BGP

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

概述

在当前所使用的计算机网络中，一个网络，通常使用一个 IP 网段来表示，要将所有网络连接起来，并且要通信，就需要将这些 IP 网段连接起来，让每个 IP 网

段都知道其它 IP 网段的信息，就可以实现全网通信。将网络与网络连接起来的设备是路由器，只要网络中每一台路由器都知道所有 IP 网段的信息，就可以为全网提供数据转发，如果某一台路由器不能得知所有的 IP 网段信息，也就表示这台路由器所连接的网络不能与其它网段通信。为了帮助路由器获得全网的 IP 网段信息，因此路由协议工作在路由器与路由器之间，路由协议将网络中每一条路由（IP 网段）在路由器与路由器之间传递，最终让网络中每一台路由器都拥有全网完整的路由信息，从而实现全网可达。

从上可以看出，路由协议在路由器之间传递路由信息，是保证网络通信的基础，如果路由协议传递了错误的路由信息，或者没有传递路由信息，将导致某些网络通信的中断，所以路由协议从一台路由器收到路由更新后，必须毫不保留地传递给其它路由器，而当一个网络失效后，也必须告知其它路由器该网段不可达，需要将相应路由删除。

当全网每一台路由器都拥有所有的路由信息，并且完全一致时，这种状态被称为收敛状态，一个网络只有在收敛状态时，才能保证全网可达。而当今所使用的最庞大的互联网，是由数万台路由器连接起来的，如果每一台路由器都拥有互联网中的每一条路由信息，那么就意味着每一台路由器都将拥有数十万甚至数百万条路由条目，这个数量是惊人的。但是由于路由协议的特征以及互联网全网通信的需求，就必须让互联网中每一台路由器将自己的路由信息与其它路由器互换，最终使整个互联网的达到收敛状态。虽然这是铁定的要求，但是请仔细想一下，这是万万不可能的，因为一个拥有数万台路由器组建起来的超大型网络，永远不可能达到收敛状态，因为当某个网络断开时，最先得知的路由器需要将这个信息告知给其它所有路由器，因为信息是一台传一台传过去的，所以一个网络断开的信息要让数万台路由器都知道，这需要很长的时间，并且可能在这个信息还没有传遍整个网络时，这个之前中断的网络就恢复了正常，那么这时，最先知道的路由器又要重新向网络中通告该网段恢复正常的信息，如此一来，互联网中不断变化的网络，会让所有路由器不停地传递路由信息，结果是导致网络中路由信息的不一致，也将导致庞大的路由更新影响所有路由器的性能。因此，互联网中，一个网络的中断与恢复，实在没有必须通告给数以万计的路由器。而网络的信息，必须向其它路由器通告，那么，一台路由器的路由信息，既然没有必须向网络中每一台路由器通告，那么，它究竟该通告给哪些路由器呢？或者换句话说，它的路由更新通告的范围究竟有多大呢？

基于以上种种原因，所以我们将一台路由器的路由更新限制在一定的范围内，也只有这样，一个被划分为更小范围的网络，才能达到收敛状态。所以现实情况是，我们的互联网被划分成了一个一个更小范围的网络，而任何一台路由器的路由更新，被限制在这个特定的范围内，而这个特定的范围，就是你应该知道的被称为自治系统的网络范围，即 autonomous system（AS） 。我们设计了互联网中路由协议的更新只应该在一个 AS 内部传递，但是互联网是需要全网通信的，所以必须让每一个 AS 都能够获得其它 AS 的路由信息才行，因此，路由协议被定义为两种截然不

同的种类，即只在一个 AS 内部更新的路由协议，称为 Interior Gateway Protocol (IGP) ，以及在 AS 与 AS 之间更新的路由协议，称为 Border Gateway Protocol (BGP)。

需要更多的解释，将互联网划分成多个 AS，目的并不仅仅是将路由协议的更新限制在特定的范围内，还有一个重要的原因是，将互联网划分成若干个小范围的网络后，那么这每一个小网络就可以单独定义各自的路由策略与安全策略，并且这样不需要干扰其它 AS，也不受其它 AS 干扰。比如网络中若干的 ISP，这些 ISP 对自己的网络需要制定自己的策略，又需要让这些策略保持私有性而不与其它 ISP 互相干扰，所以划分 AS，帮他们实现了这个目的。

只能在一个 AS 内部传递更新的 IGP 路由协议有 RIP，EIGRP，OSPF，IS-IS，可以在 AS 之间传递更新的路由协议目前只有 BGP。但是有个特别之处是，EIGRP 也使用了 AS 的概念来工作，运行 EIGRP 的网络也会被划分成多个 AS，虽然默认情况下，EIGRP 不能在 AS 与 AS 之间更新路由信息，但是 EIGRP 也可以实现 AS 之间的路由更新。需要说明的是，EIGRP 概念中的 AS 与 BGP 的 AS 并无任何关联，它们之间没有任何共同操作性，真正的 AS 是指 BGP 的 AS，而 EIGRP 不管有什么样的 AS 特征，它永远被限制在 BGP 的单一 AS 之中。

注：BGP 支持 classless interdomain routing (CIDR)

BGP AS

对于 BGP 的 AS 号码的分配，是由 Internet Assigned Number Authority (IANA) 机构来统一规划和分配的，IOS 中运行的 BGP，目前最多支持 4 字节长度的 AS 号码，但并不表示所有 AS 号码都能任意配置，在 2009 年 1 月之前，只能使用最多 2 字节长度的 AS 号码，即 1-65535，在 2009 年 1 月之后，(IANA)决定使用 4 字节长度 AS，范围是 65536 -4294967295。

当前，通常还是使用 2 字节长度的 AS，也就是 1-65535，所以不对 4 字节的 AS 号码做太多讨论。因为 BGP 是使用在互联网之中的，互联网由多个 BGP AS 域组成，所以互联网中不能出现 AS 号码相同的域，如果一台路由器要接入互联网并运行 BGP，那么必须向 IANA 申请合法的 AS 号码。为了考虑到某些大型企业需要使用 BGP 与 ISP 对接，而又没有足够的 AS 号码用来分给企业用户，所以将 AS 号码划分为公有 AS 和私有 AS，公有 AS 的范围是 1-64511，私有 AS 范围是 64512-65534；公有 AS 只能用于互联网，并且全球唯一，不可重复，而私有 AS 可以在得不到合法 AS 的企业网络使用，可以重复。很显然，因为私有 AS 可以被多个企业网络重复使用，所以这些私有 AS 不允许传入互联网，ISP 在企业用户边缘，需要过滤掉带有私有 AS 号码的路由条目。

BGP 邻居

如果你在自己的 PC 上从某个 FTP 服务器去下载文件，那么你的 PC 只要和 FTP 服务器是通畅的即可，也就是说你的 PC 只要 ping 得通 FTP 服务器就行，不管距离有多远，因为不可能每个从 FTP 服务器上下载文件的 PC 都与之是直接连在一起的; PC 从 FTP 服务器下载文件时，使用的是 TCP 传输，当数据在中途出现丢包时，被丢弃的数据包能够得到重新传递，从而保证下载的文件是完整的。由于 BGP 运行在整个互联网，传递着数量庞大的路由信息，因此需要让 BGP 路由器之间的路由传递具有高可靠性和高准确性，所以 BGP 路由器之间的数据传输使用了 TCP 协议，端口号为 179，并且指的是会话的目标端口号为 179，而会话源端口号是随机的。

正因为 BGP 使用了 TCP 协议传递，所以两台运行 BGP 的路由器只要通信正常，也就是说只要 ping 得通，而不管路由器之间的距离有多远，都能够形成 BGP 邻居，从而互换路由信息。

一个配置了 BGP 进程的路由器只能称为 BGP-Speaker，当和其它运行了 BGP 的路由器形成邻居之后，就被称为 BGP-Peer。如果一个网络中的多台路由器都运行 OSPF 之后，那么这些路由器会在相应网段去主动发现 OSPF 邻居，并主动和对方形成 OSPF 邻居。而一个路由器运行 BGP 后，并不会主动去发现和寻找其它 BGP 邻居，BGP 的邻居必须手工指定。

BGP 和其它路由协议一样，传递的是网络层协议，如 IP 协议，除此之外，BGP 还能够传递除 IP 协议之外的其它网络层协议，能够传递的协议如下：

IP Version 4 (IPv4),

IP Version 6 (IPv6),

Virtual Private Networks version 4 (VPNv4),

Connectionless Network Services (CLNS),

Layer 2 VPN (L2VPN).

这些协议被称为 address family，配置时，需要进入相应的协议 address family 模式，而 Ipv4 除外。所有命令在 address family 中独立配置，独立生效，并且都拥有独立的数据库。正常的 BGP 配置模式被称为 NLRI 模式，而 address family 模

式称为 AFI 模式，像 MPLS，只能在 AFI 中配置，而不能在 NLRI 模式中配置，在 NLRI 模式中配置的参数只对 Ipv4 单播生效。

IOS 支持四个 AFI 模式，为：IPv4, IPv6, CLNS, VPNv4，并且 IPv4 和 IPv6 还有单播和组播之分。

思科路由器运行的 BGP 为 version 4，一台路由器只能运行一个 BGP 进程，并且整台路由器只能属于一个 AS，但是一台路由器可以承载多个 address family，而一个支持多个 address family 的 BGP 和一个不支持的可以正常通信，但这也仅限于 Ipv4。

一台 BGP 路由器运行在一个单一的 AS 内，在和其它 BGP 路由器建立邻居时，如果对方路由器和自己属于相同 AS，则邻居关系为 internal BGP (iBGP)，如果属于不同 AS，则邻居关系为 external BGP (eBGP)。BGP 要求 eBGP 邻居必须直连，而 iBGP 邻居可以任意距离，但这些都是可以改变的。

在 BGP 形成邻居后，最开始会交换所有路由信息，但是之后都采用增量更新，也就是只有在路由有变化时才更新，并且只更新有变化的路由。

BGP 建立邻居后，会通过相互发送类似 hello 包的数据来维持邻居关系，这个数据包称为 Keepalive，默认每 60 秒发送一次，hold timer 为 180 秒，即到达 180 秒没有收到邻居的 Keepalive，便认为邻居丢失，则断开与邻居的连接。

BGP 之间建立邻居，需要经历如下几个过程：

Idle—BGP 进程被启动或被重置，这个状态是等待开始，比如等于指定一个 BGP peer，当收到 TCP 连接请求后，便初始化另外一个事件，当路由器或 peer 重置，都会回到 idle 状态。

Connect—检测到有 peer 要尝试建立 TCP 连接。

Active—尝试和对方 peer 建立 TCP 连接，如有故障，则回到 idle 状态

OpenSent— TCP 连接已经建立，BGP 发送了一个 OPEN 消息给对方 peer，然后切换到 OpenSent 状态，如果失败，则切换到 Active 状态。

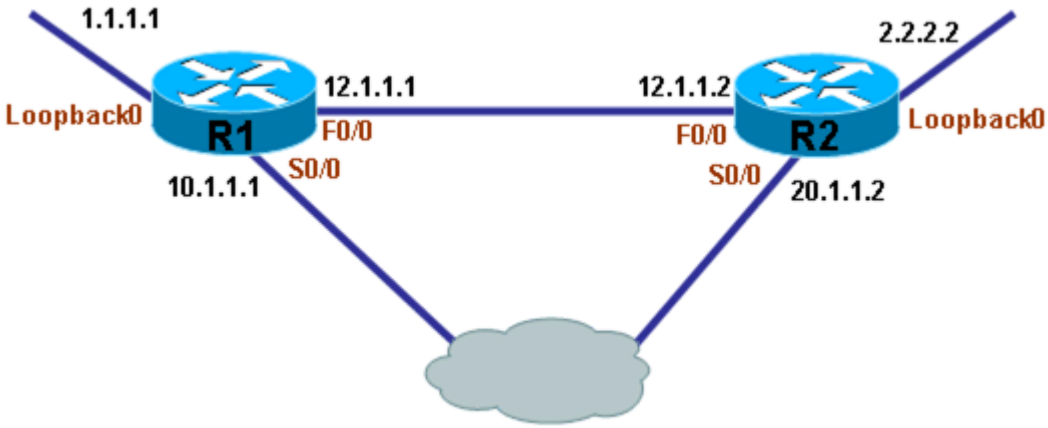
OpenReceive— 收到对方 peer 的 OPEN 消息，并等待 keepalive 消息，如果收到 keepalive，则转到 Established 状态，如果收到 notification，则回到 idle 状态，比如错误或配置改变，都会发送 notification 而回到 idle 状态。

Established— 从对端 peer 收到了 keepalive，并开始交换数据，收到 keepalive 后，hold timer 都会被重置，如果收到 notification，就回到 idle 状态。

BGP 更新源

BGP 并不能主动在网络中寻找邻居，必须手工指定 BGP 邻居的地址，那么 BGP 才会将数据包发往指定的地址来请求建立邻居，与此同时，BGP 发出的请求数据包除了写明目标 IP 地址外，还要写上自己的 IP 地址，即 BGP 源地址。路由器自己产生流量后从接口发出时，流量从哪个接口被发出，那么这些数据包的源 IP 地址就是哪个接口的地址。因此当 BGP 发出数据包寻找邻居时，这些数据包从哪个接口被发出，那么 BGP 源 IP 地址就是哪个接口的地址。要两台 BGP 路由器要正常建立邻居，必须双方路由器都相互指定邻居，相互发送数据包才行。当一台 BGP 路由器收到建立邻居的请求后，如果发现数据包的目标 IP 不是自己的 BGP 源地址，那么就拒绝该连接请求，只有当请求数据包的目标 IP 与自己的 BGP 源地址相同时，才可建立 BGP 邻居。需要注意的是，这个条件只在两个邻居之间，任意一个邻居满足条件即可，并不需要双方都满足，也就是说一方收到的数据包目标 IP 与自己的 BGP 源地址相同即可，另一方收到的数据包目标 IP 与它的 BGP 源地址不同也没关系，只要单方面符合条件就行，但我们通常都将 BGP 两端的源与目标保持一致。BGP 的源地址是可以随意更改的，但只能是路由器上的接口地址。

如下图



在上图中，R1 与 R2 之间有两条链路，当配置 BGP 邻居时，如果 R1 指定邻居地址为 12.1.1.2，R2 指定邻居地址为 12.1.1.1，那么在建立邻居过程中，R1 将请求数据包从接口 F0/0 发出，数据包的目标 IP 为 12.1.1.2，BGP 源地址为 F0/0 的接口地址 12.1.1.1， 当 R2 将请求数据包从接口 F0/0 发出时，数据包的目标 IP 为 12.1.1.1，BGP 源地址为 F0/0 的接口地址 12.1.1.2，由于 R1 发出的数据包目标 IP12.1.1.2 与 R2 的 BGP 源地址 12.1.1.2 完全相同，所以最终能够正常建立 BGP 邻居。R1 在检测地址时，R2 的目标 IP 与 R1 的源也完全相同，通常我们都保证双方一致。

当 R1 与 R2 之间的直连接口 F0/0 中断后，如果双方将数据包从 S0/0 发出，那么 R1 的源地址就是 10.1.1.1，R2 的源地址就是 20.1.1.2，由此可以看出，双方发出的数据包的目标 IP 都与对方的源地址不符，所以无法建立 BGP 邻居。虽然在上面的网络环境中，双方路由器之间都拥有多条链路，在中断某条链路之间，仍然可以通信，但是这并不能保证 BGP 邻居的永久连接。为了使拥有多条链路的 BGP 邻居之间永远保持连接，考虑到路由器的 loopback 口在设备正常工作的情况下，不会像物理接口那样出现中断，所以建议在 BGP 邻居之间使用 loopback 接口的地址来建立 TCP 连接，当指定邻居时，不再将邻居的地址指定为对方物理接口地址，而改为指定对方的 loopback 地址，这样一来，既然物理接口中断，只要还有通畅的链路，那么 BGP 邻居仍然可以保持连接。在将 BGP 邻居地址指定为对方 loopback 地址时，为了使数据包的目标 IP 与对方的 BGP 源地址相同，所以邻居也要将 BGP 源地址更改为自己的 loopback 接口地址，从而使得双方正常建立 BGP 连接。

在上图中，当 R1 指定邻居地址为 2.2.2.2，BGP 源地址为 1.1.1.1，而 R2 指定邻居地址为 1.1.1.1，BGP 源地址为 2.2.2.2，这样一来，双方的目标 IP 都与对方的 BGP 源地址相同，所以可以正常建立邻居，并且在双方链路中，任何一条链路断开，都不影响邻居的会话，BGP 的连接仍然保持而不会中断，实现了连接的冗余性和稳定性。

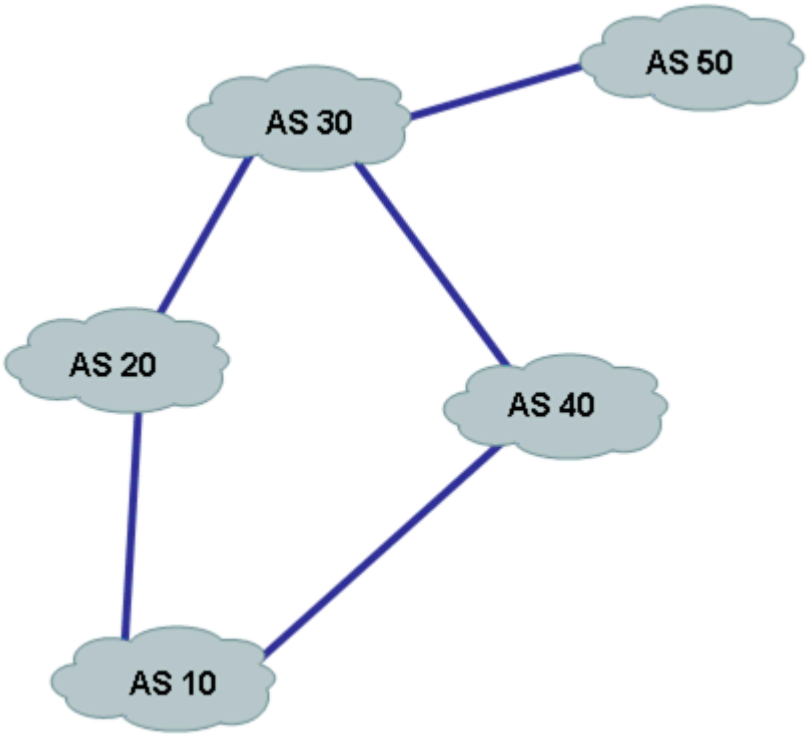
BGP TTL

一台 BGP 路由器只属于一个 AS，在建立 BGP 邻居时，如果对方路由器和自己属于相同 AS，即在同一自治系统内部，则邻居关系为 internal BGP（iBGP），如果属于不同 AS，即邻居在自治系统外部，则邻居关系为 external BGP（eBGP）。考虑到外部自治系统的路由器对 BGP 发起 DOS 攻击，所以 BGP 要求外部 BGP 邻居，即 eBGP 邻居必须与自己直连，而 iBGP 邻居可以任意距离。这些控制是通过控制 BGP 数据包的 TTL 值来实现的，将建立 eBGP 邻居时发出的数据包的 TTL 值限制为 1，就限制了 eBGP 邻居必须直连，而由于 iBGP 邻居可以在任意位置，所以建立 iBGP 邻居时发出的数据包的 TTL 值为最大，即 255。对于建立 eBGP 的数据包的 TTL 值可以随意修改，甚至改为 255 都行。

BGP AS_Path

BGP 的路由可能会从一个 AS 发往另外一个 AS，从而穿越多个 AS。但是由于运行 BGP 的网络会是一个很大的网络，路由从一个 AS 被发出，可能在经过转发之后，又回到了最初的 AS 之中，最终形成路由环路，所以出于防止环路的目的考虑，BGP 在将路由发往其它 AS 时，也就是发给 eBGP 邻居时，需要在路由中写上自己的 AS 号码，下一个 AS 收到路由后，再发给其它 AS 时，除了保留之前的 AS 号码之外，也要添加上自己的 AS 号码，这样的写在路由中的 AS 被称为 AS-path，如果 BGP 收到的路由的 AS_PATH 中包含自己的 AS 号码，就认为路由被发了回来，以此断定出现了路由环路，最后就会丢弃收到的路由。BGP 只有在将路由发给 eBGP 时，才会在 AS-path 中添加自己的 AS 号码，而在发给 iBGP 时，是不会添加 AS 号码的，因为 iBGP 邻居在同一个 AS 中，即使要添加，AS 号码全是一样的，所以没有必要。

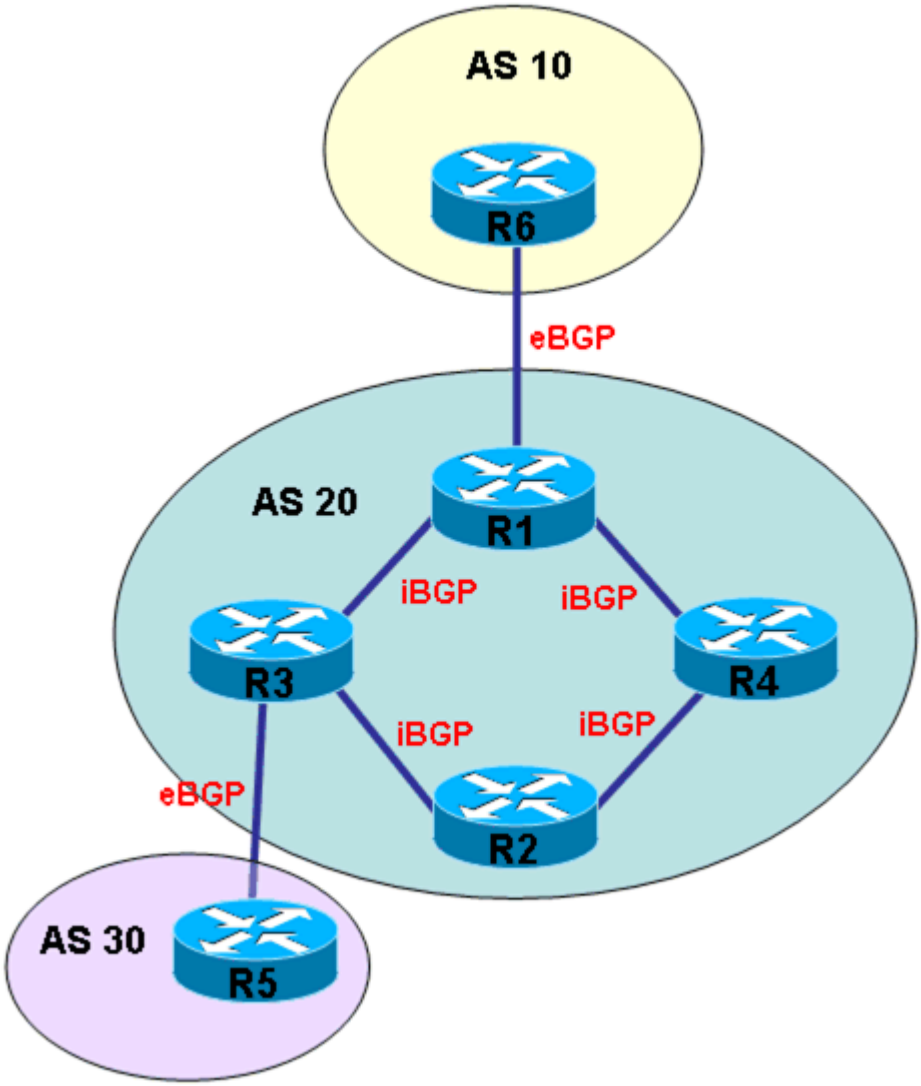
如下图：



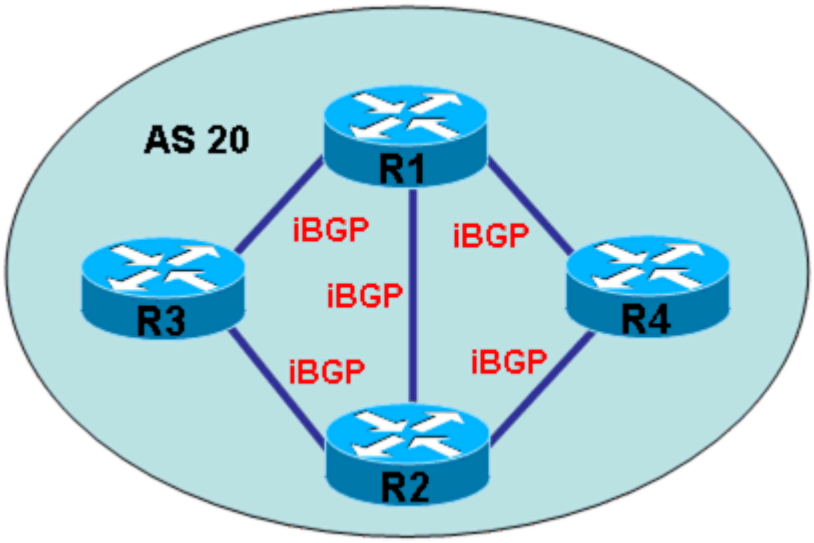
在上图中，当路由穿越各个 AS 时，所有发给 eBGP 邻居的路由，都会在 AS-path 中添加自己的 AS，自己的 AS 总是添加在 AS-path 的最前面。例如一条路由从 AS 10 被发往 AS 20，则 AS-path 为“10”，当 AS 20 将路由发往 AS 30 时，添加上自己的 AS 号码 20 之后，AS-path 变成“20,10”，当 AS 30 将路由发往 AS 50 时，最终 AS 50 收到的路由的 AS-path 为“30,20,10”。当 AS 30 将路由发给 AS 40，AS 40 再将路由发给 AS 10 时，路由的 AS-path 为“40,30,20,10”，由于 AS 10 在收到路由后，发现 AS-path 中包含自己的 AS 号码 10，所以认为出现环路，便丢弃收到的所有路由。

在 IGP 协议中，我们将路由协议分成两大类：距离矢量路由协议和链路状态路由协议，而 BGP 被划分为路径矢量路由协议（path-vector routing），路径矢量算法结合了距离矢量特性与 AS-path 防环特性。

因为 BGP 在将路由发给 eBGP 邻居时，会将自己的 AS 号码添加到 AS-path 中，所以可以以此来防止环路，而在将路由发给 iBGP 时，是不会往 AS-path 添加 AS 号码的，因此在 iBGP 之间传递路由时，没有防止环路的机制。考虑到为 iBGP 之间的路由传递也加入防环机制，因而强制将 BGP 路由在 AS 内部只传一跳，具体操作为：一台 BGP 路由器从 eBGP 邻居收到路由，发给 iBGP 邻居之后，iBGP 邻居收到后就不能再传给其它任何 iBGP 邻居，只能传递给 eBGP 邻居。此规则被多数人称为 BGP 的水平分割，就是一台 BGP 路由器从 iBGP 邻居收到的路由，不能传递给其它 iBGP 邻居，只能传给 eBGP 邻居。如下图：



在上图中，当 R1 从 eBGP 邻居 R6 那里收到路由后，可以发给任何 iBGP 邻居，包括 R3 和 R4，当 R3 从 iBGP 邻居 R1 那里收到路由后，不可以转发给任何 iBGP 邻居，只可以转发给 eBGP 邻居，所以 R3 从 R1 收到路由后，只能转发给 eBGP 邻居 R5。由于 R3 和 R4 从 R1 收到路由后，都不可以转发给 iBGP 邻居 R2，在上图环境中，最终造成 R2 无法接收任何路由，要让 R2 收到路由，建议在 R1 与 R2 之间再建立一条 BGP 会话，所以如此一来，在同一个 AS 中，要将路由全网传递，就需要在 iBGP 邻居之间配置全互联，最终 AS 20 内的邻居关系如下图：



在 AS 内部,除了建立全互联的 iBGP 邻居关系外,还可以通过 BGP Reflector(BGP 反射器)和 BGP Confederation（BGP 联邦）的方式来实现路由全网传递，将在后续介绍。

BGP 路由表

当路由器之间建立 BGP 邻居之后，就可以相互交换 BGP 路由。一台运行了 BGP 协议的路由器，会将 BGP 得到的路由与普通路由分开存放，所以 BGP 路由器会同时拥有两张路由器，一张是存放普通路由的路由表，被称为 IGP 路由表，就时平时我们使用命令 show ip route 看到的路由表，IGP 路由表的路由信息只能从 IGP 协议和手工配置获得，并且只能传递给 IGP 协议；另外一张就是运行 BGP 之后创建的路由表，称为 BGP 路由表，需要通过命令 show ip bgp 才能查看，BGP 路由表的路由信息只能传递给 BGP 协议，如果两台 BGP 邻居的 BGP 路由表为空，就不会有任何路由传递。在初始状态下，BGP 的路由表为空，没有任何路由，要让 BGP 传递相应的路由，只能先将该路由导入 BGP 路由表，之后才能在 BGP 邻居之间传递。默认情况下，任何路由都不会自动进入 BGP 路由表，BGP 路由表的路由获得有多种方式，可以从 BGP 邻居获得，也可以手工将 IGP 路由导入 BGP 路由表，还可以将其它路由重分布进 BGP，只要 BGP 的路由不是从邻居学习到的而是手工导入的，那么这样的路由被称为 BGP 本地路由。

因为 BGP 的邻居类型分为两种：eBGP 和 iBGP，所以 BGP 路由的 AD 值也有区分，如果 BGP 的路由是从 eBGP 学习到的，AD 值为 20，可以发现，从 eBGP 邻居学习到的路由，将优于任何 IGP 协议；从 iBGP 学习到的路由的 AD 值为 200，同样可以发

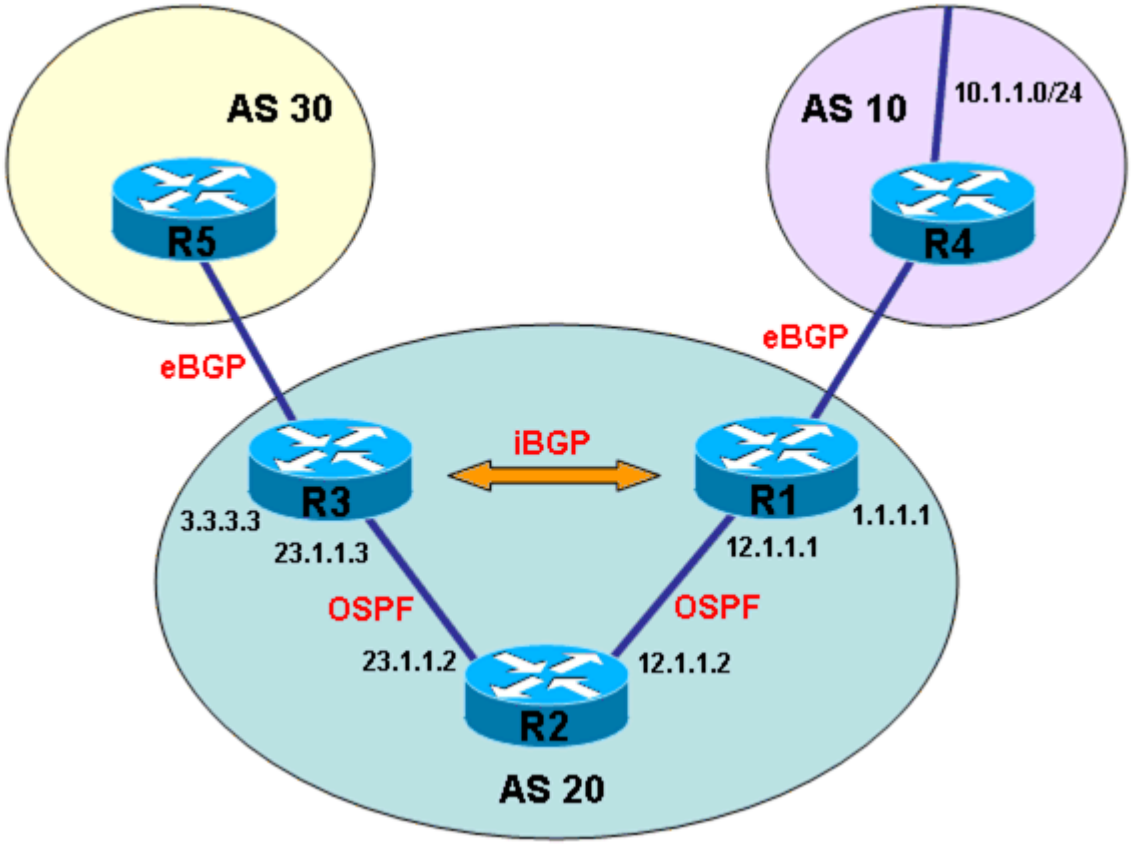
现，此类路由的优先级低于任何 IGP 协议。BGP 除了以上两种 AD 值之外，如果 BGP 路由是从本地手工导入的，即 BGP 本地路由，则 BGP 本地路由的 AD 值为 200，与 iBGP 路由的 AD 值相同，优先级低于任何 IGP 协议。

如果某一条相同的路由同时从 eBGP 和 iBGP 以及本地路由学习到，那么究竟哪条路由会被选择为最优路径呢？路由的 AD 值并不一定会影响到路径选择，因为 BGP 并不会在一开始，就通过比较 AD 值来选择最优路径。

BGP Synchronization

BGP 邻居之间的通信与交流运行在 TCP 的基础上，在两个节点之间，只要网络是通的，就能够建立 TCP 建立，网络的连通，可以是任何 IGP 路由协议，甚至是静态路由，总之，只要网络是通的，TCP 连接就一定能够建立起来。只要让两台路由器之间连通，保证 TCP 能够正常连接，就能够保证 BGP 的通信。在一个 AS 中，除了需要建立 BGP 连接之外，同时还需要运行 IGP 协议，其中运行 BGP 的目的是为了在大型网络中传递庞大的路由表或路由信息，而运行 IGP 协议的目的可想而知，并不是为了传递庞大的路由信息，在 AS 中运行 IGP 的根本目的是为了让 BGP 路由器之间能够建立 TCP 连接，从而为 BGP 的通信服务。因此可以看出，BGP 就像是一辆运货的卡车，BGP 的路由就是卡车要运的货，而 IGP 协议就是为了在站与站之间铺平道路，如果没有 IGP 去让道路连通，那么 BGP 就无法在站与站之间运送货物。

因为 BGP 在建立邻居时，BGP 的源地址可以是任意地址，这些地址可以不是直连的，只要是能通信的，能建立 TCP 连接即可。当 BGP 在向邻居发送流量时，只要将流量发往邻居的对端地址，因为邻居的地址并不一定是直连的，所以要找到去往邻居地址的路径，可能需要查询 IGP 路由表，因为 IGP 为 BGP 的通信与连接提供了保证。由此可见，BGP 要将数据发给邻居，BGP 在查询去往邻居的路径时，采用的是递归查询，BGP 查询去往邻居的过程中，可能要多次查询 IGP 路由表，只要在 IGP 路由表中找到了去往邻居地址的相应路径或相应下一跳，那么就会将数据发给这个下一跳。



在上图的网络中，R1 与 R4 建立 eBGP 连接，R3 与 R5 建立 eBGP 连接，而 R1 与 R3 建立 iBGP 连接。在 R1 与 R3 建立 iBGP 连接时，R1 通过目标地址 3.3.3.3 找到邻居 R3，R1 的 BGP 源地址为 1.1.1.1，而 R3 也通过目标地址 1.1.1.1 找到邻居 R1，R3 的 BGP 源地址为 3.3.3.3，为了让 1.1.1.1 和 3.3.3.3 能够正常通信，从而建立 TCP 连接，R1、R2、R3 之间启用了 IGP 协议 OSPF，OSPF 的目的只是为了使 1.1.1.1 能够与 3.3.3.3 通信，并不传递 AS 中庞大的路由信息。

当 AS 10 中的 R4 将网段 10.1.1.0/24 通告给 AS 20 中的 R1 后，因为 R1 与 R3 之间是 iBGP 邻居，所以 R1 将路由 10.1.1.0/24 传递给 R3，最终 R3 将路由 10.1.1.0/24 传递给 AS 30 中的 R5。当 R5 将目的地为 10.1.1.0/24 的流量发给 R3 时，R3 在查询路由表后得知，去往 10.1.1.0/24 的数据包需要发给 iBGP 邻居 1.1.1.1 才能够到达，于是 R3 便执行递归查询，查询如何去往 1.1.1.1，正因为 R1 与 R3 之间的通信是靠 OSPF 提供的，所以 R3 得知去往 1.1.1.1 必须将数据包交给 R2，即交给下一跳 23.1.1.2，因为 R2 只运行了 OSPF 为 BGP 服务，所以 R2 没有 BGP 的路由 10.1.1.0/24，当 R2 发现数据包的目标地址为 10.1.1.0/24 后，只能将数据包全部丢弃，这就类似于路由黑洞。

从以上情况中可以看出，当 BGP 从 iBGP 收到路由时，因为邻居之间可能跨越了多台 IGP 路由器，所以 BGP 在将数据包发往目的地时，通常会发给一台只运行了 IGP 的路由器，而只运行 IGP 的路由器并没有 BGP 的路由，因而最终导致数据包丢失，

造成路由黑洞。要杜绝此类问题的发生，其实答案很明了，就是让 AS 中只运行 IGP 的路由器同时也拥有 BGP 的路由表即可。由于以上原因，在 BGP 路由传递中，有以下一条规则：当 BGP 要将从 iBGP 邻居学习到的路由信息传递给其它邻居之前（这个邻居通常是 eBGP 邻居），这些路由必须在 IGP 路由表中也能学到，否则认为此路由无效而不能发给其它邻居。

此规则称为 iBGP 与 IGP 路由同步。

在上图环境中,在 R3 将从 iBGP 邻居 R1 学习到的路由传递给 eBGP 邻居 R5 之前,必须确定这条路由在自己的 IGP 路由表中也存在，否则不使用该路由。要查看路由在 IGP 路由表中是否存在，使用命令 show ip route 即可。

注意，只有从 iBGP 邻居学习到的路由，才受 iBGP 与 IGP 路由同步规则的限制,如果路由是从 eBGP 邻居学习到的，则不受此规则限制，并且此规则可以手工开启或关闭。

BGP 同步默认是开启的，在 IOS 12.2(8)T 以及之后的版本默认都是关闭的。

Path Attributes

在默认情况下，到达同一目的地，BGP 只走单条路径，并不会在多条路径之间执行负载均衡。对于 IGP 路由协议，当有多条路径可以到达同一目的地时，则根据最小 metric 值来选择最优路径，而 BGP 存在多条路径到达同一目的地时，对于最优路径的选择，BGP 并不会以 metric 值大小为依据，BGP 对于最优路径的选择，需要靠比较路由条目中的 Path Attributes，即路径属性，只有在比较多条路由的属性之后，才能决定选择哪条为最优路径。BGP 的每条路由都带有路径属性，BGP 的路径属性可以划分为以下四类：

公认强制 （Well-Known Mandatory）

公认自选 （Well-Known Discretionary）

可选可传递 （Optional Transitive）

可选不可传递（Optional Nontransitive）

对于各属性的各特征如下：

公认强制 （Well-Known Mandatory）

对于任何一台运行 BGP 的路由器，都必须支持公认强制属性，并且在将路由信息发给其它 BGP 邻居时，必须在路由中写入公认强制属性，这些属性是被强制写入路由中的，一条不带公认强制属性的路由被 BGP 路由器被视为无效而被丢弃，一个不支持公认强制属性的 BGP，是不正常的，不合法的 BGP。

BGP 路由必须携带的公认强制属性有三个：Origin，Next_Hop，AS-path。

公认自选 （Well-Known Discretionary）

公认自选属性并不像公认强制属性那么严格，任何一台运行 BGP 的路由器都必须支持公认自选属性，必须理解和认识公认自选属性，但是为路由写入公认自选属性并不是必须的，是否要为路由写入公认自选属性可以自由决定，为路由写上公认自选属性之后，所有 BGP 路由器都能认识和理解，并且都会自动保留和传递该属性。

可选可传递 （Optional Transitive）

并不是所有运行 BGP 的路由器都能理解和支持可选可传递属性，路由的可选可传递属性是任意写入的，其它 BGP 路由器并不一定能理解，也并不一定能保留和传递该属性，但是当为路由设置了可选可传递属性后，可以明确要求 BGP 路由器保留和传递该属性。

可选不可传递（Optional Nontransitive）

只有特定的 BGP 路由器才理解和支持可选不可传递属性，并且可选不可传递属性理论上是不能手工设置的，即使手工设置了可选不可传递属性，这些属性也不能任意传递，只可以传递到特定的 BGP 路由器。

以下是一些常用 BGP 属性的介绍：

1. Origin（公认强制属性）：

在路由器之间建立 BGP 邻居之后，邻居之间只能相互传递 BGP 路由表中的路由，在初始状态下，BGP 的路由表为空，没有任何路由，要让 BGP 传递相应的路由，只能先将该路由导入 BGP 路由表，之后才能在 BGP 邻居之间传递。默认情况下，任何路由都不会自动进入 BGP 路由表，只能手工导入，对于路由是怎么进入 BGP 路由表的，这种方式会被记录在路由条目中，称为 Origin 属性，Origin 属性就反映出了路由是如何进入 BGP 路由表的。要将路由导入 BGP 路由表，有三种方式，

第一：因为路由器上默认会有 IGP 路由表，通过命令 show ip route 可以查看到，这些 IGP 表中的路由可以被手工导入 BGP，通过在 BGP 进程模式下使用命令 network，即可将 IGP 表中的相应路由导入 BGP 路由表，并且需要指定掩码，只有 network 后面的网段和掩码在 IGP 路由表中能找到时，才会进入 BGP 路由表，并不能通过这种方式将一条不存在的路由凭空导入 BGP，通过命令 network 被导入 BGP 的路由的 Origin 属性为 IGP 属性。

第二：BGP 可以从 EGP 路由协议中获得路由信息，而 EGP 已经被淘汰，被 BGP 所取代，所以我们很难遇见 EGP 路由协议，从 EGP 路由协议获得的路由的 Origin 属性为 EGP。

第三：BGP 路由表除了从 IGP 和 EGP 获得路由外，还可以将路由重分布进 BGP 路由表，而重分布的路由的 Origin 属性为 Incomplete。

当 BGP 路由表中到达同一目的地存在多条路径时，会通过比较路由的 Origin 属性来选择最优路径，它们的优先级为 IGP 优于 EGP，EGP 优于 Incomplete，即 IGP>EGP>Incomplete。

2. AS_Path（公认强制属性）：

AS_Path 中包含了 BGP 路由器到达目的地所经过的所有 AS 的集合，AS_Path 中会包含了多个 AS 号码，号码的多少，逻辑上反映了到达目的地的远近。

AS_Path 还能细分为：

AS_SEQUENCE（有序的 AS 号码，即 AS 号码在 AS_Path 中是按一定顺序排列的）

AS_SET（无序的 AS 号码，即 AS 号码在 AS_Path 中的排列是没有顺序的，通常是将多条拥有不同 AS_Path 的路由汇总后产生的）

当 BGP 路由表中到达同一目的地存在多条路径时，会优选 AS-Path 最短的路径。

3. Next_Hop（公认强制属性）：

也就是 BGP 将数据包发往目的地的下一跳，BGP 路由的下一跳，就是 BGP 建立邻居时的地址，也是 BGP 之间建立 TCP 连接所使用的地址。因为这个地址可以是路由器上任意接口的地址，是要能通信即可（其连通性由 IGP 提供保证），所以 BGP 在将数据包发往下一跳时,通常需要采用递归查询在 IGP 路由表中查询该下一跳地址。默认情况下，一台 BGP 路由器将路由传递给 eBGP 邻居时，会将 Next-hop 属性改为自己的地址，也就是和对方建立邻居所使用的地址，而在将路由传递给 iBGP 邻居时，不会改变 Next-hop 属性。

对于将路由传递给 BGP 邻居时，是否改变 Next-hop 属性的功能，可以自由关闭或启用。

BGP 路由表中由本地产生的路由而不是从 BGP 邻居学习来的，即本地发起路由的 Next-hop 属性都为 0.0.0.0。

4. Local_Pref（公认自选属性）：

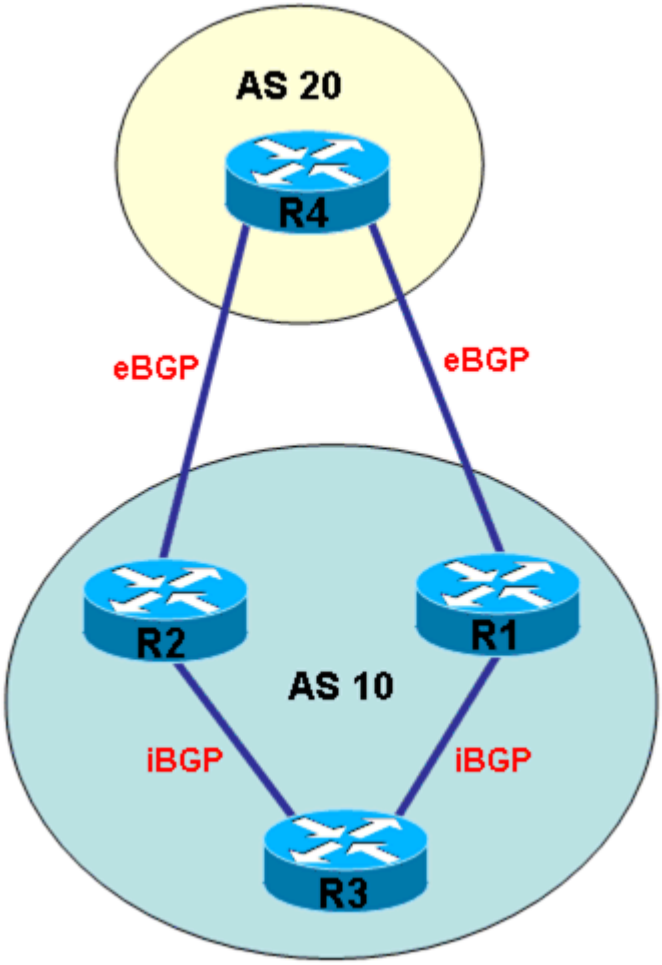
Local_Pref 称为本地优先级，其中的（Local）本地就是指本 AS，或 AS 内的意思，所以可以想象得出，Local_Pref 属性的传递范围，只在同一个 AS 内有效，一条路由的 Local_Pref 属性只能在同一 AS 内部传递，出了 AS 后就会被还原成默认值。

Local_Pref 属性在 BGP 邻居之间是自动传递的，只有在将路由发给 iBGP 时才会传递，而在发给 eBGP 时，是没有 Local_Pref 值的，一条路由的 Local_Pref 属性在一个 AS 内的所有 BGP 路由器上是完全相同的。Local_Pref 的默认值为 100，由此可以看出，一条路由在 AS 内的所有路由器上默认值为 100。

本地优先级属性是用于区分到同一目的地的各个路由优先程度的。本地优先级越高，路由优先级越高。默认值为 100。

当 BGP 路由表中到达同一目的地存在多条路径时,会比较 Local_Pref 值的大小，Local_Pref 值大的会被选为最优路径，如 110 与 100，那么 110 会被选为最优路径。

Local_Pref 值可以被随意修改，修改后将在整个 AS 内传递，所以推荐使用 Local_Pref 属性来控制一个 AS 的路由器去往目的地在其它 AS 的路径。如下图：



在上图中，AS 10 中的 BGP 路由器 R3 可以同时通过 R1 与 R2 去往目的地在 AS 20 中的 R4 上时，可以通过在 AS 10 内部修改路由的 Local_Pref 值来影响选路，比如在 R1 上将路由的 Local_Pref 值改为 110，而路由器 R2 上不作任何改动，最终 R3 将选择从 R1 去往 AS 20，因为 R1 的 Local_Pref 值为 110，而 R2 的 Local_Pref 值为 100（默认），所以 R3 选择 R1 为最优路径。

因为 R1 和 R2 在将路由发给 iBGP 邻居 R3 时会携带 Local_Pref 属性，所以 R3 同时比较 iBGP 邻居 R1 与 iBGP 邻居 R2 时，才合适使用 Local_Pref 属性，因为下一跳都是 iBGP 邻居，如果下一跳不都是 iBGP 邻居，并不建议修改 Local_Pref 属性来影响选路。

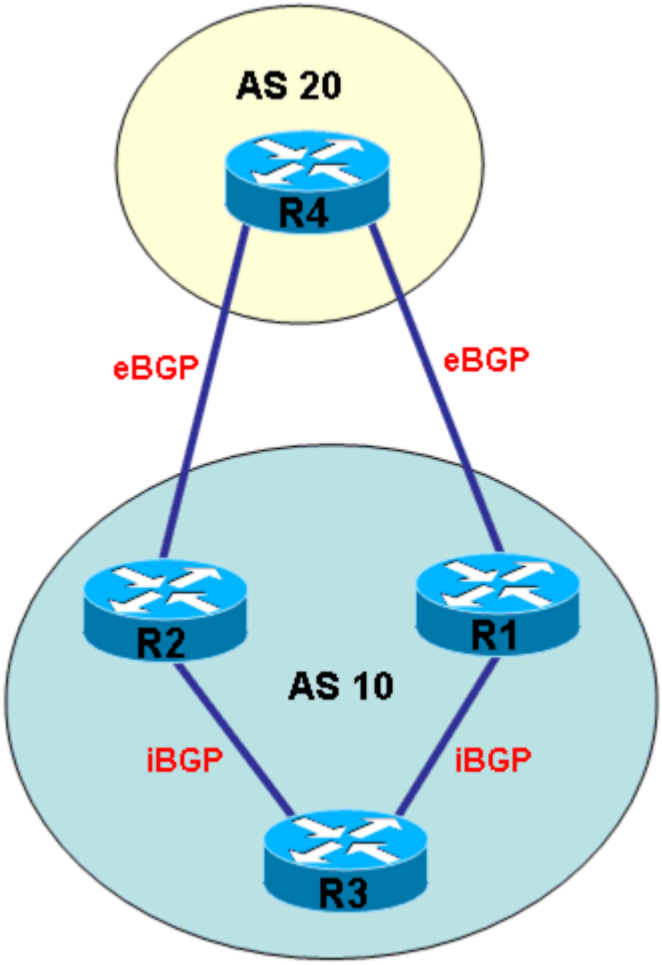
5. MULTI_EXIT_DISC（MED，可选不可传递属性）：

MED 就是 BGP 路由中的 metric，是被设计用来影响在多个下一跳都为 eBGP 邻居时，如何选择最优路径，因为在多个下一跳都为 iBGP 时，是建议使用修改 Local_Pref

属性来影响选路的，而多个下一跳都为 eBGP 时，则使用 MED。MED 是 BGP 路由的 metric，所以多条路径中拥有最小 MED 值的路径会被优先使用。MED 默认值为 0。

Local_Pref 属性只在同一个 AS 内部传递，而 MED 只能在 AS 之间传递，只有在将路由发给 eBGP 邻居时，才会传递 MED，在发给 iBGP 时，是不会传递 MED 的。当一条路由被设置 MED 值后传递给 eBGP 邻居，在 eBGP 邻居收到后，如果将该路由继续传递给 iBGP 邻居，那么这个值会被还原为默认值 0，也就是说同一个 AS 内，所有发给 iBGP 邻居的路由的 MED 值都为 0，这是为了让所有 AS 内部路由器都能够拥有相同的选路结果。

MED 值也是可以随意修改的。



在上图中，当 AS 20 中的路由器 R4 要去往目的地为 AS 10 的网段时，由于下一跳 R1 与 R2 都为 eBGP 邻居,所以可以通过修改 MED 值来影响 R4 对于下一跳的选择。比如将 R2 的 MED 改为 10，而 R1 的 MED 保持默认不变，那么最终 R4 将选择 R1 去往 AS 10 中的目的地，因为 R1 的 MED 值 0 小于 R2 的 MED 值 10，所以被优先使用。

默认情况下，只有当去往目的地的多个下一跳 eBGP 邻居都为相同 AS 时，才会比较 MED 值，如果多个 eBGP 邻居为不同 AS 时，是不会比较 MED 的，若是要强制在多个不同的 eBGP 邻居之间比较 MED 值，需要在 BGP 进程下输入命令：bgp always-compare-med。

6.Weight

Weight 属性为 Cisco 私有属性，只有 Cisco 的路由器才能认识和理解 Weight。路由的 Weight 属性只在路由器本地起作用，BGP 将路由传递给邻居时，并不会保留 Weight。Weight 值的范围为 0~65535，默认为 0，如果是 BGP 本地路由，则 Weight 值为 32768。可以手工任意修改路由的 Weight 值，可以对路由进行修改，也可以对整个邻居进行修改，但也只能对本地起作用，路由的 Weight 值并不会传递给邻居。

当 BGP 路由表中到达同一目的地存在多条路径时,会优选 Weight 值最大的路径。在 Cisco 路由器中，比较最优路径的第一条规则就是比较 Weight 值，所以只要改动 Weight 值，就绝能够控制 Cisco 路由器的 BGP 选路。

其它属性，在应用中继续介绍。

BGP RIB-Failure

在 BGP 的路由表中，并非所有的路由都会被 BGP 使用，默认情况下，BGP 到任何目的地，只选择单一路径。在 BGP 路由表中，只有最优路由才会被 BGP 使用，也只有最优路由才会发给 BGP 邻居，需要说明的是，思科官方强调 BGP 会将所有路由发给邻居，请以实际为准。BGP 路由要被标为最优路由必须达到以下两个条件：

★下一跳可达

★如果是从 iBGP 收到的路由，则必须满足 IGP 与 iBGP 同步，除非该规则已被关闭。

如果某 BGP 路由的状态为 RIB-Failure,则是不能被使用的,被定为 RIB-Failure 的原因有：

★该路由在 IGP 中已经拥有比 BGP 更高优先级的 AD 值。

★内存错误

★超出 VRF 中的路由限制数。

而 BGP 建立邻居的条件如下：

★双方需要建立邻居的 IP 地址在网络上是互通的，可以建立 TCP 会话。

★双方指定的 AS 号码必须匹配

★双方 BGP 数据包必须可达（eBGP 默认 TTL 为 1，需要注意）。

★对方 BGP 数据包的目的 IP 和自己的源 IP 必须相同（单向满足即可）

BGP 最优路径选择

在默认情况下，到达同一目的地，BGP 只走单条路径，并不希望在多条路径之间执行负载均衡。当 BGP 路由表中有多条路径可以到达同一目的地时，需要靠比较路由条目中的路径属性，只有在比较多条路由的属性之后，才能决定选择哪条为最优路径。BGP 的每条路由都带有路径属性，对于通过比较路径属性来选择最优路径，BGP 需要在多条路径之间按照一定的顺序比较属性，当多条路由的同一属性完全相同时，需要继续比较顺序中的下一条属性。BGP 在选择最优路径时，需要按照以下顺序来做比较：

1.最高 Weight 值

（选择最高 Weight 值的路由，Weight 值为 Cisco 路由器特有，并且只在本地路由器有效，默认 Weight 值为 0，本地发起路由为 32768。）

2.最高 LOCAL_PREF 值

（如果 Weight 值相同，则选择拥有最高 LOCAL_PREF 值的路由，默认为 100。）

3.本地发起路由

（如果 LOCAL_PREF 值相同，则选择 BGP 本地发起的路由，换句话说，也就是下一跳为 0.0.0.0 的路由，本地发起的路由有多种方式，如通过在 BGP 进程下命令 network 命令从 IGP 路由表导入，将其它路由协议重分布进 BGP 路由表，最后是汇总路由表。而通过命令 network 和重分布的路由优先于手工汇总的路由。）

4.最短 AS_PATH

（如果本地发起路由无法比出最优路径，则选择拥有最短 AS_PATH 的路由，但是可以跳过这一步,输入命令 bgp bestpath as-path ignore 后,就会忽略对 AS_PATH 的比较，而直接比较下一属性。需要更加注意的是，AS_SET 被认为是 1 个 AS，而无论 AS_SET 中包含多少个 AS，并且 BGP 联邦内部 AS 不被计算。）

5.最低 Origin 类型

（如果 AS_PATH 无法比出最优路径,则选择拥有最低 Origin 类型的路由,Origin 表示路由最初是如何进入 BGP 路由表的,目前有三种进行 BGP 路由表的方法,从 IGP 导入,从 EGP 学习,以及重分布,它们的优先级为 IGP 优于 EGP,EGP 优于 Incomplete,即 IGP>EGP>Incomplete。

6.最小 MED 值

（如果 Origin 类型无法比出最优路径，则选择拥有最小 MED 值的路由，并且只有当多个下一跳邻居在同一 AS 时才比较 MED 值。如果要在多个不同 AS 的下一跳中比较 MED，可在 BGP 进程中输入命令 bgp always-compare-med，注意须保证此命令

在整个 AS 的路由器上输入，否则可能产生路由环路。默认的 MED 值为 0，如果收到一条没有 MED 的路由，也认为是 0。）

7.eBGP 优于 iBGP

（如果 MED 值无法比出最优路径，则选择下一跳为 eBGP 的邻居而不选择 iBGP 邻居。都知道 eBGP 的路由 AD 值为 20，而 iBGP 的路由 AD 值为 200，但 BGP 并不在 eBGP 与 iBGP 之间比较 AD 值，并且在比到此步时，邻居类型才影响了最优路径的选择，这种影响是受邻居类型的影响，而不是受 AD 值的影响。注意：BGP 联邦内部没有此规则。）

8.最小 IGP metric 到达下一跳的路由

（如果多条路径的下一跳邻居同为 eBGP 或 iBGP，则选择拥有最小 IGP metric 到达下一跳的路由。）

9.负载均衡（如果开启的话）

（BGP 并不是不能负载均衡的，如果之前的属性都无法选出最优路径，则执行负载均衡，但必须是之前的所有属性均完全相同，缺一不可。需要注意：只有负载均衡功能开启了，BGP 才会执行负载均衡，否则，继续比较下一属性。）

在开启负载均衡功能时，在 BGP 进程下输入以下命令：

maximum-paths n （多条路径的下一跳邻居都为 eBGP 时，输入此命令）

maximum-paths ibgp n（多条路径的下一跳邻居都为 iBGP 时，输入此命令）

其中 n 为执行负载均衡的路径数量，最大值为 6，默认为 1，也就是不执行负载均衡。

如果在 eBGP 和 iBGP 邻居之间同时执行负载均衡，输入命令 maximum-paths eibgp n，此命令只支持在 Ipv4 VRF 模式下输入，就是只能支持 MPLS VPN 下的 eBGP 和 iBGP 邻居之间负载均衡。

10.如果下一跳都为 eBGP，则选择最早学习到的路由（即时间最长的路由）

（为了避免路由翻动，所以选择最早学习到的路由，如果要忽略比较路由学习到的时间长短，可在 BGP 进程下输入命令 bgp best path compare-routerid，某些 IOS 已经自动加入此命令，并且不能删除。如果多条路由拥有相同的 Router-ID，比如路由是从同一个邻居学习到的，同样也会忽略比较路由学习到的时间长短。）

11.最低 Router-ID 下一跳

（BGP 的 Router-ID 选举如同 OSPF，在此步，拥有最低 Router-ID 的下一跳路由将被选为最优路径。）

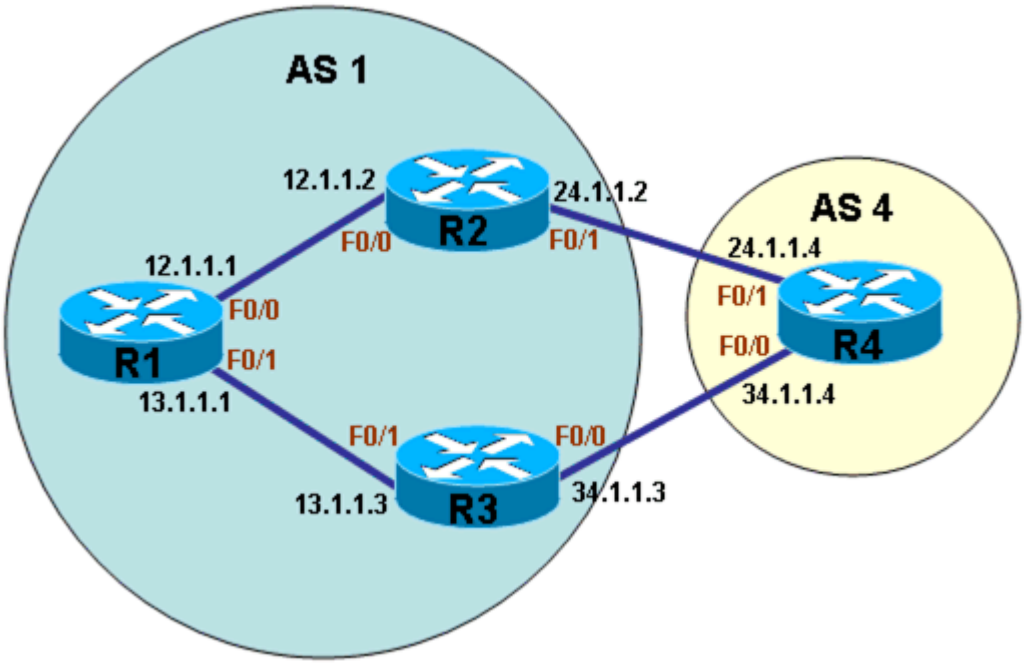
12.最短 cluster list（如同 AS_PATH）

[cluster list 只在 BGP reflector (RR)的环境下才有，功能如同 AS_PATH]

13.最小下一跳的邻居地址

（如果比较之前的所有属性都无法选出最优路径，最终选择下一跳的邻居地址最小的路由，这个地址就是在建立邻居时所指的地址，也是邻居和自己建立 TCP 连接所使用的源地址，建立不同邻居，不可能使用相同地址，所以不可能两个不同路径的邻居地址是相同的，在这一步一定能够选出最优路径。）

BGP 基础实验



说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32 Loopback 11 11.1.1.1/24

R2 Loopback 0 2.2.2.2/32 Loopback 22 22.2.2.2/24

R3 Loopback 0 3.3.3.3/32 Loopback 33 33.3.3.3/24

R4 Loopback 0 4.4.4.4/32 Loopback 44 44.4.4.4/24

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的，以此来作为 BGP 的连接地址。

1. IGP 使全网 Loopback 0 互通

说明：使用 OSPF 保证 Loopback 0 之间的通信，从而建立 BGP 连接。

（1）配置各路由器的 OSPF

R1：

```
r1(config)#router ospf 1
r1(config-router)#router-id 1.1.1.1
r1(config-router)#network 12.1.1.1 0.0.0.0 area 0
r1(config-router)#network 13.1.1.1 0.0.0.0 area 0
r1(config-router)#network 1.1.1.1 0.0.0.0 area 0
```

R2：

```
r2(config)#router ospf 1
r2(config-router)#router-id 2.2.2.2
r2(config-router)#network 12.1.1.2 0.0.0.0 area 0
r2(config-router)#network 24.1.1.2 0.0.0.0 area 0
r2(config-router)#network 2.2.2.2 0.0.0.0 area 0
```

R3：

```
r3(config)#router ospf 1
r3(config-router)#router-id 3.3.3.3
r3(config-router)#network 13.1.1.3 0.0.0.0 area 0
r3(config-router)#network 34.1.1.3 0.0.0.0 area 0
```

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

R4：

```
r4(config)#router ospf 1
r4(config-router)#router-id 4.4.4.4
r4(config-router)#network 24.1.1.4 0.0.0.0 area 0
r4(config-router)#network 34.1.1.4 0.0.0.0 area 0
r4(config-router)#network 4.4.4.4 0.0.0.0 area 0
```

说明：发布各路由器的直连网段与 Loopback 0 到 OSPF 中。

2.检查 IGP 连接

（1）检查 R1 上的 OSPF 邻居

```
r1#show ip ospf neighbor

  Neighbor ID     Pri   State           Dead
Time   Address         Interface

  3.3.3.3           1   FULL/BDR        00:00:34    13.1.1.3        Fa
stEthernet0/1

  2.2.2.2           1   FULL/BDR        00:00:38    12.1.1.2        Fa
stEthernet0/0

  r1#
```

说明：R1 与 R2 和 R3 的 OSPF 邻居正常。

（2）检查 R4 上的 OSPF 邻居

```
r4#show ip ospf neighbor


  Neighbor ID     Pri   State           Dead
Time   Address         Interface

  3.3.3.3           1   FULL/DR         00:00:34    34.1.1.3        Fa
stEthernet0/0

  2.2.2.2           1   FULL/DR         00:00:29    24.1.1.2        Fa
stEthernet0/1

  r4#
```

说明：R4 与 R2 和 R3 的 OSPF 邻居正常。

（3）在 R1 上查看全网的 loopback 0 通信情况

```
r1#ping 2.2.2.2 source loopback 0


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#

r1#ping 3.3.3.3 source loopback 0
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#ping 4.4.4.4 source loopback 0


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#
```

说明：全网的 loopback 0 通信正常，可以用此地址建立 BGP 连接。

3.建立 BGP 邻居

（1）在 R1 与 R2 之间建立 BGP 邻居

```
r1(config)#router bgp 1

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

r1(config-router)#neighbor 2.2.2.2 remote-as 1
```

说明：配置 R1 的 Router-ID，并指定邻居为 2.2.2.2，邻居 AS 为 1。

（2）在 R1 与 R2 之间建立 BGP 邻居

```
r2(config)#router bgp 1

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

r2(config-router)#neighbor 1.1.1.1 remote-as 1
```

说明：配置 R2 的 Router-ID，并指定邻居为 1.1.1.1，邻居 AS 为 1。

(3)查看 BGP 邻居

```
r1#show ip bgp summary

BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 1, main routing table version 1


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

2.2.2.2         4     1       0       0        0    0    0
never    Active

r1#
```

说明：R1 无法与 R2 建立 BGP 邻居，因为自己目的地址是 2.2.2.2，而对方源地址是 12.1.1.2，同样对方目的是 1.1.1.1，而自己源是 12.1.1.1，源与目的不匹配，所以必须修改。

（4）修改 R1 的 BGP 源地址

```
r1(config)#router bgp 1

r1(config-router)#neighbor 2.2.2.2 update-source loopback 0
```

说明：将 R1 的源地址改为 loopback 0，即 1.1.1.1，而 R2 的目标地址也是 1.1.1.1，与 R1 的源相同。

（5）查看 R1 的 BGP 邻居

```
r1#show ip bgp summary

BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 1, main routing table version 1


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

2.2.2.2         4     1       9       9        1    0    0
00:05:43        0

r1#
```

说明：因为 R1 的源与 R2 的目的相匹配，所以双方正常建立 BGP 邻居。

（6）修改 R2 的 BGP 源地址

```
r2(config)#router bgp 1

r2(config-router)#neighbor 1.1.1.1 update-source loopback 0
```

说明：虽然 R1 的源与 R2 的目的相匹配，已正常建立 BGP 邻居，但为了统一，也配置使 R2 的源与 R1 的目的相匹配。

4.建立 R2 与 R4 的 BGP 邻居

（1）配置 R2 的 BGP 参数

```
r2(config)#router bgp 1

r2(config-router)#neighbor 4.4.4.4 remote-as 4

r2(config-router)#neighbor 4.4.4.4 update-source loopback 0
```

说明：在 R2 上指定邻居为 4.4.4.4，邻居 AS 为 4，并且 R2 的源为 loopback 0，即 2.2.2.2。

（2）配置 R4 的 BGP 参数

```
r4(config)#router bgp 4

r4(config-router)#bgp router-id 4.4.4.4

r4(config-router)#neighbor 2.2.2.2 remote-as 1

r4(config-router)#neighbor 2.2.2.2 update-source loopback 0
```

说明：配置 R4 的 Router-ID，并指定邻居为 2.2.2.2，邻居 AS 为 1，R4 的源地址为 4.4.4.4。

（3）查看 BGP 邻居

```
r2#sh ip bgp summary

BGP router identifier 2.2.2.2, local AS number 1

BGP table version is 1, main routing table version 1


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

1.1.1.1         4     1      12      12        1    0    0
00:08:45        0

4.4.4.4         4     4       0       0        0    0    0
never    Idle
```

r2#

说明：因为 R2 与 R4 之间为 eBGP 邻居，hello 的 TTL 值默认为 1，而 R2 的源 2.2.2.2 到达 R4 的源 4.4.4.4，不止经过了一个网段，所以 TTL 值必须修改，才能够建立连接。

（4）修改 R2 与 R4 的 TTL 值

r2(config)#router bgp 1

r2(config-router)#neighbor 4.4.4.4 ebgp-multihop

r4(config)#router bgp 4

r4(config-router)#neighbor 2.2.2.2 ebgp-multihop

说明：将 R2 与 R4 之间的 TTL 值改大，默认改为 255。

（5）查看 BGP 邻居

r2#sh ip bgp summary

BGP router identifier 2.2.2.2, local AS number 1

BGP table version is 1, main routing table version 1

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

1.1.1.1 4 1 14 14 1 0 0 00:10:27 0

4.4.4.4 4 4 4 4 1 0 0 00:00:16 0

r2#

说明：由于邻居参数配置，所以邻居已经正常建立。

5.发布 BGP 路由

（1）查看 BGP 路由表

R1：

r1#sh ip bgp

r1#

R2：

r2#sh ip bgp

r2#

R4：

r4#sh ip bgp

r4#

说明：默认情况下，BGP 的路由表为空。

（2）在 R1 上导入 BGP 路由表

r1(config)#router bgp 1

r1(config-router)#network 11.1.1.0 mask 255.255.255.0

说明：将路由 11.1.1.0/24 导入 BGP 路由表，命令 network 后面的网段必须在 IGP 表中真实存在，也就是 show ip route 必须能够看见，且掩码匹配，否则无法导入 BGP 路由表。

（3）查看 R1 的 BGP 路由表

1#sh ip bgp

BGP table version is 2, local router ID is 1.1.1.1

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

r RIB-failure, S Stale

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

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 11.1.1.0/24	0.0.0.0	0		32768	i

r1#

说明：因为 R1 已经通过命令 network 将 11.1.1.0/24 导入 BGP 路由表中，所以能够看见该路由，由于此路由是自己导入的而并非从邻居学习到的，所以为本地路由，默认 Next Hop 为 0.0.0.0，并且 weight 值为 32768，最后的 i 表示该路由的 origin 属性为 IGP，就表示是从 IGP 路由表被导入 BGP 路由表的。

（4）在 R2 上导入 BGP 路由表

r2(config)#route-map loop permit 10

r2(config-route-map)#match interface loopback 22

```
r2(config)#router bgp 1

r2(config-router)#redistribute connected route-map loop
```

说明：R2 通过重分布的方法将 22.2.2.0/24 导入 BGP 路由表。

（5）查看 R2 的 BGP 路由表

```
r2#sh ip bgp

BGP table version is 2, local router ID is 2.2.2.2

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

* i11.1.1.0/24     1.1.1.1                  0    100      0 i

*> 22.2.2.0/24     0.0.0.0                  0         32768 ?

r2#
```

说明：因为 R2 是通过重分布的方法将路由导入 BGP 路由表的，所以为本地路由，默认 Next Hop 为 0.0.0.0，并且 weight 值为 32768，并且因为是使用重分布的方法，所以该路由的 origin 属性为 incomplete；而路由 11.1.1.0/24 是从邻居 R1 学习到的，所以 下一跳为邻居 1.1.1.1 的地址，前面的 i 表示是从 iBGP 邻居学习到的，而后面的 i 同样表示该路由的 origin 属性为 IGP。

（6）在 R4 上导入 BGP 路由表

```
r4(config)#router bgp 4

r4(config-router)#network 44.4.4.0 mask 255.255.255.0
```

说明：通过命令 network 将路由 44.4.4.0/24 发布到 BGP 路由表中。

（7）查看 R4 的 BGP 路由表

```
r4#sh ip bgp

BGP table version is 3, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 22.2.2.0/24      2.2.2.2                  0             0 1 ?

*> 44.4.4.0/24      0.0.0.0                  0         32768 i

r4#
```

说明：因为 R4 是通过命令 network 将路由导入 BGP 路由表的，所以为本地路由，默认 Next Hop 为 0.0.0.0，并且 weight 值为 32768， origin 属性为 IGP；而路由 22.2.2.0/24 是从 eBGP 邻居 R2 学习到的，所以 下一跳为邻居 2.2.2.2 的地址，后面的?表示该路由的 origin 属性为 incomplete。

6.保证 BGP 全网可达

（1）查看 R1 的 BGP 路由表

```
r1#sh ip bgp
```

```
BGP table version is 2, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

* i22.2.2.0/24      2.2.2.2                  0    100       0 ?

* i44.4.4.0/24      4.4.4.4                  0    100       0 4 i

r1#
```

说明：R1 已经学习到三条路由，11.1.1.0/24 为自己本地路由，所以路由前面出现>符号，表示该路由为最优路径，被 BGP 选中并使用，而 22.2.2.0/24 和 44.4.4.0/24 没有>符号，所以并不是最优路由，就不会被 BGP 使用，被标为最优路径的两个条件中，下一跳可达已经符合，而由于这两条路由前面有 i 的标记，所以是从 iBGP 学习到的，因此必须完成 iBGP 与 IGP 之间的同步后，才能为最优路由。

（2）在 R1 上关闭 iBGP 与 IGP 之间的同步

```
r1(config)#router bgp 1

r1(config-router)#no synchronization
```

（3）再次查看 R1 的 BGP 路由表

```
r1#sh ip bgp

BGP table version is 4, local router ID is 1.1.1.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
*>i44.4.4.0/24      4.4.4.4                  0    100      0 4 i
r1#
```

说明：由于所有路由的下一跳都可达，并且 iBGP 与 IGP 之间的同步已不需要满足，所以所有路由都成为了最优路由。

（4）查看 R4 的 BGP 路由表

```
r4#sh ip bgp
BGP table version is 3, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 22.2.2.0/24      2.2.2.2                  0             0 1 ?
*> 44.4.4.0/24      0.0.0.0                  0         32768 i
```

```
r4#
```

说明：R4 所有的路由都不需要 iBGP 与 IGP 之间的同步，而下一跳都可达，所以全部被标为最优路由，但是缺少 R1 发布的路由 11.1.1.0/24。

（5）查看 R2 的 BGP 路由表

```
r2#sh ip bgp
BGP table version is 3, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i11.1.1.0/24     1.1.1.1                  0    100      0 i
*> 22.2.2.0/24     0.0.0.0                  0         32768 ?
*> 44.4.4.0/24     4.4.4.4                  0             0 4 i
r2#
```

说明：因为 11.1.1.0/24 是从 iBGP 学习到的，并且不满足 iBGP 与 IGP 之间的同步，所以该路由没有被标为最优路由，所以就没有发给邻居 R4。

（6）在 R2 上关闭 iBGP 与 IGP 之间的同步

```
r2(config)#router bgp 1
r2(config-router)#no synchronization
```

（7）再次查看 R2 的 BGP 路由表

```
r2#sh ip bgp

BGP table version is 4, local router ID is 2.2.2.2

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i11.1.1.0/24      1.1.1.1                  0    100      0 i

*> 22.2.2.0/24      0.0.0.0                  0         32768 ?

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

r2#
```

说明：由于所有路由的下一跳都可达，并且 iBGP 与 IGP 之间的同步已不需要满足，所以所有路由都成为了最优路由。

（8）查看 R4 的 BGP 路由表

```
r4#sh ip bgp

BGP table version is 4, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path
```

```
*> 11.1.1.0/24       2.2.2.2                                0 1 i

*> 22.2.2.0/24       2.2.2.2                  0             0 1 ?

*> 44.4.4.0/24       0.0.0.0                  0         32768 i

r4#
```

说明：R1，R2，R4 都已经拥有了全部的路由。

RIB failure

1.R4 创建网段并发布进 OSPF

（1）R4 创建网段并发布进 OSPF

```
r4(config)#int loopback 100

r4(config-if)#ip address 100.1.1.1 255.255.255.0

r4(config-if)#ip ospf network point-to-point


r4(config)#router ospf 1

r4(config-router)#network 100.1.1.1 0.0.0.0 area 0
```

说明：在 R4 上创建网段 100.1.1.0/24，并将其发布进 OSPF

（2）R4 将 100.1.1.0/24 导入 BGP 路由表

```
r4(config)#router bgp 4

r4(config-router)#network 100.1.1.0 mask 255.255.255.0
```

（3）查看 R4 的 BGP 路由表

```
r4#sh ip bgp

BGP table version is 7, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      2.2.2.2                              0 1 i

*> 22.2.2.0/24      2.2.2.2                  0           0 1 ?

*> 44.4.4.0/24      0.0.0.0                  0       32768 i

*> 100.1.1.0/24     0.0.0.0                  0       32768 i

r4#
```

说明：100.1.1.0/24 已经被发布进 BGP，成为本地路由，所以下一跳为 0.0.0.0，weight 值为 32768。

2.查看 RIB failure

（1）查看 R2 的 BGP 路由表

```
r2#sh ip bgp

BGP table version is 7, local router ID is 2.2.2.2

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

```
                r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i11.1.1.0/24      1.1.1.1                  0    100      0 i

*> 22.2.2.0/24      0.0.0.0                  0         32768 ?

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

*> 100.1.1.0/24     4.4.4.4                  0             0 4 i

r2#
```

说明：因为 R2 从 eBGP 邻居 R4 学习到的 100.1.1.0/24 的 AD 值为 20，并且还从 OSPF 学习到一次，AD 为 110，而 OSPF 的 AD 比 BGP 学习到的 AD 高，所以 BGP 中的路由正常。

（2）查看 R1 的 BGP 路由表

```
r1#sh ip bgp

BGP table version is 6, local router ID is 1.1.1.1

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

                r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
```

```
*>i44.4.4.0/24      4.4.4.4                  0    100      0 4 i

r>i100.1.1.0/24     4.4.4.4                  0    100      0 4 i

r1#
```

说明：因为 R2 从 iBGP 邻居 R2 学习到的 100.1.1.0/24 的 AD 值为 200，并且还从 OSPF 学习到一次，AD 为 110，而 OSPF 的 AD 比 BGP 学习到的 AD 低，所以 BGP 中的路由为 RIB failure，所以路由前面标记为 r。

(3)查看 R1 的 RIB failure 表

```
r1#sh ip bgp rib-failure

Network          Next Hop                RIB-failure   RIB-NH Matches

100.1.1.0/24     4.4.4.4                 Higher admin distance   n/a

r1#
```

说明：可以看到 100.1.1.0/24 为 RIB failure 路由，原因为已经拥有比 BGP 更优先的 AD 值。

BGP 选路规则实验

1.使 R1，R2，R3，R4 全网建立 BGP，并且互通

（1）将 R3 加入 BGP 中

R1：

```
r1(config)#router bgp 1
```

```
r1(config-router)#neighbor 3.3.3.3 remote-as 1

r1(config-router)#neighbor 3.3.3.3 update-source loopback 0
```

R3：

```
r3(config)#router bgp 1

r3(config-router)#bgp router-id 3.3.3.3

r3(config-router)#neighbor 1.1.1.1 remote-as 1

r3(config-router)#neighbor 1.1.1.1 update-source loopback 0


r3(config-router)#neighbor 4.4.4.4 remote-as 4

r3(config-router)#neighbor 4.4.4.4 update-source loopback 0

r3(config-router)#neighbor 4.4.4.4 ebgp-multihop
```

R4：

```
r4(config)#router bgp 4

r4(config-router)#neighbor 3.3.3.3 remote-as 1

r4(config-router)#neighbor 3.3.3.3 update-source loopback 0

r4(config-router)#neighbor 3.3.3.3 ebgp-multihop
```

（2）查看 R3 的 BGP 邻居

```
r3#sh ip bg summary
```

```
BGP router identifier 3.3.3.3, local AS number 1

BGP table version is 4, main routing table version 4

3 network entries using 351 bytes of memory

3 path entries using 156 bytes of memory

3/2 BGP path/bestpath attribute entries using 372 bytes of memory

1 BGP AS-PATH entries using 24 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 903 total bytes of memory

BGP activity 3/0 prefixes, 3/0 paths, scan interval 60 secs


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

1.1.1.1         4     1       7       7        4    0    0
00:02:08        1

4.4.4.4         4     4       7       5        4    0    0
00:00:08        2

r3#
```

说明：R3 已经与其它路由器建立 BGP 邻居。

2.改变 AS 1 内部下一跳

（1）查看 R1 的 BGP 路由表

```
r1#sh ip bgp
```

```
BGP table version is 6, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?

* i44.4.4.0/24      4.4.4.4                  0    100      0 4 i

*>i                 4.4.4.4                  0    100      0 4 i

r i100.1.1.0/24     4.4.4.4                  0    100      0 4 i

r>i                 4.4.4.4                  0    100      0 4 i

r1#
```

说明：因为 R1 都是从 iBGP 收到的路由，所以到达 R4 的路由 44.4.4.0 的下一跳都为 4.4.4.4，而没有被 R2 和 R3 改变。

（2）改变 R2 与 R3 对 R1 的下一跳为自己

```
r2(config)#router bgp 1

r2(config-router)#neighbor 1.1.1.1 next-hop-self


r3(config)#router bgp 1

r3(config-router)#neighbor 1.1.1.1 next-hop-self
```

（3）再次查看 R1 的 BGP 路由表

```
r1#sh ip bgp
BGP table version is 8, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
* i44.4.4.0/24      3.3.3.3                  0    100      0 4 i
*>i                 2.2.2.2                  0    100      0 4 i
r i100.1.1.0/24     3.3.3.3                  0    100      0 4 i
r>i                 2.2.2.2                  0    100      0 4 i
r1#
```

说明：学习到的路由 44.4.4.0/24 已经被 R2 和 R3 改为自己。

测试选路规则说明：

测试 R1 通过 R2 与 R3 到达 R4 的网段 44.4.4.0/24 的选路，

以及测试 R4 通过 R2 与 R3 到达 R1 的网段 11.1.1.0/24 的选路，

要测试的选路顺序为

1. 最高 Weight 值

2. 最高 LOCAL_PREF 值

3. 本地发起路由

4. 最短 AS_PATH

5. 最低 Origin 类型

6. 最小 MED 值

7. eBGP 优于 iBGP

8. 最小 IGP metric 到达下一跳的路由

9. 负载均衡（如果开启的话）

10. 如果下一跳都为 eBGP，则选择最早学习到的路由（即时间最长的路由）

11. 最低 Router-ID 下一跳

12. 最短 cluster list（如同 AS_PATH）

13. 最小下一跳的邻居地址

因为选路顺序为由上至下，当上一个属性已经比较出最优路径，则下一属性被忽略，所以我们实验从下往上修改来进行比较，因为改过下面的属性影响选路之后，只要再改上一条，就能再次影响选路，就能证明，上一条是比下一条优先的。

测试第 13 条 最小下一跳的邻居地址

说明：因为只有下一跳邻居的 Router-ID 相同的情况下，才会比较下一跳邻居的地址大小，所以先将 R2 与 R3 的 Router-ID 改为相同，以测试比较下一跳地址。

（1）修改 R3 的 Router-ID 与 R2 相同

```
r3(config)#router bgp 1

r3(config-router)#bgp router-id 2.2.2.2

r3(config-router)#
```

（2）查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp

BGP table version is 8, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?

* i44.4.4.0/24      3.3.3.3                  0    100      0 4 i

*>i                 2.2.2.2                  0    100      0 4 i

r i100.1.1.0/24     3.3.3.3                  0    100      0 4 i

r>i                 2.2.2.2                  0    100      0 4 i

r1#
```

说明：R1 到达网段 44.4.4.0/24 选则最小下一跳邻居 R2 为最优路径。

测试第 12 条 最短 cluster list

说明：因为比较最短 cluster list 只在 BGP Route Reflector (RR)环境中才有，所以此步跳过。

测试第 11 条 最低 Router-ID 下一跳

说明：选择下一跳有最小 Router-ID 的邻居为最优路径。

（1）修改 R3 的 Router-ID

```
r3(config)#router bgp 1

r3(config-router)#bgp router-id 1.1.1.3
```

说明：将 R3 的 Router-ID 改为 1.1.1.3

（2）查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp

BGP table version is 10, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i
```

```
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
*>i44.4.4.0/24      3.3.3.3                  0    100      0 4 i
* i                 2.2.2.2                  0    100      0 4 i
r>i100.1.1.0/24     3.3.3.3                  0    100      0 4 i
r i                 2.2.2.2                  0    100      0 4 i
r1#
```

说明：因为 R2 的 Router-ID 为 2.2.2.2，而 R3 的 Router-ID 为 1.1.1.3，所以最小 Router-ID 的 R3 被选为最优路径。

测试第 10 条 如果下一跳都为 eBGP，则选择最早学习到的路由（即时间最长的路由）

说明：因为只有下一跳都为 eBGP，才比较选择最早学习到的路由，所以测试 R4 通过 R2 与 R2 到达 11.1.1.0/24 的选路。

（1）查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip bgp
BGP table version is 7, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  11.1.1.0/24      3.3.3.3                              0 1 i
*>                  2.2.2.2                              0 1 i
```

```
*> 22.2.2.0/24       2.2.2.2                  0             0 1 ?

*> 44.4.4.0/24       0.0.0.0                  0         32768 i

*> 100.1.1.0/24      0.0.0.0                  0         32768 i

r4#
```

说明：R4 选择 R2 到达 11.1.1.0/24

(2)查看 R2 与 R3 的邻居时间

```
r4#sh ip bg summary

BGP router identifier 4.4.4.4, local AS number 4

BGP table version is 7, main routing table version 7

4 network entries using 404 bytes of memory

5 path entries using 240 bytes of memory

3 BGP path attribute entries using 180 bytes of memory

1 BGP AS-PATH entries using 24 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 848 total bytes of memory

BGP activity 5/1 prefixes, 8/3 paths, scan interval 60 secs


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

2.2.2.2         4     1      51      49        7    0    0
00:43:27        2
```

```
3.3.3.3          4      1       27       33          7     0     0
00:02:23         1
```

```
r4#
```

说明：因为 R2 的邻居时间比 R3 长，所以 R2 为最优路径。

（3）清除 R2 的邻居，以刷新邻居时间

```
r4(config)#router bgp 4
r4(config-router)#neighbor 2.2.2.2 shutdown
r4(config-router)#
*Mar  1 01:16:09.823: %BGP-5-ADJCHANGE: neighbor 2.2.2.2 Down Admin.
shutdown
r4(config-router)#no neighbor 2.2.2.2 shutdown
r4(config-router)#
*Mar  1 01:16:37.452: %BGP-5-ADJCHANGE: neighbor 2.2.2.2 Up
r4(config-router)#
```

说明：将邻居 R2 断开，再建立，从而刷新邻居的建立时间。

（4）再次查看邻居的建立时间

```
r4#sh ip bg summary
BGP router identifier 4.4.4.4, local AS number 4
BGP table version is 10, main routing table version 10
4 network entries using 404 bytes of memory
5 path entries using 240 bytes of memory
```

```
3 BGP path attribute entries using 180 bytes of memory

1 BGP AS-PATH entries using 24 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 848 total bytes of memory

BGP activity 5/1 prefixes, 10/5 paths, scan interval 60 secs


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

2.2.2.2         4     1      57      55       10    0    0
00:00:27        2

3.3.3.3         4     1      28      36       10    0    0
00:03:45        1

r4#
```

说明：R3 的邻居时间比 R2 长。

（5）再次查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip bgp

BGP table version is 10, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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

```
   Network          Next Hop            Metric LocPrf Weight Path
*  11.1.1.0/24      2.2.2.2                                0 1 i
*>                  3.3.3.3                                0 1 i
*> 22.2.2.0/24      2.2.2.2                  0             0 1 ?
*> 44.4.4.0/24      0.0.0.0                  0         32768 i
*> 100.1.1.0/24     0.0.0.0                  0         32768 i
r4#
```

说明：因为 R3 的邻居时间比 R2 长，所以选择了 R3 为最优路径。

测试第 9 条 BGP 负载均衡

说明：只有在前面 8 条属性都相同的话，才能开启 BGP 的负载功能，前 8 条属性任何一条不同，都不能负载。

（1）查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip route bgp
     22.0.0.0/24 is subnetted, 1 subnets
B       22.2.2.0 [200/0] via 2.2.2.2, 00:38:33
     44.0.0.0/24 is subnetted, 1 subnets
B       44.4.4.0 [200/0] via 3.3.3.3, 00:05:12
r1#
```

说明：R1 到达 44.4.0/24 只走 R3，默认没有负载。

（2）开启 BGP 负载功能

```
r1(config)#router bgp 1

r1(config-router)#maximum-paths ibgp 2
```

说明：因为两个下一跳都为 iBGP，所以开启 iBGP 的负载功能。

（3）再次查看 R1 到达 44.4.0/24 的选路

```
r1#sh ip route bgp

     22.0.0.0/24 is subnetted, 1 subnets

B       22.2.2.0 [200/0] via 2.2.2.2, 00:39:16

     44.0.0.0/24 is subnetted, 1 subnets

B       44.4.4.0 [200/0] via 2.2.2.2, 00:00:18

                 [200/0] via 3.3.3.3, 00:00:18

r1#
```

说明：R1 到达 44.4.4.0/24 已经执行负载。

（4）查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip route bgp

     22.0.0.0/24 is subnetted, 1 subnets

B       22.2.2.0 [20/0] via 2.2.2.2, 00:03:03

     11.0.0.0/24 is subnetted, 1 subnets

B       11.1.1.0 [20/0] via 3.3.3.3, 00:03:31

r4#
```

说明：R4 到达 11.1.1.0/24 没有负载。

（5）开启 R4 到达 11.1.1.0/24 的负载

```
r4(config)#router bgp 4

r4(config-router)#maximum-paths 2
```

说明：因为两个下一跳都为 eBGP，所以开启 eBGP 的负载。

（6）再次查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip route bgp

     22.0.0.0/24 is subnetted, 1 subnets

B       22.2.2.0 [20/0] via 2.2.2.2, 00:03:33

     11.0.0.0/24 is subnetted, 1 subnets

B       11.1.1.0 [20/0] via 2.2.2.2, 00:00:15

                 [20/0] via 3.3.3.3, 00:00:15

r4#
```

说明：R4 到达 11.1.1.0/24 已经执行负载。

测试第 8 条 最小 IGP metric 到达下一跳的路由

说明：拥有最小 IGP metric 到达下一跳的路由为最优路径。

（1）查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp

BGP table version is 14, local router ID is 1.1.1.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
* i44.4.4.0/24      2.2.2.2                  0    100      0 4 i
*>i                 3.3.3.3                  0    100      0 4 i
r i100.1.1.0/24     2.2.2.2                  0    100      0 4 i
r>i                 3.3.3.3                  0    100      0 4 i
r1#
```

说明：R1 选择 R3 到达 44.4.0/24，但并不是因为 IGP metric。

（2）查看到达两个下一跳 R2 与 R3 的 IGP metric 值

```
r1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
```

```
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
O       34.1.1.0 [110/2] via 13.1.1.3, 00:28:17, FastEthernet0/1
     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 12.1.1.2, 00:28:17, FastEthernet0/0
     100.0.0.0/24 is subnetted, 1 subnets
O       100.1.1.0 [110/3] via 12.1.1.2, 00:28:17, FastEthernet0/0
                  [110/3] via 13.1.1.3, 00:28:17, FastEthernet0/1
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/2] via 13.1.1.3, 00:28:18, FastEthernet0/1
     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/3] via 12.1.1.2, 00:28:18, FastEthernet0/0
                [110/3] via 13.1.1.3, 00:28:18, FastEthernet0/1
     22.0.0.0/24 is subnetted, 1 subnets
B       22.2.2.0 [200/0] via 2.2.2.2, 00:41:12
     24.0.0.0/24 is subnetted, 1 subnets
```

```
O        24.1.1.0 [110/2] via 12.1.1.2, 00:28:21, FastEthernet0/0
     11.0.0.0/24 is subnetted, 1 subnets
C        11.1.1.0 is directly connected, Loopback11
     12.0.0.0/24 is subnetted, 1 subnets
C        12.1.1.0 is directly connected, FastEthernet0/0
     13.0.0.0/24 is subnetted, 1 subnets
C        13.1.1.0 is directly connected, FastEthernet0/1
     44.0.0.0/24 is subnetted, 1 subnets
B        44.4.4.0 [200/0] via 2.2.2.2, 00:02:14
                  [200/0] via 3.3.3.3, 00:02:14
r1#
```

说明：到达两个下一跳 R2 与 R3 的 IGP metric 值相同。

（3）改大到达下一跳 R3 的 IGP metric 值，使最优路径走 R2

```
r1(config)#int f0/1
r1(config-if)#ip ospf cost 2
```

（4）再次查看到达两个下一跳 R2 与 R3 的 IGP metric 值

```
r1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
          E1 - OSPF external type 1, E2 - OSPF external type 2
          i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
area
          * - candidate default, U - per-user static route, o - ODR
          P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
O       34.1.1.0 [110/3] via 13.1.1.3, 00:01:10, FastEthernet0/1
                 [110/3] via 12.1.1.2, 00:01:10, FastEthernet0/0
     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 12.1.1.2, 00:01:10, FastEthernet0/0
     100.0.0.0/24 is subnetted, 1 subnets
O       100.1.1.0 [110/3] via 12.1.1.2, 00:01:10, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/3] via 13.1.1.3, 00:01:11, FastEthernet0/1
     4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/3] via 12.1.1.2, 00:01:11, FastEthernet0/0
     22.0.0.0/24 is subnetted, 1 subnets
B       22.2.2.0 [200/0] via 2.2.2.2, 00:43:18
```

```
     24.0.0.0/24 is subnetted, 1 subnets
O       24.1.1.0 [110/2] via 12.1.1.2, 00:01:11, FastEthernet0/0
     11.0.0.0/24 is subnetted, 1 subnets
C       11.1.1.0 is directly connected, Loopback11
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, FastEthernet0/0
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, FastEthernet0/1
     44.0.0.0/24 is subnetted, 1 subnets
B       44.4.4.0 [200/0] via 2.2.2.2, 00:00:14
r1#
```

说明：到达 R2 的 metric 值为 2,到达 R3 的 metric 值为 3，大于 R2。

（5）再次查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp
BGP table version is 16, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
```

```
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
*>i44.4.4.0/24      2.2.2.2                  0    100      0 4 i
* i                 3.3.3.3                  0    100      0 4 i
r>i100.1.1.0/24     2.2.2.2                  0    100      0 4 i
r i                 3.3.3.3                  0    100      0 4 i
r1#
```

说明：因为到达 R2 的 IGP metric 值比 R3 小，所以最优路径为 R2。

测试第 7 条 eBGP 优于 iBGP

说明：因为没有两个下一跳同时存在 eBGP 和 iBGP 的，所以此步跳过。

测试第 6 条 最小 MED 值

说明：最小 MED 值的路由为最优路径，一条没有 MED 的路由，默认为 0。

因为 MED 希望是在下一跳都为 eBGP，也就是出 AS 时比较，但现在要证明，只要有 MED 值存在，无论邻居是何类型，都将比较 MED 值，所以选择比较 R1 到达 44.4.4.0/24 的选路。

（1）查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp
BGP table version is 16, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
```

```
                r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?

*>i44.4.4.0/24      2.2.2.2                  0    100      0 4 i

* i                 3.3.3.3                  0    100      0 4 i

r>i100.1.1.0/24     2.2.2.2                  0    100      0 4 i

r i                 3.3.3.3                  0    100      0 4 i

r1#
```

说明：R2 和 R3 的 MED 默认为 0，相同。

（2）加大 R2 的 MED 值，使其走 R3

```
r1(config)#access-list 44 permit 44.4.4.0


r1(config)#route-map med permit 10

r1(config-route-map)#match ip address 44

r1(config-route-map)#set metric 44

r1(config-route-map)#exit

r1(config)#route-map med permit 20
```

```
r1(config)#router bgp 1
r1(config-router)#neighbor 2.2.2.2 route-map med in
```

自动刷新：

```
r1(config)#router bgp 1
r1(config-router)#neighbor 2.2.2.2 soft-reconfiguration inbound
```

说明：修改属性后，BGP 无法得知，所以配置自动刷新策略，只能为 in 方向。

（3）再次查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp
BGP table version is 17, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
* i44.4.4.0/24      2.2.2.2                 44    100      0 4 i
*>i                 3.3.3.3                  0    100      0 4 i
r>i100.1.1.0/24     2.2.2.2                  0    100      0 4 i
```

r i 3.3.3.3 0 100 0 4 i

r1#

说明：因为 R3 的 MED 小于 R2，所以 R3 为最优路径。

注：在 R4 上修改 MED 影响选路的方法不再举例。

测试第 5 条 最低 Origin 类型

说明：优先级为 IGP 优于 EGP，EGP 优于 Incomplete，即 IGP>EGP>Incomplete。

（1）查看 R1 到达 44.4.4.0/24 的选路

r1#sh ip bgp

BGP table version is 17, local router ID is 1.1.1.1

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

r RIB-failure, S Stale

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

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 11.1.1.0/24	0.0.0.0	0		32768	i
*>i22.2.2.0/24	2.2.2.2	0	100	0	?
* i44.4.4.0/24	2.2.2.2	44	100	0	4 i
*>i	3.3.3.3	0	100	0	4 i

```
r>i100.1.1.0/24     2.2.2.2                  0     100       0 4 i

r i                 3.3.3.3                  0     100       0 4 i

r1#
```

说明：两个下一跳邻居的 origin 属性都为 IGP。

（2）将 R3 的 origin 属性改为 incomplete，使其走 R2

```
r1(config)#route-map ori permit 10

r1(config-route-map)#match ip address 44

r1(config-route-map)#set origin incomplete

r1(config)#route-map ori permit 20


r1(config)#router bgp 1

r1(config-router)#neighbor 3.3.3.3 route-map ori in

r1(config-router)#neighbor 3.3.3.3 soft-reconfiguration inbound
```

（3）再次查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp

BGP table version is 18, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
*>i44.4.4.0/24      2.2.2.2                 44    100      0 4 i
* i                 3.3.3.3                  0    100      0 4 ?
r>i100.1.1.0/24     2.2.2.2                  0    100      0 4 i
r i                 3.3.3.3                  0    100      0 4 i
r1#
```

说明：因为 R3 的 origin 属性为为 incomplete，R2 的 origin 属性为 IGP，所以选 R2 为最优路径。

测试第 4 条最短 AS_PATH

说明：修改 AS_Path 只能在 eBGP 邻居之间，iBGP 邻居是不能改动 AS_Path，所以测试 R4 通过 eBGP 邻居 R2 和 R3 到达 11.1.1.0 的选路。

（1）查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip bgp
BGP table version is 14, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop            Metric LocPrf Weight Path
*  11.1.1.0/24      2.2.2.2                                0 1 i
*>                  3.3.3.3                                0 1 i
*> 22.2.2.0/24      2.2.2.2                  0             0 1 ?
*> 44.4.4.0/24      0.0.0.0                  0         32768 i
*> 100.1.1.0/24     0.0.0.0                  0         32768 i
r4#
```

说明：R2 与 R3 的 AS_Path 长短相同。

（2）加长 R3 路径上的 AS_Path，使其走 R2

```
r4(config)#access-list 3 permit 11.1.1.0

r4(config)#route-map as permit 10
r4(config-route-map)#match ip address 3
r4(config-route-map)#set as-path prepend 3
r4(config-route-map)#exit
r4(config)#route-map as permit 20

r4(config)#router bgp 4
r4(config-router)#neighbor 3.3.3.3 route-map as in
```

```
r4(config-router)#neighbor 3.3.3.3 soft-reconfiguration inbound
```

（3）再次查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip bgp
BGP table version is 15, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      2.2.2.2                              0 1 i
*                   3.3.3.3                              0 3 1 i
*> 22.2.2.0/24      2.2.2.2                  0           0 1 ?
*> 44.4.4.0/24      0.0.0.0                  0       32768 i
*> 100.1.1.0/24     0.0.0.0                  0       32768 i
r4#
```

说明：因为 R3 的 AS_Path 比 R2 长，所以最优路径选择 R2。

测试第 3 条 本地发起路由

说明：因为一条路由是不是由本地引入 BGP，无法修改，所以此步跳过。

测试第 2 条 最高 LOCAL_PREF 值

因为 LOCAL_PREF 希望是在下一跳都为 iBGP，也就是 AS 内部比较，但现在要证明，只要有 LOCAL_PREF 值存在，无论邻居是何类型，都将比较 LOCAL_PREF 值，所以选择比较 R4 到达 11.1.1.0/24 的选路。

(1) 查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip bgp

BGP table version is 15, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      2.2.2.2                              0 1 i

*                   3.3.3.3                              0 3 1 i

*> 22.2.2.0/24      2.2.2.2                  0           0 1 ?

*> 44.4.4.0/24      0.0.0.0                  0       32768 i

*> 100.1.1.0/24     0.0.0.0                  0       32768 i

r4#
```

说明：R2 为最优路径。

（2）改 R3 的 LOCAL_PREF 值比 R2 大，让其走 R3

```
r4(config)#access-list 11 permit 11.1.1.0


r4(config)#route-map r3 permit 10

r4(config-route-map)#match ip address 11

r4(config-route-map)#set local-preference 3

r4(config-route-map)#exit

r4(config)#route-map r3 permit 20


r4(config)#route-map r2 permit 10

r4(config-route-map)#match ip address 11

r4(config-route-map)#set local-preference 2

r4(config-route-map)#exit

r4(config)#route-map r2 permit 20


r4(config)#router bgp 4

r4(config-router)#neighbor 3.3.3.3 route-map r3 in

r4(config-router)#neighbor 2.2.2.2 route-map r2 in

r4(config-router)#neighbor 2.2.2.2 soft-reconfiguration inbound
```

（3）再次查看 R4 到达 11.1.1.0/24 的选路

```
r4#sh ip bgp
BGP table version is 5, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  11.1.1.0/24      2.2.2.2                  2      0 1 i
*>                  3.3.3.3                  3      0 1 i
*> 22.2.2.0/24      2.2.2.2                  0             0 1 ?
*> 44.4.4.0/24      0.0.0.0                  0         32768 i
*> 100.1.1.0/24     0.0.0.0                  0         32768 i
r4#
```

说明：因为 R3 的 LOCAL_PREF 值比 R2 大，所以最优路径为 R3。

注：在 R1 上修改 LOCAL_PREF 影响选路的方法不再举例。

测试第 1 条　最高 Weight 值

说明：可针对路由修改 Weight 值，也可针对整个邻居修改 Weight 值。

（1）查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp
BGP table version is 23, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
* i44.4.4.0/24      3.3.3.3                  0    100      0 4 ?
*>i                 2.2.2.2                 44    100      0 4 i
r i100.1.1.0/24     3.3.3.3                  0    100      0 4 i
r>i                 2.2.2.2                  0    100      0 4 i
r1#
```

说明：R2 为最优路径。

（2）改大 R3 的 weight 值，使其走 R3

```
r1(config)#router bgp 1
r1(config-router)#neighbor 3.3.3.3 weight 3
```

（3）再次查看 R1 到达 44.4.4.0/24 的选路

```
r1#sh ip bgp
BGP table version is 6, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 ?
* i44.4.4.0/24      2.2.2.2                 44    100      0 4 i
*>i                 3.3.3.3                  0    100      3 4 ?
r i100.1.1.0/24     2.2.2.2                  0    100      0 4 i
r>i                 3.3.3.3                  0    100      3 4 i
r1#
```

说明：因为 R3 的 weight 值大于 R2，所以 R3 为最优路径。

结论：可以证明，按照选路顺序，虽然改变了某个属性，从而改变了选路，但是前一个属性的改变，再次影响了选路，就表示顺序必须是由前往后，只有在前一个属性无法比出结果，才会比较后面一个，如果前面的属性已经比出结果，则后面的属性已经无关紧要了，所以以上选路规则顺序成立。

BGP 路由聚合

路由表的大小，可以影响到路由器的转发速度，对于拥有庞大路由表的 BGP，如果能够尽可能地减小其路由表的大小，那么性能可以得到明显的提高。减少路由表的条目，缩小路由表的空间，可以使用对路由的汇总来实现，在 BGP 中，称为路由聚合。在 BGP 中做路由汇总，需要手工创建，只要有一条路由包含在汇总路由中，那么这条汇总路由即可生效。

当创建了 BGP 汇总路由后，并不表示一定能够缩小路由表大小，因为在创建汇总路由后，被汇总的明细路由默认依然会通告给邻居，所以路由条目并没有减少，路由表的大小也就没有缩小。对于被包含在汇总路由中的明细路由是否需要通告给邻居，是可以自定义的，只要将某些路由抑制住，那么这些路由就不会通告给邻居。也可以选择抑制所有明细路由而只发送汇总路由给邻居。

因为汇总路由往往包含多条明细路由，而这些明细路由可能会拥有各不相同的 AS_Path 属性，默认情况下，汇总路由会将所有明细路由的 AS_Path 全部去掉，当汇总路由发给其它邻居之后，由于 AS_Path 的丢失，所以很有可能造成路由环路，因此 BGP 会在汇总路由中附加一定的属性来提示该路由产生了路径丢失，需要 BGP 路由器额外小心，这个属性就是 atomic-aggregate。

在 BGP 中创建汇总路由之后，默认会去掉明细路由中的所有 AS_Path，但也可以选择让汇总路由保留所有明细路由的 AS_Path，这个在汇总路由中的 AS_Path 称为 AS-SET，AS-SET 包含了所有明细路由的所有 AS_Path，而这些 AS_Path 的排列是没有固定顺序的，并且放在括号中，如 AS 15，AS 25，AS 35，AS 45，变成 AS-SET，很有可能就是{35，15，45，25}。由此可见，拥有 AS-SET 的汇总路由没有丢失路径，所以这样的汇总路由就不需要携带 atomic-aggregate 属性，也不会携带 atomic-aggregate 属性。汇总路由是否使用 AS-SET，可以自由决定。因为 AS-SET 中可能包含多个 AS，但即使一个 AS-SET 中有多个 AS，但在计算 AS_Path 长度时，只被计算为 1 个 AS。

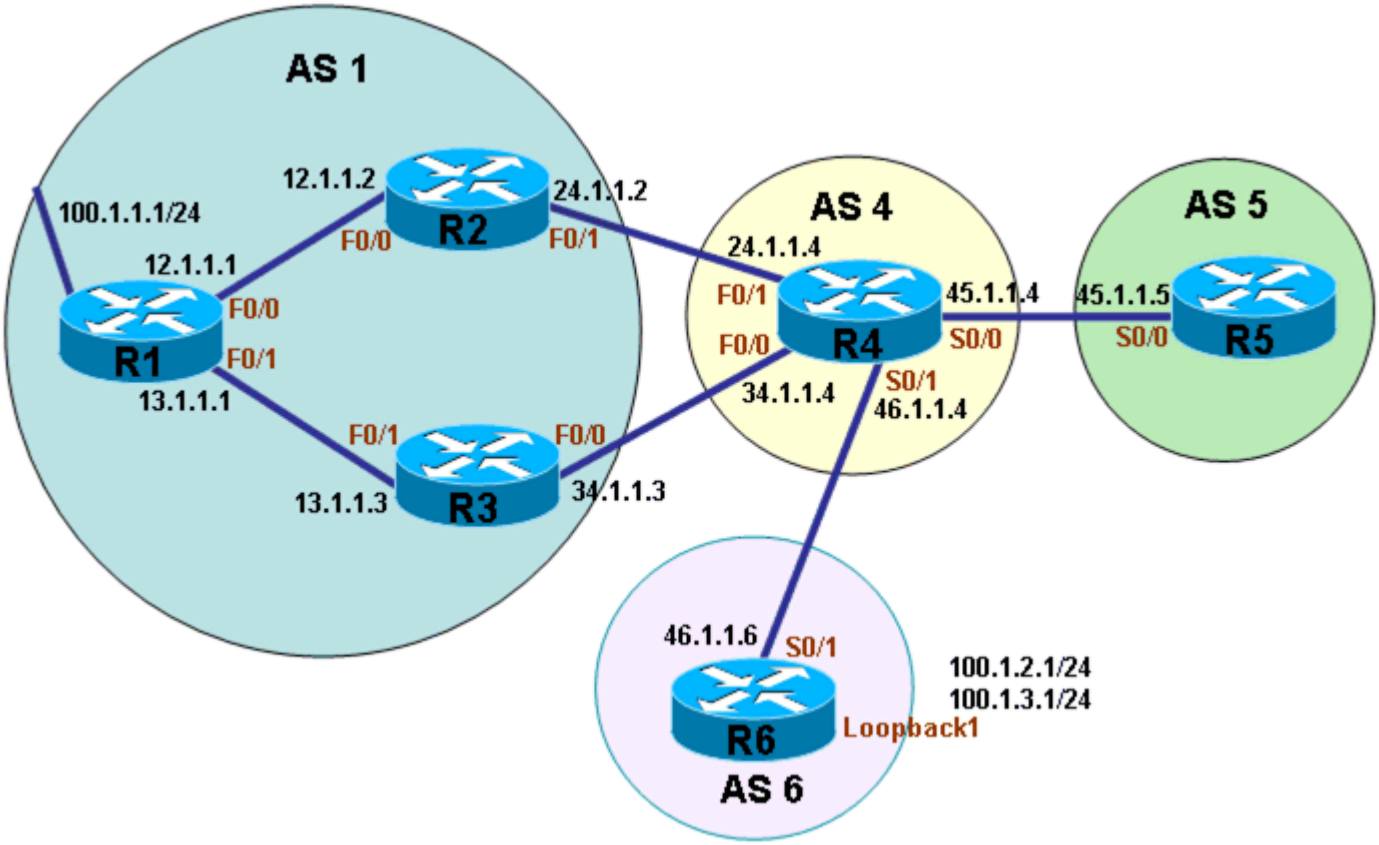
创建汇总路由的 BGP 路由器被认为是该路由的起源，所以汇总路由在该路由器上为本地路由，AD 值为 200，下一跳属性为 0.0.0.0。

创建 BGP 汇总路由的方法有两种：

1．手工命令创建汇总路由

2．创建静态路由并重分布进 BGP。

配置 BGP 路由聚合



说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32 Loopback 11 11.1.1.1/24

R2 Loopback 0 2.2.2.2/32 Loopback 22 22.2.2.2/24

R3 Loopback 0 3.3.3.3/32 Loopback 33 33.3.3.3/24

R4 Loopback 0 4.4.4.4/32 Loopback 44 44.4.4.4/24

R5 Loopback 0 5.5.5.5/32 Loopback 55 55.5.5.5/24

R6 Loopback 0 6.6.6.6/32 Loopback 66 66.6.6.6/24

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

1. IGP 保证全网 Loopback 0 互通

（1）配置 OSPF

说明：此步略，请参见之前配置。

（2）测试全网 Loopback 0 连通性

```
r1#ping 2.2.2.2 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/50/88 ms
r1#ping 3.3.3.3 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/43/76 ms
r1#ping 4.4.4.4 source loopback 0
```

```
    Type escape sequence to abort.
    Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
    Packet sent with a source address of 1.1.1.1
    !!!!!
    Success rate is 100 percent (5/5), round-trip min/avg/max = 16/74/116
ms
    r1#ping 5.5.5.5 source loopback 0

    Type escape sequence to abort.
    Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
    Packet sent with a source address of 1.1.1.1
    !!!!!
    Success rate is 100 percent (5/5), round-trip min/avg/max = 132/172/228
ms
    r1#ping 6.6.6.6 source loopback 0

    Type escape sequence to abort.
    Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
    Packet sent with a source address of 1.1.1.1
    !!!!!
    Success rate is 100 percent (5/5), round-trip min/avg/max = 96/169/248
ms
    r1#
```

说明：全网 Loopback 0 连通性连通性正常。

2.配置全网 BGP

（1）配置 R1 的 BGP

```
r1(config)#router bgp 1
r1(config-router)#bgp router-id 1.1.1.1
r1(config-router)#neighbor 2.2.2.2 remote-as 1
r1(config-router)#neighbor 2.2.2.2 update-source loopback 0
r1(config-router)#neighbor 3.3.3.3 remote-as 1
r1(config-router)#neighbor 3.3.3.3 update-source loopback 0
r1(config-router)#network 11.1.1.0 mask 255.255.255.0
```

（2）配置 R2 的 BGP

```
r2(config)#router bgp 1
r2(config-router)#bgp router-id 2.2.2.2
r2(config-router)#neighbor 1.1.1.1 remote-as 1
r2(config-router)#neighbor 1.1.1.1 update-source loopback 0
r2(config-router)#neighbor 4.4.4.4 remote-as 4
r2(config-router)#neighbor 4.4.4.4 update-source loopback 0
r2(config-router)#neighbor 4.4.4.4 ebgp-multihop
r2(config-router)#network 22.2.2.0 mask 255.255.255.0
```

（3）配置 R3 的 BGP

```
r3(config)#router bgp 1
r3(config-router)#bgp router-id 3.3.3.3
r3(config-router)#neighbor 1.1.1.1 remote-as 1
r3(config-router)#neighbor 1.1.1.1 update-source loopback 0
r3(config-router)#neighbor 4.4.4.4 remote-as 4
r3(config-router)#neighbor 4.4.4.4 update-source loopback 0
r3(config-router)#neighbor 4.4.4.4 ebgp-multihop
r3(config-router)#network 33.3.3.0 mask 255.255.255.0
```

（4）配置 R4 的 BGP

```
r4(config)#router bgp 4
r4(config-router)#bgp router-id 4.4.4.4
r4(config-router)#neighbor 2.2.2.2 remote-as 1
r4(config-router)#neighbor 2.2.2.2 update-source loopback 0
r4(config-router)#neighbor 2.2.2.2 ebgp-multihop
r4(config-router)#neighbor 3.3.3.3 remote-as 1
r4(config-router)#neighbor 3.3.3.3 update-source loopback 0
r4(config-router)#neighbor 3.3.3.3 ebgp-multihop
r4(config-router)#neighbor 5.5.5.5 remote-as 5
r4(config-router)#neighbor 5.5.5.5 update-source loopback 0
r4(config-router)#neighbor 5.5.5.5 ebgp-multihop
r4(config-router)#neighbor 6.6.6.6 remote-as 6
r4(config-router)#neighbor 6.6.6.6 update-source loopback 0
```

```
r4(config-router)#neighbor 6.6.6.6 ebgp-multihop

r4(config-router)#network 44.4.4.0 mask 255.255.255.0
```

（5）配置 R5 的 BGP

```
r5(config)#router bgp 5

r5(config-router)#bgp router-id 5.5.5.5

r5(config-router)#neighbor 4.4.4.4 remote-as 4

r5(config-router)#neighbor 4.4.4.4 update-source loopback 0

r5(config-router)#neighbor 4.4.4.4 ebgp-multihop

r5(config-router)#network 55.5.5.0 mask 255.255.255.0
```

（6）配置 R6 的 BGP

```
r6(config)#router bgp 6

r6(config-router)#bgp router-id 6.6.6.6

r6(config-router)#neighbor 4.4.4.4 remote-as 4

r6(config-router)#neighbor 4.4.4.4 update-source loopback 0

r6(config-router)#neighbor 4.4.4.4 ebgp-multihop

r6(config-router)#network 66.6.6.0 mask 255.255.255.0
```

3.创建 BGP 路由汇总

（1）在 R1 和 R6 上添加 BGP 明细路由

```
r1(config)#int loopback 100

r1(config-if)#ip address 100.1.1.1 255.255.255.0
```

```
r1(config)#router bgp 1

r1(config-router)#network 100.1.1.0 mask 255.255.255.0



r6(config)#int loopback 100

r6(config-if)#ip address 100.1.2.1 255.255.255.0

r6(config-if)#ip address 100.1.3.1 255.255.255.0 secondary


r6(config)#router bgp 6

r6(config-router)#network 100.1.2.0 mask 255.255.255.0

r6(config-router)#network 100.1.3.0 mask 255.255.255.0
```

说明：在 R1 和 R6 上添加路由，以供 BGP 汇总使用。

（2）在 R5 上查看 BGP 路由

```
r5#sh ip bgp

BGP table version is 10, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      4.4.4.4                                0 4 1 i
```

```
*> 22.2.2.0/24       4.4.4.4                              0 4 1 i
*> 33.3.3.0/24       4.4.4.4                              0 4 1 i
*> 44.4.4.0/24       4.4.4.4                  0           0 4 i
*> 55.5.5.0/24       0.0.0.0                  0       32768 i
*> 66.6.6.0/24       4.4.4.4                              0 4 6 i
*> 100.1.1.0/24      4.4.4.4                              0 4 1 i
*> 100.1.2.0/24      4.4.4.4                              0 4 6 i
*> 100.1.3.0/24      4.4.4.4                              0 4 6 i
r5#
```

说明： R5 上看到 R1 和 R6 发出的路由 100.1.1.0/24, 100.1.2.0/24, 100.1.3.0/24，并且 AS_Path 有所不同，分别为 4,6 和 4,1，可以看出分别由 AS 1 和 AS4 发起，并经过了 AS 4。

（3）在 R4 上创建 BGP 路由汇总

```
r4(config)#router bgp 4
r4(config-router)#aggregate-address 100.1.0.0 255.255.252.0
```

说明： 创建路由汇总 100.1.0.0/22

4.查看 BGP 汇总路由

(1)在 R5 上查看 BGP 汇总路由

```
r5#sh ip bgp
BGP table version is 11, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
```

```
                r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      4.4.4.4                                0 4 1 i

*> 22.2.2.0/24      4.4.4.4                                0 4 1 i

*> 33.3.3.0/24      4.4.4.4                                0 4 1 i

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

*> 55.5.5.0/24      0.0.0.0                  0         32768 i

*> 66.6.6.0/24      4.4.4.4                                0 4 6 i

*> 100.1.0.0/22     4.4.4.4                  0             0 4 i

*> 100.1.1.0/24     4.4.4.4                                0 4 1 i

*> 100.1.2.0/24     4.4.4.4                                0 4 6 i

*> 100.1.3.0/24     4.4.4.4                                0 4 6 i

r5#
```

说明：可以看到，R5 除了收到 BGP 汇总路由 100.1.0.0/22 之外，其中的明细路由也同样收到，并且汇总路由 100.1.0.0/22 的 AS_Path 为 4，说明 AS_Path 缺失。

（2）查看 atomic-aggregate

```
r5#sh ip bgp 100.1.0.0

BGP routing table entry for 100.1.0.0/22, version 32

Paths: (1 available, best #1, table Default-IP-Routing-Table)

  Not advertised to any peer
```

4, (aggregated by 4 4.4.4.4)

4.4.4.4 (metric 65) from 4.4.4.4 (4.4.4.4)

Origin IGP, metric 0, localpref 100, valid, external, atomic-aggregate, best

r5#

说明：因为 100.1.0.0/22 产生了路径丢失，所以该路由中携带 atomic-aggregate 属性，说明路径丢失。

5.调整 BGP 汇总路由路径信息

（1）调整 AS-SET

r4(config)#router bgp 4

r4(config-router)#aggregate-address 100.1.0.0 255.255.252.0 as-set

说明：在 R4 创建汇总路由时，使用 AS-SET，以保留所有明细路由的 AS_Path。

（2）再次查看 R5 的 BGP 汇总路由

r5#sh ip bgp

BGP table version is 12, local router ID is 5.5.5.5

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

r RIB-failure, S Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	11.1.1.0/24	4.4.4.4			0	4 1 i
*>	22.2.2.0/24	4.4.4.4			0	4 1 i

```
*> 33.3.3.0/24       4.4.4.4                                0 4 1 i
*> 44.4.4.0/24       4.4.4.4                  0             0 4 i
*> 55.5.5.0/24       0.0.0.0                  0         32768 i
*> 66.6.6.0/24       4.4.4.4                                0 4 6 i
*> 100.1.0.0/22      4.4.4.4                  0             0 4 {1,6} i
*> 100.1.1.0/24      4.4.4.4                                0 4 1 i
*> 100.1.2.0/24      4.4.4.4                                0 4 6 i
*> 100.1.3.0/24      4.4.4.4                                0 4 6 i
r5#
```

说明：汇总路由 100.1.0.0/22 包含了所有明细路由的 AS_Path，因此没有路径丢失。

（3）查看 atomic-aggregate

```
r5#sh ip bgp 100.1.0.0
BGP routing table entry for 100.1.0.0/22, version 33
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Not advertised to any peer
  4 {1,6}, (aggregated by 4 4.4.4.4)
    4.4.4.4 (metric 65) from 4.4.4.4 (4.4.4.4)
      Origin IGP, metric 0, localpref 100, valid, external, best
r5#
```

说明：因为 100.1.0.0/22 没有产生路径丢失，所以该路由中没有携带 atomic-aggregate 属性。

6.调整 BGP 汇总路由抑制

（1）在 R4 上抑制不需要通告的明细路由

```
r4(config)#access-list 1 permit 100.1.1.0

r4(config)#route-map sup permit 10

r4(config-route-map)#match ip address 1


r4(config)#router bgp 4

r4(config-router)#aggregate-address 100.1.0.0 255.255.252.0 as-set
suppress-map sup
```

说明：将 100.1.1.0/24 放入 route-map，在 suppress-map 中的 route-map 所包含的路由便会被抑制而不会通告给邻居。

（2）在 R5 上查看 BGP 路由

```
r5#sh ip bgp

BGP table version is 12, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      4.4.4.4                                0 4 1 i
```

```
*> 22.2.2.0/24       4.4.4.4                              0 4 1 i

*> 33.3.3.0/24       4.4.4.4                              0 4 1 i

*> 44.4.4.0/24       4.4.4.4                  0           0 4 i

*> 55.5.5.0/24       0.0.0.0                  0       32768 i

*> 66.6.6.0/24       4.4.4.4                              0 4 6 i

*> 100.1.0.0/22      4.4.4.4                  0           0 4 {1,6} i

*> 100.1.2.0/24      4.4.4.4                              0 4 6 i

*> 100.1.3.0/24      4.4.4.4                              0 4 6 i

r5#
```

说明：可以看到，R5 并没有收到明细路由 100.1.1.0/24，说明被抑制了。

（3）在 R4 上抑制全部明细路由

```
r4(config)#router bgp 4

r4(config-router)#aggregate-address 100.1.0.0 255.255.252.0 as-set
summary-only
```

说明：在 R4 上抑制全部明细路由，只发汇总路由。

（4）在 R5 上查看 BGP 路由

```
r5#sh ip bgp

BGP table version is 15, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      4.4.4.4                                0 4 1 i
*> 22.2.2.0/24      4.4.4.4                                0 4 1 i
*> 33.3.3.0/24      4.4.4.4                                0 4 1 i
*> 44.4.4.0/24      4.4.4.4                  0             0 4 i
*> 55.5.5.0/24      0.0.0.0                  0         32768 i
*> 66.6.6.0/24      4.4.4.4                                0 4 6 i
*> 100.1.0.0/22     4.4.4.4                  0             0 4 {1,6} i
r5#
```

说明：R5 只能收到汇总路由，其它明细路由已经被全部抑制。

（5）在 R4 上查看抑制路由

```
r4#sh ip bgp
BGP table version is 15, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      2.2.2.2                                0 1 i
*                   3.3.3.3                                0 1 i
*> 22.2.2.0/24      2.2.2.2                  0             0 1 i
```

```
*> 33.3.3.0/24       3.3.3.3                  0             0 1 i
*> 44.4.4.0/24       0.0.0.0                  0         32768 i
*> 55.5.5.0/24       5.5.5.5                  0             0 5 i
*> 66.6.6.0/24       6.6.6.6                  0             0 6 i
*> 100.1.0.0/22      0.0.0.0                        100   32768 {1,6} i
s   100.1.1.0/24      3.3.3.3                        0 1 i
s>                   2.2.2.2                        0 1 i
s> 100.1.2.0/24      6.6.6.6                  0             0 6 i
s> 100.1.3.0/24      6.6.6.6                  0             0 6 i
r4#
```

说明：路由前面的标识 s 表示该路由在汇总时被抑制而不发给邻居。

（6）使用不抑制

```
r4(config)#access-list 3 permit 100.1.3.0

r4(config)#route-map unsup permit 10
r4(config-route-map)#match ip address 3

r4(config)#router bgp 4
r4(config-router)#neighbor 5.5.5.5 unsuppress-map unsup
```

说明：在使用了抑制路由之后，如果希望某些路由还是发给某邻居而不抑制，那么可以使用不抑制的映射列表 unsuppress-map，在该列表后面包含的路由都将被发给邻居。

（7）在 R5 上查看 BGP 路由

```
r5#sh ip bgp

BGP table version is 16, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      4.4.4.4                                0 4 1 i

*> 22.2.2.0/24      4.4.4.4                                0 4 1 i

*> 33.3.3.0/24      4.4.4.4                                0 4 1 i

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

*> 55.5.5.0/24      0.0.0.0                  0         32768 i

*> 66.6.6.0/24      4.4.4.4                                0 4 6 i

*> 100.1.0.0/22     4.4.4.4                  0             0 4 {1,6} i

*> 100.1.3.0/24     4.4.4.4                                0 4 6 i

r5#
```

说明：可以看见，因为 R4 上配置上对 100.1.3.0/24 不抑制，所有 R5 在收到汇总路由之后，还能收到不被抑制的路由。

7.重分布汇总路由

（1）在 R4 上创建静态路由并重分布进 BGP

```
r4(config)#ip route 100.1.0.0 255.255.252.0 null 0

r4(config)#router bgp 4

r4(config-router)#redistribute static
```

说明：手工创建静态路由方式的汇总路由，并指向 null 0 以防止路由黑洞。手工创建的默认路由是不能被重分布进 BGP 的。

（2）在 R5 上查看路由表

```
r5#sh ip bgp

BGP table version is 35, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      4.4.4.4                                0 4 1 i

*> 22.2.2.0/24      4.4.4.4                                0 4 1 i

*> 33.3.3.0/24      4.4.4.4                                0 4 1 i

*> 44.4.4.4.0/24    4.4.4.4                  0             0 4 i

*> 55.5.5.5.0/24    0.0.0.0                  0         32768 i

*> 66.6.6.6.0/24    4.4.4.4                                0 4 6 i
```

```
*> 100.1.0.0/22      4.4.4.4                  0             0 4 ?
*> 100.1.1.0/24      4.4.4.4                                0 4 1 i
*> 100.1.2.0/24      4.4.4.4                                0 4 6 i
*> 100.1.3.0/24      4.4.4.4                                0 4 6 i
```

R5

说明：R5 已经收到重分布进的汇总路由信息。

BGP 默认路由

BGP 的默认路由只能创建，而不能通过静态重分布。创建默认路由，可以选择对所有邻居生效，也可以只针对某个邻居发布默认路由。

配置 BGP 默认路由

说明：以上个实验环境为基础，继续 BGP 默认路由的实验。

1.创建 BGP 默认

(1)在 R4 上发布 BGP 默认路由

```
r4(config)#router bgp 4
r4(config-router)#default-information originate
```

说明：没有指定邻居，则向所有邻居通告默认路由。

（2）在 R5 上查看默认路由

```
r5#sh ip bgp
BGP table version is 37, local router ID is 5.5.5.5
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          4.4.4.4                  0             0 4 ?
*> 11.1.1.0/24      4.4.4.4                                0 4 1 i
*> 22.2.2.0/24      4.4.4.4                                0 4 1 i
*> 33.3.3.0/24      4.4.4.4                                0 4 1 i
*> 44.4.4.0/24      4.4.4.4                  0             0 4 i
*> 55.5.5.0/24      0.0.0.0                  0         32768 i
*> 66.6.6.0/24      4.4.4.4                                0 4 6 i
*> 100.1.0.0/22     4.4.4.4                  0             0 4 ?
*> 100.1.1.0/24     4.4.4.4                                0 4 1 i
*> 100.1.2.0/24     4.4.4.4                                0 4 6 i
*> 100.1.3.0/24     4.4.4.4                                0 4 6 i
*> 200.0.0.0/8      4.4.4.4                  0             0 4 ?
r5#
```

说明：R5 收到 R4 发来的默认路由。

（3）只对单个邻居发送默认路由

```
r4(config-router)#neighbor 5.5.5.5 default-originate
```

说明：此邻居包含邻居，即可针对单个邻居通告默认路由。

BGP 路由过滤

对于在 BGP 中过滤路由，可以有多种方法，比如通过利用 access-list 或 prefix-list 来匹配中特定路由，然后在 BGP 进程中使用，可以针对所有邻居使用，也可以针对特定邻居使用。

因为 BGP 的路由通常会携带 AS_Path，所以除了根据路由的 IP 来过滤之外，还可以根据路由携带的 AS_Path 来过滤，要匹配路由的 AS_Path，需要使用 Regular Expressions（正则表达式）来匹配 AS 特征，对于 Regular Expressions（正则表达式）的匹配规则如下：

. 任何一个单一字符，包括空格

* 字符或模式出现 0 次或多次

^ 一行的开始

_ 类似于逗号

$ 一行的结束

举例：

.*

表示匹配任意

^123$

表示只匹配 AS 123

^$

表示没有经过任何 AS，即本地 AS 的路由

^12[0-3]$

表示匹配 120 121 122 123

^12.

表示匹配 12 ，120 - 129 开始的 AS 号

_12_

表示匹配经过了 AS 12 的路由

配置 BGP 路由过滤

说明：以上个实验环境为基础，继续 BGP 路由过滤的实验。

1.配置 R5 使用 As-path filter 路由过滤

（1）查看 R5 当前路由情况

```
r5#sh ip bgp
BGP table version is 13, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      4.4.4.4                              0 4 1 i
*> 22.2.2.0/24      4.4.4.4                              0 4 1 i
*> 33.3.3.0/24      4.4.4.4                              0 4 1 i
*> 44.4.4.0/24      4.4.4.4                  0           0 4 i
*> 55.5.5.0/24      0.0.0.0                  0       32768 i
*> 66.6.6.0/24      4.4.4.4                              0 4 6 i
*> 100.1.0.0/22     4.4.4.4                  0           0 4 {1,6} i
*> 100.1.1.0/24     4.4.4.4                              0 4 1 i
*> 100.1.2.0/24     4.4.4.4                              0 4 6 i
*> 100.1.3.0/24     4.4.4.4                              0 4 6 i
r5#
```

说明：R5 当前拥有所有网段的路由。

（2）只收起源于 AS 6 的路由，过滤其它所有路由

http://itcto.taobao.com 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

```
r5(config)#ip as-path access-list 5 permit ^6$


r5(config)#router bgp 5

r5(config-router)#neighbor 4.4.4.4 filter-list 5 in
```

说明：在 R5 方向应用 As-path filter

(3)查看过滤后的路由情况

```
r5#sh ip bgp

BGP table version is 22, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 55.5.5.0/24      0.0.0.0                  0         32768 i

r5#
```

说明：除了自己本地路由之外，没有收到任何路由，因为^6$不与任何路由匹配，并不表示为起源于 AS 6 的路由。

（3）改写起源于 AS 6 的路由

```
r5(config)#ip as-path access-list 55 permit _6$

r5(config)#router bgp 5
```

```
r5(config-router)#neighbor 4.4.4.4 filter-list 55 in
```

说明：起源于 AS 6 的路由格式应该为_6$，因为 AS 的方向问题，并且符合_可以表示逗号的意思。

（4）查看正确过滤后的路由情况

```
r5#sh ip bgp
BGP table version is 19, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 55.5.5.0/24      0.0.0.0                  0         32768 i
*> 66.6.6.0/24      4.4.4.4                                0 4 6 i
*> 100.1.2.0/24     4.4.4.4                                0 4 6 i
*> 100.1.3.0/24     4.4.4.4                                0 4 6 i
r5#
```

说明：使用 As-path filter 过滤路由成功，只有起源于 AS 6 的路由。

2.在 R4 上配置 distribute-list 路由过滤

（1）配置全局过滤路由

```
r4(config)#access-list 2 permit 100.1.2.0


r4(config)#router bgp 4

r4(config-router)#distribute-list 2 in
```

说明：该方式配置全局过滤路由，只接收 100.1.2.0 的路由。

（2）查看过滤后的路由情况

```
r4#sh ip bgp

BGP table version is 9, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 44.4.4.0/24      0.0.0.0                  0         32768 i

*> 100.1.0.0/22     0.0.0.0                        100  32768 {1,6} i

*> 100.1.2.0/24     6.6.6.6                  0             0 6 i

r4#
```

说明：除了本地路由外，只接收到 100.1.2.0 的路由，说明路由过滤成功。

（3）针对单个邻居过滤路由

```
r4(config)#router bgp 4
```

r4(config-router)#no distribute-list 2 in

r4(config-router)#neighbor 6.6.6.6 distribute-list 2 in

说明：配置过滤路由只针对某个邻居 R6 生效。

（4）查看过滤后的路由情况

r4#sh ip bgp

BGP table version is 9, local router ID is 4.4.4.4

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

r RIB-failure, S Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	11.1.1.0/24	3.3.3.3			0	1 i
*>		2.2.2.2			0	1 i
*>	22.2.2.0/24	2.2.2.2	0		0	1 i
*>	33.3.3.0/24	3.3.3.3	0		0	1 i
*>	44.4.4.0/24	0.0.0.0	0		32768	i
*>	55.5.5.0/24	5.5.5.5	0		0	5 i
*>	100.1.0.0/22	0.0.0.0		100	32768	{1,6} i
*	100.1.1.0/24	3.3.3.3			0	1 i
*>		2.2.2.2			0	1 i
*>	100.1.2.0/24	6.6.6.6	0		0	6 i

r4#

说明：只从 R6 收到 100.1.2.0 的路由，说明针对单个邻居过滤路由生效。

BGP 条件路由

BGP 可以使用有条件的路由，通过定义某个条件，来限制路由的发送。

配置 BGP 条件路由后，只有当定义的条件满足时，相应的路由才会被通告给邻居，否则是被抑制的。

BGP 条件路由通过两个 route-map 来实现，advertise map 和 exist map，定义 route-map 中的路由，可以使用 access list 或 IP prefix list，配置完之后，BGP 会跟踪 exist map 中的路由，只有相应路由存在 BGP 路由表中时，advertise map 中的路由才会被通告给邻居，否则是被抑制的。

路由的条件除了存在路由表中外，还可以是消失于路由表中，需要 nonexist map 来定义，而 advertise map 保持不变。

需要重点说明的是，BGP 条件路由为 IOS 中一个不稳定的技术，效果并不明显，也不稳定，所以使用中请仔细斟酌并小心使用，否则会有预料之外的效果。

配置 BGP 条件路由

说明：以上个实验环境为基础，继续 BGP 条件路由的实验。

配置 R4 只有当 100.1.2.0 存在于 BGP 路由表时，才向邻居 5.5.5.5 通告 11.1.1.0。

1.配置 BGP 条件路由

（1）查看 R5 当前路由情况

```
r5#sh ip bgp
BGP table version is 9, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      4.4.4.4                                  0 4 1 i
*> 22.2.2.0/24      4.4.4.4                                  0 4 1 i
*> 33.3.3.0/24      4.4.4.4                                  0 4 1 i
*> 44.4.4.0/24      4.4.4.4                  0               0 4 i
*> 66.6.6.0/24      4.4.4.4                                  0 4 6 i
*> 100.1.1.0/24     4.4.4.4                                  0 4 1 i
*> 100.1.2.0/24     4.4.4.4                                  0 4 6 i
*> 100.1.3.0/24     4.4.4.4                                  0 4 6 i
r5#
```

说明：R5 当前拥有全部路由，包括 11.1.1.0。

（2）查看 R4 当前路由情况

```
r4#sh ip bgp
BGP table version is 20, local router ID is 4.4.4.4
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  11.1.1.0/24      3.3.3.3                                0 1 i
*>                  2.2.2.2                                0 1 i
*> 22.2.2.0/24      2.2.2.2                  0             0 1 i
*> 33.3.3.0/24      3.3.3.3                  0             0 1 i
*> 44.4.4.0/24      0.0.0.0                  0         32768 i
*> 55.5.5.0/24      5.5.5.5                  0             0 5 i
*> 66.6.6.0/24      6.6.6.6                  0             0 6 i
*  100.1.1.0/24     3.3.3.3                                0 1 i
*>                  2.2.2.2                                0 1 i
*> 100.1.2.0/24     6.6.6.6                  0             0 6 i
*> 100.1.3.0/24     6.6.6.6                  0             0 6 i
r4#
```

说明：R4 当前拥有全部路由，包括 11.1.1.0 和 100.1.2.0。

（3）在 R4 上配置 BGP 条件路由

```
r4(config)#access-list 11 permit 11.1.1.0
r4(config)#access-list 2 permit 100.1.2.0
```

```
r4(config)#route-map adv permit 10
r4(config-route-map)#match ip address 11
r4(config-route-map)#exit

r4(config)#route-map exi permit 10
r4(config-route-map)#match ip address 2

r4(config)#router bgp 4
r4(config-router)#neighbor 5.5.5.5 advertise-map adv exist-map exi
```

说明：配置只有当 100.1.2.0 存在于 BGP 路由表时，才向邻居 5.5.5.5 通告 11.1.1.0。

（4）测试使 100.1.2.0 消失

```
r6(config)#router bgp 6
r6(config-router)#no network 100.1.2.0 mask 255.255.255.0
```

说明：使 100.1.2.0 消失。

（5）查看 R4 的 BGP 路由情况

```
r4#sh ip bgp
BGP table version is 21, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
```

```
                r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*  11.1.1.0/24      3.3.3.3                                0 1 i

*>                  2.2.2.2                                0 1 i

*> 22.2.2.0/24      2.2.2.2                  0             0 1 i

*> 33.3.3.0/24      3.3.3.3                  0             0 1 i

*> 44.4.4.0/24      0.0.0.0                  0         32768 i

*> 55.5.5.0/24      5.5.5.5                  0             0 5 i

*> 66.6.6.0/24      6.6.6.6                  0             0 6 i

*  100.1.1.0/24     3.3.3.3                                0 1 i

*>                  2.2.2.2                                0 1 i

*> 100.1.3.0/24     6.6.6.6                  0             0 6 i

r4#
```

说明： 100.1.2.0 已经从 BGP 路由表中消失。

（6）查看 R5 的 BGP 路由情况

```
r5#sh ip bgp

BGP table version is 23, local router ID is 5.5.5.5

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

                r RIB-failure, S Stale
```

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

   Network          Next Hop            Metric LocPrf Weight Path
*> 22.2.2.0/24      4.4.4.4                                0 4 1 i
*> 33.3.3.0/24      4.4.4.4                                0 4 1 i
*> 55.5.5.0/24      0.0.0.0                  0         32768 i
*> 66.6.6.0/24      4.4.4.4                                0 4 6 i
*> 100.1.1.0/24     4.4.4.4                                0 4 1 i
*> 100.1.3.0/24     4.4.4.4                                0 4 6 i

r5#
```

说明：由于路由 100.1.2.0 的消失，所以 11.1.1.0 被抑制，导致 R5 没有收到 11.1.1.0。

BGP Peer Group

从 BGP 的配置中可以看到，在配置邻居时，需要使用多条命令指定多个参数，比如 AS 号码，BGP 更新源地址，TTL 等等，才能够配置一个正常的邻居；而 BGP 是使用在大型网络中的，这就意味着一台 BGP 路由器将要使用许许多多的命令来完成邻居的建立，而这其中势必会有许多邻居都拥有相同的配置参数。

为了能够简化 BGP 对邻居的参数配置，BGP 使用了 Peer Group 的概念，BGP 的 Peer Group 就相当于是一个容器，这个容器拥有着 BGP 参数和策略，只要将 BGP 邻居放入这个容器中，那么该邻居即可获得容器的所有参数和策略，从而大大简化为每个邻居重复配置相同参数和策略。

BGP 的 Peer Group 创建之后，就可以为其配置参数，所有可以为邻居配置的一切参数，都可以为 Peer Group 配置，在配置了 Peer Group 之后，就可以不必再为每个邻居一一配置参数，只要将邻居划入 Peer Group 即可，对 Peer Group 配置的参数会对 Peer Group 中所有邻居生效。

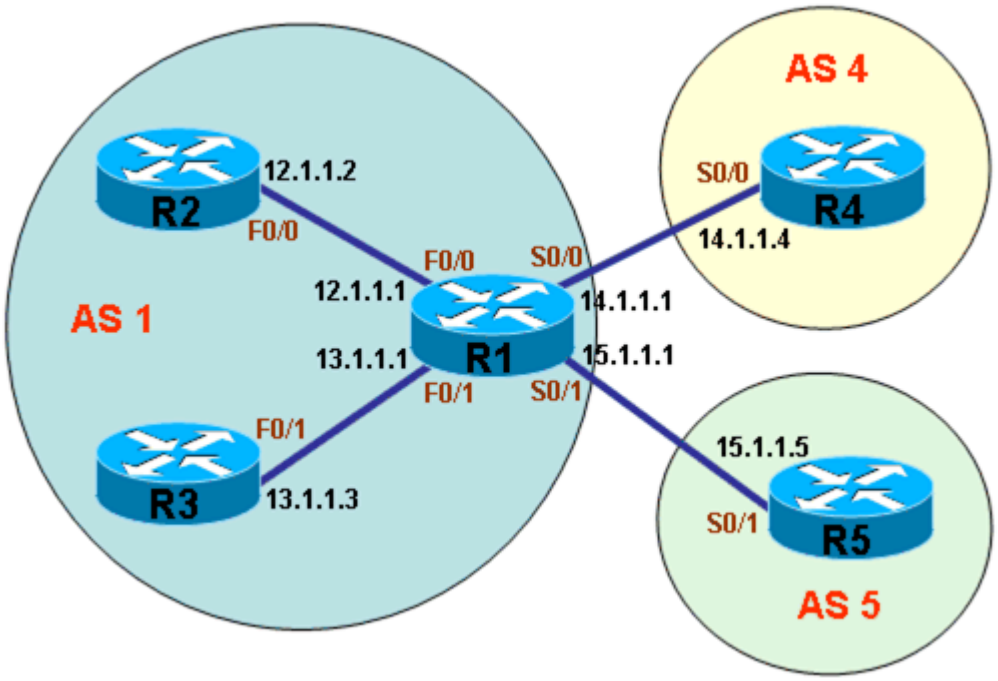
在使用普通方式配置 BGP 邻居时，假如配置一个特定的邻居需要 4 条命令，那么配置 10 个邻居就需要 40 条命令，在使用 Peer Group 时，创建 Peer Group 使用 1 条命令，再使用 4 条命令为 Peer Group 配置参数，最后再使用 10 条命令将 10 个邻居全部划入 Peer Group，可以看出，使用 Peer Group 配置 10 个邻居所使用的 15 条命令，远远少于使用普通方式的 40 条命令，从而体现出使用 Peer Group 对配置 BGP 工作量的简化是相当明显的。

除了可以对 Peer Group 配置各种参数外，各种可以为邻居配置的属性和策略，也完全可以对 Peer Group 进行配置。

Peer Group 唯一的限制就是，同一个 Peer Group 中的所有邻居，必须全部为 iBGP 邻居，或者全部为 eBGP 邻居，也就是说不能将 iBGP 邻居和 eBGP 邻居同时混杂在同一个 Peer Group 中，但是如果全部都为 eBGP 邻居，这些邻居可以是任意 AS 的，而不必所有邻居都是同一个 AS 的。

在使用 Peer Group 配置邻居后，可以对 Peer Group 配置参数和策略，也可以对 Peer Group 中的单个邻居配置参数和策略，如果对单个邻居配置，那么配置只对单个特定的邻居生效，而不影响 Peer Group 中其它邻居，所以在使用 Peer Group 配置减少工作量的同时，也能保证邻居策略的多样化。

配置 BGP Peer Group



说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32 Loopback 11 11.1.1.1/24

R2 Loopback 0 2.2.2.2/32 Loopback 22 22.2.2.2/24

R3 Loopback 0 3.3.3.3/32 Loopback 33 33.3.3.3/24

R4 Loopback 0 4.4.4.4/32 Loopback 44 44.4.4.4/24

R5 Loopback 0 5.5.5.5/32 Loopback 55 55.5.5.5/24

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

因为 R1 拥有多个 BGP 邻居，所以使用 BGP Peer Group 的方式来配置 R1，从而简化工作量。

1. IGP 保证全网 Loopback 0 互通

说明：使用 IGP 协议 OSPF 来保证全网 Loopback 0 互通，从而使用 Loopback 0 来传递 BGP 信息。

（1）测试全网 Loopback 0 连通性：

```
r1#ping 2.2.2.2 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 72/126/192 ms
r1#ping 3.3.3.3 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/99/192 ms
r1#
r1#ping 4.4.4.4 source loopback 0

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/70/152 ms

r1#

r1#ping 5.5.5.5 source loopback 0



Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:

Packet sent with a source address of 1.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/129/196
ms

r1#
```

说明：所有路由器之间的 Loopback 0 通信正常。

2.使用 Peer Group 配置 R1

（1）使用 Peer Group 配置同 AS 内的 iBGP 邻居

```
r1(config)#router bgp 1

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

r1(config-router)#neighbor as1 peer-group

r1(config-router)#neighbor as1 remote-as 1

r1(config-router)#neighbor as1 update-source loopback 0
```

```
r1(config-router)#neighbor 2.2.2.2 peer-group as1

r1(config-router)#neighbor 3.3.3.3 peer-group as1
```

说明：创建 Peer Group 为 as1 ，配置 as1 中的邻居为 as 1，更新时使用地址为 Loopback0 的地址，然后将邻居 2.2.2.2 和 3.3.3.3 放入 Peer Group(as1)中。

（2）使用 Peer Group 配置不同 AS 内的 eBGP 邻居

```
r1(config)#router bgp 1

r1(config-router)#neighbor as45 peer-group

r1(config-router)#neighbor as45 update-source loopback 0

r1(config-router)#neighbor as45 ebgp-multihop


r1(config-router)#neighbor 4.4.4.4 remote-as 4

r1(config-router)#neighbor 4.4.4.4 peer-group as1

% Peer with AS 4 cannot be in this peer-group, members must be all internal
or all external   （报错提示）

r1(config-router)#neighbor 4.4.4.4 peer-group as45


r1(config-router)#neighbor 5.5.5.5 remote-as 5

r1(config-router)#neighbor 5.5.5.5 peer-group as1

% Peer with AS 5 cannot be in this peer-group, members must be all internal
or all external   （报错提示）

r1(config-router)#neighbor 5.5.5.5 peer-group as45

r1(config-router)#
```

说明：创建 Peer Group 为 as45 ，配置 as45 更新时使用地址为 Loopback0 的地址，指定 TTL 为 255，但并没有指定 as45 中的邻居 AS，因为并不是所有 eBGP 邻居都在同一 AS,所以需要单独为各个邻居指定 AS 之后,才将邻居放入 Peer Group(as1)中，也可以看见，在指定邻居为其它 AS 之后，就不可能再放入本 AS 的 Peer Group 中，否则有如上报错信息。

3.使用普通方式配置其它 BGP 路由器

（1）使用普通方式配置 R2

```
r2(config)#router bgp 1

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

r2(config-router)#neighbor 1.1.1.1 remote-as 1

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

r2(config-router)#network 22.2.2.0 mask 255.255.255.0
```

说明：正常配置 R2，并发布相应网段，以做测试用。

（2）使用普通方式配置 R3

```
r3(config)#router bgp 1

r3(config-router)#bgp router-id 3.3.3.3

r3(config-router)#neighbor 1.1.1.1 remote-as 1

r3(config-router)#neighbor 1.1.1.1 up loopback 0

r3(config-router)#network 33.3.3.0 mask 255.255.255.0
```

说明：正常配置 R3，并发布相应网段，以做测试用。

（3）使用普通方式配置 R4

```
r4(config)#router bgp 4

r4(config-router)#bgp router-id 4.4.4.4

r4(config-router)#neighbor 1.1.1.1 remote-as 1

r4(config-router)#neighbor 1.1.1.1 up loopback 0

r4(config-router)#neighbor 1.1.1.1 ebgp-multihop

r4(config-router)#network 44.4.4.0 mask 255.255.255.0
```

说明：正常配置 R4，并发布相应网段，以做测试用。

（4）使用普通方式配置 R5

```
r5(config)#router bgp 5

r5(config-router)#bgp router-id 5.5.5.5

r5(config-router)#neighbor 1.1.1.1 remote-as 1

r5(config-router)#neighbor 1.1.1.1 update-source loopback 0

r5(config-router)#neighbor 1.1.1.1 ebgp-multihop

r5(config-router)#network 55.5.5.0 mask 255.255.255.0
```

说明：正常配置 R5，并发布相应网段，以做测试用。

4.查看结果

（1）查看 BGP 邻居状态

```
r1#sh ip bgp summary
```

```
BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 5, main routing table version 5

4 network entries using 516 bytes of memory

4 path entries using 208 bytes of memory

4/3 BGP path/bestpath attribute entries using 496 bytes of memory

2 BGP AS-PATH entries using 48 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 1268 total bytes of memory

BGP activity 4/0 prefixes, 4/0 paths, scan interval 60 secs


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

2.2.2.2         4     1       8       9        5    0    0
00:03:01        1

3.3.3.3         4     1       7       8        5    0    0
00:02:18        1

4.4.4.4         4     4       6       7        4    0    0
00:01:15        1

5.5.5.5         4     5       5       6        4    0    0
00:00:28        1

r1#
```

说明：在 R1 上可以看见，所有 BGP 邻居均已正常建立。

（2）查看 BGP 路由情况

鸿鹄培训+考试群：301283867

```
r1#sh ip bgp

BGP table version is 5, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i22.2.2.0/24      2.2.2.2                  0    100      0 i

*>i33.3.3.0/24      3.3.3.3                  0    100      0 i

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

*> 55.5.5.0/24      5.5.5.5                  0             0 5 i

r1#
```

说明：已经学习到所有 BGP 路由。

5.测试邻居参数

（1）修改对 Peer Group 的参数配置

```
r1(config)#router bgp 1

r1(config-router)#neighbor as1 weight 111
```

说明：将 Peer Group “as1”的 weight 值改为 111，将对 Peer Group 中所有邻居生效。

（2）查看修改结果

```
r1#sh ip bgp
```

```
BGP table version is 7, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i22.2.2.0/24      2.2.2.2                  0    100     111 i

*>i33.3.3.0/24      3.3.3.3                  0    100     111 i

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

*> 55.5.5.0/24      5.5.5.5                  0             0 5 i

r1#
```

说明：对 Peer Group 修改的参数已对 Peer Group 中所有邻居生效。

（3）修改单个邻居的参数配置

```
r1(config)#router bgp 1

r1(config-router)#neighbor 5.5.5.5 weight 5
```

说明：将 Peer Group “as45”的中邻居 5.5.5.5 的 weight 值改为 5，此改动只对 Peer Group 中的单个邻居生效，而不会影响其它邻居。

（4）查看修改结果

```
r1#sh ip bgp

BGP table version is 8, local router ID is 1.1.1.1

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

```
                    r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i22.2.2.0/24      2.2.2.2                  0    100     111 i

*>i33.3.3.0/24      3.3.3.3                  0    100     111 i

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

*> 55.5.5.0/24      5.5.5.5                  0             5 5 i

r1#
```

说明：对 Peer Group 中单个邻居修改的参数，只对 Peer Group 中单个邻居生效，而不影响其它邻居。

BGP Community

BGP Community 只是 BGP 路由可以携带的一种属性，是 BGP 中的一个可选可传递属性（Optional Transitive），所以可以选择为路由配置 Community，也可以不配，如果配置，Community 需要明确要求 BGP 路由器保留和传递该属性，否则邻居收不到路由的 Community 属性。

都知道 Peer Group 是用来简化 BGP 邻居配置的，而 BGP Community 则可以简化 BGP 路由在某方面的配置。对于 Community 属性，它只是 BGP 路由中携带的一个标签而已，可以分成不同的类别，分为众所周知的 Community 和私有 Community，也可以分为标准 Community 和扩展 Community（Extended Community），它们的区别如下：

众所周知的 Community

众所周知的 Community 被所有 BGP 路由器认识和理解，并且必须对携带众所周知的 Community 的路由做相应的操作，BGP 拥有 4 个预先定义的众所周知的 Community，分别为

no-export—不将路由发给任何 eBGP 邻居，也就是只能将该路由在本 AS 内部传递。

no-advertise—不将路由发给任何 BGP 邻居。

internet—可以将路由发给任何 BGP 邻居。

local-as—同 no-export，即不将路由发到 AS 外。

注：众所周知的 Community 为固定格式，不可自定义，只能使用预定义的格式。

私有 Community

私有 Community 可以理解为 BGP 路由的自定义标签，所以可以通过为 BGP 路由配置私有 Community 来配置任何自定义的标签，该标签可以在任何时候被利用。例如一台 BGP 路由器为某个 Community 标签配置策略后，那么所有携带该 Community 标签的路由都将获得相应策略。

在正常情况下，BGP 路由器要对某些一定范围内的路由配置策略，必须使用 prefix list 或 access list 将所有符合条件的路由匹配出来，然后调用之后再配置相应策略，如果是网络中所有路由器都要对这些路由设置策略，那么就必须在网络中每台设备上单独使用大量重复的配置将路由匹配出来，再做相应策略，工作烦琐并且容易出错，而在使用私有 Community 之后，就可以将特定的路由设置私有 Community，并将其传递给所有邻居，最终所有路由器都对拥有该私有 Community 的路由配置策略，并且对大量路由设置私有 Community 只需要在一台路由器上完成后，发给所有邻居即可，可见私有 Community 可以减少网络中路由器对相同路由的匹配工作，这就是标签的效果。

私有 Community 的类型为数字，长度为 32bit，但被分为两种格式：

单个 32bit，如 123，666

或者 2 字节长度的 AS 号码加两字节普通数字，称为 AS:NN 格式，范围为 1:0 至 65534:65535。

默认路由器支持单个 32bit 格式，若要支持 AS:NN 格式，必须开启 BGP-Community New-Format 功能。

标准 Community

标准 Community 就是普通路由可以设置的 Community，包括上面提到的众所周知 Community 和私有 Community。

扩展 Community（Extended Community）

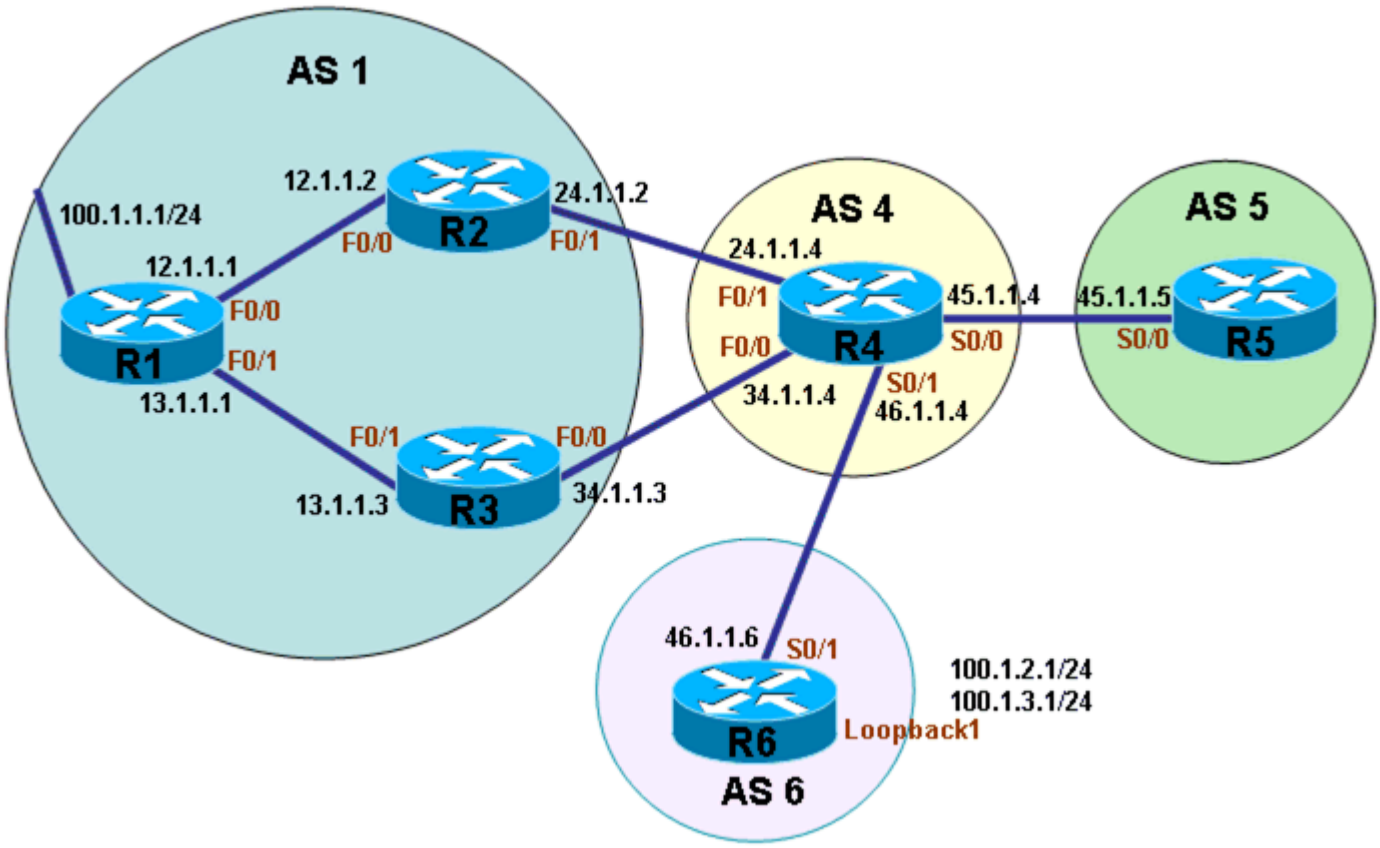
扩展 Community 为 MPLS 中的 VRF 路由传递所定义的，在 MPLS 中有所解释。

注： BGP Community 必须明确要求传递，否则邻居收不到相应 Community。

要匹配携带 Community 的路由，方法为使用 Community List，并且有数字 list 和命名 list 两种，每两条语句之间相隔 10，以 10 递增，一组数字 list 最多支持 100 条语句，而命名 list 则不受此限制，但并不是所有 IOS 都支持命名 list。在使用 Community List 匹配指定路由条目后，则可使用 route-map 来调用 Community List，从而为指定路由设置相应参数和策略，最终应用该 route-map。

注： Community 可以传递的距离不受限制，邻居可以再传给其它邻居，Community 可以被多个路由器多次使用。

配置 BGP Community



说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1	Loopback 0	1.1.1.1/32	Loopback 11	11.1.1.1/24
R2	Loopback 0	2.2.2.2/32	Loopback 22	22.2.2.2/24
R3	Loopback 0	3.3.3.3/32	Loopback 33	33.3.3.3/24
R4	Loopback 0	4.4.4.4/32	Loopback 44	44.4.4.4/24
R5	Loopback 0	5.5.5.5/32	Loopback 55	55.5.5.5/24
R6	Loopback 0	6.6.6.6/32	Loopback 66	66.6.6.6/24

鸿鹄培训+考试群：301283867

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

1. IGP 保证全网 Loopback 0 互通

说明：此步略，请参见之前配置。

2. 基础 BGP 配置

说明：此步略，请参见之前配置。

3. 测试 no-export

说明：在 R5 上将路由 55.5.5.0/24 设置 no-export 后传递给 R4，由于带 no-export 的路由不能传递给任何 eBGP 邻居，所以 R6 将无法收到 R4 转发的 55.5.5.0/24。

（1）查看 R4 的 BGP 路由

```
r4#sh ip bgp
BGP table version is 11, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  11.1.1.0/24      3.3.3.3                                  0 1 i
*>                  2.2.2.2                                  0 1 i
*> 22.2.2.0/24      2.2.2.2                  0               0 1 i
```

```
*> 33.3.3.0/24      3.3.3.3                  0             0 1 i

*> 44.4.4.0/24      0.0.0.0                  0         32768 i

*> 55.5.5.0/24      5.5.5.5                  0             0 5 i

*> 66.6.6.0/24      6.6.6.6                  0             0 6 i

*  100.1.1.0/24     3.3.3.3                                0 1 i

*>                  2.2.2.2                                0 1 i

*> 100.1.2.0/24     6.6.6.6                  0             0 6 i

*> 100.1.3.0/24     6.6.6.6                  0             0 6 i

r4#
```

说明：R4 拥有全部的路由，其中包括 R5 发来的所有路由。

（2）查看 R6 的 BGP 路由

```
r6#sh ip bgp

BGP table version is 156, local router ID is 6.6.6.6

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      4.4.4.4                                0 4 1 i

*> 22.2.2.0/24      4.4.4.4                                0 4 1 i

*> 33.3.3.0/24      4.4.4.4                                0 4 1 i
```

```
*> 44.4.4.0/24      4.4.4.4                  0             0 4 i
*> 55.5.5.0/24      4.4.4.4                                0 4 5 i
*> 66.6.6.0/24      0.0.0.0                  0         32768 i
*> 100.1.1.0/24     4.4.4.4                                0 4 1 i
*> 100.1.2.0/24     0.0.0.0                  0         32768 i
*> 100.1.3.0/24     0.0.0.0                  0         32768 i
r6#
```

说明： R4 已将自己的全部路由发给 R6。

（3）R5 将自己的路由 55.5.5.0/24 设置 no-export 后发给 R4

```
r5(config)#access-list 55 permit 55.5.5.0

r5(config)#route-map noe permit 10
r5(config-route-map)#mat ip address 55
r5(config-route-map)#set community no-export
r5(config-route-map)#exit
r5(config)#route-map noe permit 20
r5(config-route-map)#exit

r5(config)#router bgp 5
r5(config-router)#neighbor 4.4.4.4 route-map noe out
r5(config-router)#neighbor 4.4.4.4 send-community
```

说明：R5 已将自己的路由 55.5.5.0/24 设置 no-export，并且指示将 community 传给 R4。

（4）查看 R4 收到的带 community 的路由

```
r4#sh ip bgp community

BGP table version is 12, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 55.5.5.0/24      5.5.5.5                  0             0 5 i

r4#
```

说明：R4 已经收到 R5 发来的带 community 的路由。

（5）查看 R4 收到的带 community 的路由

```
r4#sh ip bgp 55.5.5.0

BGP routing table entry for 55.5.5.0/24, version 12

Paths: (1 available, best #1, table Default-IP-Routing-Table, not
advertised to EBGP peer)

Flag: 0x880

  Not advertised to any peer

  5
```

5.5.5.5 (metric 65) from 5.5.5.5 (5.5.5.5)

Origin IGP, metric 0, localpref 100, valid, external, best

Community: no-export

r4#

说明：R4 收到的路由 55.5.5.0/24 的 community 为 no-export，也就是不将该路由传给任何 eBGP 邻居。

（6）查看 R4 的 eBGP 邻居 R6 的路由情况

r6#sh ip bgp

BGP table version is 157, local router ID is 6.6.6.6

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

r RIB-failure, S Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	11.1.1.0/24	4.4.4.4				0 4 1 i
*>	22.2.2.0/24	4.4.4.4				0 4 1 i
*>	33.3.3.0/24	4.4.4.4				0 4 1 i
*>	44.4.4.0/24	4.4.4.4		0		0 4 i
*>	66.6.6.0/24	0.0.0.0		0	32768	i
*>	100.1.1.0/24	4.4.4.4				0 4 1 i
*>	100.1.2.0/24	0.0.0.0		0	32768	i

*> 100.1.3.0/24 0.0.0.0 0 32768 i

r6#

说明：因为 R4 的路由 55.5.5.0/24 的 community 为 no-export，也就是不将该路由传给任何 eBGP 邻居，所以 R6 没有收到 55.5.5.0/24。

4.测试 no-advertise

说明：在 R4 上将路由 44.4.4.0/24 设置 no-advertise 后传递给 R2，由于带 no-advertise 的路由不能传递给任何 BGP 邻居，所以 R1 将无法收到 R2 转发的 44.4.4.0/24。

（1）查看 R2 的 BGP 路由

r2#sh ip bgp

BGP table version is 124, local router ID is 2.2.2.2

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

r RIB-failure, S Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>i	11.1.1.0/24	1.1.1.1	0	100	0	i
*>	22.2.2.0/24	0.0.0.0	0		32768	i
*>	44.4.4.0/24	4.4.4.4	0		0	4 i
*>	66.6.6.0/24	4.4.4.4			0	4 6 i
*>i	100.1.1.0/24	1.1.1.1	0	100	0	i

*> 100.1.2.0/24 4.4.4.4 0 4 6 i

*> 100.1.3.0/24 4.4.4.4 0 4 6 i

r2#

说明：R2 拥有全部的路由，其中包括 R4 发来的所有路由。

（2）查看 R1 的 BGP 路由

r1#sh ip bgp

BGP table version is 186, local router ID is 1.1.1.1

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

r RIB-failure, S Stale

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

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 11.1.1.0/24	0.0.0.0	0		32768	i
*>i22.2.2.0/24	2.2.2.2	0	100	0	i
*>i33.3.3.0/24	3.3.3.3	0	100	0	i
*>i44.4.4.0/24	2.2.2.2	0	100	0	4 i
* i	3.3.3.3	0	100	0	4 i
*>i66.6.6.0/24	2.2.2.2	0	100	0	4 6 i
* i	3.3.3.3	0	100	0	4 6 i
*> 100.1.1.0/24	0.0.0.0	0		32768	i
* i100.1.2.0/24	3.3.3.3	0	100	0	4 6 i

```
*>i                  2.2.2.2                  0    100      0 4 6 i

*>i100.1.3.0/24      2.2.2.2                  0    100      0 4 6 i

* i                  3.3.3.3                  0    100      0 4 6 i

r1#
```

说明：R2 已将自己的全部路由发给 R1。

（3）R4 将自己的路由 44.4.4.0/24 设置 no-advertise 后发给 R2

```
r4(config)#access-list 44 permit 44.4.4.0


r4(config)#route-map noa permit 10

r4(config-route-map)#match ip address 44

r4(config-route-map)#set community no-advertise

r4(config-route-map)#exit

r4(config)#route-map noa permit 20

r4(config-route-map)#exit


r4(config)#router bgp 4

r4(config-router)#neighbor 2.2.2.2 route-map noa out

r4(config-router)#neighbor 2.2.2.2 send-community
```

说明：R4 已将自己的路由 44.4.4.0/24 设置 no-advertise，并且指示将 community 传给 R2。

（4）查看 R2 收到的带 community 的路由

```
r2#sh ip bgp community

BGP table version is 125, local router ID is 2.2.2.2

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

r2#
```

说明：R2 已经收到 R4 发来的带 community 的路由。

（5）查看 R2 收到的带 community 的路由

```
r2#sh ip bgp 44.4.4.0

BGP routing table entry for 44.4.4.0/24, version 125

Paths: (1 available, best #1, table Default-IP-Routing-Table, not
advertised to any peer)

Flag: 0x880

  Not advertised to any peer

  4

    4.4.4.4 (metric 2) from 4.4.4.4 (4.4.4.4)
```

Origin IGP, metric 0, localpref 100, valid, external, best

Community: no-advertise

r2#

说明：R2 收到的路由 44.4.4.0/24 的 community 为 no-advertise，也就是不将该路由传给任何 BGP 邻居。

（6）查看 R2 的邻居 R1 的路由情况

r1#sh ip bgp

BGP table version is 187, local router ID is 1.1.1.1

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

r RIB-failure, S Stale

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

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 11.1.1.0/24	0.0.0.0	0		32768	i
*>i22.2.2.0/24	2.2.2.2	0	100	0	i
*>i33.3.3.0/24	3.3.3.3	0	100	0	i
*>i44.4.4.0/24	3.3.3.3	0	100	0	4 i
*>i66.6.6.0/24	2.2.2.2	0	100	0	4 6 i
* i	3.3.3.3	0	100	0	4 6 i
*> 100.1.1.0/24	0.0.0.0	0		32768	i
* i100.1.2.0/24	3.3.3.3	0	100	0	4 6 i

```
*>i                  2.2.2.2                  0    100      0 4 6 i

*>i100.1.3.0/24      2.2.2.2                  0    100      0 4 6 i

* i                  3.3.3.3                  0    100      0 4 6 i

r1#
```

说明：因为 R2 的路由 44.4.4.0/24 的 community 为 no-advertise，也就是不将该路由传给任何邻居，所以 R1 没有收到 R2 发来的 44.4.4.0/24。

5.测试普通私有 Community

说明：私有 Community 的功能为路由标签，通过利用 Community 标签来匹配路由。

（1）R4 将 100.0.0.0/8 的全部路由的 Community 设置为 123，并发给 R5

```
r4(config)#access-list 10 permit 100.0.0.0 0.255.255.255


r4(config)#route-map com permit 10

r4(config-route-map)#match ip address 10

r4(config-route-map)#set community 123

r4(config-route-map)#exit

r4(config)#route-map com permit 20

r4(config-route-map)#exit
```

```
r4(config)#router bgp 4

r4(config-router)#neighbor 5.5.5.5 route-map com out

r4(config-router)#neighbor 5.5.5.5 send-community
```

说明：R4 将 100.0.0.0/8 的全部路由的 Community 设置为 123，并发给 R5

（2）查看 R5 收到的带 community 的路由

```
r5#sh ip bgp community

BGP table version is 41, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 100.1.1.0/24     4.4.4.4                                0 4 1 i

*> 100.1.2.0/24     4.4.4.4                                0 4 6 i

*> 100.1.3.0/24     4.4.4.4                                0 4 6 i

r5#
```

说明：R4 收到的带 community 的路由为全部 100.0.0.0/8。

（3）查看 R5 的 community 的路由情况

```
r5#sh ip bgp 100.1.1.0

BGP routing table entry for 100.1.1.0/24, version 39
```

```
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x880
  Not advertised to any peer
  4 1
    4.4.4.4 (metric 65) from 4.4.4.4 (4.4.4.4)
      Origin IGP, localpref 100, valid, external, best
      Community: 123
r5#
```

说明：R4 收到的路由 100.0.0.0/8 的 community 为 123

（4）查看 R5 当前 BGP 路由情况

```
r5#sh ip bgp
BGP table version is 41, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      4.4.4.4                                0 4 1 i
*> 22.2.2.0/24      4.4.4.4                                0 4 1 i
*> 33.3.3.0/24      4.4.4.4                                0 4 1 i
```

```
*> 44.4.4.0/24      4.4.4.4                  0             0 4 i
*> 55.5.5.0/24      0.0.0.0                  0         32768 i
*> 66.6.6.0/24      4.4.4.4                                0 4 6 i
*> 100.1.1.0/24     4.4.4.4                                0 4 1 i
*> 100.1.2.0/24     4.4.4.4                                0 4 6 i
*> 100.1.3.0/24     4.4.4.4                                0 4 6 i
r5#
```

说明：R5 当前 BGP 路由正常，包括 10.0.0.0/8。

（5）R5 将所有 community 为 123 的路由的 weight 值改为 123

```
r5(config)#ip community-list 5 permit 123

r5(config)#route-map wei permit 10
r5(config-route-map)#match community 5
r5(config-route-map)#set weight 123
r5(config-route-map)#exit
r5(config)#route-map wei permit 20
r5(config-route-map)#exit

r5(config)#router bgp 5
r5(config-router)#neighbor 4.4.4.4 route-map wei in
```

说明：R5 上 community-list 匹配 community 为 123 的路由，并将 weight 值改为 123。

（6）查看 R5 修改后的路由情况

```
r5#sh ip bgp

BGP table version is 44, local router ID is 5.5.5.5

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      4.4.4.4                                0 4 1 i

*> 22.2.2.0/24      4.4.4.4                                0 4 1 i

*> 33.3.3.0/24      4.4.4.4                                0 4 1 i

*> 44.4.4.0/24      4.4.4.4                  0             0 4 i

*> 55.5.5.0/24      0.0.0.0                  0         32768 i

*> 66.6.6.0/24      4.4.4.4                                0 4 6 i

*> 100.1.1.0/24     4.4.4.4                              123 4 1 i

*> 100.1.2.0/24     4.4.4.4                              123 4 6 i

*> 100.1.3.0/24     4.4.4.4                              123 4 6 i

r5#
```

说明：community 为 123 的路由 100.0.0.0/8 的 weight 值已被改为 123。

6.测试 AS:NN私有 Community

（1）R4 将 100.0.0.0/8 的全部路由的 Community 设置为 4:123，并发给 R5

```
r4(config)#access-list 10 permit 100.0.0.0 0.255.255.255

r4(config)#route-map com2 permit 10
r4(config-route-map)#match ip address 10
r4(config-route-map)#set community 4:123
r4(config-route-map)#exit
r4(config)#route-map com2 permit 20
r4(config-route-map)#exit

r4(config)#router bgp 4
r4(config-router)#neighbor 5.5.5.5 route-map com2 out
r4(config-router)#neighbor 5.5.5.5 send-community
```

说明：R4 将 100.0.0.0/8 的全部路由的 Community 设置为 4: 123，并发给 R5，

（2）如果双方不做如下配置，则 Community 格式不是 AS:NN：

```
r4(config)#ip bgp-community new-format
r5(config)#ip bgp-community new-format
```

```
r5#sh ip bgp 100.1.1.0
BGP routing table entry for 100.1.1.0/24, version 11
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x880
  Not advertised to any peer
  4 1
    4.4.4.4 (metric 65) from 4.4.4.4 (4.4.4.4)
      Origin IGP, localpref 100, valid, external, best
      Community: 262267
r5#
```

说明：不配置 bgp-community new-format，格式不为 AS:NN。

（3）配置 bgp-community new-format 后，格式为 AS:NN：

```
r5#sh ip bgp 100.1.1.0
BGP routing table entry for 100.1.1.0/24, version 11
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  4 1
    4.4.4.4 (metric 65) from 4.4.4.4 (4.4.4.4)
      Origin IGP, localpref 100, valid, external, best
      Community: 4:123
```

r5#

说明：双方配置 bgp-community new-format 后，Community 变为 4:123，即为 AS:NN 格式。

7.测试错误 AS 号码下的 BGP 邻居

说明：当在指定 BGP 邻居时，如果 AS 号码错误，可以通过 AS 欺骗的形式来建立邻居。

（1）在 R5 上配置 BGP

```
r5(config)#router bgp 5
r5(config-router)#bgp router-id 5.5.5.5
r5(config-router)#neighbor 4.4.4.4 remote-as 100
r5(config-router)#neighbor 4.4.4.4 up loopback 0
r5(config-router)#neighbor 4.4.4.4 ebgp-multihop
r5(config-router)#network 55.5.5.0 mask 255.255.255.0
```

说明：邻居 4.4.4.4 应该是 AS 4 的，而 R5 错指为 100。

（2）在 R4 上配置 AS 欺骗

```
r4(config)#router bgp 4
r4(config-router)#neighbor 5.5.5.5 remote-as 5
r4(config-router)#neighbor 5.5.5.5 update-source loopback 0
r4(config-router)#neighbor 5.5.5.5 ebgp-multihop
r4(config-router)#neighbor 5.5.5.5 local-as 100
```

说明：R4 将自己针对邻居 5.5.5.5 的 AS 改为 100，以建立 BGP 邻居。

（3）查看 R5 的 BGP 邻居

```
r5#sh ip bgp summary

BGP router identifier 5.5.5.5, local AS number 5

BGP table version is 10, main routing table version 10

9 network entries using 1161 bytes of memory

9 path entries using 468 bytes of memory

5/4 BGP path/bestpath attribute entries using 620 bytes of memory

3 BGP AS-PATH entries using 72 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 2321 total bytes of memory

BGP activity 9/0 prefixes, 9/0 paths, scan interval 60 secs


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

4.4.4.4         4   100      10       7       10    0    0
00:00:34        8

r5#
```

说明：可以看到，R5 已经与 R4 建立邻居关系，并且认为对方是 AS 100。

（4）查看 R5 的 BGP 路由

```
r5#sh ip bgp
BGP table version is 10, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      4.4.4.4                              0 100 4 1 i
*> 22.2.2.0/24      4.4.4.4                              0 100 4 1 i
*> 33.3.3.0/24      4.4.4.4                              0 100 4 1 i
*> 44.4.4.0/24      4.4.4.4                  0           0 100 4 i
*> 55.5.5.0/24      0.0.0.0                  0       32768 i
*> 66.6.6.0/24      4.4.4.4                              0 100 4 6 i
*> 100.1.1.0/24     4.4.4.4                              0 100 4 1 i
*> 100.1.2.0/24     4.4.4.4                              0 100 4 6 i
*> 100.1.3.0/24     4.4.4.4                              0 100 4 6 i
r5#
```

说明：R4 在将所有路由发给 R5 时，除了原有的所有真正 AS 外，虚假 AS 100 也被添加进了 AS_Path 中，用以欺骗 R5。

（5）查看 R6 的 BGP 路由

```
r6#sh ip bgp
BGP table version is 158, local router ID is 6.6.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      4.4.4.4                              0 4 1 i
*> 22.2.2.0/24      4.4.4.4                              0 4 1 i
*> 33.3.3.0/24      4.4.4.4                              0 4 1 i
*> 44.4.4.0/24      4.4.4.4                  0           0 4 i
*> 55.5.5.0/24      4.4.4.4                              0 4 100 5 i
*> 66.6.6.0/24      0.0.0.0                  0       32768 i
*> 100.1.1.0/24     4.4.4.4                              0 4 1 i
*> 100.1.2.0/24     0.0.0.0                  0       32768 i
*> 100.1.3.0/24     0.0.0.0                  0       32768 i
r6#
```

说明：对于 R4 与 R5 之间的虚假 AS 100，在将路由发给 R6 时，这个 AS 同样存在。

（6）在 AS_Path 中移除虚假 AS 100

```
r4(config)#router bgp 4
```

r4(config-router)#neighbor 5.5.5.5 local-as 100 no-prepend

说明：指定虚假 AS 100 不添加在 AS_Path 中。

（7）再次查看 R5 的 BGP 路由表

r5#sh ip bgp

BGP table version is 26, local router ID is 5.5.5.5

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

r RIB-failure, S Stale

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

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 11.1.1.0/24	4.4.4.4			0	100 4 1 i
*> 22.2.2.0/24	4.4.4.4			0	100 4 1 i
*> 33.3.3.0/24	4.4.4.4			0	100 4 1 i
*> 44.4.4.0/24	4.4.4.4	0		0	100 4 i
*> 55.5.5.0/24	0.0.0.0	0		32768	i
*> 66.6.6.0/24	4.4.4.4			0	100 4 6 i
*> 100.1.1.0/24	4.4.4.4			0	100 4 1 i
*> 100.1.2.0/24	4.4.4.4			0	100 4 6 i
*> 100.1.3.0/24	4.4.4.4			0	100 4 6 i

r5#

说明：对于 R5 来说，虚假的 AS 100 无法去除，因为对方邻居指定为 100，如果 AS_Path 中没有 AS 100，则会有错误。

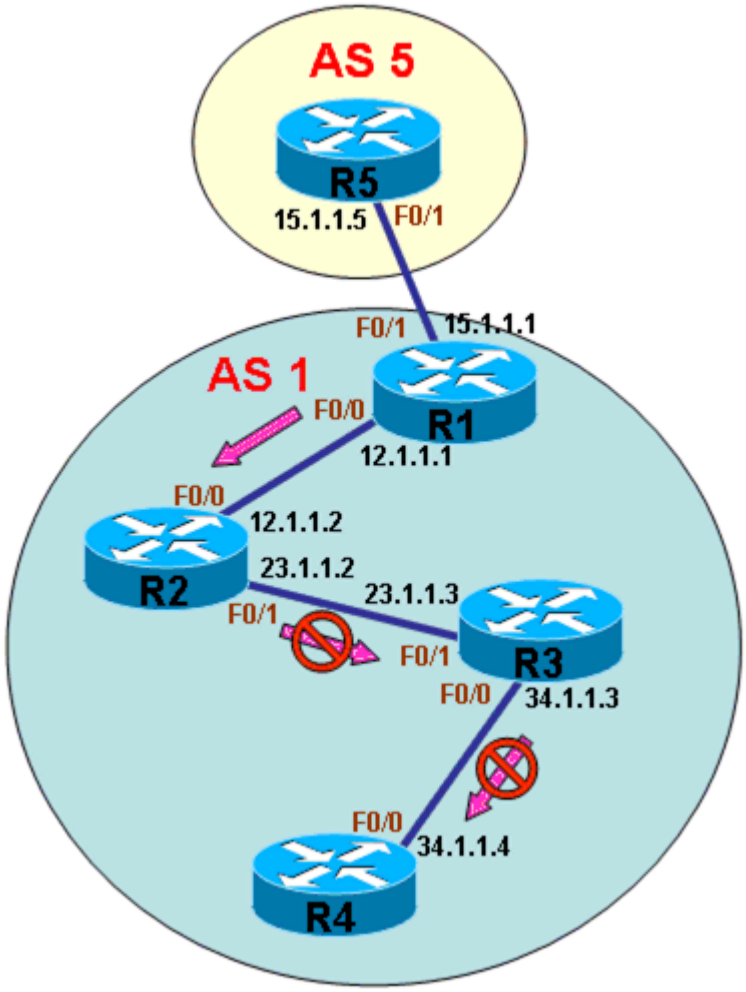
（8）再次查看 R6 的 BGP 路由

```
r6#sh ip bgp
BGP table version is 160, local router ID is 6.6.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      4.4.4.4                                0 4 1 i
*> 22.2.2.0/24      4.4.4.4                                0 4 1 i
*> 33.3.3.0/24      4.4.4.4                                0 4 1 i
*> 44.4.4.0/24      4.4.4.4                  0             0 4 i
*> 55.5.5.0/24      4.4.4.4                                0 4 5 i
*> 66.6.6.0/24      0.0.0.0                  0         32768 i
*> 100.1.1.0/24     4.4.4.4                                0 4 1 i
*> 100.1.2.0/24     0.0.0.0                  0         32768 i
*> 100.1.3.0/24     0.0.0.0                  0         32768 i
r6#
```

说明：在将路由发给与虚假 AS 100 不相关的邻居时，虚假 AS 可以不添加到 AS_Path 中。

BGP Reflector(BGP 反射器)

因为 BGP 在将路由发给 eBGP 邻居时，会将自己的 AS 号码添加到 AS-path 中，所以可以以此来防止环路，而在将路由发给 iBGP 时，是不会往 AS-path 添加 AS 号码的，因此在 iBGP 之间传递路由时，没有防止环路的机制。考虑到为 iBGP 之间的路由传递也加入防环机制，因而强制将 BGP 路由在 AS 内部只传一跳，导致一台 BGP 路由器从 eBGP 邻居收到路由时，发给 iBGP 邻居之后，iBGP 邻居收到就不再传给其它任何 iBGP 邻居了，而只能传递给 eBGP 邻居，最终使得 AS 内部邻居过多时，很难保证每台路由器都能收到所有路由。



如上图环境中，R1 从 eBGP 邻居 R5 收到路由后，可以传递给 iBGP 邻居 R2，而 R2 为了防止环路，不能将从 iBGP 邻居 R1 收到的路由传递给 iBGP 邻居 R3，最后导致 BGP 路由器 R3 和 R4 都不能拥有完整的 BGP 路由表。

为了解决 AS 内，不能将从 iBGP 邻居收到的路由发给 iBGP 邻居的问题，通过创建全互联的邻居关系，将所有 iBGP 邻居都与拥有 eBGP 邻居的路由器建立邻居。这种方法工作繁琐且消耗系统资源，如上图，需要所有路由器都与 R1 建立邻居，从而使 R1 成为单点故障。

除了创建全互联的 BGP 邻居关系外，还可以使用 BGP Reflector(BGP 反射器)的方式来将从 iBGP 邻居学习到的路由传递给其它 iBGP 邻居。

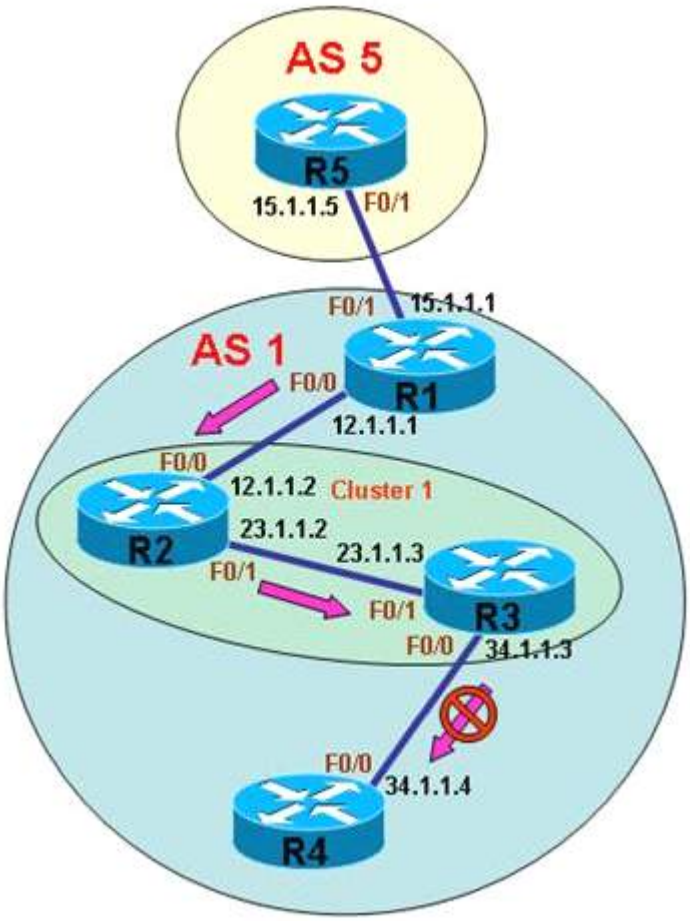
BGP Reflector 可以将自己的任何 BGP 路由反射给自己的 client，从而可以突破 iBGP 路由传递的限制，具体规则为：

★从 eBGP 邻居学习到的路由会发送给所有 client 和所有非 client，也就是发给所有邻居。

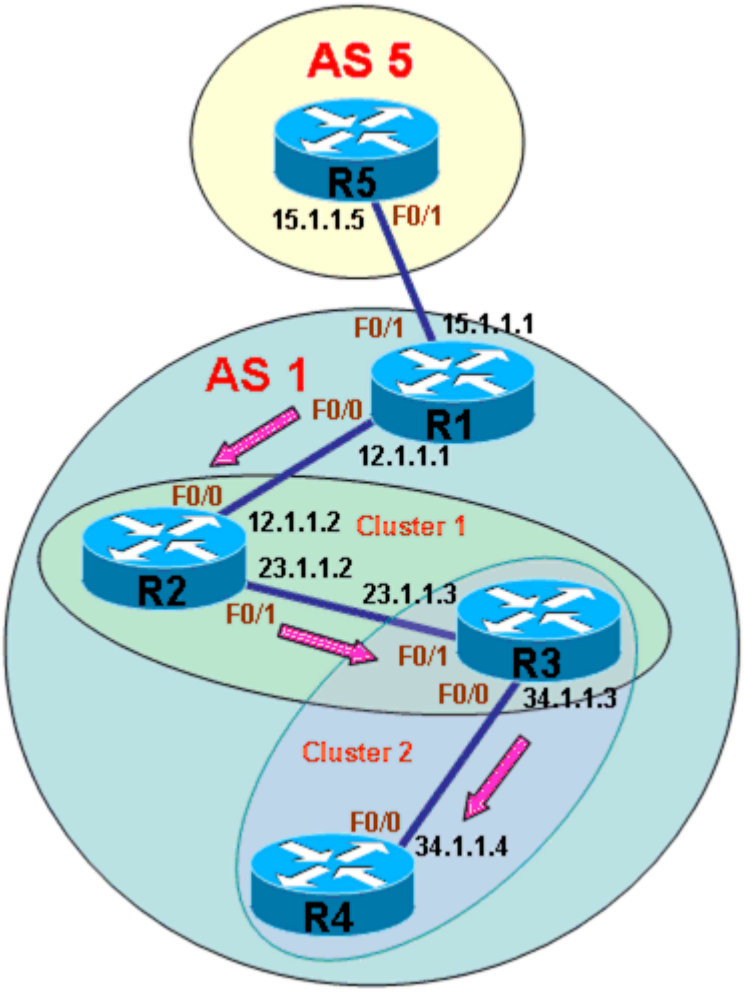
★从非 client 学习到的路由将发送给所有 client。

★从 client 学习到的路由将发送给所有 client 和所有非 client，也就是发给所有邻居。

BGP Reflector 和自己的 Client 称为一个 cluster，如下图：



我们将 R2 配置为 BGP Reflector，将 R3 配置为 client，那么 R2 和 R3 就是一个 cluster。在配置 BGP Reflector 时，只需要在 Reflector 上配置参数，而不需要在 client 做任何配置，所以 client 并不知道自己是 client，因此一个 cluster 的 client，同样还可以配置成另一个 cluster 的 Reflector，类似于嵌套，如下图：

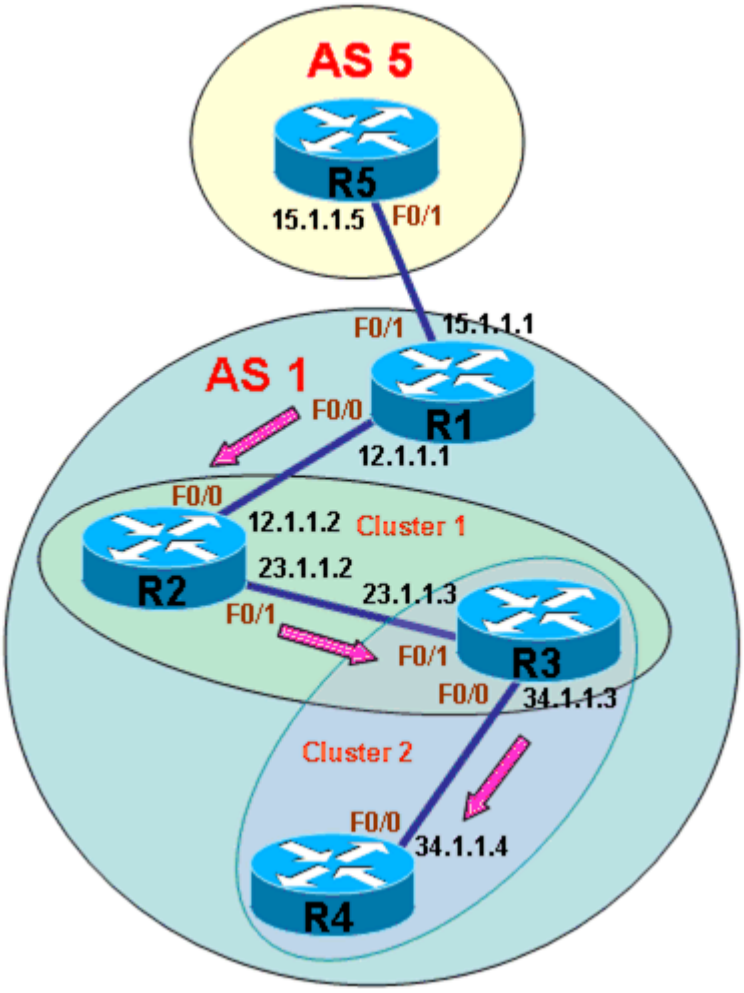


在 Cluster 1 中,R2 为 Reflector,R3 为 client,因为 R3 并不知道自己是 client,所以又将 R3 配置成了 Cluster 2 中的 Reflector，并将 R4 配置成为 Cluster 2 中的 client。最终因为 Reflector R2 收到了 R1 的路由，将所有路由发给 client R3,又因为 R3 也是 Reflector，再将路由发给 client R4，并将所有从 R4 的路由也发送给 R2，最后所有路由器都拥有全网的路由。

一个 AS 中可以有多个 Cluster，所以为了防止环路，引入了类似于 AS_Path 的技术，一个 Cluster 拥有一个唯一的 Cluster ID，这个 Cluster ID 默认就是 Reflector 的 Router-ID，也可以随意设置，并且一个 Cluster 中可以有多个 Reflector 互为备份，所以当一个 Cluster 中有多个 Reflector 时，Cluster ID 必须手工设置。Reflector 在将路由反射出去时，都会写入自己的 Cluster ID，在路由发送到其它 Cluster 后，其它 Reflector 在写入自己的 Cluster ID 时，还会保留之前的 Cluster ID，就像保留 AS_Path 一样，如果收到一条路由带有与自己相同的 Cluster ID，就说明路由发回了原来的 Cluster，则认为环路产生，将接收到的路由丢失，以此来防止环路。

除了 Cluster ID 外，路由还带有 Originator ID，这个 Originator ID 是产生这条路由的路由器的 Router-ID，发回起源路由器，也认为环路产生，则被丢弃。在路由是从别的 AS 发过来时，Originator ID 就是 AS 边界接收的第一台 BGP 路由器。

配置 BGP Reflector



以上图环境为例，配置 BGP Reflector

说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32 Loopback 11 11.1.1.1/24

R2 Loopback 0 2.2.2.2/32 Loopback 22 22.2.2.2/24

R3 Loopback 0 3.3.3.3/32 Loopback 33 33.3.3.3/24

R4 Loopback 0 4.4.4.4/32 Loopback 44 44.4.4.4/24

R5 Loopback 0 5.5.5.5/32 Loopback 55 55.5.5.5/24

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

1. IGP 保证全网 Loopback 0 互通

说明：此步略，请参见之前配置。

2. 全网配置 BGP

（1）配置 R5 的 BGP

```
r5(config)#router bgp 5
r5(config-router)#bgp router-id 5.5.5.5
r5(config-router)#neighbor 1.1.1.1 remote-as 1
r5(config-router)#neighbor 1.1.1.1 update-source loopback 0
r5(config-router)#neighbor 1.1.1.1 ebgp-multihop
r5(config-router)#network 55.5.5.0 mask 255.255.255.0
```

（2）配置 R1 的 BGP

```
r1(config)#router bgp 1
r1(config-router)#bgp router-id 1.1.1.1
r1(config-router)#neighbor 5.5.5.5 remote-as 5
r1(config-router)#neighbor 5.5.5.5 update-source loopback 0
r1(config-router)#neighbor 5.5.5.5 ebgp-multihop
r1(config-router)#neighbor 2.2.2.2 remote-as 1
r1(config-router)#neighbor 2.2.2.2 update-source loopback 0
r1(config-router)#network 11.1.1.0 mask 255.255.255.0
```

（3）配置 R2 的 BGP

```
r2(config)#router bgp 1
r2(config-router)#bgp router-id 2.2.2.2
r2(config-router)#neighbor 1.1.1.1 remote-as 1
r2(config-router)#neighbor 1.1.1.1 update-source loopback 0
r2(config-router)#neighbor 3.3.3.3 remote-as 1
r2(config-router)#neighbor 3.3.3.3 update-source loopback 0
r2(config-router)#network 22.2.2.0 mask 255.255.255.0
```

（4）配置 R3 的 BGP

```
r3(config)#router bgp 1
r3(config-router)#bgp router-id 3.3.3.3
r3(config-router)#neighbor 2.2.2.2 remote-as 1
```

```
r3(config-router)#neighbor 2.2.2.2 update-source loopback 0
r3(config-router)#neighbor 4.4.4.4 remote-as 1
r3(config-router)#neighbor 4.4.4.4 update-source loopback 0
r3(config-router)#network 33.3.3.0 mask 255.255.255.0
```

（5）配置 R4 的 BGP

```
r4(config)#router bgp 1
r4(config-router)#bgp router-id 4.4.4.4
r4(config-router)#neighbor 3.3.3.3 remote-as 1
r4(config-router)#neighbor 3.3.3.3 update-source loopback 0
r4(config-router)#network 44.4.4.0 mask 255.255.255.0
```

3.查看 BGP 路由情况

（1）查看 R5 的 BGP 路由情况

```
r5#sh ip bgp
BGP table version is 4, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
```

```
*> 11.1.1.0/24      1.1.1.1                  0             0 1 i

*> 22.2.2.0/24      1.1.1.1                                0 1 i

*> 55.5.5.0/24      0.0.0.0                  0         32768 i

r5#
```

说明：因为 R2 无法将 R3 和 R4 的路由 33.3.3.0 和 44.4.4.0 发给 R1，所以导致 R5 路由表不完整。

（2）查看 R1 的 BGP 路由情况

```
r1#sh ip bgp

BGP table version is 4, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*>i22.2.2.0/24      2.2.2.2                  0    100       0 i

*> 55.5.5.0/24      5.5.5.5                  0             0 5 i

r1#
```

说明：因为 R2 无法将 R3 和 R4 的路由 33.3.3.0 和 44.4.4.0 发给 R1，所以导致 R1 路由表不完整。

（3）查看 R2 的 BGP 路由情况

```
r2#sh ip bgp

BGP table version is 5, local router ID is 2.2.2.2

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i11.1.1.0/24      1.1.1.1                  0    100      0 i

*> 22.2.2.0/24      0.0.0.0                  0         32768 i

*>i33.3.3.0/24      3.3.3.3                  0    100      0 i

*>i55.5.5.0/24      5.5.5.5                  0    100      0 5 i

r2#
```

说明：因为 R3 无法将 R4 的路由 44.4.4.0 发给 R2，所以导致 R2 路由表不完整。

（4）查看 R3 的 BGP 路由情况

```
r3#sh ip bgp

BGP table version is 4, local router ID is 3.3.3.3

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

              r RIB-failure, S Stale

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

```
   Network          Next Hop            Metric LocPrf Weight Path
*>i22.2.2.0/24      2.2.2.2                  0    100      0 i
*> 33.3.3.0/24      0.0.0.0                  0         32768 i
*>i44.4.4.0/24      4.4.4.4                  0    100      0 i
r3#
```

说明：R3 只能收到 R2 和 R4 的直连路由。

（5）查看 R4 的 BGP 路由情况

```
r4#sh ip bgp
BGP table version is 3, local router ID is 4.4.4.4
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i33.3.3.0/24      3.3.3.3                  0    100      0 i
*> 44.4.4.0/24      0.0.0.0                  0         32768 i
r4#
```

说明：R4 只能收到 R3 的直连路由。

4.配置 BGP Reflector

鸿鹄培训+考试群：301283867

（1）配置 R2 为 BGP Reflector

```
r2(config)#router bgp 1

r2(config-router)#neighbor 3.3.3.3 route-reflector-client
```

说明：配置 R2 为 BGP Reflector，R3 为 Client。

（2）查看 R3 的路由情况

```
r3#sh ip bgp

BGP table version is 8, local router ID is 3.3.3.3

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i11.1.1.0/24      1.1.1.1                  0    100      0 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 i

*> 33.3.3.0/24      0.0.0.0                  0         32768 i

*>i44.4.4.0/24      4.4.4.4                  0    100      0 i

*>i55.5.5.0/24      5.5.5.5                  0    100      0 5 i

r3#
```

说明：R2 将 R1 和 R5 的路由都发来了，R3 的路由表已经完整。

（3）查看 R1 的路由情况

```
r1#sh ip bgp
```

```
BGP table version is 5, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 i

*>i33.3.3.0/24      3.3.3.3                  0    100      0 i

*> 55.5.5.0/24      5.5.5.5                  0             0 5 i

r1#
```

说明：因为从 client 学习到的路由将发送给所有 client 和所有非 client，所以 R1 收到了 R2 发来的所有路由，但不包括 R4 的路由。

（4）查看 R2 的路由情况

```
r2#sh ip bgp

BGP table version is 7, local router ID is 2.2.2.2

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path
```

```
*>i11.1.1.0/24      1.1.1.1                  0    100      0 i

*> 22.2.2.0/24      0.0.0.0                  0         32768 i

*>i33.3.3.0/24      3.3.3.3                  0    100      0 i

*>i55.5.5.0/24      5.5.5.5                  0    100      0 5 i

r2#
```

说明：因为 R3 不能将 iBGP R4 的路由发给任何 iBGP 邻居，所以 R2 也没有 R4 的路由。

（5）在 R3 上查看 R4 发来的路由 44.4.4.0

```
r3#sh ip bgp 44.4.4.0

BGP routing table entry for 44.4.4.0/24, version 4

Paths: (1 available, best #1, table Default-IP-Routing-Table)

  Not advertised to any peer

  Local

    4.4.4.4 (metric 2) from 4.4.4.4 (4.4.4.4)

      Origin IGP, metric 0, localpref 100, valid, internal, best

r3#
```

说明：R4 发给 R3 的路由为正常路由，所以没有不同。

（6）在 R3 上查看 R2 发来的路由 55.5.5.0

```
r3#sh ip bgp 55.5.5.0

BGP routing table entry for 55.5.5.0/24, version 8
```

```
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  5
    5.5.5.5 (metric 4) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 1.1.1.1, Cluster list: 2.2.2.2
r3#
```

说明：因为 R2 是 Reflector,所以发出的路由带有 Cluster ID，即为自己的 Router-ID 2.2.2.2，还有个 Originator ID，就是 AS 边界第一台路由器 R1 的 Router-ID。

（7）改变 Reflector 路由器 R2 对 Client 的下一跳属性

```
r2(config)#router bgp 1
r2(config-router)#neighbor 3.3.3.3 next-hop-self
```

（8）再次查看 R3 的路由情况

```
r3#sh ip bgp
BGP table version is 8, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
```

```
*>i11.1.1.0/24      1.1.1.1                  0    100      0 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 i

*> 33.3.3.0/24      0.0.0.0                  0         32768 i

*>i44.4.4.4/24      4.4.4.4                  0    100      0 i

*>i55.5.5.5/24      5.5.5.5                  0    100      0 5 i

r3#
```

说明：可以看到，从 R2 反射过来的 R5 的路由 55.5.5.0 的下一跳还是 R5，所以在 Reflector 环境中，下一跳属性是不能被改变的。

5.配置 Cluster 2

(1)查看 R4 的路由情况

```
r4#sh ip bgp

BGP table version is 3, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i33.3.3.0/24      3.3.3.3                  0    100      0 i

*> 44.4.4.4/24      0.0.0.0                  0         32768 i

r4#
```

说明：由于 R3 不能将任何从 iBGP 接收到的路由发给 R4，导致 R4 只有 R3 和 R4 两台路由器的路由。

（2）将 R3 配置为另一个 Cluster 的 Reflector

```
r3(config)#router bgp 1

r3(config-router)#neighbor 4.4.4.4 route-reflector-client
```

说明：将 R3 配置为 Reflector，R4 配置为 Client。

（3）查看 R4 的路由情况

```
r4#sh ip bgp

BGP table version is 8, local router ID is 4.4.4.4

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i11.1.1.0/24      1.1.1.1                  0    100      0 i

*>i22.2.2.0/24      2.2.2.2                  0    100      0 i

*>i33.3.3.0/24      3.3.3.3                  0    100      0 i

*> 44.4.4.0/24      0.0.0.0                  0         32768 i

*>i55.5.5.0/24      5.5.5.5                  0    100      0 5 i

r4#
```

说明：R3 将所有路由反射给 client R4，最后 R4 拥有了全部的路由。

（4）查看 R4 从 Reflector 反射过来的路由信息

```
r4#sh ip bgp 55.5.5.0
BGP routing table entry for 55.5.5.0/24, version 8
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  5
    5.5.5.5 (metric 5) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 1.1.1.1, Cluster list: 3.3.3.3, 2.2.2.2
r4#
```

说明：路由从一个 cluster 发到另外一个 cluster，cluster ID 会像 AS_Path 一样被保留累积。

（5）在 R3 上修改 cluster ID

```
r3(config)#router bgp 1
r3(config-router)#bgp cluster-id 123
```

说明：R3 上修改 cluster ID 为 0.0.0.123。

（6）查看 R4 从 Reflector 反射过来的路由信息

```
r4#sh ip bgp 55.5.5.0
BGP routing table entry for 55.5.5.0/24, version 11
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x800
  Not advertised to any peer
  5
    5.5.5.5 (metric 5) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 1.1.1.1, Cluster list: 0.0.0.123, 2.2.2.2
r4#
```

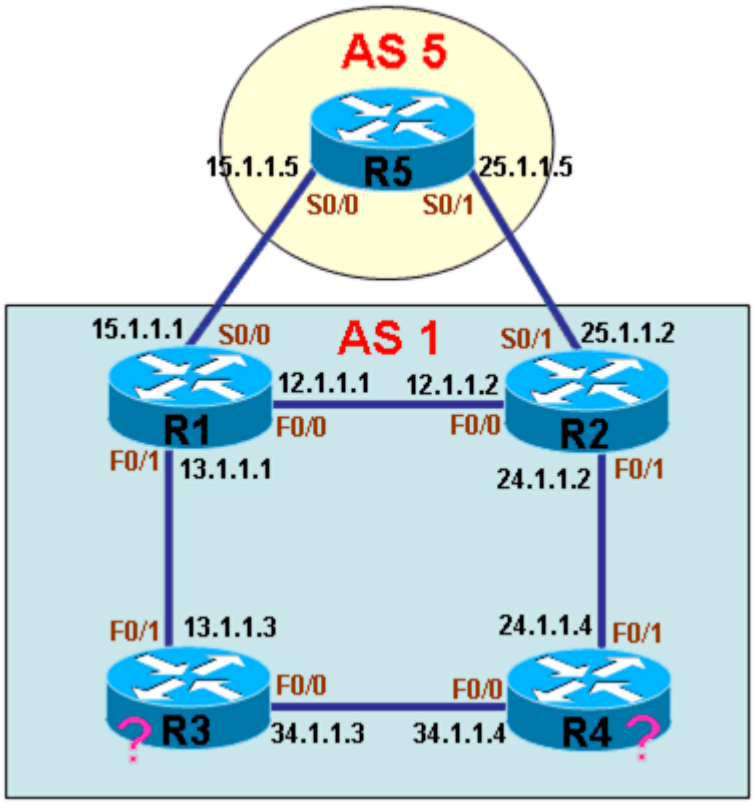
说明：R3 已经将 cluster ID 改为 0.0.0.123。

BGP Confederation(BGP 联邦)

为了解决由于从 iBGP 邻居收到的路由不能转发给其它 iBGP 邻居的限制问题，除了可以使用在 iBGP 邻居之间创建全互联的邻居关系和使用 BGP Reflector 之外，还可以使用 BGP Confederation(BGP 联邦)。

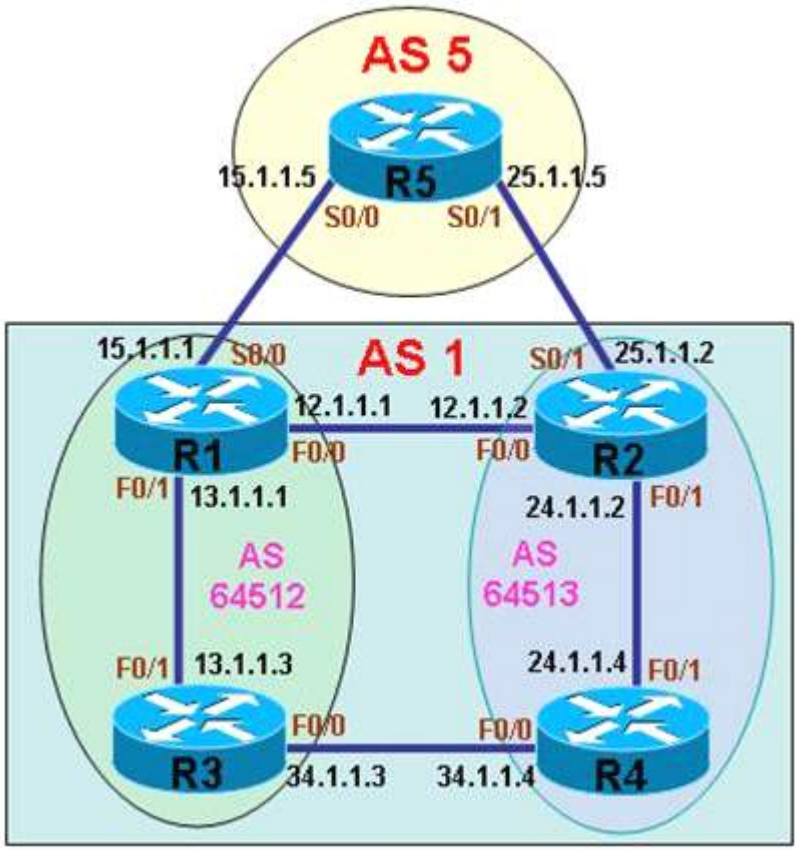
因为只有从 iBGP 邻居收到的路由才不能转发给其它 iBGP 邻居，而从 eBGP 邻居收到的路由可以转发给任何邻居，包括 iBGP 邻居，所以在拥有多个路由器的大型 AS 中，BGP Confederation 采用在 AS 内部建立多个子 AS 的方法，从而将一个大的 AS 分割成多个小型 AS，让 AS 内部拥有足够数量的 eBGP 邻居关系来解决路由限制问题。

如下图



在上图中，当 R3 从 iBGP 邻居 R1 收到路由后，不能再转发给 iBGP 邻居 R4，而 R2 从 eBGP 邻居 R5 收到 R1 的路由后，因为拥有自己的 AS 号码，最后将路由丢弃而不转发给 R4，最终造成 R4 拥有不完整的路由表，同样 R3 也像 R4 一样不能拥有完整的路由表。

对于上述问题，可以创建全互联的 BGP 邻居关系，或者在 R3 和 R4 上配置 BGP Reflector 的方法来解决。除此之外，还可以使用在 AS 内部创建 BGP Confederation 的方法来解决，如下：



在上图环境中,通过 BGP Confederation 的方式在 R1 与 R3 之间创建子 AS 64512,而在 R2 与 R4 之间创建子 AS 64513，这样一来，在 R1 将全部路由发给 R3，以及 R2 将全部路由发给 R4 之后，

因为 R3 与 R4 是 eBGP 邻居的关系，所以 R3 与 R4 之间可以任意转发 BGP 路由信息，从而使双方都拥有完整的全网路由表。

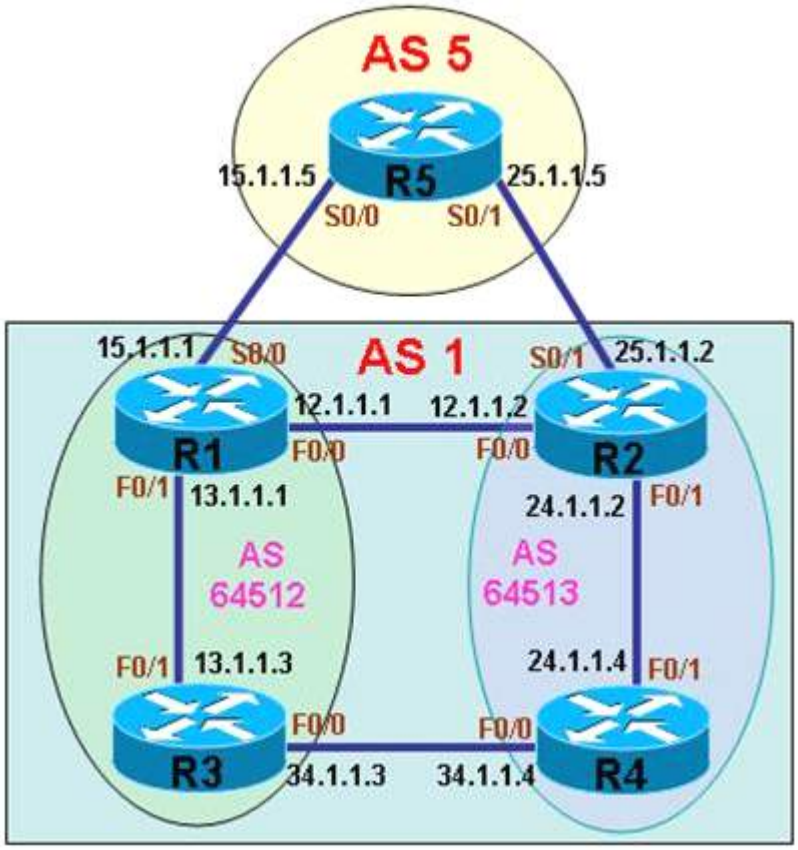
在使用 BGP Confederation 在 AS 内部创建子 AS 时，建议使用私有 AS 号码，范围是 64512-65534，所有 BGP Confederation 内部的子 AS，对于外界都是不可见的,如上图中，R1 与 R2 在 AS 1 中分别为 AS 64512 和 AS 64513，但是对于 R5 来说，R1 和 R2 都为 AS 1 的，而 AS 64512 和 AS 64513 对于 R5 来说是透明的，外界并不知道 AS 内部是否创建了 BGP Confederation，对于子 AS 的号码只在 AS 内部传递路由时才会添加到 AS_Path 中去，在出 AS 时，这些子 AS 号码是不会写入 AS_Path 的。

注：

★在路径属性中，联邦内部的子 AS 是不被 AS_Path 计算在内的。

★在选路规则中，比较 eBGP 与 iBGP 邻居类型时，AS 内部的子 AS 之间是不作 eBGP 与 iBGP 邻居类型比较的。

配置 BGP Confederation



说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32 Loopback 11 11.1.1.1/24

R2 Loopback 0 2.2.2.2/32 Loopback 22 22.2.2.2/24

R3 Loopback 0 3.3.3.3/32 Loopback 33 33.3.3.3/24

R4 Loopback 0 4.4.4.4/32 Loopback 44 44.4.4.4/24

R5 Loopback 0 5.5.5.5/32 Loopback 55 55.5.5.5/24

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

1. IGP 保证全网 Loopback 0 互通

（1）配置 OSPF

说明：此步略，请参见之前配置。

（2）测试全网 Loopback 0 连通性

```
r5#ping 1.1.1.1 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 5.5.5.5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/114/228
ms
r5#ping 2.2.2.2 source loopback 0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 5.5.5.5
!!!!!
```

```
    Success rate is 100 percent (5/5), round-trip min/avg/max = 56/84/128
ms

    r5#

    r5#

    r5#ping 3.3.3.3 source loopback 0


    Type escape sequence to abort.

    Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:

    Packet sent with a source address of 5.5.5.5

    !!!!!

    Success rate is 100 percent (5/5), round-trip min/avg/max = 16/106/180
ms

    r5#ping 4.4.4.4 source loopback 0


    Type escape sequence to abort.

    Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:

    Packet sent with a source address of 5.5.5.5

    !!!!!

    Success rate is 100 percent (5/5), round-trip min/avg/max = 48/124/224
ms

    r5#
```

说明：全网 Loopback 0 连通性连通性正常。

2.配置 BGP Confederation

（1）在 R5 上配置 BGP

```
r5(config)#router bgp 5

r5(config-router)#bgp router-id 5.5.5.5

r5(config-router)#neighbor 1.1.1.1 remote-as 1

r5(config-router)#neighbor 1.1.1.1 update-source loopback 0

r5(config-router)#neighbor 1.1.1.1 ebgp-multihop

r5(config-router)#neighbor 2.2.2.2 remote-as 1

r5(config-router)#neighbor 2.2.2.2 update-source loopback 0

r5(config-router)#neighbor 2.2.2.2 ebgp-multihop

r5(config-router)#network 55.5.5.0 mask 255.255.255.0
```

说明： R5 的配置常规不变。

（2）在 R1 上配置 BGP Confederation

```
r1(config)#router bgp 64512

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

r1(config-router)#bgp confederation identifier 1

r1(config-router)#bgp confederation peers 64513

r1(config-router)#neighbor 5.5.5.5 remote-as 5

r1(config-router)#neighbor 5.5.5.5 update-source loopback 0

r1(config-router)#neighbor 5.5.5.5 ebgp-multihop

r1(config-router)#neighbor 2.2.2.2 remote-as 64513
```

```
r1(config-router)#neighbor 2.2.2.2 update-source loopback 0

r1(config-router)#neighbor 2.2.2.2 ebgp-multihop

r1(config-router)#neighbor 3.3.3.3 remote-as 64512

r1(config-router)#neighbor 3.3.3.3 update-source loopback 0

r1(config-router)#network 11.1.1.0 mask 255.255.255.0
```

说明：指定子 AS 为 64512，而真正的 AS 为 1，并指明与 AS 64513 同属一个 AS，在联邦内部与 R2 为 eBGP 关系，与 R3 为 iBGP 关系。

（3）在 R2 上配置 BGP Confederation

```
r2(config)#router bgp 64513

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

r2(config-router)#bgp confederation identifier 1

r2(config-router)#bgp confederation peers 64512

r2(config-router)#neighbor 5.5.5.5 remote-as 5

r2(config-router)#neighbor 5.5.5.5 update-source loopback 0

r2(config-router)#neighbor 5.5.5.5 ebgp-multihop

r2(config-router)#neighbor 1.1.1.1 remote-as 64512

r2(config-router)#neighbor 1.1.1.1 update-source loopback 0

r2(config-router)#neighbor 1.1.1.1 ebgp-multihop

r2(config-router)#neighbor 4.4.4.4 remote-as 64513

r2(config-router)#neighbor 4.4.4.4 update-source loopback 0

r2(config-router)#network 22.2.2.0 mask 255.255.255.0
```

说明：指定子 AS 为 64513，而真正的 AS 为 1，并指明与 AS 64512 同属一个 AS，在联邦内部与 R1 为 eBGP 关系，与 R4 为 iBGP 关系。

（4）在 R3 上配置 BGP Confederation

```
r3(config)#router bgp 64512
r3(config-router)#bgp router-id 3.3.3.3
r3(config-router)#bgp confederation identifier 1
r3(config-router)#bgp confederation peers 64513
r3(config-router)#neighbor 1.1.1.1 remote-as 64512
r3(config-router)#neighbor 1.1.1.1 update-source loopback 0
r3(config-router)#neighbor 4.4.4.4 remote-as 64513
r3(config-router)#neighbor 4.4.4.4 update-source loopback 0
r3(config-router)#neighbor 4.4.4.4 ebgp-multihop
r3(config-router)#network 33.3.3.0 mask 255.255.255.0
```

说明：指定子 AS 为 64512，而真正的 AS 为 1，并指明与 AS 64513 同属一个 AS，在联邦内部与 R4 为 eBGP 关系，与 R1 为 iBGP 关系。

（5）在 R4 上配置 BGP Confederation

```
r4(config)#router bgp 64513
r4(config-router)#bgp router-id 4.4.4.4
r4(config-router)#bgp confederation identifier 1
r4(config-router)#bgp confederation peers 64512
r4(config-router)#neighbor 2.2.2.2 remote-as 64513
r4(config-router)#neighbor 2.2.2.2 update-source loopback 0
r4(config-router)#neighbor 3.3.3.3 remote-as 64512
```

```
r4(config-router)#neighbor 3.3.3.3 update-source loopback 0

r4(config-router)#neighbor 3.3.3.3 ebgp-multihop

r4(config-router)#network 44.4.4.0 mask 255.255.255.0
```

说明：指定子 AS 为 64513，而真正的 AS 为 1，并指明与 AS 64512 同属一个 AS，在联邦内部与 R3 为 eBGP 关系，与 R1 为 iBGP 关系。

3.查看 BGP 邻居关系

（1）查看 R5 的 BGP 邻居状况

```
r5#sh ip bgp summary

BGP router identifier 5.5.5.5, local AS number 5

BGP table version is 6, main routing table version 6

5 network entries using 645 bytes of memory

9 path entries using 468 bytes of memory

4/3 BGP path/bestpath attribute entries using 496 bytes of memory

1 BGP AS-PATH entries using 24 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 1633 total bytes of memory

BGP activity 5/0 prefixes, 9/0 paths, scan interval 60 secs


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

1.1.1.1         4     1      25      26        6    0    0
00:17:30        4
```

```
   2.2.2.2         4      1       15       16         6    0    0
00:07:08        4

   r5#
```

说明：R1 与 R2 在联邦内部虽然为 AS 64512 和 AS 64513，但对于 R5 来说，它们都为 AS 1，子 AS 则透明不可见。

（2）查看 R1 的 BGP 邻居状况

```
   r1#sh ip bgp summary

   BGP router identifier 1.1.1.1, local AS number 64512

   BGP table version is 6, main routing table version 6

   5 network entries using 645 bytes of memory

   6 path entries using 312 bytes of memory

   6/4 BGP path/bestpath attribute entries using 744 bytes of memory

   3 BGP AS-PATH entries using 72 bytes of memory

   0 BGP route-map cache entries using 0 bytes of memory

   0 BGP filter-list cache entries using 0 bytes of memory

   BGP using 1773 total bytes of memory

   BGP activity 5/0 prefixes, 6/0 paths, scan interval 60 secs


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

   2.2.2.2         4 64513      13       13         6    0    0
00:06:27        3

   3.3.3.3         4 64512       8       11         6    0    0
00:03:14        1
```

5.5.5.5 4 5 26 25 6 0 0
00:17:51 1

r1#

说明：子 AS 在联邦内部 AS 之间可见。

注：其它 BGP 邻居关系类同，不再一一查看。

4.查看 BGP 路由情况

（1）查看 R5 的 BGP 路由情况

r5#sh ip bgp

BGP table version is 6, local router ID is 5.5.5.5

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

r RIB-failure, S Stale

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

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	11.1.1.0/24	2.2.2.2			0	1 i
*>		1.1.1.1	0		0	1 i
*	22.2.2.0/24	1.1.1.1			0	1 i
*>		2.2.2.2	0		0	1 i
*	33.3.3.0/24	2.2.2.2			0	1 i

```
*>                  1.1.1.1                                0 1 i

*  44.4.4.0/24      1.1.1.1                                0 1 i

*>                  2.2.2.2                                0 1 i

*> 55.5.5.0/24      0.0.0.0                  0         32768 i

r5#
```

说明：R5 已经收到全部的路由，说明对方 AS 内部正常稳定运行。

（2）查看 R1 的 BGP 路由情况

```
r1#sh ip bgp

BGP table version is 6, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*> 22.2.2.0/24      2.2.2.2                  0     100       0 (64513) i

*>i33.3.3.0/24      3.3.3.3                  0     100       0 i

*> 44.4.4.0/24      4.4.4.4                  0     100       0 (64513) i

*  55.5.5.0/24      5.5.5.5                  0     100       0 (64513)
5 i

*>                  5.5.5.5                  0               0 5 i

r1#
```

说明：R1 也收到全部路由，说明 BGP 邻居正常运行，且路由收发和预期相同。需要注意，虽然在 AS 内部，R1 与 R2 为 eBGP 邻居关系，但下一跳属性并没有作修改，所以需要手工修改。

（3）修改 R2 对 R1 的下一跳属性

```
r2(config)#router bgp 64513

r2(config-router)#neighbor 1.1.1.1 next-hop-self
```

（4）再次查看 R1 的 BGP 路由情况

```
r1#sh ip bgp

BGP table version is 7, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 11.1.1.0/24      0.0.0.0                  0         32768 i

*> 22.2.2.0/24      2.2.2.2                  0    100      0 (64513) i

*>i33.3.3.0/24      3.3.3.3                  0    100      0 i

* i44.4.4.0/24      4.4.4.4                  0    100      0 (64513) i

*>                  2.2.2.2                  0    100      0 (64513)
i

*  55.5.5.0/24      2.2.2.2                  0    100      0 (64513)
5 i

*>                  5.5.5.5                  0             0 5 i
```

```
r1#
```

说明：R2 对 R1 的下一跳属性成功修改。

（5）AS 内部都将改变下一跳属性

R1:

```
r1(config)#router bgp 64512
r1(config-router)#neighbor 2.2.2.2 next-hop-self
r1(config-router)#neighbor 3.3.3.3 next-hop-self
```

R3:

```
r3(config)#router bgp 64512
r3(config-router)#neighbor 1.1.1.1 next-hop-self
r3(config-router)#neighbor 4.4.4.4 next-hop-self
```

R4:

```
r4(config)#router bgp 64513
r4(config-router)#neighbor 2.2.2.2 next-hop-self
r4(config-router)#neighbor 3.3.3.3 next-hop-self
```

5.测试 BGP 联邦内部选路

（1）查看 R1 当前的 BGP 路由情况

```
r1#sh ip bgp
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*> 22.2.2.0/24      2.2.2.2                  0    100      0 (64513) i
*>i33.3.3.0/24      3.3.3.3                  0    100      0 i
*> 44.4.4.0/24      2.2.2.2                  0    100      0 (64513) i
*  55.5.5.0/24      2.2.2.2                  0    100      0 (64513)
5 i
*>                  5.5.5.5                  0             0 5 i
r1#
```

说明：R1 到达 R5 的网段 55.5.5.0/24 从 S0/0 出去。

（2）在 R1 上改变去往 55.5.5.0/24 的路径

说明：因为在路径比较中，联邦内部 AS_Path 是不被计算在内的，所以要证明虽然 R1 去往 55.5.5.0/24，从 R2 走的 AS_Path 要长于 R5，但并不是因为 R2 的 AS_Path 比 R5 长，因为子 AS 不被计算，所以只要选路规则的属性中 AS_Path 后面一个属性的变更影响到选路后，就能证明子 AS 是被忽略的，所以在此选择修改 AS_Path 后面的属性，如 MED。

```
r1(config)#access-list 55 permit 55.5.5.0


r1(config)#route-map med permit 10
r1(config-route-map)#match ip address 55
r1(config-route-map)#set metric 55
r1(config-route-map)#exit
r1(config)#route-map med permit 20
r1(config-route-map)#exit


r1(config)#router bgp 64512
r1(config-router)#neighbor 5.5.5.5 route-map med in
```

说明：将走 R5 的 MED 值设置为 55，大于 R2 的 MED 值 0。

（3）再次查看 R1 去往 55.5.5.0/24 的路径

```
r1#sh ip bgp
BGP table version is 8, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete


   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*> 22.2.2.0/24      2.2.2.2                  0     100       0 (64513) i
```

```
   *>i33.3.3.0/24       3.3.3.3                  0    100      0 i

   *> 44.4.4.0/24       2.2.2.2                  0    100      0 (64513) i

   *> 55.5.5.0/24       2.2.2.2                  0    100      0 (64513) 5
i

   *                    5.5.5.5                 55             0 5 i

   r1#
```

说明：R1 去往 55.5.5.0/24 选择从 R2 走，虽然 R1 的 AS_Path 看起来比 R5 长，但因为联邦内部的子 AS 不被计算，所以最终因为 R2 的低 MED 值影响了选路，所以 eBGP 邻居优于 iBGP 邻居的规则在 BGP 联邦内部是被忽略的。

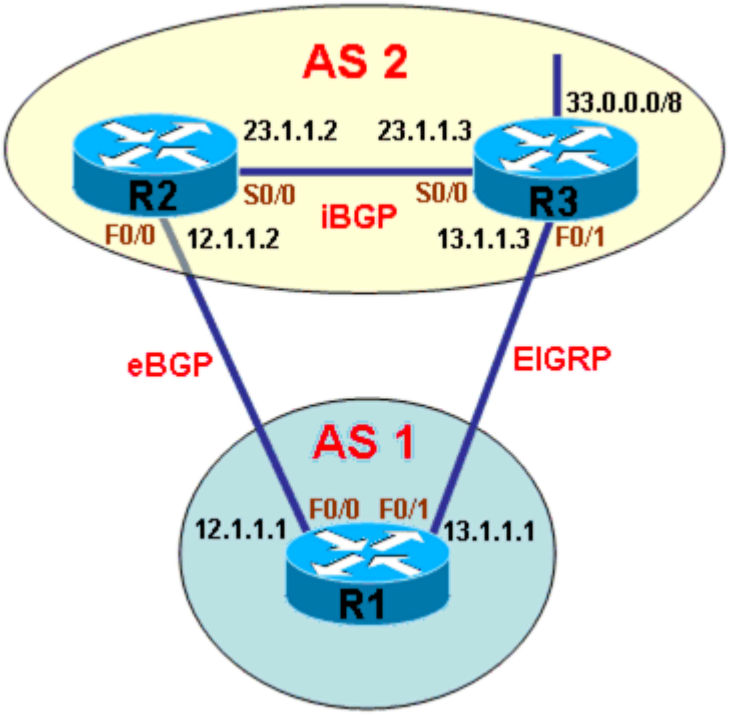
（4）查看 R3 的 BGP 路径

```
   r3#sh ip bgp

   BGP table version is 12, local router ID is 3.3.3.3

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

                 r RIB-failure, S Stale

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


      Network          Next Hop            Metric LocPrf Weight Path

   *>i11.1.1.0/24       1.1.1.1                  0    100      0 i

   *  22.2.2.0/24       4.4.4.4                  0    100      0 (64513)
i

   *>i                  1.1.1.1                  0    100      0 (64513)
i

   *> 33.3.3.0/24       0.0.0.0                  0         32768 i
```

```
   *  44.4.4.0/24      4.4.4.4                0      100      0 (64513)
i

   *>i                 1.1.1.1                0      100      0 (64513)
i

   *  55.5.5.0/24      4.4.4.4                0      100      0 (64513)
5 i

   *>i                 1.1.1.1                0      100      0 (64513)
5 i

   r3#
```

说明：可以看见，R3 去往 55.5.5.0/24，下一跳选择了 iBGP 邻居 R1 而没有选择 eBGP 邻居 R4，所以再一次证实了 eBGP 邻居优于 iBGP 邻居的选路规则在 BGP 联邦内部是被忽略的。

BGP 后门路由



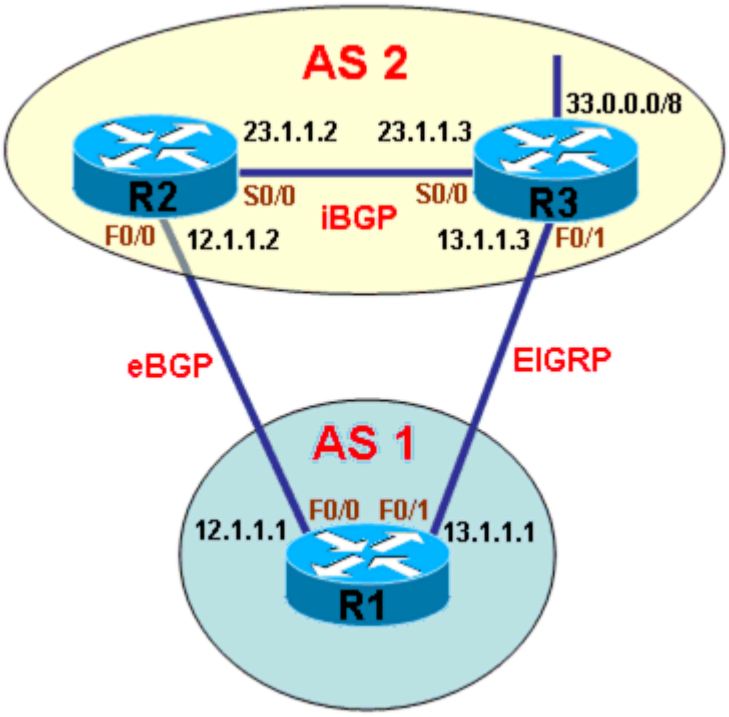
如上图网络环境中，AS 1 与 AS 2 存在多种业务流量，AS 1 中的路由器 R1 与 AS 2 中的路由器 R2 建立 eBGP 邻居关系，当 R2 将 AS 2 内的路由传递给 R1 之后，便可以实现 AS 1 与 AS 2 的互通，R1 将所有去往 AS 2 的流量全部交给 R2。

R1 除了与 R2 建立 eBGP 邻居之外，由于某些特殊业务需求，AS 1 到达 AS 2 中 33.0.0.0/8 网段的流量需要在 R1 和 R3 直连的链路上传输，因此在 R1 与 R3 之间建立了 EIGRP 邻居关系，R3 将 33.0.0.0/8 通过 EIGRP 传给 R1，但是为了冗余性，R3 除了将 33.0.0.0/8 通告在 EIGRP 之外，还通告在了 BGP 中，当 R1 与 R3 的直连链路断掉之后，便可以启用 R1 与 R2 之间的备用链路，最后 R1 通过 R2 去往 33.0.0.0/8。由于 R1 同时从 eBGP 邻居 R2 和 EIGRP 邻居 R3 都收到 33.0.0.0/8 的路由，而 eBGP 的 AD 值为 20，EIGRP 的 AD 值为 90，这样一来，R1 便将所有去往 33.0.0.0/8 的流量都从 R1 与 R2 之间的直连链路上转发，因此不符合要求。

如果要让 R1 直接从与 R3 直连的链路上到达 33.0.0.0/8，就需要将 eBGP 的 AD 值调整到高于 EIGRP，如果改动整个协议的 AD 值，势必会影响全局所有路由的走向，但只要求 33.0.0.0/8 的流量被改动。

基于上述问题，eBGP 有种特殊的路由处理方式，就是将某些路由在 BGP 进程中通过命令 network 导入 BGP 路由表，将其变成本地路由，从而将 AD 值改为 200，大于任何 IGP 协议的 AD 值，但是该路由只对自己本地生效，只影响自己的选路，并不会传递给其它 BGP 邻居。这样的路由处理被称为 BGP backdoor（BGP 后门路由）。可以看出，BGP 后门路由的使用可以使 BGP 中的某些路由不通过 BGP 传递，而优先使用其它后门链路，可以实现对特定流量的路径调整。

配置 BGP 后门路由



说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32

R2 Loopback 0 2.2.2.2/32

R3 Loopback 0 3.3.3.3/32

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

1. IGP 保证全网 Loopback 0 互通

（1）配置 OSPF

说明：此步略，请参见之前配置。

（2）测试全网 Loopback 0 连通性

说明：此步略，请参见之前配置。

2.配置 BGP

（1）配置 R1 与 R2 之间的 BGP

```
r1(config)#router bgp 1

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

r1(config-router)#neighbor 2.2.2.2 remote-as 2

r1(config-router)#neighbor 2.2.2.2 update-source loopback 0

r1(config-router)#neighbor 2.2.2.2 ebgp-multihop
```

说明：在 R1 与 R2 之间建立 eBGP 邻居。

（2）配置 R2 与 R1 和 R3 的 BGP

```
r2(config)#router bgp 2

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

r2(config-router)#neighbor 1.1.1.1 remote-as 1

r2(config-router)#neighbor 1.1.1.1 update-source loopback 0

r2(config-router)#neighbor 1.1.1.1 ebgp-multihop

r2(config-router)#neighbor 3.3.3.3 remote-as 2

r2(config-router)#neighbor 3.3.3.3 update-source loopback 0
```

说明：在 R2 与 R1 之间建立 eBGP 邻居，在 R2 与 R3 之间建立 iBGP 邻居

（3）配置 R3 与 R2 的 BGP

```
r3(config)#router bgp 2

r3(config-router)#bgp router-id 3.3.3.3
```

```
r3(config-router)#neighbor 2.2.2.2 remote-as 2

r3(config-router)#neighbor 2.2.2.2 update-source loopback 0

r3(config-router)#network 33.0.0.0 mask 255.0.0.0
```

说明：在 R3 与 R2 之间建立 iBGP 邻居,并将 33.0.0.0/8 导入 BGP 路由表。

3.查看路由

（1）查看 R2 的 BGP 路由

```
r2#sh ip bgp

BGP table version is 4, local router ID is 2.2.2.2

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*>i33.0.0.0         3.3.3.3                  0    100      0 i

r2#
```

说明：R2 从 iBGP 邻居 R3 收到 33.0.0.0。

（2）查看 R1 的 BGP 路由

```
r1#sh ip bgp

BGP table version is 7, local router ID is 1.1.1.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 33.0.0.0         2.2.2.2                                0 2 i
r1#
```

说明：R1 也从 eBGP 邻居 R2 收到 33.0.0.0。

（3）查看 R1 的 IGP 路由表

```
r1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 12.1.1.2, 00:15:39, FastEthernet0/0
B    33.0.0.0/8 [20/0] via 2.2.2.2, 00:00:13
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/66] via 12.1.1.2, 00:15:39, FastEthernet0/0
     23.0.0.0/24 is subnetted, 1 subnets
O       23.1.1.0 [110/65] via 12.1.1.2, 00:15:39, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, FastEthernet0/0
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, FastEthernet0/1
r1#
```

说明：R1 去往 33.0.0.0 的最终路径是从 R2 走。

（4）跟踪 R1 去往 33.0.0.0 的路径

```
r1#traceroute 33.3.3.3

Type escape sequence to abort.
Tracing the route to 33.3.3.3
```

```
  1 12.1.1.2 16 msec 84 msec 16 msec

  2 23.1.1.3 184 msec *  316 msec

r1#
```

说明：R1 从 R1 与 R2 之间的链路去往 33.0.0.0。

4.配置 EIGRP

（1）配置 R1 与 R3 之间的 EIGRP

R1：

```
r1(config)#router eigrp 100

r1(config-router)#no auto-summary

r1(config-router)#network 13.1.1.1 0.0.0.0
```

R3：

```
r3(config)#router eigrp 100

r3(config-router)#no auto-summary

r3(config-router)#network 13.1.1.3 0.0.0.0

r3(config-router)#network 33.3.3.3 0.0.0.0
```

说明：R1 与 R3 之间建立 EIGRP，R3 将 33.0.0.0 通告进 EIGRP。

（2）查看 R1 从 EIGRP 收到的路由

```
r1#sh ip eigrp topology
```

```
IP-EIGRP Topology Table for AS(100)/ID(1.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 13.1.1.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/1
P 33.0.0.0/8, 0 successors, FD is Inaccessible
        via 13.1.1.3 (156160/128256), FastEthernet0/1
r1#
```

说明：R1 从与 R3 的直连链路收到 33.0.0.0。

5.配置 BGP 后门路由

（1）R1 通过 BGP 后门路由将 33.0.0.0 变为本地路由

```
r1(config)#router bgp 1
r1(config-router)#network 33.3.3.0 backdoor
```

（2）查看 R1 的 BGP 路由表

```
r1#sh ip bgp
BGP table version is 4, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
```

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

   Network          Next Hop            Metric LocPrf Weight Path
r> 33.0.0.0         2.2.2.2                                0 2 i
r1#
```

说明：33.0.0.0 前面标记为 r，说明该路由在 BGP 中为 RIB-failure，所以不会被使用。

（3）查看 R1 的 IGP 路由表

```
r1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
```

```
     2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 12.1.1.2, 00:19:28, FastEthernet0/0
D    33.0.0.0/8 [90/156160] via 13.1.1.3, 00:02:35, FastEthernet0/1
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/66] via 12.1.1.2, 00:19:28, FastEthernet0/0
     23.0.0.0/24 is subnetted, 1 subnets
O       23.1.1.0 [110/65] via 12.1.1.2, 00:19:28, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, FastEthernet0/0
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, FastEthernet0/1
r1#
```

说明：R1 去往 33.0.0.0 的最终路径是直接从 R3 走，符合需求。

（4）跟踪 R1 去往 33.0.0.0 的路径

```
r1#traceroute 33.3.3.3

Type escape sequence to abort.
Tracing the route to 33.3.3.3

  1 13.1.1.3 204 msec *  56 msec
r1#
```

说明：R1 直接从 R3 去往 33.0.0.0。

BGP Dampening

BGP 通常是在大型网络中使用，拥有数量庞大的路由条目。虽然在路由表达到收敛状态后，并不会定期更新路由表，但是只要路由有变动，则会立马通告给邻居，而邻居还会通告给其它邻居，最终变动信息将通告给网络中所有的 BGP 路由器。

如果由于管理员误操作导致路由频繁变动，或者由于物理原因以及软件原因导致路由多次翻动，那么这些变化的路由将在整个网络中不停传播，将带来严重影响，所以为了防止路由翻动而带来的大量路由更新，BGP 采取抑制翻动路由的方法，来将不稳定的路由抑制住，而只有稳定后的路由才会被通告给邻居。

对于衡量什么样的路由算是不稳定的路由，什么样的路由才能传递给邻居，BGP 有一套自己的机制，称为 BGP Dampening，具体过程如下：

对于路由每次翻动，BGP 都会给该路由加上一个惩罚值，并且如果翻动多次，惩罚值都会全部累加，当惩罚值累加到一定程度，也就是累加到最大抑制值，那么该路由就被认为是不稳定的，也就不再发给邻居，但是路由的惩罚值会随着时间而减少，当减到释放值时，该路由又可以重新发给邻居。惩罚值的减少，是和某个时间有关系的，这个时间称为半衰期，每过一个半衰期的时间，惩罚值就减少到原来的一半。虽然路由每翻动一次，都会累加惩罚值，但惩罚值并不是无限累加的，是有一定限制的，这就是最大抑制值，无论路由翻动多少次，累加的惩罚值都不会超过最大抑制值。

以下是各个值的具体参数：

Penalty(惩罚值)

路由每翻动一次加 1000。

Suppress limit(抑制值)

默认为 2000，当某条路由的惩罚值累加到 2000 时，便会被抑制而不发给邻居。

Half-life（半衰期）

默认为 15 分钟，每经过一个半衰期时间，惩罚值减到原来的一半。（每 5 秒会计算一次）

Resume limit（释放值）

默认为 750，当某条被抑制的路由的惩罚值减到释放值时，就可以再次发给邻居。（每 10 秒查看一次）

Maximum suppress limit(最大抑制值)

默认为过 4 个半衰期时间可以减到释放值，即 60 分钟，按释放值 750 计算，那么 4 个半衰期减到 750，原来的值就是 12000。

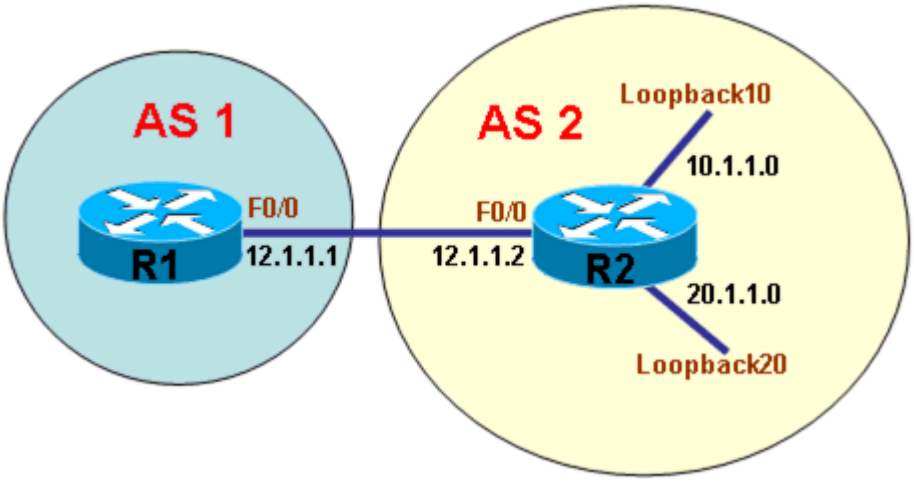
注：

★并不是所有的路由都能配置 BGP Dampening，只有从 eBGP 收到的路由才能配置 BGP Dampening，从 iBGP 收到的路由是不可以的。

★BGP Dampening 可以针对所有 BGP 路由配置，也可以针对特定路由配置，但不可以针对特定邻居配置。

★BGP Dampening 所有值都可自定义，但有范围限制。

配置 BGP Dampening



说明：

上图中所有路由器都配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32

R2 Loopback 0 2.2.2.2/32

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

1. IGP 保证全网 Loopback 0 互通

说明：此步略，请参见之前配置。

2. 配置 BGP

（1）在 R1 上配置 BGP

```
r1(config)#router bgp 1
```

```
r1(config-router)#bgp router-id 1.1.1.1
r1(config-router)#neighbor 2.2.2.2 remote-as 2
r1(config-router)#neighbor 2.2.2.2 update-source loopback 0
r1(config-router)#neighbor 2.2.2.2 ebgp-multihop
```

说明：R1 与 R2 建立 eBGP 邻居关系。

（2）在 R2 上配置 BGP

```
r2(config)#router bgp 2
r2(config-router)#bgp router-id 2.2.2.2
r2(config-router)#neighbor 1.1.1.1 remote-as 1
r2(config-router)#neighbor 1.1.1.1 update-source loopback 0
r2(config-router)#neighbor 1.1.1.1 ebgp-multihop
r2(config-router)#network 10.1.1.0 mask 255.255.255.0
r2(config-router)#network 20.1.1.0 mask 255.255.255.0
```

说明：R2 将 10.1.1.0/24 和 20.1.1.0/24 导入 BGP 路由表中。

（3）查看 R1 的 BGP 路由

```
r1#sh ip bgp
BGP table version is 3, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
```

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

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      2.2.2.2                  0             0 2 i
*> 20.1.1.0/24      2.2.2.2                  0             0 2 i
r1#
```

说明：R1 已经收到 R2 发来的 10.1.1.0/24 和 20.1.1.0/24。

3.针对所有路由配置 BGP Dampening

（1）在 R1 上对所有路由配置 BGP Dampening

```
r1(config)#router bgp 1
r1(config-router)#bgp dampening
```

（2）查看 BGP Dampening 参数

```
r1#sh ip bgp dampening parameters
 dampening 15 750 2000 60 (DEFAULT)
  Half-life time       : 15 mins       Decay Time       : 2320 secs
  Max suppress penalty: 12000          Max suppress time: 60 mins
  Suppress penalty     :  2000         Reuse penalty    : 750
```

说明：以上值为 BGP Dampening 默认值。

（3）查看被 BGP Dampening 监控的路由

```
r1#sh ip bgp dampening dampened-paths


r1#
```

说明：因为没有路由发生翻动，所以路由为空。

（4）测试 bgp dampening

```
r2(config)#int loopback 10
r2(config-if)#shutdown
```

说明：将 10.1.1.0/24 的接口断开，表示翻动一次。

（5）R1 上查看 10.1.1.0/24

```
r1#sh ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24, version 4
Paths: (1 available, no best path)
Flag: 0x820
  Not advertised to any peer
  2 (history entry)
    2.2.2.2 (metric 65) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, external
      Dampinfo: penalty 980, flapped 1 times in 00:00:29
r1#
```

说明：可以看到，由于 10.1.1.0/24 发生了一次翻动，所以有了 penalty 值，为 980。

（6）R1 上再次查看 10.1.1.0/24

```
r1#sh ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24, version 6
Paths: (1 available, no best path)
Flag: 0x820
  Not advertised to any peer
  2 (history entry)
    2.2.2.2 (metric 65) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, external
      Dampinfo: penalty 1903, flapped 2 times in 00:04:25
r1#
```

说明：10.1.1.0/24 翻动了两次，现在 penalty 值为 1903，但还是低于抑制值 2000。

（7）R1 上再次查看 10.1.1.0/24

```
r1#sh ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24, version 8
Paths: (1 available, no best path)
Flag: 0x820
  Not advertised to any peer
```

```
  2 (history entry)

    2.2.2.2 (metric 65) from 2.2.2.2 (2.2.2.2)

      Origin IGP, metric 0, localpref 100, external

      Dampinfo: penalty 2780, flapped 3 times in 00:06:03

r1#
```

说明：翻动 3 次后，penalty 值为 2780，大于抑制值 2000，可以被抑制了。

（8）查看被抑制的路由

```
r1#sh ip bgp dampening dampened-paths

BGP table version is 8, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          From             Reuse    Path

*d 10.1.1.0/24      2.2.2.2          00:04:49 2 i

r1#
```

说明：可以看到 10.1.1.0/24 是被抑制的路由，需要注意，只有该路由重新活动后，才能看见被抑制，否则断开的路由也是不会显示在抑制表中的。

（9）查看 R1 的 BGP 路由表

```
r1#sh ip bgp
```

```
BGP table version is 8, local router ID is 1.1.1.1

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

              r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*d 10.1.1.0/24      2.2.2.2                  0             0 2 i

*> 20.1.1.0/24      2.2.2.2                  0             0 2 i

r1#
```

说明：BGP 路由表中也显示了 10.1.1.0/24 是被抑制的。

4.针对特定路由配置 BGP Dampening

（1）在 R1 上只针对 20.1.1.0 配置 BGP Dampening

```
r1(config)#access-list 20 permit 20.1.1.0

r1(config)#route-map damp permit 10

r1(config-route-map)#match ip address 20

r1(config-route-map)#set dampening 15 800 2100 60

r1(config-route-map)#exit

r1(config)#route-map damp permit 20


r1(config)#router bgp 1
```

```
r1(config-router)#bgp dampening route-map damp
```

说明：只对 20.1.1.0/24 配置 BGP Dampening，并且自定义 Dampening 值。

（2）查看 BGP Dampening 参数

```
r1#sh ip bgp dampening parameters
 dampening 15 800 2100 60 (route-map damp 10)
  Half-life time       : 15 mins       Decay Time       : 2345 secs
  Max suppress penalty: 12800          Max suppress time: 60 mins
  Suppress penalty     :  2100         Reuse penalty    : 800

r1#
```

说明：以上值为自定义的值。

（3）查看 20.1.1.0/24 的情况

```
r1#sh ip bgp 20.1.1.0
BGP routing table entry for 20.1.1.0/24, version 4
Paths: (1 available, no best path)
Flag: 0x820
  Not advertised to any peer
  2 (history entry)
    2.2.2.2 (metric 65) from 2.2.2.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, external
```

```
      Dampinfo: penalty 1000, flapped 1 times in 00:00:04

r1#
```

说明：由于 20.1.1.0/24 翻动了一次，当前 penalty 值为 1000。

（4）再次查看 20.1.1.0/24 的抑制情况

```
r1#sh ip bgp 20.1.1.0

BGP routing table entry for 20.1.1.0/24, version 8

Paths: (1 available, no best path)

Flag: 0x820

  Not advertised to any peer

  2 (history entry)

    2.2.2.2 (metric 65) from 2.2.2.2 (2.2.2.2)

      Origin IGP, metric 0, localpref 100, external

      Dampinfo: penalty 2886, flapped 3 times in 00:03:08

r1#
```

说明：由于 20.1.1.0/24 翻动 3 次后，penalty 值为 2886，大于抑制值 2000，可以被抑制了。

（5）查看被抑制的路由

```
r1#sh ip bgp dampening dampened-paths

BGP table version is 8, local router ID is 1.1.1.1

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

```
                r RIB-failure, S Stale

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


   Network          From            Reuse    Path

*d 20.1.1.0/24      2.2.2.2         00:05:39 2 i

r1#
```

说明：20.1.1.0/24 已经被抑制。

（6）查看 R1 的 BGP 路由表情况

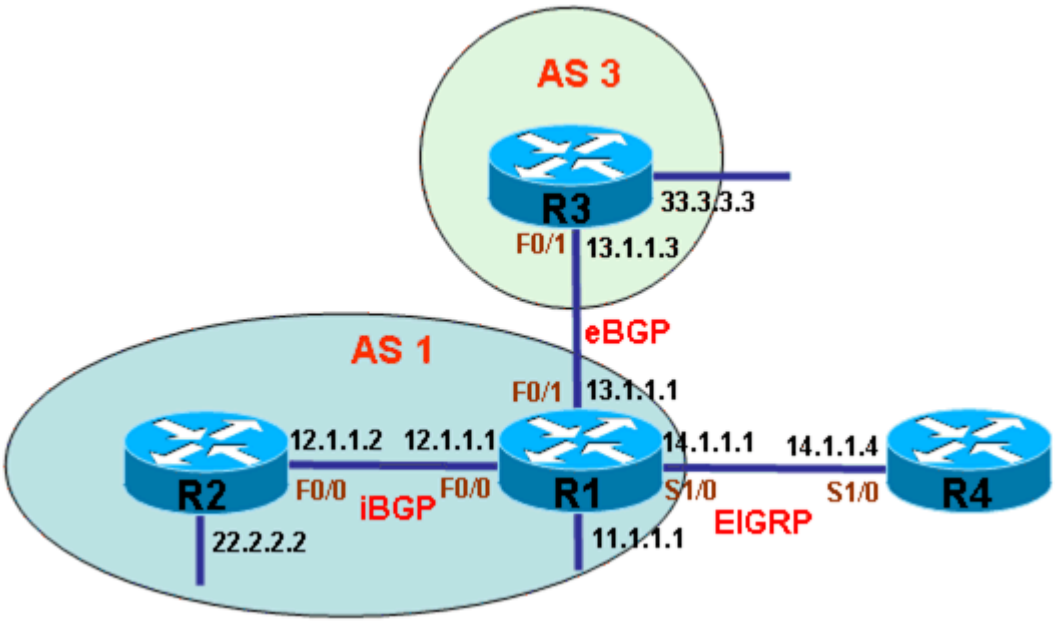
```
r1#sh ip bgp

BGP table version is 8, local router ID is 1.1.1.1

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

                r RIB-failure, S Stale

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


   Network          Next Hop            Metric LocPrf Weight Path

*> 10.1.1.0/24      2.2.2.2                  0             0 2 i

*d 20.1.1.0/24      2.2.2.2                  0             0 2 i

r1#
```

说明：BGP 路由表中也显示 20.1.1.0/24 是被抑制的。

BGP 重分布进 IGP

因为 BGP 通常拥有庞大的路由表，所以在将 BGP 路由表重分布进 IGP 时，很有可能导致 IGP 协议停止工作或路由器崩溃，所以为了预防此类事件的发生，慢慢的，IOS 默认不允许将 BGP 重分布进 IGP，但是并非所有 BGP 都不能重分布进 IGP，为了放宽限制，默认情况下，只可以将从 eBGP 邻居学习到的路由和本地路由重分布进 IGP，也就是说 iBGP 路由是不能重分布进 IGP 的，但是可以手工调整允许将 iBGP 学习到的路由重分布进 IGP。

配置 BGP 重分布进 IGP



说明：

上图中路由器 R1，R2，R3 配有 Loopback 地址，地址分别为：

R1 Loopback 0 1.1.1.1/32

R2 Loopback 0 2.2.2.2/32

R3 Loopback 0 3.3.3.3/32

所有路由器之间运行 OSPF，并将 Loopback 0 的地址发布到 OSPF 中，保证全网 Loopback 0 之间是可以通信的。

1. IGP 保证全网 Loopback 0 互通

说明：此步略，请参见之前配置。

2. 配置 BGP

（1）配置 R1 的 BGP

```
r1(config)#router bgp 1
r1(config-router)#bgp router-id 1.1.1.1
r1(config-router)#neighbor 2.2.2.2 remote-as 1
r1(config-router)#neighbor 2.2.2.2 update-source loopback 0
r1(config-router)#neighbor 3.3.3.3 remote-as 3
r1(config-router)#neighbor 3.3.3.3 update-source loopback 0
r1(config-router)#neighbor 3.3.3.3 ebgp-multihop
r1(config-router)#network 11.1.1.0 mask 255.255.255.0
```

说明：R1 与 R2 建立 iBGP 邻居，与 R3 建立 eBGP 邻居。

（2）配置 R2 的 BGP

```
r2(config)#router bgp 1
r2(config-router)#bgp router-id 2.2.2.2
```

```
r2(config-router)#neighbor 1.1.1.1 remote-as 1

r2(config-router)#neighbor 1.1.1.1 update-source loopback 0

r2(config-router)#network 22.2.2.0 mask 255.255.255.0
```

说明：R2 与 R1 建立 iBGP 邻居。

（3）配置 R3 的 BGP

```
r3(config)#router bgp 3

r3(config-router)#bgp router-id 3.3.3.3

r3(config-router)#neighbor 1.1.1.1 remote-as 1

r3(config-router)#neighbor 1.1.1.1 update-source loopback 0

r3(config-router)#neighbor 1.1.1.1 ebgp-multihop

r3(config-router)#network 33.3.3.0 mask 255.255.255.0
```

说明：R3 与 R1 建立 eBGP 邻居。

(4)查看 R1 的 BGP 路由表

```
r1#sh ip bgp

BGP table version is 4, local router ID is 1.1.1.1

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



internal,

              r RIB-failure, S Stale

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

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 11.1.1.0/24      0.0.0.0                  0         32768 i
*>i22.2.2.0/24      2.2.2.2                  0    100      0 i
*> 33.3.3.0/24      3.3.3.3                  0             0 3 i
r1#
```

说明：R1 中包含本地路由 11.1.1.0/24，iBGP 路由 22.2.2.0/24，eBGP 路由 33.3.3.0/24。

3.配置 EIGRP

（1）在 R1 上配置 EIGRP

```
r1(config)#router eigrp 100
r1(config-router)#no auto-summary
r1(config-router)#network 14.1.1.1 0.0.0.0
```

说明：在 R1 与 R4 之间建立 EIGRP。

（2）在 R4 上配置 EIGRP

```
r4(config)#router eigrp 100
r4(config-router)#no auto-summary
r4(config-router)#network 14.1.1.4 0.0.0.0
```

说明：在 R4 与 R1 之间建立 EIGRP。

（3）查看 R4 的 EIGRP 邻居

```
r4#sh ip eig neighbors

IP-EIGRP neighbors for process 100

   H   Address                 Interface       Hold
Uptime   SRTT   RTO  Q  Seq

                                               (sec)          (ms)
 Cnt Num

   0   14.1.1.1                Se1/0            137 00:00:44
1590  5000  0  2

r4#
```

说明：R4 已经与 R1 正常建立 EIGRP 邻居。

4.配置 BGP 重分布进 EIGRP

（1）在 R1 上重分布 BGP 进 EIGRP

```
r1(config)#router eigrp 100

r1(config-router)#redistribute bgp 1 metric 10000 100 255 1 1500
```

说明：R1 将 BGP 重分布 BGP 进 EIGRP。

（2）在 R4 上查看 EIGRP 路由

```
r4#sh ip route

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

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

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

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

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

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

          o - ODR, P - periodic downloaded static route


   Gateway of last resort is not set


        33.0.0.0/24 is subnetted, 1 subnets

   D EX     33.3.3.0 [170/2195456] via 14.1.1.1, 00:00:49, Serial1/0

        11.0.0.0/24 is subnetted, 1 subnets

   D EX     11.1.1.0 [170/2195456] via 14.1.1.1, 00:00:49, Serial1/0

        14.0.0.0/24 is subnetted, 1 subnets

   C        14.1.1.0 is directly connected, Serial1/0

   r4#
```

说明：默认情况下，R1 只能将本地路由 11.0.0.0/24 和 eBGP 路由 33.3.3.0/24 重分布进 IGP。

（3）允许将 iBGP 重分布进 IGP

```
r1(config)#router bgp 1

r1(config-router)#bgp redistribute-internal
```

说明：配置 BGP 允许将 iBGP 重分布进 IGP

(4)再次查看 R4 的 EIGRP 路由

```
r4#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     33.0.0.0/24 is subnetted, 1 subnets
D EX    33.3.3.0 [170/2195456] via 14.1.1.1, 00:01:15, Serial1/0
     22.0.0.0/24 is subnetted, 1 subnets
D EX    22.2.2.0 [170/2195456] via 14.1.1.1, 00:00:02, Serial1/0
     11.0.0.0/24 is subnetted, 1 subnets
D EX    11.1.1.0 [170/2195456] via 14.1.1.1, 00:01:15, Serial1/0
     14.0.0.0/24 is subnetted, 1 subnets
C       14.1.1.0 is directly connected, Serial1/0
r4#
```

说明：配置允许将 iBGP 重分布进 IGP 后，就表示允许所有 BGP 路由重分布进 IGP，所以 R4 从 EIGRP 收到 BGP 重分布进来的所有路由。

Feature

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

Directed-broadcast（定向广播）

概述

在默认情况下，Cisco 路由器在收到任何广播数据的时候，默认都是丢弃而不转发，在某些时候，有些广播是必须的，比如我们通常使用的 NetBios Name Server 协议，DNS 协议，还有 DHCP 协议。就像 DHCP 协议，当主机在没有地址的情况下，向服务器请求 IP 地址，正因为自己没有 IP 地址，却又不知道服务器在哪，所以需要使用广播来查找，这时，如果服务器在远程网络，而路由器又拒绝转发广播，

鸿鹄培训+考试群：301283867

那么将给我们的 DHCP 带来麻烦。

要如何才能在这种情况下让网络正常工作呢，那就是让路由器帮我们把广播转发到我们需要到达的目标网络，这样需要到达远程网络的广播，我们称为定向广播，比如我们本地网络是 10.1.1.0/24 的网段，我们需要将广播发到 192.168.1.0/24 的网段，那么只想让广播在 192.168.1.0/24 网段发送，广播地址为 192.168.1.255，但如果直接这样发，是到达不了 192.168.1.0/24 网段的，所以需要得到路由器的允许。

Directed-broadcast

每个网络都是与路由器的某个接口相连的，这个网络需要向远程网络发送广播，首先就是发送到路由器与之相连的接口，那么当路由器从该接口收到广播之后，决定是丢弃还是转发，就在于路由器的接口是否允许定向广播功能，如果我们需要让路由器帮我们转发定向广播，就需要在接口上手工配置 directed-broadcast 转发功能。

配置

1. 配置 Directed-broadcast

（1）定义 ACL

R1(config)#access-list 10 permit any

注：ACL 是用来告诉接口哪些数据包是可以传递定向广播的，加了 ACL 之后，只传递该 ACL

所有允许的数据，如果不加 ACL，默认传递所有定向广播数据。

（2）在接口上开启 directed-broadcast

R1(config)#int f0/0

R1(config-if)#ip directed-broadcast 10

注：IOS 12.0 开始，默认接口上是关闭 directed-broadcast 的。（注意 IOS 版本，请以自身为准。）

Forward-protocol

在路由器接口上开了 directed-broadcast 之后，路由器便能够将该接口上收到的定向广播发往相应的远程网络，比如从 10.1.1.0/24 收到的广播发往 192.168.1.255，但是如果一个目标地址为 255.255.255.255（此广播称为本地广播）的 DHCP 广播请求包，需要发往 DHCP 服务器所在的 192.168.1.0/24 网络（DHCP 服务器地址为 192.168.1.100），路由器会自动将该广播包转到 192.168.1.100 吗？答案当然是不会，那么又如何让路由器在收到目标地址为 255.255.255.255 的广播包，就知道要转发到 192.168.1.100 呢？这就需要在路由器接口上定义 ip helper-address，之后路由器就会将广播转成单播发到 ip helper-address 后面的目标 IP，但是要注意的是，什么样的广播才会被转发到 ip helper-address 后面的目标 IP，是需要根据 forward-protocol 来定义的，如果 forward-protocol 不允许任何协议，那么路由器就不会将任何广播发到 ip helper-address 后面的目标 IP。

forward-protocol 默认允许通过的协议为：

Trivial File Transfer (TFTP) (port 69)

Domain Name System (port 53)

Time service (port 37)

NetBIOS Name Server (port 137)

NetBIOS Datagram Server (port 138)

Boot Protocol (BOOTP) client and server datagrams (ports 67 and 68)

TACACS service (port 49)

所以默认情况下，在接口上加了 ip helper-address，路由器也就只能转发这些默认的端口。

配置

1. 配置 forward-protocol

（1）定义可以转发的协议和端口

```
R1(config)#ip forward-protocol udp 3001
```

注：默认是开启 forward-protocol 的，且只能转发默认协议和端口。

（2）在接口下配置 ip helper-address

R1(config)#int f0/0

R1(config-if)# ip helper-address 192.168.1.100

注：默认是关闭 ip helper-address 的。

说明：通过以上配置之后，当路由器从 f0/0 接口上收到目标地址为 255.255.255.255 的广播，并且目标为 UDP 3001 的广播之后，就会转成单播发往 192.168.1.100。如果没有配置 forward-protocol，路由器就只能转发默认的协议和端口。

DHCP

概述

DHCP 应该是一个大家都非常熟悉的协议，可能算不上什么 feature，平时我们使用电脑时，经常会用到，它的功能也是大家所知道的：动态地为主机分配 IP 地址以节省工作量。由此可以看出，DHCP 可以由两个部分组成，即 DHCP 服务器和 DHCP 客户端（通常为我们所使用的 PC）。那么在 Cisco 设备分别扮演 DHCP 服务器和 DHCP 客户端时，与其它设备有什么不同之处呢，下面来详细地介绍。

DHCP 是一个协议，无论它运行在主机上，还是在路由器或交换机上，其运行的规则，就像 TCP/IP 协议一样，是不允许我们使用任何命令更改的，DHCP 前身是 BOOTP 协议，使用的端口号为：服务器端是 UDP 67(cisco 设备上称为 bootps)，客户端是 UDP 68（cisco 设备上称为 bootpc）,CCIE 考生需牢记。当一台主机接入网络后，在没有 IP 地址的情况下，向网络上发送 DHCP 请求获得 IP 地址时，正因为自己还没有 IP 地址，所以发送数据包时，源 IP 为 0.0.0.0,源 MAC 正常，而自己也不可能知道谁是 DHCP 服务器，所以数据包是目标 IP 地址为 255.255.255.255、目标 MAC 地址为 FFFFFF 的广播包。当本地网络中如果存在 DHCP 服务器，那么 DHCP 服务器从某个网卡收到请求后，便向客户端发送一个地址信息，其中包含我们需要使用的 IP 地址，子网掩码，网关，DNS 等信息，同时这些信息会携带一个租约时间（即这个地址客户端可以使用多久，过了这个时间，那么服务器将该地址提供给其它客户端使用），当然，客户端也可以提前结束该地址的使用。

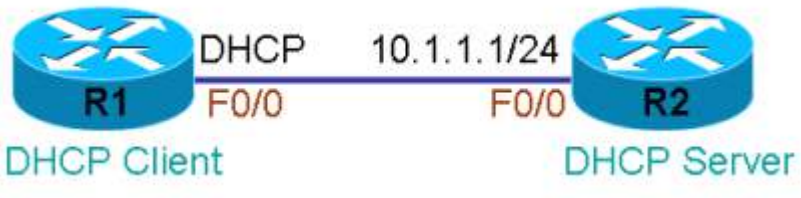
当我们使用的客户端为 windows 主机，其接入网络之后，当网卡配置为 DHCP 获得地址时，就开始向网络中请求地址，先发送一个广播包，等待 1 秒之后，如果

没有服务器应答，就开始尝试发送第二个广播包，如果又等了 9 秒没有收到应答，则发送第三个广播包，第三个是等 13 秒，还没有应答，最后再发送一个包，等待 16 秒后，最终在四个广播包没有应答的情况下，也就是从发送第一个请求包到 39 秒之后，默认是放弃请求，但最后也会为网卡自动配上一个私有 IP 地址，地址段为 169.254.0.0/16,并且你会看到网卡连接图标上出现黄色感叹号，状态名为“受限制或无连接”，即使这时网络中出现了 DHCP 服务器，也救不了这台主机，这就是 windows 主机作为 DHCP 客户端的情况。那么使用 Cisco 设备作为 DHCP 客户端的不同之处是什么呢？当 Cisco 设备的接口配置为 DHCP 获得地址时，便向网络中发出广播，如果等待 5 秒都没有收到应答，就再次发送，再等 10 秒，没有收到就再发，然后再等 15 秒,20 秒，25 秒，30 秒•••60 秒，由此可以看出其间隔是随着次数的增加而每次加 5 秒，但尝试到 60 秒后，便不再增加，又会重新回到 5 秒，以次类推，Cisco 设备在得不到地址的情况下，并不会为接口自动配上网段为 169.254.0.0/16 的地址，所以如果网络中之后出现 DHCP 服务器，就会为该设备的接口分发一个 IP 地址。当 DHCP 客户端得到 IP 地址后，因为地址使用是有时间限制的，当这个时间过去一半的时候，客户端会向服务器续约，以请求继续使用该地址，服务器同意后，该地址的使用时间会被刷新，如果在时间过去一半时，续约不成功，便会在总时间过去 75%的时候再续约一次，如果还不成功，就会放弃该地址的使用权。

服务器与客户端之间并没有 hello 这样的数据包来保持会话状态，所以当一台客户端得到一个 IP 地址的使用权后，中途离开网络，服务器是无法知道的，也就无法将该 IP 地址重新分配给他人使用，所以建议大家在配置服务器时，可以将租约配的越短越好，以免造成一个地址发给客户使用，而这台客户机已经离开了，该 IP 地址还长时间不能重新发给新的客户使用，建议租约配置为 1 分钟，因为一个地址在租约过半时，客户端会续约，也就是可以再次使用同一个地址，所以不用担心一分钟之后，客户端会重新获得别的 IP 地址。当一台 DHCP 客户端收到服务器提供的 IP 地址后，会使用 Gratuitous ARP 来查讯网络，即使用该 IP 地址为目的 IP，目标 MAC 为 FFFFFF 发到网络里，如果有人回答该数据包，则证明该 IP 地址在网络中已经有他人在使用，那么将向 DHCP 服务器重新请求获得别的 IP 地址。（注：Gratuitous ARP 通过正常手段无法关闭）

配置

DHCP 基础



1. 配置 DHCP Server

（1）开启 DHCP 功能

r2(config)#service dhcp

（2）配置 DHCP 地址池

r2(config)#ip dhcp pool ccie1 地址池名为 ccie1

r2(dhcp-config)#network 10.1.1.0 255.255.255.0 可供客户端使用的地址段

r2(dhcp-config)#default-router 10.1.1.1 网关

r2(dhcp-config)#dns-server 10.1.1.1 10.1.1.2 DNS

r2(dhcp-config)#lease 1 1 1 租期为 1 天 1 小时 1 分（默认为一天）

r2(config)#ip dhcp pool ccie2 地址池名为 ccie1

r2(dhcp-config)#network 20.1.1.0 255.255.255.0 可供客户端使用的地址段

r2(dhcp-config)#default-router 20.1.1.1 网关

r2(dhcp-config)#dns-server 20.1.1.1 20.1.1.2 DNS

r2(dhcp-config)#lease 1 1 1 租期为 1 天 1 小时 1 分（默认一天）

（3）去掉不提供给客户端的地址

注：因为某些 IP 地址不希望提供给客户端，比如网关地址，所以我们要将这些地址从地址池中移除，这样服务器就不会将这些地址发给客户端使用。

r2(config)#ip dhcp excluded-address 10.1.1.1 10.1.1.10 移除 10.1.1.1 到 10.1.1.10

r2(config)#ip dhcp excluded-address 20.1.1.1 20.1.1.10 移除 20.1.1.1 到 20.1.1.10

2. 配置 DHCP Client

（1）配置接口使用 DHCP

r1(config)#int f0/1

r1(config-if)#ip address dhcp

3. 查看命令：

（1）在服务器上查看哪些地址分配给了哪些主机：

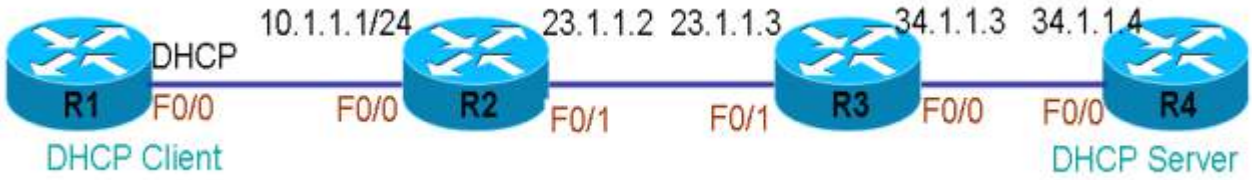
R2#Show ip dhcp binding

4. 查看结果

查看 DHCP Client 会看到接口 F0/0 的 IP 地址为 10.1.1.11 并且产生一条指向 10.1.1.1 的默认路由（换成 PC 就会变成网关是 10.1.1.1），路由器并不需要得到 DNS。

在这里，DHCP Server 上明明配了两个地址池，网段分别为 10.1.1.0/24 和 20.1.1.0/24,为什么客户端向服务器请求地址的时候，服务器就偏偏会把 10.1.1.0/24 网段的地址发给客户，而不会错把 20.1.1.0/24 网段的地址发给客户呢。这是因为服务器从哪个接口收到 DHCP 请求，就只能向客户端发送地址段和接收接口地址相同的网段，如果不存在相同网段，就会丢弃请求数据包。图中接收接口地址为 10.1.1.1，而地址池 ccie1 中的网段 10.1.1.0/24 正好和接收接口是相同网段，所以向客户端发送了 IP 地址 10.1.1.11。

DHCP 中继



如图中所示，当 R1 的接口配置为 DHCP 获得地址后，那么将从 F0/0 发出目的地为 255.255.255.255 的广播请求包，如果 R2 为 DHCP 服务器，便会响应客户端，但它不是 DHCP 服务器，因此 R2 收到此广播包后便默认丢弃该请求包。而真正的 DHCP 服务器是 R4，R1 的广播包又如何能到达 R4 这台服务器呢，R4 又如何向 R1 客户端发送正确的 IP 地址呢。

路由器是不能够转发广播的，因此，除非能够让 R2 将客户端的广播包单播发向 R4 这台服务器。 我们的做法就是让 R2 将广播包通过单播继续前转到 R4 这台服务器，称为 DHCP 中继，通过 IP help-address 功能来实现。

1. R2 配置

（1）配置将 DHCP 广播前转到 34.1.1.4

注： IP help-address 功能默认能够前转 DHCP 协议，所以无需额外添加。

```
R2(config)#int f0/0
R2(config-if)#ip helper-address 34.1.1.4
```

2. 配置 DHCP Server:

（1）开启 DHCP 功能

```
R4(config)#service dhcp
```

（2）配置 DHCP 地址池

```
R4(config)#ip dhcp pool ccie1                    地址池名为 ccie1
R4(dhcp-config)#network 10.1.1.0 255.255.255.0   可供客户端使用的地址段
R4(dhcp-config)#default-router 10.1.1.1          网关
```

R4(config)#ip dhcp pool ccie2 地址池名为 ccie1

R4(dhcp-config)#network 34.1.1.0 255.255.255.0 可供客户端使用的地址段

R4(dhcp-config)#default-router 34.1.1.4 网关

（3）去掉不提供给客户端的地址

R4(config)#ip dhcp excluded-address 10.1.1.1 10.1.1.10 移除 10.1.1.1 到 10.1.1.10

R4(config)#ip dhcp excluded-address 34.1.1.1 34.1.1.10 移除 20.1.1.1 到 20.1.1.10

（4）配置正确地址池的路由

R4(config)#ip route 10.1.1.0 255.255.255.0 34.1.1.3

注： R3 无需做任何配置！

3. 查看结果

查看 DHCP Client 会看到接口 F0/0 的 IP 地址为 10.1.1.11，那么 DHCP 服务器 R4 又是根据什么来判断出客户端需要的是哪个网段的 IP 地址呢，为什么还是没有错把 34.1.1.0/24 网段的地址发给客户呢。不是说服务器从哪个接口收到请求，就把这个接口相同网段的地址发给客户端吗？按照之前的理论，应该是发送 34.1.1.0/24 的地址给客户啊。在这里，能够指导服务器发送正确 IP 地址给客户端，是因为有一个被称为 option 82 的选项，这个选项只要 DHCP 请求数据包被中继后便会自动添加，此选项,中继路由器会在里面的 giaddr 位置写上参数，这个参数，就是告诉服务器，客户端需要哪个网段的 IP 地址才能正常工作。中继路由器从哪个接口收到客户的 DHCP 请求，就在 option 82 的 giaddr 位置写上该接收接口的 IP 地址，然后服务器根据 giaddr 位置上的 IP 地址，从地址池中选择一个与该 IP 地址相同网段的地址给客户，如果没有相应地址池，则放弃响应，所以，服务器 R4 能够正确发送 10.1.1.0/24 的地址给客户，正是因为 R2 在由于 IP help-address 的影响下，将 giaddr 的参数改成了自己接收接口的地址，即将 giaddr 参数改成了 10.1.1.1，通过 debug 会看到如下过程：

*Mar100:28:36.666: DHCPD: setting giaddr to 10.1.1.1.

鸿鹄培训+考试群：301283867

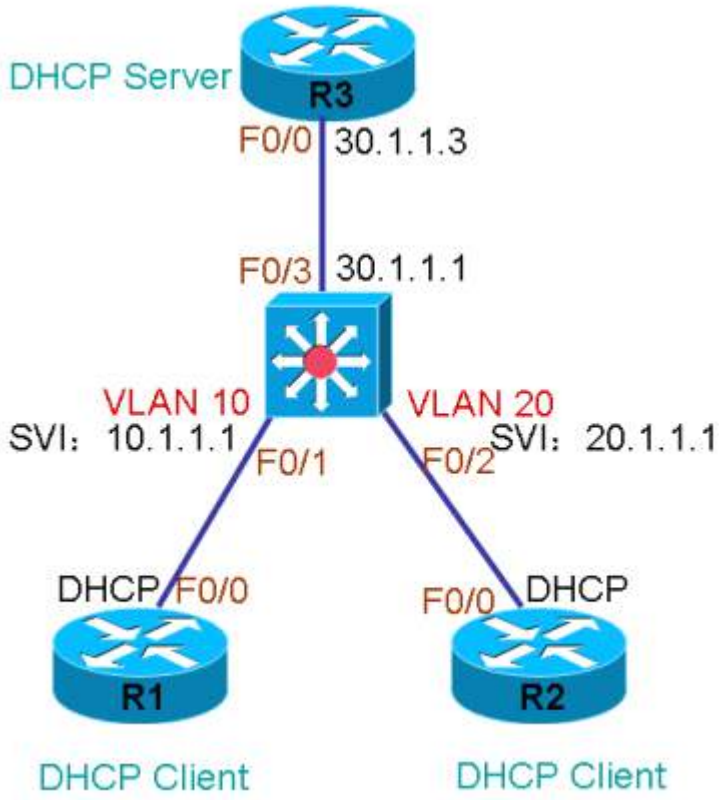
*Mar100:28:36.666:DHCPD:BOOTREQUESTfrom0063.6973.636f.2d30.3031.322e.

6439.6639.2e63.3638.302d.4661.302f.30 forwarded to 34.1.1.4.

从上面 debug 信息可以看到 R2 是将 giaddr 改成 10.1.1.1 后发中继发向 34.1.1.4 的，需要知道的是，经过中继后发来的 DHCP 请求包如果 giaddr 位置不是某个 IP 地址而是 0.0.0.0 的话，服务器是丢弃该请求而不提供 IP 地址的。

注：当服务器上存在 10.1.1.0/24 网段的地址池时，服务器要将该地址池发送给客户，就必须存在到达 10.1.1.0 网段的路由(默认路由也行)，并且客户端必须位于该路由的方向，如果方向不对，该地址池也是不能够发给客户使用的。

不同 VLAN 分配不同地址



如图 3 中所示，两个 DHCP 客户端分别位于交换机上两个不同的 VLAN，交换机上的 VLAN 接口将作为他们的网关，R3 是 DHCP 服务器，这两个客户端必须得到不同网段的地址，否则无法与外网通信，在这种情况下，服务器 R3 也必须正确

为 R1 分配 10.1.1.0/24 网段的地址，必须为 R2 分配 20.1.1.0/24 的地址，配置如下：

1. 配置 DHCP Server

（1）开启 DHCP 功能

R3(config)#service dhcp

（2）配置 DHCP 地址池

R3(config)#ip dhcp pool ccie1 地址池名为 ccie1

R3(dhcp-config)#network 10.1.1.0 255.255.255.0 可供客户端使用的地址段

R3(dhcp-config)#default-router 10.1.1.1 网关

R3(config)#ip dhcp pool ccie2 地址池名为 ccie1

R3(dhcp-config)#network 20.1.1.0 255.255.255.0 可供客户端使用的地址段

R3(dhcp-config)#default-router 20.1.1.1 网关

（3）去掉不提供给客户端的地址

R3(config)#ip dhcp excluded-address 10.1.1.1 10.1.1.10 移除 10.1.1.1 到 10.1.1.10

R3(config)#ip dhcp excluded-address 20.1.1.1 20.1.1.10 移除 20.1.1.1 到 20.1.1.10

（4）配置正确地址池的路由

R3(config)#ip route 10.1.1.0 255.255.255.0 30.1.1.1

R3(config)#ip route 20.1.1.0 255.255.255.0 30.1.1.1

2. 配置交换机

（1）配置相应接口信息

sw(config)#vlan 10

sw(config-vlan)#exit

sw(config)#vlan 20

sw(config-vlan)#exit

sw(config)#int f0/1

sw(config-if)#switchport mode access

sw(config-if)#switchport access vlan 10

sw(config-if)#exit

sw(config)#int f0/2

sw(config-if)#switchport mode access

sw(config-if)#switchport access vlan 20

sw(config-if)#exit

sw(config)#int vlan 10

sw(config-if)#ip address 10.1.1.1 255.255.255.0

sw(config-if)#ip helper-address 30.1.1.3 单播前转 DHCP 广播到 30.1.1.3

sw(config-if)#exit

sw(config)#int vlan 20

sw(config-if)#ip address 20.1.1.1 255.255.255.0

sw(config-if)#ip helper-address 30.1.1.3 单播前转 DHCP 广播到 30.1.1.3

3. 配置 DHCP Client

（1）配置 R1

```
r1(config)#int f0/1

r1(config-if)#ip address dhcp
```

(2)配置 R2

```
r2(config)#int f0/1

r1(config-if)#ip address dhcp
```

4. 查看结果：

按上述配置完之后，客户端 R1 的 F0/0 便能够收到地址 10.1.1.11,客户端 R2 便能够收到地址 20.1.1.11，然后就可以全网通信。在上述的情况下，服务器 R3 能够正确为 R1 分配 10.1.1.0/24 网段的地址，能够正确为 R2 分配 20.1.1.0/24 网段的地址，同样也是因为交换机在收到 R1 的 DHCP 广播包后，将 giaddr 的参数改成了 10.1.1.1，收到 R2 的广播包后，将 giaddr 的参数改成了 20.1.1.1，所以最后服务器 R3 能够根据 giaddr=10.1.1.1 的包分配 10.1.1.0/24 的地址，根据 giaddr=20.1.1.1 的包分配 20.1.1.0/24 的地址。

IP 与 MAC 地址绑定

在配置 DHCP 时，地址池中除了移除掉的 IP 地址之外，所有的地址都会按顺序分配给客户，所以客户机得到的 IP 地址是无法固定的，有时需要每次固定为某些 PC 分配相同的 IP 地址，那么这时就可以配置 DHCP 服务器以静态将 IP 地址和某些 MAC 绑定，只有相应的 MAC 地址才能获得相应的 IP 地址。在 Cisco 设备上静态将 IP 与 MAC 绑定的方法为，需要将某个 IP 地址绑定给 MAC 地址，就为该 IP 地址单独创建地址池，称为 host pool，地址池中需要注明 IP 地址和掩码位数，并且附上一个 MAC 地址，以后这个 IP 地址就只分配给这个 MAC 地址，所以 host pool 只能有一个 IP 地址和一个 MAC 地址，如果需要为多个客户绑定 IP 和 MAC，就必须得单独为每个客户都配置各自的 host pool，还要注意的是，在 host pool 中，MAC 地址的表示方法和平常不一样，比如一个主机网卡的 MAC 地址为 aabb.ccdd.eeff，在地址池中，需要在前面加上 01（01 表示为以太网类型），结果为 01aa.bbcc.ddee.ff

1. 配置 host pool:

（1） 配置 pool 名

r1(config)#ip dhcp poo ccie

（2）配置 IP 地址

r1(dhcp-config)#host 10.1.1.100 /24

（3）配置与该 IP 地址对应的 MAC 地址

r1(dhcp-config)#client-identifier 01aa.bbcc.ddee.ff

2. 查看配置结果：

（1）查看服务器地址分配状态

r1#sh ip dhcp binding

Bindings from all pools not associated with VRF:

IP address Client-ID/ Lease expiration Type

Hardware address/

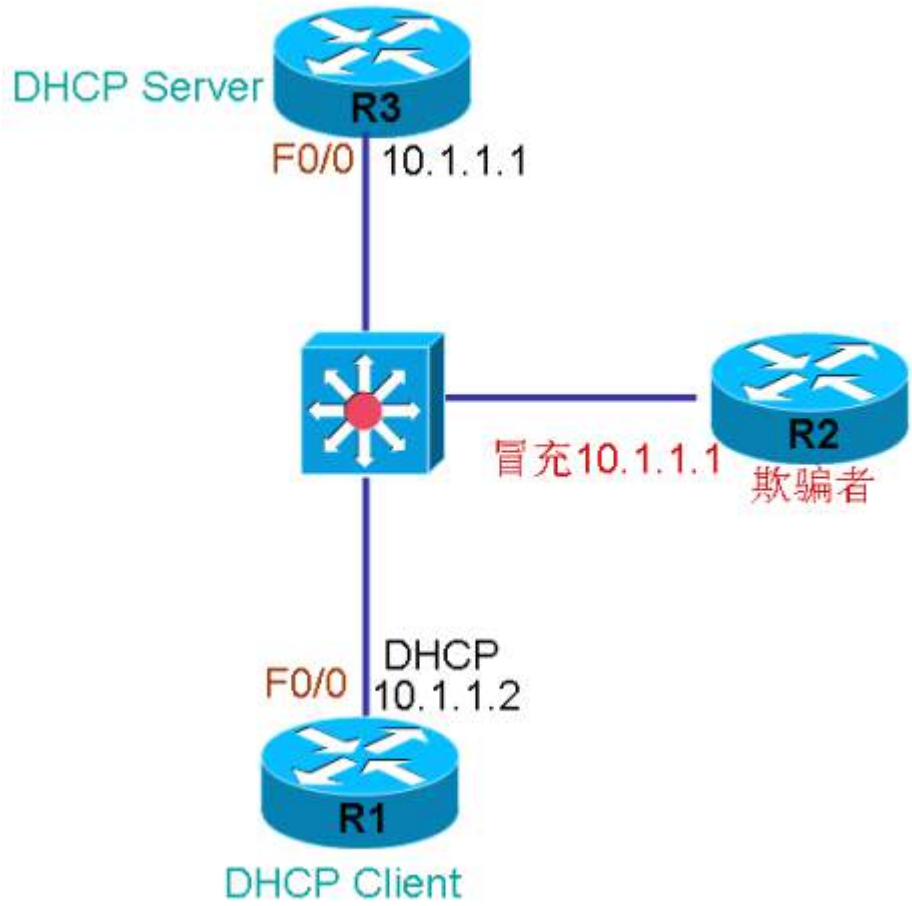
User name

10.1.1.100 01aa.bbcc.ddee.ff Infinite Manual

r1#

说明：从以上结果可以看出，IP 地址 10.1.1.100 已经手工与 MAC 地址 aabb.ccdd.eeff 做了绑定，以后只要 MAC 地址为 aabb.ccdd.eeff 的客户端请求 IP 地址时，才能获得 IP 地址 10.1.1.100。

DHCP 安全 ARP



Cisco 设计的 DHCP 安全 ARP 也许不是绝对的安全，但也起到了一定的作用，原本设计为一个需要计费的公共热点 PVLAN（公共无线场所），如图 4 中所示，R3 为 DHCP 服务器，为付费的 R1 提供正确 IP 地址以提供网络服务，当服务器 R3 为客户端 R1 提供 IP 地址 10.1.1.2 之后，就已经记住了它的 MAC 地址，在正常情况下，如果 R1 退出，服务器是不知道的，并且当网络中有欺骗者接入后，也可冒充 10.1.1.2 这个地址进行上网，当然 R1 和 R2 的 MAC 地址肯定是不一样的，如果这时服务器 R3 由于自动更新 ARP 表的 MAC 地址，就能够顺利让 R2 上网。

基于上述原因，需要在服务器 R3 和客户端 R1 之间提供某种安全机制，即服务器定期 ARP 讯问 10.1.1.2 是否还存在，在讯问时，只有 R1 能够回答。

在完成这种机制，需要两个 feature 来支持，第一个是 Update Arp，在地址池模式下开启，这个 feature 便是定期讯问网络中 DHCP 客户端的;第二个是 Authorized ARP（ARP 授权）,只能在以太网接口下开启，功能是禁止该接口下通过 ARP 自动更新和学习 MAC 地址，这样一来，接口下将不能有手动配置 IP 的设备接入，因为手工配置 IP 接入后，服务器不会更新自己的 ARP 表，也就无法完成到新设备的二层 MAC 地址封装，也就无法和新设备进行通信，只有合法的 DHCP 客户端才能正

常通信，所以，如果为远程客户端分配 IP 地址，就无法做这样的保护，并且到远程客户端的下一跳必须是自己的客户端，因为如果不是，是无法通信的，因为 ARP 不存在到它的条目。

1. 配置安全 ARP：

（1）开启 DHCP 功能

R3(config)#service dhcp

（2）配置 DHCP 地址池

R3(config)#ip dhcp pool ccie1 地址池名为 ccie1

R3(dhcp-config)#network 10.1.1.0 255.255.255.0 可供客户端使用的地址段

R3(dhcp-config)#default-router 10.1.1.1 网关

R3(dhcp-config)#update arp 开启定期 ARP 讯问

（3）去掉不提供给客户端的地址

R3(config)#ip dhcp excluded-address 10.1.1.1 10.1.1.10 移除 10.1.1.1 到 10.1.1.10

（4）在接口下开启 Authorized ARP

R3(config)#int f0/0

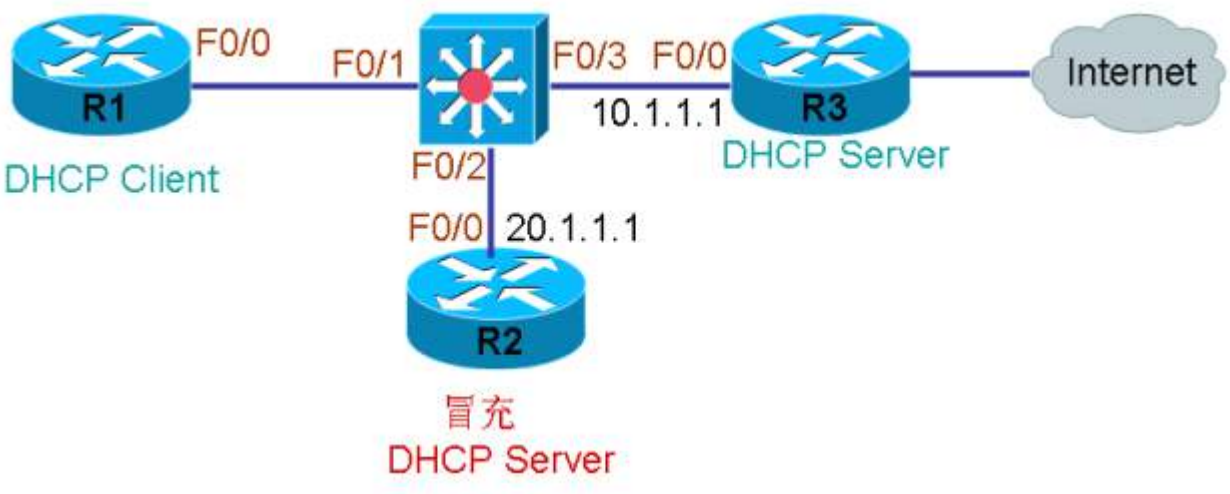
R3(config-if)#Router(config-if)# arp authorized 禁止动态更新 ARP

R3(config-if)# arp timeout 60 60 秒客户无应答则删除 ARP 条目

说明：通过以上配置之后，当 DHCP 客户端从服务器获得 IP 地址后，服务器便会定期查讯该 IP 地址，如果 60 秒没有回答，便从 ARP 表中删除该条目。

DHCP 监听

鸿鹄培训+考试群：301283867



如图 5 中所示，客户端 R1 只有正确从服务器 R3 中获得 10.1.1.0/24 网段的 IP 地址才能够正确上网，如果当网络中出现另外一台错误的 DHCP 服务器（图中 R2），R2 向客户端 R1 发出 20.1.1.0/24 的地址，那么将导致 R1 网络中断，在这样的情况下，就需要禁止不合法的 DHCP 服务器向网络中提供 DHCP 服务，这就需要 DHCP 监听（DHCP Snooping）。DHCP Snooping 是在交换机上完成的，如上图中，只要告诉交换机，只有 F0/3 发来的 DHCP 应答地址才转发给客户端，其它接口发来的应答地址统统被丢弃。要做到这一点，就要告诉交换机，F0/3 接口是它可能信任的 DHCP 地址，其它接口都是不可信的，不能提供 DHCP 应答，那么在实现这个功能时，就需要将交换机上的接口分为可信任接口和不可信任接口两种，默认交换机全为不可信任接口，也就是说交换机开启 DHCP Snooping 之后，没有任何一个接口上的 DHCP 服务器能提供服务。在交换机上配置 DHCP Snooping 时，必须指明在哪个 VLAN 上进行监听，其它没有监听的 VLAN 不受上述规则限制。

1. 交换机上配置 DHCP Snooping

注：交换机上所有接口全部划入 VLAN1

（1）在交换机上开启 DHCP Snooping

sw(config)#ip dhcp snooping　　开启 DHCP Snooping

sw(config)#ip dhcp snooping vlan 1　　在交换机上启用 DHCP Snooping

（2）将相应接口变为信任接口（默认全部为不可信）

鸿鹄培训+考试群：301283867

sw(config-if)#ip dhcp snooping trust

2. 查看命令：

（1）查看 dhcp snooping

Sw#sh ip dhcp snooping

说明：通过以上配置之后，只有交换机 F0/3 接口上（信任接口）的设备能够应答 DHCP 请求，而其它所有接口，比如 R2 过来的 DHCP 应答是会被丢弃的。但是你会发现，在这之后，R1 还是无法获得服务器 R3 发来的 DHCP 地址。这是因为开了 DHCP Snooping 的交换机默认会产生中继效果，即将 DHCP 请求包的 giaddr 的参数改成 0.0.0.0，交换机的这种中继效果是无法关闭的，当一个服务器收到中继后并且将 giaddr 设置为 0.0.0.0 而不是 IP 地址的请求包时，默认是要丢弃该数据包而不作应答的，所以服务器 R3 丢弃了该请求数据包。要让客户 R1 能够正常收到 DHCP 提供的 IP 地址，就要让 DHCP 服务器对即使 giaddr 为 0.0.0.0 的请求包也作出应答。配置如下：

R3(config-if)#ip dhcp relay information trusted

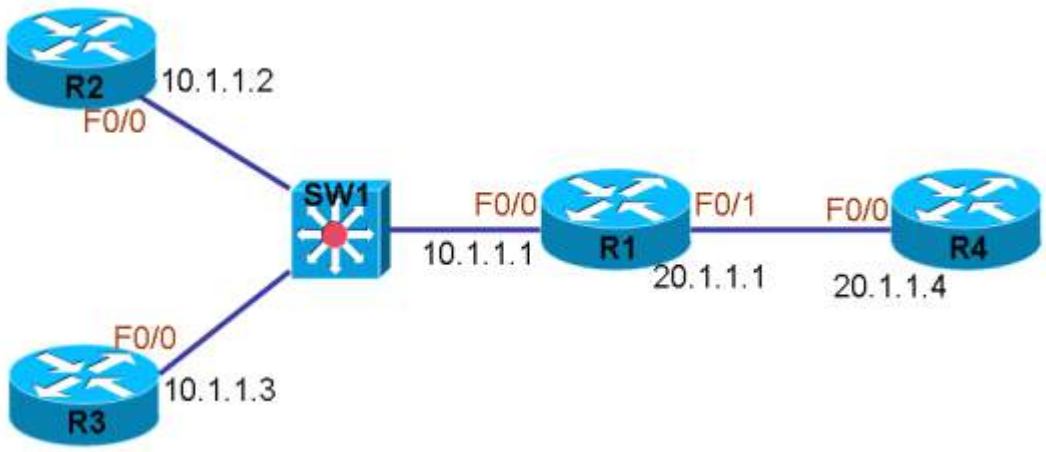
最后，从上图中，如果 R3 本身还不是 DHCP 服务器，如果 DHCP 服务器还在远程网络，需要 R3 提供中继并转发该请求包到服务器的话，那么 R3 除了在接口下配置 ip dhcp relay information trusted 之外，还必须配置 ip helper-address，两者缺一不可。

IP Accounting（记账）

有时需要在路由器上查看某台主机通过路由器的流量是多少，这时就需要路由器能够记录下该主机的数据量。主机之间进行通信时，发出的数据包都有源 IP 和目的 IP 共两个 IP 地址，路由器在记录流量时，可根据这些 IP 地址来定义要记录的流量。虽然通信时数据包有两个地址，但路由器只需要定义一个地址即可，这个地址不需要说明是源还是目的，也就是说一个数据包无论是源 IP 匹配还是目的 IP 匹配，都会被路由器记录，但是在记录条目里，会同时写上流量发生的源 IP 和目的 IP。在记录流量时，单位是 Byte,其中包含了数据包的包头和数据大小。在路由器开启记账功能后，会影响到路由器的工作性能，请慎用。需要注意的是，路由器只能记录从接口出去的流量，并且从自己发出的流量和发往自己的流量是不能记录的。路由器记录的每一条包含一个源 IP 和一个目标 IP，这一对称为一个条目，路由器能

记录的条目数量是可以随意更改的，默认最多能存储 512 条。

配置



1. 配置 IP Accounting

（1）在路由器接口下直接开启 IP Accounting

r1(config)#int f0/0

r1(config-if)#ip accounting

注：只能记录从接口 f0/0 出去的数据包

2. 测试结果

（1）在 R2 上 ping 20.1.1.4，以测试 R2 到 R4 的数据不作记录，只有 R4 返回 R2 的才记录。

r2#ping 20.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

（2）再从 R4 ping R2

r4#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms

(3)查看 R1 上的流量记录

r1#sh ip accou

*Mar 1 00:08:53.750: %SYS-5-CONFIG_I: Configured from console by console

r1#sh ip accounting

Source	Destination	Packets	Bytes
20.1.1.4	10.1.1.2	10	500

Accounting data age is 0

说明：可以看出，R2 到 R4 的流量没有做记录，因为没有目的为 20.1.1.4 的记录，只看到有 R4 到 R2 的流量记录，因为有目的为 10.1.1.2 的记录。

（4）再测试 R1 的接口 f0/0 是否还会记录别的流量

r3#ping 20.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

r3#

r1#sh ip accounting

Source	Destination	Packets	Bytes
20.1.1.4	10.1.1.2	10	1000
20.1.1.4	10.1.1.3	5	500

Accounting data age is 2

说明：从以上数据表明，R1 会记录从接口 f0/0 出去的任何流量

（5）再测试到 R1 的流量

r3#ping 20.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

r3#

r1#sh ip accounting

Source	Destination	Packets	Bytes
20.1.1.4	10.1.1.2	10	1000
20.1.1.4	10.1.1.3	5	500

Accounting data age is 2

说明：发现没有记录到 R1（即 IP 地址为 20.1.1.1）的流量，说明从 R1 发起的流量和发到 R1 的流量是不做记录的。

3. 配置单独记录到某固定 IP 的流量

（1）配置 R1 只记录到 R3(10.1.1.3)的流量

r1(config)#ip accounting-list 10.1.1.3 0.0.0.0 （后面为通配符掩码 wildcard）

（2）在接口上应用记账功能

r1(config)#int f0/0

r1(config-if)#ip accounting output-packets

注：命令 ip accounting output-packets 和 ip accounting 功能相同。

（3）测试 R3 到 R4 的流量

r3#ping 20.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

（4）测试 R2 到 R4 的流量

R2#ping 20.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

（5）查看流量记录

r1#sh ip accounting

Source	Destination	Packets	Bytes
20.1.1.4	10.1.1.3	5	500

Accounting data age is 1

说明：从结果中看出，只要数据包中有 10.1.1.3 这个地址，便会记录，其它统统不作记录。

NetFlow

概述

在需要对 Cisco 设备上查看数据的流量传输情况的时候，我们可以用到的技术是 IP Accounting，但是可以看出这个技术记录出的流量是非常简洁的，只能看到数据包的量，却看不到数据包的协议。下面有一项技术在 IP Accounting 的基础上增加了一些有用的附加功能，它不仅能记录数据的数量，并且还可以记录出协议等信息，这就是 NetFlow。

Cisco 开发的 NetFlow 共 4 个版本，分别是 ver 1 ，ver 5，ver 8，ver 9，它们的特点是：

V9 不向后兼容 v8 和 v5，需要独立开启。

V8 只支持聚合缓存，不支持新 feature。

V5 只支持主缓存，不支持新 feature。

V1，不要使用，建议用 5 和 9。

Netflow 在网络设备上抓包，在每台设备上独立进行，不需要所有设备开启。

配置 Netflow 的条件：

（1）必须先开启路由功能，

（2）CEF 必开

（3）并且会消耗额外 CPU 和内存资源。

Netflow 抓的包包含：

IP-to-IP

IP-to- MPLS

Frame Relay

ATM

egress (outgoing)

下面 7 个参数相同即被认为是同一流，作相同记录，否则另作记录

（1）Source IP address

（2）Destination IP address

（3）Source port number

（4）Destination port number

（5）Layer 3 protocol type

（6）Type of service (ToS)

（7）Input logical interface

Netflow 抓到的包可以向远程主机发送，发送时使用协议 UDP，端口号默认为 9991，这些远程主机的 IP 和端口号可以随意更改。

配置

说明：Netflow 是基于接口开启的，在某个接口开启后，就在相应接口的相应方向抓取数据包，

在接口上开启 Netflow 时，有先决条件需要打开。

1. 全局开启 CEF（必开）

R1(config)ip cef

2. 接口开启 Netflow（可开多个接口）

（1）早于 IOS 版本 12.2(14)S, 12.0(22)S, or 12.2(15)T 的

r1(config)#int f0/0

r1(config-if)#ip route-cache flow

（2）等于或晚于 IOS 版本 12.2(14)S, 12.0(22)S, or 12.2(15)T 的

r1(config)#int f0/0

r1(config-if)#ip flow ingress

3. 定义远程采集主机，(最多两台)

（1)定义远程 IP 地址，设备将抓过的数据包发向远程主机 默认端口为 UDP 9991）

R1(config)Ip flow-export destination 10.1.1.2 90

(2)配置数据包源地址

R1(config)# ip flow-export source f0/0

4. 定义 Netflow 版本

（1）全局定义 Netflow 版本（默认版本 1，不同 IOS 支持的版本不同）

r1(config)#Ip flow-export ver 5

5. 查看效果：

（1）在 r2 上 ping R3 的 20.1.1.3

r2#ping 20.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/12 ms

r2#

(2)在 R1 上查看记录：

r1#sh ip cache flow

IP packet size distribution (10 total packets):

1-32 64 96 128 160 192 224 256 288 320 352 384 416 44
8 480

.000 .000 .000 1.00 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

512 544 576 1024 1536 2048 2560 3072 3584 4096 4608

.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes

2 active, 4094 inactive, 2 added

6 ager polls, 0 flow alloc failures

Active flows timeout in 30 minutes

Inactive flows timeout in 15 seconds

last clearing of statistics never

Protocol	Total Flows	Flows /Sec	Packets /Flow	Bytes /Pkt	Packets /Sec	Active(Sec) /Flow	Idle(Sec) /Flow

SrcIf	SrcIPaddress	DstIf	DstIPaddress	Pr	SrcP	DstP	Pkts
Fa0/0	10.1.1.2	Fa0/1	20.1.1.3	01	0000	0800	5
Fa0/1	20.1.1.3	Fa0/0	10.1.1.2	01	0000	0000	5

说明：从以上结果可以看出，拥有很详细的数据包记录。

（3）查看更详细

r1#sh ip cache verbose flow

IP packet size distribution (10 total packets):

1-32 64 96 128 160 192 224 256 288 320 352 384 416 448 480

.000 .000 .000 1.00 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

512 544 576 1024 1536 2048 2560 3072 3584 4096 4608

.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes

0 active, 4096 inactive, 2 added

鸿鹄培训+考试群：301283867

32 ager polls, 0 flow alloc failures

Active flows timeout in 30 minutes

Inactive flows timeout in 15 seconds

last clearing of statistics never

Protocol Total Flows Packets Bytes Packets Active(Sec) Idle(Sec)

-------- Flows /Sec /Flow /Pkt /Sec /Flow /Flow

ICMP 2 0.0 5 100 0.0 0.0 15.0

Total: 2 0.0 5 100 0.0 0.0 15.0

SrcIf SrcIPaddress DstIf DstIPaddress Pr TOS Flgs Pkts

Port Msk AS Port Msk AS NextHop B/Pk Active

r1#

并且看到协议也有所记录。

(4).可清除总记录

R1#clear ip flow stats

6. 聚合参数

说明：可以将数据包中某些需要的数据聚合后显示，比如 BGP AS 号码，或者前缀等信息

（1）配置聚合 BGP AS（事先配好 BGP）

r1(config)#ip flow-aggregation cache as 进入聚合配置模式

r1(config-flow-cache)#cache entries 2000 配置可聚合的最多条

目

r1(config-flow-cache)#cache timeout inactive 200 配置不活动会话的超时时间

r1(config-flow-cache)#cache timeout active 50 配置活动会话的时间

r1(config-flow-cache)#enabled 开启

（2）配置记录中 Origin-as 在源和目的中包含 as

R1(config)#ip flow-export version 5 peer-as

7. 查看结果：

(1)从 R2 ping r3

R2#ping 20.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms

R2#

（2）查看记录：

r1#sh ip cache flow aggregation as

IP Flow Switching Cache, 132104 bytes

4 active, 1996 inactive, 4 added

398 ager polls, 0 flow alloc failures

Active flows timeout in 50 minutes

Inactive flows timeout in 200 seconds

Src If	Src AS	Dst If	Dst AS	Flows	Pkts	B/Pk	Active
Fa0/1	30	Fa0/0	0	1	10	100	0.9
Fa0/0	0	Fa0/1	30	1	10	100	0.9

r1#

说明：可以看到拥有的 AS 记录情况。

WCCP

概述

WCCP 被称为网页信息缓存协议，目的在于通过缓存用户曾经访问过的网页数据，在用户下次访问相同网页时读取缓存信息来加快用户的访问速度。

WCCP 的工作原理为每当用户访问一个页面时，WCCP 就将这些页面进行缓存保留，当用户下次请求网页数据时，路由器将用户的请求发到引擎，如果缓存中有相同网页的备份，则直接从缓存中发给用户，提高速度；但是如果没有，就自己重新请求远程网页，然后再发给用户。

配置

1. 开启 WCCP：

（1）启用 WCCP：

Router(config)# ip wccp web-cache

（2）开启 WCCP ver2

(共两个版本，默认为 ver2)

router(config)#ip wccp version 2

（3）在接口上启用：

Router(config)# interface f0/0

Router(config-if)# ip wccp web-cache redirect out

注：在 f0/0 接口的 out 方向启用，表明用户接在 f0/0 上，路由器只将从网站服务器发回的数

据从 f0/0 发给用户时，才会缓存。

（4）限制 WCCP 缓存哪些 WEB 服务器过来的数据包（可选配置）：

例：比如只接收服务器 10.1.1.1（即 ACL10 过来的数据）发来的数据：

router(config)# access-list 10 permit host 10.1.1.1

router(config)# ip wccp web-cache group-list 10

（5）限制到哪些主机或哪些服务器的不被缓存（可选配置）：

例：从服务器发回到目标主机 192.168.196.51 的不缓存，其它全部缓存

Router(config)# access-list 100 deny ip any host 192.168.196.51

Router(config)# access-list 100 permit ip any any

Router(config)# ip wccp web-cache redirect-list 100

（6）缓存组播能力，事先需要路由器配置组播（可选配置）：

例：让路由器缓存到组播目标为 224.2.2.2 的数据

Router(config)# ip wccp web-cache group-address 224.2.2.2

Router(config)# interface interface-number f0/0

Router(config-if)# ip wccp web-cache group-listen

DRP （Director Response Protocol）

导向应答协议

概述

当大型网络里面有多台分布的服务器时，用户在请求服务器数据的时候，可能无法定位与自己最近的服务器，那么这个时候可能需要在网络里放置导向器,导向器的功能就是将用户的请求信息重新定位（重定向）到与之最近的服务器。所以导向器要能够查讯网络里路由器上的 BGP 和 IGP 路由信息，这就需要路由器提供支持，如果路由器能够完全提供这种路由信息的支持，就必须配置成 DRP Server Agent（DRP 服务器代理）。并且建议为网络里所有路由器均配置为 DRP 服务器代理，因为导向器每次将用户的请求定向到服务器代理后，他们将都能够向用户提供响应，而用户便能够采用最快响应的数据，而忽略其它所有。

配置路由器为 DRP 服务器代理，让网内的导向器能够有效的指导和分发流量，这些导向器需要查讯 BGP 和 IGP 的路由信息，所以需要 DRP 协议。

路由器要成为服务器代理，必须满足以下要求：

1.必须在拓朴上和导向器是邻近的且路由上是可达的。

2.应该具有全网的路由信息，IGP 和 BGP 都不能少，须保证自己全网可达。

3.保证能够访问所有导向器。

配置

1.打 DRP agent.

```
r1(config)#ip drp server
```

2.写 ACL 允许哪些能访问

（1）定义 ACL

r1(config)#access-list 10 permit 10.1.1.1

（2）加上 ACL:

r1(config)#ip drp access-group 10

3.使用认证以提供安全：

（1）定义 KeyChain

r1(config)#key chain ccie

r1(config-keychain)#key 1

r1(config-keychain-key)#key-string cisco

r1(config-keychain-key)#exit

r1(config-keychain)#exit

(2)使用认证

r1(config)#ip drp authentication key-chain ccie

4.查看结果：

r1#sh ip drp

IP Event Dampening(IP 事件惩罚)

概述

在网络中，有时会因为一些外在的或者人为的因素，可能引起设备接口翻动（一会儿可用，一会儿不可用），这些接口状态的变化将导致路由器重新计算最优路径，并且将这些路由信息通告给邻居，从而影响的不仅是自身的路由计算和系统资源消耗，还将影响到整个网络的所有设备重新计算路由而消耗巨大的系统资源，这样的结果将是不可想象的。

IP Event Dampening 监控接口的状态，在接口状态经常性变化，不稳定的时候，

就抑制路由协议对这些接口的计算，并且也限制在这些接口上建立路由邻居，直到这些接口回复稳定状态为止。那么，IP Event Dampening 认为什么样的接口为稳定状态，什么样的接口为经常性变化的呢，在这里，有一套自己的标准，这些标准包含四个参数，分别为：

(1)Suppress Threshold（抑制阀值）， 1 to 20000; the default is 2000.

(2)Half-Life Period （半衰期） 1 to 30 seconds. The default is 5 seconds.

(3)Reuse Threshold （重新使用阀值） 1 to 20000 default value is 1000 penalties

(4)Maximum Suppress Time （最大抑制时间） 1 to 20000 seconds.deis 20 seconds (即 4 倍的半衰期)

(1))Suppress Threshold（抑制阀值）

当一个接口由于翻动而要被 IP Event Dampening 抑制住，这接口的惩罚值必须累加到一定的数额才行这个数额就是 Suppress Threshold（抑制阀值），默认是 2000，范围是 1-20000。

(2)Half-Life Period （半衰期）

当一个接口的惩罚值到达抑制阀值被抑制住后，自己的抑制阀值会随着时间的流逝而慢慢降低，这个下降的速度由 Half-Life Period （半衰期）来控制，也就是每过去一个半衰期的时间，惩罚值的数额就降为总数额的一半，默认半衰期为 5 秒，范围是 1-30 少，比如一个接口的惩罚值为 2000，5 秒钟过去后，这个值就为 2000 的一半，即 1000。

(3)Reuse Threshold （重新使用阀值）

当一个接口被抑制住后，如果还要重新被路由协议接受或重新使用，这个接口的惩罚值必须降到一定的数额才行，这个数额就是 Reuse Threshold （重新使用阀值） ,默认为 1000，范围是 1-20000。

(4)Maximum Suppress Time （最大抑制时间）

接口每经过一个 UP 和 down 的状态，就被认为是翻动一次，每翻动一次，惩

罚值就会加 1000，但是为了防止一个接口由于翻动次数过多，而真正等到稳定之后，由于抑制时间过长而不能重新被使用的可能，所以定义了最大抑制值，但定义的不是值，而是一个时间，这个时间意为一个接口被抑制住后，最多过多少时间可以再次被使用，默认为 20 秒，即为半衰期的 4 倍。

在使用 IP Event Dampening 时，会有一些限制，这些限制为：不支持子接口的监控，不支持虚拟接口（即 Virtual Templates）的监控，

当惩罚出现后，与之接口的路由将不出现在路由表中，（包括静态路由）

协议包含：RIP, OSPF, EIGRP, IS-IS, and BGP:，HSRP，CLNS

配置

1.在接口下开启 Dampening

例：配置 半衰期为 30 秒，重新使用阀值为 500，抑制阀值为 1000，最大抑制时间为 100 秒

r1(config)#int f0/0

r1(config-if)#dampening 30 500 1000 100

2. 查看配置

r1#show interface dampening

3.测试效果

（1）让接口翻动，即让接口 shutdown，再 up

r1(config)#int f0/0

r1(config-if)#shutdown

r1(config-if)#no shutdown

（2）查看状态

r1#sh dampening interface

1 interface is configured with dampening.

1 interface is being suppressed.

No features are using interface dampening.

r1#

r1#sh int dampening

FastEthernet0/0

			Flaps		Penalty				Supp
ReuseTm	HalfL	ReuseV	SuppV	MaxSTm	MaxP	Restart			
1	811	TRUE	21	30	500	1000	100	5039	0

r1#

说明：从上面可以看到显示有 1 个接口已被抑制，接口已翻动一次，当前还剩惩罚值为 811,离重新使用时间还剩 21 秒，半衰期为 30 秒，重新使用阀值为 500，抑制阀值为 1000，最大抑制时间为 100 秒。

Core dump（核心崩溃）

概述

核心崩溃是路由器检测到自己发生了一些无法恢复的错误但需要重启自己才能正常工作，并非所有错误都会造成路由器核心崩溃。当出现核心崩溃，路由器重启自己时，其实所有接口都是能够正常工作的，也是能够发送数据包的，但这个时候它并不会为用户转发数据包，只会转发系统必须的数据包。

既然能够在接口上发送数据包，路由器便考虑将自己内存中所有的数据内容复制到远程服务器,这时内容中所有的内容将打包成一个二进制文件,称为 core dump file，这个文件可能会很能够大；远程服务器的类型包括(FTP), (TFTP), 和 Remote Copy Protocol (RCP,建议使用 FTP，如果不指定,默认使用 TFTP。当远程服务器为

路由器时，发生核心崩溃的路由器只将文件的前 16MB 传送过去，即使对方内容再大也不会完全传送过去。

配置

1. 配置 core dump

（1）指定核心崩溃时的传输协议 （默认为 TFTP）

```
r1(config)#exception protocol ftp
```

（2）定义 FTP的用户名和密码

```
r1(config)#ip ftp username ccie
r1(config)#ip ftp password cisco
```

（3）定义远程的服务器

```
r1(config)#exception dump 10.1.1.1
```

（4）定义核心崩溃时将内存中文件打包的名字（默认为 主机名-core）

```
r1(config)#exception core-file dumpfile
```

2. 测试

注:此测试只是测试传送文件，并不测试系统崩溃）

```
R1#write core
Remote host [10.1.1.1]?
Base name of core files to write [dumpfile]?
writing uncompressed ftp://10.77.233.129/cdfile1
Writing cdfile1 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!
```


鸿鹄培训+考试群：301283867

GLBP （Gateway Load Balancing Protocol ）

网关负载均衡协议

概述

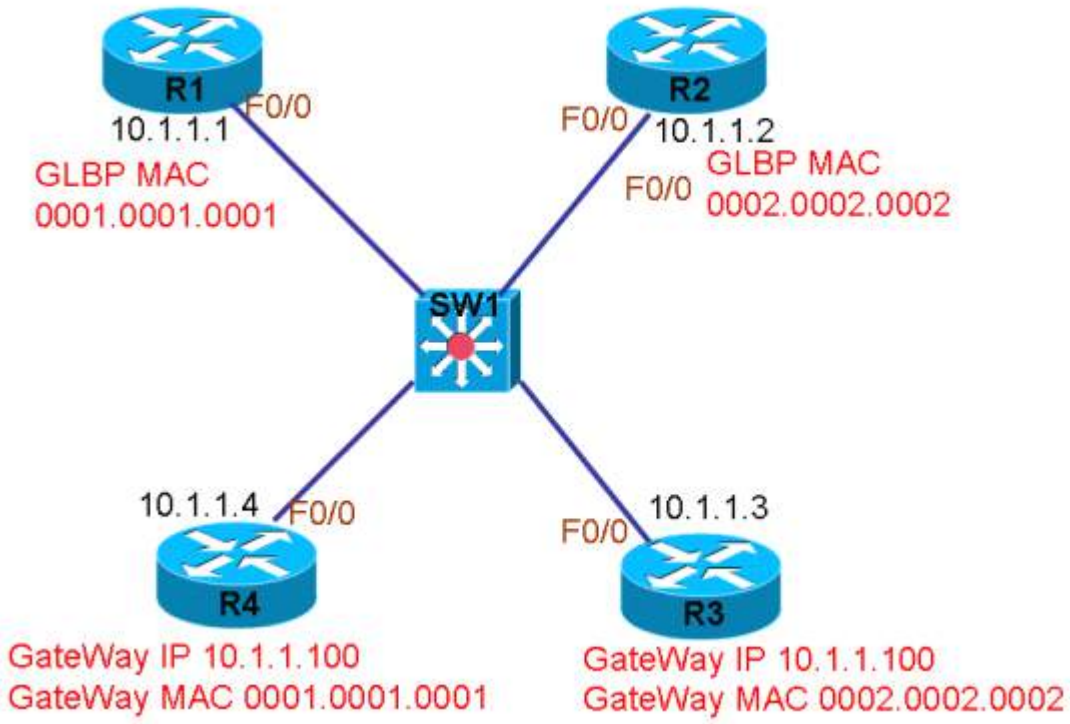
GLBP 是思科的私有协议，在一般高端设备才会支持。GLBP 和 HSRP、VRRP 的目的是一样的，用多台路由器为 LAN 提供网关负载能力，但是 HSRP 和 VRRP 是将多台路由器配成一个组并虚拟成一个 IP 地址，用户将数据发到这个 IP 地址，组内只有一台活动路由器，其它都是备份路由器，只有活动路由器才能转发用户的数据包，而备份路由器只有当活动路由器不可用时才有机会为用户提供数据转发。可以看出，HSRP and VRRP 的备份路由器在平时是没有使用的，要想让所有的路由器都被使用，只有配置多个组，并且用户的主机都要分别指到相应的组 IP。

考虑到以上问题，GLBP 在将多台路由器配置成一个组时，也同时向用户提供一个单一的 IP 地址，额外增加的功能是让组内所有的成员都能够为用户提供数据转发，而不存在空闲的路由器。做法是在为用户提供单一 IP 的同时，每台路由器都为用户提供不同的 MAC 地址，这样，用户发到组 IP 的数据包就成功地被分担各个路由器上了。而组内的各个成员之间也有 hello 数据包来进行交流，3 秒发一个，使用组播地址 224.0.0.102, UDP 协议，端口号 3222。

在 GLBP 组中，选择一台路由器做为 active virtual gateway (AVG),其它路由器做为 AVG 的备份，在 AVG 不可用时顶替 AVG 的角色。在组中，AVG 的任务只是为各成员分配虚拟 MAC 地址，因为大家都是同一个 IP，只要大家的 MAC 地址不同，就可以同时为用户提供数据转发，用户的数据包目标 MAC 发到谁身上，谁就转发相应的数据包。

这些能够为用户提供数据转发的路由器被称为 active virtual forwarders (AVFs) ，所以 AVG 也是 AVF，

图例说明 GLBP：



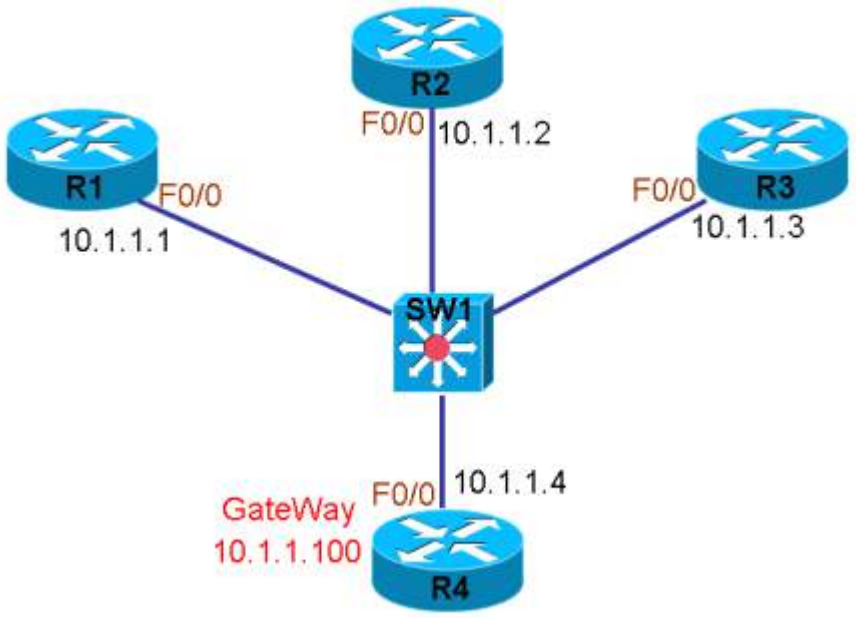
从上图中，R1 和 R2 配置为 GLBP 中成员，组 IP 为 10.1.1.100,而 R3 和 R4 为主机，两台主机的网关 IP 都为 10.1.1.100，但是由于 GLBP 中两个组成员的虚拟 MAC 分别为 0001.0001.0001 和 0002.0002.0002,所以虽然主机的网关 IP 都是相同的，只要他们映射的 10.1.1.100 的 MAC 不一样，即可发送到不同的 GLBP 成员。现在 R3 学习到 10.1.1.100 的 MAC 为 0002.0002.0002，所以最终它发向 10.1.1.100 的数据包将被 R2 接收，而 R4 学习到 10.1.1.100 的 MAC 为 0001.0001.0001，所以最终它发向 10.1.1.100 的数据包将被 R1 接收。这样因为 GLBP 中组成员的 IP 虽然相同，但 MAC 不相同，所以能够为用户实现负载均衡。

GLBP 组最多可以有 024 virtual routers 和四个虚拟 MAC 地址，主的为活动网关，其它都停在 listen state 状态，可以定义优先级来竞选：1 到 255 ，然后是 higher IP address

GLBP 的抢夺模式是关闭的。一台路由器转发用户流量的多少，可以利用 weighting 来控制，也可以 track 一个接口，接口失效就降低相应权重。

组内所有成员必须配置相同的组号，并且如果是 VLAN 不同，那么组号也必须是不同的。

配置



1. 配置 GLBP

说明：配置 R1，R2，R3 为 GLBP 组成员

（1）配置 R1（并配置为 AVG，即优先级最高）

R1(config)# interface fastethernet 0/0

R1(config-if)#ip add 10.1.1.1 255.255.255.0

R1(config-if)# glbp 10 ip 10.1.1.100

R1(config-if)# glbp 10 priority 200 配置优先级为 200，使其成为 AVG，默认 100。

R1(config-if)# glbp 10 preempt

（2）可选配置

R1(config-if)#glbp 10 weighting 200 配置权重，影响分担流量

R1(config-if)#glbp 10 weighting track 1 decrement 30 配置监控信息

(3)配置 R2

R2(config)# interface fastethernet 0/0

鸿鹄培训+考试群：301283867

R2(config-if)#ip add 10.1.1.2 255.255.255.0

R2(config-if)# glbp 10 ip 10.1.1.100

R2(config-if)# glbp 10 preempt

(4)配置 R3

R3(config)# interface fastethernet 0/0

R3(config-if)#ip add 10.1.1.3 255.255.255.0

R3(config-if)# glbp 10 ip 10.1.1.100

R3(config-if)# glbp 10 preempt

2. 查看结果：

（1）查看 AVG（R1）的状态

r1#sh glbp

FastEthernet0/0 - Group 10

State is Active

2 state changes, last state change 00:05:52

Virtual IP address is 10.1.1.100

Hello time 3 sec, hold time 10 sec

Next hello sent in 1.910 secs

Redirect time 600 sec, forwarder time-out 14400 sec

Preemption enabled, min delay 0 sec

Active is local

Standby is 10.1.1.3, priority 100 (expires in 9.366 sec)

Priority 200 (configured)

Weighting 100 (default 100), thresholds: lower 1, upper 100

Load balancing: round-robin

There are 3 forwarders (1 active)

Forwarder 1

State is Active

1 state change, last state change 00:05:42

MAC address is 0007.b400.0a01 (default)

Owner ID is 0013.1a2f.0680

Redirection enabled

Preemption enabled, min delay 30 sec

Active is local, weighting 100

Forwarder 2

State is Listen

MAC address is 0007.b400.0a02 (learnt)

Owner ID is 0012.da88.3560

Redirection enabled, 599.563 sec remaining (maximum 600 sec)

Time to live: 14399.563 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.2 (primary), weighting 100 (expires in 9.559 sec)

Forwarder 3

State is Listen

MAC address is 0007.b400.0a03 (learnt)

Owner ID is 0013.19f8.8be0

Redirection enabled, 598.625 sec remaining (maximum 600 sec)

Time to live: 14398.625 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.3 (primary), weighting 100 (expires in 8.625 sec)

r1#

说明：可以看出 R1 状态为 Active，即自己为 AVG，且自己的虚拟 MAC 为：0007.b400.0a01，其它 AVF 的 MAC 分别为 0007.b400.0a02 和 0007.b400.0a03。

（2）查看 R2

r2#

r2#sh gl

r2#sh glbp

FastEthernet0/0 - Group 10

State is Listen

Virtual IP address is 10.1.1.100

Hello time 3 sec, hold time 10 sec

Next hello sent in 1.105 secs

Redirect time 600 sec, forwarder time-out 14400 sec

Preemption enabled, min delay 0 sec

Active is 10.1.1.1, priority 200 (expires in 9.715 sec)

Standby is 10.1.1.3, priority 100 (expires in 7.171 sec)

Priority 100 (default)

Weighting 100 (default 100), thresholds: lower 1, upper 100

Load balancing: round-robin

There are 3 forwarders (1 active)

Forwarder 1

State is Listen

MAC address is 0007.b400.0a01 (learnt)

Owner ID is 0013.1a2f.0680

Time to live: 14399.711 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.1 (primary), weighting 100 (expires in 9.711 sec)

Forwarder 2

State is Active

1 state change, last state change 00:02:55

MAC address is 0007.b400.0a02 (default)

Owner ID is 0012.da88.3560

Preemption enabled, min delay 30 sec

Active is local, weighting 100

Forwarder 3

State is Listen

MAC address is 0007.b400.0a03 (learnt)

Owner ID is 0013.19f8.8be0

Time to live: 14397.712 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.3 (primary), weighting 100 (expires in 7.604 sec)

r2#

说明：从上可以看出 R2 的状态为 Listen，因为还有个 R3，虽然优先级相同，但对方 IP 地址比自己大，所以对方状态是 Standby,即为 AVG 的备份。

（3）查看 R3（AVG 的备份）

r3#sh glbp

FastEthernet0/0 - Group 10

State is Standby

1 state change, last state change 00:03:01

Virtual IP address is 10.1.1.100

Hello time 3 sec, hold time 10 sec

Next hello sent in 1.750 secs

Redirect time 600 sec, forwarder time-out 14400 sec

Preemption enabled, min delay 0 sec

Active is 10.1.1.1, priority 200 (expires in 8.294 sec)

Standby is local

Priority 100 (default)

Weighting 100 (default 100), thresholds: lower 1, upper 100

Load balancing: round-robin

Group members:

0012.da88.3560 (10.1.1.2)

0013.19f8.8be0 (10.1.1.3) local

0013.1a2f.0680 (10.1.1.1)

There are 3 forwarders (1 active)

Forwarder 1

State is Listen

MAC address is 0007.b400.0a01 (learnt)

Owner ID is 0013.1a2f.0680

Time to live: 14398.290 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.1 (primary), weighting 100 (expires in 7.304 sec)

Forwarder 2

State is Listen

MAC address is 0007.b400.0a02 (learnt)

Owner ID is 0012.da88.3560

Time to live: 14398.690 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.2 (primary), weighting 100 (expires in 8.690 sec)

Forwarder 3

State is Active

1 state change, last state change 00:03:09

MAC address is 0007.b400.0a03 (default)

Owner ID is 0013.19f8.8be0

Preemption enabled, min delay 30 sec

Active is local, weighting 100

r3#

3. 测试结果

（1）在 R4 上通过 telnet 10.1.1.100，看最终的结果是登陆哪里：

r4#telnet 10.1.1.100

Trying 10.1.1.100 ... Open

r1>

说明：可以看出，结果是 telnet 上 R1。

（2）查看 10.1.1.100 的 ARP

r4#sh arp

Protocol Address Age (min) Hardware Addr Type Interface

Internet 10.1.1.4 - 00e0.1e60.7c86 ARPA Ethernet0

Internet 10.1.1.100 0 0007.b400.0a01 ARPA Ethernet0

r4#

说明：可以看出，10.1.1.100 的 MAC 为 0007.b400.0a01，即为 R1（AVG）的 MAC

（3）清除 ARP，再次测试 telnet 10.1.1.100 的结果，因为 GLBP 中有三个成员，所以会负载，这次应该 telnet 到第二台了。

r1#clear arp-cache

r1#

r4#telnet 10.1.1.100

Trying 10.1.1.100 ... Open

r2>

说明：可以看出，已经不再是到 R1，而是到 R2，说明负载成功。

r4#sh arp

Protocol Address Age (min) Hardware Addr Type Interface

Internet 10.1.1.2 0 0012.da88.3560 ARPA Ethernet0

Internet 10.1.1.4 - 00e0.1e60.7c86 ARPA Ethernet0

Internet 10.1.1.100 0 0007.b400.0a02 ARPA Ethernet0

r4#

说明：查看 ARP 表也发现，第二次得到 10.1.1.100 的 MAC 为 0007.b400.0a02，说明 GLBP 中的组成员是为用户分担数据的。

（4）清除 ARP，再测试一次：

r1#clear arp-cache

r1#

r4#telnet 10.1.1.100

Trying 10.1.1.100 ... Open

R3>

说明：看到结果 telnet 到了 R3，说明 GLBP 中负载成功。

4. 配置 GLBP 认证

说明：认证由于 IOS 版本的原因，可能支持明文和 MD5 认证，可使用直接输入密码，也可使用 key chain 的方法。

（1）为 R1 配置 GLBP 认证密码：

r1(config-if)#glbp 10 authentication text cisco 密码为 cisco

（2）为 R2 配置 BLGP 认证密码

R2(config-if)#glbp 10 authentication text ccie 密码为 ccie,密码同其它不一样

（3）为 R3 配置 BLGP 认证密码

R3(config-if)#glbp 10 authentication text cisco 密码为 cisco

5.查看 GLBP 认证后的结果

说明：由于 R2 配置的密码与其它成员和 AVG 不一样，查看时会发现 R2 不在此组中。

（1）在 R3 上查看 GLBP 组状态

```
r3#sh glbp
FastEthernet0/0 - Group 10
  State is Standby
   4 state changes, last state change 00:00:42
  Virtual IP address is 10.1.1.100
  Hello time 3 sec, hold time 10 sec
   Next hello sent in 2.595 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
```

Authentication text "cisco"

Preemption enabled, min delay 0 sec

Active is 10.1.1.1, priority 200 (expires in 8.597 sec)

Standby is local

Priority 100 (default)

Weighting 100 (default 100), thresholds: lower 1, upper 100

Load balancing: round-robin

Group members:

0013.19f8.8be0 (10.1.1.3) local

0013.1a2f.0680 (10.1.1.1)

There are 3 forwarders (2 active)

Forwarder 1

State is Listen

2 state changes, last state change 00:00:52

MAC address is 0007.b400.0a01 (learnt)

Owner ID is 0013.1a2f.0680

Time to live: 14397.055 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.1 (primary), weighting 100 (expires in 7.055 sec)

Forwarder 2

State is Active

1 state change, last state change 00:00:44

MAC address is 0007.b400.0a02 (learnt)

Owner ID is 0012.da88.3560

Time to live: 14311.038 sec (maximum 14359 sec)

Preemption enabled, min delay 30 sec

Active is local, weighting 100

Forwarder 3

State is Active

1 state change, last state change 00:08:12

MAC address is 0007.b400.0a03 (default)

Owner ID is 0013.19f8.8be0

Preemption enabled, min delay 30 sec

Active is local, weighting 100

r3#

SLA （Service Level Agreements）

概述

SLA 这个技术，算是一个非常实用，功能相当广泛的技术，但是 SLA 许多功能，许多结果，需要特定的软件才能读懂，需要特定的人才能分析。那么我们就介绍一些我们能够读懂，我们能够分析的功能。

如果有一个人在上海，他从未去过南京，但是他现在想要测试从上海到南京的路程上要花多少时间，很显然，只要有人亲自从上海去一趟南京，这个时间就能够计算出来，并且，从上海往南京多跑几趟，再算个平均值，那么这样计算出来的时间将更为理想。正好这个想测试从上海到南京之间的路程要花多少时间的人，听说

上海有朋友要去南京买东西，这样他就抓住了一个测试的机会，他就去跟朋友说这个事，比如他的朋友 10 点从上海出发，那么这个人记下了出发时间，然后等到他朋友从南京回到上海后，这个人记下回到上海的时间是 15 点，如果这个人把 15 点减去 10，等于 5,然后再除以 2，结果为 2.5，那么他认为从上海到南京的路程要花 2.5 个小时，这样计算的结果是否准确呢。很明显，这样计算出来的路程是很马虎的，也是不可靠的，因为可能他的朋友在南京买东西就花去了一个小时，也就是说他的朋友从上海去南京再回来，路程上花掉的时间是 4 个小时，而不是 5 个小时，中间的路程时间就应该是 4 小时除以 2，结果等于 2 小时，所以按之前的算法算出 2.5 小时是错误的。那么又怎样才能正确计算出路程应该是 2 小时呢，这也很明显，那就是在他朋友 10 点从上海出发时，记下这个出发时间，然后他朋友到达南京时，应该是 12 点，他的朋友也应该把这个时间记下来，等他朋友在南京买到东西，准备离开南京时，应该是 13 点，那么这个时间也应该做记录，最后回到上海是 15 点，再记下这个到达时间，最后得出来的，才是真实的路程时间。所以中间买东西的误差，应该用科学的方法去除。

以上是生活的真实例子，如果拿到网络中，用来测试我们网络上从一个主机到另外一个主机之间的网络速度，又应该是什么样的方法呢，又应该用什么样的方法好呢？比如，要测试从一台主机到目标为 cisco.com 的主机，中间跨越了许多网络设备，我们想要测试从这台主机到 cisco.com 之间网络中的传输该用去多少时间，中间的网络性能到底如何，我们该用什么样的方法呢？假如我们用常用的方法“ping”,我们从这台主机去 ping cisco.com，结果看到花去的时间为 20ms，那么我们认为从这台主机到 cisco.com 中间的网络需要 20ms 除以 2 等于 10ms，那我们就真的错了，因为我们算出的这 10ms，根本就不是中间网络需要使用的时间，也不能因此而认为网络的性能是差的。也许 cisco.com 这台主机正处于繁忙状态，或者是 CPU 负荷不够，CPU 处理 ICMP 数据需要 10ms，那么这样一来，我们应该计算出，两台主机之间的网络传输数据需要多少时间呢？很明显，应该是来回的时间 20ms 减去 CPU 繁忙时处理 ICMP 的时间 10ms，结果为剩下的 10ms 除以 2，应该为 5ms，所以我们之前并没有算出网络的时间和性能。那么我们在测试网络时间和性能的时候，能不能像上面测试上海到南京路程那样，在分别出发的时候，下记时间，然后让朋友在南京到达和出发时，也记下时间呢。网络设备能不能这样为我们记下时间呢，答案是不能。既然不能，那就说明我们不能准确的计算网络延时，这个时候，如果要让网络设备像我们测试路程记录时间一样来测试网络，就可以利用 SLA 这个技术，SLA 正是在我们测试网络中两个节点之间的数据时，为数据记录下我们需要的重要参数，这就是 SLA 完成的，我们在发数据从设备上发出去，只要我们正确配置 SLA，便可记录参数，但是，数据到达目标之后，目标是否又能协助我们添加重要参数呢？如果需要对方这样做，就必须将对方配置为 SLA responder。需要说明的是，SLA 在测试网络的时候，用到的数据可以由我们任意定义，包括数据的协议，端口，并且这些测试的数据，是程序在后台发起的，不需要我们人工干预，我们需要做的只是定义数据类型，和数据发起时间，以及频率，

其它的，交给后台处理。

配置

根据图 1 中的拓朴，来测试 R1 到 R3 之间的网络延迟和性能，按照理论，我们需要在 R1 上配置好 SLA，并且需要将 R3 配置成为 SLA responder,否则结果有如掩耳盗铃。

1. 配置 R1 的 SLA

（1）定义 SLA 的数据类型，发起时间，频率

注：定义会话号码 1：

R1(config)#ip sla monitor 1

（2）定义数据类型为 TCP 23，(其它协议也可用，但需网络允许)，源端口可不配：

R1(config-sla-monitor)#type tcpConnect dest-ipaddr 13.1.1.3 dest-port 23 source-ipaddr 13.1.1.1**（3）定义频率为 60 秒一次：**

R1(config-sla-monitor-tcp)#frequency 60

定义会话发起时间为现在，并且在没有人工干预的情况下永不停止：

R1(config)#ip sla monitor schedule 1 start-time now life forever

2.配置目标为 SLA responder

r3(config)#ip sla monitor responder

3. 检查测试结果：

（1）查看 R1

R1#sh ip sla monitor statistics

Round trip time (RTT) Index 2

Latest RTT: 8 ms

Latest operation start time: *01:25:07.252 UTC Fri Mar 1 2002

Latest operation return code: OK

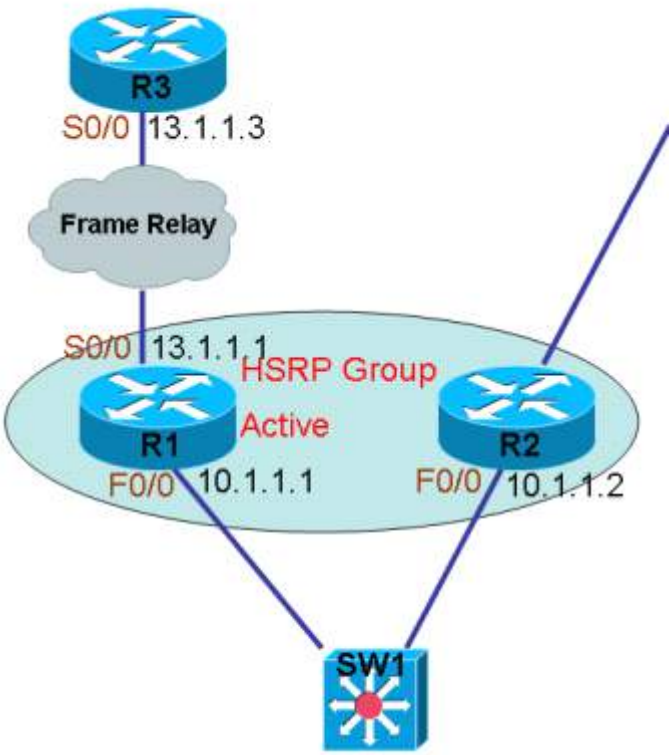
Number of successes: 1

Number of failures: 0

Operation time to live: Forever

R1#

说明：从结果可以看出，测试成功次数为 1 次，并且可以看到时间为 8ms，如果对方不配 SLA，将会失败。



来看图 2，在 LAN 中，R1 和 R2 为 HSRP，为 LAN 中主机提供网关备份，我们配置 R1 为主网关，即 Active,R1 负责将 LAN 中主机的数据包从 S0/0 送出去，如果当 R1 的 S0/0 失效之后，理所当然主网关应该变成 R2，由 R2 为 LAN 内的主机提供数据转发。在通常情况下，我们配置 R1 的 HSRP 时，可以监控 S0/0 的接口状态，当接口 down 掉之后，降低自身优先级而后让出 Active 的角色。虽然我们已经为 HSRP 监控了接口 S0/0 的状态，可是当 S0/0 与对端网络失去连接后，R1 并不会因此而降低优先级来让出 Active 的角色，因为 S0/0 出去是帧中继网络，就算对方接口有什么变化，因为 S0/0 是和帧中继交换机相连的，所以 S0/0 接口没有出现接口 down 的状态，从而导致 R1 的 S0/0 已经不能通信了，但自身还是 HSRP 的 Active 状态，而导致用户不能与外界通信。

最理想的做法就是，R1 不能单纯的只监控 S0/0 接口的状态，不能听认定接口状态 down 了，网络才失去连接，而应该主动去尝试是否可以从 S0/0 发送数据包出去得到远程主机的回应，如果数据包从 S0/0 发出去，并无主机回应，那么就认为 S0/0 已不能与外界通信，这时就应该降低优先级来让出 Active 的角色。那么要怎样才能测试从 S0/0 发数据包出去测试与远程主机的连通性呢，在本图中，我们完全可以利用之前的 SLA，让 SLA 从 R1 向 13.1.1.3 发出 ICMP 数据包，正常情况下是应该得到对方的回应的(必须在对方放开 ICMP 的通信)，如果 13.1.1.3 不对 ICMP 作出回应，我们便认为 S0/0 已失去网络连通性。这样的组合，Cisco 称为 Object Tracking,即基于目标的跟踪。

配置

1. 配置 Object Tracking：

（1）配置到 13.1.1.3 的 SLA，并利用 ICMP 作为数据协议：

r1(config)#ip sla monitor 5

r1(config-sla-monitor)#type echo protocol ipIcmpEcho 13.1.1.1 source-ipaddr 13.1.1.3

r1(config-sla-monitor)#exit

r1(config)#ip sla monitor schedule 5 start-time now life forever

（2）配置 Track 组:

r1(config)#track 10 rtr 5 10 为 track 组号码，5 为 SLA 会话号

2. 将 Track 组应用到 HSRP：

（1）在接口应用 track 组 10

r1(config)#int f0/0

r1(config-if)#standby 1 track 10

3. 检查结果

说明：如果 R1 能 ping 通 13.1.1.3，就说明 Track 组状态是 up 的，那么 HSRP 就不会降低优先级，只要 ping 不通 13.1.1.3，Track 组状态 down 了，HSRP 就会降低优先级成为 Standby 状态。

（1）检查 SLA 状态

r1#sh ip sla monitor statistics

Round trip time (RTT) Index 5

Latest RTT: 35 ms

Latest operation start time: *01:54:30.942 UTC Fri Mar 1 2002

Latest operation return code: OK

Number of successes: 25

Number of failures: 0

Operation time to live: Forever

说明 SLA 正常

（2）检查 Track 组状态

r1#sh track

Track 10

Response Time Reporter 5 state

State is Up

1 change, last change 00:04:04

Latest operation return code: OK

Latest RTT (millisecs) 36

Tracked by:

HSRP FastEthernet0/0 1

r1#

说明：Track 组正常，则 HSRP 角色正常。

（3）检查 HSRP 角色：

r1#sh standby

FastEthernet0/0 - Group 1

State is Active

4 state changes, last state change 00:00:07

Virtual IP address is 10.1.1.100

Active virtual MAC address is 0000.0c07.ac01

Local virtual MAC address is 0000.0c07.ac01 (v1 default)

Hello time 3 sec, hold time 10 sec

Next hello sent in 1.950 secs

Preemption enabled

Active router is local

Standby router is unknown

Priority 105 (configured 105)

Track interface Serial0/0 state Up decrement 10

Track object 10 state Up decrement 10

IP redundancy name is "hsrp-Fa0/0-1" (default)

r1#

说明：HSRP 角色正常。

4. 测试 SLA 状态失效后，HSRP 的反应：

（1）测试 R1 到对端 ICMP 断开

r1#sh ip sla monitor statistics

Round trip time (RTT) Index 5

Latest RTT: NoConnection/Busy/Timeout

Latest operation start time: *01:58:50.941 UTC Fri Mar 1 2002

Latest operation return code: Timeout

Number of successes: 50

Number of failures: 1

Operation time to live: Forever

r1#

说明：可以看出 ICMP 失败了

（2）查看 track 组状态：

r1#sh track

Track 10

Response Time Reporter 5 state

State is Down

2 changes, last change 00:00:37

Latest operation return code: Timeout

Tracked by:

HSRP FastEthernet0/0 1

r1#

说明：由于 SLA 失效，所以 Track 组也 down 了，那么 HSRP 就应该降低优先级而改变角色身份 。

（3）查看 HSRP 角色状态：

r1#sh standby

FastEthernet0/0 - Group 1

State is Standby

```
  3 state changes, last state change 00:01:18
 Virtual IP address is 10.1.1.100
 Active virtual MAC address is 0000.0c07.ac01
  Local virtual MAC address is 0000.0c07.ac01 (v1 default)
 Hello time 3 sec, hold time 10 sec
  Next hello sent in 2.083 secs
 Preemption enabled
 Active router is 10.1.1.2, priority 100 (expires in 8.065 sec)
 Standby router is local
 Priority 95 (configured 105)
  Track interface Serial0/0 state Up decrement 10
  Track object 10 state Down decrement 10
 IP redundancy name is "hsrp-Fa0/0-1" (default)
r1#
```

说明：可以看出 HSRP 已将优先级从 105 降到 95，从 Active 降到了 Standby 状态。

除加 Track 组额外功能

我们不仅可以在 HSRP 中应用 Track，利用 Track 的状态来做出自己的决定。有时我们可以在其它方面也利用 Track 组的功能，比如我们写路由的时候，某条路由也可以调用 Track 组的状态，当 Track 状态 down 后，相应的路由也消失。下面我们以静态路由为例，来看看当 Track 组状态 down 后，静态路由消息的例子。

1. 监控接口的路由能力

说明：监控某个接口是否能够传送数据包，如果没有传送数据包的能力，那么 Track 组状态将 down，也就是我们把接口的 IP 地址去掉，Track 组即 down

（1）配置 Track 接口

r1(config)#track 1 interface s0/0 ip routing

（2）去掉 S0/0 的接口 IP

r1(config)#int s0/0

r1(config-if)#no ip address

（3）查看接口 IP

r1#sh ip int s0/0

Serial0/0 is up, line protocol is up

Internet protocol processing disabled

r1#

（4）查看 Track 组状态：

r1#sh track

Track 1

Interface Serial0/0 ip routing

IP routing is Down (ip disabled, no ip addr)

1 change, last change 00:03:09

说明：从上可以看出，接口没有数据传递功能后，Track 组变成 Down 状态。

如果被静态路由调用，那么相应静态路由将消失。

2 监控接口 line-protocol 状态

鸿鹄培训+考试群：301283867

说明：如果接口的 line-protocol 为 down，那么 Track 组即 down，只要 line-protocol 是 up 的，不管有没有接口 IP 地址，不管是否有路由能力，track 组的状态都为 up。

（1）配置 track 接口的 line-protocol

r1(config)#track 2 interface s0/0 line-protocol

（2）查看接口信息：

r1#sh ip interface s0/0

Serial0/0 is up, line protocol is up

Internet protocol processing disabled

r1#

说明：可以看出接口 S0/0 line protocol 是 up 的，虽然没有路由能力

（3）查看 Track 组状态：

r1#sh track

Track 2

Interface Serial0/0 line-protocol

Line protocol is Up

1 change, last change 00:01:25

说明：可以看出，Track 组此时只关心接口的 line-protocol 状态，并不状态接口的路由能力。

3 监控路由表中路由条目

说明：监控路由表中是否有某些路由，如果被监控的路由条目在路由表中存在，则 Track 组 UP，被监控路由消失，则 Track 组 down:

（1）写出一条被监控路由作测试

鸿鹄培训+考试群：301283867

r1(config)#ip route 100.1.1.0 255.255.255.0 s0/0

（2）将 100.1.1.0/24 作为监控目标：

r1(config)#track 3 ip route 100.1.1.0/24 reachability

（3）添加路由

说明：最后写一条静态路由，如果 100.1.1.0 这条路由消失（也就是 Track 组 down），则自己也跟着消失：

r1(config)#ip route 30.1.1.0 255.255.255.0 s0/0 track 3

（4）查看是否路由都存在：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

100.0.0.0/24 is subnetted, 1 subnets

S 100.1.1.0 is directly connected, Serial0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial0/0

30.0.0.0/24 is subnetted, 1 subnets

S 30.1.1.0 is directly connected, Serial0/0

r1#

说明：可以看出，自身路由 30.1.1.0/24 是存在的，以及 100.1.1.0/24 也是存在的。

（5）查看 track 组状态

r1#sh track

Track 3

IP route 100.1.1.0 255.255.255.0 reachability

Reachability is Up (static)

3 changes, last change 00:00:04

First-hop interface is Loopback3

Tracked by:

STATIC-IP-ROUTING 0

（5）测试 100.1.1.0/24 消失

r1(config)#no ip route 100.1.1.0 255.255.255.0 s0/0

（6）查看 Track 组状态：

r1#sh track

Track 3

IP route 100.1.1.0 255.255.255.0 reachability

Reachability is Down (no route)

2 changes, last change 00:00:09

First-hop interface is unknown

Tracked by:

STATIC-IP-ROUTING 0

说明：100.1.1.0/24 消失，track 组也消失。

（7）查看路由 30.1.1.0/24

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial0/0

r1#

说明：因为路由 100.1.1.0/24 消失，Track 组也 down，所以 30.1.1.0/24 也从路由表中消失。

NTP（网络时间协议）

概述

某些时候，我们需要在 Cisco 设备上做一些基于时间的策略或访问控制，让这些策略或控制在特定的时间内生效，所以设备上必须存在着准确的时间。但是如果手工给设备配好时间，当设备因为某些原因重启后，时间将被刷新到出厂时的时间，这样就影响到我们所做的策略或服务。这时我们就需要设备能够借助于远程时间服务器上的时间来同步自己的本地时间，让设备在正常工作时，本地的时间和远程时间服务器的时间保持一致。

本地设备的时间和远程时间服务器即使能够同步，也会存在毫秒级的误差，如果自己和远程时间服务器同步，那么别人再和自己同步，就意味着别人的时间误差可能更大。在这里，时间的精准度就会有高低，Cisco 设备的 NTP 把这样的精准度高低称为 stratum，如果 stratum 值越大，就表示精准度越差，stratum 值越小表示精准度就越好。比如远程一台时间服务器的 stratum 是 2，本地设备和它同步后，自己的 stratum 就是 3，精准度就差了一些，如果这时别的设备再和自己同步，那么它得到的 stratum 就是 4，精准度就意味着更差。

Cisco 设备即可以做为 NTP 客户端，即自己和远程时间服务器同步，也可作为 NTP 服务器，即向别的设备提供自己的时间，让别的设备和自己的时间同步，如果将 Cisco 设备作为 NTP 服务器，默认的 stratum 是 8，就表示远程设备和自己同步后，stratum 就是 9。

时间和时区

如果一台 Cisco 设备需要做 NTP 时间服务器，就得先为自己配上时间，但是，因为设备可能出现在全球的任何一个国家，而可能各个国家的时区是不一样的，所以这时还需要为设备配置时区，至于每个国家用哪个时区，不在本篇讨论范围内，请自行查阅各国相关时区和时差，中国使用东 8 区时。

配置

1. 配置时间

（1）为设备配置时区：

R1(config)#clock timezone GMT +8 配置时区为东 8 区时

（2）为设备配置时间：

r1#clock set 20:00:00 1 oct 2008 配置时间为 2008 年 10 月 1 日 20 点整

注：此时间为东 8 区时 2008 年 10 月 1 日 20 点整，如果将时区更新，设备会自行计算时差将时间调整到对应时区的时间。

2. 查看结果：

（1）r1#show clock

2. 配置 NTP

（1）配置 NTP 服务器：

注：配置 master 和 stratum(默认为 8)

R1(config)#ntp master 5 stratum 为 5

（2）配置 NTP 数据包的源地址：

注：此地址为数据发出时的源地址，并不影响 NTP 时间同步

R1(config)#ntp source Loopback0

3.配置 NTP Client

（1）指定 NTP 服务器地址

R2(config)# ntp server 10.1.1.1

(2)配置 clock timezone

R2(config)# clock timezone GMT +8

鸿鹄培训+考试群：301283867

（3）查看结果：

R2#show clock

说明：看本地时间和服务器的时间是否一致。

NTP 认证

服务器和客户端之间可以使用 MD5 来提供安全认证，只有双方在密码相同的情况下，时间才能同步，双方可以同时配置多个 key，号码也可以不一样，但当前使用的的 key 密码必须是相同的，否则时间不能同步。

配置

1. 配置 NTP 服务器：

（1）开启认证

R1(config)# ntp authenticate

(2)配置密码

R1(config)# ntp authentication-key 5 md5 cisco

(3)使用某个密码

注：在 key 只有一个的情况下，可以不配

R1(config)#ntp trusted-key 5

2. 配置 NTP Client;

(1) 开启认证

R2(config)# ntp authenticate

（2）配置密码

R2(config)# ntp authentication-key 20 md5 cisco

（3）使用某个密码

R2(config)#ntp trusted-key 20

（4）打开对服务器的密码使用，让发送给服务器的数据中携带密码

R2(config)#ntp server 10.1.1.1 key 20

3.查看结果

（1）未同步的：

R2#sh ntp status

Clock is unsynchronized, stratum 16, no reference clock

R2#show ntp association detail

12.0.0.1 configured, insane, invalid, unsynced, stratum 16

（2）已同步的：

R2#sh ntp status

Clock is synchronized, stratum 6, reference is 127.127.7.1

r2#sh ntp associations detail

10.1.1.1 configured, authenticated, our_master, sane, valid, stratum 6

Summer-time 夏令时

概述

众所周知，在冬天，如果某个地方是早上 7 点钟天亮，晚上是 6 点钟天黑，那么到了夏天，这个地方就很可能是早上 5 点钟就天亮了，并且晚上 7 点才天黑，如果人们一般是 7 点半起床，9 点上班的话，那么可以相象，人们绝对可以在夏天的时候 6 点半起床，甚至可以更早，因为天早已经亮了，人们可以大大利用白天的时间。 但是如果因为到了夏天而让人们的学习和工作提前一个小时，可能会让人产生厌恶情绪。因此，就利用调时钟的方法，在夏天到来的时间，将时间往前拨快一小时，也就是时间明明还是早上 5 点，咱们就把时钟调到 6 点，大家到了夏天都遵守这个规则，这样一来，大家夏天就可以每天多利用一小时。但是到了冬天，时钟还是应该回归到夏天之前的时间，也就是时间往后拨慢 1 小时。但是这钟时钟拨快和拨慢，当然并不是人们手工去执行，而只要时钟支持这种夏令时机制，就会自行在相应的季节里调整自己的时间。

那么究竟在什么样的一个时间范围内，才需要让时间调快一小时呢，这段时间范围就称为夏令时范围。中国已经没有这样的机制，而现在 Cisco 所在的美国夏令时范围是从每年 3 月的第二个星期日，到 11 月的第一个星期日。这样的夏令时范围，在 Cisco 设备上，可以随意配置，当时钟进入夏令时范围，时间会马上往前跳过一小时，比如夏令时范围是 3 月 1 日早上 9 点，那么时间到了 3 月 1 日早上 9 点，下一分钟就不再是 9 点零 1 分，而应该是 10 点零 1 分。在结束的时候，如果是 10 月 1 日早上 9 点结束，那么到了 10 月 1 日早上 9 点，下一分钟就不再是 9 点零 1 分，而应该是 8 点零 1 分。

配置

1.配置夏令时

注：在配置之前，应该配好正确的时区和时间

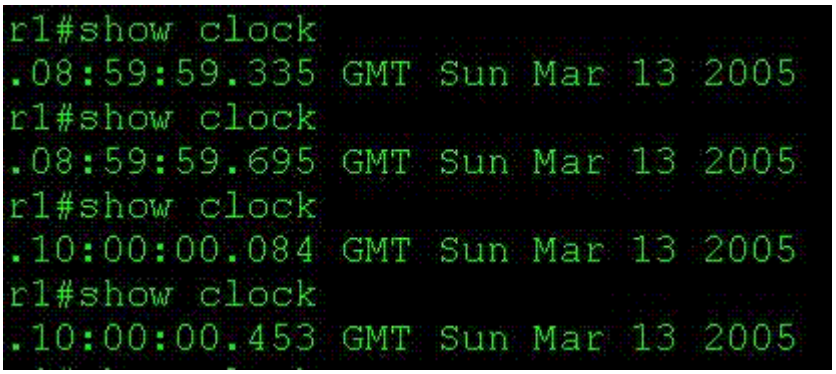
（1）配置夏令时范围

r1(config)#clock summer-time GMT recurring 2 sun march 9:00 2 sun nov 9:00

说明：夏令时范围从 3 月的第二个周日上午 9:00 到 11 月第二个周日的上午 9:00

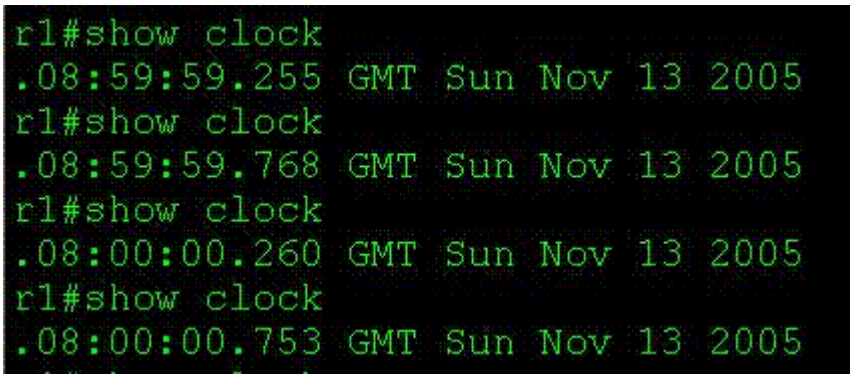
2 查看结果：

（1）查看进入夏令时的时间变动



说明：从上面可以看出，时间从 8:59 过了之后，按正常情况应该是 9:00，可是因为受进入夏令时的影响，时间直接加快一小时，跳到了 10:00

（2）查看结束夏令时的时间变动



说明：从上面可以看出，时间从 8:59 过了之后，按正常情况应该是 9:00，可是因为受结束夏令时的影响，时间直接变慢一小时，跳到了 8:00。

Syslog and local logging

Local logging

在 Cisco 设备上往往会发生许多事件，有些是我们预料中的，是我们自己配置

的，有些是无法预料的故障。如果当我们使用中的设备出现某些故障时，我们需要知道设备究竟发生了哪些事件，或者曾经发生过哪些事件，以便我们能够排错故障，这时，就需要设备有记录事件的功能。Logging 功能就能够帮助我们记录设备上发生的大大小小的事件。

设备上所发生的事件是有等级之分的，有时我们并不需要设备将任何事件都记录下来，那么我们就可以随意设定哪种级别的事件是我们需要记录的，哪些是不需要记录的。Cisco 将设备上的事件级别分为 0 到 7 总共 8 个级别，分别是 0 级代表最严重的事件，0 (emergencies)， 1(alerts)，2(critical)，3(errors)，4，(warnings) ，5 (notifications)， 6(informational) ，

7 (debugging) ，0 是最严重的，如果我们指定记录 5 级别的，那么 0 到 5 级别的会全部记录。

我们可以将事件记录到设备本地存储器中，称为 buffered，也可以将事件记录到远程服务器中。如果要记录到远程服务器，远程服务器的空间大小在 Cisco 设备上是无法控制的，但是如果将事件记录在设备本地存储器，就可以定义存储器的空间大小 ，默认为 4096 bytes。存放在本地存储器时，我们查看的时候，事件的排列是由上往下的，上面的日志是最老的，也就是最早发生的，最下面的日志是最新的，如果存储器满了，那么最新的日志将替换最老的日志。

当设备出现故障时，我们需要查找曾经发生的件事和原因，有时要根据事件发生的时间来判断，那么在设备记录事件时，就应该为每条发生的件事打上相应的时间戳。在默认情况下，设备会为事件写上发生的时间，但是这个时间并不是设备本地设置的时间，这样也就不利于我们排错，所以，我们需要手工告诉设备将时间戳写成本机的时间。

在我们登陆到设备上进行配置的时候，如果引起设备发生一些事件，我们希望设备能够给我们一些反馈信息，也就是能跳出一个提示信息。因为我们可以通过 console 登陆设备，也可以 telnet 登陆，但设备默认是 console 登陆设备的是能够看见日志提示的，而 telnet 登陆的是无法看见的，如果要让 telnet 方式登陆也看见日志信息，也需要手工进行配置。

配置

1. 配置 logging:

（1）开启 logging 服务（默认是开启的）

鸿鹄培训+考试群：301283867

r1(config)#logging on

（2）开启本地记录功能

r1(config)#logging buffered

（3）定义本地存储器的空间大小（单位 Byte）

r1(config)#logging buffered 9090

(4)定义可以存储的日志级别（默认 7 debugging）

r1(config)#logging buffered errors 改为记录日志的级别为 3 errors

(5)定义在 console 下面是否弹出日志提示(默认开启)

r1(config)#logging console 定义 console 下弹出日志

（6）定义 telnet 方式登陆设备时也能看到弹出日志（默认关闭）

r1#terminal monitor

（7）定义能够传到 telnet 下的事件等级（默认为 7 debugging）

r1(config)#logging monitor errors 定义能够传到 telnet 下的事件等级为 3 errors

（8）定义时间戳（默认非本地时间）

r1(config)#service timestamps log datetime msec show-timezone localtime

注：在记录时间戳时，精确到毫秒级，并且使用本地时间，附带时区信息。

（9）查看日志：

r1#sh logging

syslog

有时我们需要将设备发生的事件记录到远程服务器上，这时就需要为本地设备

定义远程服务器的 IP 地址，并且在远程服务器上，需要知道远程的存储类型，称为 facility，要事先了解清楚，通常我们使用的有 local0 local1 local2 local3 local4 local5 local6 local7 共 8 个类型，默认为 local7。

配置

1. 配置远程记录日志

（1）配置远程服务器地址

r1(config)#logging host 10.1.1.1 定义远程服务器 IP 为 10.1.1.1

（2）定义远程存储类型(默认为 local7)

r1(config)#logging facility local6 定义远程存储类型为 local6

（3）定义发送到远程服务器事件等级（默认为 6 informational）

r1(config)#logging trap 7

FTP&TFTP

概述

在很多时候，我们需要从别的地方往 Cisco 设备上传送文件，或者从 Cisco 设备上往别的地方传送文件。在传输文件时，我们可以使用的协议有很多，在这里需要介绍的是两个传输协议，第一个是大家非常熟悉的 FTP，第二个是简单文件传输协议（TFTP）。也许 FTP 在平时用的非常多，但是到了 Cisco 设备上，并不是如此。Cisco 设备可以利用 FTP 协议从别的 FTP 服务器上往本地传送文件，也可以将自身配置成一台 FTP 服务器，为别人提供文件传送。要让 Cisco 设备成为一台 FTP 服务器，共享出自己存储器中的文件让别人拷贝，这需要特定的 IOS 支持，而 Cisco 从远程 FTP 服务器往本地拷贝文件，几乎所有的 IOS 都支持。另外一个 TFTP 协议，也几乎是所有的 IOS 都支持的协议，TFTP 和 FTP 的功能是相同的，但是 TFTP 不需要认证，而 FTP 在传送文件之前，可以采用认证。Cisco 设备可以将自己配置成一台 TFTP 服务器，共享出自己的文件和目录，也可以从远程 TFTP 服务器往本地拷贝文件。

配置

1. 配置 FTP Server

注：要配置 Cisco 设备成为一台 FTP 服务器，需要特定 IOS 支持

（1）开启 FTP server 功能：

Router(config)# ftp-server enable

（2）指定 FTP 的共享目录

如：共享出 flash 中某文件

Router(config)# ftp-server topdir flash:abc.bin

2. 配置 FTP Client

说明：此配置目的在于让 Cisco 设备能够从远程 FTP 服务器往本地拷贝文件，即配置用户名和密码

（1）配置远程 FTP 服务器用户名

Router(config)# ip ftp username ccie 配置远程 FTP 服务器的用户名为 ccie

（2） 配置远程 FTP 服务器密码

Router(config)# ip ftp password cisco 配置远程 FTP 服务器的用户名为 cisco

（3）配置本地 FTP 源 IP（可选配置）

Router(config)# ip ftp source-interface

3 配置 TFTP

说明：将 Cisco 设备配置成一台 TFTP 服务器，以共享出自己的目录和文件

（1）配置 Cisco 设备为 TFTP 服务器（需要指明文件名）

R1(config)#tftp-server flash:abc.bin

HTTP&HTTPS

概述

我们在对 Cisco 设备进行配置和管理时，可以通过 Consolde 连接上去，也可以通过 VTY 上去，还有 TTY，最后还有 AUX 拨号上去，不管我们通过以上哪种方式连上去管理，我们能看到的都是命令行界面（CLI 界面）。除了以上的管理方式外，我们还可以为设备配置图形界面的管理方式，也就是通常我们用的网页浏览器去连接设备进行管理，既然要使用网页浏览器来管理设备，那就要使用的 HTTP 协议。

要想使用网页浏览器通过 HTTP 协议对 Cisco 设备进行图形化界面的管理，这时就需要设备开启 HTTP 服务，默认是关闭的。一般情况下，我们使用的 HTTP 都是使用 TCP 端口号 80 进行连接的，这种方式的传输也只是明文的，如果我们需要让数据安全的传输，可以使用安全的 HTTP，即 HTTPS，端口号为 443，这两种 HTTP 方式，Cisco 设备是否完全支持，取决于 IOS 文件。

配置

1 配置 HTTP Server

说明：当路由器配置成 HTTP Server 后，我们就可以通过网页浏览器对设备进行管理。

（1）开启 HTTP Server(默认关闭)

R1(config)# ip http server

（2）改变 HTTP 端口号（默认 80）

R1(config)#secure-server ip http port 8080 端口号改为 8080

（3）设定 HTTP 连接的认证方式（默认为 enable 密码）

注：共支持认证方式为 aaa ， enable(默认) ， local ， tacacs，连接时，只需输入相应的认证用户信息即可

R1(config)# ip http authentication local 将认证方式改为 local （即用户名密码认证）

（4）限制访问

注：通过使用 ACL 来限制特定的 IP 地址能够通过 HTTP 连接

R1(config)# ip http access-class 20

（5）限制最大连接数，默认为 5

R1(config)# ip http max-connections 10

（6）测试连接

如测试路由器的 IP 地址为 192.168.27.66，则在 PC 网页浏览器输入 http://192.168.27.66:8080/即可看见如下认证框

输入已配置的用户名和密码即可。

（7）查看状态

R1#sh ip http server all

1. 配置 HTTPS Server

说明：以上是配置 HTTP 的方式，因为 HTTP 使用的是明文不安全的连接方式，我们可以将其改为安全的 HTTP 连接方式，即 HTTPS，在开启 HTTPS 时，必须使用 no ip http server 来关闭 HTTP，才能开启 HTTPS

（1）开启 HTTPS Server

R1(config)#ip http secure-server

其它配置基本和 HTTP 一致，不再重复。

SNMP

概述

当网络到一定规模的时候，对于网络管理员来说，要管理好这个网络，如果纯粹靠着命令行的方式去连接网络设备，去查看设备当前的状态，查看自己需要的信息，这些将变的烦琐而难以实现。在这样的环境下，很多人都在试图寻找一个网络管理软件，如果这个软件能够通过图形化的界面对设备状态进行监测，能够利用图形界面看到当前设备的各种参数，能够随时看到设备上发生的各种事件，这将大大改善管理员的工作效率。其实，这样的网络管理软件已经存在，在 Cisco 厂商主推荐的有 CiscoWorks，CiscoWorks 含有多个版本，既然一个管理软件能够图形化显示设备上的各种软硬件信息，那么在软件与设备之间就必须存在着一种交流，只有设备将自身的各种软硬件信息和状态发送给管理软件，那么管理员才能通过图形化界面看到这一切。在担任管理软件与设备之间交流的协议也就 SNMP（简单网络管理协议）。

从上面可以看出，SNMP 协议为两个组件在服务，这两个组件就是我们的网络设备和我们的网络管理软件，SNMP 正是它们之间的桥梁。要让管理软件成功的利用图形界面为我们显示出设备的一切信息，就必须正确的理解和正确的配置 SNMP。

SNMP 工作在网络设备和网络管理软件这两个组件之间，SNMP 将设备上的相关信息通告给管理软件，在配置 SNMP 时，配置了 SNMP 的网络设备被称为 SNMP 代理，而装了管理软件的主机被称为 SNMP 管理，也就是 SNMP 代理将自己的状态信息发送给 SNMP 管理，SNMP 将其整理后，通过图形界面汇报给用户，并且 SNMP 管理上的相关软件被称为 NMS（网络管理系统）。

在 SNMP 代理与 SNMP 管理之间，SNMP 代理使用 UDP 162,SNMP 管理使用 UDP 161。一个完整的 SNMP 共包含三个组件，除了以上说的 SNMP 代理和 SNMP 管理两个组件之外，还有一个组件就是 MIB，这个 MIB 的具体内容就是，之所有 SNMP 管理能够图形化显示设备的一切，正是因为收到了 SNMP 代理发来的相关数据，而 SNMP 代理的所有硬件信息和软件信息，就全部存储在 MIB 里面。SNMP 代理需要将 MIB 的内容发送给 SNMP 管理，才能对用户真正有用。

SNMP 代理将 MIB 发送给 NMS，可以使用两种数据包，这两种数据包分别是 trap 和 informs。

他们之间的区别是，当设备上发生了各种事件之后，产生的 MIB 信息由 SNMP 代理主动向 NMS 发送，是不需要 NMS 请求的，并且在将 MIB 信息发送比值 NMS 之后，NMS 即使收到了，也不需要向 SNMP 代理发送确认消息，也就是说 SNMP 代理发出去的 trap 是根本就不知道 NMS 是否收到，而自己发送完毕之后，这些 MIB 信息会马上删除，不会驻留在内存里面。

而 informs 和 trap 正好恰恰相反，当设备发生事件后，这些信息并不会主动向 NMS 通告，除非 NMS 发送 requst 来查询，然后才会发出去，但是这些已经发出去的 informs 在发送之后是会保留在内存里面的，当 NMS 收到 informs 之后必须向 SNMP 代理发送确认消息，如果不发送确认消息，SNMP 代理会重复多次发送 informs。从上可以看出 informs 具有可靠性。

网络管理员不仅可以通过 NMS 来查询设备上的各种信息，而且还可以通过 NMS 来更改设备上的配置，如果是要查询设备信息，发送的数据包被称为 get，如果是要更改设备的配置，被称为 set。

SNMP 版本

SNMP 协议目前为止分为 3 个版本，分别是 version 1，version 2c，version3。

因为网络设备上配置 SNMP 之后，就可以利用 NMS 软件对设备进行查看和修改配置，但是为了安全性，所以在 SNMP 代理与 SNMP 管理之间必须存在着某种安全机制，这些安全机制，在各个版本上的实现是不一样的。

版本 1：版本 1 在 SNMP 代理与 SNMP 管理之间提供的安全机制是使用密码的方式，只有拥有相应密码的 NMS，才能够对 SNMP 代理进行操作，这种密码也分了级别，可以分别定义相应密码是否具有读权限或者是否同时具有读和写的权限。这样的密码被称为 community。版本 1 不仅提供密码认证的方式，除此之外，还可以利用 ACL 的方式来限制只有某些主机能够对其进行访问，也就是说即使主机提供了相应的密码，同样不能够对设备进行操作，还必须 IP 地址是被允许的。

版本 2c：版本 2c 的认证方式和版本 1 是一样的。

版本 3：版本 3 提供的认证方式是使用用户名和密码的方式，并且密码可以是 MD5 加密的。

如果要让 SNMP 代理和 SNMP 管理正常的协同工作，必须将双方的版本配置一致。需要注意的是，一台已配置 SNMP 的设备，可以同时允许多个 SNMP 管理对其进行操作，那么这时就可以在设备上分别为每个 SNMP 管理配置各自的 SNMP 版本。

MIB

设备的所有信息都包含在 MIB 中，接口信息在 MIB 中共分三类：

1：ifIndex(ifEntry 1) （接口索引）， 是接口在 MIB 中的区分唯一性的方法，这个值是大于 0 的。查看可使用命令：show snmp mib ifmib ifindex

2：ifAlias(ifXEntry 18) （接口别名）,是用户给接口定义的名称，方便识别，在接口下 description 定义。 snmp ifmib ifalias long 命令用来扩展最长可用字符为 256（默认 64）。

别名是在 show interfaces 时才看到。

3：ifName (ifXEntry 1)（接口名），即接口的原名，

所有信息在 MIB 中存储时，以词汇表的顺序编排，（即字典上 A-Z）

所有软硬件在 MIB 库里，都有与自己相关联的标识，而这些标识，并不是永久不变的，有的可能因为重启后，就改变了，也可以通过配置使其固定不变。方法可以使用命令

snmp mib persist ，如果要保存，使用 write mib-data 写入 NVRAM。

让接口在 MIB 里固定使用某个信息，全局命令 snmp mib persist circuit 来完成。

Event MIB（MIB 事件）

MIB 认为设备上的软硬件在多大的范围内变化可被认为是事件发生，这些采集方法可分为三种：

Absolute Sampling ，绝对采集也就是定义绝对变化多少之后 ，认为是事件

Delta Sampling 相对采集 定义从多少到多少这样一个变化范围，认为是事件

Changed Sampling

变化采集，只要变化便认为是事件，多用于硬件状态。

这些参数，对配置 RMON 时有用。

配置

注：没有一条特定的开启 SNMP 的命令，全部以 snmp-server 打头的命令便开启了 SNMP。

1.配置 SNMP v1 和 V2

（1）配置 ACL，通过此 ACL，可以限制相应的 IP 才能访问 SNMP 代理：

r1(config)#access-list 10 permit 10.1.1.2 定义源 IP 为 10.1.1.2 的才能访问

（2）配置密码（即 community）

鸿鹄培训+考试群：301283867

说明：这个密码必须定义，并且需要说明通过这个密码，能够执行什么样的操作，操作分为读和读写。并且后面可以跟上 ACL 作源 IP 限制。

r1(config)#snmp-server community cisco rw 10

注：源 IP 为 ACL 10 的主机通过密码 cisco 具有读写权限。

（3）配置 trap（也可用 informs 代理）

说明：设备在默认情况下，许多事件是不进行记录的，如果要记录，就需要打开相应的事件记录功能。

r1(config)#snmp-server enable traps bgp

注：配置为记录 BGP 的事件，如果 traps 后面不跟参数，就为开启所有事件记录功能。

（4）配置对 SNMP 管理的 trap（须已配置 trap，否则无效）

说明：当配置完 trap 之后，相应的事件并不会对 SNMP 管理进行反馈，还必须手工指明对哪些主机进行反馈什么样的事件。在这里，对 SNMP 管理反馈的事件必须全局已配，而且还可以在此配置对相应主机要求的密码，SNMP 版本等信息。

r1(config)#snmp-server host 10.1.1.100 version 2c cisco bgp

注：配置主机 10.1.1.100 使用密码 cisco，并且向主机发送 BGP 事件信息，而且 SNMP 版本为 2c。

某些 trap 不需要开，因为本身就是开启的，如果需要开启对接口状态的监控（即 up 和 down 状态的监控），需要使用如下命令：

r1(config)#snmp trap link-status （此命令因 IOS 不同而命令有所不同）

（5）配置关闭系统功能

说明：通过 NMS 默认情况下不能对设备进行重启操作，如需开启重启的权限，需要额外配置：

r1(config)#snmp-server system-shutdown

（6）配置接口与 MIB 绑定：

r1(config)#snmp-server ifindex persist

（7）配置 MIB 所有信息均绑定：

r1(config)#snmp mib persist

r1#write mib-data

说明：虽然没有一条特定的命令来开启 SNMP 服务，但有特定的命令可以关闭所有 SNMP 服务：

r1(config)#no snmp-server

2.查看信息：

（1）查看 SNMP 总信息：

r1#sh snmp

（2）查看与 SNMP 管理之间的会话

r1#sh snmp sessions

（3）查看 MIB 所有软硬件信息

r1#sh snmp mib：

（4）查看所有 MIB 中接口的信息（此信息对 RMON 有用）

r1#sh snmp mib ifmib ifindex

RMON（远程监视）

概述

功能有点像 SNMP，但可以用来增强 SNMP，对 SNMP MIB 中各种信息的变量进行轮询（查询），查询相关事件的变量分为上升门限和下降门限，上升门限即

超过一定的值，下降门限即降到一定的值。当 MIB 变量达到我们预期的效果后，就会发送 SNMP trap。

RMON 的组：

警报组

警报组(alarm)是 RMON 中的第 3 组，以指定的时间间隔监控一个特定的 MIB(Management

Information Base)对象，当这个 MIB 对象的值超过一个设定的上限值或低于一个设定的下限值

时，会触发一个警报。警报被当作事件来处理，处理事件的方式可以是记录日志或发送 SNMP Trap 的方式。

事件组

事件组(event)是 RMON 中的第 9 组，决定当由于警报而产生事件时，处理行为是产生一个日志. 记录表项或者一个 SNMP Trap。

取样时，对结果可以用两种方式来判断，分为 delta 和 absolute，关键字 delta 表示取样的值在 MIB 变量中两次取样间值的变化，而关键字 absolute 表示直接使扩 MIB 变量的值作为取样值也相当于平均值。

当配置 RMON 时，需要指定以下一些参数：

log ：(可选)输入这个关键值，则警报产生时，会将这个事件记录到日志中

trap：(可选)输入这个关键值，则警报产生时，会产生一个 SNMP Trap。

community：(可选)发送 Trap 时使用的认证名。

descrīption string：(可选)对这个事件的描述。

鸿鹄培训+考试群：301283867

sowner string：(可选)标志这个事件的拥有者。

需要说明的是，在配置 RMON 对目标进行监控时，输入目标的标识，这个目标的标识是在 SNMP MIB 中的标识符，也称为 OID，一般 SNMP MIB 里面的标识是我们看不懂的，如果需要理解其含义，可查询 CISCO OID 工具，地址为：

http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en

配置

1. 配置 SNMP（此步略过，详细见前面 SNMP 部分）

2. 配置 RMON

（1）例配置对接口 f0/0 的监控，

r1(config)#rmon alarm 1 ifEntry.1.1 2 absolute rising-threshold 2 falling-threshold 1 owner ccie

说明：配置接口 f0/0(OID: ifEntry.1.1 2,各目标的 OID 请自行查阅)，采样时间为 2 秒，当阀值上升到 2 或降到 1 的时候都产生警报，并注明了标识符为 ccie。

（2）并配置事件发生后通告的内容

r1(config)#rmon event 1 trap cisco description "intover" owner ccie

说明：community 为之前 SNMP 的配置，intover 表示消息，owner 可选标识。

3. 查看结果：

（1）查看产生过的警报

r1#show rmon alarms

（2）查看产生过的事件

r1#show rmon events

Embedded Event Manager (EEM)

之前的任何一种网络管理技术，如 SNMP，RMON，在检测到事件发生后，并不能解决问题，这些传统的网管技术只有监测功能，却没有解决故障的功能，因此，为了更有效的管理网络，能够在事件发生时，便采用有效的动作来杜绝网络问题，Cisco 推出更进一步的网管技术—Embedded Event Manager (EEM)。

EEM 在正常工作时，能够定期监视指定的事件，当被监测的事件发生后，EEM 可以产生指定的信息或指定的动作。EEM 如何检测指定的事件，需要指定相应的监测方法和监测标准，当事件发生后，需要产生的信息或执行的动作也需要定义，这一系列的事件和事件发生后需要执行的动作集合起来称为 EEM policy ；由于 EEM 工作复杂，所以 EEM 需要根据不同的分工定义不同的组件，EEM 共有如下几个组件：

EEM server

相当于 EEM 主程序。

Core Event Publishers (Event Detectors)

也就是 EEM 用于检测事件的组件，负责检测各种定义好的事件，事件的检测可以基于其它网管技术，Event Detectors 会在事件发生时向 Server 报告。

Event Subscribers (Policies)

当 Event Detector 检测到指定的事件发生后，Event Subscribers 便执行指定的动作，动作包括产生特定的消息，或执行特定的命令。

EEM 可以单独使用，也可以和其它网管技术配合使用,在配置 EEM 时，就是配置 EEM Policy，因为 Policy 就是事件和事件发生后需要执行的动作集合，配置 Policy，分两种方式：

Applet

Tool Command Language (Tcl)

其中 Applet 是使用 IOS 的 CLI 来配置的，操作相对简单，而 Tool Command Language (Tcl)是一种编程所使用的脚本工具，比较专业，需要使用外置的第三方 ASCII editor 才能编辑和配置，所以，在理论上，单纯只学习 Cisco 课程应该没有能力编写 ASCII editor 的，CCIE 考试中，目前推算几乎不太可能会考到，而 Applet 却直接就能在设备上进行配置。

由于 Cisco 的 IOS 版本众多，所以 EEM 的版本也是相当多，对于事件的检测和能够执行的动作，会因为 EEM 版本的不同而有所不同，基本上新版本会包含老版本的功能，对于各个 EEM 版本所支持的检测方法和执行的动作只作统一列举，而不一一列举，目前所有 EEM 的版本和对应的 IOS 版本信息如下：

EEM 1.0

支持的 IOS ： 12.0(26)S 、 12.3(4)T 以及后续版本。

EEM 2.0

支持的 IOS ： 12.2(25)S 以及后续版本。

EEM 2.1

支持的 IOS ： 12.3(14)T, 12.2(18)SXF5, 12.2(28)SB, 12.2(33)SRA 以及后续版本。

EEM 2.2

支持的 IOS ： 12.4(2)T, 12.2(31)SB3, 12.2(33)SRB 以及后续版本。

EEM 2.3

支持的 IOS ： Catalyst 6500 交换机上 12.2(33)SXH 以及后续版本。

EEM 2.4

支持的 IOS ： 12.4(20)T, 12.2(33)SXI, 12.2(33)SRE 以及后续版本。

EEM 3.0

支持的 IOS ： 12.4(22)T, 12.2(33)SRE 以及后续版本。

EEM 3.1

支持的 IOS ： 15.0(1)M 以及后续版本。

对于事件的检测，总体上支持如下一些方式，具体哪个版本支持哪个方式，请以实际 IOS 为准，不在该文档中详细说明，事件检测方式如下：

Application-Specific

CLI

Counter

Custom CLI

Enhanced Object Tracking

GOLD

Interface Counter

IPSLA

NF

None

OIR

Resource

RF

Routing

RPC

SNMP

SNMP Notification

SNMP Object

Syslog

System Manager

Timer

IOSWDSysMon (Cisco IOS watchdog)

WDSysMon (Cisco IOS Software Modularity watchdog)

当事件发生后，能够执行的动作如下：

Execute a CLI command

Generate a CNS event

Generate a prioritized syslog message

Generate an SNMP trap

Manually run an EEM policy

Publish an application-specific event

Read the state of a tracked object

Reload the Cisco IOS software

Request system information

Send a short e-mail

Set or modify a named counter

Set the state of a tracked object

Switch to a secondary RP

对于配置 Policy，只对 Applet 做出介绍，由于 Tool Command Language (Tcl)已经超过范围，不再讨论。

EEM Applet

在配置 Applet 时，共有 3 种配置状态：Event， Action， Set。

Event 用于定义事件标准，当指定的要求发生或阀值触发时，则表示该事件产生。

Action 当事件发生后执行的动作。

Set 是设置变量的，目前只支持_exit_。

在 EEM Applet 配置中，一个 Policy 只支持一个 event，也就是一个 Policy 只能检测一个事件，如果退出 Policy 配置时并没有 event，则会有警告，表示 Applet 没有注册成功；但如果没有 action，事件照样被检测，只是事件发生后不会执行任何动作，一个 Policy 中可以配置多个 action。

注：如果修改配置，在没有退出配置模式前，是不会生效的。

在配置 Applet 时，每一个 action 动作是有标签的，多个 action 将由标签的顺序来执行，标签可以是字母，也可以是数字，如果是数字，需要写成如 01.0,，02.0 等等，或 1.0，2.0。

配置 EEM

说明： 配置中共包含

1.配置 EEM 监测内存使用率

2.配置 EEM 监测 Enhanced Object Tracking 状态

3.配置 EEM 监测 CPU 利用率

4.配置 EEM 在事件触发时发送简短 E-mail

1.配置 EEM 监测内存使用率：

（1）查看当前内存情况：

Router#show processes memory

```
Processor Pool Total:   30623072 Used:   17889156 Free:   12733916
      I/O Pool Total:    6291456 Used:    4429312 Free:    1862144

 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0   35624856   13980528   18744396        608         78
*Init*
   0   0      12128     122652      12128          0          0
*Sched*
   0   0     248956     895688        564          1          0
*Dead*

Router#
```

说明：从结果中看出，内存总大小为 30623072，空闲大小为 12733916。

（2）查看当前路由协议状态：

```
Router#sh ip protocols

Router#
```

说明：路由器当前没有配置任何路由协议。

（3）配置 EEM 监测内存使用率：

```
Router(config)#event manager applet MEM
```

```
Router(config-applet)#event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1
get-type exact entry-op lt entry-val 30623072 poll-interval 90

Router(config-applet)#action 01.0 cli command "enable"

Router(config-applet)#action 02.0 cli command "conf t"

Router(config-applet)#action 03.0 cli command "router eigrp 100"

Router(config-applet)#exit
```

说明：EEM 当前监测内存的使用情况，如果空闲大小低于 30623072，则事件被触发，采集间隔为 90 秒一次，如果事件触发后，执行的第一个动作为在命令行下输入命令 enable，执行的第二个动作为在命令行下输入命令 conf t，执行的第三个动作为在命令行下输入命令 router eigrp 100，其实结果就是在事件发生后，自动启用一个 EIGRP 进程，AS 号为 100；结合之前可以得知，内存总大小为 30623072，所以内存空闲空间肯定会小于 30623072，那么该 EEM policy 配置后，事件肯定是被触发的。其中动作标签为 01.0 格式。

（4）查看 EEM Policy 注册情况：

```
Router#show event manager policy registered

No.  Class   Type    Event Type          Trap  Time
Registered          Name

1    applet  system  snmp                Off   Fri Mar 1 00:10:53
2002   MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1} get_type exact entry_op lt entry_val
{30623072} poll_interval 90.000

 action 01.0 cli command "enable"

 action 02.0 cli command "conf t"

 action 03.0 cli command "router eigrp 100"
```

```
Router#
```

说明：正常工作的 EEM Policy 名字为 MEM，并且其它详细信息也能看见。

（5）查看当前内存使用情况：

```
Router#show processes memory

Processor Pool Total:   30623072 Used:   18103504 Free:   12519568
      I/O Pool Total:    6291456 Used:    4429312 Free:    1862144

 PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
   0   0   35624856   13980528   18744396        608         78
*Init*
   0   0      12128     155784      12128          0          0
*Sched*

Router#
```

说明：当前内存空闲大小为 12519568，小于事件定义的 30623072，所以事件肯定已经触发。

（6）再次查看路由协议情况：

```
Router#sh ip protocols

Routing Protocol is "eigrp 100"

  Outgoing update filter list for all interfaces is not set

  Incoming update filter list for all interfaces is not set
```

```
  Default networks flagged in outgoing updates

  Default networks accepted from incoming updates

  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

  EIGRP maximum hopcount 100

  EIGRP maximum metric variance 1

  Redistributing: eigrp 100

  EIGRP NSF-aware route hold timer is 240s

  Automatic network summarization is in effect

  Maximum path: 4

  Routing for Networks:

  Routing Information Sources:

    Gateway         Distance      Last Update

  Distance: internal 90 external 170


Router#

Router#
```

说明：由于 EEM 事件被触发，所以自动启用一个 EIGRP 进程，AS 号为 100。

（7）查看 EEM 事件记录：

```
Router#show event manager history events detailed

No.   Time of Event             Event Type          Name

1     Fri Mar 1  00:14:53 2002  snmp                applet: MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1}
```

```
 val {12725680}

2    Fri Mar 1  00:16:23 2002  snmp                applet: MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1}

 val {12532448}

Router#
```

说明：结果显示了相应的 EEM 事件触发了两次。

2:配置 EEM 监测 Enhanced Object Tracking 状态

（1）开启接口 F0/0：

```
Router(config)#int f0/0

Router(config-if)#no shutdown

Router(config-if)#exi

Router(config)#exi


Router#sh interfaces f0/0

FastEthernet0/0 is up, line protocol is up

  Hardware is Gt96k FE, address is c000.03d0.0000 (bia c000.03d0.0000)
```

说明：接口 F0/0 处于双 up 状态。

（2）配置 Enhanced Object Tracking，跟踪接口 F0/0 的状态：

```
Router(config)#track 1 interface f0/0 line-protocol
Router(config-track)#exit
Router(config)#exit
```

```
Router#show track
Track 1
  Interface FastEthernet0/0 line-protocol
  Line protocol is Up
    1 change, last change 00:00:06
Router#
```

说明：当前 Enhanced Object Tracking 跟踪接口 F0/0 的 line-protocol 状态，由于接口 F0/0 目前为双 up 状态，所以 Enhanced Object Tracking 的状态也为 up。

（3）配置 EEM 监测 Enhanced Object Tracking 的状态：

```
Router(config)#event manager applet EOT
Router(config-applet)#event track 1 state down
Router(config-applet)#action a cli command "enable"
Router(config-applet)#action b cli command "conf t"
Router(config-applet)#action c cli command "router ospf 100"
```

```
Router(config-applet)#exit
```

说明：EEM 监测 track 1 的状态，如果为 down，则事件触发，并且在事件触发后，自动配置 ospf，进程为 100。其中动作标签为字母格式。

（4）查看 EEM Policy 注册情况：

```
Router#sh event manager policy registered

No.  Class   Type    Event Type          Trap  Time
Registered          Name

1    applet  system  snmp                Off   Fri Mar 1 00:13:23
2002   MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1} get_type exact entry_op lt entry_val
{30623072} poll_interval 90.000

 action 01.0 cli command "enable"

 action 02.0 cli command "conf t"

 action 03.0 cli command "router eigrp 100"



2    applet  system  track               Off   Fri Mar 1 00:22:49
2002   EOT

 track 1 state down

 action a cli command "enable"

 action b cli command "conf t"

 action c cli command "router ospf 100"
```

```
Router#
```

说明：显示了正常工作的 EEM Policy，包含之前的 MEM 和现在的 EOT。

（5）将 Enhanced Object Tracking 的状态变为 down：

```
Router(config)#int f0/0
Router(config-if)#shutdown
Router(config-if)#exit
```

```
Router#show track
Track 1
  Interface FastEthernet0/0 line-protocol
  Line protocol is Down (hw admin-down)
    2 changes, last change 00:00:08
  Tracked by:
    EEM applet EOT
Router#
```

说明：由于接口 F0/0 被关闭，所以 Enhanced Object Tracking 的状态变为 down，也显示了当前 Enhanced Object Tracking 的状态正被 EEM 所监测。

（6）再次查看路由器上的路由协议：

```
Router#sh ip protocols summary

Index Process Name

0     connected

1     static

2     eigrp 100

3     ospf 100

Router#
```

说明：由于 Enhanced Object Tracking 的状态变为 down，EEM 事件被触发，所以自动配置了 OSPF，进程号 100。

（7）查看 EEM 事件记录：

```
Router#show event manager history events

No.   Time of Event              Event Type          Name

1     Fri Mar 1  00:14:53 2002   snmp                applet: MEM

2     Fri Mar 1  00:16:23 2002   snmp                applet: MEM

3     Fri Mar 1  00:17:53 2002   snmp                applet: MEM

4     Fri Mar 1  00:19:23 2002   snmp                applet: MEM

5     Fri Mar 1  00:20:53 2002   snmp                applet: MEM

6     Fri Mar 1  00:22:23 2002   snmp                applet: MEM

7     Fri Mar 1  00:23:53 2002   snmp                applet: MEM

8     Fri Mar 1  00:24:28 2002   track               applet: EOT

Router#
```

说明：结果显示了 Enhanced Object Tracking 引起的 EEM 事件。

3.配置 EEM 监测 CPU 利用率

（1）查看当前 CPU 利用率：

```
Router#show processes cpu

CPU utilization for five seconds: 8%/0%; one minute: 4%; five minutes: 4%

 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process

   1           4        47         85  0.00%  0.00%  0.00%   0 Chunk
Manager

   2          12       312         38  0.08%  0.01%  0.00%   0 Load
Meter

   3       31580      4185       7545  8.35%  3.53%  3.35%   0
Exec


Router#
```

说明：当前 CPU 利用率大于 4%。

（2）配置 EEM 监测 CPU 使用率：

```
Router(config)#event manager applet CPU

Router(config-applet)#event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3.1
get-type exact entry-op ge entry-val 1 poll-interval 10
```

```
Router(config-applet)#action 1.0 syslog msg "CPU Over"

Router(config-applet)#exit
```

说明：配置 EEM 监测 CPU 的使用率，每 10 秒种采集一次，如果使用率超过 1%，则事件被触发，当事件触发后，自动产生 syslog 消息”CPU Over”，CPU 使用率肯定超过了 1%，所以事件已经触发。

（3）开启 syslog 功能：

```
Router(config)#logging on

Router(config)#logging buffered 10000 7
```

说明：开启 syslog 缓存信息功能。

（4）查看 EEM Policy 注册情况：

```
Router#show event manager policy registered

No.  Class   Type    Event Type          Trap  Time
Registered          Name

1    applet  system  snmp                Off   Fri Mar 1 00:13:23
2002   MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1} get_type exact entry_op lt entry_val
{30623072} poll_interval 90.000

 action 01.0 cli command "enable"

 action 02.0 cli command "conf t"

 action 03.0 cli command "router eigrp 100"
```

```
2    applet  system  track              Off   Fri Mar 1 00:22:49
2002   EOT

 track 1 state down

 action a cli command "enable"

 action b cli command "conf t"

 action c cli command "router ospf 100"



3    applet  system  snmp               Off   Fri Mar 1 00:40:12
2002   CPU

 oid {1.3.6.1.4.1.9.9.109.1.1.1.1.3.1} get_type exact entry_op ge
entry_val {1} poll_interval 10.000

 action 1.0 syslog msg "CPU Over"



Router#
```

说明：显示了正常工作的 EEM Policy，包含之前的 MEM，EOT 和现在的 CPU。

（5）查看 syslog 状态：

```
Router#sh logging

Syslog logging: enabled (11 messages dropped, 0 messages rate-limited,

                0 flushes, 0 overruns, xml disabled, filtering disabled)

    Console logging: level debugging, 78 messages logged, xml disabled,

                     filtering disabled

    Monitor logging: level debugging, 0 messages logged, xml disabled,
```

```
                    filtering disabled
    Buffer logging: level debugging, 2 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled

No active filter modules.

ESM: 0 messages dropped

    Trap logging: level informational, 83 message lines logged

Log Buffer (10000 bytes):

*Mar  1 00:40:32.155: %HA_EM-6-LOG: CPU: CPU Over
*Mar  1 00:40:39.395: %SYS-5-CONFIG_I: Configured from console by console
Router#
```

说明：因为 CPU 使用率超过 1%，所以自动产生了 syslog 信息” CPU Over”。

（6）查看 EEM 事件记录：

```
Router#
```

```
Router#show event manager history events detailed

No.   Time of Event              Event Type          Name

1     Fri Mar 1  00:34:23 2002   snmp                applet: MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1}

 val {12133372}

2     Fri Mar 1  00:35:53 2002   snmp                applet: MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1}

 val {12013372}

3     Fri Mar 1  00:37:23 2002   snmp                applet: MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1}

 val {12026628}

4     Fri Mar 1  00:38:53 2002   snmp                applet: MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1}

 val {12020216}

5     Fri Mar 1  00:40:22 2002   snmp                applet: CPU

 oid {1.3.6.1.4.1.9.9.109.1.1.1.1.3.1}

 val {11}

6     Fri Mar 1  00:40:23 2002   snmp                applet: MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1}

 val {12014284}

7     Fri Mar 1  00:40:32 2002   snmp                applet: CPU

 oid {1.3.6.1.4.1.9.9.109.1.1.1.1.3.1}
```


鸿鹄培训+考试群：301283867

```
 val {1}

8    Fri Mar 1  00:40:52 2002  snmp                applet: CPU

 oid {1.3.6.1.4.1.9.9.109.1.1.1.1.3.1}

 val {1}

Router#

Router#
```

说明：结果显示了 CPU 利用率触发的事件。

4.配置 EEM 在事件触发时发送简短 E-mail

（1）配置 EEM 在 CPU 利用率超过阀值时发送简短 E-mail:

```
Router(config)#event manager applet EMAIL

Router(config-applet)#event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3.1
get-type exact entry-op ge entry-val 75 poll-interval 10

Router(config-applet)#action 1.0 cli command "enable"

Router(config-applet)#action 2.0 cli command "show process cpu"

Router(config-applet)#action 3.0 mail server "192.168.1.146" to
"engineer@cisco.com" from "eem@cisco.com" subject "CPU Alert" body "CPU
Alert 75"
```

说明：EEM Policy 定义了 CPU 利用率在超过 75%时，将命令 show process cpu 的结果发送到邮箱地址"engineer@cisco.com" 邮箱源地址为"eem@cisco.com" ，邮件主题为 "CPU Alert"，正文同时添加"CPU Alert 75"。

（2）查看 EEM Policy 注册情况：

```
Router#show event manager policy registered

No.  Class   Type    Event Type          Trap  Time
Registered           Name

1    applet  system  snmp                Off   Fri Mar 1 00:13:23
2002   MEM

 oid {1.3.6.1.4.1.9.9.48.1.1.1.6.1} get_type exact entry_op lt entry_val
{30623072} poll_interval 90.000

 action 01.0 cli command "enable"

 action 02.0 cli command "conf t"

 action 03.0 cli command "router eigrp 100"



2    applet  system  track             Off   Fri Mar 1 00:22:49
2002   EOT

 track 1 state down

 action a cli command "enable"

 action b cli command "conf t"

 action c cli command "router ospf 100"



3    applet  system  snmp                Off   Fri Mar 1 00:40:12
2002   CPU

 oid {1.3.6.1.4.1.9.9.109.1.1.1.1.3.1} get_type exact entry_op ge
entry_val {1} poll_interval 10.000

 action 1.0 syslog msg "CPU Over"
```

```
4    applet  system  snmp                Off   Fri Mar 1 00:53:37
2002   EMAIL

 oid {1.3.6.1.4.1.9.9.109.1.1.1.1.3.1} get_type exact entry_op ge
entry_val {75} poll_interval 10.000

 action 1.0 cli command "enable"

 action 2.0 cli command "show process cpu"

 action 3.0 mail server "192.168.1.146" to "engineer@cisco.com" from
"eem@cisco.com" subject "CPU Alert" body "CPU Alert 75"


Router#
```

说明：结果显示了正常工作中的 EEM policy。

SCP（安全复制协议）

概述

SCP 通过认证的方式为 Cisco 设备的配置和映像复制提供安全，这种安全是基于 SSH 的。

在配置 SCP 之前，必须配置 SSH，认证和授权，所以必须有 RSA 密码。正因为有了 SSH 和 AAA 这样的机制所以能够确定用户是否是合法的。

配置

1. 配置 AAA：

注：AAA 是必配，详细 AAA 介绍和配置，请参见本站的安全部分。

（1）开启 AAA

Router (config)# aaa new-model

（2）开启认证并指明认证方式

Router (config)# aaa authentication login default local 认证方式为本地用户数据库

（3）开启授权指明授权方式

Router (config)# aaa authorization exec default local 授权方式为本地用户数据库

（4）创建本地用户数据库

Router (config)# username ccie privilege 15 password cisco 必须为 15 级

2. 配置 SSH：

注：SSH 是必配，详细 SSH 介绍和配置，请参见本站的安全部分。

（1）配置域名

r1(config)#ip domain-name cisco.com 配置域名为 cisco.com

(2) 配置 RSA 密码

r1(config)#crypto key generate rsa

3. 配置 SCP

（1）开启 SCP 服务

Router(config)# ip scp server enable

4.检查配置

（1）检查配置

Router# show running-config

5.SCP 测试：

(1)测试从远程路由器向这台开了 SCP 的路由器传送文件

说明：在远程路由器将 flash:下的文件 r1-confg 传送到开了 SCP 服务的路由器，目录是在 flash 下的。注：前面需要跟用户名，但密码只能在提示的时候才输入。

r1#copy flash:r1-confg scp://ccie@10.1.1.2/

Address or name of remote host [10.1.1.2]?

