



图 7 下游主动分发



2. DOD (Downstream On Demand)

下游LSR如果工作在DOD方式（下游按需分发）下，只有在接收到上游LDP邻居的Label request message后才回应Label mapping message 分发标签（针对标记请求消息所指定的FEC）。下图中LSR-C工作在DOD模式下，LSR-A的触发策略生效（LSR-A转发到172.16.1.0/24的报文流量达到设定阀值）后将向172.16.1.0/24的下游发送标记请求消息Label request message（请求172.16.1.0/24的标签）。最终LSR-C收到请求，发送Label mapping message响应。

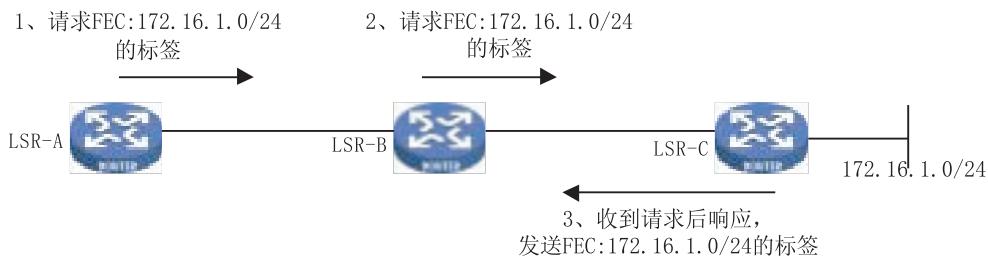


图 8 下游按需分发

标签控制方式

1. 独立控制方式 (Independent)

LSR如果工作在独立控制方式下，如果标签分发方式是DU，即使在没有获得下游标签的情况下也会直接向上游分发标签。在标签控制的方式上显得很“独立”，不依赖下游LSR；如果标签分发方式是DOD，发送标签请求的LSR的直连下游LSR会直接回应标签，而不必等待来自最终下游的标签。



图 9 独立控制方式

在上图中，在LSR-B上采用独立控制方式。LSR-B路由表中有172.16.1.0/24的路由，但没有收到下游来的标签绑定。由于LSR-B工作于独立控制方式，所以对路由表中的所有路由都向上游发送标签。继而，无论LSR-A工作在独立模式还是有序模式，将向上游继续发送标签。这时，如果有目的IP为172.16.1.0/24的报文进入LSR-A，它将采用MPLS转发。但数据到LSR-B后，由于没有关联172.16.1.0/24的LSP，所以采用传统IP转发。

2. 有序控制方式 (Ordered)

LSR 如果工作在有序控制方式下, 如果标签分发模式为DU, 则只有收到下游LSR 分发的标签时才会向自己的上游LSR 通告标签, 如果没有收到下游的标签映射则不向上游LSR 通告。VRP 系统缺省工作在有序方式。



图 10 有序控制方式

在上图中, LSR-B 路由表中有 $172.16.1.0/24$ 的路由, 但由于 LSR-B 没有收到下游的标签且工作在有序模式, 因而不向上游通告关于 $172.16.1.0/24$ 的Label。如果 LSR-A 收到目的 IP为 $172.16.1.0/24$ 的报文将采用传统 IP转发。可以看出, 在有序控制方式下, 是否向上游LSR 分发标签取决于自己是否收到下游LSR 的标签。

标签保留方式

1. 自由保留模式 (Liberal retention mode)

收到无效的Label通告后(没有对应的IP路由或路由通告与Label通告的下一跳不一致), 虽然不生成LSP, 但在标签绑定表里存储, 并且LSR 向上游通告其它FEC 的Label绑定时也不占用这些标签, 这种方式的优点是LSR 应对网络拓扑变化的响应较快, 缺点是浪费标签, 所有不能生成LSP 的Label通告都需要保留。

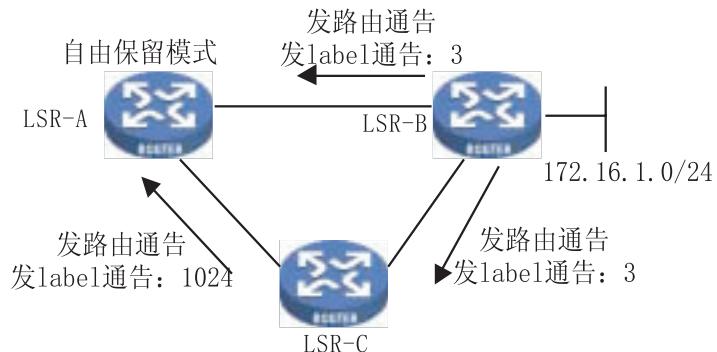


图 11 自由保留模式



在上图中LSR-A工作于自由保留方式，对于FEC为172.16.1.0/24将生成下一跳为LSR-B的LSP，LSR-C发来的Label通告将保留。如LSR-A和LSR-B之间的直连链路down掉，对于FEC：172.16.1.0/24的将很快生成下一跳为LSR-C的LSP。

2. 保守保留模式（Conservative retention mode）

工作于保守保留模式的LSR收到无效的Label通告后将不存放到标签绑定表里，在向上游通告Label时可以自由使用这些标签。保守保留模式的缺点是对拓扑变化的响应较慢，优点是节省标签。

Step1：互发Hello消息，Hello消息中包括LDP-ID和Transport Address。双方用Transport Address建立LDP会话，再进一步比较Transport Address确定由谁作为主动方发起TCP连接。Transport Address大的一方将作为主动方发起TCP连接。被动方等待对方发起连接。在下图中将由LSR-B作为主动方发起TCP连接。

Step2：TCP连接完成后由LSR-B发送Initialization消息来协商参数，包括：LDP协议版本、Label分发方式、HoldTime、接收者的LSR-ID等。

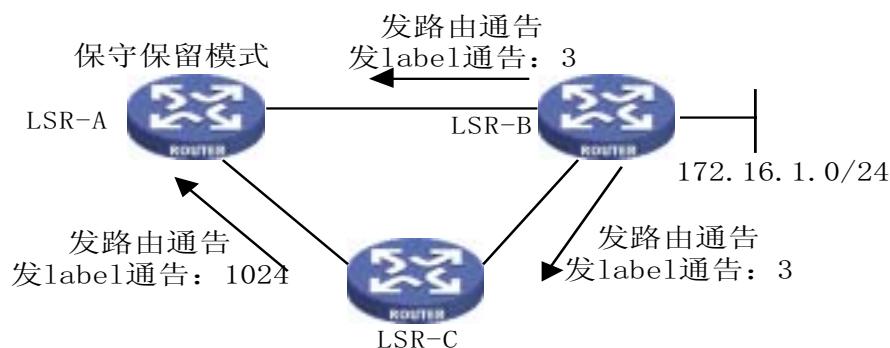


图 12 保守保留模式

在上图中，LSR-A工作于保守保留模式，对于FEC为172.16.1.0/24将生成下一跳为LSR-B的LSP，LSR-C发来的Label通告将不保留。如LSR-A和LSR-B之间的直连链路down掉，对于FEC：172.16.1.0/24将不能很快生成下一跳为LSR-C的LSP。

LDP会话建立过程

下图中用一个示例来演示LDP会话建立过程：

Step3：如果接收Initialization的LSR-A发觉对方的参数自己不能接受，则发送Notification消息结束会话；否则的话由LSR-A回应Initialization消息同时也发KeepAlive消息，两个消息可以在一个报文中同时携带。

Step4：如果LSR-B接受Initialization消息中携带的参数则发送KeepAlive，LDP会话成功建立。可以在同一个报文中携带KeepAlive消息和其它Session消息，如Address消息和Label mapping消息。

报文交互过程如下：

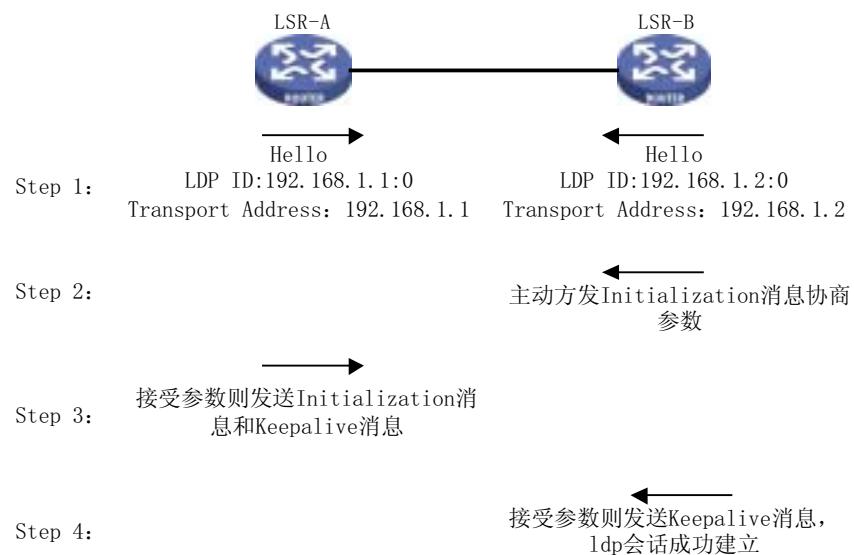


图 13 LDP会话建立过程

在此过程中，LSR检测到任何错误都会发Notification报文关闭连接。



状态机描述

1、NON EXISTENT状态：该状态类似BGP的Idle状态，为LDP会话的最初状态。在此状态双方发送Hello消息，选举主动方，在收到TCP连接建立成功事件的触发后变为INITIALIZED状态。

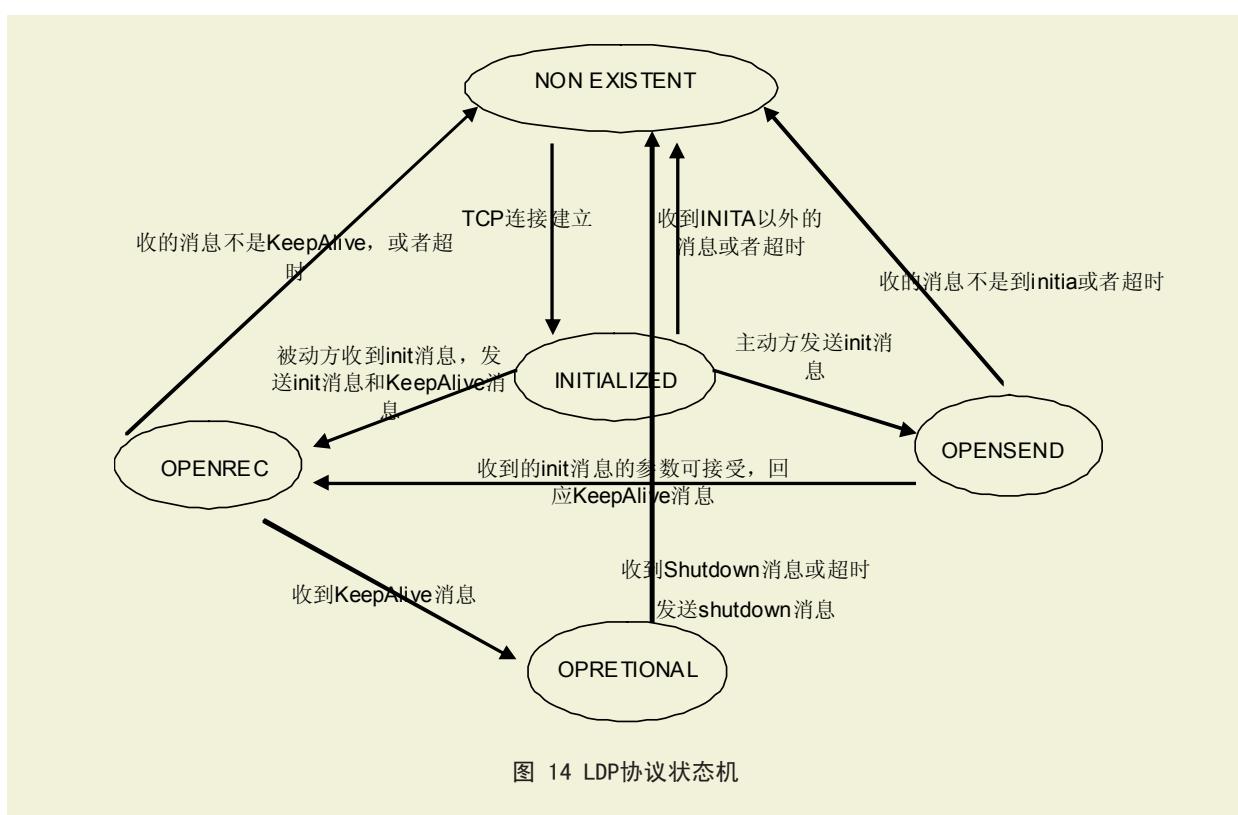
2、INITIALIZED状态：在该状态下分主动方与被动方两种情况，主动方将发送Initialization报文，转向OPENSENT状态，等待回应的Initialization消息；被动方在此状态等待主动方发给自己Initialization消息，如果收到的Initialization报文的参数可接收，则发送Initialization和KeepAlive转向OPENREC状态。主动方和被动方在此状态下收到任何非Initialization消息或等待超时，都会转向NON EXISTENT状态。

3、OPENSENT状态：此状态为主动方发送Initialization报文后的状态，在此状态等待被

动方回答Initialization消息和KeepAlive消息，如果收到的Initialization消息中的参数可以接受则转向OPENREC状态；如果参数不能接受或Initialization消息超时则断开TCP连接转向NON EXISTENT状态。

4、OPENREC状态：在此状态不管主动方还是被动方都是发出KeepAlive后的状态，在等待对方回应KeepAlive，只要收到KeepAlive消息就转向OPERATIONAL状态；如果收到其它消息或者KeepAlive超时，则转向NON EXISTENT状态。

5、OPERATIONAL状态：它是LDP Session成功建立的标志。在此状态下可以发送和接收所有其它的LDP消息。在此状态如果KeepAlive超时或者收到致命错误的Notification消息（Shutdown消息）或者自己主动发送Shutdown消息主动结束会话，都会转向NON EXISTENT状态。





MPLS LSP

Ping/Traceroute

陆强

前 言

在MPLS网络中，标签交换路径（LSP）转发用户数据失败时，控制平面常常没有办法进行有效的故障检测，于是就需要一种能在短时间内发现和隔离路由黑洞或者路由丢失等故障的方法。本文介绍了一种能够对MPLS标签交换路径进行故障检测的简易而有效的机制——MPLS LSP Ping/Traceroute。

本文档包括三部分内容：MPLS转发路径故障、MPLS LSP Ping/Traceroute原理以及MPLS LSP Echo的报文格式。

MPLS转发路径故障

相关信息。

MPLS转发路径故障的产生

在MPLS网络中LSP发生故障总体来说有四种情况：LDP邻居关系中断、MPLS（全局或者接口）错误的去使能、标签分发错误以及软硬件故障等等。

MPLS网络中运用传统的故障检测面临的问题

在传统IP网络中，我们利用IP Ping来进行网络连通性检测，用Traceroute进行逐跳的错误定位和路径跟踪。在MPLS网络，如果我们继续用传统的IP Ping和Traceroute进行故障检测会面临如下问题：

1. 传统的Ping并不能够对MPLS LSP的连通性进行有效的检测。传统Ping能通只能说明IP转发是正常的，而不能够说明LSP是没有问题的。在IP路由正常而LSP中断的情况下，传统Ping报文依旧可以通过IP转发到达目的地。

2. 传统的Traceroute并不能够对MPLS LSP故障进行有效的逐跳定位和返回LSP的相关信息。因为IP转发通并不能够代表LSP是通的，并且标准ICMP报文不能返回诸如标签栈、下游映射等LSP的

MPLS LSP Ping/Traceroute

与传统的Ping/Traceroute类似，MPLS LSP Ping/Traceroute同样是基于Echo request和Echo reply的模式；但是LSP Ping/Traceroute并不使用ICMP协议来实现，而是使用IPv4/IPv6的UDP协议来实现的。MPLS LSP Ping/Traceroute的基本思路是使用特定FEC转发类的分组来验证对应该FEC的LSP（从入口LSR到出口LSR）的完整性。

在LSP Ping Echo请求消息中携带所属FEC的信息。LSP Ping分组信息封装在UDP包中，包含序列号和时戳参数。MPLS在处理MPLS LSP Ping请求消息时采用了与该FEC分组相同的转发策略。在进行连通性测试时，分组将到达LSP的出口，出口LSR对该分组进行检查，验证该LSR是否是该FEC的真正出口。

Traceroute模式可以作为故障定位的一种手段，发起测试的LSR向目的LSR发送Traceroute分组，该分组的TTL初始值为1，步进值为1。这些LSR对该分组执行各种检查，进一步返回相关控制和数据平面的信息。

如果Ping失败可以采用Traceroute对故障进行定位，也可以通过周期性的Traceroute FEC验证实际数据转发路径和控制平面路径是否一致。



MPLS LSP

Ping/Traceroute原理

MPLS LSP Ping 原理

MPLS LSP Ping使用IPv4/IPv6的UDP协议来实现的，其中Echo Request的UDP端口为3503，需要说明的是只有使能MPLS的路由器才能够识别该端口号。我们以下图为例说明MPLS LSP Ping的一般过程。

文进行初始化，在IP头部填入127/8的目的地址，同时将10.10.10.10填入Echo Request报文中的Target FEC Stack中，然后查找标签转发表中对应项压入标签栈，将报文发送给R-2；

Step3：在R-2和R-3上，采用和该FEC相同的转发策略，将Echo Request当作普通MPLS数据报文进行转发；

Step4：如果在上述过程中，MPLS转发失败，

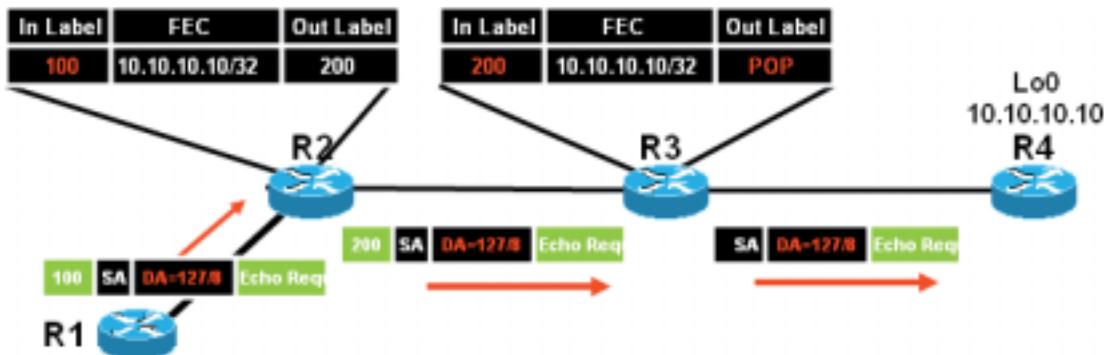


图 1 MPLS LSP Ping

假设我们在R-1上操作LSPPing10.10.10.10（10.10.10.10为R-4的Loopback接口地址），执行步骤如下：

Step 1：在R-1上，应用程序先查找这条LSP是否存在（特别的，对于TE的Tunnel我们查找这个Tunnel接口是否存在，而且CR-LSP是否已经建立好），如果不存在，直接返回错误信息，停止Ping；否则进入Step 2；

Step 2：在R-1上，应用程序对Echo Request报

MPLS LSP Ping 应用程序会作相应处理，回应的Echo Reply携带相应的错误码（错误码请参见MPLS LSP Echo报文格式部分）；

Step5：如果在上述过程中，MPLS转发正常，该Echo Request到达R-4；

Step6：在R-4上，应用程序检查Echo Request中的Target FEC Stack 包含的目的IP 10.10.10.10为自己的Loopback接口地址，回应正确的Echo Reply后，整个LSP Ping过程结束，完成LSP的连通

性检测。

从前面的例子我们可以看Echo Request报文的目的IP地址是127开头的地址，该地址是一个环回地址。为什么这样呢？这样做的目的是为了防止LSP断路时Echo Request进行IP转发，从而保证LSP的连通性测试。

MPLS LSP Traceroute 原理

和传统的Traceroute类似，MPLS LSP Traceroute通过连续发送一个TTL步进为1的Echo Request报文，让LSP沿途的每一个LSR都会收到TTL超时的Echo Request报文，同时回送一个携带下游信息（可选）以及相应返回码的Echo Reply给发送者。这样发送者就会得到该LSP沿途每一个节点的信息。我们以下图为例说明MPLS LSP Traceroute的一般过程。

Step 1：在R-1上，应用程序先查找这条LSP是否存在（特别的，对于TE的Tunnel我们查找这个Tunnel接口是否存在，而且CR-LSP是否已经建立好），如果不存在，直接返回错误信息，停止Ping；否则进入Step 2；

Step 2：在R-1上，应用程序对Echo Request报文进行初始化，在IP头部填入127/8的地址作为目的地址，同时将10.10.10.10填入Echo Request报文中的Target FEC Stack中。Echo Request报文可以包含Downstream Mapping TLV（用来携带LSP在当前节点的下游信息，主要包括下一跳地址、出标签等），然后查找标签转发表中对应项压入标签栈并且将TTL设置为1，将报文发送给R-2；

Step 3：在R-2上，将Echo Request中TTL减1为0发现TTL超时，应用程序检查是否存在该LSP，同时检查报文中Downstream Mapping TLV中的值（下一跳地址，出标签）：

- ①比较其中的下一跳地址是否为本地接口地址；
 - ②比较其中的出标签是否为本地该FEC的入标签；
- 如果存在该LSP并且比较值均为真，则回应正

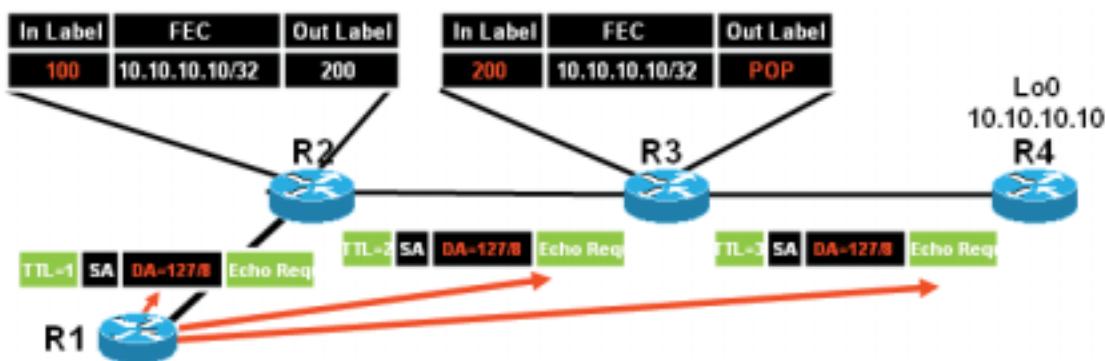


图 2 MPLS LSP Traceroute

假设我们在R-1上操作LSP Traceroute 10.10.10.10（10.10.10.10为R-4的Loopback接口地址），执行步骤如下：

确的Echo Reply，并且报文中必须携带Downstream Mapping TLV给发送者进行检查分析。如果上述检查不为真，则返回带有错误码的Echo Replay的报文；

Step4：在R-1上，应用程序对Echo Reply进行相关处理，对第二个Echo Request进行初始化，初始化过程基本和Step 1中的初始化一样，值得指出的是这时候应用程序会将收到Echo Reply报文中的Downstream Mapping TLV复制到第二个MPLS Echo Request中并且TTL=2，然后发送给R-2；

Step5：在R-2上，将TTL减1，采用和该FEC相同的转发策略，将Echo Request当作普通MPLS数据报文发送给R-3；

Step6：在R-3上，进行和Step3相同的程序步

骤，返回Echo Reply给R-1；

Step7：在R-1上，进行和Step4相同的程序步骤，发送TTL=3的Echo Request给R-2，经R-3到达R-4；

Step8：在R-4上，进行和Step3相同的程序步骤，返回Echo Reply给R-1，因为R-4已经是该LSP的出口节点，因此不返回任何下游信息。至此整个LSP Traceroute 过程结束，在入口处也得到该LSP沿途每一个节点的信息。

MPLS LSP Echo报文格式

0	1	2
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8
+-----+-----+-----+	+-----+-----+-----+	+-----+-----+-----+
Version Number		Must Be Zero
+-----+-----+-----+	+-----+-----+-----+	+-----+-----+-----+
Message Type Reply mode Return Code Return S		
+-----+-----+-----+	+-----+-----+-----+	+-----+-----+-----+
	Sender's Handle	
+-----+-----+-----+		+-----+-----+-----+
	Sequence Number	
+-----+-----+-----+		+-----+-----+-----+
	TimeStamp Sent (seconds)	
+-----+-----+-----+		+-----+-----+-----+
	TimeStamp Sent (microseconds)	
+-----+-----+-----+		+-----+-----+-----+
	TimeStamp Received (seconds)	
+-----+-----+-----+		+-----+-----+-----+
	TimeStamp Received (microseconds)	
+-----+-----+-----+		+-----+-----+-----+
	TLVs ...	
.		
.		
.		
+-----+-----+-----+		+-----+-----+-----+

Version Number :

MPLS Echo Packet的版本号，当前版本为1。

Message Type :

消息类型，用来标识该MPLS Echo Packet是Echo Request还是Echo Reply。消息类型按如下定义：

Value	Meaning
---	---
1	MPLS Echo Request
2	MPLS Echo Reply

Reply Mode :

回应模式，指示Replier Router用什么方式来回应这个Echo Request。回应模式按如下定义：

Value	Meaning
---	---
1	Do not reply
2	Reply via an IPv4/IPv6 UDP packet
3	Reply via an IPv4/IPv6 UDP packet with Router Alert

下面分别对这三种模式进行简要介绍：

Do not reply：发送方不要求接收方作出应答；

Do not reply模式可以用作单向的连通性测试。

Reply via an IPv4/IPv6 UDP packet：发送方要求接收方作出应答。

Reply via an IPv4/IPv6 UDP packet with Router Alert：发送方要求接收方不仅用UDP来回答，同时要启动Router Alert。Router Alert并不是MPLS自己的功能，是IP的功能。一般IP头有20字节，有时会在后面加一些Option，Router Alert就是其中一个Option。LSR收到一个IP包里面含有Router Alert这个Option，它并不会马上转发出去，而是交给上层协议栈进一步处理。

Return Code :

返回码，回应的LSR对Echo Request报文中的Target FEC Stack进行验证后返回相应的代码值给发送者。

返回码按如下定义：

Value	Meaning
---	-----
0	The Error Code Is Contained in the Error Code TLV
1	Malformed echo request received
2	One or more of the TLVs was not understood
3	Replies Router Is an Egress for the FEC
4	Replies Router Has No Mapping for the FEC
5	Replies Router Is Not One of the “Downstream Routers”
6	Replies Router Is One of the “Downstream Routers”，and Its Mapping for this FEC on the Received Interface Is the Given Label

Sender's Handle :

发送者句柄，是用来标识一个MPLS Echo的，其值是在应用程序发送一个MPLS Echo Request时随机生成的。单次的LSP Ping操作可以产生多个Echo Request，但是这些Echo Request所包含的Sender's Handle的值是相同，即单次LSP Ping操作仅能产生一个Sender's Handle的Echo Request。

Sequence Number :

序列号，Sequence Number同样是用来标识MPLS Echo的，它是一个进程的概念，进程内有效，可以用来检测丢失的Reply的个数，从而可以对网络进行延时和抖动统计。单次LSP Ping操作可以产生多个Sequence Number，其值一般从零开始逐一递增。

Timestamp :

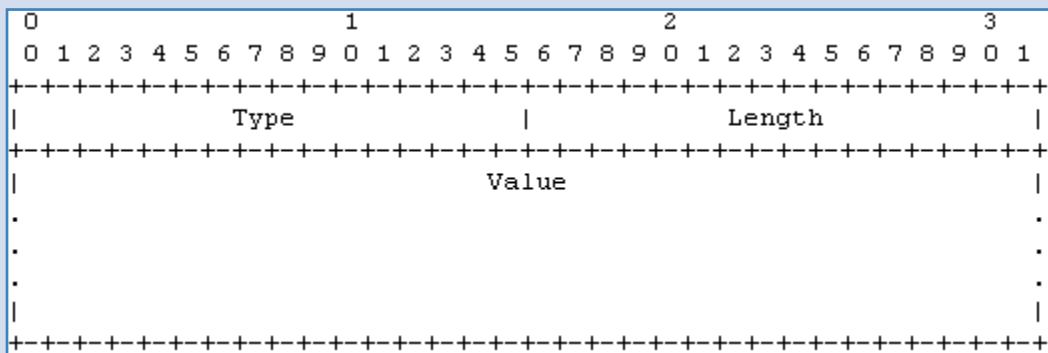
时间戳，采用NTP协议的时间格式，包含两部分：收到的时间戳和发送时间戳；可以用来计算报文从一个节点到另一个节点所需要花费的时间。



TLVs :

TLVs (Type-Length-Value tuples) 的报文格式按下

图定义：



其中类型 (Type) 值按如下定义,

Value	Meaning
-----	-----
1	Taget FEC Stack
2	Downstream Mapping
3	Pad
4	Error Code
5	Vendor Enterprise Code

长度 (Length) 包括Length字段后该TLV (包括子TLV) 的所有内容, 单位为字节。

值 (Value) 值由类型决定，它可以嵌套多个子TLV。

Target FEC Stack :

所谓的Target FEC Stack就是所要Ping或者Traceroute的对象，真正Ping/Traceroute的对象并不是填充在目标IP地址中，而是填充在Target FEC Stack中的。Target FEC Stack包含着下面一系列的sub-TLV，按如下定义：

Sub-Type	Length	Value Field
---	-----	-----
1	5	LDP IPv4 Prefix
2	17	LDP IPv6 Prefix
3	20	RSVP IPv4 Session Query
4	56	RSVP IPv6 Session Query
5		Not Assigned
6	13	VPN IPv4 Prefix
7	25	VPN IPv6 Prefix
9	10	L2 Circuit ID

我们看这样一个例子，假设今天要Ping一个LDP IPv4 Prefix（10.10.10.10/32）地址，那么该Echo Request报文中的Target FEC Stack的报文格式如下：

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Type = 1 (LDP IPv4 FEC)	Length = 5		
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
IPv4 prefix(10.10.10.10)			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
PreLength(32) Must Be Zero			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+

大家注意这个Target FEC Stack TLV仅仅会出现在Echo Request里面有，不会出现在Echo Reply里。

Downstream Mapping：

Downstream Mapping TLV主要包含了下游接口地址以及出标签栈等信息。在一个Echo Request中至多可以包含一个Downstram Mapping TLV。当接收LSR收到一个包含 Downstram Mapping TLV的Echo Request报文时，它回应请求报文时也应该包含相应的Downstream Mapping TLV。假如回应的LSR是该Echo Request中所请求的目的FEC的出口节点，那么该LSR在发送的Echo Reply报文中就不应该包含任何下游映射对象类，因为对于该FEC来说已经是终点LSR，不存在下游节点。

Downstream Mapping TLV的报文格式按如下定义：

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
MTU Address Type Resvd (SBZ)			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Downstream IP Address (4 or 16 octets)			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Downstream Interface Address (4 or 16 octets)			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Hash Key Type Depth Limit Multipath Length			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
.	.	.	.
.	(Multipath Information)	.	.
.	.	.	.
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Downstream Label Protocol			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
.	.	.	.
.	.	.	.
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+
Downstream Label Protocol			
+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+	+-----+-----+-----+-----+

Pad

Pad TLV当前并未对其用途进行详细地说明，
仅对其作如下定义：

Value	Meaning
----	-----
1	Drop Pad TLV from reply
2	Copy Pad TLV to reply
3	Reserved for future use

Error Code :

Error Code TLV当前未定义。

Vendor Enterprise Code :

Vendor Enterprise Code TLV 包含各自厂家的
私有信息，长度一般为4字节，其值为SMI Private
Enterprise Numbers，由IANA指派。

MPLS L3 VPN

基础

蔡金龙

VPN技术概述

VPN技术的产生及现状

VPN (virtual private network) 技术的定义

多个站点之间的客户通过部署在相同的基础设施之上进行互联，它们之间的访问及安全策略与专用网络（专线）相同。通俗点说就是使用公共网络设施实现私有的连接，各个私有连接在公共网络上对其他的私有网络是不可见的。相对于专线的物理隔离技术来说，VPN技术更多意义上是一种逻辑隔离技术。它主要是在客户要求各个站点通过服务提供商公共网络进行连接的需求背景下，由专线网络的概念引申而来的。

任何符合如下两个条件的网络我们都可以泛泛地称它为VPN网络：

- 1、使用共享的公共环境实现各个私有网络的连接；
- 2、不同的私有网络间（除非有特殊要求，如两个公司间的互访要求）是相互不可见的。

VPN技术的分类

VPN的概念早在10多年前便已经提出，至今经历了较长的演进过程。VPN的概念最早是从专线引发的。专线网络具有以下特点：

- 1、安全性高，线路为客户专用，不同用户间是物理隔离的；
- 2、价格昂贵；
- 3、带宽浪费严重。

正因为专线网络具有一些固有缺陷，随着统计复用技术的出现，一些新的共享带宽技术逐渐替代了专用线路，并可以提供与专线相同的服务。如帧中继、X.25技术等，通过VC在公共的交换设备上提供虚链路，实现用户的私有通信。由于是共享带宽，所以价格比专线便宜很多。这些技术构成了早期的VPN网络。随着新技术（如加密技术、隧道技术、MPLS技术）的不断涌现，及新的客户需求的出现，目前VPN的概念变得越来越复杂。所以引入了VPN的分类，以区分各种不同的VPN技术。

- VPN技术主要从四个方面进行分类：
- 1、VPN要解决的业务问题；
 - 2、服务提供商在哪一层与客户交换拓扑信息；
 - 3、在服务提供商中用于实现VPN服务的第二层或第三层技术；
 - 4、网络的拓扑结构。

每一个分类与其它的分类间都不是独立的，都是有相互关联的，它们共同决定了VPN网络的架构。

VPN是一门很复杂的技术，相关的知识点很多。在本文中我们主要介绍一下和MPLS VPN相关的VPN知识点。其它的内容我们暂不介绍，大家可以查阅相关资料获取感兴趣的信息。

VPN的模型及典型实现技术

首先我们介绍一下VPN网络中的几个关键角色：

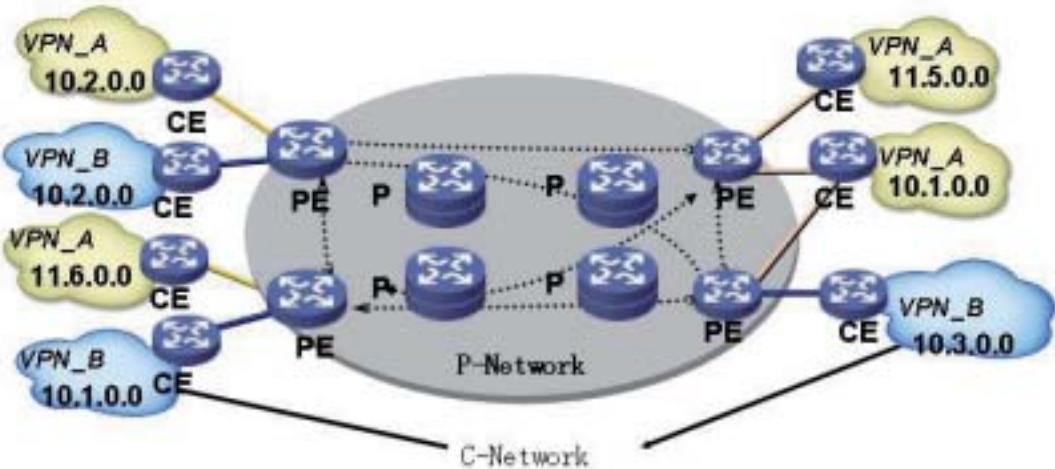


图1 VPN网络结构示意图

如上图所示，构成一个VPN网络的关键组件如下：

CE (Customer Edge)：直接与服务提供商相连的用户设备。

PE (Provider Edge Router)：指骨干网上的边缘设备（如路由器、ATM交换机、帧中继交换机等），与CE相连，主要负责VPN业务的接入。

P (Provider Router)：指骨干网上的核心路由器，主要完成路由和快速转发功能。P设备根据网络结构及规模可有可无。

任何一个VPN网络都是由这几个组件全部或部分组成的，下面我们来介绍一下VPN网络的几种主要模型。

VPN的主要模型为两种，分别是：

Overlay VPN—覆盖VPN模型

Overlay VPN的主要特点是客户的路由协议总

是在客户设备之间交换，而服务提供商对客户网络的内部结构一无所知。

典型的Overlay VPN技术有：

第二层隧道技术如X.25、帧中继、ATM；第三层隧道技术如IP-over-IP隧道技术等。其中IP-over-IP隧道技术通过专用IP主干或INTERNET来实现覆盖VPN网络，最常用的技术是GRE和IP-SEC等。

Overlay VPN的主要缺陷是：

连接性比较复杂时管理开销非常大；

要正确提供VC的容量，必须了解站点间的流量情况，比较难统计。

Peer-to-Peer VPN一对等VPN模型

Peer-to-Peer VPN的主要特点是服务提供商的PE设备直接参与CE设备的路由交换。该VPN模型的实现依据是：如果去往某一特定网络的路由未被



安装在路由器的转发表中，在那台路由器上，该网络不可达。实施Peer-to-Peer VPN的前提是所有CE端的地
址是全局唯一的！

典型的Peer-to-Peer VPN技术有：

1. 专用PE接入技术
2. 共享PE接入技术

其中，共享PE Peer-to-Peer VPN（如下图所示）的主要特点是：

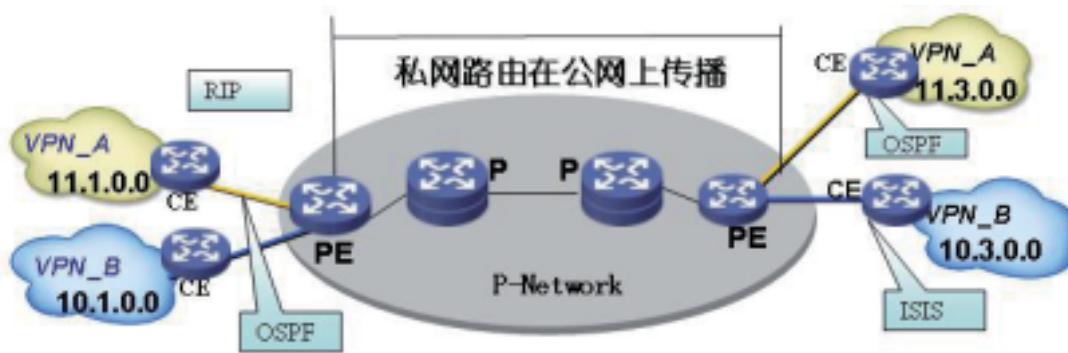


图2 共享PE Peer-to-Peer VPN组网图

1. 所有VPN用户的CE都连到同一台PE上，PE与不同的CE之间运行不同的路由协议（或者是相同路
由协议的不同进程，比如OSPF）。
2. 由始发PE将VPN路由发布到公网上，在接收端的PE上将这些路由过滤后再发给相应的CE设备。
3. 共享路由接入方式为避免不同VPN用户间的路由外漏，需要配置大量的ACL进行过滤，对于设备
的性能及管理开销都会有很大的负担。

专用PE Peer-to-Peer VPN（如下图所示）的主要特点是：

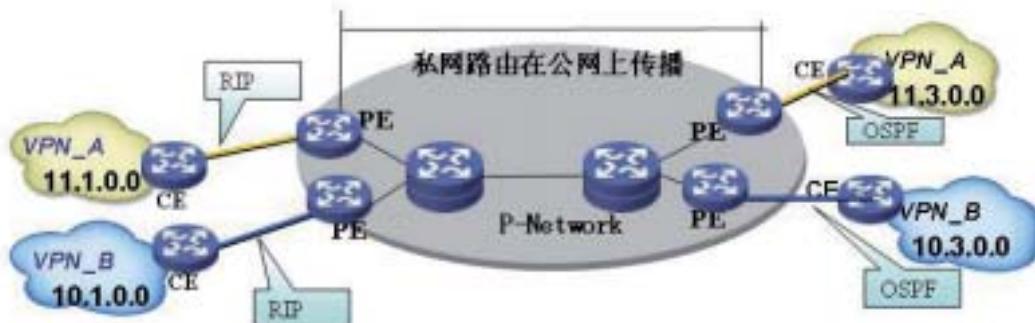
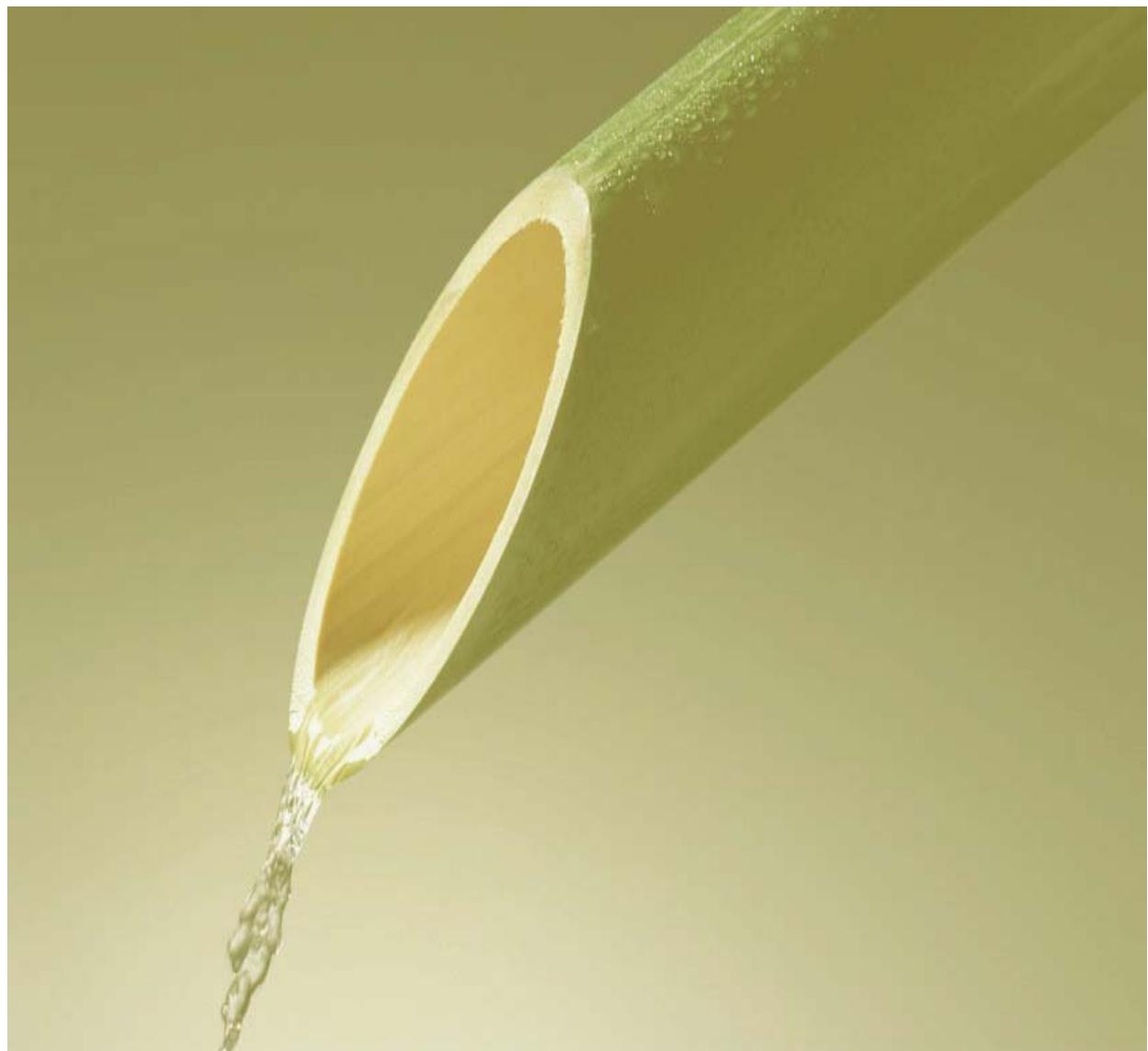


图3 专用PE Peer-to-Peer VPN组网图

总 结

现有VPN技术基本满足了客户私有网络通过公共基础设施互联的要求，但同时也存在一些固有缺陷，典型的是覆盖模型和对等模型不能很好的兼容。但客户往往要求其VPN网络可以具备这两者的优点，如下图3所示的要求：对于黄色站点和绿色站点的VPN客户都要求可以使用私有地址（不同VPN的私有地址可以是重复的），并且可以像对等模型VPN那样动态建立隧道。这种需求目前的VPN技术（覆盖型、对等型）是实现不了的，所以急需一种新技术可以整合这两种VPN模型的优点，满足客户的需求。MPLS VPN在这种背景下孕育而生。下面我们来介绍一下MPLS L3 VPN的一些相关技术。



MPLS L3 VPN 技术综述

MPLS L3 VPN技术综述

MPLS VPN技术产生背景

现有VPN的一些固有的缺陷导致客户组网的很多需求不能满足，实施比较复杂。MPLS VPN技术将两种VPN模型完美的整合到一起，推动了VPN的继续发展。下面对MPLS L3 VPN的基础知识进行介绍。

MPLS L3 VPN体系结构综述

MPLSVPN的出现主要是为了要解决传统VPN技术的一些固有缺陷，这有很多技术问题需要解决的，其中最重要的是地址重叠的问题。必须有一种技术可以保证不同的用户VPN可以使用相同的私有地址空间，而且可以在公共的骨干网络上互相不影响地交换数据。要解决地址空间重叠的问题主要有以下几个问题：

1. 本地路由冲突问题，即：在同一台PE上如何区分不同VPN的相同路由。
2. 路由在网络中的传播问题，两条相同的路由，都在网络中传播，对于接收者如何分辨。
3. 报文的转发问题，即使成功的解决了路由表的冲突，但是当PE接收到一个IP报文时，它又如何能够知道该发给哪个VPN？因为IP报文头中唯一可用的信息就是目的地址。而很多VPN中都可能存在这个地址。

从上述这些技术难点来看，主要问题都存在于和路由相关的特性，所以要解决这些问题必须从路由协议上进行考虑。但现有的路由协议都不具备

解决这些问题的条件，所以必须通过对现有的路由协议进行改造来实现。选择一个合适的路由协议来实现改造就成了首先要解决的问题。作为候选的路由协议必须可以适应数量巨大的VPN路由，而且要求协议具有良好的可扩展性。具备条件的协议一定是基于TLV元素的（扩展方便）。分析现有的路由协议，OSPF是应用最广泛的路由协议，但是这个协议的扩展性比较差，而且它是一种链路状态路由协议，需要对收到的LSA进行大量的计算，很难适应公共网络上数量巨大的VPN路由。RIP从扩展的角度上看它是TLV架构的，扩展性应该还可以，但是RIP的运行机制不适合大型网络。其它的路由协议也都有相应的不适合的地方，最终这个重任就落到了现有的INTERNET使用的骨干路由协议BGP身上。BGP具备很多特点使其十分适合改造以满足VPN网络上地址重叠的问题，它的特点如下：

1. 公共网络中VPN路由数目非常大，BGP是目前唯一支持大量路由的路由协议；
 2. BGP也是为在不直接相连的路由器间交换信息而设计的，这使得P路由器中无需包含VPN路由信息；
 3. BGP可以运载附加在路由后的任何信息，作为可选的BGP属性，任何不了解这些属性的BGP路由器都将透明的转发它们（当然这些属性都是可传递的），这使在PE路由器间传播路由非常简单。这里主要是考虑到可以通过扩展属性的方式来标识不同VPN中相同的路由。
- 正因为BGP具备了如上的优势，所以解决前面提到的技术难点的重任就落到了它的身上。通过对现有BGP协议的改造，前面提到的几个需要解决的问题基本可以迎刃而解了，解决思路如下：

- 1、本地路由冲突问题，可以通过在同一台路

由器上创建不同的路由表解决，而不同的接口可以分属不同的路由表中，这就相当于将一台共享PE模拟成多台专用PE。

2、可以在路由传递的过程中为这条路由再添加一个标识，用以区别不同的VPN。

3、由于IP报文的格式不可更改，但可以在IP头之外加上一些信息，由始发的VPN打上标记，这样PE在接收报文时可以根据这个标记进行转发。

本地路由冲突问题的解决思路

本地路由冲突问题的解决思路主要是从专用PE对等VPN模型上借鉴过来的。专用路由器方式分工明确，每个PE只保留自己VPN的路由。P只保留公网路由。而现在的思路是：将这些所有设备的功能整合在一台PE上完成。如下图所示：

由器包括如下元素：

1、一张独立的路由表，当然也包括了独立的地址空间；

2、一组归属于这个VRF的接口的集合，特定的VRF通过数据包从哪个接口接收来进行判断；

3、一组只用于本VRF的路由协议。

对于每个PE，可以维护一个或多个VRF，同时维护一个公网的路由表（也叫全局路由表）。多个VRF实例相互分离独立。

相对于复杂的ACL配置，多转发表的机制大大简化了PE路由器为每个VPN路由信息提供隔离的支持。

实现VRF本地路由区分并不困难，但是解决远端路由区分就要在PE上使用特定的策略规则来协调各VRF和全局路由表之间的关系。这个主要是指

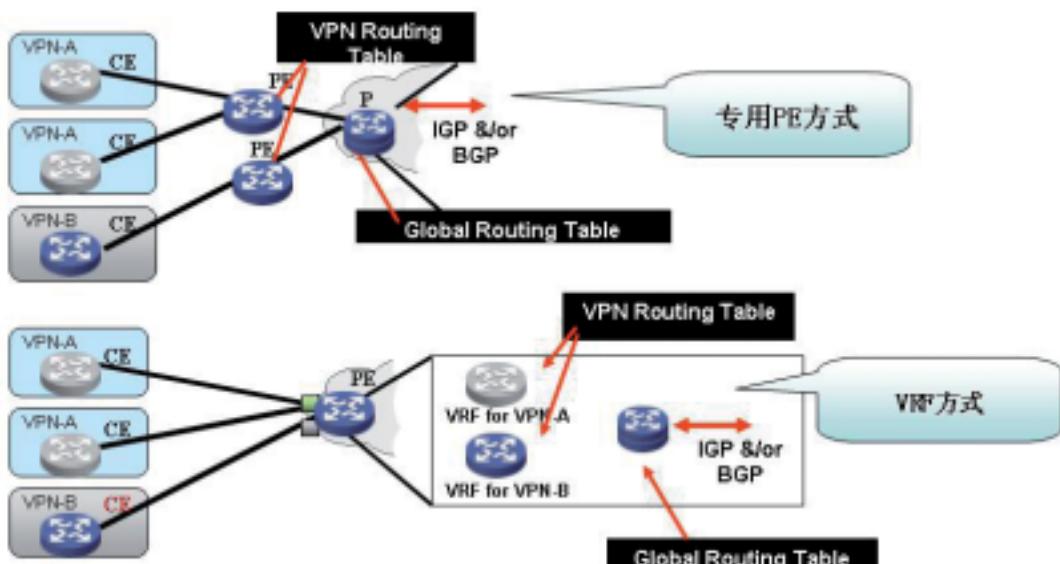


图4 本地路由冲突问题解决思路

具体的操作是在PE设备上划分不同的VRF（VPN路由转发实例-VPN Routing & Forwarding Instance，我司称之为VPN Instance），每一个VRF可以看作虚拟的路由器，好像是一台专用的PE设备。该虚拟路

PE设备如何分辨收到的路由是属于VPN的路由还是公共网络上的全局路由。解决的办法是给VPN路由加上一个特定的标记，由不同的标记来标示不同的路由，并由PE设备根据这个标记来判断该路由应该



被写入哪个VRF中。

要解决标记的问题我们需要从BGP的属性上想解决办法。BGP协议的路由属性是十分灵活的而且扩展性极佳。回忆一下BGP的各种属性，团体属性的特点最适合用于解决这个问题。

团体属性是一个任选可传递属性，它主要用于简化策略的执行。团体属性标明一个目的地作为一些目的地团体中的一个成员，这些目的地共享一个或者多个共同的特性。也就是说团体属性是针对具体路由的，而与之功能类似的对等体组是针对对等路由器的，在这并不适用。所以我们可以通过对团体属性进行改造来解决我们所面临的问题。改造后的团体属性功能上和以前是一样的，只不过为了区别于传统的团体属性，我们给它一个新的名字叫—RT。这样PE设备就可以通过在特定路由条目

扩展的community有如下两种格式：其中type字段为0x0002或者0x0102时表示RT。RT是作为BGP路由的属性进行传递的，并且它是任选可传递的属性。扩展的团体属性值是32bit的，可以提供更多的路由区分。

RT的本质是每个VRF表达自己的路由取舍及喜好的方式。可以分为两部分：Export Target与Import Target；前者表示了我发出的路由的属性，而后者表示了我对哪些路由感兴趣。同时，RT的应用是比较灵活的，每个RT Export Target与Import Target都可以配置多个属性，例如：我对红色或者蓝色的路由都感兴趣。接收时是“或”操作，红色的、蓝色的以及同时具备两种颜色的路由都会被接受。这样就可以实现非常灵活的VPN访问控制。下面我们通过一个实例分析来说明一下RT的应用：

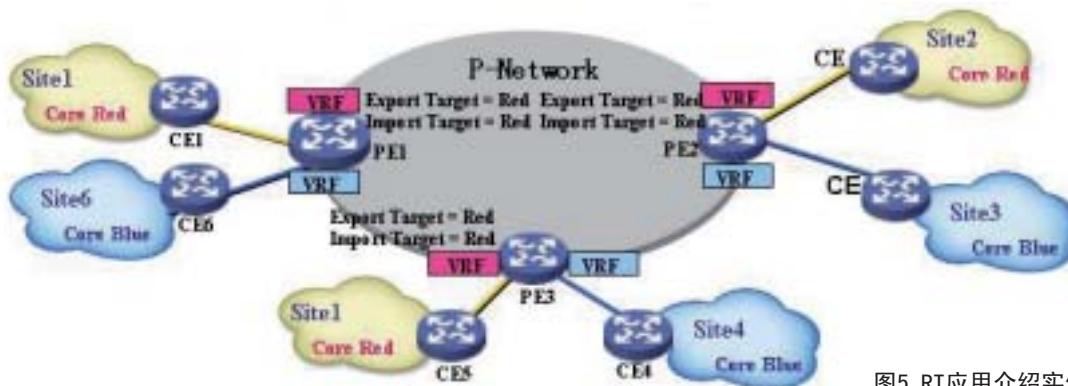


图5 RT应用介绍实例图

中加入RT属性来区分不同VRF的路由。

通过对比，我们来看一下团体属性是如何改造使之成为RT属性的：

传统的团体属性格式如下：

Type	AS#(16bit)	Value(16bit)
------	------------	--------------

扩展后的团体属性（也就是 RT）格式如下：

Type(0x0002)	AS#(16bit)	Value(32bit)
--------------	------------	--------------

Type(0x0102)	IP address(16bit)	Value(32bit)
--------------	-------------------	--------------

如图所示，每个红色公司的站点与PE路由器上的红色VRF关联。PE为每个红色VRF配置一个全局唯一的RT（RED），作为其输入输出目标。该RT不会再分配给其它任何VRF作为它们的RT（如蓝色VRF），这样就保证红色公司的VPN中只包含自己VPN中的路由。具体的步骤如下：

SITE-1：我发的路由是红色的，我也只接收红色的路由。

SITE-2：我发的路由是红色的，我也只接收红色的路由。



SITE-5：我发的路由是红色的，我也只接收红色的路由。

SITE-3：我发的路由是蓝色的，我也只接收蓝色的路由。

SITE-4：我发的路由是蓝色的，我也只接收蓝色的路由。

SITE-6：我发的路由是蓝色的，我也只接收蓝色的路由。

这样， SITE-1、2、5中就只有自己和对方的路由，两者实现了互访。同理SITE-3、4、6之间也一样。这时我们就可以把SITE-1、2、5称为VPN-A，而把SITE3、4、6称为VPN-B。

下面我们通过专用PE方式与VRF方式的一个对比来对这部分内容做一个小结：

路由在网络中传递时的冲突问题的解决思路

在成功的解决了本地路由冲突的问题之后，下一步我们就需要解决路由在网络中传递时的冲突问题。标准的BGP只能处理IPv4路由，所以如果不不同的VPN使用相同的IPv4地址前缀，在接收端就无法分辨不同VPN的路由。使用RT属性是可以部分解决这个问题的，但同时也存在一定的局限性。我们来分析一下通过RT如何解决这个问题及它的局限性。

- 当PE收到不同VPN发过来的路由后，根据RT属性决定路由进入哪个VRF，这样就可以保证不同VPN的路由不具备可比性，操作可以正常进行。

- 路由撤销的时候BGP报文是不带属性的，

	发出路由	接收路由
专用PE方式	在属于特定VPN的路由器上，使用BGP的community属性，将本VPN的路由打上特殊标记。并将路由发给P路由器。	在P路由器上接收所有的路由，并根据路由中的community属性发给特定的VPN的PE设备。
VRF方式	在一个VRF中，在发布路由时使用RT的export规则。直接发送给其他的PE设备。	在接收端的PE上，接收所有的路由，并根据每个VRF配置的RT的import规则进行检查，如果与路由中的RT属性match，则将该路由加入到相应的VRF中。



RT肯定也就不起作用了，会导致所有VPN中的相同路由都被撤销掉。

所以RT虽然具备了这个功能但并不是所有的时候都好用，必须有一种标记可以和IPv4地址绑定到一起来从根本上解决这个问题—这个标记我们称之为RD。

RD是附加在IPv4地址前面的一种标记，它的格式如下所示：

RD。VPN-IPv4地址仅用于服务供应商网络内部。在PE发布路由时添加，在PE接收路由后放在本地路由表中，用来与后来接收到的路由进行比较。CE不知道使用的是VPN-IPv4地址。在其穿越供应商骨干时，在VPN数据流量的包头中没有携带VPN-IPv4地址。RD只在骨干网路由协议交换路由时使用。并且PE从CE接收的标准的路由是IPv4路由，如果需要发布给其它的PE路由器，此时需要为

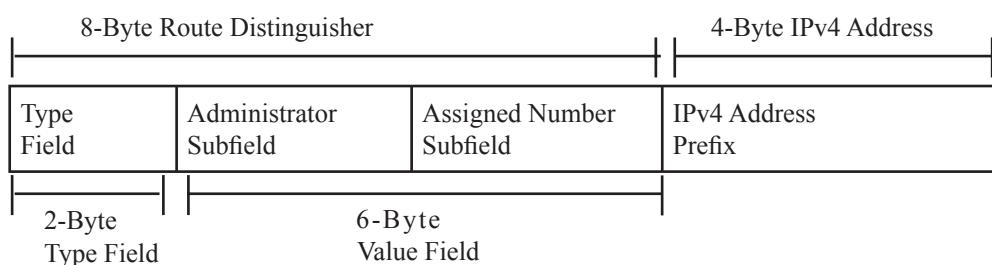


图6 RD格式示意图

其中类型字段定义了两个值：0和1

对于类型0，管理器子区域包括2字节，分配数值字段包括4字节。管理器子区域使用自治系统号码（ASN），分配数值子区域为服务提供商管理的数值空间。类型0不能使用私有自治系统号码，可能会造成冲突。如果要使用私有的自治系统，可以使用类型1。

对于类型1，管理器子区域包括4字节，分配数值字段包括2字节。管理器子区域使用IPv4地址，分配数值子区域为服务提供商管理的数值空间。

RD的结构和RT相似，但它们是有本质区别的，RT是BGP路由的扩展属性，而RD是附加在IPv4地址前的作为地址的一部分存在，这点需要大家注意。

关于RD的一些应用上的特点如下：在IPv4地址加上RD之后，就变成VPN-IPv4地址族了。理论上可以为每个VRF配置一个RD，但要保证这个RD全球唯一。通常建议为每个VPN都配置相同的

这条路由附加一个RD。正因为RD具有这些特点，所以如果两个VRF中存在相同的地址，但是RD不同，则两个VRF一定不能互访，间接互访也不行。这是因为在数据转发时数据报文中并不携带RD，这样数据到达目的地时PE就会在不同的VRF中查找到去往相同目的地的路由条目，从而造成错误的转发。虽然RD是在PE设备路由交换的过程中携带，但是RD并不会影响不同VRF之间的路由选择以及VPN的形成，这些事情是由RT搞定的。

数据报文转发问题的解决思路

前面两个问题：在PE本地的路由冲突和路由网络传播过程中的冲突问题都已解决。但是在数据转发时如果接收端PE的两个本地VRF中同时存在10.0.0.0/24的路由，当它接收到一个目的地址为10.0.0.1的报文时，它如何知道该把这个报文发给与哪个VRF相连的CE？肯定还需要在被转发的报文中增加一些信息。当然这个信息可以由RD担当，

只需改造一下MPLS VPN的处理流程，使数据转发时也携带RD即可解决。但是RD一共有64个bit，太大了，这会导致转发效率的降低。为保证效率，只需要一个短小、定长的标记即可。由于公网的隧道已经由MPLS来提供，而且MPLS支持多层标签的嵌套，这个标记可以定义成MPLS标签的格式。那这个标签由谁来分配呢？路由是私网VPN的，LDP对其一无所知，这个分配VPN私网路由标签的任务也

只能由扩展的BGP来完成了。

和LDP协议类似，标签的分配是在数据转发发生之前完成的。不同的是MP-IBGP分配标签是和路由交换同时进行的。我们知道BGP交换路由是通过NLRI（Network Layer Reachability Information）来完成的，通过对BGP协议的改造，改造后的MP-IBGP进行NLRI信息交换时会附加RD、标签等各种信息。格式如下：

MP_REACH_NLRI:	
address-family :	VPN-IPV4地址族
next-hop:	就是PE路由器自己，通常是loopback地址。
NLRI:	
label:	24个bit，与MPLS标签一样，但没有TTL。
prefix:	RD:64bit + ip前缀

图7 MP-IBGP NLRI报文格式

在这之后是RT信息，如下图所示：

Extended_Communities(TR1)
Extended_Communities(TR2)
Extended_Communities(TR3)

图8 MP-IBGP NLRI报文RT列表

这样，整个MPLS VPN的路由交换及数据转发问题就都解决了。下面我们来介绍一下MPLS L3 VPN的路由交换及数据转发的流程。



MPLS L3 VPN路由 交换及数据转发流程

MPLS L3 VPN技术综述

MPLS L3 VPN路由交换过程

前面我们已经介绍过，MPLS L3 VPN的路由交换时，PE路由器运行单个路由协议（MP-IBGP），来交换所有的VPN路由。为支持VPN客户空间重叠的情况，给VPN地址空间加上RD，使其是唯一的。并使用RT属性来标示路由所属的VRF。我们可以对其进行如下的总结。