Destination username [ccie]?

Destination filename [r1-confg]?

Writing r1-confg

Password:

!

1054 bytes copied in 6.428 secs (164 bytes/sec)

r1#

IGP

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

鸿鹄培训+考试群：301283867

静态路由

ICMP 重定向（ICMP Redirect）

网络中的路由器通过相互之间的共同努力，将用户的数据包转发到目的地。通常情况下，主机都会将去往远程网络的数据包发送到路由器，路由器再尽最大努力转发数据。但是在某些情况下，收到数据包的路由器可能并不是去往目的地的最优选择，也就是说该路由器并不在源与目标的路径当中，或者说数据源应该将数据交给其它路由器来转发。如果某台路由器真的发现自己不应该为用户转发数据，而希望让用户选择其它路由器来转发数据，那么它就会通过向数据源发送 ICMP 重定向（ICMP Redirect）来告诉对方，让对方不要再将数据包发向自己，而应该发到其它路由器。

需要路由器向源发送 ICMP 重定向的情况有两种：

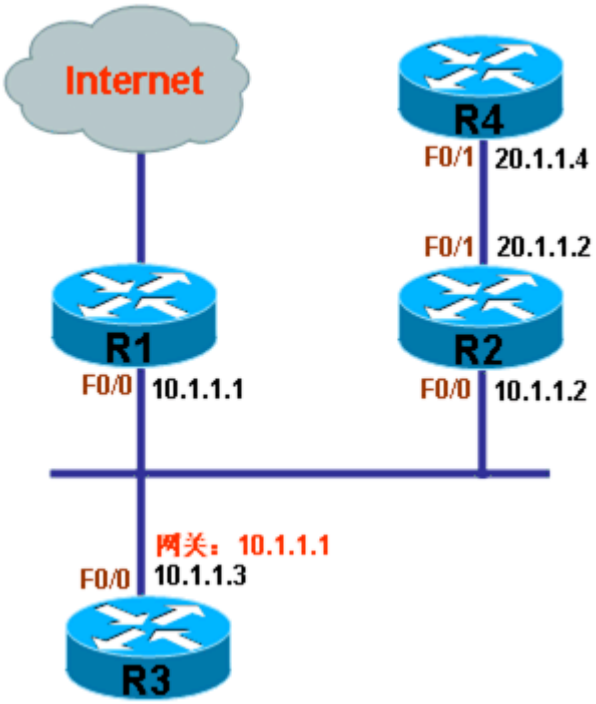
★ **1.** 当路由器从某个接口收到数据包后，还要将数据包从同一个接口发往目的地，就是路由器收到数据包的接口正是去往目的地的出口时，则会向源发送 ICMP 重定向，通告对方直接将数据包发向自己的下一跳即可，不要再发给自己。

★ **2.** 数据包的源 IP 和自己转发时的下一跳 IP 地址是同网段时，则会向源发送 ICMP 重定向，通告对方直接将数据包发向自己的下一跳。

注：路由器在向数据源发送 ICMP 重定向的同时，也会正常转发收到的数据包，并不会中断网络。

配置 ICMP 重定向

说明:ICMP 重定向是基于接口配置的，默认为开启状态。



说明：以上图为例，测试 ICMP 重定向，其中，R1，R2，R3 的接口 F0/0 在 10.1.1.0/24 网段，R2 和 R4 的接口 F0/1 在 20.1.1.0/24 网段，而 R3 将去往任何目的的数据全部交给 R1。

1.配置基础网络环境

（1）配置 R1:

r1(config)#int f0/0

r1(config-if)#ip address 10.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config-if)#exit

r1(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.2

说明：R1 将去往任何目的地的数据包全部交给 10.1.1.2，即交给 R2。

（2）配置 R2:

r2(config)#int f0/0

r2(config-if)#ip add 10.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config-if)#exit

r2(config)#int f0/1

r2(config-if)#ip add 20.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config-if)#exit

说明：R2 同时连接 10.1.1.0/24 和 20.1.1.0/24。

（3）配置 R3:

鸿鹄培训+考试群：301283867

r3(config)#int f0/0

r3(config-if)#ip add 10.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config-if)#exit

r3(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1

说明：R3 将去往任何目的地的数据包全部交给 10.1.1.1，即选择 R1 作为网关。

（4）配置 R4:

r4(config)#int f0/1

r4(config-if)#ip add 20.1.1.4 255.255.255.0

r4(config-if)#no sh

r4(config-if)#exit

r4(config)#ip route 0.0.0.0 0.0.0.0 20.1.1.2

说明：R4 在 20.1.1.0/24。

2.测试 ICMP 重定向

（1）在 R3 上向目标网络 20.1.1.0 发送数据包来测试 ICMP 重定向，并且打开 debug 观察数据包：

r3#debug ip icmp

ICMP packet debugging is on

r3#

r3#ping 20.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/84/200 ms

r3#

*Mar 1 00:15:39.075: ICMP: redirect rcvd from 10.1.1.1- for 20.1.1.4 use gw 10.1.1.2

*Mar 1 00:15:39.175: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:15:39.291: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:15:39.323: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:15:39.383: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:15:39.403: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

r3#

说明：从上面信息可以看出，由于 R3 的网关是 10.1.1.1，所以会将去往 20.1.1.0/24 的数据包发给网关 R1，但是 R1 从接口 F0/0 收到数据包后，检查路由表得知需要再将数据包从相同接口 F0/0 发给 10.1.1.2，不仅满足发送 ICMP 重定向情况的第一条同接口进出，也满足第二条源和下一跳同网段，所以 R1 向源发送了 ICMP 重定向，数据包中明确告诉 R3 将去往 20.1.1.4 的数据包直接交给 10.1.1.2，即交给 R2。从上也可以看出，让 R1 来转发数据包确实是无谓举动。

（2）更改 R3 的路由方式:

r3(config)#ip route 0.0.0.0 0.0.0.0 f0/0

说明：如果将 R3 的路由改为直接指定出接口，而不使用下一跳 IP 地址，则不会造成 R1 发送 ICMP 重定向，因为 R3 在此类路由方式下，并不会将数据包发向

R1。

（3）测试 R3 更改路由方式后的情况:

r3#debug ip icmp

ICMP packet debugging is on

r3#

r3#ping 20.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 52/86/176 ms

r3#

*Mar 1 00:33:41.511: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:33:41.607: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:33:41.663: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:33:41.719: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

*Mar 1 00:33:41.779: ICMP: echo reply rcvd, src 20.1.1.4, dst 10.1.1.3

r3#

说明：可以看出，R1 并没有再发送 ICMP 重定向，因为 R3 并没有将去往 20.1.1.0/24 的数据包发向 R1，具体原因，由普通 ARP 的原理可以得知。

3.关闭 ICMP 重定向

（1）在 R1 接口上关闭 ICMP 重定向:

r1(config)#int f0/0

r1(config-if)#no ip redirects

说明：需要开启 ICMP 重定向，输入命令 ip redirects；ICMP 重定向功能不建议关闭。

注：在接口上开启 HSRP 后，默认会关闭 ICMP 重定向的功能，在 IOS 12.1(3)T 和以后的版本可以手工开启 ICMP 重定向功能。

代理 ARP（Proxy-ARP）

数据包在网络上寻址时，需要靠 OSI 七层模型中的第二层数据链路层地址和第三层网络层地址来完成，只要完成这两个地址的封装，数据包便能够发往目的地，也必须完成这两个地址的封装，数据包才能发往目的地。

在当前的网络中，第三层网络层地址就是 IP 地址，主机之间要通信，必须封装好双方的 IP 地址，这是无可争议的；IP 地址在主机通信的过程中，不会因为网络设备或物理介质的改变而改变。而第二层数据链路层地址则会因为链路介质的不同而发生变化，并且在主机通信中，链路层地址会不断发生变化，因为链路层地址只在同一段介质中才有效，如果介质是多路访问类型，如以太网，则只在同一 IP 网段有效。 在这里需要讨论的是代理 ARP，故介质默认为以太网。

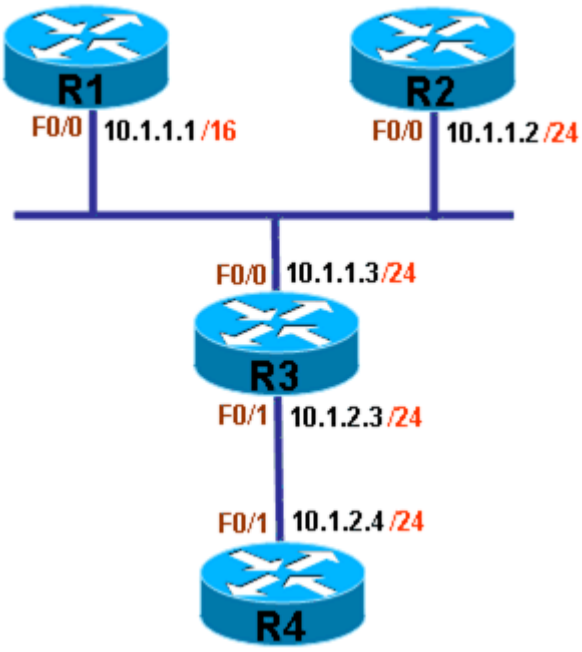
如果通信的主机在同一网段，源主机则直接请求目标主机的二层链路地址（以太网为 MAC 地址），其它介质的网络同样也是直接请求目标主机的二层链路地址，只不过不叫 MAC 地址；如果得不到二层链路地址的回复，数据包就不能完成封装，就不能发送。如果通信的主机不在同一网段，必须经过路由后才能到达目标的话，那么源主机就无法直接请求目标主机的二层链路地址，因为请求是用广播发送的，所以，此时源主机就直接请求网关的二层链路地址，将数据包的目标地址封装为网关的二层链路地址，从而将数据包交给网关处理，如果是路由器需要发送数据包到远程网络，同样的道理，路由器将数据包的二层目标地址封装为下一跳的二层链路地址，从而交给下一跳路由器来处理。

当向网络中发送广播请求目标的二层链路地址时，如果收到的路由器有去往目标网络的路由，那么路由器将会使用自己接口的二层链路地址来回复数据源，声称自己的二层链路地址就是目标的二层链路地址，这就是路由器的代理 ARP 功能。主机收到路由器的回复后，便将数据包的目标二层链路地址封装为路由器的二层链路地址，从而将数据包发到路由器，因为路由器是有到目标网络的路由的，所以通信不会有问题。

不难发现，当主机需要将数据包发到远程网络时，因为开启了代理 ARP 的路由器会以自己的二层链路地址充当目标的二层链路地址，所以主机不需要配置网关，就能够将去往远程网络的数据包发到路由器，最终完成通信。

配置代理 ARP

说明：代理 ARP 在路由器接口上默认是开启的，并且是基于接口打开或关闭的。



说明：以上图为例，测试代理 ARP，其中 R3 和 R4 的接口 F0/1 在 10.1.2.0/24

网段，而 R1，R2 和 R3 的接口 F0/0 在 10.1.1.0/24 网段，但由于 R1 的接口 F0/0 的掩码为 16 位，所以 R1 会认为整个 10.1.0.0/16 都是接口 F0/0 的直连网段，其中包含 10.1.2.0/24;但是 R2 的接口 F0/0 的掩码为 24 位，所以 R2 会认为 10.1.2.0/24 是在远程网络。

1.配置基础网络环境

（1）配置 R1:

```
r1(config)#int f0/0
r1(config-if)#ip add 10.1.1.1 255.255.0.0
r1(config-if)#no sh
```

说明：R1 配置网段 10.1.0.0/16，其中包含网段 10.1.1.0/24 和 10.1.2.0/24。

（2）配置 R2:

```
r2(config)#int f0/0
r2(config-if)#ip add 10.1.1.2 255.255.255.0
r2(config-if)#no sh
r2(config-if)#exit
r2(config)#ip route 10.1.2.0 255.255.255.0 f0/0
```

说明：R2 直连网段 10.1.1.0/24，并且通过配置静态路由将去往远程网段 10.1.2.0/24 定义为直连网段。

（3）配置 R3:

```
r3(config)#int f0/0
```

r3(config-if)#ip add 10.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config-if)#exit

r3(config)#int f0/1

r3(config-if)#ip add 10.1.2.3 255.255.255.0

r3(config-if)#no sh

r3(config-if)#exit

说明：R3 同时与 10.1.1.0/24 和 10.1.2.0/24 直连。

（4）配置 R4:

r4(config)#int f0/1

r4(config-if)#ip add 10.1.2.4 255.255.255.0

r4(config-if)#no sh

r4(config-if)#exit

r4(config)#ip route 0.0.0.0 0.0.0.0 10.1.2.3

说明：R4 连 10.1.2.0/24。

2.测试开启代理 ARP 的情况

说明：路由器接口默认已经开启代理 ARP，无需再开。

（1）查看 R1 的路由情况:

r1#sh ip route

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

鸿鹄培训+考试群：301283867

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

C 10.1.0.0 is directly connected, FastEthernet0/0

r1#

说明：可以看见，R1 与 10.1.0.0/16 直连，R1 会认为 10.1.2.0/24 也是自己的直连网段，所以我们不用写到 10.1.2.0/24 的静态路由。

（2）测试 R1 到 10.1.2.0/24 的连通性：

r1#ping 10.1.2.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.2.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 48/79/180 ms

r1#

说明：因为 R1 认为自己直连 10.1.0.0/16 网段，所以在向 10.1.2.0/24 发送数据包时，直接在本网段广播请求目标二层链路地址，这个广播被 R3 收到，又因为 R3 能够到达 10.1.2.0/24，并且开启代理 ARP 功能，所以 R3 将自己接口的二层链路地址回复给 R1，最终 R1 将去往 10.1.2.0/24 的数据包封装为 R3 的二层链路地址，从而将数据包交给 R3 处理，最后网络通信成功。

（3）查看 R4（10.1.2.4）的 F0/1 的 MAC 地址与 R3（10.1.1.3）的 F0/0 的 MAC 地址，并且查看 R1 去往 10.1.2.4 的 ARP 表：

R4：

r4#sh interfaces f0/1

FastEthernet0/1 is up, line protocol is up

Hardware is Gt96k FE, address is c000.1370.0001 (bia c000.1370.0001)

Internet address is 10.1.2.4/24

R3：

r3#sh int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is c000.1124.0000 (bia c000.1124.0000)

Internet address is 10.1.1.3/24

R1：

r1#sh arp

Protocol Address Age (min) Hardware Addr Type Interface

```
Internet  10.1.1.3          8   c000.1124.0000  ARPA   FastEthernet0/0
Internet  10.1.1.1          -   c000.141c.0000  ARPA   FastEthernet0/0
Internet  10.1.2.4          4   c000.1124.0000  ARPA   FastEthernet0/0
r1#
```

说明：从结果中可以看出，R1 获得的 10.1.2.4 的 MAC 地址并非目标 R4 的 MAC 地址，而是 R3 的接口 F0/0 的 MAC 地址，这就是由于 R3 的代理 ARP 功能，使得 R3 会代替目标 R4 回复源主机的二层链路地址请求。

（4）查看 R2 的路由表并测试到 10.1.2.0/24 的连通性：

```
r2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 2 subnets
```

S 10.1.2.0 is directly connected, FastEthernet0/0

C 10.1.1.0 is directly connected, FastEthernet0/0

r2#r2#ping 10.1.2.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.2.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/80/156 ms

r2#

说明：因为 R2 路由表中的静态路由指示去往 10.1.2.0/24 为直连网段，所以同上原因，因为 R3 代理 ARP 的功能，最后与 10.1.2.0/24 的网络通信正常。

3.测试关闭代理 ARP 的情况

（1）关闭 R3 接口 F0/0 的代理 ARP 功能:

r3(config)#int f0/0

r3(config-if)#no ip proxy-arp

说明：关闭了 R3 接口 F0/0 的代理 ARP 功能，要开启，输入命令 ip proxy-arp。

（2）查看 R1 与 10.1.2.0/24 的连通性和 ARP 情况：

r1#ping 10.1.2.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.2.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r1#

r1#sh arp

Protocol Address Age (min) Hardware Addr Type Interface

Internet 10.1.1.3 0 c000.1124.0000 ARPA FastEthernet0/0

Internet 10.1.1.1 - c000.141c.0000 ARPA FastEthernet0/0

Internet 10.1.2.4 0 Incomplete ARPA

r1#

说明：可以看见，当 R3 关闭了代理 ARP 功能后，R1 不能与 10.1.2.0/24 通信，因为 R1 认为目标与自己直连，所以会在直连网段直接请求目标的二层链路地址，但 R3 关闭了代理 ARP 功能，即使自己与目标可达，但也不会使用自己的 MAC 地址去回复 R1，最后 R1 也无法获得任何目标的 MAC 地址，ARP 表中显示 10.1.2.4 的记录为 Incomplete，最终与 10.1.2.0/24 的通信以失败告结。

（3）查看 R2 与 10.1.2.0/24 的连通性：

r2#ping 10.1.2.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.2.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

```
r2#

r2#sh arp

Protocol  Address          Age (min)  Hardware Addr   Type   Interface

Internet  10.1.1.2              -   c000.13e8.0000  ARPA   FastEthernet0/0

Internet  10.1.1.3              4   c000.1124.0000  ARPA   FastEthernet0/0

Internet  10.1.2.4              0   Incomplete      ARPA

r2#
```

说明：虽然 R2 拥有到 10.1.2.0/24 的静态路由，但因为静态路由指定去往目标为直连接口，所以 R2 会认为 10.1.2.0/24 与接口 F0/0 直连，由于 R3 关闭了代理 ARP，R2 与 R1 一样，不能与 10.1.2.0/24 通信。

（4）更改 R2 的静态路由方式，并查看路由表：

```
r2(config)#no ip route 10.1.2.0 255.255.255.0 f0/0

r2(config)#ip route 10.1.2.0 255.255.255.0 10.1.1.3

r2(config)#exi

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 2 subnets

S 10.1.2.0 [1/0] via 10.1.1.3

C 10.1.1.0 is directly connected, FastEthernet0/0

r2#

说明：R2 将去往目标 10.1.2.0/24 的静态路由改为下一跳指向 10.1.1.3（R3），所以 R2 并不会再认为 10.1.2.0/24 是自己的直连网段，因此在需要与 10.1.2.0/24 通信时，会请求下一跳地址 10.1.1.3 的二层链路地址，最终将数据包交给 10.1.1.3（R3）处理。

（5）查看更改静态路由后 R2 与 10.1.2.0/24 的通信情况：

r2#ping 10.1.2.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.2.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/68/192 ms

r2#

说明：更改静态路由到达目标网络为下一跳地址后，R2 不再直接请求目标 10.1.2.4 的二层链路地址，而是改为请求下一跳地址的二层链路地址，因为请求的地址 10.1.1.3，所以 R3 作出了回应，最后 R2 与目标通信成功。

说明：由以上情况可以看出，当配置静态路由时，如果指定远程目标为直连，则可能因为下一跳路由器关闭了代理 ARP 而造成通信失败，但静态路由指定为下一跳地址时，通信不会受到任何影响。

代理 ARP 在没有配置默认网关或不使用路由的网络中，比较有优势。

默认路由

当路由器收到目标地址不在路由表中的数据包时，全部发送到默认路由所定义的地方，作为未知地址数据包的一种最后求助，这就是默认路由的功能。

默认路由的使用，可以大大节省系统资源，缩减路由表的大小，而要使路由器生产默认路由，产生对未知地址数据包的最后求助，可以有三种方法：

IP Default-Gateway

IP Default-Network

IP route 0.0.0.0 0.0.0.0

下面来一一详细解释:

IP Default-Gateway

IP route 0.0.0.0 0.0.0.0

IP Default-Network

Classless 与 Classful

IP Default-Gateway

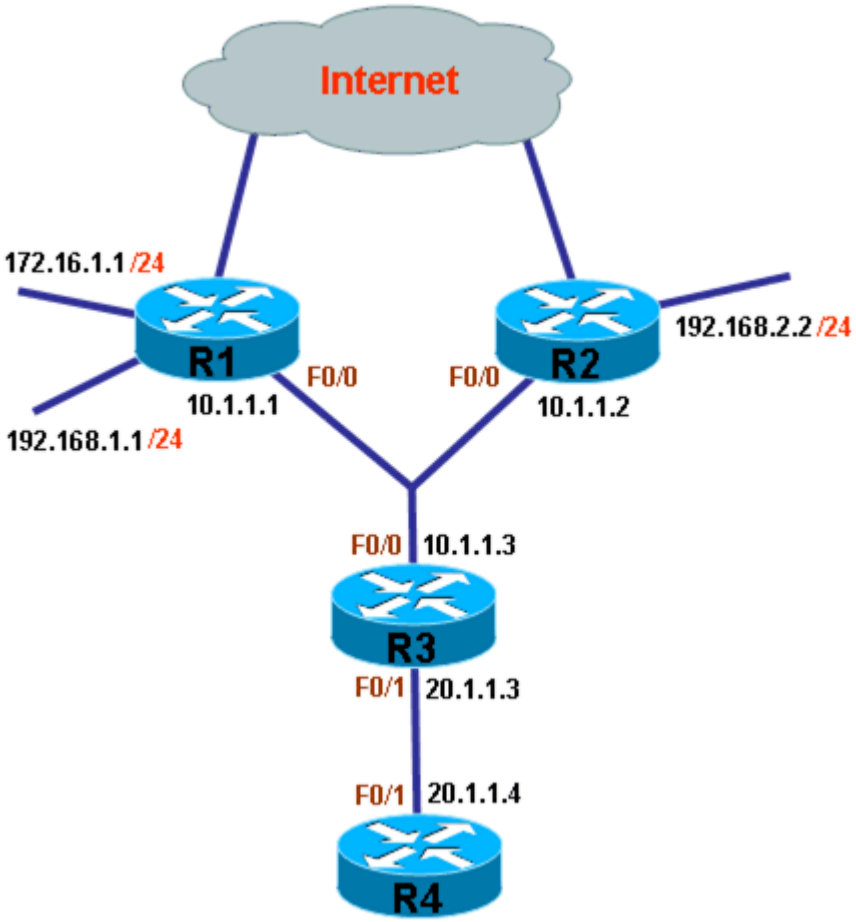
大家都非常清楚，我们最熟悉的 PC，在配置 IP 地址后，通常需要配置默认网关，除了目标地址在本网段的数据包直接发到目的地之外，其它所有数据包都发给网关处理。

而有时由于种种原因，路由器不可能获知网络中每一个网段，所以这时，路由器也需要像 PC 一样配置网关，将所有未知目标地址的数据包全部交给网关。

通过在路由器上配置命令 IP Default-Gateway 加上 IP 地址，可以手工为路由器指定一个默认网关，该默认网关的作用与 PC 完全相同。而命令 IP Default-Gateway 只有在路由器关闭路由功能后（命令 no ip routing），才能使用，如果路由器处于 boot 模式时，同样也可以通过该命令配置默认网关，这样便可以帮助像 TFTP 这样的传输。

配置 IP Default-Gateway

说明：必须先关闭路由功能（命令 no ip routing）



说明：以上图为例，测试 IP Default-Gateway。

1.在 R3 上配置 IP Default-Gateway

（1）在 R3 上配置 IP Default-Gateway:

r3(config)#no ip routing

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

r3(config)#ip default-gateway 10.1.1.1

说明：在 R3 上关闭路由功能，并指定默认网关为 10.1.1.1（R1）。

（2）在 R3 上查看默认网关:

r3#sh ip route

Default gateway is 10.1.1.1

Host Gateway Last Use Total Uses Interface

ICMP redirect cache is empty

r3#

说明：R3 上路由功能已关闭，并且所有数据包全部交给网关 10.1.1.1。

（3）测试网络连通性:

r3#ping 172.16.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/47/168 ms

r3#

r3#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/61/192 ms

r3#

说明：因为 R3 的网关为 10.1.1.1，而 172.16.1.1 与 192.168.1.1 也在 R1 上，所以 R3 与 172.16.1.1 和 192.168.1.1 通信正常。

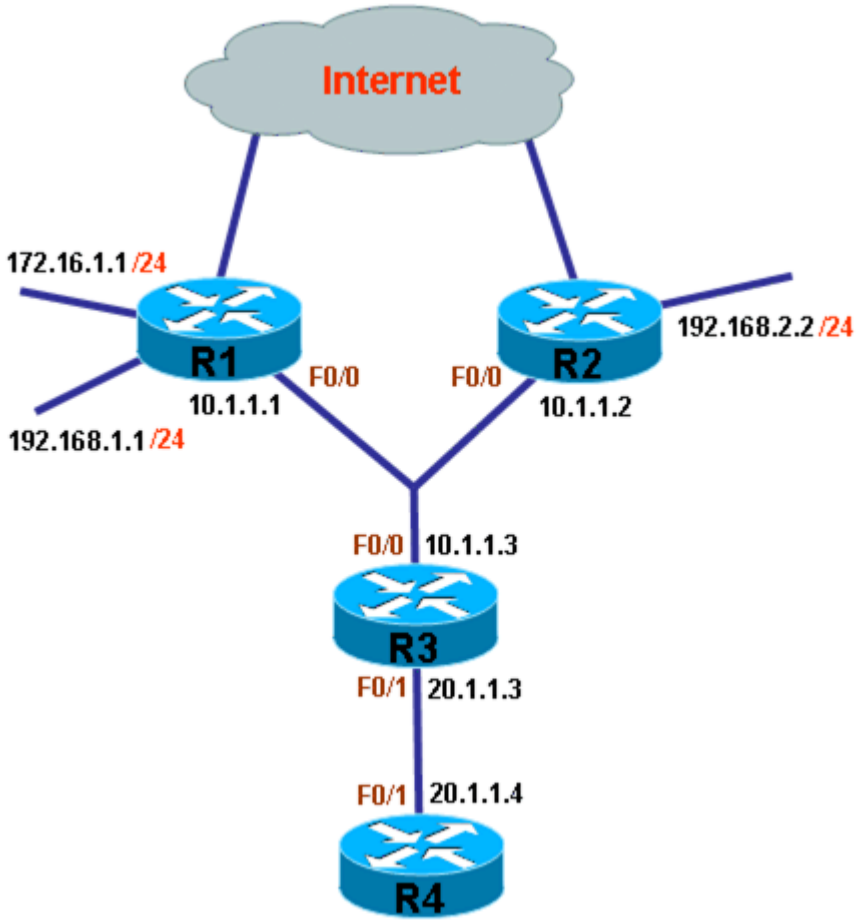
IP route 0.0.0.0 0.0.0.0

为路由器配置默认网关时，IP Default-Gateway 只能在关闭路由功能后起作用，在路由功能开启的情况下，通过命令 IP route 0.0.0.0 0.0.0.0 同样可以为路由器配置默认网关。

两者的区别在于，IP Default-Gateway 只能在路由功能关闭时工作，并且一台路由器只能配置一条，而 IP route 0.0.0.0 0.0.0.0 可以在路由功能开启时工作，一条路由器可以配置多条 IP route 0.0.0.0 0.0.0.0。

配置 IP route 0.0.0.0 0.0.0.0

说明：必须开启路由功能。



说明：以上图为例，配置 ip route 0.0.0.0 0.0.0.0

1.在 R3 上配置 ip route 0.0.0.0 0.0.0.0

（1）在 R3 上配置 ip route 0.0.0.0 0.0.0.0:

r3(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1

说明：R3 的网关指向 10.1.1.1（R1）。

（2）查看 R3 的路由情况:

```
r3#sh ip route

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

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

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

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

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

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


route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is 10.1.1.1 to network 0.0.0.0


     20.0.0.0/24 is subnetted, 1 subnets

C       20.1.1.0 is directly connected, FastEthernet0/1

     10.0.0.0/24 is subnetted, 1 subnets

C       10.1.1.0 is directly connected, FastEthernet0/0

S*   0.0.0.0/0 [1/0] via 10.1.1.1

r3#
```

说明：R3 路由表中有一条指向 10.1.1.1 的默认网关。

（3）测试网络连通性:

r3#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/80/196 ms

r3#

r3#ping 172.16.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 24/77/200 ms

r3#

说明：因为 R3 的网关为 10.1.1.1，而 172.16.1.1 与 192.168.1.1 也在 R1 上，所以 R3 与 172.16.1.1 和 192.168.1.1 通信正常。

2.配置多条 ip route 0.0.0.0 0.0.0.0

（1）在 R3 上配置多条 ip route 0.0.0.0 0.0.0.0:

r3(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1

r3(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.2

说明：命令 ip route 0.0.0.0 0.0.0.0 可以在单台路由器上重复配置。

（2）查看 R3 路由表情况:

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.2 to network 0.0.0.0

     20.0.0.0/24 is subnetted, 1 subnets
C       20.1.1.0 is directly connected, FastEthernet0/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 10.1.1.2
                  [1/0] via 10.1.1.1
r3#
```

说明：当配置多条 ip route 0.0.0.0 0.0.0.0 后，路由器将同时在其间执行负载均衡，需要注意的是，负载均衡会受到 CEF 的影响，所以默认情况，可能不是你想要的结果。

IP Default-Network

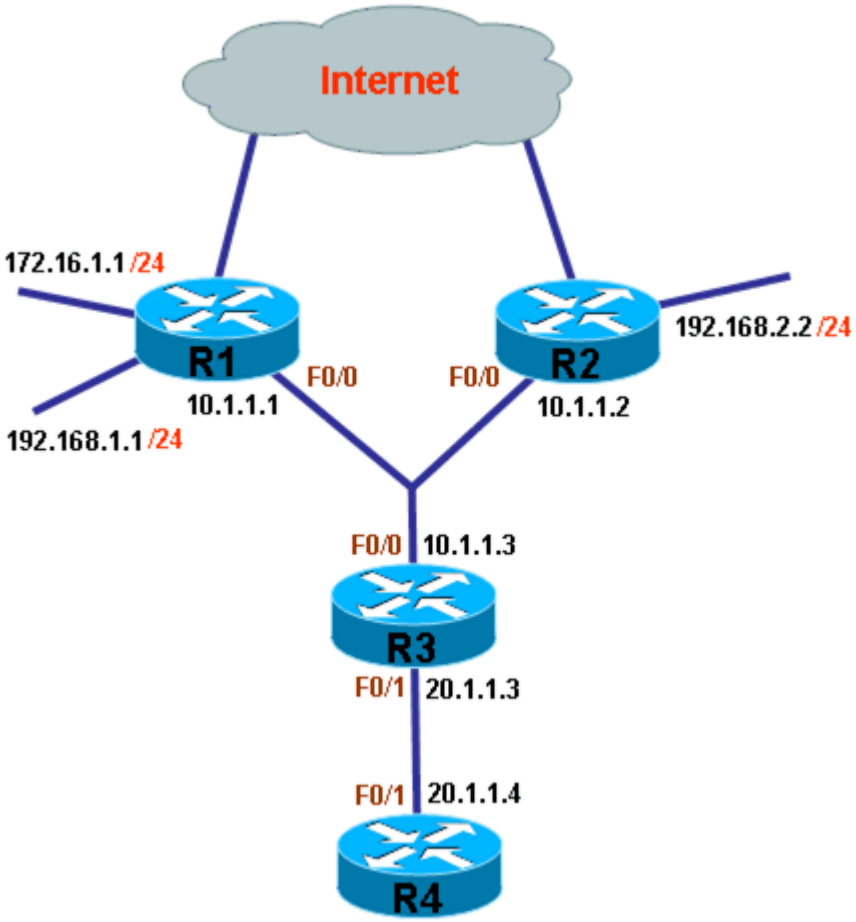
为路由器配置默认网关的方法除了 IP Default-Gateway 与 IP route 0.0.0.0 0.0.0.0 之外，还有 IP Default-Network，不同之处在于，IP Default-Gateway 只能工作在非路由模式下，而 IP route 0.0.0.0 0.0.0.0 可以工作在路由模式下，但不会自动被动态路由协议传递给邻居。如果使用 IP Default-Network，则被 IP Default-Network 所定义的网络将成为路由器的默认网关，所有未知目标的数据包全部发往该网络，IP Default-Network 的不同之处是它所定义的默认网关，会自动被动态路由协议传递，能够自动传递 IP Default-Network 默认网关的路由协议有 RIP，IGRP，EIGRP，而 OSPF 和 IS-IS 不会传递。

IP Default-Network 是 Classful 的，所指定的网段必须是没有划过子网的主类网络，否则不会产生默认网关。如果需要 IGRP 和 EIGRP 自动传递 IP Default-Network 的默认网关，那么

IP Default-Network 所指定的网络必须在 EIGRP 进程里通告，或者将该网络重分布进 EIGRP；对于 RIP，不需要在进程下通告便会自动传递，但由于 IOS 的不同，RIP 的操作可能存在着不同，某些 IOS 只能在 IP Default-Network 所指定的网络为直连网络时，才会被 RIP 传递，否则无效，所以请以自身 IOS 为准，因为思科并没有文档指出 IOS 版本号。

注：RIP version 1 与 version 2 都支持对 IP Default-Network 默认网关的自动传递。

配置 IP Default-Network



说明：以上图为例，测试 IP Default-Network。

1.在 R3 上配置 IP Default-Network

（1）在 R3 上手工配置到 192.168.1.0/24 的静态路由:

r3(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1

说明：配置该静态路由，目的在于让 192.168.1.0/24 事先存在于路由表中。

（2）查看 R3 的路由表，并测试到远程网络 192.168.1.0/24 与 172.16.1.0/24 的连通性：

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

S 192.168.1.0/24 [1/0] via 10.1.1.1

r3#

r3#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/56/172 ms

r3#

r3#ping 172.16.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r3#

说明：因为路由表中只有去往 192.168.1.0/24 的静态路由，所以 R3 与 192.168.1.0/24 的通信正常，而与 172.16.1.0/24 不能通信。

（3）在 R3 上配置 ip default-network:

r3(config)#ip default-network 192.168.1.0

说明：配置默认网关的网段为 192.168.1.0

（4）查看 R3 的路由表情况并再次测试连通性：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.1.1.1 to network 192.168.1.0
     20.0.0.0/24 is subnetted, 1 subnets
C       20.1.1.0 is directly connected, FastEthernet0/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
S*   192.168.1.0/24 [1/0] via 10.1.1.1
r3#
```

测试：

```
r3#ping 192.168.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/52/192 ms

r3#

r3#ping 172.16.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/84/156 ms

r3#

说明：可以看见，R3 当前的路由表中，存在一条指向 192.168.1.0 的默认网关，所以会将所有未知目标的数据包发往 192.168.1.0，最终 R3 能够与 192.168.1.0/24 和 172.16.1.0/24 通信。

（5）查看 R3 的路由协议:

r3#sh ip protocols

r3#

说明：ip default-network 与协议无关。

2.配置更多 ip default-network

说明：测试多条 ip default-network

（1）在 R3 上增加 172.16.1.0/24 与 ip default-network 的关联：

r3(config)#ip route 172.16.1.0 255.255.255.0 10.1.1.1

r3(config)#ip default-network 172.16.1.0

说明：配置到 172.16.1.0/24 的静态路由，并通过 ip default-network 指定该网段为默认网关。

（2）查看 R3 当前的路由表情况：

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 192.168.1.0

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

S 172.16.0.0/16 [1/0] via 172.16.1.0

S 172.16.1.0/24 [1/0] via 10.1.1.1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

S* 192.168.1.0/24 [1/0] via 10.1.1.1

r3#

说明：从结果中看出，172.16.1.0/24 并没有成为默认网关，原因是，ip default-network 只支持主类网络，而 172.16.1.0/24 是 172.16.0.0/16 的子网，所以被忽略。

（3）修改 172.16.0.0/16 为默认网关：

r3(config)#no ip default-network 172.16.1.0

r3(config)#ip route 172.16.0.0 255.255.0.0 10.1.1.1

r3(config)#ip default-network 172.16.0.0

说明：增加静态路由 172.16.0.0/16，并指定为默认网关，需要说明的是，ip default-network 所指定的网段必须在路由表中真实存在，所以当前必须手工指定到 172.16.0.0/16 的静态路由。

（4）再次查看 R3 的路由表：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static


route
       o - ODR, P - periodic downloaded static route


Gateway of last resort is 10.1.1.1 to network 172.16.0.0


     20.0.0.0/24 is subnetted, 1 subnets
C       20.1.1.0 is directly connected, FastEthernet0/1
 *   172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
S*      172.16.0.0/16 [1/0] via 10.1.1.1
S       172.16.1.0/24 [1/0] via 10.1.1.1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
S*   192.168.1.0/24 [1/0] via 10.1.1.1
```


http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

r3#

说明：可以看见，之前的默认网关是 192.168.1.0，而现在已经变成 172.16.0.0，原因是，当路由器上配置多条 ip default-network 后，拥有最低 AD 值的被使用，但当前两条网络 AD 值相同，都为 1，最后比较路由条目在路由表中的上下排列顺序，也就是使用命令 show ip route 时，所显示在最上面的条目被优先使用，正因为 172.16.0.0/16 在 192.168.1.0/24 上面，所以被优先使用。

3.测试 ip default-network 与 RIP 的关联

说明：测试 ip default-network 的默认网关在 RIP 中的传递。

（1）在 R3 与 R4 之间配置 RIP:

R3：

r3(config)#router rip

r3(config-router)#network 20.0.0.0

R4：

r4(config)#router rip

r4(config-router)#network 20.0.0.0

r4(config-router)#exit

（2）在 R3 上指定 10.0.0.0 为默认网关，并查看路由表情况：

r3(config)#ip default-network 10.0.0.0

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

* 10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

r3#

说明：因为路由表中 10.1.1.0/24 是 10.0.0.0/8 的子网，所以并没成为自己的默认网关，但这并不影响协议的自动传递。

需要注意，如果不是直连网段，可能无法传递。

（3）查看 R4 的路由表情况：

r4#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 20.1.1.3 to network 0.0.0.0

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

R* 0.0.0.0/0 [120/1] via 20.1.1.3, 00:00:14, FastEthernet0/1

r4#

说明：R4 已经成功从 RIP 中收到默认网关，并且指向 R3 的方向。

4.测试 ip default-network 与 EIGRP 的关联

说明：测试 ip default-network 的默认网关在 EIGRP 中的传递。

（1）在 R3 与 R4 之间配置 EIGRP:

R3：

r3(config)#router eigrp 1

r3(config-router)#no auto-summary

r3(config-router)#network 20.1.1.3 0.0.0.0

R4：

r4(config)#router eigrp 1

r4(config-router)#no auto-summary

r4(config-router)#network 20.1.1.4 0.0.0.0

（2）在 R3 上指定 192.168.1.0 为默认网关:

r3(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1

r3(config)#ip default-network 192.168.1.0

（3）查看 R3 的路由表情况：

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 192.168.1.0

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

S* 192.168.1.0/24 [1/0] via 10.1.1.1

r3#

说明：R3 已经成功将 192.168.1.0 定为默认网关。

（4）查看 R4 的路由表情况：

r4#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

r4#

说明：因为 ip default-network 指定的网段 192.168.1.0 并没有在 EIGRP 进程中，所以默认网关无法被传递。

（5）R3 将默认网关的网段 192.168.1.0 引入 EIGRP：

r3(config)#router eigrp 1

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

说明：将一条路由导入 EIGRP，可以原本就是 EIGRP 进程的，或者重分布，或者通过命令 network，但 network 的网段必须为直连。

（6）再次查看 R4 的路由表情况：

r4#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 20.1.1.3 to network 192.168.1.0

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

D*EX 192.168.1.0/24 [170/284160] via 20.1.1.3, 00:00:27, FastEthernet0/1

r4#

说明：R4 已经成功收到指向 192.168.1.0 的默认网关。

5.测试 ip default-network 同时在 RIP 与 EIGRP 的关联

说明：测试 ip default-network 的默认网关同时在 RIP 与 EIGRP 中的传递。

（1）在 R3 与 R1 之间配置 RIP，R3 与 R2 之间配置 EIGRP：

R1：

r1(config)#router rip

r1(config-router)#network 192.168.1.0

r1(config-router)#network 10.0.0.0

R2：

r2(config)#router eigrp 1

r2(config-router)#no auto-summary

r2(config-router)#network 192.168.2.2 0.0.0.0

r2(config-router)#network 10.0.0.0

R3：

r3(config)#router rip

r3(config-router)#network 10.0.0.0

r3(config-router)#exit

r3(config)#router eigrp 1

r3(config-router)#network 10.0.0.0

说明：R1 通过 RIP 向 R3 通告 192.168.1.0/24，R2 通过 EIGRP 向 R3 通告 192.168.2.0/24。

（2）在 R3 上将 192.168.1.0 配置为默认网关，并查看路由表：

r3(config)#ip default-network 192.168.1.0

r3(config)#exi

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 192.168.1.0

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R* 192.168.1.0/24 [120/1] via 10.1.1.1, 00:00:14, FastEthernet0/0

D 192.168.2.0/24 [90/156160] via 10.1.1.2, 00:00:42, FastEthernet0/0

r3#

说明：192.168.1.0/24 已经成功成为默认网关。

（3）增加 192.168.2.0/24 为默认网关，并查看路由表：

r3(config)#ip default-network 192.168.2.0

r3(config)#exi

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.2 to network 192.168.2.0

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R* 192.168.1.0/24 [120/1] via 10.1.1.1, 00:00:23, FastEthernet0/0

D* 192.168.2.0/24 [90/156160] via 10.1.1.2, 00:01:17, FastEthernet0/0

r3#

说明：因为当路由器上配置多条 ip default-network 后，拥有最低 AD 值的被使用，由于 EIGRP 通告的 192.168.2.0/24 的 AD 值为 90，而 RIP 通告的 192.168.1.0/24 的 AD 值为 120,所以 192.168.2.0 被优先使用，而忽略了路由条目在路由表中的上下排列顺序。

5.测试 ip default-network 与 ip route 0.0.0.0 0.0.0.0 共存

说明：测试 ip default-network 与 ip route 0.0.0.0 0.0.0.0 共同存在于路由表时，路由器对默认网关的选择。

（1）在 R3 使用命令 ip route 0.0.0.0 0.0.0.0 配置指向 R4 的默认网关，并查看路由表：

r3(config)#ip route 0.0.0.0 0.0.0.0 20.1.1.4

r3(config)#exi

r3#sh ip route

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

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

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

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

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

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


route

        o - ODR, P - periodic downloaded static route


Gateway of last resort is 20.1.1.4 to network 0.0.0.0


     20.0.0.0/24 is subnetted, 1 subnets

C       20.1.1.0 is directly connected, FastEthernet0/1

     10.0.0.0/24 is subnetted, 1 subnets

C       10.1.1.0 is directly connected, FastEthernet0/0

R    192.168.1.0/24 [120/1] via 10.1.1.1, 00:00:10, FastEthernet0/0

D    192.168.2.0/24 [90/156160] via 10.1.1.2, 00:03:57, FastEthernet0/0

S*   0.0.0.0/0 [1/0] via 20.1.1.4

r3#
```

说明：R3 当前的默认网关为 20.1.1.4，即 R4。

（2）在 R3 上配置静态路由到 172.16.0.0/16，并使用命令 ip default-network 指定为默认网关，并查看路由表：

```
r3(config)#ip default-network 172.16.0.0

r3(config)#exi
```

r3#

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 172.16.0.0

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, FastEthernet0/1

S* 172.16.0.0/16 [1/0] via 10.1.1.1

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R 192.168.1.0/24 [120/1] via 10.1.1.1, 00:00:16, FastEthernet0/0

D 192.168.2.0/24 [90/156160] via 10.1.1.2, 00:08:46, FastEthernet0/0

S* 0.0.0.0/0 [1/0] via 20.1.1.4

r3#

说明：因为 ip default-network 后面的网段 172.16.0.0 是通过静态路由指定的，所以优先于 ip route 0.0.0.0 0.0.0.0，最终 172.16.0.0 成为了默认网关，需要注意，只有当 ip default-network 后面的网段是通过静态路由指定时，才优先于 ip route 0.0.0.0 0.0.0.0 被使用。

（3）在 R3 上将 ip default-network 的网段改为通过 RIP 学习到的 192.168.1.0 指定为默认网关，并查看路由表：

r3(config)#no ip default-network 172.16.0.0

r3(config)#ip default-network 192.168.1.0

r3(config)#

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

```
Gateway of last resort is 20.1.1.4 to network 0.0.0.0

     20.0.0.0/24 is subnetted, 1 subnets
C       20.1.1.0 is directly connected, FastEthernet0/1
S    172.16.0.0/16 [1/0] via 10.1.1.1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
R*   192.168.1.0/24 [120/1] via 10.1.1.1, 00:00:13, FastEthernet0/0
D    192.168.2.0/24 [90/156160] via 10.1.1.2, 00:10:05, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 20.1.1.4
r3#
```

说明：因为只有当 ip default-network 后面的网段是通过静态路由指定时，才优先于 ip route 0.0.0.0 0.0.0.0 被使用，而 192.168.1.0/24 是通过 RIP 学习到的，所以此时 ip route 0.0.0.0 0.0.0.0 优先。

Classless 与 Classful

大家都非常清楚，可以将一个 IP 网段划分成多个子网，子网的掩码可以是任意位数，比如将一个 10.0.0.0/8 的大网划分出 10.1.1.0/24，10.1.2.0/24，10.1.3.0/24 等等，划分出来的更小的网络叫做子网，而原来的大网络叫主网，也称为主类网络，A 类地址掩码必须为 8 位才是主类网络，B 类地址掩码必须为 16 位才是主类网络，C 类地址掩码必须为 24 位才是主类网络。

无论是主网还是子网，都会被路由器放入路由表，只是某些路由协议不能精确传递子网而已，但只要路由协议传递了子网和掩码，路由器就一定会将其放入路由表中。

支持子网的功能被称为 Classless，支持 Classless 可以与子网很好的协作，如果不支持子网，则被称为 Classful，所以，一个路由协议是工作在 Classless 还是 Classful，直接关系到路由信息中是否存在精确的子网信息，如 RIP 和 EIGRP，并且这些功能可以在协议中手工开启或关闭。

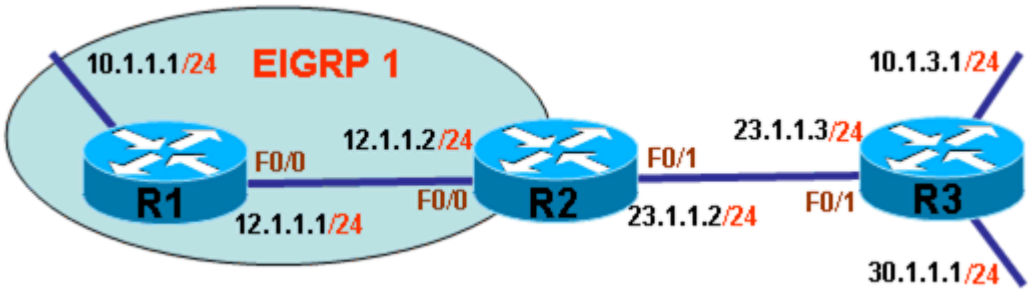
路由协议有 Classless 与 Classful 的说法，而 IOS 本身也有运行在 Classless 还是 Classful 的说法，IOS 是工作在 Classless 还是 Classful，并不影响路由表中是否有子网条目，也就是说，IOS 工作在 Classless 还是 Classful，并不影响路由表的建立，路由表不会有任何区别，但是，Classless 与 Classful 会决定路由器转发数据包的进程，影响如下：

对于某个主类网络，如 10.0.0.0/8，当路由表中存在其中部分子网，如 10.1.1.0/24 和 10.1.2.0/24，当工作在 Classless 时，对于已经知道的子网，路由器会将数据包精确地发送到相应出口，而对于并不知道的子网和其它所有未知目标网络，如 10.1.3.0/24 和 30.1.1.0/24，如果存在默认路由的话，路由器便将他们全部发送到默认路由所指示的出口；但是当路由器工作在 Classful 时，路由器知道了子网 10.1.1.0/24，就始终会认为其它所有 10.0.0.0/8 范围内的子网都应该真实存在于网络中，会认为 10.1.2.0/24、10.1.3.0/24 等等都存在于网络中，只是自己没有详细路由，这时，当路由器收到去往 10.1.1.0/24 的数据包时，可以正常转发，但是如果收到去往 10.1.3.0/24 和 30.1.1.0/24 的数据包，当路由表中存在默认路由时，去往 30.1.1.0/24 的数据包会被发送到默认路由指示的出口，而去往 10.0.0.0/8 中的未知子网 10.1.3.0/24 的数据包则被全部丢弃而不走默认路由。

由以上情况可以看出，路由器工作在 Classful 时，如果知道了某个主类网络中的部分子网后，其它所有未知子网的数据包将被全部丢弃而不转发，即使存在默认路由，也不会转发，而其它主类网络的数据包还是会正常转发。IOS 的 Classless 与 Classful 可以通过命令 ip classless 和 no ip classless 开启或关闭。

配置 Classless 与 Classful

说明：以下图为例，测试 Classless 与 Classful



1。配置基础网络环境

（1）配置 R1：

r1(config)#int f0/0

r1(config-if)#ip add 12.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config-if)#exi

r1(config)#int loopback 10

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config-if)#exit

说明：R1 连接 10.1.1.0/24。

（2）配置 R2：

r2(config)#int f0/0

r2(config-if)#ip add 12.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config-if)#exi

r2(config)#int f0/1

r2(config-if)#ip address 23.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config-if)#exit

（3）配置 R3：

r3(config)#int f0/1

r3(config-if)#ip address 23.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config-if)#exit

r3(config)#int loopback 10

r3(config-if)#ip add 10.1.3.1 255.255.255.0

r3(config)#int loopback 30

r3(config-if)#ip address 30.1.1.1 255.255.255.0

说明：说明：R3 连接 10.1.3.0/24 和 30.1.1.0/24

2.测试基础网络

说明：R1 与 R2 之间启用 EIGRP，R1 将子网 10.1.1.0/24 通告给 R2。

（1）R1 与 R2 之间启用 EIGRP:

R1：

r1(config)#router eigrp 1

r1(config-router)#network 10.0.0.0

r1(config-router)#network 12.0.0.0

r1(config-router)#no auto-summary

R2：

r2(config)#router eigrp 1

r2(config-router)#network 12.0.0.0

r2(config-router)#no auto-summary

（2）查看 R2 的路由表：

r2#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/24 is subnetted, 1 subnets

D 10.1.1.0 [90/156160] via 12.1.1.1, 00:01:46, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

r2#

说明：R2 已经从 EIGRP 学习到 10.1.1.0/24。

（3）分别测试 R2 到 10.1.1.0/24，10.1.3.0/24，30.1.1.0/24 的连通性：

r2#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/87/216 ms

r2#

r2#ping 10.1.3.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.3.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

r2#

r2#ping 30.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 30.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：10.1.1.0/24 可以通信，但由于没有去往 10.1.3.0/24 和 30.1.1.0/24 的路由，所以通信失败。

3.测试 Classless 与 Classful

（1）在 R2 上配置指向 R3 的默认路由:

r2(config)#ip route 0.0.0.0 0.0.0.0 23.1.1.3

（2）测试到 10.1.3.0/24 和 30.1.1.0/24 的连通性：

r2#ping 10.1.3.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.3.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 44/136/240 ms

r2#

r2#

r2#ping 30.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 30.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/104/272 ms

r2#

r2#

说明：因为配置了指向 R3 的默认路由，而 10.1.3.0/24 和 30.1.1.0/24 与 R3 直连，所以通信成功。因为 IOS 默认工作在 Classless。

（3）关闭 IO 的 Classless 功能，并查看路由表：

r2(config)#no ip classless

r2(config)#

r2#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 23.1.1.3 to network 0.0.0.0

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/24 is subnetted, 1 subnets

D 10.1.1.0 [90/156160] via 12.1.1.1, 00:07:23, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

S* 0.0.0.0/0 [1/0] via 23.1.1.3

r2#

说明：可以看见，Classless 与 Classful 并不影响路由表的建立。

（4）再次测试 R3 到 10.1.3.0/24 和 30.1.1.0/24 的连通性：

r2#ping 10.1.3.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.3.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

r2#

r2#ping 30.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 30.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 60/114/196 ms

r2#

说明：结果证明，Classful 让 R2 丢弃了去往 10.0.0.0/8 范围内的子网 10.1.3.0/24 的数据包，而其它主类网络的流量 30.1.1.0/24 则走了默认路由，所以 R2 到

10.1.3.0/24 通信失败，而到 30.1.1.0/24 通信正常。

（5）开启 Classless 后再次测试到 10.1.3.0/24 的连通性：

r2(config)#ip classless

r2(config)#exit

r2#ping 10.1.3.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.3.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/116/260 ms

r2#

说明：由于 R2 开启了 Classless 功能，所以所有未知目标的数据包全部走默认路由，所以最后 R2 到 10.1.3.0/24 通信成功。

默认路由重点总结：

★ip default-gateway 只能为路由器配置一个默认网关，并且只能在关闭路由功能的模式下使用；

★如果使用命令 ip default-network 配置了多条默认网关，拥有最低 AD 值的网段将被选为默认网关，如果多条网段 AD 值全部相同，最后比较路由条目在路由表中的上下排列顺序，也就是使用命令 show ip route 时，所显示在最上面的条目被优先使用；

★如果同时使用 ip default-network 和 ip route 0.0.0.0 0.0.0.0 配置了多条默认网关，ip default-network 会被优先选用，如果 ip default-network 指定的路由是靠动态路由协议学来的，则 ip route 0.0.0.0 0.0.0.0 被优先选用。

★但如果使用多条 ip route 0.0.0.0 0.0.0.0 配置了多条默认网关，并不会只选择其中一条，而是在多条 ip route 0.0.0.0 0.0.0.0 中执行负载均衡。

RIP

概述

RIP 是在小型 TCP/IP 网络中使用最普遍的动态路由协议，是一个稳定的协议。

RIP 是一个内部网关路由协议（Interior Gateway Protocol，即 IGP），只能在单个 AS 内传递路由信息。

RIP 被定义为距离矢量路由协议，而距离矢量路由协议的根本特征就是自己的路由表是完全从其它路由器学来的，并且将收到的路由条目一丝不变地放进自己的路由表，以供数据转发。正因为如此，对于路由是否正确，对于目标是否可达，RIP 全然不知。

RIP 使用跳数作为 metric，跳数就是到达目标网络所需要经过的路由器个数，因为直连网络不需要经过任何路由器，所以直连网络的 metric 为 0。RIP 所支持网络的最大跳数为 15，也就是 metric 值最大为 15，一条大于 15，如 16，被 RIP 认为目标不可达，由此可见，RIP 并不适合大型网络。

RIP 共有两个版本，ver 1 和 ver2，管理距离(Ddministrative Distance)为 120。

RIP 使用 UDP 协议，端口号 520 将路由条目从开启了 RIP 进程的接口上发出，ver 1 使用广播地址 255.255.255.255 发出，而 ver 2 使用组播地址 224.0.0.9 发出。

无论是 ver 1 还是 ver 2，都会将路由表每 30 秒定期向网络中发送，RIP 没有邻居的概念，所以自己并不知道发出去的路由更新是不是有路由器收到，而收到的路由更新，RIP 并不会绝对接受，只有当路由的发送 IP 地址和自己接收的接口 IP 地址处于同网段时，才会接收，否则忽略。如果路由表中的路由超过 180 秒都没有再次收到更新，则被标记为不可用，如果连续 240 秒没收到更新，最后将相应路由从路由表中删除。

RIP 在将路由发出去之前，都会在原来的 metric 值基础上加 1 后发出去，虽然 RIP 采用定期更新，但是当路由发生变化时，也会立即发送更新，如果检测到某路由不可用，则将路由的 metric 值设为 16，然后发出去，这样，接收到的路由器自然会将该路由从自己的路由表中删除。

RIP 有两个版本，发送的路由更新同样分为两个版本 ver 1 和 ver2，如果在配置 RIP 时，没有指定版本，那么默认可以同时接收 ver 1 和 ver 2 的更新，但默认只发送 ver 1，可以通过手工配置接口接收或发送更新版本的能力。

正因为 RIP 发出去的更新任何路由器都可以接收，所以为了安全，RIP 可以加密更新，但只有 ver 2 才可以启用认证，而 ver 1 是不可以的，认证同时支持明文和 MD5。

RIP ver 1 会将收到的路由自动汇总为主类网络，并且无法关闭该功能，为什么 RIP ver1 会自动汇总网络而不可关闭，是因为 RIP 发送的 ver 1 更新中，路由条目只包含 IP 网络号，却没有掩码信息，这是个不可思议的麻烦事情，对于路由信息，RIP ver 1 有一套奇怪的规则，在后面的实验中，将会详细解释。

RIP ver 2 改掉了 RIP ver 1 的某些缺点，就是在发送路由更新时，将路由的掩码一起发送，于是成就了 RIP ver 2 能够支持 Classless Interdomain Routing (CIDR) 和 Variable-Length Subnet Masks (VLSMs)的功能，但 RIP ver 2 默认也会自动汇总，只不过该功能可以手工关闭，同时，RIP 还支持手工汇总路由信息，但手工汇总也是有条件限制的，需要明确说明的是，汇总是针对发出的路由有效，也就是对其它路由器生效。

无论是 RIP ver 1 还是 RIP ver 2，都可以在接口上关闭发送路由的功能，该功能称为 Passive-Interface（即被动接口），两个版本都可以通过 Offset list 来增加路由的 metric，只可以增加，不可以减少。

RIP 也有自己的 Database，会将收到的路由存放于 Database 中，而 RIP 的 Database 并没有特别之处，只是个存放路由的仓库。

RIP 可以使用接口上的 Secondary IP 地址作为更新的源地址。

为了防止引起路由环路，RIP 引入了水平分割（Split Horizon）的机制，即从一个接口收到路由更新，不会再从这个接口将收到的路由发回去。默认情况下，Frame-Felay 主接口是关闭的，而无论是 Frame-Felay 点到点子接口还是多点子接口都是默认开启的，除此之外，普通 HDLC 封装的串口，以太网接口也都是默认开启的。

RIP ver 1 路由更新

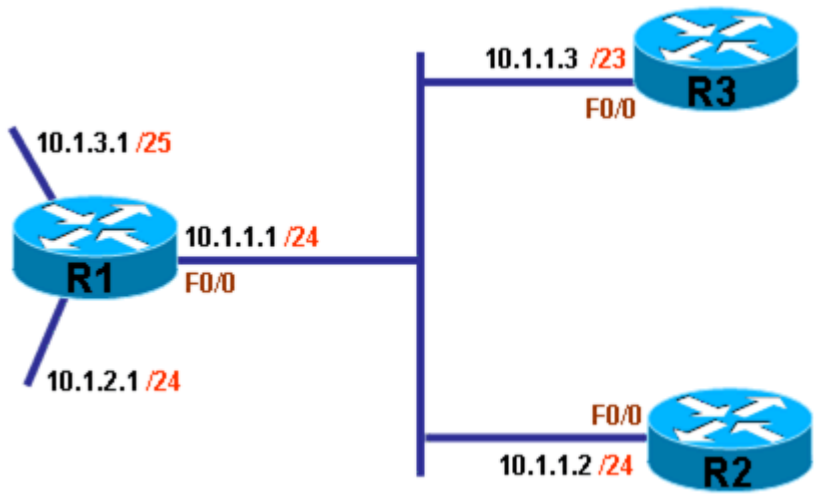
RIP ver 1 使用广播地址 255.255.255.255 发送路由更新，所以同网段任何节点都能收到该更新信息。

RIP ver 1 会将收到的路由自动汇总为主类网络，并且无法关闭该功能，结果为 RIP ver1 的收到的路由条目全部都为主类，即掩码总是为 8 位，16 位，24 位其中的一种，但并不永远都是这样子，因为 RIP 发送的 ver 1 更新中，路由条目只包含 IP 网络号，并没有掩码长度，所以对于收到的路由条目，掩码是接收路由器自己加上去的，由于自动汇总，从而造成所有路由的掩码长度都归为了 8 位，16 位，24 位；但有一种情况是绝对的例外，那就是如果收到的路由条目与接收该路由的接口属于同一主类网络时，那么该路由则被加上与该接收接口 IP 地址相同的掩码长度，比如路由器的接口地址是 10.1.1.2/24，从该接口上收到一条路由 10.1.2.0，则给 10.1.2.0 加上 24 位的掩码长度，而不会自动汇总是 8 位掩码。由此可以看出，RIP 给路由的掩码，只是猜测的，也许该路由并不是自己所猜的掩码长度，因此可能会造成网络中路由表的不精确。但是当接收路由的接口配有 Secondary 地址时，则采用 Secondary 地址的掩码长度赋予接收路由，例如收到的路由为 10.1.2.0，而接口主 IP 地址为 10.1.1.2/24，Secondary IP 地址为 10.1.1.3/25，那么将设定 10.1.2.0 的掩码长度为 25 位，所以 RIP 采用 Secondary IP 的掩码长度优先。

RIP 始终认为与自己直连的对端路由器的 IP 地址和自己是同子网的，认为双方掩码是相同的，其实这也只是 RIP 路由器自己认为而已，它并没有办法知道对方是否真的与自己在同一子网。有时，RIP 希望避免造成不必要的错误，比如自己有条路由条目为 10.1.3.0/25，当要将该路由从某个接口发出去之前，都会做一次检测，如果检测到该接口的 IP 地址与需要发送的路由属于同一主类，那么两者必须拥有相同的掩码长度，才会将路由发出，例如接口 IP 地址为 10.1.1.1/25，要发送的路由为 10.1.3.0/25，这时要发送的路由与接口属于同一主类，并且路由的掩码长度与接口掩码长度也相同，所以可以成功发出 10.1.3.0/25，但是如果接口 IP 地址的掩码长度不是 25 位，例如 IP 地址为 10.1.1.1/24，那么最后因为两者的掩码长度不同，从而放弃发送 10.1.3.0，因为 RIP 认为接口对端的接收者应该和自己属于同一子网，应该是 10.1.1.0/24 的 IP 地址，如果对方收到 10.1.3.0，必定会将掩码定义为 24 位（10.1.3.0/24），所以 RIP 知道将该路由发给对方后，会造成对方路由表错误，因此在明知对方会犯错的情况下，采用不发送该路由的方式来避免对方犯错误。

测试 RIP ver 1 路由更新

说明：以下图为例，测试 RIP ver 1 路由更新



1.配置基础网络环境

（1）配置 R1:

r1(config)#int f0/0

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config-if)#exit

r1(config)#int loopback 10

r1(config-if)#ip add 10.1.2.1 255.255.255.0

r1(config)#int loopback 11

r1(config-if)#ip address 10.1.3.1 255.255.255.128

说明：F0/0 为 10.1.1.0/24 网段，loopback 10 为 10.1.2.0/24 网段，loopback 11 为 10.1.3.0/25 网段。

（2）配置 R2:

r2(config)#int f0/0

r2(config-if)#ip address 10.1.1.2 255.255.255.0

r2(config-if)#no sh

说明：F0/0 为 10.1.1.0/24 网段。

（3）配置 R3:

r3(config)#int f0/0

r3(config-if)#ip address 10.1.1.3 255.255.254.0

r3(config-if)#no sh

说明：F0/0 为 10.1.1.0/23 网段。

2.测试 RIP ver 1 路由更新

（1）同时配置 3 台路由器的 RIP 进程:

r1(config)#router rip

r1(config-router)#network 10.0.0.0

r2(config)#router rip

r2(config-router)#network 10.0.0.0

r3(config)#router rip

r3(config-router)#network 10.0.0.0

说明：在 3 台路由器上开启 RIP，默认发送 ve 1 更新，RIP 在发布网段时，即使只需要发布某子网，但也只能使用主类网络。

（2）查看 R1 发出的 RIP 更新：

r1#debug ip rip

RIP protocol debugging is on

r1#

*Mar 1 00:14:12.939: RIP: sending v1 update to 255.255.255.255 via

FastEthernet0/0 (10.1.1.1)

*Mar 1 00:14:12.939: RIP: build update entries

*Mar 1 00:14:12.939: subnet 10.1.2.0 metric 1

*Mar 1 00:14:14.959: RIP: sending v1 update to 255.255.255.255 via

Loopback10 (10.1.2.1)

*Mar 1 00:14:14.959: RIP: build update entries

*Mar 1 00:14:14.959: subnet 10.1.1.0 metric 1

*Mar 1 00:14:16.287: RIP: sending v1 update to 255.255.255.255 via

Loopback11 (10.1.3.1)

*Mar 1 00:14:16.287: RIP: build update entries - suppressing null update

r1#

说明：RIP 默认向广播地址 255.255.255.255 发送 ve 1 更新，自己直连的网段 10.1.2.0 虽然 metric 值为 0，但在发出去时，都是加 1 再发送的，而 RIP 收到更新时，是不会加 metric 的，收到时是多少，放进路由表还是多少。还可以看出 RIP ve 1 的路由更新中，并没有掩码长度。

（3）查看 R2 收到的 RIP 更新:

r2#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 2 subnets

R 10.1.2.0 [120/1] via 10.1.1.1, 00:00:26, FastEthernet0/0

C 10.1.1.0 is directly connected, FastEthernet0/0

r2#

说明：R2 成功收到了 10.1.2.0，因为自己接口 IP 地址为 10.1.1.2/24，所以赋予了 10.1.2.0 的掩码长度也是 24 位，这与原路由完全相同。对于为什么没有收到 10.1.3.0/25，从之前 R1 上的 debug 就可以看出，由于要发送的路由与接口属于同一主类，但是路由的掩码长度与接口掩码长度不同，所以 R1 就已经放弃了发送 10.1.3.0。

（4）查看 R3 收到的 RIP 更新:

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/23 is subnetted, 2 subnets

R 10.1.2.0 [120/1] via 10.1.1.1, 00:00:16, FastEthernet0/0

C 10.1.0.0 is directly connected, FastEthernet0/0

r3#

说明：R3 也成功收到了 10.1.2.0，2，但因为自己接口 IP 地址为 10.1.1.3/23，所以赋予了 10.1.2.0 的掩码长度也是 23 位，结果这条路由被 R3 认为是 10.1.2.0/23，很显然，这是错误的，所以可以看出，当 RIP 路由器双方 IP 地址在同一主类，但掩码长度不同时，会造成路由表的不精确，或者造成错误。

（5）使用 Secondary 地址让 R3 纠正路由信息：

r3(config-if)#ip add 10.1.1.33 255.255.255.0 secondary

查看路由：

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

R 10.1.2.0/24 [120/1] via 10.1.1.1, 00:00:03, FastEthernet0/0

C 10.1.1.0/24 is directly connected, FastEthernet0/0

C 10.1.0.0/23 is directly connected, FastEthernet0/0

r3#

说明：可以看见，当接收路由的接口配有 Secondary 地址时，优先采用了 Secondary 地址的掩码长度赋予接收到的路由，最终 10.1.2.0 被纠正为 10.1.2.0/24，该信息现在与原路由完全相同。

（6）测试 R2 添加 Secondary 地址后的情况：

r2(config)#int f0/0

r2(config-if)#ip add 10.1.1.22 255.255.255.128 secondary

```
r2#sh ip route

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

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

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

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

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

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


route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

R       10.1.2.0/25 [120/1] via 10.1.1.1, 00:00:02, FastEthernet0/0

C       10.1.1.0/25 is directly connected, FastEthernet0/0

C       10.1.1.0/24 is directly connected, FastEthernet0/0

r2#
```

说明：在 R2 给接口配置 Secondary 地址为 10.1.1.22/25 后，由于优先采用了 Secondary 地址的掩码长度赋予接收到的路由，所以原本的 10.1.2.0/24 变成了 10.1.2.0/25。

（7）查看路由变化时的状态：

r1#debug ip rip

RIP protocol debugging is on

r1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

r1(config)#int loopback 10

r1(config-if)#shutdown

*Mar 1 00:24:41.887: %LINK-5-CHANGED: Interface Loopback10, changed state to

administratively down

*Mar 1 00:24:41.895: RIP: sending v1 flash update to 255.255.255.255 via

FastEthernet0/0 (10.1.1.1)

*Mar 1 00:24:41.895: RIP: build flash update entries

*Mar 1 00:24:41.895: subnet 10.1.2.0 metric 16

*Mar 1 00:24:41.899: RIP: sending v1 flash update to 255.255.255.255 via

Loopback11 (10.1.3.1)

*Mar 1 00:24:41.899: RIP: build flash update entries - suppressing null

update

*Mar 1 00:24:42.887: %LINEPROTO-5-UPDOWN: Line protocol on Interface

Loopback10, changed state to down

*Mar 1 00:24:44.079: RIP: received v1 update from 10.1.1.3 on

FastEthernet0/0

*Mar 1 00:24:44.083: 10.1.2.0 in 16 hops (inaccessible)

*Mar 1 00:24:44.087: RIP: received v1 update from 10.1.1.2 on

FastEthernet0/0

*Mar 1 00:24:44.087: 10.1.2.0 in 16 hops (inaccessible)

r1(config-if)#exi

说明：因为 RIP 所支持网络的最大跳数为 15，而 16 跳被 RIP 认为目标不可达，便会从路由表中删除，所以 RIP 如果检测到某路由不可用，则将路由的 metric 值设为 16，然后发出去，这样，接收到的路由器自然会将该路由从自己的路由表中删除。上面 R1 上 10.1.2.0 的接口 shutdown 后，R1 立即将 10.1.2.0 设置 metric 为 16 后发出去。

RIP ver 1 主机路由

我们都知道，IP 地址由网络位和主机位组成，网络位表示某个网络，而主机位则表示该网络中的某台主机，如果我们要表示某个网络，就将主机位全部变成 0，例如 10.1.0.0/16，因为最后 16 个主机位全部为 0，所以 10.1.0.0/16 是个网络地址；

如果主机位不全为 0，只要任何一位为 1，则表示主机地址，而不是网络地址，例如 10.1.1.0/16，可以看见最后 16 个主机位中，并不全部为 0，因为主机位 1.0 变成二进制为:

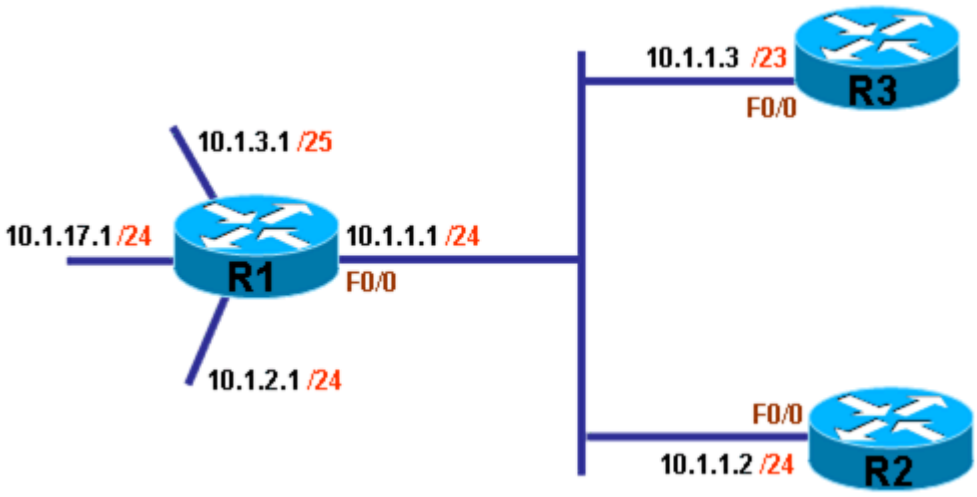
0000 0001 0000 0000，

其中有一个 bit 为 1，所以 10.1.1.0/16 是主机地址。

路由表是用来存放网段信息的，所有路由表中的内容都应该是网络地址，而不应该有主机地址，但是路由表并不是不允许主机地址存在于路由表中，当路由协议收到的路由更新为一个主机地址时，也就是主机位不是全 0 的条目，那么该条目被认为是一个主机地址，以 32 位的掩码存放在路由表中。

RIP 协议在收到主机位不全为 0 的路由信息时，同样会认为是主机地址，从而以 32 位的掩码存放在路由表中。由于 RIP ver 1 的路由条目中并不包含掩码长度，所以也就并不知道网络位是哪部分，主机位又是哪部分，因此，如果收到的路由与接收接口不属于同一主类，则一律使用主类地址来检测，但如果收到的路由与接收接口属于同一主类，则以该接口 IP 地址的掩码长度来检测，最后计算出是否是主机地址，如果是，就以 32 位的掩码存放在路由表中。

测试 RIP ver 1 主机路由



说明：以类似上一个实验的拓朴，来测试 RIP ver 1 主机路由

1.增加测试网段

说明：以上一个实验的拓朴为基础，测试 RIP ver 1 主机路由，网络基础配置同上一个实验，并且已经全部启用 RIP。

（1）在 R1 上增加测试网段：

r1(config)#int loopback 17

r1(config-if)#ip address 10.1.17.1 255.255.255.0

说明：在 R1 上增加 10.1.17.0/24 网段。

2.测试主机路由

（1）查看 R3 的路由情况:

```
r3#sh ip route

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

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

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

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

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

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


route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

R       10.1.2.0/23 [120/1] via 10.1.1.1, 00:00:20, FastEthernet0/0

C       10.1.0.0/23 is directly connected, FastEthernet0/0

R       10.1.17.0/32 [120/1] via 10.1.1.1, 00:00:20, FastEthernet0/0

r3#
```

说明：因为 R3 的接收接口地址为 23 位掩码长度，并且收到的路由 10.1.17.0 与该接收接口地址属于同一主类，所以使用 23 位掩码长度来计算 10.1.17.0 是否属于主机路由，过程为：

23 位掩码长度的主机位为 9 位，那么查看收到的路由条目最后 9 位是否全为 0，如果不是，则为主机路由，而 10.1.17.0 的最后 16 位 17.0 换算成二进制为:

0001 0001 0000 0000

很明显，后面 9 位为 100000000，不是全部都为 0，所以 10.1.17.0 被 R3 认为是主机路由，从而以 32 位长度的掩码放在路由表中，由此可见，RIP 路由器双方如果掩码不匹配，则会造成路由表不精确或路由表错误。

（2）查看 R2 的路由情况

r2#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 3 subnets

R 10.1.2.0 [120/1] via 10.1.1.1, 00:00:19, FastEthernet0/0

C 10.1.1.0 is directly connected, FastEthernet0/0

R 10.1.17.0 [120/1] via 10.1.1.1, 00:00:19, FastEthernet0/0

r2#

说明：R2 接收到 10.1.2.0，而接收接口掩码长度为 24 位，所以以 24 位长度计算，10.1.2.0 不属于主机路由，最终以 10.1.2.0/24 存放在路由表中，该信息完全正确。

（3）继续在 R1 增加主机路由:

r1(config)#int loopback 17

r1(config-if)#ip address 10.1.31.1 255.255.255.0 secondary

r1(config-if)#ip address 10.1.41.1 255.255.255.0 secondary

说明：R1 增加了 10.1.31.0 和 10.1.41.0。

（4）再次查看 R3 的路由表情况：

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

R 10.1.2.0/23 [120/1] via 10.1.1.1, 00:00:17, FastEthernet0/0

C 10.1.0.0/23 is directly connected, FastEthernet0/0

R 10.1.31.0/32 [120/1] via 10.1.1.1, 00:00:16, FastEthernet0/0

R 10.1.17.0/32 [120/1] via 10.1.1.1, 00:00:17, FastEthernet0/0

R 10.1.41.0/32 [120/1] via 10.1.1.1, 00:00:11, FastEthernet0/0

r3#

说明：同上原理，R3 以 23 位掩码计算 10.1.31.0 和 10.1.41.0，结果两条路由的 9 个主机位并不全为 0，所以被当作主机路由，要产生主机路由，可以手工配置对于接收者来说主机位并不是全为 0 的网段即要。

（5）再次查看 R2 的路由表情况：

r2#sh ip route

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

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

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

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

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

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


route

        o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     10.0.0.0/24 is subnetted, 5 subnets

R       10.1.2.0 [120/1] via 10.1.1.1, 00:00:08, FastEthernet0/0

C       10.1.1.0 is directly connected, FastEthernet0/0

R       10.1.31.0 [120/1] via 10.1.1.1, 00:00:08, FastEthernet0/0

R       10.1.17.0 [120/1] via 10.1.1.1, 00:00:08, FastEthernet0/0

R       10.1.41.0 [120/1] via 10.1.1.1, 00:00:08, FastEthernet0/0

r2#
```

说明：路由表正常，计算方法同上，不再解释。

RIP 路由更新源

RIP 会每隔 30 秒定期使用广播或者组播向开启了 RIP 的接口上发出路由更新，RIP 没有邻居的概念，所以自己并不知道发出去的路由更新被什么样的路由器收到，从之前的实验我们可以看出，当 RIP 路由器相互之间 IP 地址或子网掩码长度不匹配时，都会造成路由不精确或者路由表错误，因此，RIP 收到的路由更新源地址必

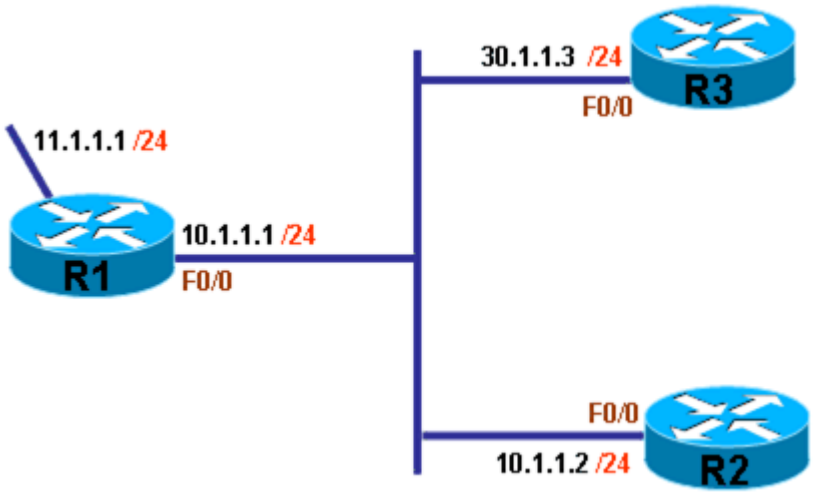
须和接收接口处于相同子网，即使是源地址和接收接口的 Secondary 地址处于相同子网也可以，否则忽略收到的路由更新，该功能称为 RIP 的更新源有效性，可以手动关闭或打开，默认为开启状态。

当接口上配置 Secondary 地址并发布到 RIP 进程后，所有路由信息从该接口发出去时，会同时使用主 IP 地址与 Secondary 地址分别发送一次。

注：接口上 Secondary 地址可以作为路由更新的源地址，但 Secondary 地址的网段不会被 RIP 更新出去。

测试 RIP 路由更新源

说明：该功能默认为开启状态，RIP ver 1 与 ver 2 均适用。



说明：以上图为例，测试 RIP 路由更新源

1.配置基础网络环境

（1）配置 R1:

r1(config)#int loopback 11

r1(config-if)#ip add 11.1.1.1 255.255.255.0

r1(config)#int f0/0

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#router rip

r1(config-router)#network 11.0.0.0

r1(config-router)#network 10.0.0.0

说明：在 R1 上配置 10.1.1.0/24 和 11.0.0.0/24，并发布到 RIP 中。

（2）配置 R2:

r2(config)#int f0/0

r2(config-if)#ip add 10.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#router rip

r2(config-router)#network 10.0.0.0

说明：在 R2 上配置 10.1.1.0/24，并发布到 RIP 中。

（3）配置 R3:

r3(config)#int f0/0

r3(config-if)#ip add 30.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#router rip

r3(config-router)#network 30.0.0.0

说明：在 R3 上配置 10.1.1.0/24，并发布到 RIP 中。

2.测试路由

（1）查看 R2 的路由情况：

r2#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R 11.0.0.0/8 [120/1] via 10.1.1.1, 00:00:29, FastEthernet0/0

r2#

说明：因为 R1 与 R2 接口地址都在 10.1.1.0/24 网段，所以满足条件，R2 正常收到路由。

（2）查看 R3 的路由情况：

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

30.0.0.0/24 is subnetted, 1 subnets

C 30.1.1.0 is directly connected, FastEthernet0/0

r3#

说明：因为 R1 发出的路由更新的源地址在 10.1.1.0，而 R3 的接收接口地址在 30.1.1.0，双方处于不同子网，所以 R3 并没有认可 R1 发来的路由。

以下是 R3 上的 debug 信息：

r3#debug ip rip

RIP protocol debugging is on

r3#

*Mar 1 00:08:32.219: RIP: sending v1 update to 255.255.255.255 via

FastEthernet0/0 (30.1.1.3)

*Mar 1 00:08:32.219: RIP: build update entries - suppressing null update

*Mar 1 00:08:35.003: RIP: ignored v1 update from bad source 10.1.1.1 on

FastEthernet0/0

r3#

说明：R3 因为 R1 不合法的源地址，所以忽略了从 10.1.1.0 发来的路由更新。

3.解决路由接收

（1）在 R3 上使用 Secondary 地址解决:

r3(config)#int f0/0

r3(config-if)#ip address 10.1.1.3 255.255.255.0 secondary

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R 11.0.0.0/8 [120/1] via 10.1.1.1, 00:00:10, FastEthernet0/0

30.0.0.0/24 is subnetted, 1 subnets

C 30.1.1.0 is directly connected, FastEthernet0/0

r3#

说明：因为 R3 在接收接口上配置与 R1 更新源地址同子网的 10.1.1.0 的地址，所以源地址和接收接口的 Secondary 地址处于相同子网，最后将接收到的路由更新放入路由表中。

（2）关闭更新源有效性检测:

r3(config)#int f0/0

r3(config-if)#no ip address 10.1.1.3 255.255.255.0 secondary

r3(config)#router rip

r3(config-router)#no va

r3(config-router)#no validate-update-source

说明：R3 关闭了更新源有效性检测，并去除了 Secondary 地址，再次查看路由表：

r3#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R 11.0.0.0/8 [120/1] via 10.1.1.1, 00:00:02

30.0.0.0/24 is subnetted, 1 subnets

C 30.1.1.0 is directly connected, FastEthernet0/0

r3#

说明：当关闭了更新源有效性检测后，所有接收到的路由更新即使地址不合法，也都被放入路由表

（3）在 R3 上配置 Secondary 地址解决:

r3(config)#router rip

r3(config-router)#validate-update-source

r1(config)#int f0/0

r1(config-if)#ip address 30.1.1.1 255.255.255.0 secondary

1(config)#router rip

r1(config-router)#network 30.0.0.0

说明：在 R1 上配置 Secondary 地址，并放入 RIP 进程，同时在 R3 上恢复更新源有效性检测。

（4）再次查看 R3 的路由接收情况:

r3#debug ip rip

RIP protocol debugging is on

*Mar 1 00:17:20.971: RIP: ignored v1 update from bad source 10.1.1.1 on

FastEthernet0/0

*Mar 1 00:17:20.971: RIP: received v1 update from 30.1.1.1 on

FastEthernet0/0

*Mar 1 00:17:20.971: 11.0.0.0 in 1 hops

r3#

```
r3#sh ip route

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

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

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

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

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

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


route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


R     11.0.0.0/8 [120/1] via 30.1.1.1, 00:00:27, FastEthernet0/0

      30.0.0.0/24 is subnetted, 1 subnets

C        30.1.1.0 is directly connected, FastEthernet0/0

r3#
```

说明：因为 R1 同时使用了接口主地址 10.1.1.1 与 Secondary 地址 30.1.1.1 各发了一份路由更新，R3 虽然忽略了 10.1.1.1 发来的更新，但接收了 30.1.1.1 的更新，所以从 30.1.1.1 收到的路由被放入了路由表中。

（5）使 R2 接收所有路由:

r2(config)#router rip

r2(config-router)#no validate-update-source

r2#sh ip route

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

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

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

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

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

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

route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R 11.0.0.0/8 [120/1] via 30.1.1.1, 00:00:06

[120/1] via 10.1.1.1, 00:00:06, FastEthernet0/0

r2#

说明：R2 关闭了更新源有效性检测后，同时接收了 10.1.1.1 和 30.1.1.1 发来的路由更新，

并且可以得知，R1 接口上 Secondary 地址作为了路由更新的源地址，但 Secondary 地址的网段并没有被 RIP 更新出去。

RIP 触发更新

由于无论是 RIP ver 1 还是 ver 2，都会将路由表每隔 30 秒定期向网络中发送，但通常情况下，广域网链路带宽相对比较低速，而且带宽宝贵，为了能够更好的利用和节省广域网上的带宽，RIP 可以调整为在广域网链路上抑制路由更新的发送，而只仅仅在路由有变化时，将有变化的路由发出去，这就是 RIP 的 触发更新机制，并且是基于接口开启或关闭的。

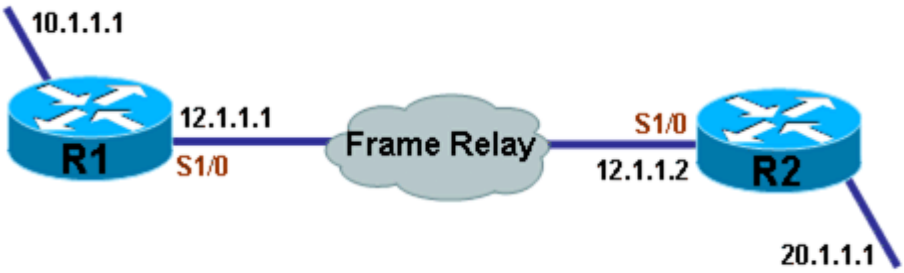
因为 RIP 是网络早期开发的，在早期时候，广域网都是串行链路，或者说是点到点链路，所以 RIP 触发更新只支持在点到点链路上开启或关闭，对于 Frame-Relay 和以太网这样的多路访问接口中，不支持 RIP 触发更新，但是 Frame-Relay 点到点子接口被 RIP 认为是点到点链路，可以开启触发更新。

因为 RIP 的路由会每隔 30 秒更新一次，如果路由表中的路由超过 180 秒都没有再次收到更新，则被标记为不可用，如果连续 240 秒没收到更新，将会从路由表中删除，所以当某台 RIP 路由器开启触发更新后，在路由没有变动的情况下，便不再向对端发送路由更新，这样势必会造成对端路由器在 240 秒之后将收到的路由从路由表中删除，为了杜绝此类问题，

RIP 触发的更新机制需要在两端路由器都开启，否则不生效。在双方都开启后，相互收到的路由都会被注明永久有效(permanent)而不需要再次收到更新。

注：RIP 触发更新支持 ver 1 和 ver 2。

测试 RIP 触发更新



说明：以上图为例，测试测试 RIP 触发更新

1.配置基础网络环境

（1）配置 R1:

r1(config)#int serial 1/0

r1(config-if)#encapsulation frame-relay

r1(config-if)#no frame-relay inverse-arp

r1(config-if)#no arp frame-relay

r1(config-if)#no ip address

r1(config-if)#no shutdown

r1(config)#int serial 1/0.12 point-to-point

r1(config-subif)#ip address 12.1.1.1 255.255.255.0

r1(config-subif)#frame-relay interface-dlci 102

r1(config)#int loopback 10

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config)#router rip

r1(config-router)#network 12.0.0.0

r1(config-router)#network 10.0.0.0

说明：在 R1 上配置网段 12.1.1.0 和 10.1.1.0，因为 R1 和 R2 通过 Frame-Felay 连接，而 Frame-Felay 只有点到点子接口才被认为是点到点链路。

（2）配置 R2:

r2(config)#int serial 1/0

r2(config-if)#en frame-relay

r2(config-if)#no frame-relay inverse-arp

r2(config-if)#no arp frame-relay

r2(config-if)#no ip address

r2(config-if)#no shutdown

r2(config)#int serial 1/0.12 point-to-point

r2(config-subif)#ip address 12.1.1.2 255.255.255.0

r2(config-subif)#frame-relay interface-dlci 201

r2(config)#int loopback 20

r2(config-if)#ip address 20.1.1.1 255.255.255.0

r2(config)#router rip

r2(config-router)#network 12.0.0.0

r2(config-router)#network 20.0.0.0

说明：在 R2 上配置网段 12.1.1.0 和 20.1.1.0，并放入 RIP 进程。

2 .测试 RIP 触发更新

（1）查看 R1 与 R2 当前的路由表情况：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R 20.0.0.0/8 [120/1] via 12.1.1.2, 00:00:15, Serial1/0.12

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Loopback10

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0.12

r1#

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

20.0.0.0/24 is subnetted, 1 subnets

C 20.1.1.0 is directly connected, Loopback20

R 10.0.0.0/8 [120/1] via 12.1.1.1, 00:00:11, Serial1/0.12

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0.12

r2#

说明：R1 与 R2 当前路由表正常，都能成功收到对方发来的路由。

（2）在 R1 接口上开启 RIP 触发更新：

r1(config)#int serial 1/0.12

r1(config-subif)#ip rip triggered

说明：在接口上成功输入 RIP 触发更新的命令，并不代表 RIP 触发更新就已经生效。

（3）在 R1 上查看 RIP 触发更新是否生效：

r1#sh ip protocols

Routing Protocol is "rip"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Sending updates every 30 seconds, next due in 12 seconds

Invalid after 180 seconds, hold down 0, flushed after 240

Redistributing: rip

Default version control: send version 1, receive any version

Interface	Send	Recv	Triggered RIP	Key-chain
Serial1/0.12	1	1 2		
Loopback10	1	1 2		

Automatic network summarization is in effect

Maximum path: 4

Routing for Networks:

10.0.0.0

12.0.0.0

Routing Information Sources:

Gateway	Distance	Last Update
12.1.1.2	120	00:00:14

Distance: (default is 120)

r1#

说明：并没有任何信息显示 RIP 触发更新已生效。

（4）在 R2 上也开启 RIP 触发更新：

r2(config)#int serial 1/0.12

r2(config-subif)#ip rip tri

r2(config-subif)#ip rip triggered

说明：在 R2 上也开启了 RIP 触发更新，保持双方一致。

（5）再次查看 RIP 触发更新是否生效:

```
r1#sh ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 3 seconds
  Invalid after 180 seconds, hold down 0, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
   Interface          Send  Recv  Triggered RIP  Key-chain
   Serial1/0.12       1     1 2   Yes
   Loopback10         1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
   10.0.0.0
   12.0.0.0
  Routing Information Sources:
   Gateway         Distance      Last Update
   12.1.1.2              120      00:00:26
```


http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

Distance: (default is 120)

r1#

说明：在双方路由器都开启 RIP 触发更新后，输出结果显示 RIP 触发更新已生效。

（6）再次查看 R1 的路由表情况：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

R 20.0.0.0/8 [120/1] via 12.1.1.2, 00:05:18, Serial1/0.12

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Loopback10

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0.12

r1#

说明：从路由表中可以看出，RIP 收到的路由 20.0.0.0/8 在 5 分 18 秒还没再次收到更新的情况下，也没有从路由表中删除，因为开了 RIP 触发更新，不会再收到路由更新。

（7）查看 R1 的 RIP 数据库状态：

r1#sh ip rip da

r1#sh ip rip database

10.0.0.0/8 auto-summary

10.1.1.0/24 directly connected, Loopback10

12.0.0.0/8 auto-summary

12.1.1.0/24 directly connected, Serial1/0.12

20.0.0.0/8 auto-summary

20.0.0.0/8

[1] via 12.1.1.2, 00:05:24 (permanent), Serial1/0.12

* Triggered Routes:

- [1] via 12.1.1.2, Serial1/0.12

r1#

说明：结果显示 R1 收到的路由 20.0.0.0/8 当前为触发更新状态，所以注明该路由不需要更新，而是永久有效（permanent）。

（8）查看 R2 的路由表与 RIP 数据库状态：

```
r2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     20.0.0.0/24 is subnetted, 1 subnets
C       20.1.1.0 is directly connected, Loopback20
R    10.0.0.0/8 [120/1] via 12.1.1.1, 00:05:38, Serial1/0.12
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, Serial1/0.12
r2#
r2#sh ip rip da
r2#sh ip rip database
10.0.0.0/8    auto-summary
```

```
10.0.0.0/8
    [1] via 12.1.1.1, 00:05:42 (permanent), Serial1/0.12
  * Triggered Routes:
    - [1] via 12.1.1.1, Serial1/0.12
12.0.0.0/8    auto-summary
12.1.1.0/24    directly connected, Serial1/0.12
20.0.0.0/8    auto-summary
20.1.1.0/24    directly connected, Loopback20
r2#
```

说明：同样，R2 从 R1 收到的路由 10.0.0.0/8 在 5 分 38 秒没有再次收到更新，也没有从路由表中删除，数据库中也显示该路由为触发更新路由，并且为永久有效。

RIP 单播更新

RIP ver 1 只能工作在 Classful 模式下，虽然 RIP ver 2 可以工作在 Classless 下，但是在使用命令 network 发布网段时，都只能发布主类网络，例如路由器上有三个网段，分别为 10.1.1.0/24，10.1.2.0/24，10.1.3.0/24，如果要将 10.1.1.0/24 放入 RIP 进程，只能使用命令 network 10.0.0.0，因为使用 network 10.1.1.0 和 network 10.0.0.0 是同样的结果，而 10.0.0.0 将路由器上三个网段全部包含在内了，所以在配置 RIP 时，只能将所有接口都放入进程，但是可以想象，只想将单个接口放入进程，结果却是将所有接口放入进程，这种做法是不理想的，因为这种配置不仅将某些不必要的网段通告给了其它路由器，同时也在不必要的接口上周期性发送路由更新。为了解决上述问题，RIP 可以抑制从某个接口发送更新，称为 RIP 被动接口（Passive-Interface）这不同于 RIP 触发更新，因为开启触发更新的接口，只

是平时不向该接口发送路由更新，而在路由在变动时，同样会发送，但 RIP 被动接口则是永远不再向该接口发送更新。

RIP ver 1 使用广播更新，RIP ver 2 使用组播更新，除此之外，无论是 RIP ver 1 还是 ver 2，都可以选择使用单播地址更新，就是将路由更新的目的地址使用单播地址来代替广播和组播，如果开启单播更新之后，RIP 除了使用单播更新外，还会继续使用原来的组播和广播更新，也就是开启单播更新后，RIP 的路由更新没有减少，反而增加了，这样就变成了掩耳盗铃，但是不用担心，在这里，有个特殊的情况，就是 RIP 被动接口虽然可以抑制从某个接口发送路由更新，但是被动接口不能抑制单播更新，只能抑制广播和组播，所以答案很明显，我们在采用单播更新的时候，可以利用被动接口消除其它不必要的路由更新。

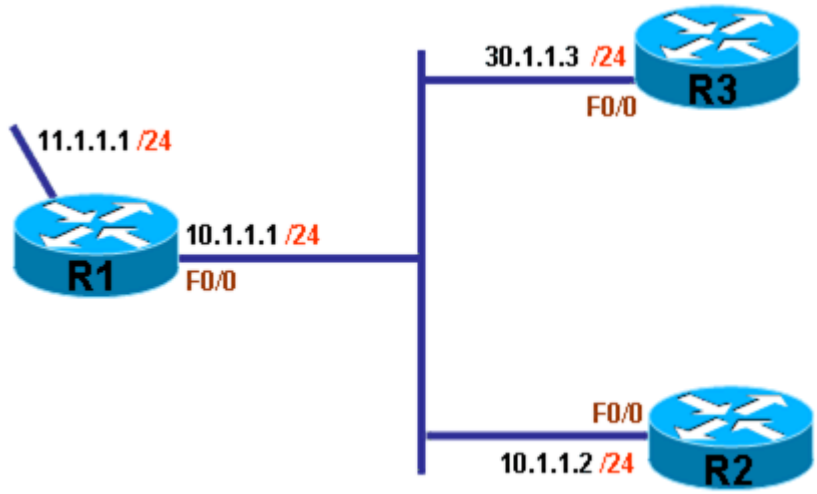
注：

★RIP 被动接口可以对单个接口生效，也可以对 RIP 进程下的所有接口生效，开启了被动接口后，该接口虽然不向外发送路由更新，但依然可以接收路由更新。

★单播更新的目标地址必须在自己的直连网段，否则不会发送更新。

测试 RIP 单播更新

说明：以下图为例，测试 RIP 单播更新。



1.配置基础网络环境

（1）配置 R1：

r1(config)#int f0/0

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#int loopback 11

r1(config-if)#ip add 11.1.1.1 255.255.255.0

r1(config)#router rip

r1(config-router)#network 10.0.0.0

r1(config-router)#network 11.0.0.0

说明：在 R1 上配置 11.1.1.0/24 和 10.1.1.0/24，并放入 RIP 进程。

鸿鹄培训+考试群：301283867

（2）配置 R2：

r2(config)#int f0/0

r2(config-if)#ip add 10.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#router rip

r2(config-router)#network 10.0.0.0

说明：在 R1 上配置 10.1.1.0/24，并放入 RIP 进程。

（3）配置 R3：

r3(config)#int f0/0

r3(config-if)#ip address 10.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#router rip

r3(config-router)#network 10.0.0.0

说明：在 R1 上配置 10.1.1.0/24，并放入 RIP 进程。

2.测试 RIP 单播更新

（1）查看 R1 发送的 RIP 更新：

r1#debug ip rip

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

RIP protocol debugging is on

r1#

*Mar 1 00:09:47.615: RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (10.1.1.1)

*Mar 1 00:09:47.615: RIP: build update entries

*Mar 1 00:09:47.615: network 11.0.0.0 metric 1

r1#

说明：默认情况下，RIP 使用广播地址发送 ver 1 路由更新。

（2）在 R1 开启单播更新：

r1(config)#router rip

r1(config-router)#neighbor 10.1.1.2

说明：单播向 10.1.1.2 发送更新。

（3）再次查看 R1 发送的 RIP 更新：

r1#debug ip rip

RIP protocol debugging is on

r1#

*Mar 1 00:39:05.859: RIP: sending v1 update to 255.255.255.255 via

FastEthernet0/0 (10.1.1.1)

*Mar 1 00:39:05.859: RIP: build update entries

*Mar 1 00:39:05.859: network 11.0.0.0 metric 1

*Mar 1 00:39:05.863: RIP: sending v1 update to 10.1.1.2 via FastEthernet0/0 (10.1.1.1)

*Mar 1 00:39:05.863: RIP: build update entries

*Mar 1 00:39:05.863: network 11.0.0.0 metric 1

*Mar 1 00:39:07.563: RIP: sending v1 update to 255.255.255.255 via Loopback11 (11.1.1.1)

*Mar 1 00:39:07.563: RIP: build update entries

*Mar 1 00:39:07.563: network 10.0.0.0 metric 1

r1#

说明：RIP 虽然向 10.1.1.2 发送单播更新，更是广播更新依然存在，这让单播更新毫无优势。

（4）开启 RIP 被动接口：

r1(config)#router rip

r1(config-router)#passive-interface default

说明：因为 RIP 已经开启单播更新，所以开启被动接口，过滤掉不必要的更新。命令 passive-interface default 在所有 RIP 接口上开启被动接口。

（5）再次查看 R1 发送的 RIP 更新：

r1#debug ip rip

RIP protocol debugging is on

r1#

*Mar 1 00:13:25.371: RIP: sending v1 update to 10.1.1.2 via FastEthernet0/0 (10.1.1.1)

*Mar 1 00:13:25.371: RIP: build update entries

*Mar 1 00:13:25.371: network 11.0.0.0 metric 1

r1#

说明：虽然在所有 RIP 接口上开启了被动接口，抑制发广播发送路由更新，但向 10.1.1.2 的单播更新照样工作正常。

（6）查看 R2 的路由表：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R 11.0.0.0/8 [120/1] via 10.1.1.1, 00:00:03, FastEthernet0/0

r2#

说明：虽然 R1 在所有 RIP 接口上开启了被动接口，但 R1 还是继续向 R2 单播发送路由更新，所以 R2 收到了 R1 发来的路由。

（7）查看 R3 的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

R 11.0.0.0/8 [120/1] via 10.1.1.1, 00:02:05, FastEthernet0/0

r3#

说明：因为 R1 在所有 RIP 接口上开启了被动接口，但 R1 并没有向 R3 单播发送路由更新，所以 R3 没有收到了 R1 发来的任何路由。

RIP 手工汇总

RIP ver 1 在发送路由更新时，不带掩码信息，在收到路由更新后，自动汇总为主类网络，并且无法关闭，虽然 RIP ver 2 在发送路由更新时，带了掩码信息，但默认也将所有收到的路由汇总为主类网络，不过 RIP ver 2 的自动汇总可以关闭。正因为 RIP ver 2 的路由更新中带了掩码长度，所以在发送路由信息时，可以手工汇总到任意比特位，从而缩小路由表的空间。

虽然 RIP ver 2 可以将路由手工汇总到任意比特位，但还是存在一定的限制条件，不能将一条路由的掩码位数汇总到短于自身主类网络的掩码长度，就是不能将 C 类地址汇总到短于 24 位的掩码长度，不能将 B 类地址汇总到短于 16 位的掩码长度，不能将 A 类地址汇总到短于 8 位的掩码长度，例如只能将 172.16.1.0/24 汇总到 172.16.0.0/16，但不能汇总到 172.16.0.0/15，因为该网络为 B 类地址，所以掩码长度不能短于 16 位。

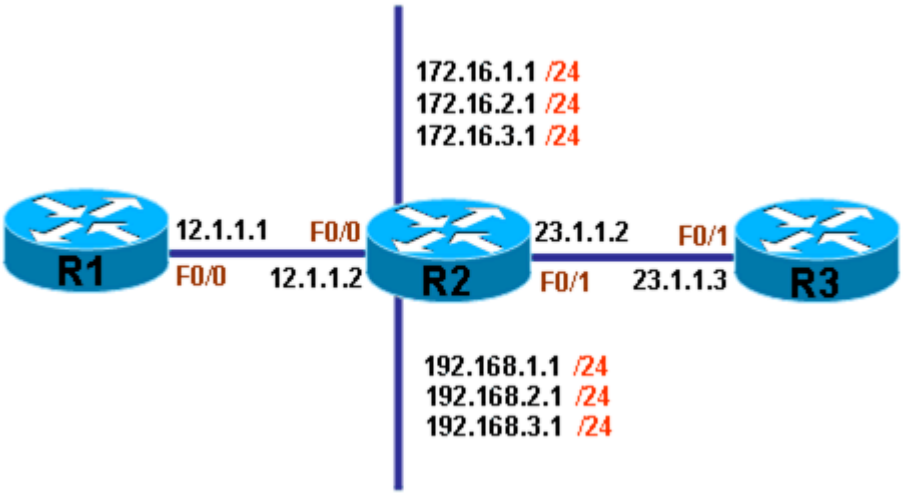
注：RIP 的自动汇总，是将路由汇总后发给邻居，而不是将收到的路由汇总后转给自己，所以自动汇总是对其它路由器产生效果，是对其它路由器的路由表生效。

RIP 手工汇总和自动汇总相似，也是将路由汇总后发给邻居，手工汇总的效果也是在其它路由器上生效，自己并不能看见效果，手工汇总是基于接口配置的。

如果自动汇总和手工汇总同时存在，则自动汇总优先，也就是说，还是只发送主类网络。

测试 RIP 手工汇总

说明：以下图为例，测试 RIP 手工汇总：



1.配置基础网络环境：

（1）配置 R1:

r1(config)#int f0/0

r1(config-if)#ip address 12.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#router rip

r1(config-router)#network 12.0.0.0

说明：在 R1 上配置 12.1.1.0/24，并将网段放入 RIP 进程。

（2）配置 R2:

r2(config)#int f0/0

r2(config-if)#ip add 12.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config)#int f0/1

r2(config-if)#ip address 23.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config)#int loopback 172

r2(config-if)#ip address 172.16.1.1 255.255.255.0

r2(config-if)#ip address 172.16.2.1 255.255.255.0 secondary

r2(config-if)#ip address 172.16.3.1 255.255.255.0 secondary

r2(config)#int loopback 192

r2(config-if)#ip add 192.168.1.1 255.255.255.0

r2(config-if)#ip add 192.168.2.1 255.255.255.0 secondary

r2(config-if)#ip add 192.168.3.1 255.255.255.0 secondary

r2(config)#router rip

r2(config-router)#version 2

r2(config-router)#network 12.0.0.0

r2(config-router)#network 23.0.0.0

r2(config-router)#network 172.16.0.0

r2(config-router)#network 192.168.1.0

r2(config-router)#network 192.168.2.0

r2(config-router)#network 192.168.3.0

说明：在 R2 上配置网段，并启动 RIP ver 2，将网段放入 RIP 进程。

（3）配置 R3:

r3(config)#int f0/1

r3(config-if)#ip address 23.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#router rip

r3(config-router)#network 23.0.0.0

说明：在 R3 上配置 23.1.1.0/24，并将网段放入 RIP 进程。

2.测试 RIP 手工汇总

（1）查看当前 R3 的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

路

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

R 172.16.0.0/16 [120/1] via 23.1.1.2, 00:00:16, FastEthernet0/1

R 12.0.0.0/8 [120/1] via 23.1.1.2, 00:00:16, FastEthernet0/1

R 192.168.1.0/24 [120/1] via 23.1.1.2, 00:00:16, FastEthernet0/1

R 192.168.2.0/24 [120/1] via 23.1.1.2, 00:00:16, FastEthernet0/1

R 192.168.3.0/24 [120/1] via 23.1.1.2, 00:00:16, FastEthernet0/1

r3#

说明：虽然 R3 为 RIP ver 2，但自动汇总默认开启，所以 R3 收到的还是汇总过的路由。

（2）在 R2 上将 172.16.0.0 中 24 位的路由汇总为 22 位发给 R3:

r2(config)#int f0/1

r2(config-if)#ip summary-address rip 172.16.0.0 255.255.252.0

说明：手工汇总为 172.16.0.0/22。

（3）查看 R2 的手工汇总情况：

r2#debug ip rip

RIP protocol debugging is on

r2#

*Mar 1 01:05:34.903: RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (23.1.1.2)

*Mar 1 01:05:34.903: RIP: build update entries

*Mar 1 01:05:34.903: 12.0.0.0/8 via 0.0.0.0, metric 1, tag 0

*Mar 1 01:05:34.907: 172.16.0.0/16 via 0.0.0.0, metric 1, tag 0

*Mar 1 01:05:34.907: 192.168.1.0/24 via 0.0.0.0, metric 1, tag 0

*Mar 1 01:05:34.907: 192.168.2.0/24 via 0.0.0.0, metric 1, tag 0

*Mar 1 01:05:34.911: 192.168.3.0/24 via 0.0.0.0, metric 1, tag 0

说明：虽然 R2 在 F0/1 上配置了手工汇总 172.16.0.0/22，但由于自动汇总还在生效，所以自动汇总优先，最后发送的路由依然是 172.16.0.0/16 这条主类网络。

查看 R3 的路由表：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     23.0.0.0/24 is subnetted, 1 subnets
C       23.1.1.0 is directly connected, FastEthernet0/1
R    172.16.0.0/16 [120/1] via 23.1.1.2, 00:00:08, FastEthernet0/1
R    12.0.0.0/8 [120/1] via 23.1.1.2, 00:00:08, FastEthernet0/1
R    192.168.1.0/24 [120/1] via 23.1.1.2, 00:00:08, FastEthernet0/1
R    192.168.2.0/24 [120/1] via 23.1.1.2, 00:00:08, FastEthernet0/1
R    192.168.3.0/24 [120/1] via 23.1.1.2, 00:00:08, FastEthernet0/1
r3#
```

说明：R3 并没有收到 R2 上配置的手工汇总 172.16.0.0/22，说明自动汇总优先于手工汇总。

（4）在 R2 上关闭自动汇总：

r2(config)#router rip

r2(config-router)#no auto-summary

（5）再次查看 R3 上的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

172.16.0.0/22 is subnetted, 1 subnets

R 172.16.0.0 [120/1] via 23.1.1.2, 00:00:05, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/1] via 23.1.1.2, 00:00:05, FastEthernet0/1

R 192.168.1.0/24 [120/1] via 23.1.1.2, 00:00:05, FastEthernet0/1

R 192.168.2.0/24 [120/1] via 23.1.1.2, 00:00:05, FastEthernet0/1

R 192.168.3.0/24 [120/1] via 23.1.1.2, 00:00:05, FastEthernet0/1

r3#

说明： R2 上关闭了自动汇总：所以 R3 收到了 R2 的手工汇总 172.16.0.0/22。

（6）查看 R2 的路由表：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

172.16.0.0/24 is subnetted, 3 subnets

C 172.16.1.0 is directly connected, Loopback172

C 172.16.2.0 is directly connected, Loopback172

C 172.16.3.0 is directly connected, Loopback172

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

C 192.168.1.0/24 is directly connected, Loopback192

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

r2#

说明：R2 上在 172.16.0.0/22 只存在 172.16.1.0，172.16.2.0，172.16.3.0，共三条。

（7）解决 R2 的路由黑洞:

说明：由于 172.16.0.0/22 包含了 172.16.0.0/24，172.16.1.0/24，172.16.2.0/24，172.16.3.0/24，所以最终 R3 会将去往 172.16.0.0/24 的数据包发给 R2，这时 R2 收到数据包会将自己的路由表全部查一遍，最终发现目标不可达，才将数据包丢弃，这是浪费时间也浪费系统资源的事情，所以应该手工将不存在的路由指向空接口（null 0），所有指向该接口的数据包将被全部丢弃。

r2(config)#ip route 172.16.0.0 255.255.252.0 null 0

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks

S 172.16.0.0/22 is directly connected, Null0

C 172.16.1.0/24 is directly connected, Loopback172

C 172.16.2.0/24 is directly connected, Loopback172

C 172.16.3.0/24 is directly connected, Loopback172

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

C 192.168.1.0/24 is directly connected, Loopback192

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

r2#

说明：R2 将 172.16.0.0/22 指向了空接口（null 0），但这样并不会有错，因为 R2 拥有 172.16.1.0/24，172.16.2.0/24，172.16.3.0/24 这些明细路由，照样会正常转发数据，只有 172.16.0.0/22 中未知的流量才会被丢弃。

（8）在 R2 上手工汇总 192 网段：

r2(config)#int f0/1

r2(config-if)#ip summary-address rip 192.168.0.0 255.255.252.0

Summary mask must be greater or equal to major net

r2(config-if)#

说明：提示不能汇总 192.168.0.0/22，因为手工汇总不能将一条路由的掩码位数汇总到短于自身主类网络的掩码长度，而 192.168.0.0 的主类网络是 192.168.0.0/24 这个 C 类地址，所以不能将掩码汇总到低于 24 位。

（9）测试 R3 自己手工汇总:

说明：在 R2 上去除手工汇总，看 R3 自己手工汇总：

r2(config)#int f0/1

r2(config-if)#no ip summary-address rip 172.16.0.0 255.255.252.0

r3(config)#int f0/1

r3(config-if)#ip summary-address rip 172.16.0.0 255.255.252.0

说明：R3 自己将汇总 172.16.0.0/22。

（10）查看 R3 的汇总情况：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     23.0.0.0/24 is subnetted, 1 subnets
C       23.1.1.0 is directly connected, FastEthernet0/1
     172.16.0.0/24 is subnetted, 3 subnets
R       172.16.1.0 [120/1] via 23.1.1.2, 00:00:26, FastEthernet0/1
R       172.16.2.0 [120/1] via 23.1.1.2, 00:00:26, FastEthernet0/1
R       172.16.3.0 [120/1] via 23.1.1.2, 00:00:26, FastEthernet0/1
     12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 23.1.1.2, 00:00:26, FastEthernet0/1
R    192.168.1.0/24 [120/1] via 23.1.1.2, 00:00:26, FastEthernet0/1
R    192.168.2.0/24 [120/1] via 23.1.1.2, 00:00:26, FastEthernet0/1
R    192.168.3.0/24 [120/1] via 23.1.1.2, 00:00:26, FastEthernet0/1
```

r3#

说明：R3 汇总并不成功，因为汇总是将路由汇总后发给其它路由器，而自己并不能对自己的路由表汇总。

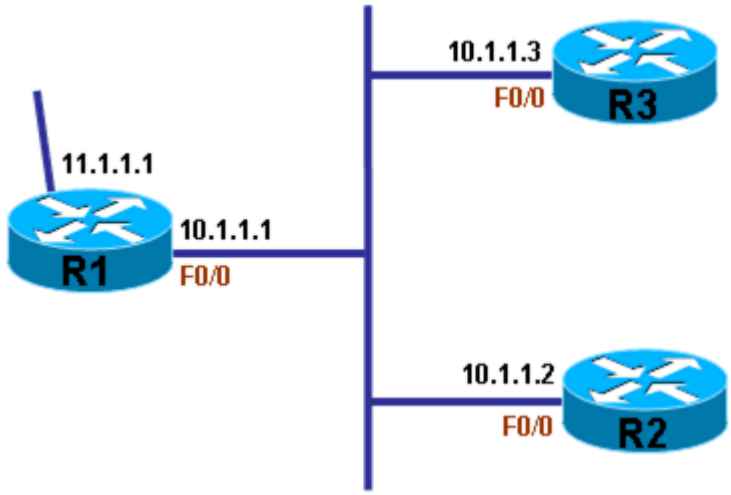
RIP ver 2 认证

由于 RIP 没有邻居的概念，所以自己并不知道发出去的路由更新是不是有路由器收到，同样也不知道会被什么样的路由器收到，因为 RIP 的路由更新是明文的，网络中无论谁收到，都可以读取里面的信息，这就难免会有不怀好意者窃听 RIP 的路由信息。为了防止路由信息被非法窃取，RIP ver 2 可以相互认证，只有能够提供密码的路由器，才能够获得路由更新。而 RIP ver 1 是不支持认证的。RIP ver 2 可以支持明文与 MD5 认证。

路由器之间，当一方开启认证之后，另一方也同样需要开启认证，并且密码一致，才能读取路由信息。认证是基于接口配置的，密码使用 key chain 来定义，key chain 中可以定义多个密码，每个密码都有一个序号，RIP ver 2 在认证时，只要双方最前面的一组密码相同，认证即可通过，双方密码序号不一定需要相同，key chain 名字也不需要相同，但在某些低版本 IOS 中，会要求双方的密码序号必须相同，才能认证成功，所以建议大家配置认证时，双方都配置相同的序号和密码。

测试 RIP ver 2 认证

说明：以下图为例，测试 RIP ver 2 认证



1.配置基础网络环境

（1）配置 R1:

r1(config)#int f0/0

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#int loopback 11

r1(config-if)#ip address 11.1.1.1 255.255.255.0

r1(config)#router rip

r1(config-router)#version 2

r1(config-router)#no auto-summary

r1(config-router)#network 10.0.0.0

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

r1(config-router)#network 11.0.0.0

说明：R1 配置了 10.1.1.0/24 和 11.1.1.0/24，并放入 RIP ver 2 进程中。

（2）配置 R2:

r2(config)#int f0/0

r2(config-if)#ip add 10.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#router rip

r2(config-router)#version 2

r2(config-router)#no auto-summary

r2(config-router)#network 10.0.0.0

说明：R2 配置了 10.1.1.0/24，并放入 RIP ver 2 进程中。

（3）配置 R3:

r3(config)#int f0/0

r3(config-if)#ip address 10.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#router rip

r3(config-router)#version 2

r3(config-router)#no auto-summary

r3(config-router)#network 10.0.0.0

说明：R3 配置了 10.1.1.0/24，并放入 RIP ver 2 进程中。

2.测试 RIP ver2 认证

（1）查看当前 R2 与 R3 的路由表：

R2：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

11.0.0.0/24 is subnetted, 1 subnets

R 11.1.1.0 [120/1] via 10.1.1.1, 00:00:10, FastEthernet0/0

r2#

R3：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

11.0.0.0/24 is subnetted, 1 subnets

R 11.1.1.0 [120/1] via 10.1.1.1, 00:00:26, FastEthernet0/0

r3#

说明：因为没有开启认证，所以 R2 与 R3 能够正常收到路由 11.1.1.0/24。

（2）在 R1 上开启认证：

r1(config)#key chain ccie

r1(config-keychain)#key 1

r1(config-keychain-key)#key-string cisco

r1(config)#int f0/0

r1(config-if)#ip rip authentication mode md5

r1(config-if)#ip rip authentication key-chain ccie

说明：配置 key chain 为 ccie，key 1 的密码为 cisco，并在接口上开启认证，注意一条为定义认证模式，一条为定义密码。

（3）查看 R2 与 R3 没有认证时的路由表：

R2：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

11.0.0.0/24 is subnetted, 1 subnets

R 11.1.1.0 [120/1] via 10.1.1.1, 00:03:04, FastEthernet0/0

r2#

R3：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

11.0.0.0/24 is subnetted, 1 subnets

R 11.1.1.0/24 is possibly down,

routing via 10.1.1.1, FastEthernet0/0

r3#

说明：因为 R2 和 R3 没有开启认证，所以无法获得路由更新，之前的路由已经快要丢失。

（4）在 R2 上开启认证:

r2(config)#key chain ccie

r2(config-keychain)#key 2

r2(config-keychain-key)#key-string cisco

r2(config)#int f0/0

r2(config-if)#ip rip authentication mode md5

r2(config-if)#ip rip authentication key-chain ccie

说明：R2 配置 key chain 为 ccie，而 key 2 为 cisco，可见双方密码的序号不同。

（5）查看 R2 开启认证后的路由表：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

11.0.0.0/24 is subnetted, 1 subnets

R 11.1.1.0 [120/1] via 10.1.1.1, 00:00:03, FastEthernet0/0

r2#

说明：R2 在开启认证后，成功获得 R1 发来的路由 11.1.1.0，虽然双方密码的序号不同，但双方最前面的密码相同，所以认证成功，请注意，会存在某些 IOS 需要双方密码序号相同。

（6）在 R3 上开启认证:

r3(config)#key chain ccie

r3(config-keychain)#key 1

r3(config-keychain-key)#ke

r3(config-keychain-key)#key-string abc

r3(config-keychain-key)#exit

r3(config-keychain)#key 3

r3(config-keychain-key)#key-string cisco

r3(config-keychain-key)#exit

r3(config-keychain)#key 4

r3(config-keychain-key)#key-string cde

r3(config)#int f0/0

r3(config-if)#ip rip authentication mode md5

r3(config-if)#ip rip authentication key-chain ccie

说明：R3 的 key chain 中定义 key 1 密码为 abc ，key 3 密码为 cisco，key 4 为 cde，R3 最前面的密码应该是 key 1，为 abc，与 R1 最前面的密码 cisco 不同。

（7）再次查看 R3 的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

r3#

说明：由于 R3 最前面的密码 key 1，为 abc，与 R1 最前面的密码 cisco 不同，所以认证未能通过。

（8）更改 R3 的 key chain:

r3(config)#key chain ccie

r3(config-keychain)#no key 1

说明：因为 R3 之前配置了 key 1 密码为 abc ，key 3 密码为 cisco，key 4 为 cde，而现在取消了 key 1，所以当前最前面的密码为 key 3 “cisco”会被使用。

（9）查看 R3 修改认证后的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

11.0.0.0/24 is subnetted, 1 subnets

R 11.1.1.0 [120/1] via 10.1.1.1, 00:00:04, FastEthernet0/0

r3#

说明：因为 R1 最前面的密码 key 1 为 cisco，R3 最前面的密码 key 3 为 cisco，双方最前面的密码相同，所以认证成功，R3 能够获得 R1 发来的路由。

EIGRP

概述

EIGRP 是思科私有协议，只能运行在思科的设备上。

EIGRP 能够支持的协议有 IP、AppleTalk 和 IPX。

EIGRP 的流量使用 IP 协议号 88。

EIGRP 采用 Diffused Update Algorithm (DUAL)算法来计算到目标网络的最短路径，EIGRP 还是一个距离矢量路由协议，因为距离矢量路由协议的根本特征就是自己的路由表是完全从其它路由器学来的，并且将收到的路由条目一丝不变地放进自己的路由表，运行距离矢量路由协议的路由器之间交换的是路由表，距离矢量路由协议是没有大脑的，路由表从来不会自己计算，总是把别人的路由表拿来就用；

但 EIGRP 协议并不是完全没有大脑，它在距离矢量路由协议的基础上却有着那么一点点的优化和提高，那就是从邻居那里收到路由表之后，会经过一些计算和比较，然后才放进路由表中使用，因此将 EIGRP 的身份提高到“增强的距离矢量路由协议”，虽然 EIGRP 是增强的距离矢量路由协议，但其根本核心还是交换路由表，EIGRP 没有任何资格称为链路状态路由协议，因为 EIGRP 从邻居那里得到的路由表可能原本就是错的，那么自己将会一错再错，并且害人害己，EIGRP 对于网络拓朴也许会有完整的认识，但不一定有正确的认识。因为 EIGRP 是距离矢量路由协议，所以 EIGRP 同样会受水平分割（Split Horizon）的影响。

EIGRP 使用了 Autonomous System（AS）的概念，即使是这样，EIGRP 也算不上外部网关路由协议（Exterior Gateway Protocol 即 EGP），因为不同 AS 之间，EIGRP 无法传递路由信息，所以 EIGRP 依然是个内部网关路由协议（Interior Gateway Protocol，即 IGP）。AS 是基于接口定义的，一台 EIGRP 路由器可以属于多个 AS。

EIGRP 扩展了对大型网络的支持，不再像 RIP 那样只支持最大跳数 15 跳，而是扩展到了最大支持 255 跳，但默认情况下最大跳数为 100 跳。

EIGRP 支持 Classless Interdomain Routing (CIDR) 和 Variable-Length Subnet Masks (VLSMs)，但默认也会自动汇总，该功能可以手工关闭， EIGRP 还支持手工汇总路由信息，并且手工汇总没有任何条件限制，可以汇总到任意掩码长度。自动汇总和手工汇总与 RIP 相似，汇总是针对发出的路由有效，也就是对其它路由器生效，是对其它路由器的路由表生效。

EIGRP 支持认证，并且只支持 MD5 认证；支持通过 Offset list 来增加路由的 metric，只可以增加，不可以减少；EIGRP 也支持 Passive-Interface（被动接口），但 EIGRP 的被动接口与 RIP 不同， RIP 的被动接口不向外发路由，但可以接收路由，而 EIGRP 的被动接口不接收也不发送路由。

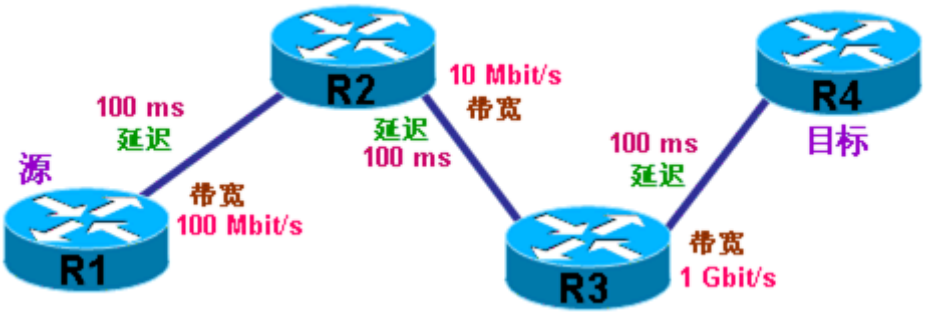
EIGRP 并不会周期性更新路由表，而采用增量更新，即只在路由有变化时，才会发送更新，并且只发送有变化的路由信息；有时 EIGRP 并不知道邻居的路径是否还依然有效，并且路由没有超时。

EIGRP 自己内部路由的管理距离(Ddministrative Distance)为 90，而从外部重分布进 EIGRP 的管理距离为 170。

EIGRP 支持非等价负载均衡，最多支持 6 条，默认为 4 条，但非等价负载均衡功能默认为关闭状态。

EIGRP Metric

EIGRP 使用多种参数计算去往目标网络的 Metric 值，包括带宽(Bandwidth)、延迟(delay)、可靠性(reliability)、负载(loading)、最大传输单元(MTU)，这 5 个参数分别使用 K 值来表示，即 K1、K2、K3、K4、K5，所以如果两台 EIGRP 路由器之间的 5 个 K 值不同，则代表双方计算 Metric 值的方法不同；无论是 EIGRP 还是其它协议，在需要使用带宽计算 Metric 时，只计算接口出方向的带宽，而接口进方向的是不计算在内的，也就是一条链路上，只有一个出接口的带宽会被计算，而进接口的带宽是被忽略的，如下图中所示，



从源路由器 R1 到目标 R4 所经过的三条链路中，带宽分别是 100 Mbit/s、10 Mbit/s、1 Gbit/s、虽然有条链路的带宽可达 1 Gbit/s，但我们都知道，从源到目标的带宽最快也始终不超过最低带宽 10 Mbit/s，永远不可能达到 1 Gbit/s，所以从源到目的的路径中，只有最低带宽最终决定传输时的带宽，而某条链路的高带宽是没有意义的，所以在 EIGRP 的 Metric 计算中，只需要计算从源到目标的最低带宽即可，但是经过各个接口的延迟却是要累加的。

EIGRP 在计算 Metric 时，使用 K 值来控制整个计算公式，公式如下：

$$\left(K1 \times Bandwidth + \frac{K2 \times Bandwidth}{256 - Load} + K3 \times Delay\right) \times \frac{K5}{Reliability + K4} \times 256$$

公式中的带宽为 1000 0000 除以链路中的最小带宽，带宽单位为 Kbit，延迟为链路中的延迟之和除以 10，延迟单位为 ms(毫秒)；

默认情况下，5 个 K 值的取值分别为: K1 = 1, K2 = 0, K3 = 1, K4 = K5 =0，由于 K5=0，由此一来，造成上面的公式算出来的最终结果为 0，所以当 K5=0 时，必须将其简化，变成如下：

$$\left(K1 \times Bandwidth + \frac{K2 \times Bandwidth}{256 - Load} + K3 \times Delay\right) \times 256$$

正因为默认 5 个 K 值的取值分别为: K1 = 1, K2 = 0, K3 = 1, K4 = K5 =0，所以默认 EIGRP 的计算公式为：

$$\left(\frac{1000\ 0000}{\text{最小 Bandwidth}} + \frac{\text{Delay之和}}{10}\right) \times 256$$

注：此公式为 EIGRP 默认计算公式，改变 K 值，将影响 Metric 值的计算公式。

EIGRP 邻居

EIGRP 使用了邻居的概念，EIGRP 的路由表并不会像 RIP 那样通过组播或广播向网络中发送，EIGRP 只向邻居发送路由表，并且是使用单播向邻居发送路由表，如果要在 EIGRP 之间交换路由表，必须成为邻居，不同 AS 不能成为邻居，EIGRP 只在直连网络中发现和建立邻居。

EIGRP 路由器之间的邻居关系通过 Hello 包来发现和维护，EIGRP 会将自己全部的路由表发给所有邻居；路由器上启动 EIGRP 之后，就会使用组播地址 224.0.0.10 在相应接口上发送 Hello 包，EIGRP 会使用一张单独的表来记录哪些路由器是自己的邻居，称为邻居表，只要收到 Hello 包，便将对方列为自己的邻居，并且写入邻居表，EIGRP 会将邻居的地址写在 Hello 包中，由此可见，EIGRP 路由器双方可能一方认为另外一方是自己的邻居，而另外一方却不认为对方是邻居，例如自己收到了另一方的 Hello 包，认为对方是邻居，而对方没有收到或过滤了自己的 Hello 包，所以如果 EIGRP 要形成双向邻居，只有在双方都发现对方的 Hello 包中列出自己的地址才行，但思科却没有这样设计。

在 EIGRP 断开或进程关闭时，会发送 Goodbye Message 结束邻居关系。

EIGRP 会定期向网络中发送 Hello 包，发送的间隔会因为链路带宽的不同而不同，间隔时间分为 5 秒和 60 秒：

Hello 间隔 60 秒

所有带宽低于或等于 1544 Kbit/s 的，如 T1，Frame Relay multipoint 接口，ATM multipoint 接口，ISDN BRI 接口等等。

Hello 间隔 5 秒

所有带宽大于 1544 Kbit/s 的，如以太网接口，Frame Relay point-to-point 子接口，and ATM point-to-point 子接口，ISDN PRI 接口。

如果超过一定的时间没有收到邻居的 Hello 包，便认为邻居无效，称为 EIGRP Hold-time，默认为 Hello 间隔的 3 倍，也就是分别为 15 秒和 180 秒，Hello 间隔时间和 Hold-time 都可以手工调整，但是如果调整了 Hello 间隔时间，Hold-time 并不会自动调整到相应的 3 倍，而是保持不变。

EIGRP 的 Hello 间隔时间和 Hold-time 是无法通过命令直接查看的，只能通过现象来推断，如：

```
router# show ip eigrp neighbor

IP-EIGRP neighbors for process 1

H   Address         Interface   Hold Uptime   SRTT   RTO  Q  Seq  Type

                                (sec)         (ms)       Cnt Num

1   10.1.1.2        Et1          11 12:00:56   12    300  0  620
```

0 10.1.2.2 S0 172 12:00:58 17 200 0 645

Hold 栏的值永远不会超过 Hold-time，也永远不会低于 Hold-time 减 Hello 间隔，否则就是不正常，或者 Hello 包有丢失。

如果 Hold 栏的值范围是 10-15，则证明 Hello 间隔是 5 秒，Hold-time 是 15 秒。

如果 Hold 栏的值范围是 120-180，则证明 Hello 间隔是 60 秒，Hold-time 是 180 秒。

注：

★Hello 间隔和 Hold-time 可以手工在接口上配置，如果发现不正常，请检查接口上是否手工配置了时间参数。

★EIGRP 双方 Hello 间隔和 Hold-time 不一致也可以建立邻居关系，接口上的 Secondary 地址不能建邻居，所有 EIGRP 的数据包源地址总是接口的 Primary 地址。

★在 Frame-Relay 环境下，需要在命令 frame-relay map 带 broadcast 关键字。

EIGRP 双方必须满足以下三个条件，才能建立邻居：

★1.双方在相同 AS

★2.双方 Hello 包正常，即双方接口 Primary 地址在同网段。

★3.双方计算 Metric 值方法相同，即双方 K1 K2 K3 K4 K5 值相同。

在一个稳定的 EIGRP 邻居之间，只有 Hello 包在传递，EIGRP 支持的邻居数量，

并没有限制，但要视内存大小，CPU 能力，路由条目数量，拓朴复杂程度，网络稳定性而定。

EIGRP 数据包

在 EIGRP 协议中，总共会使用 5 种类型的数据包，分别为 Hello、Update、Query、Reply、Ack，下面介绍各种数据包的功能与用途：

Hello

是用来发现和维护 EIGRP 邻居关系的，目标地址为 224.0.0.10，Hello 包在邻居收到后不需要确认。

Update

发给邻居的路由表，通过单播发送 Update 数据包，邻居收到后必须回复确认消息。

Query

当路由信息丢失并没有备用路由时，使用 Query 数据包向邻居查询，邻居必须回复确认。

Reply

是对邻居 Query 数据包的回复，也需要邻居回复确认。

Ack

是对收到的数据包的确认，告诉邻居自己已经收到数据包了，收到 Ack 后，不需要再对 Ack 做回复，因为这是没有意义的，并且可能造成死循环。

由以上可以看出，5 种数据包中，Update、Query、Reply 在对方收到后，都需要回复确认，这些数据包是可靠的，回复是发送 Ack；而 Hello 和 Ack，是不需要回复的，因此被认为不可靠。

EIGRP 运行过程：

两台路由器 A 和 B 启动 EIGRP

1.路由器 A 启动 EIGRP 后，在相应接口上向外发送 Hello 包。

2.B 收到 A 发来的 Hello 包后，将 A 列入自己的邻居表，然后向 A 单播发送 Update，也就是向 A 单播发送自己的路由表；EIGRP 路由器在收到一个 Hello 包时，就直接向对方发送路由表了，B 后面的 Hello 包就会写明路由器 A 已是自己的邻居。

3.A 在收到 B 的 Update 之后，向 B 发送 Ack 通知对方已经收到 Update，然后再向 B 单播发送自己的 Update，在 A 收到 B 的 Hello 包后，会发现里面列出了自己的地址，双方都看见对方 Hello 中列出自己后，双向邻居关系便建立成功。

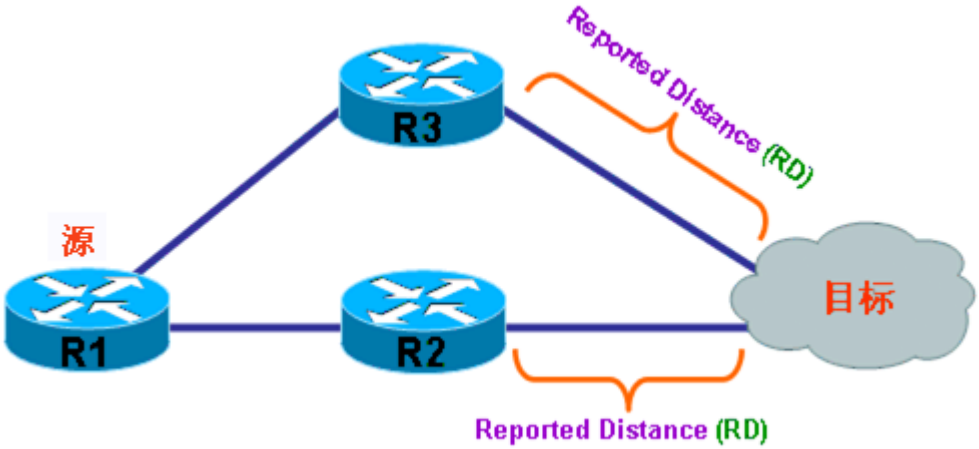
4.B 在收到 A 的 Update 之后，向 A 发送 Ack 确认 Update 已收到。

5.A 和 B 都将收到的 Update 放入拓朴数据库中，计算路由表。

EIGRP 拓朴

虽然 EIGRP 是距离矢量路由协议，但是当从邻居收到路由后，并不是不经过任何计算就直接放进路由表中使用，EIGRP 会将从邻居收到的路由全部放入拓朴数据库（Topology Database）中，经过 DUAL 的无环算法计算之后，才将最优的路由放入路由表中；因为 EIGRP 可能有多个邻居，也可能从多个邻居收到相同的路由，所以需要从中选中最优路由放入路由表中使用，而不是最优的路由则放在拓朴数据库中备份，等路由表中的路由失效后，便从拓朴数据库中查找备用路由继续放入路由表中使用。

当 EIGRP 将从邻居收到的路由信息放入拓朴数据库之后，要经过 DUAL 算法选出最优路由，以下图为例介绍计算方法：



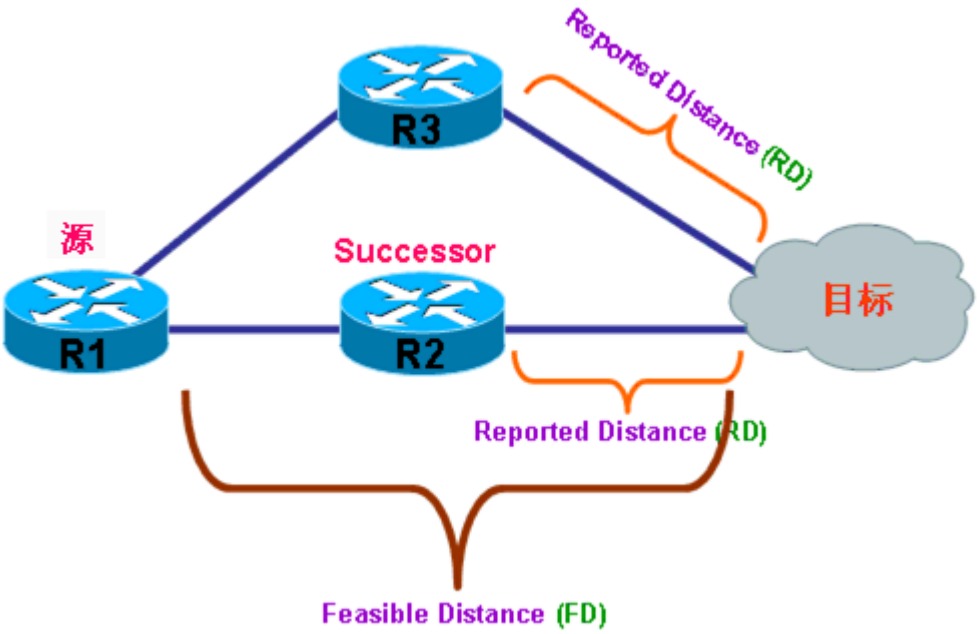
要了解 EIGRP 选路的具体计算过程，需要先了解以下几个术语：

Reported Distance（RD）

因为 EIGRP 拓朴数据库中的信息就是从邻居收到的路由表，目的地对于邻居来说肯定是可达的，Reported Distance 是表示邻居到达目的地的 Metric 值是多少。在上图中，R1 从 R2 和 R3 收到去往目标网络的路由后，R2 去往目标网络的 Metric 值对于 R1 来说就是 RD，同样，R3 去往目标网络的 Metric 值对于 R1 来说也是 RD。

Feasibledistance（FD）

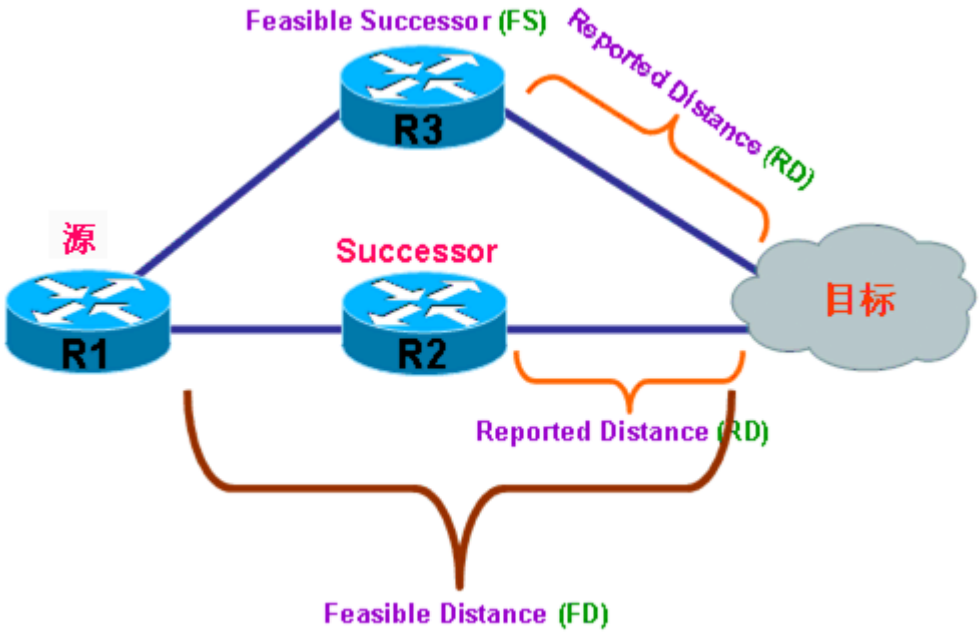
当从邻居收到路由信息后，RD 只是邻居去往目的地的 Metric，而自己去往目的地还得在 RD 的基础上，再加自己到邻居的这段距离，所以自己到目的地的真正 Metric 应该是自己到邻居这段距离的 Metric 加上 RD，但是拓朴数据库中可能存在多条去往目的地的路径，而被放入路由表的最优的那条被称为 Feasible distance（FD）。在上图中，如果 R1 选择从 R2 去往目的地，那么结果将如下图：



如果 R1 选择从 R2 去往目的地，那么 R2 到达目的地的 metric（RD）加上 R1 到 R2 的 Metric，就是 R1 到达目的地的 FD，R1 将 R2 的路径放入路由表中，这时，R2 也称为 Successor；默认情况下，拓朴数据库中有多条路径可到目的地时，被放入路由表的那条就是 FD，FD 就一定会被放入路由表。

FeasibleSuccessor（FS）

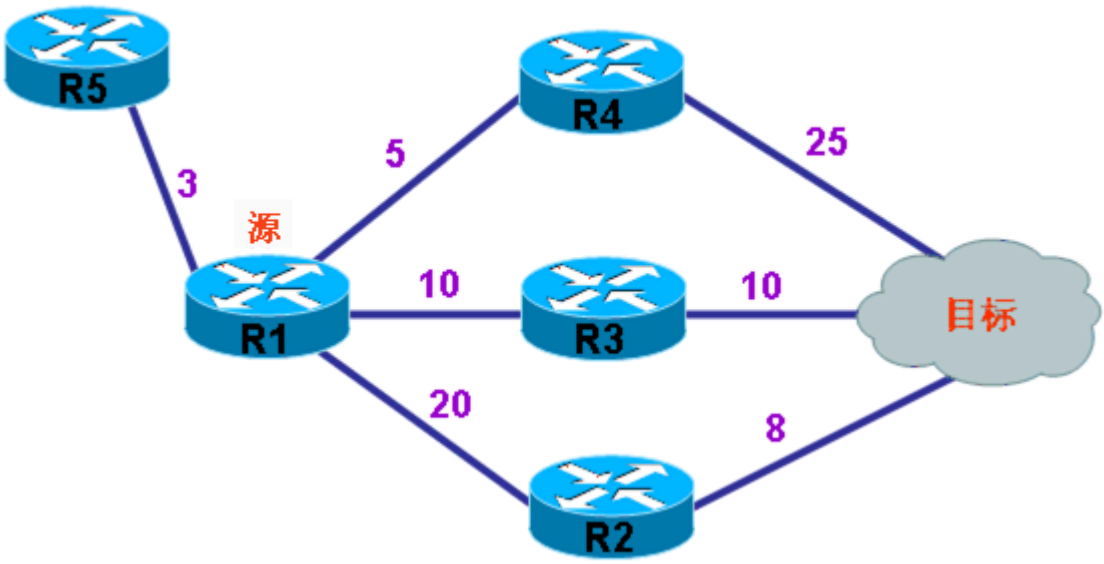
因为拓朴数据库中可能存在多条路径可以到达目的地，但被选为 FD 的最优的那条被放入路由表中使用，而留在拓朴数据库中的备用路由称为 Feasible Successor（FS），如下图：



R2 被选为 Successor，当 R3 继续存放于拓朴数据库时，R3 就是 FS。

Feasibility Condition (FC)

拓朴数据库中的 FS 最多可以有 6 条，如果一个 EIGRP 有 8 个邻居可以去往目的地，选出一条 FD 放入路由表之后，并不是其它 7 条全部都可以存放于拓朴数据库，拓朴数据库最多只能有 6 条（其中已经包含 FD），也并不是一定会有 6 条被放入拓朴数据库，因为要存放于拓朴数据库，是必须达到一定条件的，称为 Feasibility Condition (FC)，就是邻居通告的 RD 必须小于 FD，这个邻居的路径才能存在于拓朴数据库，计算 FC 的过程如下：



上图中，R1 共有 4 个邻居，分别为 R2、R3、R4、R5，当 R1 从每个邻居那里收到去往目标网络的路由后，结果如下：

R2 通告的 RD 为 8，

R3 通告的 RD 为 10，

R4 通告的 RD 为 25，

当 R1 要去目标网络时，必须加上自己到达邻居的 Metric，计算结果为：

从 R2 到达目的地的 Metric 为 20 + 8 = 28，

从 R3 到达目的地的 Metric 为 10 +10 = 20，

从 R4 到达目的地的 Metric 为 5 + 25 = 30，

由此可见，R1 从 R3 去往目的地的 Metric 值最小，因此 R3 的路径被选为 FD 并放入路由表，R3 这时就是 Successor，R1 去往目的地的 FD 为 20，其它邻居的

路由要成为 FS 存放在拓朴数据库中，那么它们的 RD 必须小于 FD，也就是必须小于 20，等于也不行；下面继续计算 FS，因为 R2 的 RD 为 10，小于 FD 20，所以 R2 就是 FS 并存放在拓朴数据库中备份，而 R4 的 RD 为 28，大于 FD，所以 R4 没有资格成为 FS，不能放在拓朴数据库中，这是因为 EIGRP 要防止环路，虽然 R4 的路径并不是环路，但由于要防患于未然，所以 R4 被拒绝于拓朴数据库之外，理由如下：

因为 R1 也是 R5 的邻居，所以 R1 会将自己的路由表全部发给 R5，由于 R1 到达目的地的 Metric 为 20，R5 到达 R1 的 Metric 值为 3，所以最后 R5 到达目标网络的 Metric 值为 20 + 3 = 23，R5 在将自己的路由表通告给 R1 后，很明显，R5 的 Metric 值 23 就是 R1 从 R5 到达目的地的 RD，对于 R1 来说，因为 R5 要从自己去往目标网络，所以对方通告的 RD 才会比自己的 FD 还要大，以致于让 EIGRP 认为，所有邻居通告的 RD 比自己当前的 FD 大的话，都统一认为它们在自己后面，它们必须经过自己才能去往目的地，那么这就是一个环路，最终邻居通告的 RD 等于或大于 FD 的，都没有资格成为 FS， 所以最后造成上图中 R4 通告了 RD 为 25 而没有资格成为 FS。

EIGRP 负载均衡

EIGRP 可以支持非等价负载均衡，最多支持 6 条，默认为 4 条，但非等价负载均衡功能默认为关闭状态。EIGRP 只能将拓朴数据库中的备用链路放入路由表执行负载均衡，拓朴数据库中可能有多条备用链路，而且多条链路的 Metric 值也可能各不相同，当启用非等价负载均衡时，需要定义什么样的 Metric 范围可以用来负载均衡，这需要通过控制 Metric 的变量（Variance）值来控制，具体方法如下：

路由表中正在使用的最优路由的 Metric 值为 FD，而拓朴数据库中备用路由的 Metric 值肯定是大于 FD 的，Variance 值通过控制备用链路的 Metric 值与 FD 的倍数关系来控制，就是 Variance 值取多少，备用链路的 Metric 在 FD 的 Variance 值倍数范围内就有资格执行负载均衡,例如当前 FD 为 20，3 条备用链路 Metric 分别为 30，50，55，如果 Variance 值取 2，那么 Metric 值范围在 20×2=40 的链路都可以执行负载均衡，所以 Metric 值为 30 的链路可以执行负载均衡，而 Metric 值为 50 和 55 的却不可以，因为大于 40，只有当 Variance 值取 3 时，Metric 值范围在 20×3=60 的链路才可以执行负载均衡，所以 Metric 值为 50 和 55 只有在 Variance

值取 3 时才可以执行负载均衡。

注： Variance 值默认取值为 1，也就是不执行非等价负载均衡，但会执行等价负载均衡。

并不是所有在 Variance 值所定义的 Metric 值范围内的链路一定会执行负载均衡，这需要根据设置的最大负载均衡条数来决定，最多为 6 条。

因为 Metric 值越大的路由，表示其链路况越低下，而 Metric 值越小的路由，其链路况越优秀，这是一个成反比的关系，所以在执行负载均衡时，我们更希望流量也能因 Metric 值的大小，成反比例传输，链路好的传递更多的数据包，而链路差的则传递更少的数据包，

通过配置命令 traffic-share balanced 即可，该功能默认为开启状态。

EIGRP Stuck In Active (SIA)

在 EIGRP 中，正常的路由称为 Passive Route；因为 EIGRP 可能会从多个邻居处收到相同的路由，默认只有最优的路由会被放入路由表中使用，其它符合 FC 条件的会放入拓朴数据库中备份，当路由表中最优路由丢失时，EIGRP 会从拓朴数据库中查询备用路由，如果当最优路由丢失后，拓朴数据库中又没有备用路由，在这种情况下，EIGRP 会向所有邻居发送 Query，试图查询邻居是否有到目的地的路由信息，并且发送 Query 后，该路由被标记为 Active route ，该状态称为 Stuck In Active (SIA)；向邻居发送的 Query 是必须回复 Ack 确认的，当邻居收到 Query 之后，就会查询自己的路由表，如果有，就向邻居回复 Reply，如果最终邻居的路由表和拓朴表中都没有相应路由条目，就会再向自己的所有邻居发出 Query，虽然 Query 是向所有邻居发出的，但它不会发向原本最优路径的下一跳，也就是 Query 不会发向该路由的 Successor，因为正是 Successor 路丢失了，才问自己要路由的，自己再反过去问对方要路由，是没有理由的。不可思议的事情是，如果发送 Query 的路由器在 3 分钟内没有收到邻居的回复，就会清除与该邻居的 EIGRP 会话，可想而知，EIGRP 路由丢失后，必须要求邻居提供路由，如果不提供，它就会六亲不认，3 分钟就和你翻脸，反目成仇，和你断绝任何关系，很费解为什么思科会将 EIGRP 设计成这样子，EIGRP 的优势分析来分析去，只有一个非等价负载均衡，和所谓的快速收敛，再无别的，至少本人是这么认为的。

造成 EIGRP 路由器无法回复邻居 Query 的原因有很多，如：CPU 繁忙，内存错误，数据包丢失，或者是单向链路故障等等。

为了杜绝 EIGRP 在 SIA 状态时，由于邻居没有提供路由而误将与邻居的会话清除，思科推荐解决方法为调整 SIA 状态时等待回复的时间，默认为 3 分钟，其实这种改时间，不是解决总是的根本，这只是在拖延时间而已，会话总是要断开的，不知道这个是否会被高手利用而成为攻击 EIGRP 的漏洞。另外的解决方法就是将相应路由器配置为 EIGRP 末节（Stub）路由器，EIGRP 在 SIA 状态是不会向 Stub 邻居发送 Query 的，但 Stub 邻居向外发送路由的功能有所限制，默认只能发送直连和汇总路由，但可以调整，也可以调整为不向外发送任何路由。

配置 EIGRP 实验

说明：实验配置共包含：

EIGRP 基础实验

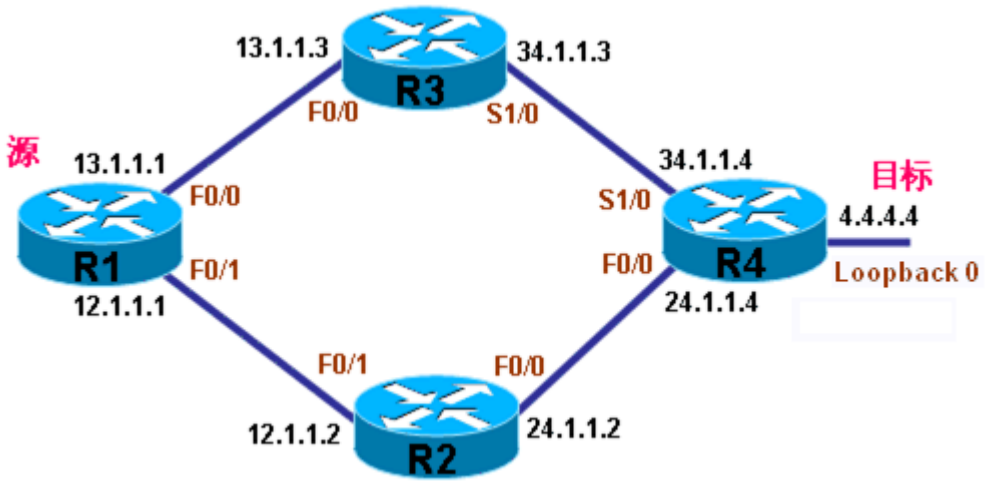
EIGRP 非等价负载均衡

EIGRP Stub

EIGRP 手工汇总

EIGRP 认证

EIGRP 默认路由



1.配置网络基础环境

（1）配置 R1:

r1(config)#int f0/0

r1(config-if)#ip address 13.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#int f0/1

r1(config-if)#ip add 12.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#router eigrp 1

r1(config-router)#no auto-summary

r1(config-router)#network 13.1.1.1 0.0.0.0

r1(config-router)#network 12.1.1.1 0.0.0.0

说明：在 R1 上配置 12.1.1.0/24，13.1.1.0/24，并放入 EIGRP 进程中。

（2）配置 R2:

r2(config)#int f0/0

r2(config-if)#ip address 24.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#int f0/1

r2(config-if)#ip address 12.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#router eigrp 1

r2(config-router)#no auto-summary

r2(config-router)#network 12.1.1.2 0.0.0.0

r2(config-router)#network 24.1.1.2 0.0.0.0

说明：在 R2 上配置 12.1.1.0/24，24.1.1.0/24，并放入 EIGRP 进程中。

（3）配置 R3:

r3(config)#int f0/0

r3(config-if)#ip address 13.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#int s1/0

r3(config-if)#encapsulation frame-relay

r3(config-if)#no frame-relay inverse-arp

r3(config-if)#no arp frame-relay

r3(config-if)#ip address 34.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#frame-relay map ip 34.1.1.4 304 broadcast

r3(config)#router eigrp 1

r3(config-router)#no auto-summary

r3(config-router)#network 13.1.1.3 0.0.0.0

r3(config-router)#network 34.1.1.3 0.0.0.0

说明：在 R3 上配置 13.1.1.0/24，34.1.1.0/24，并放入 EIGRP 进程中，其中连接 R4 的链路为 Frame-Relay 环境。

（4）配置 R4:

r4(config)#int f0/0

r4(config-if)#ip add 24.1.1.4 255.255.255.0

r4(config-if)#no sh

```
r4(config)#int s1/0
r4(config-if)#encapsulation frame-relay
r4(config-if)#no frame-relay inverse-arp
r4(config-if)#no arp frame-relay
r4(config-if)#ip address 34.1.1.4 255.255.255.0
r4(config-if)#no shutdown
r4(config-if)#frame-relay map ip 34.1.1.3 403 broadcast

r4(config)#int loopback 0
r4(config-if)#ip address 4.4.4.4 255.255.255.0

r4(config)#router eigrp 1
r4(config-router)#no auto-summary
r4(config-router)#network 24.1.1.4 0.0.0.0
r4(config-router)#network 34.1.1.4 0.0.0.0
r4(config-router)#network 4.4.4.4 0.0.0.0
```

说明：在 R4 上配置 24.1.1.0/24，34.1.1.0/24，4.4.4.0/24，并放入 EIGRP 进程中，其中连接 R4 的链路为 Frame-Relay 环境。

2.测试 EIGRP 邻居

（1）查看 R1 当前的 EIGRP 邻居:

r1#show ip eigrp neighbors

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
1	13.1.1.3	Fa0/0	10	00:04:38	213	1278	0	4
0	12.1.1.2	Fa0/1	10	00:06:32	165	990	0	11

r1#

说明：R1 已经与 R2 和 R3 建立邻居，因为 Hold 栏的值在 10-15 范围内，所以 Hello 间隔是 5 秒，Hold-time 是 15 秒。

（2）查看 R3 当前的 5 个 K 值：

r3#sh ip protocols

Routing Protocol is "eigrp 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 1

Redistributing: eigrp 1

EIGRP NSF-aware route hold timer is 240s

Automatic network summarization is not in effect

Maximum path: 4

Routing for Networks:

13.1.1.3/32

34.1.1.3/32

Routing Information Sources:

Gateway	Distance	Last Update
13.1.1.1	90	00:01:45
34.1.1.4	90	00:01:55

Distance: internal 90 external 170

r3#

说明：R3 当前的 5 个 K 值分别为 K1=1, K2=0, K3=1, K4=0, K5=0，与默认值相同。

（3）修改 R3 的 5 个 K 值：

r3(config)#router eigrp 1

r3(config-router)#metric weights 0 1 1 1 0 0

说明：手工将 R3 的 5 个 K 值修改为 K1=1, K2=1, K3=1, K4=0, K5=0 ，命令中第一个值为 TOS ，IOS 中必须为 0 。

（4）查看 R3 修改后的 5 个 K 值：

r3#sh ip protocols

Routing Protocol is "eigrp 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Default networks flagged in outgoing updates

Default networks accepted from incoming updates

EIGRP metric weight K1=1, K2=1, K3=1, K4=0, K5=0

EIGRP maximum hopcount 100

EIGRP maximum metric variance 1

Redistributing: eigrp 1

EIGRP NSF-aware route hold timer is 240s

Automatic network summarization is not in effect

Maximum path: 4

Routing for Networks:

13.1.1.3/32

34.1.1.3/32

Routing Information Sources:

Gateway	Distance	Last Update
13.1.1.1	90	00:00:17
34.1.1.4	90	00:00:17

Distance: internal 90 external 170

r3#

说明：输入表示 R3 的 5 个 K 值已经被修改为 K1=1, K2=1, K3=1, K4=0, K5=0。

（5）再次查看 R1 的邻居：

r1#show ip eigrp neighbors

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
0	12.1.1.2	Fa0/1	13	00:11:43	200	1200	0	21

r1#

*Mar 1 00:18:52.287: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 13.1.1.3 (FastEthernet0/0) is down: Interface Goodbye received

r1#

说明：R1 已经断开与 R3 的邻居关系，正是因为双方 K 值不同，因为 R1 为默认值，而 R3 为改而的值。

★最后将 R3 的值恢复默认，并建立正常邻居关系。

3.测试 EIGRP 带宽计算

（1）查看 R2 到目标 4.4.4.4 的出口 F0/0 的带宽与延迟：

r2#sh int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is c000.0a34.0000 (bia c000.0a34.0000)

Internet address is 24.1.1.2/24

MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,

说明：R2 到目标 4.4.4.4 的出口 F0/0 的带宽为 100000 Kbit，延迟 100 usec

（2）查看 R4 到目标 4.4.4.4 的出口 loopback 0 的带宽与延迟：

r4#sh interfaces loopback 0

Loopback0 is up, line protocol is up

Hardware is Loopback

Internet address is 4.4.4.4/24

MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,

说明：R4 到目标 4.4.4.4 的出口 loopback 0 的带宽为 8000000 Kbit，延迟 5000 usec。

（3）使用公式计算 R2 到目标 4.4.4.4 的 Metric 值：

公式为：

$$\left(\frac{1000\,0000}{\text{最小 Bandwidth}} + \frac{\text{Delay之和}}{10}\right) \times 256$$

R2 到 4.4.4.4 链路中的最小带宽为 100 000 Kbit，延迟之和为 100 usec+5000 usec=5100 usec

应用到公式中为：

（1000 0000/100 000+5100/10）×256

↓

（100+510）×256

↓

610×256=156160

说明：所以 R2 到目标 4.4.4.4 的 Metric 值为 156160。

（4）查看 R2 到目标 4.4.4.4 的 Metric 值：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

D 34.1.1.0 [90/2172416] via 24.1.1.4, 00:10:50, FastEthernet0/0

4.0.0.0/24 is subnetted, 1 subnets

D 4.4.4.0 [90/156160] via 24.1.1.4, 00:14:27, FastEthernet0/0

24.0.0.0/24 is subnetted, 1 subnets

C 24.1.1.0 is directly connected, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

D 13.1.1.0 [90/2174976] via 24.1.1.4, 00:06:14, FastEthernet0/0

r2#

说明：R2 到 4.4.4.4 的 Metric 值确实为 156160。

（5）修改 R2 到目标 4.4.4.4 的最小带宽，影响最终 Metric：

r2(config)#int f0/0

r2(config-if)#bandwidth 50000

说明：之前 R2 的出口 F0/0 的带宽为 100000 Kbit，现在改为 50000 Kbit，该值将影响最终到目标 4.4.4.4 的 Metric 值。

（6）查看 R2 修改最小带宽后到目标 4.4.4.4 的 Metric：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     34.0.0.0/24 is subnetted, 1 subnets

D       34.1.1.0 [90/2172416] via 24.1.1.4, 00:00:18, FastEthernet0/0

     4.0.0.0/24 is subnetted, 1 subnets

D       4.4.4.0 [90/181760] via 24.1.1.4, 00:00:07, FastEthernet0/0

     24.0.0.0/24 is subnetted, 1 subnets

C       24.1.1.0 is directly connected, FastEthernet0/0

     12.0.0.0/24 is subnetted, 1 subnets

C       12.1.1.0 is directly connected, FastEthernet0/1

     13.0.0.0/24 is subnetted, 1 subnets

D       13.1.1.0 [90/30720] via 12.1.1.1, 00:02:14, FastEthernet0/1

r2#
```

说明：R2 修改最小带宽后到目标 4.4.4.4 的 Metric 为 181760，大于之前的 156160，因为带宽越小，性能越差，Metric 就越大；修改接口的带宽会影响到 EIGRP

对接口真实带宽的认知，并自动调整与带宽有关的所有参数，但须注意， Hello 包的间隔不会因此改变。

（7）调整 EIGRP 在接口上的最大使用率：

说明：默认 EIGRP 认为自己的流量可占用接口带宽的 50%，而修改接口的带宽会影响到 EIGRP 对接口真实带宽的认知，并自动调整 EIGRP 可使用该接口的最大带宽，所以在修改接口带宽之后，调整 EIGRP 流量使用率到相应值，注意，在没有手工修改接口带宽的情况下，也可以随意调整 EIGRP 在接口上的使用率。

r2(config)#int f0/0

r2(config-if)#ip bandwidth-percent eigrp 1 150

说明：调整 EIGRP 在接口上 F0/0 上的最大使用率为 150%，也就是 50Mbit/s ×150%=75 Mbit/s。

4.测试 EIGRP 非等价负载均衡

（1）查看 R1 到目标 4.4.4.4 的 RD：

r1#show ip eigrp topology

IP-EIGRP Topology Table for AS(1)/ID(13.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,

r - reply Status, s - sia Status

P 4.4.4.0/24, 1 successors, FD is 158720

via 12.1.1.2 (158720/156160), FastEthernet0/1

P 12.1.1.0/24, 1 successors, FD is 28160

via Connected, FastEthernet0/1

P 13.1.1.0/24, 1 successors, FD is 2562560

via Connected, FastEthernet0/0

via 12.1.1.2 (2177536/2174976), FastEthernet0/1

P 24.1.1.0/24, 1 successors, FD is 30720

via 12.1.1.2 (30720/28160), FastEthernet0/1

P 34.1.1.0/24, 1 successors, FD is 2174976

via 12.1.1.2 (2174976/2172416), FastEthernet0/1

via 13.1.1.3 (3074560/2169856), FastEthernet0/0

r1#

说明：因为 R1 与 R2 是 EIGRP 邻居，R2 将路由信息发给 R1 之后，R2 到目标 4.4.4.4 的 Metric 值 156160 就成为了 R1 到目标 4.4.4.4 的 RD 值，而当前 R1 到目标 4.4.4.4 的 FD 为 158720，这条信息将被放入路由表中使用；拓朴数据库中显示确实如此；而拓朴数据库中为什么没有 R1 经过 R3 到目标 4.4.4.4 的路径，下面来查看：

（2）查看 R1 经过 R3 到目标 4.4.4.4 的路径：

说明：R1 经过 R3 到目标 4.4.4.4 的路径不能存放于拓朴数据库中，应该是不满足 FC 的条件（R3 到目标 4.4.4.4 的 Metric 值必须小于 R1 当前的 FD 值 158720）

所以，我们手工计算 R3 到目标 4.4.4.4 的 Metric 值

查看带宽与延迟：

r4#sh interfaces loopback 0

Loopback0 is up, line protocol is up

Hardware is Loopback

Internet address is 4.4.4.4/24

MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,

r3#sh int s1/0

Serial1/0 is up, line protocol is up

Hardware is M4T

Internet address is 34.1.1.3/24

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec

R3 到 4.4.4.4 链路中的最小带宽为 1544 Kbit，延迟之和为 20000 usec +5000 usec=25000 usec

应用到公式中为：

（1000 0000/1544 +25000/10）×256

↓

（6476.6 + 2500）×256=2297856

说明：所以很明显，R3 到目标 4.4.4.4 的 Metric 值必须小 2297856 大于 R1 当前的 FD 值 158720，所以无法存放于拓朴数据库中，所以当前 R1 只使用经过 R2 到目标 4.4.4.4 的路径，如下：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

D 34.1.1.0 [90/2174976] via 12.1.1.2, 00:07:46, FastEthernet0/1

4.0.0.0/24 is subnetted, 1 subnets

D 4.4.4.0 [90/158720] via 12.1.1.2, 00:15:36, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

D 24.1.1.0 [90/30720] via 12.1.1.2, 00:19:50, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/0

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

r1#

★查看 R3 到目标 4.4.4.4 的路径：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, Serial1/0

4.0.0.0/24 is subnetted, 1 subnets

D 4.4.4.0 [90/161280] via 13.1.1.1, 00:13:15, FastEthernet0/0

24.0.0.0/24 is subnetted, 1 subnets

D 24.1.1.0 [90/33280] via 13.1.1.1, 00:13:15, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

D 12.1.1.0 [90/30720] via 13.1.1.1, 00:13:15, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/0

r3#

说明：由于 R3 从 S1/0 到目标 4.4.4.4 的 Metric 值太大，所以 R3 自己都没从 S1/0 到 4.4.4.4，而选择从 R1 到 4.4.4.4。

（3）计算 R3 成为 R1 到目标 4.4.4.4 的 FS 的条件：

因为 R1 当前的 FD 为 158720，所以 R3 到目标 4.4.4.4 的 Metric 值必须小于 158720，才能成为 FS，因为 R3 的出口 S1/0 为帧中继接口，带宽实在太低，即使没有延迟，也不能成为 FS，所以我们事先将接口带宽改为 100 000 Kbit/s，从而再修改延迟到相应值，延迟需要改成多少，需要将公式进行反运算：

公式为：

$$\left(\frac{1000\ 0000}{\text{最小 Bandwidth}} + \frac{\text{Delay之和}}{10}\right) \times 256$$

R4 loopback 0 的延迟为 5000 usec，设置总延迟为 X，则：

（1000 0000 / 100 000 + X）× 256 = 158720

↓

（100 + X）=158720 /256

↓

（100 + X）= 620

↓

X=520

所以 R3 成为 R1 到目标 4.4.4.4 的 FS 的条件的总延迟必须小于 520，等于 520 也不行，

因为延迟除以 10 得到 520，所以原始延迟为 5200，而 R4 loopback 0 的延迟为 5000 usec，得 R3 S1/0 的延迟为 200，为了取小一点的值，我们取 190，下面配置 R3 S1/0 的接口延迟为 190：

r3(config)#int s1/0

r3(config-if)#delay 19

说明：在配置时，会自动乘以 10，所以要配置 190，就配置 19。

（4）查看 R3 修改接口延迟后的情况：

r3#sh int s1/0

Serial1/0 is up, line protocol is up

Hardware is M4T

Internet address is 34.1.1.3/24

MTU 1500 bytes, BW 100000 Kbit, DLY 190 usec,

说明：延迟已经改成预计的 190 了。

（5）查看 R3 到目标 4.4.4.4 的情况：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, Serial1/0

4.0.0.0/24 is subnetted, 1 subnets

D 4.4.4.0 [90/158464] via 34.1.1.4, 00:00:53, Serial1/0

24.0.0.0/24 is subnetted, 1 subnets

D 24.1.1.0 [90/33024] via 34.1.1.4, 00:00:53, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

D 12.1.1.0 [90/30720] via 13.1.1.1, 00:00:53, FastEthernet0/0

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/0

r3#

说明：R3 已经选择从 S1/0 到 4.4.4.4，说明改动有效果。

（6）查看 R1 拓朴数据库中到目标 4.4.4.4 的情况：

r1#sh ip eigrp topology

IP-EIGRP Topology Table for AS(1)/ID(13.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,

r - reply Status, s - sia Status

P 4.4.4.0/24, 1 successors, FD is 158720

via 12.1.1.2 (158720/156160), FastEthernet0/1

via 13.1.1.3 (161024/158464), FastEthernet0/0

P 12.1.1.0/24, 1 successors, FD is 28160

via Connected, FastEthernet0/1

P 13.1.1.0/24, 1 successors, FD is 28160

via Connected, FastEthernet0/0

P 24.1.1.0/24, 1 successors, FD is 30720

via 12.1.1.2 (30720/28160), FastEthernet0/1

P 34.1.1.0/24, 1 successors, FD is 33024

via 13.1.1.3 (33024/30464), FastEthernet0/0

r1#

说明：R1 当前的拓朴数据库中同时存在 R2 和 R3 到达目标 4.4.4.4，并且显示经过 R2 的路径为 FD，值为 158720，而经过 R3 的 Metric 为 161024，明显比 FD 大，但很微小。

（7）通过修改 variance 值使 R1 到目标 4.4.4.4 执行非等价负载均衡

r1(config)#router eigrp 1

r1(config-router)#variance 2

说明：因为当前 FD 为 158720，要包含 161024，只需要将 FD 为 158720 扩大 2 倍即可，值为 158720×2=317440。

（8）查看 R1 负载均衡路由表：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

D 34.1.1.0 [90/33024] via 13.1.1.3, 00:00:21, FastEthernet0/0

4.0.0.0/24 is subnetted, 1 subnets

D 4.4.4.0 [90/161024] via 13.1.1.3, 00:00:21, FastEthernet0/0

[90/158720] via 12.1.1.2, 00:00:21, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

D 24.1.1.0 [90/30720] via 12.1.1.2, 00:00:21, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/0

r1#

说明：R1 已经执行到 4.4.4.4 的负载均衡。

（9）测试负载均衡:

r1#traceroute 4.4.4.4

Type escape sequence to abort.

Tracing the route to 4.4.4.4

1 13.1.1.3 92 msec

12.1.1.2 144 msec

13.1.1.3 156 msec

2 24.1.1.4 92 msec

34.1.1.4 112 msec *

r1#

说明：R1 已经执行到 4.4.4.4 的负载均衡。

（10）查看邻居发送的路由条目：

说明：当 EIGRP 用于复杂大型网络时，有时需要查看从邻居收到的路由条目情况。

r1#sh ip eigrp 1 accounting

IP-EIGRP accounting for AS(1)/ID(13.1.1.1)

Total Prefix Count: 5 States: A-Adjacency, P-Pending, D-Down

State	Address/Source	Interface	Prefix Count	Restart Count	Restart/ Reset(s)
A	13.1.1.3	Fa0/0	3	0	0
A	12.1.1.2	Fa0/1	2	0	0

r1#

说明：R1 从 13.1.1.3（R3）收到 3 条，从 12.1.1.2（R2）收到 2 条。

5:测试 EIGRP Stub

（1）查看 R2 当前的邻居详情：

r2#sh ip eigrp neighbors detail

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
1	24.1.1.4	Fa0/0	12	00:55:48	185	1110	0	29
	Version 12.4/1.2, Retrans: 3, Retries: 0, Prefixes: 2							
0	12.1.1.1	Fa0/1	10	00:59:57	238	2142	0	32
	Version 12.4/1.2, Retrans: 4, Retries: 0, Prefixes: 2							

r2#

说明：R2 当前有两个邻居 12.1.1.1（R1），24.1.1.4（R4），并且为正常邻居。

（2）配置 R4 为 EIGRP Stub:

r4(config)#router eigrp 1

r4(config-router)#eigrp stub

说明：将 R4 配置为 EIGRP Stub，默认只向外发送直连和汇总路由。

（3）查看配置 R4 为 EIGRP Stub 后，R2 的邻居详情：

r2#sh ip eigrp neighbors detail

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
1	24.1.1.4	Fa0/0	12	00:00:06	234	1404	0	32

Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 2

Stub Peer Advertising (CONNECTED SUMMARY) Routes

Suppressing queries

0	12.1.1.1	Fa0/1	12	01:00:32	191	1146	0	41

Version 12.4/1.2, Retrans: 5, Retries: 0, Prefixes: 2

r2#

说明：结果中显示 24.1.1.4（R4）当前为 EIGRP Stub peer，并且是默认的只发送直连和汇总路由。

（4）修改 R4 的 EIGRP Stub 参数:

```
r4(config)#router eigrp 1
r4(config-router)#eigrp stub receive-only
```

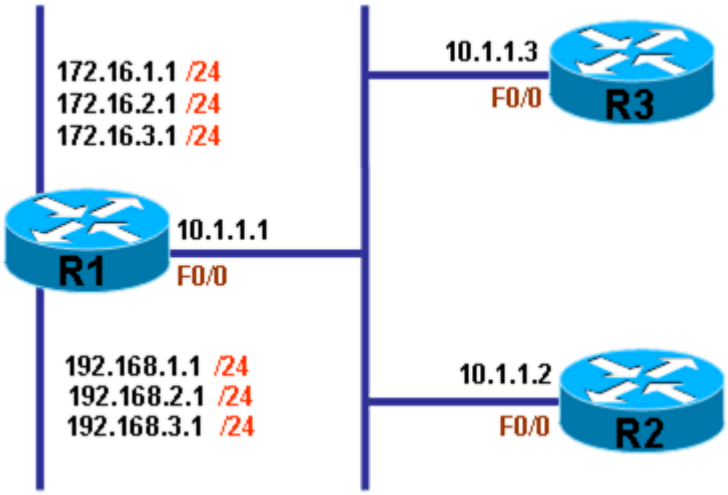
说明：EIGRP Stub 默认只向外发送直连和汇总路由，现在将 R4 改为只收路由，但不发送任何路由。

（5）再次查看 R2 的路由情况:

```
r2#sh ip route

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

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

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

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

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

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

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     34.0.0.0/24 is subnetted, 1 subnets

D       34.1.1.0 [90/35584] via 12.1.1.1, 00:00:50, FastEthernet0/1

     24.0.0.0/24 is subnetted, 1 subnets

C       24.1.1.0 is directly connected, FastEthernet0/0

     12.0.0.0/24 is subnetted, 1 subnets

C       12.1.1.0 is directly connected, FastEthernet0/1

     13.0.0.0/24 is subnetted, 1 subnets

D       13.1.1.0 [90/30720] via 12.1.1.1, 00:14:35, FastEthernet0/1

r2#
```

说明：因为 4.4.4.4 是 R4 的直连网络，而 R4 当前只收路由，却不发任何路由，所以 R2 没有从 R4 收到任何路由。即使是 4.4.4.4。

以下图为例，配置 EIGRP 手工汇总，EIGRP 认证，EIGRP 默认路由



1.配置基础网络环境

（1）配置 R1：

r1(config)#int f0/0

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#int loopback 172

r1(config-if)#ip address 172.16.1.1 255.255.255.0

r1(config-if)#ip address 172.16.2.1 255.255.255.0 secondary

r1(config-if)#ip address 172.16.3.1 255.255.255.0 secondary

r1(config)#int loopback 192

r1(config-if)#ip address 192.168.1.1 255.255.255.0

r1(config-if)#ip address 192.168.2.1 255.255.255.0 secondary

r1(config-if)#ip address 192.168.3.1 255.255.255.0 secondary

r1(config)#router eigrp 1

r1(config-router)#network 10.0.0.0

r1(config-router)#redistribute connected metric 10000 100 255 1 1500

说明：在 R1 上配置了 10.1.1.0/24， 172.16.1.0/24， 172.16.2.0/24，172.16.3.0/24，192.168.1.0/24，192.168.2.0/24，192.168.3.0/24，将启动 EIGRP，但默认为关闭自动汇总；将 10.1.1.0/24 放入 EIGRP 进程，并将其它直连路由重分布进 EIGRP。

（2）配置 R2:

r2(config)#int f0/0

r2(config-if)#ip address 10.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#router eigrp 1

r2(config-router)#network 10.0.0.0

说明：在 R2 上配置了 10.1.1.0/24，并将其放入 EIGRP 进程。

（3）配置 R3:

r3(config)#int f0/0

r3(config-if)#ip address 10.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#router eigrp 1

r3(config-router)#network 10.0.0.0

说明：在 R3 上配置了 10.1.1.0/24，并将其放入 EIGRP 进程。

2.测试 EIGRP 手工汇总

（1）查看 R2 当前自动汇总的状态：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

```
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 3 subnets
D EX     172.16.1.0 [170/284160] via 10.1.1.1, 00:01:43, FastEthernet0/0
D EX     172.16.2.0 [170/284160] via 10.1.1.1, 00:01:43, FastEthernet0/0
D EX     172.16.3.0 [170/284160] via 10.1.1.1, 00:01:43, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
D EX 192.168.1.0/24 [170/284160] via 10.1.1.1, 00:01:43, FastEthernet0/0
D EX 192.168.2.0/24 [170/284160] via 10.1.1.1, 00:01:43, FastEthernet0/0
D EX 192.168.3.0/24 [170/284160] via 10.1.1.1, 00:01:43, FastEthernet0/0
r2#
```

说明：因为 EIGRP 无法对外部路由进行自动汇总，从 R2 的路由表中也可以看出，外部路由不受自动汇总影响。

（2）修改 R1 直连路由为 EIGRP 内部路由：

```
r1(config)#router eigrp 1
r1(config-router)#no redistribute connected metric 10000 100 255 1 1500
r1(config-router)#network 172.16.0.0 0.0.255.255
r1(config-router)#network 192.168.0.0 0.0.255.255
```

说明：取消重分布直连路由，并直连网段发布为 EIGRP 内部路由。

（3）再次查看 R2 当前自动汇总的状态：

```
r2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

D     172.16.0.0/16 [90/156160] via 10.1.1.1, 00:00:20, FastEthernet0/0
      10.0.0.0/24 is subnetted, 1 subnets
C        10.1.1.0 is directly connected, FastEthernet0/0
D     192.168.1.0/24 [90/156160] via 10.1.1.1, 00:00:14, FastEthernet0/0
D     192.168.2.0/24 [90/156160] via 10.1.1.1, 00:00:14, FastEthernet0/0
D     192.168.3.0/24 [90/156160] via 10.1.1.1, 00:00:14, FastEthernet0/0
r2#
```

说明： EIGRP 自动汇总对内部路由产生了效果，将 172.16.1.0/24，

172.16.2.0/24，172.16.3.0/24 自动汇总成了 B 类网段 172.16.0.0/16。

（4）在 R1 上配置手工汇总：

r1(config)#int f0/0

r1(config-if)#ip summary-address eigrp 1 172.16.0.0 255.255.252.0

r1(config-if)#ip summary-address eigrp 1 192.168.0.0 255.255.252.0

说明：EIGRP 手工汇总同 RIP，是将路由汇总后发出去，是对出去的路由生效，而不是对进来的路由生效。

（5）查看 R2 当前的路由表：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

D 172.16.0.0/22 [90/156160] via 10.1.1.1, 00:00:42, FastEthernet0/0

D 172.16.0.0/16 [90/156160] via 10.1.1.1, 00:02:02, FastEthernet0/0

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

D 192.168.0.0/22 [90/156160] via 10.1.1.1, 00:00:32, FastEthernet0/0

r2#

说明：EIGRP 手工汇总可以将路由汇总为任意掩码长度，不受主类地址掩码长度影响，例如将 C 类地址汇总成了低于 24 位的掩码长度，并且从路由表中可以看出，在自动汇总和手工汇总都开启时，两者同时生效。

（6）关闭自动汇总：

r1(config)#router eigrp 1

r1(config-router)#no auto-summary

说明：关闭自动汇总。

（7）查看 R2 的路由表：

r2#sh ip route

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

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

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

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

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

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

        o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     172.16.0.0/22 is subnetted, 1 subnets

D       172.16.0.0 [90/156160] via 10.1.1.1, 00:01:35, FastEthernet0/0

     10.0.0.0/24 is subnetted, 1 subnets

C       10.1.1.0 is directly connected, FastEthernet0/0

D    192.168.0.0/22 [90/156160] via 10.1.1.1, 00:01:25, FastEthernet0/0

r2#
```

说明：因为关闭了自动汇总，所以只有手工汇总生效，路由表中已经没有明细路由了。

3.测试 EIGRP 认证：

（1）在 R1 上配置 EIGRP 认证：

```
r1(config)#key chain ccie

r1(config-keychain)#key 1

r1(config-keychain-key)#key-string cisco
```

r1(config)#int f0/0

r1(config-if)#ip authentication mode eigrp 1 md5

r1(config-if)#ip authentication key-chain eigrp 1 ccie

说明：在 R1 上开启 EIGRP 认证，并在接口 F0/0 上启用，1 号密码为 cisco。

（2）查看 R2 当前 EIGRP 邻居状态：

r2#sh ip eig neighbors

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
1	10.1.1.3	Fa0/0	11	00:08:32	154	924	0	29

r2#

说明：由于 R1 启用了 EIGRP 认证，而 R2 没有启用认证，所以 R2 无法与 R1 保持邻居关系，但与 R3 的邻居关系正常。

（3）在 R2 上开启 EIGRP 认证：

r2(config)#key chain ccie

r2(config-keychain)#key 1

r2(config-keychain-key)#key-string abc

r2(config-keychain-key)#exi

r2(config-keychain)#key 2

r2(config-keychain-key)#key-string cisco

r2(config)#int f0/0

r2(config-if)#ip authentication mode eigrp 1 md5

r2(config-if)#ip authentication key-chain eigrp 1 ccie

说明：在 R2 上配置了 EIGRP 认证，共设置了两个密码，1 号密码为 abc，2 号密码为 cisco。

（4）再次查看 R2 的 EIGRP 邻居状态：

r3#sh ip eigrp neighbors

IP-EIGRP neighbors for process 1

r3#

说明：当 R2 开启认证后，已经没有任何邻居了，即使与开启了认证的 R1 也不能成为邻居，因为虽然双方有相同有密码，但是密码序号不一样，R1 是 1 号密码为 cisco，而 R2 是 2 号密码为 cisco。

（5）在 R3 上开启 EIGRP 认证：

r3(config)#key chain ccie

r3(config-keychain)#key 1

r3(config-keychain-key)#key-string cisco

r3(config-keychain-key)#exit

r3(config-keychain)#key 2

r3(config-keychain-key)#key-string abc

r3(config)#int f0/0

r3(config-if)#ip authentication mode eigrp 1 md5

r3(config-if)#ip authentication key-chain eigrp 1 ccie

说明：在 R3 上配置了 EIGRP 认证，共设置了两个密码，1 号密码为 abc，2 号密码为 cisco。

（6）查看 R3 的 EIGRP 邻居状态：

r3#sh ip eigrp neighbors

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
0	10.1.1.1	Fa0/0	13	00:00:38	215	1290	0	31

r3#

说明：因为 R1 的 1 号密码为 cisco，而 R3 的 1 号密码也为 cisco，双方相同，所以 R3 与 R1 建立了邻居关系，由于 R2 的 1 号密码为 abc，与大家不同，所以没能建立邻居；结果证明，需要注意，EIGRP 认证时，需要双方密码相同，并且号码相同，才能建立邻居，而且必须是双方最上面的一组密码相同才可以。

注：有的文档会提示 EIGRP 不用号码匹配也能建立邻居，所以不排除某些 IOS 版本可能是那样，但是为了保险起见，请一定保证双方第一组号码和密码相同，否则结果不能保证。

4.测试 EIGRP 路由

（1）查看 R3 当前的路由表：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/22 is subnetted, 1 subnets
D       172.16.0.0 [90/156160] via 10.1.1.1, 00:00:49, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
D    192.168.0.0/22 [90/156160] via 10.1.1.1, 00:00:49, FastEthernet0/0
r3#
```

说明：当前 R3 与 R1 为正常邻居，所以 R3 收到了 R2 发来的两条直连路由 172.16.0.0/22 和 192.168.0.0/22。

（2）在 R1 上配置静态路由：

r1(config)#ip route 100.1.1.0 255.255.255.0 loopback 172

r1(config)#ip route 172.16.100.0 255.255.255.0 loopback 172

说明：R1 上配置了指向 100.1.1.0/24 和 172.16.100.0/24 的静态路由，并且出口为直连出口。

（3）再次查看 R3 当前的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

D 172.16.0.0/22 [90/156160] via 10.1.1.1, 00:02:41, FastEthernet0/0

D 172.16.100.0/24 [90/156160] via 10.1.1.1, 00:00:41, FastEthernet0/0

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

D 192.168.0.0/22 [90/156160] via 10.1.1.1, 00:02:41, FastEthernet0/0

r3#

说明：R1 上配置的指向接口的静态路由 172.16.100.0/24 被通告给了邻居，所以手工配置的静态路由如果是指的直连出口，只要静态路由被 network 命令包含在内，就会被通告进 EIGRP 进程，而不管其是真正的直连路由，还是手工指定的静态路上。

（4）在 R1 上 network 静态路由:

r1(config)#router eigrp 1

r1(config-router)#network 100.0.0.0

说明：将静态路由 100.1.1.0/24 通过命令 network 放入 EIGRP 进程。

（5）查看 R3 的路由表：

r3#sh ip route

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

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

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

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

```
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     100.0.0.0/24 is subnetted, 1 subnets
D       100.1.1.0 [90/156160] via 10.1.1.1, 00:00:14, FastEthernet0/0
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.16.0.0/22 [90/156160] via 10.1.1.1, 00:06:18, FastEthernet0/0
D       172.16.100.0/24 [90/156160] via 10.1.1.1, 00:04:17, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
D    192.168.0.0/22 [90/156160] via 10.1.1.1, 00:06:18, FastEthernet0/0
r3#
```

说明：因为指向接口的静态路由 100.1.1.0/24 被 network 命令包含在内，所以被通告进了 EIGRP 进程。

5.测试 EIGRP 默认路由

说明：EIGRP 发布默认路由的方法有三种，一是直接手工配置默认路由指向空接口（null 0），并将其通过命令 network 发布进 EIGRP；二是直接手工配置默认路由指向空接口（null 0），并将其重分布进 EIGRP；三是通过命令 ip default-network

指定默认网关，然后会自动被 EIGRP 传递。

（1）配置指向空接口（null 0）的默认路由：

r1(config)#ip route 0.0.0.0 0.0.0.0 null 0

（2）network 默认路由进 EIGRP：

r1(config)#router eigrp 1

r1(config-router)#network 0.0.0.0

说明：配置指向空接口的默认路由。

（3）查看 R3 的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 0.0.0.0

100.0.0.0/24 is subnetted, 1 subnets

D 100.1.1.0 [90/156160] via 10.1.1.1, 00:02:34, FastEthernet0/0

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

D 172.16.0.0/22 [90/156160] via 10.1.1.1, 00:08:38, FastEthernet0/0

D 172.16.100.0/24 [90/156160] via 10.1.1.1, 00:06:37, FastEthernet0/0

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, FastEthernet0/0

D* 0.0.0.0/0 [90/28160] via 10.1.1.1, 00:00:20, FastEthernet0/0

D 192.168.0.0/22 [90/156160] via 10.1.1.1, 00:08:38, FastEthernet0/0

r3#

说明：因为指向接口的默认路由被 network 命令包含在内，所以被自动通告进了 EIGRP 进程。

（4）重分布默认路由：

r1(config)#router eigrp 1

r1(config-router)#no network 0.0.0.0

r1(config-router)#redistribute static metric 10000 100 255 1 1500

说明：在 R1 上将默认路由通过重分布的方法取代 network 命令，默认路由也是静态路由的一种，所以是重分布静态路由。

（5）查看 R3 的路由表：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 0.0.0.0

     100.0.0.0/24 is subnetted, 1 subnets
D       100.1.1.0 [90/156160] via 10.1.1.1, 00:03:20, FastEthernet0/0
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.16.0.0/22 [90/156160] via 10.1.1.1, 00:09:24, FastEthernet0/0
D EX    172.16.100.0/24 [170/284160] via 10.1.1.1, 00:00:16, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
D*EX 0.0.0.0/0 [170/284160] via 10.1.1.1, 00:00:16, FastEthernet0/0
D    192.168.0.0/22 [90/156160] via 10.1.1.1, 00:09:24, FastEthernet0/0
```

r3#

说明：默认路由被重分布进了 EIGRP。

（6）通过命令 ip default-network 发布默认路由

说明：通过命令 ip default-network 发布默认路由的方法请参见前面章节，不再详述。

OSPF

概述

路由协议 OSPF 全称为 Open Shortest Path First，也就开放的最短路径优先协议，因为 OSPF 是由 IETF 开发的，它的使用不受任何厂商限制，所有人都可以使用，所以称为开放的，而最短路径优先（SPF）只是 OSPF 的核心思想，其使用的算法是 Dijkstra 算法，最短路径优先并没有太多特殊的含义，并没有任何一个路由协议是最长路径优先的，所有协议，都会选最短的。

OSPF 的流量使用 IP 协议号 89。

OSPF 工作在单个 AS，是个绝对的内部网关路由协议（Interior Gateway Protocol，即 IGP）。

OSPF 对网络没有跳数限制，支持 Classless Interdomain Routing (CIDR)和 Variable-Length Subnet Masks (VLSMs)，没有自动汇总功能，但可以手工在任意比特位汇总，并且手工汇总没有任何条件限制，可以汇总到任意掩码长度。

OSPF 支持认证，并且支持明文和 MD5 认证；OSPF 不可以通过 Offset list 来改变路由的 metric。

OSPF 并不会周期性更新路由表，而采用增量更新，即只在路由有变化时，才会发送更新，并且只发送有变化的路由信息；事实上，OSPF 是间接设置了周期性更新路由的规则，因为所有路由都是有刷新时间的，当达到刷新时间阀值时，该路由就会产生一次更新，默认时间为 1800 秒，即 30 分钟，所以 OSPF 路由的定期更新周期默认为 30 分钟。

OSPF 所有路由的管理距离(Ddministrative Distance)为 110，OSPF 只支持等价负载均衡。

距离矢量路由协议的根本特征就是自己的路由表是完全从其它路由器学来的，并且将收到的路由条目一丝不变地放进自己的路由表，运行距离矢量路由协议的路由器之间交换的是路由表，距离矢量路由协议是没有大脑的，路由表从来不会自己计算，总是把别人的路由表拿来就用；而 OSPF 完全抛弃了这种不可靠的算法，OSPF 是典型的链路状态路由协议，路由器之间交换的并不是路由表，而是链路状态，OSPF 通过获得网络中所有的链路状态信息，从而计算出到达每个目标精确的网络路径。

OSPF 术语

Router-ID

假设这个世界上的人名字没有重复，每个人的名字都不相同，当有一天，遇上个陌生人告诉你，有任何麻烦可以找他，他一定能够帮你解决；等到你有麻烦的时候，你想找那个人帮忙，可是如果你连那个人的名字都不知道，那么也就不可能找到那个人帮忙了。OSPF 就类似于上述情况，网络中每台 OSPF 路由器都相当于一个人，OSPF 路由器之间相互通告链路状态，就等于是告诉别人可以帮别人的忙，如此一来，如果路由器之间分不清谁是谁，没有办法确定各自的身份，那么通告的链路状态就是毫无意义的，所以必须给每一个 OSPF 路由器定义一个身份，就相当于人的名字，这就是 Router-ID，并且 Router-ID 在网络中绝对不可以有重名，否则路由器收到的链路状态，就无法确定发起者的身份，也就无法通过链路状态信息确定网络位置，OSPF 路由器发出的链路状态都会写上自己的 Router-ID，可以理解为该链路状态的签名，不同路由器产生的链路状态，签名绝不会相同。

每一台 OSPF 路由器只有一个 Router-ID，Router-ID 使用 IP 地址的形式来表示，确定 Router-ID 的方法为：

★1 .手工指定 Router-ID。

★2 .路由器上活动 Loopback 接口中 IP 地址最大的，也就是数字最大的，如 C 类地址优先于 B 类地址，一个非活动的接口的 IP 地址是不能被选为 Router-ID 的。

★3 .如果没有活动的 Loopback 接口，则选择活动物理接口 IP 地址最大的。

注：如果一台路由器收到一条链路状态，无法到达该 Router-ID 的位置，就无法到达链路状态中的目标网络。

Router-ID 只在 OSPF 启动时计算，或者重置 OSPF 进程后计算。

COST

OSPF 使用接口的带宽来计算 Metric，例如一个 10 Mbit/s 的接口，计算 Coast 的方法为：

将 10 Mbit 换算成 bit，为 10 000 000 bit，然后用 10000 0000 除以该带宽，结果为 10000 0000/10 000 000 bit = 10，所以一个 10 Mbit/s 的接口，OSPF 认为该接口的 Metric 值为 10，需要注意的是，计算中，带宽的单位取 bit/s，而不是 Kbit/s，例如一个 100 Mbit/s 的接口，Cost 值为 10000 0000 /100 000 000=1，因为 Cost 值必须为整数，所以即使是一个 1000 Mbit/s（1GBbit/s）的接口，Cost 值和 100Mbit/s 一样，为 1。如果路由器要经过两个接口才能到达目标网络，那么很显然，两个接口的 Cost 值要累加起来，才算是到达目标网络的 Metric 值，所以 OSPF 路由器计算到达目标网络的 Metric 值，必须将沿途中所有接口的 Cost 值累加起来，在累加时，同 EIGRP 一样，只计算出接口，不计算进接口。

OSPF 会自动计算接口上的 Cost 值，但也可以通过手工指定该接口的 Cost 值，手工指定的优先于自动计算的值。

OSPF 计算的 Cost，同样是和接口带宽成反比，带宽越高，Cost 值越小。到达目标相同 Cost 值的路径，可以执行负载均衡，最多 6 条链路同时执行负载均衡。

链路（Link）

就是路由器上的接口，在这里，应该指运行在 OSPF 进程下的接口。

链路状态（Link-State）

链路状态（LSA）就是 OSPF 接口上的描述信息，例如接口上的 IP 地址，子网掩码，网络类型，Cost 值等等，OSPF 路由器之间交换的并不是路由表，而是链路状态（LSA），OSPF 通过获得网络中所有的链路状态信息，从而计算出到达每个目标精确的网络路径。OSPF 路由器会将自己所有的链路状态毫不保留地全部发给邻居，邻居将收到的链路状态全部放入链路状态数据库（Link-State Database），邻居再发给自己的所有邻居，并且在传递过程种，绝对不会有任何更改。通过这样的过程，最终，网络中所有的 OSPF 路由器都拥有网络中所有的链路状态，并且所有路由器的链路状态应该能描绘出相同的网络拓朴。比如现在要计算一条地铁线路图，如上海地铁二号线某段的图，如果不直接将该图给别人看，图好比是路由表，现在只是报给别人各个站的信息，该信息好比是链路状态，通过告诉别人各个站左边一站是什么，右边一站是什么，别人也能通过该信息（链路状态），画出完整的线路图（路由表），如得到如下各站信息（链路状态）：

★南京东路-站 （左边一站是人民广场，右边一站是陆家嘴）

★南京西路-站 （左边一站是静安寺，右边一站是人民广场）

★静安寺-站 （右边一站是南京西路）

★人民广场-站 （左边一站是南京西路，右边一站是南京东路）

★陆家嘴-站 （左边一站是南京东路）

还原线路图（路由表）如下：

根据分析以下两站信息（两条链路状态）：

★南京西路-站 （左边一站是静安寺，右边一站是人民广场）

★静安寺-站 （右边一站是南京西路）

计算 因为静安寺右边是南京西路，而南京西路左边是静安寺，所以静安寺和南京西路是相邻的，为 静安寺 — 南京西路，并且由于南京西路右边是人民广场，所以通过这两条信息，得出线路为 静安寺 — 南京西路 — 人民广场，继续往下

再根据如下两站信息（链路状态）：

★人民广场-站 （左边一站是南京西路，右边一站是南京东路）

★南京东路-站 （左边一站是人民广场，右边一站是陆家嘴）

计算 因为之前南京西路右边是人民广场，人民广场左边是南京西路，所以南京西路和人民广场是相邻的两站，并且人民广场右边是南京东路，得出线路为 南京西路 — 人民广场 — 南京东路，并且因为南京东路右边是陆家嘴，所以 这部分线路得知为南京西路 — 人民广场 — 南京东路 — 陆家嘴，继续往下

再根据如下一站信息（链路状态）：

★陆家嘴-站 （左边一站是南京东路）

计算 因为南京东路右边是陆家嘴，而陆家嘴左边是南京东路，所以两站相邻，得出为 南京东路 — 陆家嘴，

通过以上各部分的线路：

静安寺 — 南京西路 — 人民广场

南京西路 — 人民广场 — 南京东路 — 陆家嘴

南京东路 — 陆家嘴

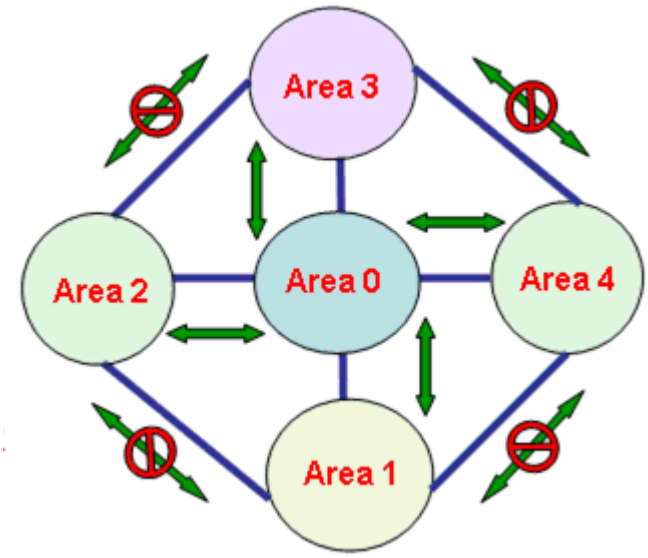
所以很轻松的就画出该段地铁线路图为：

静安寺 — 南京西路 — 人民广场 — 南京东路 — 陆家嘴

从以上计算过程可以知道，因为得到各站的信息，就能画出整条线路图，而 OSPF 也同样根据路由器各接口的信息（链路状态），计算出网络拓朴图，OSPF 之间交换链路状态，就像上面交换各站信息，而不像 RIP 和 EIGRP 直接交换路由表，交换路由表，就等于直接给人看线路图，可见 OSPF 的智能算法，比距离矢量协议对网络有更精确的认知。

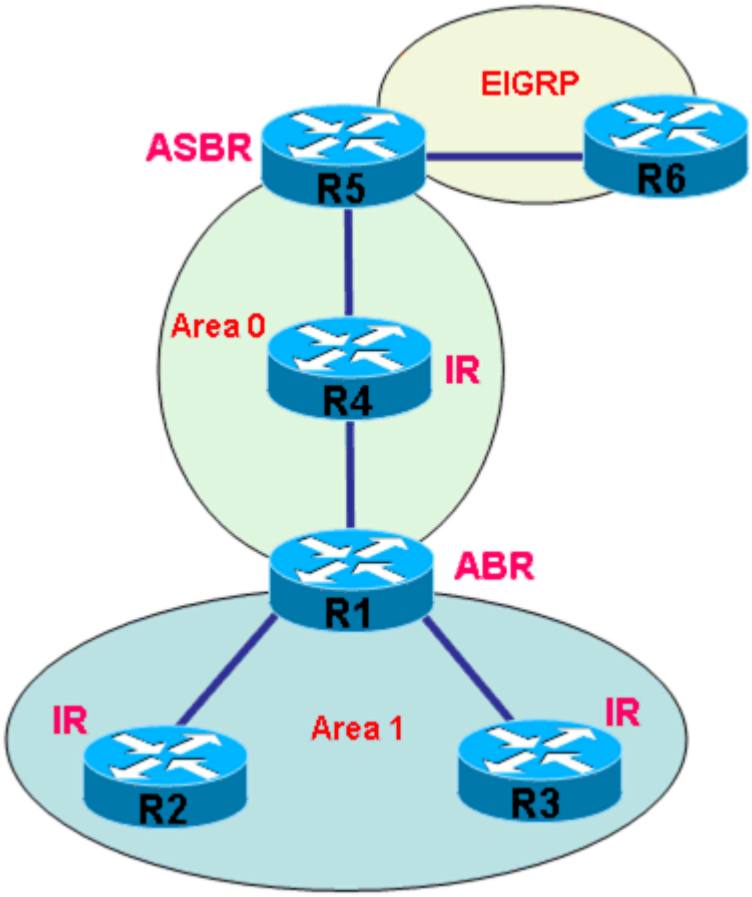
OSPF 区域

因为 OSPF 路由器之间会将所有的链路状态（LSA）相互交换，毫不保留，当网络规模达到一定程度时，LSA 将形成一个庞大的数据库，势必会给 OSPF 计算带来巨大的压力；为了能够降低 OSPF 计算的复杂程度，缓存计算压力，OSPF 采用分区域计算，将网络中所有 OSPF 路由器划分成不同的区域，每个区域负责各自区域精确的 LSA 传递与路由计算，然后再将一个区域的 LSA 简化和汇总之后转发到另外一个区域，这样一来，在区域内部，拥有网络精确的 LSA，而在不同区域，则传递简化的 LSA。区域的划分为了能够尽量设计成无环网络，所以采用了 Hub-Spoke 的拓朴架构，也就是采用核心与分支的拓朴，如下图：



区域的命名可以采用整数数字，如 1、2、3、4，也可以采用 IP 地址的形式，0.0.0.1、0.0.0.2，因为采用了 Hub-Spoke 的架构，所以必须定义出一个核心，然后其它部分都与核心相连，OSPF 的区域 0 就是所有区域的核心，称为 BackBone 区域（骨干区域），而其它区域称为 Normal 区域（常规区域），在理论上，所有的常规区域应该直接和骨干区域相连，常规区域只能和骨干区域交换 LSA，常规区域与常规区域之间即使直连也无法互换 LSA，如上图中 Area 1、Area 2、Area 3、Area 4 只能和 Area 0 互换 LSA，然后再由 Area 0 转发，Area 0 就像是一个中转站，两个常规区域需要交换 LSA，只能先交给 Area 0，再由 Area 0 转发，而常规区域之间无法互相转发。

OSPF 区域是基于路由器的接口划分的，而不是基于整台路由器划分的，一台路由器可以属于单个区域，也可以属于多个区域，如下图：



如果一台 OSPF 路由器属于单个区域，即该路由器所有接口都属于同一个区域，那么这台路由器称为 Internal Router（IR），如上图中的 R2，R3 和 R4；如果一台 OSPF 路由器属于多个区域，即该路由器的接口不都属于一个区域，那么这台路由器称为 Area Border Router （ABR），如上图中的 R1，ABR 可以将一个区域的 LSA 汇总后转发至另一个区域；如果一台 OSPF 路由器将外部路由协议重分布进 OSPF，那么这台路由器称为 Autonomous System Boundary Router （ASBR），如上图中，R5 将 EIGRP 重分进 OSPF，那么 R5 就是 ASBR，但是如果只是将 OSPF 重分布进其它路由协议，则不能称为 ASBR。

可以配置任何 OSPF 路由器成为 ABR 或 ASBR。

由于 OSPF 有着多种区域，所以 OSPF 的路由在路由表中也以多种形式存在，共分以下几种：

如果是同区域的路由，叫做 Intra-Area Route，在路由表中使用 O 来表示；

如果是不同区域的路由，叫做 Inter-Area Route 或 Summary Route，在路由表中使用 O IA 来表示；

如果并非 OSPF 的路由，或者是不同 OSPF 进程的路由，只是被重分布到 OSPF 的，叫做 External Route，在路由表中使用 O E2 或 OE 1 来表示。

当存在多种路由可以到达同一目的地时，OSPF 将根据先后顺序来选择要使用的路由，所有路由的先后顺序为：

Intra-Area — Inter-Area — External E1 — External E2，即 O — O IA — O E1 — O E2。

注：

★一台路由器可以运行多个 OSPF 进程，不同进程的 OSPF，可视为没有任何关系，如需要获得相互的路由信息，需要重分布。

★每个 OSPF 进程可以有多个区域，而路由器的链路状态数据库是分进程和分区域存放的。

邻居（Neighbor）

OSPF 只有邻居之间才会交换 LSA，路由器会将链路状态数据库中所有的内容毫不保留地发给所有邻居，要想在 OSPF 路由器之间交换 LSA，必须先形成 OSPF 邻居，OSPF 邻居靠发送 Hello 包来建立和维护，Hello 包会在启动了 OSPF 的接口上周期性发送，在不同的网络中，发送 Hello 包的间隔也会不同，当超过 4 倍的 Hello 时间，也就是 Dead 时间过后还没有收到邻居的 Hello 包，邻居关系将被断开。

两台 OSPF 路由器必须满足 4 个条件，才能形成 OSPF 邻居，4 个必备条件如下：

1.Area-id（区域号码）

即路由器之间必须配置在相同的 OSPF 区域，否则无法形成邻居。

2.Hello and Dead Interval（Hello 时间与 Dead 时间）

即路由器之间的 Hello 时间和 Dead 时间必须一致，否则无法形成邻居。

3.Authentication（认证）

路由器之间必须配置相同的认证密码，如果密码不同，则无法形成邻居。

4.Stub Area Flag（末节标签）

路由器之间的末节标签必须一致，即处在相同的末节区域内，否则无法形成邻居。

注：

★OSPF 只能使用接口的 Primary 地址建立邻居，不能使用 Secondary 建立邻居。

★路由器双方接口要么都为手工配置地址（Numbered），要么都为借用地址（Unnumbered），否则无法建立邻居。

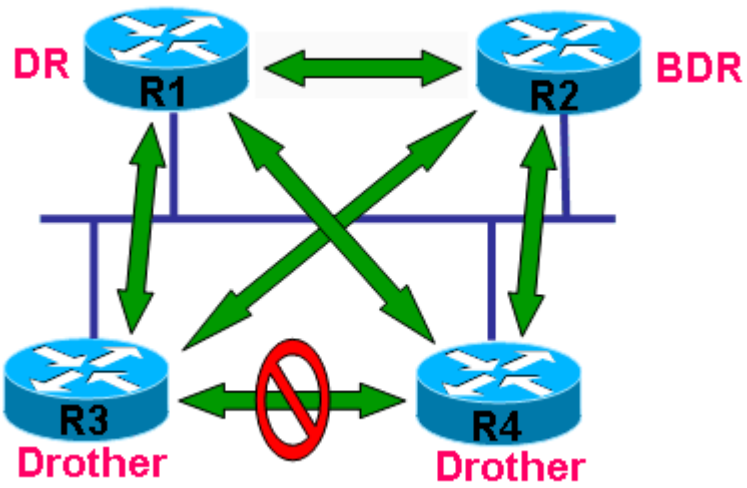
邻接（Adjacency）

两台 OSPF 路由器能够形成邻居，但并不一定能相互交换 LSA，只要能交换

LSA，关系则称为邻接（Adjacency）。邻居之间只交换 Hello 包，而邻接（Adjacency）之间不仅交换 Hello 包，还要交换 LSA。

DR/BDR

当多台 OSPF 路由器连到同一个多路访问网段时，如果每两台路由器之间都相互交换 LSA，那么该网段将充满着众多 LSA 条目，为了能够尽量减少 LSA 的传播数量，通过在多路访问网段中选择出一个核心路由器，称为 DR（Designated Router），网段中所有的 OSPF 路由器都和 DR 互换 LSA，这样一来，DR 就会拥有所有的 LSA，并且将所有的 LSA 转发给每一台路由器；DR 就像是该网段的 LSA 中转站，所有的路由器都与该中转站互换 LSA，如果 DR 失效后，那么就会造成 LSA 的丢失与不完整，所以在多路访问网络中除了选举出 DR 之外，还会选举出一台路由器作为 DR 的备份，称为 BDR（Backup Designated Router），BDR 在 DR 不可用时，代替 DR 的工作，而既不是 DR，也不是 BDR 的路由器称为 Drother，事实上，Dother 除了和 DR 互换 LSA 之外，同时还会和 BDR 互换 LSA，如下图：



上图中 R1 被选为 DR，R2 被选为 BDR，而 R3 和 R4 为 Drother，R3 同时和 R1 与 R2 互换 LSA，R4 也同时和 R1 与 R2 互换 LSA，但 R3 与 R4 却不能互换 LSA。

其实不难看出，DR 与 BDR 并没有任何本质与功能的区别，只有在多路访问的网络环境，才需要 DR 和 BDR，DR 与 BDR 的选举是在一个二层网段内选举的，即在多个路由器互连的接口范围内，与 OSPF 区域没有任何关系，一个区域可能有多个多路访问网段，那么就会存在多个 DR 和 BDR，但一个多路访问网段，只能有一个 DR 和 BDR；选举 DR 和 BDR 的规则为：

★比较接口优先级

选举优先级最高的成为 DR，优先级数字越大，表示优先级越高，被选为 DR 的几率就越大，次优先级的为 BDR，优先级范围是 0-255，默认为 1，优先级为 0 表示没有资格选举 DR 和 BDR。

★Route-Id 大小

如果在优先级都相同的情况下，Route-Id 最大的成为 DR，其次是 BDR，数字越大，被选为 DR 的几率就越大。

因为所有路由器都能与 DR 和 BDR 互换 LSA，所以所有路由器都与 DR 和 BDR 是邻接（Adjacency）关系，而 Drother 与 Drother 之间无法互换 LSA，所以 Drother 与 Drother 之间只是邻居关系。

在一个多路访问网络中，选举 DR 和 BDR 是有时间限制的，该时间为 Wait 时间，默认为 4 倍的 Hello 时间，即与 Dead 时间相同，如果 OSPF 路由器在超过 Wait 时间后也没有其它路由器与自己竞争 DR 与 BDR 的选举，那么就选自己为 DR；当一个多路访问网络中选举出 DR 与 BDR 之后，在 DR 与 BDR 没有失效的情况下，不会进行重新选举，也就是在选举出 DR 与 BDR 之后，即使有更高优先级的路由器加入网络，也不会影响 DR 与 BDR 的角色，在越出选举时间（Wait 时间）后，只有 DR 与 BDR 失效后，才会重新选举。DR 失效后，会同时重新选举 DR 与 BDR，而在 BDR 失效后，只会重新选举 BDR。

DR 和 BDR 与 Drother 的数据包处理会有所不同，

所有 OSPF 路由器，包括 DR 与 BDR，都能够接收和传递目标地址为 224.0.0.5

的数据包。

只有 DR 和 BDR 才能接收和传递目标地址为 224.0.0.6 的数据包。

由此可见，Drother 路由器将数据包发向目标地址 224.0.0.6，只能被 DR 和 BDR 接收，其它 Drother 不能接收；而 DR 和 BDR 将数据包发向目标地址 224.0.0.5，可以被所有路由器接收。

OSPF 数据包交换过程

从 OSPF 建立邻居，到 LSA 的互换，到路由表的计算，需要经过一系列的数据包交换过程，过程如下：

Hello

↓

Database Description Packets （DBD）

↓

Link-state Request （LSR）

↓

Link-state update（LSU）

↓

LSDB

具体情况如下：

Hello

Hello 包是用来建立和维护 OSPF 邻居的，要交换 LSA，必须先通过 Hello 包建立 OSPF 邻居。

Database Description Packets （DBD）

当一个人去书店买书时，想要决定买哪本书，并不会先将书店里所有的书都看一遍，才做决定买哪本好，通常是只看书的目录，或者大概翻一翻，再对比一下，就能决定买哪本；而 OSPF 的 LSA 交换也是一样的，邻居建立之后，并不会立刻就将自己链路状态数据库中所有的 LSA 全部发给邻居，而是将 LSA 的基本描述信息发给邻居，这就是 Database Description Packets （DBD），是 LSA 的目录信息，相当于书的目录，邻居在看完 DBD 之后，就能知道哪些 LSA 是需要邻居发送给自己的。

Link-state Request （LSR）

邻居在看完发来的 LSA 描述信息（DBD）之后，就知道哪些 LSA 是需要邻居发送给自己的，自己就会向邻居发送 LSA 请求（LSR），告诉邻居自己需要哪些 LSA。

Link-state update（LSU）

当邻居收到其它路由器发来的 LSA 请求（LSR）之后，就知道对方需要哪些 LSA，然后根据 LSR，将完整的 LSA 内容全部发给邻居，以供计算路由表。

LSDB

就是已经收到了所有需要邻居发给自己的 LSA，这时的链路状态数据库已经达到收敛状态。

OSPF 启动过程

路由器从启动 OSPF 进程，到根据链路状态数据库计算出路由表，同样需要经历一系列的启动过程，总共有 8 种可能的启动过程，但并不是一定会经历这 8 个过程，具体过程如下：

Down → Attempt → Init → Two-way → Exstart → Exchange → Loading → Full

每个过程详细情况如下：

Down

路由器刚刚启动 OSPF 进程，还没有从任何路由器收到任何数据包，Hello 包也没有收到，在此进程，可以向外发送 Hello 包，以试图发现邻居。

Attempt

因为 OSPF 使用组播发送数据包，如使用组播发送 Hello 包，如果 Hello 包不能发出去被其它路由器收到，就不能和其它路由器建立 OSPF 邻居；在一些组播不能发送的网络中，例如帧中继这样的非广播网络环境，组播不能够传递，在这种情况下，就需要指定 OSPF 使用单播向邻居发送 Hello 包，以此试图和指定的邻居建立 OSPF 邻居关系，在此状态下，OSPF 称为 Attempt 状态。

Init

只是 OSPF 路由器一方收到了另一方的 Hello，但并没有双方都交换 Hello，也

就是对方的 Hello 中还没有将自己列为邻居。

Two-way

双方都已经交换了 Hello 信息，并且从 Hello 中看到对方已经将自己列为邻居，此状态，就表示 OSPF 邻居关系已经建立，并且如果是需要选举 DR 和 BDR 的话，也已经选举出来，但 OSPF 邻居之间并不一定就会交换 LSA，如果不需要交换 LSA，则永远停留在此状态，如果需要形成邻接并互相交换 LSA，则状态继续往下进行。（比如 Drother 与 Drother 之间将永远停留在 Two-way 状态，因为 Drother 与 Drother 之间不需要交换 LSA。）

Exstart

因为在 OSPF 邻居之间交换完整的 LSA 之前，会先发送 Database Description Packets （DBD），Link-state Request （LSR）等数据包，邻居之间是谁先发，谁后发，需要确定顺序，在 Exstart

状态，就是确定邻居之间的主从关系（Master—Slave 关系），Router-ID 数字大的为主路由器，另一端为从路由器，由主路由器先向从路由器发送信息。在选举 DR 与 BDR 的网络环境中，并不一定 DR 就是主路由器，BDR 就是从路由器，因为 DR 和 BDR 可以通过调整接口优先级来控制，所以 DR 也许是因为优先级比 BDR 高，而 Router-ID 并不比 BDR 高。

注：在任何网络环境下，OSPF 在交换 LSA 之前，都需要确定主从关系。

Exchange

就是交换 Database Description Packets （DBD）的过程，DBD 只是 LSA 的简单描述，只包含 LSA 的一些头部信息，收到 DBD 的路由器会和自己的链路状态数据库作对比，确定需要哪些 LSA 的完整信息，就会发送 LSR 请求给邻居。

Loading

邻居根据收到的 LSR（Link-State Request），向对方回复 Link-state update（LSU）。

Full

等到 OSPF 都收到了邻居回复的所有 Link-state update（LSU），那么此时的数据库状态就变成了收敛状态，此状态就是 Full 状态，但此时只是数据库已经同步，但路由表却还在计算当中。

注：除了 Two-way 和 Full 这两个状态，邻居停留在任何状态，都是不正常。

OSPF 网络类型（Network Type）

OSPF 是一个在各方面都考虑比较周全的路由协议，也会因此将该协议变得更为复杂化，OSPF 并不像 RIP 与 EIGRP 那样，RIP 与 EIGRP 在运行时，并不考虑 OSI 模型在二层所定义的内容，即并不关心二层的链路介质类型，而 OSPF 在运行时，必须考虑链路层的类型，称为 OSPF 网络类型（Network Type），对于不同二

层介质类型，OSPF 将有不同的操作和运行过程，网络类型，可分为如下几种：

点到点（Point-To-Point）

点到多点（Point-To-Multipoint ）

广播（Broadcast ）

非广播（Non-Broadcast ）

点到多点非广播（Point-To-Multipoint Non-Broadcast）

对于不同的网络类型，将会影响到 OSPF 的 Hello 时间与 Dead 时间，关系到 DR 与 BDR 的选举与否，影响到 OSPF 邻居是自动建立还是手工建立，总结如下表：

网络类型	Hello 时间	选举 DR/BDR	邻居建立方式
点到点 （Point-To-Point）	10 秒	否	自动
点到多点 （Point-To-Multipoint）	30 秒	否	自动
广播 （Broadcast ）	10 秒	是	自动
非广播 （Non-Broadcast ）	30 秒	是	手工
点到多点非广播 （Point-To-Multipoint Non-Broadcast）	30 秒	否	手工

注：

★OSPF 网络类型（Network Type）是根据二层链路层的介质决定的，但也可以手工定义网络类型，因此可以在各类型之间手工切换。

★OSPF 邻居的成功建立，并不要求双方网络类型一致，但双方网络类型不一致，将可能导致链路状态数据库中的条目无法进入路由表。

OSPF 链路类型（Link Type）

巧玲珑 OSPF 确实因为考虑问题的全面，而导致路由协议的复杂，OSPF 不仅因为不同的二层链路层介质定义了不同的 OSPF 网络类型（Network Type），还因为链路上的邻居，而定义了 OSPF 链路类型（Link Type） 。

OSPF 网络类型（Network Type）是完全根据二层链路层的介质决定的，而 OSPF 链路类型（Link Type）不仅受二层链路层介质的影响，还受到链路中 OSPF 邻居的影响，同时还影响到 LSA，因此变得复杂。

注：

★OSPF 链路类型（Link Type）不会影响人们操作 OSPF，所以可以选择不深入理解 OSPF 链路类型（Link Type），但 OSPF 网络类型（Network Type）却影响到 OSPF 的操作，OSPF 网络类型（Network Type）必须理解和牢记。

★OSPF 链路类型（Link Type）与 OSPF 网络类型（Network Type）没有对应关系，没有因果关系。

OSPF 链路类型（Link Type）分为以下几种：

Stub Network Link

在一个网段中只有一台 OSPF 路由器的情况下，该网段被 OSPF 链路类型定义为 Stub Network Link；因为一个网段中只有一台 OSPF 路由器，所以在这个网段

就不可能有 OSPF 邻居，一个接口被通告进 OSPF，无论其二层链路是什么介质，只要在该接口上没有 OSPF 邻居，那么就是 Stub Network Link；Loopback 接口永远被定义为 Stub Network Link，默认使用 32 位掩码表示，无论将 Loopback 接口改为哪种 OSPF 网络类型（Network Type），始终改变不了它的 OSPF 链路类型（Link Type）属性，但可以改变它在 LSA 中的掩码长度。

Point-To-Point Link

OSPF 网络类型（Network Type）为 Point-To-Point 的接口，OSPF 链路类型（Link Type）为 Point-To-Point Link，但 Loopback 接口除外；而网络类型为点到多点（Point-To-Multipoint）的接口，同样链路类型也为 Point-To-Point Link。

Point-To-Point Link 可以是手工配置的地址（Numbered），也可以是借用的地址（Unnumbered），也可以是物理接口或逻辑子接口。

Transit Link

拥有两台或两台以上 OSPF 路由器的链路，简单理解为有邻居的 OSPF 接口就是 Transit Link，但网络类型为 Point-To-Point 和点到多点（Point-To-Multipoint）的接口除外，因为它们被定义为 Point-To-Point Link。

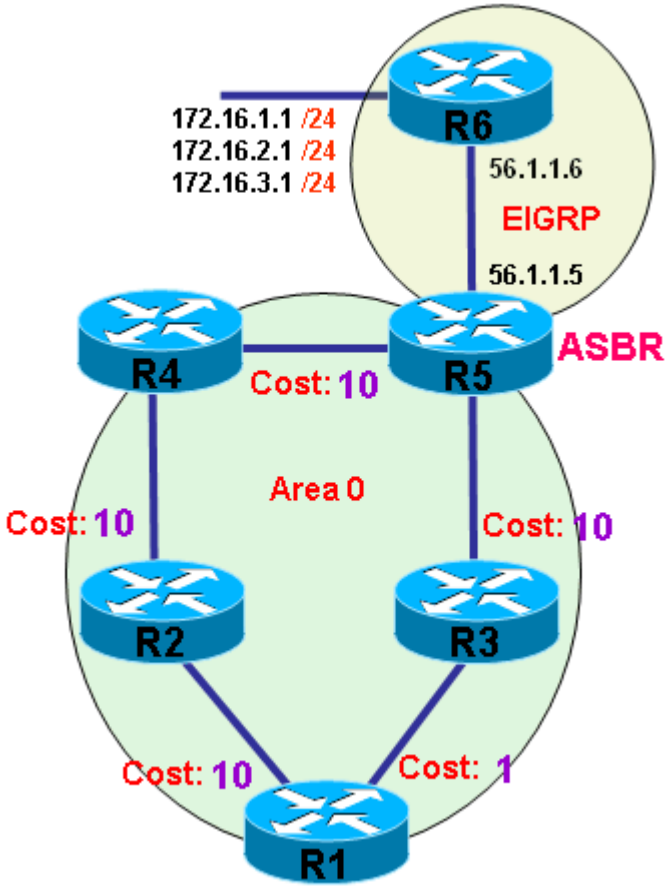
Virtual link

就是 OSPF 虚链路（Virtual Link），但希奇的是，虚链路（Virtual Link）被定义为手工配置的地址（Numbered）的 Point-To-Point Link。

OSPF 外部路由

OSPF 同其它路由协议一样，可以将其它外部协议的路由信息或其它 OSPF 进程的路由信息重分布进自己的域内，这样的路由在 OSPF 域内就是 OSPF 外部路由（External Route），在路由表中的表示方法和 OSPF 自己的路由会有所不同，因为 OSPF 外部路由可以分为两类，分为 Type 2 和 Type 1，所以在路由表中分别表示为 O E2 和 OE 1。

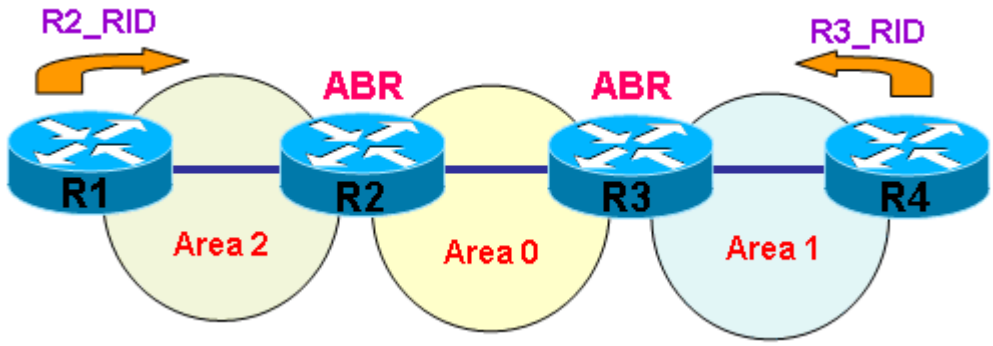
O E2 与 OE 1 在路由条目上没有任何区别，只是对于路由的 Metric 值计算有区别；类型为 O E2 的外部路由，在该路由进入 OSPF 之前的 Metric 值为多少，进入 OSPF 域后，所有 OSPF 路由器看到关于该路由的 Metric 值全部相同，不会再为该路由增加任何 Metric 值，O E2 默认 Metric 值为 20。而 O E1 的路由在 OSPF 路由器上的 Metric 值包含该路由进入 OSPF 域之前的 Metric 值，再加上在 OSPF 域内传递的 Metric 值，也就是到达外部路由的 Metric 值为到达 ASBR 的 Metric 值再加上进入 OSPF 域之前的 Metric 值之和，不同 OSPF 路由器看到 O E1 的路由的 Metric 值可能会有不同。如下图：



在上图中，R5 将 EIGRP 重分布进 OSPF，如果使用 O E2 类型重分布进 OSPF，并且取默认 Metric 值 20，那么 OSPF 域内的路由器 R1，R2，R3，R4，R5 看到外部路由的 Metric 值全部都为 20，不会有任何变化。如果使用 O E1 类型重分布进 OSPF，并且取默认 Metric 值 20，那么 OSPF 域内的路由器在计算 Metric 值时，还会在原有 Metric 值的基础上，再加上到达 ASBR（R5）所需的 Metric 值，假设 R1 选择从 R3 到 R5 再到外部路由，那么 R1 到外部路由的 Metric 值为 20+1+10=31，所以使用 O E2 时，R1 到外部路由的 Metric 值为 20，而在使用 O E1 时，R1 到外部路由的 Metric 值为 31。

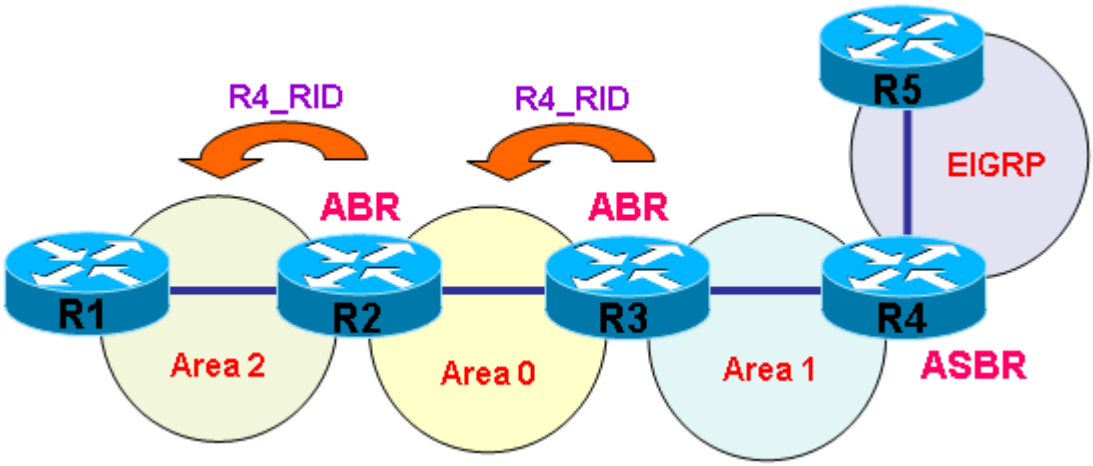
所以必须给每一个 OSPF 路由器定义一个身份，就相当于人的名字，这就是 Router-ID，并且 Router-ID 在网络中绝对不可以有重名，否则路由器收到的链路状态，就无法确定发起者的身份，也就无法通过链路状态信息确定网络位置，OSPF 路由器发出的链路状态都会写上自己的 Router-ID，可以理解为该链路状态的签名，不同路由器产生的链路状态，签名绝不会相同。

之前说过，每一台 OSPF 路由器都有一个 Router-ID，在自己产生 LSA 时，都会在 LSA 中写上自己的 Router-ID，表示 LSA 的身份，类似于签名，如果一台路由器收到一条链路状态，无法到达该 Router-ID 的位置，就无法到达链路状态中的目标网络。其实，在同区域内，每台 OSPF 路由器的 Router-ID 对于每一台路由器都是可达的，因为同区域内会有精确的 LSA 信息，包含 Router-ID，但需要说明，同区域路由器的 Router-ID 并不是单独通过 LSA 来通告的，而是通过精确的 LSA 计算出来的，也就是说 Router-ID 的位置是推算出来的，但您放心，这不会有错，就像之前推算地铁线路图一样；因为同区域路由器的 Router-ID 都有精确的路径信息，所以网络是通畅的，然而，不同区域的 OSPF 路由器，Router-ID 是不知道的，也可以理解为 Router-ID 不会跨区域传递，那么是否就意味着不同区域的路由是不可达的呢？答案当然是可达的，只要同区域所有路由器可达，那么不同区域自然就是可达的，理由是，不同区域是通过 ABR 相连的，因为 ABR 连接着不同区域，所以只要各个区域路由器和 ABR 是通的，那么不同区域当然可以实现网络连通，所以，ABR 在将一个区域的 LSA 转发至另一个区域时，产生该 LSA 的 Router-ID 就不再是原来的 Router-ID，会被修改为 ABR 的 Router-ID，这样一来，和 ABR 相通的路由器只要到达 ABR，就能到达其它区域；如下图所示：



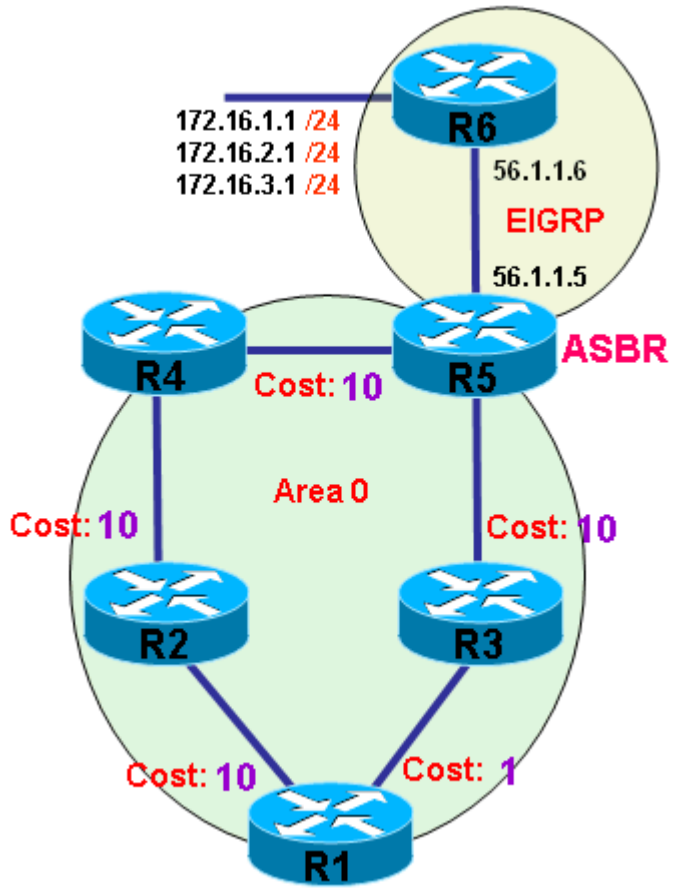
上图中，R2 和 R3 是 ABR，Area 0 一定拥有全网的 LSA，R2 可以在 Area 0 与 Area 2 之间转发 LSA，而 R3 可以在 Area 0 与 Area 1 之间转发 LSA，但是对于其它区域的路由，R1 无法到达，因为 R1 只有 Area 2 每台路由器的 Router-ID，R4 的情况同样如此；最终结果是，R2 在将 Area 0 的 LSA 转发给 Area 2 时，已经将所有 LSA 的 Router-ID 改成了自己的，所以所有的路由对于 R1 来说，只要交给 ABR（R2）即可，因为 R2 能够全网可达，在 Area 1 也是一样，R3 在将 Area 0 的 LSA 转发给 Area 1 时，已经将所有 LSA 的 Router-ID 改成了自己的，所以所有的路由对于 R4 来说，只要交给 ABR（R3）即可，由此可见，OSPF 中，不同区域之间的路由互通，是由于 ABR 在转发区域是的 LSA 时，将 LSA 的 Router-ID 改成了自己的 Router-ID，才使得该区域可以与其它区域通信，如果没有 ABR 这种自动修改 Router-ID 的行为，那么 OSPF 不同区域间将会无法通信。

对于外部路由，执行重分布的路由器同样需要在 LSA 中写上自己的 Router-ID，其实就是 ASBR 的 Router-ID，因为外部路由会在多个 OSPF 区域之间传递，所以会被多个 ABR 转发，而 ABR 在转发外部路由的 LSA 时，是没有权限修改 LSA 的 Router-ID，这样一来，外部路由的 Router-ID 在所有 OSPF 路由器上都不会改变，永远是 ASBR 的 Router-ID，最终造成的结果是只有与 ASBR 同在一个区域的路由器才能到达外部路由，因为只有与 ASBR 同在一个区域的路由器才知道如何到达 ASBR 的 Router-ID，而其它区域的路由器对此却无能为力；为了能够让 OSPF 所有区域都能与外部路由连通，在 ABR 将外部路由从 ASBR 所在的区域转发至其它区域时，需要发送单独的 LSA 来告知如何到达 ASBR 的 Router-ID，因为 ABR 将外部路由的 LSA 告诉了其它区域，是有义务让它们与外部路由可达的，所以额外发送了单独的 LSA 来告知如何到达 ASBR 的 Router-ID；从这里也可以看出，任何一个 ASBR 所在区域外的其它区域，都必须靠 ABR 通告一条通往 ASBR 的 Router-ID 的 LSA，此 LSA 就是后面将会详细解释的 LSA 类型的第 4 类。如下图：



在上图中，因为 ASBR（R4）将外部路由（EIGRP）重分布进 OSPF 后，自己的 Router-ID 对于 Area 1 的所有路由器是可达的，但对于 Area 2 和 Area 0 中的路由器是不可达的，所以在 ABR 将外部路由的 LSA 发向 Area 0 和 Area 2 时，会额外通过单独的 LSA 4 将 ASBR（R4）的 Router-ID 发向这些区域。

OSPF 外部路由有许多是需要理解的地方，外部 LSA 的 Forward Address 是一个其它 LSA 没有的特征，每一条外部 LSA 都带有一个 Forward Address，该地址是用来告诉收到此 LSA 的路由器如何到达外部路由，那么一条外部路由的 LSA，Forward Address 究竟该是什么地址呢？在 ASBR 的路由表中，外部路由的下一跳地址是什么，那么在外部路由的 LSA 中 Forward Address 就是什么，但是 OSPF 内部路由器是通过 Forward Address 来到达外部路由的，如果它们连这个 Forward Address 都到达不了，比如 Forward Address 本身就是外部路由而不包含在 OSPF 进程中，那么可想而知，Forward Address 的意义等于 0，所以，如果 ASBR 的路由表中，外部路由的下一跳地址是 OSPF 进程自己的路由，那么外部路由 LSA 的 Forward Address 就是该地址，所有 OSPF 内部路由器则通过该地址去往外部路由，但如果 ASBR 的路由表中，外部路由的下一跳地址不在 OSPF 进程中，那么该地址对于其它所有 OSPF 路由器都不可达，这时，ASBR 就将外部路由 LSA 的 Forward Address 设置为 0.0.0.0，当 Forward Address 为 0.0.0.0 时，所有 OSPF 路由器通过外部 LSA 的 Router-ID 去往外部路由；如下图所示：



在上图中，当 R5（ASBR）将外部路由 EIGRP 重分布进 OSPF 时，因为 R5 到达外部路由的下一跳地址是 56.1.1.6，如果该地址的网段被通告进 OSPF 进程，那么此时外部路由的 Forward Address 就是 56.1.1.6，所有 OSPF 路由器通过去往地址 56.1.1.6 来去往外部路由；但如果该地址并不在 OSPF 进程中，那么外部路由的 Forward Address 就是 0.0.0.0，则所有 OSPF 路由器通过去往 ASBR（R5）的 Router-ID 去往外部路由。

重点提示：

重分布外部路由时，默认类型为 O E2，如果通过两个 ASBR 能到达相同的外部路由，选择 O E1 的优先，其次是 O E2，但如果都为 O E1 或 O E2，则选择到达 Forward Address 最小 Metric 的路径优先，如果 Forward Address 都为 0.0.0.0，最后选择到达 ASBR 最小 Metric 的路径优先，但如果 Forward Address 地址一个为 0.0.0.0，一个为真实地址，统一比较到 ASBR 的 Metric。

OSPF 末节区域

如果路由增加，就意味着 LSA 的增加，有时，在一个末梢网络中，许多路由信息是多余的，并不需要通告进来，因为一个 OSPF 区域内的所有路由器都能够通过该区域的 ABR 去往其它 OSPF 区域或者 OSPF 以外的外部网络，既然一个区域的路由器只要知道去往 ABR，就能去往区域外的网络，所以可以过滤掉区域外的路由进入某个区域，这样的区域称为 OSPF 末节区域（Stub Area）；一个末节区域的所有路由器虽然可以从 ABR 去往区域外的网络，但路由器上还是得有指向 ABR 的路由，所以末节区域的路由器只需要有默认路由，而不需要明细路由，即可与区域外的网络通信，根据末节区域过滤掉区域外的不同路由，可将末节区域分为如下四类：

Stub Area（末节区域）

Totally Stub Area（完全末节区域）

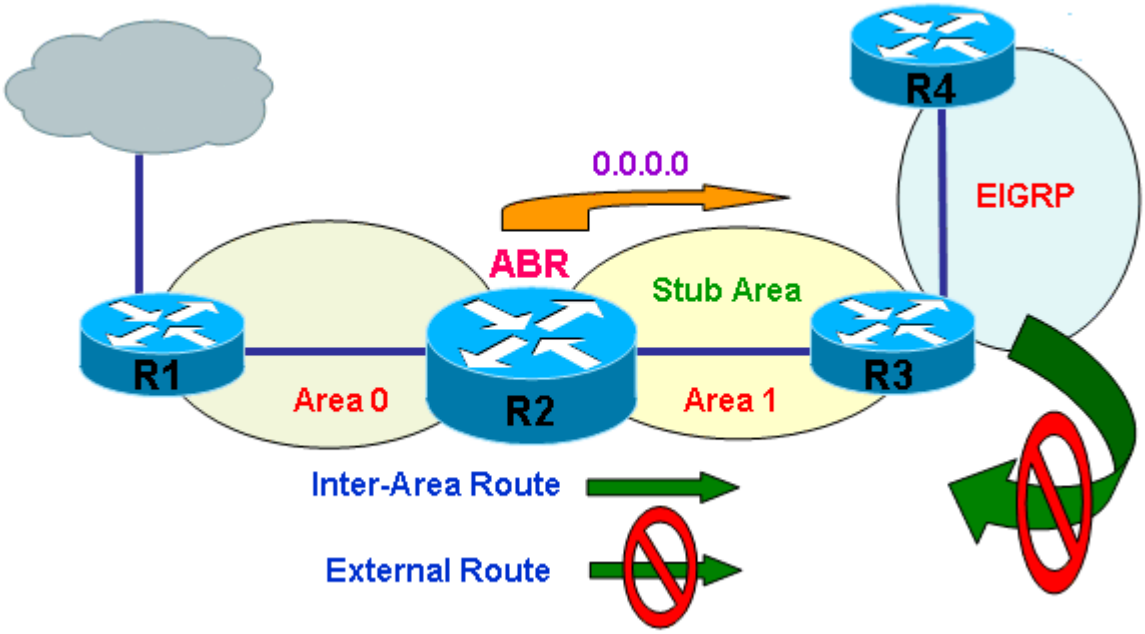
Not-so-Stubby Area（NSSA）

Totally Not-so-Stubby Area（Totally NSSA）

各类型的特征如下：

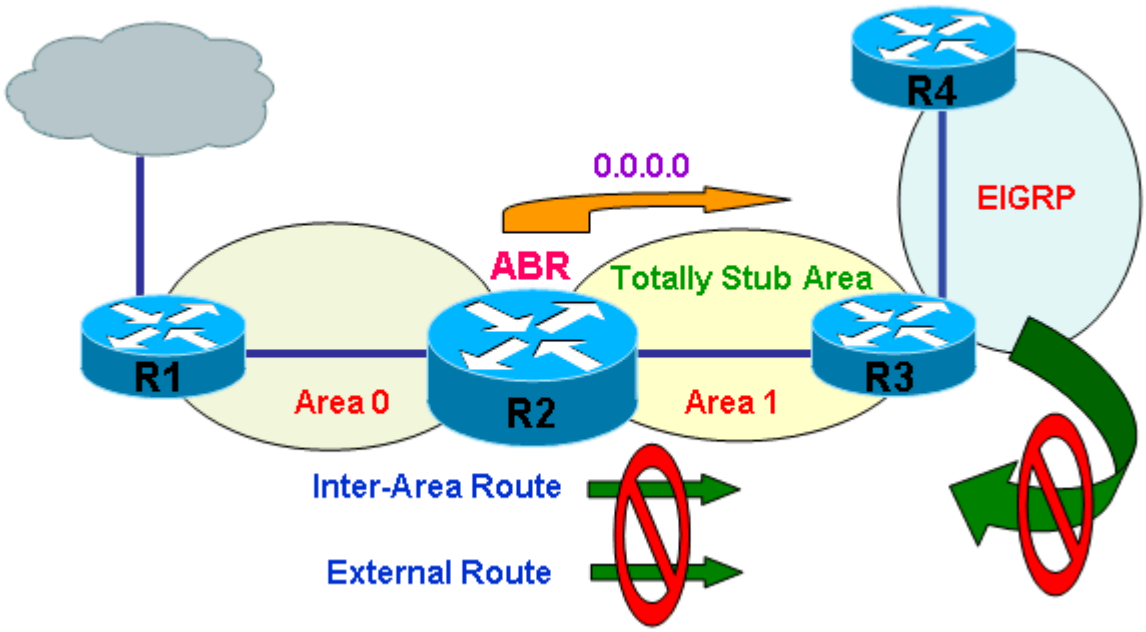
Stub Area（末节区域）

在 Stub Area（末节区域）下，ABR 将过滤掉所有外部路由进入末节区域，同时，末节区域内的路由器也不可以将外部路由重分布进 OSPF 进程，即末节区域内的路由器不可以成为 ASBR，但其它 OSPF 区域的路由（Inter-Area Route）可以进入末节区域，由于没有去往外部网络的路由，所以 ABR 会自动向末节区域内发送一条指向自己的默认路由，如下图：



Totally Stub Area（完全末节区域）

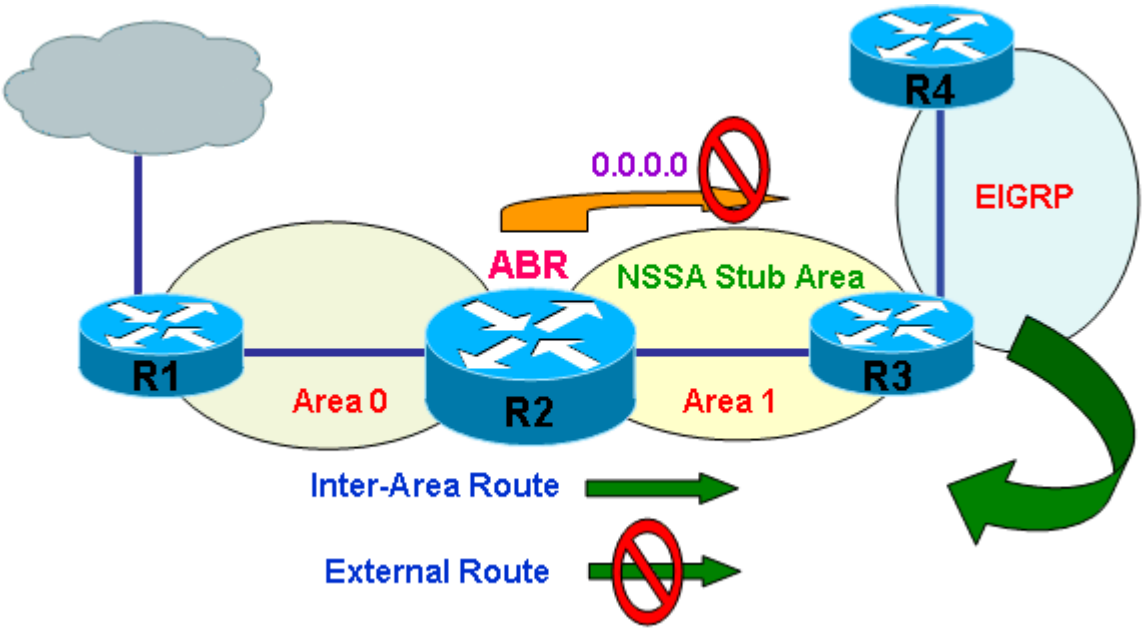
在 Totally Stub Area（完全末节区域）下，ABR 将过滤掉所有外部路由和其它 OSPF 区域的路由（Inter-Area Route）进入完全末节区域，同时，末节区域内的路由器也不可以将外部路由重分布进 OSPF 进程，即完全末节区域内的路由器不可以成为 ASBR，由于没有去往外部网络的路由，所以 ABR 会自动向完全末节区域内发送一条指向自己的默认路由，如下图：



可以发现，末节区域与完全末节区域的不同之处在于，末节区域可以允许其它 OSPF 区域的路由（Inter-Area Route）进入，而完全末节区域却不可以。

Not-so-Stubby Area（NSSA）

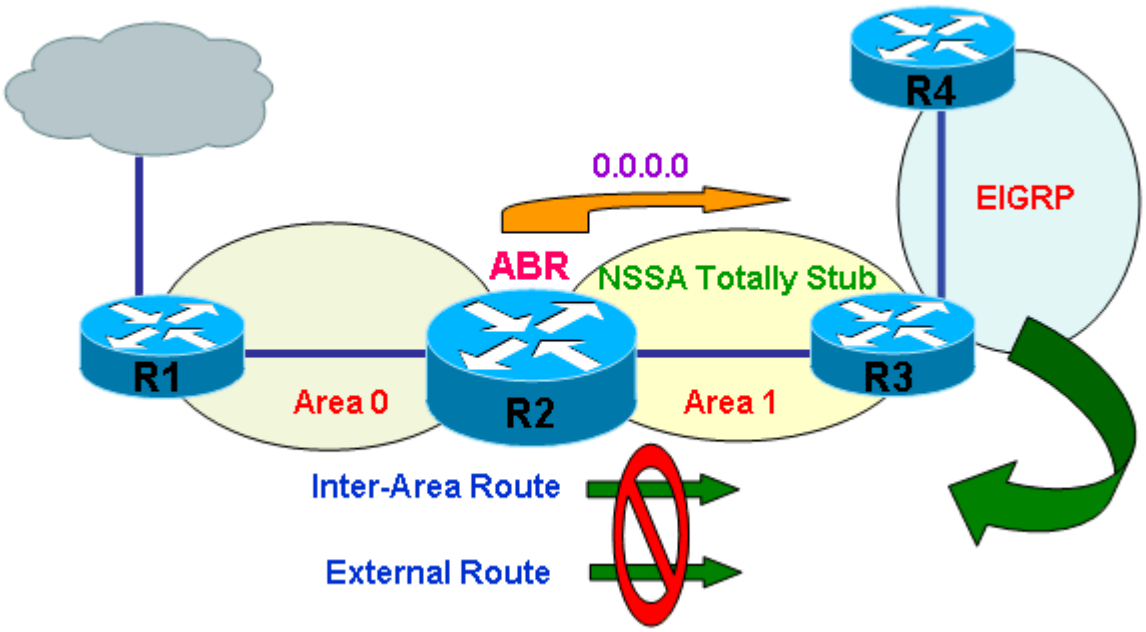
在 Not-so-Stubby Area（NSSA）下，ABR 将过滤掉所有外部路由进入末节区域，同时也允许其它 OSPF 区域的路由（Inter-Area Route）进入 NSSA 区域，并且路由器还可以将外部路由重分布进 OSPF 进程，即 NSSA 区域内的路由器可以成为 ASBR，由于自身可以将外部网络的路由重分布进 OSPF 进程，所以 ABR 不会自动向 NSSA 区域内发送一条指向自己的默认路由，但可以手工向 NSSA 域内发送默认路由，并且只可在 ABR 上发送默认路由；如下图：



NSSA 与末节区域的最大区别在于，NSSA 区域可以允许自身将外部路由重分布进 OSPF，而末节区域则不可以。

Totally Not-so-Stubby Area（Totally NSSA）

在 Totally Not-so-Stubby Area（Totally NSSA）下，ABR 将过滤掉所有外部路由和其它 OSPF 区域的路由（Inter-Area Route）进入 Totally NSSA 区域，但路由器可以将外部路由重分布进 OSPF 进程，即 Totally NSSA 区域内的路由器可以成为 ASBR，由于没有去往其它 OSPF 区域的路由，所以 ABR 会自动向 Totally NSSA 内发送一条指向自己的默认路由，如下图：



Totally NSSA 与 NSSA 的区别在于，NSSA 区域可以允许其它 OSPF 区域的路由（Inter-Area Route）进入，而 Totally NSSA 区域却不可以，但 Totally NSSA 区域的 ABR 会自动向 Totally NSSA 区域内发送一条指向自己的默认路由。

总结各区域的特征如下：

区域类型	接收区域间路由	ABR 是否自动发送默认路由	是否可以重分布外部路由
Stub Area （末节区域）	是	是	否
Totally Stub Area （完全末节区域）	否	是	否
Not-so-Stubby Area （NSSA）	是	否	是
Totally Not-so-Stubby Area （Totally NSSA）	否	是	是

注：

★在末节区域下，ABR 自动发出的默认路由，Metric 值默认为 1，可通过命令 area area-id default-cost cost 修改，默认路由除了默认的 Cost 值以外，还会累加真实接口的 Cost 值。

★骨干区域不能配置为任何末节区域。

★当将某个区域配置为末节区域后，则区域中所有路由器都必须配置为末节区域，因为配置为末节区域的路由器上所有接口发出的 Hello 包中都会有末节标签，所有如果对方没有末节标签，则不能成为邻居。

OSPF LSA 类型

OSPF 由于有着多种区域类型，多种网络类型，多种链路类型，多种路由器身份，所以 LSA（Link-State Advertisements）也是多样的，并不用埋怨设计 OSPF 的人为什么要将 OSPF 设计的这么复杂，因为这种工作，通常是一群人，合伙只干一件事，工作分担到他们每个人身上，并没有多少，复杂程度他们自然不会感觉到，但他们多个人的复杂结果，却要分到一个人身上。

在详细讲解 LSA 之前，需要重点说明，只有同一个区域内的 LSA，才是精确的，区域外的 LSA，并不一定包含所有必备的信息，因此，所有 LSA 知识信息，并不一定可以套用到每一类 LSA。

目前得到的消息为止，OSPF 中共有 11 类 LSA，而在 CCIE 的要求中，只需要理解 1、2、3、4、5、7 共 6 类即可，这些 LSA 会因为区域类型，网络类型，链路类型，路由器身份的不同而不同，以下是详细介绍：

LSA 各类的用途为：

类型 1 （Router Link）

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

类型 1 的 LSA 是任何一台 OSPF 路由器都会产生的，每一台 OSPF 路由器的每一个 OSPF 接口都会有自己的链路状态，但是每台 OSPF 路由器只能产生一条类型 1 的 LSA，即使有多个 OSPF 接口，也只有一条类型 1 的 LSA，因为所有 OSPF 接口的链路状态是被打包成一条类型 1 的 LSA 发送的。

一个区域正是由于 LSA 1 的存在，才有精确的路由表，一个区域如果只有 LSA 1，同样可以正常通信。LSA 1 只能在单个区域内传递，ABR 不能将 LSA 1 转到发另外一个区域，并且没有任何权利修改 LSA 1。

类型 2 （Network Link）

类型 2 的 LSA 只有在需要选举 DR/BDR 的网络类型中才会产生，并且只是 DR 产生，BDR 没有权利产生，LSA 2 与 LSA 1 没有任何关联，没有任何依存关系，是想互独立的。

类型 3 （Summary Link）

类型 3 的 LSA 就是将一个区域的 LSA 发向另一个区域时的汇总和简化，ABR 其实就是将 LSA 1 汇总和简化，变成 LSA 3 后再发到另一个区域的，如果是详细完整的 LSA 1，是绝不允许的，LSA 3 是 LSA 1 的缩略版。

类型 4 （ASBR Summary Link）

对于外部路由，执行重分布的路由器 ASBR 在 LSA 中写上自己的 Router-ID，然后传递到多个 OSPF 区域，所以会被多个 ABR 转发，而 ABR 在转发外部路由的 LSA 时，是没有权限修改 LSA 的 Router-ID，这样一来，外部路由的 Router-ID 在所有 OSPF 路由器上都不会改变，永远是 ASBR 的 Router-ID，最终造成的结果是只有与 ASBR 同在一个区域的路由器才能到达外部路由，因为只有与 ASBR 同在一个区域的路由器才知道如何到达 ASBR 的 Router-ID，而其它区域的路由器对此却无能为力；为了能够让 OSPF 所有区域都能与外部路由连通，在 ABR 将外部路由从 ASBR 所在的区域转发至其它区域时，需要发送单独的 LSA 来告知如何到达 ASBR 的 Router-ID，因为 ABR 将外部路由的 LSA 告诉了其它区域，是有义务让它们与外部路由可达的，所以额外发送了单独的 LSA 来告知如何到达 ASBR 的 Router-ID；这个单独的 LSA 就是类型 4 的 LSA，LSA 4 是包含的 ASBR 的 Router-ID，只要不是 ASBR 所在的区域，都需要 ABR 发送 LSA 4 来告知如何去往

ASBR。

类型 5 （External Link）

类型 5 的 LSA 就是外部路由重分布进 OSPF 时产生的，并且是由 ASBR 产生的，LSA 中包含 ASBR 的 Router-ID，任何路由器都不允许更改该 Router-ID，LSA 5 中还包含 Forward Address，对于 LSA 5 的 Metric 值计算与选路规则也有所不同，详细信息请见 OSPF 外部路由部分。

类型 7 （NSSA Link）

因为 NSSA 区域可以将外部路由重分布进 OSPF 进程，而 NSSA 不是一般的常规区域，所以在 NSSA 将外部路由重分布进 OSPF 时，路由信息使用类型 7 来表示，LSA 7 由 NSSA 区域的 ASBR 产生，LSA 7 也只能在 NSSA 区域内传递，如果要传递到 NSSA 之外的其它区域，需要同时连接 NSSA 与其它区域的 ABR 将 LSA 7 转变成 LSA 5 后再转发。

LSA 各参数

在 LSA 的内容中，将有多个参数来表示，这些参数会因为 LSA 类型，区域类型，网络类型，链路类型，路由器身份的不同而不同，是真正的变化多端，非常的复杂，这些参数在我们操作 OSPF 时，可以帮助我们更好的分析 LSA，但并不会起决定性的作用，所以对 LSA 参数的理解与否，不会影响到 OSPF 的配置与排错，若无特殊要求，LSA 参数需要大家了解即可，不需要掌握，不需要牢记。

LSA 中包含的参数有 **LS Type，Link State ID，Link ID，Link Data**，具体如下：

LS Type

LS Type 就是前面讲到的 LSA 类型，如 LSA 1，LSA 2，LSA 3，LSA 4，LSA 5，LSA 7。

Link State ID

因为 OSPF 接口的链路状态，是使用 LSA 发送的，接口的相关信息，如网络号，掩码等等，它们算是 LSA 真正的内容，而 LSA 也是有简明信息的，或者说是 LSA 的标题，或者说是 LSA 的名称，这就是 Link State ID，如果将 LSA 比作一个包裹，那么 Link State ID 就是包裹外面写的信息，比如包裹里是一件衣服，那么 Link State ID 可能就是写的衣服是什么牌子，什么尺寸，什么颜色，等等；但不同类型的 LSA，其 Link State ID 的表示也会不同，如下表：

LSA 类型	Link State ID 内容
LSA 1	是产生 LSA 1 的路由器的 Router-ID。
LSA 2	因为 LSA 2 是由 DR 产生的，所以 LSA 2 的 Link State ID 是 DR 的接口地址。
LSA 3	是目标网络的网络地址，其实这个等同于路由条目，也就是路由表里显示的是什么，LSA 3 的 Link State ID 就是什么。
LSA 4	是 ASBR 的 Router-ID
LSA 5	和 LSA 3 的一样，还是目标网络的网络地址，路由表里显示的是什么，LSA 5 的 Link State ID 就是什么。

Link ID

Link ID 是用来表示链路自己的，也就是表示 OSPF 接口自己，再换句话说，就是 OSPF 接口的链路状态，可以理解为 LSA 的内容部分

Link ID 会因为链路类型（Link Type）的不同而不同，对于链路类型（Link Type）的详细解释，请参见前面部分。

需要注意，Link ID 的内容有时与 Link State ID 的内容相同，但 Link ID 并非完全等于 Link State ID，因为 Link State ID 是由 LSA 类型的不同而不同，而 Link ID 是由 Link Type 的不同而不同。

Link ID 的具体内容如下表：

链路类型（Link Type）	Link ID 内容
Stub Network Link	使用接口的网络号和子网掩码来表示。
Point-To-Point Link	邻居的 Router-ID。
Transit Link	是 DR 的接口地址。
Virtual link	同 Point-To-Point Link，是邻居的 Router-ID。

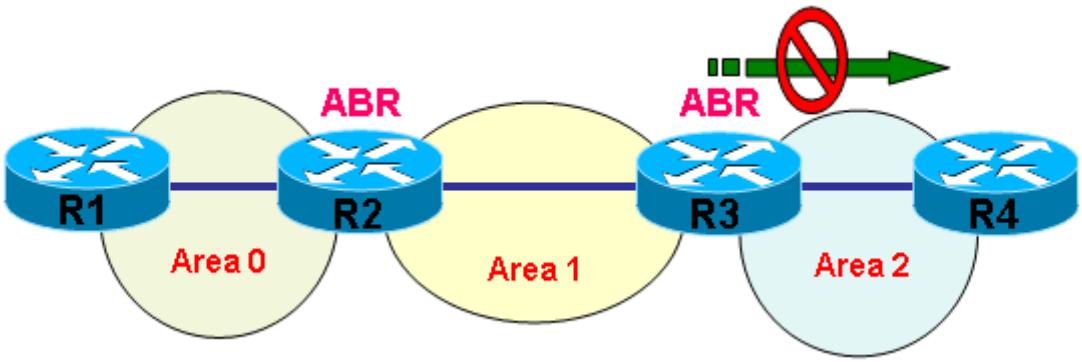
Link Data

是接口上的 IP 地址，如果链路类型（Link Type）为 Stub Network Link，则 Link Data 是子网掩码。

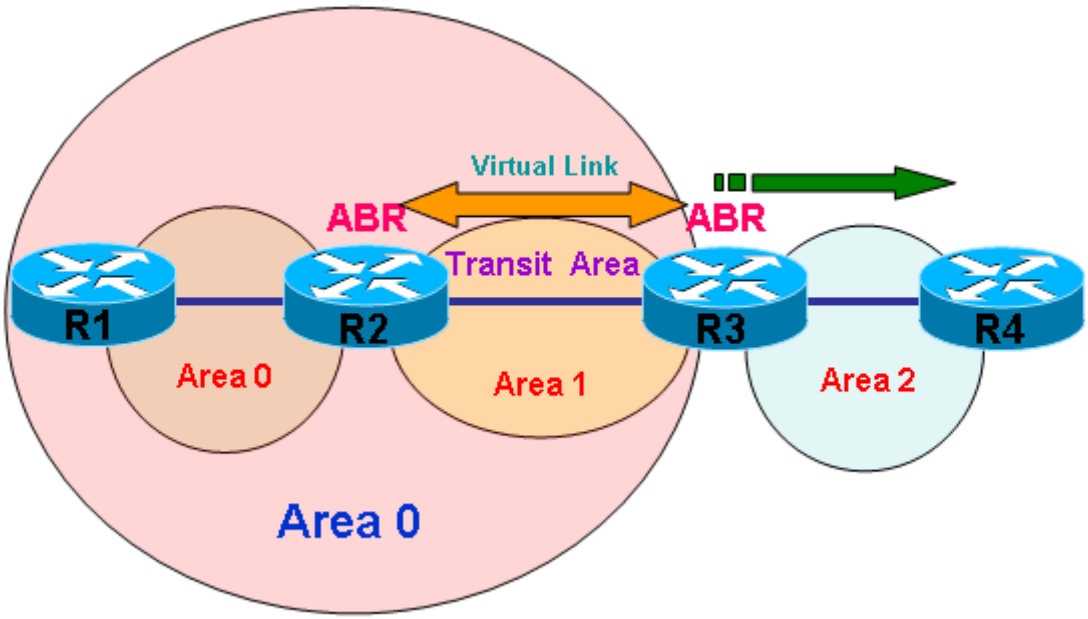
OSPF 虚链路（Virtual Link）

因为 OSPF 采用了区域化的设计，并且区域也采用了 Hub-Spoke 的架构，所有区域中定义出一个核心，然后其它部分都与核心相连，OSPF 的区域 0 就是所有区域的核心，称为 BackBone 区域（骨干区域），而其它 Normal 区域（常规区域）应该直接和骨干区域相连，常规区域只能和骨干区域交换 LSA，常规区域与常规区域之间即使直连也无法互换 LSA，但在某些情况下，某些常规区域无法与骨干区域直连，这时便无法得到其它区域的路由，因此，设计了将骨干区域的范围通过虚拟的方法进行扩展到相邻常规区域的位置，因而让不能直接与骨干区域相连的区域，最终可以与骨干区域直连，这种对骨干虚拟的扩展和拉伸就是 OSPF 虚链路（Virtual Link）能实现的；因为某些常规区域不能与骨干区域直连而只能与其它常规区域直连，所以 OSPF 虚链路（Virtual Link）通过将相邻的常规区域虚拟为骨干区域，从而让那些不能与骨干区域直连的常规区域也能获得其它 OSPF 区域的路由。与骨干区域相邻的常规区域被扩展后，该区域被称为 Transit Area，理论上 Transit Area

不应该为末节区域；在扩展后，原本为常规区域的 Transit Area，将变成骨干区域，所以路由将从 Inter-Area Route 转变为 Intra-Area Route，路由表示形式也将从 O IA 改变为 O 的形式；在进行 OSPF 虚链路扩展时，是将 Transit Area 中与骨干区域直连的 ABR 和连接另一个常规区域的 ABR 相连，连接这两个 ABR 时，使用双方的 Router-ID 来连接。如下图：

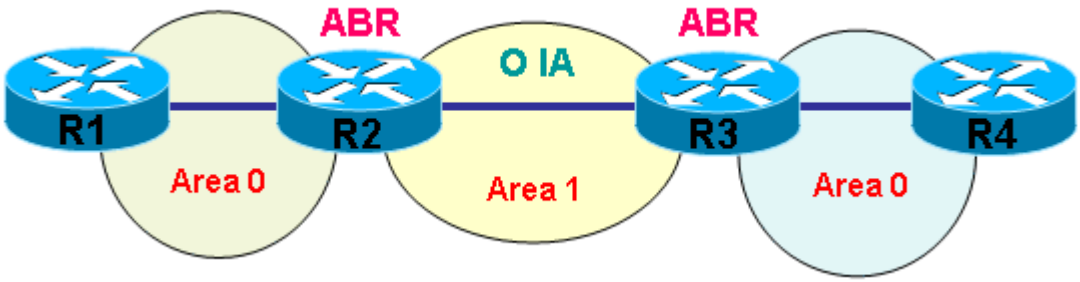


在上图中，区域 2 只能与区域 1 直连，而无法与骨干区域直连，在这种情况下，由于常规区域与常规区域之间即使直连也无法互换 LSA，所以 R3 虽然是 ABR，但因为没有连接骨干区域，最后不可能将任何区域的 LSA 发进区域 2，最终导致区域 2 无法外其它区域通信，在这种情况下，需要通过 OSPF 虚链路将骨干区域的范围扩展到相邻的区域 1，如下图：

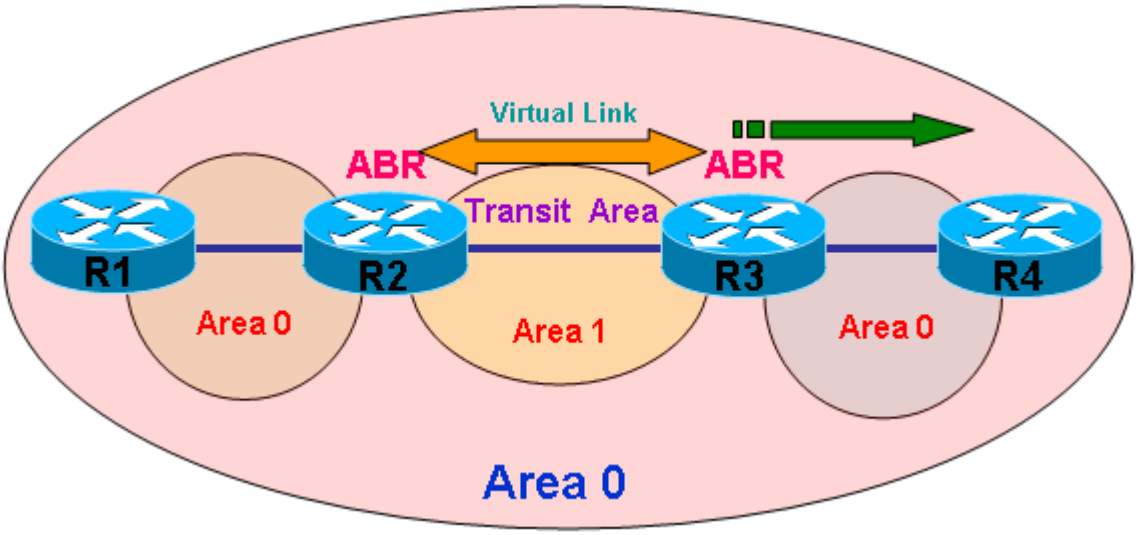


在进行 OSPF 虚链路扩展后，区域 1 被虚拟成了骨干区域，而这时的 R3 等同于连接骨干区域和区域 2 的 ABR，所以可以将自己所有的 LSA 发进区域 2。在扩展 OSPF 虚链路时，是通过连接 R2（ABR）与 R3（ABR）的 Router-ID 来建立的。

因为 OSPF 虚链路（Virtual Link）能将骨干区域扩展到相邻的常规区域，从而将常规区域虚拟为骨干区域，所以在某些情况下，如公司合并，或者为了备份骨干区域，可能出现骨干区域被常规区域所隔离，如下图：



图中骨干区域被区域 1 分割为两部分，将使得骨干区域自己的路由无法相互传递，在使用 OSPF 虚链路后，可以将区域 1 也扩展为骨干区域，如下图：



在经过 OSPF 虚链路将区域 1 也扩展为骨干区域后，可见所有的区域都变成了骨干区域，从而使网络中骨干区域能够收到另外一边被分割的骨干区域的路由，并且最后骨干区域自己的路由都为 Intra-Area Route，但 Area 1 与骨干区域的路由将仍然为 O IA。

注：

★OSPF 虚链路必须是在两个拥有共同区域的 ABR 之间建立的，其中必须至少有一个 ABR 是连接骨干的。

★OSPF 虚链路被认为是骨干区域的一个接口，一条链路，也需要建立 OSPF 邻居，但在邻居建立之后，链路上是没有 Hello 包传递的。

OSPF 认证

同 RIP 和 EIGRP 一样，出于安全考虑，OSPF 也使用了认证，OSPF 同时支持明文和 MD5 认证，在启用 OSPF 认证后，Hello 包中将携带密码，双方 Hello 包中的密码必须相同，才能建立 OSPF 邻居关系，需要注意，空密码也是密码的一种。

当 OSPF 邻居的一方在接口上启用认证后，从该接口发出的 Hello 包中就会带有密码，双方的 Hello 包中拥有相同的密码时，邻居方可建立；一台 OSPF 路由器可能有多个 OSPF 接口，也可能多个接口在多个 OSPF 区域，只要在接口上输入 OSPF 认证的命令后，便表示开启了 OSPF 认证，可以在每个接口上一个一个启用，也可以一次性开启多个接口的认证，如果需要开启多个接口的认证功能，那么认证的命令就并非直接在接口上输入，而是到 OSPF 进程模式下输入，并且是对某个区域全局开启的，当在进程下对某个区域开启 OSPF 认证后，就表示在属于该区域的所有接口上开启了认证。所以，在进程下对区域配置认证，是快速配置多个接口认证的方法，与在多个接口上一个一个开启，没有本质区别。因为 OSPF 虚链路被认为是骨干区域的一个接口，一条链路，所以在 OSPF 进程下对骨干区域开启认证后，不仅表示开启了区域 0 下所有接口的认证，同时也开启了 OSPF 虚链路的认证，但 OSPF 虚链路在建立后，并没有 Hello 包的传递，所以认证在没有重置 OSPF 进程的情况下，是不会生效的。

OSPF 汇总路由

在 OSPF 同区域内，LSA 是绝对不允许以任何形式或任何手段更改的，但在一个区域与另一个区域之间，LSA 可以被 ABR 修改后传递，从而得知，在同一个区域内，OSPF 路由是不能被汇总的，而是当路由从一个区域被 ABR 转发到另外一个区域时，就可以执行路由汇总，并且汇总必须是在 ABR 上执行的，但该汇总不对 OSPF 外部路由生效；在将外部路由重分布进 OSPF 时，也可以执行路由汇总，此时的汇总必须在 ASBR 上配置。

为了防止路由黑洞，需要在执行路由汇总的路由器上将汇总由指向空接口（null0），在 IOS 12.1(6)以后的版本，配置汇总后会自动产生指向空接口的路由，但在 IOS 12.1(6)及以前的版本需要手工创建。

注：OSPF RFC (1583)并没有规定一个区域适合多少台路由器，一个网段多少个邻居，或如何布署网络。

配置 OSPF 实验

说明：实验配置共包含：

OSPF 邻居

OSPF LSA

OSPF 虚链路

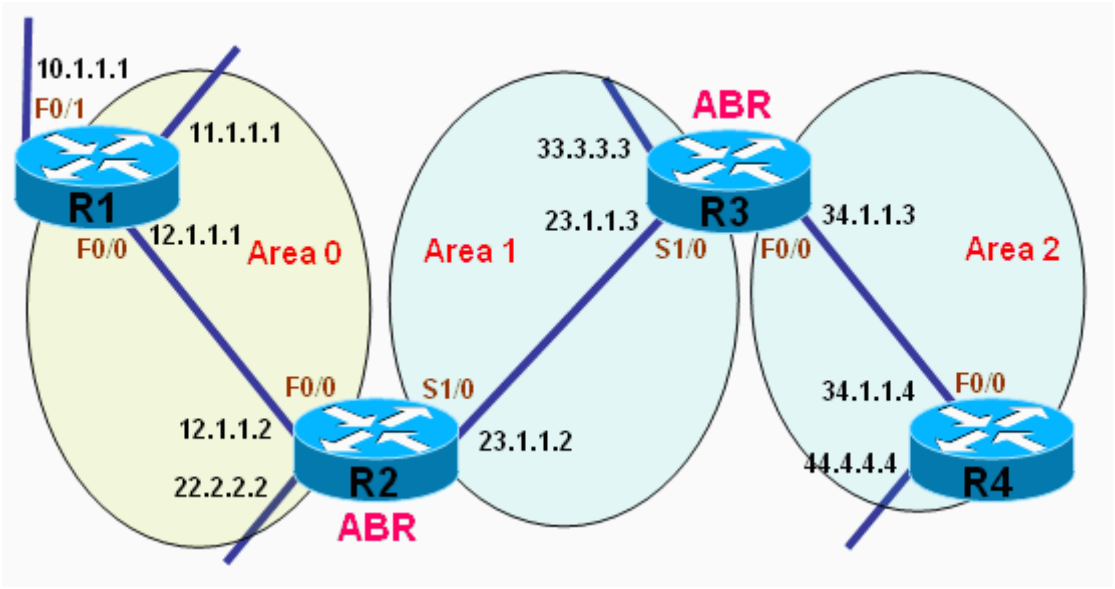
OSPF 认证

OSPF 外部路由

OSPF 路由汇总

OSPF 末节区域

以下图为例，配置 OSPF 邻居，OSPF LSA，OSPF 虚链路，OSPF 认证：



1.配置基础网络环境

（1）配置 R1：

r1(config)#int f0/0

r1(config-if)#ip address 12.1.1.1 255.255.255.0

r1(config-if)#no sh

```
r1(config)#int f0/1

r1(config-if)#ip address 10.1.1.1 255.255.255.0

r1(config-if)#no sh


r1(config)#int loopback 11

r1(config-if)#ip address 11.1.1.1 255.255.255.0


r1(config)#router ospf 1

r1(config-router)#router-id 1.1.1.1

r1(config-router)#network 12.1.1.1 0.0.0.0 area 0

r1(config-router)#network 11.1.1.1 0.0.0.0 area 0

r1(config-router)#network 10.1.1.1 0.0.0.0 area 0
```

说明：在 R1 上配置 10.1.1.0/24，11.1.1.0/24，12.1.1.0/24，并放入 OSPF 进程。

（2）配置 R2：

```
r2(config)#int f0/0

r2(config-if)#ip address 12.1.1.2 255.255.255.0

r2(config-if)#no shutdown


r2(config)#int s1/0

r2(config-if)#en frame-relay
```

r2(config-if)#no fram inverse-arp .

r2(config-if)#no arp frame-relay

r2(config-if)#no ip address

r2(config-if)#no shutdown

r2(config-if)#exit

r2(config)#int s1/0.23 point-to-point

r2(config-subif)#ip address 23.1.1.2 255.255.255.0

r2(config-subif)#frame-relay interface-dlci 203

r2(config)#int loopback 22

r2(config-if)#ip address 22.2.2.2 255.255.255.0

r2(config)#router ospf 1

r2(config-router)#network 12.1.1.2 0.0.0.0 area 0

r2(config-router)#network 22.2.2.2 0.0.0.0 area 0

r2(config-router)#network 23.1.1.2 0.0.0.0 area 1

说明：在 R2 上配置 12.1.1.0/24，23.1.1.0/24，22.2.2.0/24，并放入 OSPF 进程。

（3）配置 R3：

```
r3(config)#int s1/0
r3(config-if)#encapsulation frame-relay
r3(config-if)#no frame-relay inverse-arp
r3(config-if)#no arp frame-relay
r3(config-if)#no ip address
r3(config-if)#no shutdown

r3(config)#int s1/0.23 point-to-point
r3(config-subif)#ip address 23.1.1.3 255.255.255.0
r3(config-subif)#frame-relay interface-dlci 302

r3(config)#int f0/0
r3(config-if)#ip add 34.1.1.3 255.255.255.0

r3(config)#int loopback 33
r3(config-if)#ip add 33.3.3.3 255.255.255.0

r3(config)#router ospf 1
r3(config-router)#router-id 3.3.3.3
r3(config-router)#network 23.1.1.3 0.0.0.0 area 1
r3(config-router)#network 33.3.3.3 0.0.0.0 area 1
```

r3(config-router)#network 34.1.1.3 0.0.0.0 area 2

说明：在 R3 上配置 23.1.1.0/24，34.1.1.0/24，33.3.3.0/24，并放入 OSPF 进程。

（4）配置 R4：

r4(config)#int f0/0

r4(config-if)#ip address 34.1.1.4 255.255.255.0

r4(config-if)#no sh

r4(config)#int loopback 44

r4(config-if)#ip address 44.4.4.4 255.255.255.0

r4(config)#router ospf 1

r4(config-router)#network 0.0.0.0 255.255.255.255 area 2

说明：在 R4 上配置 34.1.1.0/24，44.4.4.0/24，并放入 OSPF 进程。

2.测试 OSPF 邻居

（1）查看 R1 的 Router-ID：

r1#sh ip protocols

Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 1.1.1.1

Number of areas in this router is 1. 1 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

10.1.1.1 0.0.0.0 area 0

11.1.1.1 0.0.0.0 area 0

12.1.1.1 0.0.0.0 area 0

Reference bandwidth unit is 100 mbps

Routing Information Sources:

Gateway Distance Last Update

Distance: (default is 110)

r1#

说明：可以看见 R1 手工配置的 Router-ID 为 1.1.1.1。

（2）查看 R2 的 Router-ID：

r2#sh ip protocols

Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 22.2.2.2

It is an area border router

Number of areas in this router is 2. 2 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

12.1.1.2 0.0.0.0 area 0

22.2.2.2 0.0.0.0 area 0

23.1.1.2 0.0.0.0 area 1

Reference bandwidth unit is 100 mbps

Routing Information Sources:

Gateway Distance Last Update

Distance: (default is 110)

r2#

说明：R2 在没有手工配置 Router-ID 的情况下，自动选择了最高 Loopback 地址 22.2.2.2 为 Router-ID。

（3）查看 R1 的 OSPF 邻居：

r1#sh ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

22.2.2.2 1 FULL/BDR 00:00:30 12.1.1.2 FastEthernet0/0

r1#

说明：R1 已经与 R2 建立 OSPF 邻居关系，显示的邻居为对方的 Router-ID 地址。由于是以太网介质，所以默认为多路访问，需要选举 DR/BDR，理论上默认选举 Router-ID 高的为 DR，而 R1 的 Router-ID 为 1.1.1.1，R2 的 Router-ID 为 22.2.2.2，应该选择 R2 为 DR，但我们看到的结果是 R2 是 BDR，R1 才是 DR，这是因为 R1 先配置，R2 后配置，在 R1 配置好 40 秒（默认以太网 wait 时间为 40 秒）后，没有路由器与它竞选 DR，那么它就选自己为 DR，当 DR 选举后，在 DR 没失效的情况下，将保持不变，除非重置 OSPF 进程。

（4）重置 R1 的 OSPF 进程并再次查看邻居关系：

r1#clear ip ospf process

Reset ALL OSPF processes? [no]: y

r1#

再看邻居：

被抢走：

r1#sh ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

22.2.2.2 1 FULL/DR 00:00:38 12.1.1.2 FastEthernet0/0

r1#

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

说明：因为在重置 OSPF 进程后，将重新进行 DR/BDR 的选举，此时由于 R2 的 Router-ID 比 R1 大，所以 R2 被选为了 DR。

（5）查看 R1 的 OSPF 接口：

```
r1#sh ip ospf interface
FastEthernet0/1 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 10.1.1.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   oob-resync timeout 40
   Hello due in 00:00:00
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
```

Last flood scan time is 0 msec, maximum is 0 msec

Neighbor Count is 0, Adjacent neighbor count is 0

Suppress hello for 0 neighbor(s)

FastEthernet0/0 is up, line protocol is up

Internet Address 12.1.1.1/24, Area 0

Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1

Transmit Delay is 1 sec, State BDR, Priority 1

Designated Router (ID) 22.2.2.2, Interface address 12.1.1.2

Backup Designated router (ID) 1.1.1.1, Interface address 12.1.1.1

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:09

Supports Link-local Signaling (LLS)

Cisco NSF helper support enabled

IETF NSF helper support enabled

Index 2/2, flood queue length 0

Next 0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 1

Last flood scan time is 0 msec, maximum is 0 msec

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 22.2.2.2 (Designated Router)

Suppress hello for 0 neighbor(s)

Loopback11 is up, line protocol is up

Internet Address 11.1.1.1/24, Area 0

Process ID 1, Router ID 1.1.1.1, Network Type LOOPBACK, Cost: 1

Loopback interface is treated as a stub Host

r1#

说明：R1 上的 OSPF 接口中，F0/1（10.1.1.0）的网络类型为 BROADCAST，F0/0（12.1.1.0）的网络类型为 BROADCAST，Loopback11（11.1.1.1）只是一个主机类型。

（6）查看 R2 与 R3 的接口状态和邻居状态：

r2# sh ip ospf int Serial1/0.23

Serial1/0.23 is up, line protocol is up

Internet Address 23.1.1.2/24, Area 1

Process ID 1, Router ID 22.2.2.2, Network Type POINT_TO_POINT, Cost: 64

Transmit Delay is 1 sec, State POINT_TO_POINT

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:04

Supports Link-local Signaling (LLS)

Cisco NSF helper support enabled

IETF NSF helper support enabled

Index 1/3, flood queue length 0

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

```
  Next 0x0(0)/0x0(0)

  Last flood scan length is 1, maximum is 3

  Last flood scan time is 0 msec, maximum is 4 msec

  Neighbor Count is 1, Adjacent neighbor count is 1

   Adjacent with neighbor 3.3.3.3

  Suppress hello for 0 neighbor(s)

r2#


r2#sh ip ospf neighbor


Neighbor ID     Pri   State           Dead Time   Address         Interface

1.1.1.1           1   FULL/BDR        00:00:32    12.1.1.1        FastEthernet0/0

3.3.3.3           0   FULL/  -        00:00:33    23.1.1.3        Serial1/0.23

r2#
```

说明：因为 R2 通过帧中继子接口与 R3 相连，该链路被 OSPF 定义为 POINT_TO_POINT 的网络类型，所以 R2 与 R3 不需要进行 DR/BDR 的选举。

3.分析 OSPF LSA

（1）查看 R2 的 LSA 数据库：

```
r2#sh ip ospf database
```

OSPF Router with ID (22.2.2.2) (Process ID 1)

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	345	0x80000005	0x003485	3
22.2.2.2	22.2.2.2	349	0x80000005	0x001A85	2

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
12.1.1.2	22.2.2.2	349	0x80000001	0x00E904

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
23.1.1.0	22.2.2.2	481	0x80000001	0x00D7EF
33.3.3.3	22.2.2.2	461	0x80000001	0x0013A2

Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
3.3.3.3	3.3.3.3	476	0x80000004	0x00E913	3
22.2.2.2	22.2.2.2	476	0x80000002	0x006EBB	2

Summary Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
10.1.1.0	22.2.2.2	343	0x80000001	0x00C739
11.1.1.1	22.2.2.2	343	0x80000001	0x0056B1
12.1.1.0	22.2.2.2	486	0x80000001	0x0049BF
22.2.2.2	22.2.2.2	486	0x80000001	0x0041C2

r2#

说明：因为 R2 是 ABR，同时连接 Area 0 和 Area 1，所以看见 R2 同时拥有 Area 0 和 Area 1 的 LSA 数据库，并且是分区域存放的；仔细查看 Area 0 的 LSA 可以发现，目前 Area 0 拥有 Router Link（类型 1），Net Link（类型 2），Summary Net Link（类型 3），拥有 LSA 1 是每台路由器都必须的，因为 Area 0 总共有两台 OSPF 路由器，所以就有两条 LSA，分别用 R1 的 Router-ID（1.1.1.1）和 R2 的 Router-ID（22.2.2.2）来表示，并且显示了 1.1.1.1 中有三条链路状态，而 22.2.2.2 中有两条链路状态；拥有 LSA 2 是因为 Area 0 中与 R1 相连的链路上选举了 DR/BDR，所以会产生 LSA2，希奇的是，这里的 Link ID 是用 DR 的地址来表示的，如果没有记错，Link ID 应该和链路类型有关，但需要说明的是，这里并没有错，因为通过命令 sh ip ospf database 所看到的并不是真正的 LSA 信息，而只是 LSA 的包头而已，在这里所看到的不能完全代表 LSA 的内容，但这里的 Link ID 在更高一层面来看，它是等同于 Link State ID 的。

还拥有 LSA 3，因为骨干区域理所当然可以拥有其它其它的汇总 LSA，也就拥有了 LSA 3，可以发现，LSA 3 中即用了网络地址来表示，也用了主机地址，这是因为链路类型造成的。

（2）查看 R1 在 Area 0 的 LSA 1 数据库：

r2#sh ip ospf database router 1.1.1.1

OSPF Router with ID (22.2.2.2) (Process ID 1)

Router Link States (Area 0)

LS age: 228

Options: (No TOS-capability, DC)

LS Type: Router Links

Link State ID: 1.1.1.1

Advertising Router: 1.1.1.1

LS Seq Number: 80000006

Checksum: 0x804A

Length: 60

Number of Links: 3

Link connected to: a Stub Network

(Link ID) Network/subnet number: 10.1.1.0

(Link Data) Network Mask: 255.255.255.0

Number of TOS metrics: 0

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

TOS 0 Metrics: 1

Link connected to: a Transit Network

(Link ID) Designated Router address: 12.1.1.2

(Link Data) Router Interface address: 12.1.1.1

Number of TOS metrics: 0

TOS 0 Metrics: 1

Link connected to: a Stub Network

(Link ID) Network/subnet number: 11.1.1.1

(Link Data) Network Mask: 255.255.255.255

Number of TOS metrics: 0

TOS 0 Metrics: 1

r2#

说明：R1（1.1.1.1）发送了 3 条链路状态，分别为自己的接口 F0/1 的网络（10.1.1.0），接口 F0/0 的网络（12.1.1.0），以及 Loopback 11 的网络（11.1.1.1），而 F0/1 的网络（10.1.1.0）被定义为 Stub Network 是因为这条链路上只有一台 OSPF 路由器，没有邻居，所以是 Stub Network，而 F0/0 的网络（12.1.1.0）被定义为 Transit Network，是因为该链路上超过了一台 OSPF 路由器，有 OSPF 邻居，并且因为不是串口，所以是认为链路类型是 Transit Network，而 Loopback11（11.1.1.1）固定没变。

注：详尽表示的信息，请阅读前面 OSPF LSA 的详细解释。

（3）查看 Area 0 的 LSA 2：

r2#sh ip ospf database network

OSPF Router with ID (22.2.2.2) (Process ID 1)

Net Link States (Area 0)

Routing Bit Set on this LSA

LS age: 840

Options: (No TOS-capability, DC)

LS Type: Network Links

Link State ID: 12.1.1.2 (address of Designated Router)

Advertising Router: 22.2.2.2

LS Seq Number: 80000001

Checksum: 0xE904

Length: 32

Network Mask: /24

Attached Router: 22.2.2.2

Attached Router: 1.1.1.1

r2#

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

说明：因为链路上选举了 DR/BDR，所以产生了 LSA 2，并使用 DR 的接口地址来表示。

（4）查看 R3 在 Area 1 的通告的 LSA 1：

r2#sh ip ospf database router 3.3.3.3

OSPF Router with ID (22.2.2.2) (Process ID 1)

Router Link States (Area 1)

LS age: 1388

Options: (No TOS-capability, DC)

LS Type: Router Links

Link State ID: 3.3.3.3

Advertising Router: 3.3.3.3

LS Seq Number: 80000004

Checksum: 0xE913

Length: 60

Number of Links: 3

Link connected to: a Stub Network

(Link ID) Network/subnet number: 33.3.3.3

(Link Data) Network Mask: 255.255.255.255

Number of TOS metrics: 0

TOS 0 Metrics: 1

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

(Link ID) Neighboring Router ID: 22.2.2.2

(Link Data) Router Interface address: 23.1.1.3

Number of TOS metrics: 0

TOS 0 Metrics: 64

Link connected to: a Stub Network

(Link ID) Network/subnet number: 23.1.1.0

(Link Data) Network Mask: 255.255.255.0

Number of TOS metrics: 0

TOS 0 Metrics: 64

r2#

说明：R3 在区域 1 通告了 3 链路状态，分别是 33.3.3.3，因为是 loopback，所以是 32 位地址，而与 R2 相连的为点到点链路，所以 Link ID 使用了邻居的 Router-ID 来表示，点到点链路还会自动产生一条 Stub Network 的链路状态。

（5）在 R1 上查看 Area 0 的 LSA 数据库：

r1#sh ip ospf database

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

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	1005	0x80000006	0x00804A	3
22.2.2.2	22.2.2.2	1000	0x80000006	0x007532	2

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
12.1.1.2	22.2.2.2	1442	0x80000001	0x00E904

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
23.1.1.0	22.2.2.2	1574	0x80000001	0x00D7EF
33.3.3.3	22.2.2.2	1554	0x80000001	0x0013A2

r1#

说明：因为 R1 是骨干区域路由器，所以拥有了 Area 0 与 Area 1 的所有 LSA。

（6）在 R1 上查看 Area 0 的 LSA 3：

r1#sh ip ospf database summary 33.3.3.3

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

Summary Net Link States (Area 0)

Routing Bit Set on this LSA

LS age: 1656

Options: (No TOS-capability, DC, Upward)

LS Type: Summary Links(Network)

Link State ID: 33.3.3.3 (summary Network Number)

Advertising Router: 22.2.2.2

LS Seq Number: 80000001

Checksum: 0x13A2

Length: 28

Network Mask: /32

TOS: 0 Metric: 65

r1#

说明：可以看见从 Area 1 到达 Area 0 的 LSA 在经过 ABR 转发时，Router-ID 被 ABR（R2）修改成了自己的 Router-ID，因为不同区域的 Router-ID 不可达，所以 R2 修改成自己的，便于 Area 0 能够通过自己到达目标网络。

（7）调整 R1 连接 R2 的接口网络类型：

r1(config)#int f0/0

r1(config-if)#ip os network point-to-point

说明：原本相连的接口类型为以太网，网络类型为多路访问，Hello 时间为 10 称，改成 point-to-point 后，Hello 时间并没有改变，查看当前邻居状态：

r1#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
22.2.2.2	0	FULL/ -	00:00:35	12.1.1.2	FastEthernet0/0

r1#

说明：因为 Hello 时间匹配，其它参数也正常，所以邻居关系正常。

（8）查看 R1 当前的路由表：

r1#sh ip route

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

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

```
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/1
     11.0.0.0/24 is subnetted, 1 subnets
C       11.1.1.0 is directly connected, Loopback11
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, FastEthernet0/0
r1#
```

说明：当邻居双方网络类型不匹配时，可以建立邻居，但有时 LSA 数据库中的条目将无法进入路由表，所以当前 R1 的路由表为空，此状态不正常，所以请谨慎修改接口网络类型。

（9）还原 R1 连接 R2 的接口网络类型并查看路由表：

r1(config)#int f0/0

r1(config-if)#ip ospf network broadcast

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

33.0.0.0/32 is subnetted, 1 subnets

O IA 33.3.3.3 [110/66] via 12.1.1.2, 00:00:06, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

O IA 23.1.1.0 [110/65] via 12.1.1.2, 00:00:06, FastEthernet0/0

22.0.0.0/32 is subnetted, 1 subnets

O 22.2.2.2 [110/2] via 12.1.1.2, 00:00:06, FastEthernet0/0

10.0.0.0/24 is subnetted, 1 subnets

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

C 10.1.1.0 is directly connected, FastEthernet0/1

11.0.0.0/24 is subnetted, 1 subnets

C 11.1.1.0 is directly connected, Loopback11

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

r1#

说明：在网络类型匹配的情况下建立邻居后，路由表完全正常，所以不能随意修改接口网络类型，虽然不影响邻居的建立，但影响路由表的计算。

（10）查看 R3 的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

33.0.0.0/24 is subnetted, 1 subnets

C 33.3.3.0 is directly connected, Loopback33

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, Serial1/0.23

22.0.0.0/32 is subnetted, 1 subnets

O IA 22.2.2.2 [110/65] via 23.1.1.2, 00:20:48, Serial1/0.23

10.0.0.0/24 is subnetted, 1 subnets

O IA 10.1.1.0 [110/66] via 23.1.1.2, 00:20:48, Serial1/0.23

11.0.0.0/32 is subnetted, 1 subnets

O IA 11.1.1.1 [110/66] via 23.1.1.2, 00:20:48, Serial1/0.23

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/65] via 23.1.1.2, 00:20:48, Serial1/0.23

44.0.0.0/32 is subnetted, 1 subnets

O 44.4.4.4 [110/2] via 34.1.1.4, 00:20:48, FastEthernet0/0

r3#

说明：因为 R3 所在的区域 1 与骨干区域直连，所以路由表正常。

（11）查看 R4 的 LSA 数据库：

r4#sh ip ospf database

OSPF Router with ID (34.1.1.4) (Process ID 1)

Router Link States (Area 2)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
3.3.3.3	3.3.3.3	1219	0x80000003	0x00DBEE	1
34.1.1.4	34.1.1.4	1214	0x80000003	0x0054F3	2

Net Link States (Area 2)

Link ID	ADV Router	Age	Seq#	Checksum
34.1.1.3	3.3.3.3	413	0x80000002	0x008150

r4#

说明：因为 R4 所在的区域 2 连接的不是骨干区域，所以不能获得区域外的 LSA，最终路由表不正常。

4.配置 OSPF 虚链路

（1）在 R2 与 R3 之间建 OSPF 虚链路：

r2(config)#router ospf 1

r2(config-router)#area 1 virtual-link 3.3.3.3

r3(config)#router ospf 1

r3(config-router)#area 1 virtual-link 22.2.2.2

说明：在 R2 与 R3 之间通过 Router-ID 建 OSPF 虚链路，Area 1 为 Transit Area。

（2）查看 OSPF 虚链路状态：

r3#sh ip ospf virtual-links

Virtual Link OSPF_VL0 to router 22.2.2.2 is up

Run as demand circuit

DoNotAge LSA allowed.

Transit area 1, via interface Serial1/0.23, Cost of using 64

Transmit Delay is 1 sec, State POINT_TO_POINT,

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

Hello due in 00:00:05

Adjacency State FULL (Hello suppressed)

Index 1/3, retransmission queue length 0, number of retransmission 1

First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)

Last retransmission scan length is 1, maximum is 1

Last retransmission scan time is 0 msec, maximum is 0 msec

r3#

说明：R2 与 R3 之间的 OSPF 虚链路已成功建立，并且显示虚链路上的 Hello 包是被抑制的。

（3）查看 R3 当前的 OSPF 邻居状态：

r3#sh ip os neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
22.2.2.2	0	FULL/ -	-	23.1.1.2	OSPF_VL0
22.2.2.2	0	FULL/ -	00:00:33	23.1.1.2	Serial1/0.23
34.1.1.4	1	FULL/BDR	00:00:34	34.1.1.4	FastEthernet0/0

r3#

说明：由于虚链路上的 Hello 包是被抑制的，所以没有 Dead 时间。

（4）查看 R3 建立虚链路后的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

33.0.0.0/24 is subnetted, 1 subnets

C 33.3.3.0 is directly connected, Loopback33

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, Serial1/0.23

22.0.0.0/32 is subnetted, 1 subnets

O 22.2.2.2 [110/65] via 23.1.1.2, 00:00:42, Serial1/0.23

10.0.0.0/24 is subnetted, 1 subnets

O 10.1.1.0 [110/66] via 23.1.1.2, 00:00:42, Serial1/0.23

11.0.0.0/32 is subnetted, 1 subnets

O 11.1.1.1 [110/66] via 23.1.1.2, 00:00:42, Serial1/0.23

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/65] via 23.1.1.2, 00:00:42, Serial1/0.23

44.0.0.0/32 is subnetted, 1 subnets

O 44.4.4.4 [110/2] via 34.1.1.4, 00:01:02, FastEthernet0/0

r3#

说明：因为之前 R3 是区域 1 的路由器，所以收到的区域 0 的路由显示为 O IA，

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

但通过虚链路将骨干区域扩展到区域 1 后，R3 也相当于是骨干区域的路由器，因此区域 0 的路由从 O IA 转变成 O。

（5）查看 R4 建立虚链路后的路由表：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

33.0.0.0/32 is subnetted, 1 subnets

O IA 33.3.3.3 [110/2] via 34.1.1.3, 00:01:43, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

O IA 23.1.1.0 [110/65] via 34.1.1.3, 00:01:43, FastEthernet0/0

22.0.0.0/32 is subnetted, 1 subnets

O IA 22.2.2.2 [110/66] via 34.1.1.3, 00:01:23, FastEthernet0/0

10.0.0.0/24 is subnetted, 1 subnets

O IA 10.1.1.0 [110/67] via 34.1.1.3, 00:01:23, FastEthernet0/0

11.0.0.0/32 is subnetted, 1 subnets

O IA 11.1.1.1 [110/67] via 34.1.1.3, 00:01:23, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/66] via 34.1.1.3, 00:01:24, FastEthernet0/0

44.0.0.0/24 is subnetted, 1 subnets

C 44.4.4.0 is directly connected, Loopback44

r4#

说明：原本 R4 所在的区域 2 与区域 1 相连，由于通过虚链路将骨干区域扩展到区域 1 后，区域 2 相当于和骨干区域相连，所以 R4 拥有了全网的路由。

5.配置 OSPF 认证

（1）查看认证之前，R1 与 R2 的邻居状态：

r1#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
22.2.2.2	1	FULL/DR	00:00:30	12.1.1.2	FastEthernet0/0

r1#

说明：R1 与 R2 邻居正常。

（2）在 R1 连接 R2 的接口上开启 MD5 认证：

r1(config)#int f0/0

r1(config-if)#ip ospf authentication message-digest

说明：在 R1 连接 R2 的接口上开启了 MD5 认证。

（3）查看 R1 连接 R2 的接口认证状态：

r1#sh ip ospf interface f0/0

FastEthernet0/0 is up, line protocol is up

Internet Address 12.1.1.1/24, Area 0

Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1

Transmit Delay is 1 sec, State BDR, Priority 1

Designated Router (ID) 22.2.2.2, Interface address 12.1.1.2

Backup Designated router (ID) 1.1.1.1, Interface address 12.1.1.1

Flush timer for old DR LSA due in 00:02:04

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:04

Supports Link-local Signaling (LLS)

Cisco NSF helper support enabled

IETF NSF helper support enabled

Index 2/2, flood queue length 0

Next 0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 1

Last flood scan time is 0 msec, maximum is 0 msec

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 22.2.2.2 (Designated Router)

Suppress hello for 0 neighbor(s)

Message digest authentication enabled

No key configured, using default key id 0

r1#

说明：在开启认证后，如果接口上不指定密码，那么则使用空密码认证，空密码也是密码的一种，也需要使用空密码来匹配。

（4）查看配置认证之后，R1 与 R2 的邻居状态：

r1#sh ip ospf neighbor

r1#

说明：由于 R1 使用了空密码认证，而 R2 没有认证，所以邻居断开了。

（5）在 R2 连接 R1 的接口上开启 MD5 认证：

r2(config)#int f0/0

r2(config-if)#ip ospf authentication message-digest

说明：在 R2 连接 R1 的接口上开启了 MD5 认证：在没有指定密码的情况下，默认使用空密码认证。

（6）查看双方配置认证之后的邻居状态：

r1#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
22.2.2.2	1	FULL/DR	00:00:36	12.1.1.2	FastEthernet0/0

r1#

说明：由于双方都在接口上配置了认证并使用空密码认证，所以邻居成功建立。

（6）在 R1 连接 R2 的接口上启用非空密码：

r1(config)#int f0/0

r1(config-if)#ip ospf message-digest-key 1 md5 cisco

说明：R1 使用密码 cisco 与 R2 认证。

（7）查看使用密码 cisco 认证后的邻居状态：

r1#sh ip ospf neighbor

r1#

说明：因为一方使用密码 cisco，一方使用空密码，所以双方密码不匹配，邻居仍然无法建立。

（8）在 R2 连接 R1 的接口上启用非空密码：

r2(config)#int f0/0

r2(config-if)#ip ospf message-digest-key 1 md5 cisco

说明：R2 使用密码 cisco 与 R1 认证。

（9）查看双方邻居状态：

r1#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
22.2.2.2	1	FULL/DR	00:00:37	12.1.1.2	FastEthernet0/0

r1#

说明：因为双方都配置了认证，并且都使用密码 cisco 认证，所以认证通过，邻居正常建立。

（10）R2 在 OSPF 进程下开启认证：

r2#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
3.3.3.3	0	FULL/ -	-	23.1.1.3	OSPF_VL0
1.1.1.1	1	FULL/DR	00:00:30	12.1.1.1	FastEthernet0/0
3.3.3.3	0	FULL/ -	00:00:36	23.1.1.3	Serial1/0.23

r2#

开认证：

r2(config)#router ospf 1

r2(config-router)#area 0 authentication message-digest

说明：R2 在与 R1 和 R3 邻居正常，并且虚链路正常的情况下，在 OSPF 进程中开启了 Area 0 的认证。

（11）查看 R2 上 OSPF 接口认证状态：

r2#sh ip ospf interface

OSPF_VL0 is up, line protocol is up

Internet Address 23.1.1.2/24, Area 0

Process ID 1, Router ID 22.2.2.2, Network Type VIRTUAL_LINK, Cost: 64

Configured as demand circuit.

Run as demand circuit.

DoNotAge LSA allowed.

Transmit Delay is 1 sec, State POINT_TO_POINT

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:02

Supports Link-local Signaling (LLS)

Cisco NSF helper support enabled

IETF NSF helper support enabled

Index 3/4, flood queue length 0

Next 0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 3

Last flood scan time is 4 msec, maximum is 4 msec

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 3.3.3.3 (Hello suppressed)

Suppress hello for 1 neighbor(s)

Message digest authentication enabled

No key configured, using default key id 0

FastEthernet0/0 is up, line protocol is up

Internet Address 12.1.1.2/24, Area 0

Process ID 1, Router ID 22.2.2.2, Network Type BROADCAST, Cost: 1

Transmit Delay is 1 sec, State BDR, Priority 1

Designated Router (ID) 1.1.1.1, Interface address 12.1.1.1

Backup Designated router (ID) 22.2.2.2, Interface address 12.1.1.2

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:02

Supports Link-local Signaling (LLS)

Cisco NSF helper support enabled

IETF NSF helper support enabled

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

Index 2/2, flood queue length 0

Next 0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 4

Last flood scan time is 4 msec, maximum is 4 msec

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 1.1.1.1 (Designated Router)

Suppress hello for 0 neighbor(s)

Message digest authentication enabled

No key configured, using default key id 0

Loopback22 is up, line protocol is up

Internet Address 22.2.2.2/24, Area 0

Process ID 1, Router ID 22.2.2.2, Network Type LOOPBACK, Cost: 1

Loopback interface is treated as a stub Host

Serial1/0.23 is up, line protocol is up

Internet Address 23.1.1.2/24, Area 1

Process ID 1, Router ID 22.2.2.2, Network Type POINT_TO_POINT, Cost: 64

Transmit Delay is 1 sec, State POINT_TO_POINT

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

oob-resync timeout 40

Hello due in 00:00:01

Supports Link-local Signaling (LLS)

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

Cisco NSF helper support enabled

IETF NSF helper support enabled

Index 1/3, flood queue length 0

Next 0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 4

Last flood scan time is 0 msec, maximum is 4 msec

Neighbor Count is 1, Adjacent neighbor count is 1

Adjacent with neighbor 3.3.3.3

Suppress hello for 0 neighbor(s)

r2#

说明：R2 在 OSPF 进程下开启 Area 0 的认证后，等于在所有 Area 0 的接口下启用了认证。

（12）查看 R2 上 OSPF 邻居状态：

r2#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
3.3.3.3	0	FULL/ -	-	23.1.1.3	OSPF_VL0
3.3.3.3	0	FULL/ -	00:00:38	23.1.1.3	Serial1/0.23

r2#

说明：由于 R2 连接 R1 的接口在 Area 0，所以在 Area 0 开启认证后，该接口默认使用了空密码认证，而 R1 并没有使用空密码认证，所以 R2 与 R2 邻居断开，因为与 R3 相连的接口并非 Area 0 的接口，所以与 R3 的 OSPF 邻居正常。

（13）查看 R1 上开启空密码认证：

对方开认证：

r1(config)#int f0/0

r1(config-if)#ip ospf authentication message-digest

（14）再次查看 R2 的 OSPF 邻居

r2#sh ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

3.3.3.3 0 FULL/ - - 23.1.1.3 OSPF_VL0

1.1.1.1 1 FULL/DROTHER 00:00:38 12.1.1.1 FastEthernet0/0

3.3.3.3 0 FULL/ - 00:00:38 23.1.1.3 Serial1/0.23

r2#

说明：由于 R1 与 R2 在 Area 0 的接口都开启了空密码认证，所以认证通过。需要注意，虽然虚链路上没有 Hello 包交换，在 Area 0 开启认证后，同时也开启了虚链路的空密码认证，如果虚链路两端不配置相同认证，稍后邻居也会断开，虚链路将无法继续维持。

（15）到虚链路对端（R3）开启空密码认证：

r3(config)#router ospf 1

r3(config-router)#area 1 virtual-link 22.2.2.2 authentication message-digest

说明：在虚链路双方都配置相同认证后，一切将会变得正常。

（16）将 R3 与 R4 的区域 2 变成骨干区域：

r3(config)#router ospf 1

r3(config-router)#network 34.1.1.3 0.0.0.0 area 0

R4：

r4(config)#router ospf 1

r4(config-router)#network 0.0.0.0 0.0.0.0 area 0

说明：当 R3 与 R4 的区域 2 变成骨干区域后，那么网络中骨干区域将被区域 1 分割为两部分。

（17）查看骨干区域内 R4 的路由表：

r4#sh ip route

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

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

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

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

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

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

        o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     34.0.0.0/24 is subnetted, 1 subnets

C       34.1.1.0 is directly connected, FastEthernet0/0

     33.0.0.0/32 is subnetted, 1 subnets

O IA    33.3.3.3 [110/2] via 34.1.1.3, 00:01:02, FastEthernet0/0

     23.0.0.0/24 is subnetted, 1 subnets

O IA    23.1.1.0 [110/65] via 34.1.1.3, 00:01:02, FastEthernet0/0

     44.0.0.0/24 is subnetted, 1 subnets

C       44.4.4.0 is directly connected, Loopback44

r4#
```

说明：虽然 R4 当前为骨干区域的路由器，但是由于骨干区域被区域 1 分割成两部分，所以无法相互传递 LSA，路由表变的不正常。

（18）创建 OSPF 虚链路连接骨干区域：

```
r2(config-router)#router os 1
```

r2(config-router)# area 1 virtual-link 3.3.3.3

r3(config)#router ospf 1

r3(config-router)#area 1 virtual-link 22.2.2.2

说明： 骨干区域被分割后必须创建 OSPF 虚链路。

（19）再次查看骨干区域内 R4 的路由表：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

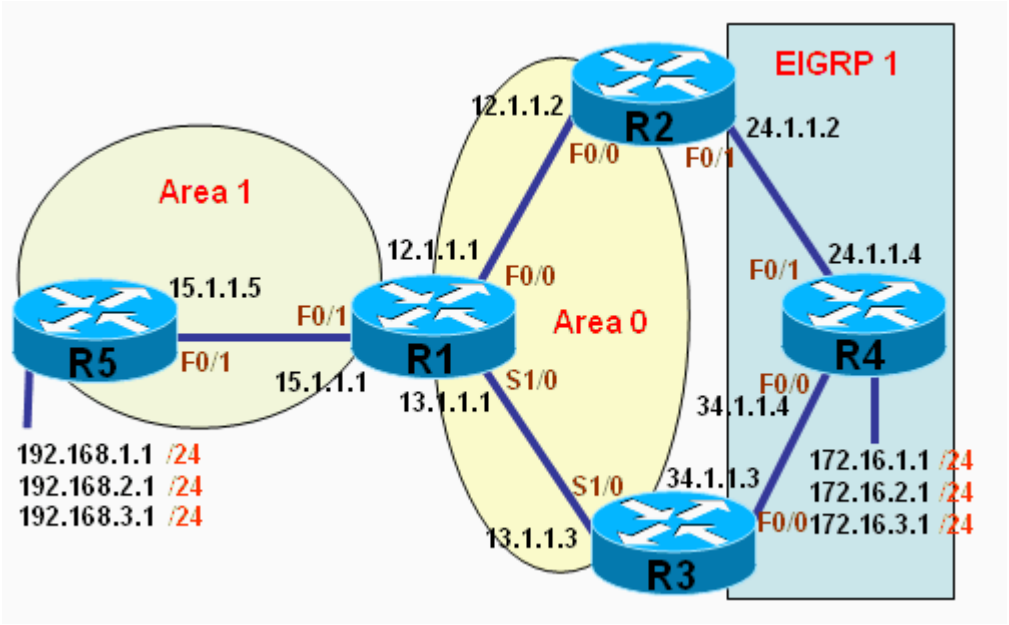
C 34.1.1.0 is directly connected, FastEthernet0/0

33.0.0.0/32 is subnetted, 1 subnets

```
O IA     33.3.3.3 [110/2] via 34.1.1.3, 00:00:04, FastEthernet0/0

    23.0.0.0/24 is subnetted, 1 subnets

O IA     23.1.1.0 [110/65] via 34.1.1.3, 00:00:04, FastEthernet0/0

    22.0.0.0/32 is subnetted, 1 subnets

O        22.2.2.2 [110/66] via 34.1.1.3, 00:00:04, FastEthernet0/0

    10.0.0.0/24 is subnetted, 1 subnets

O        10.1.1.0 [110/67] via 34.1.1.3, 00:00:04, FastEthernet0/0

    11.0.0.0/32 is subnetted, 1 subnets

O        11.1.1.1 [110/67] via 34.1.1.3, 00:00:04, FastEthernet0/0

    12.0.0.0/24 is subnetted, 1 subnets

O        12.1.1.0 [110/66] via 34.1.1.3, 00:00:04, FastEthernet0/0

    44.0.0.0/24 is subnetted, 1 subnets

C        44.4.4.0 is directly connected, Loopback44

r4#
```

说明：由于创建了 OSPF 虚链路将分割的骨干区域连接起来，所以 R4 获得了区域外的路由信息，并且另外部分骨干区域的路由显示为 O，表示是同区域路由，而 Area 1 还是与骨干区域属于不同区域，所以使用 O IA 来表示。

以下图为例，配置 OSPF 外部路由，OSPF 路由汇总，OSPF 末节区域：



1.配置基础网络环境

（1）配置 R1：

r1(config)#int f0/0

r1(config-if)#ip add 12.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#int f0/1

r1(config-if)#ip add 15.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#int s1/0

r1(config-if)#encapsulation frame-relay

r1(config-if)#no frame-relay inverse-arp

r1(config-if)#no arp frame-relay

r1(config-if)#ip address 13.1.1.1 255.255.255.0

r1(config-if)#no shutdown

r1(config-if)#frame-relay map ip 13.1.1.3 103 broadcast

r1(config-if)#ip ospf network point-to-point

r1(config)#router ospf 1

r1(config-router)#router-id 1.1.1.1

r1(config-router)#network 12.1.1.1 0.0.0.0 area 0

r1(config-router)#network 13.1.1.1 0.0.0.0 area 0

r1(config-router)#network 15.1.1.1 0.0.0.0 area 1

说明：在 R1 上配置 12.1.1.0/24，13.1.1.0/24，15.1.1.0/24，并放入 OSPF 进程。

（2）配置 R2：

r2(config)#int f0/0

r2(config-if)#ip address 12.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#int f0/1

r2(config-if)#ip address 24.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#router ospf 1

r2(config-router)#router-id 2.2.2.2

r2(config-router)#network 12.1.1.2 0.0.0.0 area 0

r2(config)#router eigrp 1

r2(config-router)#no auto-summary

r2(config-router)#network 24.1.1.2 0.0.0.0

说明：在 R2 上配置 12.1.1.0/24，24.1.1.0/24，并将 121.1.0 放入 OSPF 进程，将 24.1.1.0 放入 EIGRP 进程。

（3）配置 R3：

r3(config)#int s1/0

r3(config-if)#encapsulation frame-relay

r3(config-if)#no frame-relay inverse-arp

r3(config-if)#no arp frame-relay

r3(config-if)#ip address 13.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config-if)#frame-relay map ip 13.1.1.1 301 broadcast

r3(config-if)#ip ospf network point-to-point

r3(config)#int f0/0

r3(config-if)#ip add 34.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#router ospf 1

r3(config-router)#router-id 3.3.3.3

r3(config-router)#network 13.1.1.3 0.0.0.0 area 0

r3(config)#router eigrp 1

r3(config-router)#no auto-summary

r3(config-router)#network 34.1.1.3 0.0.0.0

说明：在 R3 上配置 13.1.1.0/24，34.1.1.0/24，并将 131.1.0 放入 OSPF 进程，将 34.1.1.0 放入 EIGRP 进程。

（4）配置 R4：

r4(config)#int f0/1

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

```
r4(config-if)#ip add 24.1.1.4 255.255.255.0

r4(config-if)#no sh


r4(config)#int f0/0

r4(config-if)#ip address 34.1.1.4 255.255.255.0

r4(config-if)#no sh


r4(config)#int loopback 172

r4(config-if)#ip add 172.16.1.1 255.255.255.0

r4(config-if)#ip add 172.16.2.1 255.255.255.0 secondary

r4(config-if)#ip add 172.16.3.1 255.255.255.0 secondary


r4(config)#router eigrp 1

r4(config-router)#no auto-summary

r4(config-router)#network 24.1.1.4 0.0.0.0

r4(config-router)#network 34.1.1.4 0.0.0.0

r4(config-router)#network 172.0.0.0 0.255.255.255
```

说明：在 R4 上配置 24.1.1.0/24，34.1.1.0/24，172.16.1.0/24，172.16.2.0/24，172.16.3.0/24，并全部放入 EIGRP 进程。

（5）配置 R5：

r5(config)#int f0/1

r5(config-if)#ip address 15.1.1.5 255.255.255.0

r5(config-if)#no sh

r5(config)#int loopback 192

r5(config-if)#ip address 192.168.1.1 255.255.255.0

r5(config-if)#ip address 192.168.2.1 255.255.255.0 secondary

r5(config-if)#ip address 192.168.3.1 255.255.255.0 secondary

r5(config)#router ospf 1

r5(config-router)#router-id 5.5.5.5

r5(config-router)#network 15.1.1.5 0.0.0.0 area 1

说明：在 R5 上配置 15.1.1.0/24， 192.168.1.0/24， 192.168.2.0/24，192.168.3.0/24，并将 15.1.1.0/24 放入 OSPF 进程。

2.测试 OSPF 外部路由

（1）查看 R1 的 OSPF 邻居状态：

r1#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
3.3.3.3	0	FULL/ -	00:00:38	13.1.1.3	Serial1/0
2.2.2.2	1	FULL/BDR	00:00:31	12.1.1.2	FastEthernet0/0
5.5.5.5	1	FULL/BDR	00:00:33	15.1.1.5	FastEthernet0/1

r1#

说明：R1 已经与所有直连路由器建立邻居关系。

（2）查看 R1 当前的路由表：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：因为 OSPF 区域内没有路由，所以 R1 没有从 OSPF 收到路由。

（3）查看 R2 与 R3 的路由表：

R2：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

D 34.1.1.0 [90/30720] via 24.1.1.4, 00:05:30, FastEthernet0/1

172.16.0.0/24 is subnetted, 3 subnets

D 172.16.1.0 [90/156160] via 24.1.1.4, 00:05:24, FastEthernet0/1

D 172.16.2.0 [90/156160] via 24.1.1.4, 00:05:24, FastEthernet0/1

D 172.16.3.0 [90/156160] via 24.1.1.4, 00:05:24, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

C 24.1.1.0 is directly connected, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

O 13.1.1.0 [110/65] via 12.1.1.1, 00:07:01, FastEthernet0/0

15.0.0.0/24 is subnetted, 1 subnets

O IA 15.1.1.0 [110/2] via 12.1.1.1, 00:07:02, FastEthernet0/0

r2#

R3：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

172.16.0.0/24 is subnetted, 3 subnets

D 172.16.1.0 [90/156160] via 34.1.1.4, 00:05:48, FastEthernet0/0

D 172.16.2.0 [90/156160] via 34.1.1.4, 00:05:48, FastEthernet0/0

D 172.16.3.0 [90/156160] via 34.1.1.4, 00:05:48, FastEthernet0/0

24.0.0.0/24 is subnetted, 1 subnets

D 24.1.1.0 [90/30720] via 34.1.1.4, 00:05:53, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/65] via 13.1.1.1, 00:07:26, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

O IA 15.1.1.0 [110/65] via 13.1.1.1, 00:07:27, Serial1/0

r3#

说明：因为 R2 与 R3 都配有 EIGRP，所以从 EIGRP 收到了路由，其中包括 172.16.1.0/24，172.16.2.0/24，172.16.3.0/24，并且 R2 与 R3 从 EIGRP 学习到的路由条目相同。

（4）在 R2 与 R3 上同时将 EIGRP 重分布进 OSPF：

r2(config)#router ospf 1

r2(config-router)#redistribute eigrp 1 subnets

r3(config)#router ospf 1

r3(config-router)#redistribute eigrp 1 subnets

说明：R2 与 R3 将 EIGRP 重分布进 OSPF，包括所有子网信息。

（5）查看 R1 从 R2 和 R3 收到的外部路由：

r1#

r1#sh ip ospf database external 172.16.1.0

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

Type-5 AS External Link States

Routing Bit Set on this LSA

LS age: 126

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 172.16.1.0 (External Network Number)

Advertising Router: 2.2.2.2

LS Seq Number: 80000001

Checksum: 0x1BBF

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 0.0.0.0

External Route Tag: 0

LS age: 91

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 172.16.1.0 (External Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000001

Checksum: 0xFCD9

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 0.0.0.0

External Route Tag: 0

r1#

说明：可以看见，默认为类型 2 的外部路由，因为 R2 与 R3 到达 OSPF 外部路由的下一跳不在 OSPF 进程中，所以 R2 和 R3 都在外部路由中将 Forward Address 写为 0.0.0.0，最终到 ASBR 最近的路径将被优先使用。

（6）R1 到达 ASBR（R2）和（R3）的距离：

r1#sh ip ospf border-routers

OSPF Process 1 internal Routing Table

Codes: i - Intra-area route, I - Inter-area route

i 2.2.2.2 [1] via 12.1.1.2, FastEthernet0/0, ASBR, Area 0, SPF 9

i 3.3.3.3 [64] via 13.1.1.3, Serial1/0, ASBR, Area 0, SPF 9

r1#

说明：很明显，R1 到达 ASBR（R2）更近，由此可以得出结论，R2 将被选为去往外部路由转发路由器。

（7）查看 R1 到达外部路由的路径：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O E2 34.1.1.0 [110/20] via 12.1.1.2, 00:02:36, FastEthernet0/0

172.16.0.0/24 is subnetted, 3 subnets

O E2 172.16.1.0 [110/20] via 12.1.1.2, 00:02:36, FastEthernet0/0

O E2 172.16.2.0 [110/20] via 12.1.1.2, 00:02:36, FastEthernet0/0

O E2 172.16.3.0 [110/20] via 12.1.1.2, 00:02:36, FastEthernet0/0

24.0.0.0/24 is subnetted, 1 subnets

O E2 24.1.1.0 [110/20] via 12.1.1.2, 00:02:36, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：和计算的一样，当外部路由类型相同时，由于 R1 到达 R2 更近，所以选择了 R2 去往 OSPF 外部路由。

（8）修改 R1 到达 ASBR （R2）的 Cost

r1(config)#int f0/0

r1(config-if)#ip ospf cost 65

说明：将连接 ASBR（R2）的 Cost 值加大。

（9）再次查看 R1 到达 ASBR（R2）和（R3）的距离：

r1#sh ip ospf border-routers

OSPF Process 1 internal Routing Table

Codes: i - Intra-area route, I - Inter-area route

i 2.2.2.2 [65] via 12.1.1.2, FastEthernet0/0, ASBR, Area 0, SPF 10

i 3.3.3.3 [64] via 13.1.1.3, Serial1/0, ASBR, Area 0, SPF 10

r1#

说明：现在 ASBR（R2）比 R3 离 R1 更远，所以 R3 将被优选。

（10）再次查看 R1 到达外部路由的路径：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O E2 34.1.1.0 [110/20] via 13.1.1.3, 00:00:32, Serial1/0

172.16.0.0/24 is subnetted, 3 subnets

O E2 172.16.1.0 [110/20] via 13.1.1.3, 00:00:32, Serial1/0

```
O E2     172.16.2.0 [110/20] via 13.1.1.3, 00:00:32, Serial1/0

O E2     172.16.3.0 [110/20] via 13.1.1.3, 00:00:32, Serial1/0

     24.0.0.0/24 is subnetted, 1 subnets

O E2     24.1.1.0 [110/20] via 13.1.1.3, 00:00:32, Serial1/0

     12.0.0.0/24 is subnetted, 1 subnets

C        12.1.1.0 is directly connected, FastEthernet0/0

     13.0.0.0/24 is subnetted, 1 subnets

C        13.1.1.0 is directly connected, Serial1/0

     15.0.0.0/24 is subnetted, 1 subnets

C        15.1.1.0 is directly connected, FastEthernet0/1

r1#
```

说明：因为修改了连接 R2 的 Cost 值，所以这时 R1 到达 R3 更近，所以选择了 R3 去往 OSPF 外部路由。

（11）将 ASBR（R2）到达外部路由的下一跳地址通告进 OSPF：

```
r2(config)#router ospf 1

r2(config-router)#network 24.1.1.2 0.0.0.0 area 0
```

说明：将 ASBR 到达外部路由的下一跳地址通告进 OSPF。

（12）查看 R1 中外部路由的 LSA 信息：

r1#sh ip os database external 172.16.1.0

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

Type-5 AS External Link States

Routing Bit Set on this LSA

LS age: 315

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 172.16.1.0 (External Network Number)

Advertising Router: 2.2.2.2

LS Seq Number: 80000005

Checksum: 0x6D4B

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 24.1.1.4

External Route Tag: 0

Routing Bit Set on this LSA

LS age: 941

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 172.16.1.0 (External Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000001

Checksum: 0xFCD9

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 0.0.0.0

External Route Tag: 0

r1#

说明：由于（R2）将到达外部路由的下一跳地址通告进了 OSPF，所以 R2 通告的外部路由的 Forward Address 为真实地址，而 R3 还是保持原状。

（13）查看 R1 到达 ASBR 的 Cost 与到达外部路由的路径：

r1#sh ip ospf border-routers

OSPF Process 1 internal Routing Table

Codes: i - Intra-area route, I - Inter-area route

i 2.2.2.2 [65] via 12.1.1.2, FastEthernet0/0, ASBR, Area 0, SPF 17

i 3.3.3.3 [64] via 13.1.1.3, Serial1/0, ASBR, Area 0, SPF 17

r1#

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O E2 34.1.1.0 [110/20] via 13.1.1.3, 00:04:19, Serial1/0

172.16.0.0/24 is subnetted, 3 subnets

O E2 172.16.1.0 [110/20] via 13.1.1.3, 00:04:19, Serial1/0

O E2 172.16.2.0 [110/20] via 13.1.1.3, 00:04:19, Serial1/0

O E2 172.16.3.0 [110/20] via 13.1.1.3, 00:04:19, Serial1/0

24.0.0.0/24 is subnetted, 1 subnets

O 24.1.1.0 [110/66] via 12.1.1.2, 00:04:19, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：R1 还是选择 R3 去往外部路由。

（14）将 ASBR（R3）到达外部路由的下一跳地址也通告进 OSPF：

r3(config)#router ospf 1

r3(config-router)#network 34.1.1.3 0.0.0.0 area 0

说明：将 ASBR 到达外部路由的下一跳地址通告进 OSPF。

（15）查看 R1 中外部路由的 LSA 信息：

r1#sh ip os database external 172.16.1.0

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

Type-5 AS External Link States

LS age: 404

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 172.16.1.0 (External Network Number)

Advertising Router: 2.2.2.2

LS Seq Number: 80000005

Checksum: 0x6D4B

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 24.1.1.4

External Route Tag: 0

Routing Bit Set on this LSA

LS age: 16

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 172.16.1.0 (External Network Number)

Advertising Router: 3.3.3.3

LS Seq Number: 80000002

Checksum: 0xC3E9

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 34.1.1.4

External Route Tag: 0

r1#

说明：由于（R3）也将到达外部路由的下一跳地址通告进了 OSPF，所以 R2 和 R3 通告的外部路由的 Forward Address 都为真实地址。

（16）查看 R1 到达外部路由的真实 Forward Address 地址的距离：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/65] via 13.1.1.3, 00:00:46, Serial1/0

172.16.0.0/24 is subnetted, 3 subnets

O E2 172.16.1.0 [110/20] via 13.1.1.3, 00:00:46, Serial1/0

O E2 172.16.2.0 [110/20] via 13.1.1.3, 00:00:46, Serial1/0

O E2 172.16.3.0 [110/20] via 13.1.1.3, 00:00:46, Serial1/0

24.0.0.0/24 is subnetted, 1 subnets

O 24.1.1.0 [110/66] via 12.1.1.2, 00:00:46, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：到达 ASBR（R3）的真实 Forward Address 地址的距离小于 R2。

（17）改大 R1 到达 R3 的外部路由的真实 Forward Address 地址的距离：

r3(config)#int f0/0

r3(config-if)#ip os cost 3

说明：改变 R1 到达 R3 的外部路由的真实 Forward Address 地址的距离大于 R2。

（18）查看 R1 到达外部路由的路径：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/67] via 13.1.1.3, 00:00:05, Serial1/0

172.16.0.0/24 is subnetted, 3 subnets

O E2 172.16.1.0 [110/20] via 12.1.1.2, 00:00:05, FastEthernet0/0

O E2 172.16.2.0 [110/20] via 12.1.1.2, 00:00:05, FastEthernet0/0

O E2 172.16.3.0 [110/20] via 12.1.1.2, 00:00:05, FastEthernet0/0

24.0.0.0/24 is subnetted, 1 subnets

O 24.1.1.0 [110/66] via 12.1.1.2, 00:00:05, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：由于 R1 到达 R2 的外部路由的真实 Forward Address 地址的距离小于 R3，所以优先 R2 到达外部路由；当有类型 2 和类型 1 可同时到达外部路由时，类型 1 被优选，如果相同类型，且 Forward Address 不全都为真实地址，则比较到达 ASBR 的距离，小的优先，如果都为真实地址，则比较到达该地址小的优先。

（19）查看 R1 的 LSA 数据库：

r1#

r1#sh ip ospf database

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

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	796	0x80000005	0x00B786	3
2.2.2.2	2.2.2.2	229	0x80000003	0x007E81	1
3.3.3.3	3.3.3.3	195	0x80000003	0x0079E9	2

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
12.1.1.1	1.1.1.1	946	0x80000001	0x004AD0

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
15.1.1.0	1.1.1.1	1058	0x80000001	0x009A8C

Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	557	0x80000002	0x00F113	1
5.5.5.5	5.5.5.5	558	0x80000002	0x00F5EB	1

Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
15.1.1.1	1.1.1.1	558	0x80000001	0x00B952

Summary Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
12.1.1.0	1.1.1.1	1063	0x80000001	0x00C168
13.1.1.0	1.1.1.1	1063	0x80000001	0x002DBC

Summary ASB Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
2.2.2.2	1.1.1.1	224	0x80000001	0x000B24
3.3.3.3	1.1.1.1	189	0x80000001	0x005596

Type-5 AS External Link States

Link ID	ADV Router	Age	Seq#	Checksum	Tag
24.1.1.0	2.2.2.2	230	0x80000001	0x005B23	0
24.1.1.0	3.3.3.3	195	0x80000001	0x003D3D	0
34.1.1.0	2.2.2.2	230	0x80000001	0x00D89B	0
34.1.1.0	3.3.3.3	195	0x80000001	0x00BAB5	0
172.16.1.0	2.2.2.2	230	0x80000001	0x001BBF	0
172.16.1.0	3.3.3.3	195	0x80000001	0x00FCD9	0
172.16.2.0	2.2.2.2	230	0x80000001	0x0010C9	0
172.16.2.0	3.3.3.3	195	0x80000001	0x00F1E3	0
172.16.3.0	2.2.2.2	230	0x80000001	0x0005D3	0
172.16.3.0	3.3.3.3	195	0x80000001	0x00E6ED	0

r1#

说明：因为 ASBR 就在区域 0 中，所以可以看出，区域 0 并没有类型 4 的 LSA，因为该区域与 ASBR 在同区域，只有与 ASBR 在不同区域才需要 LSA 4。

（20）查看 R5 的 LSA 数据库：

r5#

r5#sh ip ospf database

OSPF Router with ID (5.5.5.5) (Process ID 1)

Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	595	0x80000002	0x00F113	1
5.5.5.5	5.5.5.5	594	0x80000002	0x00F5EB	1

Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
15.1.1.1	1.1.1.1	595	0x80000001	0x00B952

Summary Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum

12.1.1.0	1.1.1.1	1101	0x80000001 0x00C168
13.1.1.0	1.1.1.1	1101	0x80000001 0x002DBC

Summary ASB Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
2.2.2.2	1.1.1.1	262	0x80000001	0x000B24
3.3.3.3	1.1.1.1	228	0x80000001	0x005596

Type-5 AS External Link States

Link ID	ADV Router	Age	Seq#	Checksum	Tag
24.1.1.0	2.2.2.2	268	0x80000001	0x005B23	0
24.1.1.0	3.3.3.3	234	0x80000001	0x003D3D	0
34.1.1.0	2.2.2.2	268	0x80000001	0x00D89B	0
34.1.1.0	3.3.3.3	234	0x80000001	0x00BAB5	0
172.16.1.0	2.2.2.2	268	0x80000001	0x001BBF	0
172.16.1.0	3.3.3.3	234	0x80000001	0x00FCD9	0
172.16.2.0	2.2.2.2	268	0x80000001	0x0010C9	0
172.16.2.0	3.3.3.3	234	0x80000001	0x00F1E3	0
172.16.3.0	2.2.2.2	268	0x80000001	0x0005D3	0
172.16.3.0	3.3.3.3	234	0x80000001	0x00E6ED	0

r5#

说明：因为 R5 所在的区域 1 与 ASBR 在不同区域，所有需要 ABR 单独发送 LSA 4 告知如何到达 ASBR 的 Router-ID。

3.测试 OSPF 汇总路由

（1）查看汇总前，R5 的路由表：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O IA 34.1.1.0 [110/68] via 15.1.1.1, 00:01:24, FastEthernet0/1

172.16.0.0/24 is subnetted, 3 subnets

O E2 172.16.1.0 [110/20] via 15.1.1.1, 00:01:24, FastEthernet0/1

O E2 172.16.2.0 [110/20] via 15.1.1.1, 00:01:24, FastEthernet0/1

O E2 172.16.3.0 [110/20] via 15.1.1.1, 00:01:24, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

O IA 24.1.1.0 [110/67] via 15.1.1.1, 00:01:24, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/66] via 15.1.1.1, 00:01:24, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Loopback192

13.0.0.0/24 is subnetted, 1 subnets

O IA 13.1.1.0 [110/65] via 15.1.1.1, 00:01:24, FastEthernet0/1

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r5#

说明：拥有到 172.16.1.0 /24，172.16.2.0 /24，172.16.3.0/24 的明细路由。

（2）在 R2 上汇总路由：

r2(config)#router ospf 1

r2(config-router)#summary-address 172.16.0.0 255.255.252.0

说明：将路由汇总为 172.16.0.0/22，因为是外部路由，所以必须在 ASBR 上进行汇总。

（3）查看汇总后，R5 的路由表：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O IA 34.1.1.0 [110/68] via 15.1.1.1, 00:03:51, FastEthernet0/1

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks

O E2 172.16.0.0/22 [110/20] via 15.1.1.1, 00:00:55, FastEthernet0/1

O E2 172.16.1.0/24 [110/20] via 15.1.1.1, 00:00:50, FastEthernet0/1

O E2 172.16.2.0/24 [110/20] via 15.1.1.1, 00:00:50, FastEthernet0/1

O E2 172.16.3.0/24 [110/20] via 15.1.1.1, 00:00:50, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

O IA 24.1.1.0 [110/67] via 15.1.1.1, 00:03:51, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

O IA 12.1.1.0 [110/66] via 15.1.1.1, 00:03:51, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Loopback192

13.0.0.0/24 is subnetted, 1 subnets

O IA 13.1.1.0 [110/65] via 15.1.1.1, 00:03:52, FastEthernet0/1

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r5#

说明：R5 拥有汇总路由与明细路由，因为有两个 ASBR，一个通告了汇总路由，一个通告了明细路由。

（4）在 R1 上过滤从一个区域发向其它区域的路由：

r1(config)#ip prefix-list NET172 deny 172.16.0.0/22

r1(config)#router ospf 1

r1(config-router)#area 0 filter-list prefix NET172 out

说明：当 R1 将区域 0 的路由发向其它任何区域时，除了 172.16.0.0/22，其它全部拒绝。

注：prefix-list 中被 deny 的条目，就是可以放行的路由，该方法只能过滤一个区域去其它区域的路由，只对其它路由器生效，自己没有效果。

（5）查看配置过滤后 R5 的路由表:

r5#sh ip route

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

Gateway of last resort is not set

     172.16.0.0/22 is subnetted, 1 subnets
O E2    172.16.0.0 [110/20] via 15.1.1.1, 00:00:09, FastEthernet0/1
C    192.168.1.0/24 is directly connected, Loopback192
C    192.168.2.0/24 is directly connected, Loopback192
C    192.168.3.0/24 is directly connected, Loopback192
     15.0.0.0/24 is subnetted, 1 subnets
C       15.1.1.0 is directly connected, FastEthernet0/1
r5#
```

说明：R5 只收到了除了 172.16.0.0/22，除了 172.16.0.0/22，其它全部被过滤了。

（6）查看 R1 的路由表:

r1#sh ip route

*Mar 1 00:54:06.991: %SYS-5-CONFIG_I: Configured from console by console

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/67] via 13.1.1.3, 00:00:35, Serial1/0

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks

O E2 172.16.0.0/22 [110/20] via 12.1.1.2, 00:00:35, FastEthernet0/0

O E2 172.16.1.0/24 [110/20] via 13.1.1.3, 00:00:35, Serial1/0

O E2 172.16.2.0/24 [110/20] via 13.1.1.3, 00:00:35, Serial1/0

O E2 172.16.3.0/24 [110/20] via 13.1.1.3, 00:00:35, Serial1/0

24.0.0.0/24 is subnetted, 1 subnets

O 24.1.1.0 [110/66] via 12.1.1.2, 00:00:35, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：虽然配置了区域过滤，但过滤只对其它路由器生效，而 R1 的路由条目没有受到任何影响。

（7）配置 R1 对自己过滤：

r1(config)#access-list 1 permit 172.16.0.0

r1(config)#router ospf 1

r1(config-router)#no area 0 filter-list prefix NET172 out

r1(config-router)#distribute-list 1 in serial 1/0

说明：去除之前的区域过滤，配置 distribute-list 对进入路由表的条目做过滤，该过滤方法对自己生效，只影响 LSA 进入自己的路由表，但 LSA 数据库中的条目仍然不变，照样会发给邻居。

（8）查看 R1 配置对自己过滤后的路由表：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O E2 34.1.1.0 [110/20] via 12.1.1.2, 00:00:24, FastEthernet0/0

172.16.0.0/22 is subnetted, 1 subnets

O E2 172.16.0.0 [110/20] via 12.1.1.2, 00:00:24, FastEthernet0/0

24.0.0.0/24 is subnetted, 1 subnets

O 24.1.1.0 [110/66] via 12.1.1.2, 00:00:25, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：distribute-list 对进入路由表的条目做了过滤。

（9）查看 R1 配置对自己过滤后的 LSA 数据库：

r1#sh ip ospf database

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

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	407	0x8000000D	0x005A98	3
2.2.2.2	2.2.2.2	1741	0x8000000A	0x003795	2
3.3.3.3	3.3.3.3	1744	0x80000009	0x00E243	3

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum

12.1.1.2 2.2.2.2 1742 0x80000003 0x000E06

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
15.1.1.0	1.1.1.1	821	0x80000006	0x009091

Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	361	0x8000000C	0x00B13D	1
5.5.5.5	5.5.5.5	116	0x8000000E	0x00AB20	1

Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
15.1.1.5	5.5.5.5	362	0x8000000B	0x006A7D

Summary Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
0.0.0.0	1.1.1.1	172	0x80000001	0x001B17

Type-5 AS External Link States

Link ID	ADV Router	Age	Seq#	Checksum Tag
24.1.1.0	3.3.3.3	35	0x80000003	0x00024E 0
34.1.1.0	2.2.2.2	407	0x80000006	0x002928 0
172.16.0.0	2.2.2.2	1518	0x80000001	0x0017C7 0
172.16.1.0	3.3.3.3	36	0x80000003	0x00C1EA 0
172.16.2.0	3.3.3.3	36	0x80000003	0x00B6F4 0
172.16.3.0	3.3.3.3	36	0x80000003	0x00ABFE 0

r1#

说明：配置 distribute-list 对进入路由表的条目做过滤，该过滤方法对自己生效，只影响 LSA 进入自己的路由表，但 LSA 数据库中的条目仍然不变，照样会发给邻居。

（10）查看 R1 配置对自己过滤后邻居 R5 的路由表：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O IA 34.1.1.0 [110/68] via 15.1.1.1, 00:01:27, FastEthernet0/1

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks

O E2 172.16.0.0/22 [110/20] via 15.1.1.1, 00:01:27, FastEthernet0/1

O E2 172.16.1.0/24 [110/20] via 15.1.1.1, 00:01:27, FastEthernet0/1

O E2 172.16.2.0/24 [110/20] via 15.1.1.1, 00:01:27, FastEthernet0/1

O E2 172.16.3.0/24 [110/20] via 15.1.1.1, 00:01:27, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

O IA 24.1.1.0 [110/67] via 15.1.1.1, 00:01:27, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/66] via 15.1.1.1, 00:01:27, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Loopback192

13.0.0.0/24 is subnetted, 1 subnets

O IA 13.1.1.0 [110/65] via 15.1.1.1, 00:01:28, FastEthernet0/1

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r5#

r5#

说明：配置 distribute-list 对进入路由表的条目做过滤后，只对自己的路由表生效，只邻居的路由表正常。

4.测试 OSPF 末节区域

（1）查看 R5 当前的路由表：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O IA 34.1.1.0 [110/68] via 15.1.1.1, 00:00:06, FastEthernet0/1

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks

O E2 172.16.0.0/22 [110/20] via 15.1.1.1, 00:00:06, FastEthernet0/1

O E2 172.16.1.0/24 [110/20] via 15.1.1.1, 00:00:06, FastEthernet0/1

O E2 172.16.2.0/24 [110/20] via 15.1.1.1, 00:00:06, FastEthernet0/1

O E2 172.16.3.0/24 [110/20] via 15.1.1.1, 00:00:06, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

O IA 24.1.1.0 [110/67] via 15.1.1.1, 00:00:06, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/66] via 15.1.1.1, 00:00:06, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Loopback192

13.0.0.0/24 is subnetted, 1 subnets

O IA 13.1.1.0 [110/65] via 15.1.1.1, 00:00:07, FastEthernet0/1

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r5#

说明：R5 同时拥有 OSPF 区域间的路由以及 OSPF 外部路由。

（2）配置区域 1 为 Stub Area（末节区域）

r1(config)#router ospf 1

r1(config-router)#area 1 stub

说明：在 R1 上配置区域 1 为 Stub Area（末节区域）。

（3）查看 R5 的 OSPF 邻居状态：

r5#sh ip ospf neighbor

r5#

说明：由于对端邻居配置了 Stub Area，所以 Hello 包中携带末节标签，导致邻居丢失。

（4）在 R5 上配置 Stub Area:

r5(config)#router ospf 1

r5(config-router)#area 1 stub

r5#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
1.1.1.1	1	FULL/BDR	00:00:37	15.1.1.1	FastEthernet0/1

r5#

说明：在双方都配置 Stub Area 后，邻居正常。

（5）查看 R5 配置 Stub Area 后的路由情况：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is 15.1.1.1 to network 0.0.0.0

34.0.0.0/24 is subnetted, 1 subnets

O IA 34.1.1.0 [110/68] via 15.1.1.1, 00:00:14, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

O IA 24.1.1.0 [110/67] via 15.1.1.1, 00:00:14, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/66] via 15.1.1.1, 00:00:14, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Loopback192

13.0.0.0/24 is subnetted, 1 subnets

O IA 13.1.1.0 [110/65] via 15.1.1.1, 00:00:14, FastEthernet0/1

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

O*IA 0.0.0.0/0 [110/2] via 15.1.1.1, 00:00:14, FastEthernet0/1

r5#

说明：Stub Area 阻拦了所有外部路由，但放行 OSPF 区域间的路由，由于没有了外部路由，ABR（R1）向域内发送了一条指向自己的默认路由，并且 Metric 值默认为 1 加到达 ABR 的接口 Cost。

（6）在 ABR 上将 Stub Area 改为 Totally Stub Area（完全末节区域）：

r1(config)#router ospf 1

r1(config-router)#area 1 stub no-summary

说明：Totally Stub Area（完全末节区域）只需要在 ABR 上配置即可，其它普通路由器只需要配置 Stub Area。

（7）查看配置 Totally Stub Area 后 R5 的路由情况：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is 15.1.1.1 to network 0.0.0.0

C 192.168.1.0/24 is directly connected, Loopback192

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

O*IA 0.0.0.0/0 [110/2] via 15.1.1.1, 00:00:11, FastEthernet0/1

r5#

说明：Totally Stub Area 阻拦了所有外部路由和 OSPF 区域间的路由，由于没有了外部路由，ABR（R1）向域内发送了一条指向自己的默认路由，并且 Metric 值默认为 1 加到达 ABR 的接口 Cost。

（8）配置区域 1 为 Not-so-Stubby Area（NSSA）：

r1(config)#router ospf 1

r1(config-router)#area 1 nssa

r5(config)#router ospf 1

r5(config-router)#area 1 nssa

r5(config-router)#redistribute connected subnets

说明：NSSA 区域的路由器可以向区域内重分布 OSPF 外部路由，NSSA 区域内将以 LSA 7 的形式向区域内重分布外部路由，但 LSA 7 只能在 NSSA 区域内传递。

（9）查看配置 Not-so-Stubby Area 后 R5 的路由情况：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O IA 34.1.1.0 [110/68] via 15.1.1.1, 00:00:22, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

O IA 24.1.1.0 [110/67] via 15.1.1.1, 00:00:22, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/66] via 15.1.1.1, 00:00:22, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Loopback192

13.0.0.0/24 is subnetted, 1 subnets

O IA 13.1.1.0 [110/65] via 15.1.1.1, 00:00:22, FastEthernet0/1

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r5#

说明：因为自己可以将外部路由重分布进 OSPF，并且自己也拥有 OSPF 区域间的路由，所以 ABR 并没有向区域内自动发送默认路由。

（10）查看配置 Totally Stub Area 后 R1 的路由情况：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/67] via 13.1.1.3, 00:01:44, Serial1/0

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks

O E2 172.16.0.0/22 [110/20] via 12.1.1.2, 00:00:57, FastEthernet0/0

O E2 172.16.1.0/24 [110/20] via 13.1.1.3, 00:00:57, Serial1/0

O E2 172.16.2.0/24 [110/20] via 13.1.1.3, 00:00:57, Serial1/0

O E2 172.16.3.0/24 [110/20] via 13.1.1.3, 00:00:57, Serial1/0

24.0.0.0/24 is subnetted, 1 subnets

O 24.1.1.0 [110/66] via 12.1.1.2, 00:01:44, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

O N2 192.168.1.0/24 [110/20] via 15.1.1.5, 00:00:57, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/0

O N2 192.168.2.0/24 [110/20] via 15.1.1.5, 00:00:58, FastEthernet0/1

O N2 192.168.3.0/24 [110/20] via 15.1.1.5, 00:00:58, FastEthernet0/1

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

r1#

说明：NSSA 区域以 LSA 7 的形式向区域内重分布外部路由，但 LSA 7 只能在 NSSA 区域内传递。

（11）手工在 ABR 上向 NSSA 区域发送默认路由：

r1(config)#router ospf 1

r1(config-router)#area 1 nssa default-information-originate

说明：必须在 NSSA 区域的 ABR 上向 NSSA 区域内发送默认路由。

（12）查看 R5 的路由表：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is 15.1.1.1 to network 0.0.0.0

34.0.0.0/24 is subnetted, 1 subnets

O IA 34.1.1.0 [110/68] via 15.1.1.1, 00:02:29, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

O IA 24.1.1.0 [110/67] via 15.1.1.1, 00:02:29, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O IA 12.1.1.0 [110/66] via 15.1.1.1, 00:02:29, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Loopback192

13.0.0.0/24 is subnetted, 1 subnets

O IA 13.1.1.0 [110/65] via 15.1.1.1, 00:02:29, FastEthernet0/1

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

O*N2 0.0.0.0/0 [110/1] via 15.1.1.1, 00:00:21, FastEthernet0/1

r5#

说明：手工配置 ABR 向 NSSA 区域发送默认路由成功。

（13）配置区域 1 为 Totally NSSA：

r1(config)#router ospf 1

r1(config-router)#area 1 nssa no-summary

说明：Totally NSSA 只需要在 ABR 上配置即可，其它普通路由器只需要配置 NSSA。

（14）查看配置 Totally NSSA 后 R5 的路由情况：

r5#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is 15.1.1.1 to network 0.0.0.0

C 192.168.1.0/24 is directly connected, Loopback192

C 192.168.2.0/24 is directly connected, Loopback192

C 192.168.3.0/24 is directly connected, Loopback192

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, FastEthernet0/1

O*IA 0.0.0.0/0 [110/2] via 15.1.1.1, 00:00:18, FastEthernet0/1

r5#

说明：Totally NSSA 阻拦了所有外部路由和 OSPF 区域间的路由，ABR（R1）自动向域内发送了一条指向自己的默认路由，并且 Metric 值默认为 1 加到达 ABR 的接口 Cost。

路由策略

路由重分布

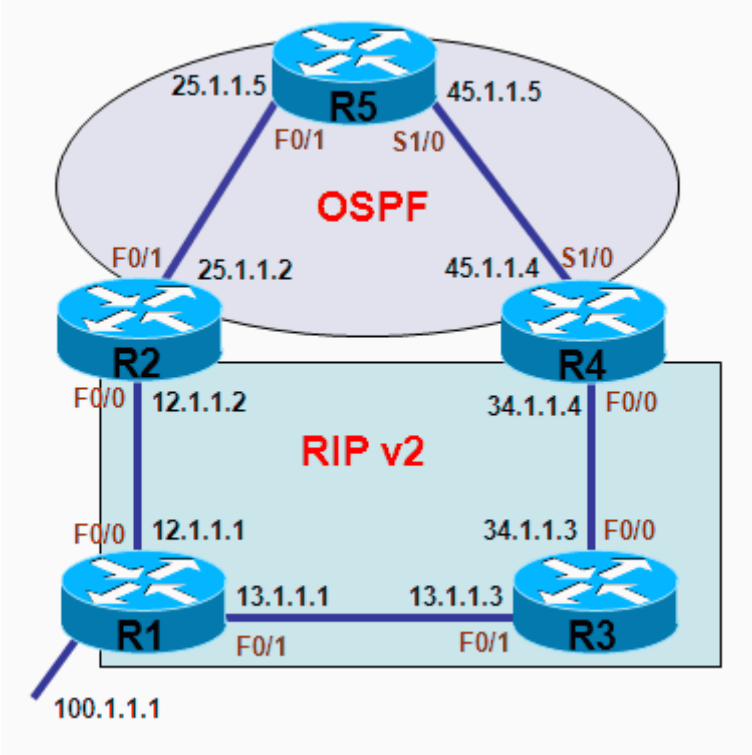
在同路由协议内，默认通过比较路由的 Metric 值来决定优先权，在不同路由协议之间，默认通过比较路由的 AD 值（Administrative Distance）来决定优先权。由于不同路由协议，Metric 值计算方法不同，所以在路由协议之间重分布时，需要额外考虑相互的 Metric 值如何理解。并且由于各协议 AD 值的不同，也需要额外考虑是否会有引起路由环路或次优路径的风险。

各协议默认的 AD 值（Administrative Distance）如下表：

Default Administrative Distances	
Connected	0
Static	1
eBGP	20
EIGRP (internal)	90
IGRP	10

	0
OSPF	110
IS-IS	115
RIP	120
EIGRP (external)	170
iBGP	200
EIGRP summary route	5

如下图所示：



RIP 区域的路由器 R1 将外部路由 100.1.1.0 重分布进 RIP，并且指定 metric 值为 5，在 RIP 范围内，到达 100.1.1.0 的 AD 值为 120，当 R2 将从 RIP 学习到的 100.1.1.0 重分布进 OSPF 区域后，路由传递到 R4 时，由于 OSPF 的 AD 值为 110，小于 RIP，最终造成 R4 到达 100.1.1.0 从更远的 OSPF 路径绕一圈；如果 R4 继续将路由 100.1.1.0 重分布进 RIP，并设置一个低于 5 的 Metric 值，又被 R3 学习到，结果又会造成 R3 经过 R4，经过 OSPF 区域，再绕回 RIP 到达目标 100.1.1.1，这就是由于不同协议的不同 AD 值造成的次优路径，甚至是路由环路，所以在协议之间配置重分布时，AD 值需要多加考虑和分析。

不同协议之间配置重分布时，还需要对 Metric 值做多加处理，如果由于 Metric 值错误或者 Metric 无法理解，将导致路由不可达或路由不可用。

当将外部路由重分布进路由协议时，各协议默认赋予外部路由的 Metric 值如下 ：

RIP

外部路由重分布进 RIP 时，默认 Metric 值为无穷大，即默认为不可用。

EIGRP

外部路由重分布进 EIGRP 时，默认 Metric 值为空，即默认没有 Metric 值，所以路由同样是不可用，但某些不同版本的 IOS，对此有所改进，并且是针对不同路由有不同操作，需要以实际 IOS 为准。（EIGRP 在 IOS 中是个例外）

OSPF

外部路由重分布进 OSPF 时，默认 Metric 值为 20，如果 BGP 重分布进 OSPF，默认 Metric 值为 1。

Route-Map

Route-Map 的使用相当广泛，而我们这里只提两个最主要的用途，那就是 Route-Map 在配置 Policy-Based Routing（PBR）所起的作用，以及 Route-Map 在重分布时所起的作用。

★需要说明，Route-Map 拥有许多特殊功能，我们这里并没有讲述完整的 Route-Map 功能，Policy-Based Routing 同样有着许多特殊功能，我们所提及的也不包含完整的 PBR 功能。

Route-Map 与 ACL（访问控制列表）有着一些共同之处，那就是一组 Route-Map 通过多条 Route-Map 语句组合而成，在配置时没有手工定义语句序号的情况下，序号默认以 10 递增，并且起始序号为 10，在系统检测 Route-Map 时，从上至下按顺序检测语句，只要检测到符合的语句，就立刻停止并执行当前动作，而不会继续检测剩下的所有语句。

因为 Route-Map 的语句总是以 10 递增，所以 Route-Map 允许在序号之间插入语句，方便修改。

在为 Route-Map 定义数据源或数据条件时，可以使用关键字 match 来匹配所需要的类型，可以匹配 ACL，prefix-list，接口，协议，以及数据包包头或内容等等，总之相当丰富，在定义匹配的条件之后，就可以定义需要对指定条件执行的动作，但如果不定义匹配条件，而只定义了执行动作，那么该 Route-Map 将对所有数据包生效。任何被上一条语句拒绝或没匹配到的所有数据，将被下一条匹配，如果还是没有匹配上，则继续往下，直到全部语句检测完毕为止。

例如：

假设源数据为 10.1.1.1，被以下 Route-MAP 检测：

★

Router(config)#access-list 10 permit host 10.1.1.1

Router(config)#route-map SET permit 10

Router(config-route-map)#match ip address 10

Router(config-route-map)#exit

Router(config)#route-map SET deny 20

Router(config-route-map)#exit

结果：被第一条允许匹配上，不再与第二条匹配，第二条无效，最后结果为放行。

★

Router(config)#access-list 10 permit host 10.1.1.1

Router(config)#route-map SET deny 10

Router(config-route-map)#exit

Router(config)#route-map SET permit 20

Router(config-route-map)#exit

Router(config)#

结果：被第一条拒绝匹配上，不再与第二条匹配，第二条无效，最后结果为拒绝。

★

Router(config)#access-list 10 permit host 10.1.1.1

Router(config)#route-map SET deny 10

Router(config-route-map)#exit

Router(config)#route-map SET deny 20

Router(config-route-map)#exit

结果：被第一条拒绝匹配上，不再与第二条匹配，第二条无效，最后结果为拒

绝。

★

Router(config)#access-list 20 permit host 20.1.1.1

Router(config)#route-map SET deny 10

Router(config-route-map)#match ip address 20

Router(config-route-map)#exit

Router(config)#route-map SET permit 20

Router(config-route-map)#exit

Router(config)#

结果：没有被第一条拒绝匹配上，继续与第二条匹配，第二条为放行，最后结果为放行。

★

Router(config)#route-map SET permit 10

Router(config-route-map)#exit

Router(config)#route-map SET deny 20

Router(config-route-map)#exit

结果：被第一条允许匹配上，不再与第二条匹配，第二条无效，最后结果为放行。

当 Route-Map 在 PBR 中使用时，被匹配到的流量将执行定义好的动作，而没有匹配到的流量，默认将按照常规来转发，所以在使用 Route-Map 做 PBR 时，没有被匹配到的流量将不受影响。

注：PBR 默认情况下，只对外部设备产生的流量生效，如果需要对配置 PBR 的设备自己产生的流量生效，需要全局输入命令：ip local policy route-map name。（将 name 替换为相应 Route-Map 的名字即可。）

当 Route-Map 在路由重分布中使用时，被匹配到并且允许的路由，将被重分布，被匹配到却被拒绝的路由将被丢弃而不重分布，如果路由没有被匹配，同样也不允许重分布，所以在使用 Route-Map 做路由重分布时，没有被匹配到的路由将全部拒绝。

路由控制

如果将不必要的路由发入错误的协议，可能导致路由环路或次优路径，因此，可以采用过滤手段将相应路由过滤掉，对于过滤路由，有以下特别注意的地方：

路由过滤可以通过 Distribute-List 来实现，Distribute-List 可以适用的协议和方向有：

RIP（距离矢量路由协议）

可适用于 in 方向和 out 方向。

EIGRP（距离矢量路由协议）

可适用于 in 方向和 out 方向。

OSPF（链路状态路由协议）

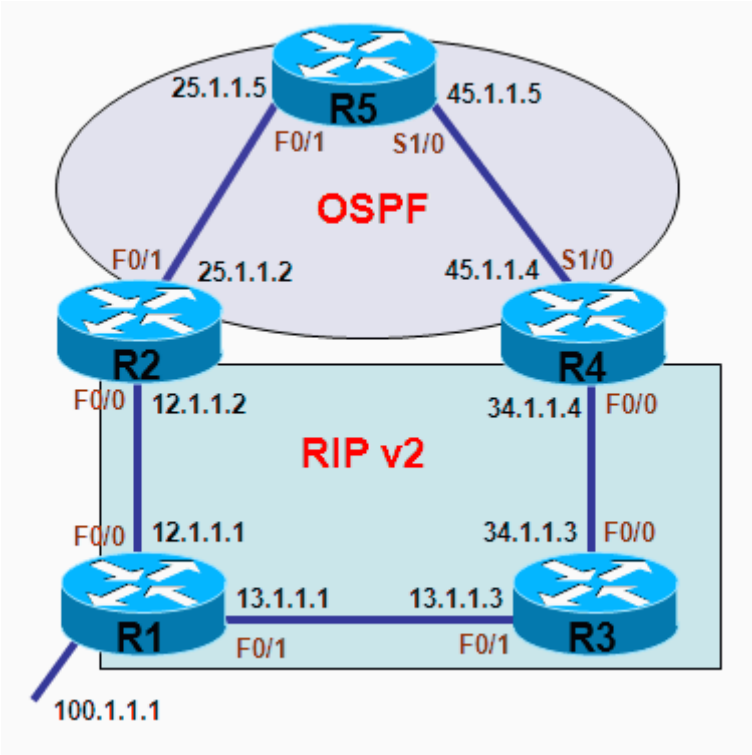
在 OSPF 本协议进程内，只适用于 in 方向，只对自己的路由表生效，无法影响邻居的路由表；在除 OSPF 之外的其它协议进程下，可用于 out 方向，在于将 OSPF 重分布进其它路由协议时做过滤，此过滤称为进程过滤，不仅适用于 OSPF 协议重分布进其它协议时适用，同样适用于其它协议重分布进 OSPF 或其它协议之间重分布。

Route-Map 可以单独用在路由重分布时控制和过滤路由。

除了在必要时，将路由过滤掉之外，当需要在不同协议或不同 AD 值之间调整路优选择优先权时，可以通过修改路由协议默认的 AD 值来实现，修改 AD 值可以是基于整个协议的修改，将对协议内所有路由条目生效，也可以对特定路由修改；如果只是需要对特定路由修改 AD 值，则需要调用 ACL 或 Prefix-Lix 来匹配特定路由，除此之外，还要定义路由去往目的地的下一跳地址，RIP 和 EIGRP 都需要在对特定路由调整 AD 值时定义路由下一跳地址，而对于 OSPF，其 LSA 中并没有明确写明去往目的地的下一跳地址，因为这个地址需要 OSPF 通过 LSA 计算后得知，所以在 OSPF 下对特定路由调整 AD 值时定义的路由下一跳地址为产生该 LSA 的

Router-ID。

如下图所示：



RIP 区域的路由器 R1 将外部路由 100.1.1.0 重分布进 RIP，并且指定 metric 值为 5，在 RIP 范围内，到达 100.1.1.0 的 AD 值为 120，当 R2 将从 RIP 学习到的 100.1.1.0 重分布进 OSPF 区域后，路由传递到 R4 时，由于 OSPF 的 AD 值为 110，小于 RIP，最终造成 R4 到达 100.1.1.0 从更远的 OSPF 路径绕一圈；如果 R4 继续将路由 100.1.1.0 重分布进 RIP，并设置一个低于 5 的 Metric 值，又被 R3 学习到，结果又会造成 R3 经过 R4，经过 OSPF 区域，再绕回 RIP 到达目标 100.1.1.1，这就是由于不同协议的不同 AD 值造成的次优路径，甚至是路由环路，所以在协议之间配置重分布时，AD 值需要多加考虑和分析。

重分布时的过滤方法除了以上的之外，还有 Tag 过滤技术，就是将相应路由打上 Tag，将该路由在重分布回原路由协议时，通过匹配 Tag 来拒绝掉，需要注意，Tag 过滤不支持 IGRP 和 RIP ver 1。

配置路由策略

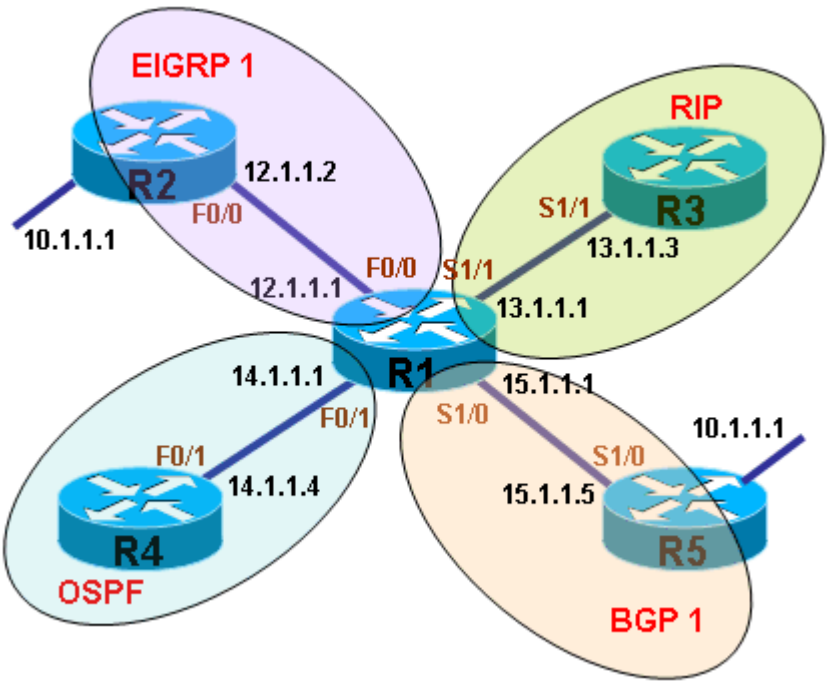
说明：实验配置共包含：

配置路由重分布

配置 PBR

配置路由控制

说明：以下图为例，配置路由重分布和 PBR：



1.配置网络基础环境

（1）配置 R1：

```
r1(config)#int f0/0
r1(config-if)#ip add 12.1.1.1 255.255.255.0
r1(config-if)#no sh

r1(config-if)#int f0/1
r1(config-if)#ip add 14.1.1.1 255.255.255.0
r1(config-if)#no sh

r1(config)#int s1/0
r1(config-if)#encapsulation frame-relay
r1(config-if)#no frame-relay inverse-arp
r1(config-if)#no arp frame-relay
r1(config-if)#ip address 15.1.1.1 255.255.255.0
r1(config-if)#no shutdown
r1(config-if)#frame-relay map ip 15.1.1.5 105 broadcast

r1(config)#int s1/1
r1(config-if)#encapsulation frame-relay
r1(config-if)#no frame-relay inverse-arp
```

r1(config-if)#no arp frame-relay

r1(config-if)#ip add 13.1.1.1 255.255.255.0

r1(config-if)#no shutdown

r1(config-if)# frame-relay map ip 13.1.1.3 113 broadcast

r1(config)#router eigrp 1

r1(config-router)#no auto-summary

r1(config-router)#network 12.1.1.1 0.0.0.0

r1(config)#router rip

r1(config-router)#version 2

r1(config-router)#no auto-summary

r1(config-router)#network 13.0.0.0

r1(config)#router ospf 1

r1(config-router)#router-id 1.1.1.1

r1(config-router)#network 14.1.1.1 0.0.0.0 area 0

r1(config)#router bgp 1

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

r1(config-router)#neighbor 15.1.1.5 remote-as 1

r1(config-router)#network 15.1.1.0 mask 255.255.255.0

说明：配置 R1 与 R2 之间运行 EIGRP 1，与 R3 之间运行 RIP，与 R4 之间运行 OSPF，与 R5 之间运行 BGP。

（2）配置 R2：

```
r2(config)#int f0/0
r2(config-if)#ip address 12.1.1.2 255.255.255.0
r2(config-if)#no sh

r2(config)#int loopback 10
r2(config-if)#ip add 10.1.1.1 255.255.255.0

r2(config)#router eigrp 1
r2(config-router)#no auto-summary
r2(config-router)#network 12.1.1.2 0.0.0.0
r2(config-router)#network 10.1.1.1 0.0.0.0
```

说明：在 R2 与 R1 之间配置 EIGRP 1，并将 10.1.1.0/24 放入 EIGRP 进程。

（3）配置 R3：