MPLS/VPN的路由交换过程主要分为四部分：

- 1、CE与PE之间的路由交换；
- 2、VRF路由注入到MP-IBGP的过程；
- 3、公网标签分配过程；
- 4、MP-IBGP路由注入到VRF的过程。

下面我们通过实例来分析一下MPLS/VPN路由在PE间交换的全过程。

CE与PE之间的路由交换

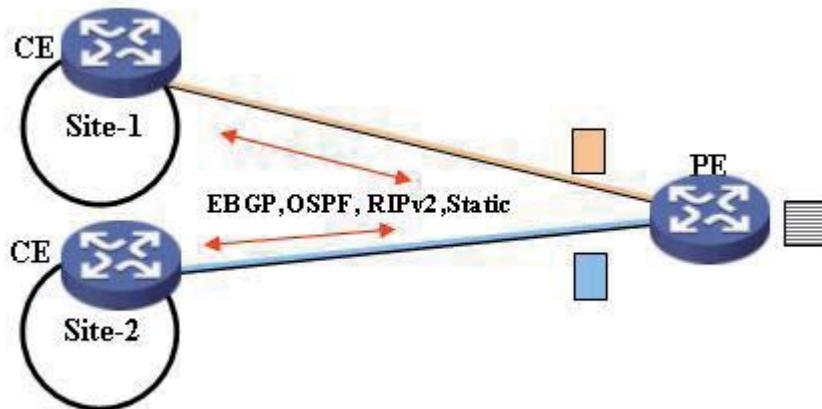


图9 CE与PE之间路由交换示意图

如图所示，交换过程如下：

在PE上为不同的VPN站点配置VRF。PE上维护多个独立的路由表，包括公网和私网(VRF)路由表，其中：

1. 公网路由表：包含全部PE和P路由器之间的路由，由骨干网IGP产生。
2. 私网路由表：包含本VPN用户可达信息的路由和转发表。

PE和CE之间通过标准的EBGP、OSPF、RIP或者静态路由交换路由信息。在这个过程中，除PE设备需要将CE设备传来的路由分别存储在不同的VRF外（这和路由接收的接口有关，和其它MPLS VPN特性无关）其它操作和普通的路由交换没有区别。

静态路由、RIP都是标准的协议，所有的CE端都可以使用相同的路由协议，但是需要在PE端的每个VRF运行不同的实例。相互之间没有干扰。

与PE的MP-IBGP之间只是简单的互相引入操作。EBGP的情况与RIP类似，也是普通的EBGP而非MP-EBPG，只交换经过PE过滤后的本VPN路由。但选择OSPF作为PE与CE之间的路由协议，情况相对复杂。需要对OSPF做很多修改，以将本site的LSA放在BGP的扩展community属性中携带，与远端VPN中的OSPF之间交换LSA。每个site中的

OSPF都可以存在area 0，而骨干网则可以看作是super area 0。此时的OSPF由两级拓扑（骨干区域+非骨干区域）变为三级拓扑（超级骨干区域+骨干区域+非骨干区域）。关于OSPF在MPLS VPN网络中的更详细的介绍请参阅MPLS VPN的其它相关文档，在此就不详细介绍了。这样CE到PE端的路由交换过程就完成了。

VRF路由注入到MP-IBGP的过程

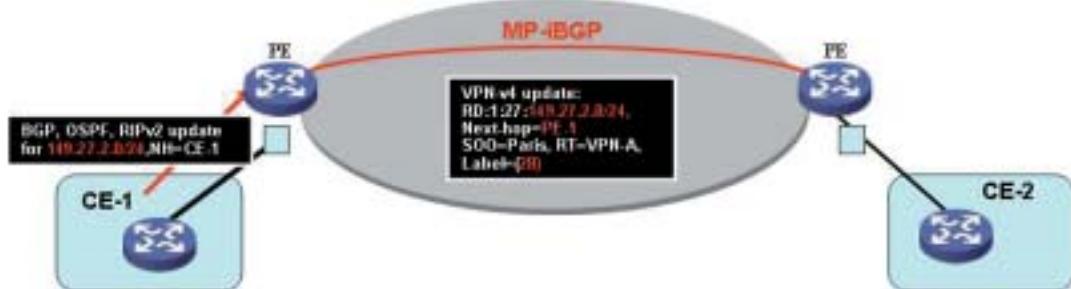


图10 VRF路由注入到MP-IBGP及PE间路由交换示意图

如图所示，VRF路由注入到MP-IBGP并通过MP-IBGP在PE设备间交换的过程如下：

在从CE端接收到路由信息后，PE路由器需要对该路由加上RD（RD为手工配置），使其变为一条VPN-IPv4路由。然后在路由通告中更改下一跳属性为自己（通常自己的loopback地址），为

这条路由加上私网标签（由MP-IBGP协议随机自动生成，无需配置）、加上RT属性（RT需手工配置）。这一系列工作完成后，由PE发给其它所有的PE邻居。其它的PE邻居也进行同样的操作用于交换不同CE端的路由。



公网标签分配过程

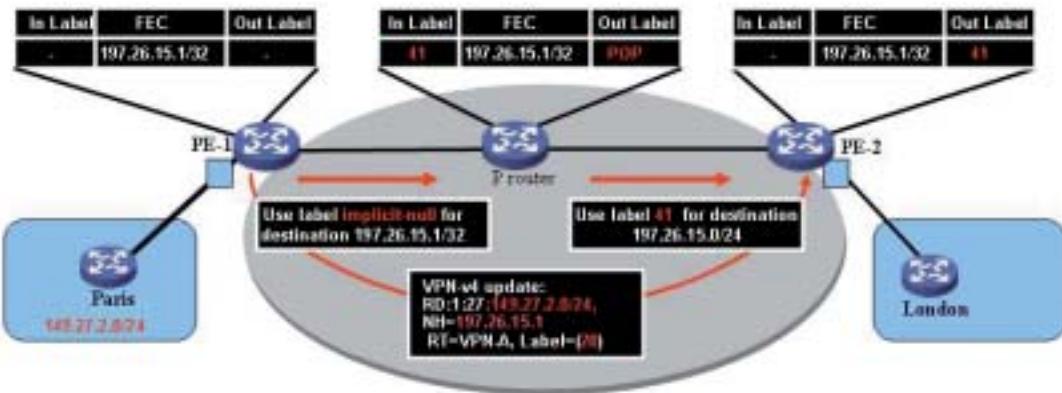


图11 公网标签分配过程示意图

PE间私网路由交换需要跨越MPLS骨干网络，在这个过程中需要进行标准的MPLS转发，所以要正确的将路由传递到对端PE，则需要知道到达对端PE的公网标签。如图所示，公网标签分配的过程如下：

首先PE和P路由器通过骨干网IGP学习到BGP邻居下一跳的地址。通过运行LDP协议，分配标签，建立LSP通道。标签栈用于报文转发，外层标签用来指示如何到达BGP下一跳，内层标签表示报文的出接口或者属于哪个VRF（属于哪个VPN）。MPLS节点转发是基于外层标签，而不管内层标签是多少。此时通过MPLS的外层标签空间，PE设备间就可以进行正常的路由交换了。

MP-IBGP路由注入到VRF的过程



图12 MP-IBGP路由注入到VRF的过程示意图

如图所示，接收端PE在接收到发送端PE发送的路由后，将VPN-v4路由变为IPv4路由，并且根据本地VRF的import RT属性将路由条目加入到相应的VRF中，私网标签保留，记录到转发表中，留做转发时使用。再由本VRF的路由协议引入并传递给相应的CE。发给CE时下一跳为接收端PE自己的接口地址。这样就完成了从MP-IBGP路由注入到VRF的过程。

经过以上四个步骤，整个MPLS VPN网络的路由交换就完成了。此时VPN构建完成，可以进行正常的业务数据转发了。

MPLS L3 VPN数据转发流程

MPLS/VPN的数据转发过程也需要分为两部分来进行处理：

- 1、从CE到Ingress PE。
 - 2、Ingress PE—>Egress PE—>CE。
- 我们还是通过具体的实例来进行分析。

数据转发—从CE到Ingress PE

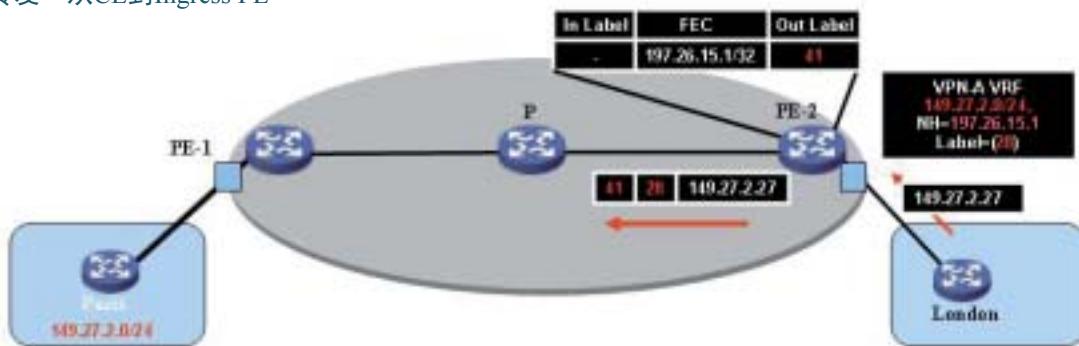


图13 数据转发——从CE到Ingress PE过程示意图

如图所示，从CE到Ingress PE的数据报文转发过程如下：

CE将报文发给与其相连的VRF接口，PE在本VRF的路由表中进行查找，得到了该路由的公网下一跳地址（即：对端PE的loopback地址）和私网标签。

在把该报文封装一层私网标签后，在公网的标签转发表中查找下一跳地址，再封装一层公网标签，交与MPLS转发。

数据转发—Ingress PE—>Egress PE—>C

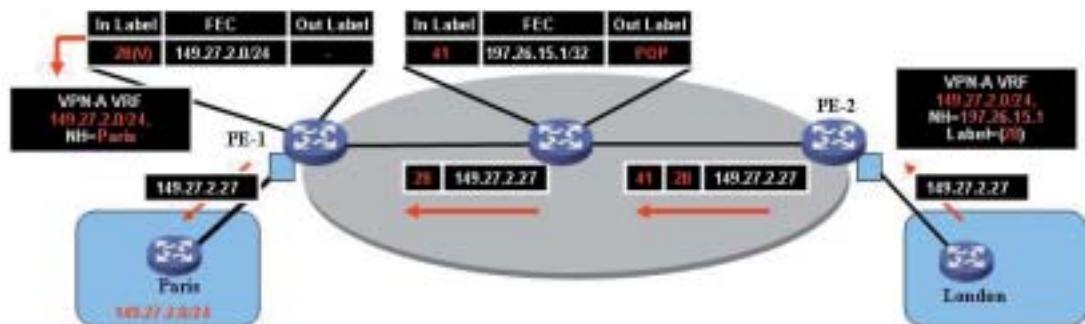


图14 数据转发——Ingress PE—>Egress PE—>CE



如图所示，从Ingress PE—>Egress PE—>CE的数据报文转发过程如下：

如前一部分所述，交由MPLS转发的报文在公网上沿着LSP转发，并根据途径的每一台P设备的标签转发表进行标签交换。在倒数第二跳P设备处，将外层的公网标签弹出，并由该P设备交给目的PE设备，PE设备根据内层的私网标签判断该报文属于哪个VRF。然后弹出内层的私网标签，在目的VRF中查找路由表，根据下一跳发给相应的CE。

这样VPN站点间的一次数据交换就完成了。

MPLS L3 VPN特性介绍

前面我们介绍了MPLS L3 VPN的整体结构及基本特征，下面对MPLS L3 VPN的一些特性做简单的介绍。

SOO特性

SOO的概念：扩展团体路由源，SOO用于当站点与MPLS/VPN主干中的多个PE路由器相连并且使用了AS覆盖特性时防止路由环路。根据路由的SOO，PE路由器确定它从哪个站点获得，这样其它的PE路由器便不会将该路由重新通告给该站点。各台相关PE都需要配置，是扩展团体属性的一部分。

下面是这个特性的一个实例：

如图所示，位于HANGZHOU的PE收到来自BEIJING PE的关于192.1.1.0/24的路由更新。更新中包含一个值为100 : 28的SOO，它是位于HANGZHOU PE路由器上配置的VRF SOO之一，因此该路由不会再被通告到客户站点。

AS覆盖特性

AS 覆盖 特性：当客户站点和PE间运行EBGP时，客户在其不同站点使用相同的AS号，通过在PE上重写VPN站点路由中的AS_PATH，使得路径中只包含MPLS/VPN主干的自治系统号，以避免对端站点不接受该路由。该特性与SOO特性一起使用以避免出现路由环路。在配置了AS覆盖特性后，PE路由器在将路由通告给CE时对路由进行检查，相关操作如下：

1. 如果AS_PATH中的最后一个AS号与邻居的AS号相同，则PE将其替换为自己的AS号。
2. 如果AS_PATH中有多个AS号与最后的AS号相同，PE将这些AS号都替换为自己的AS号。

每执行一次常规EBGP流程，PE会将自己的AS号加到AS_PATH。这是传统EBGP的处理流程。我司实现是通过允许AS号重复实现的。

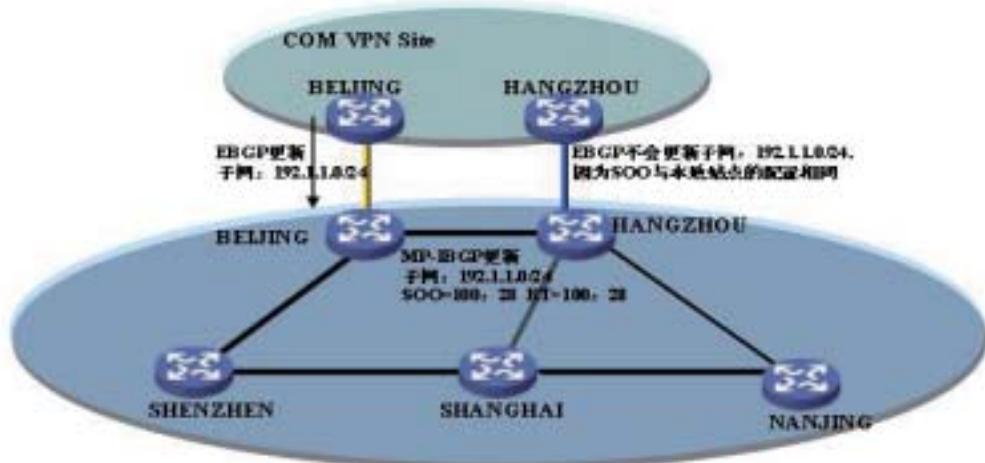


图15 SOO特性示意图

如下面这个例子所示：

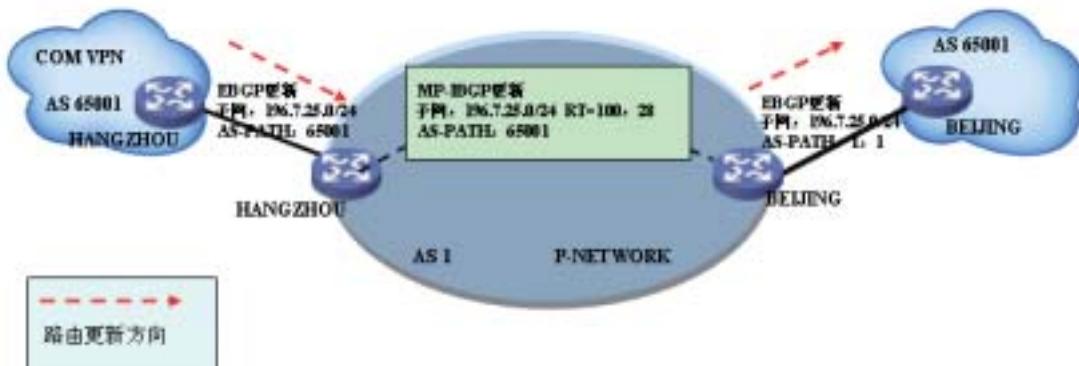


图16 AS覆盖特性示意图

如图所示，从HANGZHOU PE传递到BEIJING PE的关于196.7.25.0/24的更新消息中，AS-PATH与BEIJING CE的AS号相同，在BEIJING PE上将该路由传递给BEIJING CE时会更改AS-PATH属性为1, 1，这样这条路由会被BEIJING CE正常接收。

自动路由过滤技术（ARF）特性

自动路由过滤技术（ARF）特性：目前MPLS/VPN进行路由交换时存在一个问题：所有的VPN路由都会存放在一个全局的BGP路由表中，

不管该PE的VRF是否需要该路由，该路由条目都会存放，这样对PE的内存和链路带宽都是一种浪费。自动路由过滤技术可以满足这种过滤的需求，该功能在所有的PE路由器上都可用。它的特点是将包含的RT与PE配置的任何一个VRF都不匹配的路由条目自动过滤掉，以减少PE必须存储到内存中的信息量。

图17 自动路由过滤技术（ARF）特性示意图

如图所示，从SHANGHAI来的路由条目携带的RT与BEIJING的PE路由器中所配置的不相匹配，导致在该PE上更新被拒绝。

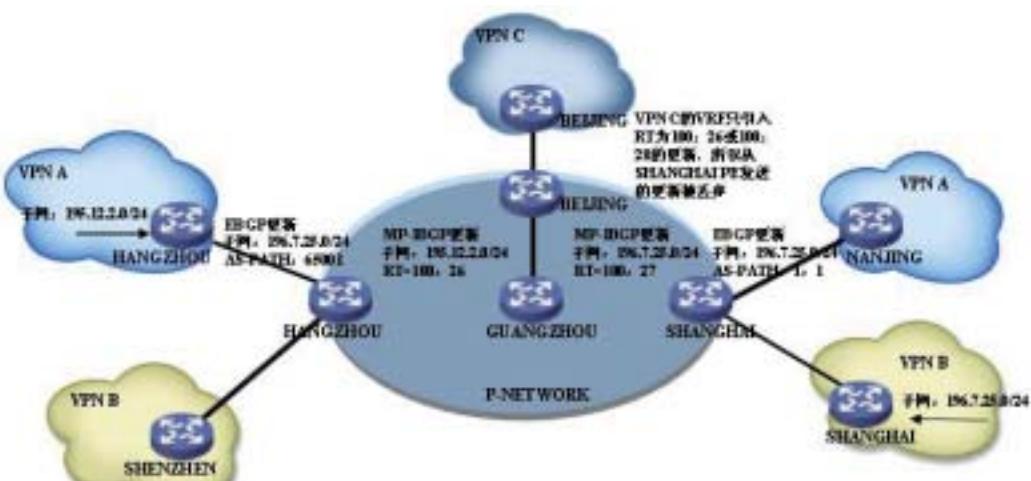


图17 自动路由过滤技术（ARF）特性示意图

路由刷新特性

路由刷新特性：当使用了自动路由过滤技术将PE上不需要的路由过滤掉以后，此时PE的策略发生了变化，如添加了新的VRF或修改了已有的VRF，则PE路由器需要获得以前被丢弃的路由。此时需要更改过滤特性的配置使其接受该路由，但是按照BGP的处理机制，路由始发者只会发送更新的路由信息给邻居而不会重传路由表，所以需要有一种机制来保证路由的正常学习。路由刷新特性用来解决这个问题。使用这个特性后，PE路由器将在其配置被修改后的一段时间后，请求其邻居重传路由选择信息，以获得前面丢弃掉的路由。延时是必要的，因为在PE上可能要做多处修改，而我们希望只发一次刷新。路由刷新特性是默认开启的不需要额外的配置。

下面是这个特性的一个实例：

如图所示：

- 1、BEIJING PE新添加了一个站点，该站点属于VPN B。
- 2、BEIJING的PE给其邻居发送一条路由刷新消息要求重传VPN路由。
- 3、其PE邻居将VPN路由重传给BEIJING的PE，使其学习到该VPN的路由。当然前期必须把过滤策略去掉。
- 4、我们也可以手工的reset BGP来实现路由刷新，但效率较低。

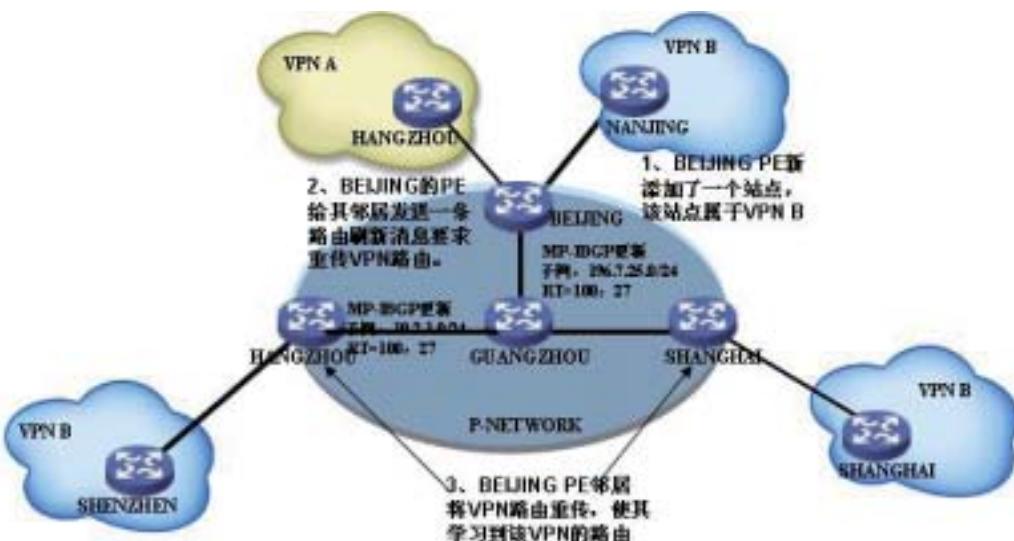


图18 路由刷新特性示意图

MPLS L3 VPN典型组网

配置实例

下面的内容是一个MPLS L3 VPN的典型组网配置实例，希望对大家学习MPLS L3 VPN能有所帮助。实例中配置以我司路由器设备为例。

如图所示：

CE1、CE3构成VPN-A，CE2、CE4构成VPN-B；

不同VPN用户之间不能互相访问。VPN-A使用的VPN-target属性为100:26，VPN-B使用的VPN-target属性为100:27

下面依次介绍各核心组件PE路由器、CE路由器和P路由器上的配置。具体配置请参见附件。

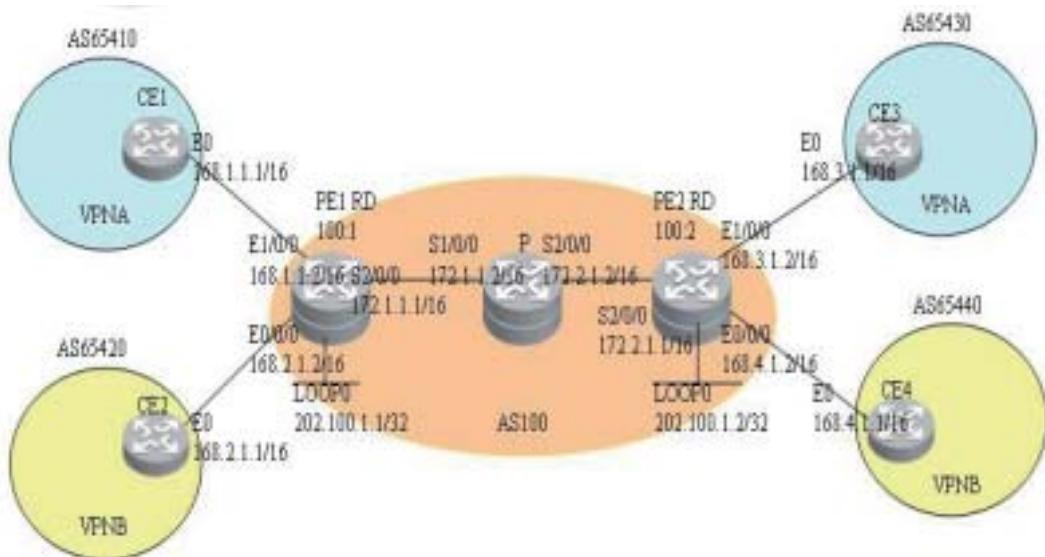


图19 MPLS L3 VPN典型组网图



L3VPN

多实例路由协议

郜忠华



路由协议的多实例

所谓路由协议的多实例，就是指在一个独立的VRF接口表和路由表上，运行一个与这个VRF相配套的路由协议。因此，在同一台路由器上公网以及不同VRF里路由协议的实例之间相互隔离，配置、功能上都是基本独立的。从理论上讲，任何路由协议都可以支持多实例，我们常见的路由协议RIP、OSPF、BGP等都支持多实例。

一般情况下，路由协议的多实例使用在PE上，即MPLS VPN这个场合下。为了实现MPLS VPN的功能，PE上的路由协议多实例需要与MP-IBGP路由协议交互。RIP、BGP等路由协议的多实例和MP-IBGP的交互，其实就是复杂一点的引用关系，即MP-IBGP将VRF中RIP、BGP多实例里面的路由加上RD变成VPN-IPv4路由引入MP-IBGP，同时携带诸如私网标签、RT LIST这些属性。对端

PE再将VPN-IPv4路由的RD去掉，根据RT LIST引入不同VRF的RIP、BGP多实例里即可。配置上也是使用“引入”这样的命令来实现交互的。这个交互过程前面已经讲过，不再赘述。OSPF和MP-IBGP的交互相对复杂，后面专门讲述。

另外，大家会有疑问，路由协议的多实例和VRF这些东西只能用在MPLS VPN体系里面么？当然不是，VRF和路由协议的多实例完全可以作为一种手段，将一个路由器分割为多个路由器。这种应用我们也称之为MultiCe，它和MPLS VPN没有关系，即多实例中的路由不需要引入MP-BGP，也不需要为这些路由分配相应的私网标签、RT LIST等等，只是简单的将路由器划分，建立独立的路由表、接口表、以及对应的路由实例。

OSPF多实例与MPLS VPN

OSPF路由协议的多实例要复杂一些，这是因为OSPF路由协议本身追求尽善尽美的特点，决定了它应用在MPLS VPN体系中能够提供更丰富、安全的使用方式。

我们将OSPF多实例在MPLS VPN里面的应用，简称为OSPF VPN。

简单的OSPF VPN应用

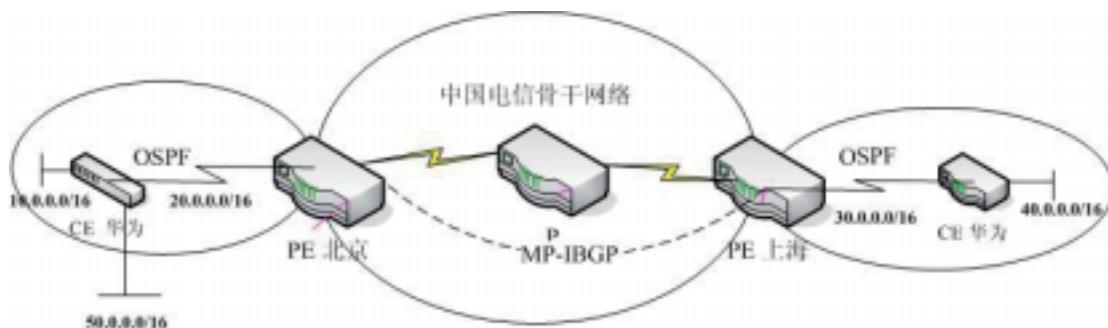


图1 简单的OSPF多实例在MPLS VPN中的应用

从路面上讲，普通路由协议的多实例应用在MPLS VPN体系中，仅仅是引入关系。如果OSPF这样简单应用也是可以的，那么从MP-BGP中插入VRF的时候，只是相当于PE上的OSPF实例引入了MP-BGP路由，因此只能将原站点的OSPF区域内路由还原成ASE等外部路由的让其他站点学习。

图1中，华为公司在北京的CE和PE运行OSPF路由协议，那么在PE上看到的10.0.0.0/16这条路自然是一个OSPF的区域内路由；如果CE上引入了50.0.0.0/16这条直连路由，那么在北京PE的OSPF多实例看来50.0.0.0/16这条路是ASE路由。

但是不管怎样，这两条路由被MP-IBGP携带到上海的PE并引入OSPF多实例的时候，都变成了两条ASE的LSA发布给所有上海站点的其他设备，因此上海的CE以及其他设备学到这两条路由都变成了ASE路由。

这是无法忍受的，因为OSPF能够支持多种多样的路由类型，一个公司的VPN网络，怎能够将ASE路由和OSPF内部路由混淆呢？为了使得路由的优先级及其他属性在MP-IBGP传输时不丢失，就必须随着路由携带更多的MP-BGP属性。

OSPF VPN的解构和扩展团体属性

为了保证OSPF路由属性（路由的类别、路由的COST、所属区域等等）在MP-IBGP传输时不丢失，自然是携带的信息越多越好。但是无论如何都不可能将远端站点的拓扑还原，因为那样就意味着要同步LSDB，也就是说即使两个站点的OSPF配置成了一个区域，也无法沟通拓扑，只能通过MP-IBGP沟通路由。相互直接沟通路由的路由算法，也就是DV算法，在OSPF体系中是存在的，即区域间路由和外部路由。那么，可以说，如果OSPF多实例不去伪造拓扑，那么通过MP-IBGP传递而来的OSPF路由最多只能还原成区域间的OSPF路由。

普通区域或者骨干区域，根据原来路由类别的不同，还原成区域间路由或者ASE路由，这就是所谓的OSPF的三层结构。在普通的OSPF两层构架之上增加了一个Super Backbone Area，虽然这个Super Backbone Area里只是通过MP-IBGP传输路由。

OSPF VPN中常用的扩展团体属性下面列举并介绍一下，相关格式以及详细定义请见RFC: OSPF as the PE/CE Protocol in BGP/MPLS VPNs。

► OSPF Domain ID：必选属性

OSPF Domain ID是一个在OSPF VPN中新增的概念，其目的就是将各个站点的OSPF实例划分到

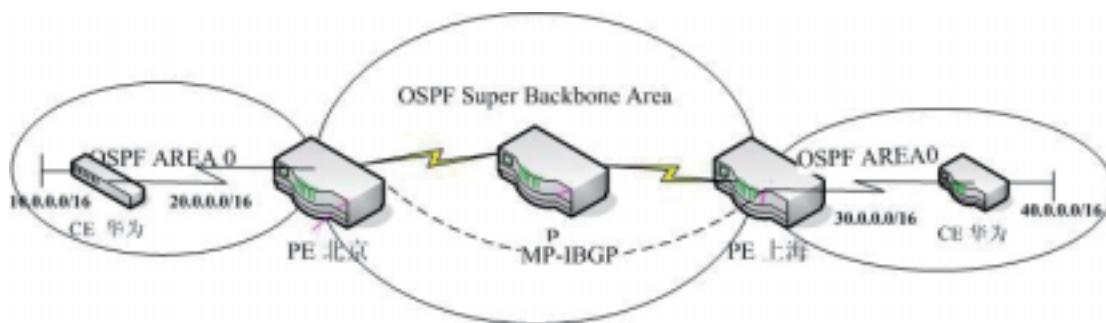


图2 OSPF VPN的三层体系

因此，OSPF多实例在MPLS VPN体系中仿佛变成了以上的三层构架，即各个站点里面可以有骨干区域，但是运营商网络部分像是一个更高层的骨干区域—Super Backbone Area，而PE就是这个Super Backbone Area的边界路由器，可以叫它Super ABR。

Super ABR在把站点内的普通区域或者骨干区域的路由注入到Super Backbone Area的同时，也会把Super Backbone Area区域内的路由注入到站点的

一个OSPF域内。我们知道，通常情况下的OSPF域是一个连通的整体，域内都是OSPF的内部路由（区域内路由或者区域间路由），域外的路由只能是ASE或NSSA这种路由，它们的优先级较低。

而在MPLS VPN体系中，所有站点的OSPF路由都会通过MP-IBGP到达任意一个站点，如果看到原有路由类别就是OSPF内部路由，就一律还原成3类LSA，还原成区域间路由，显然变成了所有站点都在一个OSPF域内了。



为了增加控制，每个站点的 OSPF 都必须配置一个域 ID，域 ID 相同的站点之间，路由可以还原成 OSPF 内部路由（只能是区域间路由），而域 ID 不相同的站点之间，路由只能还原成 ASE 或 NSSA 路由，即来自不同的域的路由按照 OSPF 外部路由对待。

按照前文中 RFC 的要求，每个站点配置的域 ID 可以是个列表，路由注入 MP-BGP 的时候只携带其中的主域 ID，在目的端通过一套匹配的过程，来决定他们是不是在一个域内。

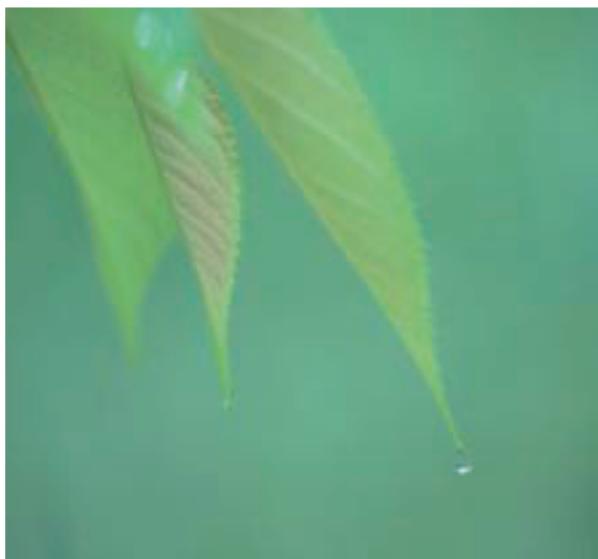
► Area Number：必选属性

Area number 即原路由所在的区域号。对于一般的路由类型来说，这个 Area number 是意义不大的，因为不论两个站点的 OSPF 是不是一个区域，路由最多也只能还原成 3 类 LSA。

但是对于 Sham Link Endpoint Address 类型的路由，Area number 是有用的，因为 Sham Link 是跨越运营商公网的一条区域内链路，Sham Link 的两端要在同一个区域内才能建立 Sham Link 连接。有关 Sham Link 的相关技术，后面会专门讲述。

► OSPF 路由类型：必选属性

就是源站点 OSPF 路由的类别，目的站点的 PE 会根据路由的这个属性，来决定还原成什么样的



LSA，其主要取值有：

1、2：区域内路由，具体是 1 或 2 取决于路由来自 Router LSA 还是 Network LSA

3：区域间路由

5：ASE 外部路由

7：NSSA 外部路由

129：Sham Link 端点地址路由（Sham Link Endpoint Address），Sham Link 建立所使用的。

► OSPF Router ID：可选属性

Router ID 作为一个可选属性，被 MP-BGP 携带到对端站点并无太多实际作用，因为不管对端 PE 将 MP-BGP 还原成 3 类或者 5、7 类 LSA，其 Router ID 都是对端 PE 的 Router ID，相应的对端 PE 也会变成 ABR 或者 ASBR。

CE 双归属与 BackDoor 链路

CE 的双归属是指一个 CE 同时连接到两个 PE 上，做上行备份。而 BackDoor 链路，指的是一个 VPN 的不同站点间，除了通过公网连接以外，还有一条后门链路备份连接。这两种连接方式有其共同点，都是一个用户的网络连接到了多个 CE（PE？）上。如图 3 所示，蓝色部分华为的 CE 设备即连接到了两个 CE（PE？）上，互为备份，称之为 CE 的双归属；而红色的后门链路，却将联想在北京和上海的站点连为一体，与公网的联接互为备份，称之为 BackDoor 链路。这种组网增强了网络的稳定性，但同是也为 OSPF VPN 的应用带来了麻烦。OSPF 本身是没有环路的，区域内路由计算通过 SPF 运算天然的解决了此问题，区域间路由计算通过区域的二层结构解决，外部路由通过携带引入者的 Router ID 解决，等等。但是由于 MPLS VPN 体系中，路由信息经过骨干网以后有丢失和转变的情况，导致路由环路的产生。

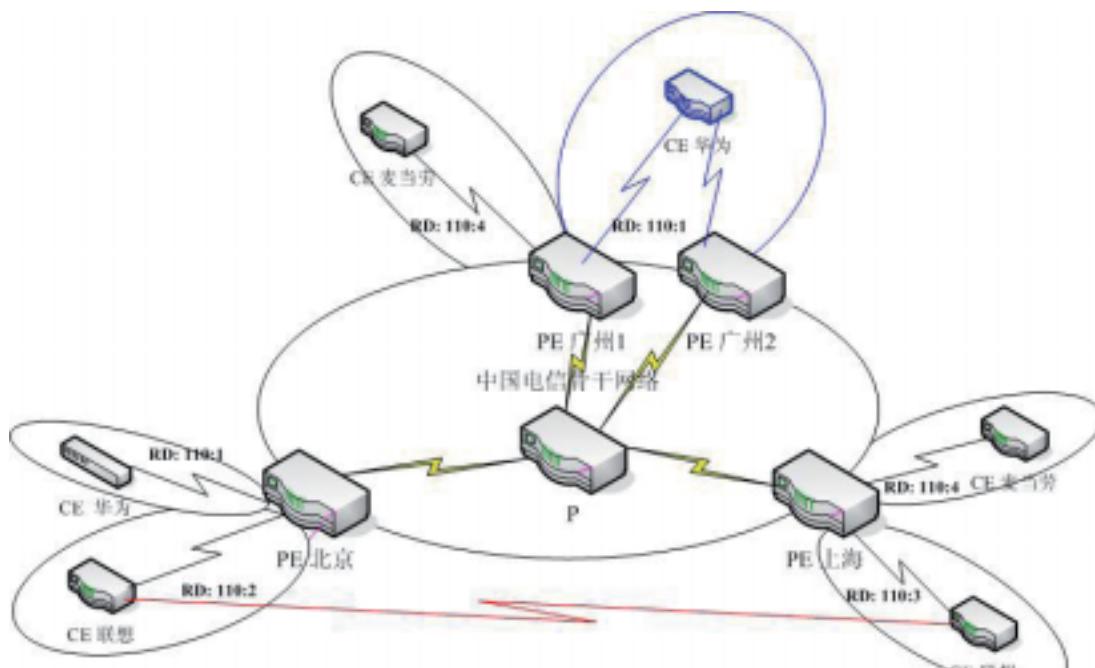


图3 CE双归属与BackDoor链路

● 外部路由的VPN Tag

外部路由避免路由环路的主要方式，就是在ASE LSA泛洪的整个过程中不更改其发布者的Router ID。但是当PE将私网路由还原成ASE LSA到OSPF多实例的时候，这些LSA的Adv Router ID就变成了PE上这个OSPF实例的Router ID，最初这条路由的引入者的信息丢掉了，这条路由变成是PE上这个OSPF实例引入的路由了。这也不太会引入什么问题，但是如果网络里面有CE双归属与BackDoor链路这种组网的话，就有了问题。

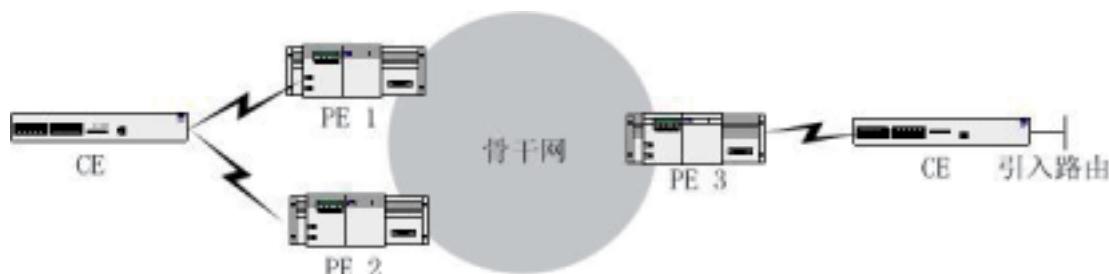


图4 ASE路由的VPN Tag



如上图所示，右边的CE设备引入了外部路由，并被PE3插入到MP-IBGP中传播给所有的PE，那么PE1和PE2都会产生相应的ASE LSA发布给CE，同样都会从CE那里学习到对方产生的ASE路由。这样就带来了混乱，两个PE上就要做一下路由的优选，即从站点内部学来的OSPF ASE路由和插入VPN的BGP路由之间作一个优选。假设PE 1上，优选了PE2引入的OSPF ASE路由，那么它就会收回自己产生的ASE LSA，反而将这条路由注入到MP-IBGP中。PE3收到这条MP-IBGP路由以后可能优选这条路由或者优选本地的ASE路由，不管怎样，都会造成路由的震荡或者带来路由环路的隐患。

因此，为了避免这种情况，PE上在将MP-IBGP路由还原成ASE LSA的时候，将产生一个VPN Tag携带在ASE LSA的Tag字段里。如果PE收到的ASE LSA的VPN Tag和本地OSPF实例的VPN Tag一致，此LSA就不参与路由计算。这个VPN Tag一般来源于MP-IBGP的AS号，因为所有的PE都在一个BGP域中，因此AS号码是一样的，生成的VPN Tag也是一样的，这样就避免了上述情况的发生。VPN Tag是可以配置的。

● 区域间路由的Down Bit

同样，如果有CE双归属或者BackDoor链路这种组网的时候，将MP-IBGP路由还原成3类LSA，不做相应处理也是一样危险的。因此在PE产生3类

LSA的时候都在LSA中设置Down Bit，PE如果收到的3类LSA携带了Down Bit就不参与路由计算。

OSPF VPN中的区域零

在站点内部署OSPF是要非常关注骨干区域部署的方法，否则会造成PE还原的远端站点的私网路由不能被所有设备学到。这是因为OSPF的如下设计决定的：

- 1.ABR设备只计算骨干区域内的3类LSA
- 2.ABR只会将骨干区域内的3类LSA转发到非骨干区

因此如图5所示，当PE将MP-IBGP带来的私网路由还原成3类LSA的时候，CE设备就不能够学习到这些路由，因为CE设备作为ABR，它只计算区域零里面的3类LSA，而PE还原的3类LSA都在Area 1里面。同时，最左边的C设备也学不到这些路由，因为CE设备也不会将PE产生的3类LSA从非骨干区域转换到骨干区域里。

要记住以下口诀：尽量不要在站点内部属骨干区域；如果站点内部署骨干区域，那么PE和CE之间的链路只能在区域零里；在CE双归属或者BackDoor链路的情况下，最好所有的链路都在区域零，否则备份功能可能有问题。

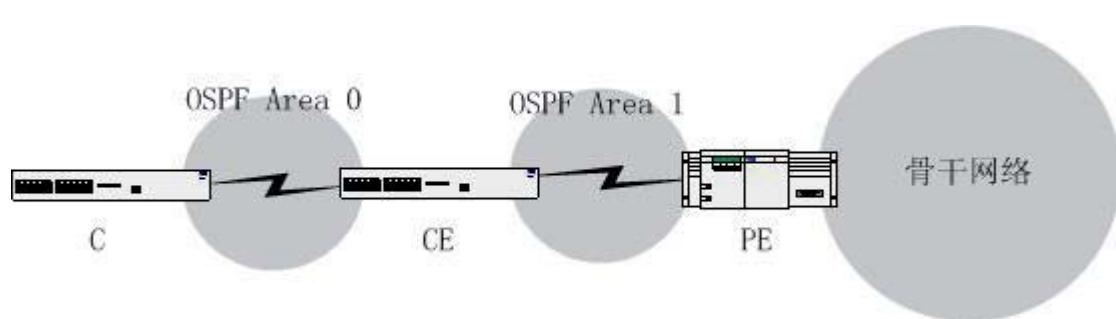


图5 OSPF VPN的区域零部署

Sham-Link概述

上面提到了，通过MP-IBGP携带私网路由的方式，只是传递路由，而且会丢失很多路由的信息，到达对端PE后的还原工作也只是尽力而为式的引入，并不能真正的使得OSPF的拓扑信息得到沟通。那么能不能做到各个站点的OSPF能够真正连通呢，只要各站点间的OSPF之间有一条跨过公网的链路即可，这就是Sham-Link。

Sham-link就应该像一条正常的OSPF链路一样，有自己的OSPF接口，能够互相发送OSPF协议报文，建立邻居，传送LSDB等等，同时作为Router Lsa的一部分参与到路由计算中。Sham-link存在于两个PE之间，而且最好各个PE之间的Sham-link是全连接的，这样携带私网路由的工作就可以由Sham-link完成，但是MP-IBGP并非不再需要了，它依然要用来携带Sham-Link端点路由。

每个站点都需要通过配置指定Sham-link在哪个区域，所有的PE上Sham-link端点都应配置在一个区域内。同时也要配置自己的Sham-Link端点路由，此路由会被MP-IBGP携带到每个PE上，这个路由的扩展团体属性中说明了它是Sham-Link端点类型的路由，因此每个PE都很清楚它的Sham-link对端有哪些，发往这些Sham-Link端点路由的OSPF报文，就自然地通过PE间的MPLS隧道到达对端PE，到达了对端Sham-Link的接口上，完成了报文的交互、LSDB的交互、邻居的建立。这也称作Sham-link的自配置，也就是说只需要在所有PE上指定自己的Sham-Link端点路由，那么Sham-link的全连接就应该能够自动的建立起来，完成OSPF的路由交互。因此，如果配置了Sham-link，那么在PE配置中，MP-BGP就不需要配置引入OSPF多实例的路由，反之，OSPF多实例也不需要配置引入MP-BGP的路由。

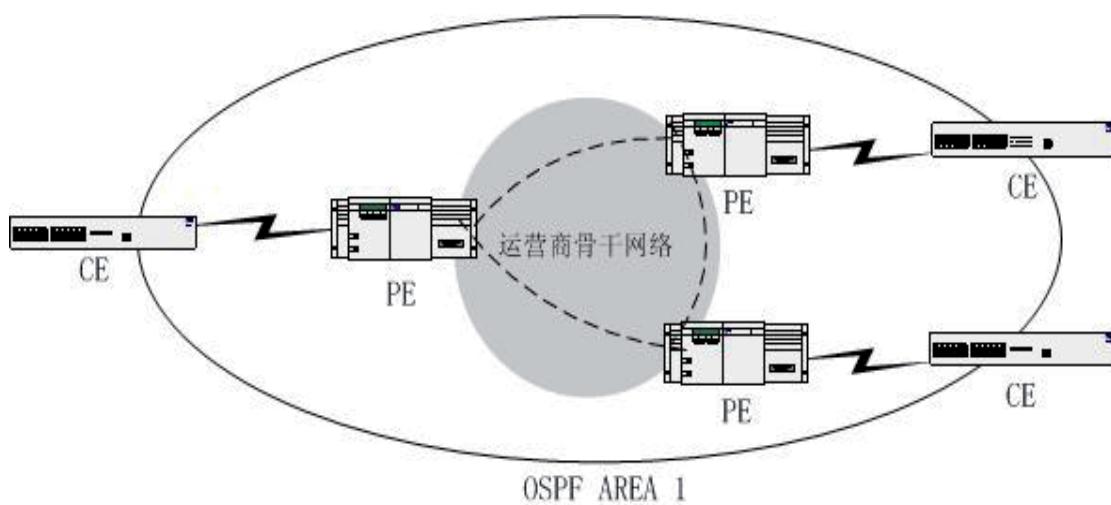


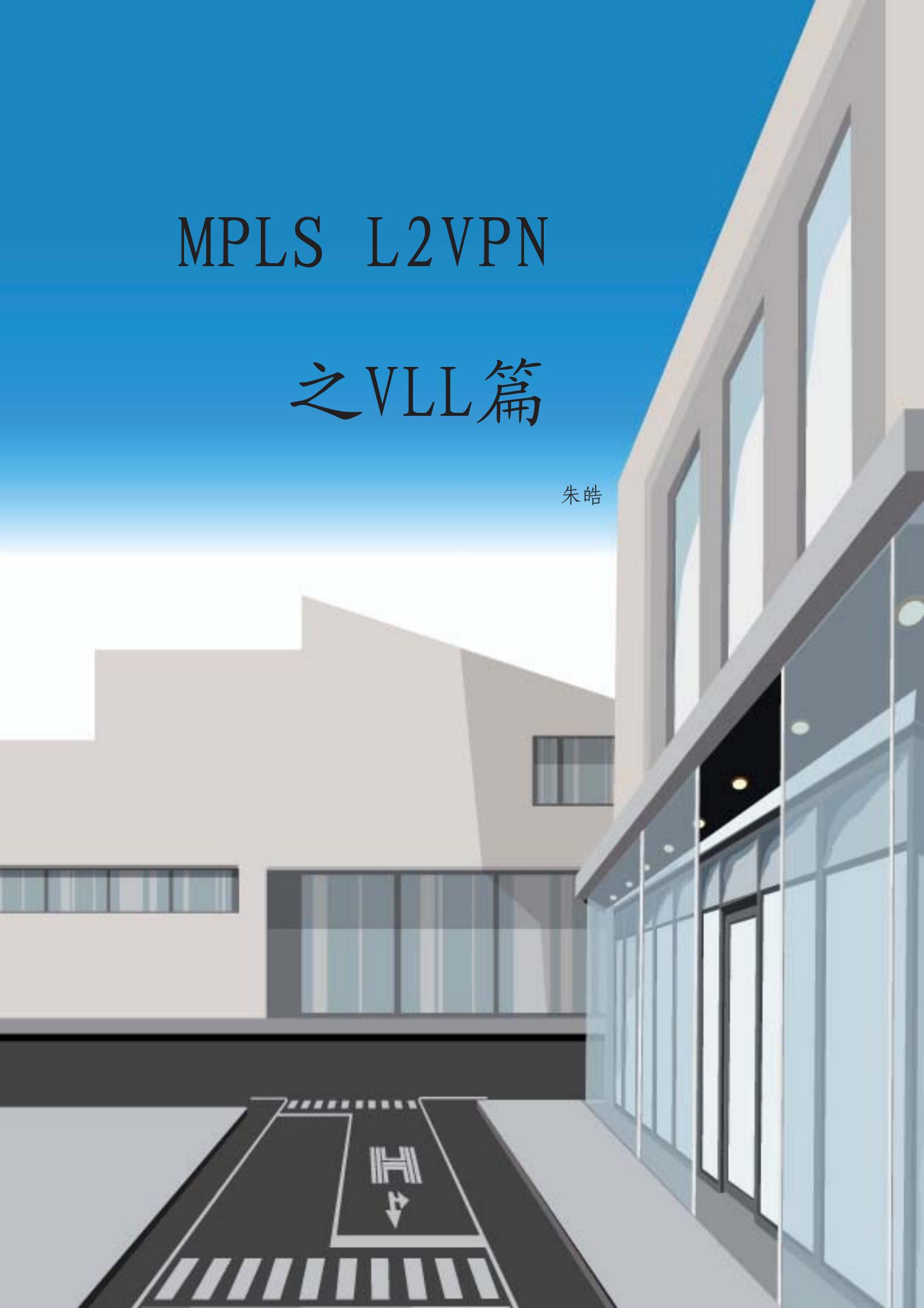
图6 OSPF Sham-Link



MPLS L2VPN

之VLL篇

朱皓



什么是VLL技术

VLL (Virtual Leased Line)，虚拟租用线路，是相对于专有租用线路而言 (dedicated leased line)，相对于专有租用线路中客户对线路的独占，VLL允许多个客户共享线路，通过VC (Virtual Circuit) 区分客户数据，为不同客户提供逻辑上的专用通道，传统的ATM、FR技术均可以提供这种服务，Ethernet通过QinQ也可以提供类似服务。

VLL作为一种点到点的虚拟专线技术，被运营商广泛应用，用于为客户提供L2VPN服务。不过，不同的客户，采用的二层技术不尽相同，网络技术的发展和变更，使得ATM、FR、Ethernet等技术都被各种客户广泛采用，这对为这些客户提供服务的运营商来说，构成了一个挑战：如何在一个统一的基础网络架构上方便的为使用不同二层技术的客户提供L2VPN服务，以最大限度的降低运营开支？

MPLS技术的出现，使得运营商得以很好的应对这一挑战。采用MPLS最新技术，运营商在统一的MPLS/IP基础网络架构上，不但可以为客户提供L3VPN服务，也可以同时为客户提供L2VPN服务，使客户跨过运营商网络实现ATM、FR、HDLC、PPP、以太网等各种二层网络的互连。

MPLS目前提供两种L2VPN服务：

- Virtual Private LAN Service (VPLS)：虚拟私有LAN服务，即在MPLS/IP网络中提供二层LAN服务，允许标准的以太设备通过MPLS/IP网络相连，就象连接在一个二层交换机上。

- Virtual Private Wire Service (VPWS)：虚拟私有线路服务，即在MPLS/IP网络中模拟点到点二层服务，客户的二层设备跨过MPLS/IP核心网络相连，就象通过一根二层线路相连。

在纯IP网络中提供L2VPN服务，可以通过

L2TPv3协议来实现，本文不涉及，有兴趣的读者可以查看IETF的L2TPEXT、L2VPN和PWE3三个工作组的相关工作文档。

事实上，VPWS服务，和本文开篇提到的VLL是同一个东西，都是实现点到点的虚拟线路，本文要介绍的就是如何在统一的MPLS/IP基础网络架构中提供适应各种不同二层技术的VLL服务。

MPLS提供VLL服务的基本原理如下：运营商网络中，为客户的每一个点到点二层连接分配不同的VC标签，一对VC标签模拟一条虚拟线路，其中，VC标签的分配可以是静态配置，也可以是通过LDP、BGP等信令协议动态分配；客户二层数据帧在PE设备上被打上VC标签，然后在运营商核心网中转发到目的PE，在转发到目的PE时，如果使用MPLS隧道，根据实现和应用方式的不同，可能需要再加上一层或多层外层标签，也可以不使用MPLS隧道，比如Martini方式中就可以使用PE到目的PE的GRE隧道；报文到达目的PE后，剥掉外层隧道封装和VC标签，还原成最初的二层帧，发送给客户；报文在运营商网络中的标签转发过程，在对客户来说是透明的，客户分布在不同地域的二层设备，感觉就象是二层链路直连。

CISCO把基于MPLS的VLL服务称为AToM (All Transport over MPLS)。将在本文的最后讲解。

本文主要是介绍在路由器设备上的VLL技术，作为一种虚拟租用线路的实现方法，主要是在接入层和汇聚层使用。首先需要明确的一点是：VLL技术是一种点到点的虚拟专线技术，能够支持几乎所有的链路层协议。为了实现点到多点的MPLS L2VPN虚拟专线，目前的实现是VPLS (Virtual Private LAN Service)。VPLS技术不在本文中讨论。



VLL技术是建立在MPLS技术下的二层隧道技术。传统的二层隧道是通过对应的二层交换设备进行接续，来完成用户节点间二层隧道的建立。因为使用了不同的二层协议，因此不同种L2网络是隔离的。MPLS标签技术解决了统一兼容二层交换网络的问题。我们知道MPLS核心（下文称为SP网络）是根据标签进行交换的，其实可以把MPLS理解成为一个特殊的二层协议，也就是说在原有的各种L2封装外层加装MPLS封装，Cisco将VLL技术叫做AToM（Any Transport over MPLS），这是一个很形象的叫法。VLL技术的好处是显而易见的，第一是同一个SP网络可以提供多种二层协议的接续和交换，另一方面也为不同的L2交换网络互连提供了可能性。

因为VLL是MPLS技术的产物，因此类似MPLS L3VPN的结构，使用内层标签来标识不同的虚拟线路（也就是二层隧道，下文延用传统的VC来表示Virtual Circuit），使用外层标签来做公共隧道。SP网络的设备不需要维护任何二层信息，只根据MPLS标签信息在公网隧道上进行MPLS转发。不同于L3VPN的是FEC（Forwarding Equivalence Class）的概念，在L3VPN中FEC实际上就是路由信息，而VLL中FEC是一些二层信息和VC标志等。

VLL技术

目前我公司的COMWARE平台全面支持VLL技术，实现方式分为4种：CCC、SVC、Martini、Kompella。

支持的链路层协议包括：ATM AAL5、FR、cisco HDLC、PPP、Ethernet（包括tag & untag两种方式）。

Cisco的AToM使用的是Martini方式。

VLL技术的CCC方式

CCC方式最早是在什么时间提出，由谁在什么情况下提出是有一些争论的。一种说法是1998年提出，为解决ATM网络穿越MPLS的问题，后来这个技术得到不断的发展，可以支持大多数的链路层协议。当前的CCC实现虽然并没有标准，但基本的思路是类似的。有兴趣的读者可以研究一下 draft-kompella-ccc-02 (<http://tech.huawei-3com.com/article.php/2787>) 这个草案的状态是Dead，里面关于历史部分的说法不一定准确，大家可以关注一下技术实现的部分。

CCC (Circuit Cross Connect) 方式是一种静态配置VC连接的方式，根据配置把VC端点收到的报文映射到一个静态的LSP隧道上去，这样二层报文在途经的每一跳设备上根据该静态LSP进行MPLS转发，最后将报文转发到VC的另一端。显然CCC的方式并不

需要多层标签，只是一层MPLS标签而已，这一层标签在每个LSR上进行标签交换。CCC方式只需要SP网络支持MPLS转发就可以了。CCC对LSP隧道是独占的，而且在两个方向都需要配置静态的LSP。CCC支持本地连接模式，可以在同一个PE上的两个CE设备间建立CCC连接，也就是通过一个二层转发表项实现CE间的二层互连。CCC方式的L2交换模式与策略路由很相似，在策略路由中也是一跳一跳根据策略路由转发，策略路由是优先于路由表查找的。CCC方式逐跳建立静态的LSP，然后将L2报文在这条LSP上进行传递。

CCC用于小型，拓扑简单的MPLS网络，需要管理员手工配置。因为不需要其它交互信息的信令协议，因此消耗资源比较小，易于理解，但维护比较麻烦。

CCC的结构

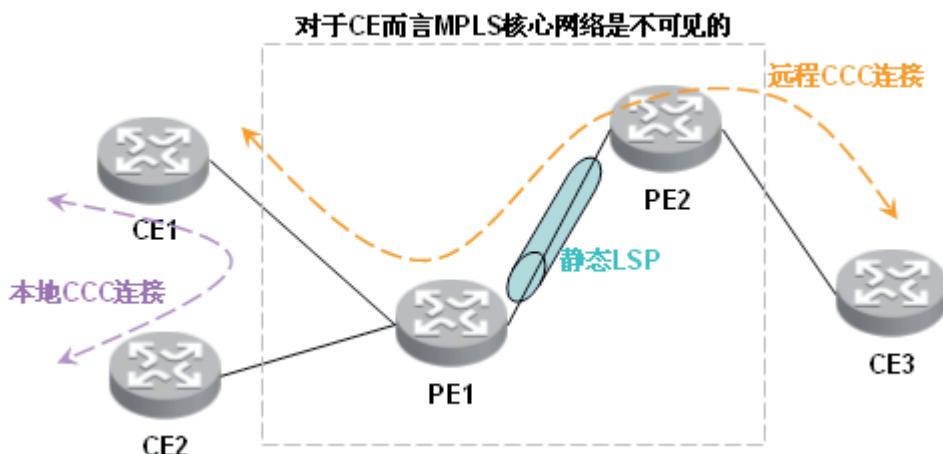


图1 CCC可以支持的拓扑模型

CE1需要为两条CCC连接各准备一个接口（支持子接口），从CE的角度认为CE间是直连的。同样的PE1侧也要有两个接口与CE1连接，第三个接口与CE2连接。



CCC的报文交互过程

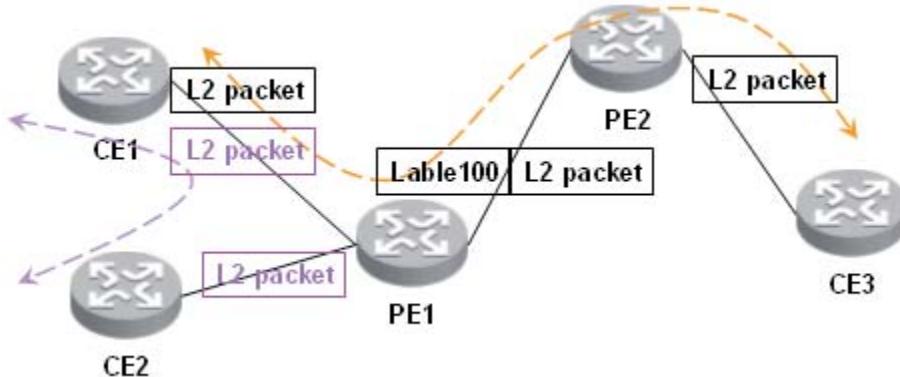


图2 CCC的报文交换过程

CE1发送报文到CE3：（反方向过程相同）

PE1收到CE1相关接口送达的二层报文后，根据CCC的关联配置查找静态LSP，得到下一跳为PE2，标签为100。于是PE1在二层报文外封装MPLS头，label=100并发送到连接PE2的接口。

PE2收到报文后查找LSP表，进行弹出操作，根据CCC的关联配置得到对应的出接口，于是将二层报文直接送到目的CE。

从这个过程中，我们可以看到，PE只关心报文是从那个接口收到的，如果这个接口关联到了某一个CCC连接，就去查找CCC的相关配置，并且进行MPLS封装和MPLS转发。因此PE连接CE侧的接口其实并不做任何二层的处理。同样在出接口的时候，PE只是解封装MPLS，并直接将报文送达到出接口。因此PE侧的CCC连接一旦建立起来后，PE连接CE接口的二层协议状态处于实质上down的状态。这一点在所有形式的VLL实现上都是相同的（当然，VLL在实现上可以在报文进入的PE侧进行二层报文头的解封装，并且在另一侧PE发向CE时进行重新封装，这样就可以实现二层协议的相互转换）。

CE1发送报文到CE2：

PE1收到CE1相关接口送达的二层报文后，根据CCC的关联配置发现是一个本地连接，直接得到出接口，然后不做任何处理，将二层报文通过出接口发送到CE2。

VLL技术的Martini方式

在VLL技术中，有两种重要的实现方法，使用了不同的信令协议来交互VC信息。这两种技术以协议草案的撰写者来命名，其中一种就是Martini方式。最新的协议草案包括：

[draft-martini-l2circuit-trans-mpls-17](http://tech.huawei-3com.com/article.php/1667) http://tech.huawei-3com.com/article.php/1667

[draft-martini-l2circuit-encap-mpls-10](http://tech.huawei-3com.com/article.php/355) http://tech.huawei-3com.com/article.php/355

Martini方式使用标准的两层标签，内层标签是采用扩展的LDP作为信令进行交互。在Martini草案中对标准的LDP进行了扩展，增加了VC FEC的FEC类型用于VC标签的交换。它采用VC TYPE + VC ID来识别一个VC。VC TYPE表明链路层封装的类型，VC ID则用于唯一标志一个VC。同一个VC



TYPE的所有VC中，其VC ID必须在整个PE中唯一。连接两个CE的PE通过LDP交换VC标签，并通过VC ID将对应的CE绑定起来，一个VC就建立起来了，两个CE通过这个VC来传递二层数据。

Martini方式没有提供象CCC方式的本地交换功能。外层标签用于将各个VC的数据在SP网络中进行传递，因为通过内层的VC标签可以对数据进行区分，因此外层隧道是可以被多个VC共享的。

既然外层隧道是用于VC数据穿越SP网络，那么外层隧道当然也可以使用IP隧道封装，比如使用GRE隧道。

Martini方式适合稀疏的二层连接，例如星型连接。部署Martini方式需要SP网络能够自动的建立LSP隧道，所以需要SP网络支持MPLS转发及MPLS LDP，如果SP网络不支持LDP，那么可以使用GRE隧道封装。