```
r3(config)#int s1/1
r3(config-if)#encapsulation frame-relay
```

r3(config-if)#no frame-relay inverse-arp

r3(config-if)#no arp frame-relay

r3(config-if)#ip add 13.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config-if)#fram map ip 13.1.1.1 311 broadcast

r3(config)#router rip

r3(config-router)#version 2

r3(config-router)#no auto-summary

r3(config-router)#network 13.0.0.0

说明：在 R3 与 R1 之间配置 RIP。

（4）配置 R4：

r4(config)#int f0/1

r4(config-if)#ip add 14.1.1.4 255.255.255.0

r4(config-if)#no sh

r4(config)#router ospf 1

r4(config-router)#router-id 4.4.4.4

r4(config-router)#network 14.1.1.4 0.0.0.0 area 0

说明：在 R4 与 R1 之间配置 OSPF。

（5）配置 R5：

r5(config)#int s1/0

r5(config-if)#encapsulation frame-relay

r5(config-if)#no frame-relay inverse-arp

r5(config-if)#no arp frame-relay

r5(config-if)#ip address 15.1.1.5 255.255.255.0

r5(config-if)#no sh

r5(config-if)#fram map ip 15.1.1.1 501 broadcast

r5(config)#int loopback 10

r5(config-if)#ip address 10.1.1.1 255.255.255.0

r5(config)#router bgp 1

r5(config-router)#bgp router-id 5.5.5.5

r5(config-router)#neighbor 15.1.1.1 remote-as 1

r5(config-router)#network 10.1.1.0 mask 255.255.255.0

说明：在 R5 与 R1 之间配置 BGP，并将 10.1.1.0/24 放入 BGP 进程。

2.测试路由重分布

（1）查看 R1 当前各协议的邻居状态：

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

r1#sh ip eigrp neighbors

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold	Uptime	SRTT	RTO	Q	Seq
			(sec)		(ms)		Cnt	Num
0	12.1.1.2	Fa0/0	13	00:04:08	153	1377	0	3

r1#sh ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
4.4.4.4	1	FULL/BDR	00:00:37	14.1.1.4	FastEthernet0/1

r1#

r1#sh ip bgp summary

BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 3, main routing table version 3

1 network entries using 129 bytes of memory

1 path entries using 52 bytes of memory

2/1 BGP path/bestpath attribute entries using 248 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 429 total bytes of memory

BGP activity 1/0 prefixes, 1/0 paths, scan interval 60 secs

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

15.1.1.5 4 1 6 5 3 0 0 00:01:42 1

r1#

说明：除 RIP 不建邻居之外，其它协议邻居正常。

（2）查看 R1 的路由表：

r1#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

D 10.1.1.0 [90/156160] via 12.1.1.2, 00:05:09, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, FastEthernet0/1

15.0.0.0/24 is subnetted, 1 subnets

C 15.1.1.0 is directly connected, Serial1/0

r1#

说明：R1 使用了 R2 发来的 10.1.1.0/24，但并没有使用 BGP 发来的，因为 iBGP 管理距离为 200，大于 EIGRP。

（3）在 R1 上将 EIGRP 学习到的路由重分布进 RIP：

r1(config)#router rip

r1(config-router)#redistribute eigrp 1

说明：将 EIGRP 1 的路由重分布进 RIP。

（4）查看 RIP 路由器 R3 的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/1

r3#

说明：RIP 中并没有收到 EIGRP 1 重分布进来的路由，因为在将路由重分布进 RIP 时，如果不指定 Metric 值，默认为无穷大，即不可用。

（5）修改 RIP 重分布时的默认 Metric 值：

r1(config)#router rip

r1(config-router)#default-metric 8

说明：在重分布的那台路由器上将默认重分布进 RIP 时的 Metric 值改为 8。

（6）再次查看 RIP 路由器 R3 的路由表：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

R 10.1.1.0 [120/8] via 13.1.1.1, 00:00:13, Serial1/1

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/8] via 13.1.1.1, 00:00:13, Serial1/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/1

r3#

说明：RIP 中成功收到从 EIGRP 重分布过来的路由，并且默认 Metric 值为修改后的 8。

（7）手工指定 EIGRP 重分布进 RIP 时的 Metric 值：

r1(config)#router rip

r1(config-router)#redistribute eigrp 1 metric 2

说明：手工指定 EIGRP 重分布进 RIP 时的 Metric 值为 2。

（8）再次查看 RIP 路由器 R3 的路由表：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
R       10.1.1.0 [120/2] via 13.1.1.1, 00:00:17, Serial1/1
     12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/2] via 13.1.1.1, 00:00:17, Serial1/1
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, Serial1/1
r3#
```

说明；Metric 值已被改为 2，说明手工指定的 Metric 值优先于默认 Metirc 值。

（9）重分布 EIGRP 进 OSPF：

r1(config)#router ospf 1

r1(config-router)#redistribute eigrp 1 subnets

说明：将 EIGRP 所有路由重分布进 OSPF。

（10）查看 OSPF 路由器 R4 的路由表：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

O E2 10.1.1.0 [110/20] via 14.1.1.1, 00:00:19, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 14.1.1.1, 00:00:19, FastEthernet0/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, FastEthernet0/1

r4#

说明：成功将 EIGRP 重分布进 OSPF，并且看到默认的类型为 2 类，并且默认 Metirc 值为 20。

（11）重分布 BGP 进 OSPF：

r1(config)#router bgp 1

r1(config-router)#bgp redistribute-internal

r1(config)#router ospf 1

r1(config-router)#redistribute bgp 1 subnets

说明：将 BGP 重分布进 OSPF。

（12）查看 OSPF 路由器 R4 的路由表：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

O E2 10.1.1.0 [110/20] via 14.1.1.1, 00:01:50, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 14.1.1.1, 00:01:50, FastEthernet0/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, FastEthernet0/1

15.0.0.0/24 is subnetted, 1 subnets

O E2 15.1.1.0 [110/1] via 14.1.1.1, 00:00:21, FastEthernet0/1

r4#

说明：成功将 BGP 重分布进 OSPF，并且看到默认的类型为 2 类，而 BGP 的默认 Metirc 值为 1，这是与其它路由协议不一样的地方。

（13）修改重分布进 OSPF 时的默认 Metric 值：

r1(config)#router ospf 1

r1(config-router)#default-metric 30

说明：将重分布进 OSPF 时的默认 Metric 值修改为 30。

（14）再次查看 OSPF 路由器 R4 的路由表：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

O E2 10.1.1.0 [110/30] via 14.1.1.1, 00:00:25, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/30] via 14.1.1.1, 00:00:25, FastEthernet0/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, FastEthernet0/1

15.0.0.0/24 is subnetted, 1 subnets

O E2 15.1.1.0 [110/30] via 14.1.1.1, 00:00:25, FastEthernet0/1

r4#

说明：所有重分布进 OSPF 的默认 Metirc 值全部变成了 30。

（15）重分布 OSPF 进 EIGRP：

r1(config)#router eigrp 1

r1(config-router)#redistribute ospf 1

说明：将 OSPF 重分布进 EIGRP：

（16）查看 EIGRP 路由器 R2 的路由表：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Loopback10

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

r2#

说明：重分布进 EIGRP 的路由为空，所以注意修改 Metric 值。

（17）修改重分布进 EIGRP 时的默认 Metric 值：

r1(config)#router eigrp 1

r1(config-router)#default-metric 10000 100 255 1 1500

说明：修改的 EIGRP 默认 Metric 值为 5 个 K 值所代表的值。

（18）再次查看 EIGRP 路由器 R2 的路由表：

r2#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Loopback10

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

14.0.0.0/24 is subnetted, 1 subnets

D EX 14.1.1.0 [170/284160] via 12.1.1.1, 00:02:58, FastEthernet0/0

r2#

说明：修改默认 Metric 值后，已经成功收到重分布进来的路由。

注：从实验中不难看出，所有重分布时修改的 Metric 值，全部都是在做重分布的那台路由器上修改的。

（19）重分布直连路由进 EIGRP 时使用 Route-MAP：

r1(config)#route-map NET15 permit 10

r1(config-route-map)#match interface serial 1/0

r1(config-route-map)#exit

r1(config)#router eigrp 1

r1(config-router)#redistribute connected route-map NET15

说明：通过 Route-Map 只匹配重分布接口 serial 1/0 进 EIGRP。

（20）查看 EIGRP 路由器 R2 的路由表：

```
r2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Loopback10
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, FastEthernet0/0
     15.0.0.0/24 is subnetted, 1 subnets
D EX    15.1.1.0 [170/2172416] via 12.1.1.1, 00:00:24, FastEthernet0/0
r2#
```

说明：只有与 Route-Map 中定义的接口相匹配的 serial 1/0 被重分布进了 EIGRP

（21）改变重分布直连路由进 EIGRP 时使用 Route-MAP：

r1(config)#route-map EIGRP deny 10

r1(config-route-map)#match interface s1/0

r1(config-route-map)#exit

r1(config)#route-map EIGRP permit 20

r1(config-route-map)#exit

r1(config)#router eigrp 1

r1(config-router)#no redistribute connected route-map NET15

r1(config-router)#redistribute connected route-map EIGRP

说明：修改 Route-Map 拒绝接口 serial 1/0，但允许其它所有接口进 EIGRP。

（22）查看 EIGRP 路由器 R2 的路由表：

r2#sh ip route

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

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

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

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

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

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

```
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Loopback10
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, FastEthernet0/0
     13.0.0.0/24 is subnetted, 1 subnets
D EX    13.1.1.0 [170/2172416] via 12.1.1.1, 00:00:11, FastEthernet0/0
     14.0.0.0/24 is subnetted, 1 subnets
D EX    14.1.1.0 [170/30720] via 12.1.1.1, 00:00:11, FastEthernet0/0
r2#
```

说明：修改 Route-Map 后，除了接口 serial 1/0 的路由没有之外，其它的都有。

3 测试 PBR

（1）将直连网络全部重分布进各协议：

```
r1(config)#router rip
r1(config-router)#redistribute connected
```

r1(config)#router bgp 1

r1(config-router)#redistribute connected

r1(config)#router ospf 1

r1(config-router)#redistribute connected subnets

r1(config)#router eigrp 1

r1(config-router)#redistribute connected

说明：将直连网络全部重分布进各协议，保证通畅。

（2）开启 R2 与 R5 和 VTY 线路：

r2(config)#line vty 0 4

r2(config-line)#no login

r2(config-line)#exit

r5(config)#line vty 0 4

r5(config-line)#no login

r5(c

说明：允许 R2 与 R5 的 VTY 使用空密码登陆。

（3）查看当前 R1 的路由表：

r1#sh ip route

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

```
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.1.1.0 [90/156160] via 12.1.1.2, 00:36:29, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
C       12.1.1.0 is directly connected, FastEthernet0/0
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, Serial1/1
     14.0.0.0/24 is subnetted, 1 subnets
C       14.1.1.0 is directly connected, FastEthernet0/1
     15.0.0.0/24 is subnetted, 1 subnets
C       15.1.1.0 is directly connected, Serial1/0
r1#
```

说明：当前 R1 选择从 R2 去往 10.1.1.0/24。

（4）测试 R1 到达 10.1.1.0/24 的路径：

r1#telnet 10.1.1.1

Trying 10.1.1.1 ... Open

r2>

r1#telnet 10.1.1.1 /source-interface serial 1/1

Trying 10.1.1.1 ... Open

r2

说明：R1 无论以什么形式，都是通过 R2 去往 10.1.1.0/24 的，因为路由表中就是这样。

（5）测试 R3 到达各网段的路径：

r3#telnet 10.1.1.1

Trying 10.1.1.1 ... Open

r2>

r3#telnet 15.1.1.5

Trying 15.1.1.5 ... Open

r5>

r3#telnet 12.1.1.2

Trying 12.1.1.2 ... Open

r2>

说明：R3 从 R2 到达 10.1.1.0/24，其它网段也均正常到达路由器。

（6）在 R1 连接 R3 的接口上配置 PBR：

r1(config)#access-list 133 permit ip any host 10.1.1.1

r1(config)#route-map SET permit 10

r1(config-route-map)#match ip address 133

r1(config-route-map)#set ip next-hop 15.1.1.5

r1(config-route-map)#exit

r1(config)#int s1/1

r1(config-if)#ip policy route-map SET

说明：配置 PBR 使任何去往 10.1.1.1/32 的流量都被强制发送到 R5。

（7）测试 R3 到达 10.1.1.1/32 的路径：

r3#telnet 10.1.1.1

Trying 10.1.1.1 ... Open

r5>

r5>

说明：R3 到达 10.1.1.1 已经从原来的 R2 被改到 R5，说明 PBR 生效。

（8）再测试 R3 到达其它网段的路径：

r3#telnet 12.1.1.2

Trying 12.1.1.2 ... Open

r2>

说明：到达其它网段的流量并没有执行 PBR，所以没有被发到 R5，只是去往 10.1.1.1/32 的流量被 PBR 生效。

（9）测试 R1 自己到达 10.1.1.1/32 的路径：

r1#telnet 10.1.1.1 /source-interface serial 1/1

Trying 10.1.1.1 ... Open

r2>

说明：自己还是通过 R2 到达 10.1.1.1/32，路径和配置 PBR 之前没有变化，说明 PBR 不对自己产生的流量生效。

（10）配置 PBR 对 R1 自己产生的流量：

r1(config)#ip local policy route-map SET

说明：允许当前配置的 PBR 对自己产生的流量生效。

（10）再次测试 R1 自己到达 10.1.1.1/32 的路径：

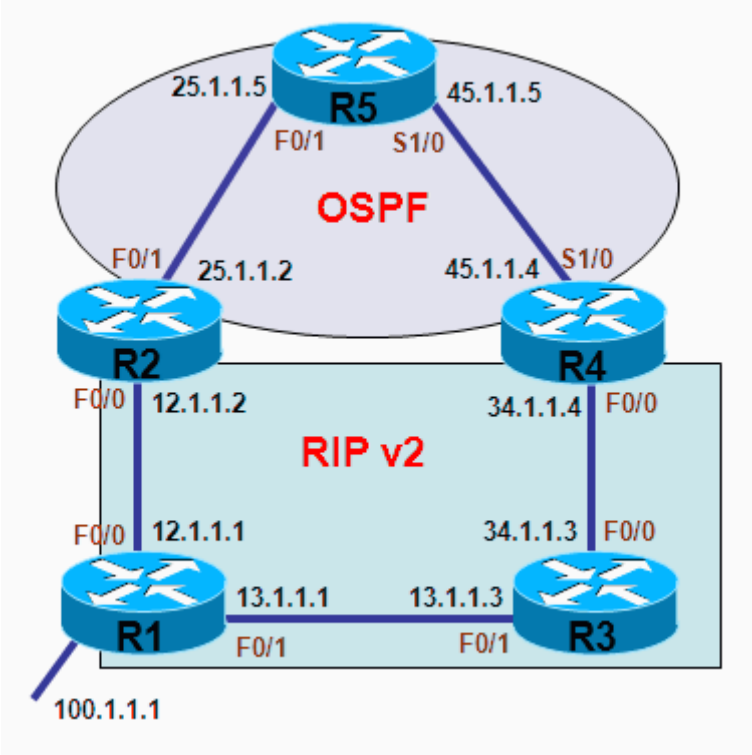
r1#telnet 10.1.1.1 /source-interface serial 1/1

Trying 10.1.1.1 ... Open

r5>

说明：修改后，PBR 已对自己产生的流量生效。

配置路由控制



说明：以上图为例，配置路由控制，解决路由环路和次优路径，其中包含配置 Distribute-List 过滤，修改特定路由 AD 值，进程过滤，以及 Tag 过滤，为了方便起见，我们只以 100.1.1.0 的路由选择为例，并且 100.1.1.0 被重分布进 RIP。

1.配置网络基础环境

（1）配置 R1：

r1(config)#int loopback 100

r1(config-if)#ip add 100.1.1.1 255.255.255.0

r1(config)#int f0/0

r1(config-if)#ip add 12.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#int f0/1

r1(config-if)#ip address 13.1.1.1 255.255.255.0

r1(config-if)#no sh

r1(config)#router rip

r1(config-router)#ver 2

r1(config-router)#no auto-summary

r1(config-router)#network 12.0.0.0

r1(config-router)#network 13.0.0.0

r1(config-router)#redistribute connected metric 5

说明：在 R1 上配置了 100.1.1.0/24，12.1.1.0/24，13.1.1.0/24，在 12.1.1.0/24，13.1.1.0/24 运行 RIP，并将 100.1.1.0/24 重分布进 RIP，重分布时的 Metric 值为 5。

（2）配置 R2：

r2(config)#int f0/0

r2(config-if)#ip add 12.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#int f0/1

r2(config-if)#ip address 25.1.1.2 255.255.255.0

r2(config-if)#no sh

r2(config)#router rip

r2(config-router)#version 2

r2(config-router)#no auto-summary

r2(config-router)#network 12.0.0.0

r2(config)#router ospf 1

r2(config-router)#router-id 2.2.2.2

r2(config-router)#network 25.1.1.2 0.0.0.0 area 0

说明：在 R2 上配置了 12.1.1.0/24，25.1.1.0/24，在 12.1.1.0/24 运行 RIP，在 25.1.1.0/24 运行 OSPF。

（3）配置 R3：

r3(config)#int f0/0

r3(config-if)#ip address 34.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#int f0/1

r3(config-if)#ip address 13.1.1.3 255.255.255.0

r3(config-if)#no sh

r3(config)#router rip

r3(config-router)#ver 2

r3(config-router)#no auto-summary

r3(config-router)#network 13.0.0.0

r3(config-router)#network 34.0.0.0

说明：在 R3 上配置了 13.1.1.0/24，34.1.1.0/24，全部网段运行 RIP。

（4）配置 R4：

r4(config)#int f0/0

r4(config-if)#ip add 34.1.1.4 255.255.255.0

r4(config-if)#no sh

r4(config)#int s1/0

r4(config-if)#encapsulation frame-relay

r4(config-if)#no frame-relay inverse-arp

r4(config-if)#no arp frame-relay

r4(config-if)#ip address 45.1.1.4 255.255.255.0

r4(config-if)#no shutdown

r4(config-if)#frame-relay map ip 45.1.1.5 405 broadcast

r4(config-if)#ip ospf network point-to-point

r4(config)#router rip

r4(config-router)#version 2

r4(config-router)#no auto-summary

r4(config-router)#network 34.0.0.0

r4(config)#router ospf 1

r4(config-router)#router-id 4.4.4.4

r4(config-router)#network 45.1.1.4 0.0.0.0 area 0

说明：在 R4 上配置了 34.1.1.0/24，45.1.1.0/24，在 34.1.1.0/24 运行 RIP，在 45.1.1.0/24 运行 OSPF。

（5）配置 R5：

r5(config)#int f0/1

r5(config-if)#ip add 25.1.1.5 255.255.255.0

r5(config-if)#no sh

r5(config)#int s1/0

r5(config-if)#encapsulation frame-relay

r5(config-if)#no frame-relay inverse-arp

r5(config-if)#no arp frame-relay

r5(config-if)#ip add 45.1.1.5 255.255.255.0

r5(config-if)#no sh

r5(config-if)#frame-relay map ip 45.1.1.4 504 broadcast

r5(config-if)#ip ospf network point-to-point

r5(config)#router ospf 1

r5(config-router)#router-id 5.5.5.5

r5(config-router)#network 25.1.1.5 0.0.0.0 area 0

r5(config-router)#network 45.1.1.5 0.0.0.0 area 0

说明：在 R5 上配置了 25.1.1.0/24，45.1.1.0/24，全部网段运行 OSPF。

（6）查看 R4 去往 100.1.1.0/24 的路由情况：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/6] via 34.1.1.3, 00:00:06, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:01:14, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/2] via 34.1.1.3, 00:00:06, FastEthernet0/0

13.0.0.0/24 is subnetted, 1 subnets

R 13.1.1.0 [120/1] via 34.1.1.3, 00:00:06, FastEthernet0/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

说明：R4 通过 RIP 内部去往 100.1.1.0/24，路径正常。

（7）查看 R3 去往 100.1.1.0/24 的路由情况:

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
C       34.1.1.0 is directly connected, FastEthernet0/0
     100.0.0.0/24 is subnetted, 1 subnets
R       100.1.1.0 [120/5] via 13.1.1.1, 00:00:09, FastEthernet0/1
     12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 13.1.1.1, 00:00:09, FastEthernet0/1
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, FastEthernet0/1
r3#
```

说明：R3 通过 RIP 内部去往 100.1.1.0/24，路径正常。

（8）跟踪 R3 去往 100.1.1.0/24 的路径情况：

r3#traceroute 100.1.1.1

Type escape sequence to abort.

Tracing the route to 100.1.1.1

1 13.1.1.1 224 msec * 80 msec

r3#

说明：R3 从 13.1.1.1 直接去往 100.1.1.0/24，路径正常。

（9）在 R2 与 R4 上配置 RIP 与 OSPF 的双向重分布：

r2(config)#router rip

r2(config-router)#redistribute ospf 1 metric 1

r2(config)#router ospf 1

r2(config-router)#redistribute rip subnets

r4(config)#router rip

r4(config-router)#redistribute ospf 1 metric 1

r4(config)#router ospf 1

r4(config-router)#redistribute rip subnets

说明：在 R2 与 R4 上配置 RIP 与 OSPF 的双向重分布，并且在 R4 将 OSPF 重分布进 RIP 时，Metirc 值为 1，小于 R1 将 100.1.1.0 重分布进 RIP 时的值。

（10）查看配置重分布后 R4 的路由表情况：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

O E2 100.1.1.0 [110/20] via 45.1.1.5, 00:01:19, Serial1/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:01:19, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:01:19, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:01:19, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

说明：因为 RIP 区域的路由器 R1 将外部路由 100.1.1.0 重分布进 RIP 后，R2 再将 100.1.1.0 重分布进 OSPF 区域，路由传递到 R4 时，由于 OSPF 的 AD 值为 110，小于 RIP，最终造成 R4 到达 100.1.1.0 从更远的 OSPF 路径绕一圈。

（11）查看配置重分布后 R3 的路由表情况：

r3#sh ip route

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

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

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

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

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

```
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
C       34.1.1.0 is directly connected, FastEthernet0/0
     100.0.0.0/24 is subnetted, 1 subnets
R       100.1.1.0 [120/1] via 34.1.1.4, 00:00:13, FastEthernet0/0
     25.0.0.0/24 is subnetted, 1 subnets
R       25.1.1.0 [120/1] via 34.1.1.4, 00:00:13, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 34.1.1.4, 00:00:13, FastEthernet0/0
                 [120/1] via 13.1.1.1, 00:00:18, FastEthernet0/1
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, FastEthernet0/1
     45.0.0.0/24 is subnetted, 1 subnets
R       45.1.1.0 [120/1] via 34.1.1.4, 00:00:13, FastEthernet0/0
r3#

r3#traceroute 100.1.1.1
```

Type escape sequence to abort.

Tracing the route to 100.1.1.1

1 34.1.1.4 64 msec 44 msec 60 msec

2 45.1.1.5 256 msec 104 msec 140 msec

3 25.1.1.2 156 msec 124 msec 76 msec

4 12.1.1.1 232 msec * 284 msec

r3#

说明：因为 RIP 区域的路由器 R1 将外部路由 100.1.1.0 重分布进 RIP 后，R2 再将 100.1.1.0 重分布进 OSPF 区域，路由传递到 R4 时，由于 OSPF 的 AD 值为 110，小于 RIP，最终造成 R4 到达 100.1.1.0 从更远的 OSPF 路径绕一圈，当 R4 再将 OSPF 重分布进 RIP 时，Metirc 值为 1，小于 R1 将 100.1.1.0 重分布进 RIP 时的值，所以 R3 也选择从 OSPF 的路径去往 100.1.1.0/24。

通过分析可以得知，造成次优路径的原因为，不应该配置 OSPF 重分布进 RIP 时，再将 100.1.1.0/24 返回 RIP，或者干脆在重分布 RIP 去往 OSPF 时，不将 100.1.1.0/24 放过去，但现在可以多种方法解决该问题。

2.配置 Distribute-List 过滤

（1）在 R4 上配置 Distribute-List 过滤:

r4(config)#access-list 4 deny 100.1.1.0

r4(config)#access-list 4 permit any

r4(config)#router ospf 1

r4(config-router)#distribute-list 4 in s1/0

说明：在 R4 上配置 Distribute-List 过滤，拒绝再次 OSPF 接收 100.1.1.0/24 的路由，从而防止次优路径。

（2）查看 R4 配置 Distribute-List 过滤后的路由表情况：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/6] via 34.1.1.3, 00:00:03, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:00:32, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:00:32, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:00:32, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

说明：因为 R4 拒绝再次 OSPF 接收 100.1.1.0/24 的路由，所以 R4 选择从内部 RIP 去往 100.1.1.0/24，配置 Distribute-List 过滤后，路径选择正常。

（3）查看 R3 配置 Distribute-List 过滤后的路径情况：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/5] via 13.1.1.1, 00:00:21, FastEthernet0/1

25.0.0.0/24 is subnetted, 1 subnets

R 25.1.1.0 [120/1] via 34.1.1.4, 00:00:14, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/1] via 34.1.1.4, 00:00:14, FastEthernet0/0

[120/1] via 13.1.1.1, 00:00:21, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

R 45.1.1.0 [120/1] via 34.1.1.4, 00:00:14, FastEthernet0/0

r3#

r3#traceroute 100.1.1.1

Type escape sequence to abort.

Tracing the route to 100.1.1.1

1 13.1.1.1 224 msec * 80 msec

r3#

说明：因为 R4 拒绝再次 OSPF 接收 100.1.1.0/24 的路由，所以 R3 选择从内部 RIP 去往 100.1.1.0/24，配置 Distribute-List 过滤后，路径选择正常。

3.修改 AD 值控制路径走向

说明：由于 100.1.1.0/24 在 R4 从 RIP 学习到的 AD 值为 120，而从 OSPF 学习到的 AD 值为 110，所以造成 R4 从 OSPF 去往 100.1.1.0/24，所以通过修改 OSPF 的 AD 值大于 RIP，便可控制路径走向，同样可以控制 AD 值来控制 R3 去往 100.1.1.0/24 的路径走向。

（1）恢复网络至初始路径状态：

r4(config-router)#no distribute-list 4 in s1/0

R3：

r3#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/1] via 34.1.1.4, 00:00:04, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

R 25.1.1.0 [120/1] via 34.1.1.4, 00:00:04, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/1] via 34.1.1.4, 00:00:04, FastEthernet0/0

[120/1] via 13.1.1.1, 00:00:19, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

R 45.1.1.0 [120/1] via 34.1.1.4, 00:00:04, FastEthernet0/0

r3#

R4：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

O E2 100.1.1.0 [110/20] via 45.1.1.5, 00:01:52, Serial1/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:03:49, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:03:49, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:03:49, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

说明：R3 和 R4 都选择从 OSPF 去往目标 100.1.1.0/24，此路径为次优路径。

（2）在 R3 上修改 AD 值控制路径走向：

r3(config)#access-list 33 permit 100.1.1.0

r3(config)#router rip

r3(config-router)#distance 109 13.1.1.1 0.0.0.0 33

说明：在 R3 上修改从 13.1.1.1 去往 100.1.1.0/24 的 AD 值为 109，大于从 R4 去往 100.1.1.0/24，配置中的地址为下一跳地址。

（3）查看修改 AD 值后 R3 的路径情况：

r3#sh ip route

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

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

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

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

```
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
C       34.1.1.0 is directly connected, FastEthernet0/0
     100.0.0.0/24 is subnetted, 1 subnets
R       100.1.1.0 [109/5] via 13.1.1.1, 00:00:23, FastEthernet0/1
     25.0.0.0/24 is subnetted, 1 subnets
R       25.1.1.0 [120/1] via 34.1.1.4, 00:00:11, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 34.1.1.4, 00:00:11, FastEthernet0/0
                 [120/1] via 13.1.1.1, 00:00:23, FastEthernet0/1
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, FastEthernet0/1
     45.0.0.0/24 is subnetted, 1 subnets
R       45.1.1.0 [120/1] via 34.1.1.4, 00:00:11, FastEthernet0/0
r3#
```

说明：修改 AD 值后，R3 选择从 13.1.1.1 直接去往 100.1.1.0/24。

（4）在 R4 上查看 100.1.1.0在 OSPF 中通告的 Router-ID：

r4#sh ip ospf database external 100.1.1.0

OSPF Router with ID (4.4.4.4) (Process ID 1)

Type-5 AS External Link States

Routing Bit Set on this LSA

LS age: 155

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 100.1.1.0 (External Network Number)

Advertising Router: 2.2.2.2

LS Seq Number: 80000001

Checksum: 0x7BB6

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 0.0.0.0

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

External Route Tag: 0

r4#

说明：因为在 OSPF 修改特定路由的 AD 值，下一跳为通告该 LSA 的 Router-ID。

（5）在 R4 上修改 AD 值控制路径走向：

r4(config)#access-list 44 permit 100.1.1.0

r4(config)#router ospf 1

r4(config-router)#distance 121 2.2.2.2 0.0.0.0 44

说明：在 R4 上修改从 OSPF 去往 100.1.1.0/24 的 AD 值为 121，大于从 RIP 内部去往 100.1.1.0/24，置中的地址为通告该 LSA 的 Router-ID。

（6）查看修改 AD 值后 R4 的路径情况：

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/6] via 34.1.1.3, 00:00:11, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:00:22, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:00:22, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:00:22, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

说明：修改 AD 值后，R4 选择从 RIP 内部去往 100.1.1.0/24，

4.通过 Route-Map 过滤路由

说明：在 R2 上将 RIP 重分布进 OSPF 时，过滤掉 100.1.1.0/24

（1）恢复网络至初始路径状态：

r4(config)#router ospf 1

r4(config-router)#no distance 121 2.2.2.2 0.0.0.0 44

R4:

r4#sh ip route

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

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

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

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

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

O E2 100.1.1.0 [110/20] via 45.1.1.5, 00:00:06, Serial1/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:01:03, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:01:03, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:01:03, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

R3:

r3(config)#router rip

r3(config-router)#no distance 109 13.1.1.1 0.0.0.0 33

r3#sh ip route

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

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

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

```
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
C       34.1.1.0 is directly connected, FastEthernet0/0
     100.0.0.0/24 is subnetted, 1 subnets
R       100.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0
     25.0.0.0/24 is subnetted, 1 subnets
R       25.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0
                 [120/1] via 13.1.1.1, 00:00:03, FastEthernet0/1
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, FastEthernet0/1
     45.0.0.0/24 is subnetted, 1 subnets
R       45.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0
r3#
```

说明：R3 和 R4 都选择从 OSPF 去往目标 100.1.1.0/24，此路径为次优路径。

（2）在 R2 上通过 Route-Map 过滤 100.1.1.0/24:

r2(config)#access-list 22 permit 100.1.1.0

r2(config)#route-map NET100 deny 10

r2(config-route-map)#exit

r2(config-route-map)#match ip address 22

r2(config-route-map)#exit

r2(config)#route-map NET100 permit 20

r2(config-route-map)#exit

r2(config)#router ospf 1

r2(config-router)#redistribute rip

r2(config-router)#redistribute rip subnets route-map NET100

说明：通过 Route-Map 在 R2 上将 RIP 重分布进 OSPF 时，过滤掉 100.1.1.0/24

（3）查看使用 Route-map 过滤后 R4 的路径情况：

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/6] via 34.1.1.3, 00:00:15, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:03:42, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:03:42, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:03:42, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

说明：从 OSPF 过滤掉 100.1.1.0/24 后，R4 选择从 RIP 内部去往 100.1.1.0/24，

（4）查看使用 Route-map 过滤后 R3 的路径情况：

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
C       34.1.1.0 is directly connected, FastEthernet0/0
     100.0.0.0/24 is subnetted, 1 subnets
R       100.1.1.0 [120/5] via 13.1.1.1, 00:00:17, FastEthernet0/1
     25.0.0.0/24 is subnetted, 1 subnets
R       25.1.1.0 [120/1] via 34.1.1.4, 00:00:11, FastEthernet0/0
     12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 34.1.1.4, 00:00:11, FastEthernet0/0
```

[120/1] via 13.1.1.1, 00:00:18, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

R 45.1.1.0 [120/1] via 34.1.1.4, 00:00:11, FastEthernet0/0

r3#

说明：从 OSPF 过滤掉 100.1.1.0/24 后，R3 选择从 RIP 内部去往 100.1.1.0/24，

（5）查看使用 Route-map 过滤后 R5 的 LSA 数据库情况：

r5#sh ip ospf database external 100.1.1.0

OSPF Router with ID (5.5.5.5) (Process ID 1)

Type-5 AS External Link States

Routing Bit Set on this LSA

LS age: 65

Options: (No TOS-capability, DC)

LS Type: AS External Link

Link State ID: 100.1.1.0 (External Network Number)

Advertising Router: 4.4.4.4

LS Seq Number: 80000001

Checksum: 0x3FEA

Length: 36

Network Mask: /24

Metric Type: 2 (Larger than any link state path)

TOS: 0

Metric: 20

Forward Address: 0.0.0.0

External Route Tag: 0

r5#

说明：在 R2 上配置 Route-Map 过滤掉 100.1.1.0/24 后，R5 并没有再从 R2 收到关于 100.1.1.0/24 的 LSA。

5.通过配置进程过滤路由控制路径走向

说明：在 R4 将 OSPF 路由重分布进 RIP 时，可以在进程中过滤掉 100.1.1.0/24 返回 RIP。

（1）恢复网络至初始路径状态：

r2(config)#router ospf 1

r2(config-router)#no redistribute rip subnets route-map NET100

r2(config-router)#redistribute rip subnets

R4:

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

O E2 100.1.1.0 [110/20] via 45.1.1.5, 00:00:18, Serial1/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:00:37, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:00:37, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:00:37, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

R3:

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

R 25.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0

[120/1] via 13.1.1.1, 00:00:07, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

R 45.1.1.0 [120/1] via 34.1.1.4, 00:00:03, FastEthernet0/0

r3#

说明：R3 和 R4 都选择从 OSPF 去往目标 100.1.1.0/24，此路径为次优路径。

（2）在 R4 将 OSPF 路由重分布进 RIP 时，在进程中过滤掉 100.1.1.0/24返回 RIP：

r4(config)#access-list 4 deny 100.1.1.0

r4(config)#access-list 4 permit any

r4(config)#router rip

r4(config-router)#distribute-list 4 out ospf 1

说明：在 R4 将 OSPF 路由重分布进 RIP 时，在进程中过滤掉 100.1.1.0/24 返回 RIP。

（3）查看配置进程过滤后 R3 的路径情况：

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/5] via 13.1.1.1, 00:00:07, FastEthernet0/1

25.0.0.0/24 is subnetted, 1 subnets

R 25.1.1.0 [120/1] via 34.1.1.4, 00:00:07, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/1] via 34.1.1.4, 00:00:07, FastEthernet0/0

[120/1] via 13.1.1.1, 00:00:07, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

R 45.1.1.0 [120/1] via 34.1.1.4, 00:00:07, FastEthernet0/0

r3#

说明：因为 R3 已经不能从 R4 收到关于 100.1.1.0/24 的路由，所以 R3 直接选择从 R1 去往目标 100.1.1.0/24。

（4）查看配置进程过滤后 R4 的路径情况：

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

O E2 100.1.1.0 [110/20] via 45.1.1.5, 00:00:27, Serial1/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 45.1.1.5, 00:00:27, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O E2 12.1.1.0 [110/20] via 45.1.1.5, 00:00:27, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

O E2 13.1.1.0 [110/20] via 45.1.1.5, 00:00:27, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, Serial1/0

r4#

说明：因为过滤是对其它路由器生效，所以 R4 的路径保持不变。

6.配置 Tag 过滤技术控制路径走向

说明：Tag 过滤技术就是将各自协议的路由都打上 Tag，在将该路由在重分布回原路由协议时，通过匹配 Tag 来拒绝掉路由返回，下面我们将 OSPF 自己的路由打上 Tag 110，将 RIP 自己的路由打上 Tag 120，并且在重分布时，通过匹配 Tag 拒绝原本属于自己的路由再返回。

（1）恢复网络至初始路径状态：

r4(config)#router rip

r4(config-router)#no distribute-list 4 out ospf 1

R3：

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/1] via 34.1.1.4, 00:00:00, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

R 25.1.1.0 [120/1] via 34.1.1.4, 00:00:00, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/1] via 34.1.1.4, 00:00:00, FastEthernet0/0

[120/1] via 13.1.1.1, 00:00:25, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

R 45.1.1.0 [120/1] via 34.1.1.4, 00:00:00, FastEthernet0/0

r3#

说明：R3 选择从 OSPF 去往目标 100.1.1.0/24，此路径为次优路径。

（2）在 R2 上配置 Tag 过滤技术：

r2(config)#route-map R2O deny 10

r2(config-route-map)#match tag 110

r2(config-route-map)#exit

r2(config)#route-map R2O permit 20

r2(config-route-map)#set tag 120

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

r2(config-route-map)#exit

r2(config)#route-map O2R deny 10

r2(config-route-map)#match tag 120

r2(config-route-map)#exit

r2(config)#route-map O2R permit 20

r2(config-route-map)#set tag 110

r2(config-route-map)#exit

r2(config)#router rip

r2(config-router)#redistribute ospf 1 metric 1 route-map O2R

r2(config)#router ospf 1

r2(config-router)#redistribute rip subnets route-map R2O

说明：在双向重分布时，将 OSPF 自己的路由打上 Tag 110，将 RIP 自己的路由打上 Tag 120，并且通过匹配 Tag 拒绝原本属于自己的路由再返回，也就是 OSPF 重分布进 RIP 时，拒绝 Tag 为 120 的路由返回，RIP 重分布进 OSPF 时，拒绝 Tag 为 110 的路由返回，因为它们原本就是自己的路由。

（3）在 R4 上配置 Tag 过滤技术：

r4(config)#route-map R2O deny 10

```
r4(config-route-map)#match tag 110

r4(config-route-map)#exit

r4(config)#route-map R2O permit 20

r4(config-route-map)#set tag 120

r4(config-route-map)#exit

r4(config)#route-map O2R deny 10

r4(config-route-map)#match tag 120

r4(config-route-map)#exit

r4(config)#route-map O2R permit 20

r4(config-route-map)#set tag 110

r4(config-route-map)#exit

r4(config)#router rip

r4(config-router)#redistribute ospf 1 metric 1 route-map O2R

r4(config-router)#exi


r4(config)#router ospf 1

r4(config-router)#redistribute rip subnets route-map R2O
```

说明：配置同 R2，在双向重分布时，将 OSPF 自己的路由打上 Tag 110，将 RIP 自己的路由打上 Tag 120，并且通过匹配 Tag 拒绝原本属于自己的路由再返回，也就是 OSPF 重分布进 RIP 时，拒绝 Tag 为 120 的路由返回，RIP 重分布进 OSPF 时，拒绝 Tag 为 110 的路由返回，因为它们原本就是自己的路由。

（4）查看配置 Tag 过滤后 R3 的路径情况：

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

100.0.0.0/24 is subnetted, 1 subnets

R 100.1.1.0 [120/5] via 13.1.1.1, 00:00:20, FastEthernet0/1

25.0.0.0/24 is subnetted, 1 subnets

R 25.1.1.0 [120/1] via 34.1.1.4, 00:00:21, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

R 12.1.1.0 [120/1] via 13.1.1.1, 00:00:20, FastEthernet0/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

R 45.1.1.0 [120/1] via 34.1.1.4, 00:00:21, FastEthernet0/0

r3#

说明：配置 Tag 过滤掉，R3 不再从 OSPF 学习到源于自身的路由，所以 R3 选择从 R1 直接去往目标 100.1.1.0/24。

（5）在 R1 上查看配置 Tag 过滤后，源于 OSPF 路由的 Tag 情况：

r3#sh ip route 25.1.1.0

Routing entry for 25.1.1.0/24

Known via "rip", distance 120, metric 1

Tag 110

Redistributing via rip

Last update from 34.1.1.4 on FastEthernet0/0, 00:00:01 ago

Routing Descriptor Blocks:

* 34.1.1.4, from 34.1.1.4, 00:00:01 ago, via FastEthernet0/0

Route metric is 1, traffic share count is 1

Route tag 110

r3#

说明：源于 OSPF 的路由 25.1.1.0/24 的 Tag 为 110，与配置预期相同。

（6）在 R1 上查看配置 Tag 过滤后，源于 RIP 路由的 Tag 情况：

```
r5#sh ip route 100.1.1.0
Routing entry for 100.1.1.0/24
  Known via "ospf 1", distance 110, metric 20
  Tag 120, type extern 2, forward metric 1
  Last update from 25.1.1.2 on FastEthernet0/1, 00:01:53 ago
  Routing Descriptor Blocks:
  * 25.1.1.2, from 2.2.2.2, 00:01:53 ago, via FastEthernet0/1
      Route metric is 20, traffic share count is 1
      Route tag 120

r5#
```

说明：源于 RIP 的路由 100.1.1.0/24 的 Tag 为 120，与配置预期相同。

IPv6

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

概述

在我们现有的网络中，几乎所有网络都使用 IP 协议作为通信的地址协议，我们的网络使用 IP 来表示地址信息，每一个节点都应该分配一个唯一的地址，才能保证通信正常。现在正常使用的 IP 协议为版本 4，用 32 位来表示，地址空间为 65536×65536，结果约为 42.9 亿，需要说明的是，虽然地址共有 42.9 亿之多，但并不表示这些地址可以供 42.9 亿个节点使用，因为我们的地址是分网段的，也就是说即使在一个节点的情况下，分配地址时，也是分配一个网段而不是一个地址，所以这样就使得版本 4 的 IP 地址一下子变得空间陕小，再加了有相当一部分地址是不可用的，那么随着网络的迅速膨胀，IP ver4 的地址空间变得几乎快耗尽了。在这样的情况下，出现了一些如 VLSM 子网技术，NAT 网络地址翻译技术，试图来缓和地址空间的快速消耗。与此同时，人们也开发出了一个地址空间更为庞大的 IP 协议，这个协议拥有比 IP ver4 多出数倍的地址空间，来解决网络地址匮乏的问题，这个 IP 协议就是 IP 版本 6，即 IPv6。

IPv6 地址格式

IPv6 拥有更为庞大的地址空间，是因为 IPv4 只是采用 32 位来表示，而 IPv6 采用 128 位来表示，这样大的一个地址空间，几乎可以容纳无数个节点。正因为 IPv6 使用了 128 位来表示地址，在表示和书写上面具有相当的困难，原来的 IPv4 使用 10 进制来表示，而 IPv6 由于地址太长，则采用 16 进制来表示，但无论我们如何表示，计算机都是处理二进制。因为 10 进制表示时，使用 0 到 9 共十个数字来表示，而 16 进制需要在 10 进制原有的基础上多出 6 个数字，即需要多出 11,12,13,14,15，这 6 个数字则采用字母的形式来表示，分别为

A(表示 10)，B(表示 11)，C(表示 12)，D(表示 13)，E(表示 14)，F(表示 15)，这些字母是不区别大小写的。

但是由于 IPv6 拥有 128 位的长度，所以不能直接表示，必须像 IPv4 那样进行分段表示。IPv6 将整个地址分为 8 段来表示，每段之间用冒号隔开，每段的长度为 16 位，表示如下：

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

从上面可以看出，IPv6 中每一个段是 16 位，每段共四个 X，其中 X 使用 4 bit 表示，一个 X 就表示一个数字或字母，一个完整的地址共 128 bit。

一个 X 使用 4 bit 表示，那么 XXXX 的取值范围就应该从 0000 到 FFFF。

IPv6 地址表示方法

对于一个完整的 IPv6 地址，需要写 128 位，已经被分成了 8 段，每段 4 个字符，也就是说完整地表示一个 IPv6 地址，需要写 32 个字母，这是相当长的，并且容易

混淆和出错，所以 IPv6 在地址的表示方法上，是有讲究的，到目前为止，IPv6 地址的表示方法分为三种，分别是：

1.首选格式

2.压缩表示

3.IPv4 内嵌在 IPv6 中

下面分别详细介绍这三种 IPv6 地址表示方法：

1.首选格式

首选格式的表示方法其实没有任何讲究，就是将 IPv6 中的 128 位，也就是共 32 个字符完完整整，一个不漏地全写出来，比如下面就是一些 IPv6 地址的首选格式表示形式：

0000:0000:0000:0000:0000:0000:0000:0000

0000:0000:0000:0000:0000:0000:0000:0001

2001:0410:0000:1234:FB00:1400:5000:45FF

3ffe:0000:0000:0000:1010:2a2a:0000:0001

FE80:0000:0000:0000:0000:0000:0000:0009

FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

从上面 IPv6 地址的首选格式表示中可以看出，每一个地址，都将 32 个字符全部

鸿鹄培训+考试群：301283867

写了出来，即使地址中有许多个 0，或者有许多个 F，也都一个不漏地写了出来，由此可见，首选格式只需要将地址完整写出即可，没有任何复杂的变化，但是容易出错。

2.压缩格式

从前面一个 IPv6 地址表示方法首选格式表示方法中可以看出，一个完整的 IPv6 地址中，会经常性的出现许多个 0,而我们知道，许多时候，0 是毫无意义的，0 表示没有，写出来，也表示没有，不写，也同样表示没有，那么我们就考虑能否将不影响地址结果的 0 给省略不写，这样就可以大大节省时间，也方便人们阅读和书写，这样的将地址省略 0 的表示方法，称为压缩格式。

而压缩格式的表示中，分三种情况，下面来分别介绍三种压缩格式：

第一种情况：

在 IPv6 中，地址分为 8 个段来表示，每个段共 4 个字符，但是一个完整的 IPv6 地址会经常碰到整个段 4 个字符全部都为 0，所以我们将整个段 4 个字符全部都为 0 的使用双冒号::来表示，如果连续多个段全都为 0，那么也可以同样将多个段都使用 双冒号::来表示，如果是多个段，并不需要将双冒号写多次，只需要写一次即可，比如一个地址 8 个段，其中有三个段全都为 0，那么我们就将这全为 0 的三个段共 48 位用::来表示，再将其它 5 个段照常写出即可，当计算机读到这样一个不足 128 位的地址时，比 128 位少了多少位，就在::的地方补上多少个 0，比如上面的::代替为 48 位，那么计算机就会在这个地址的::位置补上 48 位的 0，这样就正确地将地址还原回去了。

下面来看一些整个段 4 个字符都为 0 的 IPv6 地址使用压缩格式来表示：

例 1：

压缩前：

0000:0000:0000:0000:0000:0000:0000:0000

压缩后：

::

说明：可以看出，由于这个地址的 8 个段全部都为 0，所以只用::就将整个地址表示出来，当计算机拿到这个压缩后的地址时，发现比正常的 128 位少了 128 位，那么就会在::的地方补上 128 个 0，结果为：

0000:0000:0000:0000:0000:0000:0000:0000

可以看出，计算机还原的地址就是压缩之前的真实地址。

例 2：

压缩前：

0000:0000:0000:0000:0000:0000:0000:0001

压缩后：

::0001

说明：可以看出，压缩后的地址比正常的 128 位少了 112 位，计算机就会在::的地方补上 112 个 0，结果为：

0000:0000:0000:0000:0000:0000:0000:0001

可以看出，计算机还原的地址就是压缩之前的真实地址。

例 3：

压缩前：

2001:0410:0000: 0000:FB00:1400:5000:45FF

压缩后：

2001:0410 :: FB00:1400:5000:45FF

说明：可以看出，压缩后的地址比正常的 128 位少了 32 位，计算机就会在::的地方补上 32 个 0，结果为：

2001:0410:0000: 0000:FB00:1400:5000:45FF

可以看出，计算机还原的地址就是压缩之前的真实地址。

例 4：

压缩前：

3ffe:0000:0000:0000:1010:2a2a:0000:0001

压缩后：

3ffe::1010:2a2a::0001

说明：当计算机拿到这个压缩后的地址，发现比正常的 128 位少了 64 位，计算机就会试图在::的地方补上少了的 64 个 0，但是我们可以看到，压缩后的地址有两个::，而计算机要补上 64 个 0，所以这时补出来的结果很可能是以下几种：

3ffe:0000:1010:2a2a: 0000:0000:0000:0001

或

3ffe:0000:00001010:2a2a::0000:0000:0001

或

3ffe:0000:0000:0000:1010:2a2a:0000:0001

从结果中可以发现，当一个 IPv6 地址被压缩后，如果计算机出现两个或多个::的时候，计算机在将地址还原时，就可能出现多种情况，这将导致计算机还原后的地址不是压缩之前的地址，将导致地址错误，最终通信失败。

所以，在压缩 IPv6 地址时，一个地址中只能出现一个::。

第二种情况：

在压缩格式的第一种情况的表示中，是在地址中整个段 4 个字符都为 0 时，才将其压缩为::来表示，但是在使用第一种情况压缩之后，我们仍然可以看见地址中还存在许多毫无意义的 0，比如 0001，0410。我们知道，0001 中，虽然前面有三个 0，但是如果我们将前面的 0 全部省略掉，写为 1，结果是等于 0001 的，而 0410 也是一样，我们将前面的 0 省略掉，写成 410，也同样等于 0410 的，所以我们在省略数字前面的 0 时，是不影响结果的，那么这个时候，表示 IPv6 地址时，允许将一个段中前导部分的 0 省略不写，因为不影响结果。但是需要注意的是，如果 0 不是前导 0，比如 2001，我们就不能省略 0 写成 21，因为 21 不等于 2001，所以在中间的 0 不能省略，只能省略最前面的 0。下面来看一些省略前导 0 的地址表示形式：

例 1：

压缩前：

0000:0000:0000:0000:0000:0000:0000:0000

压缩后：

0:0:0:0:0:0:0:0

从结果中可以看出，计算机根本就不需要对这样的地址还原，压缩后的结果和压缩前的结果是相等的。

例 2：

压缩前：

0000:0000:0000:0000:0000:0000:0000:0001

压缩后：

0:0:0:0:0:0:0:1

从结果中可以看出，计算机根本就不需要对这样的地址还原，压缩后的结果和压缩前的结果是相等的。

例 3：

压缩前：

2001:0410:0000:1234:FB00:1400:5000:45FF

压缩后：

2001:410:0:1234:FB00:1400:5000:45FF

从结果中可以看出，计算机根本就不需要对这样的地址还原，压缩后的结果和压缩前的结果是相等的。

第三种情况：

在前面两种 IPv6 地址的压缩表示方法中，第一种是在整段 4 个字符全为 0 时，才将其压缩后写为::,而第二种是将无意义的 0 省略不写，可以发现两种方法都能节省时间，方便阅读。第三种压缩方法就是结合前两种方法，既将整段 4 个字符全为 0 的部分写成::， 也将无意义的 0 省略不写，结果就可以出现以下一些最方便的表示方法：

例 1：

压缩前：

0000:0000:0000:0000:0000:0000:0000:0001

压缩后：

:: 1

可以看到，结合了两种压缩格式的方法，但为简便。

例 2：

压缩前：

2001:0410:0000:0000:FB00:1400:5000:45FF

压缩后：

2001:410:: FB00:1400:5000:45FF

可以看到，结合了两种压缩格式的方法，但为简便。

3.IPv4 内嵌在 IPv6 中

在网络还没有全部从 IPv4 过渡到 IPv6 时，就可能出现某些设备即连接了 IPv4 网络，又连接了 IPv6 网络，对于这样的情况，就需要一个地址即可以表示 IPv4 地址，又可以表示 IPv6 地址。

因为一个 IPv4 地址为 32 位，一个 IPv6 地址为 128 位，要让一个 IPv4 地址表示为 IPv6 地址，明显已经少了 96 位，那么就将一个正常的 IPv4 地址通过增加 96 位，结果变成 128 位，来与 IPv6 通信。在表示时，是在 IPv4 原有地址的基础上，增加 96 个 0，结果变成 128 位，增加的 96 个 0 再结合原有的 IPv4 地址，表示方法为

0:0:0:0:0:0:A.B.C.D 或者::A.B.C.D.，如下：

0000: 0000: 0000: 0000: 0000:0000:A.B.C.D

9 6 个 0　　32 位

例：

IPv4 地址为　138.1.1.1

表示 IPv6 地址为 0:0:0:0:0:0:138.1.1.1

注：IPv6 中没有广播地址，IPv6 不建议划子网，如果需要划子网，网络位请不要低于 48 位。

IPv6 地址类型

在 IPv4 地址中，地址分许多类型，比如代表节点自己的 127.0.0.0/8,私有地址段，

鸿鹄培训+考试群：301283867

组播地址段，广播地址，以及一些不可用的地址。在 IPv6 中，同样地址也像 IPv4 那样分了许多类型，我们需要了解的有 3 种类型，为 Unicast（单播），Anycast（任意播）和 Multicast（组播），下面分别来详细介绍这几种地址类型。

Unicast（单播）

即使是在 IPv4 中，单播地址的类型也分好多种，就是我们常用的也分私有，公有，还有回环地址，在 IPv6 中，单播地址也分好几种，我们需要知道的有：Link-Local Address（链路本地地址），Unique Local Address （本地站点地址），Aggregatable Global Address（可聚合全球），回环地址。

下面详细介绍几种单播地址：

Link-Local Address（链路本地地址）

即使网络再大，每两点之间，都有链路相连，在一个节点将数据包发给下一个节点时，必须在数据包中封装三层 IP 地址，再封装下一节点的二层链路地址（如以太网中的 MAC 地址），才能将数据包发给下一节点，并且只有当封装的二层链路地址确实为下一节点的真实链路地址时，对方才能接收，这就是普通二层链路地址的功能，这样的地址在一条链路的范围内明确了每个节点，并且这样的地址是不能被路由的。

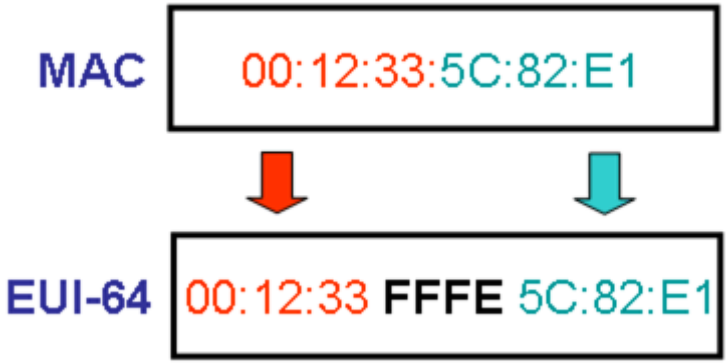
而在 IPv6 网络中，两个 IPv6 的节点通过链路相连，必须在这条链路之间为各自确立一个 Link-Local Address（即链路本地地址），在一条链路上，IPv6 节点能够确定对方节点的身份，能够将数据包发向对方节点，必须知道对方节点的链路本地地址，如果不知道，将是不能通信的，所以一条链路中的 IPv6 节点要通信，必须拥有链路本地地址，并且这个链路本地地址只在一条链路中有效，也不能被路由，而不同链路的链路本地地址是可以重复的。

因为在链路上没有链路本地地址的情况下，IPv6 是不能通信的，所以每个节点必须拥有一个链路本地地址，当一个节点上正常启动了 IPv6 之后，链路本地地址是不需要人工干预，会自己生成的，但也可以自己手工配置链路本地地址。

自动生成的链路本地地址，有默认的特殊格式，是以 FE80::/10 (1111 1110 10) 打头，再加 54 个 0，还差 64 位，这后面的 64 位，再使用 EUI-64 来填充，表示如下：

EUI-64 结构

一个链路本地地址的后 64 位使用 EUI-64 来填充，EUI-64 其实就是接口的 MAC 地址，而 MAC 地址共长度为 48 位，要填充 64 位的 EUI-64，还少 16 位。一个完整的 EUI-64 是将 MAC 地址的 48 位平均分成两部分，前面 24 位，后面 24 位，然后在中间补上 FFFE(16 位)，如一个 MAC 地址为 00:12:33:5C:82:E1，将其变为 EUI-64 的结果如下表示：



只有以太网链路才会有 MAC 地址，而串行链路是没有 MAC 地址的，当一个接口上启用 IPv6 之后，此接口会自动产生一个链路本地地址，而链路本地地址需要借用接口上的 MAC 地址才能产生，这在有 MAC 地址的以太网接口上可以轻松实现，但是当一个没有 MAC 地址的串行接口上开启 IPv6 之后，由于自己没有 MAC 地址，所以不能产生 EUI-64，也就无法完成链路本地地址。在这种情况下，所有没有 MAC 地址的接口，如串行接口，在开启 IPv6 后，需要产生 EUI-64 时，统统借用设备上第一个以太网插槽的第一个接口，也可以理解为没有 MAC 地址的接口，统统使用设备上 MAC 地址池中的第一个地址，比如设备上为接口 s2/3 开启 IPv6 之后，就很有可能借用 F0/0 或 G0/0 接口的 MAC 地址。

EUI-64 不仅在产生链路本地地址时可以使用，在正常配置 IPv6 地址时，同样可以使用 EUI-64 来填充后 64 位。

本地站点地址

本地站点地址是单播中一种受限制的地址，只在一个站点内使用，不会默认启用，这个地址不能在公网上路由，只能在一个指定的范围内路由，需要手工配置。IPv6 中的本地站点地址类似 IPv4 中私有地址，如 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16。

得不到合法 IPv6 地址的机构可配置本地站点地址，表示方法为：

FC00::/7 + 41bit 子网标识 +16bit 链路标识+ EUI-64

可聚合全球单播地址

可聚合全球单播地址相当于 IPv4 的公网地址，可以被路由的，可以正常使用的地址，但网络位最少为 48 位。可聚合全球单播地址的范围是

2000:0000:0000:0000:0000:0000:0000:0000

到

3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

以上面可以看出，可聚合全球单播地址也就是 2 和 3 打头的地址，因为 IPv6 使用 16 进制来表示，一个字符的取值范围从 0 到 F 共 16 个，而可聚合全球单播地址地址占了 2 和 3 两个，由此说明，可聚合全球单播地址占 IPv6 总地址空间的 8 分之 1，也就是说，所有 IPv6 地址中，只有 8 分之 1 是可以给网络正常使用的。

回环地址

回环地址表示节点自身，类似 IPv4 的 127.0.0.0/8

回环地址表示为

0000:0000:0000:0000:0000:0000:0000:0001

0:0:0:0:0:0:0:1

::1

任意播地址

任意播地址表示一组接口，当一个发向某个任意播地址的数据包，只被最近的接口收到，这个地址是由路由协议定义的，不能手工配置，但是我们无法看到一个地址就能区别出到底是单播地址还是任意播地址，因为任意播地址的表示格式和单播地址是一样的，也就是说任意播地址就是用普通的单播地址来表示的。任意播地址只能出现在路由器上，并且不能作为数据包的源地址来使用。

组播

组播地址就是一个目标为组播地址的数据包将被多个节点收到，地址以 FF00::/8 (11111111)打头，.表示为

FF00:0000:0000:0000:0000:0000:0000:0000/8

FF00:0:0:0:0:0:0:0/8

FF00::/8

详细的 IPv6 多播知识将在后面的 IPv6 Multicast 部分介绍。

配置 IPv6 地址

1.激活 IPv6 功能

默认情况下，Cisco 设备的 IPv6 流量转发功能是关闭的，需要使用 IPv6，必须先开启 IPv6 流量转发功能。

（1）开启 IPv6 流量转发功能

r1(config)#ipv6 unicast-routing

2.配置正常的 IPv6 地址

（1）在接口下配置正常 IPv6 地址

r1(config)#int f0/0

r1(config-if)#ipv6 address 2011:1:2:3:1:1:1:1/64

说明：配置的地址前 64 位为网络地址，即 2011:1:2:3; 后 64 位为主机位，即 1:1:1:1。

（2）查看接口的 IPv6 地址

r1#show ipv6 interface brief f0/0

FastEthernet0/0 [up/up]

FE80::C200:EFF:FEB0:0

2011:1:2:3:1:1:1:1

r1# r1#

说明：可以看到接口 F0/0 已经接受我们配置的地址 2011:1:2:3:1:1:1:1。

3.使用 EUI-64 格式配置静态地址：

(1)配置包含 EUI-64 的 IPv6 地址

r1(config)#int f0/1

r1(config-if)#ipv6 address 2022:2:2:22::/64 eui-64

（2）查看接口的 IPv6 地址

r1#show interfaces f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is Gt96k FE, address is c000.0eb0.0000 (bia c000.0eb0.0000)

r1#show ipv6 interface brief f0/1

FastEthernet0/1 [up/up]

FE80::C200:EFF:FEB0:1

2022:2:2:22:C200:EFF:FEB0:1

r1#

说明：可以看到，F0/1 成功使用接口上的 MAC 地址为 EUI-64 来填充后 64 位。

4.仅启用接口 IPv6 功能

说明：一个接口上可以仅启用 IPv6 功能，而不配置 IPv6 地址

（1）启用接口 IPv6 功能

r1(config)#int s1/0

r1(config-if)#ipv6 enable

（2）查看接口 IPv6 状态

r1#show ipv6 interface brief serial 1/0

Serial1/0 [up/up]

FE80::C200:EFF:FEB0:0

r1#

可以看到，接口 S1/0 可以只开启 IPv6 功能而不配地址，但开了 IPv6 功能的接口也会自动产生一个链路本地地址。

5.配置无编号地址

当地址紧缺时，可以配置一个接口使用另外一个接口的地址，这样的地址称为无编号地址，即 unnumbered 地址，当从无编号接口产生数据包时，该接口使用借用

的那个接口的地址作为源地址，配置这样的地址，需要允许双方不同网段协议的配合。

（1）为接口配置无编号地址

r1(config)#int s1/1

r1(config-if)#ipv6 unnumbered f0/0

（2）查看接口 IPv6 地址情况。

r1#show ipv6 interface brief serial 1/1

Serial1/1 [up/up]

FE80::C200:EFF:FEB0:0

unnumbered (FastEthernet0/0)

r1#

说明：可以看到结果显示为接口 S1/1 借用 F0/0 的地址。

IPv6 静态路由

在 IPv6 中，静态路由的写法分三种，分别为：

1.直连静态路由（Directly Attached Static Routes）

写法为只指定路由的出口，目标网络被认为是和此接口直连的，但此方法在接口为多路访问时，会有问题。

例配：

ipv6 route 2022:2:2:22::/64 s1/1

说明：到达目标网络 2022:2:2:22::/64 的数据包从接口 s1/1 发出去。

2.递归静态路由（Recursive Static Routes ）

写法为只指定路由的下一跳地址，此方法在任何网络环境中可行。

例配：

r1(config)#ipv6 route 2022:2:2:22::/64 2012:1:1:11::2

说明：到达目标网络 2022:2:2:22::/64 的数据包发给下一跳地址 2012:1:1:11::2。

3.完全静态路由（Fully Specified Static Routes）

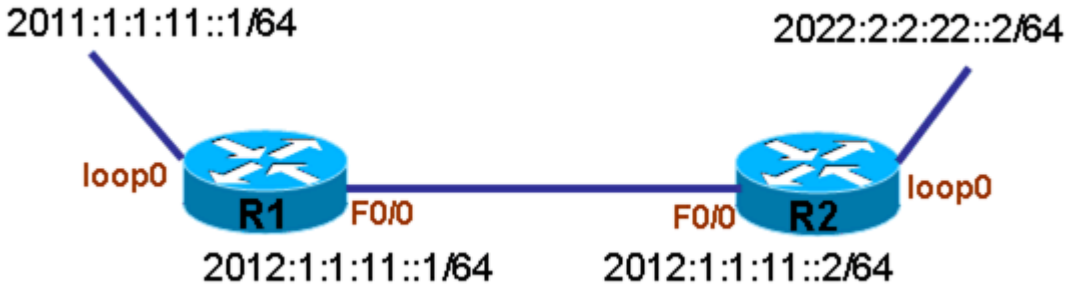
写法为同时指定出口和下一跳地址，只有当出口为多路访问时，并且确实需要明确指定下一跳时，才需要写完全静态路由，下一跳必须是和出口同网段的。

例配：

r1(config)#ipv6 route 2022:2:2:22::/64 f0/0 2012:1:1:11::2

说明：到达目标网络 2022:2:2:22::/64 的数据包从接口 F0/0 发出去，并且交给下一跳地址 2012:1:1:11::2。

IPv6 静态路由配置实验



说明：配置静态路由，使双方都能 ping 通互相 loopback 接口的网段。

由于是多路访问接口，所以省去配置直连静态路由的方法。

1.网络初始配置：

（1）R1 初始配置：

r1(config)#ipv6 unicast-routing

r1(config)#int f0/0

r1(config-if)#ipv address 2012:1:1:11::1/64

r1(config)#int loopback 0

r1(config-if)#ipv6 address 2011:1:1:11::1/64

r1(config-if)#

（2）R2 初始配置：

r2(config)#ipv unicast-routing

r2(config)#int f0/0

r2(config-if)#ipv address 2012:1:1:11::2/64

r2(config)#int loopback 0

r2(config-if)#ipv6 address 2022:2:2:22::2/64

r2(config-if)#

2.在 R1 上配置递归静态路由

（1）配置递归静态路由

r1(config)#ipv6 route 2022:2:2:22::/64 2012:1:1:11::2

说明：到达目标网络 2022:2:2:22::/64 的数据包发给下一跳地址 2012:1:1:11::2。

鸿鹄培训+考试群：301283867

（2）检查静态路由

r1#show ipv6 route static

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

S 2022:2:2:22::/64 [1/0]

via 2012:1:1:11::2

r1#

说明：从结果中看出，手工配置的递归静态路由已生效。

（3）测试连通性

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/48/140 ms

r1#

说明：由于正确配置静态路由，R1 到 R2 的 loopback 接口的网段通信正常。

3.在 R2 上配置完全静态路由

（1）配置完全静态路由

r2(config)#ipv6 route 2011:1:1:11::/64 f0/0 2012:1:1:11::1

说明：到达目标网络 2011:1:1:11::/64 的数据包从接口 F0/0 发出去，并且交给下一跳地址 2012:1:1:11::1。

（2）检查静态路由

r2#show ipv6 route static

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

S 2011:1:1:11::/64 [1/0]

via 2012:1:1:11::1, FastEthernet0/0

r2#

说明：从结果中看出，手工配置的完全静态路由已生效。

（3）测试连通性

r2#ping 2011:1:1:11::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2011:1:1:11::1, timeout is 2 seconds:

!!!!

*Mar 1 00:36:50.387: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/0 (not full duplex), with Router FastEthernet0/2 (full duplex).!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/92/156 ms

r2#

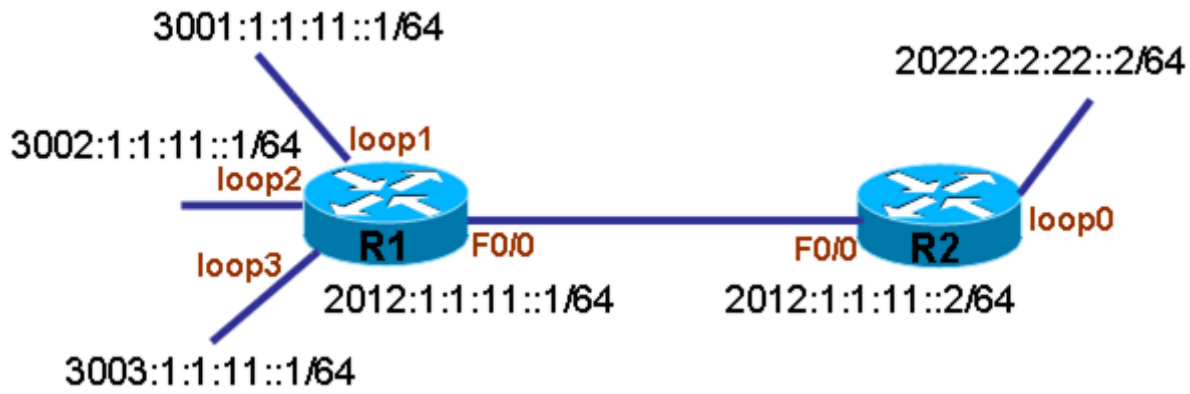
说明：由于正确配置静态路由，R2 到 R1 的 loopback 接口的网段通信正常。

IPv6 RIP (RIPng)

IPv6 的 RIP，所有路由规则与 IPv4 RIPv2 基本相同，不同之处是 IPv4 RIPv2 使用 UDP 端口 520，而 RIPng 使用 UDP 端口 521，IPv4 RIPv2 数据包更新使用地址 224.0.0.9，而 RIPng 使用更新地址为 FF02::9。

在配置 RIPng 时，方法不同于 IPv4 RIP，RIPng 是采用先配置进程，然后需要让哪些接口运行在 RIPng 下，就必须到相应的接口下明确指定，并不像 IPv4 RIP 那样在进程下通过 network 来发布。

配置 RIPng



1.初始配置

（1）R1 初始配置：

r1(config)#ipv6 unicast-routing

r1(config)#int f0/0

r1(config-if)#ipv6 address 2012:1:1:11::1/64

r1(config)#int loopback 1

r1(config-if)#ipv6 address 3001:1:1:11::1/64

r1(config)#int loopback 2

r1(config-if)#ipv6 address 3002:1:1:11::1/64

r1(config)#int loopback 3

r1(config-if)#ipv6 address 3003:1:1:11::1/64

鸿鹄培训+考试群：301283867

（2）R2 初始配置：

r2(config)#ipv6 unicast-routing

r2(config)#int f0/0

r2(config-if)#ipv6 address 2012:1:1:11::2/64

r2(config)#int loopback 0

r2(config-if)#ipv6 address 2022:2:2:22::2/64

2.启动 RIPng 进程

说明：Cisco IOS 最多同时支持 4 个 RIPng 进程，不同进程使用不同名字来区分，并且进程名为本地有效。

（1）在 R1 上启动 RIPng 进程

r1(config)#ipv6 router rip ccie

r1(config-rtr)#exit

（2）在 R2 上启动 RIPng 进程

r2(config)#ipv6 router rip ccie

r2(config-rtr)#exi

3.配置 RIPng 接口

（1）将 R1 上的接口放进 RIPng 进程

r1(config)#int f0/0

鸿鹄培训+考试群：301283867

r1(config-if)#ipv6 rip ccie enable

r1(config)#int loopback 1

r1(config-if)#ipv6 rip ccie enable

（2）将 R2 上的接口放进 RIPng 进程

r2(config)#int f0/0

r2(config-if)#ipv6 rip ccie enable

r2(config)#int loopback 0

r2(config-if)#ipv6 rip ccie enable

4.查看 RIPng 路由

（1）查看 R1 的 RIPng 路由

r1#show ipv6 route rip

IPv6 Routing Table - 11 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

R 2022:2:2:22::/64 [120/2]

via FE80::C200:DFF:FEC4:0, FastEthernet0/0

r1#

说明：由于 RIPng 配置正确，成功收到对方路由条目，并且可以看出，动态路由学习到的 IPv6 路由条目，下一跳地址均为对端的链路本地地址。

（2）查看 R2 的 RIPng 路由

r2#show ipv6 route rip

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

R 3001:1:1:11::/64 [120/2]

via FE80::C200:BFF:FE48:0, FastEthernet0/0

r2#

说明：由于 RIPng 配置正确，成功收到对方路由条目。

5.测试连通性

说明：因为动态路由学习到的 IPv6 路由条目，下一跳地址均为对端的链路本地地址，所以如果到对端的链路本地地址不通，那么到对端 IPv6 网络也不会通。

(1)测试 R1 到对端链路本地地址的连通性

r1#ping FE80::C200:DFF:FEC4:0

Output Interface: FastEthernet0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FE80::C200:DFF:FEC4:0, timeout is 2 seconds:

Packet sent with a source address of FE80::C200:DFF:FEC4:0

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/70/184 ms

r1#

说明：到对端链路本地地址的通信正常。

（2）测试 R1 到对端 IPv6 网络的连通性

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/75/240 ms

r1

说明：由于到对端链路本地地址的通信正常，所以到对端 IPv6 网络的通信也正常。

（3）测试 R2 到对端 IPv6 网络的连通性

r2#ping 3001:1:1:11::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3001:1:1:11::1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/84/248 ms

r2#

说明：到对端 IPv6 网络的通信也正常。

6.重分布 IPv6 网段

说明：将 R1 上的剩余网段重分布进 RIPng

（1）在 R1 上配置重分布剩余网段进 RIPng

r1(config)#route-map con permit 10

r1(config-route-map)#match interface loopback 2

r1(config-route-map)#exit

r1(config)#route-map con permit 20

r1(config-route-map)#match interface loopback 3

r1(config-route-map)#exit

r1(config)#ipv6 router rip ccie

r1(config-rtr)#redistribute connected route-map con

r1(config-rtr)#

（2）在 R2 上查看重分布进 RIPng 的剩余网段

r2#show ipv6 route rip

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

R 3001:1:1:11::/64 [120/2]

via FE80::C200:BFF:FE48:0, FastEthernet0/0

R 3002:1:1:11::/64 [120/2]

via FE80::C200:BFF:FE48:0, FastEthernet0/0

R 3003:1:1:11::/64 [120/2]

via FE80::C200:BFF:FE48:0, FastEthernet0/0

r2#

说明：可以看到，R1 上的剩余网段成功被重分布进 RIPng。

7.过滤 IPv6 路由

说明：在 R2 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 过滤

（1）配置只留 3002:1:1:11::/64 网段

r2(config)#ipv6 prefix-list abc permit 3002:1:1:11::/64

r2(config)#ipv6 router rip ccie

r2(config-rtr)#distribute-list prefix-list abc in f0/0

注：ipv6 的 prefix-list 同样支持 ge , le 等关键字来匹配范围。

（2）查看过滤后的路由表情况

r2#show ipv6 route rip

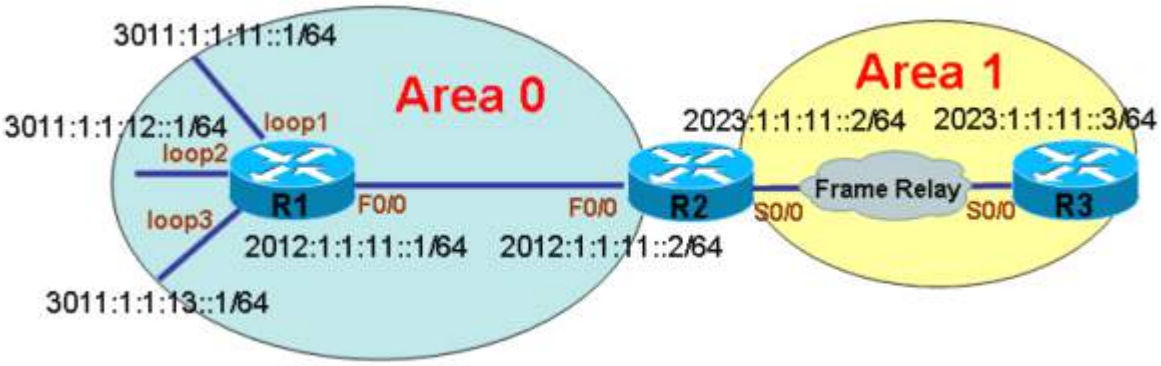
```
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
R    3002:1:1:11::/64 [120/2]
     via FE80::C200:BFF:FE48:0, FastEthernet0/0
r2#
```

说明：路由表中只剩想要的网段，说明过滤成功。

IPv6 OSPF （OSPFv3）

OSPFv3 与 OSPFv2（IPv4 OSPF）的原理都是相同的，OSPFv3 选举 Router-ID 的规则与 OSPFv2 相同，OSPFv3 也是选择路由器上的 IPv4 地址作为 Router-ID，如果设备上没有配置 IPv4 地址，那么必须手工指定 Router-ID。在配置 OSPFv3 时，先配置进程，然后需要让哪些接口运行在 OSPFv3 下，就必须到相应的接口下明确指定，并不像 OSPFv2 那样在进程下通过 network 来发布。

配置 OSPFv3



1.初始配置

（1）R1 初始配置：

r1(config)#ipv6 unicast-routing

r1(config)#interface f0/0

r1(config-if)#ipv6 address 2012:1:1:11::1/64

r1(config)#int loopback 1

r1(config-if)#ipv6 address 3011:1:1:11::1/64

r1(config)#int loopback 2

r1(config-if)#ipv6 address 3011:1:1:12::1/64

r1(config)#int loopback 3

r1(config-if)#ipv6 address 3011:1:1:13::1/64

鸿鹄培训+考试群：301283867

（2）R2 初始配置：

r2(config)#ipv6 unicast-routing

r2(config)#interface f0/0

r2(config-if)#ipv6 address 2012:1:1:11::2/64

r2(config)#interface s1/0

r2(config-if)#encapsulation frame-relay

r2(config-if)#no frame-relay inverse-arp

r2(config-if)#no arp frame-relay

r2(config-if)#ipv6 address 2023:1:1:11::2/64

r2(config-if)#frame-relay map ipv6 2023:1:1:11::3 203 broadcast

r2(config-if)#

（3）R3 初始配置：

r3(config)#ipv6 unicast-routing

r3(config)#interface s1/0

r3(config-if)#encapsulation frame-relay

r3(config-if)#no frame-relay inverse-arp

r3(config-if)#no arp frame-relay

r3(config-if)#ipv6 address 2023:1:1:11::3/64

r3(config-if)#frame-relay map ipv6 2023:1:1:11::2 302 broadcast

2.启动 OSPFv3 进程

（1）启动 R1 的 OSPFv3 进程

r1(config)#ipv6 router ospf 2

r1(config-rtr)#router-id 1.1.1.1

说明：由于没有配置 IPv4 地址，所以必须手工配置 Router-ID

（2）启动 R2 的 OSPFv3 进程

r2(config)#ipv6 router ospf 2

r2(config-rtr)#router-id 2.2.2.2

（3）启动 R3 的 OSPFv3 进程

r3(config)#ipv6 router ospf 2

r3(config-rtr)#router-id 3.3.3.3

3.配置 OSPFv3 接口

（1）将 R1 上的接口放进 OSPFv3 进程

r1(config)#int f0/0

r1(config-if)#ipv6 ospf 2 area 0

r1(config)#int loopback 1

r1(config-if)#ipv6 ospf 2 area 0

（2）将 R2 上的接口放进 OSPFv3 进程

r2(config)#int f0/0

r2(config-if)#ipv6 ospf 2 area 0

鸿鹄培训+考试群：301283867

r2(config)#int s1/0

r2(config-if)#ipv6 ospf 2 area 1

（3）将 R3 上的接口放进 OSPFv3 进程

r3(config)#int s1/0

r3(config-if)#ipv6 ospf 2 area 1

4.查看 OSPFv3 邻居

（1）查看 r1 邻居：

r1#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/BDR	00:00:39	4	FastEthernet0/0

r1#

说明：R1 与 R2 的 OSPFv3 邻居正常。

（2）查看 r2 邻居：

r2#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
1.1.1.1	1	FULL/DR	00:00:35	4	FastEthernet0/0

r2#

说明：R2 与 R2 的 OSPFv3 邻居正常，但与 R3 的邻居没有。

鸿鹄培训+考试群：301283867

（3）**（3）查看 r3 邻居：**

r3#show ipv6 ospf neighbor

r3#

说明：R3 没有 OSPFv3 邻居。

5.解决 OSPFv3 邻居问题

说明：由于 R2 与 R3 之间属于 NBMA 非广播网络，所以无法自动建邻居，要解决邻居问题，有两种方法：第一，手工指定邻居，在指定时，只须在一方指定即可，并且 OSPFv3 在手工指定邻居时，需要到接口下指定而不是在进程下指定，并且指定的为对方链路本地地址。第二，将网络类型从非广播网络类型改为允许广播的网络类型，如改为 Point-to-point 类型。

（1）查看 R3 连 R2 接口的链路本地地址

r3#show ipv6 interface brief s1/0

Serial1/0 [up/up]

FE80::C200:DFF:FEAC:0

2023:1:1:11::3

r3#

（2）在 R2 上指定 R3 为邻居，在接口下指定对方的链路本地地址

r2(config)#int s1/0

r2(config-if)#ipv6 ospf neighbor FE80::C200:DFF:FEAC:0

r2(config-if)#

（3）测试 R2 到 R3 接口链路本地地址的连通性

r2#ping FE80::C200:DFF:FEAC:0

Output Interface: Serial1/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FE80::C200:DFF:FEAC:0, timeout is 2 seconds:

Packet sent with a source address of FE80::C200:BFF:FE94:0

.....

Success rate is 0 percent (0/5)

r2#

说明：由于指定邻居时，指定为对方接口的链路本地地址，所以双方接口的链路本地地址不通，邻居将仍然不能建立。

（4）解决帧中继网络下双方接口的链路本地地址的 PVC 映射

注：必须互相映射

R2:

r2(config)#int s1/0

r2(config-if)#fram map ipv6 FE80::C200:DFF:FEAC:0 203 broadcast

R3:

R3(config)#int s1/0

R3config-if)#fram map ipv6 FE80::C200:BFF:FE94:0 302 broadcast

(5)查看邻居

r3#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/BDR	00:01:42	6	Serial1/0

r3#

说明：由于已经手工指定邻居，并且也映射了双方的链路本地地址，所以邻居成功建立。

6.查看 OSPFv3 路由

（1）在 R1 上查看 OSPFv3 路由

```
r1#sh ipv6 route ospf
IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
OI  2023:1:1:11::/64 [110/74]
     via FE80::C200:BFF:FE94:0, FastEthernet0/0
r1#
```

说明：由于邻居已经全部正常建立，所以学习到了远程网络的路由条目。

（2）在 R2 上查看 OSPFv3 路由

```
r2#show ipv6 route ospf
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
```

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 3011:1:1:11::1/128 [110/10]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

r2#

说明：由于邻居已经全部正常建立，所以学习到了远程网络的路由条目。

（3）在 R3 上查看 OSPFv3 路由

r3#show ipv6 route ospf

IPv6 Routing Table - 6 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

OI 2012:1:1:11::/64 [110/74]

via FE80::C200:BFF:FE94:0, Serial1/0

OI 3011:1:1:11::1/128 [110/74]

via FE80::C200:BFF:FE94:0, Serial1/0

r3#

说明：由于邻居已经全部正常建立，所以学习到了远程网络的路由条目。

7.解决 OSPFv3 路由掩码问题

说明：由于学习到的路由中，属于 loopback 接口的网段原本为 64 位，而学习到的为 128 位，为主机路由，所以应让路由掩码与原来的掩码一致，需要将网络类型改为 Point-to-point 类型。

（1）在 R1 改 loopback 接口的网络类型改为 Point-to-point

r1(config)#int loopback 1

r1(config-if)#ipv6 ospf network point-to-point

r1(config-if)#

（2）查看改后的路由情况

r2#show ipv6 route ospf

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 3011:1:1:11::/64 [110/11]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

r2#

说明：已经成功变成原来的掩码位数。

8.重分布 IPv6 网段

说明：将 R1 上的剩余网段重分布进 OSPFv3

（1）在 R1 上配置重分布剩余网段进 OSPFv3

r1(config)#route-map con permit 10

r1(config-route-map)#match interface loopback 2

r1(config-route-map)#exit

r1(config)#route-map con permit 20

r1(config-route-map)#match interface loopback 3

r1(config-route-map)#exit

r1(config)#ipv6 router ospf 2

r1(config-rtr)#redistribute connected route-map con

（2）在 R2 上查看重分布进 OSPFv3 的剩余网段

r2#show ipv6 route ospf

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 3011:1:1:11::/64 [110/11]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

OE2 3011:1:1:12::/64 [110/20]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

OE2 3011:1:1:13::/64 [110/20]

via FE80::C200:AFF:FE28:0, FastEthernet0/0

r2#

说明：可以看到，R1 上的剩余网段成功被重分布进 OSPFv3。

（3）在 R3 上查看重分布进 OSPFv3 的剩余网段

r3#show ipv6 route ospf

IPv6 Routing Table - 8 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

OI 2012:1:1:11::/64 [110/74]

via FE80::C200:BFF:FE94:0, Serial1/0

OI 3011:1:1:11::/64 [110/75]

via FE80::C200:BFF:FE94:0, Serial1/0

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

OE2 3011:1:1:12::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

OE2 3011:1:1:13::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

r3#

说明：可以看到，R1 上的剩余网段成功被重分布进 OSPFv3。

9.过滤 IPv6 路由

说明：在 R3 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 过滤

（1）配置只留 3011 打头的网段

r3(config)#ipv6 prefix-list abc permit 3011::/16 ge 64 le 64

r3(config)#ipv6 router ospf 2

r3(config-rtr)#distribute-list prefix-list abc in s1/0

（2）查看过滤后的路由表情况

r3#show ipv6 route ospf

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

OI 3011:1:1:11::/64 [110/75]

via FE80::C200:BFF:FE94:0, Serial1/0

OE2 3011:1:1:12::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

OE2 3011:1:1:13::/64 [110/20]

via FE80::C200:BFF:FE94:0, Serial1/0

r3#

说明：路由表中只剩 3011 打头的网段了，说明过滤成功。

10.汇总 OSPFv3 外部路由

说明：对从外部重分布进 OSPFv3 的路由进行汇总，OSPF 内的路由汇总，命令格式基本同 IPv4，需要注意的是，汇总必须在重分布的路由器上配置，即必须在 ASBR 上配置。

（1）在 ASBR（R1）上配置外部路由的汇总

说明：将 3011:1:1:11::/64 ， 3011:1:1:12::/64 ， 3011:1:1:13::/64 三条路由汇总成 3011:1:1::/48

r1(config)#ipv6 router ospf 2

r1(config-rtr)#summary-prefix 3011:1:1::/48

r1(config-rtr)#

(2)在 R2 上查看汇总后的路由表情况

r2#show ipv6 route ospf

IPv6 Routing Table - 8 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

```
        O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
        ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
        D - EIGRP, EX - EIGRP external
OE2   3011:1:1::/48 [110/20]
     via FE80::C200:AFF:FE28:0, FastEthernet0/0
O    3011:1:1:11::/64 [110/11]
     via FE80::C200:AFF:FE28:0, FastEthernet0/0
r2#
```

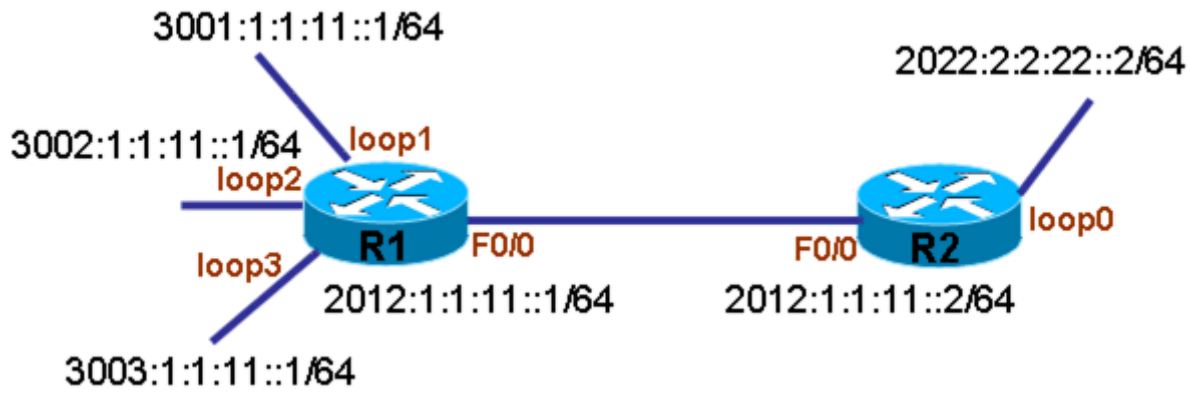
说明：可以看到，汇总成功。

IPv6 EIGRP (EIGRP v6)

EIGRP v6 与 IPv4 EIGRP 的原理都是相同的，但是 EIGRP v6 必须有 router-id 才能运行，所以在 EIGRP v6 不能获得 router-id 时，请手工配置 router-id；更多的是 EIGRP v6 进程有个 shutdown 的特性，要用 no shutdown 开启进程；在配置 EIGRP v6 时，先配置进程，然后需要让哪些接口运行在 EIGRP v6 下，就必须到相应的接口下明确指定，并不像 IPv4 EIGRP 那样通过 network 来发布。

EIGRP hello 时间默认是 5 秒一个，在低链路是 60 秒一个，比如 NBMA，或者所有低于或等于 T1 的链路（1.544M）。Hold time 是 hello 的三倍。

配置 EIGRP v6



1.初始配置

（1）R1 初始配置：

r1(config)#ipv6 unicast-routing

r1(config)#int f0/0

r1(config-if)#ipv6 address 2012:1:1:11::1/64

r1(config)#int loopback 1

r1(config-if)#ipv6 address 3001:1:1:11::1/64

r1(config)#int loopback 2

r1(config-if)#ipv6 address 3002:1:1:11::1/64

r1(config)#int loopback 3

r1(config-if)#ipv6 address 3003:1:1:11::1/64

鸿鹄培训+考试群：301283867

（2）R2 初始配置：

r2(config)#ipv6 unicast-routing

r2(config)#int f0/0

r2(config-if)#ipv6 address 2012:1:1:11::2/64

r2(config)#int loopback 0

r2(config-if)#ipv6 address 2022:2:2:22::2/64

2.配置 EIGRP v6 进程

（1）在 R1 上启动 EIGRP v6 进程

r1(config)#ipv6 router eigrp 10

r1(config-rtr)#router-id 1.1.1.1

（2）在 R1 上启动 EIGRP v6 进程

r2(config)#ipv6 router eigrp 10

r2(config-rtr)#router-id 2.2.2.2

3.配置 EIGRP v6 接口

（1）将 R1 上的接口放进 EIGRP v6 进程

r1(config)#interface f0/0

r1(config-if)#ipv6 eigrp 10

r1(config)#int loopback 1

r1(config-if)#ipv6 eigrp 10

（2）将 R2 上的接口放进 EIGRP v6 进程

r2(config)#int f0/0

r2(config-if)#ipv6 eigrp 10

r2(config)#int loopback 0

r2(config-if)#ipv6 eigrp 10

（3）查看 EIGRP v6 邻居状态

r1#show ipv6 eigrp neighbors

IPv6-EIGRP neighbors for process 10

% EIGRP 10 is in SHUTDOWN

r1#

说明：从结果中看出，EIGRP v 进程默认是 shutdown 的，必须手工开启。

（4）开启 EIGRP v6 进程

r1(config)#ipv6 router eigrp 10

r1(config-rtr)#no shutdown

（5）查看邻居

r1#show ipv6 eigrp neighbors

IPv6-EIGRP neighbors for process 10

H Address Interface Hold Uptime SRTT RTO Q Seq

(sec) (ms) Cnt Num

0 Link-local address: Fa0/0 11 00:00:36 192 1152 0 2

FE80::C200:AFF:FE50:0

说明：开启 EIGRP v6 进程后，邻居正常建立。

4.查看 EIGRP v6 路由

（1）查看 R1 的 EIGRP v6 路由

r1#show ipv6 route eigrp

IPv6 Routing Table - 11 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

D 2022:2:2:22::/64 [90/409600]

via FE80::C200:AFF:FE50:0, FastEthernet0/0

r1#

说明：由于 EIGRP v6 配置正确，成功收到对方路由条目。

（2）查看 R2 的 EIGRP v6 路由

r2#sh ipv6 route eigrp

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

D 3001:1:1:11::/64 [90/409600]

via FE80::C200:9FF:FE54:0, FastEthernet0/0

r2#

说明：由于 EIGRP v6 配置正确，成功收到对方路由条目。

5.重分布 IPv6 网段

说明：将 R1 上的剩余网段重分布进 EIGRP v6

（1）在 R1 上配置重分布剩余网段进 EIGRP v6

r1(config)#route-map con permit 10

r1(config-route-map)#match interface loopback 2

r1(config-route-map)#exit

r1(config)#route-map con permit 20

r1(config-route-map)#match interface loopback 3

r1(config)#ipv6 router eigrp 10

r1(config-rtr)#redistribute connected route-map con

r1(config-rtr)#exit

（2）在 R2 上查看重分布进 EIGRP v6 的剩余网段

r2#sh ipv6 route eigrp

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

D 3001:1:1:11::/64 [90/409600]

via FE80::C200:9FF:FE54:0, FastEthernet0/0

EX 3002:1:1:11::/64 [170/409600]

via FE80::C200:9FF:FE54:0, FastEthernet0/0

EX 3003:1:1:11::/64 [170/409600]

via FE80::C200:9FF:FE54:0, FastEthernet0/0

r2#

说明：可以看到，R1 上的剩余网段成功被重分布进 EIGRP v6。

6.过滤 IPv6 路由

说明：在 R2 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 过滤

（1）配置只留 3002:1:1:11::/64 网段

r2(config)#ipv6 prefix-list abc permit 3002:1:1:11::/64

r2(config)#ipv6 router eigrp 10

r2(config-rtr)#distribute-list prefix-list abc in f0/0

r2(config-rtr)#

（2）查看过滤后的路由表情况

```
r2#sh ipv6 route eigrp
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
EX   3002:1:1:11::/64 [170/409600]
     via FE80::C200:9FF:FE54:0, FastEthernet0/0
r2#
```

说明：路由表中只剩想要的网段，说明过滤成功。

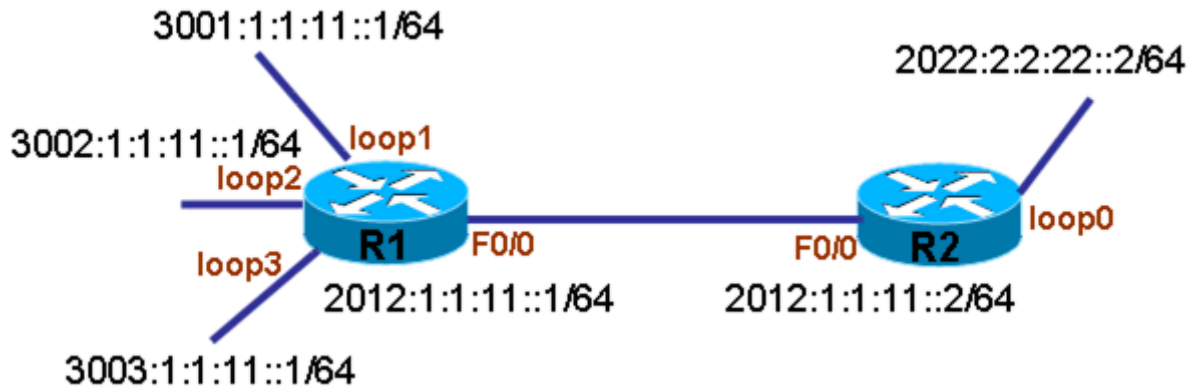
IPv6 BGP

普通情况下配置的 BGP，是用来传递 IPv4 路由的，所传递的信息是 IPv4 的协议，如果要让 BGP 传递其它路由或协议，这就需要将 BGP 扩展为支持更多协议的 BGP，如扩展 BGP 支持 IPv6 协议，支持 vpnv4，这样的支持多协议的 BGP，称为 Multiprotocol BGP，即 MP-BGP，

要配置 MP-BGP，就需要为除 IPv4 之外的协议单独创建 address-family，但是建立 BGP 邻居和正常情况下一样，当邻居建立之后，还得到 address-family 下活动，这是 MP-BGP 的特性，而需要发布的网段，也需要到 address-family 下发布。传递单播 IPv6 的 address-family 应该是 address-family ipv6 unicast，但关键字

unicast 如果省略，默认就是 address-family ipv6 unicast。下面根据以上特征，来配置 MP-BGP 传递 IPv6 路由。

配置 IPv6 MP-BGP



1.初始配置

（1）R1 初始配置：

r1(config)#ipv6 unicast-routing

r1(config)#int f0/0

r1(config-if)#ipv6 address 2012:1:1:11::1/64

r1(config)#int loopback 1

r1(config-if)#ipv6 address 3001:1:1:11::1/64

r1(config)#int loopback 2

r1(config-if)#ipv6 address 3002:1:1:11::1/64

r1(config)#int loopback 3

r1(config-if)#ipv6 address 3003:1:1:11::1/64

（2）R2 初始配置：

r2(config)#ipv6 unicast-routing

r2(config)#int f0/0

r2(config-if)#ipv6 address 2012:1:1:11::2/64

r2(config)#int loopback 0

r2(config-if)#ipv6 address 2022:2:2:22::2/64

2.配置 MP-BGP 中的 IPv6 邻居

说明：所有邻居正常配置，但需要到 IPv6 的 address-family 下激活邻居。

（1）在 R1 上配置 BGP 邻居

r1(config)#router bgp 100

r1(config-router)#bgp router-id 1.1.1.1

r1(config-router)#neighbor 2012:1:1:11::2 remote-as 100

r1(config-router)#address-family ipv6

r1(config-router-af)#neighbor 2012:1:1:11::2 activate

r1(config-router-af)#exit

（2）在 R2 上配置 BGP 邻居

r2(config)#router bgp 100

r2(config-router)#bgp router-id 2.2.2.2

r2(config-router)#neighbor 2012:1:1:11::1 remote-as 100

r2(config-router)#address-family ipv6

r2(config-router-af)#neighbor 2012:1:1:11::1 activate

r2(config-router-af)#exit

3.查看 IPv6 BGP 邻居

（1）在 R1 上查看 IPv6 BGP 邻居

r1#show bgp sum

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2012:1:1:11::2	4	100	5	4	1	0	0	00:01:35	0

r1#

说明：由于配置正确，所以已正常建立 IPv6 BGP 邻居命令。命令 show bgp sum 为隐藏命令。

（2）在 R2 上查看 IPv6 BGP 邻居

r2#show bgp sum

BGP router identifier 2.2.2.2, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2012:1:1:11::1	4	100	5	6	1	0	0	00:02:02	0

r2#

说明：由于配置正确，所以已正常建立 IPv6 BGP 邻居命令。

4.发布 IPv6 路由进 IPv6 BGP

(1)在 R1 上发布路由进 IPv6 BGP

r1(config)#router bgp 100

r1(config-router)#address-family ipv6

r1(config-router-af)#network 3001:1:1:11::/64

(2)在 R2 上发布路由进 IPv6 BGP

r2(config)#router bgp 100

r2(config-router)#address-family ipv6

r2(config-router-af)#network 2022:2:2:22::/64

(3)在 R1 上查看 IPv6 BGP 路由

r1#show bgp all

For address family: IPv6 Unicast

BGP table version is 3, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i2022:2:2:22::/64	2012:1:1:11::2	0	100	0	i
*> 3001:1:1:11::/64	::	0		32768	i

r1#

说明：已成功学习到对方邻居发来的 IPv6 路由。

(4)在 R2 上查看 IPv6 BGP 路由

r2#show bgp all

For address family: IPv6 Unicast

BGP table version is 3, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2022:2:2:22::/64	::	0		32768	i
*>i3001:1:1:11::/64	2012:1:1:11::1	0	100	0	i

r2#

说明：已成功学习到对方邻居发来的 IPv6 路由。

（5）测试网络连通性

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/96/208 ms

r1#

r2#ping 3001:1:1:11::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3001:1:1:11::1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 24/88/200 ms

r2#

说明：由于双方路由学习正常，所以网络连通性正常。

5.重分布 IPv6 网段

说明：将 R1 上的剩余网段重分布进 IPv6 BGP

（1）在 R1 上配置重分布剩余网段进 IPv6 BGP

r1(config)#route-map con permit 10

r1(config-route-map)#match interface loopback 2

r1(config-route-map)#exit

r1(config)#route-map con permit 20

鸿鹄培训+考试群：301283867

```
r1(config-route-map)#match interface loopback 3

r1(config-route-map)#exit


r1(config)#router bgp 100

r1(config-router)#address-family ipv6

r1(config-router-af)#redistribute connected route-map con
```

（2）在 R2 上查看重分布进 IPv6 BGP 的剩余网段

```
r2#show bgp all

For address family: IPv6 Unicast

BGP table version is 11, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

              r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete


   Network          Next Hop            Metric LocPrf Weight Path

*> 2022:2:2:22::/64 ::                   0         32768 i

*>i3001:1:1:11::/64 2012:1:1:11::1       0    100      0 i

*>i3002:1:1:11::/64 2012:1:1:11::1       0    100      0 ?

*>i3003:1:1:11::/64 2012:1:1:11::1       0    100      0 ?

r2#
```

说明：可以看到，R1 上的剩余网段成功被重分布进 RIPng。

6.过滤 IPv6 路由

说明：在 R2 上过滤掉 IPv6 路由，只留想要的网段，使用 distribute-list 对指定邻居进行过滤

（1）配置只留 3002:1:1:11::/64 网段

r2(config)#ipv6 prefix-list abc permit 3002:1:1:11::/64

r2(config)#router bgp 100

r2(config-router)#address-family ipv6

r2(config-router-af)#neighbor 2012:1:1:11::1 prefix-list abc in

（2）查看过滤后的路由表情况

r2#clear bgp ipv6 unicast *

r2#sh bgp all

For address family: IPv6 Unicast

BGP table version is 3, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2022:2:2:22::/64	::	0		32768	i
*>i3002:1:1:11::/64	2012:1:1:11::1	0	100	0	?

r2#

说明：路由表中只剩想要的网段，说明过滤成功。

7.使用链路本地地址建立 IPv6 BGP 邻居

说明：正常情况下，IPv6 BGP 使用全局地址建立邻居，也可以配置使用链路本地地址建立邻居。

（1）在 R1 上配置 IPv6 BGP 用链路本地地址建立邻居

```
r1(config)#router bgp 100
r1(config-router)#neighbor FE80::C200:DFF:FEC8:0 remote-as 100
r1(config-router)#neighbor FE80::C200:DFF:FEC8:0 update-source f0/0
r1(config-router)#address-family ipv6
r1(config-router-af)#neighbor FE80::C200:DFF:FEC8:0 activate
r1(config-router-af)#
```

（2）在 R2 上配置 IPv6 BGP 用链路本地地址建立邻居

```
r2(config)#router bgp 100
r2(config-router)#neighbor FE80::C200:8FF:FE10:0 remote-as 100
r2(config-router)#neighbor FE80::C200:8FF:FE10:0 update-source f0/0
r2(config-router)#address-family ipv6
r2(config-router-af)#neighbor FE80::C200:8FF:FE10:0 activate
r2(config-router-af)#
```

（3）查看邻居建立情况

```
r1#show bgp sum
BGP router identifier 1.1.1.1, local AS number 100
BGP table version is 1, main routing table version 1
```

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd

FE80::C200:DFF:FEC8:0

            4  100      6       7        1    0    0 00:01:30        0

r1#
```

```
r2#show bgp sum

BGP router identifier 2.2.2.2, local AS number 100

BGP table version is 1, main routing table version 1
```

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd

FE80::C200:8FF:FE10:0

            4  100      8       7        1    0    0 00:02:20        0

r2#
```

说明：从结果中看出，双方 IPv6 BGP 已成功使用链路本地地址建立邻居

（4）查看路由学习情况

```
r1#sh bgp all

For address family: IPv6 Unicast

BGP table version is 14, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

*>i2022:2:2:22::/64 FE80::C200:DFF:FEC8:0

0 100 0 i

*> 3001:1:1:11::/64 :: 0 32768 i

*> 3002:1:1:11::/64 :: 0 32768 ?

*> 3003:1:1:11::/64 :: 0 32768 ?

r1#

r2#show bgp all

For address family: IPv6 Unicast

BGP table version is 6, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

*> 2022:2:2:22::/64 :: 0 32768 i

*>i3001:1:1:11::/64 FE80::C200:8FF:FE10:0

0 100 0 i

*>i3002:1:1:11::/64 FE80::C200:8FF:FE10:0

0 100 0 ?

*>i3003:1:1:11::/64 FE80::C200:8FF:FE10:0

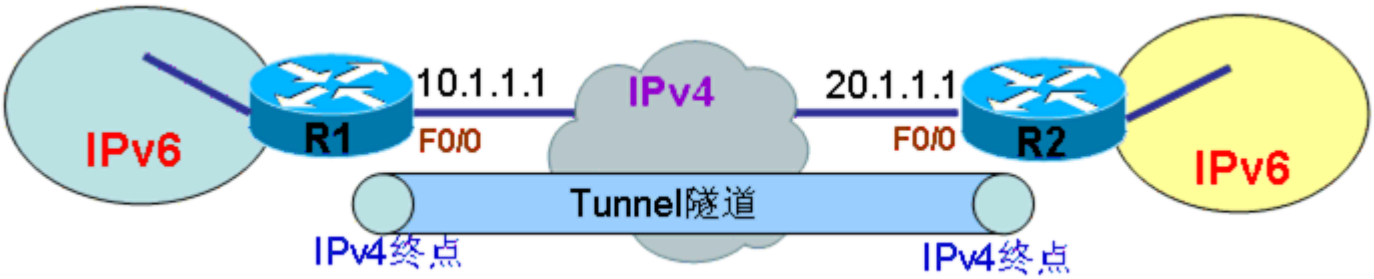
0 100 0 ?

r2#

说明：从结果中看出，双方 IPv6 BGP 已成功学习到相互的 IPv6 路由条目。

、

IPv6 隧道



如上图所示，当两个 IPv6 网络需要通信时，如果中间需要穿越 IPv4 网络，而由于 IPv4 网络中只能识别 IPv4 包头，并不能为 IPv6 数据提供正确的路径传输，这时就需要在 IPv4 网络中为 IPv6 创建一条隧道，来提供 IPv6 在 IPv4 中的传递，这样的隧道，就是把 IPv6 的数据全部封装在 IPv4 中，将 IPv4 当作链路层来传递的隧道形式，称为覆盖型隧道（Overlay Tunnels ）。由于隧道是建立在 IPv4 基础上的，隧道又必须有起点和终点来明确隧道的路径，所以覆盖型隧道的起点和终点最好是使用 IPv4 地址，有时必须是 IPv4 地址，并且隧道在传输 IPv6 数据时，也应该在隧道的两端添加 IPv6 地址，来完成两端 IPv6 网络的通信。隧道的起点和终点必须同时支持 IPv4 和 IPv6。

当前在 IOS 中支持的覆盖型隧道共有以下几中：

Manual

点对点，只传递 IPv6 数据包。

模式为：ipv6ip

Generic routing encapsulation (GRE)

点对点，可以传递多种协议。

模式为：gre ip

IPv4-compatible

点到多点的，思科不建议使用。

模式为：ipv6ip auto-tunnel

6to4

点到多点的，使用地址为 2002::/16。

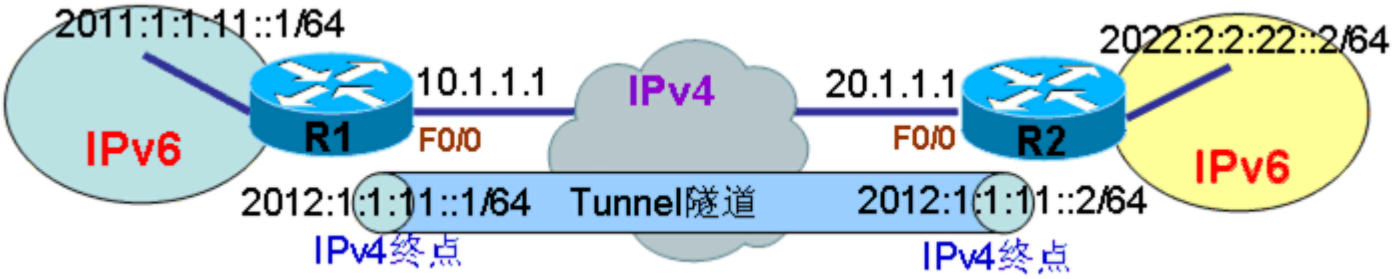
模式为：ipv6ip 6to4

Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

是点到多点的。

模式为：ipv6ip isatap

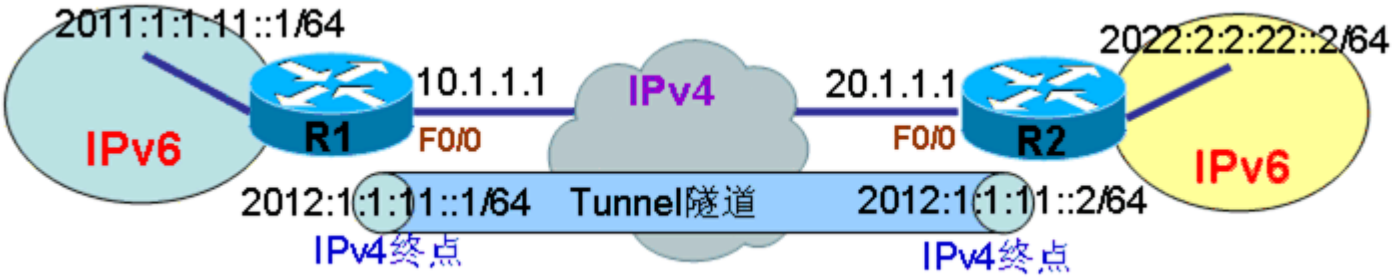
以上隧道中，所有隧道的源均为 IPv4 地址，但是只有点对点隧道的终点为 IPv4 地址,其它都不需要。更多的是，点对点隧道必须要有 IPv6 地址，点对点隧道如下图：



注： CCIE 考试中，IPv6 隧道的考点为 Manual 类型的隧道。

配置 IPv6 隧道：

说明： 原来 R1 上的 IPv6 网络无法与 R2 上的 IPv6 网络通信，通过配置 IPv6 隧道之后，在隧道与相应 IPv6 接口上启用 IPv6 路由协议，如 OSPFv3，使得两端 IPv6 网络可以通信。



1.初始配置

（1）R1 初始配置：

```
r1(config)#int f0/0

r1(config-if)#ip address 10.1.1.1 255.255.255.0

r1(config-if)#exi
```

r1(config)#ip route 0.0.0.0 0.0.0.0 f0/0

r1(config)#

r1(config)#ipv6 unicast-routing

r1(config)#int loopback 0

r1(config-if)#ipv6 address 2011:1:1:11::1/64

（2）R2 初始配置：

r2(config)#int f0/1

r2(config-if)#ip add 20.1.1.1 255.255.255.0

r2(config-if)#exit

r2(config)#ip route 0.0.0.0 0.0.0.0 f0/0

r2(config)#ipv6 unicast-routing

r2(config)#int loopback 0

r2(config-if)#ipv6 address 2022:2:2:22::2/64

测试 IPv4 连通性：

r1#ping 20.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/108/196 ms

r1#

说明：R1 与 R2 之间的 IPv4 连通性正常。

2.配置 Manual 类型的 IPv6 隧道

（1）在 R1 上配置 IPv6 隧道

说明：配置的 IPv6 隧道的类型为 Manual 类型，即模式为 ipv6ip

r1(config)#int tunnel 0

r1(config-if)#ipv6 address 2012:1:1:11::1/64

r1(config-if)#tunnel source f0/0

r1(config-if)#tunnel destination 20.1.1.1

r1(config-if)#tunnel mode ipv6ip

（2）在 R2 上配置 IPv6 隧道

r2(config)#int tunnel 0

r2(config-if)#ipv6 address 2012:1:1:11::2/64

r2(config-if)#tunnel source f0/0

r2(config-if)#tunnel destination 10.1.1.1

r2(config-if)#tunnel mode ipv6ip

(3)查看两端隧道情况

r1#sh ipv6 interface brief tunnel 0

Tunnel0 [up/up]

FE80::A01:101

2012:1:1:11::1

r1#

r2#show ipv6 interface brief tunnel 0

Tunnel0 [up/up]

FE80::1401:101

2012:1:1:11::2

r2#

（4）测试隧道连通性：

r1#ping 2012:1:1:11::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2012:1:1:11::2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 212/279/332 ms

r1#

说明：隧道通信正常。

3.配置 IPv6 路由协议

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

说明：在路由器之间启用 IPv6 路由协议，在隧道之上传递两端 IPv6 网络信息。

(1)在 R1 上配置 OSPFv3

r1(config)#ipv6 router ospf 2

r1(config-rtr)#router-id 1.1.1.1

r1(config)#int loopback 0

r1(config-if)#ipv6 ospf network point-to-point

r1(config-if)#ipv6 ospf 2 area 0

r1(config)#int tunnel 0

r1(config-if)#ipv6 ospf 2 area 0

(2)在 R2 上配置 OSPFv3

r2(config)#ipv6 router ospf 2

r2(config-rtr)#router-id 2.2.2.2

r2(config)#int loopback 0

r2(config-if)#ipv6 ospf network point-to-point

r2(config-if)#ipv6 ospf 2 area 0

r2(config)#int tunnel 0

r2(config-if)#ipv6 ospf 2 area 0

4.查看结果

（1）查看邻居状态

R1：

r1#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
2.2.2.2	1	FULL/ -	00:00:31	14	Tunnel0

r1#

R2：

r2#show ipv6 ospf neighbor

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
1.1.1.1	1	FULL/ -	00:00:38	14	Tunnel0

r2

说明：两端 OSPFv3 邻居正常。

（2）查看路由信息

R1：

r1#show ipv6 route ospf

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 2022:2:2:22::/64 [110/11112]

via FE80::1401:101, Tunnel0

r1#

R2：

r2#show ipv6 route ospf

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 2011:1:1:11::/64 [110/11112]

via FE80::A01:101, Tunnel0

r2#

说明：两端通过 OSPFv3 学习到的 IPv6 路由正常。

5.测试两端 IPv6 网络通信情况

（1）测试 R1 到 R2 端 IPv6 网络的通信情况

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 120/181/268 ms

r1#

说明：由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

（2）测试 R2 到 R1 端 IPv6 网络的通信情况

r2#ping 2011:1:1:11::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2011:1:1:11::1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 88/164/284 ms

r2#

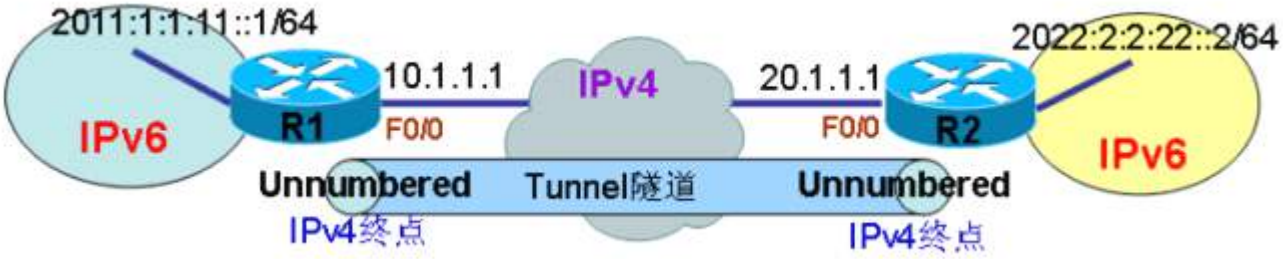
说明：由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

隧道借用地址

说明：在现有 IPv4 网络上创建覆盖型 IPv6 隧道，隧道的起点和终点都使用了 IPv4 地址来定义，然后要使隧道运行正常，使隧道具有路由协议的连接功能，需要赋予隧道两端 IPv6 地址，从而提供 IPv6 的连通性，而隧道两端的 IPv6 地址可以不属于同一网段，当然属于同一网段是最好的选择。无论隧道两端的 IPv6 地址是否属于同一网段，IPv6 路由协议都是可以正常使用的。如果隧道两端的 IPv6 地址属于

同一网段，那么一切正常，隧道两端的地址可以相互 ping 通，路由协议也无须更多操作，而当隧道两端的 IPv6 地址不属于同一网段时，那么两端的地址是无法 ping 通的，但 IPv6 路由协议可以照常使用，这时，路由协议需要将隧道的地址当作额外路由进行重新通告一次。

下面在创建隧道时，将隧道两端的 IPv6 地址改为无编号借用地址（unnumbered），这时两端地址不属于同网段，再使用 IPv6 路由协议连通两端 IPv6 网络。



1.初始配置

r1:

r1(config)#int f0/0

r1(config-if)#ip address 10.1.1.1 255.255.255.0

r1(config-if)#exi

r1(config)#ip route 0.0.0.0 0.0.0.0 f0/0

r1(config)#

r1(config)#ipv6 unicast-routing

r1(config)#int loopback 0

r1(config-if)#ipv6 address 2011:1:1:11::1/64

R2

r2(config)#int f0/1

r2(config-if)#ip add 20.1.1.1 255.255.255.0

r2(config-if)#exit

r2(config)#ip route 0.0.0.0 0.0.0.0 f0/0

r2(config)#ipv6 unicast-routing

r2(config)#int loopback 0

r2(config-if)#ipv6 address 2022:2:2:22::2/64

测试 IPv4 连通性：

r1#ping 20.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/108/196 ms

r1#

说明：R1 与 R2 之间的 IPv4 连通性正常。

2.配置 unnumbered 地址的 IPv6 隧道

（1）在 R1 上配置 IPv6 隧道

```
r1(config)#int tunnel 0
r1(config-if)#ipv6 unnumbered loopback 0
r1(config-if)#tunnel source f0/0
r1(config-if)#tunnel destination 20.1.1.1
r1(config-if)#tunnel mode ipv6ip
```

（2）在 R2 上配置 IPv6 隧道

```
r2(config)#int tunnel 0
r2(config-if)#ipv6 unnumbered loopback 0
r2(config-if)#tunnel source f0/0
r2(config-if)#tunnel destination 10.1.1.1
r2(config-if)#tunnel mode ipv6ip
```

(3)查看两端隧道情况

```
r1#show ipv6 interface brief tunnel 0
Tunnel10                [up/up]
    FE80::A01:101
    unnumbered (Loopback0)
r1#
```

```
r2#show ipv6 interface brief tunnel 0
```

Tunnel10 [up/up]

FE80::1401:101

unnumbered (Loopback0)

r2#

（4）测试隧道连通性：

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r1#

说明：由于隧道两端地址不属于同一网段，所以没有对端路由信息，无法 ping 通。

3.配置 IPv6 路由协议

(1)在 R1 上配置 OSPFv3

r1(config)#ipv6 router ospf 10

r1(config-rtr)#router-id 1.1.1.1

r1(config)#int loopback 0

r1(config-if)#ipv6 ospf network point-to-point

r1(config-if)#ipv6 ospf 10 area 0

r1(config)#int tunnel 0

r1(config-if)#ipv6 ospf 10 area 0

(2)在 R2 上配置 OSPFv3

r2(config)#ipv6 router ospf 10

r2(config-rtr)#router-id 2.2.2.2

r2(config)#int loopback 0

r2(config-if)#ipv6 ospf network point-to-p

r2(config-if)#ipv6 ospf 10 area 0

r2(config)#int tunnel 10

r2(config-if)#ipv6 ospf 10 area 0

4.查看结果

（1）查看邻居状态

r1#show ipv6 ospf neighbor

Neighbor ID Pri State Dead Time Interface ID Interface

2.2.2.2 1 FULL/ - 00:00:37 15 Tunnel10

r1#

r2#show ipv6 ospf neighbor

Neighbor ID Pri State Dead Time Interface ID Interface

1.1.1.1 1 FULL/ - 00:00:36 15 Tunnel10

r2#

说明：两端 OSPFv3 邻居正常。

（2）查看路由信息

R1：

r1#show ipv6 route ospf

IPv6 Routing Table - 7 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 2022:2:2:22::/64 [110/11112]

via FE80::1401:101, Tunnel10

r1#

R2：

r2#show ipv6 route ospf

IPv6 Routing Table - 7 entries

鸿鹄培训+考试群：301283867

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

O 2011:1:1:11::/64 [110/11112]

via FE80::A01:101, Tunnel10

r2#

说明：两端通过 OSPFv3 学习到的 IPv6 路由正常。

5.测试两端 IPv6 网络通信情况

r1#ping 2022:2:2:22::2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2022:2:2:22::2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 120/181/268 ms

r1#

说明：由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

（2）测试 R2 到 R1 端 IPv6 网络的通信情况

r2#ping 2011:1:1:11::1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2011:1:1:11::1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 88/164/284 ms

r2#

说明：由于隧道成功建立，并且通过 OSPFv3 正常学习到路由，到对端 IPv6 网络通信正常。

IPv6 组播

概述

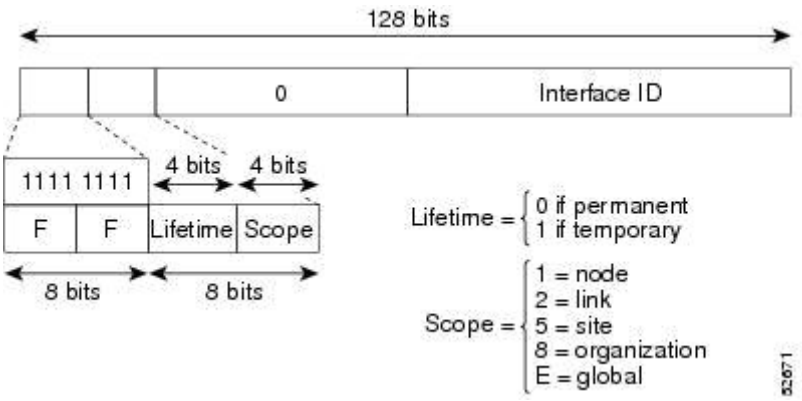
在理解 IPv6 组播之前，必须了解 IPv4 组播，了解 IPv4 PIM，了解 IGMP，这些知识在本篇不再详细讲述，相关详细内容，请参见 IPv4 组播部分。

要启用 IPv6 组播，必须先开 IPv6 单播。

IPv6 组播地址：

IPv6 组播地址的范围是 FF00::/8 (1111 1111)。

因为一个正常的 IPv6 地址包含 128 位，在 IPv6 组播地址中，第一段共 16 位的格式被拆分成三部分：第一部分共 8 位，全部为 1，即使用 FF 来表示。第二部分共 4 位，表示组播地址的存活期，如果为 0 表示永久，如果为 1 表示临时。第三部分共 4 位，表示组播地址的范围，分为 node, link, site, organization,global 分别表示为 1, 2, 5, 8,E，除了此五种以外，0 和 F 为保留范围，而其它全部称为未分配，建议使用未分配的地址范围。组播地址的表示格式如下图：



在 IPv6 中没有广播地址，只有组播，所以使用组播代替广播。

无论是路由器还是主机，所有 IPv6 接口默认加入 FF02::1 ，

而所有路由器的 IPv6 接口默认加入 FF02::2。

MLD （Multicast Listener Discovery）

在 IPv6 组播中，MLD 协议与 IPv4 组播中的 IGMP 协议功能相同，是用于发现接收者的协议。

路由器发送 MLD 查询消息来确认接收者，而主机发送 MLD 报告来加入一个组，主机可以在同一时间属于多个组。

MLD 共有两个版本，ver1 和 ver2，

MLD ver 1 是基于 IPv4 IGMP v2

MLD ver 2 是基于 IPv4 IGMP v3

IOS 同时使用两个。

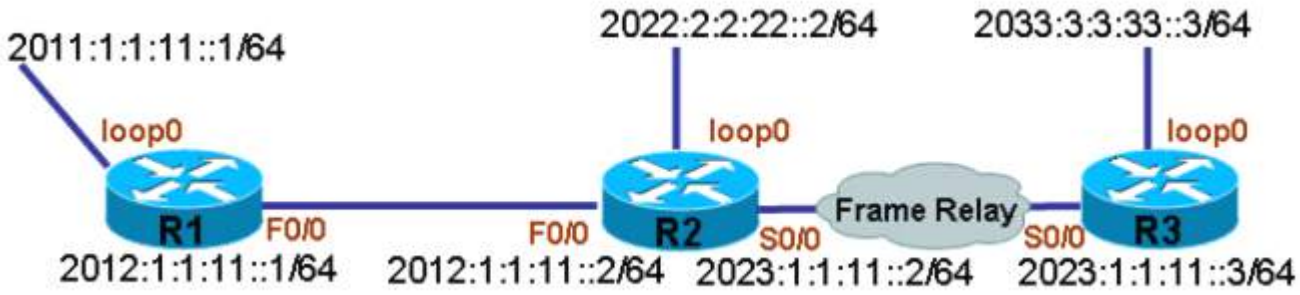
PIM

IPv6 PIM 的功能同 IPv4 PIM，而 IPv6 PIM 只使用 SM（稀疏）模式，所以网络中必须存在 RP，而 RP 的位置可以通过静态配置和 BSR 通告两种方法确认。

在配置 IPv6 PIM 时，当开启 IPv6 组播功能后，所有正常启用 IPv6 功能的接口

自动开启 IPv6 PIM，所以 IPv6 PIM 无须手工配置；并且须明白 DR 在组播中的作用，详细内容请参见 IPv4 组播部分。

配置 IPv6 组播



1.初始配置

（1）R1 初始配置：

r1(config)#ipv6 unicast-routing

r1(config)#ipv6 router ospf 10

r1(config-rtr)#router-id 1.1.1.1

r1(config)#int f0/0

r1(config-if)#ipv6 address 2012:1:1:11::1/64

r1(config-if)#ipv6 ospf 10 area 0

r1(config)#int loopback 0

r1(config-if)#ipv6 address 2011:1:1:11::1/64

r1(config-if)#ipv6 ospf network point-to-point

r1(config-if)#ipv6 ospf 10 area 0

（2）R2 初始配置：

r2(config)#ipv6 unicast-routing

r2(config)#ipv6 router ospf 10

r2(config-rtr)#router-id 2.2.2.2

r2(config)#interface f0/0

r2(config-if)#ipv6 address 2012:1:1:11::2/64

r2(config-if)#ipv6 ospf 10 area 0

r2(config)#int loopback 0

r2(config-if)#ipv6 address 2022:2:2:22::2/64

r2(config-if)#ipv6 ospf network point-to-point

r2(config-if)#ipv6 ospf 10 area 0

r2(config)#int s0/0

r2(config-if)#encapsulation frame-relay

r2(config-if)#no frame-relay inverse-arp

r2(config-if)#no arp frame-relay

r2(config-if)#ipv6 address 2023:1:1:11::2/64

r2(config-if)#frame-relay map ipv6 2023:1:1:11::3 203 broadcast

鸿鹄培训+考试群：301283867

r2(config-if)#frame-relay map ipv6 FE80::213:1AFF:FE2F:380 203 broadcast

r2(config-if)#ipv6 ospf network point-to-point

r2(config-if)#ipv6 ospf 10 area 0

（3）R3 初始配置：

r3(config)#ipv6 unicast-routing

r3(config)#ipv6 router ospf 10

r3(config-rtr)#router-id 3.3.3.3

r3(config)#interface loopback 0

r3(config-if)#ipv6 address 2033:3:3:33::3/64

r3(config-if)#ipv6 ospf network point-to-point

r3(config-if)#ipv6 ospf 10 area 0

r3(config)#int s0/0

r3(config-if)#encapsulation frame-relay

r3(config-if)#no frame-relay inverse-arp

r3(config-if)#no arp frame-relay

r3(config-if)#ipv6 address 2023:1:1:11::3/64

r3(config-if)#frame-relay map ipv6 2023:1:1:11::2 302 broadcast

r3(config-if)#frame-relay map ipv6 FE80::213:1AFF:FE2F:1200 302 broadcast

r3(config-if)#ipv6 ospf network point-to-point

r3(config-if)#ipv6 ospf 10 area 0

2.开启 IPv6 组播

（1）在 R1 上开启 IPv6 组播

r1(config)#ipv6 multicast-routing

（2）在 R2 上开启 IPv6 组播

r2(config)#ipv6 multicast-routing

（3）在 R3 上开启 IPv6 组播

r3(config)#ipv6 multicast-routing

3.配置 IPv6 PIM

说明：在配置 IPv6 PIM 时，当开启 IPv6 组播功能后，所有正常启用 IPv6 功能的接口自动开启 IPv6 PIM，所以 IPv6 PIM 无须手工配置。

（1）查看 R1 上的 PIM 状态

r1#show ipv6 pim neighbor

Neighbor Address Interface Uptime Expires DR pri Bidir

FE80::213:1AFF:FE2F:1200 FastEthernet0/0 00:00:43 00:01:31 1 (DR) B

r1#

说明：由于 IPv6 单播和 IPv6 组播已正常开启，所以 IPv6 PIM 邻居也已经正常建立。

（2）查看 R2 上的 PIM 状态

r2#show ipv6 pim neighbor

Neighbor Address Interface Uptime Expires DR pri Bidir

FE80::212:D9FF:FEF9:C8A0 FastEthernet0/0 00:00:56 00:01:18 1 B

FE80::213:1AFF:FE2F:380 Serial0/0 00:00:55 00:01:28 10 (DR) B

r2#

说明：由于 IPv6 单播和 IPv6 组播已正常开启，所以 IPv6 PIM 邻居也已经正常建立。

（3）查看 R3 上的 PIM 状态

r3#show ipv6 pim neighbor

Neighbor Address Interface Uptime Expires DR pri Bidir

FE80::213:1AFF:FE2F:1200 Serial0/0 00:01:09 00:01:37 1 B

r3#

说明：由于 IPv6 单播和 IPv6 组播已正常开启，所以 IPv6 PIM 邻居也已经正常建立。

4.配置 MLD

说明：因为 MLD 的功能同 IGMP，所以配置的目的为加入某个组。

（1）在 R1 上配置加入组 ff04::1

说明：建议使用未分配的组播地址范围。

r1(config)#interface loopback 0

r1(config-if)#ipv6 mld join-group ff04::1

5.配置静态 RP

说明：静态配置所有设备的 RP 为 R1 的 loopback 0

（1）在 R1 上配置静态 RP

r1(config)#ipv6 pim rp-address 2011:1:1:11::1

（2）在 R2 上配置静态 RP

r2(config)#ipv6 pim rp-address 2011:1:1:11::1

（3）在 R3 上配置静态 RP

r3(config)#ipv6 pim rp-address 2011:1:1:11::1

6.查看结果

（1）查看 R1 上的 RP 情况

r1#show ipv6 pim group-map ff04::

FF00::/8*

SM, RP: 2011:1:1:11::1

RPF: Tu2,2011:1:1:11::1 (us)

Info source: Static

Uptime: 00:01:15, Groups: 1

r1#

说明：由于已手工配置 RP，所以 RP 正常。

（2）查看 R2 上的 RP 情况

r2#show ipv6 pim group-map ff04::

FF00::/8*

SM, RP: 2011:1:1:11::1

RPF: Fa0/0,FE80::212:D9FF:FEF9:C8A0

Info source: Static

Uptime: 00:01:40, Groups: 0

r2#

说明：由于已手工配置 RP，所以 RP 正常。

（3）查看 R3 上的 RP 情况

r3#show ipv6 pim group-map ff04::

FF00::/8*

SM, RP: 2011:1:1:11::1

RPF: Se0/0,FE80::213:1AFF:FE2F:1200

Info source: Static

Uptime: 00:01:41, Groups: 0

r3

说明：由于已手工配置 RP，所以 RP 正常。

7.测试组播通信情况

（1）测试 R1 的组播通信情况

r1#ping ff04::1

Output Interface: Loopback0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2011:1:1:11::1

Reply to request 0 received from 2011:1:1:11::1, 16 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 0 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/3/16 ms

5 multicast replies and 0 errors.

r1#

说明：在测试组播时，出口必须全部写出。从结果中可以看出，由于 PIM 已成功建立，RP 已正确学到，所以组播通信正常。

（2）测试 R2 的组播通信情况

r2#ping ff04::1

Output Interface: FastEthernet0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2012:1:1:11::2

Reply to request 0 received from 2011:1:1:11::1, 4 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 0 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms

5 multicast replies and 0 errors.

r2#

说明：从结果中可以看出，由于 PIM 已成功建立，RP 已正确学到，所以组播通信正常。

（3）测试 R3 的组播通信情况

r3#ping ff04::1

Output Interface: Serial0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2023:1:1:11::3

Request 0 timed out

Request 1 timed out

Request 2 timed out

Request 3 timed out

Request 4 timed out

Success rate is 0 percent (0/5)

0 multicast replies and 0 errors.

r3#

说明：R3 的组播无法 ping 通，由于 PIM 的 DR 选举问题。原因请参见 IPv4 组播部分。

8.解决组播通信问题

说明：由于 R3 与 R2 之间为多路访问，DR 位置错误，所以组播无法通信，切换 DR 位置以解决组播通信问题。

（1）改 R2 为网络中的 DR

r2(config)#interface s0/0

r2(config-if)#ipv6 pim dr-priority 100

（2）查看当前网络中 DR 情况

r3#show ipv6 pim neighbor

Neighbor Address Interface Uptime Expires DR pri Bidir

FE80::213:1AFF:FE2F:1200 Serial0/0 00:10:02 00:01:24 100 (DR) B

r3#

说明：DR 已成功变为 R2。

（3）测试 R3 的组播通信情况

r3#ping ff04::1

Output Interface: Serial0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2023:1:1:11::3

Reply to request 0 received from 2011:1:1:11::1, 68 ms

Reply to request 0 received from 2011:1:1:11::1, 80 ms

Reply to request 1 received from 2011:1:1:11::1, 64 ms

Reply to request 1 received from 2011:1:1:11::1, 76 ms

Reply to request 2 received from 2011:1:1:11::1, 65 ms

Reply to request 2 received from 2011:1:1:11::1, 77 ms

Reply to request 3 received from 2011:1:1:11::1, 68 ms

Reply to request 3 received from 2011:1:1:11::1, 80 ms

Reply to request 4 received from 2011:1:1:11::1, 124 ms

Reply to request 4 received from 2011:1:1:11::1, 168 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/87/168 ms

10 multicast replies and 0 errors.

r3#

说明：修改 DR 后，R3 的组播通信正常。

配置 BSR

说明：前面通过手工静态配置 RP 来使组播正常通信，下面通过配置 BSR 来自动选举 RP。

配置 R1 的 loopback0 为 C-BSR 和 C-RP。

1.配置 C-BSR

（1）配置 R1 的 loopback0 为 C-BSR

r1(config)#ipv6 pim bsr candidate bsr 2011:1:1:11::1

2.配置 C-RP

（1）配置 R1 的 loopback0 为 C-RP

r1(config)#ipv6 pim bsr candidate rp 2011:1:1:11::1

3.查看结果

（1）查看 R1 上的 RP 情况

r1#show ipv6 pim group-map ff04::

FF00::/8*

SM, RP: 2011:1:1:11::1

RPF: Tu2,2011:1:1:11::1 (us)

Info source: BSR From: 2011:1:1:11::1(00:01:31), Priority: 192

Uptime: 00:00:58, Groups: 1

r1#

说明：由于 BSR 配置正确，所以 RP 正常。

（2）查看 R2 上的 RP 情况

r2#show ipv6 pim group-map ff04::

FF00::/8*

SM, RP: 2011:1:1:11::1

RPF: Fa0/0,FE80::212:D9FF:FEF9:C8A0

Info source: BSR From: 2011:1:1:11::1(00:02:20), Priority: 192

Uptime: 00:01:09, Groups: 1

r2#

说明：由于 BSR 配置正确，所以 RP 正常。

（3）查看 R3 上的 RP 情况

r3#show ipv6 pim group-map ff04::

FF00::/8*

SM, RP: 2011:1:1:11::1

RPF: Se0/0,FE80::213:1AFF:FE2F:1200

Info source: BSR From: 2011:1:1:11::1(00:02:09), Priority: 192

Uptime: 00:01:20, Groups: 0

r3#

说明：由于 BSR 配置正确，所以 RP 正常。

4.测试组播通信情况

（1）测试 R1 的组播通信情况

r1#ping ff04::1

Output Interface: Loopback0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2011:1:1:11::1

Reply to request 0 received from 2011:1:1:11::1, 8 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 0 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/8 ms

5 multicast replies and 0 errors.

r1#

说明： RP 已正确学到，所以组播通信正常。

（2）测试 R2 的组播通信情况

r2#ping ff04::1

Output Interface: FastEthernet0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2012:1:1:11::2

Reply to request 0 received from 2011:1:1:11::1, 12 ms

Reply to request 1 received from 2011:1:1:11::1, 0 ms

Reply to request 2 received from 2011:1:1:11::1, 8 ms

Reply to request 3 received from 2011:1:1:11::1, 0 ms

Reply to request 4 received from 2011:1:1:11::1, 0 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/4/12 ms

5 multicast replies and 0 errors.

r2#

说明： RP 已正确学到，所以组播通信正常。

（3）测试 R3 的组播通信情况

r3#ping ff04::1

Output Interface: Serial0/0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FF04::1, timeout is 2 seconds:

Packet sent with a source address of 2023:1:1:11::3

Reply to request 0 received from 2011:1:1:11::1, 120 ms

Reply to request 0 received from 2011:1:1:11::1, 132 ms

Reply to request 1 received from 2011:1:1:11::1, 100 ms

Reply to request 1 received from 2011:1:1:11::1, 116 ms

Reply to request 1 received from 2011:1:1:11::1, 128 ms

Reply to request 2 received from 2011:1:1:11::1, 100 ms

Reply to request 2 received from 2011:1:1:11::1, 116 ms

Reply to request 3 received from 2011:1:1:11::1, 100 ms

Reply to request 3 received from 2011:1:1:11::1, 116 ms

Reply to request 4 received from 2011:1:1:11::1, 101 ms

Reply to request 4 received from 2011:1:1:11::1, 117 ms

Success rate is 100 percent (5/5), round-trip min/avg/max = 100/113/132 ms

11 multicast replies and 0 errors.

r3#

说明： RP 已正确学到，所以组播通信正常。

IPv6 邻居发现协议（ND 协议）

由于 IPv6 ND 协议中，几乎所有有用功能均为自动开启，无须手工干预，所以在此不再详细解释 ND 协议的运行过程，详细内容请自行参考 Cisco 文档中 IOS 12.4 T 部分。

MPLS / MPLS_VPN

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

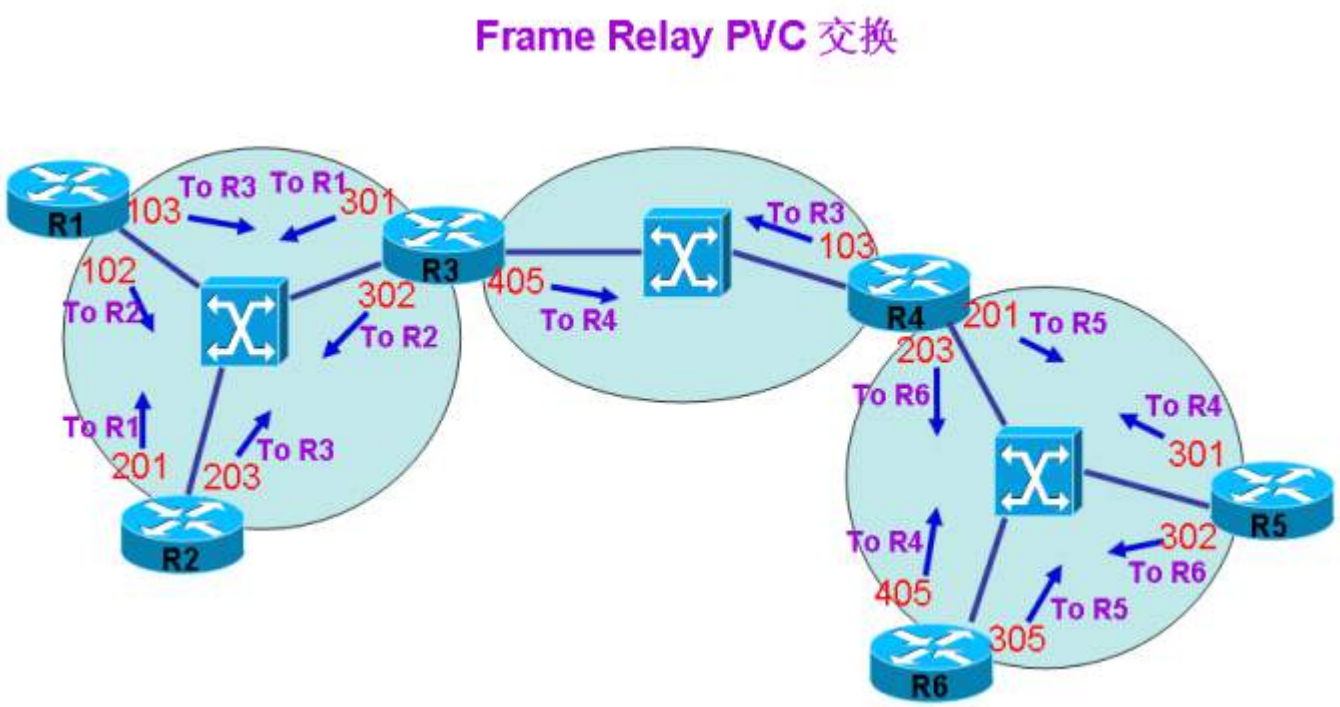
鸿鹄培训+考试群：301283867

鸿鹄培训+考试群：301283867

MPLS

非 IP 包头交换过程

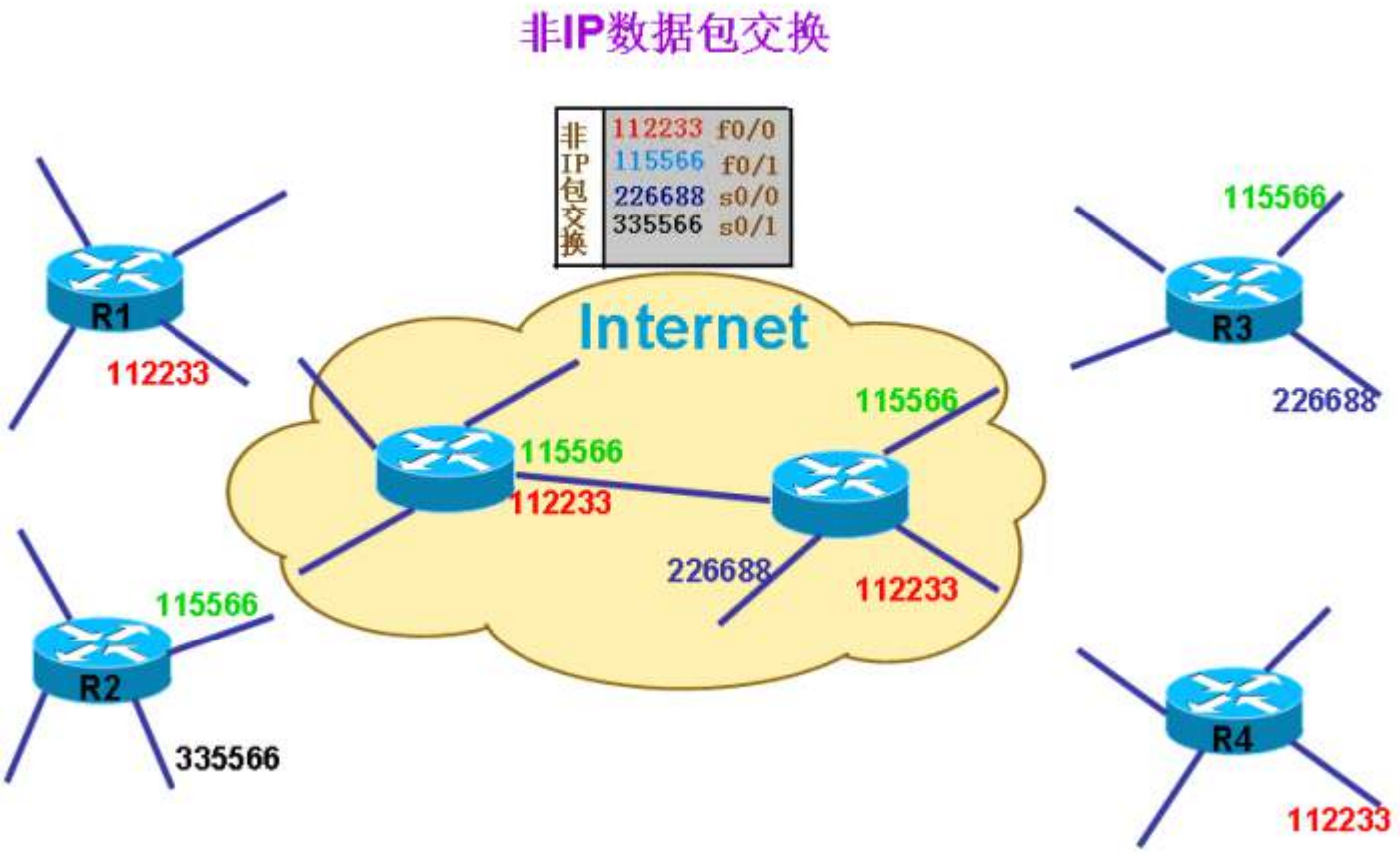
1.帧中继 PVC 交换方式



在我们现有的网络当中，IP 数据包网络占绝大部分，这样的 IP 数据包网络，在网络设备传递数据包时，是根据数据包的 IP 包头信息进行交换的，也就是网络设备根据包头中的目标 IP 地址，来决定从哪个接口转发出去。所以在数据包当中，指导设备正确转发数据包的就是 IP 地址信息，而 IP 地址只是数据包的一个标识而已。既然数据包的包头信息能够指导设备正确转发，那数据包的包头只要能够被设备正确接受，就能够做出正确的转发决策，正因为如此，网络就产生了其它不同于 IP 数据包交换方式，比如我们应当熟悉的帧中继网络（Frame Relay）。在帧中继网络中，很明显，帧中继设备（帧中继交换机）在决定数据包该从哪个接口被发出去时，查看的就是包头 PVC 号码，而不是 IP 地址，这个 PVC 号码，对于帧中继设备来说，就关系到这个数据包应该从哪个接口被转发出去。如上图所示，在帧中继交换机中，只关系数据包的 PVC 号码是多少，只要看到这个号码，就知道该从哪个接口出去，等数据包到了下一台交换机之后，下一台交换机也做同样的操作，即查看数据包的 PVC 号码后就从相应接口发出去，但是不同数据包的 PVC 号码肯定应该是不一样的，因为同一个 PVC 号码，对于交换机来说，都应该从同一个接口出去。所以说一台交换机上的每个接口相关联的 PVC 号码都是不一样的。但是，这台交

换机用过的 PVC 号码，到了下一台交换机之后，还是可以使用的，因为前面一台交换机根据某个 PVC 号码对数据包转发之后，自己再根据数据包的 PVC 进行转发，与前面是不冲突的，因为是各自关联好的。从这里也可以想象出，一个数据包经过一台帧中继交换机之后，到了下面一台，数据包的 PVC 是应该被设备进行重新改写才交换的，因为不可能一个 PVC 经过 N 台交换机还是一样的。所以可以得出一个理论就是，帧中继数据包的包头信息（即 PVC 号码）仅一跳有效，也就是本地有效，不同交换机之间，包头信息可重复，在这里用过的 PVC 号码，在别的交换机上也可能出现一样的，只要保证在单台交换机唯一就可以了，所以每次经过一个交换机之后，需要重新改写包头信息。

2. 非 IP 数字包头交换方式



下面来看一下既不是 IP 数据包交换，也不是帧中继交换的网络，那么这样的网络给数据包写上什么样的包头来指导设备正确转发呢？就写一个号码而已，我们暂且称它为非 IP 数字包头交换。在这样的网络中，设备看到包头中的这个数字，就知道该从哪个接口转发出去，每台都是一样的。这种网络跟帧中继交换相同的是设备也是查看一个号码，而不同的是，一个数据包写上一个号码之后，永远都不会被任

何设备改写，直接传到目的地为止，可想而知，网络中任意两台主机之间，他们的号码必须是唯一的，因为每台交换机都要根据这个号码来做出转发，如果两个数据包的号码相同，那么所有的交换机都做同样的转发，结果就导致这些数据包被发到同一台设备。这样的数字网络，全球中，每两点之间仅有一个号码表示，第一个点的数据包头写上此号码，必定是发到第二个点，不可能发送到第三个点，因为第一个点和第三个点，会使用另外一个号码，所以此号码为两点唯一，网络设备中有每两点（即每个号码）的出口信息，收到任何一个包，都能不看 IP 地址而根据此号码选择从相应接口发出去，每台设备执行相同的过程，即可完成任意两点间的传输。此交换方式其实并没有在计算机网络中应用，但是我们使用的电话网络，就是这种交换方式，即任何两台电话之间打电话，号码唯一，不可能有相同号码，如果你拨打电话，别人也拨一个电话号码，你们拨的号码如果是一样的，那肯定就打到同一个人那里去了。所以要实现此交换方式，网络中所有设备需要计算出任意两个点之间的号码，每一个号码都是唯一的，不可重复，与到目的地的相应出口作对应，生成转发表。但是如果全球计算机网络使用这样的方式，那就是任何一台设备为任何一台主机计算路径时，都要所有全球的设备共同参与，如果不全部都参与，就可能和没参与的计算出重复号码，可想而知工作量之庞大。

3.交换方式总结

以上两种交换方式，都是在不看 IP 地址（IP 包头），只看号码的情况下，做出的交换选择。

可以仔细想一下，在使用帧中继交换时，因为一个 PVC 号码只要保证单台设备不重复就可以了，这个号码跟接口是关联着的，也就是说一个数据包写上的 PVC 号码，这个 PVC 号码的范围只要比交换机的接口多就行，比如范围是 1024，所以帧中继交换的包头，号码不是很庞大，也就是说包头并不是很长。而非 IP 包交换的网络中，因为每两点之间都要有独立的号码，所以如果网络中有 10 亿个点，那么这个号码的范围就应该比 10 亿还要大，所以非 IP 包交换，数据包头肯定要比帧中继的包头大。

但是从结论中，我们能不能说哪个好，哪个不好呢？当然不能，因为帧中继的包头虽然比非 IP 的包头要小，但是每经过一台设备都要重新改写，也就是说帧中继网络中，设备都在不停地为每个数据包改写 PVC 号码，这也是巨大的工作量啊。而非 IP 包头虽然要大一些，但是这个号码写好之后，就永远不会再变了，只要中间的设备看到号码直接转发就行，不用改写了。

4.MPLS（多协议标签交换）

在使用 IP 包交换网络的时候，人们总是认为设备要根据 IP 地址查路由表做出转发决定，觉得这样很耗时，总想着寻找一种新的交换技术来代替 IP 包交换。最初就考虑使用数字号码的方式来代替 IP 地址，从上面介绍的交换方式中，由于第二种非 IP 包交换技术，需要在网络中对任意两点计算出一个全球唯一的号码，因为一个号码即代表了两个点之间的传输，如果其它的点之间的号码和别人出现重复，那么数据的走向也就会发生错误。当前没有研发出协议敢保证计算出任意两点的号码一定是唯一的。

在帧中继的交换中，只要保证每个网段（每台交换机）之间的 PVC 号码唯一即可，因为经过每个网段（每个交换机）号码都会重新修改。此交换方式也可避免设备检查数据包的 IP 地址来作做转发决定。（但也不要忘记，这种交换方式的弊端，在似乎节省了时间的同时，其实也浪费了许多时间。）

而当前人们认为效率比较高的 MPLS（多协议标签交换）方式，它的数据交换思想则倾向帧中继的交换方式，即认为设备在查看 IP 地址之后做出转发决定，会比较慢，会耗更多时间，则给数据包写上了额外的号码，根据此号码而不看 IP 地址，便能找出相应出口从而转发出去，这正是标签交换，而 MPLS 称此额外的号码为标签。在此可以看出，MPLS 的交换号码（标签）并不是全球唯一，只是每个网段唯一，或者说是每跳唯一，所以，经过一跳之后，此号码对下一跳设备毫无意义，经过一跳之后，此号码要修改成对下一跳有意义的号码，让其根据号码做出转发决定。由此可见，MPLS 的标签交换，是每跳都会改写标签，因为根据标签，便能够做出转发决定，所以省略了查看 IP 地址的过程，被人们认为比 IP 交换要快。

而 MPLS 根据自己的标签交换，需要给数据包先写上自己的标签，然后设备才能查看标签之后就转发，此标签是需要在原有的数据包的基础上加进去的，并没有将以前的包头删除，MPLS 的标签加在了第二层帧的帧头之后，但又在第三层数据包的包头之前，而 MPLS 不管是什么协议的数据包，不管以前的包头是什么，都能够在包中加入对自己有利的标签，所以称 MPLS 交换为多协议标签交换。

MPLS 优势：

1.（不正确的理由）：因为 IP 在路由当中，总是根据目的地址在路由表中查找目标网段，并且逐条匹配最优路径，速度慢。

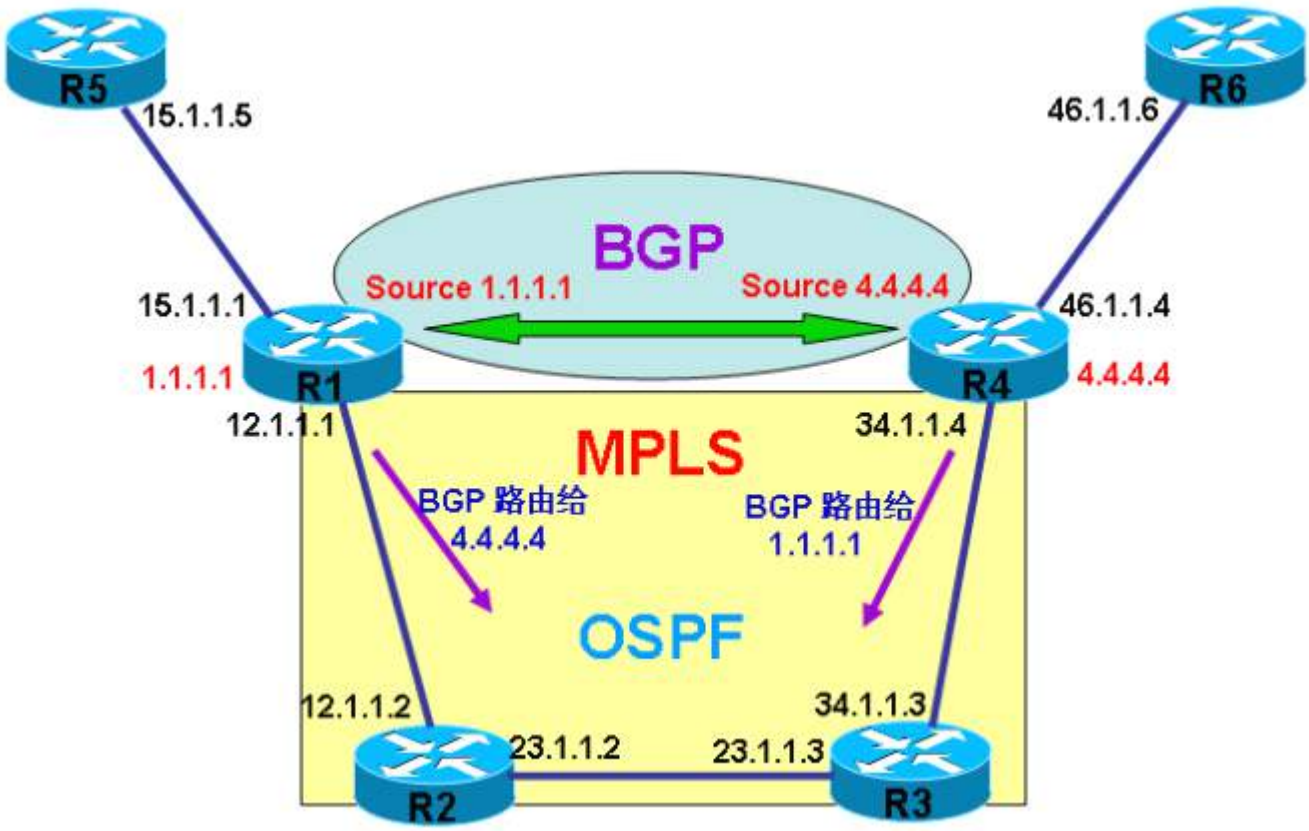
2. MPLS 只根据数据包顶部标签来查找并转发，速度快。

结论：由于现在设备采用 ASIC（专用集成电路）交换，所以速度并不慢，而 MPLS 借签了帧中继交换方式，在数据包每经过一台设备时，都要重新封装，所以 MPLS 在速度上，并不是优势。

但 MPLS 可以给数据包加上标签，以做流量控制，这是优势。MPLS 还可以承载各种协议，如 IPv4,IPv6，以太网，HDLC，PPP，以及其它第二层帧。

注：MPLS 在骨干网中传输任意第二层帧的特征被称为 MPLS 的任意传输（AToM）。

MPLS 中的 BGP



BGP 在决定一个数据包该如何被转发出去，是通过查找 IP 地址在路由表中的下一跳，有时不能避免这下一跳不是跟自己直连的，但是 BGP 只要知道如何到达那个下一跳即可，所以 BGP 路由的下一跳，也许自己不清楚，但是只要 IGP 的路由能帮助自己到达下一跳就行。在大型的核心网络中，我们完全可以设计出网络这边的 BGP 路由器，它的下一跳在网络那边，那么如何到达网络那边的下一跳，中间就可以使用 IGP 去完成，只要中间的设备能够帮助 BGP 到达最终下一跳地址就足够了，所以这样的网络，需要 BGP 协议的只是网络的边缘路由器，而中间的路由器，只要做一件事，那就是帮 BGP 找下一跳，就不用启用 BGP 了，这就大大节省了系统资源。而 MPLS 的标签交换，就可以用在这样的网络中，来为 BGP 寻找下一跳，也就是 MPLS 只要为 BGP 路由的下一跳打上标签，能够帮助 BGP 找到下一跳，那么其它的问题，都不是问题，其它的路由，BGP 就能够自己完成。

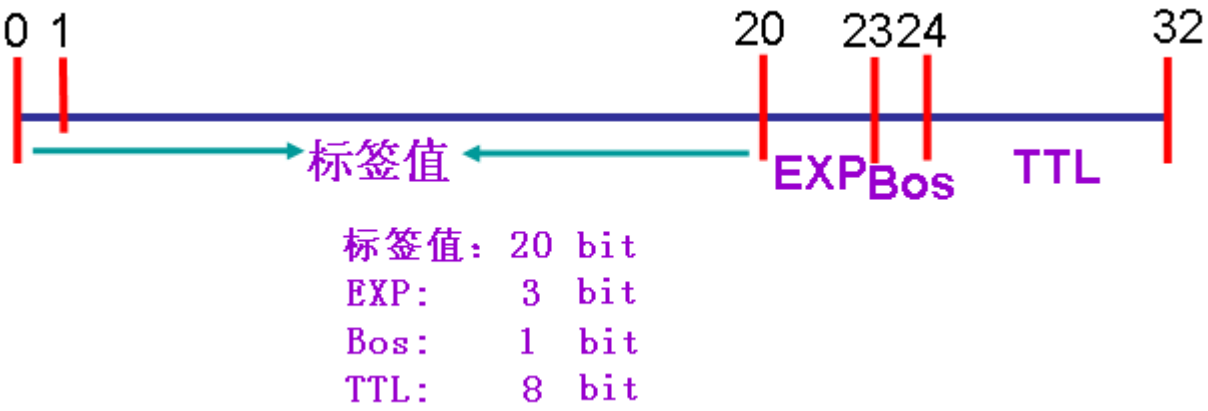
略： MPLS 流量工程，路径是由首端路由器指定的，所以又称为基于源的路由。

思科 MPLS 历史

最初 Cisco 在 IP 报文顶部加入标签时，称其为标记交换（tag switching），而标记，现在改叫标签了，为每个路由条目分配好标记并写进去，所以这就需要一张表来指导标记交换，称为转发信息库（TFIB），每一台标记交换路由器查看数据包入站的标记，并转为出站标记后发出去。

注： 思科第一个支持标记交换的 IOS 就支持流量工程，就是资源预留协议（RSVP）

MPLS 标签



一个标签由 32 个 bit 组成

前 20 为标签值，范围从 0 到 2 的 20 次方减一，即 1048575。

其中前 16bit 不能随便定义，有特定含义，从 21 到 23bit 共 3 位试验用（EXP），用于 QOS。

第 24 比特是栈底 Bos 位，值为 0，如果是栈底，就为 1，标签栈中，标签数量没有限制。

从 25 到 32 共 8 个 bit 是 TTL

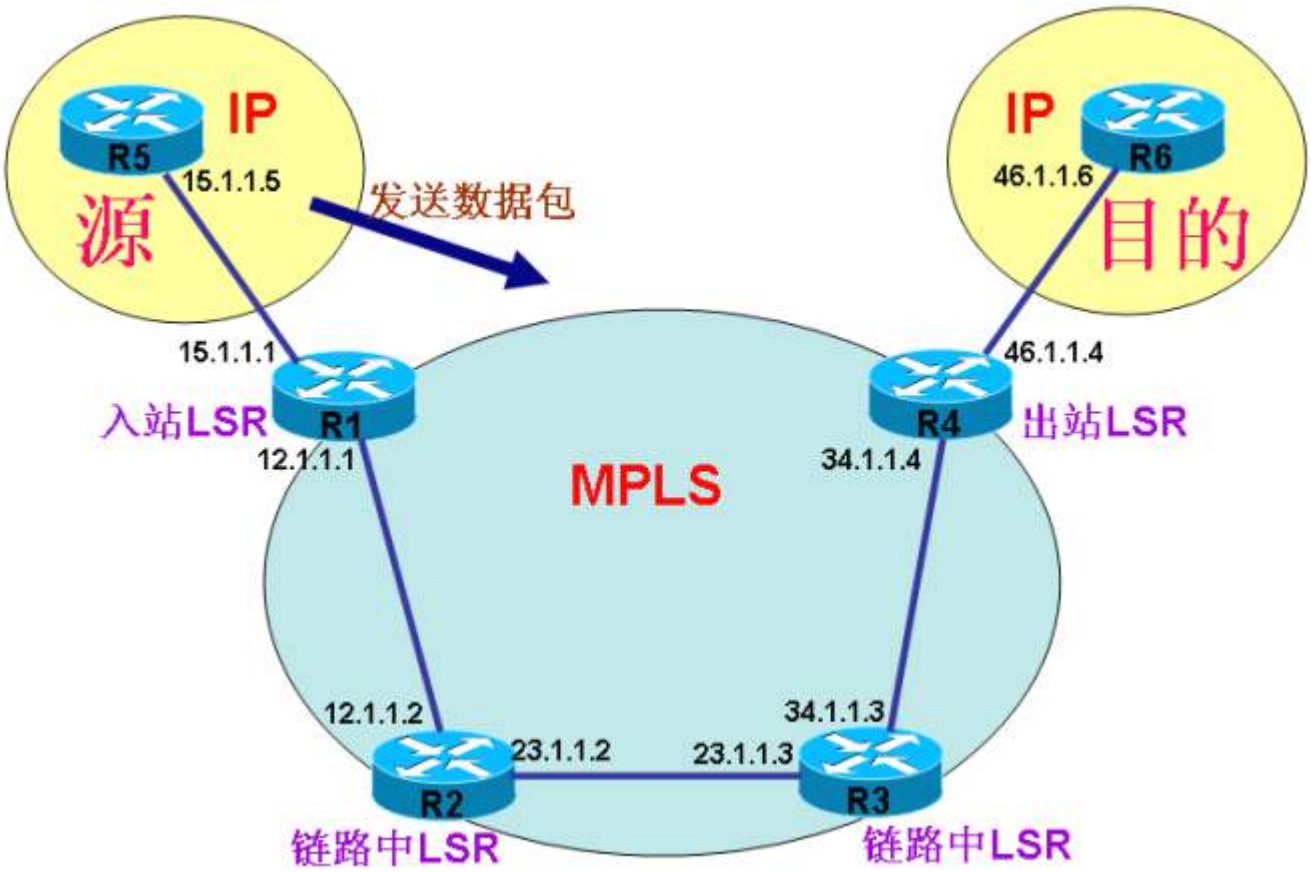
MPLS 标签栈

MPLS 路由器对数据包可能添加一个标签，也可能添加多个标签，这些标签集合起来叫做标签栈，第一个为顶部标签，最后一个为底部标签，中间数量可以无限，底部标签 BOS 总是 1，否则就是 0。而在数据包传输过程中，设备只根据第一个顶部标签来决定怎么转发。

有些情况是需要两个标签的，两个典型是 MPLS VPN 和 AToM。MPLS VPN 要用两个标签，是因为在骨干中传输时，用一个，等出了骨干，再用另外一个。

那么设备收到一个 MPLS 标签数据包时，又怎么知道这个数据包是要查 IP 路由表来决定转发呢，还是查标签表做出转发呢？那是因为标签是在第二层帧和第三层数据包之间，第二层会在数据链路层的协议字段写上新的值，以说明后面是一个带有 MPLS 标签的报文，所以设备能够做出正确的转发决策。

MPLS 设备类型



能够理解 MPLS 标签并根据标签转发数据包的路由器称为 LSR

共有以下 3 种 LSR：

入站 LSR：接收没有标签的数据包，打上标签并发出

出站 LSR：接收带有标签的数据包，移除标签，并发出，出站和入站 LSR 都是边缘 LSR，所以它们同时连接了 IP 网络和 MPLS 网络。

链路中 LSR：接收到带标签的数据包，对其进行操作，然后按正确的接口交换出去，所以链路中的 LSR 只进行标签转发。

LSR 操作过程

LSR 可以执行三种操作：提取，添加和交换

提取，即从标签栈的顶部移除一个或多个标签，移除全部标签是出站 LSR 必须做的。

添加，向报文添加标签，如果没有标签，就加新的，入站 LSR 必须做的。

交换，收到一个有标签的报文，用新的标签交换到顶部，再发出，是链路中的 LSR 做的。

注：在 MPLS VPN 中，入站和出站 LSR 就是提供商边缘 PE，链路中 LSR 就是 P，术语 PE 和 P 在没有运行 MPLS VPN 时，也可使用。

标签交换路径 LSP

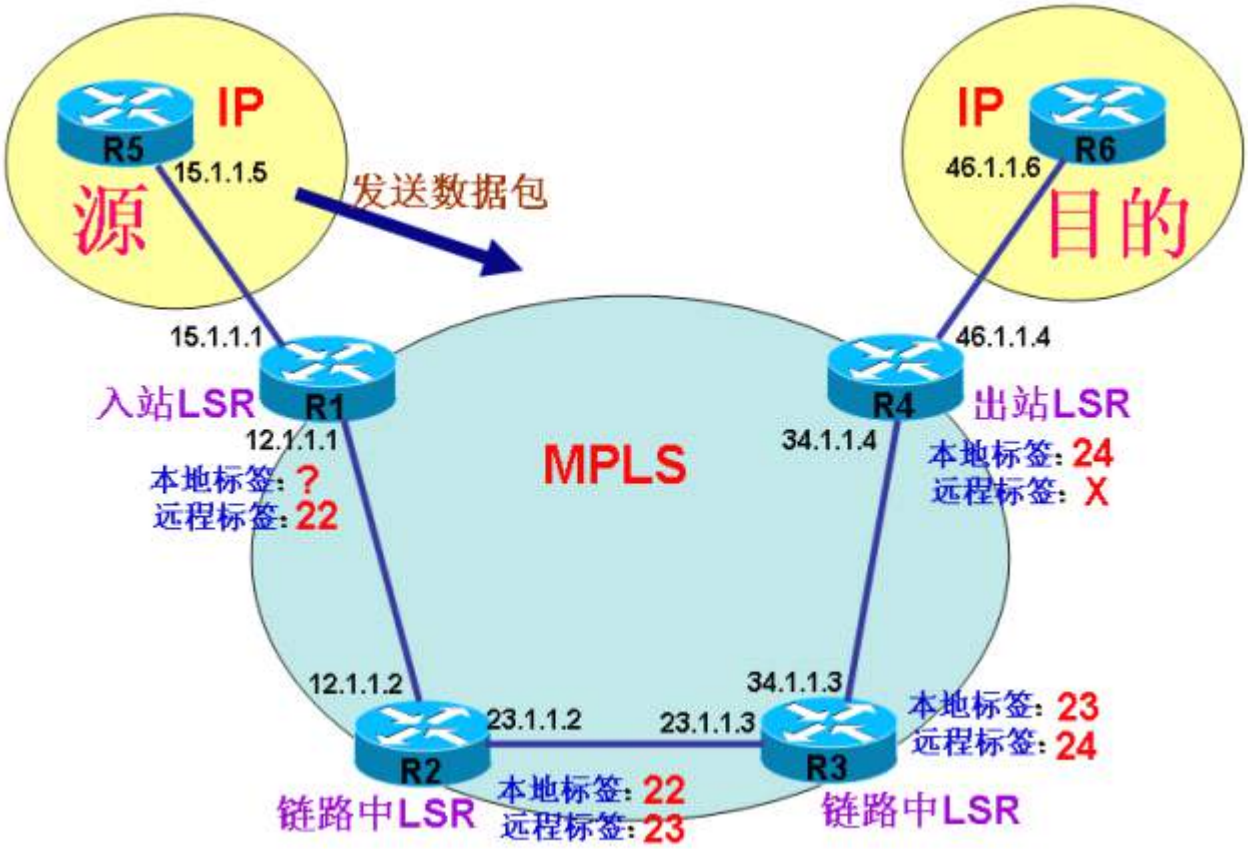
LSP 指的就是一个源到最终目的需要经过的路径，而在路径中，是要被修改多次标签，比如一个源到目的要分别被改为 20，50，35，68，那么也可以简单认为这两个点之间的 LSP 是 20-50-35-68。LSP 是 LSR 在 MPLS 网络中转发标签数据后的产生的，是标签报文穿越 MPLS 网络的路径。LSP 不需要记住，只需要知道是什么。LSP 中第一台 LSR 就是入站 LSR，最后一台就是出站 LSR，之间的就是链路中 LSR。

转发等价类（FEC）

FEC 可以认为是同一条路由，或者说是到达目标主机的路径是相同的，或者说是相同转发路径的数据流。同一个 FEC，所有标签都相同，并不是拥有相同标签的报文都是同一个 FEC，可能 EXP 值不同。报文属于哪个 FEC，由入站 LSR 决定。FEC 和 LSP 一样只需要知道是什么，不用记住。

注：一条 FEC 可以包含多个流，但不是一个流一个 FEC，比如一台主机在看新浪的网页，这是一个流，又在看新浪的视频，这又是一个流，这两个流在新浪发给远程主机时，走的路径应该是相同的，所以一个 FEC 有多个流，但是每个流并没有属于单独的 FEC。

MPLS 标签交换过程



虽然一条路由可以打了多标签，但是中间的 LSR 只根据最顶部的标签便可以做出转发。但是，每台 LSR 的转发表里都会为一条路由显示两个标签，一个是本地标签，一个是远程标签，要显示两个标签，是因为一台 LSR 收到数据包之后，就查看它的顶部标签，如果这个标签是某一条相应的本地标签，那么就从相应的接口发出去，同时在发出去的时候，就将数据包的顶部标签改为与这个本地标签对应的远程标签，这是每台 LSR 在传输时都必须改的，因为改了相应的远程标签，对下一台 LSR 做出正确转发是有帮助的。所以 LSR 自己对于某一条路由，别人要给数据打上什么标签，它才能够正确转发，它是要明确告诉邻居的，最后结果就是每一台 LSR 都会将这个能指导自己做过正确转发的标签发给邻居，邻居就认定这个发来的标签是远程标签，那么邻居在发出去数据包之前，都会改成这个远程标签，最后就能按正确路径转发了。

上图中，比如 R4 要看到数据包中有标签 24，就能够正确往 R6（目的地）的方向发，那么这个标签 24 对于 R4 来说就是本地标签，那么别人 LSR 把数据包发给

R4 之前，就得为它打上标签 24 才发出来，所以 R4 就把这个标签 24 告诉给它的邻居 R3，邻居 LSR 收到标签之后，就认为是远程标签，所以 R3 认为远程标签是 24，而自己要拿到一个数据后，是什么标签自己就会改成 24 发给 R4 呢，自己也会产生一个本地标签，可以看到，R3 产生的本地标签是 23，数据包中只要顶部标签是 23，就能够让 R3 把标签改成 24 后发给 R4，所以 R3 也把这个标签 23 发给了邻居 R2，邻居 R2 也同样为路由产生一个本地标签，是 22，也就是说 R2 收到数据包标签为 22 的，就改成 23，然后传给下一跳 R3，最后 R2 也把对自己有利的标签 22 发给了 R1，那么 R1 就知道发给 R2 之前，要把标签改成 22，要不然 R2 就会转发出错的。

而因为 R1 是入站 LSR，所以从外面发给它的数据包都是 IP 包，是不会有标签的，所以它是第一个向数据包加标签的 LSR，可以看出只要它一开始往数据包里加上标签 22 后发给 R2，最后就能一跳一跳地被发到 R6，因为中间 LSR 的标签大家都是协商好的。在数据包到达 R4 时，R4 就会将标签全部移除后发给 R6，因为是进入 IP 网络，没有必要再打标签发出去。要说明的是，该为一个路由条目产生什么样的本地标签，是 LSR 自己计算的，没有规则可言，从前面的传输过程可以看出，在 MPLS 网络中，LSR 每次收到数据包，都要将标签改成对下一跳有利的标签才转发出去，这个对下一跳有利的标签就是自己看到的远程标签，也称为出口标签，这个标签就是邻居告诉自己的，因为看到什么标签才能正确转发，邻居是知道的，所以告诉给其它邻居之后，才能保证最终路径的正确。

打标签

所有 LSR，根据路由表，将数据包打上标签，发出去，打的这个标签不能乱打，是有意义的，因为下一跳路由器要根据这个标签做出自己的决策，收到的路由器将顶部标签（上一跳路由器加的标签）去除，再加入出站标签，然后再下一跳路由器重复之前的动作，所以每一台路由器都需要对路由条目的标签达成共识，要不然这台路由器为数据包打上标签，给下一跳路由器，却不是按设想的接口发出去的，就不能到达目标网络。因此，每一台 LSR 必须明确哪个出站标签来交换哪个入站标签，标签对于邻接 LSR 来说是本地有效，没有全局意义，这和帧中继的 PVC 是一个道理。所以，基于这些，需要有一种标签分发协议来为所有的 LSR 分发一个正确的标签，只有这样，LSR 才能根据标签将数据包正确地发到目标网络。

标签分发方式

标签分发有两种方式：

1．在已存在的 IP 路由协议中分发标签。

2．使用一种独立的协议来分发标签。

1.在现有的路由协议中分发

距离矢量路由协议，比如 EIGRP 很好做，直接绑在前缀上，因为这样的路由协议可以随意修改路由条目的内容。

链路状态路由协议，比如 OSPF 和 ISIS 就比较难，因为是发链路状态，而链路状态必须毫无更改地发给邻居，更改链路状态数据包是违背原理。

所以也就没有 IGP 做标签分发的工作，但是 BGP 却可以同时发前缀和标签（注: BGP 发标签是有条件限制的）。

2.标签分发协议

最终建议使用独立的分发协议，不影响路由协议，但需要在 LSR 上额外运行协议。

以下是可以分发标签的协议

标记分发协议（TDP）

标签分发协议（LDP）

资源预留协议（RSVP）

TDP 是思科专有的标记分发协议，已经被 LDP 所取代。

LDP 是 IETF 开发，功能广泛，所以只涉及 LDP，因为 TDP 已不用了。

RSVP 只用在 MPLS TE 中。

标签分发协议 LDP

对于 IP 路由表中每一条 IGP 前缀，每台 LSR 都会进行本地捆绑，也就是为路由条目加上一个标签，这个标签称为本地标签，到时收到一个数据包后，看到顶部标签如果是自己所拥有的本地标签，那么就根据这个标签从相应接口转发出去，所以邻居把数据包发给自己时，必须在数据包上写好自己对自己有利的标签，那么要怎样才能让邻居写上一个能让自己看了就正确转发的标签呢，这就得自己把这个标签告诉邻居，自己把本地标签发给邻居后，这个标签对于邻居来说就称为远程标签，每当邻居要发送数据包给自己的时候，就先把数据包的顶部标签改成远程标签，也就是改成之前发给邻居的标签，邻居改好标签后，把数据包发给我们，我们就能够做出正确转发了，那么邻居也会像我们一样，把自己的本地标签再发给它们的邻居，因为他要把数据正确发给我们，也是由它的邻居在数据包上写好标签告诉它的。所以在邻居和邻居之后，是要大家协商好每条路由该写上什么标签后发给邻居。LSR 把自己生成的本地标签，和邻居发过来的远程标签，不管是用的着的还是用不着的，统统都保存在一张标签表里，这个表称为 LIB 表（标签信息库）。LDP 就是用来发送标签的协议。

已经介绍过，MPLS 的标签，就像帧中继的 PVC 号码一样，只是每台设备唯一的，但是 MPLS 的标签，在设备上保持唯一还分两种，基于设备唯一和基于接口唯一。

如果基于设备，就是一条路由在一台设备上只有一个唯一的标签。

如果基于接口，就是一条路由在一台设备上，是每个接口都有一个唯一的标签。也就是说标签只要能保证在每个接口上是唯一的即可，在整台设备上可以重复多次。

但 LSR 可能有多个远程标签，因为可能有多个邻居 LSR。

路由协议 EIGRP 会将所有邻居发给自己的路由条目存放在拓朴表里，再从拓朴表里选中最优最好的放到路由表也供自己使用，当路由表中的条目失效后，再从拓朴表中拿出次优的使用。而 MPLS 标签交换的 LSR 也像 EIGRP 那样，会把所有邻居发来的标签都存放在 LIB 表里（就像 EIGRP 的拓朴表），然后从路由会条目的多个标签中选择一个最优的使用，这个选择方法可以通过 IGP 路由表，选到的下一跳是谁，那就用谁发来的标签，被选中的正在使用的标签，全部都是存放在 LFIB（标

签转发信息库）表里的，就像 EIGRP 的路由表。

注：IOS 中，LDP 不会为 BGP 的 IPv4 前缀捆绑标签。

标签的选择都是根据 IGP 最优路径。

标签也可以不是 LDP 分发，比如 TE 中，由 RSVP 分发，在 MPLS VPN 中，由 BGP 分发。

因为 MPLS 的标签是加在数据包的二层帧头之上，三层包头之下，三层包头，被认为是上层协议，也就是有效负载,中间的 LSR 并不知道上层协议类型，因为标签不会写，但它也不需要知道，自己只会根据标签来做出转发决定；但是出站 LSR 需要知道，所以会为 FEC 分配一个本地标签，以用作报文的入站标签，这样就可以了解有效负载了。

LSR 在查看标签时，是要看基于接口还是设备，如果是基于接口，不能单看标签，还要看接口，如果是基于设备的那就只看标签。

注:在 IOS 中，所有的标签交换控制的 ATM（LC-ATM）接口都采用基于接口的模式，其它通通基于设备，也就是说 CCIE R&S 的考生，只需要关心基于设备的标签即可。

标签分发模式

LSR 在向邻居分发标签的时候，有三个需要注意的地方，这三个地方分为：

1．标签分发模式

2．标签保持模式

3．LSP 控制模式

1.标签分发模式：

是用来定义标签该什么时候发给邻居，分为两种方式：

（1）下游被动 DOD 模式

（2）下游主动 UD 模式

（1）在 DOD，即被动模式中，是 LSR 请求下游（路由表的下一跳）为某条路由分发标签，也就是说一台 LSR 并不知道某些路由自己该写上什么标签后发给下一跳，所以这时就去问邻居要，要来的自己就作为该路由的远程标签存放。

（2）在 UD，即主动模式中，LSR 不需要为路由请邻居请求标签，标签是邻居会主动发过来的，不用请求，所以在 UD 模式中，LSR 发现一条路由，就马上将自己的本地标签发送给邻居作为远程标签。

在此可以得出一个结论，在 DOD 中，LSR 会向路由条目的下一跳请求标签，所以 LIB 只显示一个远程标签。在 UD 中，因为可能有多个邻居发送标签过来，所以一条路由可以看到多个标签。

注：IOS 除了 LC-ATM，全部使用 UD 模式，也就是说有多个标签从邻居发来。

2.标签保存模式

用来定义 LSR 在将标签保存时，该保存多久，分两种方式：

（1）自由的标签保持模式（LLR）

（2）保守的标签保持模式（CLR）

（1）在 LLR 中，LSR 将所有的标签存放在 LIB 中，然后使用的放到 LFIB 中，不使用的也保存在 LIB 中，当路由变化时，马上从 LIB 中找到新的。

（2）在 CLR 中，LSR 将用到的标签放入 LFIB 之后，不会在 LIB 中保存任何标签。

注：IOS 中，除了 LC-ATM 接口，其它所有都使用 LLR，也就是能在 LIB 中看到标签。

3.LSP 控制模式

用来定义 LSR 什么时候应该为一条路由创建标签，创建出的这个标签就是自己的本地标签，发给邻居之后，邻居就称其为远程标签。分两种创建方式：

（1）独立于 LSP 的控制模式

（2）非独立于 LSP 的控制模式

（1）LSR 可以独立于其它 LSR 创建本地标签，称为独立模式，路由器在路由表中发现一个路由，就马上为该路由创建一个标签。

（2）在非独立模式时，LSR 只有意识到它是某 FEC 的出站 LSR 时，或者从下一跳收到某路由的标签时，才会为路由条目创建本地标签，然后发给邻居作为远程标签，邻居收到后，然后又会再创建了发给它的邻居。

由上可以看出，独立于 LSP 的模式在 LSP 中还没有让所有的 LSR 完成标签，有些 LSR 就开始标签转发，所有这些数据包有可能不能被正确转发，有可能被丢弃。而非独立的模式，只有从邻居收到标签了，开始自己的标签，所以自己使用标签转发后，下一跳邻居肯定是能接受的，不可能因为下一跳不认识标签而被丢弃 。

注:IOS 使用独立的 LSP 控制模式，也就是说发现一条 FEC，就马上创建标签，ATM 除外。

在标签转发中，LSR 查看标签前 20bit，并在 LFIB 中查找相应的值。

在 IP 转发中，查看 IP 地址，在 CEF 表中做出转发。

路由器可以查看第二层头部的协议字段，就知道是标签报文，然后查找 LFIB，则带标签出去，所以查 CEF 就等于是查 IP，则不带标签出去。

路由器要为路由条目打上标签，就必须有功能支持改写数据包包头，CEF 是唯一一种可以用于标签报文转发模式的，CEF 是可以改写数据包包头的，所以启用 MPLS 时，必须在路由器上开 CEF，否则无标签。

比如查看路由 10.1.1.0 在 CEF 中的操作，可以使用命令

show ip cef 10.1.1.0 查看这条路由的过程

MPLS 负载均衡

在 IPv4 存在多条相同 metric 出口时，标签的出站也会对应多个接口，出站的标签可以是相同的，也可以不同的；如果下一跳是同台设备，肯定相同，是不同设备

会不同，也就独立分配。

当 LSR 中某条路由有多个下一跳，如果是有标签的和无标签的，无标签的不走，是考虑到 MPLS VPN 中数据会丢包，因为在 MPLS VPN 网络中，P 路由器是没有两边私有网络的 IP 路由的，所以无法路由，最终造成丢包。

MPLS 未知标签

通常 LSR 只接收和发送带标签的并且能理解的数据包，因为某些原因，标签没找到的话，IOS 是默认采用丢弃行为，如果找不到标签也传，也不能保证别人能传不丢弃，所以就自己开始丢弃。

MPLS 保留标签

MPLS 标签范围中，并不是所有的标签都是可以随便用的，有些是保留的，范围是 0-15，有特殊作用，0 是显式空（null）,3 是隐式空，1 是路由器报警标签，14 是 OAM 报警，其它还没定义。下面是某些保留标签的重要用途：

隐式空 3 标签

在 MPLS 网络中，P 路由器是完全按照标签交换的，而边缘路由器 PE 是同时连接了 MPLS 网络和 IP 网络，因此，一个数据包在 MPLS 网络中传输到 PE 路由器的时候，PE 路由器的工作是：结束标签交换过程，从而转入 IP 网络，而转入 IP 网络就要执行 IP 地址的查找。那么从此可以看出，一个标签数据包到达 PE 路由器之后，PE 路由器第一步开始根据数据包的标签去查找 LFIB 表，通过查找 LFIB 之后发现已经不再是标签交换了，第二步就马上转入查 IP 路由表，最终在 IP 路由表里查到了结果，从 IP 网络中发出去。很明显，PE 路由器既然最后不可能使用标签交换，而要使用 IP 交换的，又何必去查了 LFIB 表才知道结果呢。所以就考虑到一个方法，能不能让 PE 的上一跳路由器不要为数据包打标签，直接改成 IP 数据包就行了，这样上一跳路由器没有写标签，那么 PE 在收到数据包之后，就能马上查 IP 路由表而做出转发。在这里，要让 PE 的上一跳路由器不要为数据打标签而直接改成 IP 数据包，这还得需要 PE 来告诉它才行。正常情况下，PE 路由器是告诉上一跳正常的标签，上一跳将这个标签变成远程标签，但现在，PE 路由器就不应该告诉上一跳正常的标签，它告诉的是隐式空（标签号为 3）标签，所以收到标签为 3 的 LSR，就不会在数据包发给下一跳时打上标签。这种终点使用隐式空标签来告诉上一跳不要打标签的行为叫倒数第二跳移除（PHP）行为，所以一台收到隐式空标签的 LSR，相应出口就不再是一个远程标签，而应该在 outgoing 显示为 pop。这种

上一跳标签移除称为标签弹出。

注：在 IOS 中，PHP 这种行为是默认的，但只会为直连路由和聚合路由通告隐式空 3，但 3 不会明写。

显示空标签

显示空的功能是在隐式空的基础上的，IPv4 标签号为 0，IPv6 为 2。

因为标签中 EXP 用于 QOS，前一跳移除后，这些信息也没了，可能希望保留，所以是上一跳将标签变为 0，来告诉终点不用为 0 查找 LFIB，只看 EXP，所以只关心 QOS 效果，这样也省事。

路由器报警标签 1，只是需要特别注意，并且软件转发。

未保留的都可以用，20 个 bit，是 16-1048575,IOS 默认是 16-10 0000,IGP 够了，但 BGP 可能不够，可以查看和修改标签范围。

MPLS TTL 行为

在正常情况下，当数据包的 TTL 值减到 0 时，路由器会向源发 ICMP 类型 11 和代码 0(时间超时)的数据包，来告诉源主机目标超时不可达。所以 TTL 无论对于 IP 网络还是 MPLS 网络都是非常重要的。

在数据包从 IP 网络进入 MPLS 网络时，IP 刚进来，以前的 TTL 是多少，PE 减 1 后，写到标签的 TTL 位，在出 MPLS 网络时，PE 再看标签中的 TTL 是多少，肯定比 IP 原来的 TTL 值小，减 1 后写回去，如果 TTL 值比 IP 原来的 TTL 值还大，就不正常，就不写了。

标签到标签，添加和交换等操作，也是减 1 后再复制，中间 P 路由器只修改顶部标签中的 TTL，顶部以下的标签是不动的。

如果遇见一个数据包 TTL 值为 0，普通的 MPLS 网络就是沿原来的 LSP 回去，只有是 IPv4 和 IPv6 才会，其它的丢弃。

MPLS VPN 中 TTL 没有后，是由终点 PE 或 CE 发回的，因为 P 没有源主机的路由，所以无法发送超时 ICMP。

MPLS MTU

OSI 第三层网络层协议的包头是在第二层帧头之上的，也就是说在封装二层帧头的时候，是将数据内容和三层包头全部作为数据封装在里面的，对于二层来说，之前的数据最大是多少，就是由 MTU 来决定的，所以正常情况下 MTU 就是第三层数据和包头的最大尺寸，这时无需分段就能传输，如果比 MTU 大，就得分段后传输。但是 MPLS 的标签是在二层帧头之后的，所以二层帧头将标签的大小和三层包的内容累加到一起作为数据封装的，因为三层包的所有内容正好和 MTU 一样大，在此基础上加上 MPLS 标签的话，就肯定比额定的 MTU 要大，所以这时 MPLS 的标签数据是会被分段后传输的，如果不想被分段，就得更新 MTU 的大小。（一般 MPLS 数据包加最多两个标签，一个标签 4 字节，所以只要改成比正常 MTU 多 8 字节即可。）而改 MTU 还必须在 MPLS 网络中所有设备上进行更改，除非允许分段。

MPLS 最大接收单元（MRU）

此内容无须配置

查看： sh mpls forwarding-table 1.1.1.1 detail

MPLS 默认超过 MTU 的数据包是和 IP 数据包一样要分段传输的，分段就是 LSR 移除标签，对 IP 数据分好相应大小后，再将原来的标签加到每个包，如果 IP 包头设置了不分段（DF），LSR 就丢掉报文，然后返回一个需要分段的 ICMP（不分段位设置为类型 3，代码 4），然后沿来的 LSP 发回去。

MTU 路径发现

（自动执行）

查看从源到目的的最小 MTU，就是类似窗口的机制，数据包试着发出去，如果被丢了，就减小后再发，如果再丢就再减再发，直到能正常发到目的地为止。有时这个不太好用，因为 ICMP 不能返回，可能是防火墙挡住了。

标签分发

要让 IGP 全部都支持标签分发并且相互协同工作，不现实，所以新分发协议要独立于所有路由协议，那么就使用 LDP，但是 BGP 可以广泛使用，范围大，最后 BGP 就自己发标签。LDP 不为 BGP 的 IPv4 路由发送标签。

LDP 运行

LDP 运行时有四大功能：

（1）运行 LDP 的 LSR 发现

（2）会话的建立和维护

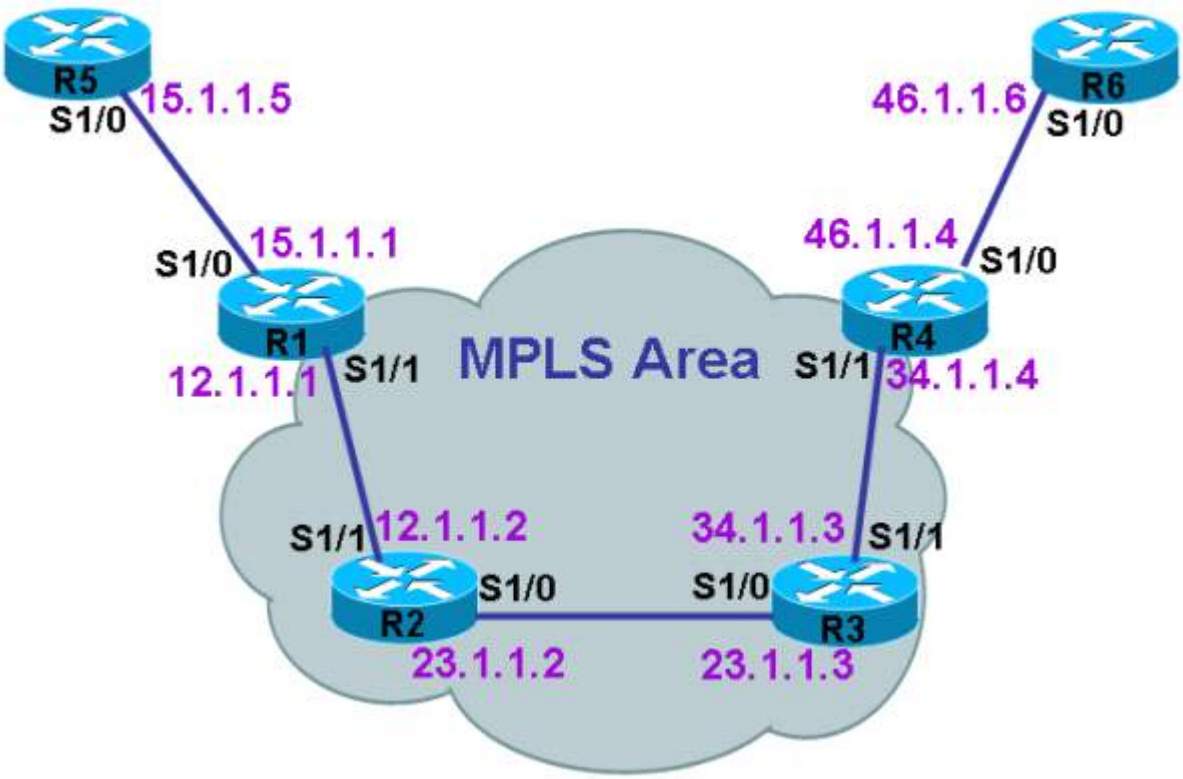
（3）标签映射通告

（4）使用通知来进行管理

LDP 是需要像 OSPF 那样建邻居的，使用 hello 包发现和维护邻居关系，LDP 会在启用了的接口上发送 hello 来找邻居，发送 hello 用 UDP 646，目的地为 224.0.0.2，hello 时间和保持时间分别是 5 秒 15 秒。这个 hello 包是不能跨网段传递的，而这个 hello 包被称为 LDP Link Hello。

LDP 在 LSR 之间除了建立邻居关系之外，还要建立 LDP 会话，建 LDP 会话就是用来交换标签的，使用的是 TCP 连接。而这个会话也只能和直连邻居建立，这样会话被称为 LDP sessions。LDP 会话的 hello 和超时分别是 60 秒和 180 秒。如果 LDP 邻居关系丢失，那么 LDP 会话也会断开。

配置 MPLS



说明：

以上图为拓朴，配置 MPLS，MPLS 网络区域的范围是 R1、R2、R3、R4，而 R1 和 R4 同时连接 MPLS 区域和 IP 网络，最终数据包在 MPLS 区域内传递时，我们将看到标签交换的效果。在开始配置之前，需要申明的是，每台路由器上都已配置 looback0，地址分别为 X.X.X.X/32,其中 X 表示设备号码，比如 R2 的 loopback0 地址为 2.2.2.2/32,R5 的 loopback0 地址为 5.5.5.5/32；在所有设备中，已经启用 OSPF 协议，除了每台设备的接口 loopback0 没有放进 OSPF 进程以外，其它所有接口均在 OSPF 进程里通告。

1.查看和修改标签范围（可选配置）

（1）看默认标签数量:

R1#sh mpls label range

Downstream Generic label region: Min/Max label: 16/100000

R1#

说明：默认标签范围是 16 到 100000

（2）改标签范围：

R1(config)#mpls label range 16 1010000

R1#sh mpls label range

Downstream Generic label region: Min/Max label: 16/1010000

R1#

说明:已将标签范围改成：16 到 1010000

2.查看和修改 MTU （可选配置）

（1）查看路由器接口 MTU:

R1#show mpls int s1/1 detail

（2）修改路由器接口 MTU:

R1(config)#int s1/1

R1 (config-if)#mpls mtu 1508

或

R1 (config)#int s1/1

R1 (config-if)#mtu 2000

注:某些 IOS 下的接口不能设置 MTU，只能分段传输。

（3）修改交换机支持小巨型帧：

sw（config）#system jumbomtu 2000

sw(config)#system mtu 2000

3.全局开启 CEF （必须配置）

注：某些 IOS 版本已默认开启的，可跳过此步，请以自身 IOS 为准。

r1(config)#ip cef

4.配置 LDP （必须配置）

（1） 全局启用 LDP:

说明：如果全局启用 LDP，就将在此路由器的所有接口都开启 LDP，但也可以选择只在某接口开启。

r1(config)#mpls label protocol ldp

（2） 接口启用 LDP:

说明：如果路由器并非全部接口都需要开启 LDP，则只在相应接口开启。

r1(config)#int s1/1

r1(config-if)# mpls label protocol ldp

（3）在接口下开启发 hello 包找邻居:

r1(config)#int s1/1

r1(config-if)#mpls ip

说明：接口上配置 mpls ip 就算打开了

IOS 有时还在使用 tag-switching 来代替 mpls ip,但功能是一样的，这两个命令相等。

注：请按上述配置 LDP 的方法，在 MPLS 区域内的所有路由器所有相关接口开启 LDP 并发出 hello 包，以方便 LDP 邻居的建立。

附：按以上拓朴，总结出需要的配置为：

R1：

r1(config)#int s1/1

r1(config-if)# mpls label protocol ldp

r1(config-if)#mpls ip

R2：

R2(config)#mpls label protocol ldp

R2(config)#int s1/0

R2(config-if)#mpls ip

R2(config-if)#exit

R2(config)#int s1/1

R2(config-if)#mpls ip

R3：

R3(config)#mpls label protocol ldp

R3(config)#int s1/0

R3(config-if)#mpls ip

R3(config-if)#exit

R3(config)#int s1/1

R3(config-if)#mpls ip

R4：

R4(config)#int s1/1

R4(config-if)#mpls label protocol ldp

R4(config-if)#mpls ip

5.查看 LDP 简单信息

（1）可以查看哪些接口开启了 mpls：

r1#sh mpls interfaces

Interface	IP	Tunnel	Operational
Serial1/1	Yes (ldp)	No	Yes

r1#

说明：可以看出，R1 相关接口 S1/1 已经运行在 LDP 下。（其它设备接口状态略过！）

（2）查看看 LDP 详情，包括包含 hello 时间，会话时间：

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 180 sec; keep alive interval: 60 sec

Discovery hello: holdtime: 15 sec; interval: 5 sec

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

r1#

说明：可以看到，默认 hello 和 hold 分别是 5s 和 15s，会话时间 hello 和 hold 分别是 60s 和 180s。

（3）修改时间机制（并不建议修改）：

注:两边保持时间不一样，选用小的一端，改了多个，也是用小的而不是最新的。

改 hello:

r1(config)#mpls ldp discovery hello interval 3

r1(config)#exi

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 180 sec; keep alive interval: 60 sec

Discovery hello: holdtime: 15 sec; interval: 3 sec

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

说明:可以看到 hello 时间被改成了 3s

r1#sh mpls ldp discovery detail

Local LDP Identifier:

12.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 3000 ms; Transport IP addr: 12.1.1.1

LDP Id: 2.2.2.2:0; no host route to transport addr

Src IP addr: 12.1.1.2; Transport IP addr: 12.1.1.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r1#

说明:也可以看到 hello 时间被改成了 3s

再改：

r1(config)#mpls ldp discovery hello interval 8

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 180 sec; keep alive interval: 60 sec

Discovery hello: holdtime: 15 sec; interval: 8 sec

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

r1#

说明:可以看到 hello 时间被改成了 8s

r1#sh mpls ldp discovery detail

Local LDP Identifier:

12.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 12.1.1.1

LDP Id: 2.2.2.2:0; no host route to transport addr

Src IP addr: 12.1.1.2; Transport IP addr: 12.1.1.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r1#

说明:在这可以看到 hello 时间还是选默认 5s，因为这个小。

改会话时间（不建议）：

r1(config)#mpls ldp holdtime 150

r1#sh adjacency detail

*Mar 1 01:32:03.063: %SYS-5-CONFIG_I: Configured from console by console

r1#sh mpls ldp parameters

Protocol version: 1

Downstream label generic region: min label: 16; max label: 100000

Session hold time: 150 sec; keep alive interval: 50 sec

Discovery hello: holdtime: 15 sec; interval: 8 sec

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

Discovery targeted hello: holdtime: 90 sec; interval: 10 sec

Downstream on Demand max hop count: 255

Downstream on Demand Path Vector Limit: 255

LDP for targeted sessions

LDP initial/maximum backoff: 15/120 sec

LDP loop detection: off

r1#

说明可以看到 hold 时间被改成了 150s。

6.查看 LDP 邻居相关信息

(1)在 R1 上查看 LDP discovery 情况:

r1#sh mpls ldp discovery detail

Local LDP Identifier:

1.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0; no route to transport addr

Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

r1#

说明: Local LDP Identifier 是每台 LSR 都必须有的，这个 ID 用 6 个字节表示，前 4 个节字称为 Rotuer-ID，先选 loopback 地址最大的，然后是物理接口，它的选举方法和 OSPF Rotuer-ID 相同，后面 2 个字节是表示标签空间的，也就是标签是基于设备还是基于接口，如果是基于设备，就是 0，可以从上面看出，1.1.1.1:0 中，1.1.1.1 表示 R1 的 Rotuer-ID，而 0 表示标签是基于设备的。再看后面还有个 Transport IP ，而这个 IP 默认是选用 Rotuer-ID 的地址，这个地址在建邻居时非常重要，是会话的源地址，如果这个地址对方没有路由可达，那么就不可能建起邻居。所以一定要保证双方 Transport IP 是路由相通的。从上面结果中还可以看出，R1 已经收到了对方 R2 的 hello，对方 Transport IP 是 2.2.2.2，也就是对方的 loopback0 地址，而因为这个地址不在 OSPF 进程里，所以 R1 不能到达，也就不能建邻居，后面提示为“no route to transport addr”。

（2）在 R2 上查看 LDP discovery 情况:

r2#sh mpls ldp discovery detail

Local LDP Identifier:

2.2.2.2:0

Discovery Sources:

Interfaces:

Serial1/0 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 2.2.2.2

LDP Id: 3.3.3.3:0; no route to transport addr

Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 2.2.2.2

LDP Id: 1.1.1.1:0; no route to transport addr

Src IP addr: 12.1.1.1; Transport IP addr: 1.1.1.1

Hold time: 15 sec; Proposed local/peer: 15/15 sec

r2#

说明:可以看出，R2 的 Rotuer-ID 是 2.2.2.2，这个地址也就是 Transport IP，而 R1 的 Transport IP 是 1.1.1.1，从上面也看出这两个地址是路由上互不相通的，所以不可能建立 LDP 邻居。

(3)解决邻居建立问题：

说明:要解决邻居建立问题，就要让双方的 Transport IP 能够相通，而 Transport IP 就是选用 Rotuer-ID 的 IP 地址，可以修改 Rotuer-ID 为可路由的接口，即可解决 Transport IP 互通。

改 R1Router-Id 为 S1/1 接口地址

r1(config)#mpls ldp router-id serial 1/1 force　　force 说明立即生效

r1(config)#exi

查看结果：

r1#sh mpls ldp discovery detail

Local LDP Identifier:

12.1.1.1:0

```
    Discovery Sources:
    Interfaces:
      Serial1/1 (ldp): xmit/recv
        Enabled: Interface config
        Hello interval: 5000 ms; Transport IP addr: 12.1.1.1
        LDP Id: 2.2.2.2:0; no route to transport addr
         Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2
         Hold time: 15 sec; Proposed local/peer: 15/15 sec
r1#
```

说明:可以看到 Rotuer-ID 已经改成接口 S1/1 的地址 12.1.1.1，Transport IP 也随即变成了 12.1.1.1。

R1 的 Transport IP 对 R2 来说已经可达了，可是 R2 的 Transport IP 还是自己的 loopback 口地址，R1 不能到达，所以还是不能建邻居。Transport IP 默认选用 Rotuer-ID 的地址，但我们可以明确指定 Transport IP 为某个接口的地址，这次我们直接改 R2 的 Transport IP 地址为 S1/1 的接口地址，Rotuer-ID 不变：

先看没改之前：

```
r2#sh mpls ldp discovery detail
 Local LDP Identifier:
    2.2.2.2:0
    Discovery Sources:
    Interfaces:
      Serial1/0 (ldp): xmit/recv
```

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 2.2.2.2

LDP Id: 3.3.3.3:0; no route to transport addr

Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 2.2.2.2

LDP Id: 12.1.1.1:0; no host route to transport addr

Src IP addr: 12.1.1.1; Transport IP addr: 12.1.1.1

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r2#conf t

现在修改：

r2(config)#int s1/1

r2(config-if)#mpls ldp discovery transport-address interface

r2(config-if)#exi

再看：

r2#sh mpls ldp discovery detail

Local LDP Identifier:

2.2.2.2:0

Discovery Sources:

Interfaces:

Serial1/0 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 2.2.2.2

LDP Id: 3.3.3.3:0; no route to transport addr

Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 12.1.1.2

LDP Id: 12.1.1.1:0; no host route to transport addr

Src IP addr: 12.1.1.1; Transport IP addr: 12.1.1.1

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r2#

说明:可以看到，s1/1 上 Transport IP 已经不再是 Router-ID 的地址，已经被改成本接口地址了。但是接口 S1/0 还是使用原来 Router-ID 的地址。

但是邻居还是不会有，因为 R2 上直接改接口为 Transport IP，是要重启进程才能生效的：

r2#cle mpls ldp neighbor *

r2#

再查看邻居：

r2#sh mpls ldp neighbor

Peer LDP Ident: 12.1.1.1:0; Local LDP Ident 2.2.2.2:0

TCP connection: 12.1.1.1.646 - 12.1.1.2.11155

State: Oper; Msgs sent/rcvd: 9/9; Downstream

Up time: 00:00:43

LDP discovery sources:

Serial1/1, Src IP addr: 12.1.1.1

Addresses bound to peer LDP Ident:

15.1.1.1 12.1.1.1 1.1.1.1

r2#

说明:可以看到，邻居已经有了，并且可以看出端口号是 646。

（4）再来关心 R2 和 R3 的邻居：

说明：因为 R2 现在连 R3 的接口 S1/0 的 Transport IP 还是使用 Router-ID 的地址 2.2.2.2，而 R3 也到不了 2.2.2.2，所以 R2 和 R3 之间的 LDP 邻居关系是建不起来的，我们还是像 R2 和 R1 建邻居那样，把 S1/0 接口的 Transport IP 改成使用本接口的地址。

r2(config)#int s1/0

r2(config-if)#mpls ldp discovery transport-address interface

r2(config-if)#exi

而 R3 的 Transport IP 还是使用自己的 Router-ID 地址 3.3.3.3，这个地址 R2 也是无法到达的，在 R3 上也可以通过将该地址放进 OSPF 进程来使 R2 能够 ping 通，从而建立 LDP 邻居。

r3(config)#router ospf 2

r3(config-router)#network 3.3.3.3 0.0.0.0 area 0

r3(config-router)#exi

在 R2 上查看建 LDP 邻居的源地址：

r2#sh mpls ldp discovery detail

Local LDP Identifier:

2.2.2.2:0

Discovery Sources:

Interfaces:

Serial1/0 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 23.1.1.2

LDP Id: 3.3.3.3:0

Src IP addr: 23.1.1.3; Transport IP addr: 3.3.3.3

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 3.3.3.3/32

Serial1/1 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 12.1.1.2

LDP Id: 12.1.1.1:0; no host route to transport addr

Src IP addr: 12.1.1.1; Transport IP addr: 12.1.1.1

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 12.1.1.0/24

r2#

说明:可以看到，R2 连 R3 的接口 S1/0 的 Transport IP 已经成功改成了 23.1.1.2。

但是 R2 还是不会有 R3 的邻居，所以按以前方法重置 LDP 进程，再看就会有了：

来查看重置后的邻居状态：

```
r2#sh mpls ldp neighbor
    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 3.3.3.3.646 - 23.1.1.2.61206
        State: Oper; Msgs sent/rcvd: 10/9; Downstream
        Up time: 00:00:09
        LDP discovery sources:
          Serial1/0, Src IP addr: 23.1.1.3
        Addresses bound to peer LDP Ident:
          23.1.1.3        34.1.1.3        3.3.3.3
    Peer LDP Ident: 12.1.1.1:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 12.1.1.1.646 - 12.1.1.2.44954
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:00:03
        LDP discovery sources:
          Serial1/1, Src IP addr: 12.1.1.1
        Addresses bound to peer LDP Ident:
```

15.1.1.1 12.1.1.1 1.1.1.1

r2#

说明:在 R2 上可以看见和 R3 的 LDP 邻居关系已经建立。

（5）R3 跟 R4 的邻居关系：

说明：R3 的 Router-ID 已经通告进 OSPF 进程，所以 R4 也能到达了，那么 R4 也选择将 Loopback 地址放 loop 进 OSPF 进程来完成和 R3 的 LDP 邻居关系建立。

说明:最终保证所有 LDP 邻居建立（R1 和 R2 的邻居，R2 和 R3 的邻居，R3 和 R4 的邻居全部都有）。

7.查看标签交换相关信息

说明:先以 R4 的 loopback0 地址 4.4.4.4/32 这条路由为例，来看别的路由器对这条路由的标签状况。

（1）在 R4 上查看 LFIB，看路由 4.4.4.4 的情况：

r4#sh mpls forwarding-table

Local Outgoing Prefix Bytes tag Outgoing Next Hop

tag tag or VC or Tunnel Id switched interface

16 Pop tag 23.1.1.0/24 0 Se1/1 34.1.1.3

17 16 12.1.1.0/24 0 Se1/1 34.1.1.3

18 18 15.1.1.0/24 0 Se1/1 34.1.1.3

19 Pop tag 3.3.3.3/32 0 Se1/1 34.1.1.3

r4#

说明:可以看出，R4 上对自己直连接口的标签并没有出现在 LFIB 表中，因此该路由不需要进行标签交换，属正常。

（2） 查看 R4 上 CEF 对 4.4.4.4 的处理情况：

注：所有路由的处理，即使是打标签，都要由 CEF 来处理。

```
r4#sh ip cef 4.4.4.4

4.4.4.4/32, version 15, epoch 0, connected, receive

  tag information set

    local tag: implicit-null

r4#
```

说明:可以看出，R4 的 CEF 对 4.4.4.4 这条路由打的本地标签是 implicit-null(隐式空标签，对于隐式空标签的解释，请参见 MPLS 正文内容)，本地路由发给邻居之后，就成为了邻居的远程标签，所以邻居到达 4.4.4.4 的路由标签都应该是隐式空标签。

（3）查看 R3 的 CEF 对 4.4.4.4 的处理情况：

```
r3#sh ip cef 4.4.4.4

4.4.4.4/32, version 19, epoch 0, cached adjacency 34.1.1.4

0 packets, 0 bytes

  tag information set

    local tag: 19

  via 34.1.1.4, Serial1/1, 0 dependencies

    next hop 34.1.1.4, Serial1/1

    valid cached adjacency

    tag rewrite with Se1/1, 34.1.1.4, tags imposed: {}

r3#
```

说明:很明显，4.4.4.4 在 R4 上的本地标签（implicit-null）发给 R3 之后，就成为了 R3 的远程标签，可以看到，因为 R4 发来时是隐式空，所以 R3 就不能为 4.4.4.4

打任何标签，所以最终结果的空的。而 R3 对于 4.4.4.4 这条路由是要生成自己的本地标签的，因为自己要对这条路由使用标签交换，可以看到 R3 自己给 4.4.4.4 打的本地标签是 19，那么这个标签发给别的邻居之后，就该变成远程标签 19。

（4）再查看 R3 的 LFIB 表：

r3#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	12.1.1.0/24	0	Se1/0	23.1.1.2
17	Pop tag	46.1.1.0/24	0	Se1/1	34.1.1.4
18	18	15.1.1.0/24	0	Se1/0	23.1.1.2
19	Pop tag	4.4.4.4/32	0	Se1/1	34.1.1.4

r3#

说明:可以看到 R3 对于路由条目 4.4.4.4 的本地标签是 19，这是要发给别人的，而出的标签是 Pop tag，也就是要移除标签，也就是当 R3 收到邻居发来一个数据包，如果查看结果为顶部标签是 19，那么自己就将该标签移除后，再从 S1/0 发出去。

（5） 查看 R3 的 FIB 表：

注：LSR 对于路由条目所有的标签都是保存在 FIB 表里的，用到的才放进 LFIB，所以在 LFIB 表里将可能看到路由的多个标签，但是只有一个标签是正被使用的。

```
r3#sh mpls ip binding
  2.2.2.2/32
      out label:     imp-null   lsr: 2.2.2.2:0
  3.3.3.3/32
      in label:      imp-null
      out label:     19         lsr: 2.2.2.2:0
```

out label: 19 lsr: 4.4.4.4:0

4.4.4.4/32

in label: 19

out label: imp-null lsr: 4.4.4.4:0 inuse

out label: 20 lsr: 2.2.2.2:0

12.1.1.0/24

in label: 16

out label: imp-null lsr: 2.2.2.2:0 inuse

out label: 17 lsr: 4.4.4.4:0

15.1.1.0/24

in label: 18

out label: 18 lsr: 2.2.2.2:0 inuse

out label: 18 lsr: 4.4.4.4:0

23.1.1.0/24

in label: imp-null

out label: imp-null lsr: 2.2.2.2:0

out label: 16 lsr: 4.4.4.4:0

34.1.1.0/24

in label: imp-null

out label: 16 lsr: 2.2.2.2:0

out label: imp-null lsr: 4.4.4.4:0

46.1.1.0/24

in label: 17

out label: 17 lsr: 2.2.2.2:0

out label: imp-null lsr: 4.4.4.4:0 inuse

r3#

说明:从以上结果看出，路由条目 4.4.4.4 的 in 标签，即本地标签是 19,而出的标签有两个，分别是 20 和 imp-null ，而 imp-null 后面有关键字 inuse，也就表示 imp-null 是正在使用中的。

（6） 查看 R2 的 LFIB：

r2#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	34.1.1.0/24	0	Se1/0	23.1.1.3
17	17	46.1.1.0/24	0	Se1/0	23.1.1.3
18	Pop tag	15.1.1.0/24	0	Se1/1	12.1.1.1
19	Pop tag	3.3.3.3/32	0	Se1/0	23.1.1.3
20	19	4.4.4.4/32	0	Se1/0	23.1.1.3

r2#

说明:正因为 R3 收到标签为 19 的数据包，才会移除标签后从 S1/1 发给 R4，所以 R2 肯定是应该将 R3 发来的本地标签 19 变成自己的远程标签 19,以上结果也显示，R2 对 4.4.4.4 打的出标签正是 19，而本地标签是 20，也就是说当 R2 收到一个顶部标签为 20 的数据包，就将标签换成 19 后从 S1/0 发出去，发出去正好就是发给了 R3，而 R3 根据自己的处理最终发到 R4。R2 给 4.4.4.4 打的本地标签 20，发给谁也就会变成谁的远程标签 20。

（7）查看 R2 的 CEF 对 4.4.4.4 的处理情况：

r2#sh ip cef 4.4.4.4

4.4.4.4/32, version 25, epoch 0, cached adjacency 23.1.1.3

0 packets, 0 bytes

tag information set

local tag: 20

fast tag rewrite with Se1/0, 23.1.1.3, tags imposed: {19}

via 23.1.1.3, Serial1/0, 0 dependencies

next hop 23.1.1.3, Serial1/0

valid cached adjacency

tag rewrite with Se1/0, 23.1.1.3, tags imposed: {19}

r2#

说明:可以看出，R2 将 4.4.4.4 的本地标签改成 20，将出的标签改成 19（19 正是 R3 上的本地标签）。所以和上面的理论全部对应。

（8）查看 R1 的 LFIB 表：

r1#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	16	34.1.1.0/24	0	Se1/1	12.1.1.2
17	Pop tag	23.1.1.0/24	0	Se1/1	12.1.1.2
18	17	46.1.1.0/24	0	Se1/1	12.1.1.2
19	19	3.3.3.3/32	0	Se1/1	12.1.1.2
20	20	4.4.4.4/32	0	Se1/1	12.1.1.2

r1#

说明:从上面结果中看出，正因为 R2 对 4.4.4.4 打的本地标签是 20，所以发给 R1，就变成 R1 的远程标签，即出口标签是 20 了，而 R1 给 4.4.4.4 打的本地标签也是 20，这是自己随意打上去的，从这个结果表明，当 R1 收到一个数据包标签为 20 的，那么就改成标签 20 从 S1/1 发出动。

（9）在 R1 上跟踪路由 4.4.4.4 的传输情况：

r1#traceroute 4.4.4.4

Type escape sequence to abort.

Tracing the route to 4.4.4.4

1 12.1.1.2 [MPLS: Label 20 Exp 0] 156 msec 220 msec 156 msec

2 23.1.1.3 [MPLS: Label 19 Exp 0] 168 msec 188 msec 196 msec

3 34.1.1.4 160 msec * 140 msec

r1#

说明:因为对于条目 4.4.4.4，R4 给 R3 的标签为隐式空，即不打标签，R3 给 R2 的标签为 19,所以 R2 会打上标签 19 后发给 R3，而 R2 给 R1 的标签是 20，所以 R1 应该打上标签 20 后发给 R2，那么在 traceroute 时，就看到了，给 12.1.1.2(R2)打上的标签是 20,然后发给 23.1.1.3（R3）时打的标签是 19,最后 R3 发给 34.1.1.4（R4）时是没有打标签就发出去了。

（10）到 IP 网络 R5 上去 traceroute 4.4.4.4：

r5#traceroute 4.4.4.4

Type escape sequence to abort.

Tracing the route to 4.4.4.4

1 15.1.1.1 136 msec 72 msec 84 msec

2 12.1.1.2 300 msec 368 msec 192 msec

3 23.1.1.3 252 msec 296 msec 148 msec

4 34.1.1.4 216 msec * 188 msec

r5#

说明:从结果中发现，IP 网络中发的数据在穿越 MPLS 区域时，标签的交换过程是看不见的，要想看到标签的交换过程，只有在 MPLS 网络中才能看见。

8.查看标签交换过程

说明:再以 6.6.6.6 这条路由条目为例，看一下它的标签如何。在边缘路由器 R4 上，会将 IP 网络的路由条目以隐式空标签发给邻居，但是默认只有自己直连的 IP 网络才会发隐式空标签，而 6.6.6.6 是在 R6 上的网段，不是 R4 的直连网段，所以对于 6.6.6.6，R4 不应该隐式空标签给邻居。

（1）查看 R4 上对 6.6.6.6 的标签情况：

r4#sh mpls ip binding

3.3.3.3/32

in label: 19

out label: imp-null lsr: 3.3.3.3:0 inuse

4.4.4.4/32

in label: imp-null

out label: 19 lsr: 3.3.3.3:0

6.6.6.6/32

in label: 20

out label: 20 lsr: 3.3.3.3:0

12.1.1.0/24

```
        in label:    17
        out label:    16       lsr: 3.3.3.3:0       inuse
    15.1.1.0/24
        in label:    18
        out label:    18       lsr: 3.3.3.3:0       inuse
    23.1.1.0/24
        in label:    16
        out label:    imp-null  lsr: 3.3.3.3:0       inuse
    34.1.1.0/24
        in label:    imp-null
        out label:    imp-null  lsr: 3.3.3.3:0
    46.1.1.0/24
        in label:    imp-null
        out label:    17       lsr: 3.3.3.3:0
r4#
```

说明:从结果中看出，因为 4.4.4.4 是 R4 的直连接口，所以本地标签（in label）是空的，而 6.6.6.6 不是自己直连网络，所以不应该打隐式空标签，可以看到它的本地标签（in label）是 20。

（2）查看 R4CEF 里面的 6.6.6.6：

```
r4#sh ip cef 6.6.6.6
6.6.6.6/32, version 20, epoch 0, cached adjacency 46.1.1.6
0 packets, 0 bytes
```

```
  tag information set
    local tag: 20
  via 46.1.1.6, Serial1/0, 0 dependencies
    next hop 46.1.1.6, Serial1/0
    valid cached adjacency
    tag rewrite with Se1/0, 46.1.1.6, tags imposed: {}
r4#
```

说明:也可以看出，6.6.6.6 的处理不同于 4.4.4.4，6.6.6.6 的本地标签确实真实存在，是 20.

（3）看到 R3 给 6.6.6.6 打的标签:

```
r3#sh ip cef 6.6.6.6
6.6.6.6/32, version 20, epoch 0, cached adjacency 34.1.1.4
0 packets, 0 bytes
  tag information set
    local tag: 20
    fast tag rewrite with Se1/1, 34.1.1.4, tags imposed: {20}
  via 34.1.1.4, Serial1/1, 0 dependencies
    next hop 34.1.1.4, Serial1/1
    valid cached adjacency
    tag rewrite with Se1/1, 34.1.1.4, tags imposed: {20}
r3#
```

说明:R3 给 6.6.6.6 打本地标签是正常的,打了远程标签(出标签)是 20,也在情理之中,因为不需要弹出标签(移除标签)。

（4）再看 R3 的 LFIB 表：

r3#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	12.1.1.0/24	1672	Se1/0	23.1.1.2
17	Pop tag	46.1.1.0/24	0	Se1/1	34.1.1.4
18	18	15.1.1.0/24	540	Se1/0	23.1.1.2
19	Pop tag	4.4.4.4/32	1132	Se1/1	34.1.1.4
20	20	6.6.6.6/32	540	Se1/1	34.1.1.4

r3#

说明:R3 对 6.6.6.6 出标签是 20，说明处理也是正常的

（5）R2 上看 6.6.6.6 的处理：

r2#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Pop tag	34.1.1.0/24	0	Se1/0	23.1.1.3
17	17	46.1.1.0/24	0	Se1/0	23.1.1.3
18	Pop tag	15.1.1.0/24	520	Se1/1	12.1.1.1

19 Pop tag 3.3.3.3/32 0 Se1/0 23.1.1.3

20 19 4.4.4.4/32 756 Se1/0 23.1.1.3

21 20 6.6.6.6/32 540 Se1/0 23.1.1.3

r2#

说明:R3 的本地标签 20，变成 R2 的出标签 20，正常，而 R2 给 6.6.6.6 打的本地标签 21，发给邻居将成为远程标签。

（6）再看 R1 对 6.6.6.6 的处理：

r1#sh mpls forwarding-table

Local Outgoing Prefix Bytes tag Outgoing Next Hop

tag tag or VC or Tunnel Id switched interface

16 16 34.1.1.0/24 0 Se1/1 12.1.1.2

17 Pop tag 23.1.1.0/24 0 Se1/1 12.1.1.2

18 17 46.1.1.0/24 0 Se1/1 12.1.1.2

19 19 3.3.3.3/32 0 Se1/1 12.1.1.2

20 20 4.4.4.4/32 0 Se1/1 12.1.1.2

21 21 6.6.6.6/32 0 Se1/1 12.1.1.2

r1#

说明:R1 将 R2 发来的标签 21 变成自己的出标签（远程标签）21，正常，自己的本地标签也正常。

（7）在 R1 上跟踪路由的标签交换过程：

r1#traceroute 6.6.6.6

Type escape sequence to abort.

Tracing the route to 6.6.6.6

1 12.1.1.2 [MPLS: Label 21 Exp 0] 356 msec 284 msec 204 msec

2 23.1.1.3 [MPLS: Label 20 Exp 0] 388 msec 196 msec 192 msec

3 34.1.1.4 [MPLS: Label 20 Exp 0] 152 msec 268 msec 260 msec

4 46.1.1.6 244 msec * 172 msec

r1#

说明:因为 R4 对于路由条目 6.6.6.6 没有采用隐式空标签，而给 R3 发了标签 20，而 R3 给 R2 也发了标签 20，R2 给 R1 发了标签 21，所以结果中显示数据包发由 R1 发给 R2（12.1.1.2）时打的标签正是 21，R2 发给 R3 时，打的标签是 20，重要的是 R3 在发给 R4 时，并没有移除标签，而是打了标签 20 发出去的。

（8）在 IP 网络 R5 上看 6.6.6.6 的标签交换情况：

r5#traceroute 6.6.6.6

Type escape sequence to abort.

Tracing the route to 6.6.6.6

1 15.1.1.1 180 msec 92 msec 60 msec

2 12.1.1.2 308 msec 256 msec 264 msec

3 23.1.1.3 368 msec 200 msec 200 msec

4 34.1.1.4 232 msec 252 msec 144 msec

5 46.1.1.6 248 msec * 408 msec

r5#

说明：IP 网络的 R5 没有看到标签交换情况，所以正常。

9.查看数据包交换数量

说明:可以在 LSR 上查看某条路由经过标签交换的数据包数量：

（1）在 R3 上查看 4.4.4.4 有多少个数据包经过了标签交换：

r3#sh mpls forwarding-table 4.4.4.4 detail

Local Outgoing Prefix Bytes tag Outgoing Next Hop

tag tag or VC or Tunnel Id switched interface

19 Pop tag 4.4.4.4/32 1744 Se1/1 34.1.1.4

MAC/Encaps=4/4, MRU=1504, Tag Stack{}

4CA18847

No output feature configured

Per-packet load-sharing

r3

（2）在 R3 上查看 6.6.6.6 有多少个数据包经过了标签交换：

r3#sh mpls forwarding-table 6.6.6.6 detail

Local Outgoing Prefix Bytes tag Outgoing Next Hop

tag tag or VC or Tunnel Id switched interface

20 20 6.6.6.6/32 2028 Se1/1 34.1.1.4

MAC/Encaps=4/8, MRU=1500, Tag Stack{20}

4CA18847 00014000

No output feature configured

Per-packet load-sharing

10.路由条目的标签限制

说明:在某些时候，并不希望 LSR 对相应的条目打上标签，那么就可以在 LSR 限制相应路由条目的标签通告或接收。

（1） 限制标签接收：

说明:在 R1 上测试限制接收 6.6.6.6 的标签：

r1(config)#access-list 6 permit 6.6.6.6

r1(config)#mpls ldp neighbor 12.1.1.2 labels accept 6

说明:命令意思为只从邻居 12.1.1.2 那里接收 ACL 6 允许的路由条目的标签，其它的不接收。

（2）查看配置标签限制后的 LFIB：

r1#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	34.1.1.0/24	0	Se1/1	12.1.1.2
17	Untagged	23.1.1.0/24	0	Se1/1	12.1.1.2
18	Untagged	46.1.1.0/24	0	Se1/1	12.1.1.2
19	Untagged	3.3.3.3/32	0	Se1/1	12.1.1.2

20 Untagged 4.4.4.4/32 0 Se1/1 12.1.1.2

21 21 6.6.6.6/32 0 Se1/1 12.1.1.2

r1#

说明:可以看到，除了 6.6.6.6 打上了标签 21 以外，其它路由条目全部都没有标签，说明配置正确。

（3）限制标签发送：

说明:可以在 LSR 限制从邻居处接收标签，同样也可以限制将什么样的路由发送标签给邻居。

下面测试在 R3 上限制只发送 6.6.6.6 的标签给 R2，其它统统不发标签：

先看 R2 上在没有做限制时的标签情况：

r2#sh mpls forwarding-table

Local Outgoing Prefix Bytes tag Outgoing Next Hop

tag tag or VC or Tunnel Id switched interface

16 Pop tag 34.1.1.0/24 0 Se1/0 23.1.1.3

17 17 46.1.1.0/24 0 Se1/0 23.1.1.3

18 Pop tag 15.1.1.0/24 3340 Se1/1 12.1.1.1

19 Pop tag 3.3.3.3/32 0 Se1/0 23.1.1.3

20 19 4.4.4.4/32 972 Se1/0 23.1.1.3

21 20 6.6.6.6/32 1188 Se1/0 23.1.1.3

说明:可以看到 6.6.6.6 和其它路由都有标签。

（4）在 R3 上配置只发 6.6.6.6 的标签给 R2：

r3(config)#access-list 2 permit 2.2.2.2 此地址必须为对方的 Router-ID

r3(config)#access-list 6 permit 6.6.6.6 匹配路由条目

s ldp advertise-labels 此命令必须配，用以禁止正常标签传送，否则所有限制无效

r3(config)#mpls ldp advertise-labels for 6 to 2 对相应邻居做相应限制

r3(config)#exi

（5）再看 R2 上 6.6.6.6 的标签情况:

r2#sh mpls forwarding-table

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	34.1.1.0/24	0	Se1/0	23.1.1.3
17	Untagged	46.1.1.0/24	0	Se1/0	23.1.1.3
18	Untagged	15.1.1.0/24	3340	Se1/1	12.1.1.1
19	Untagged	3.3.3.3/32	0	Se1/0	23.1.1.3
20	Untagged	4.4.4.4/32	972	Se1/0	23.1.1.3
21	20	6.6.6.6/32	1188	Se1/0	23.1.1.3

说明:可以看到，除了 6.6.6.6 有标签，其它都没有标签，说明配置生效。

（6）在 R3 上也可以看配置的效果：

r3#sh mpls ldp bindings advertisement-acls

Advertisement spec:

Prefix acl = 6; Peer acl = 2

```
  tib entry: 2.2.2.2/32, rev 13
  tib entry: 3.3.3.3/32, rev 4
  tib entry: 4.4.4.4/32, rev 15
  tib entry: 6.6.6.6/32, rev 18
      Advert acl(s): Prefix acl 6; Peer acl 2
  tib entry: 12.1.1.0/24, rev 8
  tib entry: 15.1.1.0/24, rev 12
  tib entry: 23.1.1.0/24, rev 6
  tib entry: 34.1.1.0/24, rev 2
  tib entry: 46.1.1.0/24, rev 10
r3#
```

LDP 邻居认证

说明:邻居之间可以配置相应密码，如果密码不同，则邻居无法建立。

（1）配置 R2 对 R1 使用密码 cisco，如果 R1 无密码，则邻居失败:

```
r2(config)#mpls ldp neighbor 12.1.1.1 password 0 cisco
r2(config)#
```

说明:指定邻居时，后面应该为邻居的 Router-ID,0 表示在内存中显示时不加密。其它邻居失败的效果略过，请自行配置查看。

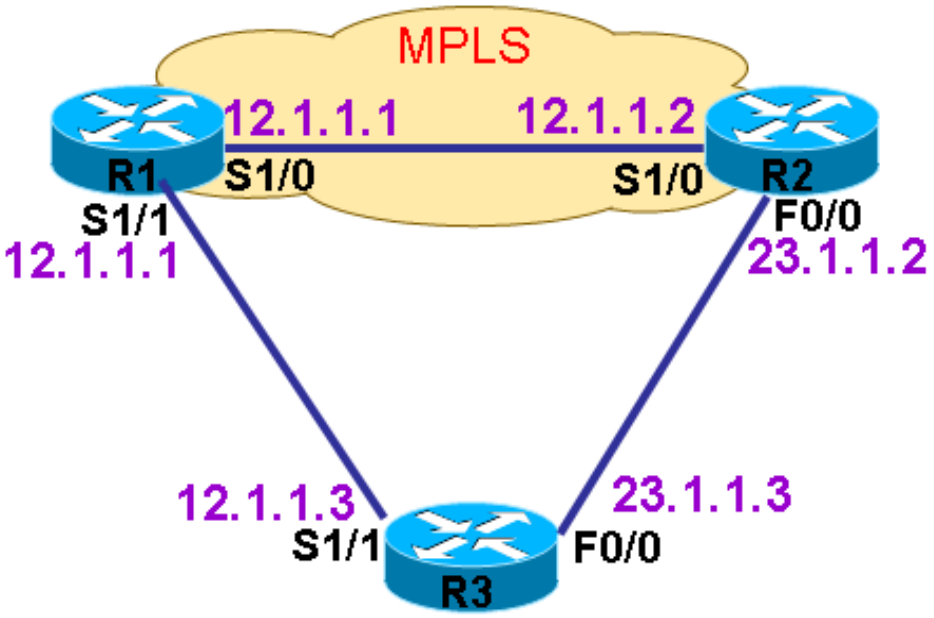
LDP 会话保护

在讲 LDP 时说过，LDP 在建邻居时，需要用到 hello 包，在直连接口上发送 hello 包出去，这个 hello 包是不能跨网段传递的，而这个 hello 包被称为 LDP Link Hello，如果对方有邻居回应了这个包，那么就建立 LDP 会话，称为 LDP sessions，LDP 会话建立后就可以传递标签，这是直连邻居。但是邻居也可以远程建立，也就是说不直连，那么这样的 hello 称为 LDP Targeted Hello，而远程建立的会话就叫 targeted session。一般情况下，两台直连 LSR 建立会话之后，如果链路断掉了，那么会话也就断掉了，所以所有标签等到会话再次建立后再次重新计算。但是当两台 LSR 之间链路断掉之后，如果他们之间还有备用链路的话，完全可以事先在备用链路上建个远程会话 targeted session，这样的话，即使两台 LSR 之间直连链路断了，也可以因为还有远程会话而不用清空所有标签，等到直连链路恢复后，再切换回来。这就是开启会话保护之后，通过建立远程会话来保护标签表的作用，但要做这样的保护，LSR 之间必须得有备用链路，否则无效。

配置保护时，指定在多少时间内，会话不要断开，流量继续发送，也可以为软重置提供保护。配置时应该双方路由器都配置会话保护，或者一方配了，另一方至少要能回应 Targeted Hello。

上面就是利用会话保护的功能来为邻居之间在备用链路上创建远程会话（targeted session），远程会话可以保护邻居断掉之间不会马上断开，因为存在备用链路。这样的远程会话可以通过会话保护的功能来创建，除此之外，还有一种创建方法，那就是手工创建远程会话，下面分别来介绍这两种方法的配置。

配置



说明:R1 的 loopback0 为 1.1.1.1/32，R2 的 loopback0 为 2.2.2.2/32，R3 的 loopback0 为 3.3.3.3/32，OSPF 在所有设备上开启，并且所有 loopback 均放入 OSPF 进程。LDP 只在 R1 和 R2 之间的 S1/0 开启。从图中可以发现，当 R1 和 R2 之间的直连链路断掉以后，LDP 会话也会断开的，但是 R1 和 R2 明明还可以通过 R3 建立远程会话连接。

1.配置会话保护

（1）在 R1 上开启会话保护：

R1(config)# mpls ldp session protection

（2）在 R2 上开启会话保护：

R2(config)# mpls ldp session protection

（3）也可以通过 ACL 指定邻居和时间，如：

router(config)#access 1 per 1.1.1.1

router(config)#mpls ldp session protection for 1 duration 90s

2.查看会话保护效果

（1）先看一下保护之前的邻居状态：

r1#sh mpls ld neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.23261 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 9/9; Downstream

Up time: 00:00:04

LDP discovery sources:

Serial1/0, Src IP addr: 12.1.1.2

Addresses bound to peer LDP Ident:

23.1.1.2 12.1.1.2 2.2.2.2

说明:可以看出没有任何远程会话的信息。

（2）查看开了会话保护的邻居状态：

r1#sh mpls ld neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.39052 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 11/11; Downstream

Up time: 00:02:27

LDP discovery sources:

Serial1/0, Src IP addr: 12.1.1.2

Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive

Addresses bound to peer LDP Ident:

23.1.1.2 12.1.1.2 2.2.2.2

r1#

说明:可以看出和 R2 之间存在远程会话信息。

（3） 再看 discovery 信息：

r1#sh mpls ld discovery detail

Local LDP Identifier:

1.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/0 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0

Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 2.2.2.2/32

Serial1/1 (ldp): xmit

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

Targeted Hellos:

1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv

Hello interval: 10000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0

Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2

Hold time: 90 sec; Proposed local/peer: 90/90 sec

Reachable via 2.2.2.2/32

r1#

说明:同样能看到远程会话信息。

（4） 断开 R1 和 R2 之间的直连链路测试:

r1#sh mpls ld discovery detail

Local LDP Identifier:

1.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

Targeted Hellos:

1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv

Hello interval: 10000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0

Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2

Hold time: 90 sec; Proposed local/peer: 90/90 sec

Reachable via 2.2.2.2/32

r1#

r1#sh mpls ld neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.39052 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 15/13; Downstream

Up time: 00:03:27

LDP discovery sources:

Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive

Addresses bound to peer LDP Ident:

23.1.1.2 12.1.1.2 2.2.2.2

r1#

说明:可以看见，R1 和 R2 之间的直连链路断开后，LDP 会话还在。

3.手工配置远程会话

说明:之前是通过会话保护的功能产生的远程会话，下面通过手工创建远程会话。

（1） R1 上创建远程会话：

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

说明:需要指定对方 Router-ID 地址

r1(config)#mpls ldp neighbor 2.2.2.2 targeted ldp 不指定 LDP，默认为 TDP

（2）R2 上创建远程会话：

R2(config)#mpls ldp neighbor 1.1.1.1 targeted ldp

（3）查看效果:

先看建立之前的状态：

r1#sh mpls ld neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.23261 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 9/9; Downstream

Up time: 00:00:04

LDP discovery sources:

Serial1/0, Src IP addr: 12.1.1.2

Addresses bound to peer LDP Ident:

23.1.1.2 12.1.1.2 2.2.2.2

r1#

说明:可以看出没有任何远程会话的信息。

再看建立之后的：

r1#sh mpls ldp neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.23261 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 10/10; Downstream

Up time: 00:01:23

LDP discovery sources:

Serial1/0, Src IP addr: 12.1.1.2

Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive

Addresses bound to peer LDP Ident:

23.1.1.2 12.1.1.2 2.2.2.2

r1#

r1#sh mpls ld discovery detail

Local LDP Identifier:

1.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/0 (ldp): xmit/recv

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0

Src IP addr: 12.1.1.2; Transport IP addr: 2.2.2.2

Hold time: 15 sec; Proposed local/peer: 15/15 sec

Reachable via 2.2.2.2/32

Serial1/1 (ldp): xmit

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

Targeted Hellos:

1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv

Hello interval: 10000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0

Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2

Hold time: 90 sec; Proposed local/peer: 90/90 sec

Reachable via 2.2.2.2/32

r1#

说明:可以看出和 R2 之间存在远程会话信息。

（4）断开 R1 和 R2 之间的直连链路测试:

r1#sh mpls ld discovery detail

Local LDP Identifier:

1.1.1.1:0

Discovery Sources:

Interfaces:

Serial1/1 (ldp): xmit

Enabled: Interface config

Hello interval: 5000 ms; Transport IP addr: 1.1.1.1

Targeted Hellos:

1.1.1.1 -> 2.2.2.2 (ldp): active/passive, xmit/recv

Hello interval: 10000 ms; Transport IP addr: 1.1.1.1

LDP Id: 2.2.2.2:0

Src IP addr: 2.2.2.2; Transport IP addr: 2.2.2.2

Hold time: 90 sec; Proposed local/peer: 90/90 sec

Reachable via 2.2.2.2/32

r1#

r1#sh mpls ldp neighbor

Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0

TCP connection: 2.2.2.2.23261 - 1.1.1.1.646

State: Oper; Msgs sent/rcvd: 15/13; Downstream

Up time: 00:02:50

LDP discovery sources:

Targeted Hello 1.1.1.1 -> 2.2.2.2, active, passive

Addresses bound to peer LDP Ident:

23.1.1.2 12.1.1.2 2.2.2.2

r1#

说明:可以看见，R1 和 R2 之间的直连链路断开后，LDP 会话还在。

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

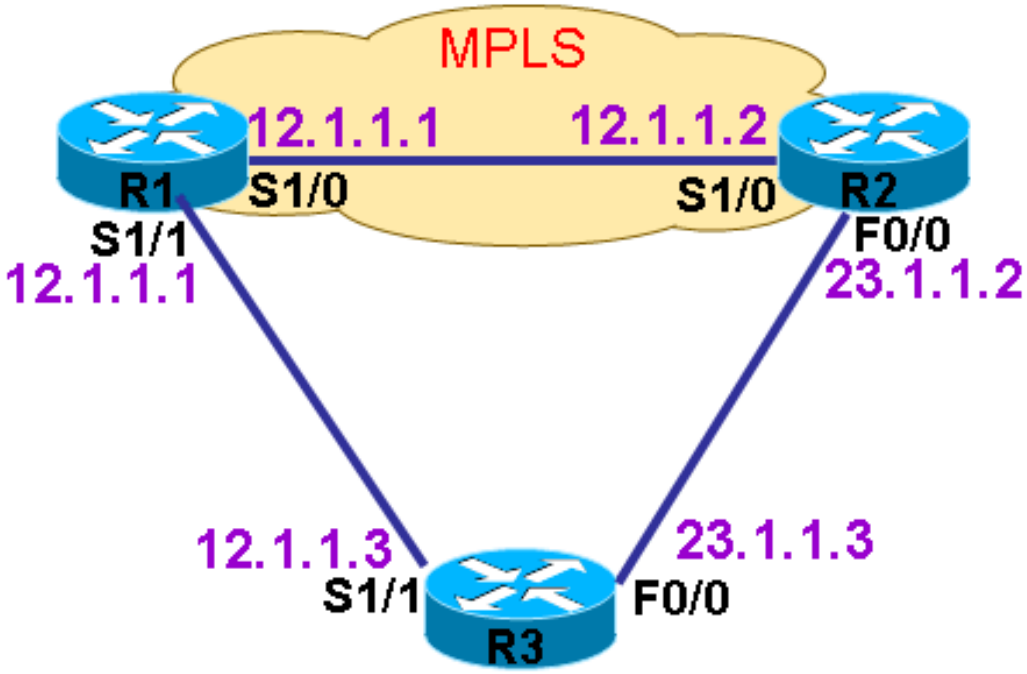
IGP 和 LDP 同步

概述

在某些情况下，当 LDP 邻居还没有建立或者邻居丢失而没有为路由发送标签时，如果这时 IGP 邻居已经建立并学到路由条目后，就会开始 IP 交换，那么后来当 LDP 正常以后，可能会出现丢包的情况，那么就利用 IGP 和 LDP 同步的特性，来限制 LSR，只有当 IGP 和 LDP 都认为某条链路该转发，才转发流量。

IGP 和 LDP 同步的特性只能在接口下配置过 mpls ip 时可以使用，并且目前只支持 OSPF 和 LDP 的同步。当配置好同步以后，相应的接口下，OSPF 邻居在 LDP 邻居还没有建立之前，自己是不会建立邻居关系的，但是可以配置一个最大等待时候，称为 holddown 时候，默认是没有配的，如果 OSPF 在这个 holddown 时间过后，LDP 邻居还没有建立起来，那么自己还是会建立邻居关系，但是，当 OSPF 在 LDP 没有建立邻居关系时，自己从邻居收到的路由条目将会被打上 Metric 65536，这个值是很大的。

配置



说明: OSPF 在 R1 和 R2 之间开启，还在 R1 和 R3 之间开启，而 LDP 只在 R1 和 R2 之间的接口开启。并且 R1 的 loopback0 为 1.1.1.1/32，R2 的 loopback0 为 2.2.2.2/32，R3 的 loopback0 为 3.3.3.3/32，所有 loopback 均放入 OSPF 进程。

1.配置 IGP 和 LDP 的同步

说明:在 R1 上配置 IGP 和 LDP 的同步

（1）在 OSPF 进程下开启 IGP 和 LDP 的同步：

注：同步只在接口下配置了 mpls ip 才会生效。同步在进程下开启之后，所以接口生效，也可以基于相应接口关闭。

R1(config)#router os 2

R1(config-router)#mpls ldp sync

2.查看配置

```
r1#sh ip ospf mpls ldp interface
Serial1/1
  Process ID 2, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Not required
  Holddown timer is disabled
  Interface is up
Serial1/0
  Process ID 2, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Required
  Holddown timer is configured : 10000 msecs
  Holddown timer is not running
  Interface is up and sending maximum metric
Loopback0
  Process ID 2, Area 0
  LDP is not configured through LDP autoconfig
  LDP-IGP Synchronization : Not required
  Holddown timer is disabled
  Interface is up
```

r1#

说明:从结果中看出，只要接口 S1/0 同步才生效，因为只有此接口配了 mpls ip，但是接口下并没有 Holddown。

r1(config)#mpls ldp igp sync holddown 10000

3.配置 Holddown

说明：配置了同步以后，OSPF 在 LDP 没有建立邻居之前，自己是不会建立邻居关系的，但是如果超过了 Holddown 所限制的时间，即使 LDP 邻居关系还没有建立，OSPF 还是会强行建立自己的邻居关系的。

4.查看同步的效果

说明:最终的效果为，在 R2 还没有配 LDP 之前，也就是 R1 的 LDP 邻居关系不能建立，在 Holddown 时间之前，OSPF 邻居关系也没有，但是过了这个时间，LDP 邻居还没有的话，OSPF 还是会建立邻居的，但是 R1 将 R2 发过来的路由条目的 Metric 值设置成 65536，然后发给邻居。

（1）在 R1 上查看 R2 发来的 OSPF 路由：

到 R3 上去看 R2 发给 R1 的路由，然后 R1 再发给自己后 Metric 是 65536，正常情况下不是

r1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

C 1.1.1.1 is directly connected, Loopback0

2.0.0.0/32 is subnetted, 1 subnets

O 2.2.2.2 [110/65536] via 12.1.1.2, 00:00:55, Serial1/0

3.0.0.0/32 is subnetted, 1 subnets

O 3.3.3.3 [110/65] via 13.1.1.3, 00:00:55, Serial1/1

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/74] via 13.1.1.3, 00:00:55, Serial1/1

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/1

r1#

说明:可以看到，R1 上对于 R2 发来的路由条目 2.2.2.2，Metric 值是 65546。

（2）在 R3 上查看路由：

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

O 1.1.1.1 [110/65] via 13.1.1.1, 00:00:08, Serial1/1

2.0.0.0/32 is subnetted, 1 subnets

O 2.2.2.2 [110/65600] via 13.1.1.1, 00:00:08, Serial1/1

3.0.0.0/32 is subnetted, 1 subnets

C 3.3.3.3 is directly connected, Loopback0

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/128] via 13.1.1.1, 00:00:08, Serial1/1

13.0.0.0/24 is subnetted, 1 subnets

C 13.1.1.0 is directly connected, Serial1/1

r3#

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

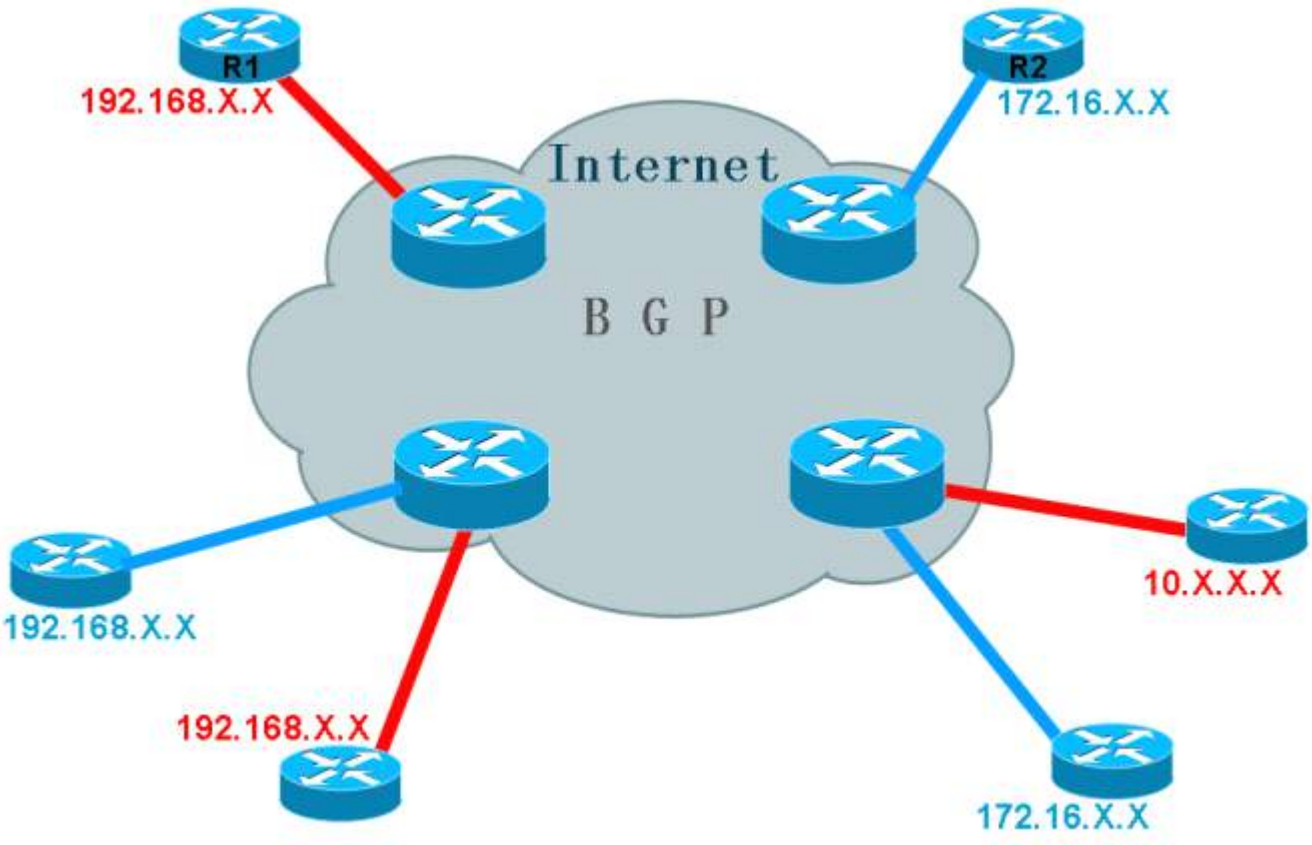
说明:在 R3 上看到 R1 发来的路由 2.2.2.2 的 Metric 是设置了 65536，再加上自己出口的 64，结果为 65600。

MPLS_VPN

概述

从前面可以看出，如果在英特网上大范围部署 MPLS 标签传输网络，这并没有给网络的的速度带来多少优势，而 MPLS 除了能够实现流量工程以外，还有一个很多人都认为很有优势的功能，那就是实现 VPN，具体 VPN 的效果和 VPN 的作用，详细过程可以参见本站“IPSec VPN”主题部分。在理解 MPLS_VPN 之前，应对 VPN 的功能有正确的认识。正因为用户的网络都是私有网络地址，所以这些用户之间的私有网络传输，并不能在 IP 公网上面很好的传递，如果一旦用户的私有网络在公网上传递时，将会有无数个相同的私有网络，这将给 IP 网络带来麻烦。而又因为核心网部署 MPLS 网络之后，这样的网络可以不检查数据包的 IP 地址而进行传输，所以能让用户的私有网络在公网上传输的目的得以实现。不要忘了 IPSec VPN 同样可以做到这样的效果，而 MPLS_VPN 还有一个特点就是可以轻松实现多用户之间的全互联网络，但是我个人认为，这并不能称为 MPLS_VPN 的优势，因为用户要实现 VPN 传输，也许想要解决的，不仅仅是用私网地址通信这么单纯的目的，而数据的加密和安全性，也是应该被重点关心的，而这些，MPLS_VPN 要做到是有难度的。不仅是这样，要完成两个远程用户网络之间的 MPLS_VPN，必须得保护这两个远程网络之间所经过的所有核心网都要支持 MPLS，而且核心网和用户网之间必须相互配合，不能出一点差错，否则也会给 MPLS_VPN 带来麻烦，也就是说，当任何一方对自己来说如果比较难以控制，那么要实施和排错 MPLS_VPN 将会存在相当的难度。而 IPSec VPN，可以将所有问题统统解决在用户手里，也就是说实施和排错 IPSec VPN 时，用户将掌握所有控制权利，将问题解决在自己手中。所以上述两种 VPN，用户可以根据自己的情况来作出选择，需要提醒的是，IPSec VPN 存在着多种 VPN 架构，需要清楚地理解其每种架构的实现方法和作用。

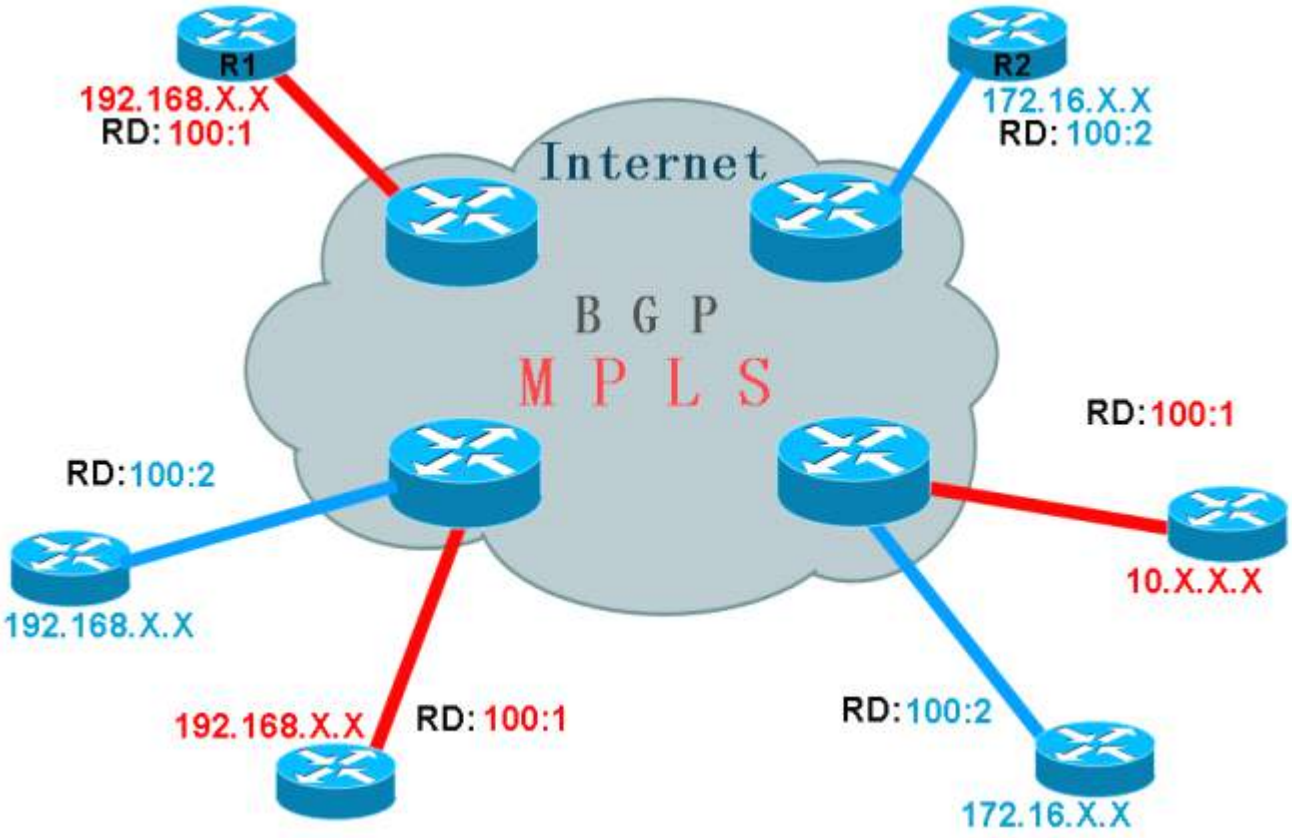
下面来详细讲解 MPLS_VPN 的实现方式。



如上图所示，在 Internet 的边缘，连接着多个用户网络，这些用户网络使用的 IP 网段都是私有网段，如：192.168.0.0，172.16.0.0，10.0.0.0，而这些网段是无法在 Internet 上进行传递的，因为如果用户的私有网段被放入 Internet 时，将意味着有无数个用户网段会发生地址冲突，而最终导致 Internet 路由器无法区分这些网段谁是谁了。既然这样，用户之间要直接通过对方的私有地址进行访问，正常情况下是不可能的。那么要帮助用户将这些私有网段在 Internet 上进行传递，除非使用一些技术能够让这些私有网络在 Internet 上进行传递时，是唯一的，是互不干扰的。这样的技术其实已经存在，那就是在 Internet 中运行 MPLS，因为在 MPLS 网络中，LSR 是不看 IP 地址进行数据转发的，所以用户的私有网络在进入 MPLS 网络时，无论他们是私有的还是重叠都的，对于 MPLS 来说，这些都不是问题，只要他们的标签是正确的，就能被 MPLS 正确转发。

RD（路由区分符）

根据以上所说，在 Internet 核心内部运行 MPLS 网络之后，用户的私有网络可以在 MPLS 里面自由传递，但是用户要想到达另外的用户网络，终究也是要出 MPLS 网络的，当用户的网络在 MPLS 中到达 Internet 边缘路由器时，如果这些路由器还是无法区分用户的私有网络，那么通信的问题依旧存在。如果要消除这个大问题，那么就必须让 Internet 边缘路由器能够正确拥有用户的私有 IP 网络，并且要正确的区分他们，只有这样，用户之间的私有网络通信才能成为可能。（请看下图）



如上图所示，要让 Internet 边缘路由器拥有用户的私有网络，那么唯一的方法就是在边缘路由器与边缘路由器之间运行 BGP，因为在 MPLS 中只要保证 BGP 之间的邻居地址可达就行，这个是可以轻易做到的，而当 BGP 在将用户的数据转入核心 MPLS 网络时，因为 MPLS 核心网络是只看标签而忽视用户私有网络的 IP 地址的，所以这不会存在任何问题，

虽然这样看来问题已经解决了，但是，还有一个最大的问题，那就是虽然

Internet 核心部分已经可以正确传递用户的私有网络了，Internet 边缘路由器也成功的拥有了用户的私有网络地址，可是，用户的网络，毕竟是私有网段，即使 BGP 拥有了用户的私有网段，那么当存在着多个用户的私有网段时，他们的地址势必会重叠和冲突，就如上图所示，有多个用户的网段都是 192.168.0.0 或者都是 172.16.0.0，这样的情况出现后，BGP 也是无法区别用户的网络谁是谁了。要让 BGP 正确区分哪条私有网段是属于哪个用户，就还得给用户这些网段加上额外的标记才行，只有这样，才能让 BGP 正确区别各个用户网段并且允许他们重叠。

既然要给用户的私有网段再加上额外的标记让 BGP 能够正确区分他们，就得考虑这些标记是否能被正确支持。当考虑使用额外标记时，对于 BGP 来说，这已经不算问题，因为 BGP 可以视任何标记为外部属性，既然自己不能理解，也能很好的传递出去。

由于上述原因，便在用户的私有网段进入 BGP 时，就被加上了额外的标记以区分它们，这样的标记被称为路由区分符（RD）。所有的用户私有网络在被 BGP 传递时，都加入了 RD，BGP 要支持这些 RD 的传递，就是多协议的 BGP（MP-BGP），所以 MP-BGP 在实现 MPLS_VPN 时，是必不可少的。

原来用户的网段地址长度都是 32 bit，而 RD 长度是 64 bit，当用户的地址被加上 RD 之后，就变成了 96 bit，这样的地址被称为 vpnv4。

RD 的格式

RD 分为两种格式：

ASN:nn（常用）和 IP-address:nn

ASN 代表 BGP AS 号码，nn 代表数字，数字可以随便定义，只要合理即可，但这个数字，对于一台路由器上的不同用户，肯定是不同的。

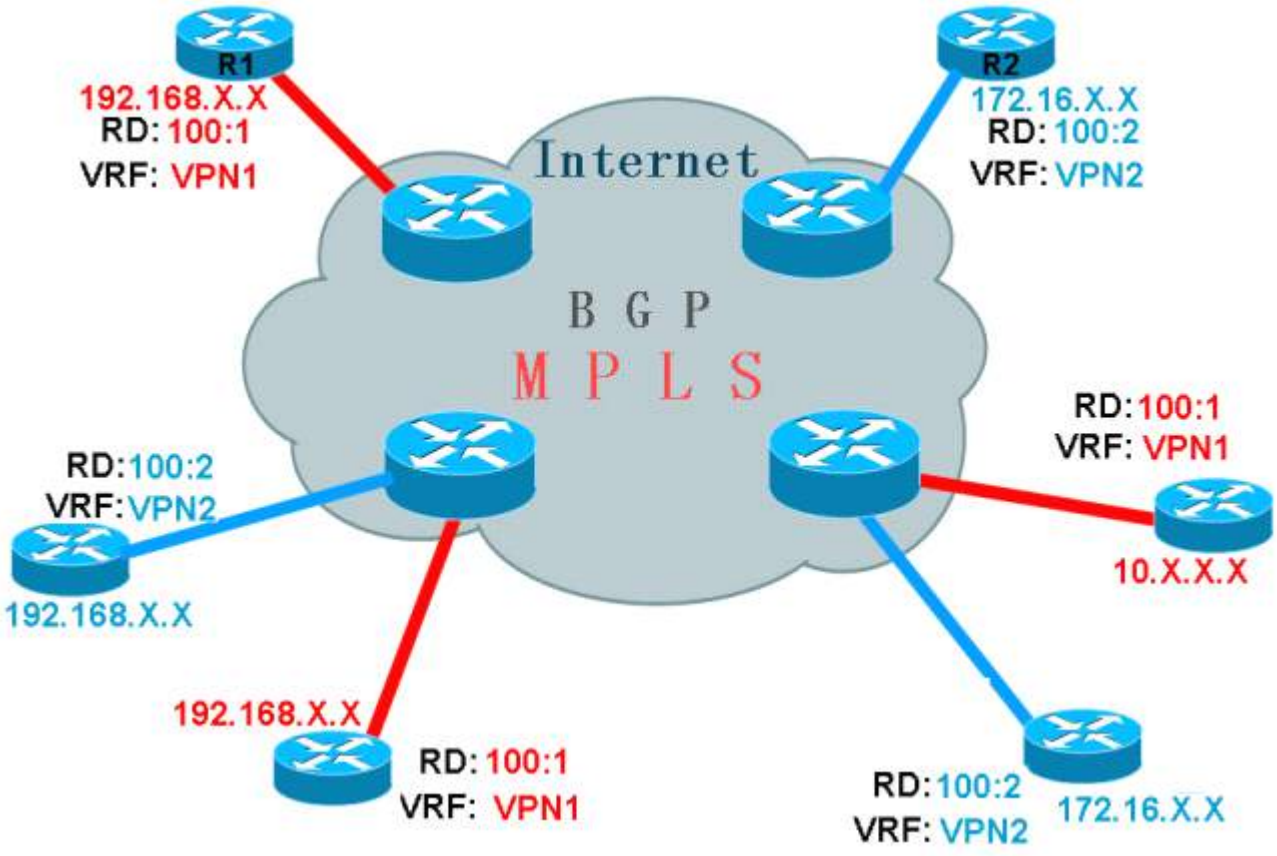
比如一个用户的网段是 10.1.1.0/24，RD 是 100:1，那么用户的 vpnv4 为 100:1:10.1.1.0/24

VRF（虚拟路由表）

到此为止，还并不能让用户网络之间直接通过内网地址访问，也就是还不能实

现 VPN 功能。

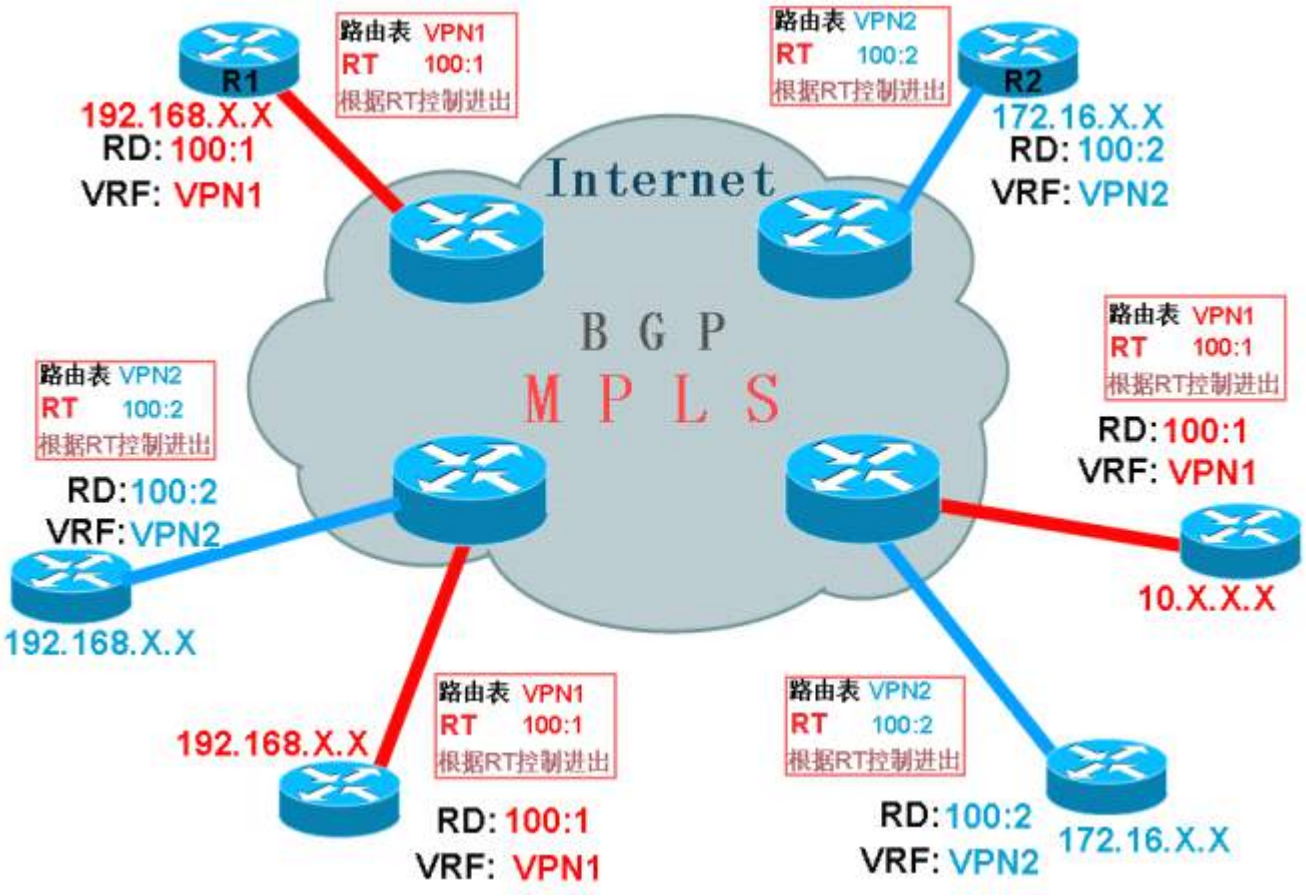
问题在于，当两个用户都连到同一台 Internet 边缘路由器时，比如他们的网段都是 192.168.0.0/24，虽然运行了 MP-BGP 的边缘路由器能够区分它们是两条不同的路由条目，如下图所示：



当左下角两个用户连到同一台边缘路由器时，因为两个用户的网段都是 192.168.0.0，既然 BGP 能够知道它们是两条网段，可是当远程有一个用户，比如左上角的用户（左上角红色网络 R1）要访问左下角红色网络时，它发出数据包的目的地址为 192.168.0.0，边缘路由器又如何能够正确将该数据包发给红色网络的 192.168.0.0，而不会错发给蓝色网络的 192.168.0.0 呢？这个问题值得思考。出现这样的问题，正因为边缘路由器中存在着两条 192.168.0.0 的网段，所以路由器无法区分数据包到底该发给谁，要解决这个问题，很明显，只要让路由器的路由表中只存在一条 192.168.0.0 的网段即可。那么又怎样才能让路由器中只存在一条不会重叠的私有网段呢，那就是在路由器上创建多个路由表，而这个路由表就只包含需要通信的用户网络，比如上图中只包含左上角红色网络和左下角红色网络，那么这样一来，当左上角红色网络 R1 要发数据包给左下角红色网络时，边缘路由器因为路由表中只存在一条 192.168.0.0，所以就再也不会错把数据发给蓝色网络了。路

由器上为需要通信的用户之间创建单独的路由表，这样的路由表被称为虚拟路由表（VRF），如果一台边缘路由器连接着多个不同用户，那么它将创建多个虚拟路由表，这个路由表和普通的路由表没有任何区别，只不过它只用来为 VPN 用户传递数据的，而与虚拟路由表相对的正常路由表被称为全局路由表。

RT（路由对象）

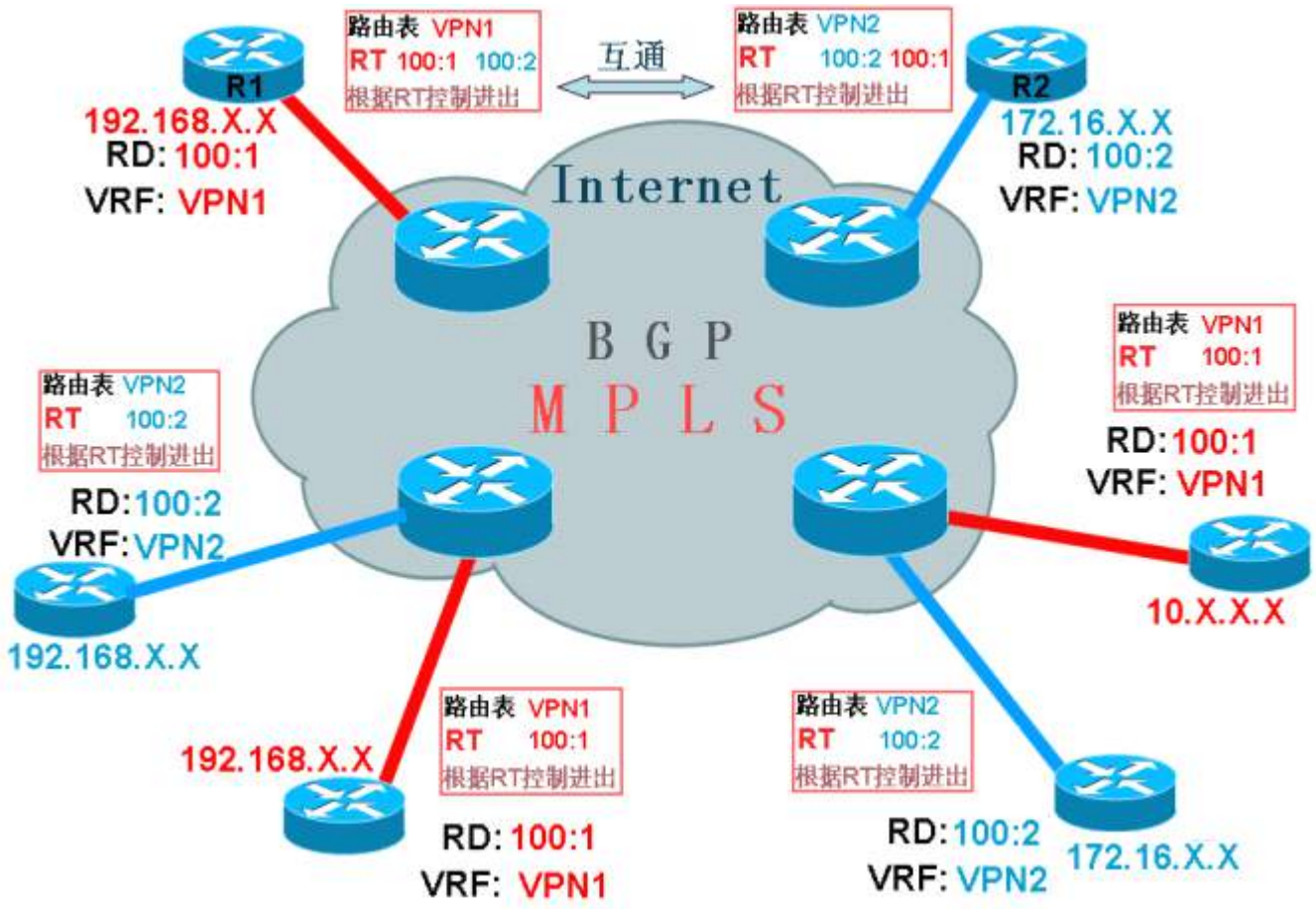


如上图所示，如果在刚才左下角的边缘路由器上为需要通信的红色网络创建 出 VRF 之后，如 VRF 名字为 VPN1，那么又如何保证只有需要通信的红色网络的网段会进入路由表，而其它不相关的网络（如蓝色网络）不会错误地被放进路由表呢？这些，都必须有所控制，否则用户之间的网络还是混乱的。

可以回想一下，之前我们说过，用户的网络在进入 BGP 之间，都是会被标记 RD 的，既然我们需要让红色的网络之间可以通信，让蓝色的网络不能进入 VPN1，那

么我们完全可以在给路由标记 RD 时，就将红色的网络标记为相同的 RD，比如 100:1，而将蓝色网络的 RD 标记为 100：2，这样一来，边缘路由器就可以正确匹配它们了。因为我们已经为红色的网络创建了 VRF 为 VPN1，而红色网络的 RD 都为 100：1，蓝色网络的 RD 都为 100：2，我们就可以完全控制只让 RD 为 100:1 的路由条目能够进入 VPN1，这样就能达到我们所有的目的，并且不会混淆不同用户的网络。这样控制路由表只能进入什么样的路由或者只能出去什么样的路由，被称为 RT（路由对象）。那么如果要让蓝色的网络之间可以相互通信，就可以为他们单独创建 VRF，如 VPN2，并且他们的 RD 是 100：2，最终就可以控制 RT 只让 RD 为 100：2 的路由条目能够进入 VPN2。通过这一切，就可以实现红色网络之间拥有单独的路由表而互通，让蓝色的网络也拥有自己单独的路由表而互通了。

上面所说的控制什么样的 RD 值能够进入什么样的 VRF 路由表，是 RT（路由对象）来控制的，比如一个 VRF 为 CCIE，用户的网段分别有 RD 为 100：1 的，也有 100：2 的和 100：3 的，我们只想让 RD 为 100：3 的路由条目能够进入 CCIE 的话，那么我们就可以设置 CCIE 的 RT 为 100：3。这样做之后，只有 RD 为 100：3 的路由条目才能进入 VRF CCIE，从而和其它 RD 为 100：1 和 100：2 的网络隔离开来。RT 是 BGP 扩展团体属性，分为输入和输出两种，可以简单理解为什么样的 RD 值可以进入该 VRF 或者该 VRF 什么样 RD 值的路由将被导出。如果路由的 RD 值和 RT 所允许的值不匹配，将不能进入或者出去的，这样不同 RD 的用户也就不能通信，想要通信，就配置为相同的 RD。但是一个 VRF 中，可以配置多个 RT，也就是说一个 VRF 可以让多个不同 RD 值的路由进入或出去，最终结果为，如果要让两个不同 RD 的用户要通信，就分别为两个用户各自的 VRF 同时配置两个用户的 RD，这样大家的 VRF 中都有互相的路由条目，也就可以互通了。如下图所示：



从上图中显示，红色网络中的 R1 和蓝色网络中的 R2，默认情况下，他们是不能通信的，因为红色网络的 VRF 只允许 RD 为 100：1 的路由条目进入，而蓝色网络的 VRF 只允许 RD 为 100：2 的路由条目进入，但是配置为大家的 VRF 都同时允许 RD 为 100：1 和 100：2 进入，最后两个 VRF 中都有了对方的路由，也就实现了互通。

一般情况下，要通信的用户之间，RD 都是配置为一样的，这样的通信称为内部通信，如果不同 RD 之间的网络也要通信，就可以分别在相互的 VRF 中配置 RT 允许双方的 RD 值都能进入和出去，也能够通信，这样的通信就称为外部通信。所以用户之间是否能通信，完全是由 RT 来控制的。RD 是让 BGP 来区分用户路由条目的，VRF 是让路由器来区分用户网络的。RT 是用来控制 VRF 的路由进入和出去的。

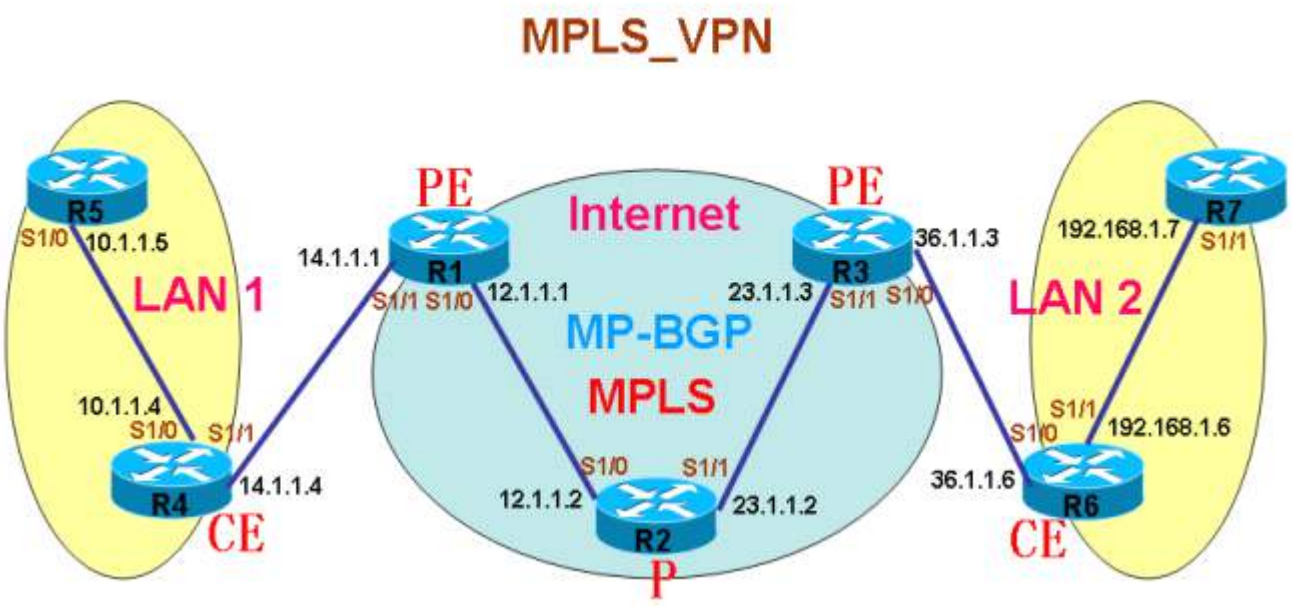
ISP 中存在 P 和 PE，其中 PE 直接和用户相连，P 则没有，但 P 和 PE 必须已经运行 MPLS。

用户中存在 CE 和 C，CE 是直接和 ISP 相连的路由器，但都不需要运行 MPLS。

CE 和 PE 直接交互，所以必须运行路由协议，或者是写静态路由，CE 只有一个对等 PE，但也可多宿主，也就是和多 PE 相连，

VPN，让 P 完全不用知道 VPN，就不用负担 VPN 的路由信息了，则使用 MPLS 来实现，为用户的私有网络打上标签，然后 P 也不用运行 BGP。但用户的路由必须在 PE 上出现，PE 是必须了解用户私有网络的。

MP-BGP



如上图所示，用户局域网 LAN1 连接 Internet，用户局域网 LAN2 也连接 Internet，当 LAN1 要和 LAN2 直接通过内网地址进行相互访问时，只要之间实现 MPLS_VPN 即可，在实施 MPLS_VPN 时，Internet 中之前是运行着 MPLS 的，我们已经知道，在 MPLS 中，中间的 LSR 称为 P，而 MPLS 中连接着用户网络的边缘 LSR 称为 PE，用户连接 PE 的路由器称为 CE，而用户自己内部的路由器可以称为是 C。

用户的网段到达 PE 时，因为要让 BGP 区分不同的用户网段，所以 PE 会为它加上 RD，加上了 RD 的网段称为 vpnv4，这时 PE 会将 vpnv4 的路由放入 MP-BGP，然后 MP-BGP 再将 vpnv4 从 MPLS 网络中通告给对端的 BGP 邻居，但是要保证 vpnv4 到达对端 BGP 邻居之后，对方还能够认识这这些 vpnv4 的 RD，那么 BGP

就必须为 vpnv4 打上相应的标签，对方 BGP 邻居看了这个标签，就能根据 RD 将其放入相应的 VRF，从而将数据包根据这些 VRF 转发给相应的 CE。正因为 BGP 要为 vpnv4 加入额外的标签，所以要使用 MP-BGP。

在 MPLS 中，数据包可以被打上多个标签，而 LSR 在转发时，只看顶部的标签，也就是说在数据包的多个标签中，只有顶部这一个标签会被使用和修改，所以 MP-BGP 在给 vpnv4 加入标签时，只要加在顶部标签的下面，这样数据包在 MPLS 中传输时，就不会被修改了，所以对端 BGP 邻居能够正确识别数据包的相应 RD，这也是为什么 MPLS_VPN 中数据包需要使用两个标签的理由。

MP-BGP 规则

普通的 BGP 通过额外配置 address-family 之后，就可以实现 MP-BGP 的功能，普通的 BGP 只是能够传递普通 IPv4 的路由，所以这样也会有个默认 address-family，称为 ipv4，但是因为 IP 分为单播和多播，所以这种默认的 address-family 就是 ipv4 unicast，如果要让 BGP 支持 ipv4 多播，那 address-family 就是 ipv4 multicast。

如果要支持 IPv6 的单播和多播，那么 address-family 就分别为 ipv6 unicast 和 ipv6 multicast。

在这里，以上的都不是我们要用到的 address-family，因为我们要传递的即不是 ipv4 单播，也不是 ipv4 多播，更不是 ipv6，我们要传递的是 vpnv4，所以就要开启 MP-BGP 支持 vpnv4，要支持 vpnv4，也需要创建相应的 vpnv4 的 address-family。并且需要创建相应的 VRF，这样相应的 vpn4 就和相应的 VRF 相关联起来。所有多协议的 BGP 在运行之前，应该保证普通的 BGP 邻居是正常的。MP-BGP 在为 vpnv4 通告标签时，并且所有信息要当作团体属性带出去，要手工指定。

注： IOS 缺省为每个 vpnv4 分配一个唯一的 MPLS 标签。

PE-CE 路由协议

不同用户网络之间要通过 MPLS_VPN 进行通信，就需要 PE 上有用户网络的路由信息，PE 再将这些路由在 MP-BGP 中通告给对端 MP-BGP 邻居。在这一切开始之前，PE 就应该获得用户的内部路由信息，而让 PE 获得这些路由信息的方法，可以使用动态路由或静态路由，如果要使用动态路由，那么 PE 和 CE 之间就必须启用某些路由协议，否则如果 PE 上没有用户的私有网络，那么远程用户之间也就不可能通信了。在 PE 有了用户的路由协议之后，必须将这些路由信息重分布进 MP-BGP，再由 MP-BGP 通告给对端邻居，从而到达远程用户，但是如果 CE 没有远程用户的私有网段信息，也是不能通信的，所以在 PE 上，MP-BGP 的路由也同样要发给 CE，也可以通过将 MP-BGP 的路由重分布进 PE 和 CE 间的路由协议，虽然普通 BGP 不允许将自己的路由重分布进 IGP，但 MP-BGP 不受此限制。

还需要说明的是，PE 和 CE 相连的某个接口，是属于某一个 VRF 的，所以从该接口进来的 CE 路由，都属于该 VRF。

PE 和 CE 之间支持的协议有

静态路由

RIPv2

OSPF

EIGRP

IS-IS

协议配置方法

在配置这些协议时，因为协议需要同时运行在 PE 和 CE 上，所以配置的步骤也分为 PE 和 CE 两步。

1.静态路由

（1）PE 上直接对某个 VRF 写静态路由:

r4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.4

（2）在 PE 上将静态路由重分布进 MP-BGP:

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute static

r1(config-router-af)#exi

(3)CE 上要有默认路由指向 PE:

r4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.1

2.RIPv2

注：RIP 路由 metric 值超过 15，路由无效，而外部路由重分布进 RIP 时，默认不指定 metric 的情况下，值为无穷大，所以默认将不显示在 RIP 中，所以请指明 metric 或配置 default-metric

（1）PE 上配置 RIP:

r1(config)#router rip

r1(config-router)#version 2

r1(config-router)#no auto-summary

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#no auto-summary

r1(config-router-af)#network 14.0.0.0

r1(config-router-af)#redistribute bgp 100 metric 1

（2）将 RIP 重分布进 MP-BGP:

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute rip

（3）CE 上正常配置 RIP 即可:

r4(config)#router rip

r4(config-router)#version 2

r4(config-router)#no auto-summary

r4(config-router)#network 14.0.0.0

r4(config-router)#network 10.0.0.0

3.OSPF

说明:OSPF 还具有 Sham-lin（伪装链路的功能，此功能将不作任何讲述，如果 CCIE R&S v4.0 考试有该要求，本篇将立即加入详细讲解和配置过程!）

（1）在 PE 上配置 OSPF:

r3(config)#router ospf 100 vrf vpn1

r3(config-router)#router-id 36.1.1.3

r3(config-router)#network 36.1.1.3 0.0.0.0 area 0

r3(config-router)#redistribute bgp 100 subnets

（2）将 OSPP 重分布进 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#redistribute ospf 100

（3）CE 正常配置 OSPF:

r6(config)#router ospf 100

r6(config-router)#router-id 6.6.6.6

r6(config-router)#network 36.1.1.6 0.0.0.0 area 0

r6(config-router)#network 192.168.1.6 0.0.0.0 area 0

r6(config-router)#exit

r6(config)#

4.EIGRP

（1）在 PE 上配置 EIGRP:

r3(config)#router eigrp 1

r3(config-router)#no auto-summary

r3(config-router)#address-family ipv4 vrf vpn2

r3(config-router-af)#no auto-summary

r3(config-router-af)#network 83.1.1.3 0.0.0.0

r3(config-router-af)#autonomous-system 1

r3(config-router-af)#redistribute bgp 100 metric 10000 100 255 1 1500

（2）将 EIGRP 重分布进 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn2

r3(config-router-af)#redistribute eigrp 1

（3）CE 正常配置 EIGRP:

r8(config)#router eigrp 1

r8(config-router)#no auto-summary

r8(config-router)#network 172.16.1.8 0.0.0.0

r8(config-router)#network 83.1.1.8 0.0.0.0

IS-IS 暂且不作说明

5.EBGP

IOS 只支持 CE 和 PE 之间运行 EBGP

（1）PE 上配置 EBGP:

r1(config)#router bg 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#neighbor 14.1.1.4 remote-as 200

r1(config-router-af)#neighbor 14.1.1.4 activate

（2）和 CE 和 EBGP 默认会重分布进 MP-BGP:

（3）CE 上配置 EBGP:

r4(config)#router bgp 200

r4(config-router)#neighbor 14.1.1.1 remote

r4(config-router)#neighbor 14.1.1.1 remote-as 100

r4(config-router)#network 10.1.1.0 mask 255.255.255.0

r4(config-router)#

如果所有客户网络 AS 都是一样的，CE 会拒绝，因为看到与自己相同的 AS，但服务提供商可以比较，如果是一样的，就修改成自己的发过去，

服务提供商配置为：

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#neighbor 14.1.1.4 as-override

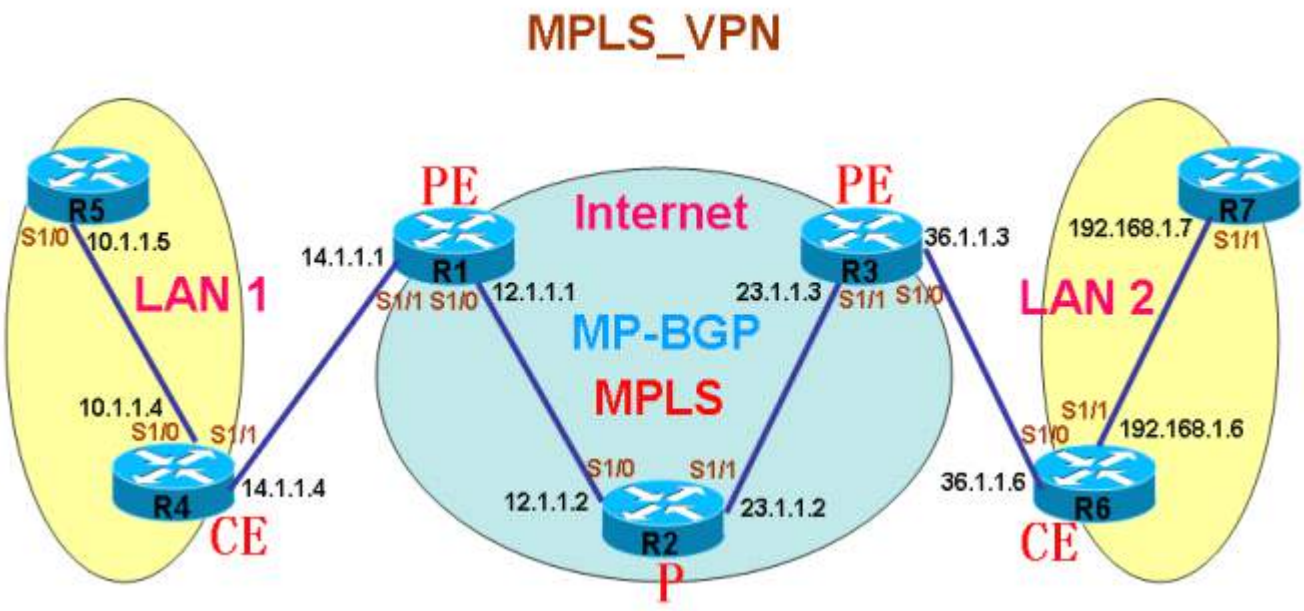
PE 给 CE 再回来，hub-spoke，就配：nei allowas-in 1-10

配置 MPLS_VPN

概述

以下面一张图为拓朴，详细介绍配置 MPLS_VPN，其中 R1、R2、R3 上均已配

置 loopback0，地址分别为 1.1.1.1/32，2.2.2.2/32，3.3.3.3/32，并且已经配置好 OSPF，让 MPLS 区域内所有直连接口和 loopback 口互通。



1.配置 MPLS

说明:因为 MP-BGP 是配置 MPLS_VPN 的必须协议，在开始配置 MPLS_VPN 之前，应该先将 MPLS 区域的相关接口实现标签交换，如各自的直连口，loopback 口均可看见已经通过标签进行交换。

（1） 配置 MPLS 区域略过，详细步骤请参见之前 MPLS 配置。

（2） 配置 MPLS_VPN 时，需要在边缘路由器 R1 和 R3 之间使用 MP-BGP 协议，我们使用路由器的 loopback 口作为源地址来建立邻居，首先测试双方 loopback 口已经实现标签交换。

r1#traceroute 3.3.3.3

Type escape sequence to abort.

Tracing the route to 3.3.3.3

1 12.1.1.2 [MPLS: Label 17 Exp 0] 104 msec 120 msec 140 msec

2 23.1.1.3 72 msec * 112 msec

r1#

说明:从结果可以看出，R1 到 R3 的 loopback0 已经实现标签交换。

r3#traceroute 1.1.1.1

Type escape sequence to abort.

Tracing the route to 1.1.1.1

1 23.1.1.2 [MPLS: Label 16 Exp 0] 100 msec 140 msec 196 msec

2 12.1.1.1 160 msec * 224 msec

r3#

说明:从结果可以看出，R3 到 R1 的 loopback0 已经实现标签交换。

2.配置普通 BGP

说明:在 R1 和 R3 之间配置普通 BGP，因为在配置 MP-BGP 之前，需要保证正常的 BGP 邻居是正常连通的

（1） 在 R1 上配置普通 BGP：

r1(config)#router bgp 100

r1(config-router)#neighbor 3.3.3.3 remote-as 100

r1(config-router)#neighbor 3.3.3.3 update-source loopback 0

（2）在 R2 上配置普通 BGP：

r3(config)#router bgp 100

r3(config-router)#neighbor 1.1.1.1 remote-as 100

r3(config-router)#neighbor 1.1.1.1 update-source loopback 0

（3）在 R1 上确认与 R3 的普通 BGP 邻居关系已建立：

r1#sh ip bgp summary

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
3.3.3.3	4	100	3	3	1	0	0	00:00:51	0

r1#

说明:可以看到 R1 与 R3 已建立正常 BGP 邻居关系。

（4）在 R3 上确认与 R1 的普通 BGP 邻居关系已建立：

r3#sh ip bgp summary

BGP router identifier 3.3.3.3, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ

Up/Down State/PfxRcd

1.1.1.1 4 100 4 4 1 0 0 00:01:07 0

r3#

说明:可以看到 R3 与 R1 已建立正常 BGP 邻居关系。

3.在 PE 上创建 VRF

说明:在 PE 上为用户创建相应的 VRF，并且指定 RD 值，需要通信的两个用户网络之间，VRF 和 RD 值保持一致。

（1）在 R1 上创建 VRF，并指定 RD 值：

r1(config)#ip vrf vpn1

r1(config-vrf)#rd 100:1

（2）在 R3 上创建 VRF，并指定 RD 值：

r3(config)#ip vrf vpn1

r3(config-vrf)#rd 100:1

4.在 PE 上将连 CE 的接口划入 VRF

说明:在 PE 上将相应的 CE 接口划入相应的 VRF，以后从该接口进入的用户数据包，则属于相应的 VRF，该用户的数据只能根据该 VRF 路由表作出转发决策。

（1）在 R1 上将连 CE R4 的接口 s1/1 划入 VRF:

r1(config)#int s1/1

r1(config-if)#ip vrf forwarding vpn1

% Interface Serial1/1 IP address 14.1.1.1 removed due to enabling VRF vpn1

r1(config-if)# ip add 14.1.1.1 255.255.255.0

说明:当一个正常的接口被划入 VRF 之后，接口上的地址会消失，所以需要重新配置一次该接口的 IP 地址。

(2) 在 R3 上将连 CE R6 的接口 s1/0 划入 VRF:

r3(config)#int s1/0

r3(config-if)#ip vrf forwarding vpn1

% Interface Serial1/0 IP address 36.1.1.3 removed due to enabling VRF vpn1

r3(config-if)#ip add 36.1.1.3 255.255.255.0

5.在 PE 上查看 VRF 的路由表情况

说明:从用户发到 PE 的数据包，PE 只能根据该用户的 VRF 路由表作出转发决策，也就是说，如果两个要通信的用户网络，如果各自的内网路由没有出现在 PE 的 VRF 路由表里，那么他们将不能通信，

（1）在 R1 上查看 VRF vpn1 的路由表:

r1#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, Serial1/1

r1#

说明:可以看见，PE 在将连 CE R4 的接口 s1/1 划入 VRF vpn1 之后，该接口就进入 VRF vpn1 的路由表。并且要说明的是，该接口就不会再出现在全局路由表里。

（2） 查看 PE R1 的全局路由表，已经不会再有连 CE 接口 s1/1 的路由了：

r1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

C 1.1.1.1 is directly connected, Loopback0

2.0.0.0/32 is subnetted, 1 subnets

O 2.2.2.2 [110/65] via 12.1.1.2, 00:14:25, Serial1/0

3.0.0.0/32 is subnetted, 1 subnets

O 3.3.3.3 [110/129] via 12.1.1.2, 00:14:25, Serial1/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/128] via 12.1.1.2, 00:14:25, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

r1#

6.创建 MP-BGP

说明:通过上面在 PE 上查看 VRF 路由表发现，VRF 路由表中并没有双方用户的路由，所以，必须创建 MP-BGP，来为双方用户网络传递路由信息。

（6 **(1)在 PE R1 上创建 MP-BGP:**

r1(config)#router bgp 100

r1(config-router)#address-family vpnv4

r1(config-router-af)#neighbor 3.3.3.3 activate

r1(config-router-af)#neighbor 3.3.3.3 send-community both

说明:因为要传递 vpnv4 的路由，所以创建的 address-family 为 vpnv4，并且将正常

的 BGP 邻居在 vpnv4 里面激活，而且还需要将这些 BGP 的扩展属性手工强行发给对端，否则对方收到的路由信息不会携带扩展属性，也就无法正常区分用户的路由信息。

（2） 在 PE R3 上创建 MP-BGP：

r3(config)#router bgp 100

r3(config-router)#address-family vpnv4

r3(config-router-af)#neighbor 1.1.1.1 activate

r3(config-router-af)#neighbor 1.1.1.1 send-community both

（3） 查看 MP-BGP 邻居：

r1#sh ip bgp all summary

For address family: IPv4 Unicast

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
3.3.3.3	4	100	23	23	1	0	0	00:03:18	0

For address family: VPNv4 Unicast

BGP router identifier 1.1.1.1, local AS number 100

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
3.3.3.3	4	100	23	23	1	0	0	00:03:18	0

r1#

说明:可以看到，R1 上已经和 R3 建立普通 BGP 邻居关系，同时也建立 MP-BGP 邻居关系。

7.查看 MP-BGP 的 VRF 路由

r1#sh ip bgp vpnv4 all

r1#

说明:在 PE 上只是已经创建了 MP-BGP，并没有为 BGP 创建 VRF，所以无法看到 BGP 中的 VRF 路由表信息，所以还需要手工创建 VRF 路由表。

8.为 MP-BGP 创建 VRF

说明:MP-BGP 在收到用户的路由信息后，必须将其放入相应的 VRF 路由表，但是这个 VRF 表是要手工创建的，并且和该用户相关联的 VRF 名字保持一致。

（1）在 R1 上为 MP-BGP 创建 VRF vpn1:

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

（2）在 R3 上为 MP-BGP 创建 VRF vpn1:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

（3）查看已创建的 VRF 路由表的路由条目:

r1#sh ip bgp vpnv4 all

r1#

r1#sh ip bgp vpnv4 vrf vpn1

r1#

说明:可以看到，BGP VRF 路由表中并没有用户的路由信息。

9.配置 RT 控制 VRF 路由信息

说明:因为 MP-BGP 的 VRF 路由表能让什么样的路由进入，是靠 RT 来控制的，所以要想让用户的路由被 MP-BGP 传递，就必须为 VRF 配置相应的 RT，只有 RT 允许的 RD 路由，才能进入和出去 VRF 表。

（1）在 R1 上为 VRF 配置相应的 RT：

r1(config)#ip vrf vpn1

r1(config-vrf)#route-target both 100:1

说明：VRF vpn1 允许 RD 为 100:1 的路由进入和出去。

（2）在 R3 上为 VRF 配置相应的 RT：

r3(config)#ip vrf vpn1

r3(config-vrf)#route-target both 100:1

10.配置 PE-CE 的路由协议

说明:虽然 MP-BGP 的 VRF 已经允许相应的用户路由进入，但是在 PE 上，此时并不能获知用户的路由信息，所以 MP-BGP 的 VRF 路由表中，依然为空，要想让 MP-BGP 的 VRF 路由表能够导入相应的用户路由，那就必须和用户 CE 之前启用路由协议，以获得对方的路由信息，从而导入 MP-BGP 的 VRF 表。

（1）在 PE R1 上配置 RIP:

说明:PE-CE 路由协议 RIP 只支持版本 2

```
r1(config)#router rip
r1(config-router)#version 2
r1(config-router)#no auto-summary
r1(config-router)#address-family ipv4 vrf vpn1
r1(config-router-af)#no auto-summary
r1(config-router-af)#network 14.0.0.0
r1(config-router-af)#redistribute bgp 100 metric 1
```

注：发布路由都是在 address-family 中进行的，并且请关闭自动汇总功能，且将 MP-BGP 的路由重分布进 RIP，否则对方 CE 将无法得知远程用户的路由信息。

（2）在 CE R4 上配置 RIP:

说明:在 CE 上的路由协议和正常路由配置没任何区别，配置完协议后，将可以从 PE 上收到路由信息。

```
r4(config)#router rip
r4(config-router)#version 2
r4(config-router)#no auto-summary
```

r4(config-router)#network 14.0.0.0

r4(config-router)#network 10.0.0.0

（3）在 PE R3 上配置 OSPF:

说明:同样也要将 MP-BGP 的路由重分布进 OSPF，以便传递给 CE 端。

r3(config)#router ospf 100 vrf vpn1

r3(config-router)#router-id 36.1.1.3

r3(config-router)#network 36.1.1.3 0.0.0.0 area 0

r3(config-router)#redistribute bgp 100 subnets

（4） 在 CE R6 上配置 OSPP:

说明:CE 上的路由协议 OSPF 与正常 OSPF 配置无区别。

r6(config)#router ospf 100

r6(config-router)#router-id 6.6.6.6

r6(config-router)#network 36.1.1.6 0.0.0.0 area 0

r6(config-router)#network 192.168.1.6 0.0.0.0 area 0

r6(config-router)#exit

r6(config)#

11.在 PE 上查看 VRF 路由

说明:已经在 PE 和 CE 配置路由协议，所以 PE 上应该已经获得用户的内部路由信息。

(1) 在 R1 上查看是否得到 CE R4 的内部路由信息 10.1.1.0/24:

鸿鹄培训+考试群：301283867

r1#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

R 10.1.1.0 [120/1] via 14.1.1.4, 00:00:27, Serial1/1

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, Serial1/1

r1#

说明:PE R1 已经成功通过路由协议 RIP 获得用户的内部路由信息。

(2) 查看 MP-BGP 的 VRF 路由表中是否已将用户的内部路由信息导入：

r1#sh ip bgp vpnv4 all

r1#

说明:MP-BGP 的 VRF 路由表还没有得到用户的内部路由信息，因为 PE 和 CE 之间运行的是 IGP 协议，所以要想让 IGP 学到的路由进入 MP-BGP 的 VRF 路由表，必须手工重分布。

12.将路由重分布进 MP-BGP

说明:PE-CE 之间在运行 IGP 时，无法自动导入 MP-BGP，所以手工重分布。

（1）在 R1 上将 RIP 路由导入 MP-BGP:

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute rip

（2）在 R3 上将 OSPP 路由导入 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#redistribute ospf 100

13.查看 MP-BGP 路由

说明:MP-BGP 已经和 IGP 之间实现重分布，查看双方路由是否获得。

（1）查看 R1 上 MP-BGP 的 VRF 路由表:

r1#sh ip bgp vpnv4 all

BGP table version is 9, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

Route Distinguisher: 100:1 (default for vrf vpn1)

*> 10.1.1.0/24 14.1.1.4 1 32768 ?

*> 14.1.1.0/24 0.0.0.0 0 32768 ?

*>i36.1.1.0/24 3.3.3.3 0 100 0 ?

*>i192.168.1.6/32 3.3.3.3 65 100 0 ?

r1#

说明:已经拥有双方用户网络的内部路由信息。

（2）查看 R3 上 MP-BGP 的 VRF 路由表:

r3#sh ip bgp vpnv4 all

BGP table version is 9, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

Route Distinguisher: 100:1 (default for vrf vpn1)

*>i10.1.1.0/24 1.1.1.1 1 100 0 ?

*>i14.1.1.0/24 1.1.1.1 0 100 0 ?

*> 36.1.1.

0/24 0.0.0.0 0 32768 ?

*> 192.168.1.6/32 36.1.1.6 65 32768 ?

r3#

14.查看 VRF 路由

说明:MP-BGP 中已经拥有双方用户的网络信息，再看 VRF 中是否全部拥有。

（1）在 R1 上查看 VRF 表：

r1#show ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

鸿鹄培训+考试群：301283867

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

36.0.0.0/24 is subnetted, 1 subnets

B 36.1.1.0 [200/0] via 3.3.3.3, 00:02:50

10.0.0.0/24 is subnetted, 1 subnets

R 10.1.1.0 [120/1] via 14.1.1.4, 00:00:04, Serial1/1

192.168.1.0/32 is subnetted, 1 subnets

B 192.168.1.6 [200/65] via 3.3.3.3, 00:02:50

14.0.0.0/24 is subnetted, 1 subnets

C 14.1.1.0 is directly connected, Serial1/1

r1#

说明:PE R1 上的 VRF 已经拥有双方用户的网络信息，再看 VRF 中是否全部拥有。

（2）在 R3 上查看 VRF 表：

r3#show ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

```
        E1 - OSPF external type 1, E2 - OSPF external type 2

        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

        ia - IS-IS inter area, * - candidate default, U - per-user static route

        o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     36.0.0.0/24 is subnetted, 1 subnets

C       36.1.1.0 is directly connected, Serial1/0

     10.0.0.0/24 is subnetted, 1 subnets

B       10.1.1.0 [200/1] via 1.1.1.1, 00:04:57

     192.168.1.0/32 is subnetted, 1 subnets

O       192.168.1.6 [110/65] via 36.1.1.6, 00:04:01, Serial1/0

     14.0.0.0/24 is subnetted, 1 subnets

B       14.1.1.0 [200/0] via 1.1.1.1, 00:04:57

r3#
```

说明:PE R1 上的 VRF 已经拥有双方用户的网络信息，再看 VRF 中是否全部拥有。

15.查看 CE 路由

说明:因为 PE 上已经拥有双方用户的路由信息，并且 PE 和 CE 之间也运行路由协议，所以这些路由应该出现在 CE 的路由表中，从而双方用户可以实现通信。

（1）查看 CE R4 上的路由表：

```
r4#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     36.0.0.0/24 is subnetted, 1 subnets
R       36.1.1.0 [120/1] via 14.1.1.1, 00:00:26, Serial1/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial1/0
     192.168.1.0/32 is subnetted, 1 subnets
R       192.168.1.6 [120/1] via 14.1.1.1, 00:00:26, Serial1/1
     14.0.0.0/24 is subnetted, 1 subnets
C       14.1.1.0 is directly connected, Serial1/1
r4#
r4#
```

说明:可以看到，CE R4 上已经拥有所有用户的内部路由。

（2）查看 CE R6 上的路由表：

```
r6#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

       E1 - OSPF external type 1, E2 - OSPF external type 2

       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

       ia - IS-IS inter area, * - candidate default, U - per-user static route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     36.0.0.0/24 is subnetted, 1 subnets

C       36.1.1.0 is directly connected, Serial1/0

     6.0.0.0/32 is subnetted, 1 subnets

C       6.6.6.6 is directly connected, Loopback0

     10.0.0.0/24 is subnetted, 1 subnets

O E2    10.1.1.0 [110/1] via 36.1.1.3, 00:00:17, Serial1/0

C    192.168.1.0/24 is directly connected, Loopback192

     14.0.0.0/24 is subnetted, 1 subnets

O E2    14.1.1.0 [110/1] via 36.1.1.3, 00:00:17, Serial1/0
```

r6#

说明:可以看到，CE R6 上已经拥有所有用户的内部路由。

16.测试用户之间通信

说明:因为用户 CE 之间已经拥有双方的内部路由信息，所以应该能够通信

（1）R5 应该具有默认路由将所有数据从 CE R4 上发出去：

r5#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.4 to network 0.0.0.0

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Serial1/0

S* 0.0.0.0/0 [1/0] via 10.1.1.4

r5#

说明:R5 已经拥有指向 CE R4 的默认路由。

（2）LAN 1 的 R5 上测试到 LAN 2 的网段 192.168.1.6 的连通性：

r5#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 188/242/288 ms

r5#

说明:R5 上已经成功穿越 MPLS 网络将数据包发送给远程用户网络，说明 MPLS_VPN 成功。

（3）跟踪路由:

r5#traceroute 192.168.1.6

Type escape sequence to abort.

Tracing the route to 192.168.1.6

1 10.1.1.4 64 msec 124 msec 48 msec

2 14.1.1.1 112 msec 92 msec 116 msec

3 12.1.1.2 260 msec 280 msec 236 msec

4 36.1.1.3 168 msec 196 msec 204 msec

5 36.1.1.6 268 msec * 248 msec

r5#

说明:R5 上已经成功穿越 MPLS 网络将数据包发送给远程用户网络，说明 MPLS_VPN 成功。

17.PE 到 CE 的通信

说明:因为 PE 将连 CE 的接口放入 VRF 之后，该接口就从全局路由表中消失，所以和 CE 的通信，由 VRF 路由表决定。

（1） 查看 PE 的到 CE 的路由:

r1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/32 is subnetted, 1 subnets

C 1.1.1.1 is directly connected, Loopback0

2.0.0.0/32 is subnetted, 1 subnets

O 2.2.2.2 [110/65] via 12.1.1.2, 00:51:02, Serial1/0

3.0.0.0/32 is subnetted, 1 subnets

O 3.3.3.3 [110/129] via 12.1.1.2, 00:51:02, Serial1/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/128] via 12.1.1.2, 00:51:02, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

r1#

说明:可以看到连 CE 的接口已经从全局路由表中消失。

（2）测试 ping:

r1#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r1#

说明:因为正常的 ping 是走的全局路由表，而全局路由表中没有到 CE 的接口，所以不通。

（3）使用 VRF 路由表:

r1#ping vrf vpn1 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 48/67/112 ms

r1#

说明:通过指定到 CE 的数据走 VRF 路由表，所以最后通信成功。

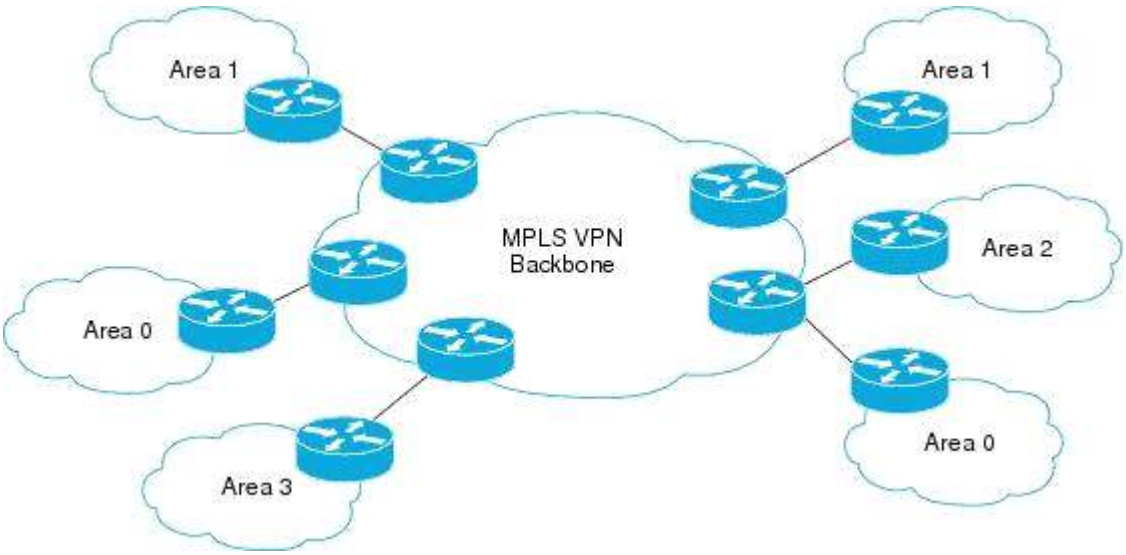
点此查看配置实例所有设备的 show running-config

OSPF Sham-Link

在 PE-CE 路由协议为 OSPF 时，当 OSPF 和 MP-BGP 之间互相重分布时，请不要改变 metric 值，因为这里的 metric 值是受保护的，如果您改了，可能造成路由环路，请小心配置。

当 PE 和 CE 使用 OSPF 时，PE 从 CE 学到路由后，都放在相应 VRF 中，然后传递给对端 PE，再由 PE 转发到远程 CE。

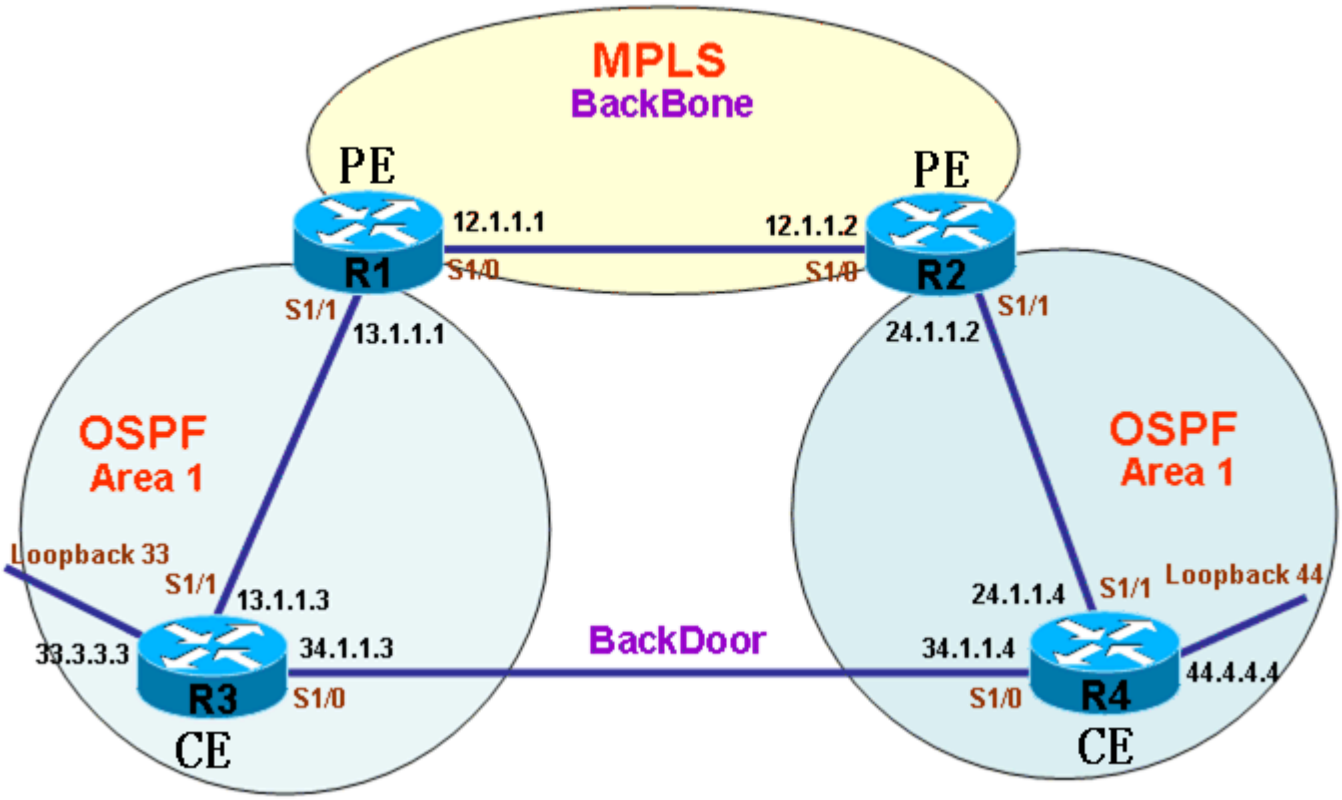
在 OSPF 中，当同时从 intra-area（同一个区域）和 inter-area(不同区域)学到相同路由时，intra-area 中的路由总是被优先选中。



在上图中，有多个 MPLS VPN 场点同时连接到 MPLS VPN 骨干网络，PE-CE 路由协议使用 OSPF，OSPF 区域分别有 0、1、2、3,当 MPLS VPN 场点之间互相再连接链路后，这样的链路被称为后门链路(BackDoor)，只要 OSPF 路由从这些后门链路上更新，那么势必会造成 PE 和 CE 路由器将 MPLS VPN 场点之间的流量优先选择从后门链路发送，而放弃从 MPLS VPN 骨干网络发送，原因为：

*MPLS VPN 骨干网络中为 BGP，且为 IBGP，管理距离为 200，高于 OSPF 的值。

*PE 和 CE 之间使用 OSPF，从同区域学习到的路由优先于其它区域路由。



根据上图中得出，当 CE 路由器 R3 从与 R4 的后门链路学习到 OSPF 路由 44.4.4.4 之后，R3 必定会优先选择从后门链路到达 44.4.4.4，因为从后门链路学习到的路由为同区域的，都为区域 1。不仅如此，PE 路由器 R1 也同样会选择从后门链路到达 44.4.4.4 而放弃从 MPLS VPN 骨干网络中传输。如果 C E 之间的后门链路仅仅是作为备份使用，那么将其选作主用链路，是不理智的。

如果要让 PE 路由器和 CE 路由器都优先选择从 MPLS VPN 骨干网络到达远程 VPN 场点，解决方法为，在 PE 路由器之间也创建一个 OSPF 区域，这样一来，从后门链路到达远程场点是 OSPF 路径，从 MPLS VPN 骨干网络中到达远程网络同样也是 OSPF 路径，所以就能轻松地控制 PE 和 CE 路由器到达远程场点的路径。

在 PE 之间创建额外的 OSPF 区域的方法是创建 OSPF Sham-Link，PE 之间的 Sham-Link 相当于一条逻辑的链路，但是这条链路也属于某个 OSPF 区域，供 PE 路由器选路使用。只要 PE 路由器想要从 MPLS VPN 骨干网络到达远程网络，就等于从 Sham-Link 上到达远程网络。

在 PE 上创建 OSPF Sham-Link 需要注意的是：

＊OSPF Sham-Link 也算是一条 OSPF 链路，这条链路可以指定为任何区域。

＊当 PE 到 CE 再从后门链路到远程网络都属于同一个 OSPF 区域，即学习到的路由为 intra-area 路由时，必须将 Sham-Link 也指定为同一区域，因为如果 Sham-Link 属于另一区域，那么就表示从 MPLS VPN 骨干网络到达远程网络是 inter-area 路由，而 inter-area 路由是不可能优先于 intra-area 路由的，所以也就不可能让到达远程网络的路径从 Sham-Link 走。

＊各个 PE 到 CE，以及 Sham-Link 都可以属于不同的区域，即使这些链路属于不同的区域，也可以控制 PE 和 CE 的选路，选路规则就是 intra-area 路由和 inter-area 路由的区别。所以当出事各类区域时，请自己控制好选路。

配置 Sham-Link 时，需要满足以下条件：

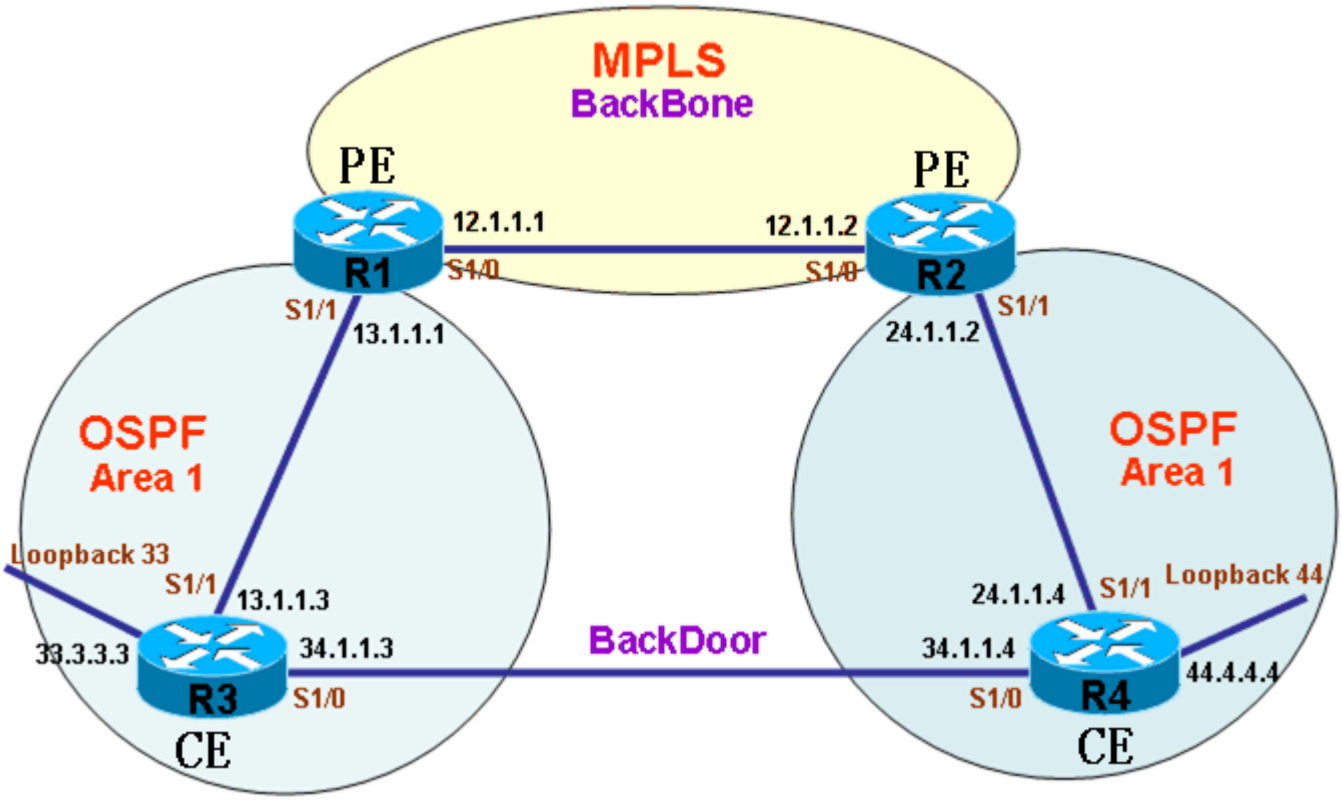
＊在 PE 上单独创建/32 位的地址，在 PE 之间使用这个地址来建立 Sham-Link。

＊这个 / 32 位地址的接口必须放入相应的 VRF。

＊这个 / 32 位地址必须在 BGP 里发布，而不能在 OSPF 里发布。

注：Sham-Link 是有 COST 值的，PE 穿越 MPLS VPN 骨干网络的 OSPF COST 值，就是 Sham-Link 的 COST 值，如果没有后门链路，Sham-Link 就不需要创建了.。

配置 OSPF Sham-Link



说明:在上图中，CE 路由器 R3 和 R4 上分别有地址 33.3.3.3 和 44.4.4.4，配置 Sham-Link 使 PE 路由器到达远程场点地址从 MPLS VPN 骨干网络中传输。

注:Sham-Link 重要配置为第 7 步

1. 配置 R1 与 R2 的路由协议为 RIP

(1)在 R1 上配置 RIP

r1(config)#router rip

r1(config-router)#ver 2

r1(config-router)#no au

r1(config-router)#network 1.0.0.0 (loopback0)

r1(config-router)#network 12.0.0.0

(2)在 R1 上配置 RIP

r2(config)#router rip

r2(config-router)#ver 2

r2(config-router)#no au

r2(config-router)#network 2.0.0.0 (loopback0)

r2(config-router)#network 12.0.0.0

2. 在 R1 与 R2 上配置 MPLS

(1)在 R1 上配置 MPLS

r1(config)#int s1/0

r1(config-if)#mpls ip

r1(config)#mpls ldp router-id loopback 0 force

(2)在 R2 上配置 MPLS

r2(config)#int s1/0

r2(config-if)#mpls ip

r2(config)#mpls ld router-id loopback 0 force

3. 配置 MPLS VPN

(1)在 R1 上配置 MPLS VPN

r1(config)#ip vrf vpn

r1(config-vrf)#rd 100:1

r1(config-vrf)#route-target both 100:1

r1(config)#int s1/1

r1(config-if)#ip vrf forwarding vpn

(2)在 R2 上配置 MPLS VPN

r2(config)#ip vrf vpn

r2(config-vrf)#rd 100:1

r2(config-vrf)#route-target both 100:1

r2(config)#int s1/1

r2(config-if)#ip vrf forwarding vpn

4．配置 OSPF

(1)在 R1 上配置 OSPF

r1(config)#router ospf 2 vrf vpn

r1(config-router)#router-id 1.1.1.1

r1(config-router)#network 13.1.1.1 0.0.0.0 a 1

(2)在 R2 上配置 OSPF

r2(config)#router ospf 2 vrf vpn

r2(config-router)#router-id 2.2.2.2

r2(config-router)#network 24.1.1.2 0.0.0.0 a 1

(3)在 R3 上配置 OSPF

r3(config)#router os 2

r3(config-router)#router-id 3.3.3.3

r3(config-router)#network 13.1.1.3 0.0.0.0 a 1

r3(config-router)#network 34.1.1.3 0.0.0.0 a 1

r3(config-router)#network 33.3.3.3 0.0.0.0 a 1

(4)在 R4 上配置 OSPF

r4(config)#router os 2

r4(config-router)#router-id 4.4.4.4

r4(config-router)#network 24.1.1.4 0.0.0.0 a 1

r4(config-router)#network 34.1.1.4 0.0.0.0 a 1

r4(config-router)#network 44.4.4.4 0.0.0.0 a 1

5．配置 MP-BGP

(1)在 R1 上配置 MP-BGP

r1(config)#router bgp 100

r1(config-router)#no au

r1(config-router)#no sy

r1(config-router)#bg router-id 1.1.1.1

r1(config-router)#neighbor 2.2.2.2 remote-as 100

r1(config-router)#neighbor 2.2.2.2 up loopback 0

r1(config-router)#address-family vpnv4

r1(config-router-af)#neighbor 2.2.2.2 activate

r1(config-router-af)#neighbor 2.2.2.2 send-community both

r1(config-router-af)#exit

r1(config-router)#address-family ipv4 vrf vpn

r1(config-router-af)#redistribute ospf 2

r1(config)#router os 2 vrf vpn

r1(config-router)#re bgp 100 subnets

(2)在 R2 上配置 MP-BGP

r2(config)#router bgp 100

r2(config-router)#no au

r2(config-router)#no sy

r2(config-router)#bg router-id 2.2.2.2

r2(config-router)#nei 1.1.1.1 remote 100

r2(config-router)#neighbor 1.1.1.1 up loopback 0

r2(config-router)#address-family vpnv4

r2(config-router-af)#neighbor 1.1.1.1 activate

r2(config-router-af)#neighbor 1.1.1.1 send-community both

r2(config-router-af)#exit

r2(config-router)#address-family ipv4 vrf vpn

r2(config-router-af)#re ospf 2

r2(config)#router ospf 2 vrf vpn

r2(config-router)#re bgp 100 subnets

6.查看路由

(1)查看 PE 路由器 R1 的路由

r1#sh ip bgp vpnv4 all

BGP table version is 11, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 100:1 (default for vrf vpn)					
* i13.1.1.0/24	2.2.2.2	192	100	0	?
*>	0.0.0.0	0		32768	?
* i24.1.1.0/24	2.2.2.2	0	100	0	?
*>	13.1.1.3	192		32768	?
*** i33.3.3.3/32**	**2.2.2.2**	**129**	**100**	**0**	**?**

***> 13.1.1.3 65 32768 ?**

* i34.1.1.0/24 2.2.2.2 128 100 0 ?

*> 13.1.1.3 128 32768 ?

*** i44.4.4.4/32 2.2.2.2 65 100 0 ?**

***> 13.1.1.3 129 32768 ?**

r1#

说明:PE 路由器 R1 到达 33.3.3.3 和 44.4.4.4 都从 CE 路由器 R3 走。

(2)查看 PE 路由器 R1 的路由

```
r1#sh ip route vrf vpn


Routing Table: vpn
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set
```

```
     34.0.0.0/24 is subnetted, 1 subnets
O       34.1.1.0 [110/128] via 13.1.1.3, 00:01:38, Serial1/1
     33.0.0.0/32 is subnetted, 1 subnets
O       33.3.3.3 [110/65] via 13.1.1.3, 00:01:39, Serial1/1
     24.0.0.0/24 is subnetted, 1 subnets
O       24.1.1.0 [110/192] via 13.1.1.3, 00:01:39, Serial1/1
     13.0.0.0/24 is subnetted, 1 subnets
C       13.1.1.0 is directly connected, Serial1/1
     44.0.0.0/32 is subnetted, 1 subnets
O       44.4.4.4 [110/129] via 13.1.1.3, 00:01:39, Serial1/1
r1#
```

说明:PE 路由器 R1 到达 33.3.3.3 和 44.4.4.4 都从 CE 路由器 R3 走。

7.在 PE 路由器之间创建 Sham-Link

(1)在 PE 路由器上创建/32 位 loopback 地址

R1:

```
r1(config)#int loopback 100
r1(config-if)#ip vrf forwarding vpn
r1(config-if)#ip add 100.1.1.1 255.255.255.255
```

说明:创建的/32 位地址必须放入 VRF。

R2:

```
r2(config)#int loopback 100
```

```
r2(config-if)#ip vrf forwarding vpn

r2(config-if)#ip add 100.1.1.2 255.255.255.255
```

说明:创建的/32 位地址必须放入 VRF。

(2)将/32 位地址在 MP-BGP 里发布

R1:

```
r1(config)#router bg 100

r1(config-router)#address-family ipv4 vrf vpn

r1(config-router-af)#network 100.1.1.1 mask 255.255.255.255
```

R2:

```
r2(config)#router bg 100

r2(config-router)#address-family ipv4 vrf vpn

r2(config-router-af)#network 100.1.1.2 mask 255.255.255.255
```

(3) 创建 Sham-Link

R1:

```
r1(config)#router ospf 2 vrf vpn

r1(config-router)#area 1 sham-link 100.1.1.1 100.1.1.2 cost 10
```

说明:创建 Sham-Link 时，要指定源地址和目的地址，并且指明 COST 值。

R2:

```
r2(config)#router ospf 2 vrf vpn

r2(config-router)#area 1 sham-link 100.1.1.2 100.1.1.1 cost 10
```

说明:创建 Sham-Link 时，要指定源地址和目的地址，并且指明 COST 值。

8. 查看结果

(1)在 PE 路由器 R1 上查看 Sham-Link

```
r1#sh ip ospf sham-links

Sham Link OSPF_SL0 to address 100.1.1.2 is up

Area 1 source address 100.1.1.1

  Run as demand circuit

  DoNotAge LSA allowed. Cost of using 10 State POINT_TO_POINT,

  Timer intervals configured, Hello 10, Dead 40, Wait 40,

    Hello due in 00:00:04

    Adjacency State FULL (Hello suppressed)

    Index 2/2, retransmission queue length 0, number of retransmission 0

    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)

    Last retransmission scan length is 0, maximum is 0

    Last retransmission scan time is 0 msec, maximum is 0 msec

r1#
```

说明: Sham-Link 建立成功。

(2)在 PE 路由器 R1 上查看 MP-BGP 的路由

```
r1#show ip bgp vpnv4 all

BGP table version is 21, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (default for vrf vpn)
*> 13.1.1.0/24      0.0.0.0                  0         32768 ?
r>i24.1.1.0/24      2.2.2.2                  0    100      0 ?
*> 33.3.3.3/32      13.1.1.3                65         32768 ?
* i34.1.1.0/24      2.2.2.2                128    100      0 ?
*>                  13.1.1.3                128         32768 ?
r>i44.4.4.4/32      2.2.2.2                  65    100      0 ?
*> 100.1.1.1/32     0.0.0.0                  0         32768 i
*>i100.1.1.2/32     2.2.2.2                  0    100      0 i
r1#
```

说明:到达远程场点 44.4.4.4 的路径选择从 MPLS VPN 骨干网络中走。

(3)在 PE 路由器 R1 上查看 VRF 的路由

```
r1#sh ip route vrf vpn
Routing Table: vpn
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
```

```
        ia - IS-IS inter area, * - candidate default, U - per-user static
route

        o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     34.0.0.0/24 is subnetted, 1 subnets

O       34.1.1.0 [110/128] via 13.1.1.3, 00:03:02, Serial1/1

     100.0.0.0/32 is subnetted, 2 subnets

C       100.1.1.1 is directly connected, Loopback100

B       100.1.1.2 [200/0] via 2.2.2.2, 00:03:19

     33.0.0.0/32 is subnetted, 1 subnets

O       33.3.3.3 [110/65] via 13.1.1.3, 00:03:02, Serial1/1

     24.0.0.0/24 is subnetted, 1 subnets

O       24.1.1.0 [110/74] via 2.2.2.2, 00:03:03

     13.0.0.0/24 is subnetted, 1 subnets

C       13.1.1.0 is directly connected, Serial1/1

     44.0.0.0/32 is subnetted, 1 subnets

O       44.4.4.4 [110/75] via 2.2.2.2, 00:03:04

r1#
```

说明:到达远程场点 44.4.4.4 的路径选择从 MPLS VPN 骨干网络中走。

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

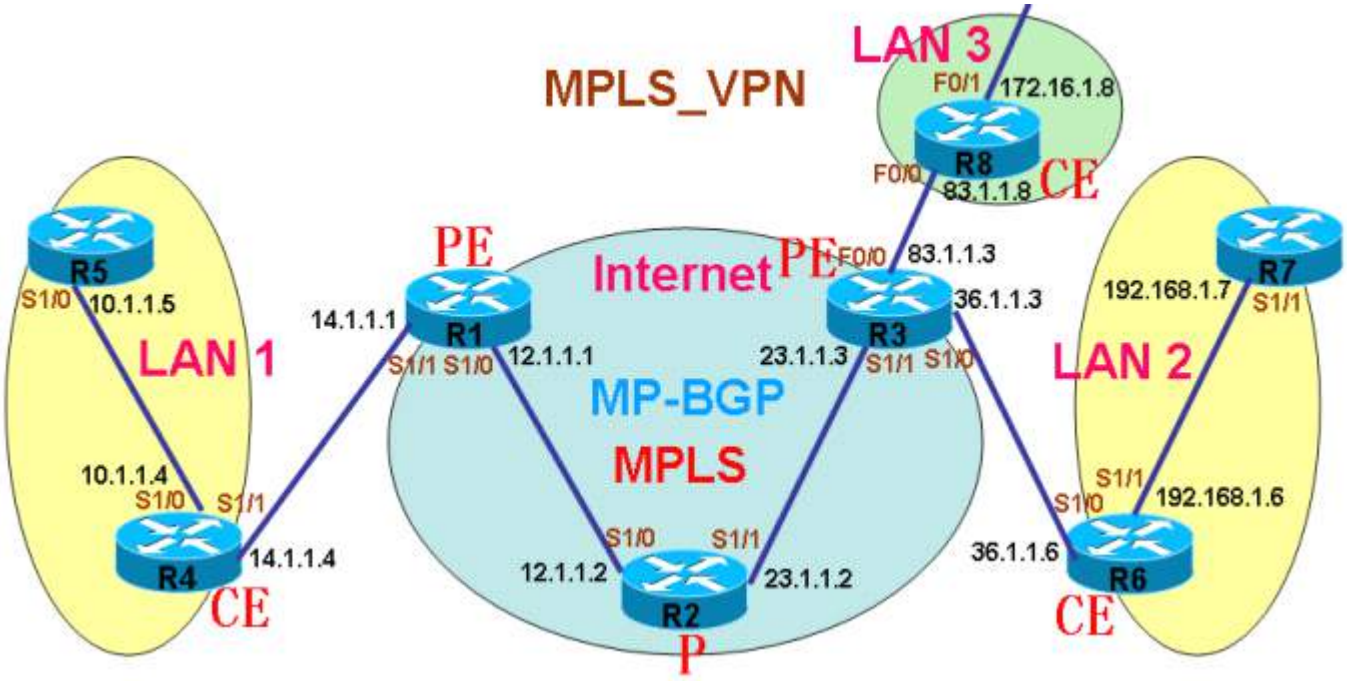
注：因为 sham-link 是要 COST 值的，相当于物理接口，要调整 CE 的选路，请调整 sham-link 和各接口 COST 值来完成。

外部通信

概述

以上是同一 VPN 的两个场点之间的通信，因为这两个场点之间 RD 是一样的，属于同一 VPN，所以这样的通信很好实现，被称为内部通信，但是如果不同 VPN 之间，即不同 RD 的 VPN 之间要通信，就要靠配置更多的 RT 来实现，这样的通信称为外部通信。

以下图为例，在原有两个 LAN 的基础上，加入 LAN 3，原有两个 LAN 之间的 RD 为 100：1，所以通信没有障碍，但是 LAN 3 的 RD 为 100：2，所以需要配置 RT 允许双方路由的导入导出，才能实现不同 LAN 的通信。



1.在 PE 为 LAN 上创建 VRF

（1） 在 PE R3 上创建 VRF，并配置不同的 RD，为 100：2

r3(config)#ip vrf vpn2

r3(config-vrf)#rd 100:2

r3(config-vrf)#route-target both 100:2

r3(config-vrf)#exit

r3(config)#int f0/0

r3(config-if)#ip vrf forwarding vpn2

% Interface FastEthernet0/0 IP address 83.1.1.3 removed due to enabling VRF vpn2

r3(config-if)#ip add 83.1.1.3 255.255.255.0

2.配置 PE-CE 路由协议

说明:配置 PE 连接 LAN 3 的路由协议，在配置 PE-CE 路由协议时，因为有多个 IGP 协议可供选择，之前我们已经举例过 RIP 和 OSPF 的协议，下面我们再使用 EIGRP 的 PE-CE 协议。

（1） 在 PE R3 上配置 EIGRP:

```
r3(config)#router eigrp 1

r3(config-router)#no auto-summary

r3(config-router)#address-family ipv4 vrf vpn2

r3(config-router-af)#no auto-summary

r3(config-router-af)#network 83.1.1.3 0.0.0.0

r3(config-router-af)#autonomous-system 1

r3(config-router-af)#redistribute bgp 100 metric 10000 100 255 1 1500
```

说明:在 PE 上配置 EIGRP 时，需要在 address-family 里面再次指定对方邻居的 AS 号，此 AS 号为对方正确配置的 AS 号码，并且将 MP-BGP 的路由重分布进 EIGR。

3.配置 EIGRP 重分布进 BGP

（1）在 R3 上配置 EIGRP 重分布进 BGP:

```
r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn2
```

r3(config-router-af)#redistribute eigrp 1

4.查看 MP-BGP 中的 VRF 路由表

r3#sh ip bgp vpnv4 all

BGP table version is 25, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 100:1 (default for vrf vpn1)					
*>i10.1.1.0/24	1.1.1.1	1	100	0	?
*>i14.1.1.0/24	1.1.1.1	0	100	0	?
*> 36.1.1.0/24	0.0.0.0	0		32768	?
*> 192.168.1.6/32	36.1.1.6	65		32768	?
Route Distinguisher: 100:2 (default for vrf vpn2)					
*> 83.1.1.0/24	0.0.0.0	0		32768	?
*> 172.16.1.0/24	83.1.1.8	409600		32768	?

r3#

说明:从 VRF 表中可以看出，LAN 3 的 VRF 和其它两个 LAN 的 VRF 并不同步，所以不可能实现通信。

5.配置 RT 允许双方 VRF 进入

说明:因为双方 VRF 的允许的 RD 不一样，所以路由表无法彼此进入，因此可以配置 RT 允许相互进出。

（1） 在 PE R3 上配置 RT 同时允许双方 RD：

r3(config)#ip vrf vpn1

r3(config-vrf)#route-target both 100:2

r3(config)#ip vrf vpn2

r3(config-vrf)#route-target both 100:1

（2）在 PE R1 上配置 RT 同时允许双方 RD:

r1(config)#ip vrf vpn1

r1(config-vrf)#route-target both 100:2

6.查看双方 VRF 路由表

说明:因为 RT 已经允许双方路由进出相互的 VRF，所以双方 VRF 路由表中应该有相互的路由信息。

（1） 在 R3 上查看 MP-BGP VRF 路由表：

r3#sh ip bgp vpnv4 all

BGP table version is 37, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 100:1 (default for vrf vpn1)					
*>i10.1.1.0/24	1.1.1.1	1	100	0	?
*>i14.1.1.0/24	1.1.1.1	0	100	0	?
*> 36.1.1.0/24	0.0.0.0	0		32768	?
*> 83.1.1.0/24	0.0.0.0	0		32768	?
*> 172.16.1.0/24	83.1.1.8	409600		32768	?
*> 192.168.1.6/32	36.1.1.6	65		32768	?
Route Distinguisher: 100:2 (default for vrf vpn2)					
*>i10.1.1.0/24	1.1.1.1	1	100	0	?
*>i14.1.1.0/24	1.1.1.1	0	100	0	?
*> 36.1.1.0/24	0.0.0.0	0		32768	?
*> 83.1.1.0/24	0.0.0.0	0		32768	?
*> 172.16.1.0/24	83.1.1.8	409600		32768	?
*> 192.168.1.6/32	36.1.1.6	65		32768	?

r3#

说明:双方的 VRF 已经同步。

（2）在 CE R4 上查看路由表:

```
r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

       E1 - OSPF external type 1, E2 - OSPF external type 2

       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

       ia - IS-IS inter area, * - candidate default, U - per-user static route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     83.0.0.0/24 is subnetted, 1 subnets