Martini方式的结构

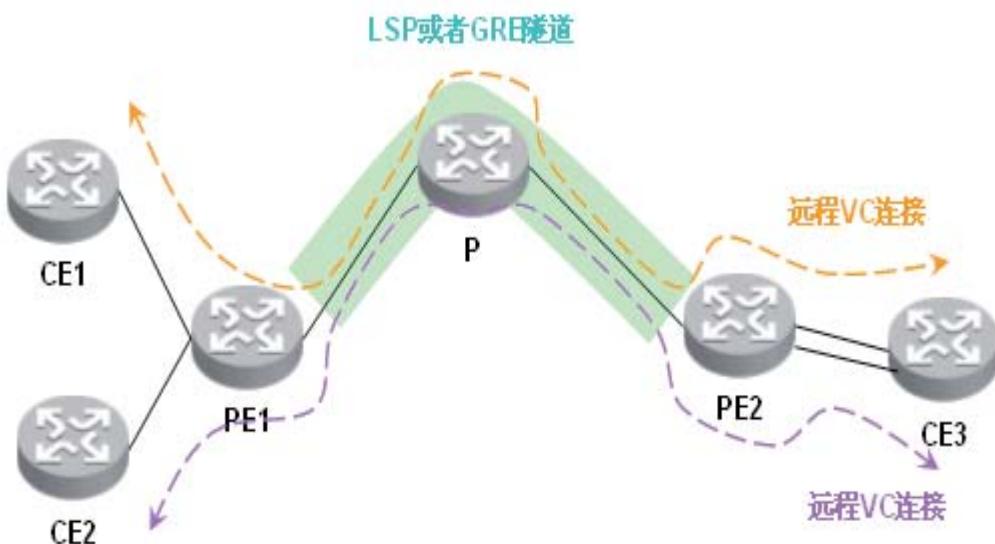


图3 Martini可以支持的拓扑模型



Martini的报文交互过程

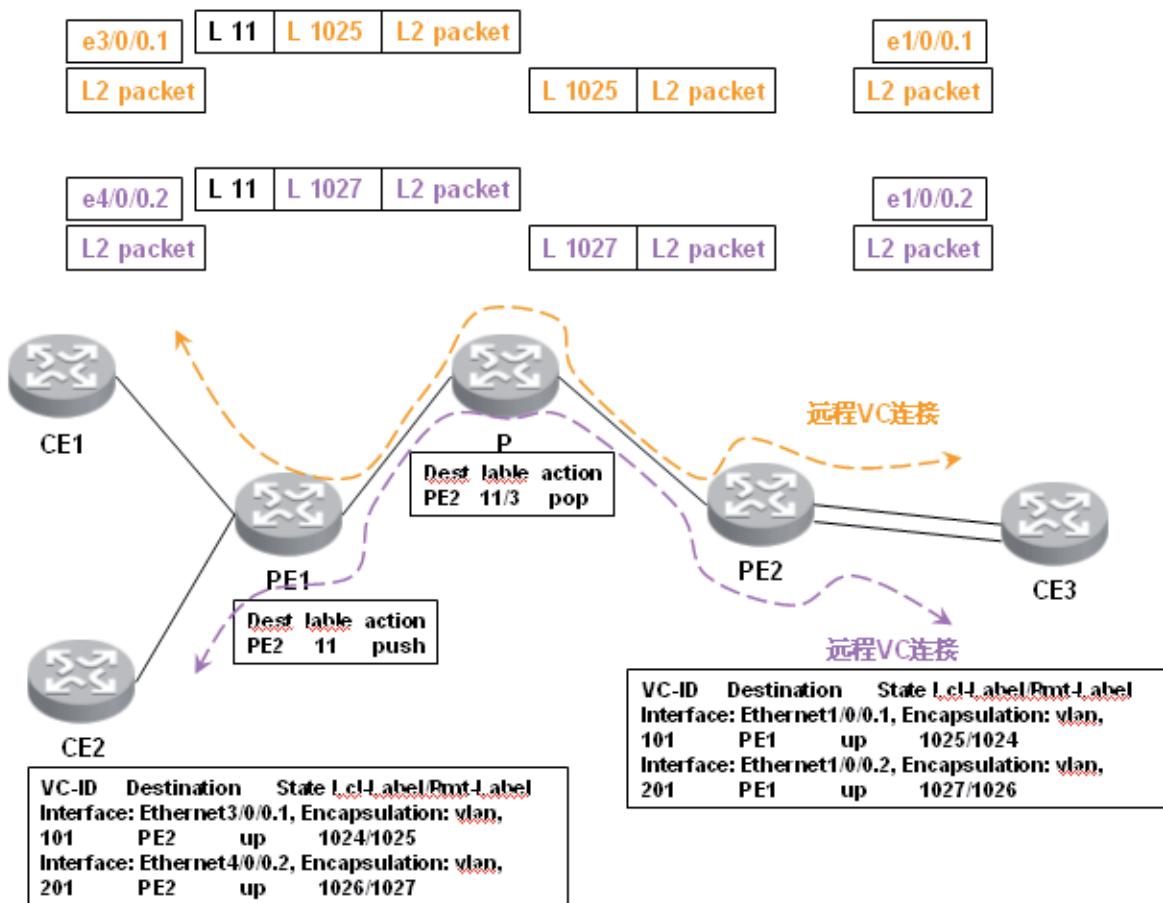


图4 Martini的报文交互过程

CE1发送报文到CE3：（橙色部分，反方向过程相同）

PE1收到CE1相关接口送达的二层报文后，根据VC的关联配置查找VC-ID对应的表项，得到下一跳为PE2，VC标签为1025。于是PE1在二层报文外封装MPLS头，label=1025。根据下一跳为PE2查找LSP隧道，得到外层标签为11，增加第二个MPLS封装label=11，并发送到P。

P设备进行MPLS转发，查找LSP表后知道自己

是次末中继，弹出外层标签，并发送到PE2

PE2收到报文后根据标签1025查找VC-ID对应的表项，得到出接口，将内层标签进行弹出操作，然后将二层报文直接送到目的CE3。

CE2发送报文到CE3的过程是类似的。从上面的交互过程中，我们可以看到，外层的LSP隧道是被共享的。PE2收到报文后会根据内层标签的不同映射到不同的VC上。

VC标签的交互指令

前面已经提到过Martini方式对传统的LDP做了扩展（LDP的标准协议文档为RFC3036），用于交互VC的信息，也就是传递VC标签，用于报文的发送。

首先Martini方式需要在PE间建立扩展的远程LDP会话，然后通过会话来交互VC信息。为了交互VC信息，增加了新的FEC的定义，新的FEC类型为128。下面是VC FEC结构。

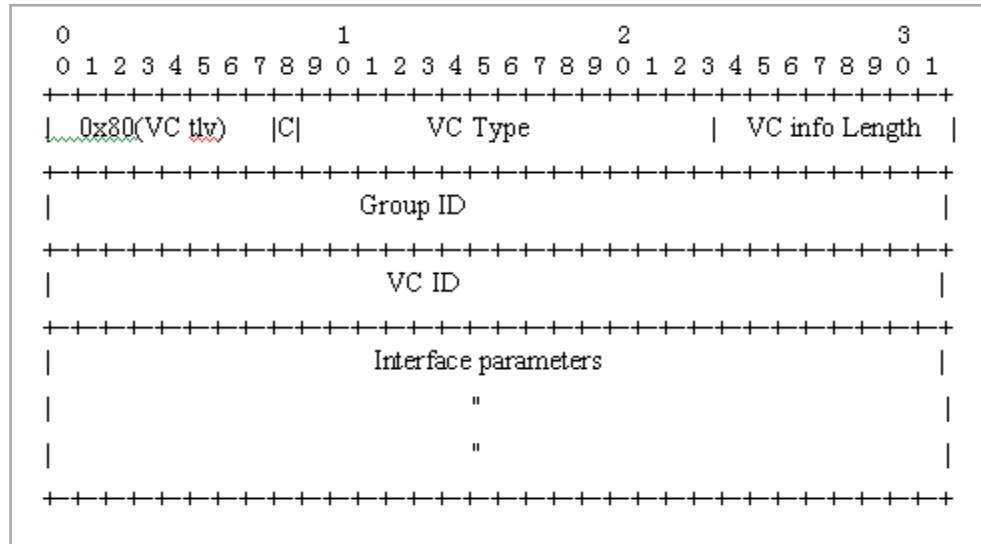


图5 128类VC FEC结构

Martini方式通过VC Type + VC ID来区分不同的VC，在Interface parameters部分描述了与CE连接的接口以及MTU等其它一些信息。

在LDP会话中VC FEC出现在标签映射消息中，下图是一个标签映射消息的结构。

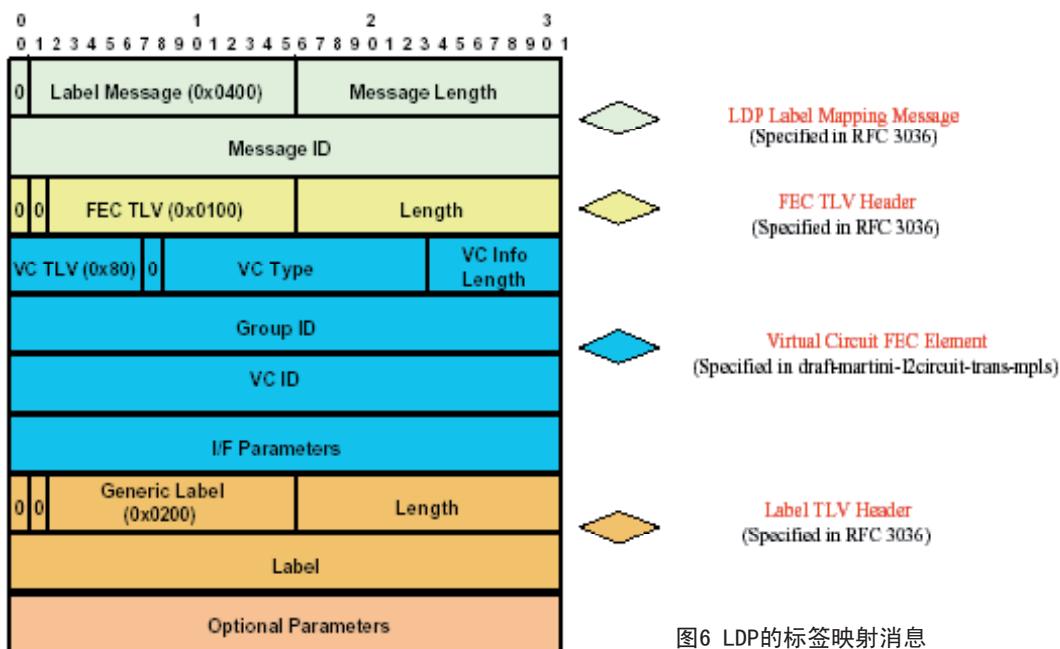


图6 LDP的标签映射消息



VLL技术的SVC方式

SVC方式是Martini方式的一种静态实现。在Martini中LDP是用于进行VC标签的交互，既然如此，是否可以不用LDP，在PE上直接根据VC ID来手动分配内层标签呢？当然是可以的，这就是SVC的模式，SVC是Martini的简化。公网隧道建立的方法与Martini相同，内层标签在配置VC的时候做好指定，这样就不需要使用VC标签的传递信令了。

所以SVC的网络拓扑模型与报文交互过程与Martini完全相同。

VLL技术的Kompella方式

Kompella是另一种VLL技术的主流模式，目前最新的协议草案是

[draft-kompella-l2vpn-l2vpn-01](http://tech.huawei-3com.com/article.php/2788)

<http://tech.huawei-3com.com/article.php/2788>

Kompella方式的L2 VPN与MPLS L3 VPN很相似，是使用BGP作为交换信令的。与MPLS L3 VPN类似，各个PE之间通过建立BGP会话自动发现L2 VPN的各个节点，并传递VPN信息，使用vpn-target来区分不同的VPN，这使得VPN组网具备了极大的灵活性。与Martini完全不同的是这里出现了真正VPN的概念，在不同的VPN内，CE ID是可以相同的。在内层标签的分配上，Kompella方式与Martini方式完全不同。Kompella采取标签块的方式，事先为每个CE分配一个标签块，这个标签块的大小决定了这个CE可以与其它CE建立多少个连接。这样做的好处是允许为VPN分配一些额外的标签，留待以后扩容使用。PE根据这些标签块进行计算，得到实际的内层标签，用于报文的传输。

无论是使用Kompella方式还是Martini方式，二层报文传递时的MPLS封装是完全相同的，都是二层标签。另外，Kompella支持本地连接。

Kompella方式的结构

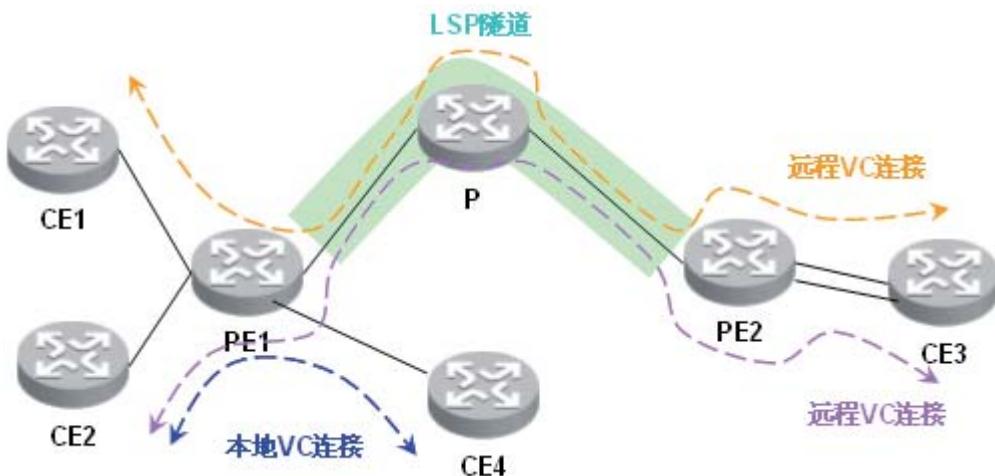


图7 Kompella可以支持的拓扑模型

Kompella方式对各种复杂的拓扑支持能力更好，这得益于BGP的节点自动发现能力。

Kompella的报文交互过程

与Martini方式相同，只不过VC表项的形式略有不同。见Martini的相关章节。

VC标签的计算

Kompella的实现相对复杂，主要是VC标签的计算部分，是最让人搞不明白的地方，因此有必要单独用一个小节来描述一下VC标签是如何计算出来的。

前面我们提到BGP交互的内容是标签块，也就是Label Block，标签块是一个连续的标签范围。那么为了清楚的描述这个标签块，需要定义几个值。首先定义标签块的起始标签LB（Label Base），然后定义这个块的大小LR（Label Range）。这样这个标签块就被清楚的定义了。

当PE上增加一个CE的相关配置时，需要指定标签块的大小LR，LB是PE自动分配的。这个标签块作为一个NLRI条目通过BGP传递到其它PE。当该CE配置被删除或者PE与该CE的连接失效，这个标签块也要被删除，BGP同样会做撤消通告。

假设，在开始部署的时候，CEm需要与远端其它CE建立两条VC，那么定义标签块的大小不能小于2。当然，为了今后扩容的考虑，我们可以定义Rang=10。无论Rang为多大，随着网络的扩容VC数量的增加，总会出现标签不够用的时候。这时候我们就需要重新定义Rang的大小，给一个更大的标签空间。但问题出现了，前面我们谈到标签块的数据是通过BGP的NLRI来传递的，并且这个标签块已经被用于计算VC标签和实际数据的转发。为了不破坏原有的VC连接，采用一个办法，就是给这个CE分配一个新的标签块，并且作为一条新的NLRI通过BGP通告。也就是说一个CE的标签空间可能是由许多个标签块组成的（Rang等于LR的总和），这种机制解决了网络扩展的需求。多个标签之间的联系通过一个偏移量来定义LO（Label-

block Offset）。LO标志出了前面所有标签块大小的总和。比如第一个标签块的LR为100，LO为0；第二个标签块的LR为50，那么LO为100；如果有第三个标签块，它的LO就是150。LO会在VC标签计算被使用。现在每一个标签块都可以用三个参数来描述，那就是LB/LR/LO。

CE ID是在同一个VPN内唯一标识CE的参数。在同一个VPN内，每个CE，其CE ID必须是不同的，CE ID会在每一个NLRI中携带，这样就可以将标签块和对应的CE关联起来。在Kompella draft中，CE ID也被用于VC标签的计算。

假设PEk本地有一个CE-k，其CE ID为k，收到远端PE-m发过来的CE-m的一个或多个标签块，根据Kompella draft规定，CE-k连接到CE-m的标签必须满足以下两个条件：

$$(1) \text{LO}_m \leq k < \text{LO}_m + \text{LR}_m$$

然后从标签块LBm/LRm/LOm中选择标签，标签值为：

$$(2) \text{LB}_m + k - \text{LO}_m$$

公式(1)的右半部分很好理解，结合公式(2)，如果不能满足 $k < \text{LO}_m + \text{LR}_m$ ，CE-k选用的标签将超出LBm/LRm/LOm标签块的范围。但为什么要要求 $\text{LO}_m \leq k$ ，其中包含着什么意义？事实上，理解了 $\text{LO}_m \leq k$ ，也就理解了Kompella标签分配算法中隐含的思想。

根据前面的叙述，LOm代表的是当前这个标签块之前CE-m的标签总和，也就是已经为CE-m分配掉的标签总数。要求 $\text{LO}_m \leq k$ ，就是规定，标签与CE ID是一一对应的。所有标签块里的第一个标签是分给CE 1的；所有标签里的第二个标签，是分给CE 2的。如果CE ID小于LOm，那么代表这个CE的标签应该是标签块LM之前的标签块已经分配过了，所以不会用标签块LM来计算这个CE的标签。CE ID如果是不连续的，那么LOm里面的标签有些会空余出来，这些已分配标签留给谁用呢？当然是留给CE ID比LOm小的CE使用。这反映了Kom-



pella标准设计者的一个默认假设，即，如果一个CE ID为k，那么从0到k的CE ID都应该会被应用，并因此为这些CE预留了标签。

这就提出了一个问题，如果在配置CE ID时，不按自然数顺序编号，而是跳跃性的配置，一些CE ID配置得特别大，就会导致跳跃区域的CE ID没有被使用，但系统仍然给其预留标签，从而浪费标签，这是标准所不乐见的。因此，用户在配置时，最好能对CE顺序分配CE ID，这样能作到最节省的使用标签。

要求 $LOm \leq k$ ，事实上反映了Kompella标准设计者的一个美好愿望：用户配置时，能顺序配置CE ID，这样标准中的标签分配算法就能作到最节省标签空间。

下面详细描述了计算VC标签的过程。假设PE-k本地有一个CE-k，CE ID为k，收到远端PEm为CE-m分配的一个标签块LBm/LRm/LOm。

首先检查从PEk收到CEm的封装类型是否与CEk的相同，如果不一致，停止处理；

检查是否 $k=m$ ，如果是，报错“CE ID k has been allocated to two CEs in VPN X (check CEm at PEk)”，然后停止处理；

如果CEm有多个标签块，检查这些标签块是

否有满足 $LOm \leq k < LOm + LRm$ ，如果任何一个标签块都不满足，报错“Cannot communicate with CE m (PE A) of VPN X:outside range”然后停止处理；

检查和CE-k相关的所有的标签块是否有满足 $LOk \leq m < LOk + LRk$ ，如果任何一个标签块都不满足，报错“Cannot communicate with CE m (PE A) of VPN X:outside range”然后停止处理；

检查PEk和PEm之间的外层通道是否正常建立，如果没有就停止处理，这里假设为LSP隧道，标签为Z；对于外层隧道的判断决定了是否为一个VC计算VC标签，这一点与L3VPN不同。如果外层隧道没有建立，就不应该为VC计算标签。

PEk为CE-m分配内层出标签为 $(LBm + k - LOm)$ ，PEk为CE-k分配内层入标签为 $(LBk + m - LOk)$ ；

PEB到PEA的外层隧道的标签为Z；

内外层标签都已经计算出来，VC处于UP状态，可以用于二层报文的传输了。

我们分阶段来看一个例子：前提是PE之间都是通过BGP来交换标签块的信息。全部公网LSP隧道是正常UP的状态，我们只关心VC标签的计算。图中CE1的CE ID为1，以此类推。在下面的例子中CE22是一个单独的VPN，并且没有建立任何连接，但是PE1同样会为CE22预留一个标签块。

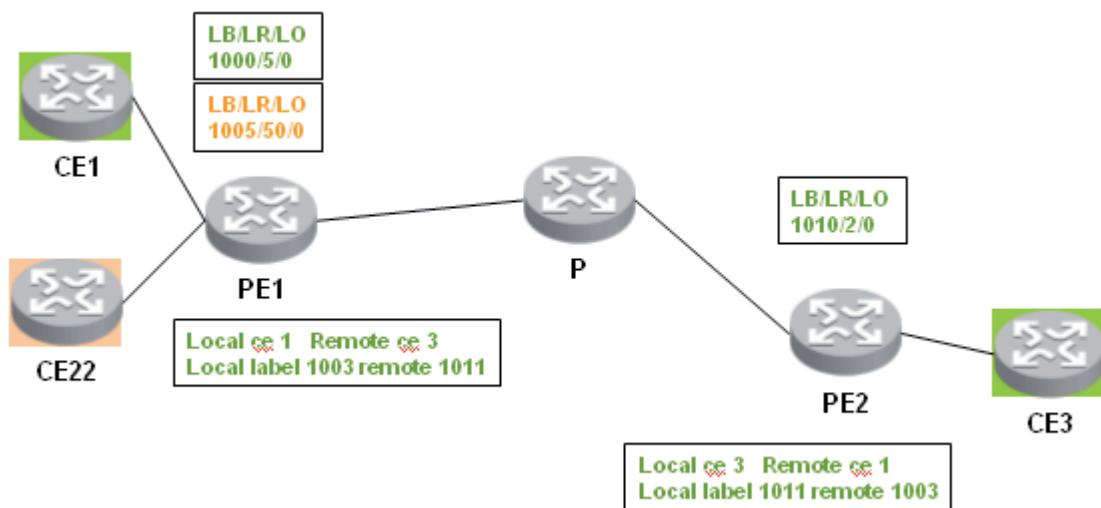


图8 Kompella的VC标签计算例子

假设标签块的分配如图所示。PE1首先为CE1分配标签块1000/5/0，并且收到CE3的标签块1010/2/0。根据前面的计算规则可以计算出VC的入标签及出标签（绿色部分）。同时PE1上还连接了另一个CE22，PE1为它分配的标签块为1005/50/0（接着CE1的标签块继续分配），CE22可能目前并没有建立连接，是为今后的扩容做准备，因此不会为CE22计算标签。CE1与CE3之间隧道的标签计算过程如下，CE1为k侧，CE3为m侧：

PE1侧的判断：

$L_{Om} \leq k < L_{Om} + LR_m$ $0 < 1 < 0+2$ 满足条件

$L_{Ok} \leq m < L_{Ok} + LR_k$ $0 < 3 < 0+5$ 满足条件

PE1进行VC标签计算：

出标签为 $LB_m + k - L_{Om}$ $1010 + 1 - 0 = 1011$

入标签为 $LB_k + m - L_{Ok}$ $1000 + 3 - 0 = 1003$

PE2侧的判断：同PE1，肯定是满足条件的。

PE2进行VC标签计算：

出标签为 $LB_m + k - L_{Om}$ $1000 + 3 - 0 = 1003$

入标签为 $LB_k + m - L_{Ok}$ $1010 + 1 - 0 = 1011$

一个新的CE13加入了绿色的VPN（黑体字部分），需要与CE1建立一条VC。目前为CE1分配的标签块大小为5，而CE ID 13大于CE1的Rang(5)，所以需要更改PE1上CE1的Rang配置，这个例子中修改为15。这样PE1会为CE1分配第二个大小为10的标签块：1055/10/5（接着CE22的标签块继续分配）。PE会收到CE1的两个标签块1000/5/0和1055/10/5，根据计算公式： $5 < 13 < 10 + 5$ 命中CE1的第二个标签块，再判断对端CE ID与本端标签的关系 $0 < 1 < 4 + 0$ 。两个条件都满足，因此根据公式进行VC标签计算（图中黑体字部分）。

根据上面的例子，我们可以看出，如果CE ID不连续，那么Kompella方式的标签分配是会造成浪费的，所以在部署的时候最好配置连续的CE ID。从另一个角度看，考虑到VLL技术的特点，在一个PE上建立的VC数量是非常有限的。因此，即使有些浪费，也完全可以忽略不计。在实际的厂商实现中对一个CE的Rang可配置大小可能会有一些限制，比如500或者1000。这个数目对于一个CE而

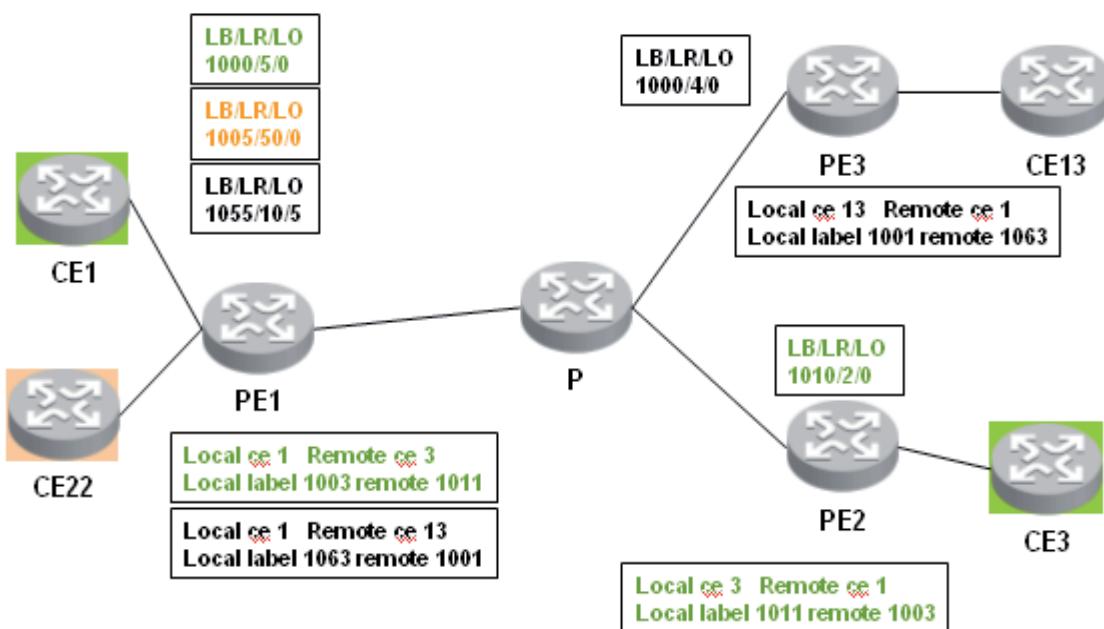


图9 Kompella的VC标签计算例子（续）



言已经足够大了。

在网络的实际部署中，有可能出现这样的情况，网络管理员习惯上用CE ID来标志这个CE的位置，因此CE ID有时候会配置的很大，这样就会造成大量的标签空间浪费，甚至有可能超出厂商的实现中对于Range的限制。解决的办法是可以用CE的名字（字符串）来描述CE的位置，并且建立一个表格记录每个CE对应的CE ID。

VC标签的交互信令

为了交互VC信息，Kompella对MPBGP的NLRI部分做了扩展。用于携带L2VPN的信息。与L3VPN类似，Kompella方式的L2VPN也使用了RD和RT的信息。需要重点提及的一点是VLL技术必然是点到点的VC，如果一个CE需要与多个CE建立VC，那么就需要有多个接口或者子接口。虽然在同一个VPN内，两个CE之间必须有VC连接才能够直接通讯。

下面是NLRI中描述标签块的信息，在可变长的TLV部分有一个CSV（Circuit Status Vector）部分用于描述标签块的LR。

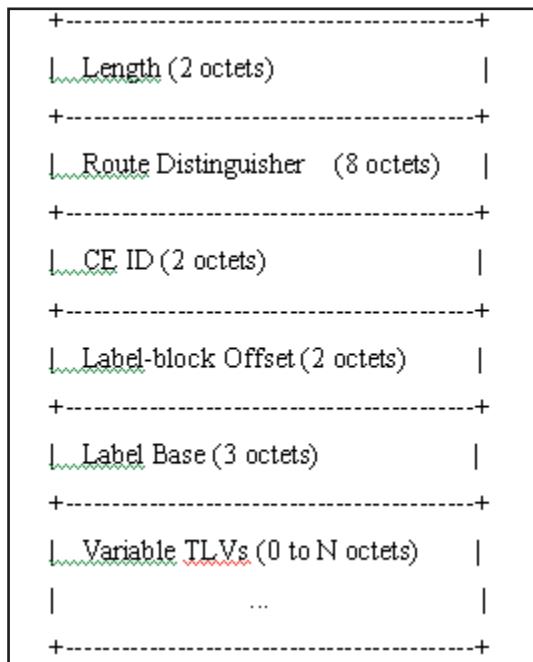


图10 对MPBGP的扩展

为了携带更多的L2VPN信息，定义了一个新的扩展属性

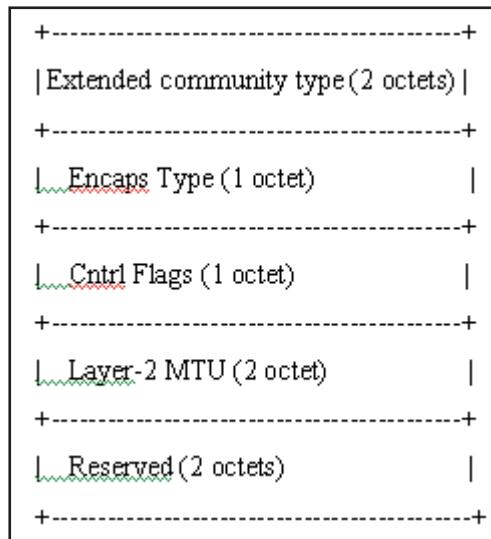
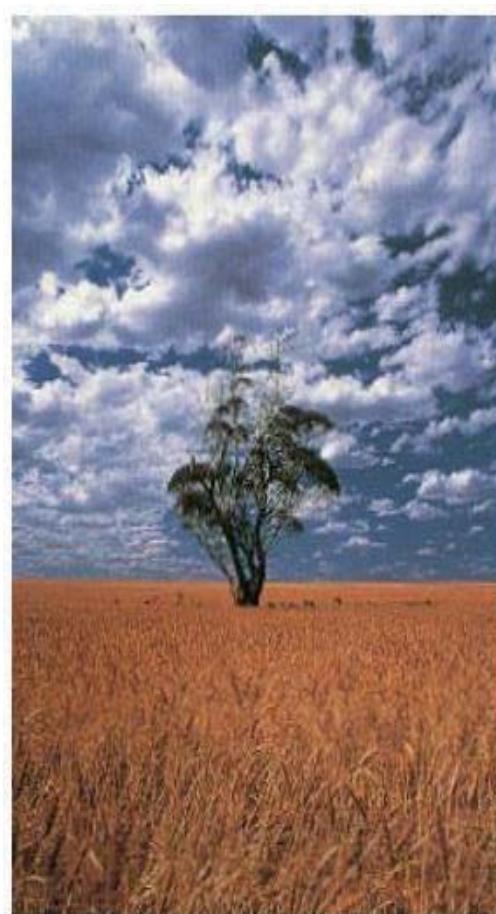


图11 layer2-info extended community



我公司设备上一个携带L2VPN信息的UPDATE报文是这样的。

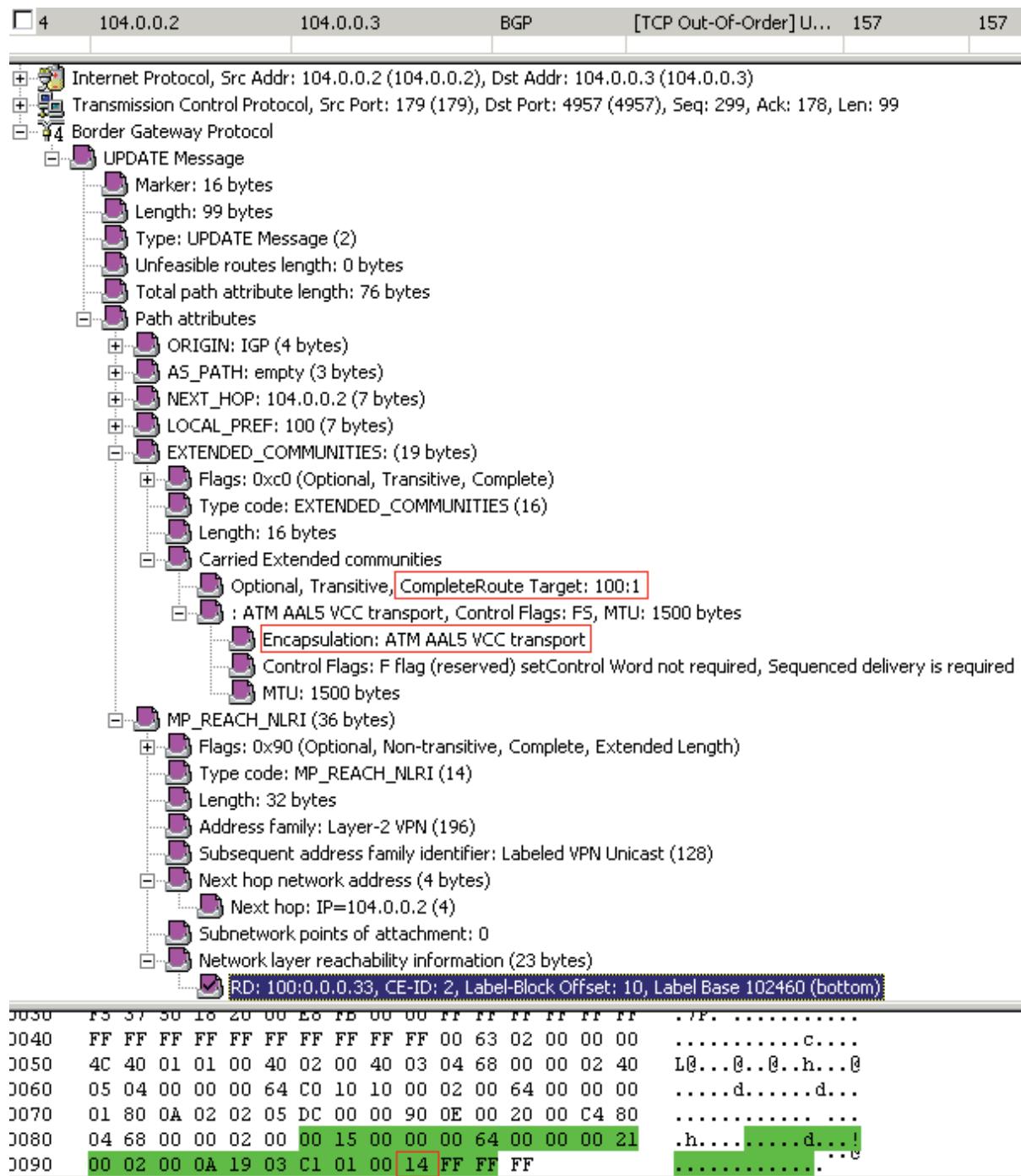


图12 携带L2VPN信息的UPDATE报文

有一个部分没有被解码出来，就是CSV，在这个例子中可以看到跟在LB后面还有几个字节的数据，其中LR是0x14，也就是20。



VLL的跨域问题

VLL的跨域问题与实现方式有关。CCC模式是单层标签，因此只要ASBR之间建立静态LSP，那么就可以完成跨域。我们对照L3VPN中的三种跨域方法来分析一下L2VPN中的其它三种方式。

Option A是部分可以实现的，ASBR互相作为CE-PE在L2VPN的应用中存在一个问题，就是ASBR之间到底使用了什么样的链路连接，是否与VC的封装是一致的。如果恰好VC数量不多，而且ASBR之间也采用了同样封装的链路，那么SVC、Martini、Kompella三种方式的跨域都是可行的，不过因为每一条VC都需要对应ASBR之间的一个子接口，与L3VPN比较，需要消耗更多的资源和更大的配置量。这种模式只能在VLL部署的初期作为一个临时方案，不推荐使用。

Option B需要在ASBR处对内层和外层标签都做交换，目前看来并不适合L2VPN。

Option C显然是最好的解决方案，SP网络设备只需要在不同AS的PE上建立外层隧道就可以了。L2VPN的信息只是在PE间交换，对资源的消耗小，对配置量也没有什么增加。



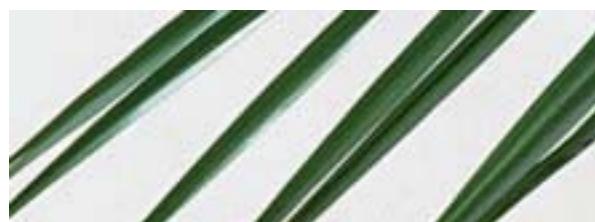
Cisco的AToM

目前Cisco实现的VLL技术有一个比较直观的名字，叫做AToM（Any Transport over MPLS）。是采用了Martini的协议草案。

目前我公司设备与Cisco的互通只能是采用Martini的方式，并且只能在部分链路层封装上互通，比如Ethernet。主要是由于对一些链路协议的2层头处理不一致，目前还不能在所有的链路层协议上互通。

AToM目前在72、75、10700、12000上支持，其中10700、12000只支持部分链路层协议的VLL技术。

可以阅读Cisco的文档《Any Transport over MPLS》做一个了解。（<http://tech.huawei-3com.com/article.php/2652>）

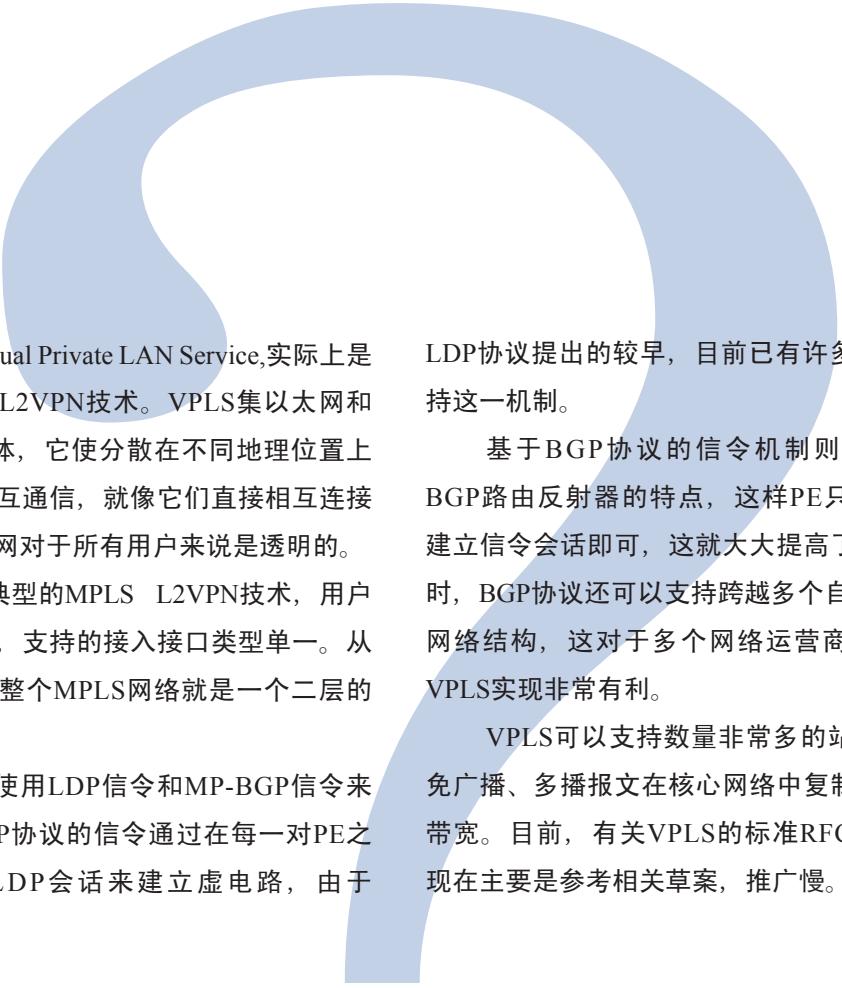


VPLS技术简介

王辉



VPLS为何物



VPLS，即Virtual Private LAN Service,实际上是一种基于以太网的L2VPN技术。VPLS集以太网和MPLS的优点于一体，它使分散在不同地理位置上的用户网络可以相互通信，就像它们直接相互连接在一起一样，广域网对于所有用户来说是透明的。

VPLS是一种典型的MPLS L2VPN技术，用户接入方式为以太网，支持的接入接口类型单一。从用户的角度来看，整个MPLS网络就是一个二层的交换网络。

VPLS可选择使用LDP信令和MP-BGP信令来构建PW，基于LDP协议的信令通过在每一对PE之间建立点到点的LDP会话来建立虚电路，由于

LDP协议提出的较早，目前已有许多厂商的产品支持这一机制。

基于BGP协议的信令机制则可以充分利用BGP路由反射器的特点，这样PE只需路由反射器建立信令会话即可，这就大大提高了可扩展性。同时，BGP协议还可以支持跨越多个自治系统（AS）网络结构，这对于多个网络运营商并存情况下的VPLS实现非常有利。

VPLS可以支持数量非常多的站点，但无法避免广播、多播报文在核心网络中复制，额外占用了带宽。目前，有关VPLS的标准RFC还没有出台，现在主要是参考相关草案，推广慢。

VPLS的基本原理

VPLS基本术语

UPE(User facing-Provider Edge) :

靠近用户侧的PE设备，主要作为用户接入VPN的汇聚设备，草案中称为PE-CLE。

NPE(Network Provider Edge) :

网络核心PE设备，处于VPLS网络的核心域边缘，提供在核心网之间的VPLS透传服务，草案中称为PE-POP。

VSI(Virtual Switch Instance) :

虚拟交换实例。通过VSI，可以将VPLS的实际接入链路映射到各条虚链接上。

PW (Pseudo Wire) :

虚链路，在两个VSI之间的一条双向的虚拟连接，它由一对单向的MPLS VC构成。

AC (Attachment Circuit) :

接入链路，指CE与PE的连接，它可以是实际的物理接口，也可以是虚拟接口。AC上的所有用户报文一般都要求原封不动的转发到对端SITE去，

包括用户的二三层协议报文。

Encapsulation :

封装，PW上传输的报文使用标准的PW封装格式和技术。PW上的报文封装有Tagged和Raw模式。

VPLS的设计思想

VPLS支持PE设备同侧的本地交换，可在本VLAN内进行二层转发。

对于与远端PE下的站点通信，原理与其他L2VPN相同，利用标签栈对用户报文进行封装，以实现其在MPLS网络中的透明传送：外层标签（也称为公网标签）用于将报文从一个PE传递到另一个PE，内层标签（在MPLS L2VPN中，称为VC标签）用于区分不同的VPN中的连接，接收方的PE根据VC标签决定将报文传递给哪个CE。转发过程中，报文的标签栈变化如下图所示：

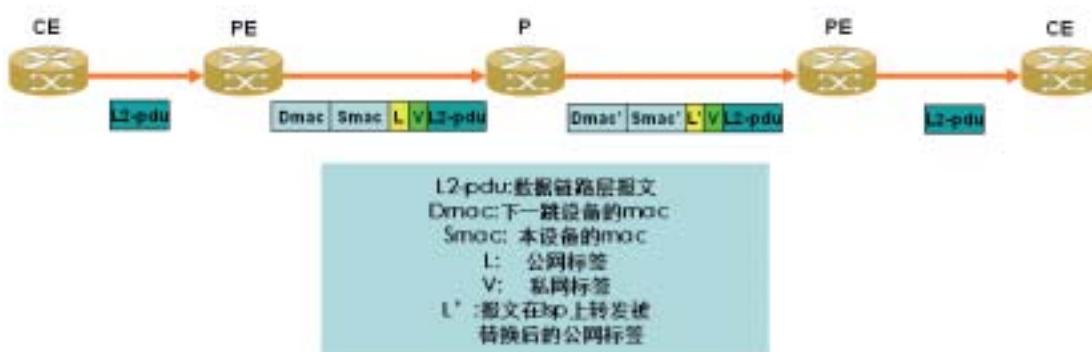


图1: VPLS转发标签变化



VPLS的基本模型

- PE全连接的基本模型

PE全连接VPLS模型的思想是：

接入VPLS的各站点的PE设备之间逻辑全连接，PE设备能在多点之间进行Mac地址学习以及数据包转发。MPLS网络提供隧道来透传VPN站点间的报文，网络中的P设备作用类似与L3VPN中的P设备，它不参与Mac地址的学习与交换，只对MPLS报文进行转发。PE上各个VPN之间的转发表相互独立，使得各VPN间的Mac地址可以重叠。

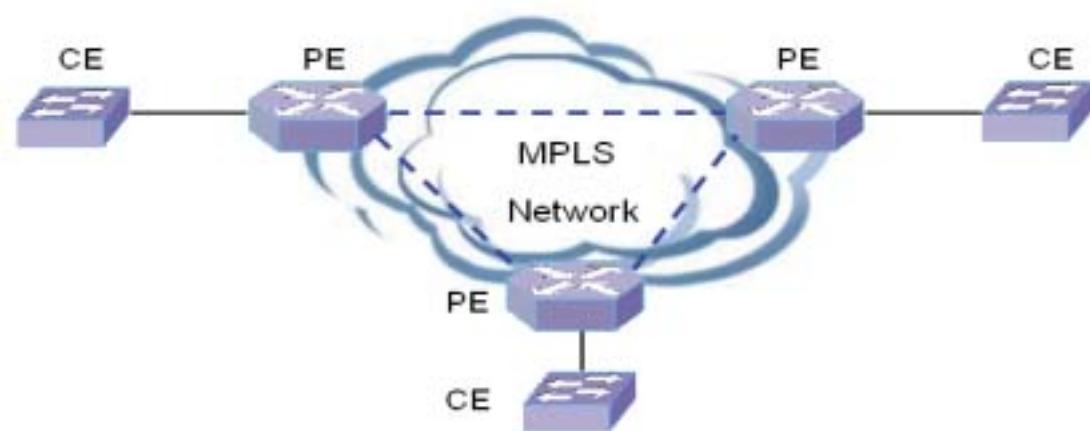


图2：PE全接连模型

- 分层VPLS模型（H-VPLS）

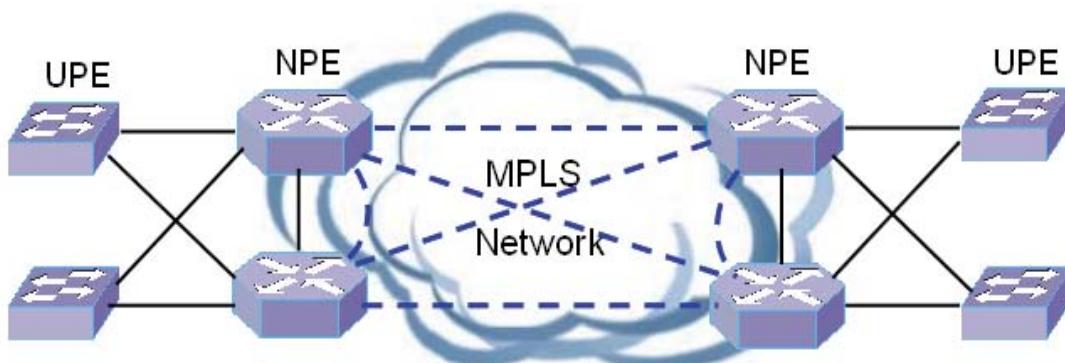


图3：分层VPLS模型

分层VPLS模型的思想是：

所有的NPE设备之间逻辑全连接，UPE只与最近的NPE建立虚连接，通过NPE与对端VPN站点进行报文交换，这样层次化了网络拓扑。H-VPLS模型对NPE性能要求高，因为这里VPN业务流量集中，而对UPE性能要求比较低，因为这里用于VPN业务的接入。另外，还可以在UPE与NPE之间增加链路备份，保证网络的健壮性。UPE与NPE之间的虚连接可以是使用QinQ或LDP来建立。

VPLS转发表中Mac地址的学习和老化

● 与PW关联的远程Mac地址学习

当PE设备在入方向的VC LSP上收到本VPLS内的数据包后将进行Mac学习，然后PW将此Mac地址与出方向的VC LSP形成映射关系，这与二层交换机将学习的Mac地址与出端口对应类似。

● 与用户相连端口的本地Mac地址学习

对于从CE设备转发上来的二层报文，需要将报文的源Mac学习到VSI的对应端口上。

● Mac地址学习能力

电信级网络的典型切换时间是50ms。那么对电信级网络设备学习能力要求为：当64K Mac地址容量时， $64K/50ms = 1.28M\text{次}/秒$ ；当16K Mac地址容量时， $16K/50ms = 320K\text{次}/秒$ 。

转发和泛洪

● 转发

VPLS中的数据转发是通过查找VPLS转发表来完成的，我司S8500设备的转发表的格式如下所示

[S8500-2]display mac-address vsi

Mac ADDR	STATE	VPN ID	PEER	AGING TIME
000d-88f7-cb9b	dynamic	1	Vlan-interface4000	AGING
0010-5ce6-764e	dynamic	1	1.1.1.1	AGING

表中显示设备学到了一条本地Mac地址和远端Mac地址。对于目的mac地址为“0010-5ce6-764e”的报文，S8500将通过NP板对其进行处理进行MPLS转发到VSI对端1.1.1.1；对于目的mac地址为“000d-88f7-cb9b”的报文，S8500直接对报文在vlan4000内进行二层转发。

● 泛洪

设备从VSI中的一个端口上收到未知单播、多播、广播报文时，采用泛洪的方式向VSI本地和所有VSI对端转发；从PW上收到的未知单播、多播、广播报文时，则只向VSI本地泛洪，而不再向其他对端PE泛洪（即，水平分割原则）。

PE全连接与环路预防

● 为什么要PE间全连接

非全连接拓扑需要多个PE之间的数据转发，容易形成环路。另外如果是非全连拓扑，

可能需要使用STP，而在运营商核心网络里他们是不愿意部署STP的。

● 全连接后环路的预防

全连接网络仍然有可能存在环路，因此在公网和私网仍然需要想一些办法来预防环路的发生。

在公网：使用水平分割原则，规则是从公网侧PW收到的数据包不再转发到其他PW上，只能转发到私网侧。即全连接后通过不使数据在PE间二次转发来避免环路（分层VPLS的UPE，SPE之间转发是特例）。

在私网：VPLS网络允许用户在VPN内运行STP协议，BPDU报文只是在ISP的网络上透传。



AC上的接入方式

- 传统的VLAN接入方式

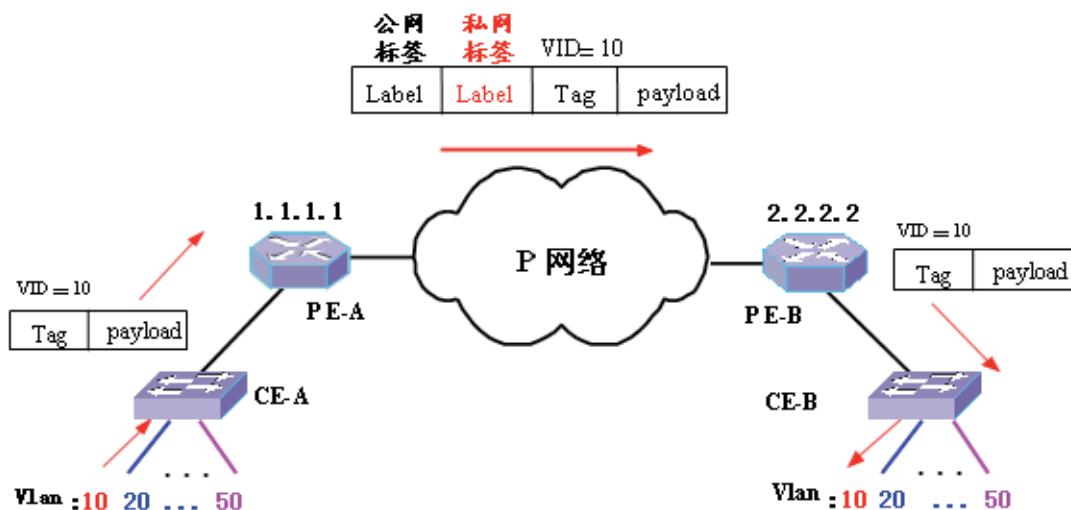


图4 传统的VLAN接入方式

如上图所示，CE-A与CE-B属于同一VPN，CE-A使用vlan10-vlan50做VPLS的接入vlan，则PE-A上也必须配置vlan10-vlan50，且两设备间端口以trunk方式连接。具体的转发流程如下：

在PE-A和PE-B上配置vlan10属于同一VSI，网络上PE、P各设备做相应路由、MPLS配置。

CE-A下行的vlan10内收到用户目的为CE-B下挂主机的报文后，在vlan10内查找MAC地址转发表后，向PE-A转发，且为报文打上Tag值10。

PE-A接收到CE-A转发来的报文，在本VSI内查找VPLS转发表，发现报文将被转发到PW对端PE-B，地址为2.2.2.2。

PE-A首先将报文封装一层私网标签，该标签是PE-B与PE-A协商后分配的，然后私网标签外

再封装一层用于在公网LSP转发的公网标签，报文在P网络中做MPLS转发到达PE-B设备，报文的格式请参考图1。

报文被转发到PE-B上后，只剩下一层私网标签，PE-B根据私网标签判断出该报文属于的VSI后，在该VSI内查找VPLS转发表，然后发给CE-B。

CE-B将收到与CE-A转发出来且未被PE-A封装前一样的报文，CE-B根据报文的Tag值在vlan10内做二层转发，使得报文到达目的主机。

但我们不难看出，传统的VLAN接入方式有明显的不足之处：

在实际网络中，一台PE设备必定下挂多台CE设备，那么同一PE下挂的不同CE间VLAN-ID一定不能重叠，否则不同CE设备上相同VLAN之间将

无法隔离；PE设备和CE设备的VLAN资源是有限的；ISP将干预用户在私网内VLAN-ID的分配，这显然是不妥当的。

如何能够对传统的vlan接入方式优化和改进呢？

● QINQ接入方式

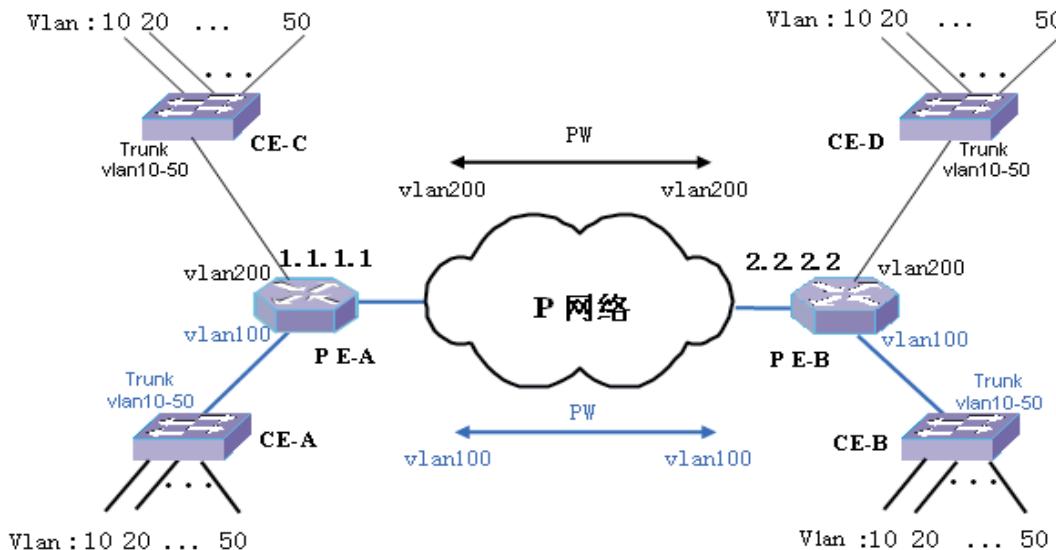


图5 QINQ接入方式

和传统vlan接入方式相比，在PE设备上使用QINQ技术后，同一PE下挂的CE设备vlan-id可以重叠、PE设备vlan-id不必与CE的vlan-id相同，完美的解决了传统vlan接入方式的不足。

在传统vlan接入方式转发流程的基础上，我们看看QINQ接入方式下的报文转发流程：

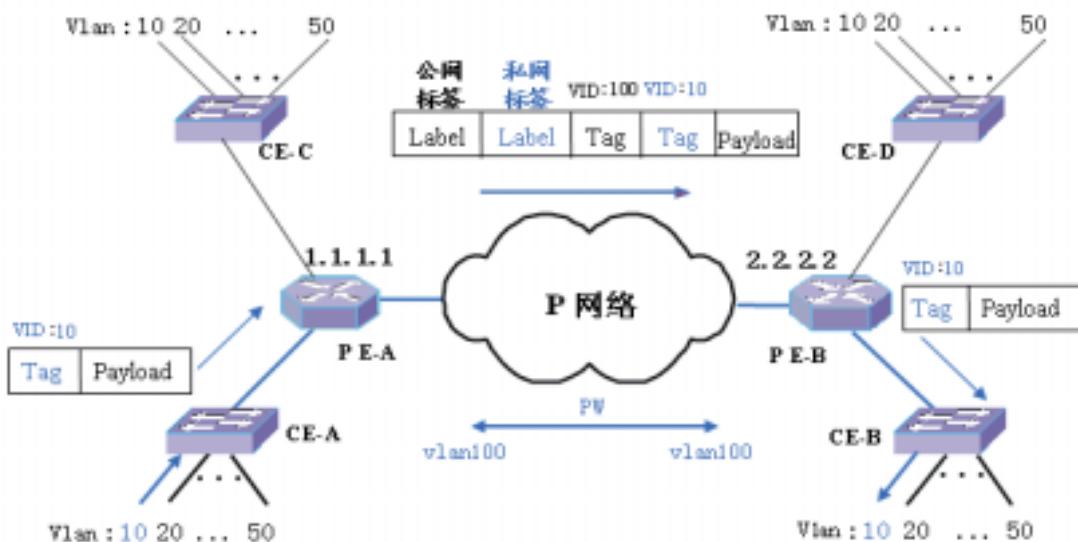


图6 QINQ接入方式下报文的转发



PE-A :

- 给从CE-A转发来的报文再加上一层vlan-id为100的Tag头

- 给报文加上为VLAN 100所对应的VC私网标签
- 给报文加上能够到达PE-B的公网标签
- 向公网转发报文

PE-B :

- 根据私网标签判断出该报文属于的VSI，再根据报文的目的mac查找到出端口，转发向CE-B
- PE的出接口为Access端口，去掉由PE-A加上的外层Tag头
- 报文转发到CE-B，此时的报文的Tag=10

PW上的封装方式

● 两个概念

U-TAG:用户定界符,对PE设备来讲， ethernet接入方式从CE转发上来报文的tag值是u-tag，原则上u-tag要求原封不动地被转发到对端站点的CE设备上去。

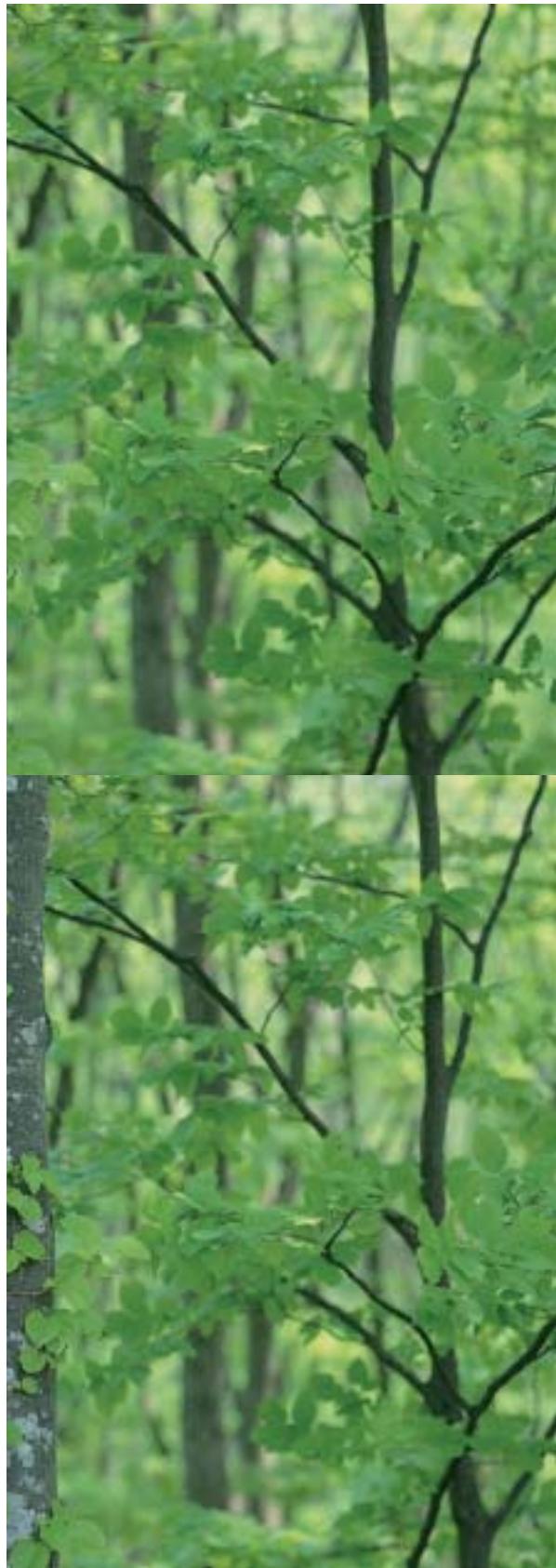
P-TAG:服务定界符， 对PE设备来讲， Vlan接入方式从CE转发上来报文的tag值是p-tag， 使用QINQ为报文封装的tag也是p-tag。而p-tag如何在PW上传输需根据PW上的封装方式来确定。

● PW上的两种封装方式

Raw封装:在PW上转发报文的时候要求去掉p-tag。

Tagged封装: 在PW上转发报文的时候要求保留p-tag。

可以视对端PE与CE连接的下行端口的link-type来灵活选择PW上的封装方式，这两个概念稍后我们将详细的介绍。



LDP信令下的VPLS

LDP信令介绍

PW间隧道的建立有两种方式：

LDP (draft-ietf-l2vpn_vpls_ldp_xx)，即Martini方式

MP-BGP (draft-ietf-l2vpn_vpls_bgp_xx)，即Kompella方式

LDP信令采用VC-TYPE+VC-ID来识别一个VC。VC-TYPE说明这个VC的类型，同一个VC-TYPE的所有VC中，其VC-ID必须在VPN内唯一。连接CE的PE通过LDP交换VC标签，并通过VC-ID将对应的CE绑定起来。当连接两个PE的LSP建立成功，双方的标签交换和绑定完成后，一个VC就建立起来了，两个CE就可以通过这个VC传递二层数据。为了在PE之间交换VC标签，Martini草案对LDP进行了扩展，增加了VC FEC类型。此外，由于交换VC标签的两个PE可能不是直接相连的，所以LDP必须使用remote peer来建立session，并在这个session上传递VC FEC和VC标签。

采用LDP相对MP-BGP简单，但LDP不能提供VPN成员自动发现机制。

LDP信令需要在每两个PE之间建立remote session，session连接数存在N平方问题。

PWid FEC Element and Mac List TLV

LDP LABEL MAPPING MESSAGE中的FEC Element中含有“FEC类型、VC-ID、VC类型、Label TLV、链路MTU”等。在进行私网标签协商时，如果VC类型、MTU不同则有效的PW无法建立。

VPLS内通过Mac地址学习和Mac老化机制来保持对Mac地址的刷新。

另外，draft-ietf-l2vpn-vpls-ldp-06提出了一种新的TLV，被LDP的Mac地址回收消息使用。当本端VPN的链路发生变化的时候，为了保证快速收敛（如双归属情况下发生的主备倒换），此时PE设备可以向对端PE发送Mac地址回收消息来回收对端学习的Mac地址。另外草案描述了一种特殊的Mac地址回收消息，即Mac List为空的消息，意思是让对端删除所有从本端PW上学习到的Mac地址。

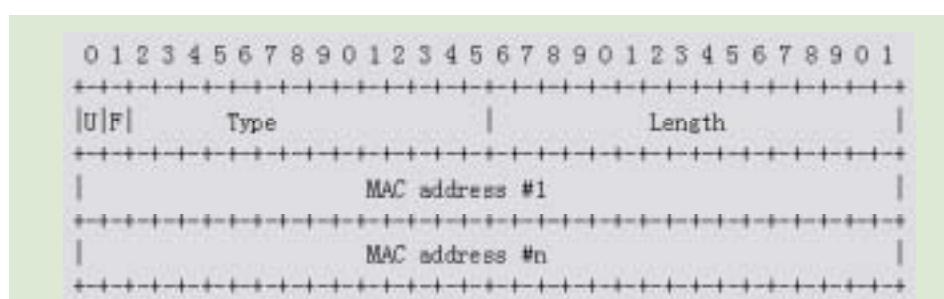


图8 MAC地址回收消息

U bit：未知比特位，必须被设置为1，表示识别该TLV

F bit：转发比特位，必须被设置为0，表示该消息不被转发

Type：类型，必须被设置为0x0404，表明是Mac List TLV

Length：长度，定义的是在该TLV中Mac地址的总长度

Mac address：被指定要回收（删除）的Mac地址



理解VPLS

在我司

S8500交换机上

如图所示，使用两台S8500和一台二层交换机搭建一个PE背靠背的简单环境。S8500-1与S8500-2上配置vsi huawei，vsi-id=500，S8500-1的lsr-id为loopback0的环回地址1.1.1.1，S8500-2的lsr-id为loopback0的环回地址2.2.2.2，两台设备互相配置remote-peer为对端lsr-id，都在vlan-interface200上启动

MPLS和LDP协议。二层交换机用来灵活的控制是否为报文填加tag值。

首先配置两台S8500设备上AC都选择为vlan接入，PW上选择Raw封装，我们通过抓包的方式来看看私网标签协商的过程。

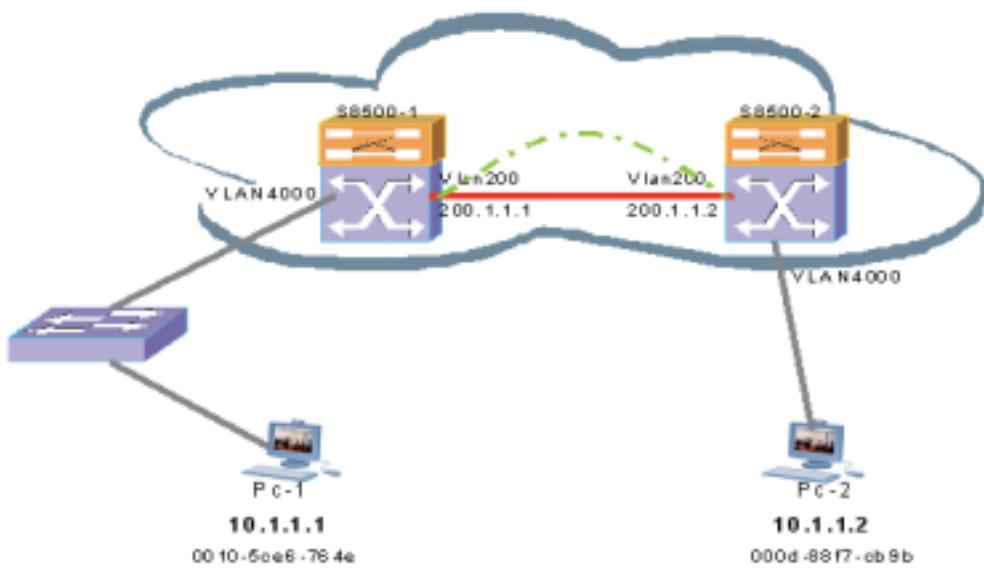


图9 S8500 VPLS背靠背组网环境

下面是S8500-1发出的LABEL MAPPING MESSAGE

```

# Frame 32 (108 bytes on wire, 108 bytes captured)
# Ethernet II, Src: 00:00:00:Fc:5a:ce (00:00:00:Fc:5a:ce), Dst: 00:00:00:Fc:6a:8a (00:00:00:Fc:6a:8a)
# Internet Protocol Version 4, Src Addr: 1.1.1.1 (1.1.1.1), Dst Addr: 2.2.2.2 (2.2.2.2)
# Transmission Control Protocol, Src Port: 646 (646), Dst Port: 1024 (1024), Seq: 0, Ack: 0, Len: 54
# Label Distribution Protocol
# Version: 1
# PDU Length: 54
# LSP ID: 3.3.3.3 (3.3.3.3)
# Label Space ID: 0
# Label Mapping Message
#     0... .... - u-bit: unknown bit not set
#     Message Type: Label Mapping Message (0x400)
#     Message Length: 40
#     Message ID: 0x000001000
# Forwarding Equivalence Classes TLV
#     00... .... - TLV unknown bits: known TLV, do not forward (0x000)
#     TLV Type: Forwarding Equivalence Classes TLV (0x100)
#     TLV Length: 16
#     FCC Elements
#         - FEC Element 1 VCID: 500
#             FEC Element Type: virtual circuit FEC (0x200)
#             0... .... - C-bit: control word NOT Present
#             .0000 0000 0000 0100 - VC Type: Ethernet VLAN (0x0004)
#             VC Info Length: 8
#             Group ID: 0
#             VC ID: 300
#             - Interface Parameter: MTU 1500
#                 ID: MTU (0x01)
#                 Length: 4
#                 MTU: 1500
# Generic Label TLV
#     00... .... - TLV unknown bits: known TLV, do not forward (0x000)
#     TLV Type: Generic Label TLV (0x200)
#     TLV Length: 4
#     Generic Label ID: 12345678
# Label Request Message ID TLV
# 0000 00 00 Fc 6a 8a 40 00 00 Fc 6a ce 04 08 00 43 c0 ..I....-3...E.
# 0001 00 34 26 00 00 00 FF 00 00 04 01 03 01 01 02 02 ..A.....P.
# 0002 02 03 02 86 04 00 04 6a 74 40 04 07 10 00 30 38 ..T.....P.
# 0003 FF FF 54 06 00 00 00 01 00 32 01 01 01 00 00 00 ..C.....P.
# 0004 04 00 00 28 00 00 18 90 01 00 00 10 00 00 04 08 ..C.....P.
# 0005 00 00 00 00 00 00 00 01 f4 01 04 05 dc 02 00 00 04 ..C.....P.
# 0006 00 02 00 00 01 06 00 00 04 00 00 00 00 00 00 00 ..C.....P.