R       83.1.1.0 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

     36.0.0.0/24 is subnetted, 1 subnets

R       36.1.1.0 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

     172.16.0.0/24 is subnetted, 1 subnets

R       172.16.1.0 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

     10.0.0.0/24 is subnetted, 1 subnets

C       10.1.1.0 is directly connected, Serial1/0

     192.168.1.0/32 is subnetted, 1 subnets

R       192.168.1.6 [120/1] via 14.1.1.1, 00:00:19, Serial1/1

     14.0.0.0/24 is subnetted, 1 subnets
```

C 14.1.1.0 is directly connected, Serial1/1

r4#

说明:双方的 VRF 已经同步。

7.测试两个 LAN 的连通性

说明:因为双方的 VRF 路由表已经相同，所以应该可以通信

（1）在 CE 路由器 R5 上测试到 LAN3 172.16.1.8 的连通性：

R5ping

r5#ping 172.16.1.8

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.8, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 188/241/340 ms

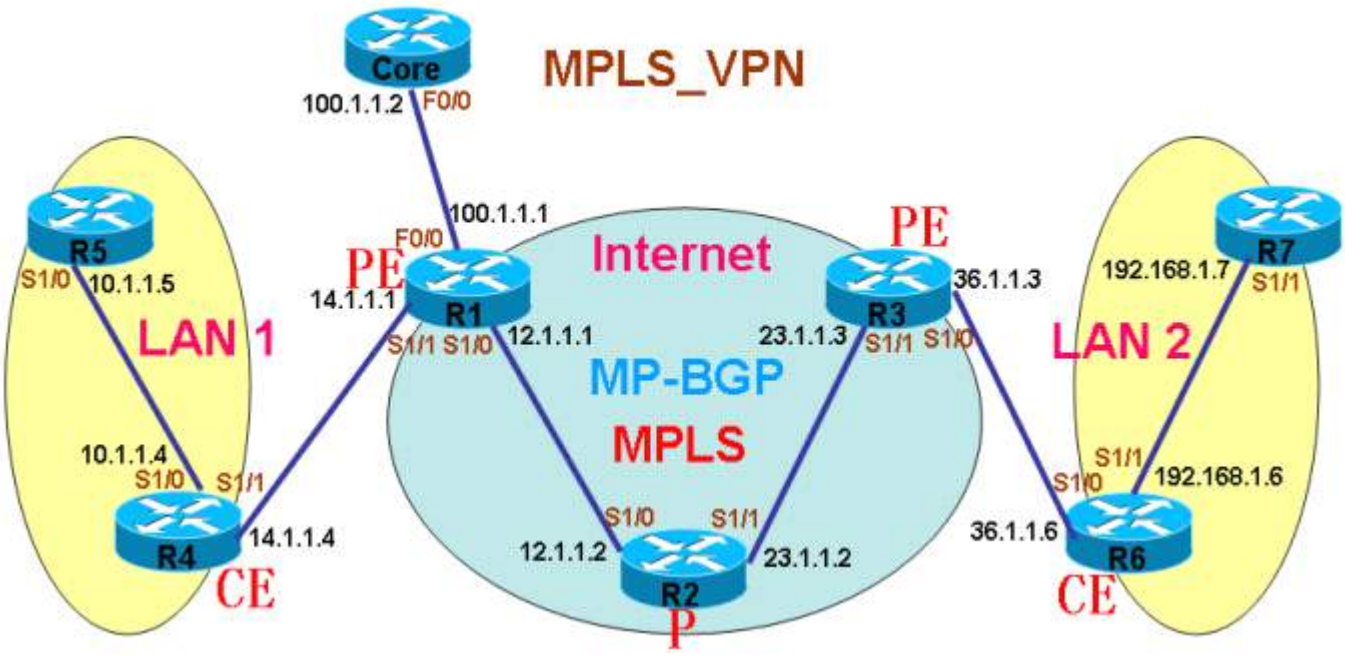
r5#

说明:成功实现不同 LAN 之间的外部通信。

点此查看配置实例所有设备的 show running-config

CE 接入英特网

概述



如上图所示，PE 路由器 R1 与 Internet 核心路由器 Core 相连，因为 PE 路由器 R1 连 CE 路由器 R4 的接口已经被划入 VRF vpn1 中，所以从 CE R4 过来的数据包，PE 都是根据 VRF vpn1 的路由表来作出转发决定的，当 CE 要将数据包发往 Internet 的时候，PE 是不会转发的，因为在 VRF vpn1 中，除了对方 LAN 的路由信息之外，并无到 Internet 核心的路由，所以要让 CE 到 Internet 核心的数据包被 PE 转发，那么就必须在 PE 上为 VRF vpn1 配置到 Internet 的路由，这样，CE 才能直接访问 Internet。

除了在 VRF 中静态为 CE 添加路由之外，还有一种方法就是，在 PE 和 CE 之间额外创建 tunnel，当 CE 到 Internet 的数据包发往 Tunnel 时，PE 就将该数据包发往 Internet。

最后一种让 CE 能够访问 Internet 的方法就是，CE 将自己的路由发往对端 LAN，让对端的 LAN 作转发处理。

配置

1.配置 VRF 静态路由

说明:直接为用户的 VRF 写到 Internet 的默认路由，让 CE 的 VRF 拥有到 Internet 的默认路由之后，就可以访问 Internet。

（1）在 PE 上为 VRF 配置默认路由:

r1(config)#ip route vrf vpn1 0.0.0.0 0.0.0.0 100.1.1.2 global

说明:为 VRF vpn1 添加默认路由指向下一跳 100.1.1.2

（2）在 PE 上还需手工指定到 CE 内部网络的路由:

r1(config)#ip route 10.1.1.0 255.255.255.0 serial 1/1 14.1.1.4

（3）最后 CE 上也要写默认路由到 PE:

R4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.1

（4）测试 CE 到 Internet 核心的连通性：

r5#ping 100.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 100.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 188/241/340 ms

r5#

说明:CE 到 Internet 核心的通信正常。

2.在 PE-CE 间配置 Tunnel

（1）在 PE 上配置到 CE 和 Tunnel:

r1(config)#int tunnel 0

r1(config-if)#ip address 50.1.1.1 255.255.255.0

r1(config-if)#tunnel source 14.1.1.1

r1(config-if)#tunnel destination 14.1.1.4

r1(config-if)#tunnel vrf vpn1

r1(config-if)#exit

说明:配置了到 CE 的 Tunnel 之后，还必须将该 Tunnel 放到 VRF 中。

（2）配置到 CE 内部网络的路由都走 tunnel:

r1(config)#ip route 10.1.1.0 255.255.255.0 tunnel 0

（3）在 CE 上配置到 PE 的 tunnel:

r4(config)#int tunnel 0

r4(config-if)#ip address 50.1.1.4 255.255.255.0

r4(config-if)#tunnel source 14.1.1.4

r4(config-if)#tunnel destination 14.1.1.1

r4(config-if)#exit

(4)指定 CE 所有的路由都从 Tunnel 发给 PE：

r4(config)#ip route 0.0.0.0 0.0.0.0 tunnel 0

（5）测试 CE 到 Internet 核心的连通性：

r5#ping 100.1.1.2

第三种方法将所有数据发到自己核心总部再转出去，略。

更多 PE-CE 路由协议

说明:之前说过 PE-CE 的路由协议有多种，我们已经举例过 RIP，OSPF，EIGRP，下面来举例静态写 VRF，还有就是 EBGP。

1.静态写 VRF 路由

说明:PE 的 VRF 中如果没有到 CE 的内部路由，将无法为不同 LAN 之间提供数据包转发，所以 PE 必须获得 CE 的内部网络路由信息，可以使用动态路由协议，也可以静态写 VRF 路由。

（1）在 PE R1 上写静态的 VRF 路由指向 CE：

r1(config)#ip route vrf vpn1 10.1.1.0 255.255.255.0 14.1.1.4

（2）在 PE R3 上写静态的 VRF 路由指向 CE：

r3(config)#ip route vrf vpn1 192.168.1.0 255.255.255.0 36.1.1.6

（3）在 PE R1 上将静态路由重分布进 MP-BGP:

说明:PE 上的 VRF 中已经拥有到 CE 内部网络的路由后，如果这个信息不发给远程 LAN，那么双方 LAN 之间也不能通信，所以可以将此静态路由重分布进 MP-BGP，从而发给远程 LAN。

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute static

r1(config-router-af)#exi

（4）在 PE R3 上将静态路由重分布进 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#redistribute static

(5)CE R4 也要写默认路由指向 PE：

r4(config)#ip route 0.0.0.0 0.0.0.0 14.1.1.1

（6）CE R6 也要写默认路由指向 PE：

r6(config)#ip route 0.0.0.0 0.0.0.0 36.1.1.3

(7)测试双方 LAN 的连通性：

r5#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 148/236/320 ms

r5#

说明:通过 PE 和 CE 之间写静态路由，双方 LAN 之间通信正常。

2.PE-CE 之间运行 EBGP

说明:PE 和 CE 之间只能运行 EBGP，因为当 PE 学到 CE 的路由后，要发给远程 LAN，所以此路由需要导入 MP-BGP，但是这种重分布在 EBGP 和 MP-BGP 之间会自动执行，无需额外配置。

（1）在 PE R1 上配置到 CE 的 EBGP：

r1(config)#router bg 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#neighbor 14.1.1.4 remote-as 200

r1(config-router-af)#neighbor 14.1.1.4 activate

（2）在 PE R3 上配置到 CE 的 EBGP：

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#neighbor 36.1.1.6 remote-as 300

r3(config-router-af)#neighbor 36.1.1.6 activate

r3(config-router-af)#

（3）在 CE R4 上配置到 PE 的 EBGP：

r4(config)#router bgp 200

r4(config-router)#neighbor 14.1.1.1 remote

r4(config-router)#neighbor 14.1.1.1 remote-as 100

r4(config-router)#network 10.1.1.0 mask 255.255.255.0

（4）在 CE R6 上配置到 PE 的 EBGP：

r6(config)#router bgp 200

r6(config-router)#neighbor 36.1.1.3 remote-as 100

r6(config-router)#network 192.168.1.0 mask 255.255.255.0

（5）解决 BGP 路由问题:

说明:因为在 LAN1 和 LAN2 之间配置的 BGP AS 号码都为 200，而本端 BGP 在收到路由时，如果发现路由携带的 AS 和自己的 AS 相同，则丢弃该路由，所以双方的 CE 都不会有对方的路由，所以这个问题要在 PE 的 BGP 上配置允许 AS 重叠。

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#neighbor 14.1.1.4 as-override

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#neighbor 36.1.1.6 as-override

（6）测试连通性：

r5#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 148/236/320 ms

r5#

说明:通过 PE 上额外的配置，CE 之间的 BGP AS 号码重叠问题被解决，双方 LAN 之间实现通信。

Multi-VRF CE / VRF-Lite

概述

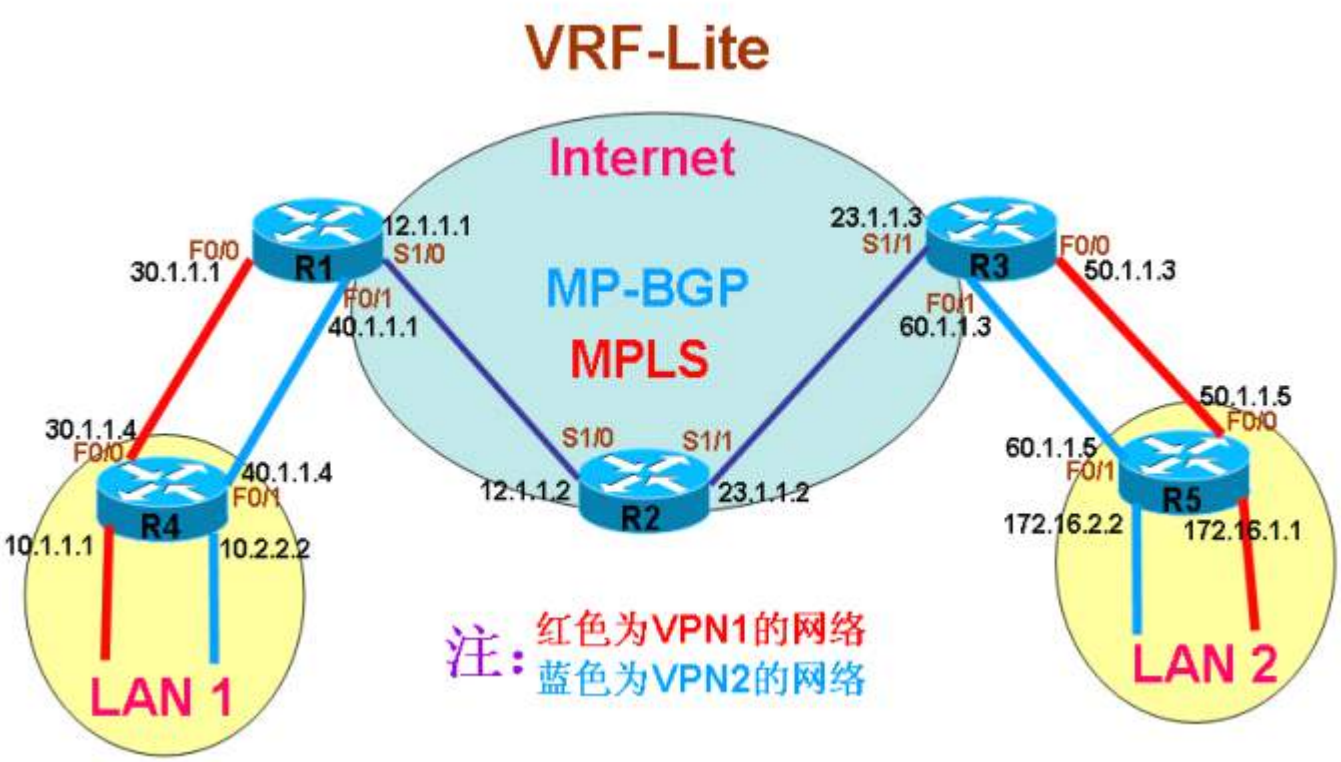
当在 PE 上将连接 CE 的接口放入 VRF 表之后，那么从此接口过来的所有 CE 数据都属于该 VRF，都根据该 VRF 来做出转发决定，所以从同一个接口发来的用户数据，都执行相同的路由查询。 如果一个很大的公司，这个公司有两个分公司通过 Internet 相连,并且每个分公司都存在多个部门，那么当在正常实施 MPLS_VPN 时，要么让两个公公司的所有部门都能通信，因为他们走的是同一个 VRF 路由表，要么都不能通信。

当某些时候因为业务要求，如果要让两个分公司的不同部门之间不能通信，按照 MPLS_VPN 的理论，就应该为要通信的部门单独创建 VRF，再为不要通信的部门之间单独创建 VRF，只有这样，才能将不同部门的数据隔离开来。

正因为 CE 连到 PE 的接口所有数据都属于同一个 VRF，所以当 CE 和 PE 之间只有一根链路相连时，要创建多个 VRF，这是不能实现的。但是，如果 PE 和 CE 之间是以太网链路，是可以支持子接口划分的，也就是可以通过为子接口划分不同的 VLAN 来实现。除了子接口之外，最好的办法就是在 PE 和 CE 间拉多条线路。

如果只通过在 PE 上将不同链路划入不同的 VRF，也可以达到隔离部门的需求。但是还有个方法就是，还可以直接在 CE 上面就将连接不同部门的接口直接划入不同的 VRF。因为在平时，要将接口划入 VRF，是在 PE 上做的，而 CE 是不了解 VRF 的，也就是说 MPLS_VPN 对于 CE 端来说是透明的，那么现在要让 CE 也能够认识 VRF，能够成功为不同接口划不同的 VRF，这就需要扩展 CE 具有 PE 的功能，这种功能被称为 Multi-VRF CE 或 VRF-Lite，在 CE 上为不同接口划 VRF 的同时，PE 上的 VRF 将继续保留不可删除，所以 PE 的功能将不作变动。而在 CE 将路由传递给 PE 时，这中间运行的协议是 OSPF，并且还要扩展 OSPF 的 vrf-lite 功能。

下面以下图为例，来详细讲解 VRF-Lite 的配置



说明:红色部分网络为需要通信的部门，但和蓝色网络的部门作隔离，其中 R1、R2、R3 上均已配置 loopback0，地址分别为 1.1.1.1/32，2.2.2.2/32，3.3.3.3/32，并且已经配置好 OSPF，让 MPLS 区域内所有直连接口和 loopback 口互通。下面以配置红色网络部门通信为例，蓝色网络部门通信与红色部门相似，请自行参考配置。

1.将 MPLS 区域网络配通

说明:此步略过，详细步骤请参见之前部分。

2.配置 MP-BGP

说明:配置 R1 和 R3 之间的 MP-BGP

（1） 在 R1 上配置 MP-BGP:

r1(config)#router bgp 100

r1(config-router)#neighbor 3.3.3.3 remote-as 100

r1(config-router)#neighbor 3.3.3.3 update-source loopback 0

r1(config-router)#address-family vpnv4

r1(config-router-af)#neighbor 3.3.3.3 activate

r1(config-router-af)#neighbor 3.3.3.3 send-community both

（2）在 R1 上配置 MP-BGP:

r3(config)#router bgp 100

r3(config-router)#neighbor 1.1.1.1 remote-as 100

r3(config-router)#neighbor 1.1.1.1 update-source loopback 0

r3(config-router)#address-family vpnv4

r3(config-router-af)#neighbor 1.1.1.1 activate

r3(config-router-af)#neighbor 1.1.1.1 send-community both

3.在 CE 上为不同部门创建不同 VRF

说明:在 CE 上为不同部门创建不同 VRF，并配置好 RD 和 RT

（1）在 R4 上为相应接口创建 VRF：

r4(config)#ip vrf vpn1

r4(config-vrf)#rd 100:1

r4(config-vrf)#route-target both 100:1

（2）在 R5 上为相应接口创建 VRF：

r5(config)#ip vrf vpn1

r5(config-vrf)#rd 100:1

r5(config-vrf)#route-target both 100:1

4.将相应部门的接口划入相应 VRF

（1） 在 R4 上将相应接口划入相应 VRF:

r4(config)#int loopback 10

r4(config-if)#ip vrf forwarding vpn1

r4(config-if)#ip add 10.1.1.1 255.255.255.0

r4(config)#int f0/0

r4(config-if)#ip vrf forwarding vpn1

r4(config-if)#ip add 30.1.1.4 255.255.255.0

（2）在 R5 上将相应接口划入相应 VRF:

r5(config)#int loopback 172

r5(config-if)#ip vrf forwarding vpn1

r5(config-if)#ip add 172.16.1.1 255.255.255.0

r5(config)#int f0/0

r5(config-if)#ip vrf forwarding vpn1

r5(config-if)#ip add 50.1.1.5 255.255.255.0

5.配置 PE-CE 间的 OSPF

（1） 在 CE R4 上启动 OSPF，并将相应路由放进 OSPF 进程，传送给 PE:

r4(config)#router ospf 100 vrf vpn1

r4(config-router)#capability vrf-lite

r4(config-router)#network 10.1.1.1 0.0.0.0 a 0

r4(config-router)#network 30.1.1.4 0.0.0.0 a 0

注：命令 capability vrf-lite 为扩展 OSPF，必须输入，否则 OSPF 无法接收路由。

（2）在 CE R5 上启动 OSPF，并将相应路由放进 OSPF 进程，传送给 PE:

r5(config)#router ospf 100 vrf vpn1

r5(config-router)#capability vrf-lite

r5(config-router)#network 172.16.1.1 0.0.0.0 a 0

r5(config-router)#network 50.1.1.5 0.0.0.0 a 0

注：命令 capability vrf-lite 为扩展 OSPF，必须输入，否则 OSPF 无法接收路由。

6.在 PE 上创建 VRF

（1）在 PE R1 上为 CE 的不同部门也创建不同的 VRF，并将连 CE 的相应线路划入相应 VRF:

r1(config)#ip vrf vpn1

r1(config-vrf)#rd 100:1

r1(config-vrf)#route-target both 100:1

r1(config)#int f0/0

r1(config-if)#ip vrf forwarding vpn1

r1(config-if)#ip add 30.1.1.1 255.255.255.0

(2)在 PE R3 上为 CE 的不同部门也创建不同的 VRF，并将连 CE 的相应线路划入相应 VRF:

r3(config)#ip vrf vpn1

r3(config-vrf)#rd 100:1

r3(config-vrf)#route-target both 100:1

r3(config)#int f0/0

r3(config-if)#ip vrf forwarding vpn1

r3(config-if)#ip add 50.1.1.3 255.255.255.0

7.在 PE 上启动 OSPF

（1）在 PE R1 上也启用 OSPF，用来接收 CE 发来的各部门的路由，以便让这些部门相通:

r1(config)#router ospf 100 vrf vpn1

r1(config-router)#network 30.1.1.1 0.0.0.0 a 0

(2)在 PE R1 上也启用 OSPF，用来接收 CE 发来的各部门的路由，以便让这些部门相通:

r3(config)#router ospf 100 vrf vpn1

r3(config-router)#network 50.1.1.3 0.0.0.0 a 0

8.在 MP-BGP 和 OSPF 间重分布

说明:将 MP-BGP 和 OSPF 相互重分布，最终让相应部门实现通信。

（1） 在 PE R1 上做 MP-BGP 和 OSPF 的双向重分布:

r1(config)#router ospf 100 vrf vpn1

r1(config-router)#redistribute bgp 100 subnets

r1(config)#router bgp 100

r1(config-router)#address-family ipv4 vrf vpn1

r1(config-router-af)#redistribute ospf 100 vrf vpn1

（2）在 PE R3 上做 MP-BGP 和 OSPF 的双向重分布:

r3(config)#router ospf 100 vrf vpn1

r3(config-router)#redistribute bgp 100 subnets

r3(config)#router bgp 100

r3(config-router)#address-family ipv4 vrf vpn1

r3(config-router-af)#redistribute ospf 100 vrf vpn1

9.查看 CE 各自 VRF 的路由

（1）在 CE R4 上查看 VRF vpn1 的路由：

r4#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

50.0.0.0/24 is subnetted, 1 subnets

O IA 50.1.1.0 [110/11] via 30.1.1.1, 00:06:00, FastEthernet0/0

172.16.0.0/24 is subnetted, 1 subnets

O IA 172.16.1.0 [110/21] via 30.1.1.1, 00:06:00, FastEthernet0/0

10.0.0.0/24 is subnetted, 1 subnets

C 10.1.1.0 is directly connected, Loopback10

30.0.0.0/24 is subnetted, 1 subnets

C 30.1.1.0 is directly connected, FastEthernet0/0

r4#

说明:CE R4 的 VRF vpn1 已经拥有双方部门的路由，应该可以实现部门间通信。

（2）在 CE R5 上查看 VRF vpn1 的路由:

r5#sh ip route vrf vpn1

Routing Table: vpn1

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

50.0.0.0/24 is subnetted, 1 subnets

C 50.1.1.0 is directly connected, FastEthernet0/0

172.16.0.0/24 is subnetted, 1 subnets

C 172.16.1.0 is directly connected, Loopback172

10.0.0.0/24 is subnetted, 1 subnets

O IA 10.1.1.0 [110/21] via 50.1.1.3, 00:05:07, FastEthernet0/0

30.0.0.0/24 is subnetted, 1 subnets

O IA 30.1.1.0 [110/11] via 50.1.1.3, 00:05:07, FastEthernet0/0

r5#

说明:CE R4 的 VRF vpn1 已经拥有双方部门的路由，应该可以实现部门间通信。

10.测试连通性

说明:双方 CE 在相应 VRF 都拥有双方部门的路由信息，因此可以实现部门间通信。

（1）在 CE R4 上 ping 远程部门地址做测试:

说明:在上图中，测试时，因为数据从 CE 发出，默认不是走 VRF 表，所以发送 ICMP 时，除了指定源地址外，还要指定数据走相应的 VRF 路由表，否则测试不正确。

r4#ping vrf vpn1 172.16.1.1 source loopback 10

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

Packet sent with a source address of 10.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 96/116/164 ms

r4#

说明:通过配置 VRF Lite，实现了远程公司之间的部门通信，而不在同一 VRF 的不

同部门，通信将被隔离。

点此查看配置实例所有设备的 show running-config

以上配置实例为红色网络部门间的通信，只要配置蓝色部门为不同的 VRF 和不同的 RD，通信将可实现隔离功能，蓝色网络的通信，请自行参照红色网络的配置。

Multicast

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

鸿鹄培训+考试群：301283867

概述

在当前的 IP 网络中，某台主机将数据包发向另一台主机时，就需要在数据包的目标 IP 位置写上那台主机的 IP 地址，再将数据包发出去，这个数据包发出去后，只有那台主机才能收到并且打开，而其它主机是不能收到和打开的。如果还想发送数据包给别的主机，就需要为数据包重新写上别的主机的 IP 地址，然后发出去。要将数据包发给几台主机，就需要为每个独立的数据包写上相应的目标 IP 地址。一个数据包包含一个特定的目标 IP 地址，并且这个数据包只能由相应的某台主机能够接收并且查看，这样的数据包称为单播（Unicast）。当要将同一份数据发送给多台主机时，如果使用单播的传送方式，那么需要发给几台主机，就需要重新封装几次数据包，并且将每份数据包单独发送给每台主机。

当使用广播来发送数据包时，目标 IP 为广播地址的数据将被网络中的每台主机接收并查看，但广播是不能被路由器转发的。

在网络上，当需要将一份同样的数据发送给多台主机时，如数字电视、视频会议等应用，这样的数据有多种传送方式，如下面两种：

1.单播

在使用单播的情况下，需要为每个接收者重复发送单播，如果接收者数目过多，那么数据源就需要多次发送而承受巨大的压大，并且低速的 WAN 链路也会成为潜在的瓶劲，如果数据对时延比较敏感，还会造成延迟。

2.广播

在使用广播的情况下，数据源只需要将同一份数据发送一次，但是负担却转移到了网络中的其他主机，因为不管想不想接收这个数据，都必须接收；并且广播是不可跨越路由器的，如果接收者在远程网络，将会造成数据丢失的情况。

3.组播

从上面的结论中可以看出，当需要将一份同样的数据发送给多台主机时，虽然使用单播可以跨越路由器，但是需要将同一份数据发送多次，不切实际；而使用广播只需要发送一次数据，但是却让网络中每个人都必须接收数据，并且数据不能穿越路由器，造成远程网络收不到数据，所以也不可行。考虑到这些因素，便开发出了一种新的数据传输方式，这样的传输方式结合了单播和广播的优势，即将一份数据发出去后，这样的数据可以同时被多台主机接收，并且数据可以穿越路由器，从而被路由到远程网络，这样的数据就是组播（Multicast），因此，组播数据发出去后，可以只被一组特定的主机接收，而不想接收的主机，是收不到的，组播还可以被路由器转发到远程网络，前提是路由器必须开启组播功能。在组播中，想要接收组播的主机，被称为组员，或组成员。

组播的三个组成部分

1.组播地址（能被组播识别的地址集）

2.组成员机制（主机加入和退出组的机制）

3.组播路由协议（路由器有效传送组播到各个网络的组成员，且不会过度消耗网络资源的路由协议）

当需要将一份同样的数据发送给多台主机时，在使用组播的情况下，就需要将需要接收数据包的主机标识出来，要区别于不接收的主机，只有想要接收的主机，才能收到相应的组播数据，这时就需要为组播数据包写上特定的 IP 地址，被写上组播地址的数据包，只能被特定的组成员接收，所以要将组播正常的发送到组成员，就必须为组播数据包写上组播地址，当网络中有多种组播数据时，每种数据应该写上不同的组播地址。

当写上了组播地址的数据包在网络中传送时，这样的数据包只应该被特定组的组成员接收，只有属于同一个组的成员，才能接收该组的数据包，所以必须确认哪些主机是组成员，哪些主机不是。要想接收组播，主机要做的事就是加入特定的组，特定的组，就是由组播地址来区分的。当主机加入了某个组之后，便能收到该组的数据，而当主机不想接收组播时，就应该退出相应的组，这样就可以停止组播的接收和转发，要完成这一切，就需要一种主机加入和退出组的机制，要区别哪些是组成员，哪些不是。

如果某个组的成员分布在不同的网络中，那么就需要路由器转发组播，才能保证远程主机能够收到组播数据。要让路由器为组播数据提供转发，就需要让路由器拥有像单播路由表一样的转发表，依据路由表来决定数据包该从哪个接口发出去。要让路由器依据路由表来转发组播，就需要有组播路由表，而路由器的组播路由器，就需要靠特定的组播路由协议来收集组播路由表，组播路由表指导路由器如何将组播正确转发到组成员。

下面分别来详细介绍组播地址、组成员机制以及组播路由协议的工作过程：

组播地址

组播 IP 地址

在我们区分一个 IP 地址是哪类地址时，只需要看第一个字节便能得出结果。

在 IP 地址一个字节的 8 个 bit 中，相应 bit 位为 1 时，便得到相应的值，每个 bit 位的取值分别如下：

128 64 32 16 8 4 2 1

A 类 IP 地址第一字节的第 1 个 bit 总是为 0,后面 7 个 bit 可以随意设置，所以 A 类 IP 地址第一个字节的取值范围为：

00000000 （最小时，值为 0）

01111111 （最大时，值为 127）

B 类 IP 地址第一字节的前面 2 个 bit 总是为 1 0,后面 6 个 bit 可以随意设置，所以 B 类 IP 地址第一个字节的取值范围为：

10000000 （最小时，值为 128）

10111111 （最大时，值为 191）

C 类 IP 地址第一字节的前面 3 个 bit 总是为 1 1 0,后面 5 个 bit 可以随意设置，所以 C 类 IP 地址第一个字节的取值范围为：

11000000 （最小时，值为 192）

11011111 （最大时，值为 223）

而 D 类 IP 地址第一字节的前面 4 个 bit 总是为 1 1 1 0,后面 4 个 bit 可以随意设置，所以 D 类 IP 地址第一个字节的取值范围为：

11100000 （最小时，值为 224）

11101111 （最大时，值为 239）

组播地址采用 **D 类** IP 地址表示，因为 IP 地址共 32 bit，D 类地址前 4 bit 总是为 1110,所以

组播地址取值范围为 224.0.0.0——239.255.255.255。

组播地址可以是永久的，也可以是临时的，“永久”是指一个组地址被永久分配给某个协议，临时的可以自由定义。

部分永久的组播地址：

224.0.0.1 子网中的所有系统和主机

224.0.0.2 子网中的所有路由器

224.0.0.4 DVMRP 路由器

224.0.0.5 所有 OSPF 路由器

224.0.0.6 OSPF 指定路由器

224.0.0.9 RIP-2 路由器

224.0.0.10 EIGRP 路由器

224.0.0.13 PIM 路由器

224.0.0.15 CBT 路由器

224.0.1.39 Cisco-RP-Announce

224.0.1.40 Cisco-RP-Discovery

组播地址除了分为永久地址和临时地址之外，还可以细分，我们需要知道的细分地址为：

保留给网络协议的地址范围：

224.0.0.0 ——224.0.0.255

全局地址范围：

224.0.1.0 ——238.255.255.255

限制私有地址范围：

239.0.0.0 ——239.255.255.255

组播二层地址：

一个 IP 数据包要在网络中传送，必须依照 OSI 七层模型由上至下封装，比如先封装 TCP 或 UDP 端口号，然后封装 IP 地址，如果是组播数据包，那么目标 IP 为组播地址，最后再封装数据链路层地址，如果介质是以太网，那么就需要封装 MAC 地址。

当一个数据包为组播数据时，那么这个数据包将被多台主机接收，所以数据包的 MAC 地址不能为某台主机的真实 MAC 地址，这时就需要根据组播 IP 地址，来封装一个拥有对应关系的组播 MAC 地址，因为这个组播 MAC 是与组播 IP 地址对应的，且不是主机的真实 MAC 地址，所以能够被多台主机接收到。

组播的 MAC 地址和组播 IP 地址拥有对应关系，也就是说组播 MAC 地址是根据组播 IP 地址计算得到的。一个 MAC 地址为 48 bit，使用十六进制表示，组播 MAC 地址的前面 24 bit 固定为 01 00 5E，第 25 bit 固定为 0，而剩下的 23 bit 则使用组播 IP 地址后 23 bit 填充。如下图表示：

从上图中可以看出，组播 MAC 地址前面 24 bit 固定为 01 00 5E，第 25 bit 固定为 0，而后面 23 bit 由组播 IP 地址的后面 23 bit 映射过来。

一个组播 IP 地址为 32 bit，而前面 4 个 bit 固定为 1110，后面 23 bit 被映射到了组播 MAC 地址，所以中间还有多余的 5 个 bit 可以随意设置，正因为这 5 bit 可以随意取值，最终造成有 2 的 5 次方共 32 个 IP 地址被映射到同一 MAC 地址。如下面出现的结果：

组播 IP 地址为：

224.0.0.1 后 23 bit 为 00 00 01

映射出组播 MAC 地址为 01 00 5E 00 00 01

225.0.0.1 后 23 bit 为 00 00 01

映射出组播 MAC 地址为 01 00 5E 00 00 01

226.0.0.1 后 23 bit 为 00 00 01

映射出组播 MAC 地址为 01 00 5E 00 00 01

等等 ••••••。

由上可以看出，将会有多个组播 IP 地址映射到同一个组播 MAC 地址，数量为 32 个，但是这并不会影响到组播的通信，因为即使组播的 MAC 地址相同，但是组播 IP 地址可能是不同的，所以数据量不会被混淆，即使组播 IP 地址是相同的，不同数据流也会使用不同的上层端口号，最终也能将不同数据流区分开来。

组成员机制

要将组播数据准确发送给组成员，必须先确定哪些网络的哪些主机是组成员，只有先确定了成员的位置，才能正确转发组播。当组成员不再需要接收组播的时候，就应该停止向成员发送组播，要确定组成员不再需要接收组播，就必须在成员退出时明确通告发送者。要确定组成员，有两种方式：查询和报告。

查询，就是一台路由器向网络中发出查询消息，查询是否要主机要加入组，如果有主机应答，那么路由器就可以请求上游路由器把组播流量前转到这个子网中，如果没有主机应答，则请求上游路由器停止向其前转组播流量。

报告，主机也可以不必等待路由器的查询，可以主动向路由器请求加入某个组，退出时也要向路由器发送退出消息，让路由器停止向其前转组播流量。

在网络中，要确定组成员，需要使用一种协议，这种协议就是 IGMP (Internet Group Management Protocol)因特网组管理协议，IGMP 运行在路由器和主机之间，因为当组播发送者和组成员在不同网络时，需要路由器为组播数据提供转发，那么路由器就必须确认自己直连的网络中是否存在组成员，可以使用查询和报告来发现组成员，IGMP 就可以完成这样的工作。

IGMP 共有 3 个版本，version 1，version 2，version 3，而思科路由器接口默认使用 version 2，下面分别来介绍 version 1 和 version 2，而 version 3 暂不介绍。

IGMP version 1

因为 IGMP 是运行在路由器和主机之间的，所以当 IGMP 分别运行在路由器和主机上时具有不同的功能。

IGMP v1 路由器：

运行了 IGMP 的路由器目的是确定哪些主机是组成员，主要靠发送 queries（查询）来确定，当路由器发送 queries 查询主机时，使用目的地址为 224.0.0.1，当有主机回复时，便认为网络中存在组成员，则将组播发送到该网络，并且 queries 每一分钟发送一次，主机每次都回复，如果连续三个 queries 没有回复，即三分钟，那么路由器便认为网络中的组成员已经离开，也就停止向网络中发送组播数据了。

IGMP v1 主机：

运行了 IGMP 的主机发送 reports 来告诉路由器自己是组成员，reports 同时也可以用来回复路由器的 queries。Reports 的目标地址是需要加入的组的地址，所以其它路由器和主机都能收到此数据包。一个网络中，因为有一个组成员时，路由器需要向该网络转发组播数据，组播是发送一次，有多个组成员时，路由器还是只向该网络发送一次组播数据，所以当一个网络中有多台主机需要成为组成员时，并不需要每台主机都向路由器发送 reports,一个网络中只需要一台主机向路由器发送 reports 即可，其它主机全部不发，因为一台发送了之后，其它要接收组播的主机也是能正常接收到的，也为了避免所有组员都发 reports 而充斥整个网络。

当网络中的组成员离开后，并没有特定的机制来通知路由器组员的离开，唯一的方法就是路由器在 3 分钟后没有收到组成员对 queries 的回复，也就认为网络中的组成员已经离开，便停止向网络中发送组播数据了。而其它主机还需要继续接收组播时，就需要再次向路由器发送 reports 要宣告自己的存在。

IGMP version 2

IGMP version 2 与 version 1 具有相同的功能，具有相同的数据包，但也有不同的数据包，下面分别来看 IGMP version 2 运行在路由器和主机上的数据包：

IGMP v2 主机：

IGMP 主机发送三种数据包来向 IGMP 路由器通告自己的状态：

1. Membership Report

2. Version 1 Membership Report

3. Leave Group

Membership Report

Membership Report 是主机发向路由器用来加入组的数据包，当路由器从接口上收到主机的 Membership Report，便认为该网络上有组成员，就开始将组播转发到该网络。主机同样使用 Membership Report 来回复路由器的查询。Membership Report 的目的地址是组地址，所以除了路由器之外，其它组员也能听到，一个网络中只会有一台主机向路由器发送 Membership Report 来通告网络中存在组员。

Version 1 Membership Report

Version 1 Membership Report 是兼容 IGMP v1 时发送的。

Leave Group

在 IGMP v1 中，并没有特定的机制来通知路由器组员的离开，唯一的方法就是路由器在 3 分钟后没有收到组成员对 queries 的回复，也就认为网络中的组成员已经离开。但是在 IGMP v2 中，组成员离开时，需要向路由器发送 Leave Group 来通告自己的离开。但是只有向路由器发送过 Membership Report 的组成员离开时才需要发送 Leave Group，其它组员离开时，也是没有任何动作的。Leave Group 目的地址为 224.0.0.2，也就是说组成员离开时，只有路由器知道，其它组员是不知道的。

IGMP v2 路由器：

IGMP 路由器靠查询消息来确认组成员的存在，运行 IGMP V2 的路由器使用 2 种查询消息：

1. General Query
2. Group-Specific Query

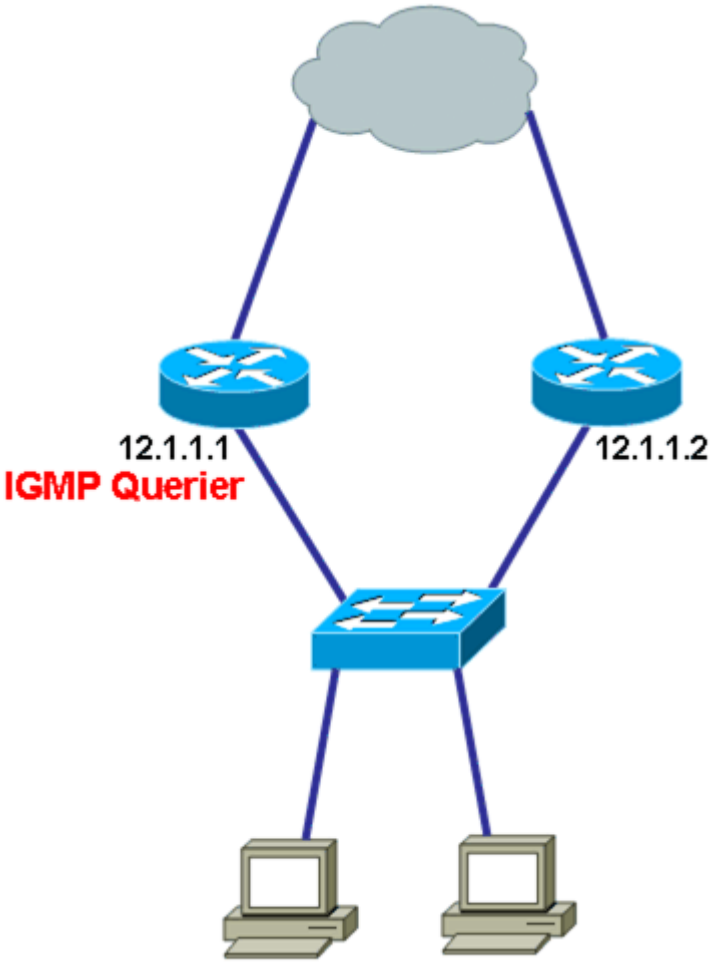
General Query

General Query 的作用和 IGMP v1 中 queries 的作用是一样的，当路由器发送 General Query 查询主机时，使用目的地址为 224.0.0.1，当有主机回复时，便认为网络中存在组成员，则将组播发送到该网络，并且 General Query 每一分钟发送一次，主机每次都回复，如果连续三个 General Query 没有回复，即三分钟，那么路由器便认为网络中的组成员已经离开，也就停止向网络中发送组播数据了。

Group-Specific Query

在 IGMP v2 中，当网络中的组成员离开之后，会向路由器发送 Leave Group 来通告自己的离开，而只有向路由器发送过 Membership Report 的组成员离开时才需要发送 Leave Group，其它组员离开时，也是没有任何动作的，因此当网络中的组成员离开后，路由器并不知道网络中是否还有其它组员，因为其它组员没有通告过自己的存在。所以如果路由器马上停止向子网转发组播的话，可能会造成某些还存在的组员无法接收组播。最终当路由器收到组成员离开时发来的 Leave Group 后，并不能马上停止组播的转发，还要向网络中发送 Group-Specific Query，目的地址为组的地址，用来查询网络中是否还有别的组员存在，如果有组员回复了，那么组播流就不会中断。为了防止包丢失而误认为没有组员，路由器每 1 秒分别发两个 Group-Specific Query。

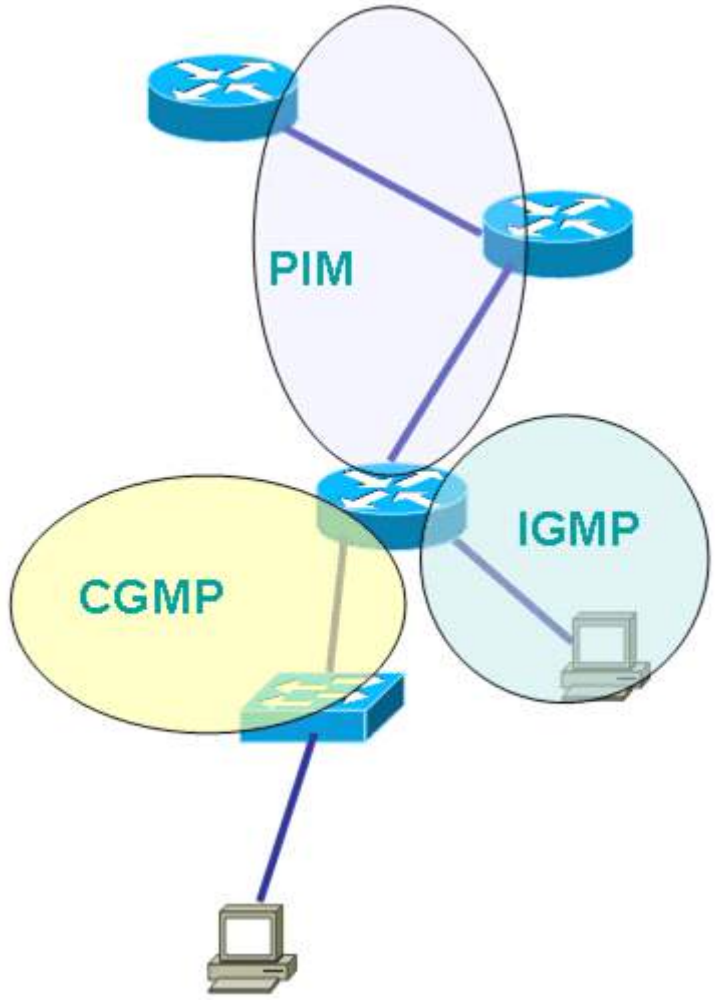
IGMP v2 Querier：



由上图可以看出，当有两台或更多路由器连接到同一个网络时，路由器要确认网络中是否有组成员，发送 General Query 就可以确认组员的存在，而当主机在使用 Membership Report 回复路由器的 General Query 时，因为 Membership Report 的目的地址是组地址，所以网络中所有路由器都能收到 Membership Report。如果网络中的每一台路由器都发送 General Query 来查询组员的存在，那么每台路由器得到的结果都是一样的，这样也就没必要让网络中的每台路由器都进行组成员的查询。这时，就从网络的多台路由器中选举出唯一的一台路由器来查询组成员，这台路由器就是 IGMP querier（IGMP 查询器），只要 IGMP 查询器查询一次，所有路由器都能得到相同的结果。网络中拥有最低 IP 地址的路由器将被选举为 IGMP 查询器，因此上图中被选为 IGMP 查询器的路由器为 12.1.1.1。当路由器正常启动 IGMP v2 后，就开始发送目标地址为 224.0.0.1 查询数据包，每台路由器收到后，查看数据包的源 IP 地址，最后得到结果谁是 IGMP 查询器，并且之后每一分钟发送一次查询数据包，如果路由器在两倍查询时间内没有收到查询器的查询数据包，便认为 IGMP 查询器已经失效，然后重新选举 IP 地址最小的为查询器。IGMP 查询器在 IGMP v1 环境下是没有的。

组播协议

要让组播正常运行起来，需要一些协议的协同工作，如下图所示，下面分别来介绍各协议的功能：



IGMP

因为组播数据只会朝着有组员的地方发去，所以要让想接收组播的主机正常收到组播数据，那主机就需要让发送者知道它的存在，当发送者知道了组成员的存在，就会向组成员发送组播数据。主机要宣告自己想要接收组播，可以发送一些数据包，而这些数据包，正是主机通过 IGMP 协议来发送的。当路由器想要知道网络中是否存在组成员时，同样也是发送 IGMP 数据包来查询的。在网络中确认组成员的存在，这就是 IGMP 协议的功能，可以看出，IGMP 是主机与路由器之间运行的组播协议。

CGMP

因为当路由器获知网络中存在组成员之后，就会将组播向该网络中转发。又因为网络中无论是一个组员，还是许多个组员，对于路由器来说，都得向网络中发送一份组播数据，如果网络中是用交换机连接了多台主机时，交换机收到组播后，会像转发广播一样，将组播发送给每一台主机。这样一来，如果只是部分主机是组成员的话，那么交换机的这种行为将导致组播转发的不精确，所以不能让交换机像处理广播那样处理广播，而必须将组播像处理单播一样，只发给需要接收的组成员。要实现这样的功能，就必须让交换机知道哪些主机才是组员，从而根据组员的 MAC 地址来转发组播。主机和交换机之间是没有协议的，却和路由器之间有协议，要准确的了解主机的 MAC 地址情况，就通过路由器得知，路由器再将此信息发送给交换机，最终使交换机能够像处理单播一样根据目标 MAC 地址来转发组播。这时就需要在路由器和交换机之间运行一种协议，那就是 CGMP（Cisco Group Management Protocol），在 CCIE R&S 考试中，CGMP 并不是考点，所以此篇中将不会对 CGMP 作出详细介绍。

PIM

只有当组播发送者知道组成员的存在，才会向组成员的方向发送组播。在某些情况下，发送者和组成员可能在远程网络，那么这样一来，发送者发出的数据，必须经过路由器才能到达组成员的网络，所以要使组播数据准确地被转发到组成员的网络，就必须让中间的路由器也知道组成员网络的位置所在。两个不同网络的主机使用单播通信时，数据可以被中间路由器准确地转发，是因为路由器的单播路由表中能够找到目标网络的位置，如果要让路由器也能像转发单播数据一样，将组播根据路由表来精确地转发到目的地，那就需要让路由器拥有像单播路由表一样的组播路由表，从而让路由器在收到组播时，就像查单播一样，去查组播该从什么样的接口被发出去才能到达目标网络。要让路由器生成一张功能完全的组播路由表，就需要在路由器之间运行一种协议，这种协议可以让组播源和目的之间的路由表生成单播表一样地生成组播表，最后路由器根据这张组播路由表来完成组播的转发。这个协议就是 PIM（Protocol Independent Multicast）。其实要让路由器知道目标组员的位置，完全可以依靠单播来找到组员，所以只要组播的源和目的之间单播是通的，那么组播路由表就能建立，而不用管单播运行的是动态路由协议还是静态路由协议，但是前提是 PIM 必须依靠单播路由表才能生成。

PIM

当组播的源和目的在远程网络时，需要依靠路由器转发组播数据，要让路由器像转发单播一样地转发组播，就必须生成一张组播路由表，而 PIM 协议正是用来帮助路由器生成组播路由表的协议，只有当路由器都拥有完整的组播路由表时，远程网络之间的组播才能正常通信。因为如果路由器不能对组播数据负责而随意转发组播，这将带来不可想象的结果，所以路由器必须像依靠单播路由表精确转发单播一样去转发组播，下面来详细介绍组播的重点知识，PIM 组播路由协议：

组播树

因为在需要将一份数据同时发给多个接收者时，而开发了组播技术，所以组播的发送者通常面临着要将数据发向多个接收者，并且这些接收者可以分布在任意网络的任意位置。如果接收者在远程网络，那么就需要路由器提供组播转发，所以要保证接收者能够正常收到组播，就必须让路由器知道自己该将组播从什么样的接口发出去，当组播到达下一跳路由器后，下一跳路由器同样也必须知道该将组播从什么样的接口发出去，即使接收者不是与自己直连的，只有这样让路由器之间协同工作，都能够记住组播的出口，最终在发送者与接收者之间形成一条连线，这样才能完成组播的转发。当多个网络存在接收者时，那么这样的连线就会有多条，组播发送者到接收者之间的这些转发线路，被称为组播转发树，而组播发送者就好比是组播树的树根，组播总是从根发向接收者。很容易想象得出，从发送者到接收者之间的路由器，都是在组播树上的，因为这些路由器在中间提供组播转发，那么如果一台根本没有与接收者相连的路由器，与组播树是没有任何关系的。要完成从发送者到接收者之间的组播转发，组播树上的路由器都应该记住组播的出口，每台中间路由器都记住出口之后，最终便形成了组播树，而要记住组播的出口信息，这就是组播路由表的工作。

组播反向路径转发

因为组播经常会有多个接收者分布在不同的网络，所以当路由器在转发组播时，也通常需要将同一份数据从多个接口发出去，如果别的路由器不小心或者由于各种原因再将组播发送回来，对于这样已经出现环路的数据，如果路由器收到后，再次

将组播转发出去，那么只会形成组播风暴，最终危害整个网络。由此可以看出，转发组播的路由器，必须拥有发现环路、避免环路的能力。要实现这样的功能，所有的组播路由器就必须只将数据往接收者的方向转发，而绝不能往发送者的方向转发，因为向发送者的方向转发，就等于是将数据往回发，那就是引起路由环路的原因。让组播路由器只将数据往接收者的方向转发，而不能往发送者的方向转发，这样的机制被称为组播反向路径转发（mRPF），这也是组播路由器必须遵循的机制。组播路由器在收到组播数据后，都要对数据进行 RPF 检测，只有从源的方向发来的数据才能被转发，从其它接口过来的数据被认为是无效的。

因为组播路由器在转发数据时，必须遵循组播反向路径转发（mRPF）机制，所以路由器除了要记住数据从什么接口发出去是接收者之外，还要记住组播发送者是在哪个接口，而数据是绝不能发向发送者的。由此可以看出，组播路由器在正常工作时，组播转发表中必须同时记住接收者的接口和发送者的接口，其它不相关的接口，可以不用记住，也可以记为剪除状态，被剪除的接口是没有组播流量的。

组播路由表记住了接收者的接口后，就能正常将数据发给组成员，而记住了发送者的接口后，就可以避免路由器将数据往回发而产生路由环路了。组播路由器上朝发送者方向的接口被称为 RPF 接口，这个接口是在单播路由表中指向组播源地址的出口，因为从 RPF 接口发出去的数据就等于将数据发向发送者，所以组播只能发往除 RPF 接口之外的其它接口，并且收到的组播数据如果不是从 RPF 接口进来的，全部都将被丢弃。

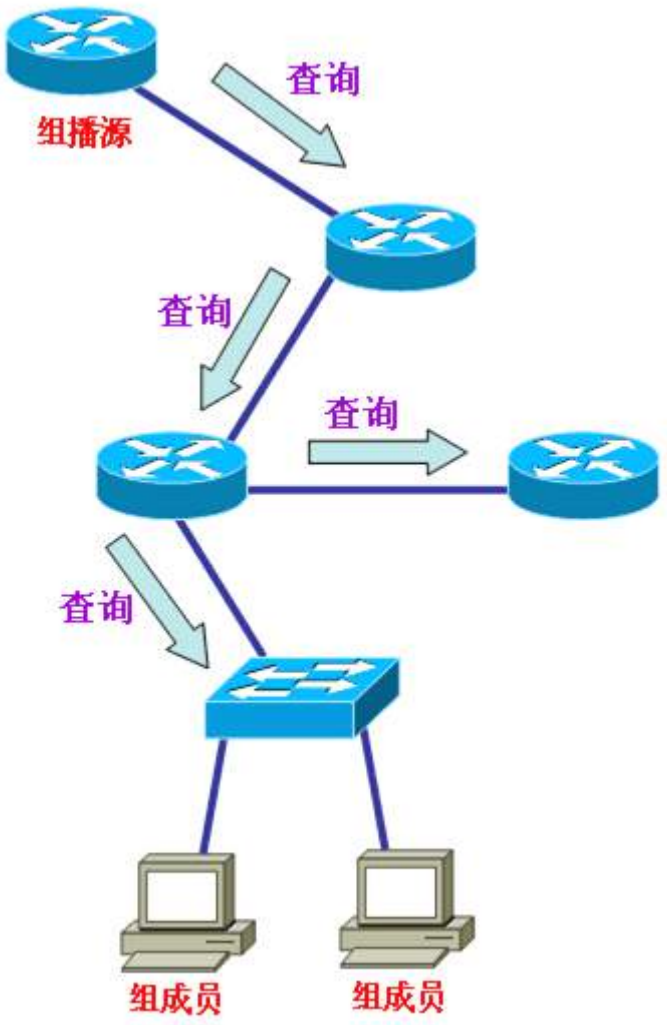
PIM 模式

在组播正常通信之前，组播路由器必须知道哪个接口是通往发送者的，即 RPF 接口，组播数据只能从 RPF 接口进来。组播路由器还必须知道哪些接口是通往接收者的，组播将从这些接口被转发出去。从组播源到接收者之间的所有路由器都记住了 RPF 接口和组播出口的组合，被称为组播树，组播树用来指导路由器如何转发组播，而组播树就是记录在组播路由表中的，因此，有了正确的组播转发表后，组播就能够正常通信了。

要让组播路由器生成组播路由表，就是 PIM 协议的工作，PIM 在组播路由器之间运行，通过路由器之间的协商，从而获得组播路由表，构建出组播树。PIM 要为路由器学习组播路由表从而建立组播树，有两种不同的方式，这两种不同方式在 PIM 中分两种模式来运行，为 PIM-DM（密集模式）和 PIM-SM（稀疏模式）,下面来详细介绍 PIM 的两种运行模式：

PIM-DM（密集模式）

组播树是用来指导路由器如何正确转发组播的，它的相关信息全部都是记录在组播路由表中的，PIM 用来帮助路由器生成组播路由表。要形成组播树，路由器需要知道哪些接口出去能够到达接收者，并记录下来，然后再记录到发送者的 RPF 口。要让路由器知道哪些接口存在接收者，有两种方式，第一种方法是接收者主动向路由器报告，第二种方法是路由器主动向网络中发出查询，而 PIM-DM 模式中，采用的方法为路由器主动向网络查询是否有接收者，如下图所示：



在上图显示的 PIM-DM 工作模式中，组播源会向所有 PIM 邻居发出查询，查询数据包中包含组的地址，下一跳 PIM 邻居还会继续向它的邻居发出查询数据包，这些查询数据包会在所有 PIM 邻居之间传递。如果查询数据包发向一个连接了组成员

的网络，这时路由器收到组成员的报告之后，就会向自己上一跳邻居（RPF 接口方向的邻居）发送加入组的消息，以宣布自己要接收组播，从而将组播转发到组成员。在这些过程中，如果某些 PIM 路由器根本没有与组成员相连，那么它将会向自己的上一跳邻居发送剪除消息，以宣布自己不需要接收组播，最终组播从源发出后，只会沿着组播树被发到连接了组成员的网络，而其它不相关的网络是不会有组播流量的。

以上的信息，路由器都会记录在组播路由表中，这个路由表是不同与单播路由表的，组播路由表拥有自己的格式，因为网络中可能存在多个组，所以路由器在记录时，应该记录下组地址，然后就是该组需要从哪些接口被发出去，最后还由于路由器需要对组播数据进行 RPF 检测，所以还必须记录与组对应的组播源地址，也就是记录发送组播数据的源 IP 地址。例如网络中有一个组，地址为：224.1.1.1，发送者为 100.1.1.1，那么就应该记录为（100.1.1.1，224.1.1.1），除了记录下组地址和源 IP 地址之外，路由器还要记录的就是组播的出口，收到的组播就会从这些出口发出去。PIM-DM 模式中这样记录组播的方式被称为(S,G)，其中 S 就是组播源地址，G 就是组地址，而出口则会被标为 forwarding。

如果只有一个组 224.1.1.1，而发送者除了 100.1.1.1 之外，还有 100.1.1.2 和 100.1.1.3，那么依照（S，G）的记录方式，就需要同时记录（100.1.1.1，224.1.1.1），（100.1.1.2，224.1.1.1），（100.1.1.3，224.1.1.1），也就是说(S,G)的记录方式，会因为组源地址的增加而增加记录条目。

可以看出，对于一个组，PIM-DM 模式中，路由器需要记录组地址，源地址，出口信息，除了这些之外，比如路由器上有 5 个接口有 PIM 邻居，其中只有一个是出口，则这个接口被标为 forwarding，再去掉 1 个 RPF 接口，那么还剩 3 个 PIM 接口是不需要接收组播的，对于不需要接收组播的 PIM 接口，PIM-DM 模式照样会将其记录在路由表中，但被标为 pruning。因为 PIM-DM 这样的记录方式，可以看出当路由器上只有少量的 PIM 接口与组成员相连时，PIM-DM 会消耗更多的资源去记录一些没有组成员的不相关的接口，而这样的事情是可以避免的。

在 PIM-DM 模式下，组播发送源将数据发给组播路由器，然后路由器依照组播路由表朝着接收者的方向转发，这样的路径，是依靠单播路由表计算出来的最短路径，也就是说从发送者到接收者之间的路径，总是最短的，所以 PIM-DM 模式建立起来的组播树，如（S，G）记录的组播树被称为最短路径树 shortest-path tree (SPT)，因为 SPT 中记录个每个组播发送者的源地址，故又被称为源树。

PIM-SM（稀疏模式）

在 PIM-DM 模式中，组播树的生成是靠路由器发送查询消息来建立的，在组播路由表中记录下组地址，发送者的源 IP 地址，同时还需要记录下所有 PIM 接口状态，

而不管这些接口是连接着组成员的，还是没有。

而 PIM-SM 模式的工作过程和 PIM-DM 模式是不同的，在建立组播树时，PIM-SM 并不会让路由器发送查询数据包去查询组成员，而组成员的发现是靠组成员自己主动向路由器发送报告数据包，当一台路由器从接口上收到组成员的报告之后，就会向自己的上一跳邻居发送加入消息，以通告自己需要接收组播，如果上一跳邻居还不是组播发送者，那么上一跳邻居会继续再向上一跳邻居发送加入消息，直到组播源收到加入消息为止，通过这样的方式，就可以建立组播源到组成员之间的组播树。

PIM-SM 模式和 PIM-DM 模式除了在发现组成员的方式上不同之外，记录路由表的方式也不一样，例如网络中有一个组，地址为：224.1.1.1，发送者为 100.1.1.1，PIM-SM 模式记录为（*，224.1.1.1），可以看出，PIM-SM 模式并不关心组播的源地址，而统统将源地址使用星号*来表示，这样一来，PIM-SM 为一个组只记录一个条目，而不管这个组有多少个发送者。这样的记录方式称为(*，G) ，其中*就是组播源地址，G 就是组地址。如果有 10 个组，每个组有 5 个发送者，使用（S，G）的记录方式，需要记录 10×5=50 条，而使用（*，G）的记录方式，则只需要 10 条，所以使用 PIM-SM 模式，可以大大缩减组播路由表的空间，从而大大节省系统资源。

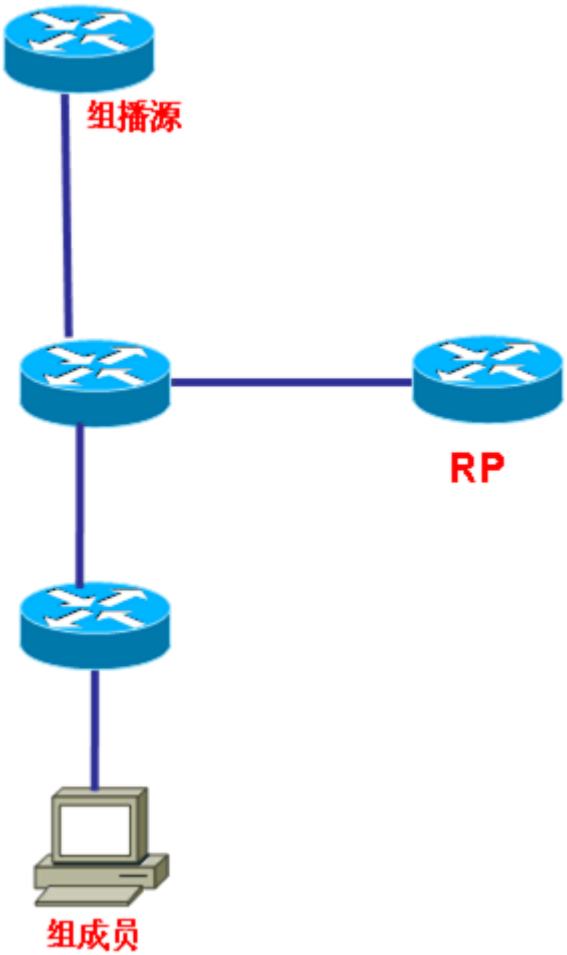
PIM-SM 模式不仅需要记录(*，G)信息，也和 PIM-DM 模式一样需要记录在该组中，哪些接口是出口，从此接口将数据发给接收者。但是与 PIM-DM 模式不同的是，PIM-SM 模式只记录连接着接收者的接口，其它没有接收者，不需要接收组播的接口是不会被记录的。比如路由器上有 5 个接口有 PIM 邻居，其中只有一个是出口，再去掉 1 个 RPF 接口，还剩 3 个 PIM 接口是不需要接收组播的，在 PIM-DM 模式中，会记录下一个 RPF 口和一个 forwarding 状态的出口，以及三个不需要接收组播的 pruning 状态的接口，而在 PIM-SM 模式中，只会记录一个 RPF 口和一个 forwarding 状态的出口，其它的都不作记录，因此，PIM-SM 模式的接口记录方式会比 PIM-DM 模式更省资源。

PIM-SM RP

在 PIM-SM 模式中，由于记录组播信息采用(*，G)的方式，而并不关心组播源地址，因此造成路由器不知道组播发送者的 IP 地址是什么，也就无法完成 RPF 反向路径检测。在这种情况下，PIM-SM 在网络中选出一个组播会聚点，即 Rendezvous Point (RP)，RP 就是组播网络的核心，发送者统一将组播数据发送到 RP，然后 RP 再将数据发到接收者，也就是说接收者收到的数据，都是由 RP 转发过来的，路由器也

就认为 RP 的地址，就是组播源的 IP 地址。

在 PIM-SM 模式中，通过 RP 转发数据，如下图所示：

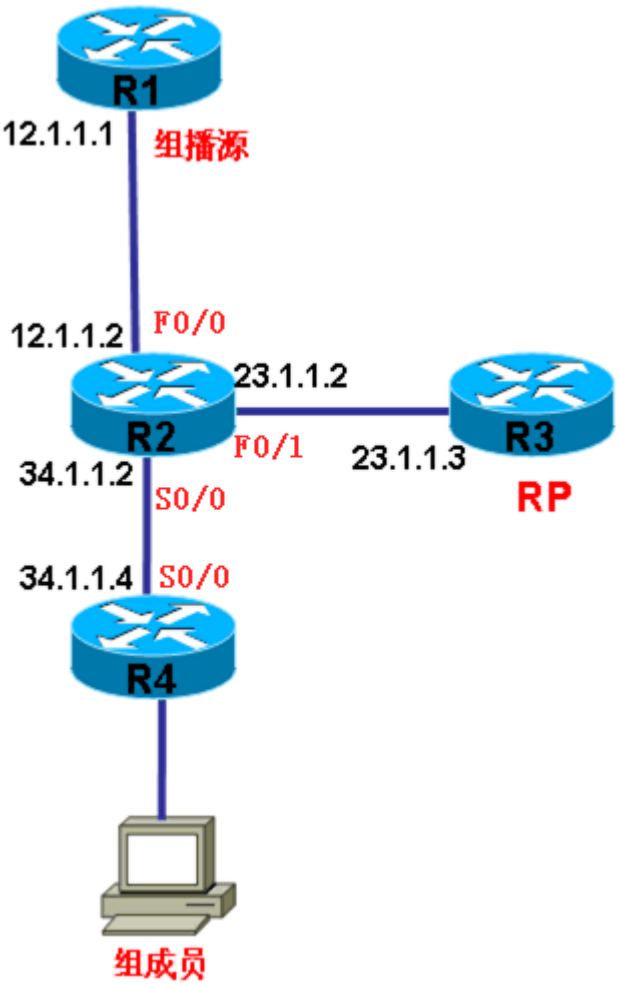


因为数据总是先发送到 RP，然后 RP 再发给组成员，如果 RP 出现了故障，也就意味着组播无法正常通信。组播中的一个组，只有一个 RP 是处于转发状态的，有时 RP 路由器的位置可能不是离每个组成员都是最近的，并且数据从源到 RP 然后再到组成员的路径也可能会很远，所以 RP 的位置，会导致源到组成员的数据包走的不是最优路径，也会在大流量的情况下，成为组播通信的瓶颈。但是在这种情况下，PIM-SM 有自己的解决方法，就是在正常组播通信时，路径会切换到源到组成员最短的路径上，也就是挑选网络中的最优路径来转发，和源树的路径方式相同，这个规则可以随意更改，定义流量超过多少时可被启用，思科路由器默认收到第一个包后即启用。

因为 RP 在 PIM-SM 中被当作核心，而(*，G)的记录方式中并不知道组播源的地址，大家都会认为 RP 就是组播源，最后 PIM-SM 中的组播树，即(*，G)形式的树被称为共享树（RPT）。

PIM-SM 模式中的 RP，可以手工指定，也可以使用动态协议来选择。如果使用手工指定，当使用中的 RP 出现故障后，也必须手工更改，才能恢复通信。而使用动态协议，则可以配置多个备份 RP，但活动 RP 只有一个，当活动中的 RP 出现故障后，协议会重新选择备用 RP 替换为当前的活动 RP，能够实现冗余功能。

在 PIM-SM 中，因为(*，G)形式的记录中不知道组播源的地址，也就无法完成 RPF 检测，但是又由于接收者收到的数据，都应该是 RP 发来的，路由器也就认为 PIM-SM 中的组播源地址，就是 RP 的地址，在这种情况下，路由器会以 RP 的地址为源地址做 RPF 检测，但是在下面的拓朴中，就会出现问题：



如上图所示，当所有路由器都认为 RP 地址 23.1.1.3 就是组播源地址时，这样去做 RPF 检测，在 R4 上检测时，因为 R4 的路由表中会显示从接口 S0/0 出去可以到达 23.1.1.3，所以组播从 R4 的 S0/0 被发进来时，则 RPF 检测可以通过，所以在 R4 上没有任何问题。

当在 R2 上做 RPF 检测时，因为 R2 的路由表中显示到达 23.1.1.3 应该从接口 F0/1 出去，所以 R2 只会接收从接口 F0/1 发进来的组播，从其它接口发进来的，都会被认为是环路而被丢弃。因此在真正的源 12.1.1.1 将组播从 R2 的 F0/0 发进来时，R2 会因为 RPF 检测失败，从而丢弃所有的组播数据，这样一来，组播也就无法通信了。

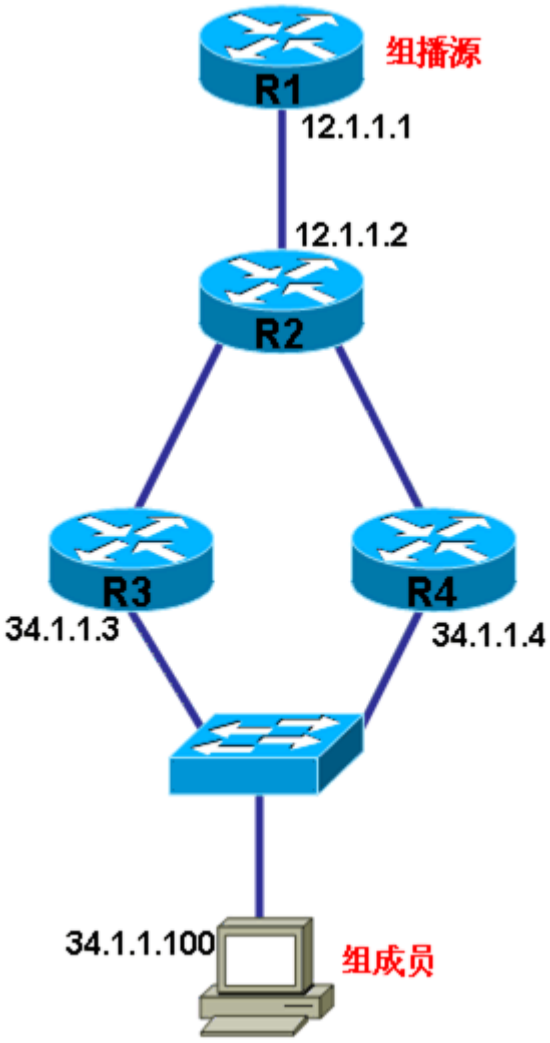
PIM DR

因为 IGMP V1 不选举 IGMP 查询器，如果有多台路由器连接同一个多路访问网段时，就必须选举 PIM Designated Router （DR）来避免重复查询，这个 DR 的选举，是由 PIM 进行的，选择 IP 地址高的为 DR。PIM DR 路由器在 IGMP v1 中充当 IGMP 查询器的功能，来向主机发送 IGMP 查询。

在 PIM-SM 中，组播源没有机制用来宣告自己的存在，又因为 PIM 路由器都认为 RP 才是组播源，而当直接的组播源向 RP 发送组播时，会导致某些路由器 RPF 检测失败，因此在 PIM 网络中，真正的源需要向 RP 发送注册消息，以宣告自己的存在，而发送注册消息在多路访问的网络中则是由 DR 来代为完成的。真正的源向 RP 发出第一个组播包时，DR 将此包封装在单播中发向 RP，这称为注册，发送的注册消息会从 DR 到 RP 之间建立一条源树，也就是（S，G）的记录，这样，在源到 RP 之间创建的源树就可以帮助避免 RPF 检测失败，当 RP 和真正的源之间创建（S，G）条目之后，就会通知 DR 停止以单播发送，从而转回发送真正的组播。因为由于误认为 RP 是源而导致 RPF 检测失败的情况，只有在 RP 到源这段上游网络中的路由器才会存在，而 RP 到组成员的这段网络中是不会出现这种情况的。由此可以想象得出，当源所在的网络需要向 RP 发送数据包时，应该将 DR 的位置选为最靠近 RP 的路由器，否则也会因为 RPF 检测失败而导致组播不通。

选择 DR 的数据包为 30 秒一次，105 秒保持时间，如果过了保持时间没有收到 DR 的数据包，则会重新选举 IP 地址最高的为 DR。

PIM 前转器



从上图中可以看出，当组播源 R1 将数据发向组成员时，数据到达 R2 之后，可以分别从 R3 和 R4 两条路径到达组成员。要知道组播通常是应用在视频和语音环境下的，假如 R2 选择将数据包以两条路径负载均衡的方式发往组成员，当 R3 和 R4 两条路径之间的延时不相同时，数据包到达组成员就会出现明显的先后顺序，这在普通数据传输时是没有问题，但是如果当数据是视频时，那么就可能出现视频画面先后不一致的情况，又如果当传输的是语音时，对方在报数字，比如 R2 向组成员发送 123456，在采用负载均衡传输时，将 1 发向 R3，将 2 发向 R4，再将 3 发向 R3，再将 4 发向 R4，再将 5 发向 R3，再将 6 发向 R4，当 R3 链路上的速度比 R3 快时，数据 135 就会先到达组成员，246 就会后到达，最终导致 R2 发出的数据 123456，在组成员收到时，就可能变成了 135246。由于这些原因，组播负载

均衡可能导致数据传输错误，所以当组播源到组成员之间有多条路径可走时，是一定要选出其中唯一的一条路径的，如上图中，就 R2 就必须选出要么从 R3 走，要么从 R4 走，而绝不能负载均衡，两个下一跳中被 R2 选中的路由器，称为 PIM 前转器，选 PIM 前转器的规则是，比较哪台路由器的路由表中，到组播源的路由的 AD 值最小，如果 AD 值相同（如使用同一种路由协议），再比较谁到源的 metric 值最小，如果 metric 值还相同，则选择 IP 地址大的。在上图中，如果 R3 和 R4 到组播源 R1 的路由条目中是使用同一个路由协议学到的，则比较 metric 值，如果 metric 值相同，则比较 R3 和 R4 的 IP 地址，这个 IP 地址是同网段相互建 PIM 邻居所使用的 IP 地址，因为 34.1.1.4 比 34.1.1.3 要大，所以前转器为 R4，因此 R2 会选择从 R4 到达组成员，并且把到 R3 的出口剪除。

PIM-DM 数据包

在 PIM-DM 模式中，所使用的源树，会产生（S，G）条目，但是也会创建（*，G）的条目，创建（*，G）是为了作为（S，G）的“父”数据结构，所有 PIM 接口都会加入输出接口，而输入接口总是空的。

在运行 PIM-DM 时，路由器会发送 5 种数据包：

1. Hello
2. Join/prune
3. Graft
4. Graft-Ack
5. Assert

1. Hello

PIM 路由器用来发现邻居，会在启动了 PIM 的接口上周期性地发送 Hello，默认 30 秒发送一个，有个保持时间，如果过了保持时间没有收到邻居的消息便认为邻居丢失，保持时间为 Hello 的 3.5 倍（105 秒）。PIM 邻居为 0.0.0.0 的， 表示邻居就是组播源。

2. Join/prune

PIM-DM 启动后，组播源开始向所有邻居发送查询消息，邻居会再向自己的邻居发送查询消息，如果有路由器连接着组成员，那么就会向上一跳邻居发送 join 数据包，用来通告自己需要接收组播，从而将自己接入组播树。其它没有连接组成员或不需要接收组播的路由器则向上一跳邻居发送 prune 数据包，用来通告自己不需要接收组播，状态被标为 prune，而不是 forward。被标为剪除状态，是方便被剪除的路由器在必要时把自己接入树中。

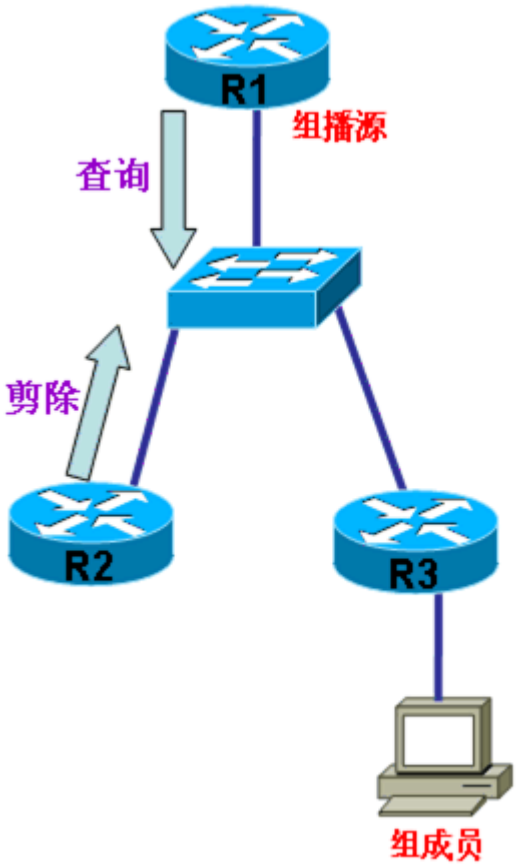
3. Graft

在路由器向下一跳邻居发送查询数据包，查询是否有组成员存在时，邻居必须发送 join 数据包来通告自己需要接收组播，然后不需要接收的，则发送 prune 将自己剪除，如果剪除后的路由器后来需要再接收组播，是不能发送 join 数据包的，而必须发送 graft 来将自己接入组播树。当被剪除的路由器向上一跳邻居发送 graft 数据包后，上一跳会将邻居接口标为组播的出口，即 forward 状态。如果上一跳邻居也是被剪除的，那么也会再向上一跳邻居发送 graft。

4. Graft-Ack

当被剪除的路由器向上一跳邻居发送 graft 数据包后，上一跳除了会将邻居接口标为组播的出口之外，还会向下一跳邻居发送一个 Graft-Ack。

Prune 消息覆盖



当一台路由器收到 PIM 查询后，如果自己不需要接收组播，或者没有连接组成员，就会向上游发送 Prune 消息将自己剪除。如上图中，因为 R1 有两个 PIM 邻居，为 R2 和 R3，当 R1 向网络中发出查询后，因为 R2 不需要接收组播，所以会发送 prune 消息将自己剪除，在正常情况下，R1 是应该将出口剪除的，但是可以看出，R3 连有组成员，是需要接收组播的，如果 R1 因为收到 R2 的剪除消息而将出口剪除后，那么 R3 发出的 join 消息是不能再收到组播的，而必须发送 graft，因为这样，就会导致 R3 在收到查询后发送 join 却不能收到组播，要避免这样的事情发生，PIM 路由器向网络中发出查询后，即使收到剪除消息，也不会马上将接口剪除，而会启动一个 3 秒的计时器，如果在 3 秒内，收到 join 后，就会覆盖前面的剪除消息，就不会将接口变成剪除状态，这被称为 Prune 消息覆盖。

5. Assert

Assert 是用来选举 PIM 前转器的，当从源到组成员有多条路径可走时，则选择其中唯一的一条路径，被选中的路由器称为 PIM 前转器。

Assert 消息中包括源和组地址，以及到源的单播路由 AD 值和 metric 值， 先选 AD 值低的，其次是 Metric，最后是最高 IP 地址，落选的把自己的出口剪除。

PIM-SM 数据包

PI-SM 模式因为组播源要把数据包先发送到 RP，然后 RP 再将数据发到组成员，这样的共享树可能出现走远路的情况，所以有时不如 PIM-DM 中的源树。但是又因为 PIM-SM 可以从共享树切换到源树，也就是同时支持共享树和源树，而且在记录组播路由表时，更节省资源，所以被多数选用。

在运行 PIM-DM 时，路由器会发送 5 种数据包：

PIM-SM 采用了 7 种 PIMv2 消息：

1. Hello
2. Bootstarp
3. Candidate-RP-Advertisement
4. Join/Prune
5. Assert
6. Register
7. Register-Stop

1. Hello

PIM-SM 中的 Hello 同 PIM-DM 中的 Hello 功能相同，是用于 PIM 邻居的建立和维护。

2. Bootstarp

Bootstarp 用于 RP 的选举，稍微将详细介绍。

3. Candidate-RP-Advertisement

Candidate-RP-Advertisement 用于 RP 的选举，稍微将详细介绍。

4. Join/Prune

PIM-SM 中的 Join/Prune 同 PIM-DM 中的 Join/Prune 功能相同，是用于路由器加入组播树或将自己从中剪除。

5. Assert

PIM-SM 中的 Assert 同 PIM-DM 中的 Assert 功能相同，用于选举 PIM 前转路由器。

6. Register

DR 在功能，在前面已经提到过，就是在 PIM-SM 中，因为 PIM 路由器都认为 RP 才是组播源，而当直接的组播源向 RP 发送组播时，会导致中间路由器 RPF 检测失败，因此在 PIM-SM 网络中，真正的源需要向 RP 发送组播，就应该建立一条（S，G）的记录来避免 RPF 检测失败。当真正的源向 RP 发出第一个组播包时，DR 将此包封装在单播中发向 RP，这称为注册，也就是 Register 消息，发送的注册消息会从 DR 到 RP 之间建立一条源树，也就是（S，G）的记录，这样，在源到 RP 之间创建的源树就可以帮助避免 RPF 检测失败，当 RP 和真正的源之间创建（S，G）条目之后，就会通知 DR 停止以单播发送，从而转回发送真正的组播。

7. Register-Stop

因为 PIM-SM 中真正的源需要向 RP 发送 Register 消息来注册，从而建立（S，G）条目，将组播正常发向 RP，当 RP 和真正的源之间创建（S，G）条目之后，就会发送 Register-Stop 通知对方停止发送单播，，从而转回发送真正的组播。

最后可以看出，在 PIM-DM 中，不仅会建立（S，G）的记录，也会建立（*，G）的记录，而在 PIM-SM 中，除了建立（*，G）的记录之外，同样也有（S，G）的记录，组播树建立后，会周期性发送 Join/Prune，60 秒一个，保持时间为 3 倍，即 180 秒，也就是说在 180 秒之后没有正常的组播流量，那么这些组播树是会被清除的。

RP 的确立

在 PIM-SM 中，组播源必须将数据发送到 RP，再由 RP 转发给组成员，当网络中没有 RP 时，组播是不正常的，因为要保证组播的正常通信，首先要让所有 PIM 路由器知道 RP 的地址，在共享树建立之前，必须确立 RP，可以单独为某个组配置一个 RP，也可以为多个组配置一个 RP，要确立 RP，有三种方法，分别是：

1．手工静态配置

2．Bootstrap Router (BSR)来通告

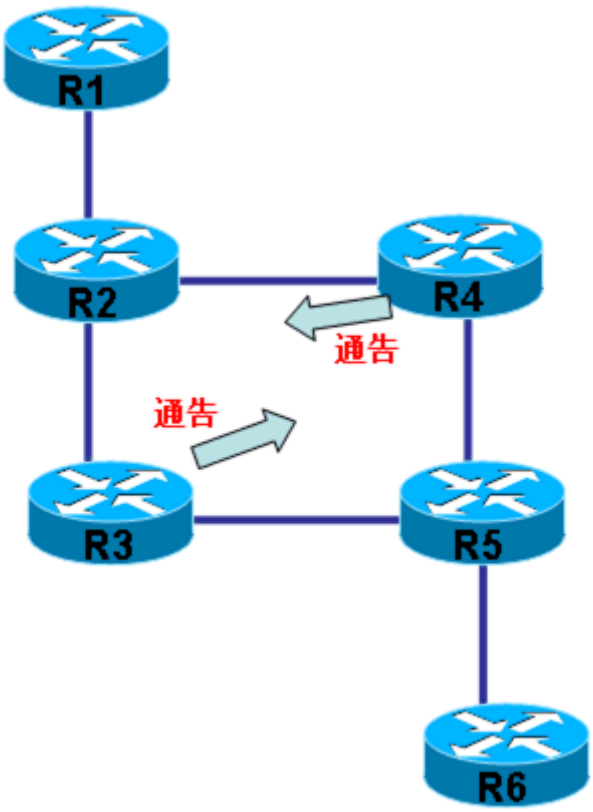
3．自动 RP（Auto-RP，思科独有协议）

1．手工静态配置

一个网络中，PIM-SM 的 RP 可以手工静态为每台 PIM 路由器配置，RP 也需要配置，因为要让每台路由器都知道谁是 RP，但是这种方式不能提供冗余功能的 RP，当配置的 RP 失效后，必须手工更改，否则组播将不能通信。

2．Bootstrap Router (BSR)来通告

RP 的确立，除了手工为每台路由器静态配置之外，还可以使用协议自动选择，使用协议的好处在于，可以在网络中配置多个 RP，起到备份的作用，当正在使用的 RP 失效后，协议可以立即重新选择其它路由器成为活动 RP。自举协议 BSR 的工作方式为在网络中配置多个 RP，称为候选 RP（C-RP），但只有一个 RP 是正在使用的活动 RP，在多个 C-RP 中，要成为活动 RP，选举规则为优先级最高的成为活动 RP，优先级默认为 0,范围 0 到 255，数字越小，优先级越高，如果优先级都相同的情况下，再比较 IP 地址高的，所以每个 C-RP 都会有一个 IP 地址用来表示自己的身份，也通过此 IP 来竞选 RP。如下图所示：



上图中，如果同时将 R3 和 R4 都配置为 C-RP，在不改变优先级的情况下，谁的 IP 地址最大，谁就是活动 RP，但是如果因为某些原因，R3 发出的 RP 竞选消息只被 R1 和 R2 收到，而 R4 发出的 RP 竞选消息只被 R5 和 R6 收到，那么 R1 和 R2 都会认为 R3 是 RP，而 R5 和 R6 都会认为 R4 是 RP，那么这样一来，网络中路由器得到的 RP 信息就变的不一致，从而导致组播故障，所以一个网络中，谁才是活动的 RP，并不能让每台路由器自己去计算结果，因为可能会出现大家计算出不同的结果。要解决这个问题，方法就是在网络中选举出一个 RP 裁判，称为 BSR，而所有 C-RP 将自己竞选 RP 的消息统一发送到 BSR，是通过单播发向 BSR 的，最后由 BSR 从收到的竞选消息中，选择出活动 RP，再将活动 RP 的地址统一发给网络的每台路由器，这样就能保证每台路由器得到的消息都是统一的，每台路由器获知的 RP 地址都是统一的。而 C-RP 发出的竞选消息是 Candidate-RP-Advertisement。而 BSR 是通过路由器发送 Bootstarp 选出的，每一台候选 BSR(C-BSR 都)有一个 0 到 255 范围的优先级，默认为 0，优先级相同的情况下，IP 地址最高的为 BSR。Candidate-RP-Advertisement 和 Bootstarp 都是 60 秒发送一次。

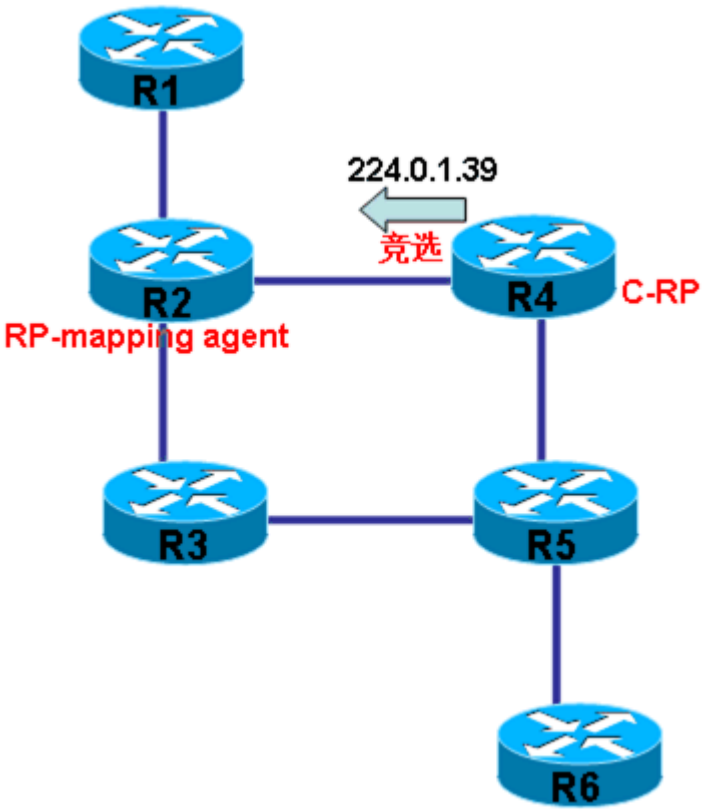
3. 自动 RP（Auto-RP，思科独有协议）

Auto-RP 的工作过程与 BSR 相同，同样是在网络中配置多个候选 RP 和 RP 裁判，候选 RP 向 RP 裁判发送竞选消息(RP-Announce),最后 RP 裁判从候选 RP 的竞选消息中选出 IP 地址最高的为活动 RP，然后发送 RP-Discovery 通告给每台路由器，不同的是，竞选活动 RP 只根据 IP 地址大小，没有优先级之分。在 Auto-RP 中，侯选 RP 仍然被称为 C-RP，但 RP 裁判被称为映射代理（RP-mapping agent）。所有的 C-RP 向映射代理发送竞选消息，使用目的地址为 224.0.1.39，每 60 秒发送一次，而映射代理从众多 C-RP 中选出活动 RP 后，以目的地址为 224.0.1.40 发给每台路由器，也是每 60 秒发送一次。

Pim sparse-dense-mode

组播路由器在运行 PIM 时，可以运行在 SM 模式下，也可以运行在 DM 模式下，当运行在 SM 模式下时，必须有 RP，否则网络不通，而运行 DM 时，不需要 RP 组播就能通信。但 PIM 路由器可以同时运行两种模式，即 sparse-dense-mode，当同时运行这两种模式时，如果一个组有 RP 时，则使用共享树，但是当 RP 失效时，则可以使用最短路径树来保证组播的通信。

在下图的网络情况下：



如果在上图网络中仅运行 SM 模式，并且使用 Auto-RP 来选举 RP，当配置 R4 为 C-RP，配置 R2 为 RP-mapping agent 时，R4 必须发送 Announce 来竞选 RP，当 R2 收到 R4 的 Announce 后，才能得到最后的结果。配置 SM 模式时，在没有 RP 的情况下，组播是不通的，所以 C-RP 要发送 Announce 将 RP 选出来，但是 C-RP 在发送 Announce 时使用的目的地为 224.0.1.39，这也就意味着 C-RP 发出的 Announce 是到达不了远程 RP-mapping agent 的，因为没有 RP，所以最终导致没有 RP 的情况下，组播就不通，组播不通就选不出 RP，结果是组播永远通不了，要解决这个问题，就是要在 RP 没选出来的情况下就要让组播能够通信，这种情况下，就可以将 PIM 模式配置为 sparse-dense-mode，因为这种混合模式下，在没有 RP 时，可以走 SPT 从而连通组播，这样目的地为 224.0.1.39 的 Announce 也就能够正常发送到 RP-mapping agent，就能选出 RP 了。

PIM Dense Mode Fallback

当将 PIM 路由器配置成 sparse-dense-mode 时，在 RP 不可用的情况下，将从 RPT 切换到 SPT，这种特性被称为 Dense mode fallback，默认是开启的，但是如果接口只配了 PIM-SM，那么就不可能往 SPT 切换的，也就必须拥有 RP 才能通信。在配置 sparse-dense-mode 后，如果想阻止从 RPT 切换回 SPT，可关闭 Dense

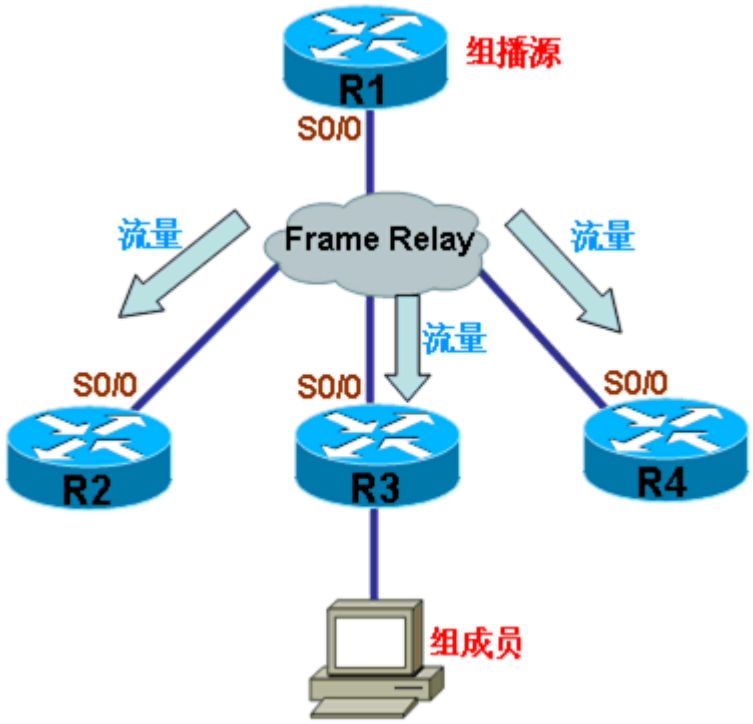
mode fallback 功能，命令为全局模式下输入 no ip pim dm-fallback。

共享树切换到源树

在 PIM-SM 中，组播源会先将数据发到 RP，然后 RP 再发给组成员，数据从源到 RP 再到组成员，这样的路径在很多时候可能不是网络中的最优路径，所以为了从源到组成员之间走最短路径，而不希望走次优路径，这就需要使用最短路径树，即 SPT。在路由器收到组播流量后，在源到组成员之间动态地创建最短路径树，从而替代共享树，这种行为在 IOS 中是默认开启的，但也可以单独为某个组开启，也可以关闭此特性。IOS 默认收到第一个组播包后，就从共享树切换到最短路径树，可以手工定义在共享树中的流量超过多少 kbps 后开始切换，流量低下去之后，60 少会再切换回去。通过在全局模式下配置命令 ip pim spt-threshold，比如希望在流量超过 4Kbps 后切换到 SPT，命令为 ip pim spt-threshold 4 。而命令 ip pim spt-threshold infinity 表示永远使用共享树而不切换到最短路径树。

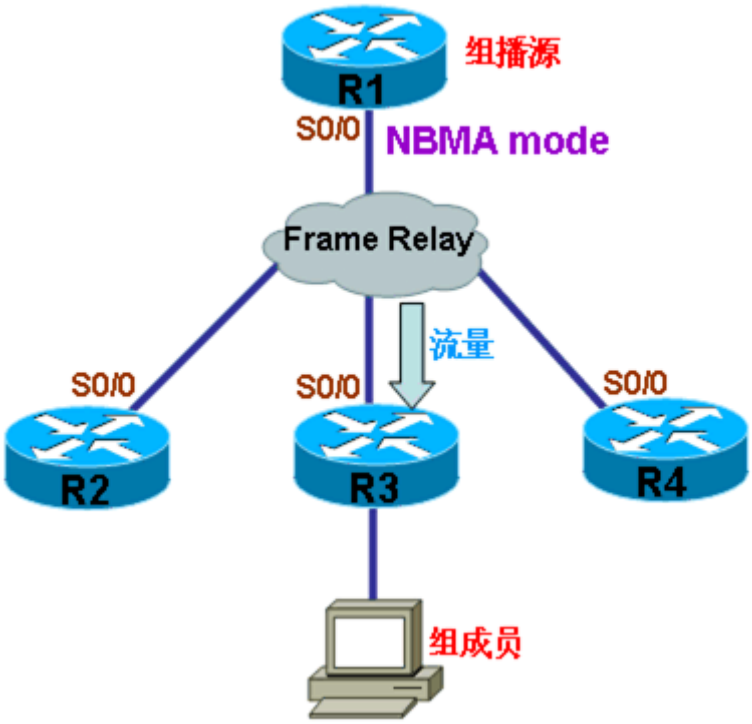
PIM-SM 之 NBMA Mode

因为帧中继中没有内置的处理广播和组播的能力，所以在收到广播和组播时，会在所有配了关键字 broadcast 的 PVC 上转发，这样当帧中继接口对端有多个邻居，而只有部分邻居要接收组播时，那么组播的被当作广播转发的行为，将影响不需要接收组播的路由器，如下图：



在上图中，当 R1 通过帧中继主接口连接多个对端时，在收到组播后，将从帧中继主接口发送给每一位配置了关键字 broadcast 的邻居，但并不是所有邻居都有组成员，而只有 R3 需要接收组播，这样就会影响到 R2 的 R4。因此需要让组播路由器在帧中继环境下，只将组播转发给要接收组播的路由器，可以通过配置子接口或者将接口模式在 PIM 下改为 NBMA Mode，子接口类似多个物理接口，会建立多个邻居，所以会根据每个邻居情况进行组播转发，而配置 NBMA Mode 后，组播在主接口上会跟踪 PIM-SM 模式的 join 消息，以记录需要接收组播的路由器地址，之后就只会将组播转发到相应的地址中。如下图：

鸿鹄培训+考试群：301283867



通过将 R1 的帧中继主接口配置成 NBMA Mode 后,R1 将在接口上跟踪 join 消息的源地址，所以可以得知只有 R3 需要接收组播，在需要转发组播时，只会转发给 R3。NBMA Mode 只能用在 PIM-SM 环境下，在 PIM-DM 下是不可用的。

配置组播

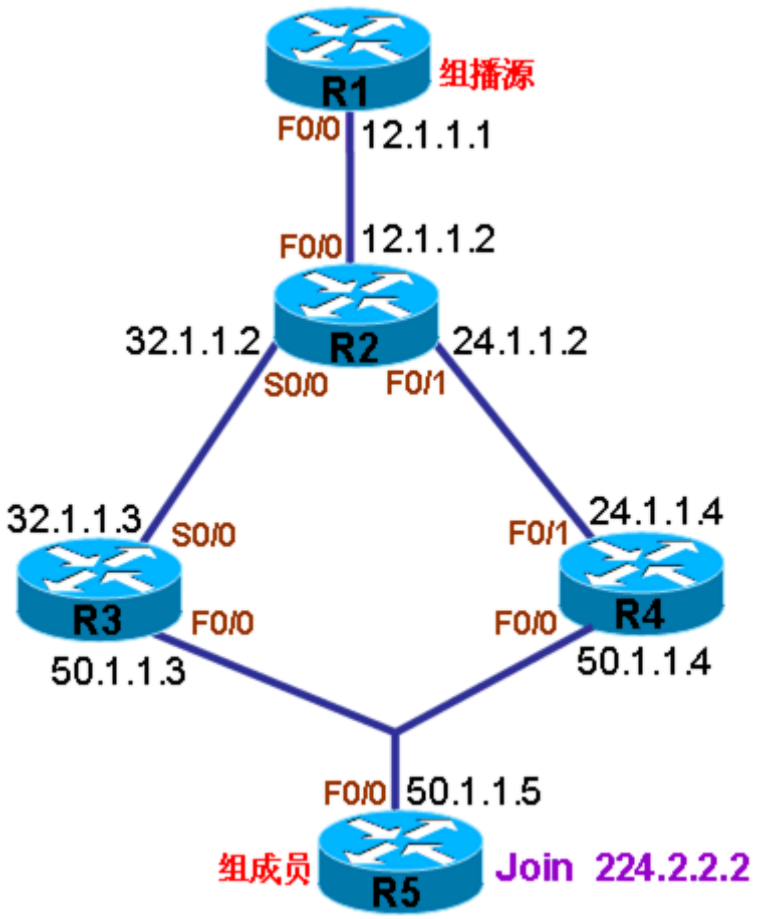
默认情况下，Cisco 路由器的组播路由功能是关闭的，所以在配置组播之间，需要在路由器上全局输入 ip multicast-routing 激活组播路由功能。

在配置组播之前，必须保证全网的单播路由是通的。

配置 PIM-DM

说明：以下图为例，其中 R1 为组播源，R5 为组成员，接收发往组 224.2.2.2 的数据。

鸿鹄培训+考试群：301283867



1 配置单播（此步略）

说明：全网配置 OSPF 来完成单播通信。

2 开启组播路由功能

（1）在每台路由器上开启组播路由功能

R1:

R1(config)#ip multicast-routing

R2:

R2(config)#ip multicast-routing

鸿鹄培训+考试群：301283867

R3:

R3(config)#ip multicast-routing

R4:

R4(config)#ip multicast-routing

R5:

R5(config)#ip multicast-routing

3 在接口上开启 PIM Dense-Mode

说明：在全网所有路由器的所有接口上开启 PIM，从而建立 PIM 邻居，要在所有接口开启，是因为要让 PIM 自己决定该在什么接口上转发组播，这样可以避免 RPF 检测失败，因为有可能只开部分接口，将导致接口不符合 RPF 接口而组播失败。

（1）在所有路由器的每个接口上开启 PIM:（只举例一个接口，其它接口配置相同）

R1(config)#int f0/0

R1(config-if)# ip pim dense-mode

4 查看 PIM 邻居

说明：相邻的两台路由器，接口上开启 PIM 后，将成为 PIM 邻居

（1）查看 R1 的 PIM 邻居

r1#show ip pim neighbor

PIM Neighbor Table

Neighbor Interface Uptime/Expires Ver DR

Address Prio/Mode

12.1.1.2 FastEthernet0/0 00:01:08/00:01:35 v2 1 / DR S

r1#

说明：R1 可以看到的 PIM 邻居有 R2,结果正常。因为双方的 DR 优先级相同，而 R2 的 IP 地址大于 R1，所以 R1 看到 R2 为 DR。

（2）查看 R2 的 PIM 邻居

r2#show ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Interface Uptime/Expires Ver DR

Address Prio/Mode

12.1.1.1 FastEthernet0/0 00:00:55/00:01:19 v2 1 / S

32.1.1.3 Serial0/0 00:01:04/00:01:40 v2 1 / DR S

24.1.1.4 FastEthernet0/1 00:00:52/00:01:22 v2 1 / DR S

r2#

说明：R2 可以看到的 PIM 邻居有 R1，R3，R4，结果正常。

（3）查看 R3 的 PIM 邻居

r3#show ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
32.1.1.2	Serial0/0	00:01:28/00:01:44	v2	1 / S
50.1.1.5	FastEthernet0/0	00:01:11/00:01:31	v2	1 / DR S
50.1.1.4	FastEthernet0/0	00:01:19/00:01:32	v2	1 / S

r3#

说明：R3 可以看到的 PIM 邻居有 R2，R4，R5，结果正常。

（4）查看 R4 的 PIM 邻居

r4#show ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
50.1.1.3	FastEthernet0/0	00:01:28/00:01:15	v2	1 / S
50.1.1.5	FastEthernet0/0	00:01:28/00:01:44	v2	1 / DR S
24.1.1.2	FastEthernet0/1	00:01:33/00:01:40	v2	1 / S

r4#

说明：R4 可以看到的 PIM 邻居有 R2，R3，R5，结果正常。

（5）查看 R5 的 PIM 邻居

r5#show ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
50.1.1.3	FastEthernet0/0	00:01:20/00:01:24	v2	1 / S
50.1.1.4	FastEthernet0/0	00:01:49/00:01:24	v2	1 / S

r5#

说明：R4 可以看到的 PIM 邻居有 R3，R4，结果正常。

5 将 R5 加入组 224.2.2.2

说明：组成员要加入组，使用的协议为 IGMP，而 Cisco 路由器开启 PIM 后，接口上自动开启 IGMP，所以 R5 在加入组时，无需额外开启 IGMP 而直接加入组 224.2.2.2

（1）配置 R5 加入组 224.2.2.2:

r5(config)#interface f0/0

r5(config-if)#ip igmp join-group 224.2.2.2

6 测试组播通信

说明：在没有组播数据之前，也就无法确认组播源的位置，在没有组播源的情况下，组播树是不能建立的。在测试 R1 向组 224.2.2.2 发送组播流量时，只需要 ping 224.2.2.2，只要数据包的目标地址为组播地址，就是使用组播方式来传递的。在 R1 上 ping 组播，那么 R1 就是组播源。

（1）从组播源 R1 上 ping 224.2.2.2

r1#ping 224.2.2.2

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.2.2.2, timeout is 2 seconds:

Reply to request 0 from 50.1.1.5, 8 ms

r1#

说明：从回包中看出，已经收到组成员 50.1.1.5 的回包，说明组播成功被转发到组成员 R5。

7 查看 IGMP 查询器

说明：因为 R3 和 R4 同时连接着同网段的组成员 R5，所以为了避免重复查询，会选举出一个 IGMP 查询器，网络中 IP 地址最小的路由器将成为查询器。

（1）查看网络中的 IGMP 查询器

r5#show ip igmp interface f0/0

FastEthernet0/0 is up, line protocol is up

Internet address is 50.1.1.5/24

IGMP is enabled on interface

Current IGMP host version is 2

Current IGMP router version is 2

IGMP query interval is 60 seconds

IGMP querier timeout is 120 seconds

IGMP max query response time is 10 seconds

Last member query count is 2

Last member query response interval is 1000 ms

Inbound IGMP access group is not set

IGMP activity: 2 joins, 0 leaves

Multicast routing is enabled on interface

Multicast TTL threshold is 0

Multicast designated router (DR) is 50.1.1.5 (this system)

IGMP querying router is 50.1.1.3

Multicast groups joined by this system (number of users):

224.0.1.40(1) 224.2.2.2(1)

r5#

说明：从结果中可以看出，R3 在网络中拥有最小地址 50.1.1.3，所以被选为 IGMP 查询器。

8 查看 PIM 前转器

说明：因为从组播源 R1 到组成员 R5 的路径中，有两条可选择，即经过 R3 或者经

过 R4，被选择的路由器就是 PIM 前转器，选择的规则为比较双方到组播源 R1 的单播 AD 值，其次是 metric 值，最后是 IP 地址最大的。

（1）查看 R3 和 R4 到组播源的单播路由表

R3：

r3#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

50.0.0.0/24 is subnetted, 1 subnets

C 50.1.1.0 is directly connected, FastEthernet0/0

32.0.0.0/24 is subnetted, 1 subnets

C 32.1.1.0 is directly connected, Serial0/0

24.0.0.0/24 is subnetted, 1 subnets

O 24.1.1.0 [110/2] via 50.1.1.4, 00:13:41, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/3] via 50.1.1.4, 00:13:41, FastEthernet0/0

r3#

R4：

r4#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

50.0.0.0/24 is subnetted, 1 subnets

C 50.1.1.0 is directly connected, FastEthernet0/0

32.0.0.0/24 is subnetted, 1 subnets

O 32.1.1.0 [110/65] via 50.1.1.3, 00:14:35, FastEthernet0/0

[110/65] via 24.1.1.2, 00:14:35, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

C 24.1.1.0 is directly connected, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/2] via 24.1.1.2, 00:14:35, FastEthernet0/1

r4#

说明：在 AD 值相同的情况下，因为 R4 到组播源的 metric 值为 2，而 R3 到源的 metric 值为 3,所以 R4 将被选为 PIM 前转器，最后结果将在组树路由表中体现出来。

9 分析组播树

说明：在组播路由表的输出中，将尽量只保留有用的信息给大家，其它信息将被抑制。

（1）查看 R1 的组播树情况

说明：查看组播树，即查看组播路由表

r1#show ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:07:14/00:02:15, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Dense, 00:07:14/00:00:00

r1#

说明：在组播源上，不会有任何组的记录信息。

（2）查看 R2 的组播树

说明：在 R2 上，即可看出 R3 和 R4 谁才是 PIM 前转器，前转器将成为组的出口。

r2#show ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.2.2.2), 00:06:18/stopped, RP 0.0.0.0, flags: D

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:06:18/00:00:00

Serial0/0, Forward/Dense, 00:06:18/00:00:00

FastEthernet0/0, Forward/Dense, 00:06:20/00:00:00

(12.1.1.1, 224.2.2.2), 00:01:57/00:02:56, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0

Outgoing interface list:

Serial0/0, Prune/Dense, 00:01:53/00:01:06, A

FastEthernet0/1, Forward/Dense, 00:01:57/00:00:00

(*, 224.0.1.40), 00:12:45/00:02:50, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:12:12/00:00:00

Serial0/0, Forward/Dense, 00:12:25/00:00:00

FastEthernet0/0, Forward/Dense, 00:12:45/00:00:00

r2#

说明：可以看出，PIM-DM 模式中同样会有（*，G）的记录，而所有正常的 PIM 接口都为输出接口，输入接口为空。在（S，G）中可以看出，进口为 F0/0，所以此接口就是组播源所在的接口，因此就是 RPF 接口，而 PIM 邻居为 0.0.0.0，表示对方邻居就是组播源。而在出口中，可以看到有两个出口，分别为 S0/0 和 F0/1，但是由于 R4 被选为 PIM 前转器，所以连 R4 的接口 F0/0 的状态就是 Forward 状态，而 S0/0 则被剪除了。

（3）查看 R4 的组播树

r4#show ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.2.2.2), 00:16:36/stopped, RP 0.0.0.0, flags: DC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:16:36/00:00:00

FastEthernet0/0, Forward/Dense, 00:16:36/00:00:00

(12.1.1.1, 224.2.2.2), 00:00:12/00:02:52, flags: T

Incoming interface: FastEthernet0/1, RPF nbr 24.1.1.2

Outgoing interface list:

FastEthernet0/0, Forward/Dense, 00:00:12/00:00:00

(*, 224.0.1.40), 00:19:08/00:02:01, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:19:04/00:00:00

FastEthernet0/0, Forward/Dense, 00:19:08/00:00:00

r4#

说明：从输出中看出，（*，G）的内容同 R3，而（S，G）中显示 R4 的 F0/1 为进口，而连着组成员的接口 F0/0 被标为 Forward，因此结果正常。

（4）查看 R2 的组播树

r3#show ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.2.2.2), 00:17:44/stopped, RP 0.0.0.0, flags: DC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Dense, 00:17:44/00:00:00

Serial0/0, Forward/Dense, 00:17:44/00:00:00

(12.1.1.1, 224.2.2.2), 00:01:19/00:01:46, flags: PT

Incoming interface: FastEthernet0/0, RPF nbr 50.1.1.4

Outgoing interface list:

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

Serial0/0, Prune/Dense, 00:01:20/00:01:39

(*, 224.0.1.40), 00:20:25/00:02:57, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Dense, 00:20:15/00:00:00

Serial0/0, Forward/Dense, 00:20:25/00:00:00

r3#

说明：（*，G）的内容同其它路由器，但是(S,G)中显示没有出口是 Forward，因此 R3 不提供组播转发，在网络中属正常。

10 改变 PIM 前转器

说明：因为 R3 和 R4 到组播源 R1 的单播路由表中，R4 拥有最小 metric 值，所以被选为 PIM 前转器，所有从组播源到组成员的流量均从 R4 被转发，要想控制从 R3 被转发，则可通过调整 R3 到组播源的单播 metric 值，只要比 R4 到组播源的 metric 值小就行。

（1）改变 R3 接口 S0/0 的 cost 值

r3(config)#int s0/0

r3(config-if)#ip ospf cost 1

（2）查看 R3 到组播源 R1 的 metric 值

r3#show ip route 12.1.1.0

Routing entry for 12.1.1.0/24

Known via "ospf 2", distance 110, metric 2, type intra area

Last update from 32.1.1.2 on Serial0/0, 00:01:37 ago

Routing Descriptor Blocks:

* 32.1.1.2, from 2.2.2.2, 00:01:37 ago, via Serial0/0

Route metric is 2, traffic share count is 1

r3#

说明： R3 到组播源 R1 的 metric 值为 2，同 R4 到组播源 R1 的 metric 值相同。

（3）查看 PIM 前转器结果：

r2#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.2.2.2), 00:13:59/stopped, RP 0.0.0.0, flags: D

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:13:59/00:00:00

Serial0/0, Forward/Dense, 00:13:59/00:00:00

FastEthernet0/0, Forward/Dense, 00:14:00/00:00:00

(12.1.1.1, 224.2.2.2), 00:00:59/00:02:54, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0

Outgoing interface list:

Serial0/0, Prune/Dense, 00:01:00/00:01:59

FastEthernet0/1, Forward/Dense, 00:01:00/00:00:00

(*, 224.0.1.40), 00:33:34/00:02:34, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:33:02/00:00:00

Serial0/0, Forward/Dense, 00:33:16/00:00:00

FastEthernet0/0, Forward/Dense, 00:33:36/00:00:00

r2#

说明：因为 R3 和 R4 到组播源 R1 的 metric 值相同，所以由最大 IP 地址的路由器成为 PIM 前转器，而 R3 和 R4 建 PIM 邻居的地址分别为 50.1.1.3 和 50.1.1.4，因为 R4 的地址比 R3 大，所以前转器仍然是路由器 R4。

（4）改大 R3 的 IP 地址

说明：只要将 R3 和 R4 建 PIM 邻居的地址改大后，方可成为 PIM 前转器

r3(config)#int f0/0

r3(config-if)#ip address 50.1.1.10 255.255.255.0

（5）查看 PIM 前转器结果：

r2#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.2.2.2), 00:16:24/stopped, RP 0.0.0.0, flags: D

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:16:24/00:00:00

Serial0/0, Forward/Dense, 00:16:24/00:00:00

FastEthernet0/0, Forward/Dense, 00:16:24/00:00:00

(12.1.1.1, 224.2.2.2), 00:03:23/00:00:31, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0

Outgoing interface list:

Serial0/0, Forward/Dense, 00:01:43/00:00:00

FastEthernet0/1, Prune/Dense, 00:00:25/00:02:34

FastEthernet0/1, Forward/Dense, 00:35:38/00:00:00

(*, 224.0.1.40), 00:35:56/00:02:39, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Dense, 00:35:23/00:00:00

Serial0/0, Forward/Dense, 00:35:36/00:00:00

FastEthernet0/0, Forward/Dense, 00:35:56/00:00:00

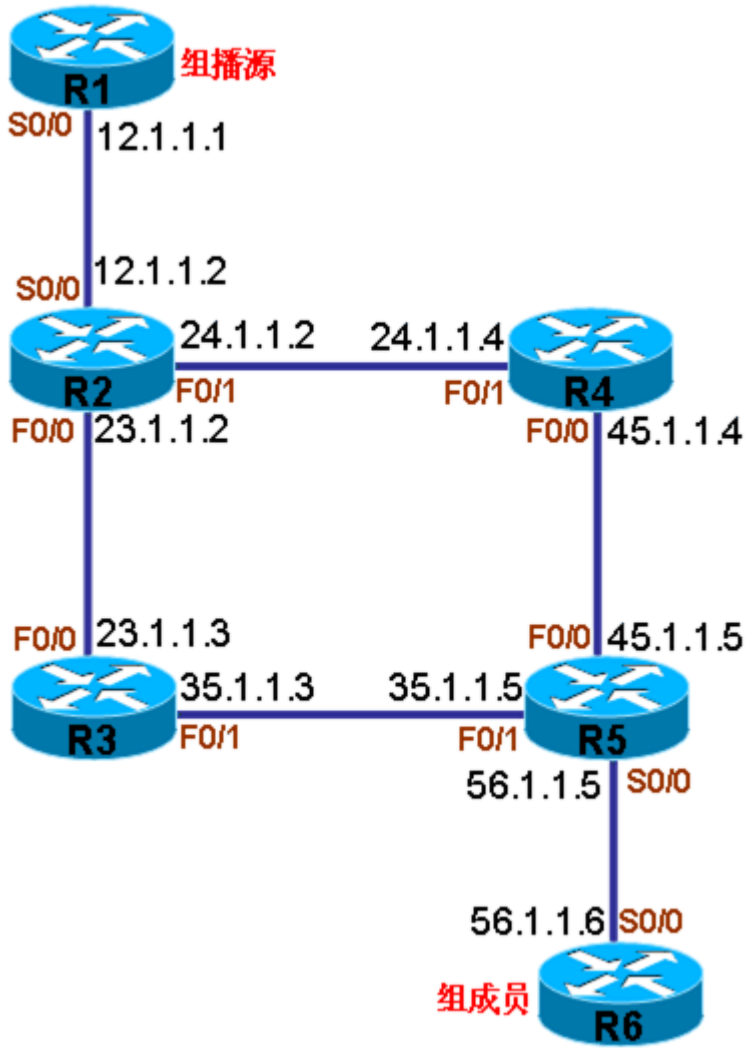
r2#

说明：因为 R3 的地址已经改为 50.1.1.10，而 R4 的地址为 50.1.1.4，所以在 AD 和 metric 值相同的情况下，R3 的 IP 地址最大，所以被选为 PIM 前转器。

配置 PIM-SM

说明：在下图中，R1 为组播源，R6 为组成员，接收发往组 224.6.6.6 的数据。

每台路由器均配有 loopback 地址，分别为 R1：1.1.1.1，R2：2.2.2.2，R3：3.3.3.3，R4：4.4.4.4，R5：5.5.5.5，R6：6.6.6.6，因为配置 PIM-SM 模式，所以网络中必须存在 RP。



1 配置单播（此步略）

说明：全网配置 OSPF 来完成单播通信。

2 开启组播路由功能

（1）在每台路由器上开启组播路由功能

R1:

R1(config)#ip multicast-routing

R2:

R2(config)#ip multicast-routing

R3:

R3(config)#ip multicast-routing

R4:

R4(config)#ip multicast-routing

R5:

R5(config)#ip multicast-routing

R6:

R6(config)#ip multicast-routing

3 在接口上开启 PIM Sparse-Mode，并建立 PIM 邻居。

说明：在全网所有路由器的所有接口上开启 PIM，从而建立 PIM 邻居，要在所有接口开启，是因为要让 PIM 自己决定该在什么接口上转发组播，这样可以避免 RPF 检测失败，因为有可能只开部分接口，将导致接口不符合 RPF 接口而组播失败。

（1）在所有路由器的每个接口上开启 PIM:（只举例一个接口，其它接口配置相同）

R1(config)#int f0/0

R1(config-if)# ip pim sparse-mode

4 将 R6 加入组 224.6.6.6

（1）配置 R6 加入组 224.6.6.6

r6(config)#int s0/0

r6(config-if)#ip igmp join-group 224.6.6.6

5 静态配置 R4 为 RP

说明：将 R4 的 loopback 地址 4.4.4.4 配置为 RP，静态配置 RP 时，RP 地址的接口可以不用开启 PIM。RP 的配置方法有多种，此处首先采用手工静态配置方法，并且需要在每一台路由器上手工配置静态 RP。

（1）在 R1 上静态配置 R4 为 RP （每台路由器的配置同 R1）

R1(config)#ip pim rp-address 4.4.4.4

说明：配置后面不跟 ACL 限制，默认为所有组的 RP。

（2）查看 PIM 路由器的 RP 情况 （所有路由器的结果都应该相同）

r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static

RP: 4.4.4.4 (?)

r1#

说明：R1 已经正常看到所有组的 RP 为 4.4.4.4。

6 查看 RP 到组成员的组播树情况

说明：因为 PIM-SM 模式中，组成员会主动加入组，所以 RP 到组成员之间在没有源的情况下，也会形成组播树。

（1）在 RP：R4 上查看组播路由表：

r4#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.6.6.6), 00:02:16/00:03:07, RP 4.4.4.4, flags: S

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:02:16/00:03:07

(*, 224.0.1.40), 00:06:13/00:03:06, RP 4.4.4.4, flags: SJCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:01:35/00:02:53

FastEthernet0/0, Forward/Sparse, 00:02:22/00:03:05

r4#

说明：可以看到，RP 到组成员方向的接口已经被标为 Forward 状态，所以 RP 在有组播流量的情况下，是会向组成员发送的。

（2）查看 R5 上的组播路由表：

r5#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.6.6.6), 00:05:36/00:02:45, RP 4.4.4.4, flags: SJC

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:02:41/00:02:45

(*, 224.0.1.40), 00:06:26/00:02:47, RP 4.4.4.4, flags: SJCL

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:01:55/00:02:33

Serial0/0, Forward/Sparse, 00:02:42/00:02:45

r5#

说明：R5 上连 RP 的接口 F0/0 就是组播的进口，而连组成员 R6 的接口 S0/0 已经是出口，状态为 Forward，所以正常。

注：从 PIM-SM 的组播路由表中看出，只记录 Forward 状态的出口，被剪除的出口是不做记录的，而 PIM-DM 会将所有出口记录下来，其中包括被剪除的接口。

6 测试组播通信

（1）从组播源 R1 上 ping 224.6.6.6

r1#ping 224.6.6.6

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.6.6.6, timeout is 2 seconds:

Reply to request 0 from 56.1.1.6, 128 ms

r1#

说明：从回包中看出，已经收到组成员 56.1.1.6 的回包，说明组播成功被转发到组成员 R6。

7 查看组播树的情况：

（1）查看 R1 的组播树情况

r1#show ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:10:04/00:02:52, RP 4.4.4.4, flags: SJPL

Incoming interface: Serial0/0, RPF nbr 12.1.1.2

Outgoing interface list: Null

r1#

说明：在组播源上，不会有任何组的记录信息。

（2）查看 R2 的组播树

r2#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.6.6.6), 00:02:05/stopped, RP 4.4.4.4, flags: SPF

Incoming interface: FastEthernet0/1, RPF nbr 24.1.1.4

Outgoing interface list: Null

(12.1.1.1, 224.6.6.6), 00:02:05/00:03:24, flags: FT

Incoming interface: Serial0/0, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:02:05/00:03:22

(*, 224.0.1.40), 00:10:33/00:02:25, RP 4.4.4.4, flags: SJCL

Incoming interface: FastEthernet0/1, RPF nbr 24.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:10:33/00:01:59

r2#

说明：可以看出，（*，G）中说明了 RP 是 4.4.4.4，但是组播源到 RP 之间会创建（S，G）的记录，并且通往 RP 的路径正常。RPF 邻居为 0.0.0.0，也正常。

（3）再次查看 RP 上的组播树

r4#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.6.6.6), 00:07:39/00:02:39, RP 4.4.4.4, flags: S

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:07:39/00:02:39

(12.1.1.1, 224.6.6.6), 00:03:35/00:02:01, flags: T

Incoming interface: FastEthernet0/1, RPF nbr 24.1.1.2

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:03:35/00:02:50

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

(*, 224.0.1.40), 00:11:37/00:02:58, RP 4.4.4.4, flags: SJCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:07:00/00:03:22

FastEthernet0/0, Forward/Sparse, 00:07:45/00:02:38

r4#

说明：RP 上也有了（S，G）的记录，并且接口状态都是正常。

（4）再次查看 R5 上的组播树

r5#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.6.6.6), 00:11:55/00:03:21, RP 4.4.4.4, flags: SJC

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:09:00/00:03:21

(12.1.1.1, 224.6.6.6), 00:04:55/00:02:11, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:04:55/00:03:21

(*, 224.0.1.40), 00:12:47/00:03:21, RP 4.4.4.4, flags: SJCL

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:08:16/00:03:08

Serial0/0, Forward/Sparse, 00:09:01/00:03:21

r5#

说明：同样也有（S，G）的记录，方面后面从 RPT 切换到 SPT，正常。

（5）再次查看 R3 上的组播树

r3#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:15:23/00:02:22, RP 4.4.4.4, flags: SJCL

Incoming interface: FastEthernet0/1, RPF nbr 35.1.1.5

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:15:23/00:02:22

r3#

说明：因为 R3 不需要转发组播，所以没有任何记录。

注：IGMP 查询器，PIM 前转器不再查看和控制，方法同 PIM-DM 模式。

配置 Auto-RP

说明：以上配置为静态 RP 的方式，下面配置 Auto-RP

1 配置 C-RP

说明：配置为 RP 的接口必须开启 PIM。

（1）配置 R3 为 C-RP：

r3(config)#access-list 24 permit 224.5.5.5

r3(config)#access-list 24 permit 224.6.6.6

r3 (config)#int loopback 0

r3 (config-if)# ip pim sparse-mode

r3(config)#ip pim send-rp-announce loopback 0 scope 10 group-list 24

说明：配置 R3 的 loopback 0 为 RP 地址，并且使用 ACL 限制只做组 224.5.5.5 和 224.6.6.6 的 RP。

（2）配置 R4 为 C-RP：

R4(config)#access-list 24 permit 224.5.5.5

R4(config)#access-list 24 permit 224.6.6.6

R4 (config)#int loopback 0

R4 (config-if)# ip pim sparse-mode

R4(config)#ip pim send-rp-announce loopback 0 scope 10 group-list 24

说明：配置 R4 的 loopback 0 为 RP 地址，并且使用 ACL 限制只做组 224.5.5.5 和 224.6.6.6 的 RP。

2 配置 RP-mapping agent

（1）配置 R1 为 RP-mapping agent

R1 (config)#int loopback 0

R1 (config-if)# ip pim sparse-mode

R1(config)#ip pim send-rp-discovery loopback 0 scope 16

说明：配置 R1 的 loopback 0 为 RP-mapping agent。

3 查看 RP 信息

说明：所有 C-RP 的竞选消息都发送到 224.0.1.39，由 RP-mapping agent 接收后将选为 RP 的地址公布给所有 PIM 路由器，要让 RP-mapping agent 成功收到 C-RP 的竞选消息，就必须能够接收组播到 224.0.1.39 的数据。

（1）查看 RP-mapping agent 的组加入情况

r1#show ip igmp interface

Serial0/0 is up, line protocol is up

Internet address is 12.1.1.1/24

IGMP is enabled on interface

Current IGMP host version is 2

Current IGMP router version is 2

IGMP query interval is 60 seconds

IGMP querier timeout is 120 seconds

IGMP max query response time is 10 seconds

Last member query count is 2

Last member query response interval is 1000 ms

Inbound IGMP access group is not set

IGMP activity: 2 joins, 0 leaves

Multicast routing is enabled on interface

Multicast TTL threshold is 0

Multicast designated router (DR) is 12.1.1.2

IGMP querying router is 12.1.1.1 (this system)

Multicast groups joined by this system (number of users):

224.0.1.39(1)

Loopback0 is up, line protocol is up

Internet address is 1.1.1.1/24

IGMP is enabled on interface

Current IGMP host version is 2

Current IGMP router version is 2

IGMP query interval is 60 seconds

IGMP querier timeout is 120 seconds

IGMP max query response time is 10 seconds

Last member query count is 2

Last member query response interval is 1000 ms

Inbound IGMP access group is not set

IGMP activity: 2 joins, 0 leaves

Multicast routing is enabled on interface

Multicast TTL threshold is 0

Multicast designated router (DR) is 1.1.1.1 (this system)

IGMP querying router is 1.1.1.1 (this system)

Multicast groups joined by this system (number of users):

224.0.1.40(1) 224.0.1.39(1)

r1#

说明：可以看出，配置成为 RP-mapping agent 后，所有的接口都已加入了 224.0.1.39，但是这并不能保证 RP-mapping agent 就一定能够收到 C-RP 的竞选消息，因为 PIM-SM 模式下，没有 RP 时，组播是不通的。

4 解决 RP-mapping agent 接收 C-RP 竞选消息

说明：因为组播不通，所以当 RP-mapping agent 和 C-RP 不是同一台路由器时，无法接收 C-RP 的竞选消息，可以通过在路由器上配置 Autorp listener 来解决，网络中最好所有路由器都配置。

（1）配置 Autorp listener（此处只举例配置 R1，所有路由器配置相同）

R1 (config)#ip pim autorp listener

注：解决方法还有就是将 PIM 的模式都改为 sparse-dense-mode，此模式配置省略。

5 查看 RP 情况

（1）查看 R1 的 RP 情况：

```
r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP-mapping agent (Loopback0)



Group(s) 224.5.5.5/32

  RP 4.4.4.4 (?), v2v1

    Info source: 4.4.4.4 (?), elected via Auto-RP

         Uptime: 00:05:49, expires: 00:02:11

  RP 3.3.3.3 (?), v2v1

    Info source: 3.3.3.3 (?), via Auto-RP

         Uptime: 00:05:14, expires: 00:02:43

Group(s) 224.6.6.6/32

  RP 4.4.4.4 (?), v2v1

    Info source: 4.4.4.4 (?), elected via Auto-RP

         Uptime: 00:05:49, expires: 00:02:10

  RP 3.3.3.3 (?), v2v1

    Info source: 3.3.3.3 (?), via Auto-RP

         Uptime: 00:05:14, expires: 00:02:45

r1#
```

说明：在 RP-mapping agent 上，同时看到所有信息，因为 R4 的地址最大，所以被

选举为两个组的 RP。

（2）查看 R5 的 RP 情况：

r5#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.5.5.5/32

RP 4.4.4.4 (?), v2v1

Info source: 1.1.1.1 (?), elected via Auto-RP

Uptime: 00:01:25, expires: 00:02:31

Group(s) 224.6.6.6/32

RP 4.4.4.4 (?), v2v1

Info source: 1.1.1.1 (?), elected via Auto-RP

Uptime: 00:01:25, expires: 00:02:32

r5#

说明：R5 正常学习到 RP 信息，其它路由器都应该正常学习到 RP 信息。

6 测试组播通信

（1）测试 R1 到各组的通信情况：

r1#ping 224.5.5.5

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.5.5.5, timeout is 2 seconds:

Reply to request 0 from 56.1.1.6, 129 ms

Reply to request 0 from 56.1.1.6, 165 ms

r1#ping 224.6.6.6

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.6.6.6, timeout is 2 seconds:

Reply to request 0 from 56.1.1.6, 124 ms

Reply to request 0 from 56.1.1.6, 152 ms

r1#

说明：R1 到组 224.5.5.5 和 224.6.6.6 通信正常。

7 限制 RP 竞选消息

说明：在 C-RP 上可以控制只竞选特定组的 RP，而在 RP-mapping agent 上，可以配置只接收某个 RP 竞选某特定组的消息，那么其它消息都将被过滤。

（1）配置 RP-mapping agent 限制 R4 的竞选消息

说明：配置 RP-mapping agent 只接收 R4 竞选 224.6.6.6 的消息，那么 R4 也就不可能竞选组 224.5.5.5 的 RP 了。

r1(config)#access-list 4 permit 4.4.4.4

```
r1(config)#access-list 6 permit 224.6.6.6

r1(config)#ip pim rp-announce-filter rp-list 4 group-list 6
```

8 查看 RP 情况

（1）在 R1 上查看 RP 信息：

```
r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP-mapping agent (Loopback0)


Group(s) 224.5.5.5/32

  RP 3.3.3.3 (?), v2v1

    Info source: 3.3.3.3 (?), elected via Auto-RP

        Uptime: 00:02:42, expires: 00:02:17

Group(s) 224.6.6.6/32

  RP 4.4.4.4 (?), v2v1

    Info source: 4.4.4.4 (?), elected via Auto-RP

        Uptime: 00:08:18, expires: 00:02:40

  RP 3.3.3.3 (?), v2v1

    Info source: 3.3.3.3 (?), via Auto-RP

        Uptime: 00:07:43, expires: 00:02:17

r1#
```

说明：因为 R4 竞选组 224.5.5.5 的消息被过滤了，所以组 224.5.5.5 的 RP 只有 R3。

（2）在 R2 上查看 RP 信息：

r2#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.5.5.5/32

RP 3.3.3.3 (?), v2v1

Info source: 1.1.1.1 (?), elected via Auto-RP

Uptime: 00:00:59, expires: 00:02:57

Group(s) 224.6.6.6/32

RP 4.4.4.4 (?), v2v1

Info source: 1.1.1.1 (?), elected via Auto-RP

Uptime: 00:20:01, expires: 00:02:56

r2#

说明：R2 上的 RP 信息和预料中相同。

配置 BSR

说明：在 BSR 中，在不修改 IP 地址的情况下，可以通过修改优先级来控制竞选。

1 配置 C-RP

（1）配置 R3 为 C-RP，优先级为 10（数字越大，优先级越高）

R3(config)#access-list 24 permit 224.5.5.5

R3(config)#access-list 24 permit 224.6.6.6

R3(config)#int loopback 0

R3 (config-if)# ip pim sparse-mode

R3(config)#ip pim rp-candidate loopback 0 group-list 24 priority 10

(2) 配置 R4 为 C-RP，优先级默认为 0

R4(config)#access-list 24 permit 224.5.5.5

R4(config)#access-list 24 permit 224.6.6.6

R4(config)#int loopback 0

R4 (config-if)# ip pim sparse-mode

R4(config)#ip pim rp-candidate loopback 0 group-list 24

2 配置 C-BSR

（1）配置 R1 为 C-BSR

R1(config)#int loopback 0

R1 (config-if)# ip pim sparse-mode

R1(config)#ip pim bsr-candidate loopback 0

3 查看 RP 情况

（1）查看 BSR 情况

```
r1#sh ip pim bsr-router
PIMv2 Bootstrap information
This system is the Bootstrap Router (BSR)
  BSR address: 1.1.1.1 (?)
  Uptime:      00:01:45, BSR Priority: 0, Hash mask length: 0
  Next bootstrap message in 00:00:14
r1#
```

说明：因为 C-BSR 只有 R1，所以被选为 BSR 的也是 R1。

（2）查看 RP 结果：

```
r1#sh ip pim rp mapping
PIM Group-to-RP Mappings
This system is the Bootstrap Router (v2)

Group(s) 224.5.5.5/32
  RP 3.3.3.3 (?), v2
    Info source: 23.1.1.3 (?), via bootstrap, priority 10
         Uptime: 00:13:25, expires: 00:02:08
```

RP 4.4.4.4 (?), v2

Info source: 24.1.1.4 (?), via bootstrap, priority 20

Uptime: 00:09:43, expires: 00:02:00

Group(s) 224.6.6.6/32

RP 3.3.3.3 (?), v2

Info source: 23.1.1.3 (?), via bootstrap, priority 10

Uptime: 00:13:25, expires: 00:02:07

RP 4.4.4.4 (?), v2

Info source: 24.1.1.4 (?), via bootstrap, priority 20

Uptime: 00:12:43, expires: 00:01:58

r1#

说明：通过查看 RP，无法看出 BSR 选择了谁成为 RP。

（3）查看其它路由器的 RP 结果：

r2#sh ip pim rp mapping in-use

PIM Group-to-RP Mappings

Group(s) 224.5.5.5/32

RP 3.3.3.3 (?), v2

Info source: 1.1.1.1 (?), via bootstrap, priority 10, holdtime 98

Uptime: 00:14:10, expires: 00:01:19

RP 4.4.4.4 (?), v2

Info source: 1.1.1.1 (?), via bootstrap, priority 20, holdtime 91

Uptime: 00:10:28, expires: 00:01:14

Group(s) 224.6.6.6/32

RP 3.3.3.3 (?), v2

Info source: 1.1.1.1 (?), via bootstrap, priority 10, holdtime 98

Uptime: 00:14:10, expires: 00:01:20

RP 4.4.4.4 (?), v2

Info source: 1.1.1.1 (?), via bootstrap, priority 20, holdtime 89

Uptime: 00:13:27, expires: 00:01:12

Dynamic (Auto-RP or BSR) RPs in cache that are in use:

Group(s): 224.6.6.6/32, RP: 3.3.3.3, expires: 00:00:51

Group(s): 224.5.5.5/32, RP: 3.3.3.3, expires: 00:00:53

r2#

说明：在 R2 上看出，因为 C-RP 的数字越 小，优先级越高，所以 RP 为 3.3.3.3。

PIM-SM 的 NBMA Mode

说明：因为帧中继中没有内置的处理广播和组播的能力，所以在收到广播和组播时，会在所有配了关键字 broadcast 的 PVC 上转发，这样当帧中继接口对端有多个邻居，而只有部分邻居要接收组播时，那么组播的被当作广播转发的行为，将影响不需要接收组播的路由器，因此需要配置 NBMA 模式来跟踪要接收组播的邻居，从而是只将组播发送给要接收的邻居。NBMA 模式只支持 PIM-SM 模式，因为是跟踪 PIM-SM 模式的 join 消息。帧中继下的解决方法除了 NBMA Mode 之外，还有配置子接口，

子接口类似多个物理接口，会建立多个邻居，此处不对子接口作过多解释。

没配之前:

1.查看 R5 没有配置 NBMA Mode 的组播树

r5#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.6.6.6), 01:53:44/00:03:13, RP 3.3.3.3, flags: SCF

Incoming interface: FastEthernet0/1, RPF nbr 35.1.1.3

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:01:16/00:03:13

Serial0/0, Prune/Sparse, 00:01:17/00:00:00

(1.1.1.1, 224.6.6.6), 00:00:48/00:03:24, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:00:49/00:03:12

(12.1.1.1, 224.6.6.6), 00:00:49/00:03:23, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:00:49/00:03:12

(*, 224.5.5.5), 01:16:35/00:03:00, RP 3.3.3.3, flags: SC

Incoming interface: FastEthernet0/1, RPF nbr 35.1.1.3

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:00:30/00:02:59

Serial0/0, Prune/Sparse, 00:01:21/00:00:00

(1.1.1.1, 224.5.5.5), 00:02:04/00:02:41, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:01:05/00:03:22

Serial0/0, Prune/Sparse, 00:01:22/00:00:00

(12.1.1.1, 224.5.5.5), 00:02:04/00:02:41, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, Forward/Sparse, 00:01:05/00:03:24

Serial0/0, Prune/Sparse, 00:01:22/00:00:00

(*, 224.0.1.40), 00:55:08/00:02:59, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Loopback0, Forward/Sparse, 00:55:09/00:00:00

Serial0/0, Forward/Sparse, 00:55:09/00:00:00

FastEthernet0/1, Forward/Sparse, 00:55:10/00:00:00

FastEthernet0/0, Forward/Sparse, 00:55:10/00:00:00

r5#

说明：从结果中看出，因为没有配置 NBMA Mode，所以 R5 将收到的组播从 S0/0 的每一条 PVC 发出去。（虽然本例中只配了一条 PVC）

2.配置 R5 连组成员 R6 的帧中继接口 S0/0 为 NBMA Mode(最好双方都配置)

r5(config)#int s0/0

r5(config-if)#ip pim nbma-mode

3.查看 R5 配置 NBMA Mode 之后的组播树

r5#sh ip mroute

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.6.6.6), 01:54:34/00:03:22, RP 3.3.3.3, flags: SCF

Incoming interface: FastEthernet0/1, RPF nbr 35.1.1.3

Outgoing interface list:

Serial0/0, 56.1.1.6, Forward/Sparse, 00:00:07/00:03:22

(1.1.1.1, 224.6.6.6), 00:01:38/00:02:34, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, 56.1.1.6, Forward/Sparse, 00:00:07/00:03:22

(12.1.1.1, 224.6.6.6), 00:01:39/00:02:33, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, 56.1.1.6, Forward/Sparse, 00:00:09/00:03:20

(*, 224.5.5.5), 01:17:25/00:03:09, RP 3.3.3.3, flags: SC

Incoming interface: FastEthernet0/1, RPF nbr 35.1.1.3

Outgoing interface list:

Serial0/0, 56.1.1.6, Forward/Sparse, 00:00:20/00:03:09

(1.1.1.1, 224.5.5.5), 00:02:56/00:03:20, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, 56.1.1.6, Forward/Sparse, 00:00:26/00:03:29

(12.1.1.1, 224.5.5.5), 00:03:02/00:03:13, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 45.1.1.4

Outgoing interface list:

Serial0/0, 56.1.1.6, Forward/Sparse, 00:00:32/00:03:25

(*, 224.0.1.40), 00:56:07/00:02:55, RP 0.0.0.0, flags: DCL

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Loopback0, Forward/Sparse, 00:56:07/00:00:00

Serial0/0, Forward/Sparse, 00:56:11/00:00:00

FastEthernet0/1, Forward/Sparse, 00:56:11/00:00:00

FastEthernet0/0, Forward/Sparse, 00:56:11/00:00:00

r5#

说明：R5 在帧中继接口上配置了 NBMA Mode，所以看到组播树中，明确显示组播将会发给特定的邻居（56.1.1.6），由于在组播树中记录了对端邻居的地址，所以在接口对端有多台路由器时，组播不会发给不想接收组播的路由器。

Source Specific Multicast（SSM）

概述

在组播树中，被记录为（*，G）的组条目表示，对于一个特定的组，任何主机都可以向该组发起组播流量，也就是说，一个组可以拥有多个组播源，任何组播源都可以发送组播流量，这样的组播被称为任意源组播 Any Source Multicast (ASM)。

因为一个组一般表示一个应用，如果网络中两个应用使用了同一个组地址，这样就会造成组成员将两个不同应用的流量误当作同一个应用来处理，就会造成数据的混乱或错误，所以当两个应用不小心使用了同一个组地址，这样会给应用带来问题。

如果一个组成员想要接收某个组的组播流量，可以通过 IGMP 向路由器报告，例如 IGMP ver1，IGMP ver2，报告中指出了组成员想要接受的组地址，当路由器收到 IGMP 报告之后，就会将发往相应组地址的流量转发到组成员。

正因为路由器会将任何组播源发到同一个组的流量转发给相同组成员，所以可能会造成多个应用使用同一个组地址时，不能只将组成员想要接收的流量发到组成员。如果要实现只将特定组播源发来的流量转发给相应的组成员，那么这样的组播机制，被称为特定源组播 Source Specific Multicast(SSM)。

因为特定源组播（SSM）只将特定的组播源发来的流量，而不是任何源发来的流量转发给组成员，所以组成员在向路由器报告自己想要接受的组播流量时，除了明确指出组地址之外，还必须指出组播源地址，而这样的 IGMP 报告，需要 IGMP ver3 来支持。

并且可以想象，如果多个应用程序在同一个源，那么就要多个组，但如果多个应用在不同源，那么组地址就可以相同，也可以不同，因为 SSM 可以根据源地址区分出不同的应用程序。

在运行 SSM 时，需要两个组件

- **Protocol Independent Multicast source-specific mode (PIM-SSM)**
- **Internet Group Management Protocol Version 3 (IGMPv3)**

其中 IGMP ver3 可以代替 ver 1 和 ver 2 的功能，但是与 ver 1 和 ver 2 不同之处在于，IGMP ver 3 支持对源地址的过滤，IGMP ver 3 在报告中，会明确指出想要接收的组播源地址。

SSM 基于(S, G) 传输， 但 SSM 也可以和其它组播树共存，只要配好自己的组地址范围即可，为 SSM 保留地址范围是：

232.0.0.0 - 232.255.255.255 （232.0.0.0/8）

但思科 IOS 可任意配地址范围。

虽然如此，要先有 PIM SM，才能有 SSM，但 SSM 也可以独立存在。

如果已经有了 PIM-SM，那么只有最后一跳路由器需要开启 SSM 即可，也就是说只需要直接连接着组成员，直接接收组成员 IGMP 数据包的路由器需要开启 SSM。

注：SSM 不需要 RP，当最后一跳路由器开启 SSM 后，正常的 PIM-SM 就失去意义了。

配置 SSM

说明：只在直接连接着组成员，直接接收组成员 IGMP 数据包的路由器 上开启 SSM

1.全局开启组播

（1）全局开启组播路由功能

```
Router(config)#ip multicast-routing
```

2.配置 SSM 组地址范围

（1）配置默认的 SSM 组地址范围

```
Router(config)#ip pim ssm default
```

说明：配置此命令后，默认的 SSM 组地址范围为：232.0.0.0 - 232.255.255.255

（2）配置 SSM 组地址为 232.1.1.1

```
Router(config)#access-list 1 permit 232.1.1.1

Router(config)#ip pim ssm range 1
```

说明：ACL 1 所匹配的地址即为 SSM 的组地址范围。

3.在接口开启 SSM

（1）在接口开启 PIM

```
Router(config-if)#ip pim sparse-mode
```

说明：必须在接口开启 sparse-mode 或 sparse-dense-mode

（2）在接口开启 IGMP ver 3

```
Router(config-if)#ip igmp version 3
```

说明：默认为 IGMP ver 2.

附：SSM 查看命令

```
Router# show ip igmp groups

Router# show ip mroute
```

MSDP (Multicast Source Discovery Protocol)

PIM 概述

通过路由器组建的网络需要通信，前提是所有路由器上都必须拥有正确的路由表，对于单播是这样子，而如果需要使用组播通信；同样道理，所有路由器上都必须拥有正确的组播路由表，准确的说，是必须拥有正确的组播树，组播树从组播源到组成员（组播接收者）之间建立，组播源发出的数据沿着组播树流向组成员，从而保证组播通信正常。换句话说就是，如果没有组播树，或组播树不正常，那么组播是不能通信的。

组播路由器需要靠组播路由协议的配合来建立组播树，目前最为流行的组播路由协议为 Protocol Independent Multicast (PIM)，PIM 可工作在两种模式下，分别为 PIM dense mode (PIM-DM)和 PIM sparse mode (PIM-SM)，PIM-DM 就是一个纯傻瓜式的工作方式，它就像咱们平时见到的傻瓜式交换机一样，交换机打开之后不需要做任何配置就能用，并且谁都会用，PIM-DM 只要在相应的路由器的相应接口上开启之后，就算是大功告成，组播源只要一发数据，组播树铁定就能正确建立，

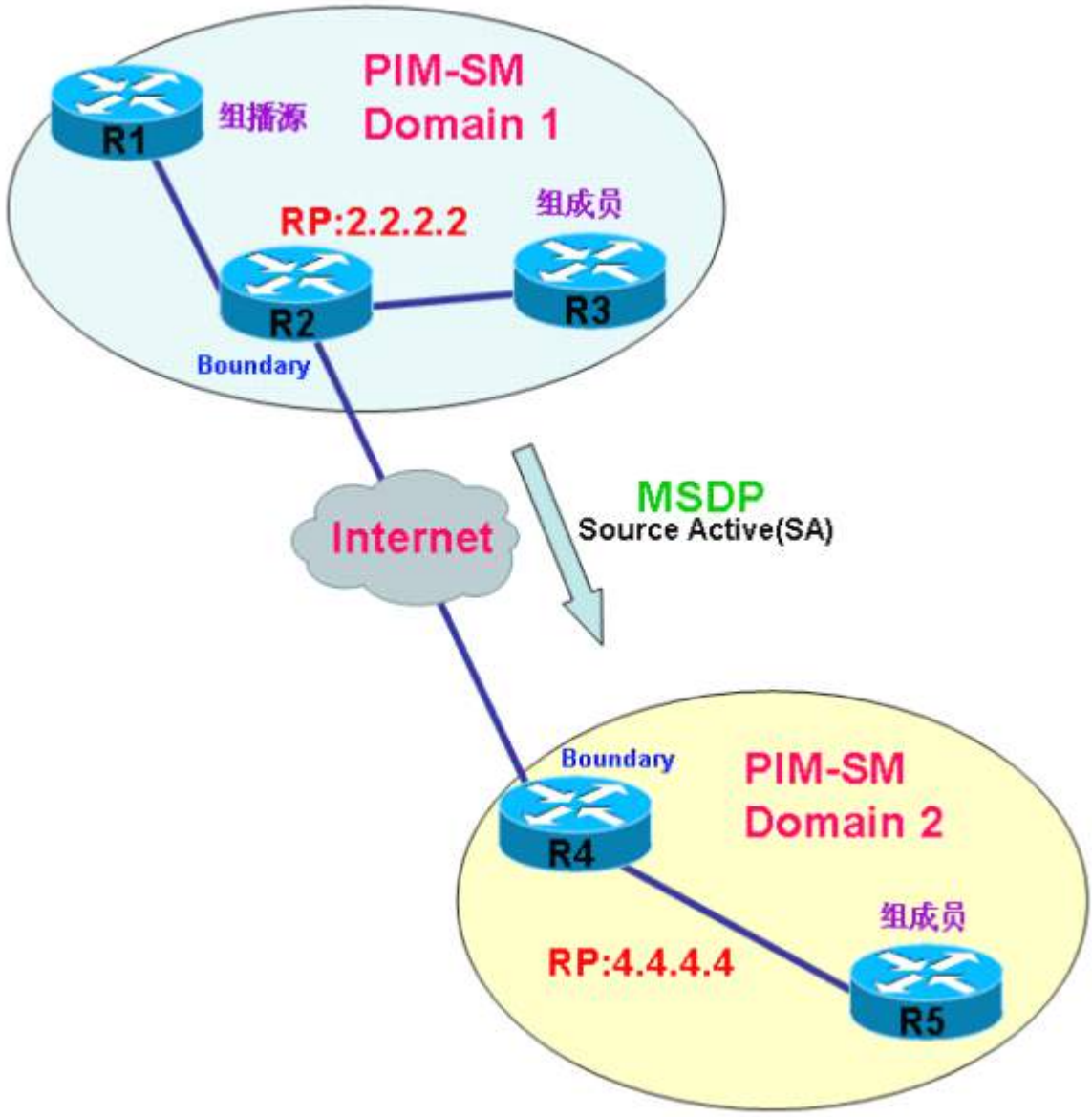
组成员也一定能够收到数据，相当快捷方便；但遗憾的是 PIM-DM 虽然简洁，但缺乏控制力度，就相当于傻瓜式交换机虽然好用，但你无法在上面做太多的策略来控制流量，并且更遗憾的是在 CCIE 考试中，基本上不考你 PIM-DM。

除 PIM-DM 之外，PIM-SM 是个重点，也是个难点，并且还是个考点，所以我们通常在解决组播问题的时候，可以说都是在解决 PIM-SM 的问题。在 PIM-SM 中，所有的流量都不是组播源直接发向组成员的，组播流量是由组播源发给 Rendezvous Point (RP)，然后 RP 再将流量转发给组成员，如果说没有了 RP，那么组播源也就不知道把流量该发给谁，而也就再也没有谁会给组成员发流量了，所以 RP 是否正常，直接关系到组播是否能通信。PIM 在建立组播树时，在 PIM-SM 中，这棵树是围绕 RP 来建立的，所以 RP 就是组播树的根，要完成组播树的建立，RP 必须要知道谁是组播源以及谁是组成员，这一点请牢记；那么 RP 是如何知道谁是组播源以及谁是组成员的呢？这一点也请牢牢地记住，过程是：组播源和组成员主动告诉 RP，因为 RP 是不会主动问它们的，事实上真正告诉 RP 的是离组播源最近的路由器以及离组成员最近的路由器，无论组播源和组成员本身是不是路由器都是这样，为什么要由路由器代为发起呢？因为通常组播员和组成员是一台 PC 或服务器，只是我们在做实验时，是用了路由器，而 PC 或服务器准确地讲是不会运行 PIM 协议的；组播源告诉 RP 自己是组播源的这个过程称为 register(注册)，而组成员告诉 RP 自己是组成员的这个过程称为 report(报告)，但也可以称为注册，Cisco 文档上也有时会把它们都称为是注册。

一个正常工作的组播网络中应该只有一个 RP，这样的只围绕一个 RP 建立组播树的组播网络被称为一个 PIM-SM 域。在某些情况下，比如一个大公司在全国各个城市都有分公司，而这个公司许多应用是靠组播来传递的，如视频会议，如果这个公司全国范围内都设计成一个 PIM-SM 域，那么就表示全国的分公司都需要使用一个共同的 RP，所以如果这个 RP 是放在北京分公司的，那么北京的这台 RP 路由器是不能关的，可能会面临着其它各个城市在开会时出现任何通信问题都往北京打电话讯问 RP 路由器是否运行正常或已被关闭；对于各个分公司的 IT 人员来说也不希望如此，他们一定是希望 RP 路由器放在自己的分公司或者自己的分公司有自己独立可控的 RP 路由器，这样就不需要依靠于其它分公司的 RP 来通信，便于在出现通信故障时能够全面地排错。ISP 的某些业务也是通过组播来开展的，并且这些业务可能需要跨 ISP 来工作，可想而知，如果将所有 ISP 都规划成一个 PIM-SM 域，势必会造成某些 ISP 在网络控制上出现被动局面，对于 ISP 来说，它不可能希望自己的组播网络依懒竞争对手的 RP 路由器来维持通信，基于以上种种原因，需要在各环境下部署独立的 PIM-SM 域，即在各个分公司或各个 ISP 都建立各自独立的 PIM-SM 域，从而保证各个组播网络的独立性与自治性。但是请不要忘记，这些 PIM-SM 域之间还是要相互通信的。

MSDP 概述

在 PIM-SM 域之间要通信，是否会存在问题呢？答案是肯定的，究竟会存在什么样的问题呢？回想一下组播要正常通信的前提条件，那就是必须建立组播树，而组播树要正常建立，RP 就必须要知道组播源和组成员，如果是同一个 PIM-SM 域内，RP 想要知道所有的组播源和组成员是非常轻松的事，但 RP 却没有办法知道其它 PIM-SM 域中的组播源，那这个问题怎么解决呢？方法也很简单，那就是让一个 PIM-SM 域中的 RP 把自己知道的组播源信息告诉其它 PIM-SM 域中的 RP 就可以了，所以我们要想办法让不同 PIM-SM 之间的 RP 能够互相共享和交换组播源信息，这个信息被称为 Source Active（SA），那么不同 PIM-SM 之间的 RP 是否会自动交换 SA 信息呢？当然不会，所以需要通过一个协议来强制这些 RP 之间交换 SA 信息，这个协议就是 Multicast Source Discovery Protocol（MSDP），即组播源发现协议，如下图所示：



从上图中可以看出，PIM-SM Domain 1 的 RP 为 2.2.2.2，R1 是组播源，PIM-SM Domain 2 的 RP 为 4.4.4.4，R5 是组成员，而正常情况下，PIM-SM Domain 2 中的组成员 R5 想要收到 PIM-SM Domain 1 中组播源 R1 发出的数据包是不可能的，因为 PIM-SM Domain 2 中的 RP 虽然知道 R5 是组成员，却不知道谁是组播源，因此无法建立组播树。但当两个 PIM-SM 域的 RP 之间建立 MSDP 连接之后，PIM-SM Domain 1 中的 RP 就会将组播源信息通过 SA 数据包发给 PIM-SM Domain 2 中的 RP，当 PIM-SM Domain 2 中的 RP 学习到组播源和组成员信息之后，两个域之间就能正常建立组播组，最终实现 PIM-SM 域间组播通信。

说到这里，谨慎的你应该发现似乎漏掉了一个问题，上图中，因为 PIM-SM Domain 2 中的 RP 只知道组成员信息，而不知道组播源信息，所以当 PIM-SM Domain 1 中的 RP 通过 MSDP 的 SA 数据包将组播源信息发给它之后事情就完美

了，但是不难看出，PIM-SM Domain 1 中的 RP 也只是知道谁是组播源，却不知道谁是组成员，为什么在只知道组播源而不知道组成员的情况下能够建立组播树的呢？为什么只考虑让 MSDP 共享组播源信息，却没有说共享组成员信息呢？难道只需要知道组播源，不知道谁是组成员也可以建立组播树吗？原因是这样的：试想一下如果你是个非常有钱的大富翁，现在你就是想把你的钱拿到街上去洒发给普通人，你能把你的钱顺利洒掉并让普通人捡到吗（当然这里不考虑街上是否有城管、ZF、以及怪兽）？其实答案咱们很清楚，是可以的，因为只要让普通人（接收者）知道你这个富翁（发送者）要洒钱，知道你是在哪里洒钱即可，而根本就不需要你（发送者）去了解普通人（接收者）有哪些且现在在哪里，也就是说只需要他们知道怎么能够找到你就 OK，你不需要知道怎么找到他们，因为他们会自动来找你；这里的你好比就是组播源，而普通人就好比是组成员，只要组成员知道了谁是组播源，就会一拥而上去找组播源，不需要组播源费脑筋去找组成员。并且这也是因为 PIM-SM 模式是靠 PULL（拉）的方式来建组播树的，也就是谁要接收流量，是由需要接收组播的组成员自己去跟 RP 申请，而不是 RP 去追着问到底谁要接收组播，但如果换了是 PIM-DM 模式，它就是靠 PUSH（推）的方式来建组播树的，就是组播源去追着每个人问到底要不要收组播，这就是区别。

需要重点强调的是，MSDP 协议的功能和目的只有一个，就是把一个 PIM-SM 域内的组播源信息（SA）发送给其它 PIM-SM 域的 RP，从而让 PIM-SM 域间的组播通信正常，MSDP 只是在 PIM-SM 域之间传个消息而已，相当于给对方 RP 打个电话，告诉对方组播源是谁，然后挂了电话就再也没 MSDP 任何事了，至于后面组播是通还是不通，都不是 MSDP 的责任；在应用中，如果一个 PIM-SM 域的 RP 已经通过 MSDP 收到了组播源信息（SA），那么就表示 MSDP 的工作已经做到位了，MSDP 只负责 SA 的传递，而不负责组播流量是否正常，如果 PIM-SM 域之间的组播还是不通，这个时候请不要再怀疑是不是 MSDP 出了问题，问题绝对不在 MSDP 身上，所以这时你应该把精力放到其它地方，MSDP 只关系到 SA，而不关系到组播数据流。

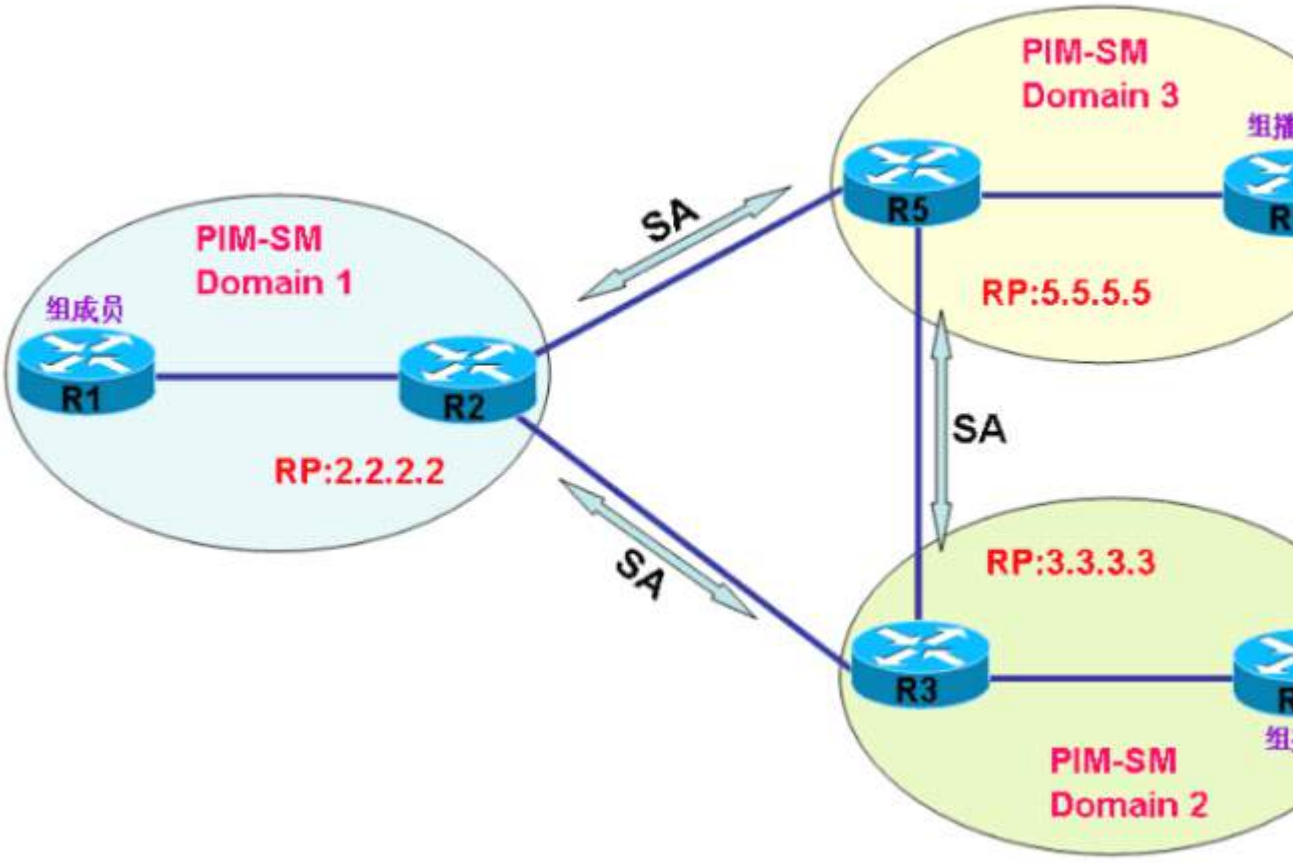
单个域内的 PIM-SM 要通信，是建立的（*，G）条目，如果 PIM-SM 域之间要通信，建立的是(S，G)条目，相当于是距离矢量的路径方式。

注：只有 PIM-SM 才有 MSDP，PIM-DM 是没有 MSDP 的，因为只有 PIM-SM 才有域的概念，才有域间组播通信的概念，PIM-DM 没有域的概念，不存在域间组播通信。

指定 MSDP peer 的命令为：ip msdp peer xxx （xxx 为对端 MSDP 路由器地址）。

MSDP RPF 检测

理论上，PIM-SM 域之间建立 MSDP 连接来交换组播源信息，是在 RP 与 RP 之间建立的，不建议在非 RP 上建立，MSDP 是通过 TCP 来建立的，端口号为 639，和 BGP 的邻居建立相似，所以要建立 MSDP，必须手工指定 MSDP peer 的地址，TCP 由地址小的一端主动发起连接，而地址高的一端则在 LISTEN 状态等待连接。待 MSDP peer 建立成功之后，便开始传递 SA 信息，MSDP 之间的 Keepalive 包每 60 秒一个，如果 75 秒没有收到 Keepalive 包，则会话被重置，而发送的 SA 信息也可以被当作 Keepalive，即如果 75 秒没有收到 Keepalive 但收到了 SA，也表示连接正常。当一个 RP 拥有多个 MSDP peer 时，在从一个 peer 那里收到 SA 之后，会转发给除发送者之外的其它所有 peer，如下图：



在上图中，PIM-SM Domain 1 中的 RP 路由器 R2 与 PIM-SM Domain 2 中的 RP 路由器 R3 以及 PIM-SM Domain 3 中的 RP 路由器 R5 之间建立全互联的 MSDP 连接，R2 从 R3 收到的 SA 会转发给 R5，从 R5 收到的 SA 也会转发给 R3，并且 R3 从 R2 收到的 SA 会转发给 R5，从 R5 收到的 SA 也会转发给 R2，R5 的动作和 R2 与 R3 一样，也会将自己收到的 SA 转发给其它 MSDP peer，这样的转发结果

就是对于同一个组播源信息会从多个 MSDP peer 收到多次重复的，比如对于 PIM-SM Domain 2 中的组播源信息，R5 会同时从 R3 和 R2 收到，那么对于哪一条是正确的，在收到的时候会做一个检测，然后将正常的 SA 缓存起来，如上图中，R5 只应该接收 R3 发来的 SA，而 R3 也应该只接收 R5 发来的 SA，对于 PIM-SM Domain 2 的 SA，R2 只应该接收 R3 发来的，对于 PIM-SM Domain 3 的 SA，R2 只应该接收 R5 发来的；（至于 SA 是谁发来的，是不是从正常路径发来的，我本人认为这并不重要，全部接收好了，我觉得没什么，因为它只是一个消息而已，并且不管是谁发来的，消息的内容都是一模一样的，有什么好纠结的呢？是谁发来的只要消息是对的，并不影响组播流量的转发！），路由器从 MSDP peer 收到 SA 之后，需要做 Reverse Path Forwarding (RPF)检测，即反向路径检测，普通的 RPF 检测的方法是查看自己的路由表中去往发送者 IP 的数据包该从哪个接口出去，那么从哪个接口收到的数据包就被认为是合法有效的，这个合法的接口也被称为 RPF 接口，事实上 MSDP 对于 SA 数据包的 RPF 检测要比普通 RPF 检测复杂的多，但这是 MSDP 唯一的重点，也是唯一的难点，如果 RPF 检测失败，就表示 SA 信息被丢弃，那么组播树就无法建立，组播也就无法通信。

注：在当前的 IOS 版本中，将 MSDP 的 SA 信息缓存起来是强制执行的，不能人为手工开启或关闭，默认情况下，当 MSDP 邻居配置之后，命令“ip multicast cache-sa-state”将被自动添加到 running configuration 中，如果 IOS 版本早于 IOS Releases 12.1(7) 和 12.0(14)S1，默认是不开启 SA 信息缓存功能的，但可以通过手工输入命令“ip multicast cache-sa-state”来开启 SA 缓存功能。

对于 SA 的数据包内容，我们不用研究太深，也没必须研究太深，但有一样东西是必须掌握的，就是某个 PIM-SM 域把 SA 发来之后，SA 数据包里面明确写清楚了那个 PIM-SM 域内的 RP 地址是多少，这是 SA 数据包内我们唯一需要关心的内容，因为该 RP 地址对 SA 数据包的 RPF 检测至关重要，但是这个 SA 里面写的 RP 地址可以被人为更改，更改命令为 ip msdp originator-id 加接口，则使用相应接口的 IP 地址为 SA 内的 RP 地址。

对收到的 SA 数据包做 RPF 检测，和普通的 RPF 检测不一样，普通的 RPF 检测是根据所有的 IGP 路由表以及所有的 BGP 单播路由表来做检测的，即命令“show ip route”看到的路由表，而对 SA 数据包做的 RPF 检测只能根据 BGP 路由表来做检测，单播 BGP 和组播 BGP 都可以，即 Unicast Border Gateway Protocol (uBGP)和 Multicast Border Gateway Protocol (MBGP)，MBGP 是 Multiprotocol-Border Gateway Protocol (MP-BGP)应用的一种，因为 MP-BGP 不仅可以扩展到组播，还可以扩展到 IPv6 以及 MPLS。如果同时存在 MBGP 和单播 BGP，则 MBGP 优先，

既然对 SA 数据包做的 RPF 检测必须依靠 BGP 路由表，那就意味着 PIM-SM 域之间必须运行 BGP，如果没有 BGP，那么 SA 的 RPF 检测将失败，最终导致 SA 数据包被丢弃。

MSDP RPF 检测详细规则

在以下 3 种情况下收到的 SA 是不需要做 RPF 检测的：（**相当重要**）

1．发送方 MSDP peer 是 default MSDP peer 或是唯一的 1 个 MSDP peer(即只配置了 1 条 ip msdp peer 命令)的情况下；

2．发送方 MSDP peer 是 MSDP Mesh Group 中的一员；

3．发送方 MSDP peer 的 IP 地址与 SA 数据包中的 RP 地址相同。

所以在以上 3 种情况下，BGP 和 MBGP 都是不需要的。

通常情况下，MSDP peer 就是 BGP 或 MBGP 邻居，但也有不是的情况，因为 BGP 的邻居类型分 interior BGP (iBGP)和 exterior (eBGP)，所以 SA 的 RPF 具体检测过程也分如下 3 种：

1．MSDP peer 是 iBGP 邻居；

2．MSDP peer 是 eBGP 邻居；

3．MSDP peer 不是 BGP 邻居（但域之间也必须有 BGP）。

注：在以上 3 种检测情况下，都需要对比 SA 中 RP 的 IP 地址信息，所以务必保证将 SA 中 RP 的 IP 地址通告进 BGP 中；同样的，也建议使用建立 MSDP peer 的地址来建立 BGP 邻居，更能保证 RPF 的检测通过。

SA 的 RPF 具体检测细节如下：

1. 当 MSDP peer 是 iBGP 邻居

如果 BGP 路由表中显示去往 SA 数据包中 RP 地址的最佳路径就是走这个 iBGP 邻居，则检测通过，否则检测失败。（先查多播路由表(MRIB)，再查单播路由表(URIB)）

解释：BGP 路由表中去往 SA 数据包中 RP 地址的最佳路径的下一跳地址等于发送该 SA 数据包的 MSDP peer 的地址。

★所以建议建 BGP 和建 MSDP 的地址使用同一个地址。

2. 当 MSDP peer 是 eBGP 邻居；

如果 BGP 路由表中显示去往 SA 数据包中 RP 地址的最佳路径的下一跳 AS 号码就是这个 eBGP 邻居的 AS 号码，则检测通过，否则检测失败。（先查多播路由表(MRIB)，再查单播路由表(URIB)）

解释： 最佳路径的下一跳 AS 就是 AS_PATH 中的第 1 个 AS，即 AS_PATH 中最左边的 AS。

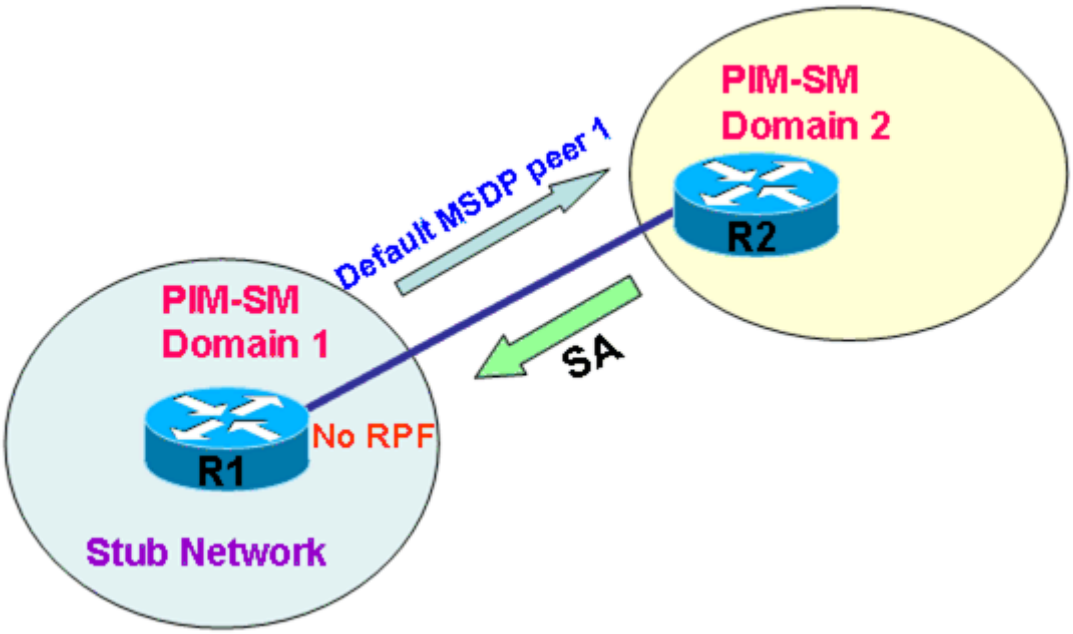
3. 当 MSDP peer 不是 BGP 邻居（但域之间也必须有 BGP）

如果 BGP 路由表中显示去往 SA 数据包中 RP 地址的最佳路径的下一跳 AS 号码和去往 MSDP peer 的最佳路径的下一跳 AS 号码相同，则检测通过，否则检测失败。（先查多播路由表(MRIB)，再查单播路由表(URIB)）

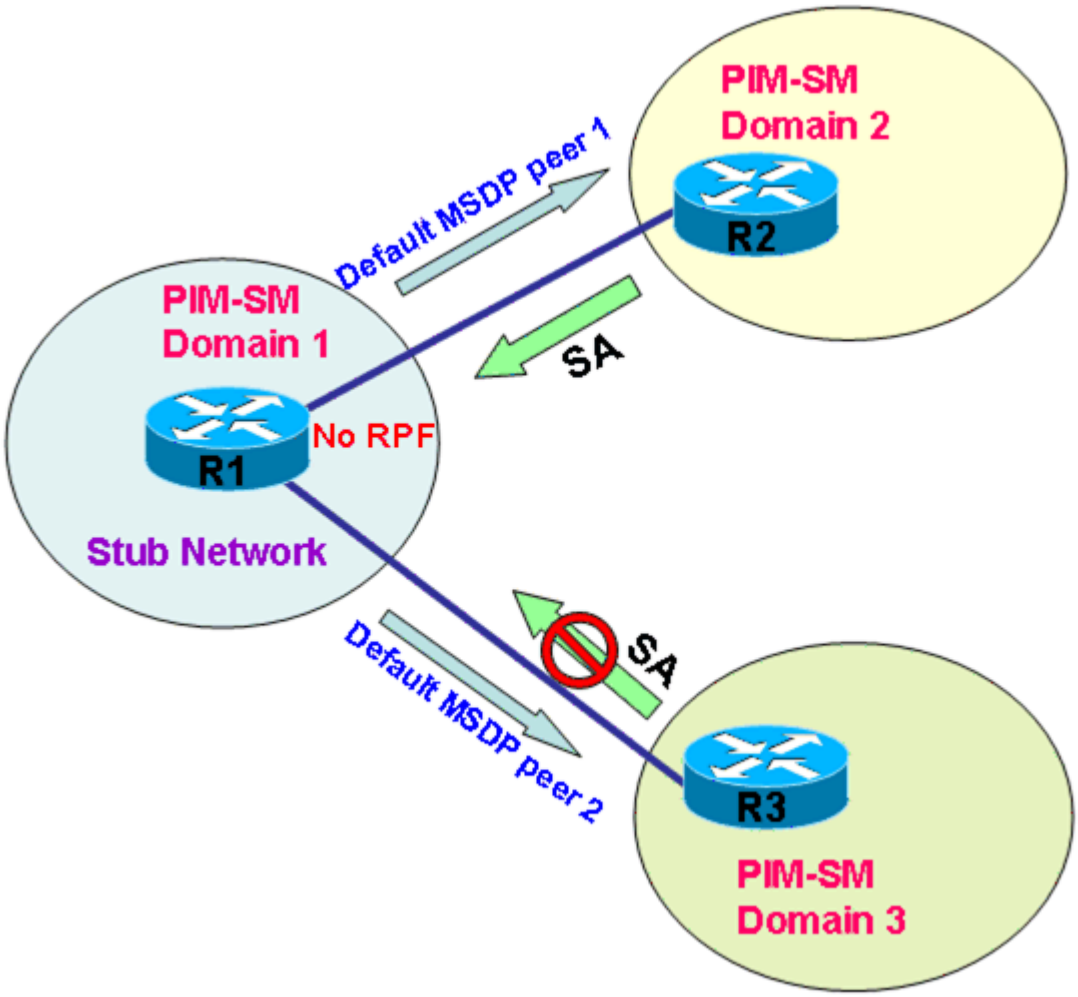
解释：不需要再解释。

Default MSDP Peer

当 MSDP 路由器收到 SA 的时候，必须依靠 BGP 来做 RPF 检测，这就强制 PIM-SM 域中的路由器运行 BGP，否则 SA 数据包会因为 RPF 检测不通过而被丢弃。然而在很多时候，这是不现实的，例如这个网络是个末节网络，或者是用户与 ISP 互联的网络，通常这样的网络只有一个出口，只需要配一条默认路由指出去就足够了，根本用不着配 BGP，如下图：



如上图所示，PIM-SM Domain 1 是个末节网络，对外面只有一个出口，所以只需要对外写一条默认路由即可，不需要配置 BGP，但是很明显，对于 MSDP 收到的所有 SA 都是应该接收的，并且也没有必要对 SA 数据包做 RPF 检测，因为没有其它办法能够收到 SA 数据包了，所以从唯一的出口收到的 SA 统统都有效。在上述情况下，就可以通过在 PIM-SM Domain 1 的 MSDP 路由器 R1 上将它的 MSDP peer 指定为 Default MSDP Peer，也就是将 PIM-SM Domain 2 中的 R2 指定为 Default MSDP Peer，这样一来，R1 从 R2 收到的所有 SA 都不需要再做 RPF 检测就能够缓存起来并使用。在某些情况下，为了网络的冗余性，一个 PIM-SM 域也可能对连了多个 PIM-SM 域，但之间并没有运行 BGP，或者为了使 MSDP 有冗余功能，可以为一个 Default MSDP Peer 再配置一个备份 Default MSDP Peer，当主 Default MSDP Peer 失效之后，再从备份 Default MSDP Peer 接收 SA，如下图：



如上图所示，PIM-SM Domain 1 同时与 PIM-SM Domain 2 和 PIM-SM Domain 3 建立 MSDP 连接，但之间并未运行 BGP，在这种情况下，R1 可以同时将 R2 与 R3 都指定为 Default MSDP Peer，但配置命令会存在先后顺序，例如先指定了 R2，再指定了 R3，那么在正常情况下，只能从先指定的 R2 那里接收 SA，只有在 R2 失效后，才能从 R3 接收 SA；在指定了多个 Default MSDP Peer 的情况下，一开始只能从第一个 Default MSDP Peer 那里接收 SA，只有当第一个不可用了，才能从第二个 Default MSDP Peer 那里接收 SA，第二个不可用再从第三个接收，依此类推。

指定 Default MSDP Peer 的命令为：ip msdp default-peer xxx（xxx 为对端 MSDP 路由器地址）；

在配置此命令之前，必须已经通过命令 ip msdp peer xxx 指定过常规 MSDP peer。

但也可以在指定多个 Default MSDP Peer 的情况下从多个 Default MSDP Peer 那里同时接收 SA，在这里有个条件限制，那就是为每个 Default MSDP Peer 配置 Prefix List 来限制只从特定 Default MSDP Peer 接收特定的 SA。

为 Default MSDP Peer 指定 Prefix List 的命令为：ip msdp default-peer xxx yyy（xxx 为对端 MSDP 路由器地址，yyy 为 Prefix List 名）。

Default MSDP Peer 的优势：

与 Default MSDP Peer 之间不需要运行 BGP，从 Default MSDP Peer 收到的所有 SA 都不需要做 RPF 检测。

重点说明：

如果 MSDP 路由器上只使用了单条 ip msdp peer 命令指定了唯一的 1 个 MSDP peer，那么该 MSDP peer 等同于 Default MSDP Peer。

MSDP Mesh Group

MSDP Mesh Group 的中文意思是 MSDP 全互联组，从字面意思就可以看出，MSDP Peer 之间的连接应该是全互联的，何为全互联？全互联就是指每两个 MSDP Peer 之间都有一条 MSDP 连接，换句话说就是每一个 MSDP Peer 都和其它任何 MSDP Peer 拥有 MSDP 连接。在全互联的模式下，每一个 MSDP Peer 收到任何 SA 都不会转发给其它 MSDP Peer，为什么呢？例如有 A、B、C 共 3 个 MSDP 路由器，它们之间是全互联的，当 B 从 A 那里收到 SA 之后，是不会转发给 C 的，因为 A 和 B 有 MSDP 连接，A 和 C 也有 MSDP 连接，既然 B 能收到 A 的 SA，那说明 A 也将同样的 SA 发过给 C 了，既然 C 也能收到和 B 收到的一样的 SA，那么 B 也没必须多此一举将自己收到的 SA 发给别人，自己能收到，证明别人也能收到，因为这个网络是全互联的。

因为 MSDP Mesh Group 中的任何 MSDP Peer 在收到 SA 之后都不会转发给其它 MSDP Peer，这就表明对于同一份 SA 数据包来说，MSDP Peer 永远只能从单个 MSDP Peer 那里收到，并且发送 SA 的那个 MSDP Peer 就是初始 MSDP Peer，这也保证了 MSDP Peer 所收到的 SA 永远都是对的，MSDP Peer 永远不可能从一个错误的路径收到不可信任的 SA。既然这样，从 MSDP Mesh Group 中的 MSDP Peer 收到的所有 SA 都不再需要做 RPF 检测便被缓存起来使用。

指定 MSDP Mesh Group 的命令为：ip msdp mesh-group test-mesh-group xxx（xxx 为对端 MSDP 路由器地址，test-mesh-group 为 Mesh Group 的组名，可以为每个 MSDP peer 配置不同的组名）。

在配置此命令之前，必须已经通过命令 ip msdp peer xxx 指定过常规 MSDP peer。

MSDP Mesh Group 的优势：

★与 MSDP Mesh Group 中的 MSDP Peer 之间不再需要运行 BGP，从 MSDP Mesh Group 中的 MSDP Peer 收到的所有 SA 都不再需要做 RPF 检测。

★杜绝了 SA 流量泛洪，减少了不必要的 SA 数据包转发。

MSDP SA Filter

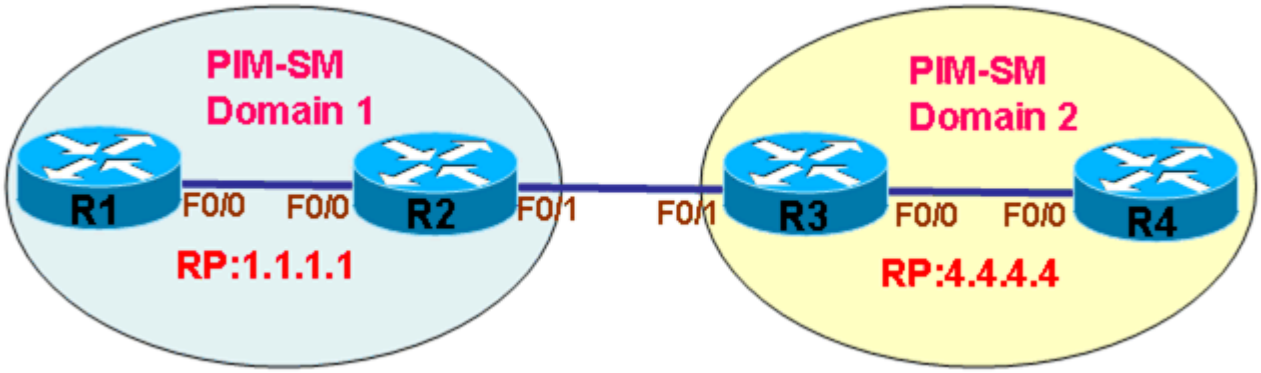
对于 SA，无论是发出去的还是收到的，无论是源自其它 MSDP Peer 还是自己产生的，都可以通过 Filter List 来做过滤，所过滤的条件可基于 ACL、route map、RP access list、RP route map，这里的 ACL 必须是扩展 ACL。

对接收的 SA 做过滤的命令为：ip msdp sa-filter in xxx yyy（xxx 为对端 MSDP 路由器地址，yyy 为 ACL、route map、RP access list、RP route map，如果不跟 yyy，则表示所有 SA）；

对发出去的 SA 做过滤的命令为：ip msdp sa-filter out xxx yyy（xxx 为对端 MSDP 路由器地址，yyy 为 ACL、route map、RP access list、RP route map，如果不跟 yyy，则表示所有 SA）。

PIM-SM 域边界

我们所定义的 PIM-SM 域，其实就是所有组播路由器都拥有同一个 RP 的网络范围，如果在网络里一部分组播路由器使用某一个 RP，而另一部分组播路由器使用另一个 RP，那么它们就属于两个不同的 PIM-SM 域。为组播网络指定 RP 的方法有 3 种，分别为：手工静态指定，Auto-RP(Cisco 私有)，BootStrap Router (BSR)，如果使用手工静态指定的方法，要明确划分 PIM-SM 域是非常简单的，因为想让某台路由器使用某个 RP 只要使用命令强制指定即可，并且任何时候都不会变动，如下图：



如上图所示，网络中有 4 台路由器，要让 R1 和 R2 在 PIM-SM Domain 1 中，使用 R1（1.1.1.1）作为 PIM-SM Domain 1 的 RP，让 R3 和 R4 在 PIM-SM Domain 2 中，使用 R4（4.4.4.4）作为 PIM-SM Domain 2 的 RP，在使用手工静态指定的情况下，只需要使用命令强制指定 R1 和 R2 的 RP 为 1.1.1.1，强制指定 R3 和 R4 的 RP 为 4.4.4.4 即可。

而在使用 Auto-RP 时，我们知道它是个动态分发 RP 的协议，组播网络中的 RP 是谁，需要靠协议自动去计算，然后自动通告出去，所以结果可能并非我们所预期的那样，如果整个组播网络中都使用 Auto-RP 来分发 RP，那么结果可想而知，这里只可能出现一个 PIM-SM 域，因为全网通过 Auto-RP 所获得的 RP 信息都是一致的，如果在使用 Auto-RP 的情况下也要划分出像上图所示的 2 个 PIM-SM 域，就必须让 Auto-RP 的协议流量只在单个域范围内通告，在上图中就是 PIM-SM Domain 1 内的 Auto-RP 通告信息只能在 R1 和 R2 之间传递，无法穿过 R2 的 F0/1 接口到达

PIM-SM Domain 2，而 PIM-SM Domain 2 内的 Auto-RP 通告信息也只能在 R3 和 R4 之间传递，无法穿过 R3 的 F0/1 接口到达 PIM-SM Domain 1，这样两个 PIM-SM 域在使用 Auto-RP 的情况下也能保证互不影响，相互独立。要让 Auto-RP 的通告信息无法穿过 R2 的 F0/1 或者无法穿过 R3 的 F0/1，可以使用过滤组播流量的方法来实现，我们都知道 Auto-RP 的流量使用的组播地址为 224.0.1.39 和 224.0.1.40，所以我们只要在 R2 或 R3 的 F0/1 接口上将这两个组的流量过滤掉即可，但其它组的流量必须放行，要不然正常的组播流量也被过滤掉了，那么组播就不通了。过滤组播流量的方法为在接口下使用命令“ip multicast boundary xxx”，其中 xxx 表示匹配组播地址的 ACL 名，该命令在接口上应用之后，将同时过滤掉接口进来和出去的所有流量，按上面所提到的需求，那就只需要在 R2 或 R3 上做如下配置即可：

```
access-list 1 deny   224.0.1.39

access-list 1 deny   224.0.1.40

access-list 1 permit any



interface f0/1

ip multicast boundary 1
```

该配置定义了将组播地址为 224.0.1.39 和 224.0.1.40 的流量过滤掉，而放行其它所有组播流量，并将该过滤应用到接口 F0/1 上，这样一来，Auto-RP 的流量就无法通过 R2 或 R3 的 F0/1 接口了，从而实现了 PIM-SM Domain 1 与 PIM-SM Domain 2 的相互独立，互不干扰。

对于在使用 BSR 的情况下划分 PIM-SM 域，方法同使用 Auto-RP 的情况类似，区别就是使用的命令不同，过滤 BSR 信息的方法为在接口下使用命令“ip pim bsr-border”，后面不需要跟任何参数，包括 ACL，这条命令将阻止接口上收到或发出的所有 BSR 数据，按上面所提到的需求，那就只需要在 R2 或 R3 上做如下配置即可：

```
interface f0/1

ip pim bsr-border
```

该配置定义了 BSR 的流量在 R2 或 R3 的 F0/1 接口上被过滤掉，但放行其它所有组播流量，这样一来，BSR 的流量就无法通过，从而实现了 PIM-SM Domain 1 与 PIM-SM Domain 2 的相互独立，互不干扰。

重点说明：

在划分多个 PIM-SM 域之后，如果 PIM-SM 域之间还需要组播通信，这两个 PIM-SM 域之间必须还有 PIM-SM 连接通路，即还有 PIM-SM 邻居，否则两个域之间连最基本的 PIM-SM 通路都没了，组播流量就无从转发；虽然 PIM-SM 域之间有 MSDP 连接，但不要忘记，MSDP 的作用是传递 SA 的，它与普通的组播流量无关。如果两个 PIM-SM 域之间被 ISP 隔开，则可通过创建 Generic Routing Encapsulation (GRE) Tunnel 的方式来实现 PIM-SM 互联，所以也必须在 GRE Tunnel 上配开启 PIM-SM。

组播流 RPF 检测详细规则

这里所提到的组播流 RPF 检测是指对普通组播流量的 RPF 检测，即对用户组播流量的 RPF 检测，它与 SA 数据包的 RPF 检测毫无关系，两个是相互独立，毫无关联的。

之前我们已经重点强调过，MSDP 协议的功能和目的只有一个，就是把一个 PIM-SM 域内的组播源信息（SA）发送给其它 PIM-SM 域的 RP，从而让 PIM-SM 域间的组播通信正常，MSDP 只是在 PIM-SM 域之间传个消息而已，只要 PIM-SM 域之间的 SA 已经确认收到了，那么就表示 MSDP 的工作已经做到位了，至于后面组播是通还是不通，都不是 MSDP 的责任；所以这时你应该把精力放到其它地方，在 PIM-SM 域之间的 SA 数据包已经正常的情况下，最有可能影响到组播流量不通的情况就是普通组播流的 RPF 检测，所以下面我们来详细介绍普通组播流的 RPF 检测如何工作。

前面所提到的对 SA 数据包的 RPF 检测只能基于 BGP 路由表，而普通组播流的 RPF 检测可以基于任何路由表，其中包括了 BGP 和 MBGP 路由表，可用的路由表如下：

1．单播路由表，即命令“show ip route”看到的所有路由表，包括 BGP。

2．MBGP 路由表，也就是组播 BGP 专用的路由表。

3．Distance Vector Multicast Routing Protocol (DVMRP) 路由表（目前无须掌握）。

4．静态组播路由表（就像静态指定单播路由表一样指定组播路由表）。

在对普通组播流做 RPF 检测时，是根据以上所有路由表 AD 值来决定先后顺利的，当收到组播流量以后，先查看发送该数据的源 IP 地址是什么，然后在以上路由表中查看去往源 IP 的数据包该从哪个接口出去，那么从哪个接口收到的数据包就被认为是合法有效的，这个合法的接口也被称为 RPF 接口，只有从 RPF 接口收到的流量才能转发，从其它所有接口收到的组播流量都将被丢弃。

注：如果是共享树（Share-Tree），则 RP 的地址被视为源 IP 地址的。

当以上路由表中路由条目的 AD 值完全一样时，则按以下顺序来做 RPF 检测：

如果都一样，那就最长匹配

1．静态组播路由表

2．DVMRP 路由表

3．MBGP 路由表

4．单播路由表

.

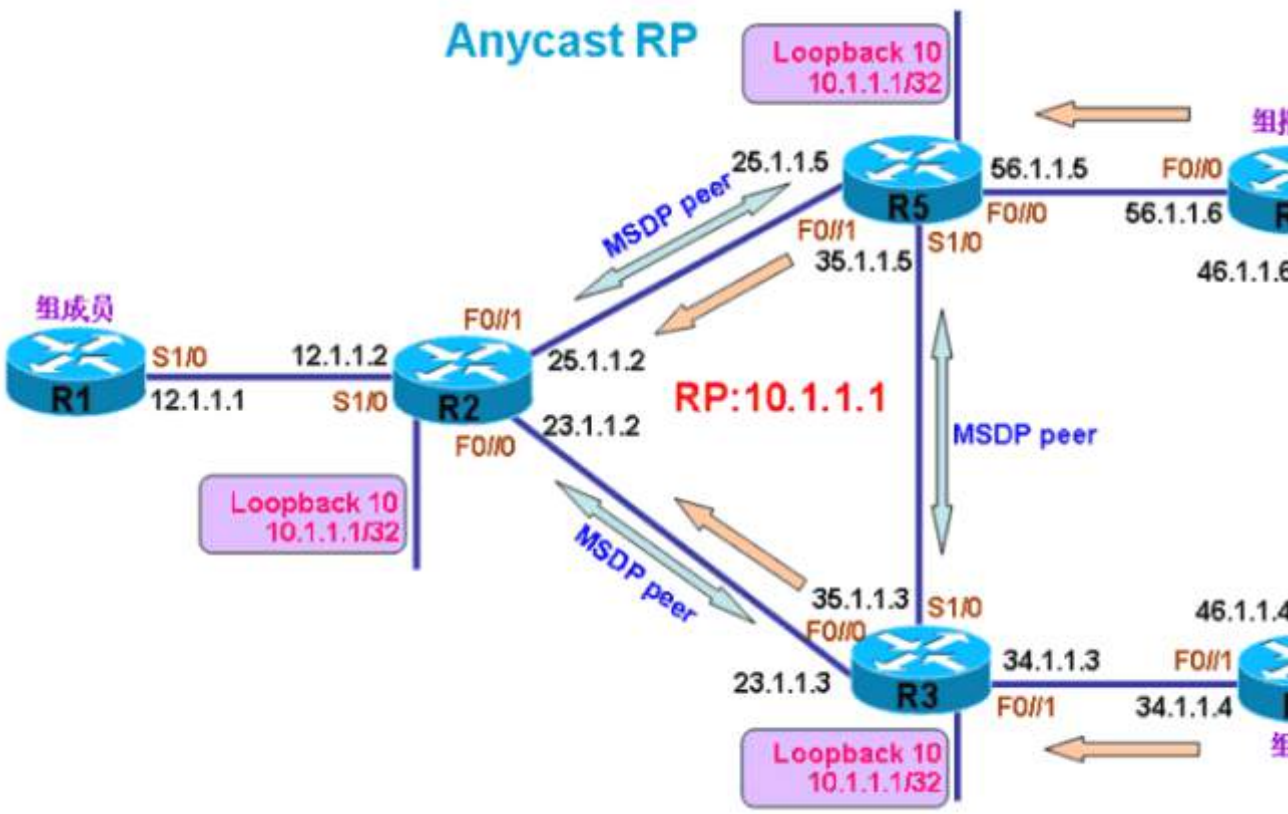
如果在同一个路由表中发现多条去往源 IP 的路由，则选择最长掩码匹配，因为在这里同一个路由表中不可能出现多条掩码相同的路由，所以一定能够选出结果，如果在负载均衡时路由出口有多个，则每个接口都被认为有效。

注：如果接口没有开启 PIM，将不参与 RPF 检测，如果需要看到更多情况下的 RPF 检测效果，请尽可能多的开启组播接口。

Anycast RP

在 PIM-SM 模式下，组播想要通信，就必须拥有正常运行的 RP，指定 RP 的方法有 3 种：手工静态指定，Auto-RP，BSR，其中通过手工静态指定的 RP 在任何时候都不会变动，只要该 RP 路由器出现故障，组播流量将中断，不能拥有备份 RP；如果使用 Auto-RP 或 BSR，则可以设置多个 RP，但同一时间只能有一个 RP 是工作的，只要工作的 RP 出现故障，就会选举备份 RP 接替之前的工作，使用 Auto-RP 或 BSR 能够实现多个 RP 的冗余备份功能，在出现故障时能够快速切换，但是多个 RP 中只能有一个 RP 可以工作，可想而知这个 RP 在大流量组播的情况下就会出现瓶胫，虽然配置了多个 RP，但这些 RP 并不能分担组播流量。

Auto-RP 和 BSR 最大的优势是能够通过多个 RP 实现备份功能，但是却不能实现多个 RP 负载均衡，为了能够在网络中配置多个 RP 既能实现备份功能，又能让所有 RP 同时工作实现流量负载功能，所以在 MSDP 的应用下扩展出了 Anycast RP。Anycast RP 采用了一种很新颖的思路来实现多个 RP 的负载与冗余，它通过在组播网络中将 2 个或者 2 个以上的 RP 配置成相同的 RP 地址来实现冗余和备份功能，每个 RP 路由器上都创建一个 loopback 接口来充当 RP，并且该接口地址的掩码必须是/32 位的，然后将该地址通过动态路由协议发布到网络中，因为所有 RP 的这个 loopback 接口地址全部是一样的，所以能够实现冗余功能，当其中某一台 RP 出现故障后，流量可以很平滑地被转移到另一台 RP 上，这都是因为所有 RP 的地址是同一个地址，如下图：



在上图中，全网运行 OSPF 路由协议，其中 R2、R3、R5 共 3 台路由器上都创建了接口 Loopback 10，并且配置的 IP 地址都是 10.1.1.1/32，然后将该地址发布到 OSPF 中，3 台路由器都以接口 Loopback 10 的地址发布为 RP，最终所有路由器都认为 RP 地址是 10.1.1.1，这样一来，3 个 RP 都能为网络中提供组播流量转发，无论其中哪台 RP 出现故障，只要网络中还剩一台可用 RP，那么组播流量还能正常通信，这就是 Anycast RP 实现的 RP 冗余功能。并且因为所有 RP 的 Loopback 10 接口地址都发布进了 OSPF，所以所有路由器选择去往 10.1.1.1 都是通过 OSPF 来选择最近的路径，例如 R4 一定是从 F0/1 出去选择 R3 的 10.1.1.1，而 R6 一定是从 F0/0 出去选择 R5 的 10.1.1.1，而 R1 一定是从 S1/0 出去选择 R2 的 10.1.1.1，这样一来，Anycast RP 不仅实现了 RP 冗余功能，同时还实现了 RP 的流量负载均衡功能，因为这时所有的流量是被分摊到多个 RP 同时传输的，如果当某个 RP 出现故障，那么流量将被转移至下一台最近的 RP，例如当 R5 出现故障后，R6 将从 S1/0 出去选择 R3 的 10.1.1.1，从而维持了组播通信。

从上述环境中我们还可以看出，组播源 R4 和 R6 以及组成员 R1 都注册到了不同的 RP，这么一来，就没有一个 RP 拥有完整的组播源和组成员信息，这样就不可能建立正常的组播树，所以必须寻求一种办法让所有的 RP 都拥有完整一致的组播源和组成员信息，这个目标可以通过在 RP 之间创建 MSDP 来实现，在上图中，

当在 R2、R3 和 R5 之间建立 MSDP 之后，通过相互交换 SA 信息，就能够让所有的 RP 都学习到完整的组播源和组成员信息，从而保证组播正常通信。这里的 MSDP 可以是全互联，也可以不全互联，只要保护 SA 信息能够让每个 RP 都收到即可。

之前提到过，MSDP 路由器发出的 SA 数据包里面明确写清楚了 RP 地址是多少，但在布署 Anycast RP 的环境下，所有 MSDP 发出的 SA 数据包里面的 RP 地址全是相同的，这种情况可能会导致 SA 无法传递，因为在遇到对 SA 数据包做 RPF 检测时，肯定是会失败的，所以需要手工配置命令将其改掉，更改命令为 ip msdp originator-id 加接口，则使用相应接口的 IP 地址为 SA 数据包内的 RP 地址。

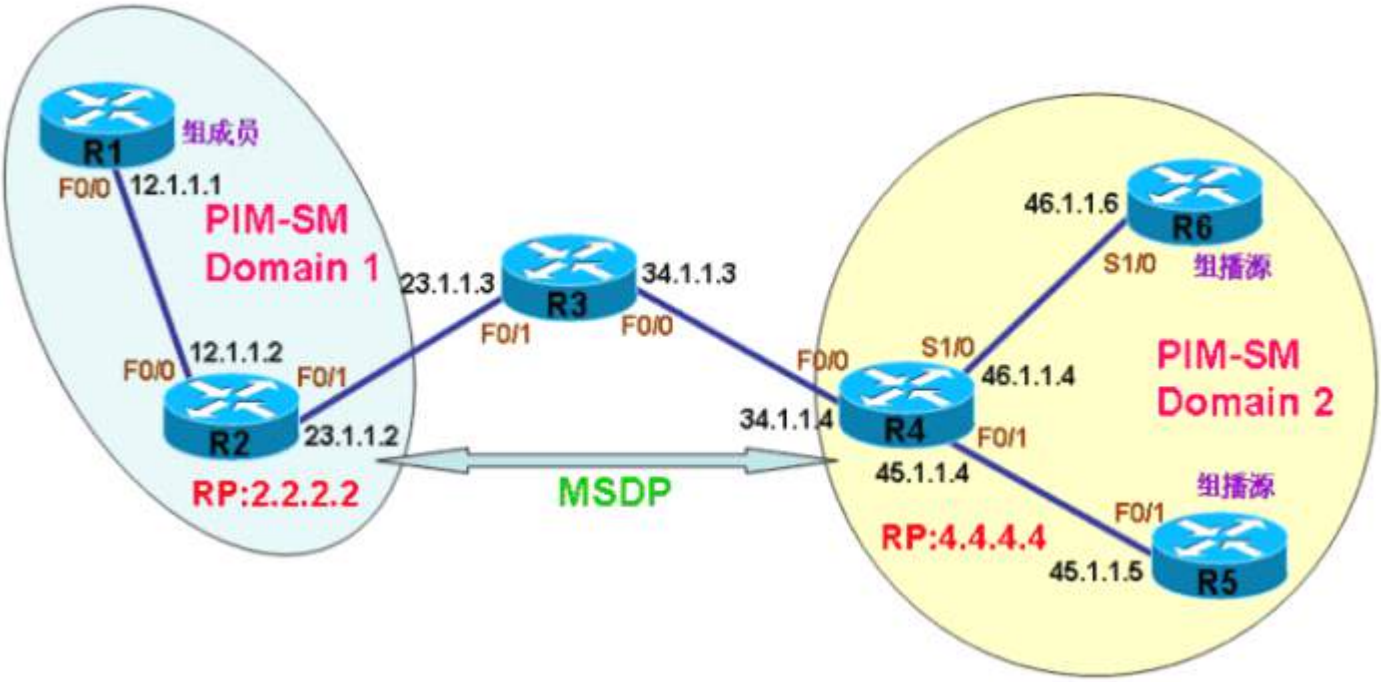
因为不同的 PIM-SM 域之间的 RP 地址是不同的，而 Anycast RP 要求所有的 RP 地址相同，所以 Anycast RP 只能在单个 PIM-SM 域范围内设计和使用，在 PIM-SM 域之间不需要实施 Anycast RP

注：无论在配置 PIM-SM 单域还是 PIM-SM 多域，都应该事先保证全网单播互通。

配置一对一 PIM-SM 域的 MSDP 实验

实验说明：

以下图为环境配置一对一 PIM-SM 域的 MSDP 实验：



上图中两个 PIM-SM 域通过 Internet 相连，其中 R1 和 R2 在 PIM-SM Domain 1 中，R4、R5 和 R6 在 PIM-SM Domain 2 中，R3 模拟 Internet 路由器，所以 R3 不需要参与组播，也不允许参与组播；PIM-SM Domain 1 和 PIM-SM Domain 2 之间需要创建 Tunnel 来穿越 Internet 实现组播通信，其中 R5 和 R6 为组 224.1.1.1 的组播源，R1 为组成员。

除了图上所标识出的接口地址外，R2 和 R4 还配有 Loopback 接口，分别为：

R2（Loopback 0 :2.2.2.2/24）

R4（Loopback 0 :4.4.4.4/24）

PIM-SM Domain 1 的 RP 为 2.2.2.2，即 R2，PIM-SM Domain 2 的 RP 为 4.4.4.4，即 R4，

所有路由器上都运行 OSPF，并将所有接口都发布进 OSPF，以保证全网单播互通。

1. 配置初始网络环境

（1）配置 R1：

r1(config)#int f0/0

r1(config-if)#ip add 12.1.1.1 255.255.255.0

r1(config-if)#no shutdown

r1(config-if)#exit

r1(config)#router ospf 1

r1(config-router)#network 0.0.0.0 0.0.0.0 area 0

r1(config-router)#exit

说明：为 R1 的 F0/0 配置接口地址，并将所有接口发布进 OSPF。

（2）配置 R2：

r2(config)#int loopback 0

r2(config-if)#ip add 2.2.2.2 255.255.255.0

r2(config-if)#ip ospf network point-to-point

r2(config-if)#exit

r2(config)#

r2(config)#int f0/0

r2(config-if)#ip add 12.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#exit

r2(config)#int f0/1

r2(config-if)#ip add 23.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#exit

r2(config)#

r2(config)#router ospf 1

r2(config-router)#network 0.0.0.0 0.0.0.0 area 0

r2(config-router)#exit

说明：为 R2 的 Loopback 0，F0/0，F0/1 配置接口地址，并将所有接口发布进 OSPF。

（3）配置 R3：

r3(config)#int f0/1

r3(config-if)#ip add 23.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#exit

r3(config)#

r3(config)#int f0/0

r3(config-if)#ip add 34.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#exit

r3(config)#router ospf 1

r3(config-router)#network 0.0.0.0 0.0.0.0 area 0

r3(config-router)#exit

说明：为 R3 的 F0/1，F0/0 配置接口地址，并将所有接口发布进 OSPF。

（4）配置 R4：

r4(config)#int loopback 0

r4(config-if)#ip add 4.4.4.4 255.255.255.0

r4(config-if)#ip ospf network point-to-point

r4(config-if)#exit

r4(config)#

r4(config)#int f0/0

r4(config-if)#ip add 34.1.1.4 255.255.255.0

r4(config-if)#no shutdown

r4(config-if)#exit

r4(config)#int f0/1

r4(config-if)#ip add 45.1.1.4 255.255.255.0

r4(config-if)#no shutdown

r4(config-if)#exit

r4(config)#

r4(config)#int s1/0

r4(config-if)#encapsulation frame-relay

r4(config-if)#no frame-relay inverse-arp

r4(config-if)#no arp frame-relay

r4(config-if)#ip add 46.1.1.4 255.255.255.0

r4(config-if)#no shutdown

r4(config-if)#frame-relay map ip 46.1.1.6 406 broadcast

r4(config-if)#ip ospf network point-to-point

r4(config-if)#exit

r4(config)#

说明：为 R4 的 Loopback 0，F0/0，F0/1，S1/0 配置接口地址，并将所有接口发布进 OSPF。

（5）配置 R5：

r5(config)#int f0/1

r5(config-if)#ip add 45.1.1.5 255.255.255.0

r5(config-if)#no shutdown

r5(config-if)#exit

r5(config)#router ospf 1

r5(config-router)#network 0.0.0.0 0.0.0.0 area 0

r5(config-router)#exit

说明：为 R5 的 F0/1 配置接口地址，并将所有接口发布进 OSPF。

（6）配置 R6：

r6(config)#int s1/0

r6(config-if)#encapsulation frame-relay

r6(config-if)#no frame-relay inverse-arp

r6(config-if)#no arp frame-relay

r6(config-if)#ip add 46.1.1.6 255.255.255.0

r6(config-if)#no shutdown

r6(config-if)#frame-relay map ip 46.1.1.4 604 broadcast

r6(config-if)#ip ospf network point-to-point

r6(config-if)#exit

r6(config)#router ospf 1

r6(config-router)#network 0.0.0.0 0.0.0.0 area 0

r6(config-router)#exit

说明：为 R6 的 S1/0 配置接口地址，并将所有接口发布进 OSPF。

（7）查看 R1 的路由学习情况：

r1#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/3] via 12.1.1.2, 00:02:19, FastEthernet0/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/2] via 12.1.1.2, 00:02:19, FastEthernet0/0

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/4] via 12.1.1.2, 00:02:19, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/2] via 12.1.1.2, 00:02:19, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/67] via 12.1.1.2, 00:02:19, FastEthernet0/0

45.0.0.0/24 is subnetted, 1 subnets

O 45.1.1.0 [110/4] via 12.1.1.2, 00:02:19, FastEthernet0/0

r1#

说明：R1 已经学习到全网的每一条路由。

（8）查看 R2 的路由学习情况：

r2#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/2] via 23.1.1.3, 00:02:50, FastEthernet0/1

2.0.0.0/24 is subnetted, 1 subnets

C 2.2.2.0 is directly connected, Loopback0

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/3] via 23.1.1.3, 00:02:50, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

C 12.1.1.0 is directly connected, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/66] via 23.1.1.3, 00:02:50, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

O 45.1.1.0 [110/3] via 23.1.1.3, 00:02:50, FastEthernet0/1

r2#

说明：R2 已经学习到全网的每一条路由。

（9）查看 R3 的路由学习情况：

R3：

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/2] via 23.1.1.2, 00:03:14, FastEthernet0/1

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/2] via 34.1.1.4, 00:03:14, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/2] via 23.1.1.2, 00:03:14, FastEthernet0/1

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/65] via 34.1.1.4, 00:03:14, FastEthernet0/0

45.0.0.0/24 is subnetted, 1 subnets

O 45.1.1.0 [110/2] via 34.1.1.4, 00:03:15, FastEthernet0/0

r3#

说明：R3 已经学习到全网的每一条路由。

（10）查看 R4 的路由学习情况：

R4：

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/3] via 34.1.1.3, 00:03:22, FastEthernet0/0

4.0.0.0/24 is subnetted, 1 subnets

C 4.4.4.0 is directly connected, Loopback0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/2] via 34.1.1.3, 00:03:22, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/3] via 34.1.1.3, 00:03:22, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

C 46.1.1.0 is directly connected, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, FastEthernet0/1

r4#

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

说明：R4 已经学习到全网的每一条路由。

（11）查看 R5 的路由学习情况：

R5：

r5#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/2] via 45.1.1.4, 00:03:48, FastEthernet0/1

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/4] via 45.1.1.4, 00:03:48, FastEthernet0/1

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/2] via 45.1.1.4, 00:03:48, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/3] via 45.1.1.4, 00:03:48, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/4] via 45.1.1.4, 00:03:48, FastEthernet0/1

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/65] via 45.1.1.4, 00:03:48, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

C 45.1.1.0 is directly connected, FastEthernet0/1

r5#

说明：R5 已经学习到全网的每一条路由。

（12）查看 R6 的路由学习情况：

R6：

r6#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/65] via 46.1.1.4, 00:04:11, Serial1/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/67] via 46.1.1.4, 00:04:11, Serial1/0

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/65] via 46.1.1.4, 00:04:11, Serial1/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/66] via 46.1.1.4, 00:04:11, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/67] via 46.1.1.4, 00:04:11, Serial1/0

46.0.0.0/24 is subnetted, 1 subnets

C 46.1.1.0 is directly connected, Serial1/0

45.0.0.0/24 is subnetted, 1 subnets

O 45.1.1.0 [110/65] via 46.1.1.4, 00:04:11, Serial1/0

r6#

说明：R6 已经学习到全网的每一条路由。

（13）在 R1 上测试到其它网段的连通性：

r1#ping

r1#ping 2.2.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/47/80 ms

r1#

r1#ping 4.4.4.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 108/191/240 ms

r1#

r1#ping 45.1.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 45.1.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 124/216/288 ms

r1#

r1#ping 46.1.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 46.1.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 140/212/268 ms

r1#

说明：R1 与 2.2.2.0/24、4.4.4.0/24、45.1.1.0/24、46.1.1.0/24 通信正常，说明已经全网单播互通。

2. 配置 PIM-SM

（1）在 R1 上配置 PIM-SM：

r1(config)#ip multicast-routing

r1(config)#

r1(config)#int f0/0

r1(config-if)#ip pim sparse-mode

r1(config-if)#ip igmp join-group 224.1.1.1

r1(config-if)#exit

r1(config)#

说明：在 R1 上开启组播功能，并且在接口 F0/0 下开启 PIM-SM 模式，接口 F0/0 加入组 224.1.1.1，成为组成员。

（2）在 R2 上配置 PIM-SM：

r2(config)#ip multicast-routing

r2(config)#int loopback 0

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#int f0/0

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#int f0/1

r2(config-if)#ip pim sparse-mode

r2(config-if)#exi

r2(config)#

说明：在 R2 上开启组播功能，并且在接口 Loopback 0，F0/0，F0/1 下开启 PIM-SM 模式；虽然 R3 不参与组播，但连接 R3 的接口 F0/1 还是开了 PIM，是因为如果接口没有开启 PIM，将不参与 RPF 检测，所以为了尽量看到 RPF 检测的效果，尽可能多的开了组播接口。

（3）在 R4 上配置 PIM-SM：

r4(config)#ip multicast-routing

r4(config)#int loopback 0

r4(config-if)#ip pim sparse-mode

r4(config-if)#exit

r4(config)#int f0/1

r4(config-if)#ip pim sparse-mode

r4(config-if)#exit

r4(config)#int s1/0

r4(config-if)#ip pim sparse-mode

r4(config-if)#exit

r4(config)#

说明：在 R3 上开启组播功能，并且在接口 Loopback 0，F0/1，S1/0 下开启 PIM-SM 模式。

（4）在 R5 上配置 PIM-SM：

r5(config)#ip multicast-routing

r5(config)#

r5(config)#int f0/1

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

说明：在 R5 上开启组播功能，并且在接口 F0/1 下开启 PIM-SM 模式。

（5）在 R6 上配置 PIM-SM：

r6(config)#ip multicast-routing

r6(config)#

r6(config)#int s1/0

r6(config-if)#ip pim sparse-mode

r6(config-if)#exit

说明：在 R6 上开启组播功能，并且在接口 S1/0 下开启 PIM-SM 模式。

（6）在 R2 上查看 PIM 邻居情况：

r2#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Interface Uptime/Expires Ver DR

Address Prio/Mode

12.1.1.1 FastEthernet0/0 00:02:42/00:01:30 v2 1 / S

r2#

说明：在 PIM-SM Domain 1 中，只有 R2 和 R1 是 PIM 邻居。

（7）在 R4 上查看 PIM 邻居情况：

r4#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Interface Uptime/Expires Ver DR

Address Prio/Mode

45.1.1.5 FastEthernet0/1 00:01:21/00:01:22 v2 1 / DR S

46.1.1.6 Serial1/0 00:00:53/00:01:20 v2 1 / DR S

r4#

说明：在 PIM-SM Domain 2 中，只有 R4 和 R5 与 R6 是 PIM 邻居。

（8）将 R2 配置为 PIM-SM Domain 1 的 RP：

r2(config)#access-list 24 permit 224.1.1.1

r2(config)#ip pim send-rp-announce loopback 0 scope 16 group-list 24

r2(config)#ip pim send-rp-discovery loopback 0 scope 16

说明：R2 以 Loopback 0 作为组 224.1.1.1 的 RP 和映射代理。

（9）将 R4 配置为 PIM-SM Domain 1 的 RP：

r4(config)#access-list 24 permit 224.1.1.1

r4(config)#ip pim send-rp-announce loopback 0 scope 16 group-list 24

r4(config)#ip pim send-rp-discovery loopback 0 scope 16

说明：R4 以 Loopback 0 作为组 224.1.1.1 的 RP 和映射代理。

（10）在 R1 上查看 RP 学习情况：

r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 2.2.2.2 (?), v2v1

Info source: 2.2.2.2 (?), elected via Auto-RP

Uptime: 00:00:47, expires: 00:02:09

r1#

说明：和预期一样，PIM-SM Domain 1 的 R1 学习到的 RP 正是 2.2.2.2，因为目前 PIM-SM Domain 1 与 PIM-SM Domain 2 之间没有 PIM 通路，所以两边的 RP 信息互不传递，互不影响。

（11）在 R2 上查看 RP 学习情况：

r2#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback0)

Group(s) 224.1.1.1/32

RP 2.2.2.2 (?), v2v1

Info source: 2.2.2.2 (?), elected via Auto-RP

Uptime: 00:01:03, expires: 00:02:58

r2#

说明：和预期一样，PIM-SM Domain 1 的 R2 学习到的 RP 也是 2.2.2.2。

（12）在 R4 上查看 RP 学习情况：

r4#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback0)

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 00:00:40, expires: 00:02:16

r4#

说明：和预期一样，PIM-SM Domain 2 的 R4 学习到的 RP 正是 4.4.4.4，，因为与 PIM-SM Domain 1 之间没有 PIM 通路，所以两边保持着 PIM-SM 域独立。

（13）在 R5 上查看 RP 学习情况：

r5#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 00:00:58, expires: 00:02:00

r5#

说明：和预期一样，PIM-SM Domain 2 的 R5 学习到的 RP 也是 4.4.4.4。

（14）在 R6 上查看 RP 学习情况：

r6#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 00:01:13, expires: 00:02:42

r6#

说明：和预期一样，PIM-SM Domain 2 的 R6 学习到的 RP 也是 4.4.4.4。

（15）测试 R1 到组 224.1.1.1 的通信情况：

r1#ping 224.1.1.1

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 4 ms

r1#

说明：在单个域内，组播通信没问题。

（16）测试 R2 到组 224.1.1.1 的通信情况：

r2#ping 224.1.1.1

Type escape sequence to abort.

Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 168 ms

Reply to request 0 from 12.1.1.1, 180 ms

r2#

（17）测试 PIM-SM Domain 2 到组 224.1.1.1 的通信情况：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r4#

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r5#

R6：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r6#

说明：因为 PIM-SM Domain 2 与 PIM-SM Domain 1 之间连最基本的 PIM 通路都没有，所以组播不通是正常的，要通了才怪了。

3. 配置 GRE Tunnel 隧道

（1）在 R2 上配置连接 R4 的 GRE Tunnel 隧道：

r2(config)#int tunnel 24

r2(config-if)#ip address 24.1.1.2 255.255.255.0

r2(config-if)#tunnel source loopback 0

r2(config-if)#tunnel destination 4.4.4.4

r2(config-if)#exit

说明：配置连接 PIM-SM Domain 1 和 PIM-SM Domain 2 的 GRE Tunnel 隧道，使用双方的 Loopback 0 地址作为 GRE Tunnel 隧道两端的端点。

（2）在 R4 上配置连接 R2 的 GRE Tunnel隧道：

r4(config)#int tunnel 24

r4(config-if)#ip add 24.1.1.4 255.255.255.0

r4(config-if)#tunnel source loopback 0

r4(config-if)#tunnel destination 2.2.2.2

r4(config-if)#exit

说明：配置连接 PIM-SM Domain 1 和 PIM-SM Domain 2 的 GRE Tunnel 隧道

（3）测试 GRE Tunnel隧道的连通性：

r2#ping 24.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 24.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 124/223/360 ms

r2#

r4#ping 24.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 24.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 72/145/276 ms

r4#

说明：目前连接 PIM-SM Domain 1 和 PIM-SM Domain 2 的 GRE Tunnel 隧道通信正常。

（4）再次测试 PIM-SM Domain 2 到组 224.1.1.1 的通信情况：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r4#

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r5#

说明：即使配置了连接 PIM-SM Domain 1 和 PIM-SM Domain 2 的 GRE Tunnel 隧道，但域间组播还是无法通信，因为两个 PIM-SM 域之间连最基本的 PIM 通路都没有，如果没有 PIM 通路，这种做了 GRE Tunnel 隧道的环境跟没做是一样的。

（5）在 R4 上查看关于组 224.1.1.1 的 PIM-SM 组播树情况：

r4#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:06:46/stopped, RP 4.4.4.4, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(4.4.4.4, 224.1.1.1), 00:01:01/00:02:24, flags: PT

Incoming interface: Loopback0, RPF nbr 0.0.0.0

Outgoing interface list: Null

(45.1.1.4, 224.1.1.1), 00:01:01/00:02:16, flags: P

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list: Null

(45.1.1.5, 224.1.1.1), 00:00:50/00:02:34, flags: PT

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list: Null

(46.1.1.4, 224.1.1.1), 00:01:00/00:02:16, flags: P

Incoming interface: Serial1/0, RPF nbr 0.0.0.0

Outgoing interface list: Null

r4#

说明：由以上组播树可以看出，R4 根本就没有能够去往 PIM-SM Domain 1 的组播出口。

（6）将 GRE Tunnel 隧道加入 PIM，形成 PIM-SM Domain 1 和 PIM-SM Domain 2 的 PIM 通道：

r2(config)#int tunnel 24

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r4(config)#int tunnel 24

r4(config-if)#ip pim sparse-mode

r4(config-if)#exit

说明：R2 和 R4 都已经在 GRE Tunnel 隧道上开启 PIM 功能，但不要忘记，GRE Tunnel 隧道接口也已经被通告进了 OSPF 进程，因为在配置 OSPF 时，是将所有接口通告进去的。

（7）查看 R4 的 PIM 邻居情况：

r4#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Interface Uptime/Expires Ver DR

Address Prio/Mode

45.1.1.5 FastEthernet0/1 00:14:49/00:01:42 v2 1 / DR S

46.1.1.6 Serial1/0 00:14:22/00:01:39 v2 1 / DR S

24.1.1.2 Tunnel24 00:00:25/00:01:19 v2 1 / S

r4#

说明：R4 已经通过 GRE Tunnel 隧道和 R2 形成 PIM 邻居关系，说明 PIM-SM Domain 1 已经通过 PIM 和 PIM-SM Domain 2 正式连通。

（8）查看所有组播路由器的 RP 情况：

r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 2.2.2.2 (?), elected via Auto-RP

Uptime: 00:09:09, expires: 00:02:58

r1#

r2#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback0)

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 00:12:08, expires: 00:02:00

RP 2.2.2.2 (?), v2v1

Info source: 2.2.2.2 (?), via Auto-RP

Uptime: 01:21:36, expires: 00:02:55

r2#

r4#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback0)

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 01:21:02, expires: 00:02:45

r4#

r5#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 01:21:14, expires: 00:02:19

r5#

r5#

r6#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 01:21:29, expires: 00:02:06

r6#

说明：因为 PIM-SM Domain 1 和 PIM-SM Domain 2 已经在 GRE Tunnel 隧道上通过 PIM 连通，所以所有路由器就等于在同一个 PIM-SM 大域里面，由于 RP 4.4.4.4 的地址高于 RP 2.2.2.2，导致所有组播路由器将得到 RP 是 4.4.4.4 的消息，因此要规划 PIM-SM 域间组播，就必须在

PIM-SM Domain 1 和 PIM-SM Domain 2 之间划分域边界来隔离两个组播网络。

（9）在 R2 上划分 PIM-SM 域边界：

r2(config)#access-list 1 deny 224.0.1.39

r2(config)#access-list 1 deny 224.0.1.40

r2(config)#access-list 1 permit any

r2(config)#int tunnel 24

r2(config-if)#ip multicast boundary 1

r2(config-if)#exit

r2(config)#

说明：因为 GRE Tunnel 隧道是导致 PIM-SM Domain 1 和 PIM-SM Domain 2 成为同一个 PIM-SM 域的根本原因，所以在 GRE Tunnel 隧道上将分发 RP 信息的 Auto-RP 流量过滤掉，从而实现 PIM-SM 域的分割，因为过滤是对进和出流量同时生效，所以只需要在 R2 一边做即可，不需要在 R4 上重复做。

（10）再次查看所有组播路由器的 RP 情况：

r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 2.2.2.2 (?), v2v1

Info source: 2.2.2.2 (?), elected via Auto-RP

Uptime: 00:00:44, expires: 00:02:15

r1#

r2#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback0)

Group(s) 224.1.1.1/32

RP 2.2.2.2 (?), v2v1

Info source: 2.2.2.2 (?), elected via Auto-RP

Uptime: 01:26:30, expires: 00:02:00

r2#

r4#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback0)

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 01:25:52, expires: 00:02:58

r4#

r5#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 01:26:05, expires: 00:02:26

r5#

r5#

r6#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 4.4.4.4 (?), v2v1

Info source: 4.4.4.4 (?), elected via Auto-RP

Uptime: 01:26:16, expires: 00:02:19

r6#

说明：划分 PIM-SM 域边界后，PIM-SM Domain 1 中的 R1 和 R2 学习到 2.2.2.2 是 RP，PIM-SM Domain 2 中的 R4，R5 和 R6 学习到 4.4.4.4 是 RP，和预期的一样，从而实现了 PIM-SM Domain 1 和 PIM-SM Domain 2 的 PIM-SM 域独立。

4. 配置 MSDP

（1）在 PIM-SM Domain 1 与 PIM-SM Domain 2 之间配置 MSDP 连接：

r2(config)#ip msdp peer 4.4.4.4 connect-source loopback 0

r4(config)#ip msdp peer 2.2.2.2 connect-source loopback 0

说明：在 PIM-SM Domain 1 的 RP 路由器 R2 和 PIM-SM Domain 2 的路由器 R4 上使用 Loopback 0 建立 MSDP 连接。

（2）查看 MSDP 的连接情况：

r2#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

4.4.4.4 ? Down 00:00:48 0 0 ?

r2#

r4#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

2.2.2.2 ? Listen 00:00:27 0 0 ?

r4#

说明：因为 R4 的 IP 地址大，所以停留在 Listen 状态对方发起连接。

（3）再次查看 MSDP 的连接情况：

r2#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

4.4.4.4 ? Up 00:01:07 0 0 ?

r2#

r4#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

2.2.2.2 ? Up 00:01:16 0 0 ?

r4#

说明：MSDP 已经正常连接，说明已经可以传递 SA 信息。

（4）在 R2 上查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 0 entries

r2#

说明：因为目前没有路由器发起组播流量，也就没有组播源，那么也就不会产生 SA 信息。

（5）在组播源路由器 R5 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

在 MSDP 路由器 R2 上开启 Debug:

r2#debug ip msdp peer

MSDP Peer debugging is on

r2#

r2#debug ip msdp detail

MSDP Detail debugging is on

r2#

在组播源路由器 R5 上发起组播流量：

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r5#

说明：组播不通，后面将根据实际情况一步一步解决。

观察 R2 上的 Debug 信息：

r2#

*Mar 1 00:49:41.563: MSDP(0): Received 20-byte TCP segment from 4.4.4.4

*Mar 1 00:49:41.567: MSDP(0): Append 20 bytes to 0-byte msg 6 from 4.4.4.4, qs 1

*Mar 1 00:49:41.567: MSDP(0): 4.4.4.4: Received 20-byte msg 6 from peer

*Mar 1 00:49:41.567: MSDP(0): 4.4.4.4: SA TLV, len: 20, ec: 1, RP: 4.4.4.4

*Mar 1 00:49:41.571: MSDP(0): 4.4.4.4: Peer RPF check passed for single peer

*Mar 1 00:49:41.571: MSDP(0): WAVL Insert SA Source 45.1.1.5 Group 224.1.1.1 RP 4.4.4.4 Successful

r2#

说明：因为目前 R2 只有单个 MSDP peer，所以对于收到的 SA 也就没必要进行 RPF 检测，最后 R2 顺利将起始 PIM-SM 域的 MSDP 路由器 4.4.4.4 发来的 SA 缓存起来，从 Debug 信息中可以看出，SA 信息中的组播源是 45.1.1.5，正是我们发起组播流量的 R5，而组地址 224.1.1.1 也是正确的，对方的 RP 是 4.4.4.4 也正确。

（6）在 MSDP 路由器 R2 上查看 SA 信息的接收情况：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 1 entries

(45.1.1.5, 224.1.1.1), RP 4.4.4.4, AS ?,00:01:04/00:04:55, Peer 4.4.4.4

Learned from peer 4.4.4.4, RPF peer 4.4.4.4,

SAs received: 1, Encapsulated data received: 0

r2#

说明：MSDP 路由器 R2 已经正常接收到 PIM-SM Domain 1 的 SA，但组播还是不通，这个原因在理论部分我们已经重点强调过，MSDP 的责任只是负责 SA 的传递，只要 SA 已经收到，那么 MSDP 的工作就已经完成，如果组播仍然不通，就不再是 MSDP 的问题，一定是其它问题，而通常影响组播不通的最大问题可能就是组播流的 RPF 检测，所以下面我们着手检查组播流 RPF 的情况，从而解决组播的通信问题。

因为普通组播流量的 RPF 检测是先查看发送该数据的源 IP 地址是什么，然后在路由表中查看去往源 IP 的数据包该从哪个接口出去，那么从哪个接口收到的数据包就被认为是合法有效的，而查看的路由表共包括：单播路由表，MBGP 路由表，DVMRP 路由表以及静态组播路由表，但目前我们只有单播路由表，其它都没有配，所以我们首先从单播路由表下手，并且也只能从单播路由表下手。

（7）在 R2 上查看去往组播源 45.1.1.5 的单播路由表情况：

r2#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/2] via 23.1.1.3, 00:06:01, FastEthernet0/1

2.0.0.0/24 is subnetted, 1 subnets

C 2.2.2.0 is directly connected, Loopback0

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/3] via 23.1.1.3, 00:06:01, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

C 24.1.1.0 is directly connected, Tunnel24

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/66] via 23.1.1.3, 00:06:01, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

O 45.1.1.0 [110/3] via 23.1.1.3, 00:06:01, FastEthernet0/1

r2#

说明：从 R2 的单播路由表可知，R2 认为去往组播源 45.1.1.5 的流量应该从接口

F0/1 出去，那么源地址为 45.1.1.5 的流量也必须从接口 F0/1 进来，如果从其它接口进来就被认为是不合法的，统统都会被丢弃，但是我们应该知道，从 PIM-SM Domain 1 发来的组播流量一定是从 GRE Tunnel 隧道进来的，因为是 GRE Tunnel 隧道通过 PIM 将 PIM-SM Domain 1 与 PIM-SM Domain 2 连接起来的。

（8）查看 GRE Tunnel 隧道在 OSPF 下的情况：

```
r2#sh ip ospf interface tunnel 24

Tunnel24 is up, line protocol is up

  Internet Address 24.1.1.2/24, Area 0

  Process ID 1, Router ID 2.2.2.2, Network Type POINT_TO_POINT, Cost: 11111

  Transmit Delay is 1 sec, State POINT_TO_POINT

  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

   oob-resync timeout 40

   Hello due in 00:00:02

  Supports Link-local Signaling (LLS)

  Cisco NSF helper support enabled

  IETF NSF helper support enabled

  Index 4/4, flood queue length 0

  Next 0x0(0)/0x0(0)

  Last flood scan length is 1, maximum is 2

  Last flood scan time is 0 msec, maximum is 4 msec

  Neighbor Count is 1, Adjacent neighbor count is 1

   Adjacent with neighbor 4.4.4.4
```

Suppress hello for 0 neighbor(s)

r2#

说明：虽然 GRE Tunnel 隧道接口已经被通告进 OSPF 进程，但 R2 去往组播源 45.1.1.5 的路径还是通过物理接口 F0/1 出去的，仍然没有走 GRE Tunnel 隧道，从上可以看出，这是因为 GRE Tunnel 隧道的 OSPF Cost 为 11111，远大于走 F0/1 的路径。

（9）查看 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: FastEthernet0/1

RPF neighbor: ? (23.1.1.3)

RPF route/mask: 45.1.1.0/24

RPF type: unicast (ospf 1)

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：从以上可以看出，R2 的系统对 45.1.1.5 的 RPF 检测结果为从物理接口 F0/1 出去，和我们之前所看到的一样，并且这个结果是通过 unicast (ospf 1)的信息获得的，但事实上这个结果与真实的组播路径不吻合，所以组播流量遭到了丢弃，如果要让组播正常通信，就要改变系统的 RPF 检测结果，让其认为从 GRE Tunnel 隧道进来的组播流量才是正确的。

5. 通过单播路由表解决 PIM-SM 域间组播通信问题

说明：因为 R2 的系统对 45.1.1.5 的 RPF 检测结果是通过 unicast (ospf 1)的信息获得的，OSPF 路由的 AD 值为 110，这时我们完全可以通过手工配置单播静态路由来取代 OSPF 学习到的路由，手工配置单播静态路由将 45.1.1.5 指向 GRE Tunnel 隧道，让从 GRE Tunnel 隧道过来的组播流量能够通过 RPF 检测。

（1）在 R2 上配置单播静态路由：

r2(config)#ip route 45.1.1.5 255.255.255.255 tunnel 24

说明：手工配置单播静态路由将组播源地址 45.1.1.5 指向 GRE Tunnel 隧道，让从 GRE Tunnel 隧道过来的组播流量能够通过 RPF 检测。

（2)）在 R2 上再次查看去往组播源 45.1.1.5 的单播路由表情况：

r2#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/2] via 23.1.1.3, 00:13:37, FastEthernet0/1

2.0.0.0/24 is subnetted, 1 subnets

C 2.2.2.0 is directly connected, Loopback0

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/3] via 23.1.1.3, 00:13:37, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

C 24.1.1.0 is directly connected, Tunnel24

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/66] via 23.1.1.3, 00:13:37, FastEthernet0/1

45.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

S 45.1.1.5/32 is directly connected, Tunnel24

O 45.1.1.0/24 [110/3] via 23.1.1.3, 00:13:37, FastEthernet0/1

r2#

说明：手工配置的将组播源地址 45.1.1.5 指向 GRE Tunnel 隧道的单播静态路由已经生效，虽然 OSPF 路由保持不变，但静态路由的子网掩码为 /32 位，优于任何路由。

（3）再次查看 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: Tunnel24

RPF neighbor: ? (24.1.1.4)

RPF route/mask: 45.1.1.5/32

RPF type: unicast (static)

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：通过手工配置将组播源地址 45.1.1.5 指向 GRE Tunnel 隧道的单播静态路由之后，目前系统的 RPF 检测结果已经成功认为从 GRE Tunnel 隧道接口进来的组播流量是合法的了，这样一来，PIM-SM Domain 1 与 PIM-SM Domain 2 之间的组播就可以通信了。

（4）再次测试从 PIM-SM Domain 2 的组播源 45.1.1.5 向 PIM-SM Domain 1 发送组播：

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 328 ms

Reply to request 0 from 12.1.1.1, 516 ms

Reply to request 1 from 12.1.1.1, 316 ms

Reply to request 2 from 12.1.1.1, 236 ms

Reply to request 3 from 12.1.1.1, 236 ms

Reply to request 4 from 12.1.1.1, 268 ms

Reply to request 5 from 12.1.1.1, 196 ms

Reply to request 6 from 12.1.1.1, 236 ms

Reply to request 7 from 12.1.1.1, 212 ms

Reply to request 8 from 12.1.1.1, 268 ms

Reply to request 9 from 12.1.1.1, 204 ms

r5#

说明：已经通过手工配置单播静态路由成功解决 PIM-SM Domain 1 与 PIM-SM Domain 2 的域间组播通信。

查看当前环境中所有路由器的配置信息请点击“running-config”

6. 通过静态组播路由表解决 PIM-SM 域间组播通信问题

（1）将之前手工配置的单播静态路由删除，以恢复组播故障问题：

r2(config)#no ip route 45.1.1.5 255.255.255.255 tunnel 24

r2(config)#exit

r2#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/2] via 23.1.1.3, 00:15:10, FastEthernet0/1

2.0.0.0/24 is subnetted, 1 subnets

C 2.2.2.0 is directly connected, Loopback0

4.0.0.0/24 is subnetted, 1 subnets

O 4.4.4.0 [110/3] via 23.1.1.3, 00:15:10, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

C 24.1.1.0 is directly connected, Tunnel24

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/66] via 23.1.1.3, 00:15:10, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

O 45.1.1.0 [110/3] via 23.1.1.3, 00:15:11, FastEthernet0/1

r2#

说明：之前手工配置的将组播源地址 45.1.1.5 指向 GRE Tunnel 隧道的单播静态路由已被删除，目前又重新选择了错误的 OSPF 路由。

（2）查看当前 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: FastEthernet0/1

RPF neighbor: ? (23.1.1.3)

RPF route/mask: 45.1.1.0/24

RPF type: unicast (ospf 1)

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：R2 的系统对 45.1.1.5 的 RPF 检测结果又回到了依靠 OSPF 做检测的状态，此状态为错误状态。

（3）测试从 PIM-SM Domain 2 的组播源 45.1.1.5 向 PIM-SM Domain 1 发送组播：

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r5#

说明：删除了手工配置的单播静态路由之后，组播问题依旧。

（4）在 R2 上配置组播静态路由：

r2(config)#ip mroute 45.1.1.5 255.255.255.255 tunnel 24

说明：手工配置组播静态路由将组播源地址 45.1.1.5 指向 GRE Tunnel 隧道，让从 GRE Tunnel 隧道过来的组播流量能够通过 RPF 检测。

（5）在 R2 上查看组播静态路由表：

r2#sh ip mroute static

Mroute: 45.1.1.5/32, interface: Tunnel24

Protocol: none, distance: 0, route-map: none

r2#

说明：手工配置的将组播源地址 45.1.1.5 指向 GRE Tunnel 隧道的组播静态路由已生效。

（6）再次查看 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: Tunnel24

RPF neighbor: ? (24.1.1.4)

RPF route/mask: 45.1.1.5/32

RPF type: unicast (static)

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：和预期的一样，系统的 RPF 检测结果已经成功认为从 GRE Tunnel 隧道接口进来的组播流量是合法的。

（7）再次测试从 PIM-SM Domain 2 的组播源 45.1.1.5 向 PIM-SM Domain 1 发送组播：

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 384 ms

Reply to request 1 from 12.1.1.1, 236 ms

Reply to request 2 from 12.1.1.1, 264 ms

Reply to request 3 from 12.1.1.1, 236 ms

Reply to request 4 from 12.1.1.1, 224 ms

Reply to request 5 from 12.1.1.1, 248 ms

Reply to request 6 from 12.1.1.1, 280 ms

Reply to request 7 from 12.1.1.1, 220 ms

Reply to request 8 from 12.1.1.1, 232 ms.

r5#

说明：通过手工配置组播静态路由也能成功解决 PIM-SM Domain 1 与 PIM-SM Domain 2 的域间组播通信。

查看当前环境中所有路由器的配置信息请点击“running-config”

7. 通过 MBGP 解决 PIM-SM 域间组播通信问题

说明：除了单播路由表和静态组播路由表之外，还可以通过 MBGP 来解决组播流的 RPF 检测问题，当然不要忘记还有 DVMRP，但这不是我们讨论的范围，如果这 4 种路由表同时出现，则靠路由条目的 AD 值来决定选谁，但如果 AD 值全部一样，请注意他们的先后顺序是：

1．静态组播路由表

2．DVMRP 路由表

3．MBGP 路由表

4．单播路由表

（1）在 R2 上删除组播静态路由，恢复组播故障：

r2(config)#no ip mroute 45.1.1.5 255.255.255.255 tunnel 24

r2(config)#exit

r2#

r2#

r2#sh ip mroute static

r2#

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: FastEthernet0/1

RPF neighbor: ? (23.1.1.3)

RPF route/mask: 45.1.1.0/24

RPF type: unicast (ospf 1)

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r5#

说明：从上面看到，删除组播静态路由后，系统对组播源 45.1.1.5 的 RPF 检测又回到了选择 OSPF 的错误状态，并且最后组播不通。

（2）在 R2 和 R4 之间配置 MBGP：

R2：

r2(config)#router bgp 2

r2(config-router)#bgp router-id 2.2.2.2

r2(config-router)#neighbor 4.4.4.4 remote-as 4

r2(config-router)#neighbor 4.4.4.4 update-source loopback 0

r2(config-router)#neighbor 4.4.4.4 ebgp-multihop

r2(config-router)#address-family ipv4 multicast

r2(config-router-af)#neighbor 4.4.4.4 activate

r2(config-router-af)#exit

r2(config-router)#exit

r2(config)#

R4：

r4(config)#router bgp 4

r4(config-router)#bgp router-id 4.4.4.4

r4(config-router)#neighbor 2.2.2.2 remote-as 2

r4(config-router)#neighbor 2.2.2.2 update-source loopback 0

r4(config-router)#neighbor 2.2.2.2 ebgp-multihop

r4(config-router)#address-family ipv4 multicast

r4(config-router-af)#neighbor 2.2.2.2 activate

r4(config-router-af)#exi

r4(config-router)#exi

r4(config)#

说明：R2 与 R4 通过双方的 Loopback 0 来建 MBGP 邻居，所以请注意，通过 Loopback 0 来建 MBGP 邻居，那么 Loopback 0 的地址就会影响到 MBGP 的路由走向，因为 BGP 的路由是根据到达邻居的地址然后去 IGP 路由表做递归路由查询的。

（3）查看 BGP 与 MBGP 邻居状态：

r2#sh ip bgp summary

BGP router identifier 2.2.2.2, local AS number 2

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
4.4.4.4	4	4	4	4	1	0	0	00:01:11	0

r2#

r2#show ip bgp ipv4 multicast summary

BGP router identifier 2.2.2.2, local AS number 2

BGP table version is 1, main routing table version 1

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
4.4.4.4	4	5	5	1	0	0	0	00:02:09	0

r2#

说明：BGP 和 MBGP 的邻居均已建立。

（4）查看 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: FastEthernet0/1

RPF neighbor: ? (23.1.1.3)

RPF route/mask: 45.1.1.0/24

RPF type: unicast (ospf 1)

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：虽然已经配置了 MBGP，但是 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果

还是选择了使用 OSPF 路由，导致最终结果错误。

（5）查看 MBGP 路由表：

r2#show ip bgp ipv4 multicast

r2#

说明：因为之前并没有在 MBGP 里面通告关于组播源地址 45.1.1.5 的路由信息，所以虽然配置了 MBGP，但毫无作用，所以需要将关于组播源地址 45.1.1.5 的路由发布进 MBGP。

（6）在 MBGP 中发布路由：

r4(config)#router bgp 4

r4(config-router)#address-family ipv4 multicast

r4(config-router-af)#network 45.1.1.0 mask 255.255.255.0

r4(config-router-af)#exit

r4(config-router)#exit

说明：在 R4 上将组播源地址 45.1.1.5 的路由发布进 MBGP，使 45.1.1.5 能够依靠 MBGP 路由顺利通过 RPF 检测。

（7）再次查看 MBGP 路由表：

r2#show ip bgp ipv4 multicast

BGP table version is 2, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 45.1.1.0/24	4.4.4.4	0		0	4 i

r2#

说明：已经从 MBGP 正常学习到关于组播源地址 45.1.1.5 的路由，该路由可用作对 45.1.1.5 的 RPF 检测。

（8）再次查看 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: FastEthernet0/1

RPF neighbor: ? (23.1.1.3)

RPF route/mask: 45.1.1.0/24

RPF type: mbgp

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：从上面可以看出，R2 的系统已经使用 MBGP 的路由对组播源地址 45.1.1.5 进行 RPF 检测，但是检测的结果却是错误的，这里我们需要注意，从之前的 MBGP 路由表可以得知，MBGP 认为去往 45.1.1.5 的数据包应该发到 4.4.4.4，即发给它的 MBGP 邻居 R4，然后 4.4.4.4 并不是与之直连的网络，所以 MBGP 就得需要到单播路由表中做递归查询去往 4.4.4.4 究竟该如何去。

（9）查询 R2 去往 4.4.4.4 的单播路由表：