```

图10 S8500-1发出的LABEL MAPPING MESSAGE

下面是S8500-2发出的LABEL MAPPING MESSAGE

```

b) Frame 16 (142 bytes on wire, 142 bytes captured)
b) ethernet II, Src: SMC 00:1e:0f:c1:b1:b4, Dst: 01:00:0c:16:ce:1b (1.1.1.1)
b) Internet Protocol Version 4, Src Addr: 2.2.2.2 (2.2.2.2), Dst Addr: 1.1.1.1 (1.1.1.1)
b) Transmission Control Protocol, Src Port: 3024 (10240), Dst Port: 646 (646), seq: 42, Ack: 98, len: 88
b) Label distribution Protocol
b) Label distribution Protocol
Version 1
PDU Length: 30
LSP ID: 2.2.2.2 (2.2.2.2)
Label Space ID: 0
= Label Mapping Message
  0... ... = U-bit: Unknown bit not set
  Message Type: Label Mapping Message (0x400)
  Message Length: 40
  Message ID: 0x00001882
= Forwarding Equivalence Classes TLV
  00... .... = TLV Unknown bits: Known TLV, do not forward (0x00)
  TLV Type: Forwarding Equivalence Classes TLV (0x100)
  TLV Length: 16
  = FEC Elements
    = FEC Element 1 VCID: 500
      FEC Element Type: Virtual Circuit FEC (128)
      0... .... = C-bit: Control Word NOT Present
      .000 0000 0000 0100 = VC Type: Ethernet VLAN (0x0004)
      VC Info Length: 8
      Group ID: 0
      VC ID: 500
    p Interface Parameter; MTU 1500
= generic Label TLV
  00... .... = TLV Unknown bits: Known TLV, do not forward (0x00)
  TLV Type: Generic Label TLV (0x200)
  TLV Length: 4
  Generic Label: 131072
= LABEL REQUEST Message ID-TLV

```

图11 S8500-2发出的LABEL MAPPING MESSAGE



通过抓包，我们可以看到，两台S8500交换机对于同一VC (id=500) 进行私网标签的协商，S8500-1分配的VC标签为131073，S8500-2分配的VC标签为131072。

在S8500-2上通过命令来查看，得到的结果如下：

```
<S8500-2>display mpls l2vc verbose
VSI name : huawei , State: , Service: VPLS, Service Status : Open
VC-ID: 500, VC State: up, Destination: 1.1.1.1
Group ID: Local 0, Remote 0, VC Label: Local 131072, Remote 131073,
Tunnel Type: LSP, Tunnel Index: 0
```

可见，交换机上的显示结果和我们通过报文分析的结果是一致的。

现在在PC-2上ping PC-1可以ping通，那么交换机上VPLS转发表是什么样子的呢？

```
[S8500-2]display mac-address vsi
MAC ADDR      STATE      VPN ID      PEER          AGING TIME
000d-88f7-cb9b  dynamic    1      Vlan-interface4000  AGING
0010-5ce6-764e  dynamic    1      1.1.1.1        AGING
```

```
[S8500-2]display mpls lsp
```

```
LSP Information: Ldp Lsp
-----
NO  FEC      NEXTHOP      I/O-LABEL      OUT-INTERFACE
1   1.1.1.1/32  200.1.1.1  ----/3      Vlan200
2   2.2.2.2/32  127.0.0.1  3/----      -----
```

这样，当S8500-2收到PC-2发出目的ip为10.1.1.1、目的mac为0010-5ce6-764e的报文后，将通过查找VPLS转发表后发送到下一跳地址为200.1.1.1的S8500-1上，S8500-1再转发给PC-1。

这里有一个问题，我们看到的转发表项是通过什么触发建立起来的？其实很简单，有兴趣的同学可以思考一下答案及表项的建立过程。

看了协商VC标签的过程后，我们再来研究一下在PW上传输的数据包是什么样子的。通过研究数据包的结构，我们可以从感官上更加清晰地了解到S8500通过VPLS究竟对数据包动了哪些手脚。

如图9所示组网环境，我们将二层交换机与S8500-1的上连口设置为Trunk permit vlan 4000，两台S8500设备AC上均为vlan接入，S8500-1与二层交换机所连物理端口设置为Trunk permit vlan 4000、PW上进行Raw模式的封装。S8500-2与pc-2连接的端口设置为access方式属于vlan4000。



在PC-1上ping PC-2，可以ping通，在S8500-1上抓经交换机处理后发出去的icmp报文，如下：

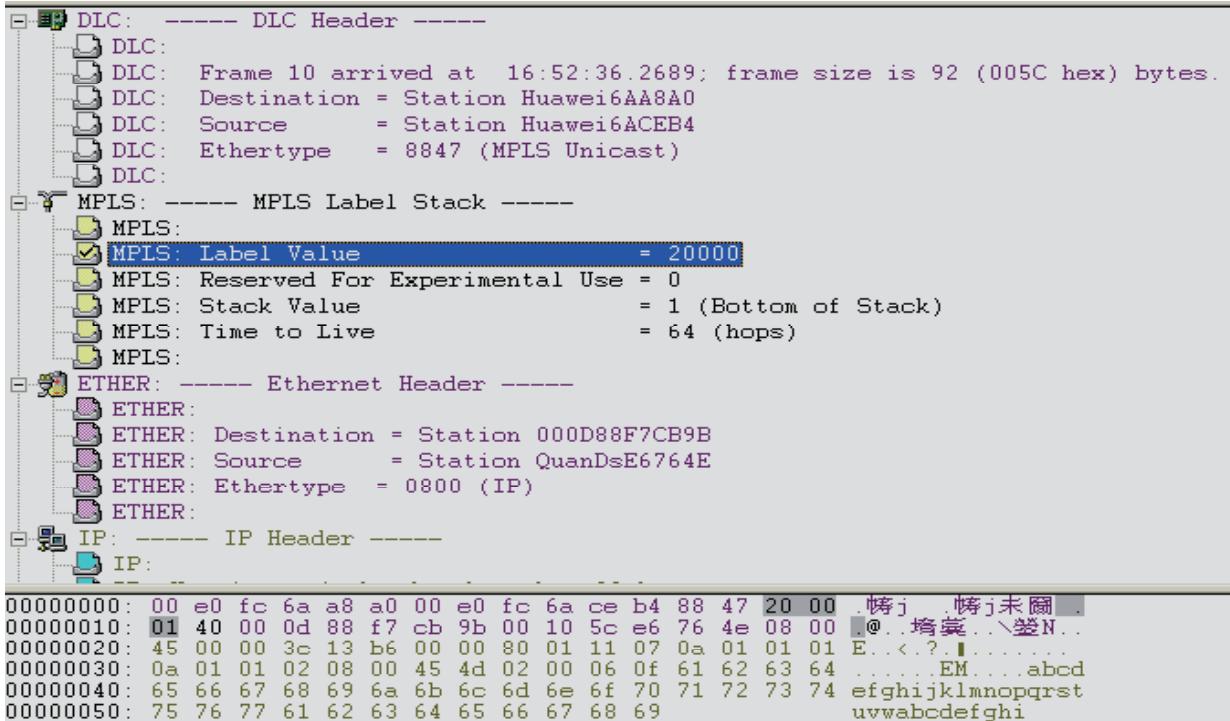


图12 S8500-1发出的icmp报文

分析这个报文的二层头部分：

00-e0-fc-6a-a8-a0：为S8500-2的三层虚接口mac地址，是此icmp报文在公网LSP上转发下一跳设备的mac地址。

00-e0-fc-6a-ce-b4：为S8500-1的三层虚接口mac地址，即发出处理后的icmp报文设备的mac地址。

88 47：在以太网中表示mpls单播报文。

20000：私网标签值，换算为十进制为131072。

01：表明是标签栈底，说明此报文没有MPLS的标签嵌套。

40：TTL值大小为64。

(下面是pc-1发出的原装报文)

00-0d-88-f7-cb-9b：为pc-2的mac地址，即被ping主机的目的地址。

00-10-5c-e6-76-4e：为pc-1的mac地址，即ping主机的源地址。

0800：ip报文

再往下就是ip包头和内容，平时大家见多识广，就不再向下分析了。

结合上面观察，再结合交换机表项的讲解部分，不难发现，数据包的私网标签值正是S8500-1和S8500-2通过LDP协商的结果。至于为什么S8500-1收到的icmp报文是带vid=4000的，但在vpls封装后却消失了呢？看看组网配置就清楚了，因为PW上是Raw模式的封装，这样S8500-1设备就必须去掉这个p-tag。



那么如果PW上进行Tagged模式封装的话，是否数据包真的会带上vid=4000的tag值呢？我们把PW上的封装模式修改为Tagged，然后再抓包，结果如下：

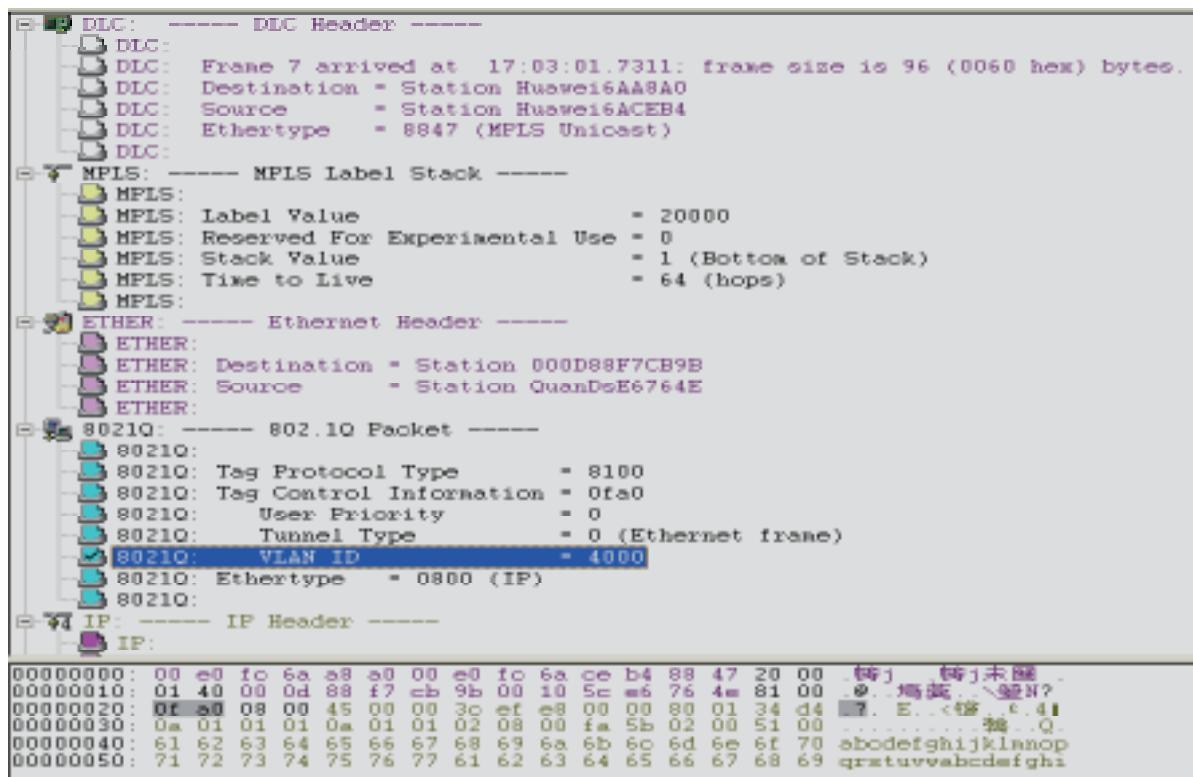


图13 S8500-1发出的icmp报文

与图12的抓包结果相比，唯一的变化就是S8500-1设备发出icmp报文保留了收到icmp报文时的tag值。

我们一起再研究一下QINQ接入情况下报文的格式，仍如图9所示组网环境，我们把上个例子中的AC接入方式配置为ethernet接入并启动VLAN-VPN功能，PW上仍为Raw封装。

仍然在pc-1上ping pc-2，则在S8500-1上抓经交换机处理后发出去的icmp报文，如图14.想一下，既然PW上Raw封装方式，那为什么S8500-1发出的icmp报文里面还有vid=4000的tag值呢？

原因是这样的：pc-1发出icmp后，从二层交

换机转发到S8500-1的时候数据包就带了u-tag（vid=4000），S8500-1收到该数据包后，因为收到数据包的端口配置了QINQ，所以还要为此数据包封装一层p-tag（vid=4000），这样，在PW上发出的报文将去掉一层p-tag，而留下了一层u-tag。

那么如果PW上进行Tagged模式封装，是否数据包真的会带上两层tag呢？我们把PW上的封装模式修改为Tagged，然后再抓包，结果如图15.

可见，此数据包的确是带两层tag的！

至此，VPLS的报文结构和封装都已经介绍完了，相信理解了这些方面的内容以后，大家能够对VPLS的转发流程有了一定的认识了。

```
DLC: ----- DLC Header -----
DLC:   Frame 6 arrived at 15:49:26.3959: frame size is 96 (0060 hex) bytes.
DLC:   Destination = Station Huawei6AA8A0
DLC:   Source      = Station Huawei6ACEB4
DLC:   Ethertype   = 8847 (MPIS Unicast)
DLC:

MPLS: ----- MPLS Label Stack -----
MPLS:
MPLS:   Label Value          = 20000
MPLS:   Reserved For Experimental Use = 0
MPLS:   Stack Value          = 1 (Bottom of Stack)
MPLS:   Time to Live         = 64 (hops)
MPLS:

ETHER: ----- Ethernet Header -----
ETHER:
ETHER:   Destination = Station 000D88F7CB9B
ETHER:   Source      = Station QuanDsE6764E
ETHER:

8021Q: ----- 802.1Q Packet -----
8021Q:
8021Q:   Tag Protocol Type      = 8100
8021Q:   Tag Control Information = 0fa0
8021Q:   User Priority          = 0
8021Q:   Tunnel Type            = 0 (Ethernet frame)
8021Q:   VLAN ID               = 4000
8021Q:   Ethertype              = 0800 (IP)
8021Q:

IP: ----- IP Header -----
00000000: 00 e0 fc 6a a8 a0 00 e0 fc 6a ce b4 88 47 20 00 僕j 僕j未圓 .
00000010: 01 40 00 0d 88 f7 cb 9b 00 10 5c e6 76 4e 81 00 @.. 僕莫..\墨N?
00000020: 0f a0 08 00 45 00 00 3c c1 61 00 00 80 01 63 5b ?. E..<霧..e..o[
00000030: 0a 01 01 01 0a 01 01 02 08 00 72 50 02 00 d9 0b .. .rP..?
00000040: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 abcdefghijklmnop
00000050: 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 qrstuvwxyzabcdefghi
```

图14 S8500-1发出的icmp报文

```

DLC: ----- DLC Header -----
  DIC: Frame 28 arrived at 16:08:43.1131; frame size is 100 (0064 hex) bytes.
  DIC: Destination = Station Huawei6AA9A0
  DIC: Source      = Station Huawei6ACEB4
  DIC: EtherType   = 0847 (MPLS Unicast)
  DIC:

NPLS: ----- MPLS Label Stack -----
  NPLS:
    NPLS: Label Value           = 20000
    NPLS: Reserved For Experimental Use = 0
    NPLS: Stack Value           = 1 (Bottom of Stack)
    NPLS: Time to Live          = 64 (hex)
    NPLS:

ETHER: ----- Ethernet Header -----
  ETHER:
    ETHER: Destination = Station 000D88F7CB9B
    ETHER: Source      = Station QuanDmX6764K
    ETHER:

8021Q: ----- 802.1Q Packet -----
  8021Q:
    8021Q: Tag Protocol Type     = 8100
    8021Q: Tag Control Information = 0fe0
    8021Q: User Priority        = 0
    8021Q: Tunnel Type          = 0 (Ethernet frame)
    8021Q: VLAN ID              = 4000
    8021Q: EtherType             = VLAN (8c1f00)
    8021Q:

V4 L7: ----- Wellknown EtherType data -----
  V4 L7:
    00000000: 00 00 fc b6 e8 a0 00 00 fc b6 ca b4 38 47 20 00 转接 纯文本
    00000010: 01 40 00 0d 88 f7 cb 9b 00 10 5c #e 76 4e 81 00 .端口 ``虚端口
    00000020: 0f a0 81 00 0f a0 08 00 46 00 00 00 3d cb a8 00 00 .端口 ? E ``虚端口
    00000030: 00 01 49 14 0a 01 01 01 0a 01 01 02 08 00 67 4e 1 1 0H
    00000040: 02 00 04 0d 61 62 63 64 65 66 67 68 69 6a 6b 6c ?abcdefghijkl
    00000050: 6d 6e 6f 70 71 72 73 74 75 76 77 61 62 63 64 65 mnopqrstuvwxyzabcde
    00000060: 66 67 68 69 fghi

```

图15 S8500-1发出的icmp报文



分层VPLS (H-VPLS)

上回书提到，为了防止环路，我们在PE间建立 full mesh lsp，这样将导致出现了两个问题：当新增加一台PE时，该台PE都需要与其他PE进行全连接的配置，这样当PE很多的时候，配置工作量相当大；另外广播包复制的复制问题，导致在PE设备间洪泛，影响带宽。

将PE分层的H-VPLS模型可以在一定程度上解决上面的问题。

H-VPLS的核心思想是通过把网络分级，NPE与其他NPE建立全连接，且转发遵循水平分割，但数据可以向NPE下挂的UPE转发。UPE间无须建立全连接，UPE与NPE间数据转发不遵循水平分割。

分层PE间的设备可以通过QinQ或者PW来连接：

聚设备，它只跟NPE-A建立一条虚连接U-PW，跟其他所有的对端都不建立虚链接。数据转发流程如下：UPE-A负责将CE-A上送的报文发给NPE-A，同时打上U-PW对应的MPLS标签，NPE-A收到报文后，根据数据包携带的MPLS标签来判断报文所属的VSI，再根据用户报文的目的MAC打上对应的VC私网标签和在公网LSP转发的公网标签，然后转发报文。NPE-B收到报文后，根据VC私网标签判断数据包所属的VSI，再查找VPLS转发表后转发出去。

如果CE-A与CE-B为本地CE之间交换数据，由于UPE-A本身具有桥接功能，UPE-A直接完成两者间的报文转发，而无需将报文上送NPE-A。不过对于未知单播、广播和多播报文，UPE-A在进行桥

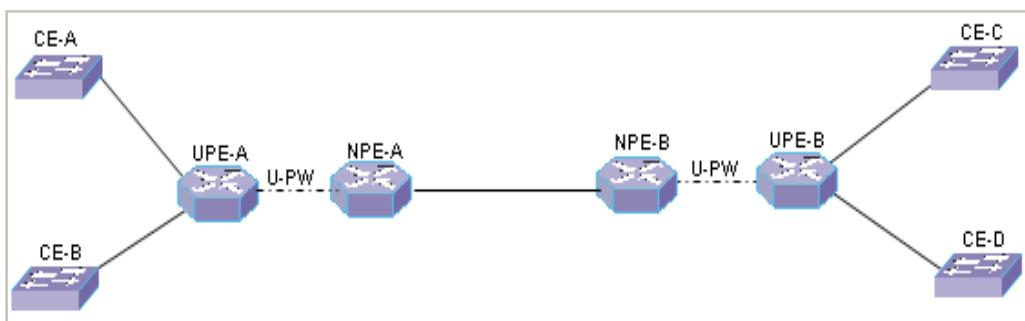


图16 基于LDP的分层VPLS模型

● 基于LDP的分层H-VPLS模型

上图所示为一个H-VPLS模型，UPE-A作为汇

广播的同时，仍然会通过U-PW转发给NPE-A，由NPE-A来完成报文的复制并转发到各PW对端。

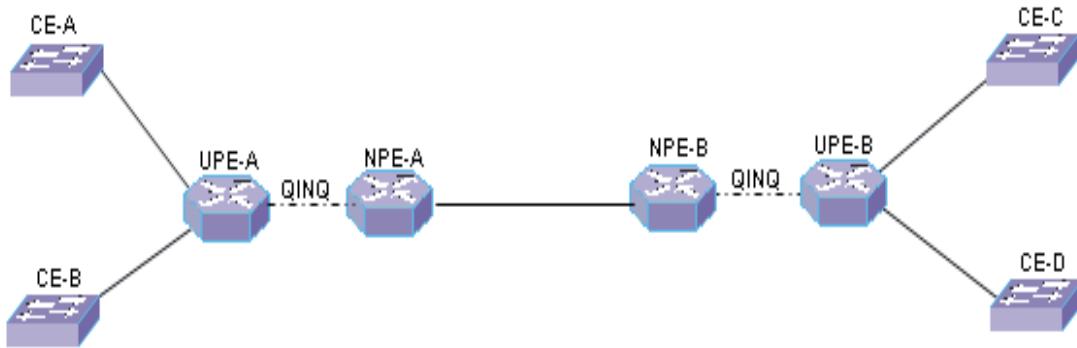


图17 基于QINQ的分层VPLS模型

● 基于QINQ的分层H-VPLS模型

与基于LDP的分层模型类似，只不过分层PE间不是使用信令生成的PW连接，而是采用QinQ隧道。如图所示，UPE-A为标准的桥接设备，在与CE-A连接的端口使能QinQ，UPE-A为CE-A转发来的数据包打上VLAN-TAG。通过QinQ隧道将报文透传到NPE-A上，NPE-A根据打VLAN-TAG判断报文所属的VSI，再根据用户报文的目的MAC打上对应的VC私网标签和公网标签进行转发。NPE-B收到报文后，根据VC私网标签判断数据包所属的VSI，再查找VPLS转发表后转发出去。

如果CE-A与CE-B为本地CE之间交换数据，由于UPE-A本身具有桥接功能，UPE-A直接完成两者间的报文转发，而无需将报文上送NPE-A。不过对于未知单播、广播和多播报文，UPE-A在进行

桥广播的同时，仍然会通过QinQ隧道转发给NPE-A，由NPE-A来完成报文的复制并转发到各PW对端。

● AC接入链路的备份

UPE与NPE设备之间只有单条链路的方案具有明显的弱点，一旦该接入链路失败，汇聚设备UPE下挂的所有VPN都将丧失连通性。所以，对于H-VPLS的两个接入模型中，我们可以设计有备份链路的存在（参考图3的分层VPLS模型）：在正常情况下，设备只使用一条主链路接入，当系统检测到主接入链路失败时，可以启用备用链路来继续提供VPN业务。

对于LSP接入的H-VPLS，由于UPE与NPE之间运行LDP会话，可以根据LDP会话的活动状态来判断主链路是否失效；对于QinQ接入的H-VPLS，可以在UPE与它相连的NPE设备之间运行STP，保证在一条链路失败以后启用另一条链路。



MPLS TE技术

原理简介

包贵新



前言

近年来，随着多媒体、视频、网络游戏、电子商务等各种应用迅猛增长，Internet服务提供商（ISP）必须不断地对链路扩容、对网络基础结构进行调整，以满足新增业务对于带宽资源的需求。与此同时，流量的高速增长对如何维持可靠的基础结构以满足重要的应用，也提出了挑战。

Internet服务提供商所面临的挑战主要来自于如何使他们的客户满意并保持高速增长。在网络部署完毕后，ISP需要将客户的业务流映射到网络的物理拓扑上。90年代初期，这项工作并不是以一种科学的方法来实现。这种映射的实现只是基于产品的路由配置一业务流简单地被分配到由内部网关协议（IGP）计算出的最短路径上。这种不规则映射的局限是，当某条链路发生阻塞时，需要通过提供额外带宽来解决。现在，ISP网络越来越大，要求的转发速度也越来越快，同时客户的需求也变得越来越高。将业务流映射到物理拓扑上的任务需要以一种完全不同的方式来实现，只有这样，网络上传输的负载才能通过一种受控和有效的方式得到支持。

多协议标签交换（MPLS），因其支持流量工程，而在新型公共网络中被作为一项重要的技术。流量工程（TE）是通过将大量的用户业务转移到预先设定的路径来实现的。这些预先设定的通过ISP MPLS网络特定节点的路径，被称作标签交换路径（LSP）。

本文对MPLS TE技术进行了描述，主要侧重于MPLS TE的功能，对于具体的小特性并没有进行细节描述。每个部分会提供相应的学习提示，希望本文对读者了解和掌握MPLS TE技术有所帮助。

本文没有对路由、MPLS、MPLS VPN、QoS等进行详细描述，要求读者已经具有相关基础。



MPLS TE概述

MPLS TE是Multi Protocol Label Switch Traffic Engineering的缩写。所谓流量工程（TE），简而言之，就是对流量进行管理、控制，是将用户的业务数据流映射到物理拓扑/链路上的一项任务。之所以称为工程，因为实现它不仅仅是一项技术或特性，而是要由一系列技术一起配合来完成。由概念可知，可实现对流量的管理和控制的技术都可以叫做TE。例如，通过修改IGP的Metric值改变路由的选路，从而使流量通过的路径发生改变，这就可以称为TE。那么，什么是MPLS TE？顾名思义，MPLS TE就是运用MPLS技术实现流量工程，也就是运用MPLS技术实现流量的管理和控制。本文以下内容只介绍MPLS TE，未涉及其他流量控制方式。

概括地说，通过MPLS技术实现TE，需要有四个步骤：

一、信息发布

为什么要进行信息发布？发布什么信息？

TE的实现需要网络中的每台设备对整个网络的链路状态有所了解。管理员在一台或几台设备上定义的资源特性需要被网络中其他设备所了解，以便通过特定的算法计算出预期的流量路径。因此需要将一些特定的信息在网络设备之间进行交互。

信息可以包括很多内容，比如链路可用的最大带宽、链路的预留带宽、链路的着色/亲和度等。那么，这些信息是如何在整个网络中的诸多设备之间进行交互的呢？这需要一种链路状态协议来帮助完成，OSPF和ISIS都可以，不过还必须对他们进行扩展。OSPF需要扩展一种LSA类型为10的报文格式，ISIS需要扩展一种TLV类型为22的报文格式。目的只有一个，就是承载TE所需要的信息。而

无论采用OSPF还是ISIS，他们所承载的信息内容都是基本相同的。

二、路径的计算

计算的依据是什么？通过什么算法进行？计算的结果是什么？

一种方式是通过动态算法计算得到的。前面已经提到，对OSPF或ISIS扩展承载TE路径计算所需要信息。MPLS TE计算路径的算法是在SPF基础上扩展的CSPF（Constraint SPF）。标准的SPF算法只根据链路的Cost值进行计算，而CSPF不仅依据链路的Cost，所有其他信息（链路可用的最大带宽、链路的预留带宽、链路的着色/亲和度等）都可以作为计算的依据，最后得到一条满足约束的路径。

还有一种方式，通过明确指定一条路径（Explicit-Path），供建立MPLS TE的Tunnel使用。可以使用严格（Strict）方式和疏松（Loose）方式。但是，对这条指定的路径，也要通过CSPF计算出路径上的资源是否满足TE Tunnel的需求。

三、路径的建立

通过第二个步骤，我们获得了一条通过CSPF计算的最佳路径或者通过静态指定的路径。但是，必须要有一种信令协议沿着这条路径进行标签请求/分配，建立一条CR-LSP路径。（回想MPLS的工作原理，通过标签进行数据转发）。

MPLS TE的路径建立协议目前可以有三种，一种是RSVP-TE，是对原来的RSVP协议进行扩展实现的，在RSVP的Path报文中增加了Label Request等字段，在Resv报文中增加了Label等字段。通过Downstream方向的Path报文请求分配标签和协商其他选项，Upstream方向的Resv报文分配标签和协商回复其他选项，建立一条CR-LSP路径。另外一

种是CR-LDP协议，是对标准LDP进行扩展，实现和RSVP-TE同样的功能。还有一种是类似于静态LSP的方式，叫做静态CR-LSP，通过手动静态设定标签来替代动态标签分配协议（RSVP-TE和CR-LDP）的功能。

四、流量的转发

实现数据流量通过TE Tunnel进行转发主要有三种方法。一种方法是通过静态路由指定到目的网络的下一跳接口为TE的Tunnel接口；第二种方法是通过策略路由指定到目的网络的下一跳接口为TE的Tunnel接口；还有一种方法是通过使TE的Tunnel参与CSPF计算，使Tunnel后的目的网络自动通过Tunnel接口进行发布，并且可控制Tunnel接口后的网络是否发布到IGP域中，称为自动路由（包括IGP Shortcut和Forwarding Adjacency两种方式）。

完成上述四个步骤之后，一条MPLS TE的隧道就建立好了，可以通过它进行流量的转发。

MPLS TE的高级特性和应用都是基于TE Tunnel进行的，隧道的正确建立是其他所有特性的基础。FRR、AutoBandwidth、Reoptimization、Load Balance等特性以及DS-TE、MPLS VPN Over TE Tunnel等应用都是在基本的TE Tunnel正确建立的基础上工作的。

目前，VRP的MPLS TE特性是通过启动封装协议为“MPLS TE”的隧道（暂且称为TE Tunnel）来应用的。可以这样理解MPLS TE技术：MPLS TE技术体系的每一部分内容都是围绕着TE Tunnel进行的。信息发布、路径的计算（OSPF扩展、ISIS扩展、CSPF）是为TE Tunnel的建立搜集信息、执行选路计算；路径的建立（RSVP扩展、CRLPD、静态CR-LSP）是为TE Tunnel分发标签；流量的转发（静态路由、策略路由、自动路由）是将流量引导入TE Tunnel；而那些高级特性是在通过TE Tunnel进行流量转发过程中的一些应用和优化。

具体内容下面分解做详细介绍。

信息发布详解

OSPF TE协议发布

标准的OSPF v2协议基本报文类型有严格的结构，只有对其进行扩展才能承载TE所需要的各种信息，包括带宽信息和管理属性等。这些承载的信息将用于建立扩展的流量工程（TE）使用的链路状态数据库（称为TE DataBase，简称TED），如同标准的Router LSA所建立的链路状态数据库一样。二者的区别就在于TED中有许多附加的链路属性，使

用TED可以监控整个网络中使能了TE功能的链路状态，还可以以自己为根节点计算出基于限制的去往目的网络的路径（CSPF）。

基于Opaque LSA类型3，OSPF扩展出Type 9、10、11三种类型LSAs，每种类型有不同的使用范围，其中Type 10 LSA被用于在一个Area内部承载扩展的链路属性信息。



ISIS TE协议发布

与OSPF类似，IS-IS协议也需要进行扩展来承载TE所需要的各种信息。

IS-IS协议通过ISO 10589 被发布，为了支持IPv4，对其进行了扩展，详见RFC 1195。IS (Router) 通过IS-IS Link State Protocol Data Units (LSPs) 发布路由信息。每一个LSP由固定长度的header和许

多个小集合构成，每个小集合由Type、Length和Value组成，这些小集合我们称之为TLVs。新的TLV用于承载关于构建TE Tunnel所需的链路附加信息。其中，TLV Type 22为扩展IS可达TLV；TLV Type 134为TE Router ID TLV；TLV Type 135为扩展IP可达TLV。

路径计算详解

CSPF计算

起始LSR通过对TED中的信息使用约束最短路径优先 (CSPF) 算法来决定每条LSP的物理路径。CSPF是一种改进的最短路径优先算法，它是一种在计算通过网络的最短路径时，将特定的约束也考虑进去的算法。CSPF算法的入口包括：

- ✓ 从IGP获得并在TED中维护的拓扑链接状态信息；
- ✓ 由IGP扩展承载并储存在TED中的与网络资源状态有关的特性（最大带宽、链路的预留带宽、链路的着色/亲和度）；
- ✓ 由用户设置得到的路径选择约束（如，带宽需求，最大跳转数，和管理策略需求等）。

当CSPF考虑一条新的LSP每个备选节点和链接时，它可基于资源的可用性或所选部分是否违反用户策略约束，而对特定的路径组成部分接受或拒绝。CSPF计算的结果是一个明确路径，该明

确路径包含了一组通过网络的最短路径并满足约束的LSR地址。这个明确路径随即传递给信令部分，信令部分实现在LSP中的LSR建立转发状态。每条LSP的起始LSR在特定时机重复执行CSPF算法。

如IS-IS和OSPF这样的链路状态协议使用Dijkstra's SPF算法计算到达网络中所有节点的最短路径树，路由表就是源于最短路径树。如果路由器执行通常的hop-by-hop路由，那么下一跳就应该是与这台路由器相连的物理接口对端地址。CSPF算法计算到达网络中指定节点的明确路径。在定义明确路径的路由器 (TE的headend端) 上，这些路径被看作逻辑接口，它们提供一条可以到达目的端 (TE的tailend端) 的Label Switched Path (LSP)，我们称为 Traffic Engineering tunnels (TE-tunnels)。TE-tunnels的创建是通过这些tunnels进行流量转发的前提。流量导入tunnel的方式有三种，将在后面的流量转发部分详细介绍。

因为CSPF的路径计算原理与SPF相同，只是

在计算中考虑的因素有增加，所以这里对算法的原理不做细致描述。

为了实现TE，每一台路由器维护从自己发起的所有TE-tunnels，也知道每一条TE-tunnel的tail-end节点路由器。在执行SPF的过程中，当路由器发现去往一个新的节点的路径（或者说，这个新节点从临时帐篷TENT中移往PATHs列表），就必须知道first-hop的信息。有三种办法来实现这点：

1. 检查是否有TE-tunnel的tail-end 路由器直接可达。如果有一条TE-tunnel可以到达这个节点，那么就是用这条TE-tunnel作为first-hop。
2. 如果没有TE-tunnel，并且这个节点是直接连接的，那么first-hop就从adjacency database中获取。
3. 如果这个节点不是直接连接的，也没有

TE-tunnel直接可达，那么去往这个新节点的first-hop就从他的父亲节点中Copy过来。每个节点都有一个或者多个父亲节点，每个节点都是0或多个下游节点的父亲节点。

这个算法的结果就是去往TE-tunnel的tail-end的流量都会通过TE-tunnels来进行转发。如果存在多条TE-tunnels分别通往多个中间节点并都能到达节点X，流量会从距离节点X最近的那条TE-tunnel进行转发。

在配置实现中，除了采用动态计算方式，还有一种方式，通过明确指定一条路径（Explicit-Path）供MPLS TE建立Tunnel使用。并且可以使用严格（Strict）方式和疏松（Loose）方式。但是，对这条手工指定的路径，也要通过CSPF检验路径上的资源是否满足TE Tunnel的需求。

路径建立详解

RSVP-TE

90年代中期，RSVP被开发以防止网络阻塞，它通过允许路由器事先判断它们是否能够满足应用流的需求，然后在可能的情况下预留所需资源来完成。最初，RSVP被设计成在主机间为特定的业务流安装资源预留有关的转发状态。到1997，RSVP成为IETF的标准。但是，RSVP并未在服务提供商网络中广泛应用，因为运行人员担心其需要潜在支持数百万条主机—主机业务流的扩展性及开销。

最初的MPLS设备选择扩展RSVP成为信令系统以支持LSP的建立，使其能够自动地绕开网络故障及阻塞。RSVP扩展后自动进行TE处理提供了简化网络运行所需的重要部分，作为TE信令协议应用RSVP与其原本在90年代中期开发者的预想已经大不相同。

支持RSVP和Multi-Protocol Label Switching (MPLS) 的主机和路由器都可以通过RSVP协议进行标签的分配，当MPLS和RSVP结合的时候，一切都变得非常



灵活。一旦一条LSP建立，流量就会在LSP的入口节点根据分配好的标签通过这条LSP进行转发。这种流量和标签间的对应可以使用许多不同的标准，总之，对应到相同标签的报文的集合被称作同一个forwarding equivalence class (FEC)，也被称为"RSVP Flow"。当通过这种方式对应到一条LSP上进行转发，我们也把这条LSP称为一条" LSP Tunnel"。

扩展的RSVP信令协议使用downstream-on-demand 模式进行标签发布。一条LSP tunnel的标签请求是由入口节点通过发送RSVP Path消息来实现。为了实现请求标签的目的，在RSVP Path消息中扩展了一个LABEL_REQUEST对象。标签请求沿downstream传递到出口节点，而标签的分配由出口节点通过RSVP Resv消息沿upstream方向传递到入口节点。为了实现分配标签的目的，RSVP Resv消息扩展了一个LABEL对象。

扩展的RSVP信令协议也支持明确路由能力，是通过在RSVP Path 消息中扩展了一个EXPLICIT_ROUTE 对象来实现的。EXPLICIT_ROUTE对象记录了明确指定路径中的节点，通过使用这个对象，可以预先定义RSVP-MPLS流的路径而不必考虑传统的IP路由。通常这条明确路径可以手工指定，也可以给予一定的QoS等条件动态计算，也称作control-driven 或者data-driven。在TE中使用explicit routing，MPLS域的入口节点可以控制通过它去往出口节点流量的LSP路径，这样，可以提高网络资源的使用率。扩展的RSVP协议支持strict和loose模式。

目前的RFC标准中，使用扩展的RSVP建立的LSP Tunnel，只支持单播LSP-Tunnels (unicast LSP-tunnels) ，不支持多播LSP-Tunnels (Multicast LSP-tunnels) ；只支持单向的LSP-Tunnels (unidirectional LSP-tunnels) ，不支持双向的LSP-Tunnels (Bidirectional LSP-tunnels) 。

CR-LDP

Label Distribution Protocol (LDP)作用在MPLS域中，为MPLS路由器进行标签的分配，具体内容详见RFC3032 (MPLS Label Stack Encoding) 。类似RSVP，为了使LDP协议可以作为信令协议建立TE LSP Tunnel，必须进行扩展。扩展后的LDP可以配合MPLS一起支持对经过网络的流量进行基于约束的路由 (Constraint-based routing) 。VRP实现的MPLS TE支持在RSVP-TE和CR-LDP这两种动态信令协议中自由选择，同时也支持静态CR-LSP。

CR-LSP over LDP 主要为了满足以下目标：

- ✓ 满足执行TE的需要，提供可以执行约束路由计算的基础
- ✓ 只要可能，尽量建立在已经发布的协议上
- ✓ 保持实现简单性

Constraint-based routing提供了TE所关心的扩展信息，通过这些信息计算一条最优路径。Constraint-based routing (CR)是一种用来实现TE的机制，一条LSP可以基于明确路由约束 (explicit route constraints) 、QoS约束 (QoS constraints) 和其它约束条件创建，被称为基于约束的LSP (constraint-based routed LSP (CR-LSPs)) 。

明确路由 (Explicit Routing) 是约束路由的一部分，他的约束条件就是明确路径 (explicit route (ER)) 。请注意这里的细微差别，前一个是动词，后一个是名词。和其它LSP一样，一条CR-LSP也是通过MPLS网络中的一条路径。区别在于其他的LSP只是简单的基于路由表或者由管理员分配，而CR-LSP是在网络的边缘基于一些标准进行计算出来的。这些用于计算的标准包含路由表但是不完全依赖路由表信息。这样设计的目的就是更关注于LSP的其他特殊特性，如一定的预留带宽、着色等。

为了支持约束路由LSP (CR-LSP) ，需要使用以下LDP机制进行支持：



- ✓ 使用基本的或者扩展的Discovery机制
- ✓ 在downstream方向使用DoD (downstream on demand) 模式的Label Request消息
- ✓ 在downstream方向使用有序 (ordered control) 的Label Request消息
- ✓ 使用通知消息 (Notification Message)
- ✓ 使用撤销 (Withdraw) 和释放 (Release) 消息
- ✓ 使用环路检测机制 (Loop Detection) (在CR-LSP的Loosely路由部分)

同时也增加了新的要求：

- ✓ 用于建立CR-LSP的Label Request消息中增加一个或者多个CR-TLVs, 例如可能增加了ERTLV
- ✓ 一个LSR通过Label Request消息中的一个或者多个CR-TLVs进行有序 (ordered control) 的推断。这表示这个LSR仍然可以被配置为独立

(independent control) 的模式, 在进行动态路由建立LSPs的时候正常工作。但是, 当一个Label Request消息中有一个或者多个CR-TLVs的时候, 就使用有序的方式来进行CR-LSP的建立。需要注意的是, 在CR-LSP的Loosely路由部分也是如此。

✓ 定义了许多新的状态码来通告建立路径过程中的各种错误。在CR-TLVs中, 所有这些新的状态码的“F”位必须被置位。

静态CR-LSP

静态建立CR-LSP原理非常简单, 就是网络管理员以Hop-by-hop方式通过命令手工建立一条从入口节点到出口节点的CR-LSP隧道。这个过程非常类似于标准的静态LSP配置。



流量转发詳解

前面描述了TE的信息发布、路径计算、路径建立部分，本部分的内容主要描述流量是如何通过TE Tunnel进行转发的。有三种方式进行流量的转发，分别是静态路由指定、策略路由指定（PBR）和自动路由（IGP AutoRoute/Shortcut）。

静态路由指定

因为TE Tunnel的接口地址网络没有任何实际意义，所以，通常情况下就不会发布到IGP中。这时候在TE Tunnel的Headend端通过定义一条到达目的网络地址通过TE Tunnel接口的静态路由，就把流量引入到TE Tunnel上进行转发。

策略路由指定

与静态路由指定类似，TE Tunnel的接口地址不发布到IGP中。在TE Tunnel的Headend端先通过ACL定义需通过TE Tunnel接口的流量，再定义路由策略，将匹配该ACL流量的下一跳的接口指向TE tunnel的接口。在流量的入接口应用策略路由，这样实现通过TE Tunnel进行转发。

自动路由发布

自动路由发布，有IGP Shortcut和Forwarding Adjacency两个特性，这两个特性的原理都是使TE Tunnel接口参与IGP的SPF计算。在TE Tunnel的Headend端，TE Tunnel可以看作是Headend的直连接口。在路由表中，目的地为TE Tunnel远端（TE Tailend后面的网络）的出接口为TE Tunnel，也就是使用CR-LSP/TE Tunnel作为出接口。在这种应用中，CR-LSP被看做点到点链路。

IGP Shortcut和Forwarding Adjacency的区别在于：

- 在IGP Shortcut应用中，使能此特性的路由器使用CR-LSP作为出接口，但它不将这条链路发布给上游邻居路由器，因此，其他路由器的链路数据库中根本没有这条路径信息存在，当然也不能使用。
- 如果配置了Forwarding Adjacency，则使能此特性的路由器在使用CR-LSP作为出接口的同时，也将这条CR-LSP作为一条普通的LSA/LSP发布给上游邻居路由器，因此，其他路由器收到后存放在链路数据库中，也能够使用此CR-LSP。

其他重要 高级 特性介绍

本部分的内容主要描述MPLS TE的一些常用高级特性，以及VRP系统的具体实现介绍，希望对读者学习MPLS TE的高级特性有所帮助。

FRR特性

FRR就是快速重路由（Fast ReRoute），也有文档称为快速恢复（Fast RestoRation），是MPLS TE中用于链路保护和节点保护的机制。

当CR-LSP链路或者节点失败时，在发现失败的节点上进行保护，这样可以允许流量继续从保护链路或者节点的隧道中通过以使得数据传输不至于发生中断，要求快速恢复的时间为小于50毫秒（<50ms），因为这个数值是一些时延敏感的业务如VoIP等所可以容忍的。同时Headend节点就可以在数据传输不受影响的同时继续发起主路径的重建。FRR的最终目的就是利用Bypass隧道绕过失败的链路或者节点从而达到保护主路径的功能。

快速重路由（FRR）是基于RSVP TE的实现，遵循RFC4090。实现快速重路由有两种方式：One-to-one Backup方式和Facility Backup方式。One-to-one Backup方式是分别为每一条被保护LSP提供保护，它实现的方法是为每一条被保护LSP创建一条保护路径，该保护路径称为Detour LSP。因此，One-to-one Backup方式又称为Detour方式。Facility Backup方式用一条保护路径保护多条LSP，该保护

路径称为Bypass LSP。因此，Facility Backup方式又称为Bypass方式。

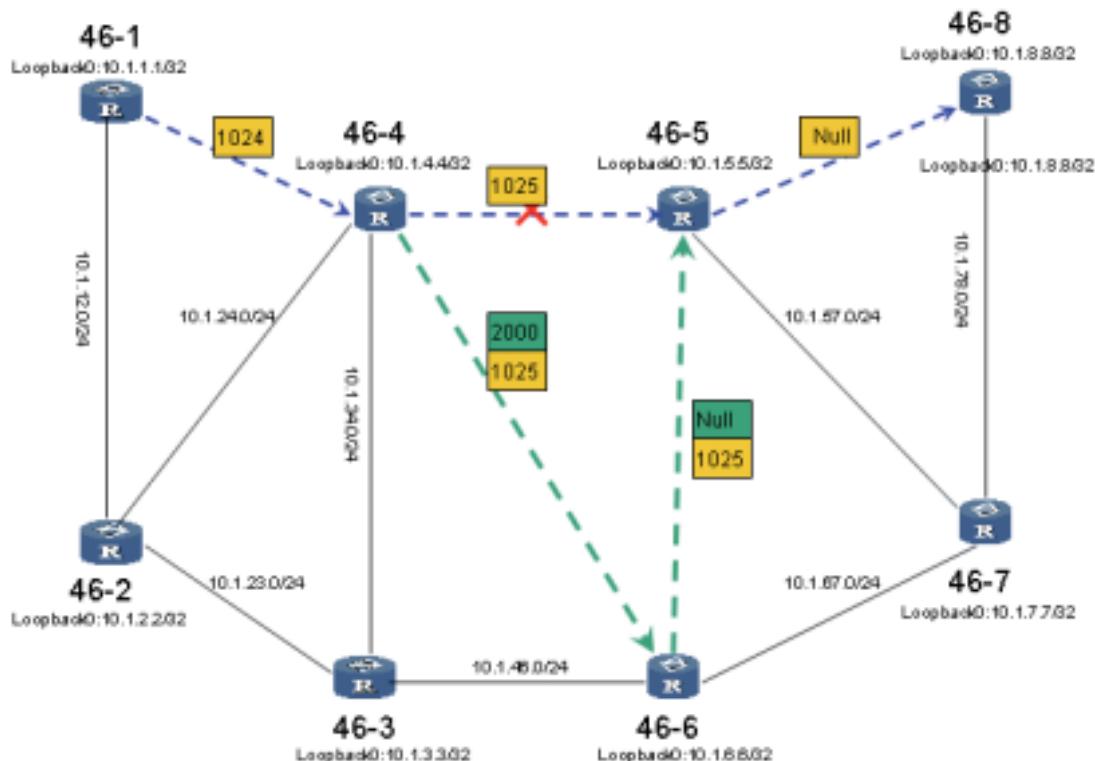
下面介绍几个概念：

- ✓ 主LSP：它是相对于Detour LSP或Bypass LSP而言的，是被保护的CR-LSP。
- ✓ PLR：Point of Local Repair，Detour LSP或Bypass LSP的头节点，它必须在主LSP的路径上，且不可能是尾节点。
- ✓ MP：Merge Point。Detour LSP或Bypass LSP的尾节点，必须在主LSP的路径上，且不可能是头节点。
- ✓ 链路保护：PLR和MP之间有直接链路连接，主LSP经过这条链路。当这条链路失效的时候，可以切换到Detour LSP或Bypass LSP上。
- ✓ 节点保护：PLR和MP之间通过一个路由器连接，主LSP经过这个路由器。当这个路由器失效的时候，可以切换到Detour LSP或Bypass LSP上。

FRR可以分为链路保护（Link Protection）、节点保护（Node Protection）和路径保护（Path Protection）三种，下面分别介绍一下。



首先看一下链路保护：



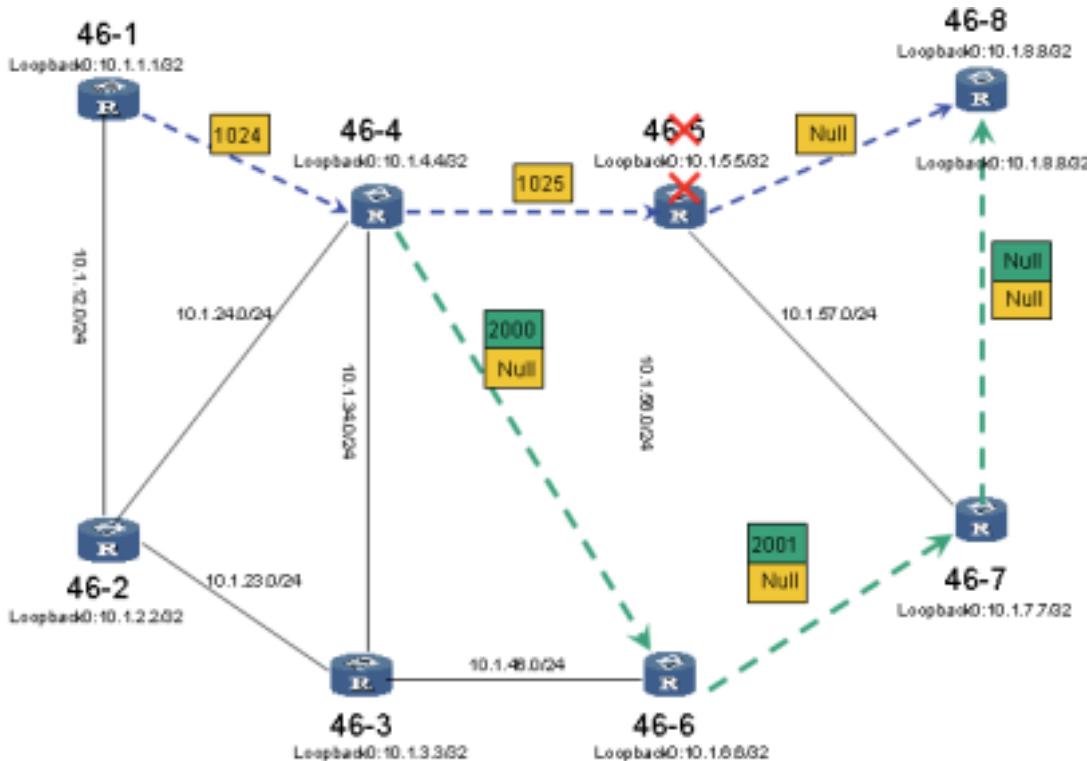
假设从46-1---> 46-4---> 46-5---> 46-8有一条CR-LSP，各个分配好的标签如图所示，因为46-5是倒数第二跳节点，所以46-8给46-5分配的是一个空标签（VRP分配隐式空标签3）；46-5给46-4分配了一个值为1025的标签；46-4给46-1分配了一个值为1024的标签。

当46-4与46-5相连接的链路出现故障的时候，在46-4上会启用46-4----> 46-6-----> 46-5这条Bypass路径进行保护，以保证流量的通过；同时，46-4会将发现的链路故障信息向upstream方向通知

46-1，46-1在下一次进行流量转发之前进行路由的重新计算，通过计算得到最优的路径。当46-4与46-5相连接的链路故障恢复的时候，46-4也会通知46-1，在46-1上的重新计算Timer到期后重新进行路径计算，通过计算的最优路径进行流量转发。

Bypass Tunnel保护通过在原来的MPLS标签栈顶压入新的标签来实现的。如图所示，因为Bypass Tunnel的出口是46-5，所以46-6是这条CR-LSP的倒数第二跳，46-5给46-6分配了一个空标签；46-6给46-4分配了一个值为2000的标签。

下面再看一下节点保护，基本原理和链路保护一样：



当46-5出现故障时，46-4发现后启用Bypass Tunnel进行保护，例子中的保护路径46-4-----> 46-6-----> 46-7-----> 46-8，绕过了节点46-5。

需要注意的是，因为MPLS标签都是只有本地意义的，那么46-4如何知道Bypass Tunnel末端所需要的标签呢？这是通过启用Record-Route和Record-Label参数来实现，Record功能可以记录CR-LSP路径中经过的每个节点以及对应的标签。另一个值得注意的问题是，节点保护的范围涵盖了链路保护的范围。节点保护与链路保护的区别是Bypass Tunnel的目的节点的位置，节点保护的目的节点绕过了被保护的节点；而链路保护的目的节点绕过了被保护的链路。

最后，介绍一下路径保护。

路径保护与前面介绍的链路和节点的区别就是保护的对象从一条链路、一个节点，变成了整条CR-LSP路径。路径保护为一条CR-LSP隧道创建备份的CR-LSP，提高CR-LSP的可靠性。当主CR-LSP失效时，系统可以把业务流量切换至备份CR-LSP上，当主CR-LSP恢复时，再把业务切换回来。

可以采用两种方法实现：

- 备份热备：创建主CR-LSP后立刻创建备份CR-LSP（无论主CR-LSP是否故障），主CR-LSP失效时，通过MPLS TE直接将业务切换至备份CR-LSP上；当主CR-LSP故障恢复后，流量自动切换回主CR-LSP上。



■ 普通备份：当发现主CR-LSP失效后立刻创建备份CR-LSP，采用Make-Before-Break方式，尽可能在主CR-LSP完全失效之前完成新的备份CR-LSP的建立。

采用普通（Ordinary）备份方式与冗余热备份方式的唯一区别在于：冗余热备份是在故障发生之前已经建立好了备份CR-LSP；而普通备份方式是在故障发生之后开始建立备份CR-LSP，配置命令是“mpls te backup ordinary”。其他的情况完全相同，这里就不再累述。

在我司产品的实现中，路径保护的FRR功能也被作为高可靠性HA（High Available）的一种方式，和其它所有的软件和硬件的高可靠性一起提供产品的整体可靠性。

关于FRR更具体更细的描述请参见RFC 4090(Fast Reroute Extensions to RSVP-TE for LSP Tunnels)。

Auto Bandwidth特性

正如前面例子看到那样，TE Tunnel接口可以配置一个预留的带宽值。这样做的一个不便的地方就是如果需要调整带宽大小，只能手工修改，如果一天需要修改多次的话会很烦琐。使用Auto Bandwidth会很简单的实现上述目的，通过启用Auto Bandwidth，可以为一个LSP配置自动进行带宽调整，在环境发生变化时能够动态分配资源，并且不中断业务。

这种需求通常是由于网络管理员最初不能确定有多少业务需要通过核心的网络传输。因此，需要希望TE具备这样一种功能，CR-LSP能在最初时建立TE Tunnel时配置一个预留带宽，当流量增多时，自动调整分配给这些CR-LSP的带宽。

MPLS TE的动态带宽调整特性（Auto Band-

width Adjust）可以实现此功能，这一特性的原理就是在TE Tunnel的接口进行流量转发速率的观察，阶段性的调整TE Tunnel接口带宽的大小以更好的进行流量的转发。当一个Tunnel第一次配置Auto Bandwidth时，一个带宽改变周期（我们称为Timer A）开始启动，在这个Timer A启动之后，每个流量采集周期（我们称为Timer C）都会采集一次Tunnel接口的输出流量，记录一个在这个流量采集周期的输出流量值（我们称为T）。当整个带宽改变周期（Timer A）过期时，会计算出所有流量采集周期中记录的输出流量值（T）的平均值。这个定义了自动带宽调整的TE Tunnel接口根据这个平均值进行接口带宽的重新配置，进行自动修改；同时，重新启动新的一个带宽改变周期（我们称为Timer A）。

VRP系统是通过在TE Tunnel接口下配置“mpls te auto-bandwidth adjustment”和“mpls te auto-bandwidth collect-bw”命令来支持此功能。

Reoptimization特性

Reoptimization特性也就是重优化，那么什么是重优化，在什么情况下需要重优化呢？当一条TE Tunnel已经建立，由于某些原因（IGP的调整、链路的增减等），从Headend到Tailend出现了一条更优的路径，更适合于TE Tunnel进行流量转发。这种情况下，Headend节点路由器将CR-LSP切换到这条更优路径上的机制就称为重优化，可以理解为CR-LSP的重优化，也可以理解为TE Tunnel的重优化，其实都是一样的。

Reoptimization特性的目的就是优化CR-LSP，以达到优化网络资源的目的。一种方法是人工配置（网络管理员手动执行Reoptimization），需要网络管理员根据网络的链路资源情况对CR-LSP进行

优化；另一种方法是使MPLS TE能够动态优化CR-LSP，从而节省人力。动态方式又分为主动（阶段性自动执行Reoptimization）和被动（事件驱动执行Reoptimization）。但是无论主动和被动，所执行的操作都是创建一个新CR-LSP，为之分配新路由，并将业务流量从旧的CR-LSP切换至新的CR-LSP，删除旧CR-LSP。

网络管理员手动执行Reoptimization是根据实际情况，通过手动执行Reoptimization命令来强行执行Reoptimization计算的。

阶段性的自动执行Reoptimization是通过在Headend节点上定义一个Reoptimization Timer来实现的，每个Timer周期结束都会根据TE Tunnel的约束条件执行计算，看是否有更优的到达相同Tailend节点的路径。这个Reoptimization Timer是对于单个TE Tunnel有效的，如果有多个TE Tunnel，每个Tunnel都会根据自己的Timer周期执行Reoptimization计算。如果发现了到达相同Tailend节点的新的更优的路径，会用“Make-Before-Break”方式进行CR-LSP的建立和拆除。

事件驱动执行Reoptimization是在某种情况下，比如某条链路的增加使得从Headend节点到Tailend节点之间最优的路径发生了变化，从而在Headend节点上执行Reoptimization计算的。这种情况下，由于原来的TE Tunnel处于正常连通状态，会通过“Make-before-Break”方式，在新的CR-LSP建立起来之后进行路径的切换，而且不会在这条新的CR-LSP建立起来之后立刻进行Reoptimization，会等待一个Reoptimization Timer之后进行切换。这是为了防止链路flapping导致切换后的业务流量丢失。如果这条最优路径的链路经常flapping，而且在它up时就立刻切换，那么它down了，导致路径中断、流量丢失。

需要注意的是，如果一条TE Tunnel状态变为down，则会立即执行Reoptimization而不是等待Reoptimization的Timer超时。在具体实现上，可以

有两种方式：一种是在全局模式下使能Reoptimization。通过在全局上配置Reoptimization使能和Reoptimization Timer，对所有的TE Tunnel生效，针对个别TE Tunnel可以在接口上进行per tunnel的设置，也可以通过参数设置该TE Tunnel接口是否参与Reoptimization计算；另外一种是不需要在全局模式下使能，只在希望使能Reoptimization计算的TE Tunnels上配置Reoptimization使能和Timer。

在VRP系统上是通过在TE Tunnel接口下配置“mpls te reoptimization frequency <1-604800> Value(in Sec)”命令来支持此功能。

Inter-Area的实现

我们知道，MPLS TE需要链路状态协议（OSPF TE或者IS-IS TE）进行TE特性信息的传递，起初的都是在一个Area内部实现的。随着MPLS TE技术的不断发展和成熟，许多新的需求，比如Inter-Area的需求在RFC 4105 (Requirements for Inter-area MPLS Traffic Engineering) 和Inter-AS的需求在RFC4216 (MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements) 中被提出来并开始逐步得到部分实现和研究。目前，VRP的TE已实现Inter-Area的需求，以下做简要介绍（这里的Area指的是OSPF area or IS-IS level）。

MPLS TE (CSPF、RSVP-TE、OSPF-TE和ISIS-TE等) 被许多网络管理员所采用并且实现，提供网络流量的测量；对网络资源的使用进行优化；支持端到端的QoS保证；进行智能的链路/节点/路径保护等功能。然而，目前的MPLS TE实现主要被限制在一个单一的IGP Area里。前面我们已经详细介绍了MPLS-TE 的几个组件：

- 路由组件
- 路径计算组件



■ 信令协议组件

这种限制的根本原因是路由组件（ISIS-TE、OSPF-TE）和路径计算组件（CSPF）。在层次化拓扑中IGP引入的区域划分会影响链路状态通告的传播范围。如果Headend节点不能了解到整个网络中的所有拓扑信息，CSPF算法就无法计算出到达Tailend节点基于约束的最优路径。

由于网络的复杂性，为了提供层次化网络结构和实现一定的管理策略，几乎所有AS内部都是采用了多个Area的结构，Inter-area MPLS-TE 的实现变得非常重要。不仅仅在一个Area中需要提供TE，在Inter-Area网络结构中也要实现。同时，对于具有了多个AS域的大型运营商，实现跨AS的MPLS TE也是必须的。

RFC4105描述了完整的Inter-area需求，要求保持TE的全路径约束计算、全路径信令传递、路径保护和QoS特性等。同时提供了两种可能的TE-LSP计算方法：

- 1.以明确路径选择扩展机制为基础，LSR head-end和中间ABR通过静态配置或动态选择来决定下一个ABR，然后对每个区域独立计算域内路径。

- 2.通过ABR的相互协作，以递归的方式执行完整的路径计算。

VRP系统目前对IGP的扩展只支持Intra-Area的LSA/TLV，因此不支持全路径的计算。通过在TE Tunnel的头节点采用严格方式或混合方式的明确指定路径，我们可以实现Inter-Area的TE tunnel。下图是一个例子：

假设Area1中的R11要建立一条目地为Area2中的R22的TE Tunnel，在R11中需要使用明确指定路径来实现。这条明确指定路径可以采用完全严格的方式，从R11到R22物理路径上的每一个节点都要通过明确指定路径顺序列出；也可以采用混合的方式，从R11到R12用松散方式，从R12到R21必须用严格方式，从R21到R22的这段用松散方式。

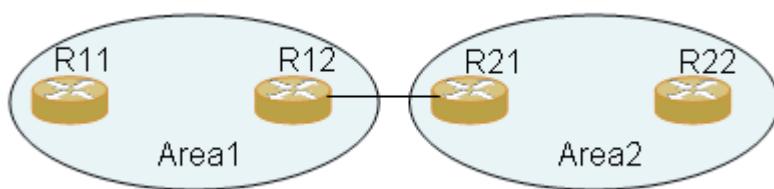
关于此问题的进一步讨论请参见RFC4105 Requirements for Inter-area MPLS Traffic Engineering）。

Load Balance特性

TE的流量负载分担（Load Balance）有三种情况，等价的TE tunnels之间的负载分担、不等价的TE Tunnels之间的负载分担和等价的TE Tunnel和IGP之间的负载分担。

第一种情况，等价的TE Tunnels之间的负载分担。CSPF算法可能会计算出多条到达目的节点等价路由，并建立多条等价的TE Tunnels；通过网络管理员指定，也可以形成多条到达目的节点的等价路由。从一条CR-LSP的Headend节点去往Tailend节点及其远端网络的流量可以通过他们之间的多条TE Tunnels进行负载分担。

第二种情况，不等价的TE Tunnels之间的负载分担。对于一条CR-LSP的Headend节点和Tailend节点之间存在多条不等价的TE Tunnels的情况，有一



种机制，在缺省情况下，流量会通过Cost值最小的那条路径转发；如果使能该机制的话，就可以根据不等价TE Tunnels的Cost值，分配不同比例的流量。

第三种情况，等价的TE Tunnel和IGP之间的负载分担不太容易理解，也不建议采用。为了方便理解，下面引用RFC3906中的例子来进行说明：



上图中所有的链路都有相同的Cost值10，假设一条TE-tunnel在rtrA 和rtrD之间（采用的是IGP Metric）开始建立。当CSPF计算时把rtrC放到临时列表中（TENTative list），会意识到rtrC不是直接连接的，因此会指定到达rtrC的下一跳（first-hop）信息到rtrC的父亲rtrB上。当rtrA上CSPF算法将rtrD从临时列表中移往路径列表（PATHS list）中时，会意识到rtrD是这条TE-tunnel的末端。因此rtrA会安装一条通过TE-tunnel接口而不是通过rtrB到达rtrD的路由。

当rtrA把rtrE放到临时列表中时，会意识到rtrE不是直接连接的并且rtrE也不是TE-tunnel的末端。因此，rtrA会指定到达rtrE的下一跳（first-hop）信息到rtrE的父亲节点（(rtrC 和 rtrD) 上。从rtrA去往rtrE的流量现在会在IGP路径rtrA->rtrB->rtrC和TE-tunnel的CR-LSP路径rtrA->rtrD（物理路径为rtrA->rtrB->rtrD）两条路径之间进行负载分担。在IGP路由和TE Tunnel都可用的情况下，可以由网络管理员根据流量策略来进行具体实现。

目前，VRP系统的TE对第一种情况，等价TE Tunnels之间的负载分担已经实现。

DS-TE

本部分内容参照RFC3564 (Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering)，对MPLS TE中应用QoS机制进行简单的描述。MPLS QoS的内容请参看RFC3270中关于MPLS Support Diff-Serv的描述。

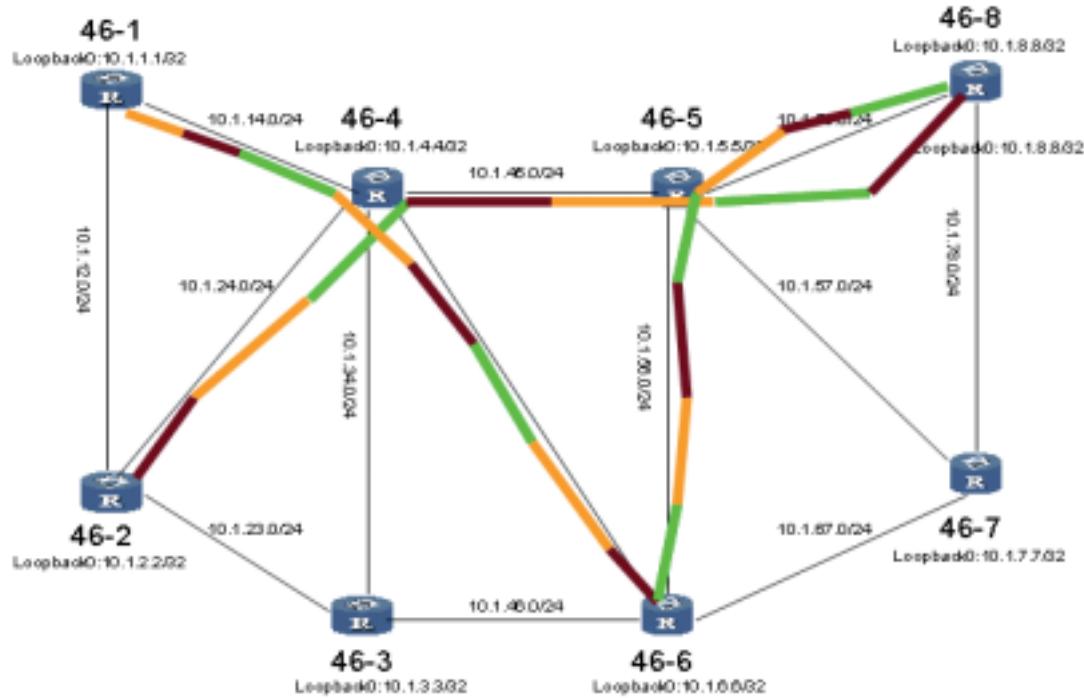
我们把在MPLS TE网络中采用区分服务模式QoS机制，把流量从Diff-Serv的分类映像到CR-LSP，针对不同流量类型，分配不同的资源进行转发的机制称为"Diff-Serv-aware Traffic Engineering(DS-TE)"。

Diff-Serv普遍被运营商采用来灵活地支持区分服务的QoS。在Diff-Serv网络中，当不需要进行网络资源优化时，MPLS TE不需要被考虑；而当需要对网络中的资源进行优化时，MPLS网络中的Diff-Serv机制就需要MPLS TE来实现。在这样的网络中，Diff-Serv和MPLS TE一起提供他们各自的优势。

在MPLS TE网络中启用QoS机制，有两种方式：基于分类方式（per-class level）和会聚方式（aggregate level）。其中对于MPLS报文中携带EXP字段的网络（如Ethernet）建议采用聚合方式（aggregate level）；对于MPLS报文中不携带EXP字段的网络（如ATM）建议采用分类方式（per-class level）。

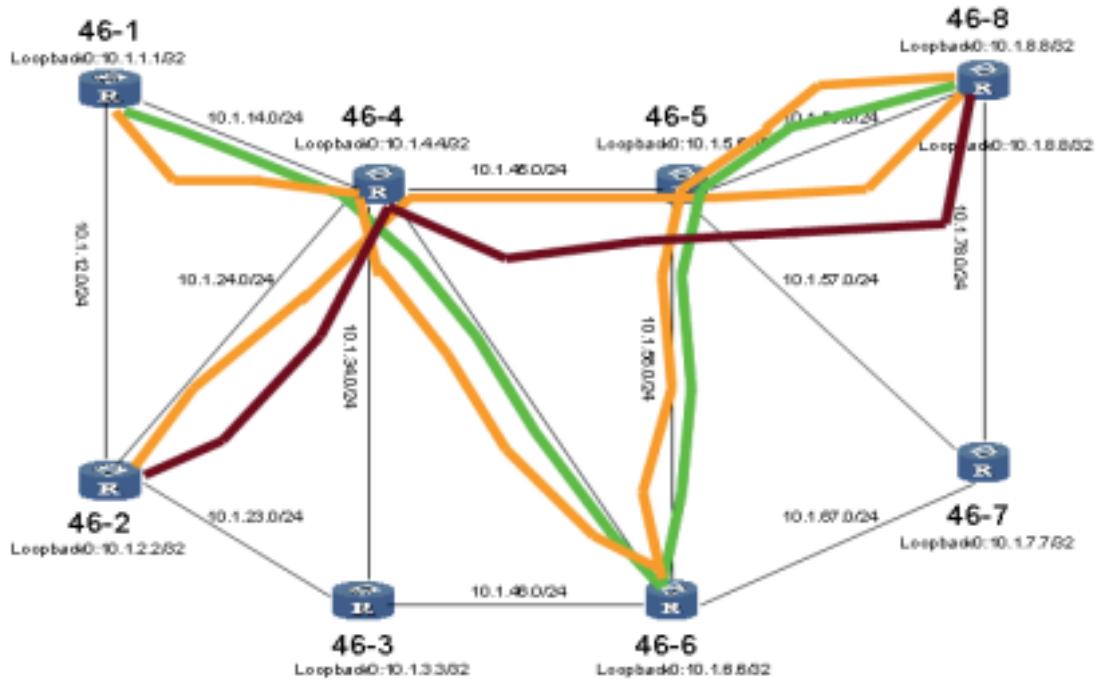
不同类别的流量从一个节点到达另外一个节点经过相同的CR-LSP，中间设备在进行转发时根据流量的类别基于EXP进行不同策略的执行，这通常被称为聚合服务模式（Aggregate Model），如下图所示：





在聚合模式QoS机制中，从TE Tunnel的入口节点（例如上图中46-2）到达TE Tunnel的出口节点（例如上图中46-8）的流量，都在一条物理链路上转发，根据流量中不同的分类执行不同的策略。

与之相对的，流量从一个节点到达另外一个节点，根据流量的类别进行区分处理。不同的类别的流量通过不同的TE Tunnel进行转发，这通常被称为区分服务QoS机制（Diff-Serv Model），如下图所示：



在区分服务模式QoS机制中，从TE Tunnel的入口节点（例如上图中46-2）到达TE Tunnel的出口节点（例如上图中46-8）的流量，根据流量的不同分类，分别通过两个TE Tunnel（例子中有两个流分类，分别在两条物理链路）进行转发。每个TE Tunnel执行不同的策略（不同的链路可用带宽、预留带宽等）。

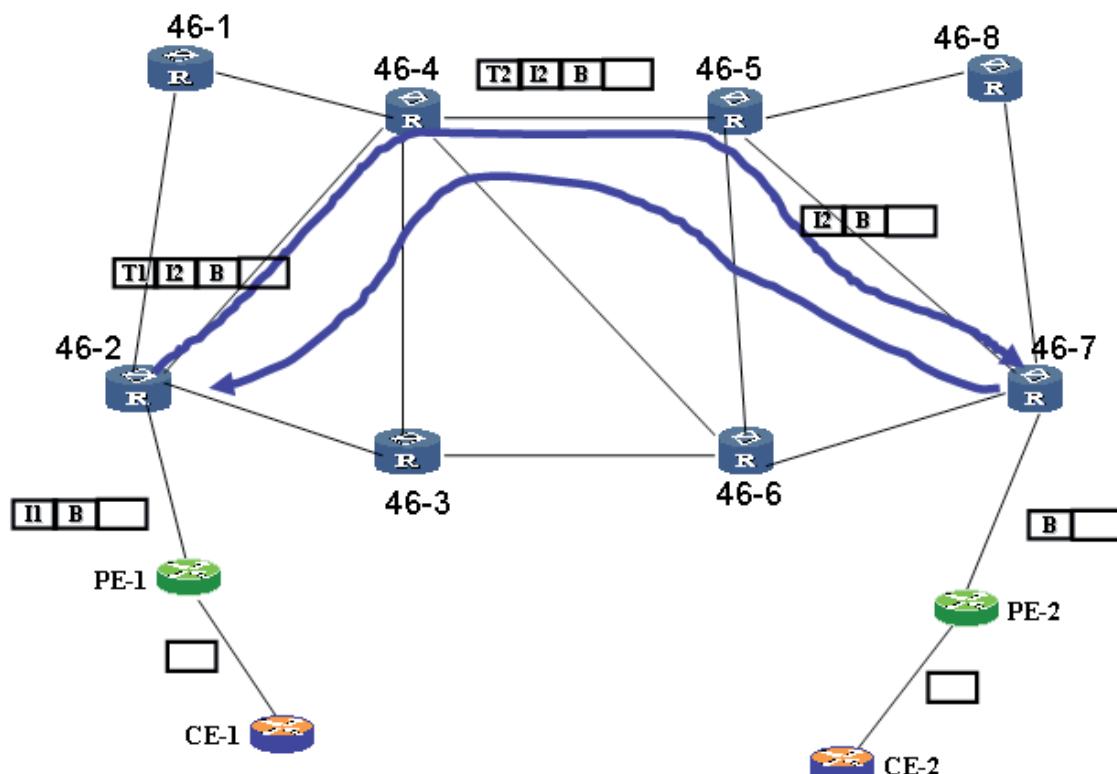
MPLS L3/L2 VPN Over TE

MPLS TE不但支持普通TE隧道，对主流的MPLS L3/L2 VPN应用也可以支持。下面通过这个示意图进行简单的描述：

假设下图中46-1, 46-2, 46-3, 46-4, 46-5, 46-6, 46-7和46-8是MPLS网络中的P设备，并且使能了TE功能，46-2与46-7之间分别建立了两条单向的TE Tunnel。网络中，PE-1下连接的CE-1和PE-2下连接的CE-2同属VPN1(L3或L2 VPN)。

下面我们来看一下从CE1发送到VPN1的另外一个站点CE-2的数据报文交换过程。

1. 普通的IP报文从CE-1到达PE-1的私网接口（连接CE1的接口）；
2. PE-1首先将vrf绑定标签（VPN1标签）PUSH到标签栈的栈底，假设标签为B；
3. PE-1把46-2通过LDP协议分配过来的标签（MPLS标签）PUSH到标签栈的栈顶，假设为I1，并且将报文发送到46-2；



4. 46-2接收到从PE-1发送过来的报文，发现I1对应的路由出标签I2来自46-7，是通过LDP协议分配的（注意，TE Tunnel接口被看作46-2和46-7的直连接口，由46-7通过LDP协议分配标签）。因此交换栈定的标签为I2；

5. 46-2发现到达下一跳46-7的接口是TE Tunnel接口，因此，将46-4分配过来的TE Tunnel标签（通过RSVP-TE或者CR-LDP或者静态LSP协议分配），假设是T1，PUSH到标签栈的栈顶，并且将报文发送到TE Tunnel的下一跳46-4（注意这时候，标签栈里有三层标签，由里向外分别为VPN1标签、MPLS标签和TE Tunnel标签）；

6. 46-4接收到从46-2发送过来的标签，发现到达TE Tunnel的下一跳46-5的TE Tunnel标签是T2，因此POP栈顶的标签T1，PUSH 标签T2，并且将报文发送到TE Tunnel的下一跳46-5（这时，标签栈有三层标签，由里向外分别为VPN1标签、MPLS标签和TE Tunnel标签）；

7. 46-5接收到从46-4发送过来的标签，发现到达TE Tunnel的下一跳46-7的TE Tunnel标签是Null，知道自己是倒数第二跳，因此POP栈顶的标签T2，并且将报文发送到TE Tunnel的末端46-7（这时，标签栈有两层标签，由里向外分别为VPN1标签和MPLS标签）；

8. 46-7接收到从46-5发送过来的标签，发现到达下一跳PE-2的MPLS标签是Null，知道自己是倒数第二跳，因此POP栈顶的标签I2，并且将报文发送到MPLS域的边界路由器PE-2（这时，标签栈有一层标签，为VPN1标签）；

9. PE-2接收到从46-7发送过来的标签，发现到达VPN1的下一跳CE-2的VPN1标签，因此POP栈顶的标签B，经过比较，将报文发送到私网接口（连接CE-2的接口），到达CE-2（这时，标签栈没有标签，转发的是普通IP报文）。

MPLS TE技术重点协议 及相关报文描述

包贵新



OSPF TE

协议及报文字段描述

对于标准的OSPF V2协议，需要对其进行扩展来承载实现TE所需要的各种信息，包括带宽信息和管理属性等。

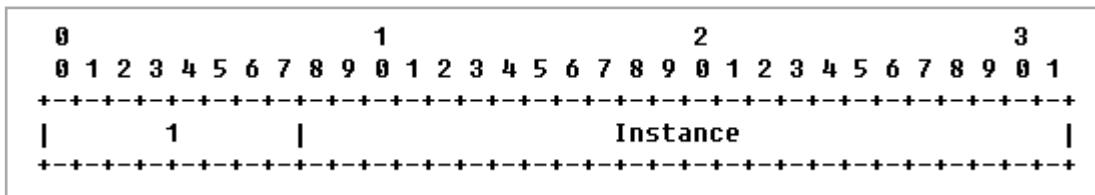
■ LSA type

OSPF基于Opaque LSA来提供TE所需的扩展，共有Type 9、10、11三种类型的Opaque LSAs，每种类型有不同的泛洪范围。

TE使用Type 10 Opaque LSA，它的泛洪范围为Area内部，在type 10 Opaque LSA的基础上定义了一个新的LSA：Traffic Engineer LSA。

■ LSA ID

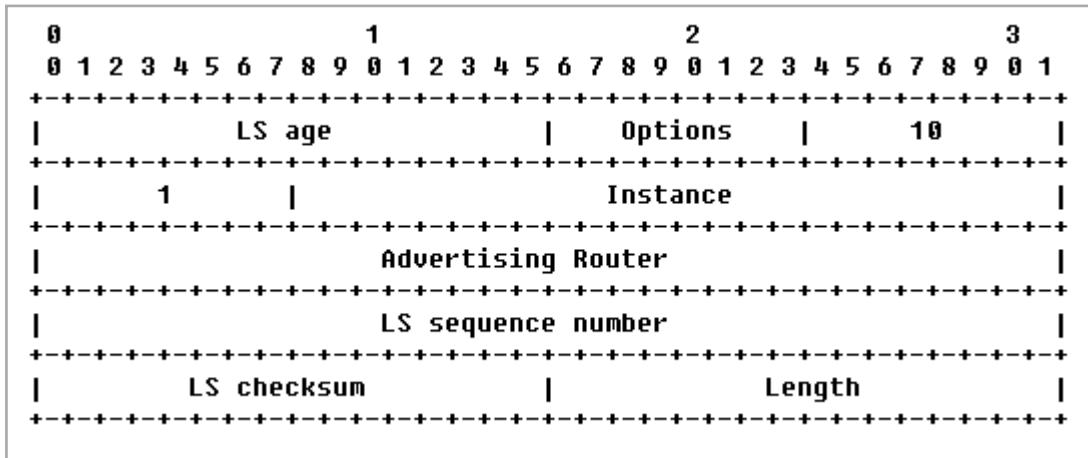
Opaque LSA中的LSA ID包含8 bits type数据和24 bits type-specific数据，Traffic Engineer LSA使用Type 1，接下来的24 bits称为Instance field。LSA ID没有任何拓扑意义，Instance field可以是任意值，用于指示多个TE LSAs，最多可以有16777216个TE LSAs从一个系统中向外发布出去。



■ LSA格式描述

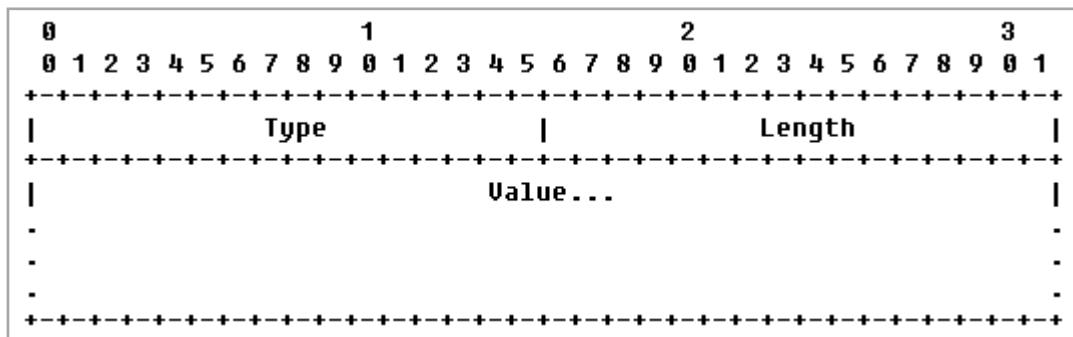
TE LSA是由一个标准的LSA Header开始，后续为PayLoad部分，是由一个或多个TLV（Type/Length/Value）构成。

✓ LSA Header



✓ TLV

LSA的payload由一个或者多个Type/Length/Value (TLV) 构成。每一个TLV的格式如下：



Length以byte为单位表示Value部分的长度。(如果一个TLV没有value部分，那么长度则为0)。

一个LSA包含一个top-level TLV。共定义了两个top-level TLVs分别为

1 - Router Address

2 - Link

Router Address TLV 指定了一个向外发布信息使用的IP地址（一般为loopback地址），如果IS-IS也被配置可用，这个地址也可以被用来比较OSPF与IS-IS拓扑。比如，一台路由器R通过OSPF和IS-IS向外发布TE LSA，假设网络中另一台路由器S通过OSPF和IS-IS都分别建立TED，路由器R会作为两个节点出现在路由器S的TED中，但是通过相同的Router Address, S可以判定收到的OSPF TE LSA 和 IS-IS TE LSA确实是从一

台路由器发布的。Router address TLV 类型为1，长度为4并且有一个4字节长的IP address。

Link TLV 描述单条链路，由一系列sub-TLVs组成，各个sub-TLV之间没有前后顺序关系。一个Link TLV只能被一个LSA所承载。Link TLV的类型为2，长度为不定长，共有以下几种sub-TLVs 被定义：

- 1 - Link type (1 octet)
- 2 - Link ID (4 octets)
- 3 - Local interface IP address (4 octets)
- 4 - Remote interface IP address (4 octets)
- 5 - Traffic engineering metric (4 octets)
- 6 - Maximum bandwidth (4 octets)
- 7 - Maximum reservable bandwidth (4 octets)
- 8 - Unreserved bandwidth (32 octets)
- 9 - Administrative group (4 octets)

对于以上定义的9种sub-TLVs，其中Link Type和Link ID sub-TLVs 强制必须出现一次，其他sub-TLVs 最多出现一次。

Link Type sub-TLV 定义链路类型，1表示Point-to-point链路；2表示Multi-access链路，1个字节长度；

Link ID sub-TLV 指出对端链路，对于point-to-point链路，为对端邻居的Router ID；对于multi-access 链路，为DR (designated router) 的接口地址。Link ID sub-TLV 为TLV 类型2，4个字节长度；

Local Interface IP Address sub-TLV指出本链路的接口IP address(es)，如果有多个地址，所有地址都会在这个sub-TLV给出。Local Interface IP Address sub-TLV 为TLV 类型3，长度为4N个字节 (N是本地接口地址的个数)；

Remote Interface IP Address sub-TLV 指出本链路对端邻居的接口地址。如果链路类型为Mul-

ti-access，则Remote Interface IP Address 被设置为0.0.0.0。Remote Interface IP Address sub-TLV 为TLV type 4，长度为4N个字节 (N是邻居接口地址的个数)；

Traffic Engineering Metric sub-TLV 定义TE计算使用的链路metric值。这个metric与标准的OSPF链路metric不同，通常是网络管理员设置。Traffic Engineering Metric sub-TLV 为 TLV 类型 5，4个字节长度；

Maximum Bandwidth sub-TLV 指出单方向的（从本机去往邻居方向）链路最大可用带宽，表明链路的真实容量，单位为bytes per second。Maximum Bandwidth sub-TLV 为TLV 类型 6，4个字节长度；

Maximum Reservable Bandwidth sub-TLV 指出单方向的（从本机去往邻居方向）本链路可以被预留的最大带宽。注意，可以比最大带宽大（表明可以过载），不过需要用户配置，缺省为最大带宽的值，单位为bytes per second。Maximum Reservable Bandwidth sub-TLV 为 TLV 类型 7，4个字节长度；

Unreserved Bandwidth sub-TLV 指出8个优先级中每个优先级没有被预留的带宽。这个带宽值可以被setup优先级从0到7的TE Tunnel所预留，每个值可以小于或等于Maximum Reservable Bandwidth，单位是bytes per second。Unreserved Bandwidth sub-TLV 为 TLV 类型 8，32个字节长度。

Administrative Group sub-TLV 包含了一个管理员设置的4个字节长度的掩码，每一组对应一个接口上设置的administrative group。一条链路可以属于多个multiple groups，通常最不重要的看作'group 0'，最重要的看作'group 31'，Administrative Group 也被称为Resource Class/Color。Administrative Group sub-TLV 为 TLV 类型 9，4个字节长度。

具体更细的描述请参见RFC3630 (TE Extensions to OSPF Version 2)。

IS-IS TE

协议及报文字段描述

为了满足流量工程的需要，IS-IS协议通过扩展的IS可达TLV（Extended IS Reachability TLV）、扩展的IP可达TLV（Extended IP Reachability TLV）和Traffic Engineering RouterID TLV来承载流量工程需要的信息。

扩展的IS可达TLV

扩展后的IS可达TLV的类型为22，是基于先前的IS可达TLV扩展而来。原先的IS可达（TLV type 2，在ISO 10589中被定义）包含一系列邻居的信息，对于每一个邻居，都有一个数据结构含有缺省的metric、delay、cost、reliability和7个字节长度的邻居的ID。

扩展的TLS Type 22包含有一个新的数据结构，包括：7个字节的system Id 和pseudo node number；3个字节的缺省metric；1个字节的sub-TLVs的Length；0-244字节的sub-TLVs。而每一个sub-TLV由如下部分组成：1个字节的sub-type；1个字节的sub-TLV中Value的Length；0-242个字节的Value。这样，如果没有sub-TLVs被使用，这种新的编码需要11个字节的长度，可以保持最多23个邻居的信息。要注意，这种新的编码允许255个字节长的sub-TLVs，实际的最大值应该是255减去11，244个字节。

为了防止在TE的CSPF计算中发生过载的情况，所有大于或等于MAX_PATH_METRIC的链路应该被看作metric等于MAX_PATH_ME-

TRIC，MAX_PATH_METRIC的值为4,261,412,864 (0xFE000000, $2^{32} - 2^{25}$)。如果一条链路有最大的链路metric值($2^{24} - 1$)，那么这条链路不能参与普通的SPF计算，这个特性使得一些链路可以只用于TE路径计算，但是不参与普通SPF计算，即不用于逐跳路由。

扩展的sub-TLVs如下：

Sub-TLV type	Length (octets)	Name
3	4	Administrative group (color)
6	4	IPv4 interface address
8	4	IPv4 neighbor address
9	4	Maximum link bandwidth
10	4	Reservable link bandwidth
11	32	Unreserved bandwidth
18	3	TE Default metric
250-254		Reserved for cisco specific extensions
255		Reserved for future expansion

■ Sub-TLV 3 : Administrative group (color, resource class)

Administrative group sub-TLV 包含4个字节的掩码（由网络管理员设置）。每个bit对应接口的一个administrative group。按约定，最低位对应的称作'group 0'；最高位对应的称作'group 31'。这个sub-TLV 是可选的，在每个扩展的TLV中应该只出现一次。

■ Sub-TLV 6: : IPv4 interface address

这个sub-TLV 包含4个字节的本接口的IPv4 address。在实现上，一定不能把一个/32 prefix的接



口地址注入到路由表和转发表中，因为当与不支持本sub-TLV的设备交互时可能产生转发环路。

■ Sub-TLV 8 : IPv4 neighbor address

这个sub-TLV 包含一个链路对端邻居的IPv4 address。在实现上，一定不能把一个/32 prefix的邻居地址注入到路由表和转发表中，因为当与不支持本sub-TLV的设备交互时可能产生转发环路。

■ Sub-TLV 9 : Maximum link bandwidth

这个sub-TLV指出单方向的（从本机去往邻居方向）链路最大可用带宽，表明链路的真实容量，单位为bytes per second。该项是可选的，最多可以出现一次。

■ Sub-TLV 10 : Maximum reservable link bandwidth

这个sub-TLV指出单方向的（从本机去往邻居方向）本链路可以被预留的最大带宽。注意，可以比Maximum link bandwidth大（表明可以过载），不过需要用户配置，缺省为最大带宽的值，单位为bytes per second。该项是可选的，最多可以出现一次。

■ Sub-TLV 11 : Unreserved bandwidth

这个sub-TLV指出本链路8个优先级中每个优先级单方向（从本机去往邻居方向）没有被预留的带宽。这个带宽值可以被“setup priority”为0到7的TE Tunnel所预留，每个值可以小于或等于Maximum Reservable Bandwidth，单位是bytes per second。由于稳定性的原因，迅速多次改变这个值不应该产生LSPs。该项是可选的，最多可以出现一次。

■ Sub-TLV 18: : Traffic Engineering Default Metric

这个sub-TLV 包含一个24 bit 的整数。这个metric是由管理员设置，并且可以通过配置指定参与CSPF计算。这个sub-TLV 是可选的，在扩展IS可达TLV中最多出现一次。如果没有这个选项，CSPF计算必须使用存在于IS可达TLV中固定部分缺省的链路metric。

扩展的IP可达TLV

扩展的IP可达TLV为TLV类型135，是在原来的IP可达TLVs (TLV 类型 128 和TLV 类型130)基础上扩展而来。为了克服原来的IP可达TLVs的限制，扩展的IP可达TLV提供了一个32 bit 的metric和增加了1 bit 指示标记，表示网络前缀已经分布到本Level中。扩展的IP可达TLV定义了一个新的数据结构，包括：4个字节的 metric 信息；1个字节的控制信息（1 bit up/down、1 bit的sub-TLVs存在指示和6 bit的前缀长度）；0-4 个字节 IPv4 前缀；0-250个字节可选sub-TLVs（其中，1个字节表示sub-TLVs的Length、0-249个字节的sub-TLVs），而每个sub-TLV由1个字节的 sub-type、1个字节的本sub-TLV的Length和0-247个字节的Value组成。

Traffic Engineering router ID TLV

Traffic Engineering router ID TLV的类型为134。Router ID TLV 包含4个字节的router ID，指出LSP 入口路由器的地址，一般采用loopback。对于TE来说，这是很有用的，因为保证了一个稳定的IP地址，即使是节点某个物理接口故障的情况下。

具体更细的描述请参见RFC3784 (Intermediate System to Intermediate System (IS-IS)Extensions for Traffic Engineering (TE))。

RSVP-TE

协议描述

对于扩展的RSVP，主要有如下重点需要了解的内容：

■ LSP Tunnels 和 TE Tunnels

一旦一条LSP建立，流量就会在LSP的入口节点根据分配好的标签通过这条LSP进行转发，这种流量和标签间的对应可以使用不同的标准，对应到相同标签的报文的集合称作forwarding equivalence class (FEC)。标签与FEC的对应使流量从LSP的入口节点经过一系列中间节点以不透明的模式转发到LSP的出口节点，因此LSP也被称为“LSP Tunnel”。协议标准中定义了新的RSVP SESSION、SENDER_TEMPLATE和FILTER_SPEC对象，称为LSP_TUNNEL_IPv4 和LSP_TUNNEL_IPv6来支持LSP Tunnel特性。通俗说，“LSP Tunnel”就是通过一条LSP，流量以Hop- by-Hop的方式进行标签转发。

一些应用需要多条LSP tunnels，特别是在重路由操作的时候需要一条流量干线可以通过多条路径。这样的多个LSP tunnels我们称之为traffic engineered tunnels (TE tunnels)。为了对TE tunnel进行标识，定义了两个标志：一个Tunnel ID（在SESSION对象中），Tunnel ID唯一的指出了一条TE Tunnel。SENDER_TEMPLATE 和 FILTER_SPEC 对象中定义了一个LSP ID。Tunnel ID和LSP ID的组合唯一标识了一条TE Tunnel。

■ 扩展RSVP的LSP Tunnels操作

扩展RSVP在LSP Tunnel方面有如下能力：

- ✓ 在有Qos需求和没有Qos需求的情况下，建立LSP tunnels

- ✓ 动态重路由时建立新的LSP tunnel
- ✓ 对于建立的LSP tunnel的路由监控
- ✓ LSP tunnels诊断
- ✓ 通过管理策略控制LSP tunnels之间的抢占
- ✓ 执行downstream-on-demand 模式的标签请求、分配和绑定

为了建立一条LSP tunnel，路径的第一个MPLS节点（入口节点）创建一个RSVP Path消息，使用的session type 是 LSP_TUNNEL_IPv4或者LSP_TUNNEL_IPv6。RSVP Path消息中携带了LABEL_REQUEST对象，LABEL_REQUEST 指出了一个标签绑定的请求并且指出了沿着这条路径的网络层协议。如果发送节点为了充分使用网络资源需要明确指定流量转发路径，那么发送节点就需要知道该路径上节点的路由，通过RSVP Path消息中增加携带EXPLICIT_ROUTE 对象实现，EXPLICIT_ROUTE 对象指出了要经过路径上的节点。Path 消息沿着ERO中指定的路径转发，路径上的每个节点都记录ERO信息，也可以修改ERO之后进行转发，节点上ERO的信息存放在缓冲区中。

如果一个LSP tunnel已经成功建立后，发送节点又发现了一条更优的路由，发送节点可以动态的重路由到这条更优的路径上。通过改变EXPLICIT_ROUTE 对象可以实现动态重路由，向RSVP Path增加一个RECORD_ROUTE对象，用于记录LSP tunnel实际经过的每一个节点，RECORD_ROUTE 对象类似于一个Path Vector，因此也可以实现Loop Detection。

RSVP Path消息中通过增加SESSION_ATTRIBUTE 对象可以进行Session 标识和诊断。一些控制信息，例如setup 和 hold 优先级、亲和度、本地保护等都包含在这个对象里。通过使用setup和hold优



先级和SENDER_TSPEC 和POLICY_DATA 对象（都在Path消息中），可以进行策略控制。

LABEL_REQUEST对象要求中间节点提供标签绑定，如果中间某个节点不支持标签绑定功能，就会向入口节点发送一个PathErr 消息，携带"unknown object class" 错误信息。

LSP的目的节点（出口节点）在RSVP Resv消息中携带LABEL对象对Path消息的LABEL_REQUEST 进行响应，Resv 消息沿着Path消息经过路径的upstream方向逐跳返回到入口节点，如果path消息使用了ERO，那么Resv消息会沿着ERO相反的方向转发。

节点收到Resv消息后，将LABEL对象中的标签作为LSP tunnel 转发流量的标签。如果不是入口节点，会分配一个新的标签重新设置在Resv消息的LABEL对象，继续沿着upstream方向发送到他的上一跳PHOP。LABEL对象新设置的标签表示从LSP tunnel入流量的标签。这个节点可以更新他的"Incoming Label Map" (ILM)了（被用作映射入标签的报文到"Next Hop Label Forwarding Entry" (NHLFE)）。当Resv 消息逐跳向上转发到入口节点，一条LSP就有效的建立起来。

■ 服务类 Service Classes

对于可以预留的Integrated Service类型进行限制，至少应该支持Controlled-Load service和Null Service。

■ 预留类型 Reservation Styles

出口节点可以从几种可用的预留类型中选择一个，并且每个RSVP session必须有一种类型，入口节点对于预留类型的选择没有影响，出口节点可以为不同的LSPs选择不同的预留类型。根据不同的预留类型，一个RSVP session可以产生一个或者多个。类型FF为一个单独的入口节点进行单独预留带宽，其他类型，WF 和 SE，可以为几个不同的入口

节点一起预留带宽，几种预留方式各有优缺点。

✓ Fixed Filter (FF) Style

Fixed Filter (FF)预留方式为每个入口节点单独预留，不能被其它入口节点所共享。一般应用在从同一个入口节点同时发出多个单独的请求时，使用FF方式在一条链路上预留带宽的总和等于为每一个入口节点预留带宽的总和。因为每个入口节点都有自己的预留带宽，因此每一个入口节点都有自己单独的标签。

✓ Wildcard Filter (WF) Style

使用Wildcard Filter (WF) 预留方式，一个共享的预留带宽可以被所有入口节点到同一个出口节点使用。在一条链路上预留带宽不管有多少个出口节点都保持相同。一个单个的multipoint-to-point LSP为相同session的所有入口节点所使用，在这条链路上，多个入口节点的流量采用一个相同的标签。这种方式适用于并不是所有的入口节点同时发送流量，例如电话会议就是所有的参与者不是在同一个时刻说话。但是，如果所有的入口节点同时发送，没有一种机制可以保证每个入口节点的预留带宽。这是WF这种预留方式对于TE应用的一个限制，此外EXPLICIT_ROUTE对象也不能用于WF这种模式，这是这样，WF这种预留模式没有被TE所采用。

✓ Shared Explicit (SE) Style

Shared Explicit (SE) 预留方式允许出口节点明确指定为哪一个或哪几个入口节点进行预留，通过一个列表记录一条链路为那些入口节点进行了预留。因为每个入口节点被明确包含在Resv 消息中，不同的标签被分配到不同的入口节点，分别建立不同的LSPs。SE预留模式可以提供multipoint-to-point LSP或者LSP per 入口节点。Multipoint-to-point LSPs 用在当path消息中没有携带EXPLICIT_ROUTE对象的情况或者Path消息中含有相同EXPLICIT_ROUTE对象的情况下，这时，会创建一个标签；当从不同的入口节点发送的Path消息中携带各自的ERO的

情况下，就会为每个EXPLICIT_ROUTE对象的记录创建不同的LSP tunnel。

■ TE Tunnels的重路由

TE的一个很重要的能力就是对已经建立好的TE tunnel进行重路由。当一个更优的TE tunnel或者现有TE tunnel的某个节点或链路故障，会用到重路由；当原TE tunnel恢复正常，从新的TE tunnel重新切换回到原来的TE tunnel上也会使用到重路由。

在进行重路由过程中最关心的就是如何减少流量的丢失。这种平滑切换需要在旧的LSP失效之前建立一条新的LSP，这个概念被称作“make-before-break”。切换的过程会发生一个问题，就是旧的LSP和新的LSP会对共同经过的链路产生竞争关系，从而会导致新的LSP因资源不足而无法建立。

使用RSVP的一个优势就是真正实现了“make-before-break”机制，在新LSP和旧LSP共同经过的链路上，在流量没有切换到新的LSP之前，旧LSP的资源不被释放并且新LSP的预留不被计算在内。RSVP的LSP_TUNNEL SESSION对象使用SE预留方式很自然的解决的这个问题，就是旧的LSP和新的LSP在共同的链路上共享资源。

在重路由或者带宽增加的操作中，TE tunnel的入口节点对于RSVP session需要作为两个不同的发送节点，通过包含在SENDER_TEMPLATE 和 FILTER_SPEC 对象中的“LSP ID”来完成。为了使重路由生效，入口节点使用了一个新的LSP ID 并且生成了一个新的SENDER_TEMPLATE，然后创建一个新的ERO来定义新的路径。这样，入口节点开始使用原来的SESSION对象、新的SENDER_TEMPLATE和ERO来发送一个新的Path消息；同时仍然使用旧的LSP，使用旧的Path消息进行刷新。在与旧的LSP的共同链路上，新的Path消息被看作一个普通的新的LSP Tunnel建立消息，在共同的链路上，共享的SESSION 对象和SE预留类型允许新的LSP共享使用旧的LSP资源正常建立，当入口节点

受到了一个新的LSP的Resv 消息，就可以将流量迅速切换过去，同时切断旧的LSP。为了使增加带宽生效，一个新的Path消息携带新的LSP_ID被使用去尝试预留一个大的带宽，即使失败，当前的LSP_ID继续刷新也会保证不会导致流量丢失。

■ Path MTU

标准的RSVP和Int-Serv提供RSVP发起端到接收端这条路径上最小的可用MTU，称为Path MTU，Path MTU 自动识别能力也可以在LSP建立时由RSVP提供。Path MTU信息在Integrated Services 或者 Null Service 对象中被承载。

通过标准的RSVP，Path MTU信息在发送端检查哪个IP报文超过了Path MTU，对于超过的报文，发送端对报文进行分片或者当IP报文中的“Don't Fragment” (DF) 位被置位的情况下，发送一个ICMP “destination unreachable message”。

目前的RFC标准中，扩展的RSVP对于建立的LSP Tunnel，只支持单播LSP-Tunnels (unicast LSP-tunnels)，不支持多播LSP-Tunnels (Multicast LSP-tunnels)；只支持单向的LSP-Tunnels (unidirectional LSP-tunnels)，不支持双向的LSP-Tunnels (Bidirectional LSP-tunnels)。

报文格式介绍

5种新扩展的对象：

Object name	Applicable RSUP messages
LABEL_TEQUEST	Path
LABEL	Resv
EXPLICIT_ROUTE	Path
RECORD_ROUTE	Path,Resv
SESSION_ATTRIBUTE	Path



■ 扩展的Path消息

扩展的Path消息的格式：

```

<Path Message> ::=      <Common Header> [ <INTEGRITY> ]
<SESSION> <RSUP_HOP>
<TIME_VALUES>
[ <EXPLICIT_ROUTE> ]
<LABEL_REQUEST>
[ <SESSION_ATTRIBUTE> ]
[ <POLICY_DATA> ... ]
<sender descriptor>

<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
[ <ADSPEC> ]
[ <RECORD_ROUTE> ]

```

■ 扩展的Resv消息

扩展的Resv消息的格式：

```

<Resv Message> ::=      <Common Header> [ <INTEGRITY> ]
<SESSION> <RSUP_HOP>
<TIME_VALUES>
[ <RESV_CONFIRM> ] [ <SCOPE> ]
[ <POLICY_DATA> ... ]
<STYLE> <flow descriptor list>

<flow descriptor list> ::= <FF flow descriptor list>
| <SE flow descriptor>

<FF flow descriptor list> ::= <FLOWSPEC> <FILTER_SPEC>
<LABEL> [ <RECORD_ROUTE> ]
| <FF flow descriptor list>
<FF flow descriptor>

<FF flow descriptor> ::= [ <FLOWSPEC> ] <FILTER_SPEC> <LABEL>
[ <RECORD_ROUTE> ]

<SE flow descriptor> ::= <FLOWSPEC> <SE filter spec list>

<SE filter spec list> ::= <SE filter spec>
| <SE filter spec list> <SE filter spec>

<SE filter spec> ::=      <FILTER_SPEC> <LABEL> [ <RECORD_ROUTE> ]

Note: LABEL and RECORD_ROUTE (if present), are bound to the
preceding FILTER_SPEC. No more than one LABEL and/or
RECORD_ROUTE may follow each FILTER_SPEC.

```

其中，扩展的PATH和RSVP消息的Common Header格式使用RFC2205 (Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification) 中的定义：

对于RSVP扩展LSP Tunnel 相关的各个对象的结构和更详细的解释请参考RFC3209 (RSVP-TE: Extensions to RSVP for LSP Tunnels) 和RFC3210 (Applicability Statement for Extensions to RSVP for LSP-Tunnels)。

0	1	2	3
Version (4 bits)	Flags (4 bits)	Message type (8 bits)	RSVP checksum (16 bits)
Send TTL (8 bits)	Reserved (8 bits)	RSVP length (16 bits)	
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0 1

CR-LDP

协议描述

对于扩展的CR-LDP，主要有如下需要了解的内容：

■ Strict 和 Loose Explicit Routes

一个明确路由存在于一个Label Request 消息里，是约束路由经过的路由器节点的集合，CR-LSP采用明确路由列表里的所有节点或者部分节点建立路径。约束路由被编码成一系列ER-Hops存放在约束路由的TLV中，每一个ER-Hop可能是一条约束路由中的一组节点。一条基于约束的路由按照TLV中的节点顺序形成了一条路径。Strict Explicit Routes中指出了从入口节点到出口节点路径中的Hop-by-Hop的每一跳节点，而Loose Explicit Routes中指出了从入口节点到出口节点路径中的部分节点。

■ 流量特性

一条路径的流量特性保存在流量参数TLV中，包括最大流量、承诺流量和服务粒度等。最大流量和承诺流量表示路径中可以提供的流量峰值和保证通过的流量大小，而服务粒度表示CR-LDP MPLS域这条路径可以提供的的延时变化等服务考虑因素。

■ 抢占机制

CR-LDP 通告一条路径中每一个节点的资源信息，如果新建一条CR-LSP，但是没有发现一条资源充足的路径，就会对当前的路径进行重路由来给一条新的路径进行资源分配，这个过程就称为路径抢占。Setup 和 holding优先级用来指示新建路径和当前已经建好的路径之间的抢占关系。新建CR-LSP的Setup优先级和当前已经建好的CR-LSP的holding优先级属性之间进行比较，如果新建CR-LSP的Setup优先级高于已经建好的CR-LSP的holding优先



级，则抢占旧的CR-LSP的资源。

对于setup 和 holding 优先级的分配可以根据一个网络的流量策略决定。Setup 和 Holding 优先级取值范围是从0到7，0表示最重要，也就是最高的优先级；7表示最不重要，也就是最低优先级。缺省优先级可以根据不同的实现来定义，建议使用4。

（我们公司的缺省为7）。一条CR-LSP的Setup优先级不能高于他的holding 优先级，因为有可能会抢占同样级别的其他LSP，也会被同样级别的其他LSP抢占。

■ Route Pinning

路由固定是对采用Loose Explicit方式路由的一个应用，也就是说对于“L”位置位的节点生效。一条使能路由固定功能的CR-LSP即使在Loosely路由时发现更优的下一跳，也不会改变当前正在使用的LSP。

■ Resource Class

网络管理员可以用许多不同的方式对网络资源进行分类。这些分类被称为“colors”或者“administrative groups”。当一条CR-LSP被建立的时候，需要指出这条CR-LSP的资源类别。

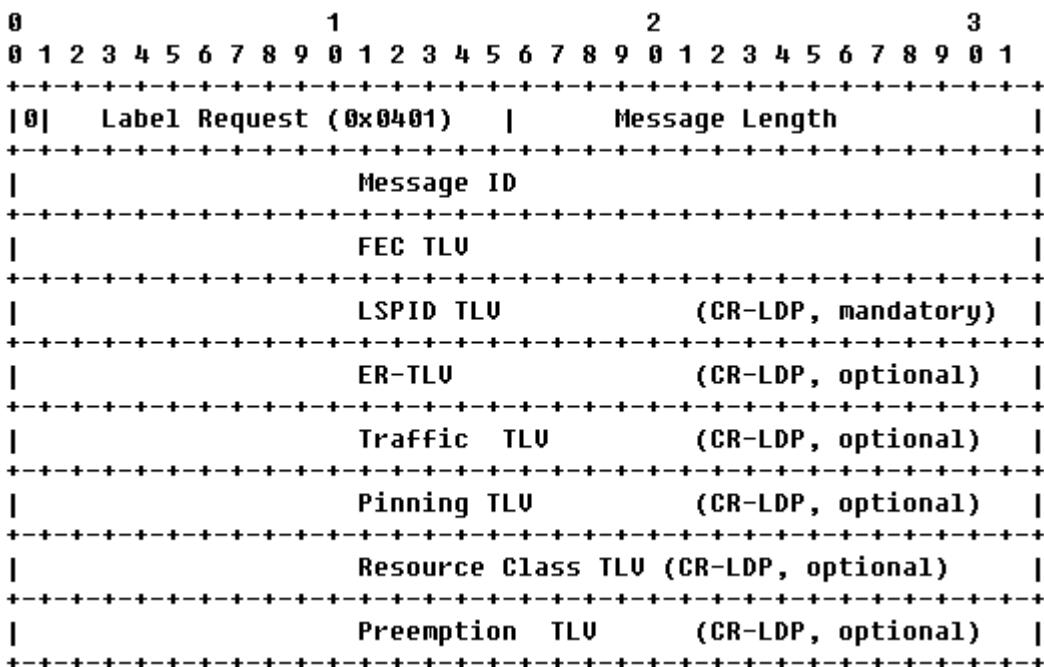
目前的RFC标准中只支持单向的点到点CR-LSPs (unidirectional point-to-point CR-LSPs)，对于点到多点 (point-to-multipoint) 和多点到点 (multipoint-to-point) 方式的CR-LSP将在未来进行研究；只支持单向的 (unidirectional) LSP建立，不支持双向的 (Bi-directional) LSP建立；只支持每个LSP分配一个唯一的标签，不支持每个LSP分配多个标签。对于CR-LDP，需要只支持DOD Ordered (Downstream On Demand Ordered) 标签分配模式和保守标签保持模式 (conservative Label Retention Mode)。



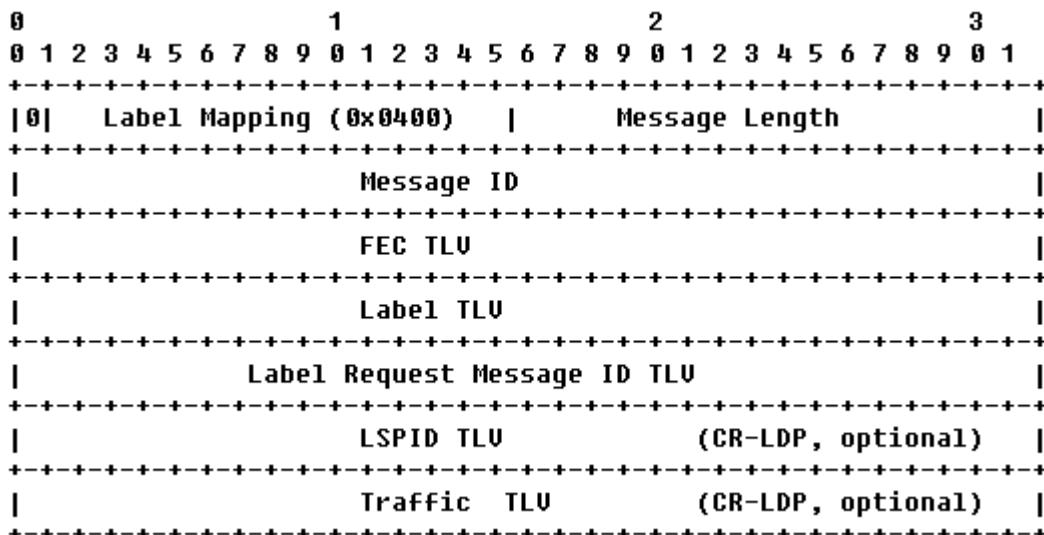
报文格式介绍

三种最主要的消息格式

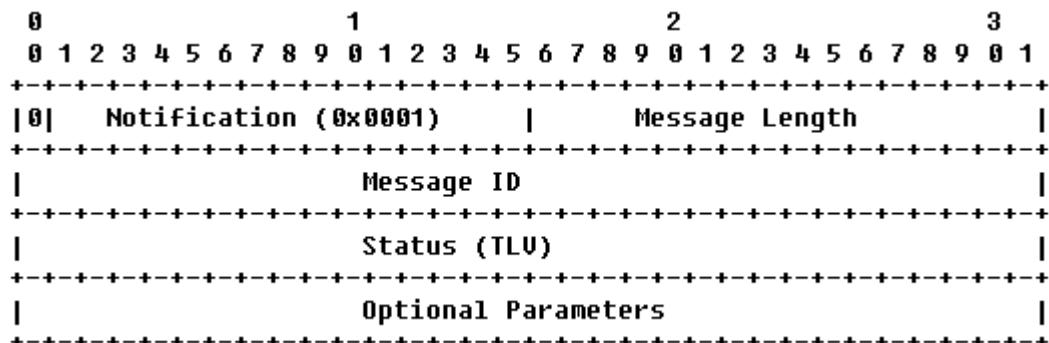
■ 扩展后的标签请求消息 (Label Request Message) 格式



■ 扩展后的标签映射消息 (Label Mapping Message) 格式



■ 扩展后的标签映射消息 (Label Mapping Message) 格式



对于每种扩展消息的相关的各个字段更详细的解释请参考RFC3212 (Constraint-Based LSP Setup using LDP) 、RFC3213 (Applicability Statement for CR-LDP) 和RFC3214 (LSP Modification Using CR-LDP) 。

DS-TE

协议及报文字段描述

为了便于理解，先介绍几个定义：

Behavior Aggregate (BA)：通过一条链路向同一个方向的具有相同(Diff-Serv)特征的报文的集合

Per-Hop-Behavior (PHB)：对于一个Diff-Serv BA应用的外部转发行为

PHB Scheduling Class (PSC)：一个属于相同流的预留分类的PHB组

Ordered Aggregate (OA)：一系列按顺序排列得 Bas，应用在这些Bas的一系列PHBs组成了一个PSC

Traffic Aggregate (TA)：具有相同PHB的报文的集合，通常是在一个DS域或者子域内。

Per-Domain Behavior (PDB)：一组可以被识别的报文以"edge-to-edge"的形式通过一个DS域所接

受的行为。一个特殊的PHB或者一组PHBs都是发 生在一个PDB。

Traffic Trunk：在同一条LSP中聚合的相同类别的流量，相同类别的流量是指从同一个FEC的流量（从DS-TE的角度应该相同对待）。

下面简单介绍一下DS-TE相关的一些内容：

■ DS-TE 兼容性

因为DS-TE可能会影响扩展性和可操作性，因此DS-TE在TE机制和Diff-Serv两者都无法解决网络设计的情况下被使用。通过DS-TE更容易使一个大网络的某个部分受益，因此一些网络管理员更愿意在他们的网络中的一个子网中来实施DS-TE。DS-TE方案必须以下面的两种方式之一来进行实施：

- 对实施的TE没有影响；2. 在一定范围内实施DS-TE（网络的一部分内或者流量类别不多，否则不好

控制)

■ Class-Types

对于DS-TE的基本要求是能够对于不同约束的流量干线 (Traffic Trunk) 分配不同的带宽。Class-Type (CT)就是经过一条链路的一些基于约束分配了不同带宽的流量干线, CT 被用作链路带宽分配、约束路由何管理控制。一个指定的流量干线在所有的链路上都是属于同一个CT。DS-TE 必须支持最多8个CTs, 被称为CT “x” ($0 \leq x \leq \text{MaxCT}-1 = 7$)。DS-TE必须可以为每一个CT分配不同的约束带宽。DS-TE必须最小支持2 CTs, 可以支持到8个 CTs。 (我们公司和Cisco公司目前都只支持2个 CTs, CT0和CT1, 分别对应两个带宽约束BC0和 BC1 (Cisco称为global pool和sub pool))。

■ Bandwidth Constraints

我们指的带宽约束 (Bandwidth Constraint) 通常包含了最大带宽限制和应用在每个CTs上的带宽限制。通过CT的定义我们知道, 每个CT被指定了一个带宽约束或者一系列的带宽约束, 我们把带宽约束表示为BC “x” ($0 \leq x \leq \text{MaxBC}-1$)。对于一个给定的CT “x” ($0 \leq x \leq \text{MaxCT}-1 = 7$), 我们定义"Reserved(CTx)" 为所有属于该CTx的CR-LSPs的预留带宽的总和。为了控制CTs, 不同模式的BCs是可以想到的。举一个例子, 一个CT会分配一个单独的BC, 这个模式被称为最大分配模式 ("Maximum Allocation Model"), 是这样定义的:

1. $\text{MaxBC} = \text{MaxCT}$; 2. 对于每一个x取值 ($0 \leq x \leq (\text{MaxCT}-1)$), $\text{Reserved}(\text{CT}x) \leq \text{BC}x$ 。

为了说明这个模式, 我们假设一条链路上有100M带宽, 使用了3个CTs, 并且规定: $\text{BC}0=20\text{M}$, $\text{BC}1=50\text{M}$, $\text{BC}2=30$, 因此, 我们可以这样定义: 所有流量干线类型为CT2的LSPs使用不超过30M (比如话音, $\text{Voice} \leq 30$) ; 所有流量干线类型为CT1的LSPs使用不超过50M (比如关键数据, $\text{Data} \leq 30$) ; 所有流量干线类型为CT0的LSPs使用不超过20M (比如其他数据, Best Effort

$\text{Data} \leq 30$)。

另外一个例子, 一个"Russian Doll"模式带宽约束BC可以这样定义 : 1. $\text{MaxBC} = \text{MaxCT}$; 2. 对于每一个x取值 ($0 \leq x \leq (\text{MaxCT}-1)$), $\text{SUM}(\text{Reserved}(\text{CT}y)) \leq \text{BC}x$ ($x \leq y \leq (\text{MaxCT}-1)$)。

为了说明这个模式, 我们假设一条链路上有100M带宽, 使用了3个CTs, 并且规定: $\text{BC}0=100\text{M}$, $\text{BC}1=80\text{M}$, $\text{BC}2=60$, 因此, 我们可以这样定义: 所有流量干线类型为CT2的LSPs使用不超过60M (话音, $\text{Voice} \leq 60$) ; 所有流量干线类型为CT1或者CT2的LSPs使用不超过80M (话音和关键数据, $\text{Voice} + \text{Data} \leq 80$) ; 所有流量干线类型为CT0或者CT1或者CT2的LSPs使用不超过100M (话音、关键数据和其他数据, $\text{Voice} + \text{Data} + \text{Best Effort Data} \leq 100$)。

■ Preemption 和 TE-Classes

DS-TE完全支持抢占机制, 抢占机制和所有的Class Types一起配合应用。Setup优先级和Holding优先级维持原来的定义并且不受到聚合模式QoS机制的LSP及其Class Type的影响。也就是说, 如果LSP1与LSP2竞争资源, 只要LSP1的Setup优先级高于LSP2的Holding优先级, 那么LSP1就可以抢占LSP2的资源而不管LSP1的OA/CT和LSP2的OA/CT。

我们把TE-Class定义为一个Class-Type和允许这个Class-Type抢占的一个优先级的组合。也就是说, 一个传输Class-Type的流量干线可以使用这个定义的抢占优先级作为Setup或者Holding优先级。

通过这样的定义, 对于一个Class-Type, 可能有一个或者多个TE-classes使用这个Class-Type, 每个TE-classes使用不同的抢占优先级; 对于一个抢占优先级, 可能有一个或者多个TE-Class(es)使用这个抢占优先级, 每个Class-Class使用不同的Class-Type。.

DS-TE必须让一个Class-Type只能被一个TE-Class使用, 这样就会确保Class-Type之间不会发生



抢占关系。DS-TE允许两个具有相同Class-Type的LSPs使用不同的抢占优先级，并且允许高Setup优先级的抢占低Holding优先级LSP的资源。换句话说，DS-TE必须允许一个指定的Class-Type可以定义多个TE-Classes，这样就会确保一个Class-Type的抢占关系。例如，可以定义一个Class-Type有3个TE-Classes，一个使用抢占优先级0；一个使用抢占优先级1；一个使用抢占优先级4。

DS-TE允许两个使用不同Class-Types的LSPs使用不同的抢占优先级，并且允许高Setup优先级的LSP可以抢占低Holding优先级的资源。

例如，可以定义2个Class-Types (CT0 和CT1)，每个CT由2个TE-Classes组成：

- TE-Class1 : CT0, 抢占优先级0；
- TE-Class2 : CT0, 抢占优先级2；
- TE-Class3 : CT1, 抢占优先级1；
- TE-Class4 : CT1, 抢占优先级3。

那么，可以在LSP上传输Setup优先级为0和Holding优先级为0的CT0的流量干线；可以在LSP上传输Setup优先级为2和Holding优先级为0的CT0的流量干线；可以在LSP上传输Setup优先级为1和Holding优先级为1的CT1的流量干线；可以在

LSP上传输Setup优先级为3和Holding优先级为1的CT1的流量干线。而不能在LSP上传输Setup优先级为1和Holding优先级为1的CT0的流量干线，也不能在LSP上传输Setup优先级为0和Holding优先级为0的CT1的流量干线。

DS-TE允许2个使用不同的Class-Types的LSPs使用相同的抢占优先级。换句话说，允许TE-classes使用有相同抢占优先级的不同CTs。这就确保在不同Class-Types的LSPs之间不会发生抢占关系。例如，可以定义3个Class-Types (CT0, CT1和CT2)，每个CT有1个TE-Class，并且都使用抢占优先级0。这种情况下，不会有抢占的情况发生。

因为有8个抢占优先级和最多8个Class-Types，因此理论上最多可以有64个TE-Classes。这已经远远超过了目前的实际需求，目前的需求是DS-TE必须支持最多8个TE-classes。现有的TE中，被抢占的LSP为断开状态，它的Head-end 节点会尝试重新建立这条LSP，这样会重新进行CSPF计算出一条新的路径。需要注意的是，如果重新建立失败，这个Head-end会阶段性的发起重建的尝试。

我们公司TE版本的TE-Classes、Class-Type和抢占优先级的实现如下图：

TE CLASS	CLASS TYPE	PRIORITY	BW RESERVED (kbps)	BW AVAILABLE (kbps)
0	0	0	0	200000
1	0	1	0	200000
2	0	2	0	200000
3	0	3	0	200000
4	0	4	0	200000
5	0	5	0	200000
6	0	6	0	200000
7	0	7	200	199800
8	1	0	0	100000
9	1	1	0	100000
10	1	2	0	100000
11	1	3	0	100000
12	1	4	0	100000
13	1	5	0	100000
14	1	6	0	100000
15	1	7	0	100000

■ 流量映像到LSPs

DS-TE可以将通过Diff-Serv进行分类的流量映像到不同的LSPs上。

■ 动态调整Diff-Serv 的PHBs

可以根据每个PHB的性能需求变化进行配置参数的动态调整。

■ 超额 (Overbooking)

因为TE可以允许一定的流量超额通过LSPs，因此DS-TE必须允许流量超额，并且可以根据CTs的不同允许的不同程度的超额。对于一个指定的CT，应该可以在网络的不同部分允许不同程度

的超额。

■ 恢复 (Restoration)

因为在TE中，恢复策略可以通过使用抢占优先级来有效地控制各条LSPs的重要级别，并且根据重要级别实现恢复的考虑。DS-TE也必须确保这点。

具体更详细地描述请参见RFC3564 (Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering)，同时建议阅读Juniper公司的《MPLS DiffServ-aware Traffic Engineering》文档。



分层PE技术简介

刘小龙



前言

近年来越来越多的企业将ERP、财务、OA、决策支持、语音、视频等关键业务承载在网络中。各业务系统需要获取相对独立的逻辑网络资源，以满足不同业务系统对安全性、服务质量、管理的要求；同时，各业务系统之间的流程整合又需要提供相互访问的途径，而且要保证互访的安全性。传统的IP网络业务单一、关键业务少，难以满足企业网的应用需求。

MPLS VPN是目前比较理想的实现网络资源分配的技术，MPLS VPN技术可以将一个物理的企业网络划分为多个独立的“逻辑网络”，每类业务

都可以获得一部分网络资源供自己使用，包括地址空间、路由转发表、带宽、隧道、服务质量等。而IT部门也可以在总部对全局的网络资源实现统一管理和分配。其中MPLS VPN广泛应用于行业企业网络资源逻辑划分和安全隔离。

MPLS VPN框架包括的组件：P设备、PE设备（包括SPE和UPE设备）、CE设备、VPN业务管理系统，其中最关键的是PE设备。MPLS VPN也包括多种技术，其中企业网中应用最广泛的是L3 MPLS VPN。

分层网络

与

MPLS VPN网络架构

MPLS VPN模型

标准的L3 MPLS VPN是一种平面式模型（RFC2547Bis），PE设备无论处于网络的哪个层次，对其性能要求和接入链路要求是相同的，对于PE设备来说不存在网络层次上的区分，所有的PE都处于对等要求。由于路由在PE之间交换，在边缘方向的

PE要维护同核心层次PE设备相同的路由数量。而典型网络是分层、分级模型，设备性能依次下降，网络规模依次扩大。这就为PE设备向网络边缘的扩展带来了困难。



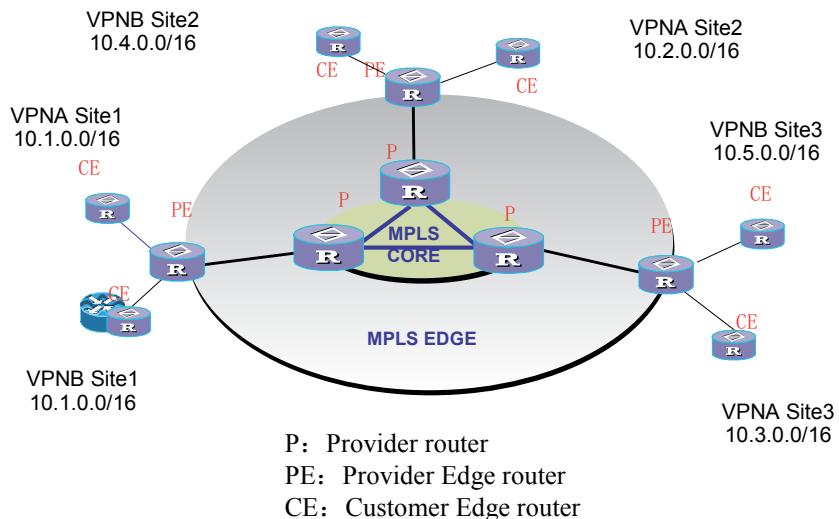


图1 MPLS VPN网络架构

如图1所示，MPLS VPN当前的框架结构中包括P、PE、CE等设备：

P router：骨干网中不与CE直接相连的设备，不感知VPN；

PE (Provider Edge) 骨干网中的边缘设备，它直接与用户的CE相连，完成VPN实现的主要功能；

CE (Custom Edge) 用户Site中直接与骨干网相连的边缘设备，不感知VPN，只需支持标准的IP功能即可。

在以上三种设备中，真正参与VPN业务部署的只有PE设备，也就是说MPLS VPN业务的智能化和业务压力都集中在PE设备上。但MPLS VPN是一种平面模型，PE在整个框架中是对等关系，无论处于网络中哪个位置，对性能的要求和接入链路要求是相同的。这种平面结构的问题在于，如果其中某些性能较低的PE存在性能和扩展性问题的时候，实际上也制约了整个网络VPN业务的广泛覆盖能力与进一步的扩展能力。

另外，当VPN用户距离PE很远的时候，需要通过WAN链路来连结，其数目至少同VPN用户的

数目相同。如果采用路由器就近接入用户，汇聚后通过一个WAN链路连结到PE，则可以节省费用，提高带宽利用率。但在这个WAN链路上需要提供区分不同的VPN用户。

网络的分层结构

目前网络设计基本都采用经典的分层结构，如城域网典型结构是核心-汇聚-接入三层模型，设备性能依次下降，网络规模依次扩大。在一个分层网络中如何部署PE节点，将PE部署到核心层、汇聚层，还是接入层？这是网络设计中经常遇到的问题。

PE设备接入用户需要大量接口，处理用户报文需要大容量的内存和转发能力，而各层次PE难以同时具备高性能和大量接口：

- 核心层性能高，但接口资源有限；
- 接入层接口数量大，但性能低；

- 汇聚层的接口容量和性能可能都不足。

由于MPLS VPN是平面结构，PE设备无论处于网络的哪个层次，对其性能要求是相同的，随着MPLS VPN业务的大规模部署，边缘网络的PE必然出现扩展性问题，形成瓶颈。

另外，在分级网络中还需要考虑跨AS的VPN部署。目前MPLS VPN跨AS技术，如VRF to VRF、MP-EBGP、Multi-hop-EBGP等方式，都是一种AS之间的对等结构，而不是一种分级结构，比较适用于运营商之间的VPN互通，而不太适合运营商内部的网络分级要求。

分级网络结构提出的问题：

既然网络的分级是必然的，能否在MPLS VPN网络框架设计中就考虑到分级网络的要求，解决扩展性问题，并实现跨AS的分级结构。

总结一下现有MPLS VPN架构的缺陷，最主要的问题是平面型的结构不能适应网络分级、分层模型的要求。由此导致了其它诸方面的问题：

- 由于网络MPLS VPN网络不能实现分层分

级，使业务的覆盖能力受到限制，使建网成本无法降低，影响了MPLS VPN的普遍部署，限制了高价值的端到端业务的开展。

- 网络的扩展能力受到制约，无法实现MPLS VPN业务不断向网络边缘延伸的需求，不利于业务的平滑演进及投资保护。

- 由于平面化模型不能充分利用网络各层次的能力，必须采用高档次的PE设备，增加了建网成本。

因此优化MPLS VPN网络架构成为必然的要求与发展方向。

针对以上问题，华为公司提出了MPLS VPN中PE的分层体系结构—HOPE解决方案，将PE的功能分布到多个设备上，它们承担不同的角色，并形成层次结构，共同完成一个集中式PE的功能。对处于较高层次的设备的路由和转发性能要求高，而对处于较低层次的设备的路由和转发性能要求低，同典型的分层分级网络模型相吻合，在分层部署L3 MPLS VPN时，解决了可扩展性问题。

HOPE的基本结构

分层PE原理

分层PE的结构如图2所示，直接连结用户的设备称为下层PE(Underlayer PE或User-end PE，用户侧PE)，简写为UPE，连结UPE并位于网络内部的设备称为上层PE(Superstratum PE)，简写为SPE。这种框架结构称为PE的分层结构(Hierarchy of PE)。多个UPE同SPE构成分层式PE，共同完成传统上一个PE的功能。它们之间的分工是：

- UPE维护其直接连接的VPN站点的路由，但不维护VPN中其它远程站点的路由或仅维护它们的聚合路由；

- SPE维护其通过UPE所连接的站点所在的VPN中的所有路由，包括本地和远程站点中的路由。

- UPE和SPE之间可以使用MP-IBGP，也可以



使用MP-EBGP。在采用MP-IBGP时，SPE作为各个UPE的路由反射器(RR)，UPE作为路由反射器的客户端，但SPE不作为其它PE的路由反射器。在采用

MP-EBGP时，UPE一般使用私有自治系统号。

分层式PE从外部来看同传统上的PE没有任何区别，因此它可以同其它PE在一个MPLS网络中共存。

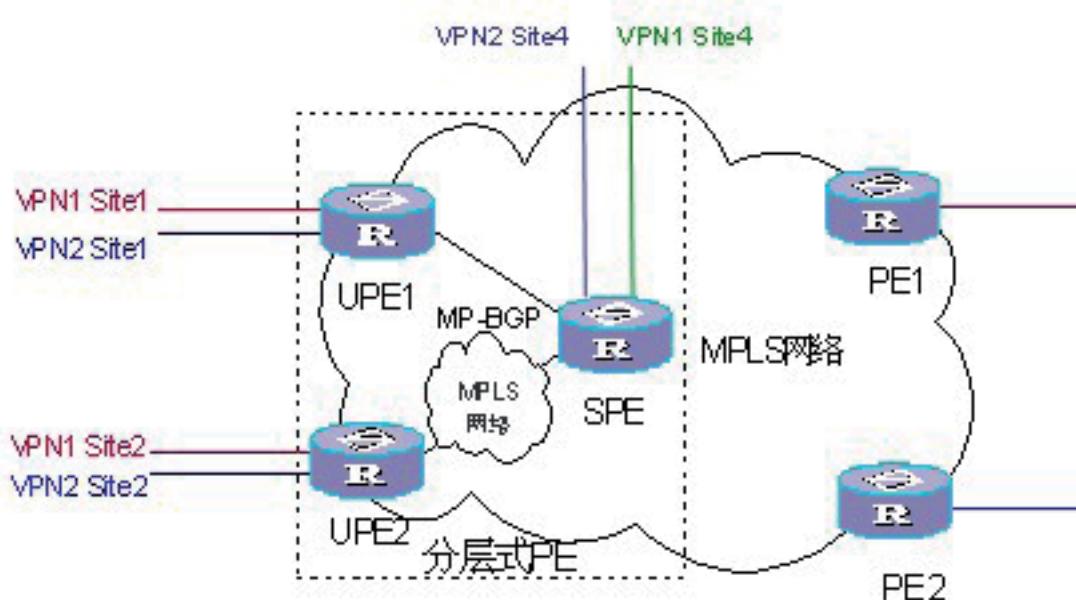


图2 HOPE框架结构

SPE—UPE连接方式

UPE和SPE之间采用标签转发，因而只需要一个(子)接口相互连接。SPE和UPE之间可以通过任何形式的接口/子接口连接，也可以通过隧道接口连接，这时候SPE和UPE之间可以相隔一个IP网络或MPLS网络，由于SPE和UPE通过MP-BGP对等，因而路由可以直接传递，无需做特殊处理，MP-EBGP的情况下配置多跳EBGP即可。在转发时，UPE或SPE发出的标签报文要经过一个隧道传递。如果是GRE隧道，要求GRE支持对MPLS报文的封装，如果是LSP，则需要中间的网络是一个MPLS网络，UPE和SPE上运行LDP/RSPV-TE等协议。

SPE和UPE之间只需要一个连接，这样SPE不需要具备大量的接口来接入用户。

路由信息的扩散

在UPE与SPE之间跑MP-IBGP协议，在UPE1上配置两个VRF(VPN1 site1, VPN2 site1)，在PE1上配置一个VRF(VPN1 site3)。以VPN1 site3中的路由10.0.0.0/8为例来描述路由信息交互流程。

扩散流程如下：D代表目的网段，N代表下一跳，label代表所携带的标签，PE之间使用云团表示，是代表之间可能还有P设备，虚线代表逻辑连接。

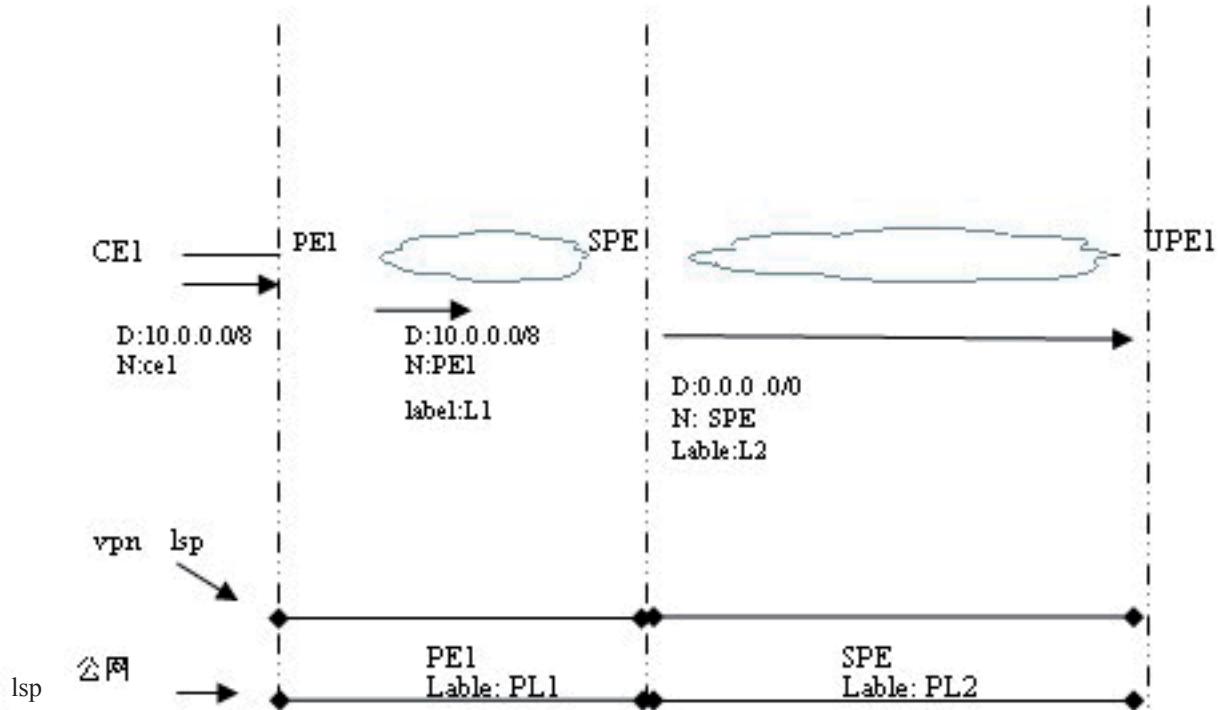


图3 路由选择过程

- ✓ PE1路由器使用RT值发布所有私网路由。
- ✓ SPE路由器接收到所有PE/SPE和UPE发布的私网路由，并把UPS发布的私网路由向所有的PE/SPE发布，对于UPE只发布带有标签的默认路由。

数据转发流程

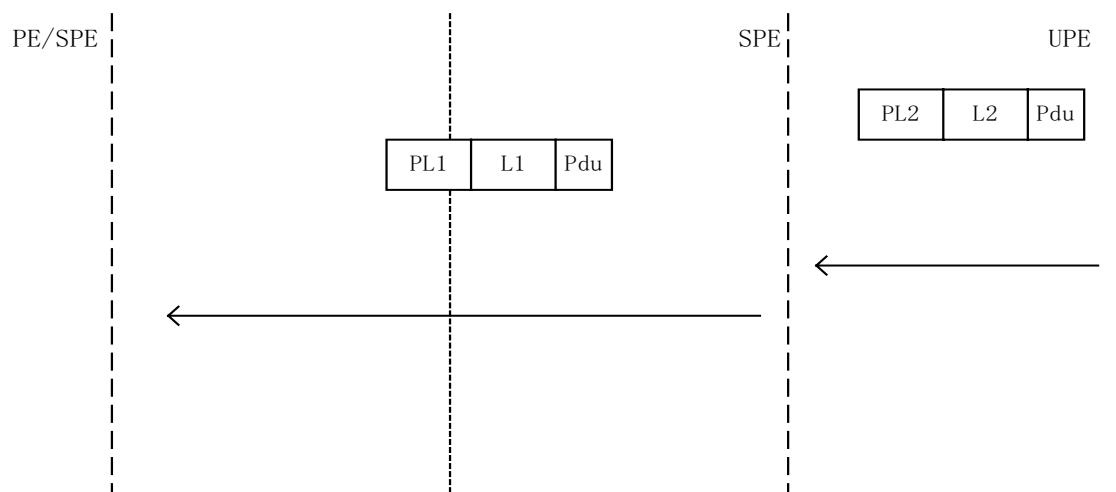


图4 报文转发过程

- ✓ UPE1路由器报文发向远端FEC，通过打上默认路由的私网标签，再加上UPE和SPE之间的隧道（若为LSP则打上lsp标签）公网标签PL2。
- ✓ SPE路由器接收到UPE发来的私网报文根据携带的私网标签L2来查找对应的VRF表，如果查找到相应的路由则，打上新的私网标签和达到对端PE的公网标签PL1。
- ✓ 返回的报文处理同正常的L3VPN的转发流程。

SPE – UPE

协议内容

SPE和UPE之间运行的MP-BGP，可以是MP-IBGP，也可以是MP-EBGP。这取决于SPE和UPE是否在一个AS内。

在同一个AS中，采用MP-IBGP，这时候SPE作为多个UPE的路由反射器，但可以不作其它PE的路由反射器。为了拒绝从其它PE发布过来的不属于本分层式PE所连接的Site的VPN中的路由，SPE上要根据各个UPE的所有VRF的import route-target list的合集生成一个全局import route-target list，用于过滤从其它PE发布过来的路由。这个全局列表可以根据SPE和UPE之间交换的信息动态生成，也可以静态的加以配置。

如果SPE和UPE属于不同的AS，它们之间运行MP-EBGP。SPE上同样需要生成一个全局Import route-target list。一般来说，UPE要

采用私有自治系统号，在SPE发布路由给其它PE时，要略去这个私有自治系统号。

SPE发布给UPE的VRF默认路由可以是动态生成的，也可以静态的配置。动态VRF默认路由应该为分层式PE所连接的所有Site所对应的VRF生成，并发布给所有UPE。在发布时，可以根据上面所提到的ORF进行过滤。

动态机制是这样的，UPE通过BGP的Route Refresh消息发布一个ORF(Outbound Route Filter)给SPE，这个ORF中包含了一个扩展团体列表，其内容是UPE上所有VRF的import route-target list的合集。SPE将所有UPE的扩展团体列表合并起来，形成全局列表。静态列表与动态列表的生成规则是一样的。

目前对于ORF并不支持，对于动态的Import route-target list交换暂时没有实现。

HOPE实现跨AS VPN部署

如图5所示，在这个案例中，骨干网和城域网属于不同的自治系统，骨干网可以设置SPE，城域网设置UPE。UPE将城域网全部路由发送给SPE，而SPE只发送VRF默认路由给UPE。这样，城域网只需要维护内部的VPN Site路由，而不需要维护城域网之外Site的路由。骨干网需要维护全局VPN Site的路由。

在跨AS方案中，SPE-UPE协议可以采用MP-EBGP或Multi-hop EBGP方式，实现灵活的部署。

采用HOPE实现的跨AS方案特点在于适应了网络分级的要求，上级网络（骨干网）处理全局业务；下级网络（城域网）只需要处理本地业务，这样就不会因为全局VPN业务发展导致下级网络出现容量和扩展性问题。

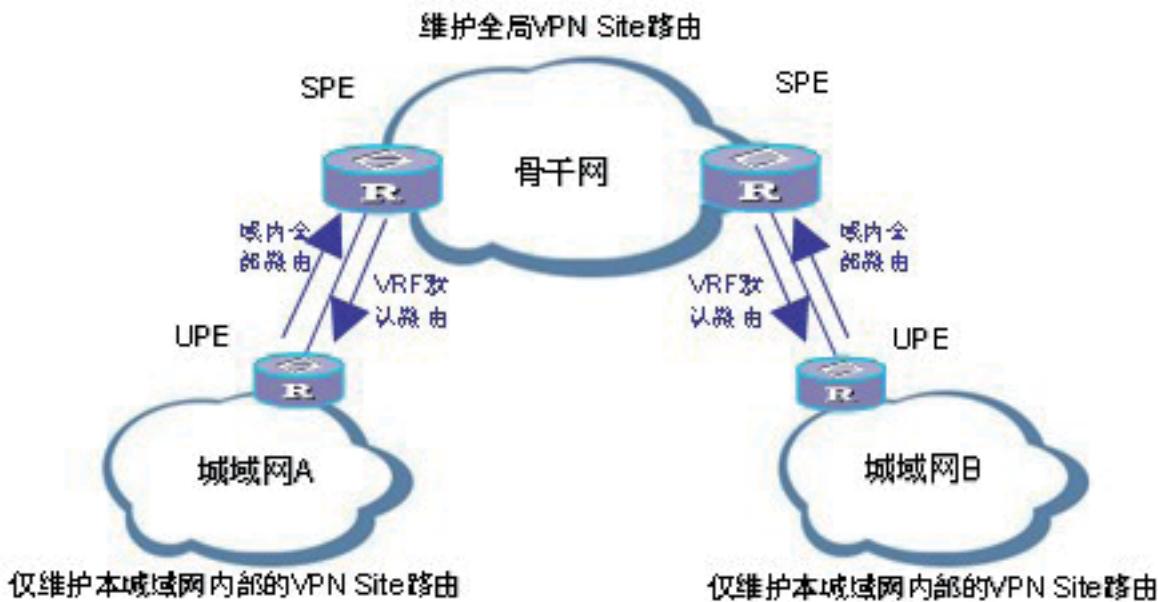


图5 HOPE实现跨AS的分级网络

HOPE分层网络的演进

PE初期部署在汇聚层。当汇聚层接口数目不足时，扩展到接入层，接入层充当UPE，汇聚层充当SPE；当汇聚层路由容量不够时，扩展到核心层，核心层充当SPE，汇聚层充当UPE；当两种情况都发生时，形成3层结构，汇聚层充当MPE（中间层PE）。这种演进方式非常适合城域网的结构。如图6。

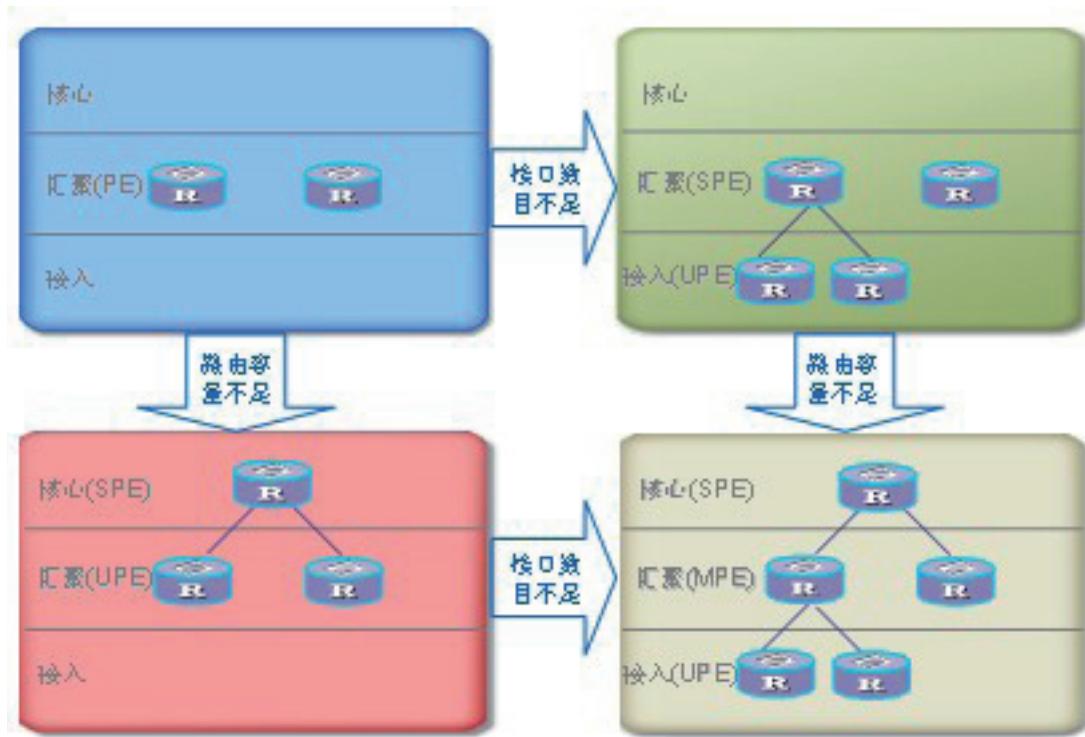


图6 HOPE分层网络的演进

HOPE分级网络的延伸

如图7所示，MPLS VPN在全国范围内部署时，通常采用一种扁平化的组网结构，也就是直接通过骨干网来提供MPLS VPN业务。在这种结构中，骨干网的PE通常设置在中心城市，用户CE都通过一条链路汇聚到PE节点。

这种方式的缺陷在于：中心城市在接入远程CE时，需要消耗大量的广域链路资源；骨干网的规模不可能无限制地扩展，其覆盖能力和扩展性面临严峻挑战。

采用HOPE之后，可以在地市甚至县部署UPE节点，形成多层结构，就近接入VPN用户。同时网络的覆盖能力得到了增强，可以根据需要实现业务的平滑演进，以及网络的扩展与延伸。SPE和UPE可以在同一个AS内，也可以实现AS之间的连接。

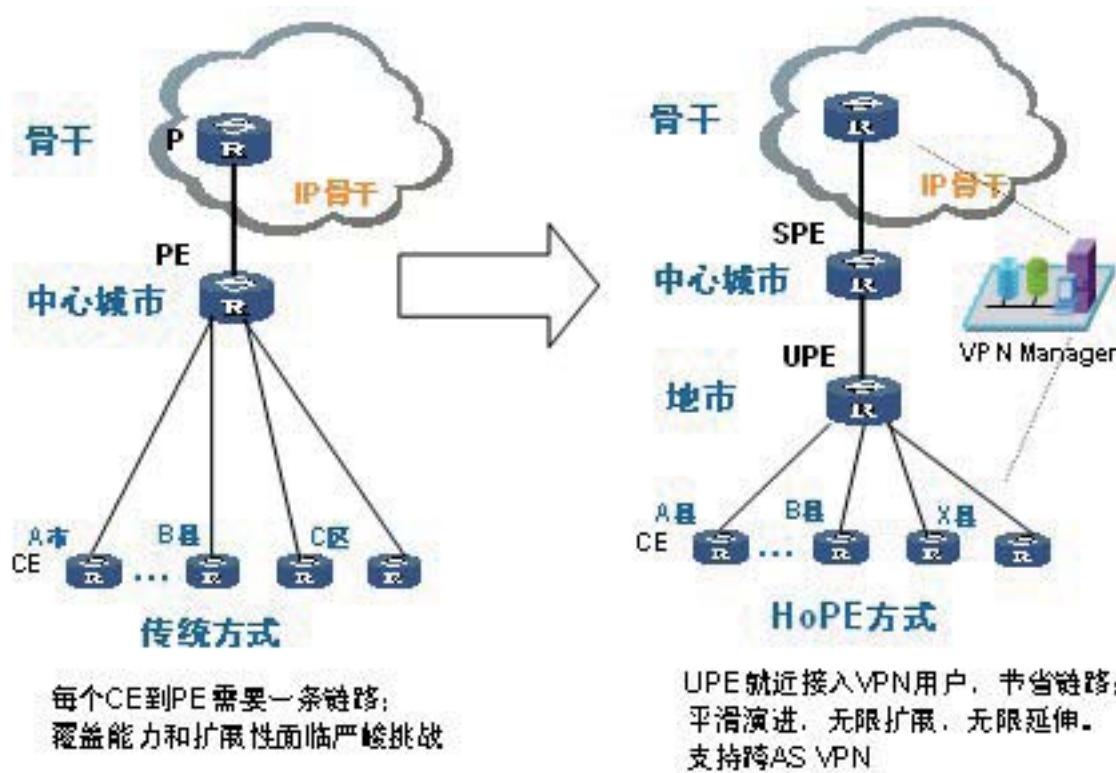


图7 HOPE分级网络的延伸

分层PE 的嵌套

一个分层式PE还可以作为一个UPE，同一个SPE组成新的分层式PE，同样一个分层式PE可以作为一个SPE，同多个UPE组成新的分层式PE，称这种位于中间的PE为MPE（Middle-level PE）。这种嵌套可以是无穷的。一个SPE在连接一个分层式PE作为UPE的同时，还可以连接单个的UPE设备。

SPE和MPE，以及MPE和UPE之间，均运行MP-BGP协议。如果是MP-IBGP协议，则SPE是各个MPE的路由反射器，而MPE是各个UPE的路由反射器。MP-BGP为上层PE发布下层PE上的所有VPN路由，而只为下层PE发布上层PE的VRF默认

路由或聚合路由。因此，SPE上维护了这个分层式PE所接入的所有站点的VPN路由，路由数目最多；MPE上维护了一部分路由，数目较多；而UPE上只维护它所直接连接的站点的VPN路由，数目最少。同典型网络中设备的能力相匹配，具有良好的可扩展性。

SPE发布的VRF默认路由携带了标签，在MPE上要对这个标签进行替换，并发布给UPE。如果MPE已经为这个VRF生成了默认路由，则在MPE上生成一个标签转发项（ILM）。

如同2层的分层式PE一样，上层PE要为下



层PE生成一个全局入口RT列表，过滤不需要的VPN路由。具体来讲，MPE以各个UPE上所有VRF的入口RT列表的合集生成一个全局入口RT列表，SPE又以各个MPE的全局入口RT列表的合集生成一个全局入口RT列表。生成方式可以是动态的或静态的。在动态的情况下，MPE要将UPE通过ORF发送的入口RT列表转发给SPE。

在转发时，从VPN本地站点始发的报文，到达UPE后，查找相应的VRF中的默认路由，

PUSH标签，转发到MPE，POP标签后，查找相应的VRF中的默认路由，转发到SPE，SPE上POP标签，查找VRF转发表，PUSH内外层标签，按照传统的BGP/MPLS VPN流程转发。

从远程站点始发的报文，通过MPLS网络到达SPE后，由于MPE已经为其目的地址分配了内层标签，因此对内层标签进行SWAP操作，转发到MPE，同样的，UPE已经为这个目的地址分配了内层标签，POP标签后转发给本地站点。

总 结

HOPE网络完全适应典型分层和分级网络模型，解决了行业网络中VPN业务广泛覆盖和扩展性问题。网络的潜力得到了充分的发挥，建网成本进一步降低。同时，HOPE在框架中已经考虑了跨AS VPN及Carrier's Carrier等边界技术，增强了业务服务能力。跨AS技术可以使网络的规模进一步扩展，Carrier's Carrier可以实现VPN的嵌套与进一步细分。

总的来说，通过HOPE的优化使MPLS VPN网络具备了端到端业务的支撑能力。

HOPE的核心理念是：平滑演进、无限扩展、无限延伸。

浅谈MPLS 测试方法

金炜



概 述

MPLS作为一种转发技术已经发展了很多年，起初以提高转发效率而提出的这种技术经过多年发展因其本身良好的扩展性，为其赋予了新的生机。随着基于MPLS技术的VPN应用、TE和QoS等各种应用不断在各大网络上部署，MPLS逐渐成为网络世界中新热点的同时，也逐渐成为我司设备的卖点之一。

虽然MPLS转发技术位于二、三层之间，但是其实现需要路由协议、LDP等标签分配协议等上层协议作为支撑，并且为了实现基于MPLS的各种应用，还对很多上层协议进行了扩展。可以说，MPLS模块涵盖了众多相关协议，是一个非常复杂的知识体系。这也为作为测试人员的我们提出了很高的要求。

说到测试方法，其实对于任何一个模块，任

何一个协议没有一个固定的、一成不变的测试方法。不同的测试人员，测试手段不同、关注点不同、思维方式不同都能形成一套自己特有的测试方法，并且随着测试不断进行，对协议、对整个模块的理解和对其应用的理解也在不断深入，测试方法也随之不断丰富、完善。测试方法不断丰富的同时，测试也会变得不断深入。

本文是平时测试中使用的一些方法和遇到的一些问题粗略总结，算是抛砖引玉，希望不断补充完善，共同丰富我们这个测试方法，共同提高测试水平！

需要特别指出的是，一个基于MPLS的重要应用：TE（Traffic Engineering）由于其具有相对独立的知识体现和自身复杂性，将有专门文章介绍它的测试方法，本文将不对其进行讨论。

MPLS基础协议

测试方法描述

MPLS基础协议是指支撑MPLS、VPN等各种应用的协议，包括：LDP、MBGP和各种路由协议多实例等。确保基础协议功能完备是其他MPLS应用功能正常的保证。因此，这里首先总结一些对MPLS基本协议测试的方法。

说到协议测试，不外乎包括基本功能测试、协议一致性测试、互通测试和性能测试等几个方面。协议测试方法也包括通用测试方法和根据不同协议而特有的测试方法，通用测试方法包括：命令行配置删除、边界值和非法值设置、接口板连接线热插拔等，这些方法相信大家都已经掌握，本文也就不再将其罗列到各个部分测试方法描述中。这里主要讨论的是我们在测试相关协议时需要关注、容易出现问题的方面，和在测试这些协议特性时通常使用的一些测试手段。

LDP测试方法

LDP（Label Distribute Protocol）是我司设备实现通用标签分配协议，可服务于所有MPLS应用。由于LDP协议本身非常复杂，定义了不同的标签分配模式、标签控制模式和标签保持模式，设备可被配置在多种模式下工作。同时，LDP还支持Loop-Detect等特性，使其测试组网、测试手段都非常复杂，下面我们从LDP协议几个主要功能部分讨论对它的测试。

基本功能测试

■ 邻居建立

LDP通过TCP建立邻居关系，并在邻居间直接传递标签映射消息。协议规定，在两个LSR设备直接只允许存在一个LDP会话关系（LDP Session），这也是测试会话功能的重点。主要测试方法包括：在两台LSR直接创建多个多种类型直连接口，并在物理接口上封装各种类型的链路协议，包括：以太网、ATM、FR子接口，PPP、MP、MFR等，同时可以指定建立LDP会话使用的IP地址，包括各种物理接口IP地址、子接口和虚接口IP地址，地址借用接口IP地址，地址协商接口IP地址等，验证此时LDP会话是否正确建立，LDP会话是否唯一。

我司设备支持在同一个接口上配置多个IP地址，同样LDP支持利用这些子IP（Sub IP）建立会话关系。子IP又分为配置在主接口、子接口和各种虚接口上等不同组合。将接口和各种类型IP地址结合是测试LDP常常使用的方法。在测试LDP邻居时，我们不但需要验证相邻设备之间建立LDP会话，还要测试任意两台设备之间创建Remote方式LDP会话。

LDP协议通过TCP建立邻居间会话关系，同时也提供基于TCP MD5认证机制。与测试其它协议认证类似，在配置LDP认证后，关注是否会影响LDP会话建立，以及各种协议报文传递，TCP、LDP各种状态显示是否正确。特别需要注意的是，对于携带MD5认证信息的TCP邻居，在显示其状态



时会有一个星号“*”进行标识，这也是确认认证是否成功的方法之一。

反复Up/Down接口、子接口、虚接口状态，插拔物理连接线，热插拔接口板，修改接口IP地址等操作是否会影响LDP会话状态。记得曾经出过这样一个网上问题：由于连接两台设备之间的光电转换器质量不好，引起LSR接口反复Up/Down，LDP会话反复建立，一定时间后LDP会话无法建立，重启设备后恢复正常。在实验室复现这个问题时还出现了由于不断插拔物理连接线引起设备宕机的严重问题。

LDP会话建立另一个方面就是LDP能力和参数协商。在修改接口参数、LDP工作状态参数后，LDP会话参数能否正确建立，协商后的参数和能力是否正确。

■ 标签分配

LDP是广泛使用的标签分配协议，主要功能是为不同FEC（路由）自动分配标签，所以标签分配是否正确是衡量协议是否正确工作的基本标准。LDP协议会为本地直连路由、动态路由分配标签，将标签和路由绑定关系封装在MAP消息中传递给上游LSR，从而在MPLS域中产生对应该路由的LSP。前面提到，由于标签分配模式、标签控制模式和标签保持模式的不同组合，LDP可能工作在多种模式下。而最常用的是DU+自由标签保持+独立标签控制模式，这也是MPLS应用的主要方式。

LSP是LSR依据路由逐跳创建的，LSP出接口应与路由出接口保持一致，并且在V5版本上还支持为等价路由创建多条不同接口LSP。对LSP的测试包括LSP与路由同步，在路由或其下一跳发生变化时，LSP能否同步变化，切换时间不应过长。LSP完整建立包括在MPLS域内对所有路由都应建立对应的LSP（除缺省路由和聚合路由外）。LSP建立完成后，可以查看LSR上MPLS LSP、ILM、FTN（V5版本对应FIB）和NHLFE等表项，

各个表项建立、相互关系应该一致正确。特别是在路由常常发生变化的网络中，我司设备曾经出现过LSP已经删除，标签已经释放，但是底层ILM和NHLFE表项未删除导致转发不通的问题。

由于我司设备支持全局标签空间，LSR为每一条路由分配一个标签，不同标签对应不同的FEC。因此，对于不应出现LSR为不同FEC分配相同标签的情况，保证在LSR上标签与FEC对应的全局唯一性。同样，在FEC对应路由消失后，LSR应及时释放为其分配的标签资源，并通过LDP消息通知上游邻居。被释放的标签能够被再次分配给其他FEC。特别是在大量路由出现路由振荡时，应特别检查是否会出现标签未被释放和不能重新分配的标签泄漏现象。

LDP另一个常常会出现问题的地方就是同一路由在多条备份链路上反复切换、等价路由某些出接口反复振荡时，LDP不能正确与路由变化保持同步，某些表项反复建立导致某些已经过期的表项无法及时删除，ILM、FTN与NHLFE对应不一致引起MPLS转发不通。

测试LSP时一个重要测试工具就是LSP ping命令。普通的ping命令只能检测路由可达性，对判断LSP逐跳是否建立完整无能为力。这时，我们可以使用LSP ping命令，为找到存在问题的LSP提供了一个很好的手段。

■ 环路检测

为了防止因为路由层面产生环路引起MPLS转发环路，LDP可以在MAP消息中携带环路检测信息，包括Hop-Count属性和Path-Vector属性。设备上通过设置允许建立LSP经过的最大跳数来防止环路产生。在实验室中由于路由产生环路比较困难，因此测试中通常通过设置较小的LDP跳数达到检测环路目的。配置了环路检查的路由器与没有配置环路检查路由器之间LDP邻居关系建立、对没有携带环路检查的LDP MAP消息处理和没有环路检查属性

的MAP消息处理。对即将达到和已经达到甚至超过规定跳数的LDP消息处理等。

协议一致性测试

目前，很多厂商的测试设备都提供了对各种基础协议进行一致性测试的测试套，包括Agilent公司的RouterTester、Spirent公司的AX4000和IXIA公司的IxANVL。这些测试套都是仪器厂商严格依据标准RFC开发的自动化脚本，几乎覆盖了对应RFC中定义的每一个功能点。利用这些协议一致性测试套，能够非常准确地测试我司设备是否按照标准协议实现，能帮助我们发现很多互通测试和遍历测试中无法发现的深层协议问题。对于MPLS部分，主要测试仪都有LDP协议测试套，协议一致性测试也早已在平台产品测试部得到广泛使用。

同时，部分厂商仪器还提供对BGP、MBGP、L2VPN和VPLS等其他MPLS应用模块的协议一致性测试套，具体支持模块的详细信息请参见原仪器集中组相关文档。

MBGP测试方法

作为BGP4对MPLS VPN应用的扩展，根据RFC2858定义的BGP多协议扩展标准，BGP协议添加了支持发送VPNv4路由、在Update报文中携带标签、RT和其他扩展团体属性的能力（由于本文只针对MPLS测试方法，对BGP在IPv6和组播方面的扩展这里不作讨论）。因此，测试MBGP时，除了可以使用BGP4基本测试方法对基本配置、路由发布、路由选路、BGP通用属性、路由策略、路由反射等进行测试外，还应当对下面MBGP几个特有方面进行测试：

能力协商

作为BGP4的一个扩展，在Open报文中将携带标示本地BGP所支持能力的参数，便于在对等体之

间建立邻居前协商双方都支持的能力交集，建立对应能力的对等体关系。对能力协商的测试包括在两台PE间配置相应BGP能力后，他们之间的MBGP邻居是否能够正确协商建立对等体。我们不但需要测试普通PE间的I-MBGP邻居，还需要测试ASBR之间和跨域方式中使用的E-MBGP邻居关系，分别使用物理接口和LOOPBACK接口建立邻居关系等各种情况。另外，我们还应该验证在设备收到Open报文种携带了本地不支持和不能识别的能力参数时其处理是否正确。根据实现，我司设备在收到无法识别的能力参数后，应当主动发送一个Notification报文，并断开TCP连接。

RD与RT

RD (Route-Distinguisher) 用于标示PE设备上不同VPN实例，其主要作用也就是实现VPN实例之间地址复用，它与IP地址一起构成了12byte的VPNv4地址空间，RD与路由一起被携带在BGP Update报文中发布给对端。一方面我们需要验证RD功能是否实现，PE设备是否能够根据不同RD实现IP地址复用，携带不同RD的相同IP路由在PE上应该对应不同VPN实例路由。同时，RD不具有选路能力，不应影响路由接收和优选，对于同一VPN携带不同RD的相同IP路由，PE设备不应根据RD优选路由或当两条不同路由进行处理。由于RD具有两种赋值形式，在测试中也需要考虑到使用不同结构RD路由的传递，特别是对临界值、非常规值（如AS号为65535，IP地址为广播、组播地址等）的测试。

RT (Route-Target) 是VPNv4路由携带的一个重要属性，它决定VPN路由的收发和过滤，PE依靠RT属性区分不同VPN之间路由，也成为MBGP测试中的一个重点。

利用RT属性对VPN路由进行过滤。RT与RD属性具有相同数据格式，但属性分为Import和



Export两种。Export属性跟随对应VPN路由通过MBGP发送到对端，而Import属性则用于与收到的VPNv4路由中携带的RT Export属性进行比较过滤路由。对RT过滤路由功能可以从匹配、不匹配等多个状态进行测试。

当PE设备上VPN实例中配置的RT export属性发生变化时，该PE发布对应这个VPN路由中携带的RT属性也应该同步变化，PE应该刷新这个VPN实例对应的VPNv4路由，更新其RT属性。同样，当VPN实例对应RT import属性变化时，被改变PE设备应该主动发出BGP refresh报文刷新VPN路由，用新配置的RT属性对路由进行过滤。

与RD不同，我们可以为一个VPN实例配置多个RT属性，并且RT属性被放置在BGP Update报文中的扩展团体属性中发布，格式与普通团体属性类似。那么当路由同时携带多个扩展团体属性和RT属性时，BGP协议、路由策略能否正确分析、处理这些不同属性，不会产生相互影响。

路由转发

作为一个路由协议，最基本最重要的功能就是必须保证路由传递正确，避免产生环路。作为BGP4的一个扩展，MBGP继承了其几乎所有特性，在路由测试方法上也与BGP大致相同，主要从选路、路由策略、环路检测、BGP各种属性等多个方面进行验证，这里也不作详细介绍。

较BGP不同的一点是：PE设备在收到VPNv4路由后，只有当接收端PE与路由发送端PE之间的LSP隧道建立成功后，这些路由才变得有效，这也为路由优选增加了一条规则。因此，我们可以利用这点，将路由发布、更新与PE间LSP反复切换、LDP邻居关系改变、MPLS域内部路由变化等其他测试手段相结合，验证由于MPLS域引起的振荡是否会影响VPNv4路由的传递和学习。

由于RD的存在实现了IP地址空间重叠，多个VPN之间可以使用相同的IP地址。那么，MBGP在

转发、处理VPNv4IP路由时能够根据RD、RT属性区分不同VPN空间路由。在测试中我们需要有意为不同VPN实例配置重叠的IP地址空间，验证MBGP能否正确处理这些路由。

标签分配

作为BGP协议另一个重要的功能扩展，MBGP具有为路由分配标签能力，路由可以是IPv4路由、VPNv4路由和IPv6路由。测试MBGP为这三种路由分配标签的功能需要搭建不同的测试环境，但其测试方法基本相同，都是验证MBGP能否为各种路由分配标签，为不同Site的VPNv4路由和IPv6路由以及不同IPv4路由分配的标签应该各不相同。并且应该考虑同一PE设备同时为这几种路由分配标签，并且存在路由振荡的情况，MBGP标签是否分配正确，被释放的标签能否及时收回等。

然而，MBGP应用并不是孤立的，它需要跟LDP等其他协议一起实现MPLS各种应用。所以，MBGP大部分功能测试还需要借助应用、组网进行。所以，MBGP相同更多方面的测试方法将在后面章节中继续讨论。

路由协议多实例测试方法

路由协议多实例用于PE设备与CE设备之间交换VPN路由。各个VPN路由在PE设备之间以VPNv4路由形式利用MBGP交换。到达PE设备后，需要通过支持多实例的路由协议向CE设备发布这些路由。目前实现多实例的路由协议包括：静态路由多实例、RIP多实例、OSPF多实例、ISIS多实例以及BGP多实例。需要说明的是：多实例只对PE设备而言，在CE设备没有多实例概念，因此对路由协议多实例的大部分测试和操作都在PE设备上进行，同时路由协议多实例不存在网络拓扑概念。

(OSPF除外，MBGP对支持OSPF路由传递进行了一些扩展，使多个Site的OSPF区域可以连接为一个整体），不必太多关注网络拓扑变化对路由协议的影响。所以，在测试路由协议多实例时，主要还是关注PE和CE之间邻居关系建立、路由交互，各个路由协议与MBGP互通、路由相互引入等方面。下面分别就各个协议具体一些测试方法进行讨论。

正如上面所说，对路由协议多实例的测试重点不再放在通过构建复杂网络结构，验证路由发布、计算和防止路由环路，因为我们只需要关注PE和CE两台设备之间的路由交换。与普通路由协议不同之处在于，多实例路由协议与VPN实例相关，一台PE设备上通常会存在多个VPN实例，一个VPN实例也会绑定多个接口，甚至同一个物理接口不同子接口下绑定的VPN实例也会不同。这种多类型接口、多VPN实例与多实例路由协议相结合成为我们测试多实例路由协议方法之一。相同VPN不同Site使用不同路由协议、他们之间路由相互引入、相同路由协议使用不用进程或不同VPN实例使用相同路由协议等都可以成为我们的测试手段。同一VPN实例可能会使用多种路由协议转发路由，这就存在各种路由协议间路由相互引入、发布的问题。同一条路由被多次反复引入后就会产生重复路由、路由振荡的问题。

当然，某些特殊的网络拓扑结构同样会引起路由环路：比如一台CE同时与多台不同PE连接使用相同或不同的路由协议发布路由，PE间又存在MBGP对等体关系。又如一台CE与PE间存在多条链路相连，而不同链路间使用不同路由协议，或被绑定到不同VPN实例都有可能引起路由环路、路由选路错误和路由不能正确刷新等问题。前段时间德国IZB项目中出现多次MPLS网上问题就是由于用户使用了CE连接到不同PE这种“双归属”网络结构，导致某些VPN路由无法及时刷新、路由产生环路甚至路由器定时重启等很多严重问题。

另一个方面，由于VPN实例支持重叠的IP地

址空间也可能导致设备之间邻居关系建立不正常。相同路由协议多个VPN实例通过同一IP地址建立邻居，交换路由。同时在MCE组网模式下，CE设备上也配置为多实例路由协议，PE、CE相互将对端看作自己CE设备，此时邻居建立、路由交换又会有所不同。

对于RIP路由多实例，由于设备之间不需要建立邻居关系，在测试中只需要考虑与其他路由协议之间相互引入路由、RIP协议不同版本之间相同版本不同目的IP之间是否能够正常收发路由、带验证时路由收发等基本方面。

OSPF多实例

这里之所以将OSPF多实例单独进行讨论，是因为与其他路由协议不同，MBGP在PE设备直接传递OSPF多实例路由时为其作了一些扩展。在OSPF路由被引入到MBGP协议中发布给对端PE设备时，Update报文中不但携带了路由、RD、RT等通常VPNv4路由信息，还携带了关于原来OSPF路由中的Domain ID等扩展信息，使接收端PE设备再次将这些VPNv4路由重新引入到OSPF进程中时，能够根据这些信息将其转换为TYPE 3 LSA，而非通常的OSPF ASE路由。这样对于VPN网络而言，各个VPN Site网络被连接为一个整体，连接各个VPN Site的服务商MPLS网络成为一个大的骨干区，使OSPF多实例在PE上也有了网络拓扑结构的概念。同时，为了防止由这种网络结构带来诸如路由环路的问题，OSPF多实例自身也进行了相应扩展，包括引入了Domain ID、VPN Tag和Sham Link等概念，这都是有别于普通OSPF协议和其他多实例路由协议的地方，也是我们重点测试OSPF多实例协议的几个方面。针对VPN Tag属性我们多采用“双归属”网络结构，在不同PE上将MBGP路由与OSPF实例路由相互引入，同时对应不同VPN Site之间或不同VPN实例采用不同或相同Tag值，验证



PE能否正确处理这些VPN路由。

由于OSPF多实例可以借助MBGP形成这样一个特殊的网络结构，在测试中我们通常会使用“双归属”网络结构，及同一CE设备和同一VPN Site不同CE设备同时连接到多个PE上，在PE上配置Sham Link以及相同或不同Domain ID，验证路由计算是否正确，是否会在PE设备上形成路由环路等。LSA类型、对VPN Tag处理、VPN路由选路和VPN路由环路是在测试OSPF多实例时主要关注的几个方面。在测试中，区域划分策略对测试结构有很大影响，由于我们默认MPLS域为骨干区，那么我们在VPN网络中部署骨干区时，如果没有将骨干区与PE设备相连同样会由于骨干区不连续而引起路由计算错误。所以，我们可以PE、CE之间的链路设置为非骨干区、骨干区和虚连接区域等几种不同情况分别进行测试。

其他MPLS应用相关模块测试方法

除了上面提到的路由协议多实例以外，还有两个比较小的模块容易被大家忽略：ARP多实例和NAT多实例。他们都不能算作一个完整的协议，只是为实现MPLS各种应用对原有功能进行了一些扩展。

ARP多实例其实是ARP表为了支持IP地址空间重叠而进行了相应扩展，为ARP各个表项添加了实例一项标识。因此测试时需要重点考虑重复地址空间时ARP表项建立情况，对应接口VPN绑定改变后，ARP表能否及时更新等。此外，当存在子接口时，不同子接口绑定到不同VPN实例时，很容易出现ARP表混乱的现象。反复变化接口和子接口绑定的VPN实例，可能会由于ARP表没有及时更新而引起转发问题。当然，频繁Shutdown/Undo Shutdown接口、热插拔网线和接口板等可靠性测试操作也是常用的测试手段。

NAT多实例是为解决VPN用户访问公网资源而提出的。其实质就是在选择NAT被转换IP时将VPN实例作为ACL一个限制条件，也就是ACL支持对指定VPN实例IP地址空间地址进行选择。其测试方法与普通NAT类似，在配置多个VPN实例并存在地址空间重叠的PE设备上对特定VPN实例地址进行NAT转换，不同VPN地址使用不同公网转换地址等。

LSPM模块测试方法

LSPM（Label Switch Path Management）不是一个独立模块，并不与某个协议对应。相比其他模块，它运行在“后台”，只有简单的几条配置和显示命令。但是，它却控制着整个MPLS标签交换操作，维护MPLS各种表项，管理隧道映射关系和所有类型隧道。LSPM可以说是MPLS控制层面与转发层面之间的一个接口，其功能主要包括：标签管理和静态LSP管理、MPLS表项维护和隧道管理三大基本功能。下面分别对这三个方面进行讨论。

静态LSP和标签管理

LSP可以利用各种协议动态产生，也可以进行手动配置。LSPM为我们提供了创建、管理静态LSP的功能。我们可以通过配置各种静态LSP，LSR在LSP中所处位置不同（Ingress、Egress或中间路由器）分别进行测试。根据实现，在Ingress实现FEC与LSP绑定，此时，LSP出接口需要与路由下一跳的出接口保持一致，而在中间路由器和Egress路由器上则不再判断LSP出入接口是否与路由保持一致，而仅仅通过手动分配的标签是否正确，出入接口状态是否正确来决定LSP的有效性。所以在测试中，我们可以采用标签会聚、多出口LSP备份并通过改变接口状态在配置的多条静态LSP间相

互切换等方法验证对静态路由管理。当一个FEC被绑定到一条静态LSP的同时，LDP又为其分配了一条动态LSP，此时静态LSP具有优先有效性。

同样的，我们可以选择各种类型的接口作为静态LSP出、入接口，并且和动态LSP相互作为备份，考虑在比较复杂的网络振荡环境下LSP是否能够正确建立，MPLS是否正确转发。

MPLS报文是否成功转发是验证静态LSP是否配置成功的唯一有效方法，但是由于缺乏上层协议维护，LSP中任何一台路由器上静态LSP配置或工作发送错误都将导致整个LSP无法正常转发，这为问题定位带来一定困难。

为了更好、更可靠地管理标签，我司设备为静态LSP单独分配了一个标签空间（通常为16—1023），不与动态LSP、MBGP使用的标签进行复用。

MPLS相关表项维护

MPLS主要表项包括：MPLS LSP、MPLS FTN（V5版本修改为FIB）、MPLS ILM和MPLS NHLFE等。这些表中记录了FEC与MPLS标签绑定关系、MPLS标签出入接口信息、MPLS标签操作类型、多个MPLS标签对应关系等关系MPLS转发层面的重要信息。路由变化、LDP邻居关系变化、接口状态变化、全局和接口下MPLS相关配置变化等很多因素都会引起设备重新创建、刷新、删除这些表项，长时间、反复对这些表项进行操作，特别是在短时间内多次删除重建同一表项很可能会引起表项内容错误、无法访问等问题。因此，这些表项信息完整性和健壮性是我们测试的重点。很多MPLS转发问题都是由于这些表项本身错误或表项之间映射关系错误导致的。在测试MPLS全过程中都需要经常查看这些表项，特别是在路由经常发生振荡、接口状态不稳定时，MPLS表项是否能够正确刷新，及时与路由等各种状态同步是我们测试中的重点。

隧道管理

这里的隧道是指能够为各种MPLS应用服务，特别地能够为连接PE所使用的通道，主要包括LSP、GRE和MPLS TE Tunnel等。每一条隧道在创建和状态改变后都会通知LSPM模块，LSPM会根据LSP绑定FEC信息和Tunnel源、目的等信息自动将其与某个PE对应的VPN隧道相关联，实现PE之间VPN报文正常转发。

对LSPM隧道管理方面的测试主要需要考虑多条隧道备份、相互切换，

异常、性能测试

提到异常、攻击测试其本身就是一个很大的测试范畴，包括异常协议报文攻击、攻击报文攻击和临界状态操作测试等等。对设备进行异常、攻击测试通常会使用各种测试仪器和测试工具，通过构造非正常协议、状态报文和状态攻击报文持续发送给设备，验证设备是否会生成异常。异常攻击测试的测试过程繁琐，测试方法也自成体系，其测试理论也在不断发展，本文就不对其进行详细讨论。这里只是就我们在测试中容易疏忽，而设备也容易出问题的两个方面进行讨论：

动态显示各种表项

这是对设备内存保护健壮性的测试。在设备对表项进行操作，特别是多进程同时访问同一个表项时，如果对内存保护不够，很容易出现内存访问错误，其后果也是致命的。但是，对这种问题的测试方法相对比较简单，向设备加入大量路由和删除这些路由的同时，反复显示路由表、LSP表、FTN、ILM等各种表项。特别是在进行删除表项操作时查看这些表，很可能会由于保护不够，使指针指向了一块已经释放的内存块引起访问错误。



携带超长属性值的协议报文

前面提到过，MBGP在传递VPNv4路由时会携带RT等属性，也会携带其他扩展团体属性，根据我司设备实现，对Update报文中所携带团体属性的个数（长度）是有所限制的，对于其他BGP属性也有类似规格。但是友商设备就未必有相同的规格，我司设备如何处理这些携带超长属性的报文是值得讨论的，但是设备不应该因此产生异常。记得一个网上问题就是由于我司设备无法识别团体属性长度大于32的Update报文，导致BGP邻居反复振荡，后来我们将规格修改到64以后，设备工作正常。以后是否还会遇到类似的问题，还需要我们仔细测试发现。

性能测试

性能测试包括对设备支持协议各种规格、配置、转发性能方面的验证。通常在进行性能测试时会使用到各种测试仪器和测试工具软件，也具有自己的测试方法论，将有其他文章对如何使用仪器测试MPLS进行专门讨论。这里我们主要从协议角度介绍归纳几个主要测试方面。

首先是配置规格测试。MPLS配置规格主要包括：BGP、LDP等协议邻居数目规格，VPN实例数目及其绑定接口数目规格，VPN实例支持RT数目规格，静态LSP规格，L2VPN对等体规格等。测试配置规格时不应该只关注是否能够完成配置，还应该验证配置是否生效，配置生效后设备功能是否正常，是否能够正确去掉这些配置，去掉配置后对应资源是否及时释放，反复配置是否存在内存泄漏等相关问题。并且设备在规格配置内，正确配置的各种协议、邻居是否能够正常建立，同时存在一定流量时设备是否依然能正常工作。

其次是路由相关规格测试。虽然在路由

协议层面上设备支持路由的规格没有具体限制与内存相关，但是对于MBGP，特别是对于MPLS L3VPN应用，PE不但要负责转发路由，还需要为路由分配标签，所以对应路由规格实际还会收到各种表项长度限制。同时，LSP表长度、NHLFE表等这些表的建立都与路由相关，也都是通过路由生成。对这些规格测试同样需要验证表项建立、删除操作和内存泄漏等方面。

最后是转发性能测试。对于MPLS转发性能测试，通常手段包括ping大包和利用测试仪器打入大流量报文。这里特别需要指出得是，我们在测试转发性能时通常就只使用SMB一种仪器（如SmartWindow/SmartFlow等），还需要多使用如Chariot等状态流测试工具。状态流能更加真是反映实际网络状态，其结果也才更准确地反映设备的性能。

互通测试

提到互通测试，我们总会马上想到与Cisco、Juniper等友商设备之间的互通、协同工作，其实这里还应该包括我司不同产品间、相同产品新老版本间以及我司与华为NE设备间的互通。。这些差异很可能会引起某些功能无法正常运行，这都需要我们能够提前发现这些不同，修改或找到其他解决的办法。

进行互通测试最基本的方法就是在测试环境中加入其他设备（包括其他厂商设备和允许其他版本的我司设备），共同构成测试网络实现一个功能。由于MPLS互连协议众多，各种应用网络拓扑比较复杂，而且还存在PE、P、ASBR等不同角色，在互通测试过程中，需要不断变化被测设备与互通设备之间关系、相对角色。

MPLS应用测试方法描述

MPLS技术能够得到日益广泛的应用，其原因本不是因为其转发技术本身有多么大的优势，特别是在计算机硬件技术快速发展的今天，路由器和交换机的IP转发效率已经能够达到甚至超过MPLS转发，何况由于MPLS转发需要多个基础协议作为实现基础，在很多实际测试中相同路由器的IP转发效率往往还会优于MPLS转发效率。为MPLS技术赋予旺盛生命力的是其具有良好的灵活性和易扩展性，能够使我们利用这个技术实现更多、更丰富、更灵活的应用，满足了服务提供商和用户两个方面对网络的需求。这些应用主要包括：MPLS L3VPN、MPLS L2VPN和MPLS TE等几类。这些应用也是MPLS技术在实际网络中的存在形式。

既然，MPLS都是以各种应用形式部署于各大网络中，对MPLS的测试也必然需要包括对这些应用的验证。与上面讨论对某个具体协议的测试不同，这些应用都没有专有协议相对应，RFC中只介绍了这些应用的基本框架和为实现这些应用而在各个协议进行的一些扩展，主要是对LDP、BGP和各种路由协议进行扩展。那么在对这些MPLS应用测试时，我们需要从这些协议扩展功能和各种应用本身功能两个方面进行验证。下面我们将对MPLS几种主要应用分别进行讨论，由于MPLS TE应用本身理论、相关协议和实现都相对比较复杂和独立，这里不作介绍。

L3VPN测试方法

MPLS L3VPN应用是目前在网络中使用的最广泛的MPLS应用。服务提供商通过为各个VPN网络维护、转发用户路由，并利用MPLS转发技术实现

各个VPN Site之间报文转发，从而达到将位于不同物理位置的用户网络连接为一个整体的最终目的。在PE上为不同VPN分别维护各自独立的路由表，为VPN用户网络提供了必要的私有性和安全性。

同时，MPLS L3VPN也是最为复杂、发展最为迅速的一种应用。为了不断满足各种用户多样性的网络应用需求，L3VPN本身也在不断变化，解决方案逐步丰富、完善。目前已经有包括基本L3VPN组网、跨域VPN组网、运营商的运营商（CSC）、嵌套VPN组网、分层PE组网、MCE、Hub-Spoke等多达十几种的网络拓扑结构。这些组网的基本结构相同，仅在某些设备（PE或CE）上实施的配置、所处位置和交互协议上有所不同，因此对这些不同组网模式的测试方法也没有太多变化。

L3VPN最初的应用是服务提供商在自己的网络内部（一般属于同一个自治系统）提供的一种VPN服务，为用户不同站点或不同用户（需要通信）之间提供安全的数据传输业务。因此，最基本的L3VPN组网就是L3VPN中所有的P和PE设备都处于服务提供商在自己的相同的自治系统内部。

对于基本L3VPN组网，根据设备的不同位置和作用分为P、PE和CE设备。在测试过程中，对不同位置的设备要进行有重点的测试：

P设备主要应用于电信级运营商或者大企业网络的骨干位置，其上承载着整个网络上的所有路由，路由容量大，对性能的要求也非常高。因此对于P设备，其测试重点首先是L3VPN流量转发能力（其实也就是MPLS报文的转发能力）以及转发延迟和稳定性等方面。可以通过使用各种测试仪器长时间持续打入大量L3VPN的路由和来进行转发能力及转发延迟的测试，通过长时间运行进行拷机测试。此外，在进行上述测试过程中执行shutdown/



undo shutdown物理接口、reset ospf、reset bgp、reset ldp进程、包括reboot系统等各种异常情况测试。

PE设备处于服务提供商网络边缘连接用户网络，由于用户网络协议配置和上连链路类型存在的多样性，并且PE需要为不同VPN分别维护独立的路由表，因此，对于PE的功能和性能要求是比较高的。PE设备的测试重点在几个方面：首先，也是最主要的测试内容就是PE与CE之间采用各种多实例的测试。包括在各种不同的网络接口类型（Frame Relay、ATM、E1、T1、E3、T3、FE、GE、POS、GRE等）上运行各种多实例（BGP多实例、OSPF多实例、RIP多实例、IpSec多实例、NAT多实例、ARP多实例等）。其次，测试MPLS L3VPN的表项建立和刷新情况。通过命令检查所有相关表项的建立、通过执行shutdown/undo shutdown私网接口、绑定/去绑定存在的VPN实例、绑定/去绑定不存在的VPN实例、重置各种多实例进程等异常操作，测试L3VPN各个表项的刷新/删除/重新建立的有效性和时间。再次，测试L3VPN用户网络不同站点之间各种应用，包括Ping（大包、小包）、Telnet、Ftp、HTTP、各种组播应用（语音、视频）、IPSec、L2VPN等的功能和性能，对于出现的问题，通过打开调试信息检查MPLS报文转发时的标签压栈、交换和弹出操作是否正确。

此外，PE之间包括PE与P之间的不同组网结构的测试，主要是MBGP的组网结构包括反射、联盟、多层次反射、反射和联盟混合结构和各种MBGP属性等的应用，这部分内容应该属于BGP测试的内容，在进行L3VPN测试的时候也要关注一下，这里需要特别注意的是地址重叠的情况，因为L3VPN需要定义VPN实例，通过BGP协议Update报文中携带的RT属性匹配远端VPN实例。配置两个VPN实例，采用相同的或者重叠的地址空间并不断的修改VPN实例的RT值并使BGP不断的发送Update报文，进行重叠地址空间情况下的异常测试。

CE设备主要应用于用户网络的边缘，用于

与运营商网络的接入。在L3VPN的实现中，CE设备作用相对小一些，不用作太多测试，重点考察下L3VPN不同站点CE之间的各种应用的功能和性能。

对于一些特殊应用的L3VPN应用，包括跨域VPN组网、运营商的运营商（CSC）组网、嵌套VPN组网、分层PE组网、MCE、Hub-Spoke等，大体上的测试方法同上，但是针对不同的组网结构或应用模式的特点，也要有相应的测试手段。

跨域VPN组网模式是L3VPN基本应用的一个扩展，VPN用户分布在不同的自治系统中。目前支持三种跨域VPN的实现方式，第一种背靠背方式其实就是相邻的两台ASBR互联的接口都绑定在VPN里，这种方式并没有什么特殊的内容，测试中可以进行一些异常情况测试，比如在ASBR上执行接口下绑定/去绑定VPN实例、shutdown/undo shutdown接口等简单测试项（其实这部分测试前面已经提过）并测试跨域的用户网络之间的各种应用，包括Ping（大包、小包）、Telnet、Ftp、HTTP、各种组播应用（语音、视频）、IPSec、L2VPN等的功能和性能。

跨域VPN的另外两种实现方式都是通过MBGP支持实现的。第二种实现方式在ASBR之间建立MBGP邻居关系，通常也称为单跳MP-EBGP方式，根据ASBR之间转发VPN路由时是否改变下一跳又分为两种情况。第三种实现方式在跨域的P之间或者PE之间建立MBGP邻居关系，通常称为多跳MP-EBGP方式。由于这两种实现方式都是通过MBGP支持，只是实现原理有点儿差异，因此测试中的方法也基本一样。首先要对这两种实现方式的三种情况重点测试PE/P和ASBR上BGP对VPN路由的标签分配和转发时的标签操作，通过在PE/P、ASBR上检查各个相关表项信息和打印调试信息进行这方面测试。其次，要重点测试各种异常操作引起的PE/P、ASBR的相关表项的刷新（删除、重新建立）情况和时间，包括重置BGP、OSPF、LDP进

程、重启系统、shutdown/undo shutdown接口等异常测试。此外，要重点测试跨域L3VPN的不同用户网络的各种应用：包括Ping（大包、小包）、Telnet、Ftp、HTTP、各种组播应用（语音、视频）、IPSec、L2VPN等。

运营商的运营商（CSC）组网模式就是将基本L3VPN组网模式中的CE作为下一级L3VPN应用的P或者PE从而实现将L3VPN应用进行分级的效果，通常也称为CSC（Carrier Support Carrier）方式。目前我司已经支持分层设备之间IGP多实例和LDP多实例的配置实现（BGP多实例的方式正在调试）。因此对于这种VPN组网应用模式，在测试中要重点测试作为层次边缘的设备IGP多实例和LDP多实例。由于IGP多实例在基本L3VPN组网中已经作为测试重点，因此在CSC组网中要重点测试通过在各种类型接口（Frame Relay、ATM、E1、T1、E3、T3、FE、GE、POS、GRE等）上LDP多实例以及通过异常操作测试LDP多实例的表项刷新/删除/重新建立过程。此外，在各台设备上打开调试信息测试报文转发过程中的标签交换过程也是必须的。还要重点测试CSC组网结构中的应用情况：包括Ping（大包、小包）、Telnet、Ftp、HTTP、各种组播应用（语音、视频）、IPSec、L2VPN等。

嵌套VPN组网也是将基本L3VPN组网进行分层，结构上和前一种组网模式（CSC）基本一样，只不过上级的PE要负责维护整网的VPN路由，而CSC结构中不需要。我司设备暂时还不支持这种实现方式。

分层PE、Hub-Spoke、多角色主机和MCE等几种L3VPN组网应用模式并没有特别的地方，按照配置进行功能测试，测试方法参照前面的描述。

L3VPN测试方法

与L3VPN不同，在L2VPN应用中服务商网络不负责维护、转发VPN用户网络路由信息，只负责

封装、转发所有由VPN用户接口接收到的数据报文。因此，可以将MPLS网络看作是连接各个用户CE设备的物理线路，用户所有数据都被透明地传递到远端CE设备上，不需要对其进行分析、处理和修改。这种应用一方面减小了对PE设备内存、处理能力的要求，另一方面却为其数据转发能力和接口类型方面提出了更高的要求。相对于传统VPN，具有更便于部署、管理维护和扩展的特点，只是由于对用户两端链路类型和链路层协议有一定要求，所以目前还没有得到广泛应用。

作为L2VPN应用，服务商为分布在各处的VPN用户网络提供二层连通性，及提供VLL服务，因此能否正确连接用户网络是测试中首要关注点。在测试中，我们建议尽量使用各种类型的物理接口，并在这些接口上封装各种类型的链路层协议如：PPP、FR、X.25等，同时配置各种常用的链路层应用，如PPP认证、地址协商、地址借用等。多种条件综合应用才能充分暴露问题。此外，基于链路协议的虚接口也是测试重点之一：MP、MFR以及各种类型子接口之间互连。验证CE之间连接是否正常的方法除了通常ping操作外，我们还可以在CE之间配置各种上层应用，验证这些应用功能是否正常，通常包括：Telnet、FTP、各种路由协议、组播应用、LDP、PPPoE、L2TP等甚至将CE配置为另一个MPLS域中的PE设备，在其上面承载其他MPLS服务等。可以看出，由于路由器支持板卡类型众多、而每种类型接口板又支持多种类型链路层协议，同时还需要为每一种情况配置多种应用，加上L2VPN本身实现方式又有不同，所以这些测试组合种类繁多、数目巨大，完成这些测试需要有极大的耐心和责任感。由于，L2VPN要求两端CE设备连接到PE设备所使用的链路类型必须相同，这对我们测试网络结构、规模和复杂性提出了较高的要求。同时，也需要测试人员对各种接口、链路层、上层应用以及他们相互之间的组合应用方式都比较熟悉，才能进行深入的测试。



上面描述的测试方法仅关注了PE、CE之间链路变化，是针对连接PE、CE之间和本地与远端CE之间网络的测试。对于MPLS域中PE和PE之间连接的测试则根据L2VPN实现形式不同而各不相同。就实现而言，目前我司设备支持4种方式L2VPN：CCC方式、SVC方式、Martini方式和Kompella方式。通常我们可以将前面两种认为是手动发现方式，需要手动配置PE间的连接关系和VPN对应关系以及手动分配VPN标签，而后两种是自动发现方式，它们分别利用LDP和BGP协议自动发现远端PE上的对应VPN实例并且协商标示VPN实例的内存标签。由于工作方式不同，对L2VPN的测试也可以分两种情况进行讨论：

手动发现方式L2VPN，由于没有任何其他协议参与协商，也没有任何自定义消息在PE间传递。在测试中我们只需要关注标签分配与释放、PE间LSP建立和切换等基本情况。测试方法于静态LSP测试方法类似，主要还是考虑与各种类型链路、LDP邻居振荡、多接口、多路由相互备份切换等各种手段相结合。同时，L2VPN支持在PE之间使用LSP和GRE（将来还会支持TE Tunnel）等多种隧道模式传递VPN数据（CCC方式除外），因此各种类型隧道之间相互备份、切换也是测试一个方面。

对于自动发现方式L2VPN，这里主要介绍对实现远端PE设备自动发现和VPN标签协商功能的LDP、BGP协议扩展功能的测试。Martini方式L2VPN使用remote方式LDP发现PE、根据配置的VPN id匹配VPN实例，并主动进行标签协商。那么remote peer方式LDP是测试重点之一，包括使用不同类型接口IP地址建立邻居关系，邻居关系振荡，带验证的邻居关系建立等。

LDP remote方式邻居关系建立后，并不会向对端发送任何FEC绑定的Map消息，而是会根据本地L2VPN的配置将VPN类型、VPN ID、本地为这个VPN分配的标签和其他选项数据一起封装在LDP消

息里面发送给对端，消息收到放设备会将消息中携带的VPN ID与本地配置相对比如果有相同ID，则取出消息中携带的标签信息和对端设备loopback对于LSP一起填写到当地L2VC表项下发给LSPM模块，形成对应转发表项。因此，LDP对L2VPN标签消息发送，设备对收到标签消息处理也是我们需要测试的重要环节，设备往往会在LDP连接出现振荡时，丢失LDP消息或由于本地配置改变而对LDP消息处理产生错误。我们在测试中可以采用不断改变L2VPN配置、修改VPN ID和LDP邻居关系等方法验证设备在网络动态变化时是否能正确处理消息。

Kompella方式L2VPN配置与L3VPN类型，需要定义VPN实例，并利用BGP协议Update报文中携带的RT属性匹配远端VPN实例。同时，支持一次性为一个Site网络分配一块标签资源，具有很好的网络扩展性，可利用BGP路由的反射器等特性完成全网PE设备自动互联具有很好的可维护性等优势。但是其配置则稍显复杂，配置项中各个参数不易理解。

由于Kompella方式具有以上特点，我们主要测试下面几个方面：首先，MBGP对L2VPN能力支持，是否能够正确建立对应能力的BGP邻居关系，同时还可以结合其他BGP实现和组网结构（如反射等）同时存在MBGP其他能力等结合进行测试，特别是对标签资源分配、收回等进行测试；其次，对L2VPN实例中配置的RD和RT匹配规则进行测试，测试方法与L3VPN中类似，可以通过完全匹配、部分匹配和空配置、不匹配几个方面验证，动态修改配置等手段结合测试；再次，是对VPN标签池再分配能力的测试，由于Kompella方式L2VPN支持动态扩展为每个CE预留的标签空间，而标签通过标签池基准标签大小加上偏移量和CE ID计算得到，标签池扩展对标签计算会产生很大影响，所以测试中应该有意不断修改标签池大小，不断修改本地和对端CE ID值验证设备对标签计算是否正确；然后，对CE ID匹配进行测试，不同实例CE ID是否能够重

复使用，CE ID匹配是否正确等；最后，也是最重要的是对标签计算的测试，Kompella方式L2VPN的标签是通过计算得到的，所以设备在不断收到Update报文时很可能由于计算错误使标签不匹配，引起MPLS转发不通。

VPLS测试方法

VPLS是另外一种MPLS二层VPN的应用，和VLL不同它应用方式是多点对多点。VPLS为许多原来使用点到点L2VPN业务的运营商提供了一种更完备的解决方案，通过VPLS服务，地域上隔离的用户站点能通过MAN/WAN相连，并且两地连接效果像在一个LAN中一样。

IETF相应的一系列草案中描述了使用MPLS的虚链路作为以太网桥链路的VPLS解决方案，通过MPLS网络提供一种透传的LAN服务。

所以在VPLS测试中对链路要求很简单，就是以太链路，但是其组网要求复杂。为了体现多点对多点，最基本的配置也是3台PE两两直连，每个下挂1台CE。

VPLS测试应该覆盖如下内容：

1、VPLS相关的草案中提供了两种基本VPLS网络架构：“虚链路(PW)逻辑全连接”的VPLS网络架构和“分层”的VPLS架构。这是VPLS最基本的两种组网应用，要分别验证这两种组网下的VPLS功能；

2、PW信令协议验证：PW信令协议是VPLS的实现基础，用于创建和维护PW，同时也可用于自动发现VSI的对端PE设备。目前，PW信令协议主要有LDP和BGP。我司都支持，所以两种信令协议都要一一测试到；

3、MAC地址学习与泛洪测试：VPLS组完最终在客户端看到仿佛就是一个大的LAN，所以它能支持MAC地址学习、老化、回收功能，要一一测试到；

4、环路检测测试：VPLS中，使用“全连接”和“水平分割转发”来避免在ISP网络上跑用户STP协议，“水平分割转发”是从指从公网侧PW收到的数据包不再转发到其他PW上，只能转发到私网侧。我们要从用户角度验证在自己L2VPN私网内运行STP协议是允许并正确的。正常情况下，所有的STP的BPDU报文都必须在ISP网络上透传。

5、在分层PE组网中测试中还需关注：UPE可以以QinQ接入NPE，此时NPE上对应实例的接入方式应为VLAN接入；我司还支持链路备份，此时需要在UPE与两个NPE之间启用STP来备份链路；

在分层PE组网中测试中还需关注：UPE可以以LSP接入NPE时，此时UPE可以以VLL、VPLS方式接入NPE，此时在NPE需要明确指明接入的设备为UPE；我司还支持链路备份，这种主备PW备份应用下需要指明NPE的主备关系；



结束语

我们同样应该了解在实际网络部署中，一个MPLS应用也不会孤立存在，它往往会根据用户类型、用户需求而不断变化，其组网模式和结构也不断变化。总的来说，用户需求不外乎包括：VPN用户访问公网资源、VPN网络之间资源互访与共享、VPN数据安全、VPN服务质量等几个方面。为满足这些需求，MPLS VPN通常与IPSec、防火墙、网管、NAT、QoS、策略路由、路由策略等其他应用方式组合实施，形成各种类型解决方案。所以，我们往往也会承担对各种解决方案的测试任务，包括：鉴定测试、综合组网测试、开局验证测试和网上问题复现等。对这些综合组网的测试需要涉及到众多模块、协议测试，已经超出了MPLS测试范畴，相信会在以后的文档中得到详细讨论。

测试方法

MPLS测试仪器

使用方法

李玉



MPLS相关测试内容

运营商关注的MPLS VPNs测试项

- 一致性测试：最新的标准和草案上规定的一致性测试(如RFC2547bis、RFC2283等)；
- 功能测试：
 - IGP问题怎样影响VPNs；
 - 运营商所需的功能是否支持；
- 性能测试：
 - 吞吐量、丢包、时延的测试；
 - 建立LSP的时间和路由振荡收敛时间测试等
- 规格测试：如VRFs、LSPs、CEs、PEs、VCs、MAC表大小等；
- 互通测试：
 - 不同设备提供商的产品之间互通测试

MPLS对DUT的测试要求

- 1、对于P和CE路由器的测试要求相对比较简单

P路由器作为MPLS网络中必不可少的组成部分也要进行MPLS的功能测试、一致性测试和性能测试，但是VPNs传递对他们是完全透明的，所以测试内容相对比较简单。

CE路由器不支持MPLS的协议，没有任何新的协议功能添加。

- 2、PE路由器是MPLS测试当中的重点

- 需要支持多个VRFs
- 需要MP-BGP维持IBGP full-mesh配置(或者使用route reflectors扩展)
- 需要使用MP-BGP广播VPN-IPv4路由

- 需要LSP label发布(LDP or RSVP-TE)
- 需要BGP Extended Communities属性
- 当收到CE发向Core的流量时，必须执行VRF查找和MPLS封装
- 从核心网向CE端转发流量时，必须弹出MPLS和VRF标签

相关的规格测试、性能测试、功能测试都是必不可少的。测试人员测试当中需要关注如下：

- 规格测试包括：同时可以建立的路由会话、虚拟链路、VRFs容量等；
- 性能测试包括：
 - 最大可支持的VRF数量
 - 每个VRF上最多可支持的路由数量
 - 转发性能 vs. VRF数量 和 每个VRF上支持的路由数量
 - 骨干网络MPLS协议(LDP vs. RSVP-TE)对其性能的影响
 - 最大可维持的MP-IBGP对等连接
 - 转发性能 vs. 路由的震荡(同一VRF)
 - 转发性能 vs. 路由的震荡(不同VRF)

当增加/删除一个VRF时，对当前VRF转发性能的影响

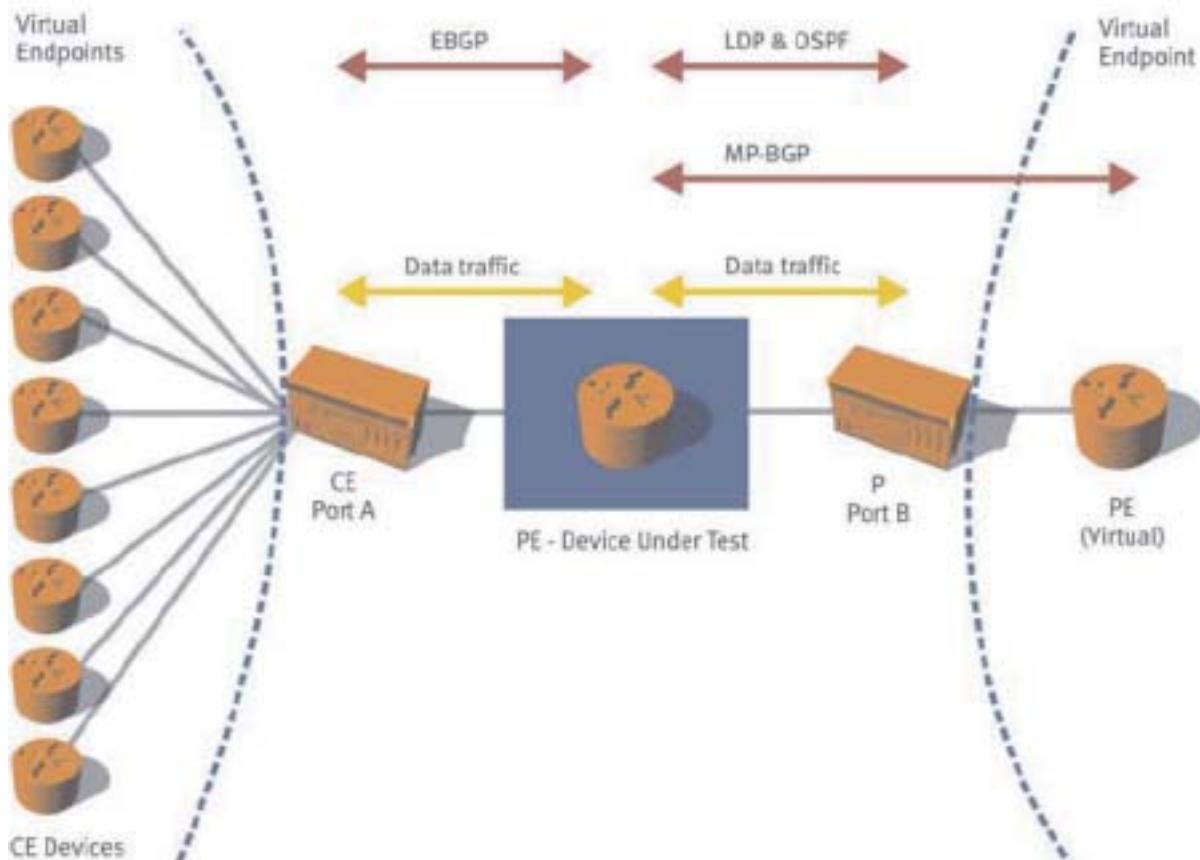
- 功能测试包括：
 - 保持VRF编号的唯一性
 - 正确转发VPN-IPv4路由给其他PE
 - 正确向不同的CE分配和发布标签
 - 正确向不同的骨干LSP分配和发布标签
 - 正确地接收和发送流量给CE
 - 学习IGP拓扑能力
 - 学习各种VPN-IPv4路由能力

MPLS 测试思想

BGP/MPLS VPN是使用三层路由来发布VPN的连通性。MPLS 的骨干网提供了VPN数据层面的连通。CE端路由器通过路由协议发布本站点的路由信息到直连的PE路由器上。PE路由器把这些路由放入在本机VPN的VRF中。为了把可达信息提供给远程的站点，PE把VPN路由发布到对端的PE路由器上。MPLS的很多性能规格测试都有相通性，只要了解测试当中的关键就很容易进行。

下面以BGP/MPLS VPN VRF 容量规格测试为例，介绍一下MPLS测试思想。

- 1、设置CE和PE（DUT）之间每个接口上的子接口数目（每个子接口仿真一个VPN站点）；
- 2、在每个子接口上提供和绑定不同的VRF：在DUT上设置VPN实例；设置RD；设置出和入的Route Target；绑定子接口和VPN实例；在DUT的核心网接口上设置MBGP；在核心网接口上设置OSPF和LDP；
- 3、设置CE-PE之间的每个子接口的路由协议（OSPF，RIP，EBGP，Static都可）；



4、发布一定数目的路由从CE到PE端（路由数目根据测试的具体设备来制定）；

5、监测通过MBGP session发布的路由。判断CE发布的所有路由全部发布到对端PE上；

6、如果被传播的路由数目不等于所发布的路由数目：记录VRF的个数和发布路由的个数；结束

测试；

7、如果被传播的路由数目等于所发布的路由数目：记录VRF的个数和发布路由的个数；构造从核心网到CE端路由的流量；判断是否可以正常转发所有的VPN的报文；结束测试；

8、改变测试参数重复进行如上的测试。

测	CE-PE之间的物理接口的数目；
试	每个物理接口的子接口的数目；
参	发布的路由数目；
数	

MPLS 测试仪器使用

支持MPLS协议测试仪器

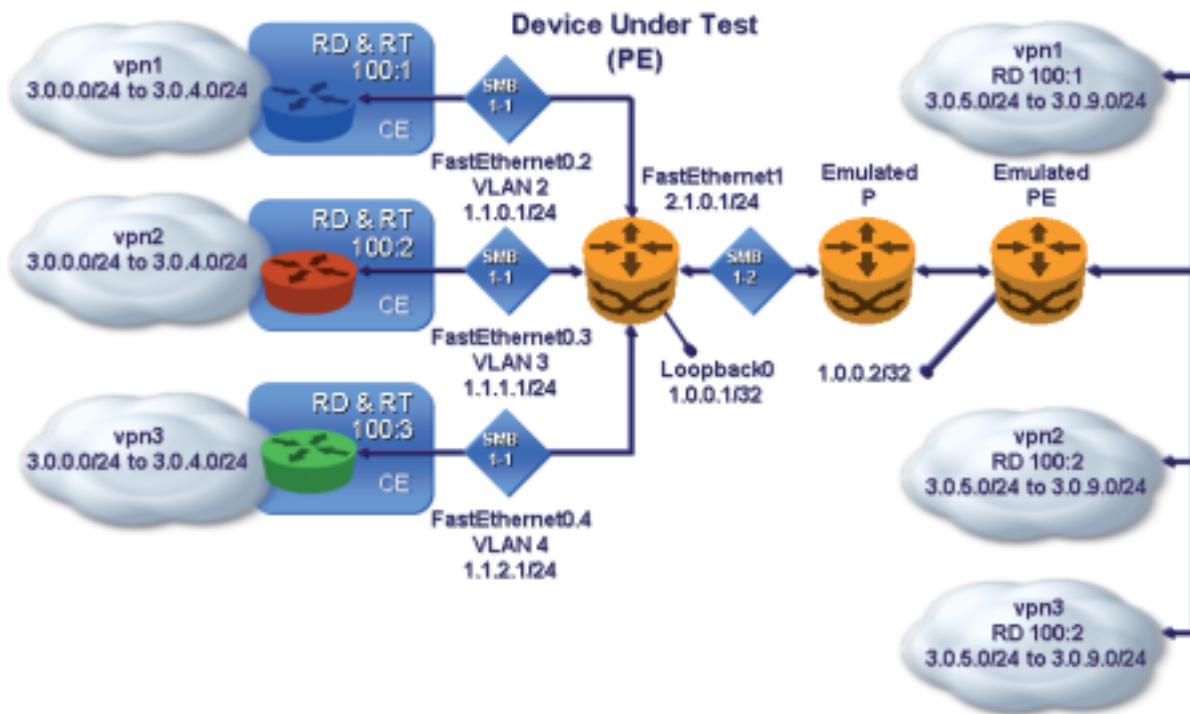
支持MPLS的测试仪器比较多，目前应用比较普遍的是Spirent的TRT，Agilent的Router Tester和IXIA的IXExplorer。使用安捷伦的设备测试MPLS以我个人看法比较麻烦，建议大家去使用TRT和IXExplorer。使用TRT和IXExplorer进行

MPLS的测试比较相似，都有Wizard（向导）方式和手工逐步配置方式。只要对其中的一种测试方法和测试理论有一定的了解就可以触类旁通了。

下面重点来讲一下使用TRT来测试MPLS的两种方式吧！

使用TRT测试MPLS测试方法

测试组网图



使用仪器很容易模拟如上的相对比较复杂的MPLS组网环境，对于使用者来说只要做简单的配置工作即可。在Tera Routing Tester上对于这样的工作提供了两种方式，一个是手工进行所有的相关配置（需要对协议非常了解），另一个是使用向导的方式进行。向导的方式十分简化了使用者的工作量和复杂度，使用起来非常方便。下面对两种方式都进行说明。

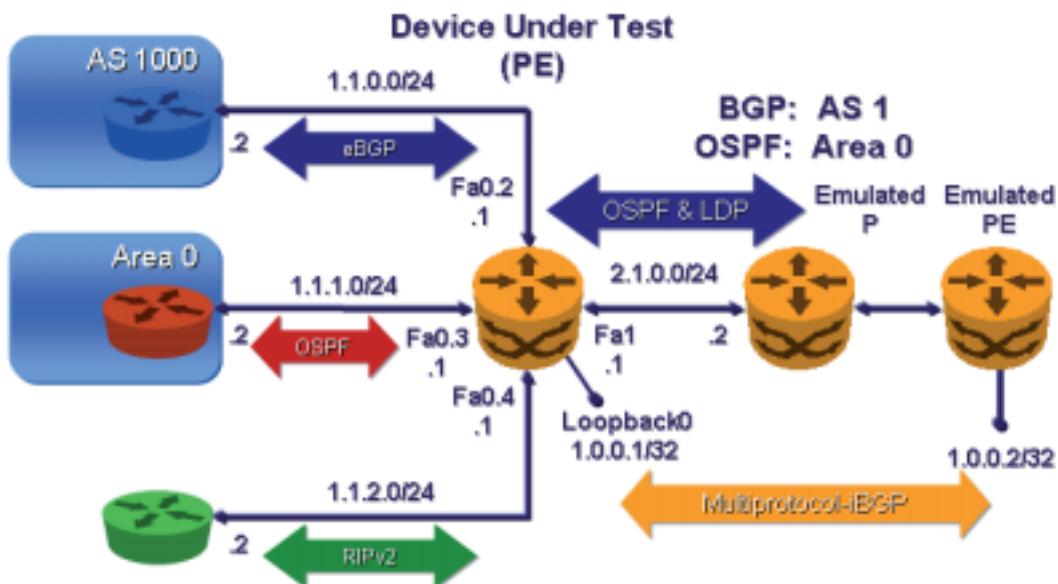
在MPLS网中测试的重点是PE设备，在如上组网环境中PE是DUT，其它的都由测试仪器来模拟。

把DUT的两个接口连接到SmartBits的两个端口上。一个端口模拟的是CE->PE的连接，另一个是PE->P之间的连接。

整个MPLS网络上有三个VPN网，并且每个VPN包含了10个仿真的路由（50%是CE-3.0.0.0/24 to 3.0.0.4/24，50%是PE端发来的-3.0.0.5/24 to 3.0.0.9/24）。

三个VPN的路由进行重叠，RD&RT分别是100:1/100:2/100:3。

对于每一个路由都发送双向的流量。



MPLS网络协议的说明如下：

CE->PE之间使用动态的路由协议：

VPN1-eBGP (AS1000)

VPN2-OSPF (Area0)

VPN3-RIP2

PE->P之间运行OSPF、LDP；

MBGP使用在DUT的loopback到对端PE之间；

DUT的设置

```
#创建VPN1-3的VPN-instance，并配置RD和
RT属性以控制VPN路由信息的发布
```

```
ip vpn-instance 1
```

```
route-distinguisher 100:1
```

```
vpn-target 100:1 export-extcommunity
```

```
vpn-target 100:2 import-extcommunity
```

```
#
```

```
ip vpn-instance 3
```

```
route-distinguisher 100:3
```

```
vpn-target 100:3 export-extcommunity
```

```
vpn-target 100:3 import-extcommunity
```

```
#MPLS基本能力使能
```

```
mpls lsr-id 1.0.0.1
```

```
#
```

```
mpls
```

```
lsp-trigger all
```

```
#
```

```
mpls ldp
```

```
#在DUT与P路由器相连的接口上使能MPLS及
LDP
```

```
interface Ethernet0/0
```

```
ip address 2.1.0.1 255.255.255.0
```

```
mpls
```

```
mpls ldp
```

```
#将DUT与CE相连的接口生成三个子接口，并
各自绑定到不同的VPN
```

```
interface Ethernet0/1
```

```
ip address 192.168.1.200 255.255.255.0
```

```

#
interface Ethernet0/1.1
  vlan-type dot1q vid 2
  ip binding vpn-instance 1
  ip address 1.1.0.1 255.255.255.0
#
interface Ethernet0/1.2
  vlan-type dot1q vid 3
  ip binding vpn-instance 2
  ip address 1.1.1.1 255.255.255.0
#
interface Ethernet0/1.3
  vlan-type dot1q vid 4
  ip binding vpn-instance 3
  ip address 1.1.2.1 255.255.255.0
#
interface LoopBack0
  ip address 1.0.0.1 255.255.255.255
#
#在PE与PE之间建立MP-IBGP邻居，进行
PE内部的VPN路由信息交换。
  vpn-target 100:1 import-extcommunity
#
  ip vpn-instance 2
    route-distinguisher 100:2
    vpn-target 100:2 export-extcommunity
  bgp 1
    peer 1.0.0.2 as-number 1
    undo synchronization
    group 1 internal
    peer 1.0.0.2 group 1
    peer 1.0.0.2 connect-interface LoopBack0
#
#在VPNv4地址族视图下激活MP-IBGP对等
体。
  ipv4-family vpnv4
                                peer 1 enable
                                peer 1.0.0.2 enable
                                peer 1.0.0.2 group 1
#
                                ipv4-family vpn-instance 1
                                peer 1.1.0.2 as-number 1000
                                import-route direct
                                group 2 external
                                peer 1.1.0.2 group 2
#
                                ipv4-family vpn-instance 2
                                import-route direct
                                import-route ospf 1
#
                                ipv4-family vpn-instance 3
                                import-route direct
                                import-route rip 1
#
                                ospf 1 vpn-instance 2
                                import-route direct
                                import-route bgp
                                area 0.0.0.0
                                network 1.1.1.0 0.0.0.255
#
#在PE1与P路由器相连的接口及loopback接口
上启用OSPF，并引入直连路由。实现PE内部
的互通。
  ospf 100
    area 0.0.0.0
    import-route direct
    network 1.0.0.1 0.0.0.0
    network 2.1.0.0 0.0.0.255
#
    rip 1 vpn-instance 3
    network 1.0.0.0
    import-route bgp

```



TRT操作过程

1. 手工操作方法

设置TRT Card Setup

设置和DUT匹配的媒体类型 (media type)、速率 (speed) 和全双工 (duplex) 模式；

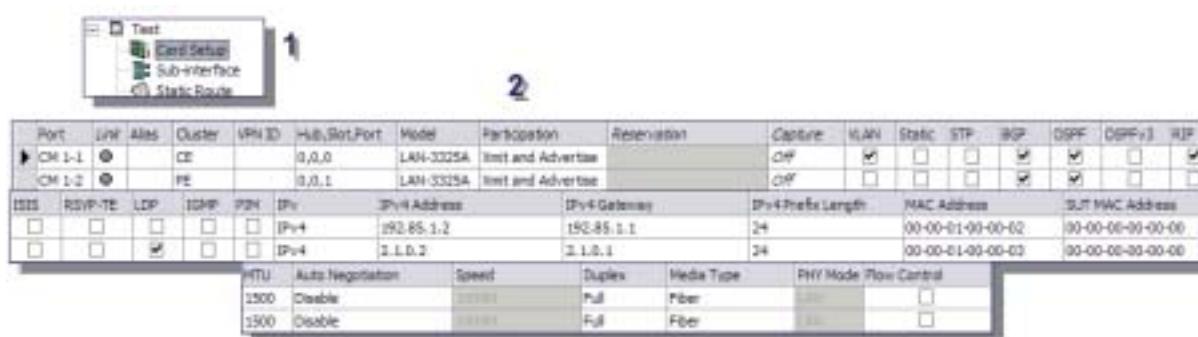
设置端口的角色 participation (在此两个端口都使用 Xmit and Advertise) ;

选择端口仿真的协议类型：

仿真CE端的端口上使能VLAN（这个端口需要扩展三个子接口）、BGP、OSPF、RIP协议；

仿真P端口的端口上需要使能BGP、OSPF、LDP协议；

注：使用VLANs的接口上不需要设置IP地址：

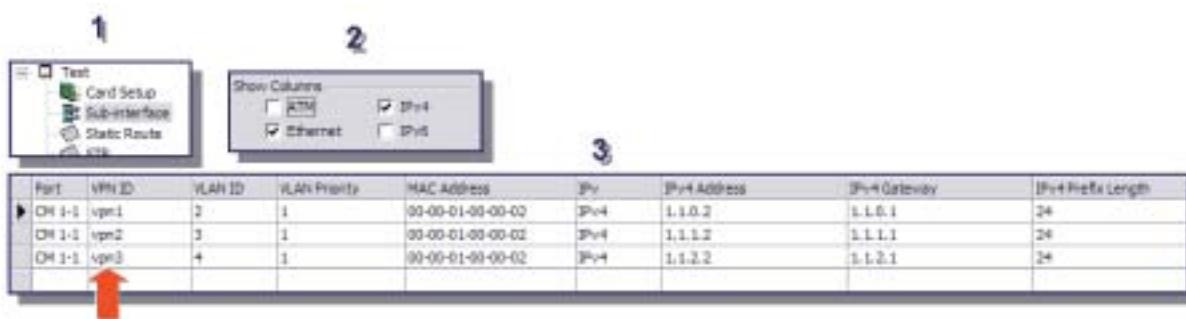


设置VLAN Sub-Interfaces

不同子接口的IP地址不可以相同的：分别设置1.1.0.2/1.1.1.2/1.1.2.2；

设置VPN ID：为了构造流量的时候映射到不同的VPNs，区分路由时使用；

设置VLAN ID：



设置MP-BGP Configuration: Sessions

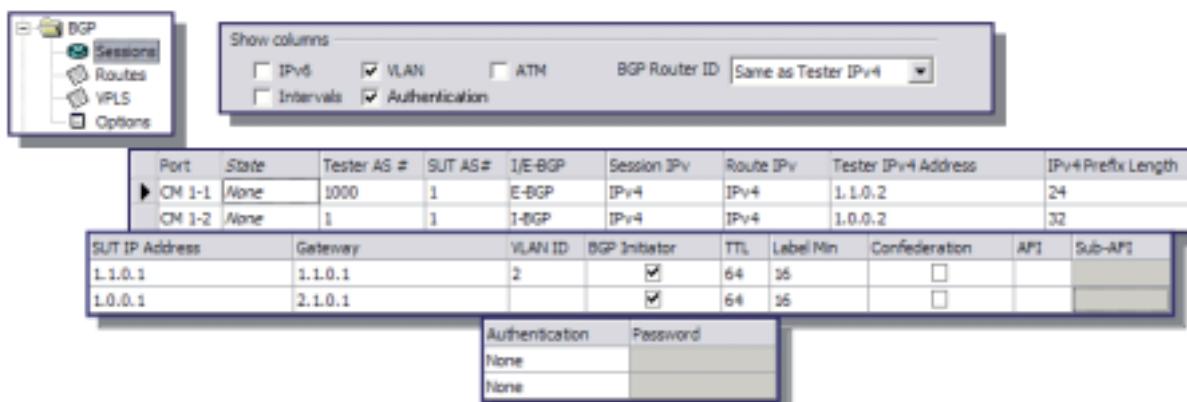
DUT和对端仿真的PE之间跑的是MP-BGP协议，入口PE路由器利用MP-IBGP穿越公网，把它从CE学到的路由信息发布给出口PE（带着MPLS标签），同时，出口PE把学到的路由信息发布到另一端的CE。

在PE和VPN1的CE端之间是通过EBGP session来传递路由的；

注：这里的Tester IPv4 Address是与BGP对等体建立session会话的IP地址，CE端是通过接口IP地址来进行会话的，因此Tester IPv4地址和SUT IP地址都是接口的地址，

IPv4 Prefix 是24位的；然而PE端之间是通过loopback接口进行会话的，Tester IPv4地址和SUT IP地址都是Loopback接口地址，IPv4 Prefix 是32位的；

对于CE端session因为使用了子接口需要填写VLAN ID；



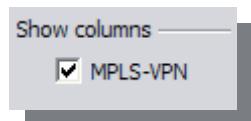
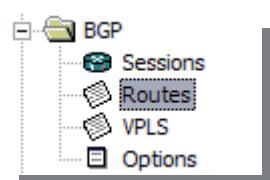
设置MP-BGP Routes

第一个端口是仿真了CE->DUT的VPN1的BGP路由；

第二个端口是仿真了PE->DUT的三个VPN的路由；

由于地址是重叠的，构造的三个路由段都是相同的，并且通过设置VPN ID和RD、Route Target来区分不同VPN的路由。

注：构造PE->PE路由时填写VPN ID，有助于构造流量时区分不同的VPN路由。



The screenshot shows a network configuration interface with two main sections:

- Route Block Configuration:** A tree view under "CM 1-1" and "CM 1-2" showing route entries. Under CM 1-1, there is one entry for #1: 1.1.0.2-1.1.0.1 AS 1000-1. Under CM 1-2, there are three entries for #1: 1.0.0.2-1.0.0.1 AS 1-1, all categorized as "Aggregate". A blue callout bubble with the text "Right-click to add/duplicate route blocks" points to the bottom right of the route block area.
- LSA Table:** A table showing Link State Advertisements. The columns are: Next Hop, Local Next Hop, MED, Local Pref, Atomic Aggr, Aggr AS, Aggr IP, Community, Originator, and Cluster List. The rows show LSA types: L.L.S.D., L.O.O.D., and L.O.O.D. for different IP ranges (1.1.0.2, 1.0.0.2, 1.0.0.2).

设置核心网IGP

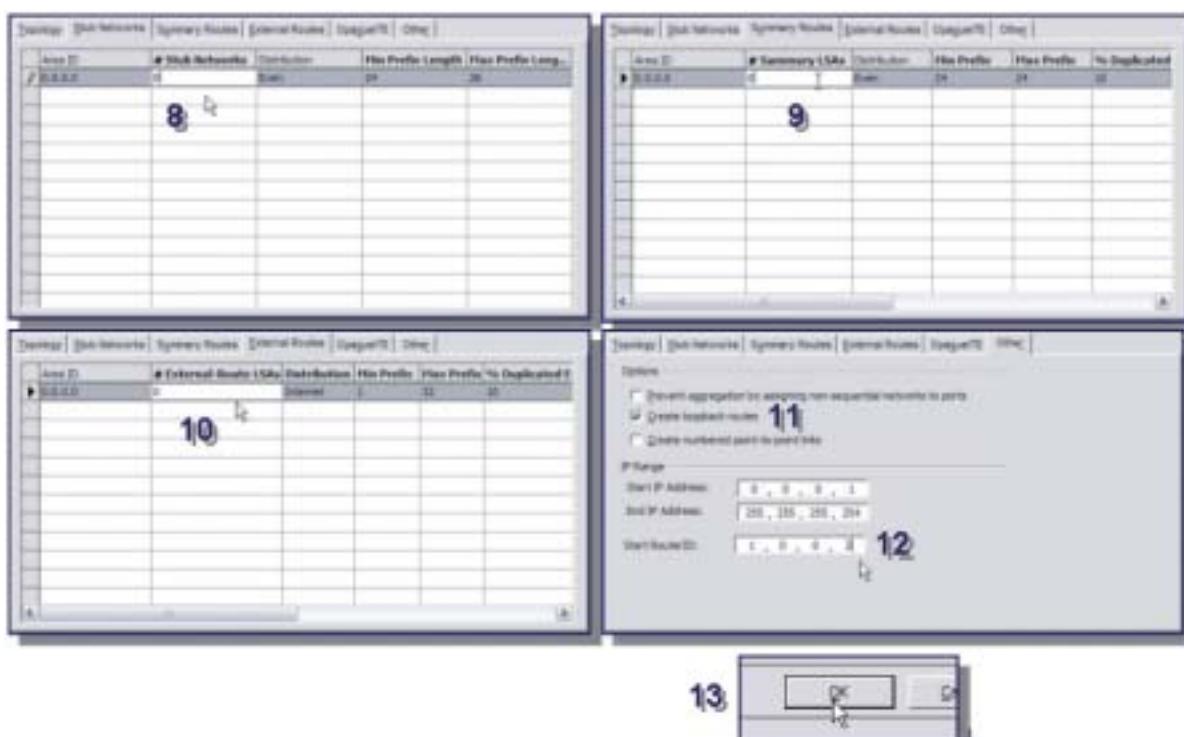
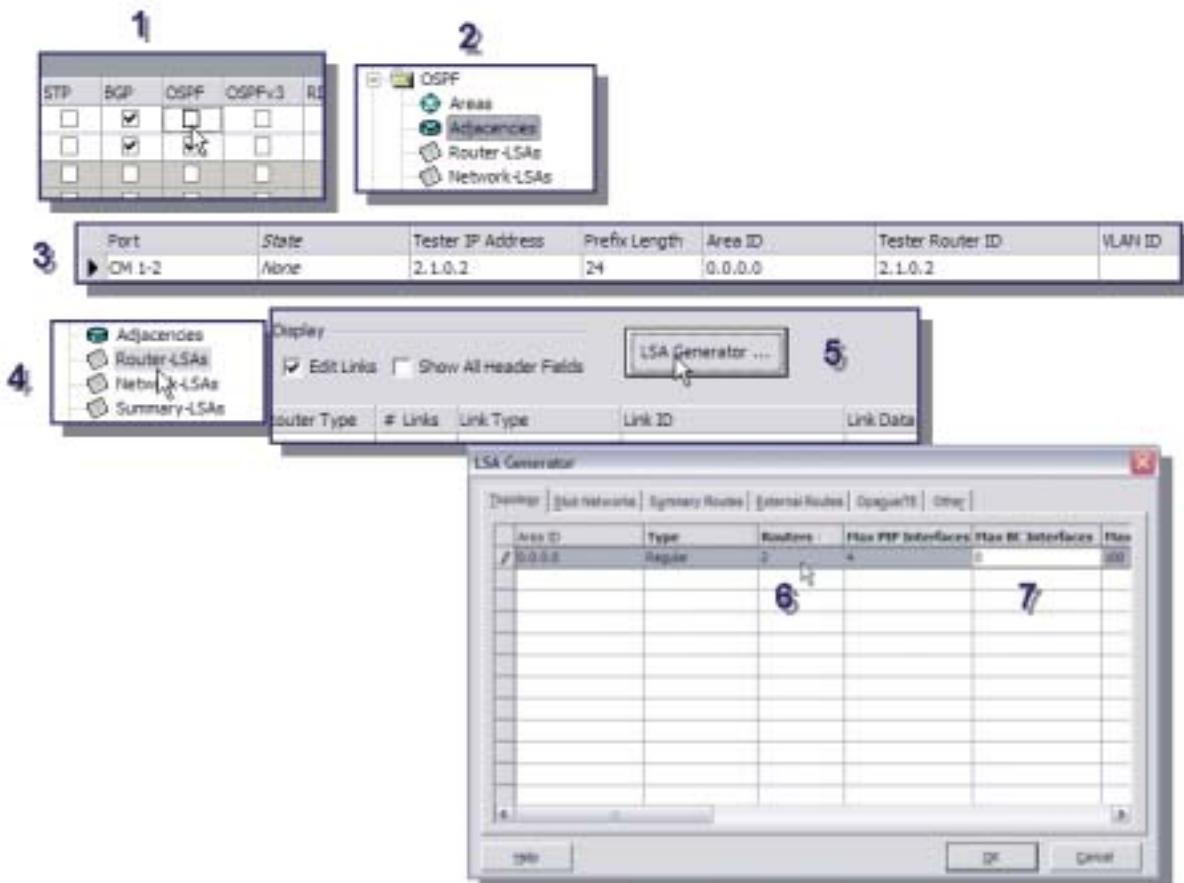
在核心网上PE-P之间是通过OSPF来传递路由信息的。

首先把CE-PE端口之间的OSPF协议使能去掉，仅仅保留PE-P端口之间的OSPF使能。虽然端口的OSPF的设置被隐藏了，但是这些设置都仍然保留着，这个功能有利于对于每个端口分别的进行

LSA的构造。

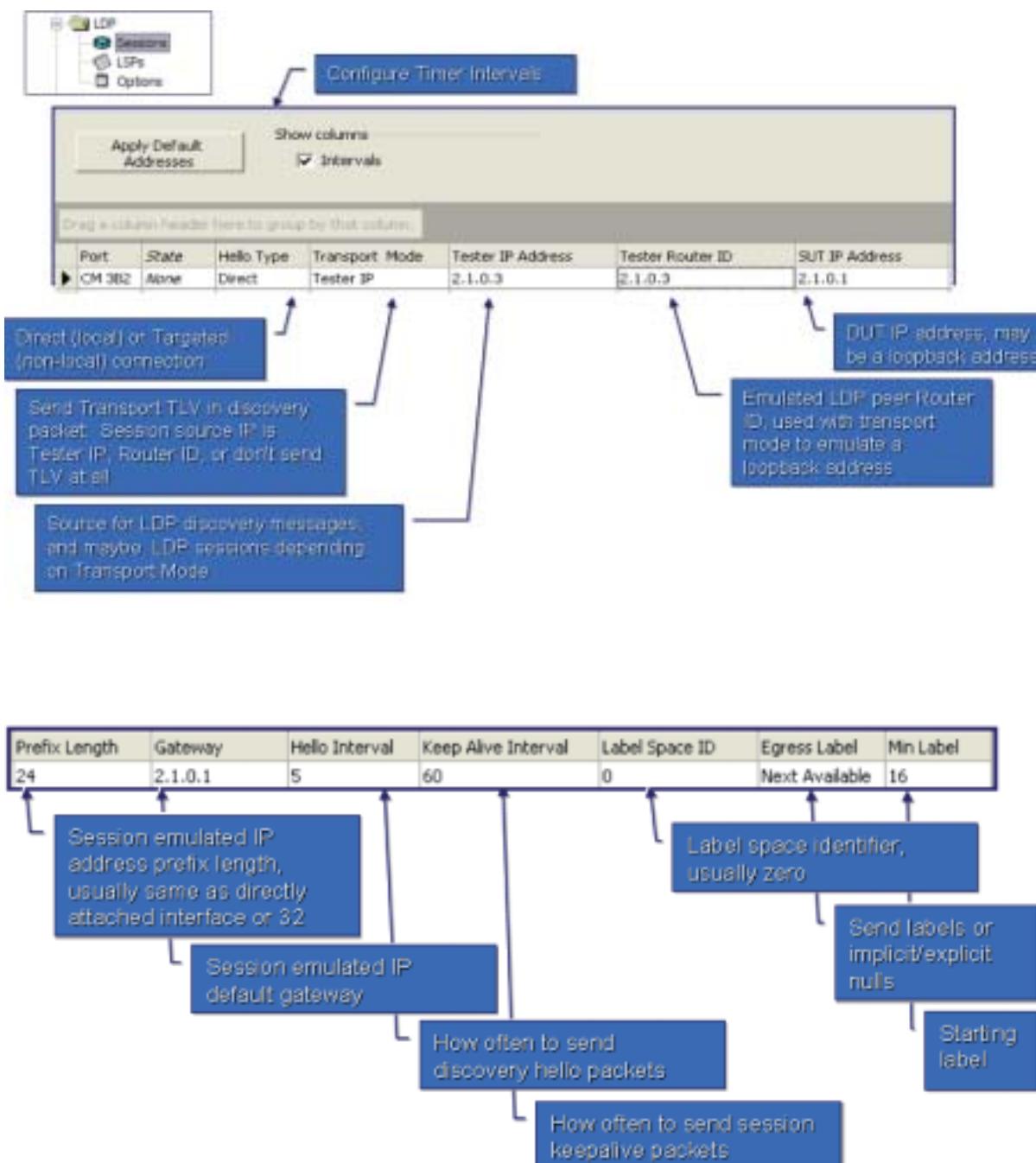
注：在核心网上要构造两个路由器的环境，也就是一个P和一个PE的环境，所以在拓扑中的Routers个数中设置2。

在测试MPLS时，需要选中other Tab中的Create loopback routers。



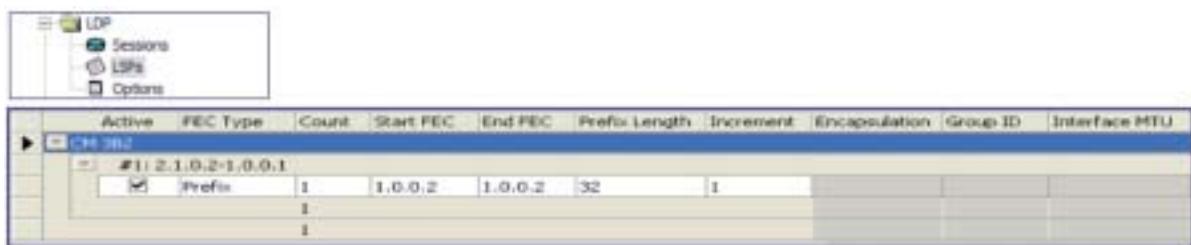
设置核心网LDP

设置LDP的session时传输模式可以使用仿真路由器的Router ID作为loopback接口。在Transport Mode中选择Router ID选项。



创建LSP标签映射

FECType定义了是否使用Prefix、host（IPv4）还是VC(PWE3/Martini/VPLS)；如果选择了VC FEC类型，可以进行不同的VC封装、Group FEC ID和MTU大小的设置；HostFEC基本上相当于使用32位的Prefix长度；



设置PE-CE 协议：RIP

RIP的配置比较简单，只要注意下面的几项即可。
版本选择RIPv2；
正确填写VLAN ID，在这里要填写4；
构造需要的RIP路由；

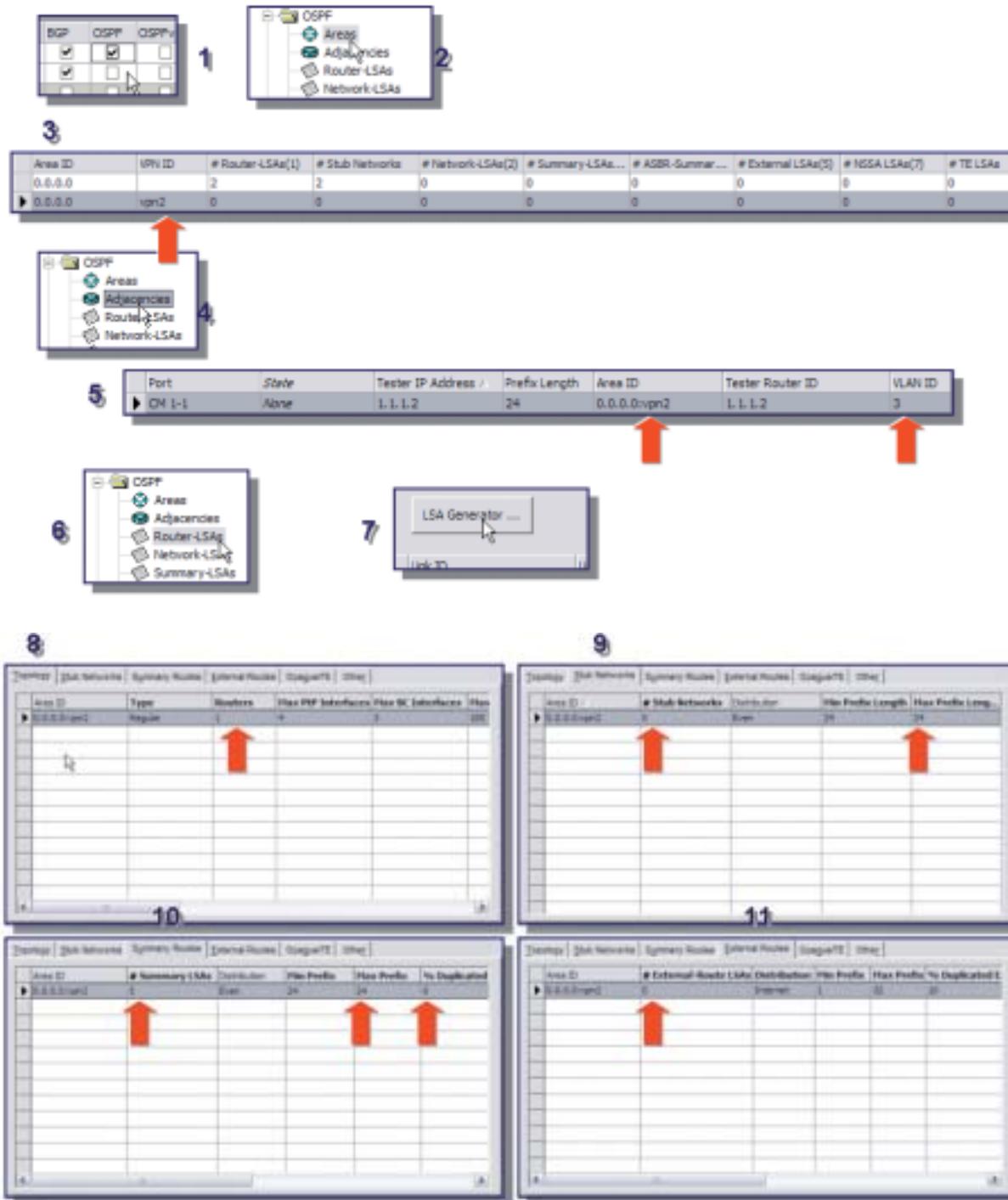


设置PE-CE 协议 : OSPF

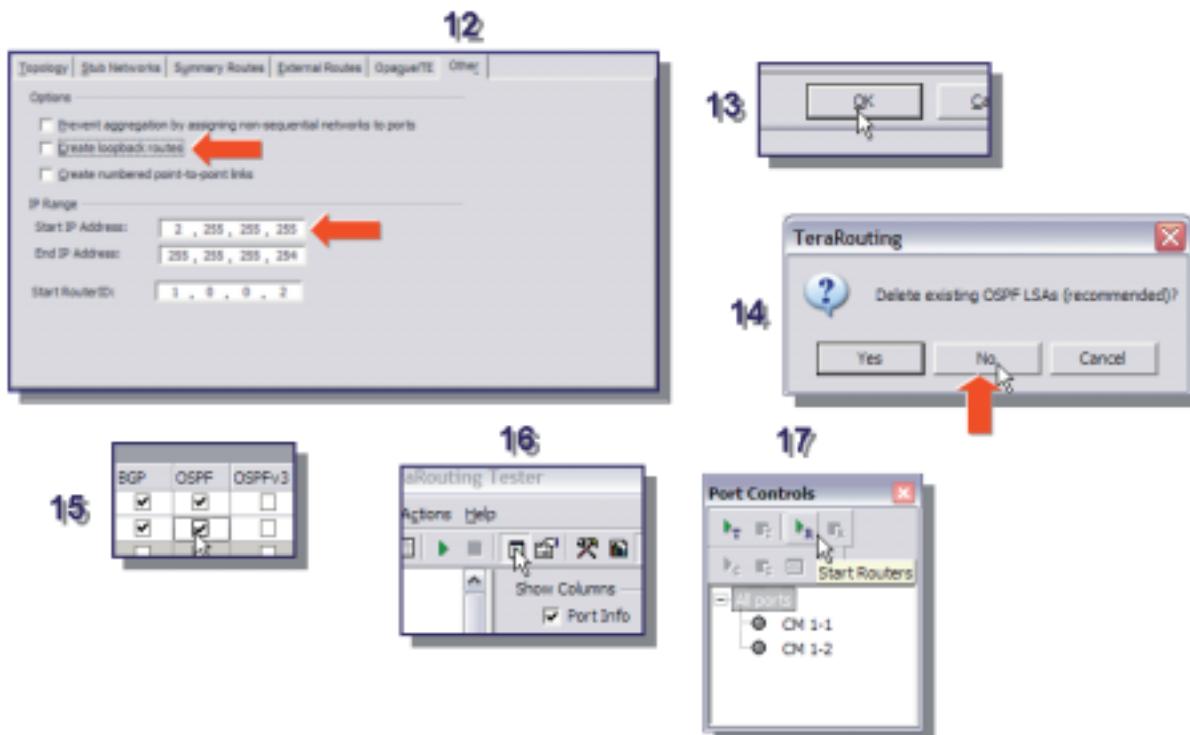
把PE-P端口之间的OSPF协议使能钩掉，仅仅保留CE-PE端口之间的OSPF使能。虽然配置隐藏着，但是刚才设置的核心网的端口配置仍然保留着。先把单独的OSPF配置完之后，再把所有的OSPF使能。

CE-PE之间的OSPF的路由属于VPN2的路由，所以先创建新的区域—VPN2的area0.0.0.0。

注：TRT不允许有两个全局的0.0.0.0，我们需要先填写VPN ID后再修改Area的ID。



在步骤10中填写Summery LSAs的个数为5，创建VPN2的5个路由。其它的LSA的个数都改成0。



在这里不需要点击create loopback routes；

TRT通过IP Range中设置的IP地址的起始地址来分配发布的路由。在Start IP Address中设置的第一个地址不可以使用，所以我们想发布的第一个路由是3.0.0.0的话，就要把Start IP Address配置成2.255.255.255。

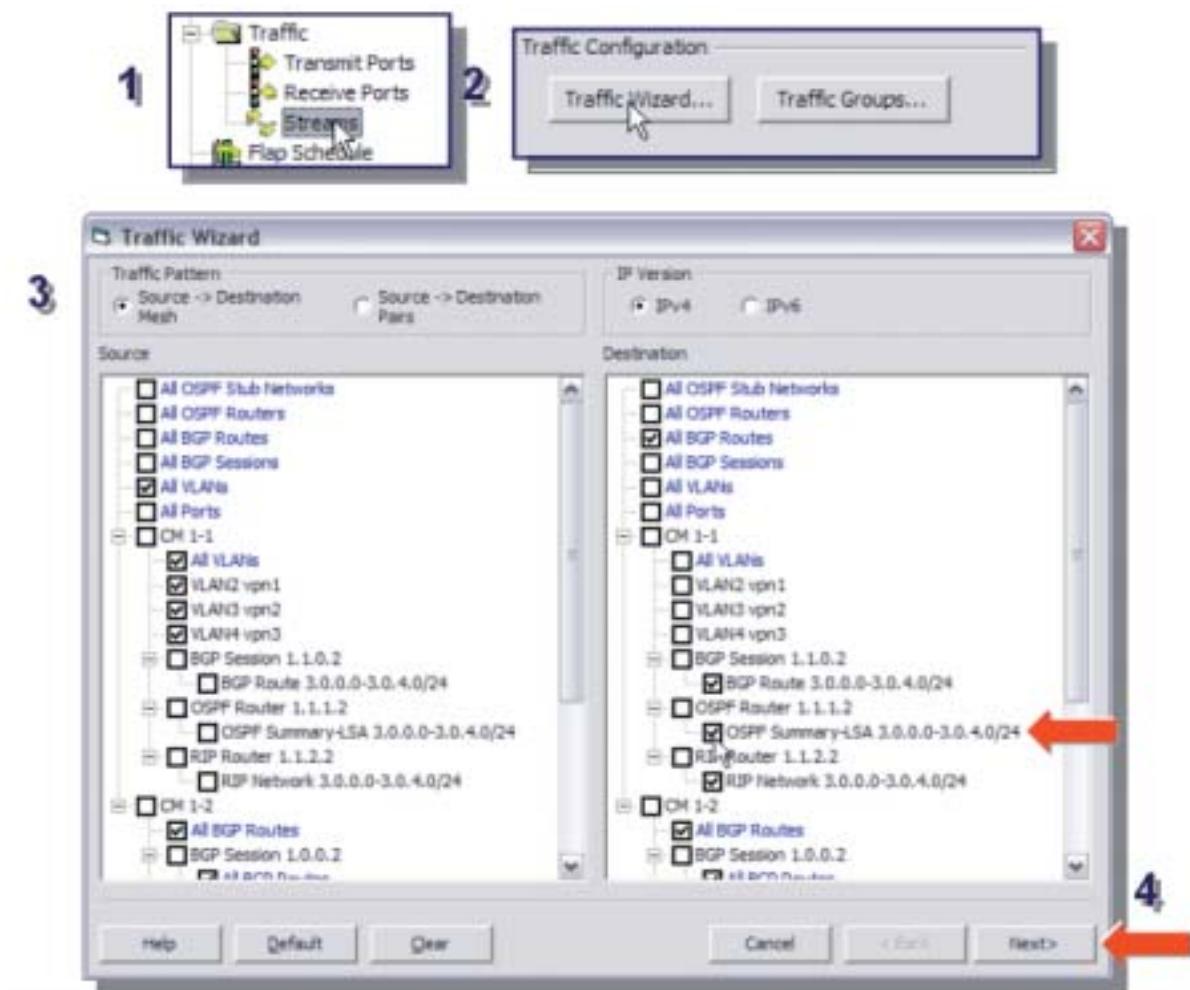
注：等我们配置完公网和私网的OSPF后，需要把两个端口的OSPF同时使能，这时刚刚配置的两个端口的OSPF信息都将显示出来。

构造流量

在TRT构造流量比较方便的方法是使用Traffic Wizard。由于我们的流量是双向的可以选择Source->Destination Mesh模式。

一部分流量是从CE->PE的，数据源应该选择Vlan2、Vlan3、Vlan4，目的选为CM1-2端口的BGP VPN路由。

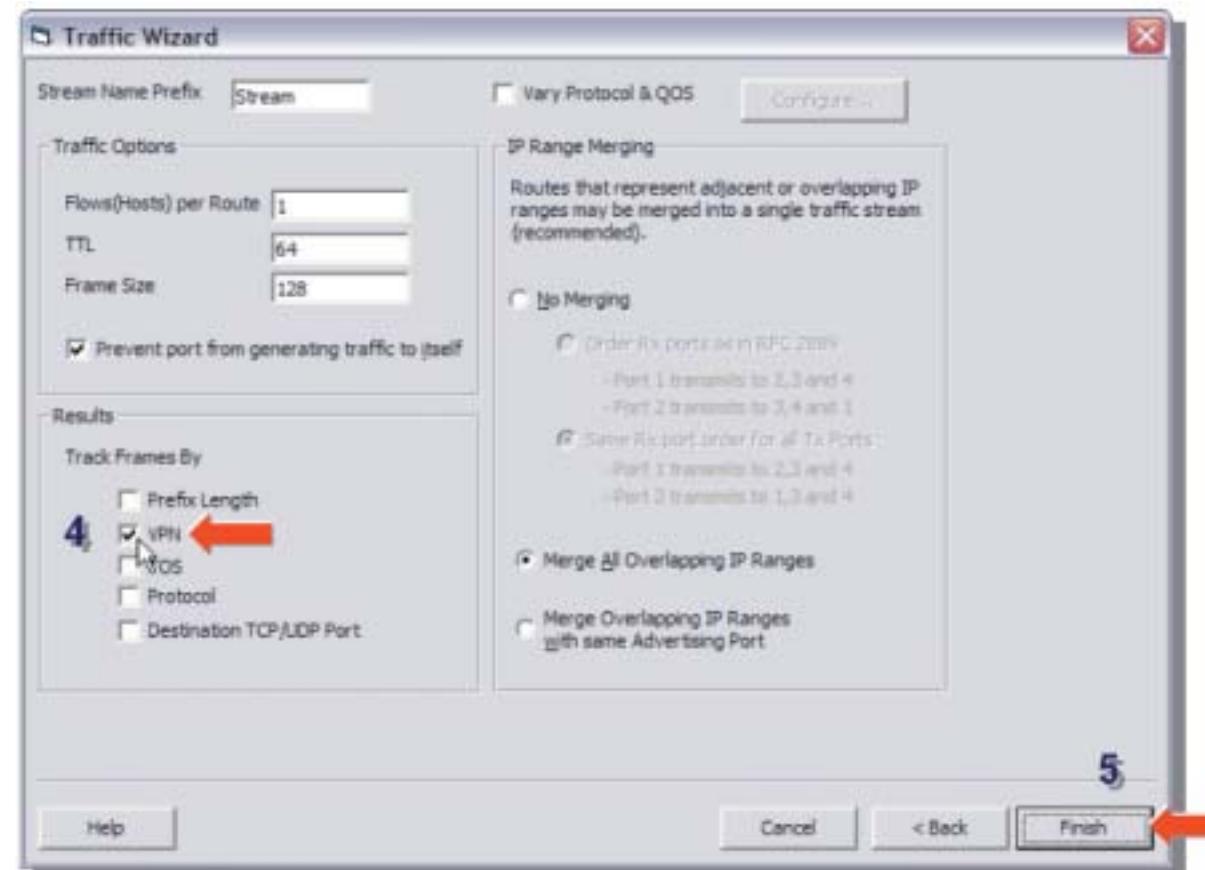
另一部分流量是从PE->CE的，数据源是CM-1-2端口BGP VPN路由到，CM1-1的BGP、OSPF、RIP路由。这些流量的选择都需要手工的进行。



为了对构造的流量通过VPN来区分，在显示界面上增加VPN的列表项，在设置时候点击VPN选项。如下步骤4中显示。

流量向导完成后，在界面中显示构造的流量，但是带标签的MPLS的流量有的时候是绿色的，却有的时候是红色的，为什么呢？这是

因为TRT目前的版本仅支持标签的连续分配，当流量多次被修改之后分配的标签如果出现了非连续分配，TRT将认为有错处理，这时候最简单的方法是使用右键点击Update MPLS Labels即可。但需要注意的是最好把被测设备的相关信息也清除掉。



CM 1-2

<input checked="" type="checkbox"/>	Stream5	vpn1	Eth EtherType 0x8847 MPLS IPv4=1.0.0.1/32 (Unresolved),TTL:15	MPLS VPNv4=
<input checked="" type="checkbox"/>	Stream6	vpn2	Eth EtherType 0x8847 MPLS IPv4=1.0.0.1/32 (Unresolved),TTL:15	MPLS VPNv4=
<input checked="" type="checkbox"/>	Stream7	vpn3	Eth EtherType 0x8847 MPLS IPv4=1.0.0.1/32 (Unresolved),TTL:15	MPLS VPNv4=
		Total:3		

Traffic Groups

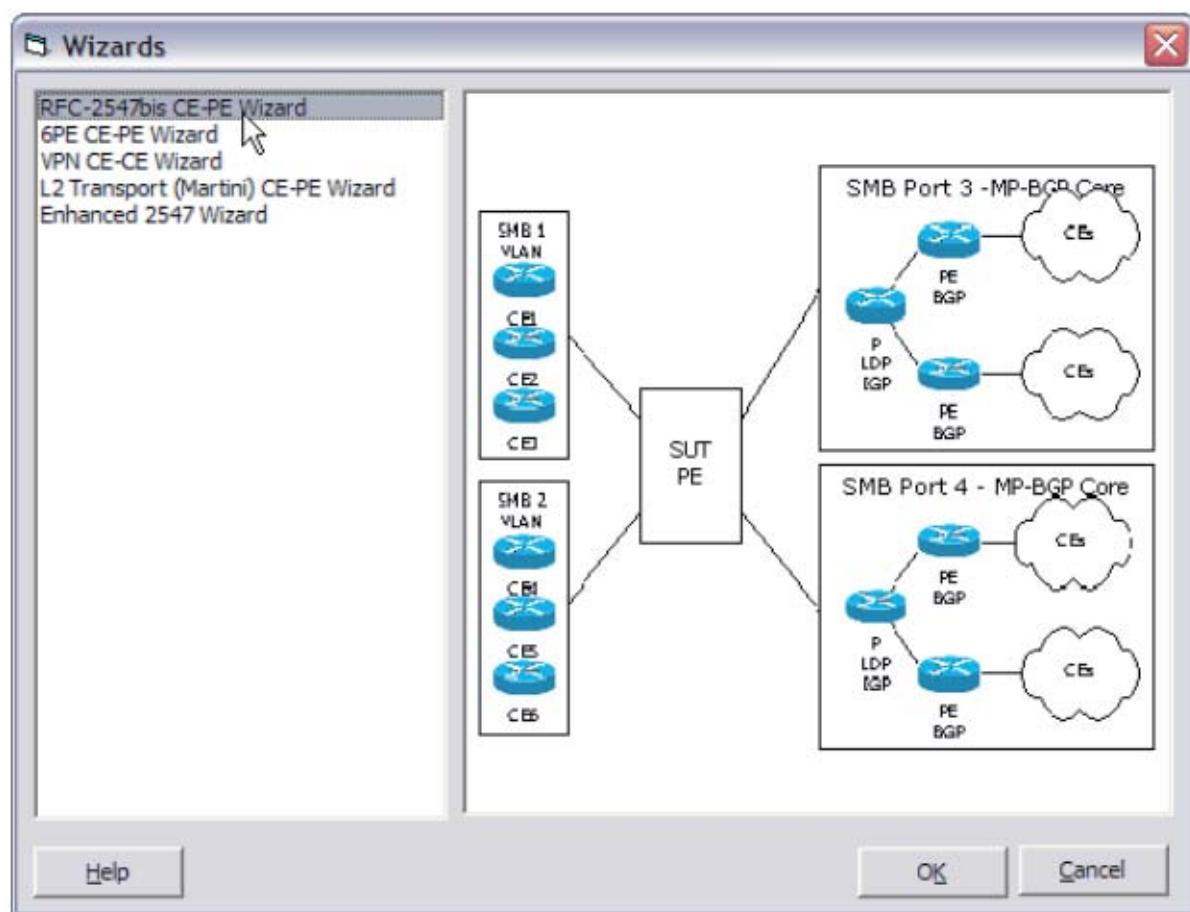
Type	Group
VPN ID	vpn1
VPN ID	vpn2
VPN ID	vpn3

New Delete Edit...
View MPLS Labels... Update MPLS Labels 6
Expand All Collapse All Filter...
Cut Streams Copy Streams Paste...
Duplicate... Select All
Copy Down... Fill Increment... Fill Decrement...

2. 2547bis Wizard 模式：

由于MPLS的测试涉及到的协议很多，使用手工的方式设置所有的配置会使得测试非常的麻烦也很容易出现配置错误的现象，为了简化测试人员的工作TRT提供了向导方式，可以很方便的构造上面进行的所有操作。如下所示：

在Wizards对话框中选择RFC-2547bis CE-PE Wizard；



在Ports Tab中选择所要使用的端口：CM 1-1作为CE端， CM 1-2作为P端设备；

在General Tab中选择，Traffic为双向的、VPN个数为3个、每个VPN的路由为10个，分别分布在CE和PE端、VPN路由可以重叠；

在Customer Setup中使用VLAN，并且开始VLAN设为2、设置起始的CE端IP地址；

在Provider Setup中根据实际要仿真的网络拓扑设置即可。

Ports General Customer Setup Provider Setup				
Active	Customer	Provider	Name	Model
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CM 1-1	LAN-3325A
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	CM 1-2	LAN-3325A
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CM 1-3	LAN-3325A
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CM 1-4	LAN-3325A
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CM 2-1	LAN-3301A
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CM 2-2	LAN-3301A

Ports General Customer Setup Provider Setup						
Traffic	Bidirectional					
Number of VPNs	3					
Routers						
Router Step	0 . 0 . 1 . 0					
Port Step	0 . 1 . 0 . 0					
Routes						
Routes per VPN	10	% CE Routes	50	% PE Routes	50	
Start Route	3 . 0 . 0 . 0		Prefix Length	24		
Increment	1				VPN Routes	Overlapping

Ports General Customer Setup Provider Setup				
CE Configuration				
<input checked="" type="checkbox"/> VLANs	Start VLAN ID	2	VLAN Scope	Unique Per Port
CE Configuration				
Start CE Address	1 . 1 . 0 . 2		/	24
CE Protocol Distribution				
<input checked="" type="checkbox"/> Router Emulation				
Protocol	% Total	# Routers		
BGP	34	1		
OSPF	33	1		
RIP	33	1		

Ports General Customer Setup Provider Setup				
Provider Core				
Provider IGP	OSPF	DUT Router ID	1 . 0 . 0 . 1	
Provider AS	1			
P Configuration				
Number of Ps	1			
Start P Router ID	2 . 1 . 0 . 2		/	24
PE Configuration				
Number of PEs	1			
Start PE Router ID	1 . 0 . 0 . 2		/	32
PE Routes				
Label	Per Site	Start Route Target	100:1	

自动生成的部分：

1、Card Setup信息：

Port	Link	Alloc	Cluster	VPN ID	Model	Participation	Reservation	Capture	VLAN	VLAN ...	Static	STP	RIP	OSPF	OSPFv3	RIP	ISIS	RSVP-TE
SPB1 101	<input checked="" type="checkbox"/>		CE	LAN-3306A		Init and Advertise	Reserved	Off	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
SPB1 102	<input checked="" type="checkbox"/>		PE	LAN-3306A		Init and Advertise	Reserved	Off	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SPB1 103	<input checked="" type="checkbox"/>			LAN-3306A		Not Used	Reserved	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SPB1 104	<input checked="" type="checkbox"/>			LAN-3306A		Not Used	Reserved	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

RSVP-TE	LDP	IGMP	PIM	IPv4	IPv4 Address	IPv4 Gateway	IPv4 Prefix Length	MAC Address	SUT MAC Address	MTU	Auto Negot
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	IPv4	192.89.1.2	192.89.1.1	24	00-00-02-00-00-01	00-00-00-00-00-00	1500	Enable
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	IPv4	2.1.0.2	2.1.0.1	24	00-00-02-00-00-02	00-00-00-00-00-00	1500	Enable

2、Sub-interface信息：

Port	VPN ID	VLAN ID	VLAN Priority	MAC Address	VPI/VCI	[IPv4]	IPv4 Address	IPv4 Gateway	IPv4 Prefix Length
SMB 1 1B1	VPN1	2	0	00-00-02-00-00-01		[IPv4]	1.1.0.2	1.1.0.1	24
SMB 1 1B1	VPN2	3	0	00-00-02-00-00-01		[IPv4]	1.1.1.2	1.1.1.1	24
SMB 1 1B1	VPN3	4	0	00-00-02-00-00-01		[IPv4]	1.1.2.2	1.1.2.1	24

3、BGP—Session信息：

Port	State	Tester AS #	SUT AS #	IE-BGP	Session IPv4	Route IPv4	Tester IPv4 Address	IPv4 Prefix Length	SUT IP Address	Gateway	BGP Insta...	TTL	Label Min
SMB 1 1B1	None	1000	1	E-BGP	IPv4	IPv4	1.1.0.2	24	1.1.0.1	1.1.0.1	<input checked="" type="checkbox"/>	64	16
SMB 1 1B2	None	1	1	I-BGP	IPv4	IPv4	1.0.0.2	32	1.0.0.1	2.1.0.1	<input checked="" type="checkbox"/>	64	16

第一个是CE1与PE之间建立的EBGP-Session；第二个是PE与远端PE之间建立的iBGP-Session。

BGP-Routes信息：

Active	IPv4	Prefix Length	# Routes	Start Route	End Route	Increment	Category	Label	VPN	VPN ID	RD	Route Target	Site of Origin	Origin
- SMB 1 1B1	#1: 1.1.0.2-1.1.0.1 AS1000-1													
	<input checked="" type="checkbox"/>	IPv4	24	5	3.0.0.0	3.0.4.0	1	Undefined	None	<input type="checkbox"/>				IGP
- SMB 1 1B2	#1: 1.0.0.2-1.0.0.1 AS1-1													
	<input checked="" type="checkbox"/>	IPv4	24	5	3.0.5.0	3.0.9.0	1	Undefined	Aggregate	<input checked="" type="checkbox"/>	vpn1	100:1	100:1	
														IGP
	<input checked="" type="checkbox"/>	IPv4	24	5	3.0.5.0	3.0.9.0	1	Undefined	Aggregate	<input checked="" type="checkbox"/>	vpn2	100:2	100:2	
														IGP
	<input checked="" type="checkbox"/>	IPv4	24	5	3.0.5.0	3.0.9.0	1	Undefined	Aggregate	<input checked="" type="checkbox"/>	vpn3	100:3	100:3	
														IGP

SMB 1B1端口构造的是从CE1发布的5个VPN1的路由；SMB 1B2端口构造的是从远端PE中发布的每一个VPN的路由，因为VPN路由可以重叠，每个VPN中构造的路由都是相同的3.0.5.0——3.0.9.0。

4、OSPF—Session信息：

Port	State	Tester IP Address	Prefix Length	Area ID	Tester Router ID	VLAN ID
SMB 1 1B1	None	1.1.1.2	24	0.0.0.0:vpn2	1.1.1.2	4
SMB 1 1B2	None	2.1.0.2	24	0.0.0.0	2.1.0.2	

第一个是CE2与PE之间的OSPF vpn2-Session；第二个是PE与远端PE之间的OSPF-Session。

OSPF-LSA:

Active	Advertising Router	Options	Prefix Length	# LSAs	Start LSA	End LSA	Increment
<input checked="" type="checkbox"/>	1.1.1.2	02	24	5	3.0.0.0	3.0.4.0	1

4、LDP-Session信息：

Port	State	Hello Type	Transport Mode	Tester IP Address	Tester Router ID	SUT IP Address	Prefix Length	Gateway	He
SMB 1 1B2	Up	Direct	Tester IP	2.1.0.2	2.1.0.2	1.0.0.1	24	2.1.0.1	5

LSPs信息：

	Active	FEC Type	Count	Start FEC	End FEC	Prefix Length	Increment
►	SMB 1 1B2						
	#1: 2.1.0.2-1.0.0.1						
		<input checked="" type="checkbox"/>	Prefix	1	1.0.0.2	1.0.0.2	32
				1			1
				1			

5、Traffic—Stream的设置信息：

Active	Name	Index					
►	SMB 1 1B1						
<input checked="" type="checkbox"/>	Stream1	1	Eth	VLAN 2	IPv4 DA 3.0.5.3-3.0.9.3/24		
<input checked="" type="checkbox"/>	Stream2	2	Eth	VLAN 3	IPv4 DA 3.0.5.3-3.0.9.3/24		
<input checked="" type="checkbox"/>	Stream3	3	Eth	VLAN 4	IPv4 DA 3.0.5.3-3.0.9.3/24		
	Total:3						
►	SMB 1 1B2						
<input checked="" type="checkbox"/>	Stream4	1	Eth	MPLS IPv4=1.0.0.1/32	MPLS VPNv4=100:1:3.0.0.0-3.0.4.0/24	IPv4 DA 3.0.0.3-3.0.4.3/24	
<input checked="" type="checkbox"/>	Stream5	2	Eth	MPLS IPv4=1.0.0.1/32	MPLS VPNv4=100:2:3.0.0.0-3.0.4.0/24	IPv4 DA 3.0.0.3-3.0.4.3/24	
<input checked="" type="checkbox"/>	Stream6	3	Eth	MPLS IPv4=1.0.0.1/32	MPLS VPNv4=100:3:3.0.0.0-3.0.4.0/24	IPv4 DA 3.0.0.3-3.0.4.3/24	
	Total:3						

从图中可以看出从SMB 1B1端口中发出来的流量有三个， 分别是从不同的VPN的CE中发出来的， 目的地址是远端PE的各自的VPN路由；然后从SMB 1B2中转发的流量是从远端PE发到CE端的路由。

测试结果分析

测试结果根据不同的类型统计在各自的表格当中， 我们来看看都有哪些统计项呢？

Stream % of Expected Rate	Stream % of Expected Rate
BGP	Event Log
OSPF	Port Pair Results
RIP	Detailed Stream Statistics
LDP	Best Performing Streams
LDP LSPs	Worst Performing Streams
Port Counters	Typical Performing Streams
Port Rates	Best Performing Streams Per Tx Port
Event Log	Worst Performing Streams Per Tx Port
Stream % of Expected Rate	Stream % of Expected Rate
Worst Performing Streams Per Tx Port	Port Avg Latency
Typical Performing Streams Per Tx Port	Port Avg Latency Data
Port Rx Rate	Stream Rx Rate
Port Rx Rate Data	Stream Rx Rate Data
Port Avg Latency	Stream % of Expected Rate
Port Avg Latency Data	Stream % of Expected Rate Data
Stream Rx Rate	Stream Avg Latency
Stream Rx Rate Data	Stream Avg Latency Data

上面的统计项非常丰富，我们可以根据自己的需求点击某项察看具体的信息。下面简单的介绍测试中会经常用到的统计项。

1. 协议报文统计信息：

实时地看到协议报文之间的交互过程。

BGP													
Port	Tester IP	SUT IP	Advertised Routes	Withdrawn Routes	Received Advertised	Received Withdrawn	Sent Notifications	Received Notifications	Sent Advertised Updates	Sent withdrawn Updates	Received Updates	Sent Keep Al	
SMB 1 1B1	1.1.0.2	1.1.0.1	5	0	6	0	0	0	1	0	2	10	
SMB 1 1B2	1.0.0.2	1.0.0.1	15	0	19	0	0	0	3	0	7	10	

BGP协议统计表中从1B1端口中发布了5个路由、撤销了0个路由，还可以看到接收到的Advertised报文个数、发送的Notifications报文个数、收到的Notifacations等协议报文信息。其它的OSPF、RIP、LDP协议的信息与BGP协议信息有点类似，不再一一介绍了。

2. 流量统计信息：

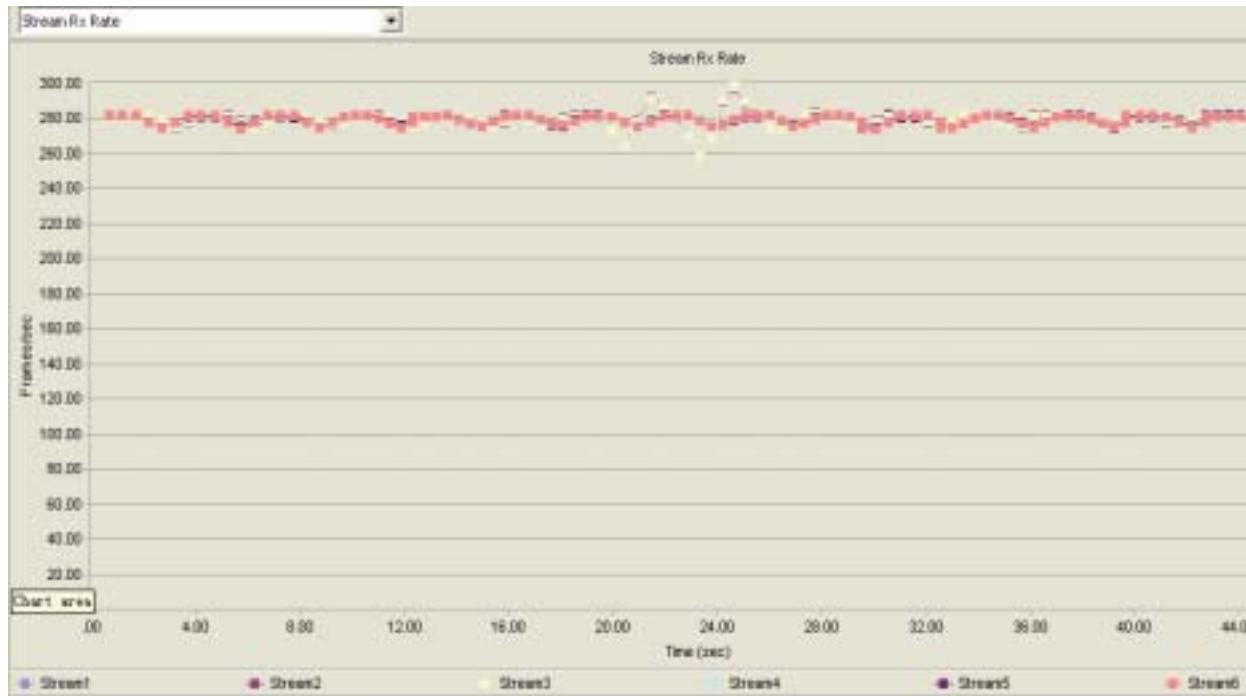
以不同流量，不同端口进行所有报文和速率的统计。

Detailed Stream Statistics														
Test	Tx Port	Stream	Expected Frames	Tx Frame	Lost Frames	% Loss	Expected	Rx Port	Rx Frames	Avg Latency (μ Secs)	Min Latency (μ Secs)	Max Latency (μ Secs)	In-sequence Frames	Out-of-sequence Frames
Test			139406	139406	0	0.0%			13797	3280.10476	951.9	106981.7	13701	96
SMB 1 1B1	Stream1	13901	13901	13901	0	0.0%	yes	SMB 1	13805	2911.331	959.4	106633.7	13709	16
SMB 1 1B1	Stream2	13905	13901	13905	0	0.0%	yes	SMB 1	13805	2925.23615	957.1	100880.7	13709	16
SMB 1 1B1	Stream3	13905	13901	13905	0	0.0%	yes	SMB 1	13805	2923.90624	951.9	100218.1	13709	16
SMB 1 1B1	Stream4	13905	13901	13905	0	0.0%	yes	SMB 1	13795	3638.86371	865.1	106979.9	13779	16
SMB 1 1B2	Stream5	13905	13901	13905	0	0.0%	yes	SMB 1	13795	3630.58768	861.3	106981.7	13779	16
SMB 1 1B2	Stream6	13905	13901	13905	0	0.0%	yes	SMB 1	13792	3631.55998	845.9	106984.2	13776	16

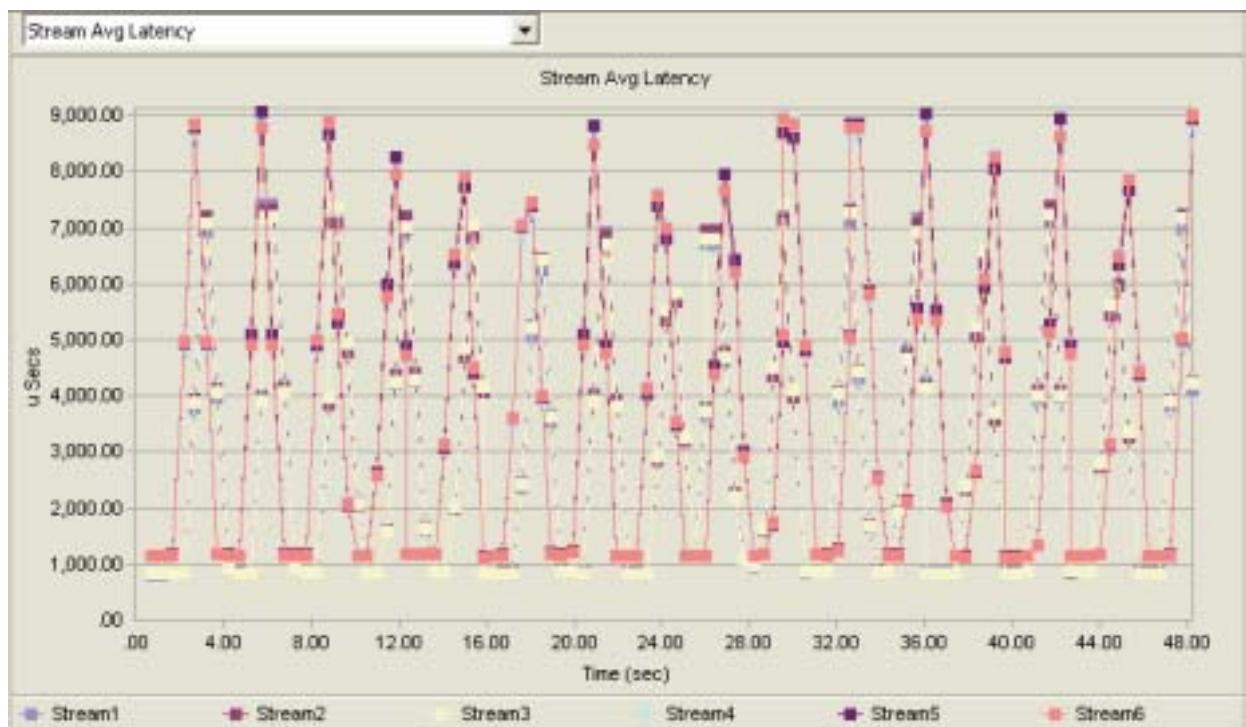
从流量统计表中可以看出：针对于每个Stream所发送的Frame个数和丢包的个数，从中可以判断哪一个VPN转发能力比较差。

Port Pair Results											
Tx Port	Rx Port	Expected Frames	Tx Frame	Rx Frame	Avg Latency (μ Secs)	Min Latency (μ Secs)	Max Latency (μ Secs)	In-sequence Frames	Out-of-sequence Frames	Lost Frames	% Loss
SMB 1 1B1	SMB 1 1B2	41703	41703	41415	2920.15879	951.9	106880.7	41357	43	285	.6906
SMB 1 1B2	SMB 1 1B1	41703	41703	41382	3640.33776	845.9	106981.7	41334	48	321	.76973

上图是基于每个端口的统计信息。



基于每条Stream的接收速率的图表统计信息，除了图表化的信息另外还有数据统计信息。



基于每条Stream的实时的时延统计信息，除了图表化的信息另外还有数据统计信息。



3. 日志信息：

从日志信息中很清楚地看到测试设备和测试仪器之间交互的协议报文信息。测试当中出现异常现象时可以通过Event Log信息来定位问题的所在，但是这种方式非常耗内存哟。

Event Log							
Protocol	Port	Entity	Direction	State	Event Type	Time (sec)	Parameters
LLDP	SMB 1 1B2	2.1.0.2:0	Tx	Operational	Hello	245.62	Type = Directed
BGP	SMB 1 1B1	1.1.0.2-1.1.0.1 v2	Rx	Established	KeepAlive	245.64	
LLDP	SMB 1 1B2	2.1.0.2:0	Rx	Operational	Hello	247.84	Type = Directed
LLDP	SMB 1 1B2	2.1.0.2:0	Rx	Operational	KeepAlive	247.84	
OSPF	SMB 1 1B2	2.1.0.2	Rx	DR/Other	Hello	248.64	RID = 7.7.7.7, AID = 0.0.0.0, DR = 2.1.0.1, BDR = 0.0.0.0, Neigh = 1
OSPF	SMB 1 1B2	2.1.0.2	Tx	DR/Other	Hello	248.81	RID = 2.1.0.2, AID = 0.0.0.0, DR = 2.1.0.1, BDR = 0.0.0.0, Neigh = 1
LLDP	SMB 1 1B2	2.1.0.2:0	Tx	Operational	KeepAlive	248.92	
OSPF	SMB 1 1B1	1.1.1.2 v3	Rx	DR/Other	Hello	249.64	RID = 1.1.1.1, AID = 0.0.0.0, DR = 1.1.1.1, BDR = 0.0.0.0, Neigh = 1
LLDP	SMB 1 1B2	2.1.0.2:0	Tx	Operational	Hello	250.72	Type = Directed
LLDP	SMB 1 1B2	2.1.0.2:0	Rx	Operational	Hello	252.84	Type = Directed
OSPF	SMB 1 1B1	1.1.1.2 v3	Tx	DR/Other	Hello	253.37	RID = 1.1.1.2, AID = 0.0.0.0, DR = 1.1.1.1, BDR = 0.0.0.0, Neigh = 1
BGP	SMB 1 1B2	1.0.0.2-1.0.0.1	Rx	Established	KeepAlive	253.64	
RIP	SMB 1 1B1	1.1.2.2 v4	Tx	Open	Update	255.19	Reach = 5, Unreach = 0
LLDP	SMB 1 1B2	2.1.0.2:0	Tx	Operational	Hello	255.82	Type = Directed
LLDP	SMB 1 1B2	2.1.0.2:0	Rx	Operational	Hello	257.84	Type = Directed
OSPF	SMB 1 1B2	2.1.0.2	Rx	DR/Other	Hello	258.64	RID = 7.7.7.7, AID = 0.0.0.0, DR = 2.1.0.1, BDR = 0.0.0.0, Neigh = 1
OSPF	SMB 1 1B2	2.1.0.2	Tx	DR/Other	Hello	258.81	RID = 2.1.0.2, AID = 0.0.0.0, DR = 2.1.0.1, BDR = 0.0.0.0, Neigh = 1
OSPF	SMB 1 1B1	1.1.1.2 v3	Rx	DR/Other	Hello	259.64	RID = 1.1.1.1, AID = 0.0.0.0, DR = 1.1.1.1, BDR = 0.0.0.0, Neigh = 1
LLDP	SMB 1 1B2	2.1.0.2:0	Tx	Operational	Hello	260.92	Type = Directed
LLDP	SMB 1 1B2	2.1.0.2:0	Rx	Operational	Hello	262.84	Type = Directed



MPLS VPN

组网应用分析

肖春喜

MPLS VPN技术

企业组网应用分析

一些大型企业组建基于MPLS VPN的网络是为了满足业务隔离的需求，针对客户的需求，我们讨论一下如何部署MPLS VPN网络并运用基于MPLS VPN的各种技术来实现客户的需求。

根据组网模型和需求，需要考虑的部分包括MPLS VPN技术中应用的方式和方法，设计时我们将MPLS VPN中的技术进行分解来说明所涉及的技术应用。

IP地址的规划问题

MPLS VPN技术，它本身是可以解决不同VPN的地址重叠问题。但在企业的实际组网中，出现地址重叠的情况比较少，因此在部署IP地址时，可以不去考虑地址重叠的问题。IP地址的设计一般有两种方式：一种是先将地址按业务进行横向分类，再按地域纵向分类；另一种是将地址先按地域横向分类，再按业务纵向分类。从网络运用的合理性来看，前一种方式显然更合理，因为它更有利于地址的聚合，对业务的扩展性较好；而后一种方式对于有些用户来说方便管理，从地址能直观识别出所属地域的特性。这些都是使用习惯问题，一般没有强制要求。

IGP协议的选择

在MPLS VPN体系结构中，IGP的主要作用是保证BGP对等体的可达性以及MPLS隧道的建立，在这一点上与普通的BGP组网中原理是一致的。和纯粹的IP网络不同，大部分情况下MPLS VPN体系

结构中的IGP不带任何客户的VPN业务，IGP路由是为了建立BGP连接和交互协议报文，因此有较多的选择。一般而言，推荐采用收敛速度快、路由振荡少、基于SPF算法的路由协议，如OSPF、IS-IS等；在部分环境中，也可以使用静态方式。IGP的另一个功能是驱动公网标签分配，以建立MPLS隧道。MPLS VPN网络中数据流的转发是基于标签的，如何让设备的标签能够正确分配，IGP协议居功至伟。

在规模较大的网络中，如果采用OSPF作为IGP，并且需要划分区域进行管理时，不要使用stub、totally stub、NSSA区域等优化应用措施；如果采用IS-IS作为IGP，在区域分级时需要将level-2中的设备loopback接口地址段引入到level-1中。

MPLS的配置中有一条命令是lsp trigger-all，这条命令的目的是为所有的IGP路由分配标签。我们知道MPLS隧道的起点和终点都是LSR的LSR-ID（一般是设备的loopback接口），因此无须为每一条IGP路由都分配标签。默认配置下只为32位主机地址分配标签，这样就可以满足MPLS VPN的部署需要了。所以建议不要使用该命令，以免造成对标签空间不必要的浪费。

针对MPLS VPN的应用

从企业组网的需求来分析，MPLS VPN的组网应用分为2个层次，域内(inter AS)MPLS VPN的规划和域间(intra AS)的规划设计。从技术角度分析，两者有共性的东西，也有相异性，在应用时我们按inter AS和intra AS来分析应用。

域内MPLS VPN的规划

■ 业务的命名

实际组网中业务的命名通常采用业务的拼音字母或是英文名，只要不重复就行。我们假设客户需要3个VPN，命名为sale, finance, manage。

```
ip vpn-instance sale
ip vpn-instance finance
ip vpn-instance manage
```

■ RD的定义

从原理上讲，RD的作用是将IPv4的地址变成全局唯一的VPNv4的地址，当不同VPN内出现重叠的IPv4地址时，RD可以将他们区分开来。采用格式通常为ASN : N方式，也有使用基于IP地址格式的，如X.X.X.X : N，不过后者不常使用。所以只要VPN地址不发生重叠，RD可以任意搭配。根据网络特点，我们采用ASN : N方式，用本AS号+N(N可以任意取值)，一般在同一个VPN中使用相同的RD是较为常见的做法。

所以RD定义：

vpn-sale	RD=ASN : 100
vpn-finance	RD=ASN : 200
vpn-manage	RD=ASN : 300

ASN为本AS的编号

■ RT的定义

RT在MPLS VPN中作用非常明显，它用来控制VPN的隔离和部分互通，格式与RD相同。对不

同的VPN，要求定义不同的RT的值，如果有互通需求，通过RT的属性来控制，分为export和import属性。export属性代表发送VPN路由时附带的属性，当另一PE设备收到此路由，通过import属性来决定接收与否或是接收时与哪个对应的VPN关联。所以针对VPN的定义，如果三个VPN不要求互通，那么：

vpn-sale	export =ASN : 100	import=ASN : 100
vpn-finance	export =ASN : 200	import=ASN : 200
vpn-manage	export =ASN : 300	import=ASN : 300

■ PE-CE间路由的规划

PE-CE之间的路由协议，不同于PE-PE之间的IGP路由，PE-CE间的IGP是传递VPN路由的，并且PE会将这些VPN路由通过BGP来发送。所以PE-CE之间的路由选择应该根据实际需要来使用，可选择直连，静态，OSPF，RIP，IS-IS或是EBGP。在使用中需要特别注意一点，所有的路由都是基于实例(vpn-instance)的，在不同的VPN中可以采用相同类型的路由协议，最终被import到MP-BGP中。因此要求PE设备对路由协议的支持要丰富，多实例是最基本需求。从组网中的双归属特性来看，采用OSPF或是EBGP比较合适(在企业网中使用IS-IS的情况比较少)。从减轻网络的复杂程度来看，使用OSPF比较通用。

在规划部署OSPF区域的时候，整个MPLS核心骨干被看做一个SuperBone，PE-CE之间的区域则属于普通的骨干或者非骨干区域。如果在不同site的位置上，PE-CE之间运行完全不同的OSPF进程，那么除了多实例的要求外，Domain ID的规划也是重点考虑的条件之一。

同时，为配合双归属的要求，在将BGP注入到OSPF时，对于cost值的灵活应用也是很重要的。按照应用需求，在CE双归属到PE上时，如地区公



司CE到地区公司的PE上行时，要求负载分担，在PE上OSPF注入BGP时，可设置相同的cost值，如果要求针对不同网段选择不同的PE设备，另一PE设备作为备份使用时，可通过设置不同的cost值来实现。

■ PE-PE之间IBGP

PE-PE的IBGP，是对BGP4协议的扩展，对BGP4是完全兼容的，所以相关的路由策略设计也是相同的。尤其是在选择路径的策略和原则上，这是BGP的魅力之一。不同的是将原有普通的BGP路由变成带有VPN属性的BGP路由。对于路由的选路控制充分利用Local-Preference和MED属性，Community属性来实现。当缺省不作设置时，应该充分利用BGP中对于loopback地址这个下一跳和IGP的优选路径的依赖来实现业务的分流。

■ 关于多角色主机

此功能在网络中应用比较广泛，主要应用于某个特定的VPN中用户，有较高的权限访问多个或者所有VPN中的业务，是一种新的技术实现手段。传统的技术手段包括：更改原有的拓扑结构，采用类似Hub-Spoken的组网技术，或是采用RT值的配对来控制路由的分发。多角色主机的技术是对传统技术手段一种较好的补充。它在发送时，采用策略路由的方式，强制进入它希望的VPN中，在返回的报文，采用静态路由的方式连续在不同的VPN中强制定义返回VPN的下一跳来实现访问。这个功能特性大大减轻了对设备的压力，并能很好的实现访问多个VPN的需求，多用于网管服务器对所有VPN中

CE的管理或是超级用户对VPN的访问。

■ HoPe的应用

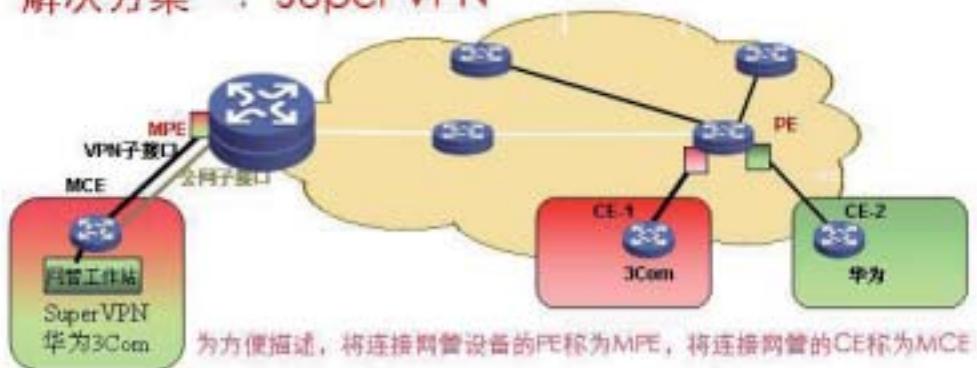
当网络规模较大时，在同一AS中，所有的PE设备因为采用IBGP的连接，因此维护路由的总量是相同的，同时对于标签的消耗也是相同的。但网络毕竟有层次之分，在接入层的设备由于硬件和软件功能特性的原因，不能承担大量的VPN路由，这样就会出现部署的问题。尤其当接入设备只有一个上行出口时，维护VPN路由更没有必要，这时分层PE的技术较好的解决了这个问题。由SPE向UPE发送相应VPN的缺省路由，尤如OSPF中stub区域从ABR获取了缺省路由一样，减轻了下层UPE设备的压力。在有多个出口的情况下，可以通过BGP的选路和HoPE的配合，减小路由表的容量。HoPE的功能是可以嵌套使用的。

在使用中，我们发现HoPE还有可改进的地方。目前它的实现方法类似于IS-IS中level-2对于level-1中发送缺省路由，这是用户不太希望看到的，希望有控制的发送部分指定路由，这要求实现类似level-2往level-1中的路由泄漏功能，如果分层PE能实现按需由SPE向UPE发送路由，而不是单纯发送缺省路由，这个技术的应用就完美了。

■ VPN中网管的使用

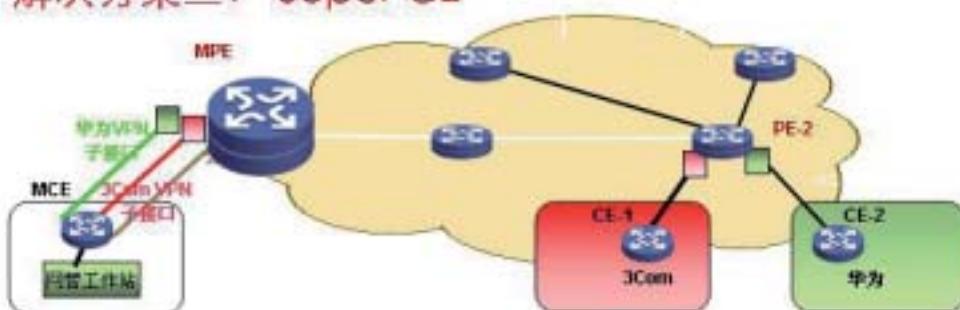
如果VPN中的网管，只需要管理PE设备，部署非常简单，将网管服务器连接到公网所在接口上，直接在公网的IGP中发布路由。如果还需要管理CE设备，建议采用如下三种方式来实现：

解决方案一：Super VPN



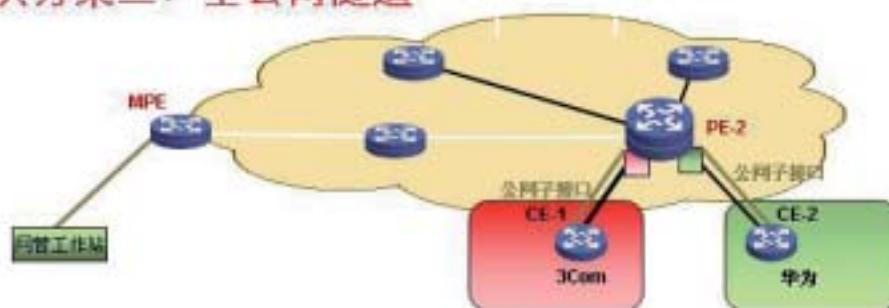
- 将网管工作站放在一个Super VPN中，使之可以访问所有的VPN中的CE设备，同时在该CE上与PE之间创建公网的子接口；
- 在CE上配置公网路由，下一跳指向PE的公网子接口，配置私网路由，下一跳指向PE的私网子接口；
- 将该网管工作站的地址同时在公网和私网中发布。

解决方案二：Super CE



- 将网管工作站放在一个Super CE下，该CE设备通过为每个VPN建立一个与PE的子接口，使之可以访问所有的VPN中的CE设备，同时在该CE上与PE之间创建公网的子接口；
- 在CE上配置公网路由，下一跳指向PE的公网子接口，为每个VPN配置私网路由，下一跳指向PE的该VPN的私网子接口；
- 将该网管工作站的地址同时在公网和私网中发布。

解决方案三：全公网隧道



- 将网管工作站直接连接在一台PE或P路由器下，在每一台CE设备上创建一个公网的子接口；
- 在CE上配置公网路由，下一跳指向PE的公网子接口；
- 将每一台CE设备的loopback地址在公网中发布。

上述三种方案的比较：

方法	网管设备位置	对VPN地址规划的要求	对设备的需求	路由发布	工作量	安全性
Super VPN	私网中，与MCE相连	VPN地址空间不能重合 至少CE的loopback地址不能重合	最好是使用单独一台MCE与网管相连	只要私网和公网的路由完全分开，可以在MCE上方便地配置静态路由	更改所有PE（RT改变）和一台CE的配置（子接口）	存在不同VPN通过Super Vpn间接互访的隐患，网管工作站易受到CE中主机的攻击
Super CE	私网中，与MCE相连	VPN地址空间不能重合 至少CE的loopback地址不能重合	最好是使用单独一台MCE与网管相连	如果私网地址无规律，必须在MPE与MCE之间运行动态路由协议的多实例	只更改一台PE和一台CE的配置（子接口）	存在不同VPN通过Super CE间接互访的隐患，网管工作站易受到CE中主机的攻击
全公网隧道	公网中，与MPE或MP相连	无特殊要求	无特殊要求	在公网可用动态路由方便的发布CE的loopback地址	更改所有PE、CE的配置（子接口）并需要重新分配公网的PE、CE互联地址	无

■ 与internet互联

PE分布式上internet

- 每个VRF中选择一台PE连接internet，连接该PE的CE上做NAT转换。

- 由该CE发布一条缺省路由给本VPN内的所有CE。

PE集中式上internet

- 选择一台健壮的CE连接internet，并且做地址转换。

- 将该CE所属的VRF配置成可以和所有VRF都互通的超级VPN（How can do that?）

- 由该CE发布一条缺省路由给全网的所有VPN的所有CE。

- 由于有超级VPN以及缺省路由的存在，会导致不同的VPN通过超级VPN互访，需要在该PE连接超级VPN的VRF上配置 ACL，丢弃源地址和目的地址都是私网地址的报文。

选择哪种方式？

- PE分布式是运营商常用的（因为运营商的每台PE都直接与internet相连），企业网不会使用。

- PE集中式太繁琐（每台PE上都要配置ACL），且不支持VPN地址重叠。

- PE分布式无需ACL，且支持VPN地址重叠。

希望节省资源，选择PE集中式；希望支持VPN地址重叠，选择PE分布式。

AS之间MPLS VPN的规划

随着MPLS VPN解决方案的越来越流行，服务的终端用户越来越多，规格和范围也在增长。在一个特殊的企业内部的站点数目越来越大，某个地理位置与另外一个服务提供商相连的需求变得非常普遍。比如我们国内运营商的不同城域网之间，或是同骨干网之间都存在着非常现实的跨越不同自治域问题。这些都需要一个不同于基本的MPLS VPN体系结构所提供的互连模型—跨域的MPLS VPN，为了支持服务提供商之间的VPN路由选择信息交换，

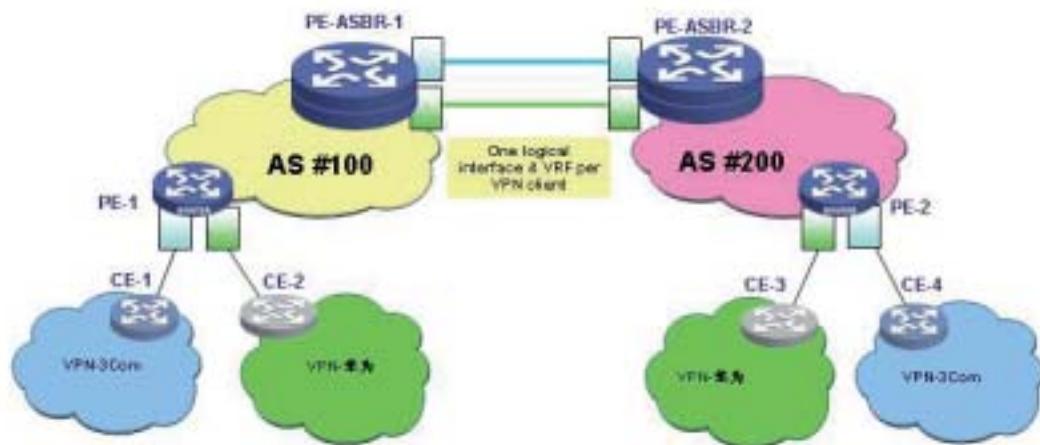
需要一个新的机制，以便可以穿过提供商间的链路来广播路由前缀和标签信息。但是一般的MPLS VPN体系结构都是在一个自治系统内运行，任何VPN的路由信息都可以在一个自治系统内按需扩散，就是没有提供一个跨域的VPN信息扩散功能。因此，为了支持跨域VPN的需求，就需要扩展现有的协议和修改MPLS VPN体系框架。

目前在组网上有三种主流模型，下面将一一介绍。



■ VRF to VRF

VRF-VRF组网结构



VRF to VRF Connectivity between PE-ASBRs

- VRF-VRF解决方案从技术上讲是最简单的，没有在“AS内部的MPLS-VPN”上作任何扩展，完全应用已有技术实现。
- ASBR对等体间，通过划分子接口方式，每个子接口分别绑定一个VRF，保证域间传播路由的私有性。
- ASBR对等体间，只运行普通BGP，不运行LDP，交互IPV4路由。
- 每个PE-ASBR路由器都把对方PE-ASBR路由器当做CE路由器看待。
- 比较适合运用在AS域间交互VPN(VRF)数量较少的情况。但是扩展性较差。

方案的主要特点：

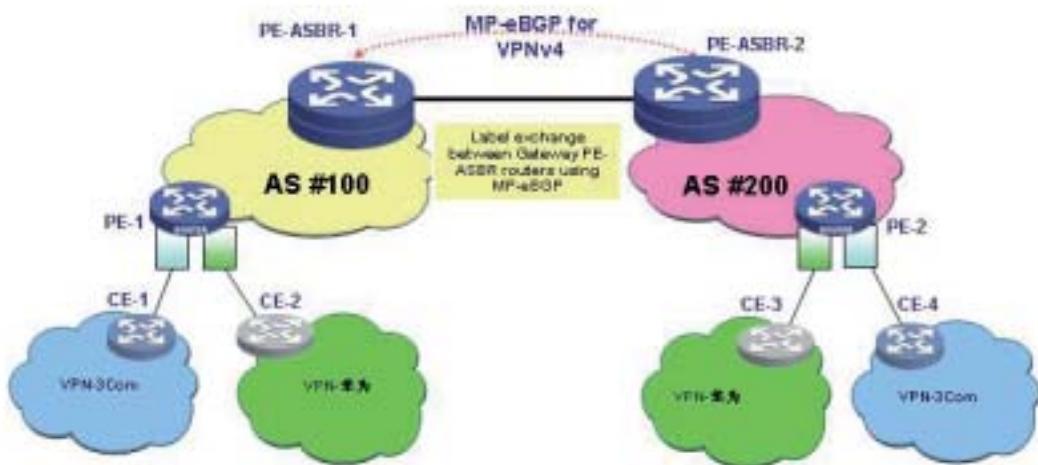
- 在两个ASBR-PE之间要为不同的VRF建立独立的物理或者逻辑链路(从节省接口的角度考虑，推荐使用逻辑链路)，但在企业网中，如果链

路是SDH的透明传输，配置起来会比较麻烦。

- LSP的建立：这种方式下PE只要有到本AS内的ASBR-PE的标签就可以了。
- 由于需要在ASBR-PE上为每个VRF配置独立的链路。在VPN数量较多时配置量与BGP的邻居数量是相当大的，所以这种跨域技术只适合VPN数量很少的情况下，适用于一些规模不大的企业网。
- 为了支持不同自治域的VPN互通，必须在ASBR路由器上对应配置相同的VPN，如果跨越多个自治域，配置工作量很大，且对中间域影响比较大，中间域必须支持VPN业务。在VPN数量较多的情况下，在每一个ASBR上的配置工作量也是很大的。
- 这种方案应用非常简单，不要扩展协议和做特殊配置，属于天然支持。在需要跨域的VPN数量比较少的情况下，可以考虑使用，属于简单实用型方案。

■ 单跳MP-EBGP的组网应用

“单跳” M-EBGP组网



MP-BGP VPNv4 prefix exchange between Gateway PE-ASBRs

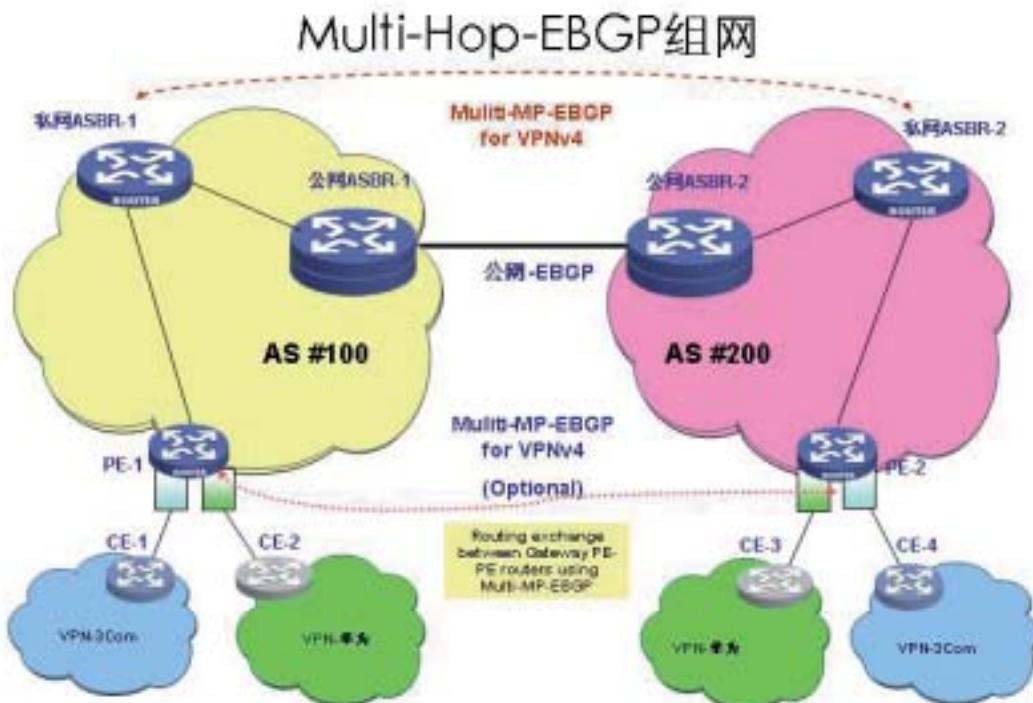
- PE-ASBR对等体之间建立单跳的MP-EBGP邻接体，传递VPN-IPv4路由，不运行IGP和LDP。
 - PE-ASBR对等体之间传递私网路由时，因为EBGP邻居关系，需要改变路由的下一跳，所以需要交换内层标签。
 - 接收端PE-ASBR，可以使用next-hop-local命令，强制修改路由的下一跳，同时再次交换内层标签，通告给MP-IBGP邻居。如果没有配置next-hop-local命令，需要把direct路由重分布（import-route）到IGP中。
 - PE-ASBR路由器上需要保存所有域间的私网路由。对于ASBR路由器来说，压力较大。
 - 和VRF-VRF方式相比，具有更好的扩展性。
- 方案的主要特点：**
- 报文转发时，需要在两个ASBR上都要对

VPN的LSP做一次交换。有一个需要注意的问题是，这种解决方案需要在ASBR上接收本域内和域外传过来的所有VPN路由，然后再把VPN给扩散出去。但是MPLS VPN的特性结构中要求，只有一个PE上有VPN匹配某条VPN路由时，这条VPN路由才会被保存下来。因此对于上述ASBR上需要保存VPN路由，这要求本地要配置它匹配的VPN实例。当然，可以使用特殊的处理手段不配置相应的VPN实例（配置undo policy vpn-target），但这会引起管理和排错的不便。

- 由于这种方案需要在ASBR上保存所有的VPN路由，因此对路由器提出了很高的要求，使ASBR更容易成为故障点。不过只要VPN的路由数量不是很多，这种方案不失为一种配置简单且实用的方案，特别适合网络规模较小的企业网用户。



■ 多跳MP-EBGP的组网应用



此方案本身也分为3种方式：

- 模式1(不改变PE下一跳方式)：本AS内PE将本地的loopback接口要泄漏到对方AS中，因为私网下一跳并没有改变，所以公网ASBR要为此loopback主机路由分配标签(BGP标签)，这样对方AS中会有大量的本AS中的主机路由，单纯的PE-PE的配置会非常繁琐，可以在本AS内选用私网ASBR的组网方式，减少了配置量。同时最内层VPN路由的私网标签没有更改。但本AS内各设备的loopback 主机路由还会泄漏到对方AS中，在运营商组网中基本是不允许的，企业网无此限制。
- 模式2(本AS内私网ASBR改变PE下一跳)：由本AS内ASBR充当一个管理者，它将本AS内PE的VPN路由的下一跳都更改为自己，然后与对方AS中相同地位的私网ASBR建立针对VPN业务的

EBGP邻居，这样ASBR只需要将私网ASBR的loopback主机路由发送到对方AS中，同时减少了配置量，注意对方AS中的PE查看到的本方AS中VPN路由的下一跳为本方的私网ASBR，也就是说最内层VPN路由的私网标签更改过一次(知道在哪儿吗？)。这种组网比较符合运营商组网。

- 模式3(本AS和对方AS中的私网ASBR都更改PE下一跳)：由本AS内私网ASBR和对方AS内的私网ASBR充当自己域内的管理者，抽象来看，就是两个私网ASBR之间建立的EBGP邻居，采用标准的EBGP发送路由方式，本方修改下一跳方式，这样公网ASBR也需要将私网ASBR的loopback主机路由发送到对方AS中，但最内层VPN路由的私网标签更改过2次(哪两次？)。这种组网也比较符合运营商组网。

方案三的总体特点

- 利用了BGP的一个新特性(RFC3107)，这个特性可以让BGP在传递公网路由的时候携带标签。
- BGP本身是靠TCP建立连接，所以只要两个端点可达到，就可以建立BGP的邻居，从而完成VPN路由的交换。第三种方式实际上就是靠两个PE设备之间建立多跳的MEBGP邻居来完成VPN路由交互的。
- MULTIHOP-EBGP的跨域方案应该说对于运营商是最理想的，因为它符合MPLS VPN的体系结构的一些要求。比如VPN的路由信息只出现在

PE设备上，而P路由器只负责报文的转发。这样就使中间域的设备可以不支持MPLS VPN业务，仅充当一个普通的支持MPLS转发的ASBR路由器，可以同时支持跨域的需求和普通的IP业务，尤其是在跨越多个域时优势更加明显。这个方案更适合支持MPLS VPN的负载分担等功能，也没有可能会成为性能瓶颈的点。不过由于这种解决方案中需要对普通的BGP做扩展，且隧道的生成也是有别于普通的MPLS VPN结构，因此维护和理解起来难度比较大，不适合用于企业网的环境。



缩略语列表

缩略语	描述	注释
CPE	Customer Premises Equipment	用户端设备
CR-LDP	Constraint-based Routing LDP	基于约束路由LDP
DS-TE	Difference Service aware traffic engineering	支持DiffServ的流量工程
FEC	Forwarding Equivalence Class	转发等价类
FIB	Forwarding Information Base	转发信息数据库
HA	High Availability	高可靠性
HoPE	Hierarchy of PE	分层PE
L2VPN	Layer 2 VPN	二层VPN
L3VPN	Layer 3 VPN	三层VPN
LDP	Label Distribution Protocol	标签分发协议
LFIB	Label Forwarding Information Base	标签转发信息数据库
LIB	Label Information Base	标签信息数据库
LSP	Label Switched Path	标签交换路径
LSR	Label Switching Router	标签交换路由器
MIB	Management Information Base	管理信息数据库
MPE	Middle PE	中间PE
MPLS	Multiple Protocol Label Switch	多协议标签交换
MPLS TE	MPLS Traffic Engineering	MPLS流量工程
MTU	Maximum Transmission Unit	最大传输单元
QoS	Quality of Service	服务质量
RD	Route Distinguisher	路由识别符
RSVP	Resource reSerVation Protocol	资源预留协议
RSVP-TE	RSVP Traffic Engineering extension protocol	RSVP TE扩展协议
RT	Route Target	路由目标
SPE	Sevice Provider-end PE	提供商侧PE
TE	Traffic Engineering	流量工程
UPE	User-end PE	用户侧PE
VC	Virtual Circuit	虚拟电路
VLL	Virtual Leased Line	虚拟租用线路
VPN	Virtual Private Network	虚拟私有网
VRF	VPN Routing and Forwarding Instance	VPN路由转发实例
VPWS	Virtual Private Wire Service	虚拟专线服务

Route to Network

网络之路

—— MPLS技术专刊

闻说双溪春尚好，也拟泛轻舟。

只恐溪头蚱蜢舟，载不动，许多愁。

—— 宋·李清照

策划：刘宇 陈旭盛 刘炜刚

主编：陆宇翔

编委：杜祥宇 文旭 徐庆伟 王慧升

王辉 朱皓 张雪莲 刘先楠

陆强 王乐 蔡金龙 王君菠

贾欣武



2006年第1季度 总第3期