```
r2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
O       34.1.1.0 [110/2] via 23.1.1.3, 00:10:15, FastEthernet0/1
     2.0.0.0/24 is subnetted, 1 subnets
C       2.2.2.0 is directly connected, Loopback0
     4.0.0.0/24 is subnetted, 1 subnets
O       4.4.4.0 [110/3] via 23.1.1.3, 00:10:15, FastEthernet0/1
     23.0.0.0/24 is subnetted, 1 subnets
C       23.1.1.0 is directly connected, FastEthernet0/1
     24.0.0.0/24 is subnetted, 1 subnets
```

C 24.1.1.0 is directly connected, Tunnel24

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/66] via 23.1.1.3, 00:10:16, FastEthernet0/1

45.0.0.0/24 is subnetted, 1 subnets

O 45.1.1.0 [110/3] via 23.1.1.3, 00:10:16, FastEthernet0/1

r2#

说明：从单播路由表中可以看出，去 4.4.4.4 的数据包应该从物理接口 F0/1 出去，所以 MBGP 路由表的结果就是去 45.1.1.5 的数据包也应该顺着 4.4.4.4 的路径从物理接口 F0/1 出去，这就与我们预期的从 GRE Tunnel 隧道出去不相符，所以我们只有想办法让 R2 从 GRE Tunnel 隧道接口去 4.4.4.4，才能改变让 MBGP 去往 45.1.1.5 从 GRE Tunnel 隧道出去，这样对组播源 45.1.1.5 的 RPF 检测才能变成 GRE Tunnel 隧道。

（10）在 R2 上手工配置从 GRE Tunnel 隧道去 4.4.4.4 的静态路由：

r2(config)#ip route 4.4.4.4 255.255.255.255 24.1.1.4

说明：配置静态路由从 GRE Tunnel 隧道去 4.4.4.4，在这里，请你注意，请你一定要注意，如果你不注意这个，那么通过这种方式配置的 MBGP 将不生效，将前功尽弃，需要注意的就是这条静态路由一定要指定下一跳地址，而不能指定出接口，所以你千万不要配成“ip route 4.4.4.4 255.255.255.255 tunnel 24”，如果你这么配了，你可以试一下，你的组播看似正常，但却不通。

（11）在 R2 上查看 GRE Tunnel 隧道情况：

r2#sh ip int brief tunnel 24

Interface IP-Address OK? Method Status Protocol

Tunnel24 24.1.1.2 YES manual up down

r2#

说明：无意间，两个 PIM-SM 域之间的关键点 GRE Tunnel 隧道的状态居然变成了“down”，这是因为我们配置的 GRE Tunnel 隧道是靠 R2 与 R4 双方的 Loopback 0 来通信的，而刚才我们配置的静态路由居然要让去往 R4 的 Loopback 0（4.4.4.4）靠 GRE Tunnel 隧道来通信，这里就形成了死循环，因为必须双方的 Loopback 0 通了，GRE Tunnel 隧道才能通，但现在又变成了必须 GRE Tunnel 隧道通了，Loopback 0 才能过，这就前后矛盾了，所以我们既然强制 Loopback 0 靠 GRE Tunnel 隧道来通信，那么我们就必须让 GRE Tunnel 隧道靠其它接口来通信而不能靠 Loopback 0 口，那我们就可以更改 GRE Tunnel 隧道靠 R2 与 R4 的物理口来通信。

（12）更改 GRE Tunnel 隧道的通信方式：

R2：

r2(config)#int tunnel 24

r2(config-if)#tunnel source f0/1

r2(config-if)#tunnel destination 34.1.1.4

r2(config-if)#exit

r2(config)#

R4：

r4(config)#int tunnel 24

r4(config-if)#tunnel source f0/0

r4(config-if)#tunnel destination 23.1.1.2

r4(config-if)#exit

说明：更改 GRE Tunnel 隧道靠 R2 与 R4 的物理口来通信。

（13）再次查看 GRE Tunnel 隧道情况：

r2#sh ip int brief tunnel 24

Interface	IP-Address	OK? Method Status	Protocol
Tunnel24	24.1.1.2	YES manual up	up

r2#

r2#ping 24.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 24.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 88/164/216 ms

r2#

说明：更改 GRE Tunnel 隧道靠 R2 与 R4 的物理口来通信之后，目前 GRE Tunnel 隧道工作正常。

（14）再次查看 R2 的系统对组播源 45.1.1.5 的 RPF 检测结果：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: Tunnel24

RPF neighbor: ? (24.1.1.4)

RPF route/mask: 45.1.1.0/24

RPF type: mbgp

RPF recursion count: 1

Doing distance-preferred lookups across tables

r2#

说明：现在 R2 的系统已经靠 MBGP 的路由正确对组播源 45.1.1.5 做出了 RPF 检测，如果您之前将从 GRE Tunnel 隧道去 4.4.4.4 的静态路由“ip route 4.4.4.4 255.255.255.255 24.1.1.4”误写成指向出接口的“ip route 4.4.4.4 255.255.255.255 tunnel 24”，那么 R2 的系统对组播源 45.1.1.5 做出的 RPF 检测结果将如下：

r2#sh ip rpf 45.1.1.5

RPF information for ? (45.1.1.5)

RPF interface: Tunnel24

RPF neighbor: ? (4.4.4.4)

RPF route/mask: 45.1.1.0/24

RPF type: mbgp

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

而这个结果将使域间组播无法通信，您应该能发现两者的 RPF neighbor 有所区别，正确的应该是真正的直连地址，错误的却是远程非直连地址。

（15）再次测试从 PIM-SM Domain 2 的组播源 45.1.1.5 向 PIM-SM Domain 1 发送组播：

r5#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 232 ms

Reply to request 1 from 12.1.1.1, 188 ms

Reply to request 2 from 12.1.1.1, 236 ms

Reply to request 2 from 12.1.1.1, 316 ms

Reply to request 3 from 12.1.1.1, 200 ms

Reply to request 4 from 12.1.1.1, 204 ms

Reply to request 5 from 12.1.1.1, 356 ms.

Reply to request 7 from 12.1.1.1, 236 ms

Reply to request 8 from 12.1.1.1, 216 ms

Reply to request 9 from 12.1.1.1, 204 ms

r5#

说明：通过配置 MBGP 也成功解决了 PIM-SM Domain 1 与 PIM-SM Domain 2 的域间组播通信，请认真观察，认真体会三种方法解决组播通信的关键所在。

附：在这里需要说明：在配置 MBGP 解决 PIM-SM 域间组播通信时，我们遇到的问题是由于 MBGP 的路由依靠单播路由表对 MBGP 路由的下一跳做递归查询时，错将组播源地址走向了物理接口而并非 GRE Tunnel 隧道接口，所以我们将被 MBGP 路由用来作递归查询的单播路由静态指向了 GRE Tunnel 隧道口，从而使得 MBGP 在单播路由表做递归查询时能被正确引导到 GRE Tunnel 隧道接口，在这里

还有另外一个做法就是直接使用 GRE Tunnel 隧道接口地址来建立 MBGP 邻居，这样 MBGP 的路由不需要做递归查询就知道直接走 GRE Tunnel 隧道，从而就不会有以上那么多步骤，但还请不要忘记，并不建议使用 GRE Tunnel 隧道建立 BGP 或 MBGP 邻居，通常是保持建立 MSDP 连接的地址和建立 BGP 或 MBGP 的地址为相同地址，因为这样做可以保证在出现 3 个或 3 个以上的 PIM-SM 域时对 SA 数据包所做的 RPF 检测能够顺利通过，虽然我们这里只有两个 PIM-SM 域。

如果使用 GRE Tunnel隧道建邻居，则 R2 和 R4 的配置如下：

r2(config)#router bgp 2

r2(config-router)#no neighbor 4.4.4.4

r2(config-router)#neighbor 24.1.1.4 remote-as 4

r2(config-router)#address-family ipv4 multicast

r2(config-router-af)#neighbor 24.1.1.4 activate

r2(config-router-af)#exit

r4(config)#router bgp 4

r4(config-router)#no neighbor 2.2.2.2

r4(config-router)#neighbor 24.1.1.2 remote-as 2

r4(config-router)#address-family ipv4 multicast

r4(config-router-af)#neighbor 24.1.1.2 activate

r4(config-router-af)#network 45.1.1.0 mask 255.255.255.0

r4(config-router-af)#exit

（16）测试从 PIM-SM Domain 2 的组播源 46.1.1.6（即 R6）向 PIM-SM Domain 1 发送组播：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r6#

说明：种种原因，PIM-SM Domain 2 的组播源 46.1.1.6（即 R6）与 PIM-SM Domain 1 的组播还无法通信。

（17）在 R2 上查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 1 entries

(45.1.1.5, 224.1.1.1), RP 4.4.4.4, BGP/AS 0, 00:02:53/00:05:43, Peer 4.4.4.4

r2#

说明：R2 上连关于组播源 46.1.1.6 的 SA 都没有，问题很严重，其实我们不难看出，R5 和 R6 的架构完全相同，但唯一的区别就是 R6 是使用的帧中继链路与 R4 相连的，而很不巧的是，R6 的 S1/0 地址大于 R4 的 S1/0 地址，所以在在 PIM 选举 DR 时，结果 R6 成了 DR，如下：

r4#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

```
Neighbor          Interface            Uptime/Expires    Ver   DR
Address                                                  Prio/Mode
24.1.1.2          Tunnel24             00:06:04/00:01:34 v2    1 / S
45.1.1.5          FastEthernet0/1      00:07:33/00:01:33 v2    1 / DR S
46.1.1.6          Serial1/0            00:05:59/00:01:41 v2    1 / DR S
r4#
```

（18）提高 R4 的 S1/0 的 DR 优先级：

```
r4(config)#int s1/0
r4(config-if)#ip pim dr-priority 4
r4(config-if)#exit

r4#sh ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface            Uptime/Expires    Ver   DR
Address                                                  Prio/Mode
24.1.1.2          Tunnel24             00:06:25/00:01:44 v2    1 / S
45.1.1.5          FastEthernet0/1      00:07:54/00:01:43 v2    1 / DR S
46.1.1.6          Serial1/0            00:06:19/00:01:21 v2    1 / S
r4#
```

说明：提高 R4 的 S1/0 接口的 DR 优先级后，R4 成为了 DR，这里不再详细讨论到底是为什么，不清楚的请复习组播前部分的理论知识。

（19）R6 再次发起组播数据并在 R2 上查看 SA 信息：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r6#

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 2 entries

(45.1.1.5, 224.1.1.1), RP 4.4.4.4, BGP/AS 0, 00:04:21/00:05:14, Peer 4.4.4.4

(46.1.1.6, 224.1.1.1), RP 4.4.4.4, BGP/AS 0, 00:00:45/00:05:14, Peer 4.4.4.4

r2#

说明：组播仍然不通，但关于组播源 46.1.1.6 的 SA 已经正常传递，那么剩下的问题就不可能是 MSDP 的问题。

（20）查看 R2 的系统关于组播源 46.1.1.6 的 RPF 检测结果：

r2#sh ip rpf 46.1.1.6

RPF information for ? (46.1.1.6)

RPF interface: FastEthernet0/1

RPF neighbor: ? (23.1.1.3)

RPF route/mask: 46.1.1.0/24

RPF type: unicast (ospf 1)

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：由于 R2 的系统还是基于 OSPF 路由对组播源 46.1.1.6 做的 RPF 检测，所以路径错误。

（21）在 R2 上配置组播静态路由解决 PIM-SM Domain 2 的组播源 46.1.1.6 与 PIM-SM Domain 1 的组播通信：

r2(config)#ip mroute 46.1.1.6 255.255.255.255 tunnel 24

说明：解决 PIM-SM Domain 2 的组播源 46.1.1.6 与 PIM-SM Domain 1 组播通信的方法有

单播路由、MBGP、组播静态路由，而我们这里采用了组播静态路由。

（22）再次查看 R2 的系统关于组播源 46.1.1.6 的 RPF 检测结果：

r2#sh ip rpf 46.1.1.6

RPF information for ? (46.1.1.6)

RPF interface: Tunnel24

RPF neighbor: ? (24.1.1.4)

RPF route/mask: 46.1.1.6/32

RPF type: static

RPF recursion count: 0

Doing distance-preferred lookups across tables

r2#

说明：配置了组播静态路由后，R2 对组播源 46.1.1.6 做的 RPF 检测成功指向 GRE Tunnel 隧道。

（23）再次测试从 PIM-SM Domain 2 的组播源 46.1.1.6 向 PIM-SM Domain 1 发送组播：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 420 ms

Reply to request 1 from 12.1.1.1, 312 ms

Reply to request 2 from 12.1.1.1, 312 ms

Reply to request 3 from 12.1.1.1, 204 ms

Reply to request 4 from 12.1.1.1, 200 ms

Reply to request 5 from 12.1.1.1, 280 ms

Reply to request 6 from 12.1.1.1, 216 ms

Reply to request 7 from 12.1.1.1, 248 ms

Reply to request 8 from 12.1.1.1, 248 ms

Reply to request 9 from 12.1.1.1, 220 ms

r6#

说明：所有问题解决后，PIM-SM Domain 1 与 PIM-SM Domain 2 之间的组播畅通无阻。

（24）查看组播树：

R2：

r2#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

(*, 224.1.1.1), 00:22:53/00:02:32, RP 2.2.2.2, flags: SJC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:22:53/00:02:40

(45.1.1.5, 224.1.1.1), 00:03:29/00:02:56, flags: MT

Incoming interface: Tunnel24, RPF nbr 24.1.1.4, Mbgp

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:03:29/00:02:40

(46.1.1.6, 224.1.1.1), 00:16:55/00:02:56, flags: MT

Incoming interface: Tunnel24, RPF nbr 24.1.1.4, Mroute

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:16:55/00:02:40

r2#

R4：

r4#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:21:18/stopped, RP 4.4.4.4, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(45.1.1.5, 224.1.1.1), 00:04:02/00:03:23, flags: TA

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list:

Tunnel24, Forward/Sparse, 00:02:00/00:03:18

(46.1.1.6, 224.1.1.1), 00:21:06/00:03:23, flags: TA

Incoming interface: Serial1/0, RPF nbr 0.0.0.0

Outgoing interface list:

Tunnel24, Forward/Sparse, 00:16:38/00:02:51

r4#

说明：从组播树中可以看见，PIM-SM 域之间的通信使用的是（S，G），而并非 PIM-SM 模式传统的（*，G），如果您之前将从 GRE Tunnel 隧道去 4.4.4.4 的静态路由“ip route 4.4.4.4 255.255.255.255 24.1.1.4”误写成指向出接口的“ip route 4.4.4.4 255.255.255.255 tunnel 24”，那么组播树将如下：

r4#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:22:34/stopped, RP 4.4.4.4, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(45.1.1.5, 224.1.1.1), 00:05:18/00:02:56, flags: PTA

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list: Null

(46.1.1.6, 224.1.1.1), 00:22:22/00:03:26, flags: TA

Incoming interface: Serial1/0, RPF nbr 0.0.0.0

Outgoing interface list:

Tunnel24, Forward/Sparse, 00:17:55/00:02:33

r4#

R2：

r2#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:25:03/00:00:22, RP 2.2.2.2, flags: SJC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:25:03/00:02:27

(45.1.1.5, 224.1.1.1), 00:05:39/00:02:26, flags: M

Incoming interface: Tunnel24, RPF nbr 4.4.4.4, Mbgp

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:05:39/00:02:27

(46.1.1.6, 224.1.1.1), 00:19:05/00:02:56, flags: MT

Incoming interface: Tunnel24, RPF nbr 24.1.1.4, Mroute

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:19:05/00:02:27

r2#

说明：R2 上正常的 flags 为 MT，而写错静态路由的 flags 为 M；R4 上正常的 flags

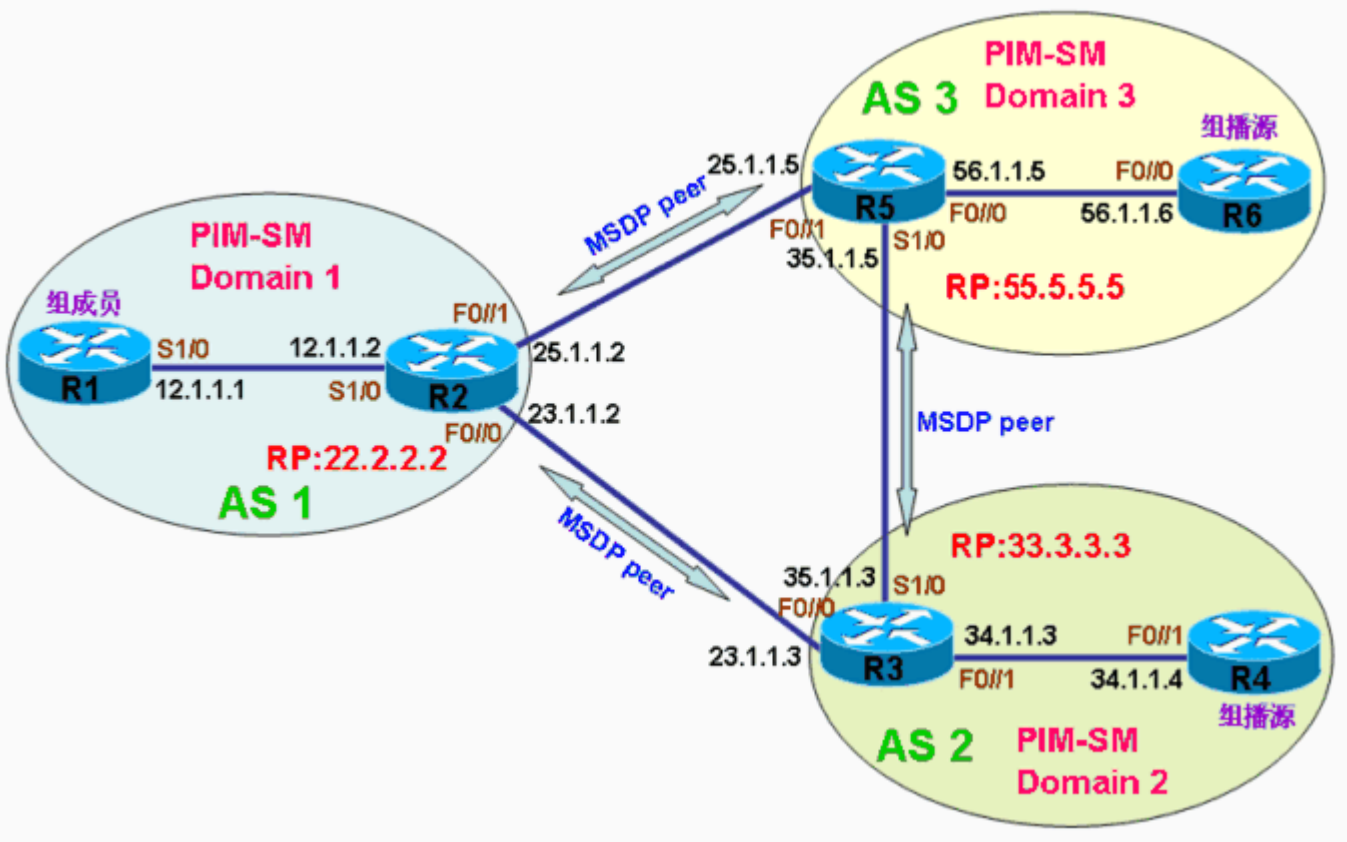
为 TA，而写错静态路由的 flags 为 PTA。

查看当前环境中所有路由器的配置信息请点击“running-config”

配置 3 个 PIM-SM 域全互联的 MSDP 实验

实验说明：

以下图为环境配置 3 个 PIM-SM 域全互联的 MSDP 实验：



上图中 3 个 PIM-SM 域互连，其中 R1 和 R2 在 PIM-SM Domain 1 中，R3 和 R4 在 PIM-SM Domain 2 中，R5 和 R6 在 PIM-SM Domain 3 中；R4 和 R6 为组 224.1.1.1 的组播源，R1 为组成员。

除了图上所标识出的接口地址外，R2、R3 和 R5 还配有 2 个 Loopback 接口，分别为：

R2（Loopback 0 :2.2.2.2/24；Loopback 1 :22.2.2.2/24）

R3（Loopback 0 :3.3.3.3/24；Loopback 1 :33.3.3.3/24）

R5（Loopback 0 :5.5.5.5/24；Loopback 1 :55.5.5.5/24）

PIM-SM Domain 1 的 RP 为 22.2.2.2，即 R2，PIM-SM Domain 2 的 RP 为 33.3.3.3，即 R3，PIM-SM Domain 3 的 RP 为 55.5.5.5，即 R5；

3 个 PIM-SM 域之间建立全互联的 MSDP 连接，连接情况如下：

R2（2.2.2.2）—— R3（3.3.3.3）

R2（2.2.2.2）—— R5（55.5.5.5）

R3（3.3.3.3）—— R5（55.5.5.5）

所有路由器上都运行 OSPF，并将所有接口都发布进 OSPF，以保证全网单播互通。

1. 配置初始网络环境

（1）配置 R1：

```
r1(config)#int s1/0
r1(config-if)#encapsulation frame-relay
r1(config-if)#no frame-relay inverse-arp
```

r1(config-if)#no arp frame-relay

r1(config-if)#ip add 12.1.1.1 255.255.255.0

r1(config-if)#no shutdown

r1(config-if)#frame-relay map ip 12.1.1.2 102 broadcast

r1(config-if)#ip ospf network point-to-point

r1(config-if)#exit

r1(config)#router ospf 1

r1(config-router)#network 0.0.0.0 0.0.0.0 area 0

r1(config-router)#exit

说明：为 R1 的 S1/0 配置接口地址，并将所有接口发布进 OSPF。

（2）配置 R2：

r2(config)#int loopback 0

r2(config-if)#ip add 2.2.2.2 255.255.255.0

r2(config-if)#ip ospf network point-to-point

r2(config-if)#exit

r2(config)#

r2(config)#int loopback 22

r2(config-if)#ip add 22.2.2.2 255.255.255.0

```
r2(config-if)#ip ospf network point-to-point

r2(config-if)#exit

r2(config)#

r2(config)#int s1/0

r2(config-if)#encapsulation frame-relay

r2(config-if)#no frame-relay inverse-arp

r2(config-if)#no arp frame-relay

r2(config-if)#ip add 12.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#frame-relay map ip 12.1.1.1 201 broadcast

r2(config-if)#ip ospf network point-to-point

r2(config-if)#exit

r2(config)#

r2(config)#int f0/0

r2(config-if)#ip add 23.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#exit

r2(config)#int f0/1

r2(config-if)#ip add 25.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#exit

r2(config)#router ospf 1
```

r2(config-router)#network 0.0.0.0 0.0.0.0 area 0

r2(config-router)#exit

说明：为 R2 的 Loopback 0，Loopback 22，S1/0，F0/0，F0/1 配置接口地址，并将所有接口发布进 OSPF。

（3）配置 R3：

r3(config)#int loopback 0

r3(config-if)#ip add 3.3.3.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#ip ospf network point-to-point

r3(config-if)#exit

r3(config)#

r3(config)#int loopback 33

r3(config-if)#ip add 33.3.3.3 255.255.255.0

r3(config-if)#ip ospf network point-to-point

r3(config-if)#exit

r3(config)#int f0/0

r3(config-if)#ip add 23.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#exit

r3(config)#

```
r3(config)#int f0/1

r3(config-if)#ip add 34.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#exit

r3(config)#int s1/0

r3(config-if)#encapsulation frame-relay

r3(config-if)#no frame-relay inverse-arp

r3(config-if)#no arp frame-relay

r3(config-if)#ip add 35.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#frame-relay map ip 35.1.1.5 305 broadcast

r3(config-if)#ip ospf network point-to-point

r3(config-if)#exit

r3(config)#router ospf 1

r3(config-router)#network 0.0.0.0 0.0.0.0 area 0

r3(config-router)#exit

r3(config)#
```

说明：为 R3 的 Loopback 0，Loopback 33，S1/0，F0/1，F0/0 配置接口地址，并将所有接口发布进 OSPF。

（4）配置 R4：

r4(config)#int f0/1

r4(config-if)#ip add 34.1.1.4 255.255.255.0

r4(config-if)#no shutdown

r4(config-if)#exit

r4(config)#router ospf 1

r4(config-router)#network 0.0.0.0 0.0.0.0 area 0

r4(config-router)#exit

r4(config)#

说明：为 R4 的 F0/1 配置接口地址，并将所有接口发布进 OSPF。

（5）配置 R5：

r5(config)#int loopback 0

r5(config-if)#ip address 5.5.5.5 255.255.255.0

r5(config-if)#ip ospf network point-to-point

r5(config-if)#exit

r5(config)#

r5(config)#int loopback 55

r5(config-if)#ip add 55.5.5.5 255.255.255.0

r5(config-if)#ip ospf network point-to-point

r5(config-if)#exit

r5(config)#int f0/0

r5(config-if)#ip address 56.1.1.5 255.255.255.0

r5(config-if)#no shutdown

r5(config-if)#exit

r5(config)#int f0/1

r5(config-if)#ip add 25.1.1.5 255.255.255.0

r5(config-if)#no shutdown

r5(config-if)#exit

r5(config)#

r5(config)#int s1/0

r5(config-if)#encapsulation frame-relay

r5(config-if)#no frame-relay inverse-arp

r5(config-if)#no arp frame-relay

r5(config-if)#ip add 35.1.1.5 255.255.255.0

r5(config-if)#no shutdown

r5(config-if)#frame-relay map ip 35.1.1.3 503 broadcast

r5(config-if)#ip ospf network point-to-point

r5(config-if)#exit

说明：为 R5 的 Loopback 0，Loopback 55，S1/0，F0/1，F0/0 配置接口地址，并将所有接口发布进 OSPF。

（6）配置 R6：

r6(config)#int f0/0

r6(config-if)#ip address 56.1.1.6 255.255.255.0

r6(config-if)#no shutdown

r6(config-if)#exit

r6(config)#

说明：为 R6 的 F0/0 配置接口地址，并将所有接口发布进 OSPF。

（7）查看 R1 的路由学习情况：

r1#sh ip route

*Mar 1 00:28:10.043: %SYS-5-CONFIG_I: Configured from console by console

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/66] via 12.1.1.2, 00:01:23, Serial1/0

35.0.0.0/24 is subnetted, 1 subnets

O 35.1.1.0 [110/129] via 12.1.1.2, 00:01:23, Serial1/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/65] via 12.1.1.2, 00:01:23, Serial1/0

33.0.0.0/24 is subnetted, 1 subnets

O 33.3.3.0 [110/66] via 12.1.1.2, 00:01:23, Serial1/0

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/66] via 12.1.1.2, 00:01:23, Serial1/0

55.0.0.0/24 is subnetted, 1 subnets

O 55.5.5.0 [110/66] via 12.1.1.2, 00:01:23, Serial1/0

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/66] via 12.1.1.2, 00:01:24, Serial1/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/65] via 12.1.1.2, 00:01:24, Serial1/0

22.0.0.0/24 is subnetted, 1 subnets

O 22.2.2.0 [110/65] via 12.1.1.2, 00:01:24, Serial1/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 12.1.1.2, 00:01:24, Serial1/0

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/66] via 12.1.1.2, 00:01:24, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

r1#

说明：R1 已经学习到全网的每一条路由。

（8）查看 R2 的路由学习情况：

```
r2#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

       E1 - OSPF external type 1, E2 - OSPF external type 2

       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

       ia - IS-IS inter area, * - candidate default, U - per-user static route

       o - ODR, P - periodic downloaded static route


Gateway of last resort is not set


     34.0.0.0/24 is subnetted, 1 subnets

O       34.1.1.0 [110/2] via 23.1.1.3, 00:01:41, FastEthernet0/0

     35.0.0.0/24 is subnetted, 1 subnets

O       35.1.1.0 [110/65] via 25.1.1.5, 00:01:41, FastEthernet0/1

                 [110/65] via 23.1.1.3, 00:01:41, FastEthernet0/0

     2.0.0.0/24 is subnetted, 1 subnets

C       2.2.2.0 is directly connected, Loopback0

     33.0.0.0/24 is subnetted, 1 subnets

O       33.3.3.0 [110/2] via 23.1.1.3, 00:01:41, FastEthernet0/0
```


http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/2] via 23.1.1.3, 00:01:41, FastEthernet0/0

55.0.0.0/24 is subnetted, 1 subnets

O 55.5.5.0 [110/2] via 25.1.1.5, 00:01:41, FastEthernet0/1

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/2] via 25.1.1.5, 00:01:42, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/0

22.0.0.0/24 is subnetted, 1 subnets

C 22.2.2.0 is directly connected, Loopback22

25.0.0.0/24 is subnetted, 1 subnets

C 25.1.1.0 is directly connected, FastEthernet0/1

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/2] via 25.1.1.5, 00:01:42, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

r2#

说明：R2 已经学习到全网的每一条路由。

（9）查看 R3 的路由学习情况：

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/1

35.0.0.0/24 is subnetted, 1 subnets

C 35.1.1.0 is directly connected, Serial1/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/2] via 23.1.1.2, 00:01:56, FastEthernet0/0

33.0.0.0/24 is subnetted, 1 subnets

C 33.3.3.0 is directly connected, Loopback33

3.0.0.0/24 is subnetted, 1 subnets

C 3.3.3.0 is directly connected, Loopback0

55.0.0.0/24 is subnetted, 1 subnets

O 55.5.5.0 [110/3] via 23.1.1.2, 00:01:56, FastEthernet0/0

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/3] via 23.1.1.2, 00:01:57, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/0

22.0.0.0/24 is subnetted, 1 subnets

O 22.2.2.0 [110/2] via 23.1.1.2, 00:01:57, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/2] via 23.1.1.2, 00:01:57, FastEthernet0/0

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/3] via 23.1.1.2, 00:01:57, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/65] via 23.1.1.2, 00:01:57, FastEthernet0/0

r3#

说明：R3 已经学习到全网的每一条路由。

（10）查看 R4 的路由学习情况：

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/1

35.0.0.0/24 is subnetted, 1 subnets

O 35.1.1.0 [110/65] via 34.1.1.3, 00:02:09, FastEthernet0/1

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/3] via 34.1.1.3, 00:02:09, FastEthernet0/1

33.0.0.0/24 is subnetted, 1 subnets

O 33.3.3.0 [110/2] via 34.1.1.3, 00:02:09, FastEthernet0/1

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/2] via 34.1.1.3, 00:02:09, FastEthernet0/1

55.0.0.0/24 is subnetted, 1 subnets

O 55.5.5.0 [110/4] via 34.1.1.3, 00:02:09, FastEthernet0/1

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/4] via 34.1.1.3, 00:02:10, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/2] via 34.1.1.3, 00:02:10, FastEthernet0/1

22.0.0.0/24 is subnetted, 1 subnets

O 22.2.2.0 [110/3] via 34.1.1.3, 00:02:10, FastEthernet0/1

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/3] via 34.1.1.3, 00:02:10, FastEthernet0/1

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/4] via 34.1.1.3, 00:02:10, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/66] via 34.1.1.3, 00:02:10, FastEthernet0/1

r4#

说明：R4 已经学习到全网的每一条路由。

（11）查看 R5 的路由学习情况：

r5#sh ip route

*Mar 1 00:29:03.083: %SYS-5-CONFIG_I: Configured from console by console

r5#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/3] via 25.1.1.2, 00:02:25, FastEthernet0/1

35.0.0.0/24 is subnetted, 1 subnets

C 35.1.1.0 is directly connected, Serial1/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/2] via 25.1.1.2, 00:02:25, FastEthernet0/1

33.0.0.0/24 is subnetted, 1 subnets

O 33.3.3.0 [110/3] via 25.1.1.2, 00:02:25, FastEthernet0/1

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/3] via 25.1.1.2, 00:02:25, FastEthernet0/1

55.0.0.0/24 is subnetted, 1 subnets

C 55.5.5.0 is directly connected, Loopback55

5.0.0.0/24 is subnetted, 1 subnets

C 5.5.5.0 is directly connected, Loopback0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/2] via 25.1.1.2, 00:02:25, FastEthernet0/1

22.0.0.0/24 is subnetted, 1 subnets

O 22.2.2.0 [110/2] via 25.1.1.2, 00:02:25, FastEthernet0/1

25.0.0.0/24 is subnetted, 1 subnets

C 25.1.1.0 is directly connected, FastEthernet0/1

56.0.0.0/24 is subnetted, 1 subnets

C 56.1.1.0 is directly connected, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/65] via 25.1.1.2, 00:02:25, FastEthernet0/1

r5#

说明：R5 已经学习到全网的每一条路由。

（12）查看 R6 的路由学习情况：

r6#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/4] via 56.1.1.5, 00:02:38, FastEthernet0/0

35.0.0.0/24 is subnetted, 1 subnets

O 35.1.1.0 [110/65] via 56.1.1.5, 00:02:38, FastEthernet0/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/3] via 56.1.1.5, 00:02:38, FastEthernet0/0

33.0.0.0/24 is subnetted, 1 subnets

O 33.3.3.0 [110/4] via 56.1.1.5, 00:02:38, FastEthernet0/0

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/4] via 56.1.1.5, 00:02:38, FastEthernet0/0

55.0.0.0/24 is subnetted, 1 subnets

O 55.5.5.0 [110/2] via 56.1.1.5, 00:02:38, FastEthernet0/0

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/2] via 56.1.1.5, 00:02:38, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/3] via 56.1.1.5, 00:02:38, FastEthernet0/0

22.0.0.0/24 is subnetted, 1 subnets

O 22.2.2.0 [110/3] via 56.1.1.5, 00:02:38, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/2] via 56.1.1.5, 00:02:38, FastEthernet0/0

56.0.0.0/24 is subnetted, 1 subnets

C 56.1.1.0 is directly connected, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/66] via 56.1.1.5, 00:02:38, FastEthernet0/0

r6#

说明：R6 已经学习到全网的每一条路由。

（13）在 R1 上测试到其它网段的连通性：

r1#ping 2.2.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/98/200 ms

r1#

r1#ping 22.2.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 22.2.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/136/196 ms

r1#

r1#

r1#ping 3.3.3.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 76/124/172 ms

r1#

r1#ping 33.3.3.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 33.3.3.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 60/136/240 ms

r1#

r1#

r1#ping 5.5.5.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/135/252 ms

r1#

r1#ping 55.5.5.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 55.5.5.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/80/104 ms

r1#

r1#ping 34.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 34.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 76/191/276 ms

r1#

r1#

r1#ping 56.1.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 56.1.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 88/134/220 ms

r1#

说明： R1 与 2.2.2.0/24、22.2.2.0/24、3.3.3.0/24、33.3.3.0/24、5.5.5.0/24、55.5.5.0/24、34.1.1.0/24、56.1.1.0/24 通信正常，说明已经全网单播互通。

2. 配置 PIM-SM

（1）在 R1 上配置 PIM-SM：

r1(config)#ip multicast-routing

r1(config)#int s1/0

r1(config-if)#ip pim sparse-mode

r1(config-if)#ip igmp join-group 224.1.1.1

r1(config-if)#exit

说明：在 R1 上开启组播功能，并且在接口 S1/0 下开启 PIM-SM 模式，接口 S1/0 加入组 224.1.1.1，成为组成员。

（2）在 R2 上配置 PIM-SM：

r2(config)#ip multicast-routing

r2(config)#int s1/0

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#int f0/0

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#

r2(config)#int f0/1

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#int loopback 22

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#

说明：在 R2 上开启组播功能，并且在接口 Loopback 22，F0/0，F0/1，S1/0 下开启 PIM-SM 模式，因为 Loopback 0 只用来建 MSDP 邻居，并不运行组播，所以没有必要运行 PIM。

（3）在 R3 上配置 PIM-SM：

r3(config)#int f0/0

r3(config-if)#ip pim sparse-mode

r3(config-if)#exit

r3(config)#

r3(config)#int f0/1

r3(config-if)#ip pim sparse-mode

r3(config-if)#exit

r3(config)#int s1/0

r3(config-if)#ip pim sparse-mode

r3(config-if)#exit

r3(config-if)#int loopback 33

r3(config-if)#ip pim sparse-mode

r3(config-if)#exit

r3(config)#

说明：在 R3 上开启组播功能，并且在接口 Loopback 33，F0/0，F0/1，S1/0 下开启 PIM-SM 模式，因为 Loopback 0 只用来建 MSDP 邻居，并不运行组播，所以没有必要运行 PIM。

（4）在 R4 上配置 PIM-SM：

r4(config)#ip multicast-routing

r4(config)#int f0/1

r4(config-if)#ip pim sparse-mode

r4(config-if)#exit

说明：在 R4 上开启组播功能，并且在接口 F0/1 下开启 PIM-SM 模式。

（5）在 R5 上配置 PIM-SM：

r5(config)#ip multicast-routing

r5(config)#int f0/1

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

r5(config)#int f0/0

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

r5(config)#

r5(config)#int s1/0

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

r5(config)#int loopback 55

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

说明：在 R5 上开启组播功能，并且在接口 Loopback 55，F0/0，F0/1，S1/0 下开启 PIM-SM 模式。

（6）在 R6 上配置 PIM-SM：

r6(config)#ip multicast-routing

r6(config)#int f0/0

r6(config-if)#ip pim sparse-mode

r6(config-if)#exit

说明：在 R6 上开启组播功能，并且在接口 F0/0 下开启 PIM-SM 模式

（7）在 R2 上查看 PIM 邻居情况：

r2#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
12.1.1.1	Serial1/0	00:09:22/00:01:15	v2	1 / S
23.1.1.3	FastEthernet0/0	00:04:22/00:01:19	v2	1 / DR S
25.1.1.5	FastEthernet0/1	00:01:59/00:01:43	v2	1 / DR S

r2#

说明：R2 与 R1，R3 以及 R5 建立 PIM 邻居关系，一切正常。

（8）在 R3 上查看 PIM 邻居情况：

r3#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
23.1.1.2	FastEthernet0/0	00:04:41/00:01:29	v2	1 / S
34.1.1.4	FastEthernet0/1	00:02:42/00:01:29	v2	1 / DR S
35.1.1.5	Serial1/0	00:02:07/00:01:35	v2	1 / DR S

r3#

说明：R3 与 R2，R4 以及 R5 建立 PIM 邻居关系，一切正常。

（9）在 R5 上查看 PIM 邻居情况：

r5#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
25.1.1.2	FastEthernet0/1	00:02:02/00:01:39	v2	1 / S
56.1.1.6	FastEthernet0/0	00:01:08/00:01:35	v2	1 / DR S
35.1.1.3	Serial1/0	00:02:20/00:01:22	v2	1 / S

r5#

说明：R5 与 R2，R3 以及 R6 建立 PIM 邻居关系，一切正常。

（10）将 R2 配置为 PIM-SM Domain 1 的 RP：

r2(config)#access-list 1 deny 224.0.1.39

r2(config)#access-list 1 deny 224.0.1.40

r2(config)#access-list 1 permit any

r2(config)#

r2(config)#access-list 24 permit 224.1.1.1

r2(config)#

r2(config)#int f0/0

r2(config-if)#ip multicast boundary 1

r2(config-if)#exit

r2(config)#

r2(config)#int f0/1

r2(config-if)#ip multicast boundary 1

r2(config-if)#exit

r2(config)#ip pim send-rp-announce loopback 22 scope 16 group-list 24

r2(config)#ip pim send-rp-discovery loopback 22 scope 16

说明：R2 以 Loopback 22 作为组 224.1.1.1 的 RP 和映射代理；因为网络划分为 PIM-SM Domain 1、PIM-SM Domain 2 以及 PIM-SM Domain 3 共 3 个 PIM-SM 域，所以必须配置 PIM-SM 域边界，这里将 R2 连接 PIM-SM Domain 2 与 PIM-SM Domain 3 的接口 F0/0 和 F0/1 都配置成了 PIM-SM 域边界，后面还需在 PIM-SM Domain 2 与 PIM-SM Domain 3 之间配置 PIM-SM 域边界，在 R3 或 R5 上配置都可以。

（11）将 R3 配置为 PIM-SM Domain 2 的 RP：

r3(config)#access-list 1 deny 224.0.1.39

r3(config)#access-list 1 deny 224.0.1.40

r3(config)#access-list 1 permit any

r3(config)#

r3(config)#access-list 24 permit 224.1.1.1

r3(config)#

r3(config)#int s1/0

r3(config-if)#ip multicast boundary 1

r3(config-if)#exit

r3(config)#ip pim send-rp-announce loopback 33 scope 16 group-list 24

r3(config)#ip pim send-rp-discovery loopback 33 scope 16

说明：R3 以 Loopback 33 作为组 224.1.1.1 的 RP 和映射代理；并将连接 PIM-SM Domain 3 的接口 S1/0 都配置成了 PIM-SM 域边界，到这里为止，PIM-SM Domain 1、PIM-SM Domain 2 以及 PIM-SM Domain 3 已经被成功分割开，相互独立。

（12）将 R5 配置为 PIM-SM Domain 3 的 RP：

r5(config)#access-list 24 permit 224.1.1.1

r5(config)#

r5(config)#ip pim send-rp-announce loopback 55 scope 16 group-list 24

r5(config)#ip pim send-rp-discovery loopback 55 scope 16

说明：R5 以 Loopback 55 作为组 224.1.1.1 的 RP 和映射代理。

（13）在 R1 上查看 RP 学习情况：

r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 22.2.2.2 (?), v2v1

Info source: 22.2.2.2 (?), elected via Auto-RP

Uptime: 00:02:37, expires: 00:02:21

r1#

说明：PIM-SM Domain 1 中的 R1 学习到的 RP 正是 22.2.2.2，和预期一样，也证明 PIM-SM 域已被成功隔离。

（14）在 R2 上查看 RP 学习情况：

r2#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback22)

Group(s) 224.1.1.1/32

RP 22.2.2.2 (?), v2v1

Info source: 22.2.2.2 (?), elected via Auto-RP

Uptime: 00:02:50, expires: 00:02:07

r2#

说明：PIM-SM Domain 1 中的 R2 学习到的 RP 正是 22.2.2.2，和预期一样，也证明 PIM-SM 域已被成功隔离。

（15）在 R3 上查看 RP 学习情况：

r3#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback33)

Group(s) 224.1.1.1/32

RP 33.3.3.3 (?), v2v1

Info source: 33.3.3.3 (?), elected via Auto-RP

Uptime: 00:01:34, expires: 00:02:23

r3#

说明：PIM-SM Domain 2 中的 R3 学习到的 RP 正是 33.3.3.3，和预期一样，也证明 PIM-SM 域已被成功隔离。

（16）在 R4 上查看 RP 学习情况：

r4#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 33.3.3.3 (?), v2v1

Info source: 33.3.3.3 (?), elected via Auto-RP

Uptime: 00:01:47, expires: 00:02:12

r4#

说明：PIM-SM Domain 2 中的 R4 学习到的 RP 正是 33.3.3.3，和预期一样，也证明 PIM-SM 域已被成功隔离。

（17）在 R5 上查看 RP 学习情况：

r5#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback55)

Group(s) 224.1.1.1/32

RP 55.5.5.5 (?), v2v1

Info source: 55.5.5.5 (?), elected via Auto-RP

Uptime: 00:01:11, expires: 00:02:47

r5#

说明：PIM-SM Domain 3 中的 R5 学习到的 RP 正是 55.5.5.5，和预期一样，也证明 PIM-SM 域已被成功隔离。

（18）在 R6 上查看 RP 学习情况：

说明：PIM-SM Domain 3 中的 R6 学习到的 RP 正是 55.5.5.5，和预期一样，也证明 PIM-SM 域已被成功隔离。

（19）测试组播源 R4 和 R6 到组 224.1.1.1 的通信情况：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r4#

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r6#

说明：因为 PIM-SM 域之间已经被隔离，各自的 RP 都没有完整的组播源和组成员信息，所以在没有 MSDP 共享组播源信息的情况下，域间组播是不可能过的。

3. 配置 MSDP

（1）在 2 个 PIM-SM 域之间建立 MSDP 连接：

r2(config)#ip msdp peer 3.3.3.3 connect-source loopback 0

r2(config)#ip msdp peer 55.5.5.5 connect-source loopback 0

r3(config)#ip msdp peer 2.2.2.2 connect-source loopback 0

r3(config)#ip msdp peer 55.5.5.5 connect-source loopback 0

r5(config)#ip msdp peer 2.2.2.2 connect-source loopback 55

r5(config)#ip msdp peer 3.3.3.3 connect-source loopback 55

说明：R2，R3 以及 R5 之间建立全互联的 MSDP 邻居关系，其中 R2 和 R3 都是使用接口 Loopback 0 的地址作为 MSDP 源地址，而 R5 使用接口 Loopback 55 的地址作为 MSDP 源地址，连接情况如下：

R2（2.2.2.2）—— R3（3.3.3.3）

R2（2.2.2.2）—— R5（55.5.5.5）

R3（3.3.3.3）—— R5（55.5.5.5）

正因为 R5 的 RP 地址也是 55.5.5.5，而 MSDP 的源地址也是 55.5.5.5，如果收到的 SA 数据包的发送方 MSDP peer 的 IP 地址与 SA 数据包中的 RP 地址相同时，是不对 SA 数据包作 RPF 检测的，所以在任何 MSDP peer 收到 R5 的 SA 时都不做 RPF 检测。

（2）查看 MSDP 的连接情况：

r2#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

3.3.3.3 ? Up 00:01:17 0 0 ?

55.5.5.5 ? Up 00:00:17 0 0 ?

r2#

r3#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

2.2.2.2 ? Up 00:01:41 0 0 ?

55.5.5.5 ? Up 00:01:03 0 0 ?

r3#

R5：

r5#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

2.2.2.2 ? Up 00:00:53 0 0 ?

3.3.3.3 ? Up 00:01:16 0 0 ?

r5#

说明：R2，R3 以及 R5 之间的全互联 MSDP 连接已经全部正常建立。

（3）在 R2 上查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 0 entries

r2#

说明：因为目前没有路由器发起组播流量，也就没有组播源，那么也就不会产生 SA 信息。

（4）在组播源路由器 R4 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

在 MSDP 路由器 R2 上开启 Debug:

r2#debug ip msdp peer

MSDP Peer debugging is on

r2#

r2#debug ip msdp detail

MSDP Detail debugging is on

r2#

在组播源路由器 R4 上发起组播流量：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r4#

说明：组播不通，后面将根据实际情况一步一步解决。

观察 R2 上的 Debug 信息：

r2#

*Mar 1 01:02:35.471: MSDP(0): Received 20-byte TCP segment from 3.3.3.3

*Mar 1 01:02:35.475: MSDP(0): Append 20 bytes to 0-byte msg 16 from 3.3.3.3, qs 1

*Mar 1 01:02:35.475: MSDP(0): 3.3.3.3: Received 20-byte msg 16 from peer

*Mar 1 01:02:35.475: MSDP(0): 3.3.3.3: SA TLV, len: 20, ec: 1, RP: 33.3.3.3

*Mar 1 01:02:35.479: MSDP(0): 3.3.3.3: Peer RPF check failed for 33.3.3.3, used ? route's peer 0.0.0.0

r2#

说明：因为目前 R2 有 1 个以上的 MSDP peer（R3 和 R5 共 2 个），所以对于收到的 SA 必须进行 RPF 检测，而对 SA 的 RPF 检测必须依靠 BGP 或 MBGP 路由表，而我们并没有配置 BGP 和 MBGP，所以 Debug 信息中显示对于 MSDP peer 3.3.3.3 发来的 SA 做 RPF 检测失败了，在这里还请记住，如果收到的 SA 数据包的发送方 MSDP peer 的 IP 地址与 SA 数据包中的 RP 地址相同时，是不对 SA 数据包作 RPF 检测的，但因为 R3 的 MSDP 地址是 3.3.3.3，而它发出的 SA 里面的 RP 地址是 33.3.3.3，不满足不做 RPF 检测的要求，所以必须对 R3 发来的 SA 做 RPF 检测。

（5）在 R2 上再次查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 0 entries

r2#

说明：由于对 R3 发来的 SA 数据包 RPF 检测失败，所以 R2 并没有缓存任何可用的 SA 信息。

（6）尝试在组播源路由器 R6 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 368 ms

Reply to request 1 from 12.1.1.1, 296 ms

Reply to request 2 from 12.1.1.1, 208 ms

Reply to request 3 from 12.1.1.1, 264 ms

Reply to request 4 from 12.1.1.1, 248 ms

Reply to request 5 from 12.1.1.1, 264 ms

Reply to request 6 from 12.1.1.1, 248 ms

Reply to request 7 from 12.1.1.1, 224 ms

Reply to request 8 from 12.1.1.1, 248 ms

Reply to request 9 from 12.1.1.1, 264 ms

r6#

说明：PIM-SM Domain 3 与 PIM-SM Domain 1 的组播居然通了，观察 R2 上的 Debug 信息，看看发生了什么：

r2#

*Mar 1 01:04:39.351: MSDP(0): Received 120-byte TCP segment from 55.5.5.5

*Mar 1 01:04:39.355: MSDP(0): Append 120 bytes to 0-byte msg 21 from 55.5.5.5, qs 1

*Mar 1 01:04:39.355: MSDP(0): 55.5.5.5: Received 120-byte msg 21 from peer

*Mar 1 01:04:39.355: MSDP(0): 55.5.5.5: SA TLV, len: 120, ec: 1, RP: 55.5.5.5, with data

*Mar 1 01:04:39.359: MSDP(0): 55.5.5.5: Peer RPF check passed for 55.5.5.5, peer is RP

*Mar 1 01:04:39.359: MSDP(0): WAVL Insert SA Source 56.1.1.6 Group 224.1.1.1 RP 55.5.5.5 Successful

r2#

说明：因为 R5 的 MSDP peer 地址 55.5.5.5 与 SA 数据包中的 RP 地址 55.5.5.5 相同，满足不对 SA 数据包作 RPF 检测的条件，所以 R5 发来的 SA 数据包可以直接使用，从 R2 的 Debug 信息中也看出的确如此，这就是最终 PIM-SM Domain 3 与 PIM-SM Domain 1 组播通信成功的原因。

（7）在 R2 上再次查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 1 entries

(56.1.1.6, 224.1.1.1), RP 55.5.5.5, AS ?,00:03:54/00:04:42, Peer 55.5.5.5

Learned from peer 55.5.5.5, RPF peer 55.5.5.5,

SAs received: 4, Encapsulated data received: 1

r2#

说明：R5 发来的 SA 信息已被成功缓存并使用。

4. 配置 MBGP 帮助 MSDP 的 SA 通过 RPF 检测

说明：因为对 SA 的 RPF 检测必须依靠 BGP 或 MBGP 路由表来完成，所以我们在这里选择采取配置 MBGP 的方式，来帮助 R2 对 PIM-SM Domain 3 的 R3 发来的 SA 通过 RPF 检测，从而实现两个 PIM-SM 域之间的组播通信。

（1）在 R2 和 R3 之间配置 MBGP：

R2：

r2(config)#router bgp 1

r2(config-router)#neighbor 3.3.3.3 remote-as 2

r2(config-router)#neighbor 3.3.3.3 update-source loopback 0

r2(config-router)#neighbor 3.3.3.3 ebgp-multihop

r2(config-router)#address-family ipv4 multicast

r2(config-router-af)#neighbor 3.3.3.3 activate

r2(config-router-af)#exit

r2(config-router)#exit

r2(config)#

R3：

r3(config)#router bgp 2

r3(config-router)#neighbor 2.2.2.2 remote-as 1

r3(config-router)#neighbor 2.2.2.2 update-source loopback 0

r3(config-router)#neighbor 2.2.2.2 ebgp-multihop

r3(config-router)#address-family ipv4 multicast

r3(config-router-af)#neighbor 2.2.2.2 activate

r3(config-router-af)#network 33.3.3.0 mask 255.255.255.0

r3(config-router-af)#exit

r3(config-router)#exit

r3(config)#

说明：为了更好的帮助 SA 通过 RPF 检测，所以使用了与建立 MSDP peer 相同的地址来建立 MBGP；因为 SA 的 RPF 检测是基于 SA 中 RP 的 IP 地址的，所以这里将 RP 的地址段 33.3.3.0/24 发布进了 MBGP，但并没有将组播源的地址段 34.1.1.0/24 发布进 MBGP，因为这条路由在 IGP 里面已经有了，对普通组播流的 RPF 检测可以基于静态组播路由表、DVMRP 路由表、MBGP 路由表、单播路由表等多种形式。

（2）查看 BGP 与 MBGP 邻居状态：

r2#show ip bgp summary

BGP router identifier 22.2.2.2, local AS number 1

BGP table version is 1, main routing table version 1

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd

3.3.3.3 4 2 2 2 0 0 0 00:00:26 0

r2#

r2#sh ip bgp ipv4 multicast summary

BGP router identifier 22.2.2.2, local AS number 1

BGP table version is 2, main routing table version 2

1 network entries using 129 bytes of memory

1 path entries using 48 bytes of memory

2/1 BGP path/bestpath attribute entries using 248 bytes of memory

1 BGP AS-PATH entries using 24 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 449 total bytes of memory

BGP activity 1/0 prefixes, 1/0 paths, scan interval 60 secs

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd

3.3.3.3 4 2 4 3 2 0 0 00:00:32 1

r2#

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

说明：R2 与 R3 之间的 BGP 和 MBGP 邻居均已建立，并且邻居类型为 eBGP，请注意对方 R3 的 AS 号码为 2，这个 AS 号码关系到它发来的 SA 能不能通过 RPF 检测，因为当 MSDP peer 是 eBGP 邻居的情况下，是看去往 SA 数据包中 RP 地址的最佳路径的下一跳 AS 号码是不是这个 eBGP 邻居的 AS 号码，如果是，则检测通过，否则检测失败，这里只要去往 RP 的地址 33.3.3.3 的 BGP 最佳路径的下一跳 AS 号码和 R3 的 AS 号码 2 一样，那么就能 SA 通过 RPF 检测，否则失败。

（3）查看 MBGP 路由表：

r2#sh ip bgp ipv4 multicast

BGP table version is 8, local router ID is 22.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 33.3.3.0/24	3.3.3.3	0		0	2 i

r2#

说明：MBGP 路由表中已经正常收到关于 R3 发来的 SA 数据包中的 RP 地址段，并且显示最佳路径的下一跳 AS 号码正是 eBGP 邻居 R3 的 AS 号码 2 相同，所以 RPF 检测一定通过。

（4）再次在组播源路由器 R4 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 396 ms

Reply to request 0 from 12.1.1.1, 444 ms

Reply to request 1 from 12.1.1.1, 240 ms

Reply to request 2 from 12.1.1.1, 288 ms

Reply to request 3 from 12.1.1.1, 220 ms

Reply to request 4 from 12.1.1.1, 216 ms

Reply to request 5 from 12.1.1.1, 288 ms

Reply to request 6 from 12.1.1.1, 304 ms

Reply to request 7 from 12.1.1.1, 240 ms

Reply to request 8 from 12.1.1.1, 196 ms

Reply to request 9 from 12.1.1.1, 188 ms

r4#

说明：和预期一样，通过 MBGP 解决 SA 的 RPF 检测问题之后，PIM-SM Domain 2 与 PIM-SM Domain 1 的组播通信已经变的正常。

观察 R2 上的 Debug 信息：

r2#

*Mar 1 01:19:43.883: MSDP(0): Received 120-byte TCP segment from 3.3.3.3

*Mar 1 01:19:43.883: MSDP(0): Append 120 bytes to 0-byte msg 56 from 3.3.3.3,

qs 1

*Mar 1 01:19:43.887: MSDP(0): 3.3.3.3: Received 120-byte msg 56 from peer

*Mar 1 01:19:43.887: MSDP(0): 3.3.3.3: SA TLV, len: 120, ec: 1, RP: 33.3.3.3, with data

*Mar 1 01:19:43.887: MSDP(0): 3.3.3.3: Peer RPF check passed for 33.3.3.3, used EMBGP peer

*Mar 1 01:19:43.891: MSDP(0): Forward decapsulated SA data for (34.1.1.4, 224.1.1.1) on Serial1/0

r2#

说明：和预期一样，R2 基于 MBGP 对 R3 发来的 SA 做 RPF 检测成功。

（5）在 R2 上再次查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 2 entries

(34.1.1.4, 224.1.1.1), RP 33.3.3.3, MBGP/AS 2, 00:00:16/00:05:43, Peer 3.3.3.3

(56.1.1.6, 224.1.1.1), RP 55.5.5.5, BGP/AS 0, 00:00:27/00:05:32, Peer 55.5.5.5

r2#

说明：R3 和 R5 发来的 SA 信息已被 R2 成功缓存并使用，所以现在 3 个 PIM-SM 域之间的组播通信均正常。

（6）查看组播树情况：

R3：

r3#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:08:20/stopped, RP 33.3.3.3, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(34.1.1.4, 224.1.1.1), 00:00:13/00:03:24, flags: TA

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:00:13/00:03:16

r3#

R2：

r2#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:49:23/00:02:28, RP 22.2.2.2, flags: SJC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:49:23/00:02:54

(34.1.1.4, 224.1.1.1), 00:00:31/00:02:37, flags: MT

Incoming interface: FastEthernet0/0, RPF nbr 23.1.1.3, Mbgp

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:00:31/00:02:54

(56.1.1.6, 224.1.1.1), 00:02:43/00:01:07, flags: T

Incoming interface: FastEthernet0/1, RPF nbr 25.1.1.5

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:02:43/00:02:53

r2#

说明：从组播树中可以看见，PIM-SM 域之间的通信使用的是（S，G），而并非 PIM-SM 模式传统的（*，G），各组流量走向进出口完全和拓朴中一致。

查看当前环境中所有路由器的配置信息请点击“running-config”

5. 测试 default MSDP peer 帮助 MSDP 的 SA 通过 RPF 检测

说明：目前 3 个 PIM-SM 域之间的组播通信完全正常，所以想要测试 default MSDP peer 帮助 MSDP 的 SA 通过 RPF 检测，首先必须制造 SA 无法通过 RPF 检测的环境，才能够利用 default MSDP peer 帮助 MSDP 的 SA 通过 RPF 检测；根据上图的实验环境，我们采用过滤掉 PIM-SM Domain 3 的 R5 直接向 PIM-SM Domain 1 的 R2 发送 SA，但还是正常发给 PIM-SM Domain 2 的 R3，这样一来，PIM-SM Domain 3 和 PIM-SM Domain 2 的 SA 都必须通过 R3 发给 R2，这就势必会造成 PIM-SM Domain 3 的 SA 因为 RPF 检测不通过而被 R2 丢弃，但 PIM-SM Domain 2 的 SA 和之前一样保持正常。

（1）将 PIM-SM Domain 3 的 R5 向 PIM-SM Domain 1 的 R2 发送的 SA 全部过滤掉：

r5(config)#ip msdp sa-filter out 2.2.2.2

说明：如果命令后面不跟任何匹配工具，则表示所有 SA。

（2）清除 R2 上的 SA 缓存信息：

r2#clear ip msdp sa-cache

r2#

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 0 entries

r2#

说明：R2 上的 SA 缓存信息已被全部清空。

（3）在组播源路由器 R6 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r6#

说明：和预期一样，PIM-SM Domain 3 与 PIM-SM Domain 1 的组播不通，这一定是 SA 数据包无法通过 RPF 检测引起的.

观察 R2 上的 Debug 信息：

r2#

*Mar 1 01:35:56.891: MSDP(0): Received 120-byte TCP segment from 3.3.3.3

*Mar 1 01:35:56.891: MSDP(0): Append 120 bytes to 0-byte msg 102 from 3.3.3.3, qs 1

*Mar 1 01:35:56.891: MSDP(0): 3.3.3.3: Received 120-byte msg 102 from peer

*Mar 1 01:35:56.891: MSDP(0): 3.3.3.3: SA TLV, len: 120, ec: 1, RP: 55.5.5.5, with data

*Mar 1 01:35:56.895: MSDP(0): 3.3.3.3: Peer RPF check failed for 55.5.5.5, used IBGP route's peer 0.0.0.0

r2#

说明：正因为 PIM-SM Domain 3 的 R5 直接向 PIM-SM Domain 1 的 R2 发送的 SA 被过滤掉了，所以这时 PIM-SM Domain 3 的 SA 是先发给 PIM-SM Domain 2 的 R3，然后由 R3 再发给 R2 的，这样一来，在对收到的 SA 做 RPF 检测时，由于 SA 数据包中 RP 地址 55.5.5.5 根本就无法在 BGP 或 MBGP 路由表中找到，也就无法做 RPF 检测，所以最终因检测失败而被丢弃了。

（4）再次在组播源路由器 R4 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 296 ms

Reply to request 1 from 12.1.1.1, 188 ms

Reply to request 2 from 12.1.1.1, 344 ms

Reply to request 3 from 12.1.1.1, 320 ms

Reply to request 4 from 12.1.1.1, 392 ms

Reply to request 5 from 12.1.1.1, 336 ms

Reply to request 6 from 12.1.1.1, 168 ms

Reply to request 7 from 12.1.1.1, 172 ms

Reply to request 8 from 12.1.1.1, 324 ms

Reply to request 9 from 12.1.1.1, 308 ms

r4#

说明：因为 PIM-SM Domain 2 与 PIM-SM Domain 1 之间并没有任何变化，所以组播通信仍然保持正常.

观察 R2 上的 Debug 信息：

r2#

*Mar 1 01:37:11.759: MSDP(0): Received 120-byte TCP segment from 3.3.3.3

*Mar 1 01:37:11.763: MSDP(0): Append 120 bytes to 0-byte msg 105 from 3.3.3.3, qs 1

*Mar 1 01:37:11.763: MSDP(0): 3.3.3.3: Received 120-byte msg 105 from peer

*Mar 1 01:37:11.767: MSDP(0): 3.3.3.3: SA TLV, len: 120, ec: 1, RP: 33.3.3.3, with data

*Mar 1 01:37:11.767: MSDP(0): 3.3.3.3: Peer RPF check passed for 33.3.3.3, used EMBGP peer

*Mar 1 01:37:11.767: MSDP(0): WAVL Insert SA Source 34.1.1.4 Group 224.1.1.1 RP 33.3.3.3 Successful

r2#

说明：PIM-SM Domain 2 中的 SA 还是被 R2 正常接收并缓存。

（5）在 R2 上再次查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 1 entries

(34.1.1.4, 224.1.1.1), RP 33.3.3.3, MBGP/AS 2, 00:00:49/00:05:44, Peer 3.3.3.3

Learned from peer 3.3.3.3, RPF peer 3.3.3.3,

SAs received: 2, Encapsulated data received: 1

r2#

说明：从结果中看出，PIM-SM Domain 2 中的 SA 被 R2 正常接收并缓存，而 PIM-SM Domain 3 还在故障中。

（6）通过 default MSDP peer 帮助 MSDP 的 SA 通过 RPF 检测

r2(config)#ip msdp default-peer 3.3.3.3

说明：因为从 Default MSDP Peer 收到的所有 SA 都不需要做 RPF 检测，所以无论是 PIM-SM Domain 2 中的 SA，还是 PIM-SM Domain 3 先发给 PIM-SM Domain 2 的 R3，然后再由 R3 发过来的 SA，R2 都不再做 RPF 检测，结果就是所有 SA 都被有效缓存。

（7）再次在组播源路由器 R6 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 456 ms

Reply to request 1 from 12.1.1.1, 408 ms

Reply to request 2 from 12.1.1.1, 220 ms

Reply to request 3 from 12.1.1.1, 200 ms

Reply to request 4 from 12.1.1.1, 192 ms

Reply to request 5 from 12.1.1.1, 280 ms

Reply to request 6 from 12.1.1.1, 264 ms

Reply to request 7 from 12.1.1.1, 248 ms

Reply to request 8 from 12.1.1.1, 312 ms

Reply to request 9 from 12.1.1.1, 276 ms

r6#

说明：default MSDP peer 已经成功解决 PIM-SM 域间的 SA 检测与组播通信问题.

观察 R2 上的 Debug 信息：

r2#

*Mar 1 01:44:04.047: MSDP(0): Received 120-byte TCP segment from 3.3.3.3

*Mar 1 01:44:04.047: MSDP(0): Append 120 bytes to 0-byte msg 118 from 3.3.3.3, qs 1

*Mar 1 01:44:04.047: MSDP(0): 3.3.3.3: Received 120-byte msg 118 from peer

*Mar 1 01:44:04.051: MSDP(0): 3.3.3.3: SA TLV, len: 120, ec: 1, RP: 55.5.5.5, with data

*Mar 1 01:44:04.051: MSDP(0): 3.3.3.3: Peer RPF check passed for 55.5.5.5, used default peer

*Mar 1 01:44:04.051: MSDP(0): WAVL Insert SA Source 56.1.1.6 Group 224.1.1.1 RP 55.5.5.5 Successful

r2#

说明：由于配置了 default MSDP peer 的原因，原本 R5 发给 R3 再发给 R2 的 SA 不能通过 RPF 检测，现在已经被 R2 正常接收并缓存了。

（8）在 R2 上再次查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 2 entries

(34.1.1.4, 224.1.1.1), RP 33.3.3.3, MBGP/AS 2, 00:07:52/00:03:56, Peer 3.3.3.3

Learned from peer 3.3.3.3, RPF peer 3.3.3.3,

SAs received: 3, Encapsulated data received: 1

(56.1.1.6, 224.1.1.1), RP 55.5.5.5, BGP/AS 0, 00:01:00/00:05:43, Peer 3.3.3.3

Learned from peer 3.3.3.3, RPF peer 3.3.3.3,

SAs received: 2, Encapsulated data received: 1

r2#

说明：无论 PIM-SM Domain 2 还是 PIM-SM Domain 3 的 SA，在 R3 发给 R2 后都被正常接收并缓存了。

查看当前环境中所有路由器的配置信息请点击“running-config”

6. 测试 MSDP Mesh Group 帮助 MSDP 的 SA 通过 RPF 检测

说明：目前 3 个 PIM-SM 域之间的组播通信完全正常，所以想要测试 MSDP Mesh Group 帮助 MSDP 的 SA 通过 RPF 检测，首先必须制造 SA 无法通过 RPF 检测的环境，才能够利用 MSDP Mesh Group 帮助 MSDP 的 SA 通过 RPF 检测；根据上图的实验环境，我们采用在 PIM-SM Domain 1 的 R2 与 PIM-SM Domain 3 的 R5 之间增加 BGP 连接，然后对于 PIM-SM Domain 2 中的 RP 地址 33.3.3.3 的网段信息，不再通过 R3 发给 R2，而通过 R5 发给 R2，从而使 SA 中 RP 地址（33.3.3.3）因为最佳路径的下一跳 AS 号码（现在为 3）与发送方 MSDP peer（R3）的 AS 号码（为 2）不匹配而导致 RPF 检测失败，从而被丢弃，然后我们在 R2 上将 R3 指定为 MSDP Mesh Group 中的成员，让 R3 发来的 SA 因为发送 MSDP peer 是 MSDP Mesh Group 中的成员而跳过 RPF 检测，最终解决 PIM-SM Domain 2 与 PIM-SM Domain 1 的域间组播通信。

（1）删除 default MSDP peer 配置：

r2(config)#no ip msdp default-peer 3.3.3.3

说明：因为 R2 对 R3 配置的 default MSDP peer 会让所有从 R3 发来的 SA 都跳过 RPF 检测而被缓存，所以必须先删除。

（2）删除 R5 对 R2 的 SA 过滤配置：

r5(config)#no ip msdp sa-filter out 2.2.2.2

说明：因为 R5 已经过滤掉发给 R2 的所有 SA，所以先将配置删除。

（3）配置 BGP 制造 RPF 检测故障（注：采用 MBGP 代替 BGP 也可以，两者一

样)：

R3 不再通告关于 RP 地址 33.3.3.3 的网段信息：

r3(config)#router bgp 2

r3(config-router)#address-family ipv4 multicast

r3(config-router-af)#no network 33.3.3.0 mask 255.255.255.0

R2 增加与 R5 之间的 BGP：

r2(config)#router bgp 1

r2(config-router)#neighbor 5.5.5.5 remote-as 3

r2(config-router)#neighbor 5.5.5.5 update-source loopback 0

r2(config-router)#neighbor 5.5.5.5 ebgp-multihop

r2(config-router)#exit

R3 取消在 MBGP 里通告关于 RP 地址 33.3.3.3 的网段：

r3(config)#router bgp 2

r3(config-router)#address-family ipv4 multicast

r3(config-router-af)#no network 33.3.3.0 mask 255.255.255.0

R5 替 R3 通告关于 RP 地址 33.3.3.3 的网段信息：

r5(config)#router bgp 3

r5(config-router)#neighbor 2.2.2.2 remote-as 1

r5(config-router)#neighbor 2.2.2.2 update-source loopback 0

r5(config-router)#neighbor 2.2.2.2 ebgp-multihop

r5(config-router)#network 33.1.1.0 mask 255.255.255.0

r5(config-router)#exit

说明：将 R5 配置为 BGP AS3，并将关于 RP 地址 33.3.3.3 的网段通告进 BGP*（因为 R5 已经通过 OSPF 从 R3 学习到该路由，所以可以使用 network 命令来通告），这就像是背地里对 R3 干的一个恶作剧。

（4）在 R2 上查看路由情况：

r2#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/2] via 23.1.1.3, 00:03:45, FastEthernet0/0

35.0.0.0/24 is subnetted, 1 subnets

O 35.1.1.0 [110/65] via 25.1.1.5, 00:21:26, FastEthernet0/1

[110/65] via 23.1.1.3, 00:21:26, FastEthernet0/0

2.0.0.0/24 is subnetted, 1 subnets

C 2.2.2.0 is directly connected, Loopback0

33.0.0.0/24 is subnetted, 1 subnets

B 33.3.3.0 [20/3] via 5.5.5.5, 00:01:54

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/2] via 23.1.1.3, 00:21:26, FastEthernet0/0

55.0.0.0/24 is subnetted, 1 subnets

O 55.5.5.0 [110/2] via 25.1.1.5, 00:21:26, FastEthernet0/1

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/2] via 25.1.1.5, 00:21:27, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/0

22.0.0.0/24 is subnetted, 1 subnets

C 22.2.2.0 is directly connected, Loopback22

25.0.0.0/24 is subnetted, 1 subnets

C 25.1.1.0 is directly connected, FastEthernet0/1

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/2] via 25.1.1.5, 00:21:27, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

r2#

说明：R2 上已经通过 BGP 从 R5 学习到关于 RP 地址 33.3.3.3 的网段。

（5）在 R2 上查看 MBGP 的路由情况：

r2#sh ip bgp ipv4 multicast

r2#

说明：因为 R3 已经取消在 MBGP 里通告关于 RP 地址 33.3.3.3 的网段，所以 R2 上的 MBGP 路由为空。

（6）清除掉 R2 上的 SA 缓存信息，然后在组播源路由器 R4 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

R2 上清除 SA：

r2#clear ip msdp sa-cache

r2#

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 0 entries

r2#

R4 上发组播：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r4#

说明：和预期一样，PIM-SM Domain 2 与 PIM-SM Domain 1 之间的组播不通.

观察 R2 上的 Debug 信息：

r2#

*Mar 1 02:09:02.451: MSDP(0): Received 120-byte TCP segment from 3.3.3.3

*Mar 1 02:09:02.455: MSDP(0): Append 120 bytes to 0-byte msg 173 from 3.3.3.3, qs 1

*Mar 1 02:09:02.455: MSDP(0): 3.3.3.3: Received 120-byte msg 173 from peer

*Mar 1 02:09:02.455: MSDP(0): 3.3.3.3: SA TLV, len: 120, ec: 1, RP: 33.3.3.3, with data

*Mar 1 02:09:02.459: MSDP(0): 3.3.3.3: Peer RPF check failed for 33.3.3.3, EBGP route/peer in AS 3/2

r2#

说明：R2 的 Debug 信息也表明 R3 发给 R2 的 SA 信息在基于 EBGP 做 RPF 检测时失败了，因而被丢弃。

（7）确认 R3 发给 R2 的 SA 数据包 RPF 检测失败的详细过程：

在 R2 上查看 MSDP peer 信息：

r2#show ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

3.3.3.3 2 Up 00:30:53 1 1 ?

55.5.5.5 ? Up 01:16:54 0 0 ?

r2#

说明：R3 作为 R2 的 MSDP peer，它的 AS 号码是 2，当 R2 对 R3 发来的 SA 做 RPF 检测时，SA 数据包中的 RP 地址在 BGP 路由表中最佳路径的下一跳 AS 号码必须也和 R3 的号码一样为 2，如果是其它号码就不能通过 RPF 检测。

在 R2 上查看 BGP 路由表中关于 RP 地址 33.3.3.3 的最佳路径情况：

r2#sh ip bgp

BGP table version is 6, local router ID is 22.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path

*> 33.3.3.0/24 5.5.5.5 3 0 3 i

r2#

说明：正是因为关于 RP 地址 33.3.3.3 的最佳路径的下一跳 AS 号码是 R5 的 AS 号码 3，与发送 MSDP peer R3 的 AS 号码 2 不一致，所以 R3 发来的 SA 不能通过

RPF 检测而被丢弃了。

（8）在 R2 上查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 0 entries

r2#

说明：由于 R3 发来的 SA 不能通过 RPF 检测，所以目前 R2 的 SA 缓存为空。

（9）通过 MSDP Mesh Group 帮助 MSDP 的 SA 通过 RPF 检测:

r2(config)#ip msdp mesh-group CCIE 3.3.3.3

说明：因为从 MSDP Mesh Group 收到的所有 SA 都不需要做 RPF 检测，所以后面 R3 发过来的 SA，R2 都不再做 RPF 检测就能被有效缓存。

（10）再次在组播源路由器 R4 上发起组播，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 360 ms

Reply to request 1 from 12.1.1.1, 344 ms

Reply to request 2 from 12.1.1.1, 264 ms

Reply to request 3 from 12.1.1.1, 304 ms

Reply to request 4 from 12.1.1.1, 300 ms

Reply to request 5 from 12.1.1.1, 308 ms

Reply to request 6 from 12.1.1.1, 388 ms

Reply to request 7 from 12.1.1.1, 320 ms

Reply to request 8 from 12.1.1.1, 264 ms

Reply to request 9 from 12.1.1.1, 452 ms

r4#

说明：MSDP Mesh Group 已经成功解决 PIM-SM 域间的 SA 检测与组播通信问题.

观察 R2 上的 Debug 信息：

r2#

*Mar 1 02:13:54.195: MSDP(0): Received 120-byte TCP segment from 3.3.3.3

*Mar 1 02:13:54.195: MSDP(0): Append 120 bytes to 0-byte msg 184 from 3.3.3.3, qs 1

*Mar 1 02:13:54.195: MSDP(0): 3.3.3.3: Received 120-byte msg 184 from peer

*Mar 1 02:13:54.199: MSDP(0): 3.3.3.3: SA TLV, len: 120, ec: 1, RP: 33.3.3.3, with data

*Mar 1 02:13:54.199: MSDP(0): 3.3.3.3: Peer RPF check bypassed, peer 3.3.3.3 in mesh-group CCIE

r2#

说明：由于配置了 MSDP Mesh Group 的原因，原本 R3 发给 R2 的 SA 不能通过 RPF 检测，现在已经被 R2 正常接收并缓存了。

（11）在 R2 上再次查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 1 entries

(34.1.1.4, 224.1.1.1), RP 33.3.3.3, BGP/AS 3, 00:03:56/00:05:50, Peer 3.3.3.3

Learned from peer 3.3.3.3, RPF peer 0.0.0.0,

SAs received: 4, Encapsulated data received: 1

r2#

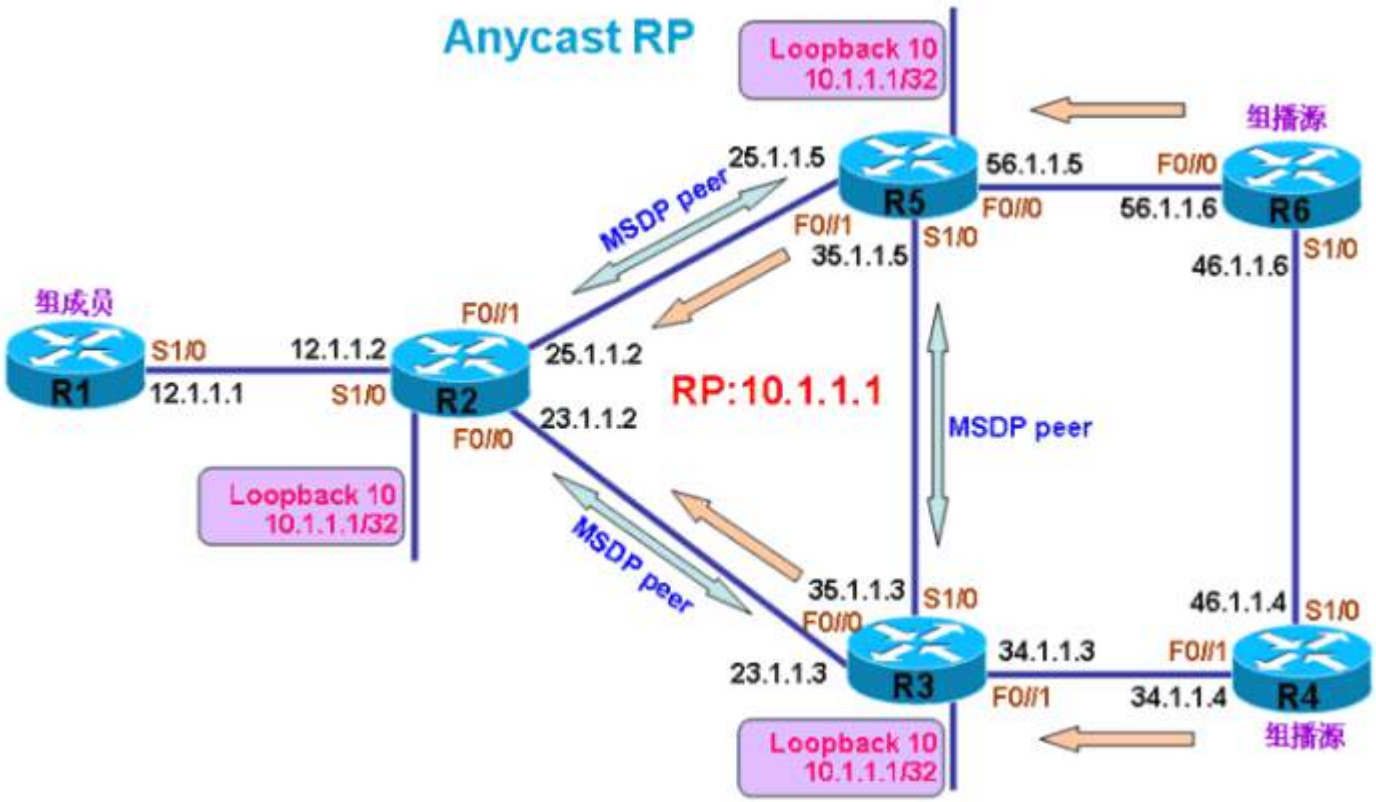
说明：由于配置了 MSDP Mesh Group 的原因，原本 R3 发给 R2 的 SA 不能通过 RPF 检测，现在已经被 R2 正常接收并缓存了。

查看当前环境中所有路由器的配置信息请点击“running-config”

配置 Anycast RP 实验

实验说明：

以下图为环境配置 Anycast RP 实验：



上图中所有路由器都在同一个 PIM-SM 域，其中 R4 和 R6 为组 224.1.1.1 的组播源，R1 为组成员；R2，R3 以及 R5 都配有 Loopback 10，并且地址都为 10.1.1.1/32；

除了图上所标识出的接口地址外，R2、R3 和 R5 均配有 Loopback 0 接口，分别为：

R2（Loopback 0 :2.2.2.2/24）

R3（Loopback 0 :3.3.3.3/24）

R5（Loopback 0 :5.5.5.5/24）

所有路由器上都运行 OSPF，并将所有接口都发布进 OSPF，以保证全网单播互通。

R2、R3 和 R5 都以接口 Loopback 10 的地址 10.1.1.1 配置为 RP，从而形成 Anycast RP，然后还需要在 RP 之间创建 MSDP 连接，连接情况如下：

R2（2.2.2.2）—— R3（3.3.3.3）

R2（2.2.2.2）—— R5（55.5.5.5）

因为从一个 MSDP peer 收到的 SA 会转发给其它所有 MSDP peer，所以 R3 和 R5 都能互相收到对方的 SA，也就没必要在 R3 和 R5 之间创建 MSDP 连接，但也可以创建。

1. 配置初始网络环境

（1）配置 R1：

```
r1(config)#int s1/0
r1(config-if)#encapsulation frame-relay
r1(config-if)#no frame-relay inverse-arp
r1(config-if)#no arp frame-relay
r1(config-if)#ip add 12.1.1.1 255.255.255.0
r1(config-if)#no shutdown
r1(config-if)#frame-relay map ip 12.1.1.2 102 broadcast
r1(config-if)#ip ospf network point-to-point
r1(config-if)#exit

r1(config)#router ospf 1
r1(config-router)#router-id 1.1.1.1
r1(config-router)#network 0.0.0.0 0.0.0.0 area 0
```

r1(config-router)#exit

说明：为 R1 的 S1/0 配置接口地址，并将所有接口发布进 OSPF。

（2）配置 R2：

r2(config)#int loopback 0

r2(config-if)#ip add 2.2.2.2 255.255.255.0

r2(config-if)#ip ospf network point-to-point

r2(config-if)#exit

r2(config)#

r2(config)#int loopback 10

r2(config-if)#ip address 10.1.1.1 255.255.255.255

r2(config-if)#exit

r2(config)#int s1/0

r2(config-if)#encapsulation frame-relay

r2(config-if)#no frame-relay inverse-arp

r2(config-if)#no arp frame-relay

r2(config-if)#ip add 12.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#frame-relay map ip 12.1.1.1 201 broadcast

r2(config-if)#ip ospf network point-to-point

```
r2(config-if)#exit

r2(config)#

r2(config)#int f0/0

r2(config-if)#ip add 23.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#exit

r2(config)#int f0/1

r2(config-if)#ip add 25.1.1.2 255.255.255.0

r2(config-if)#no shutdown

r2(config-if)#exit

r2(config)#router ospf 1

r2(config-router)#router-id 2.2.2.2

r2(config-router)#network 0.0.0.0 0.0.0.0 area 0

r2(config-router)#exit
```

说明：为 R2 的 Loopback 0，Loopback 10，S1/0，F0/0，F0/1 配置接口地址，并将所有接口发布进 OSPF，OSPF 的 router-id 改为 Loopback 0 的地址，因为默认为 Loopback 10 的地址，但 Loopback 10 的地址在网络中是重复的。

（3）配置 R3：

```
r3(config)#int loopback 0

r3(config-if)#ip add 3.3.3.3 255.255.255.0

r3(config-if)#no shutdown
```

```
r3(config-if)#ip ospf network point-to-point

r3(config-if)#exit

r3(config)#

r3(config)#int loopback 10

r3(config-if)#ip address 10.1.1.1 255.255.255.255

r3(config-if)#exit



r3(config)#int f0/0

r3(config-if)#ip add 23.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#exit

r3(config)#

r3(config)#int f0/1

r3(config-if)#ip add 34.1.1.3 255.255.255.0

r3(config-if)#no shutdown

r3(config-if)#exit

r3(config)#int s1/0

r3(config-if)#encapsulation frame-relay

r3(config-if)#no frame-relay inverse-arp

r3(config-if)#no arp frame-relay

r3(config-if)#ip add 35.1.1.3 255.255.255.0

r3(config-if)#no shutdown
```

r3(config-if)#frame-relay map ip 35.1.1.5 305 broadcast

r3(config-if)#ip ospf network point-to-point

r3(config-if)#exit

r3(config)#router ospf 1

r3(config-router)#router-id 3.3.3.3

r3(config-router)#network 0.0.0.0 0.0.0.0 area 0

r3(config-router)#exit

r3(config)#

说明：为 R3 的 Loopback 0，Loopback 10，S1/0，F0/1，F0/0 配置接口地址，并将所有接口发布进 OSPF，OSPF 的 router-id 改为 Loopback 0 的地址，因为默认为 Loopback 10 的地址，但 Loopback 10 的地址在网络中是重复的。

（4）配置 R4：

r4(config)#int f0/1

r4(config-if)#ip add 34.1.1.4 255.255.255.0

r4(config-if)#no shutdown

r4(config-if)#exit

r4(config)#int s1/0

r4(config-if)#encapsulation frame-relay

r4(config-if)#no frame-relay inverse-arp

r4(config-if)#no arp frame-relay

r4(config-if)#ip address 46.1.1.4 255.255.255.0

r4(config-if)#no shutdown

r4(config-if)#frame-relay map ip 46.1.1.6 406 broadcast

r4(config-if)#ip ospf network point-to-point

r4(config-if)#exit

r4(config)#router ospf 1

r4(config-router)#router-id 4.4.4.4

r4(config-router)#network 0.0.0.0 0.0.0.0 area 0

r4(config-router)#exit

r4(config)#

说明：为 R4 的 F0/1，S1/0 配置接口地址，并将所有接口发布进 OSPF。

（5）配置 R5：

r5(config)#int loopback 0

r5(config-if)#ip address 5.5.5.5 255.255.255.0

r5(config-if)#ip ospf network point-to-point

r5(config-if)#exit

r5(config)#

r5(config)#int loopback 10

r5(config-if)#ip address 10.1.1.1 255.255.255.255

r5(config-if)#exit

```
r5(config)#int f0/0

r5(config-if)#ip address 56.1.1.5 255.255.255.0

r5(config-if)#no shutdown

r5(config-if)#exit

r5(config)#int f0/1

r5(config-if)#ip add 25.1.1.5 255.255.255.0

r5(config-if)#no shutdown

r5(config-if)#exit

r5(config)#

r5(config)#int s1/0

r5(config-if)#encapsulation frame-relay

r5(config-if)#no frame-relay inverse-arp

r5(config-if)#no arp frame-relay

r5(config-if)#ip add 35.1.1.5 255.255.255.0

r5(config-if)#no shutdown

r5(config-if)#ip ospf network point-to-point

r5(config-if)#frame-relay map ip 35.1.1.3 503 broadcast

r5(config-if)#exit
```

说明：为 R5 的 Loopback 0，Loopback 10，S1/0，F0/1，F0/0 配置接口地址，并将所有接口发布进 OSPF，OSPF 的 router-id 改为 Loopback 0 的地址，因为默认为 Loopback 10 的地址，但 Loopback 10 的地址在网络中是重复的。

（6）配置 R6：

```
r6(config)#int f0/0
r6(config-if)#ip address 56.1.1.6 255.255.255.0
r6(config-if)#no shutdown
r6(config-if)#exit
r6(config)#


r6(config)#int s1/0
r6(config-if)#encapsulation frame-relay
r6(config-if)#no frame-relay inverse-arp
r6(config-if)#no arp frame-relay
r6(config-if)#ip add 46.1.1.6 255.255.255.0
r6(config-if)#no shutdown
r6(config-if)#frame-relay map ip 46.1.1.4 604 broadcast
r6(config-if)#ip ospf network point-to-point
r6(config-if)#exit
r6(config)#
r6(config)#router ospf 1
r6(config-router)#router-id 6.6.6.6
r6(config-router)#network 0.0.0.0 0.0.0.0 area 0
r6(config-router)#exit
```

说明：为 R6 的 F0/0，S1/0 配置接口地址，并将所有接口发布进 OSPF。

（7）查看 R1 的路由学习情况：

```
r1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     34.0.0.0/24 is subnetted, 1 subnets
O       34.1.1.0 [110/66] via 12.1.1.2, 00:00:38, Serial1/0
     35.0.0.0/24 is subnetted, 1 subnets
O       35.1.1.0 [110/129] via 12.1.1.2, 00:00:38, Serial1/0
     2.0.0.0/24 is subnetted, 1 subnets
O       2.2.2.0 [110/65] via 12.1.1.2, 00:00:38, Serial1/0
     3.0.0.0/24 is subnetted, 1 subnets
O       3.3.3.0 [110/66] via 12.1.1.2, 00:00:38, Serial1/0
     5.0.0.0/24 is subnetted, 1 subnets
```

O 5.5.5.0 [110/66] via 12.1.1.2, 00:00:38, Serial1/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/65] via 12.1.1.2, 00:00:38, Serial1/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/65] via 12.1.1.2, 00:00:39, Serial1/0

10.0.0.0/32 is subnetted, 1 subnets

O 10.1.1.1 [110/65] via 12.1.1.2, 00:00:39, Serial1/0

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/66] via 12.1.1.2, 00:00:39, Serial1/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/130] via 12.1.1.2, 00:00:39, Serial1/0

r1#

说明：R1 已经学习到全网的每一条路由。

（8）查看 R2 的路由学习情况：

r2#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/2] via 23.1.1.3, 00:00:55, FastEthernet0/0

35.0.0.0/24 is subnetted, 1 subnets

O 35.1.1.0 [110/65] via 25.1.1.5, 00:00:55, FastEthernet0/1

[110/65] via 23.1.1.3, 00:00:55, FastEthernet0/0

2.0.0.0/24 is subnetted, 1 subnets

C 2.2.2.0 is directly connected, Loopback0

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/2] via 23.1.1.3, 00:00:55, FastEthernet0/0

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/2] via 25.1.1.5, 00:00:55, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

C 25.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/32 is subnetted, 1 subnets

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

C 10.1.1.1 is directly connected, Loopback10

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/2] via 25.1.1.5, 00:00:56, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, Serial1/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/66] via 25.1.1.5, 00:00:56, FastEthernet0/1

[110/66] via 23.1.1.3, 00:00:56, FastEthernet0/0

r2#

说明：R2 已经学习到全网的每一条路由。

（9）查看 R3 的路由学习情况：

r3#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/1

35.0.0.0/24 is subnetted, 1 subnets

C 35.1.1.0 is directly connected, Serial1/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/2] via 23.1.1.2, 00:01:12, FastEthernet0/0

3.0.0.0/24 is subnetted, 1 subnets

C 3.3.3.0 is directly connected, Loopback0

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/3] via 23.1.1.2, 00:01:12, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/2] via 23.1.1.2, 00:01:12, FastEthernet0/0

10.0.0.0/32 is subnetted, 1 subnets

C 10.1.1.1 is directly connected, Loopback10

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/3] via 23.1.1.2, 00:01:12, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/65] via 23.1.1.2, 00:01:12, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/65] via 34.1.1.4, 00:01:12, FastEthernet0/1

r3#

说明：R3 已经学习到全网的每一条路由。

（10）查看 R4 的路由学习情况：

r4#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

C 34.1.1.0 is directly connected, FastEthernet0/1

35.0.0.0/24 is subnetted, 1 subnets

O 35.1.1.0 [110/65] via 34.1.1.3, 00:01:27, FastEthernet0/1

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/3] via 34.1.1.3, 00:01:27, FastEthernet0/1

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/2] via 34.1.1.3, 00:01:27, FastEthernet0/1

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/4] via 34.1.1.3, 00:01:27, FastEthernet0/1

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/2] via 34.1.1.3, 00:01:27, FastEthernet0/1

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/3] via 34.1.1.3, 00:01:28, FastEthernet0/1

10.0.0.0/32 is subnetted, 1 subnets

O 10.1.1.1 [110/2] via 34.1.1.3, 00:01:28, FastEthernet0/1

56.0.0.0/24 is subnetted, 1 subnets

O 56.1.1.0 [110/4] via 34.1.1.3, 00:01:28, FastEthernet0/1

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/66] via 34.1.1.3, 00:01:28, FastEthernet0/1

46.0.0.0/24 is subnetted, 1 subnets

C 46.1.1.0 is directly connected, Serial1/0

r4#

说明：R4 已经学习到全网的每一条路由。

（11）查看 R5 的路由学习情况：

r5#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/3] via 25.1.1.2, 00:00:23, FastEthernet0/1

35.0.0.0/24 is subnetted, 1 subnets

C 35.1.1.0 is directly connected, Serial1/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/2] via 25.1.1.2, 00:00:23, FastEthernet0/1

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/3] via 25.1.1.2, 00:00:23, FastEthernet0/1

5.0.0.0/24 is subnetted, 1 subnets

C 5.5.5.0 is directly connected, Loopback0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/2] via 25.1.1.2, 00:00:23, FastEthernet0/1

25.0.0.0/24 is subnetted, 1 subnets

C 25.1.1.0 is directly connected, FastEthernet0/1

10.0.0.0/32 is subnetted, 1 subnets

C 10.1.1.1 is directly connected, Loopback10

56.0.0.0/24 is subnetted, 1 subnets

C 56.1.1.0 is directly connected, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/65] via 25.1.1.2, 00:00:23, FastEthernet0/1

46.0.0.0/24 is subnetted, 1 subnets

O 46.1.1.0 [110/65] via 56.1.1.6, 00:00:23, FastEthernet0/0

r5#

说明：R5 已经学习到全网的每一条路由。

（12）查看 R6 的路由学习情况：

r6#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

O 34.1.1.0 [110/4] via 56.1.1.5, 00:00:39, FastEthernet0/0

35.0.0.0/24 is subnetted, 1 subnets

O 35.1.1.0 [110/65] via 56.1.1.5, 00:00:39, FastEthernet0/0

2.0.0.0/24 is subnetted, 1 subnets

O 2.2.2.0 [110/3] via 56.1.1.5, 00:00:39, FastEthernet0/0

3.0.0.0/24 is subnetted, 1 subnets

O 3.3.3.0 [110/4] via 56.1.1.5, 00:00:39, FastEthernet0/0

5.0.0.0/24 is subnetted, 1 subnets

O 5.5.5.0 [110/2] via 56.1.1.5, 00:00:39, FastEthernet0/0

23.0.0.0/24 is subnetted, 1 subnets

O 23.1.1.0 [110/3] via 56.1.1.5, 00:00:39, FastEthernet0/0

25.0.0.0/24 is subnetted, 1 subnets

O 25.1.1.0 [110/2] via 56.1.1.5, 00:00:39, FastEthernet0/0

10.0.0.0/32 is subnetted, 1 subnets

O 10.1.1.1 [110/2] via 56.1.1.5, 00:00:39, FastEthernet0/0

56.0.0.0/24 is subnetted, 1 subnets

C 56.1.1.0 is directly connected, FastEthernet0/0

12.0.0.0/24 is subnetted, 1 subnets

O 12.1.1.0 [110/66] via 56.1.1.5, 00:00:39, FastEthernet0/0

46.0.0.0/24 is subnetted, 1 subnets

C 46.1.1.0 is directly connected, Serial1/0

r6#

说明：R6 已经学习到全网的每一条路由。

（13）在 R1 上测试到其它网段的连通性：

r1#ping 2.2.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 48/98/132 ms

r1#

r1#ping 3.3.3.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 60/120/232 ms

r1#

r1#ping 5.5.5.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 84/166/244 ms

r1#

r1#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/108/200 ms

r1#

r1#

r1#ping 34.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 34.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 64/170/220 ms

r1#

r1#

r1#

r1#ping 56.1.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 56.1.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 160/195/280 ms

r1#

r1#ping 46.1.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 46.1.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 84/156/268 ms

r1#

说明：R1 与 2.2.2.0/24、3.3.3.0/24、5.5.5.0/24、10.1.1.1/32、34.1.1.0/24、56.1.1.0/24、46.1.1.0/24 通信正常，说明已经全网单播互通。

（14）在 R4 上测试到 RP 地址 10.1.1.1 的连通性和路径走向：

r4#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 24/45/76 ms

r4#

r4#

r4#traceroute 10.1.1.1

Type escape sequence to abort.

Tracing the route to 10.1.1.1

1 34.1.1.3 96 msec * 112 msec

r4#

说明：R4 到 10.1.1.1/32 通信正常，并且是选择的 R3，路径和预期相同。

（15）在 R6 上测试到 RP 地址 10.1.1.1 的连通性和路径走向：

r6#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 60/103/160 ms

r6#

r6#traceroute 10.1.1.1

Type escape sequence to abort.

Tracing the route to 10.1.1.1

1 56.1.1.5 136 msec * 116 msec

r6#

说明：R6 到 10.1.1.1/32 通信正常，并且是选择的 R5，路径和预期相同。

2. 配置 PIM-SM

（1）在 R1 上配置 PIM-SM：

r1(config)#ip multicast-routing

r1(config)#int s1/0

r1(config-if)#ip pim sparse-mode

r1(config-if)#ip igmp join-group 224.1.1.1

r1(config-if)#exit

说明：在 R1 上开启组播功能，并且在接口 S1/0 下开启 PIM-SM 模式，接口 S1/0 加入组 224.1.1.1，成为组成员。

（2）在 R2 上配置 PIM-SM：

r2(config)#ip multicast-routing

r2(config)#int s1/0

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#int f0/0

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#

r2(config)#int f0/1

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#int loopback 10

r2(config-if)#ip pim sparse-mode

r2(config-if)#exit

r2(config)#

说明：在 R2 上开启组播功能，并且在接口 Loopback 10，F0/0，F0/1，S1/0 下开启 PIM-SM 模式，因为 Loopback 0 只用来建 MSDP 邻居，并不运行组播，所以没有必要运行 PIM。

（3）在 R3 上配置 PIM-SM：

r3(config)#int f0/0

r3(config-if)#ip pim sparse-mode

```
r3(config-if)#exit

r3(config)#

r3(config)#int f0/1

r3(config-if)#ip pim sparse-mode

r3(config-if)#exit

r3(config)#int s1/0

r3(config-if)#ip pim sparse-mode

r3(config-if)#exit

r3(config)#int loopback 10

r3(config-if)#ip pim sparse-mode

r3(config-if)#exit

r3(config)#
```

说明：在 R3 上开启组播功能，并且在接口 Loopback 10，F0/0，F0/1，S1/0 下开启 PIM-SM 模式，因为 Loopback 0 只用来建 MSDP 邻居，并不运行组播，所以没有必要运行 PIM。

（4）在 R4 上配置 PIM-SM：

```
r4(config)#ip multicast-routing

r4(config)#int f0/1

r4(config-if)#ip pim sparse-mode

r4(config-if)#exit

r4(config)#int s1/0
```

r4(config-if)#ip pim sparse-mode

r4(config-if)#exit

说明：在 R4 上开启组播功能，并且在接口 F0/1，S1/0 下开启 PIM-SM 模式。

（5）在 R5 上配置 PIM-SM：

r5(config)#ip multicast-routing

r5(config)#int f0/1

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

r5(config)#int f0/0

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

r5(config)#

r5(config)#int s1/0

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

r5(config)#int loopback 10

r5(config-if)#ip pim sparse-mode

r5(config-if)#exit

说明：在 R5 上开启组播功能，并且在接口 Loopback 10，F0/0，F0/1，S1/0 下开启 PIM-SM 模式，因为 Loopback 0 只用来建 MSDP 邻居，并不运行组播，所以没有必要运行 PIM。

（6）在 R6 上配置 PIM-SM：

r6(config)#ip multicast-routing

r6(config)#int f0/0

r6(config-if)#ip pim sparse-mode

r6(config-if)#exit

r6(config)#int s1/0

r6(config-if)#ip pim sparse-mode

r6(config-if)#exit

说明：在 R6 上开启组播功能，并且在接口 F0/1，S1/0 下开启 PIM-SM 模式。

（7）在 R2 上查看 PIM 邻居情况：

r2#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
23.1.1.3	FastEthernet0/0	00:03:05/00:01:38	v2	1 / DR S
25.1.1.5	FastEthernet0/1	00:01:22/00:01:21	v2	1 / DR S
12.1.1.1	Serial1/0	00:03:36/00:01:37	v2	1 / S

r2#

说明：R2 与 R1，R3 以及 R5 建立 PIM 邻居关系，一切正常。

（8）在 R3 上查看 PIM 邻居情况：

r3#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode
23.1.1.2	FastEthernet0/0	00:02:48/00:01:25	v2	1 / S
34.1.1.4	FastEthernet0/1	00:01:54/00:01:19	v2	1 / DR S
35.1.1.5	Serial1/0	00:01:32/00:01:40	v2	1 / DR S

r3#

说明：R3 与 R2，R4 以及 R5 建立 PIM 邻居关系，一切正常。

（9）在 R5 上查看 PIM 邻居情况：

r5#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Address	Interface	Uptime/Expires	Ver	DR Prio/Mode

56.1.1.6 FastEthernet0/0 00:01:28/00:01:15 v2 1 / DR S

25.1.1.2 FastEthernet0/1 00:02:00/00:01:42 v2 1 / S

35.1.1.3 Serial1/0 00:01:57/00:01:16 v2 1 / S

r5#

说明：R5 与 R2，R3 以及 R6 建立 PIM 邻居关系，一切正常。

（10）在 R4 上查看 PIM 邻居情况：

r4#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

S - State Refresh Capable

Neighbor Interface Uptime/Expires Ver DR

Address Prio/Mode

34.1.1.3 FastEthernet0/1 00:01:37/00:01:36 v2 1 / S

46.1.1.6 Serial1/0 00:01:14/00:01:30 v2 1 / DR S

r4#

说明：R4 与 R3 和 R6 建立 PIM 邻居关系，一切正常。

（11）分别将 R2，R3 以及 R5 都以 Loopback 10 配置为 RP，从而形成 Anycast RP：

r2(config)#access-list 24 permit 224.1.1.1

r2(config)#ip pim send-rp-announce loopback 10 scope 16 group-list 24

r2(config)#ip pim send-rp-discovery loopback 10 scope 16

r3(config)#access-list 24 permit 224.1.1.1

r3(config)#ip pim send-rp-announce loopback 10 scope 16 group-list 24

r3(config)#ip pim send-rp-discovery loopback 10 scope 16

r5(config)#access-list 24 permit 224.1.1.1

r5(config)#ip pim send-rp-announce loopback 10 scope 16 group-list 24

r5(config)#ip pim send-rp-discovery loopback 10 scope 16

说明：R2，R3 以及 R5 都以 Loopback 10 配置为 RP，形成 Anycast RP。

（12）查看全网路由器的 RP 学习情况：

r1#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 10.1.1.1 (?), v2v1

Info source: 10.1.1.1 (?), elected via Auto-RP

Uptime: 00:01:22, expires: 00:02:35

r1#

r2#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback10)

Group(s) 224.1.1.1/32

RP 10.1.1.1 (?), v2v1

Info source: 10.1.1.1 (?), elected via Auto-RP

Uptime: 00:01:44, expires: 00:02:13

r2#

r3#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback10)

Group(s) 224.1.1.1/32

RP 10.1.1.1 (?), v2v1

Info source: 10.1.1.1 (?), elected via Auto-RP

Uptime: 00:01:53, expires: 00:02:04

r3#

r3#

r4#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 10.1.1.1 (?), v2v1

Info source: 10.1.1.1 (?), elected via Auto-RP

Uptime: 00:03:41, expires: 00:02:13

r4#

r5#sh ip pim rp mapping

PIM Group-to-RP Mappings

This system is an RP (Auto-RP)

This system is an RP-mapping agent (Loopback10)

Group(s) 224.1.1.1/32

RP 10.1.1.1 (?), v2v1

Info source: 10.1.1.1 (?), elected via Auto-RP

Uptime: 00:03:51, expires: 00:02:09

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

r5#

r6#sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32

RP 10.1.1.1 (?), v2v1

Info source: 10.1.1.1 (?), elected via Auto-RP

Uptime: 00:04:41, expires: 00:02:12

r6#

说明：全网的路由器学习到的 RP 地址都为 10.1.1.1，然而 R2，R3 以及 R5 就是以 10.1.1.1 地址为 RP，所以每台路由器都走拓朴上离自己最近的 RP，如 R1 选择 R2，R4 选择 R3，而 R6 选择 R5，这就是 Anycast RP。

（13）测试组播源 R4 和 R6 到组 224.1.1.1 的通信情况：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r4#

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r6#

说明：因为 R1 选择 R2 作 RP，R4 选择 R3 作 RP，而 R6 选择 R5 作 RP，所以每一台 RP 上都没有完整的组播源和组成员信息，最终全网的组播是无法通信的，只要通过 MSDP 让这些 RP 之间共享自己已知的组播源或组成员信息，这样才能保证每一台 RP 都有完整的组播源和组成员信息，才能保证全网组播互通。

（14）查看 R3 和 R2 的组播树情况：

r3#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:01:17/stopped, RP 10.1.1.1, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(34.1.1.4, 224.1.1.1), 00:01:17/00:02:01, flags: PT

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list: Null

r3#

r2#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:13:45/00:02:39, RP 10.1.1.1, flags: SJC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:13:45/00:02:39

r2#

说明：正是因为 R3 没有组成员信息，R2 没有组播源信息，所以全网的组播树无法形成，导致全网组播无法通信。

3. 配置 MSDP

（1）在 R2 和 R3 之间建立 MSDP 连接：

r2(config)#ip msdp peer 3.3.3.3 connect-source loopback 0

r2(config)#ip msdp originator-id loopback 0

r3(config)#ip msdp peer 2.2.2.2 connect-source loopback 0

r3(config)#ip msdp originator-id loopback 0

说明：在 R2 和 R3 之间以 loopback 0 的地址为源建立 MSDP 连接，并且将自己发出去的 SA 数据包中的 RP 地址改成 loopback 0 的地址，因为默认都是 loopback 10 的地址，这样会造成全网都一样，并且会导致 RPF 检测失败。

（2）查看 MSDP 的连接情况：

r2#sh ip msdp summary

MSDP Peer Status Summary

Peer Address AS State Uptime/ Reset SA Peer Name

Downtime Count Count

3.3.3.3 ? Up 00:00:27 0 0 ?

r2#

说明：R2 与 R3 之间的 MSDP 连接已经正常建立。

（3）再次测试组播源 R4 到组 224.1.1.1 的通信情况，并在 MSDP 路由器 R2 上通过 Debug 观察 MSDP 信息：

在 MSDP 路由器 R2 上开启 Debug:

r2#debug ip msdp peer

MSDP Peer debugging is on

r2#

r2#debug ip msdp detail

MSDP Detail debugging is on

r2#

在组播源路由器 R4 上发起组播流量：

r4#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 276 ms

Reply to request 1 from 12.1.1.1, 208 ms

Reply to request 2 from 12.1.1.1, 220 ms

Reply to request 3 from 12.1.1.1, 204 ms

Reply to request 4 from 12.1.1.1, 240 ms

Reply to request 5 from 12.1.1.1, 236 ms

Reply to request 6 from 12.1.1.1, 200 ms

Reply to request 7 from 12.1.1.1, 264 ms

Reply to request 8 from 12.1.1.1, 236 ms

Reply to request 9 from 12.1.1.1, 296 ms

r4#

说明：和预期一样，在 RP 之间建立 MSDP 共享 SA 信息之后，组播通信正常.

观察 R2 上的 Debug 信息：

r2#

*Mar 1 00:50:00.291: MSDP(0): Received 120-byte TCP segment from 3.3.3.3

*Mar 1 00:50:00.291: MSDP(0): Append 120 bytes to 0-byte msg 6 from 3.3.3.3, qs 1

*Mar 1 00:50:00.291: MSDP(0): 3.3.3.3: Received 120-byte msg 6 from peer

*Mar 1 00:50:00.295: MSDP(0): 3.3.3.3: SA TLV, len: 120, ec: 1, RP: 3.3.3.3, with data

*Mar 1 00:50:00.295: MSDP(0): 3.3.3.3: Peer RPF check passed for single peer

*Mar 1 00:50:00.299: MSDP(0): WAVL Insert SA Source 34.1.1.4 Group 224.1.1.1 RP 3.3.3.3 Successful

*Mar 1 00:50:00.299: MSDP(0): Forward decapsulated SA data for (34.1.1.4, 224.1.1.1) on Serial1/0

r2#

说明：因为目前 R2 只有单个 MSDP peer，所以对于 R3 发来的 SA 也就没必要进行 RPF 检测，最后 R2 顺利将关于组播源 34.1.1.4（R4）的 SA 缓存起来。

（4）在 R2 上查看 SA 信息的接收情况：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 1 entries

(34.1.1.4, 224.1.1.1), RP 3.3.3.3, AS ?,00:01:52/00:05:41, Peer 3.3.3.3

Learned from peer 3.3.3.3, RPF peer 3.3.3.3,

SAs received: 3, Encapsulated data received: 1

r2#

说明：R3 发来的 SA 信息已被成功缓存并使用。

（5）查看 R4 的组播树情况：

r4#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:00:18/stopped, RP 10.1.1.1, flags: SPF

Incoming interface: FastEthernet0/1, RPF nbr 34.1.1.3

Outgoing interface list: Null

(34.1.1.4, 224.1.1.1), 00:00:14/00:03:22, flags: T

Incoming interface: Serial1/0, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:00:14/00:03:15

r4#

说明：从 R4 的组播树可以看出，流量是从接口 F0/1 发出去的，即是走 R3 的，与预期一样。

（6）查看 R3 的组播树情况：

r3#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:00:23/stopped, RP 10.1.1.1, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(34.1.1.4, 224.1.1.1), 00:00:23/00:03:24, flags: TA

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:00:19/00:03:10

r3#

说明：从 R3 的组播树可以看出，R4 的组播流量路径正常，与预期一样。

（7）查看 R2 的组播树情况：

r2#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:00:51/stopped, RP 10.1.1.1, flags: SJC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:00:51/00:02:08

(34.1.1.4, 224.1.1.1), 00:00:51/00:02:25, flags: MT

Incoming interface: FastEthernet0/0, RPF nbr 23.1.1.3

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:00:51/00:02:08

r2#

说明：从 R2 的组播树可以看出，R4 的组播流量路径正常，与预期一样。

（8）测试组播源 R6 到组 224.1.1.1 的通信情况：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

.....

r6#

说明：因为 R4 是选择 R3 作为 RP 的，而 R6 是选择 R5 作为 RP 的，R2 与 R3 之间的 MSDP 正常，但 R2 与 R5 却没有 MSDP，所以 R6 的组播不通是正常的，要解决该问题，应该先在 R2 与 R5 之间创建 MSDP 连接。

（9）在 R2 和 R5 之间建立 MSDP 连接：

r2(config)#ip msdp peer 5.5.5.5 connect-source loopback 0

r5(config)#ip msdp peer 2.2.2.2 connect-source loopback 0

r5(config)#ip msdp originator-id loopback 0

说明：在 R2 和 R5 之间以 loopback 0 的地址为源建立 MSDP 连接，R5 也将自己发出去的 SA 数据包中的 RP 地址改成 loopback 0 的地址。

（10）在 R2 上再次查看 MSDP 的连接情况：

r2#sh ip msdp summary

MSDP Peer Status Summary

Peer Address	AS	State	Uptime/ Downtime	Reset Count	SA Count	Peer Name
3.3.3.3	?	Up	00:08:05	0	1	?
5.5.5.5	?	Up	00:00:05	0	1	?

r2#

说明：R2 与 R5 之间的 MSDP 也已经正常建立。

（11）再次测试组播源 R6 到组 224.1.1.1 的通信情况，并观察 R2 上的 Debug 信

息：

r6#ping 224.1.1.1 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 212 ms

Reply to request 1 from 12.1.1.1, 256 ms

Reply to request 2 from 12.1.1.1, 212 ms

Reply to request 3 from 12.1.1.1, 304 ms

Reply to request 4 from 12.1.1.1, 212 ms

Reply to request 5 from 12.1.1.1, 152 ms

Reply to request 6 from 12.1.1.1, 192 ms

Reply to request 7 from 12.1.1.1, 184 ms

Reply to request 8 from 12.1.1.1, 160 ms

Reply to request 9 from 12.1.1.1, 212 ms

r6#

说明：在 R2 与 R5 之间建立 MSDP 连接之后，便能发送组播源信息，所以此时 R6 的组播通信正常。

观察 R2 上的 Debug 信息：

r2#

r2#Received 20-byte msg 11 from peer

*Mar 1 00:53:52.855: MSDP(0): 5.5.5.5: SA TLV, len: 20, ec: 1, RP: 5.5.5.5

*Mar 1 00:53:52.859: MSDP(0): 5.5.5.5: Peer RPF check passed for 5.5.5.5, peer is RP

*Mar 1 00:53:52.859: MSDP(0): WAVL Insert SA Source 56.1.1.6 Group 224.1.1.1 RP 5.5.5.5 Successful

*Mar 1 00:53:52.863: MSDP(0): 3.3.3.3: Add SA entry (56.1.1.6, 224.1.1.1) RP: 5.5.5.5

*Mar 1 00:53:52.867: MSDP(0): 3.3.3.3: Send 20-byte SA message to 3.3.3.3, 1 entries

r2#

说明：虽然 R2 上有两个 MSDP 连接，但是因为 R3 和 R5 建立 MSDP 的地址和 SA 数据包中的 RP 地址都是 Loopback 0，所以能够正常通过 RPF 检测，这就是为什么之前要将 SA 数据包中 RP 地址更改为 Loopback 0 的地址的原因。

（12）在 R2 上再次查看 SA 信息：

r2#sh ip msdp sa-cache

MSDP Source-Active Cache - 2 entries

(34.1.1.4, 224.1.1.1), RP 3.3.3.3, AS ?,00:07:38/00:00:51, Peer 3.3.3.3

Learned from peer 3.3.3.3, RPF peer 3.3.3.3,

SAs received: 4, Encapsulated data received: 1

(56.1.1.6, 224.1.1.1), RP 5.5.5.5, AS ?,00:03:45/00:03:14, Peer 5.5.5.5

Learned from peer 5.5.5.5, RPF peer 5.5.5.5,

SAs received: 2, Encapsulated data received: 0

r2#

说明：R3 和 R5 发来的 SA 信息已被 R2 成功缓存并使用，所以现在全网的组播通信均正常。

（13）查看 R6 的组播树情况：

r6#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:00:05/stopped, RP 10.1.1.1, flags: SPF

Incoming interface: FastEthernet0/0, RPF nbr 56.1.1.5

Outgoing interface list: Null

(56.1.1.6, 224.1.1.1), 00:00:05/00:02:59, flags: T

Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:00:14/00:02:16

r6#

说明：从 R6 的组播树可以看出，流量是从接口 F0/0 发出去的，即是走 R5 的，与预期一样。

（14）查看 R5 的组播树情况：

r5#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:04:32/stopped, RP 10.1.1.1, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(46.1.1.4, 224.1.1.1), 00:01:33/00:01:56, flags:

Incoming interface: FastEthernet0/0, RPF nbr 56.1.1.6

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:01:33/00:02:55

(56.1.1.6, 224.1.1.1), 00:04:32/00:02:50, flags: TA

Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:04:32/00:03:07

r5#

说明：从 R5 的组播树可以看出，R6 的组播流量路径正常，与预期一样，从目前的情况来看，Anycast RP 确实为组播网络带来了负载均衡的效果。

4. 测试 Anycast RP 冗余性

说明：因为 R2，R3 以及 R5 的 RP 地址都配置为 10.1.1.1，并且该地址在 OSPF

里面全网可达，如果当 R4 使用的 RP 路由器 R3 出现故障，那么 R4 应该能够根据 OSPF 路由的变化从而切换到 R5，实际冗余性，对于 R6 也一样，当 R6 使用的 RP 路由器 R5 出现故障，那么 R6 应该能够根据 OSPF 路由的变化从而切换到 R3，下面我们来测试让 R6 使用的 RP 路由器 R5 出现故障，看看 R6 的组播通信情况。

（1）调整 R4 与 R6 之间的 PIM 关系：

```
r4(config)#int s1/0

r4(config-if)#ip pim dr-priority 4

r4(config-if)#exit
```

```
r6#sh ip pim neighbor

PIM Neighbor Table

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,

      S - State Refresh Capable

Neighbor          Interface              Uptime/Expires     Ver    DR

Address                                                          Prio/Mode

56.1.1.5          FastEthernet0/0        00:01:38/00:01:35 v2     1 / S

46.1.1.4          Serial1/0              00:07:17/00:01:21 v2     4 / DR S

r6#
```

说明：提高 R4 的 S1/0 接口的 DR 优先级后，R4 成为了 DR。

（2）关闭 RP 路由器 R5，并观察 R4 的组播流量情况：

r4#ping 224.1.1.1 repeat 1000

Type escape sequence to abort.

Sending 1000, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 144 ms

Reply to request 0 from 12.1.1.1, 444 ms

Reply to request 1 from 12.1.1.1, 312 ms

Reply to request 2 from 12.1.1.1, 328 ms

Reply to request 3 from 12.1.1.1, 308 ms

Reply to request 4 from 12.1.1.1, 272 ms

Reply to request 5 from 12.1.1.1, 208 ms

Reply to request 6 from 12.1.1.1, 132 ms

Reply to request 7 from 12.1.1.1, 208 ms

Reply to request 8 from 12.1.1.1, 216 ms

Reply to request 9 from 12.1.1.1, 200 ms

Reply to request 10 from 12.1.1.1, 216 ms

Reply to request 11 from 12.1.1.1, 216 ms

Reply to request 12 from 12.1.1.1, 200 ms

Reply to request 13 from 12.1.1.1, 172 ms

Reply to request 14 from 12.1.1.1, 188 ms

Reply to request 15 from 12.1.1.1, 188 ms

Reply to request 16 from 12.1.1.1, 188 ms

Reply to request 17 from 12.1.1.1, 156 ms

Reply to request 18 from 12.1.1.1, 156 ms

Reply to request 19 from 12.1.1.1, 208 ms

Reply to request 20 from 12.1.1.1, 280 ms

Reply to request 21 from 12.1.1.1, 200 ms

Reply to request 22 from 12.1.1.1, 216 ms

Reply to request 23 from 12.1.1.1, 280 ms

Reply to request 24 from 12.1.1.1, 216 ms

Reply to request 25 from 12.1.1.1, 216 ms

Reply to request 26 from 12.1.1.1, 216 ms

Reply to request 27 from 12.1.1.1, 296 ms

Reply to request 28 from 12.1.1.1, 216 ms

Reply to request 29 from 12.1.1.1, 172 ms

Reply to request 30 from 12.1.1.1, 172 ms

Reply to request 31 from 12.1.1.1, 192 ms

Reply to request 32 from 12.1.1.1, 196 ms

r4#

说明：在关闭 RP 路由器 R5 期间，R4 的组播通信不受影响，因为 R4 是靠 R3 来通信的，与 R5 无关。

（3）观察关闭 RP 路由器 R5 后，R6 的组播流量情况：

r6#ping 224.1.1.1 repeat 1000

Type escape sequence to abort.

Sending 1000, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 12.1.1.1, 440 ms

Reply to request 0 from 12.1.1.1, 692 ms

Reply to request 1 from 12.1.1.1, 200 ms

Reply to request 2 from 12.1.1.1, 216 ms

Reply to request 3 from 12.1.1.1, 192 ms.....

*Mar 1 00:10:42.131: %OSPF-5-ADJCHG: Process 1, Nbr 5.5.5.5 on FastEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer expired...

Reply to request 23 from 12.1.1.1, 460 ms

Reply to request 24 from 12.1.1.1, 264 ms

Reply to request 25 from 12.1.1.1, 404 ms

Reply to request 26 from 12.1.1.1, 360 ms

Reply to request 27 from 12.1.1.1, 324 ms

Reply to request 28 from 12.1.1.1, 372 ms

Reply to request 29 from 12.1.1.1, 388 ms

Reply to request 30 from 12.1.1.1, 328 ms

r6#

说明：因为 R6 是优先选择 RP 路由器 R5 来通信的，所以当 R5 出现故障后，R6

的组播被中断，但最终又重新选择另一台可用的 RP 路由器 R3，从而又恢复了组播通信，体现了 Anycast RP 的冗余性。

（4）查看 R4 的组播树情况：

r4#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:01:43/stopped, RP 10.1.1.1, flags: SPF

Incoming interface: FastEthernet0/1, RPF nbr 34.1.1.3

Outgoing interface list: Null

(34.1.1.4, 224.1.1.1), 00:01:43/00:02:21, flags: PFT

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list: Null

(46.1.1.6, 224.1.1.1), 00:01:35/00:02:59, flags: FT

Incoming interface: Serial1/0, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:00:50/00:02:54

(56.1.1.6, 224.1.1.1), 00:00:50/00:02:39, flags:

Incoming interface: Serial1/0, RPF nbr 46.1.1.6

Outgoing interface list:

FastEthernet0/1, Forward/Sparse, 00:00:50/00:02:39

r4#

说明：在 R6 优先选择 RP 路由器 R5 出现故障后，重新选择了从备用路径 R4 走。

（5）查看 R3 的组播树情况：

r3#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:02:55/stopped, RP 10.1.1.1, flags: SP

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list: Null

(34.1.1.4, 224.1.1.1), 00:02:55/00:02:21, flags: TA

Incoming interface: FastEthernet0/1, RPF nbr 0.0.0.0

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:02:55/00:02:32

(46.1.1.6, 224.1.1.1), 00:02:08/00:02:31, flags: TA

Incoming interface: FastEthernet0/1, RPF nbr 34.1.1.4

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:01:24/00:03:20

(56.1.1.6, 224.1.1.1), 00:01:24/00:02:05, flags:

Incoming interface: FastEthernet0/1, RPF nbr 34.1.1.4

Outgoing interface list:

FastEthernet0/0, Forward/Sparse, 00:01:24/00:03:19

r3#

说明：R3 上的组播树也体现了 R6 的流量是从 R4 再到 R3 被转发给 R2 的。

（6）查看 R2 的组播树情况：

r2#sh ip mroute 224.1.1.1

IP Multicast Routing Table

Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,

L - Local, P - Pruned, R - RP-bit set, F - Register flag,

T - SPT-bit set, J - Join SPT, M - MSDP created entry,

X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,

U - URD, I - Received Source Specific Host Report,

Z - Multicast Tunnel, z - MDT-data group sender,

Y - Joined MDT-data group, y - Sending to MDT-data group

Outgoing interface flags: H - Hardware switched, A - Assert winner

Timers: Uptime/Expires

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:10:20/00:00:28, RP 10.1.1.1, flags: SJC

Incoming interface: Null, RPF nbr 0.0.0.0

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:10:20/00:02:48

(34.1.1.4, 224.1.1.1), 00:08:15/00:01:29, flags: MT

Incoming interface: FastEthernet0/0, RPF nbr 23.1.1.3

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:08:15/00:02:48

(46.1.1.6, 224.1.1.1), 00:02:31/00:01:29, flags: MT

Incoming interface: FastEthernet0/0, RPF nbr 23.1.1.3

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:02:31/00:02:48

(56.1.1.6, 224.1.1.1), 00:02:31/00:00:39, flags: M

Incoming interface: FastEthernet0/0, RPF nbr 23.1.1.3

Outgoing interface list:

Serial1/0, Forward/Sparse, 00:02:31/00:02:48

r2#

说明：R2 上的组播树表明 R4 和 R6 的流量都是从 R3 过来的，和我们预期的路径

相同。

查看当前环境中所有路由器的配置信息请点击“running-config”

IPv6 Multicast

IPv6 Multicast 在此主题中不再详细讨论,具体内容请阅读 IPv6 部分的 IPv6 Multicast,点此进入.

附:

在测试组播时，除了使用 ICMP 之外，Cisco IOS 中还内置了专门测试组播的工具-MRM(Multicast Routing Monitor),由于时间关系，本站不再对 MRM 以及其它组播技术进行详细介绍。对于 Cisco MRM 的应用及配置，请自行参阅思科文档，点此进入

QOS

（提示：由于内容较多，阅读时，建议开启 文档结构图. ）

目录

鸿鹄培训+考试群：301283867

概述

在普通的网络中，当用户将数据发向网络设备后，网络设备都是尽最大努力传输数据，直到超出自己的最大负荷为止。当设备达到最大负荷后，如果还有用户发来的数据，那么这些数据将因为网络设备不能提供服务而被丢弃。这样的提供最大化服务的网络被称为尽力而为服务的网络。在尽力而为服务的网络中，所有的数据都被看成是同等重要的，用户的数据有时无法得到保证，所以在某些时候，必须让网络通过放弃传输相对不重要的数据来保证用户的重要数据和传输。因此，就需要在网络中实施 Quality of Service，即 QOS。实施了 QOS 的网络中，可以为特定数据保证带宽，同时也可以限制宽带，可以避免网络拥塞和管理拥塞，甚至可以为数据设置不同的优先级。

QOS 模型

在网络中实施 QOS 时，有三种模型可供参考，这三种模型并不是 QOS 技术，而是用来指导在各种需求下，如何实施 QOS，分为以下三种模型：

Best-Effort Service 尽力而为服务模型

鸿鹄培训+考试群：301283867

Integrated Service 综合服务模型，简称 Intserv

Differentiated Service 区分服务模型，简称 Diffserv

Best-Effort Service（尽力而为服务模型）

在尽力而为服务模型中，所有网络设备全部都是尽自己最大努力传输数据，所有数据尽管传，不需要得到许可，有多少传多少，任何数据都不能得到保证，延迟也无法预计，所以尽力而为服务模型，其实并没有实施任何 QOS，默认的网络都工作在这种模型下。

Integrated Service（综合服务模型）

在实施了综合服务模型 QOS 的网络中，应用程序在发送数据之前，必须先向网络申请带宽，例如一个视频程序在正常通信下需要 100K 的带宽，那么视频程序在连接之前，必须向网络申请自己需要 100K 的带宽，当网络同意后，视频便可连接，并且将保证能够得到 100K 的带宽，而不会有任何延迟。但是如果某些程序在连接之前没有向网络申请带宽，那么它的流量只能得到尽力而为的服务。由此可见，当某些程序流量需要绝对保证带宽时，可以在综合服务模型的网络中通过申请带宽来保证自己的流量，在申请带宽时，所用到的协议为 Resource Reservation Protocol (RSVP)。在综合服务模型中，重要的数据可以通过申请带宽而得到保证，但是在传送之前必须申请，也需要耗费额外一些时间，在现有的网络中，综合服务模型的 QOS 通常并不被采用。

Differentiated Service（区分服务模型）

在实施了区分服务模型 QOS 的网络中，网络将根据不同数据提供不同服务，因此，所有数据都被分成不同的类别，或者设置为不同的优先级，在网络发生拥塞时，网络总是先保证传输高优先级的数据，从而放弃传输低优先级的数据，但是在网络没有拥塞时，所有数据全部照常传输。在实施区分服务模型的 QOS，就必须先将数据分成不同的类别，或设置成不同的优先级。现在的网络中，实施 QOS 时通常采用区分服务模型。

在网络中，数据从源到目的地，所有的网络设备，包括路由器、交换机、防火墙等，每一台单一的设备对数据包做出的区分服务 QOS 行为称为 per-hop behavior.（PHB 每跳行为），如果数据包从源到目的路径中所有设备都为某类数据执行相同的区分服务行为，即都执行相同的 QOS 策略，那么这样的 QOS 就被称为 end-to-end QoS（端到端 QOS）。

每一台单一的设备对数据包做出的区分服务 QOS 行为称为 per-hop behavior.（PHB 每跳行为），如果数据包从源到目的路径中所有设备都为某类数据执行相同的区分服务行为，那么就被称为 end-to-end QoS(端到端 QOS)

注：本篇将着重介绍区分服务模型 QOS，之后再介绍综合服务模型 QOS。

QOS 组件

在实施区分服务模型 QOS 时，需要考虑四个 QOS 组件，这些组件相互组合，可以设计出完整的 QOS 策略，而每个组件中，都会有相应的 QOS 技术提供支持，以下是 QOS 四个组件：

分类和标记（classification and marking）

管制和整形（Policing and Shaping）

拥塞管理（Congestion management）

拥塞避免（Congestion avoidance）

分类和标记

要提供区分服务的 QOS，就必须先将数据分为不同的类别，或者将数据设置为不同的优先级。将数据分为不同的类别，称为分类(classification)，分类并不修改原来的数据包。将数据设置为不同的优先级称为标记(marking)，而标记会修改原来的数据包。分类和标记是实施 QOS 的前提，也是基础。

管制和整形

在实施 QOS 策略时，可以将用户的数据限制在特定的带宽，当用户的流量超过额定带宽时，超过的带宽将不能被传输，只能采取其它方式来处理，如果处理方式为丢弃超出带宽，那么这种行为称为管制(Policing)，如果是将超出的带宽缓存在内存中，等到下一秒再传递，这种行为称为整形（Shaping）。

拥塞管理

当网络发生拥塞后，数据还是要被传递的，正因为接收到的数据远多于自身的传输能力，所以数据被传输时就出现了先后顺序，而依照什么样的方式来传数据，就需要队列的指导，QOS 中的队列定义了数据包被传输的先后顺序。

拥塞避免

当网络发生拥塞后，超出的流量将采取其它方式处理，如果处理方式为管制，那么数据包就会被丢弃，通常情况下，网络设备默认丢弃后到的数据包而传输先到的数据包，这样的丢弃方式称为尾丢弃，但也可以让网络设备在发生拥塞时，先丢低优先级的数据包而传输高优先级的数据包。

并不是所有的 QOS 技术都适合所有网络，边缘路由器和核心路由器操作是不一样的。

比如语音数据，边缘和核心要同时考虑。而通常情况是：

边缘路由器执行：数据包分类和标记

核心路由器执行：拥塞管理，拥塞避免

后面将详细介绍 QOS 四个组件中的各个工具。

MQC（Modular QoS Command-Line）

MQC 就是模块化 QOS 命令行，是配置 QOS 处理数据的一种方式。MQC 可以配置对特定的数据采取特定的动作，步骤为三步：

定义流量

设置策略

应用策略

定义流量

在使用 MQC 时，只能在命令行下使用，定义流量通过创建 class-map 来匹配特定的数据。

例：

匹配源主机 10.1.1.1 发出的数据

1 创建 ACL 匹配主机 10.1.1.1 发出的数据

Router(config)#access-list 1 permit 10.1.1.1 0.0.0.0

2 创建 class-map，调用 ACL 的数据

Router(config)# class-map match-all ccie

Router(config-cmap)#match access-group 1

说明:class-map 中可以匹配多个数据，当存在多条匹配时，是不是所有条件都需要满足，则靠创建 class-map 时的关键字来判断，关键字 match-all 表示所有条件都要同时满足，默认为 match-all，如果关键字为 match-any，则任一条满足即可。

注： 名为 class-default 的 class-map，表示匹配所有数据。

设置策略

当匹配到特定的数据之后，就需要对其设置相应的策略，通过创建 policy-map，然后调用 class-map 匹配到的数据，从而设置相应的策略或动作。

例：

对名为 ccie 的 class-map 所匹配到的数据全部丢弃

Router(config)#policy-map cisco

Router(config-pmap)#class ccie

Router(config-pmap-c)#drop

说明:一个 policy-map 里面可以调用多个 class-map，如果调用 class-default，那么表示之前没有匹配到的流量，全部都会被 class-default 所匹配。

应用策略

当策略设置完毕之后，就需要应用到接口上。

例：

将 policy-map 应用到接口 F0/0 出方向上

Router(config)#interface f0/0

Router(config-if)#service-policy output cisco

多动作 MQC

在配置 MQC 时，可以对匹配到的流量做出多个处理动作

例：

将 class-map ccie 中超出额定带宽的流量设置 IP 优先级为 4DE 为 1，然后再传输

Router(config)#policy-map cisco

Router(config-pmap)#class ccie

Router(config-pmap-c)#police cir percent 10 bc 100 ms

Router(config-pmap-c-police)#conform-action transmit

Router(config-pmap-c-police)#exceed-action set-prec-transmit 4

Router(config-pmap-c-police)#exceed-action set-frde-transmit

说明:当 CIR 设置为百分比时，Bc 则为时间，单位 ms。

令牌桶算法（token bucket algorithm）

在实施 QOS 策略时，可以将用户的数据限制在特定的带宽，当用户的流量超过额定带宽时，超过的带宽将采取其它方式来处理。要衡量流量是否超过额定的带宽，网络设备并不是采用单纯的数字加减法来决定的，也就是说，比如带宽为 100K，而用户发来的流量为 110K，网络设备并不是靠 110K 减去 100K 等于 10K，就认为用户超过流量 10K。网络设备

衡量流量是否超过额定带宽，需要使用令牌桶算法来计算。下面详细介绍令牌桶算法机制：

当网络设备衡量流量是否超过额定带宽时，需要查看令牌桶，而令牌桶中会放置一定数量的令牌，一个令牌允许接口发送或接收 1bit 数据（有时是 1 Byty 数据），当接口通过 1bit 数据后，同时也要从桶中移除一个令牌。当桶里没有令牌的时候，任何流量都被视为超过额定带宽,只有当桶中有令牌时，数据才可以通过接口。令牌桶中的令牌不仅仅可以被移除，同样也可以往里添加，所以为了保证接口随时有数据通过，就必须不停地往桶里加令牌，由此可见，往桶里加令牌的速度，就决定了数据通过接口的速度。因此，我们通过控制往令牌桶里加令牌的速度从而控制用户流量的带宽。而设置的这个用户传输数据的速率被称为承诺信息速率（CIR），通常以秒为单位。比如我们设置用户的带宽为 1000 bit 每秒，只要保证每秒钟往桶里添加 1000 个令牌即可。

例：

将 CIR 设置为 8000 bit/s，那么就必须每秒将 8000 个令牌放入桶中，当接口有数据通过时，就从桶中移除相应的令牌，每通过 1 bit，就从桶中移除 1 个令牌。当桶里没有令牌的时候，任何流量都被视为超出额定带宽，而超出的流量就要采取额外动作。

每秒钟往桶里加的令牌就决定了用户流量的速率，这个速率就是 CIR，但是每秒钟需要往桶里加的令牌总数，并不是一次性加完的，一次性加进的令牌数量被称为 Burst size（Bc），如果 Bc 只是 CIR 的一半，那么很明显每秒钟就需要往桶里加两次令牌，每次加的数量总是 Bc 的数量。

还有就是加令牌的时间，Time interval（Tc），Tc 表示多久该往桶里加一次令牌，而这个时间并不能手工设置，因为这个时间可以靠 CIR 和 Bc 的关系计算得到，Bc/ CIR= Tc。

例：

如果 CIR 是 8000,Bc 是 4000，那就是每秒加两次，Tc 就是 4000/8000=0.5，也就是 0.5 秒，即 500 ms。

如果 Bc 设为 2000，那 Tc 就是 2000/8000=0.25，也就是 250 ms。

单速双色

在单速双色的令牌桶算法中，只存在一个令牌桶，并且流量只会出现两种结果，即符合 CIR（conform）和超出 CIR（exceed）。

例：

将 CIR 设置为 8000 bit，每一秒都会往桶里加 8000 个令牌，在一秒钟结束后，没有用完的令牌会被全部清空，由下一秒重新加入。

如

第 1 秒，加入 8000 令牌，用户使用 5000 后，剩余 3000 被清空

第 2 秒，加入 8000 令牌，用户使用 6000 后，剩余 2000 被清空

第 3 秒，加入 8000 令牌，用户使用 8000 后，没有剩余

第 4 秒，加入 8000 令牌，用户使用 7000 后，剩余 1000 被清空

从以上过程可以看出，用户每秒都可以使用 8000 令牌，也就是每秒速度均可达到 8000 bit，而无论上一秒钟是否传过数据，这一秒都可以保持在 8000 bit/s，并且如果每秒流量超过了 8000 后，超过的流量都会采取已经设定的动作。

单速三色

在单速三色的令牌桶算法中，使用两个令牌桶，用户每秒的可用带宽，总是两个桶的令牌之和，第一个桶的令牌机制和单速双色算法没有任何区别，关键在于第二个桶。第二个桶的令牌不能直接加入，只有当一秒钟结束后，第一个桶中存在剩余令牌时，这些剩余令牌就可以从第一个桶中被转移到第二个桶中。但不是第一个桶所有未用令牌都可以放入第二个桶，是有限制的，最大数量被称为 Excess Burst size（Be），由此可见，Be 是不可能超过 CIR 的，因为第一个桶每秒的所有令牌就是 CIR，即使所有令牌全部被移到第二个桶，Be 最多也只能等于 CIR 而不能超过。而 Be 和 Bc 却毫无关系。需要注意的是，在每一秒结束时，如果用户没有将第二个桶的令牌用完，那么第二个桶的令牌也是要全部被清除的，第二个桶中的令牌，都是来自于上一秒第一个桶没用完的令牌。

由于使用了两个桶，所以用户的流量也会出现三种结果：

小于或等于 CIR（也就是符合 CIR） （conform）

大于 CIR 并小于或等于 CIR 与 Be 之和（也就是符合两个桶令牌之和）（exceed）

超过 CIR 与 Be 之和（也就是超过两个桶令牌之和）（violate）

例：

将 CIR 设置为 8000 bit，Be 设置为 2000，每一秒都会往第一个桶里加 8000 个令牌，每一秒结束后，所有第一个桶未使用完的令牌都将放入第二个桶，并且用户每一秒能使用的带宽总是两个桶之和。

如：

	第一个桶令牌数	用户用掉的带宽数	第二个桶令牌数	用户可用带宽总数
第 1 秒	8000	6000	0	8000
第 2 秒	8000	7000	2000	10000
第 3 秒	8000	5000	1000	9000

第 4 秒	8000	9000	2000	10000
第 5 秒	8000	8000	0	8000
第 6 秒	8000	6000	0	8000
第 7 秒	8000	10000	2000	10000

说明:

在第 1 秒时，第一个桶加入 CIR 的数量 8000 个令牌后，第二个桶为空，所以用户可用带宽总数为 8000。用户实际使用了 6000；

在第 2 秒时，第一个桶加入 8000 个令牌后，由于上一秒用户实际使用了 6000，所以第二个桶获得 2000 令牌，此时用户可用带宽为 10000，用户实际用户了 7000；

在第 3 秒时，第一个桶加入 8000 个令牌后，由于上一秒用户实际使用了 7000，所以第二个桶获得 1000 令牌，此时用户可用带宽为 9000，用户实际用户了 5000；

在第 4 秒时，第一个桶加入 8000 个令牌后，由于上一秒用户实际使用了 5000，所以第二个桶获得 2000 令牌，此时用户可用带宽为 10000，用户实际用户了 9000；

在第 5 秒时，第一个桶加入 8000 个令牌后，由于上一秒将第一个桶中令牌用光，所以第二个桶没有获得令牌，此时用户可用带宽为 8000，用户实际用户了 8000；

在第 6 秒时，第一个桶加入 8000 个令牌后，由于上一秒将第一个桶中令牌用光，所以第二个桶没有获得令牌，此时用户可用带宽为 8000，用户实际用户了 6000；

在第 7 秒时，第一个桶加入 8000 个令牌后，由于上一秒用户实际使用了 6000，所以第二个桶获得 2000 令牌，此时用户可用带宽为 10000，用户实际用户了 10000；

从上面可以看出，第一个桶中的令牌数永远都是 CIR 的数量，而第二个令牌桶只能在上一秒第一个桶存在没有用完的令牌的情况下，才能够获得令牌，但获得令牌的最大数量不能超过 Be。用户的流量也可以出现三种结果：

即小于或等于 CIR,即小于 8000，如 6000

大于 CIE 并小于或等于 CIR+Be，如 9000

大于两个桶之后，如 11000

要使用户在某一秒的速度能够达到 CIR+Be，唯一的办法是用户在上一秒钟以低于 CIR 的速度传输。因此，用户不可能每一秒都以 CIR+Be 的速度传输。

双速三色

在单速三色的令牌桶算法中，用户若想要在某一秒以 CIR+Be 的速度传输，只能在上一秒钟以低于 CIR 的速度传输。因此，用户不可能每一秒都以 CIR+Be 的速度传输。而在双速三色的令牌桶算法中，同样使用两个令牌桶，然而这两个桶是相互独立的，并不会将第一个桶未用的令牌放入第二个桶。第一个桶与以往的算法相同，也就是每秒都有 CIR 的数量，而第二个桶可以直接设置为 CIR+Be 之和，称为 PIR，也就是说第二个桶总是比第一个桶要大，用户的流量总是以第二个桶的大小传输，而不用像单速三色的令牌桶算法中，需要在上一秒钟以低于 CIR 的速度传输。当用户的数据通过接口时，总是先检查第二个桶的最大速率，即 PIR，如果超出则采取动作，如果未超出，再检查是是否符合第一个桶的 CIR，如果超出 CIR，则采取相应动作，如果未超过，则正常传输。

虽然在双速三色的令牌桶算法中，直接设置两个速率，然而，用户可以直接以 CIR+Be 之和的速率进行传输，此外，还可以判断出三种结果。

分类和标记（Classification and Marking）

要实施区分服务的 QOS，就必须先将数据分为不同的类别，或者将数据设置为不同的优先级。将数据分为不同的类别，称为分类(classification)，分类并不修改原来的数据包。将数据设置为不同的优先级称为标记(marking)，而标记会修改原来的数据包。分类和标记是实施 QOS 的前提，也是基础。要正确对数据包进行分类和标记，需要了解数据包的某些特征，最重要的就是数据包的包头：

分类时，可以使用下列任何一种标准来识别特定的流量：

OSI 参考模式第一层特征：

物理接口

子接口

PVC

OSI 参考模式第二层特征：

MAC 地址

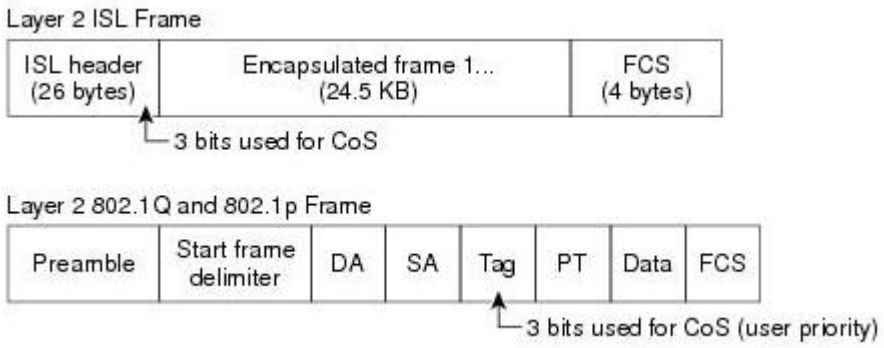
802.1Q 或 802.1P 服务类别（COS）

VLAN 标识

帧中继可丢弃指标符（DE）位

802.1Q 或 802.1P 服务类别（COS）

在以太网中，穿越 trunk 的数据将会被额外封装，如封装成 IEEE 802.1Q 或者 ISL 帧，因此，可以利用 trunk 帧头的额外部分来标记，这些字段被称为 class of service (CoS)，如下图：



Inter-Switch Link (ISL)帧中，预留有 1-byte 的 IEEE 802.1p 字段，其中有 3 bits 可以标记 CoS。

IEEE 802.1Q 帧中，预留有 2-byte 字段，其中同样只有 3 bits 可以标记 CoS，而 IEEE 802.1Q 帧中，native VLAN 是不能被标记的，因为没有额外封装。

CoS 中由于只有 3 bit 可以标记，所以只能标记出 0-7 共 8 类数据，默认标为 0，然而 6 和 7 是被保留的，因此只有 0-5 共 6 类可供用户标记使用。

帧中继可丢弃指标符（DE）位

在帧中继数据包中，有额外的一个字段可以用来指示该数据包的优先级，这个字段被称为可丢弃指标符 Discard eligible (DE)位，默认为 0，设置为 1 表示该数据不重要而优先被丢弃。

注：二层帧头会因为物理传输介质的改变而改变，所以二层标记并不具备端到端的意义。

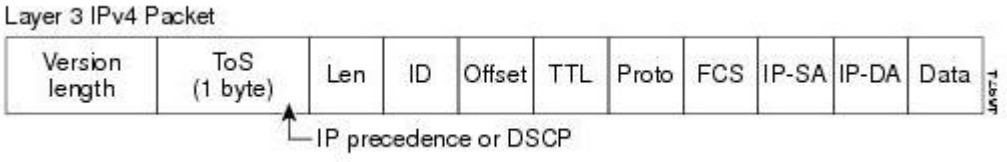
OSI 参考模式第三层特征：

IP Precedence （IP 优先级）

DiffServ 代码点（DSCP）

源 IP 地址或目的 IP 地址

在第三层 IP 数据包头，其中 ToS 字段预留 1-Byte 可供标记使用，如下图：



IP 优先级同 CoS 一样只使用 3 bit 标记，范围是 0-7 共 8 类数据，默认标为 0，然而 6 和 7 是被保留的，只有 0-5 共 6 类可供用户标记使用。

5 建议用于语音

4 由视频会议和流式视频共享

3 建议用于呼叫信令

而 DSCP 则使用 6 bit 标记，范围是 0-63 共 64 类数据。

由于 IP 数据包包头 ToS 字段只有 8 bit 可用，IP 优先级标记时使用 3bit，而 DSCP 标记时使用 6 bit， 要同时标记 IP 优先级和 DSCP 需要使用 9 bit，所以 ToS 中，只能标记 IP 优先级和 DSCP 其中一种。

注： IP 优先级 和 DSCP 被广泛用于标记选项，具有端到端的意义。

IP 优先级 、DSCP、CoS 之间的值可以互相映射。

OSI 参考模式第四层特征：

TCP 或 UDP 端口号

OSI 参考模式第三层特征：

URL

基于网络的应用识别 network-based application recognition (NBAR)，也就是匹配应用程序，通过使用数据包描述语言模块 PDLM 来识别。

注：

建议在尽可能靠近源的地方实施分类和标记

在使用 NBAR 匹配应用程序和使用 Set 标记数据包时，需要开启 CEF 功能

流（Flow）

当数据包的源 IP，目的 IP，协议，端口号，以及会话的 socket 全部相同时，这样的数据被认为是同一个流（flow），同一个流，通常应该得到相同的 QOS 服务。

比如使用某一台主机访问某网站的网页和视频，网页数据和视频数据的源 IP、目的 IP、协议即使都是相同的，但是端口号会不同，即使端口号是相同的，但会话的 socket 也绝不相同，因此，网页的数据因为五个参数都相同而被归为同一个流，而视频的数据也因为五个参数都相同而被归为另一个流，每个流便可设置不同的 QOS 策略。

管制和整形（Policing and Shaping）

在实施 QOS 策略时，可以将用户的数据限制在特定的带宽，当用户的流量超过额定带宽时，超过的带宽将不能被传输，只能采取其它方式来处理，如果处理方式为丢弃超出带宽，那么这种行为称为管制(Policing)，如果是将超出的带宽缓存在内存中，等到下一秒再传递，这种行为称为整形（Shaping）。

当使用管制时，用户的数据包超过额定带宽后，将会被丢弃，当使用的传输协议为 TCP 时，被丢弃的数据会重传，而传输协议为 UDP 时，用户是无法知道数据被丢弃的，所以管制需要小心使用。

当使用整形时，用户的数据包超过额定带宽后，是不会被丢弃的，而是缓存到内存中等到下一秒再传输，整形使超额的数据能够从接口平滑地输出，但并不表示整形就永远不会丢包。

整形分为三种：

Generic Traffic Shaping (GTS)

Frame Relay Traffic Shaping (FRTS)

Class-Based Shaping

鸿鹄培训+考试群：301283867

Generic Traffic Shaping (GTS)为通用流量整形，适用于任何接口。

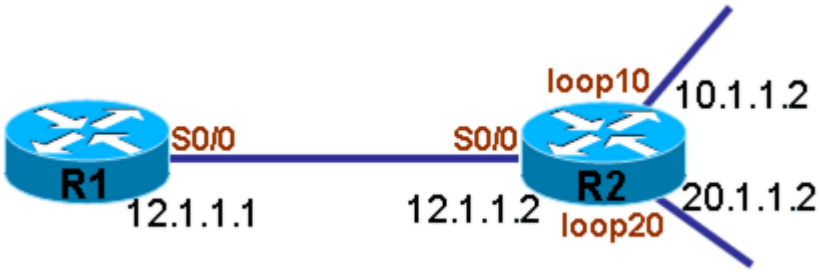
Frame Relay Traffic Shaping (FRTS) 为帧中继流量整形，是专门针对帧中继接口使用的整形技术，通过帧中继专有技术 map-class 来实现。

Class-Based Shaping 是基于类的整形，可以将 GTS 嵌入 FRTS。

注：在整形中，Bc 建议设置为 CIR 的百分之一，即 Bc/CIR=1/100。

配置管制

要对特定的流量进行管制，从而限制其可用带宽，方法为通过 MQC 的形式，先匹配特定的流量，再使用令牌桶算法将流量限制在额定的带宽，对于超额的流量再做其它处理。



说明:以上图为例，将 R1 到目标网络 10.1.1.0/24 的流量管制到 CIR 为 8000 bit，超过的流量则丢弃，R1 到目标网络 20.1.1.0/24 的流量正常通过。

1. 匹配 R1 到目标网络 10.1.1.0/24 的流量

（1）通过 ACL 匹配到网络 10.1.1.0/24 的流量

r1(config)#access-list 100 permit ip any 10.1.1.0 0.0.0.255

（2）class-map 调用 ACL 中的流量

r1(config)#class-map net10

r1(config-cmap)#match access-group 100

2. 管制到目标网络 10.1.1.0/24 的流量

(1)在 policy-map 中调用 class-map 的流量并管制带宽

r1(config)#policy-map band

r1(config-pmap)#class net10

r1(config-pmap-c)# police cir 8000 bc 1000 be 1000 conform-action transmit exceed-action drop

说明:配置中 CIR 为 8000 bit，超过的流量为 drop，关键字 police 则是开启管制（Policing）。

3. 应用策略到接口

（1）将 policy-map 应用到 R1 的 S0/0 出方向

r1(config)#int s0/0

r1(config-if)#service-policy output band

4. 测试效果

（1）测试 R1 到目标网络 20.1.1.0/24 的流量

r1#ping 20.1.1.2 size 800 repeat 20

Type escape sequence to abort.

Sending 20, 800-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

!!!!!!!!!!!!!!!!!!!!

Success rate is 100 percent (20/20), round-trip min/avg/max = 404/406/409 ms

r1#

说明:因为并没有对 R1 到目标网络 20.1.1.0/24 的流量进行管制，所以当数据包每个以 800 字节通过时，一切正常，并且速度正常。

（2）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 20

Type escape sequence to abort.

Sending 20, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!.!.!.!.!.!.!.!.!.!.

Success rate is 50 percent (10/20), round-trip min/avg/max = 405/405/409 ms

r1#

说明:因为 R1 到目标网络 10.1.1.0/24 的流量被管制为 8000bit 每秒，所以当数据包每个以 800 字节通过时，流量超出额定带宽，从而出现了超额流量被均衡地丢弃。

（3）查看 policy-map 参数

r1#sh policy-map interface s0/0

Serial0/0

Service-policy output: band

Class-map: net10 (match-all)

55 packets, 49020 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: access-group 100

police:

cir 8000 bps, bc 1000 bytes

conformed 20 packets, 16080 bytes; actions:

transmit

exceeded 35 packets, 32940 bytes; actions:

drop

conformed 0 bps, exceed 0 bps

Class-map: class-default (match-any)

237 packets, 86407 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: any

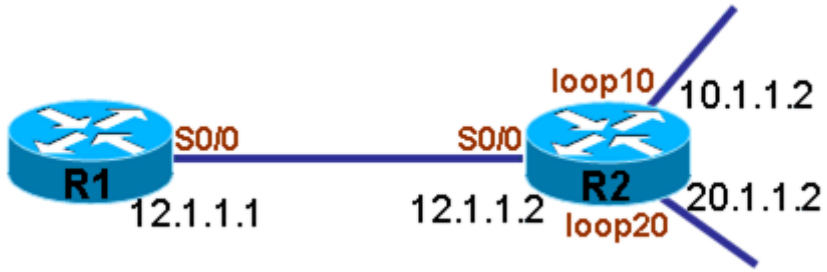
r1#

说明:从输出中可以看出超额后丢弃的数据包个数。

配置整形

要对特定的流量进行整形，从而限制其可用带宽，方法为先匹配特定的流量，再使用令牌桶算法将流量限制在额定的带宽，但是对于超额的流量不需要做其它处理，因为这些超额的流量是需要被缓存的。下面分别介绍三种整形技术的配置方法：

Generic Traffic Shaping (GTS)



说明:以上图为例，将 R1 到目标网络 10.1.1.0/24 的流量整形到 CIR 为 8000 bit，而超额的流量不需要做其它处理，这些超额的流量默认被缓存；R1 到目标网络 20.1.1.0/24 的流量正常通过。

1. 匹配 R1 到目标网络 10.1.1.0/24 的流量

（1）通过 ACL 匹配到网络 10.1.1.0/24 的流量

r1(config)#access-list 100 permit ip any 10.1.1.0 0.0.0.255

（2）class-map 调用 ACL 中的流量

r1(config)#class-map net10

r1(config-cmap)#match access-group 100

2. 整形到目标网络 10.1.1.0/24 的流量

(1)在 policy-map 中调用 class-map 的流量并整形带宽

r1(config)#policy-map SSS

r1(config-pmap)#class net10

r1(config-pmap-c)#shape average 8000 1000 0

说明:整形的 CIR 为 8000bit,虽然建议 Bc 为 CIR 的 1/100,但 Bc 请不要小于 1000,否则整形不会生效，而 Be 可以为 0。

3. 应用策略到接口

（1）将 policy-map 应用到 R1 的 S0/0 出方向

r1(config)#int s0/0

r1(config-if)#service-policy output SSS

4. 测试效果

（1）测试 R1 到目标网络 20.1.1.0/24 的流量

r1#ping 20.1.1.2 size 800 repeat 20

Type escape sequence to abort.

Sending 20, 800-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

!!!!!!!!!!!!!!!!!!!!

Success rate is 100 percent (20/20), round-trip min/avg/max = 404/406/409 ms

r1#

说明:因为并没有对 R1 到目标网络 20.1.1.0/24 的流量进行整形，所以当数据包每个以 800 字节通过时，一切正常，并且速度正常。

（2）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 20

Type escape sequence to abort.

Sending 20, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!!!!!!!!!!!!!!!!

Success rate is 100 percent (20/20), round-trip min/avg/max = 405/406/409 ms

r1#

说明:因为 R1 到目标网络 10.1.1.0/24 的流量被整形为 8000bit 每秒，所以当数据包每个以 800 字节通过时，并没有丢包，但是速度却不能超过 8000bit 每秒，此效果在文本下无法看见，需要自己在实验时查看。

（3）查看 policy-map 参数

r1#show policy-map interface

Serial0/0

Service-policy output: SSS

Class-map: net10 (match-all)

20 packets, 16080 bytes

5 minute offered rate 1000 bps, drop rate 0 bps

Match: access-group 100

Traffic Shaping

Target/Average Rate	Byte Limit	Sustain bits/int	Excess bits/int	Interval (ms)	Increment (bytes)
8000/8000	125	1000	0	125	125

Adapt Active	Queue Depth	Packets	Bytes	Packets Delayed	Bytes Delayed	Shaping Active

- 0 20 16080 19 15276 no

Class-map: class-default (match-any)

29 packets, 16197 bytes

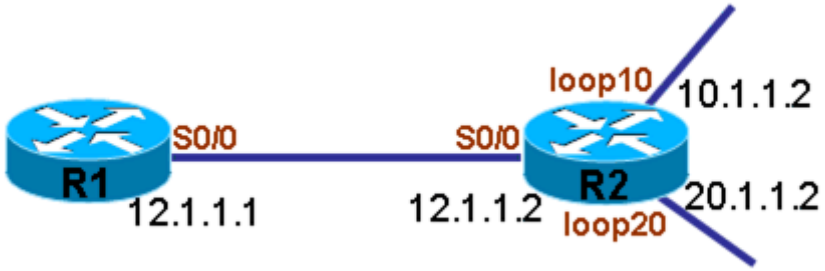
5 minute offered rate 0 bps, drop rate 0 bps

Match: any

r1#

说明:从输出中可以看出被整形的数据包个数以及其它一些参数。

Frame Relay Traffic Shaping (FRTS)



说明:以上图为例，将 R1 的出口流量整形到 CIR 为 8000 bit，而超额的流量不需要做其它处理，这些超额的流量默认被缓存。

1. 配置 map-class实现 FRTS

（1）配置 map-class

r1(config)#map-class frame-relay TTT

r1(config-map-class)#frame-relay cir 8000

r1(config-map-class)#frame-relay bc 1000

r1(config-map-class)#frame-relay be 0

说明:FRTS 的工具 map-class 不能针对特定流量整形，而只能对接口所有流量整形。

2. 应用 FRTS

（1）在帧中继接口上开启整形

r1(config)#int s0/0

r1(config-if)#frame-relay traffic-shaping

（2）应用 map-class

r1(config-if)#frame-relay interface-dlci 102

r1(config-fr-dlci)#class TTT

说明:map-class 可以单独应用到某条 PVC 上，也可以应用在整个接口上。

3. 测试效果

（1）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 20

Type escape sequence to abort.

Sending 20, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

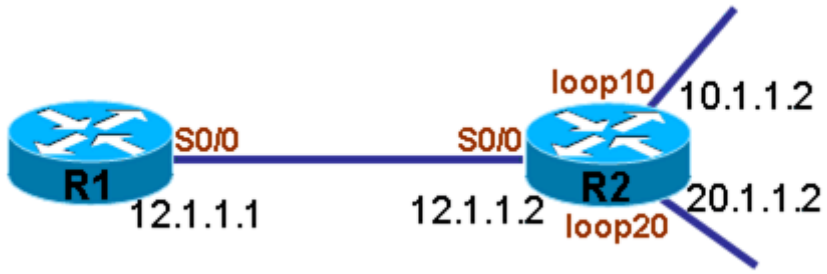
!!!!!!!!!!!!!!!!!!!!

Success rate is 100 percent (20/20), round-trip min/avg/max = 405/406/409 ms

r1#

说明:因为 R1 的出口流量被整形到 CIR 为 8000bit 每秒，所以当数据包每个以 800 字节通过时，并没有丢包，但是速度却不能超过 8000bit 每秒，此效果在文本下无法看见，需要自己在实验时查看。

Class-Based Shaping



说明:以上图为例，将 R1 的出口流量整形到 CIR 为 8000 bit，而超额的流量不需要做其它处理，这些超额的流量默认被缓存。Class-Based Shaping 和 FRTS 一样只能对所有流量整形。class-default

1. 配置 GTS

（1）policy-map 中调用 class-map 的流量并整形带宽

r1(config)#policy-map ccc

r1(config-pmap)#class class-default

r1(config-pmap-c)#shape average 8000 1000 0

说明: Class-Based Shaping 只支持对 class-default 的整形，即对所有流量整形。

2. 配置 Class-Based Shaping

（1）在 map-class 中调用 GTS

r1(config)#map-class frame-relay FFF

r1(config-map-class)#service-policy output ccc

3. 应用 Class-Based Shaping

（1）在帧中继接口上应用 Class-Based Shaping

r1(config)#int s0/0

r1(config-if)#frame-relay class FFF

说明:在应用 Class-Based Shaping 时，接口上必须禁用 frame-relay traffic shaping。

4. 测试效果

（1）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 20

Type escape sequence to abort.

Sending 20, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!!!!!!!!!!!!!!!!

Success rate is 100 percent (20/20), round-trip min/avg/max = 405/406/409 ms

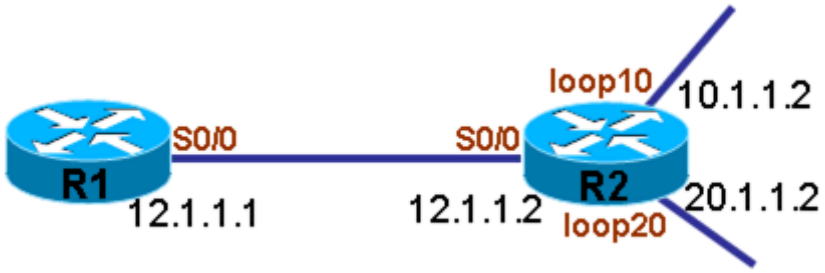
r1#

说明:因为 R1 的出口流量被整形到 CIR 为 8000bit 每秒，所以当数据包每个以 800 字节通过时，并没有丢包，但是速度却不能超过 8000bit 每秒，此效果在文本下无

法看见，需要自己在实验时查看。

接口直接开启整形

除了以上的整形方法之外，接口上可以直接配置流理整形，这也是对所有流量生效



说明:以上图为例，将 R1 的出口流量整形到 CIR 为 8000 bit。

1. 开启接口流量整形

（1）在 R1 的 S0/0 上开启整形

r1(config)#int s0/0

r1(config-if)#traffic-shape rate 8000 1000 0

2. 测试效果

（1）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 20

Type escape sequence to abort.

Sending 20, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!!!!!!!!!!!!!!!!

Success rate is 100 percent (20/20), round-trip min/avg/max = 405/406/409 ms

r1#

说明:因为 R1 的出口流量被整形到 CIR 为 8000bit 每秒，所以当数据包每个以 800 字节通过时，并没有丢包，但是速度却不能超过 8000bit 每秒，此效果在文本下无法看见，需要自己在实验时查看。

（2）查看接口整形

r1#sh traffic-shape statistics

	Acc. Queue Depth	Packets	Bytes	Packets Delayed	Bytes Delayed	Shaping Active
I/F	List					
Se0/0	0	20	16080	19	15276	no

r1#

说明:可以看到被整形的数据包个数。

Committed access rate (CAR)承诺访问速率

CAR 是一个 Cisco 工具，有两种功能，可以对数据包进行标记，也可以对流量实现管制。

既然 CAR 可以对数据标记，也就是可以改变数据包的包头信息，CAR 同样可以通过令牌桶算法对流量进行管制，将流量限制在额定的带宽。

在配置 CAR 时，流量可以是进和出任意方向的，因为 CAR 可以标记，所以在配置 CAR 时，必须开启 CEF，并且可以配置在主接口和子接口上，但 Fast EtherChannel, tunnel,，PRI 接口是不能配的，以及不支持 CEF 的接口也不能配置 CAR。

CAR 可以基于接口方向，基于 IP 优先级，以及基于 ACL 和 MAC 地址生效，但是需要注意 MAC 地址是二层地址，经过二层转换之后，MAC 地址信息将会丢失。

一个接口上可以配置多条 CAR。

对于匹配的流量，CAR 的动作有：

continue （就是检测下一条 CAR）

drop （丢弃匹配的包）

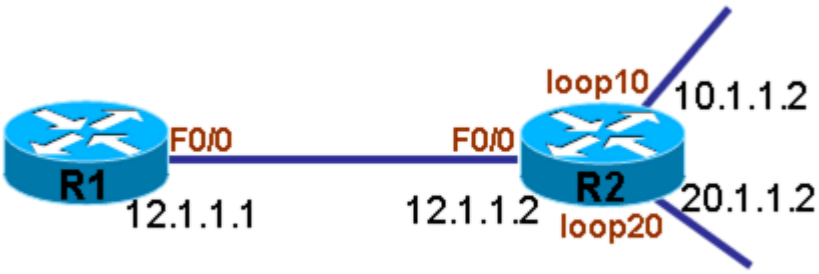
set-prec-continue new-prec （设置新的 IP 优先级，然后检测下一条 CAR）

set-prec-transmit new-prec （设置新的 IPP 然后传输）

transmit （直接传输）

注：以上信息，根据 IOS 不同，会有所增减。

配置 CAR



配置基于接口的 CAR

说明:配置 R1，将所有从接口 F0/0 出去的流量全部管制到 CIR 8000，超过的流量则丢弃

1. 在接口 F0/0 上配置 CAR 管制所有流量

（1）在 R1 的接口 F0/0 上配置 CAR

r1(config)#int f0/0

r1(config-if)#rate-limit output 8000 1500 2000 conform-action transmit exceed-action drop

2. 测试效果

（1）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 10

Type escape sequence to abort.

Sending 10, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!.!!.!!!.

Success rate is 70 percent (7/10), round-trip min/avg/max = 1/2/4 ms

r1#

说明:因为配置了 CAR 管制所有流量，所以 R1 到 10.1.1.0/24 的流量只能以 8000 bit 每秒的速度传输，超过的则被丢弃。

（2）测试 R1 到目标网络 20.1.1.0/24 的流量

r1#ping 20.1.1.2 size 800 repeat 10

Type escape sequence to abort.

Sending 10, 800-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

!!!.!!.!!!

Success rate is 80 percent (8/10), round-trip min/avg/max = 1/2/4 ms

r1#

说明:因为配置了 CAR 管制所有流量，所以 R1 到 20.1.1.0/24 的流量只能以 8000 bit 每秒的速度传输，超过的则被丢弃。

（3）查看接口 CAR 信息

r1#sh interfaces rate-limit

FastEthernet0/0

Output

matches: all traffic

params: 8000 bps, 1500 limit, 2000 extended limit

conformed 47 packets, 38258 bytes; action: transmit

exceeded 13 packets, 10582 bytes; action: drop

last packet: 17529ms ago, current burst: 1668 bytes

last cleared 00:01:46 ago, conformed 2000 bps, exceeded 0 bps

r1#

说明:可以看到接口上被 CAR 匹配到数据的状况。

配置基于 ACL 的 CAR

说明:配置 R1，将所有去往 10.1.1.0/24 的流量全部管制到 CIR 8000，超过的流量

则丢弃

1. 配置 ACL

（1）使用 ACL 匹配去往 10.1.1.0/24 的流量

r1(config)#access-list 100 permit ip any 10.1.1.0 0.0.0.255

2. 配置 CAR

（1）在 R1 的接口 F0/0 上配置 CAR

r1(config)#int f0/0

r1(config-if)#rate-limit output access-group 100 8000 1500 2000 conform-action transmit exceed-action drop

3. 测试效果

（1）测试 R1 到目标网络 20.1.1.0/24 的流量

r1#ping 20.1.1.2 size 800 repeat 10

Type escape sequence to abort.

Sending 10, 800-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

!!!!!!!!!!

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/3/4 ms

r1#

说明:因为 CAR 不对去往 20.1.1.0/24 的流量进行管制，所以去往 20.1.1.0/24 的流量正常通过。

（2）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 10

Type escape sequence to abort.

Sending 10, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!.!.!!!

Success rate is 80 percent (8/10), round-trip min/avg/max = 1/2/4 ms

r1#

说明:因为 CAR 对去往 10.1.1.0/24 的流量进行管制，所以去往 10.1.1.0/24 的流量只能以 8000bit 每秒的速度通过，超过的流量被丢弃。

（3）查看接口 CAR 信息

r1#sh interfaces rate-limit

FastEthernet0/0

Output

matches: access-group 100

params: 8000 bps, 1500 limit, 2000 extended limit

conformed 8 packets, 6512 bytes; action: transmit

exceeded 2 packets, 1628 bytes; action: drop

last packet: 16483ms ago, current burst: 1672 bytes

last cleared 00:01:25 ago, conformed 0 bps, exceeded 0 bps

r1#

说明:可以看到接口上被 CAR 匹配到数据的状况。

配置基于 DSCP 的 CAR

说明:R1 将去往 10.1.1.0/24 的流量的 DSCP 值设置为 5，让对方路由器 R2 根据 DSCP 值设置 CAR，将 DSCP 为 5 的流量全部丢弃，其它正常通过。

1. 配置 ACL

（1）使用 ACL 匹配去往 10.1.1.0/24 的流量

r1(config)#access-list 100 permit ip any 10.1.1.0 0.0.0.255

2. 配置 CAR

（1）在 R1 的接口 F0/0 上配置 CAR

r1(config)#int f0/0

r1(config-if)#rate-limit output access-group 100 8000 1500 2000 conform-action set-dscp-transmit 5 exceed-action set-dscp-transmit 5

说明:CAR 将所有符合与超出的流量的 DSCP 值全部设置成 5 之后再发出去。

3. 在 R2 上配置 CAR

（1）在 R1 的接口 F0/0 上配置 CAR 丢弃 DSCP 值为 5 的流量

r2(config)#int f0/0

r2(config-if)#rate-limit input dscp 5 8000 1500 2000 conform-action drop exceed-action drop

说明:CAR 将 DSCP 值为 5 的流量，无论是符合还是超出都全部丢弃。

4. 测试效果

（1）测试 R1 到目标网络 20.1.1.0/24 的流量

r1#ping 20.1.1.2 size 800 repeat 10

Type escape sequence to abort.

Sending 10, 800-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

!!!!!!!!!!

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/3/4 ms

r1#

说明:因为 CAR 只丢弃 DSCP 值为 5 的流量，而去往 20.1.1.0/24 的流量的 DSCP 值并没有被设置成 5，所以正常通过。

（2）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 10

Type escape sequence to abort.

Sending 10, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

.....

Success rate is 0 percent (0/10)

r1#

说明:因为 CAR 只丢弃 DSCP 值为 5 的流量，而去往 10.1.1.0/24 的流量的 DSCP 值全部被设置成 5，所以全部被丢弃。

（3）查看接口 CAR 信息

r2#sh interfaces rate-limit

FastEthernet0/0

Input

matches: dscp 5

params: 8000 bps, 1500 limit, 2000 extended limit

conformed 10 packets, 8140 bytes; action: drop

exceeded 0 packets, 0 bytes; action: drop

last packet: 169518ms ago, current burst: 0 bytes

last cleared 00:03:33 ago, conformed 0 bps, exceeded 0 bps

r2#

说明:可以看到接口上被 CAR 匹配到数据的状况。

配置基于 MAC 地址的 CAR

说明:配置 R1，将源 MAC 地址为 R2 接口 F0/0 的数据包全部丢弃。

1. 配置匹配 MAC 地址的 ACL

（1）在 R2 上查看接口 F0/0 的 MAC 地址

r2#sh int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is AmdFE, address is 0013.1a2f.1200 (bia 0013.1a2f.1200)

Internet address is 12.1.1.2/24

说明:R2 上接口 F0/0 的 MAC 地址为 0013.1a2f.1200。

（2）在 R1 上配置匹配 R2 接口 F0/0 的 MAC 地址的 ACL

r1(config)#access-list rate-limit 100 0013.1a2f.1200

2. 配置 CAR

（1）配置 R1，将源 MAC 地址为 R2 接口 F0/0 的数据包全部丢弃

r1(config)#int f0/0

r1(config-if)#rate-limit input access-group rate-limit 100 8000 1500 2000 conform-action drop exceed-action drop

3. 测试效果

（1）测试 R2 去往 R1 的流量

r2#ping 12.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明:因为 R2 去往 R1 的数据包的源 MAC 地址，即为接口 F0/0 的 MAC 地址，所以流量全被丢弃。

（2）测试 10.1.1.0/24 去往 R1 的流量

r2#ping 12.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明:因为 10.1.1.0/24 去往 R1 的数据包的源 MAC 地址，将被 R2 改为接口 F0/0 的 MAC 地址，所以流量全被丢弃。

（3）查看接口 CAR 信息

r1#sh interfaces rate-limit

FastEthernet0/0

Input

matches: access-group rate-limit 100

params: 8000 bps, 1500 limit, 2000 extended limit

conformed 15 packets, 1710 bytes; action: drop

exceeded 0 packets, 0 bytes; action: drop

last packet: 7973ms ago, current burst: 0 bytes

last cleared 00:02:28 ago, conformed 0 bps, exceeded 0 bps

r1#

说明:可以看到接口上被 CAR 匹配到数据的状况。

拥塞管理（Congestion management）

当网络发生拥塞后，数据还是要被传递的，正因为接收到的数据远多于自身的传输能力，所以数据被传输时就出现了先后顺序，而规定数据按什么样的顺序来传输，这就是拥塞管理。也只有当拥塞发生时，拥塞管理才会生效，对超额的数据流，使用队列算法来决定如何将数据发出，而每种队列技术都指定为解决特定的网络流量问题和获得特定的性能。队列技术需要依懒已经做好的分类和标记，因为队列技术需要根据数据包的不同特征做出不同处理。

一个接口只能使用一个队列技术，需要了解的队列技术有：

FIFO queuing

Priority queuing (PQ)

Custom queuing (CQ)

Weighted fair queuing (WFQ)

Class-based WFQ (CBWFQ)

Low Latency Queuing (LLQ)

IP RTP Priority

下面分别对不同的队列技术进行详细介绍

FIFO Queuing （First In First Out Queuing）

FIFO 队列为先进先出队列，FIFO 队列不对数据包进行分类，当数据包到达接口后，数据包按照到达接口的先后顺序通过接口，数据包没有优先级之分，即使接口发生了拥塞，数据包是先到的就先通过，后到的就后通过。

某些接口是默认开启 FIFO 队列的，不需要手工配置。

速率大于 E1 (2.048 Mbps)的接口全部默认开启 FIFO 队列。

查看接口队列：

```
r1#show interfaces f0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0013.1a85.d160 (bia 0013.1a85.d160)
  Internet address is 12.1.1.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:07, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
 Queueing strategy: fifo
  Output queue: 0/40 (size/max)
```

说明:因为接口 F0/0 的速率为 100000 Kbit，大于 E1 (2.048 Mbps)，所以默认队列为 FIFO。

Priority Queuing （PQ）

PQ 被称为优先级队列，是因为 PQ 在发生拥塞时，只传优先级最高的数据，只有当优先级最高的数据全部传完之后，才会传次优先级的数据。PQ 中有 4 个队列，分别是 high，medium，normal，low ，可根据数据包的 IP 优先级或 DSCP 等标识将数据包分配到各个队列中，在发生拥塞时，PQ 先传 high 中的数据，直到全部传完之后，才会传 medium 中的数据，同样只有 medium 中的数据传完之后，才会传 normal 中的数据，最后等前面三个队列的所有数据都传完之后，才轮到 low 中的数据。由此可见，如果高优先级的队列没有传送完毕，低优先级的数据将永远不会传递，造成两级分化，使较低优先级的数据转发困难。

虽然 PQ 只有在高优先级队列数据包全部传完的情况下，才会传下一个队列，但是可以限制每个队列一次性传输的最大数据包个数，当某个队列传输的数据包达到最大数量之后，无论是否还有数据包，都必须传递下一个队列。

配置 PQ

说明:在配置时，可以根据数据包的协议，以及 ACL 等技术将特定流量放入特定队列，除了明确将特定的流量放入某个队列之后，还可以将默认没有匹配的数据统统放入默认的队列。

1. 配置 PQ 将特定流量放入特定队列

（1）将源 IP 为 10.1.1.0/24 的数据放入队列 high 中

```
r1(config)#access-list 10 permit 10.1.1.0 0.0.0.255
r1(config)#priority-list 1 protocol ip high list 10
```

（2）将源 IP 为 20.1.1.0/24 的数据放入队列 medium 中

```
r1(config)#access-list 20 permit 20.1.1.0 0.0.0.255
r1(config)#priority-list 1 protocol ip medium list 20
```

（3）将端口号为 TCP 23 的数据放入队列 normal 中

r1(config)#priority-list 1 protocol ip normal tcp 23

（4）默认其它数据都放入队列 low 中

r1(config)#priority-list 1 default low

2. 限制每个队列的最大数据包个数

（1）限制 high 为 400，medium 为 300，normal 为 200，low 为 100，默认分别为 20,40,60,80。

r1(config)#priority-list 1 queue-limit 400 300 200 100

3. 应用 PQ 到接口

（1）将 PQ 应用到接口 F0/0

r1(config)#int f0/0

r1(config-if)#priority-group 1 ?

4. 查看 PQ

（1）查看接口 F0/0 上的队列情况

r1#show interfaces f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is AmdFE, address is 0013.1a85.d160 (bia 0013.1a85.d160)

Internet address is 12.1.1.1/24

MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, loopback not set

Keepalive set (10 sec)

Full-duplex, 100Mb/s, 100BaseTX/FX

ARP type: ARPA, ARP Timeout 04:00:00

Last input 00:00:00, output 00:00:07, output hang never

Last clearing of "show interface" counters never

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: priority-list 1

Output queue (queue priority: size/max/drops):

high: 0/400/0, medium: 0/300/0, normal: 0/200/0, low: 0/100/0

说明:可以看到接口 F0/0 上应用的队列为 PQ。

（2）查看 PQ 参数

r1#sh queueing priority

Current DLCI priority queue configuration:

Current priority queue configuration:

List Queue Args

1 low default

1 high protocol ip list 10

1 medium protocol ip list 20

1 normal protocol ip tcp port telnet

1 high limit 400

1 medium limit 300

1 normal limit 200

1 low limit 100

r1#

Custom queuing (CQ)

CQ 中有 1 到 16 共 16 个队列轮循，每个队列可以限制可传的数据包总数，但实时数据不能得到保证。将数据包分配到 CQ 的各个队列中，当网络发生拥塞时，CQ 先传第 1 个队列中的数据，当传到额定的数据包个数后，就接下去传第 2 个队列中的数据，同样是传到额定的数据包个数后，再传下一个队列，以此类推荐，直到传到第 16 个队列后，再回过去传第一个队列。CQ 除了 1 到 16 个队列外，还有一个 0 号队列，但是 0 号队列是超级优先队列，路由器总是先把 0 号队列中的数据发送完后才处理 1 到 16 号队列中的数据包，所以 0 号队列一般作为系统队列，许多 IOS 不支持手工将指定数据分配到 0 号队列。在配置 1 到 16 号队列时，用户可以配置每个队列同一时间可以占用接口带宽的比例，相当于限速。

配置 CQ

说明:在配置时，可以像配置 PQ 一样，可根据数据包的协议，以及 ACL 等技术将特定流量放入特定队列，除了明确将特定的流量放入某个队列之后，还可以将默认没有匹配的数据统统放入默认的队列。

1. 配置 CQ 将特定流量放入特定队列

（1）将源 IP 为 10.1.1.0/24 的数据放入 1 号队列中

r1(config)#access-list 10 permit 10.1.1.0 0.0.0.255

r1(config)#queue-list 1 protocol ip 1 list 10

（2）将源 IP 为 20.1.1.0/24 的数据放入 2 号队列中

r1(config)#queue-list 1 protocol ip 2 list 20

（3）将端口号为 TCP 23 的数据放入 3 号队列中

r1(config)#queue-list 1 protocol ip 3 tcp 23

（4）默认其它数据都放入 4 号队列中

r1(config)#queue-list 1 default 4

2. 限制队列数据包

（1）限制各个队列每次可传的最大字节数，1 为 100，2 为 200，3 为 300，4 为 400

，默认全部为 1500。当传到最大字节数后，将转到传递下一个队列。

r1(config)#queue-list 1 queue 1 byte-count 100

r1(config)#queue-list 1 queue 2 byte-count 200

r1(config)#queue-list 1 queue 3 byte-count 300

r1(config)#queue-list 1 queue 4 byte-count 400

（2）限制各个队列每次可传的最大数据包个数，1 为 10，2 为 20，3 为 30，4 为 40

，默认全部为 20。而这个数据包个数，似乎是带宽限制，此参数不确定。

r1(config)#queue-list 1 queue 1 limit 10

r1(config)#queue-list 1 queue 2 limit 20

r1(config)#queue-list 1 queue 3 limit 30

r1(config)#queue-list 1 queue 4 limit 40

3. 应用 CQ 到接口

（1）将 CQ 应用到接口 F0/0

r1(config)#int f0/0

r1(config-if)#custom-queue-list 1

4. 查看 CQ

（1）查看接口 F0/0 上的队列情况

r1#sh int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is AmdFE, address is 0013.1a85.d160 (bia 0013.1a85.d160)

Internet address is 12.1.1.1/24

MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, loopback not set

Keepalive set (10 sec)

Full-duplex, 100Mb/s, 100BaseTX/FX

ARP type: ARPA, ARP Timeout 04:00:00

Last input 00:00:01, output 00:00:07, output hang never

Last clearing of "show interface" counters never

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: custom-list 1

Output queues: (queue #: size/max/drops)

0: 0/20/0 1: 0/10/0 2: 0/20/0 3: 0/30/0 4: 0/40/0

5: 0/20/0 6: 0/20/0 7: 0/20/0 8: 0/20/0 9: 0/20/0

10: 0/20/0 11: 0/20/0 12: 0/20/0 13: 0/20/0 14: 0/20/0

15: 0/20/0 16: 0/20/0

说明:可以看到接口 F0/0 上应用的队列为 CQ。

（2）查看 CQ 参数

r1#show queueing custom

Current custom queue configuration:

List	Queue	Args
1	4	default
1	1	protocol ip list 10
1	2	protocol ip list 20
1	3	protocol ip tcp port telnet
1	1	byte-count 100 limit 10
1	2	byte-count 200
1	3	byte-count 300 limit 30
1	4	byte-count 400 limit 40

r1#

Weighted Fair Queuing (WFQ)

WFQ 是一个基于 Weight 的公平队列，之所以说 WFQ 是公平的，是因为 WFQ 根据数据包的 IP 优先级来分配相应的带宽，优先级高的数据包，分到的带宽就多，优先级低的数据包，分到的带宽就少，并且所有的数据包在任何时刻都可以分到带宽，这就是它的公平之处。WFQ 在根据 IP 优先级给数据包分配带宽时，是基于流

（flow）来分配的，也就是说每个流的数据包分配相同的带宽，只有不同的流，才可能分配不同的带宽，如果两个流的 IP 优先级是一样的，那么这两个流分配到的带宽也是一样的。要区分数据包是不是同一个流，需要五个参数完全相同，也就是数据包的源 IP，目的 IP，协议，端口号，以及会话的 socket 全部相同时，这样的数据才被认为是同一个流，既然如此，所以手工是没有办法将两个不同的数据包划分到同一个流的，而计算数据包是不是同一个流，必须由系统自己计算，不需要人工干预。

在配置 WFQ 时，最好已经将不同的数据设置好不同的 IP 优先级，否则所有的流得到的带宽都是一样的，而没办法保证重要的流量。WFQ 根据每个流的 IP 优先级，将接口的可用带宽分配给每个流，计算方法为：

例如现有 4 个流，IP 优先级分别为 0 、1、3、5，WFQ 将所有流的 IP 优先级相加，结果为 0+1+3+5=9，而 9 做为分母；然后每个流的 IP 优先级作为分子，最后得出每个流分配到的带宽为 优先级为 0 得到的带宽为 0/9，优先级为 1 得到的带宽为 1/9，优先级为 3 得到的带宽为 3/9，优先级为 5 得到的带宽为 5/9，这样就可以依靠流的 IP 优先级分配相应的带宽了，但是从上面的算法中可以看出，此算法并不可行，因为优先级为 0 的流得到的带宽为 0/9，而 0/9 就等于 0，也就是优先级为 0 的流得到的带宽为 0，那就是没有带宽，大家知道默认所有的数据包的 IP 优先级恰恰为 0，所以为了防止默认的数据包得不到带宽，最后需要将上面的算法重新调整，结果为 4 个流的 IP 优先级分别为 0 、1、3、5，将原来的优先级全部加 1，分别为 0+1=1 、1+1=2、3+1=4、5+1=6，然后将这些值全部相加，结果为 1+2+4+6=13，将 13 做为分母，然后每个流的 IP 优先级加 1 后的值做为分子，最后得出每个流分配到的带宽为 优先级为 0 得到的带宽为 1/13，优先级为 1 得到的带宽为 2/13，优先级为 3 得到的带宽为 4/13，优先级为 5 得到的带宽为 6/13，这样就有效地防止了默认数据包无法分配到带宽的情况，而且又保证了优级先为 0 的流分到的带宽最少。

注：所有带宽小于或等于 E1 (2.048 Mbps)的接口，默认都启用了 WFQ，即使手工更改带宽后，仍无法改变默认队列机制。

配置 WFQ

1. 查看接口默认 WFQ：

（1）查看接口 S1/0 的默认队列

```
Router#show interfaces s1/0
Serial1/0 is administratively down, line protocol is down
  Hardware is M4T
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Restart-Delay is 0 secs
  CRC checking enabled
  Last input never, output 00:01:21, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
```

说明:因为接口 S1/0 的带宽为 1544 Kbit，也就是小于 E1 (2.048 Mbps)，所以看到接口默认的队列为 weighted fair（WFQ）。

2. 配置 WFQ

（1）在接口 F0/0 上开启 WFQ

Router(config)#int f0/0

Router(config-if)#fair-queue

3. 查看 WFQ

（1）查看接口 F0/0 的队列情况

Router#show interfaces f0/0

FastEthernet0/0 is administratively down, line protocol is down

Hardware is Gt96k FE, address is c000.0fec.0000 (bia c000.0fec.0000)

MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, loopback not set

Keepalive set (10 sec)

Half-duplex, 10Mb/s, 100BaseTX/FX

ARP type: ARPA, ARP Timeout 04:00:00

Last input never, output never, output hang never

Last clearing of "show interface" counters never

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: weighted fair

Output queue: 0/1000/64/0 (size/max total/threshold/drops)

Conversations 0/0/256 (active/max active/max total)

Reserved Conversations 0/0 (allocated/max allocated)

Available Bandwidth 7500 kilobits/sec

说明:虽然接口 F0/0 的带宽大于 E1 (2.048 Mbps)，但由于手工配置，所以现在的队

列为 weighted fair（WFQ）。

Class-based WFQ (CBWFQ)

CBWFQ 的工作原理和 WFQ 是一样的，是基于 WFQ 的，也是对 WFQ 的扩展。因为 CBWFQ 和 WFQ 原理一样，所以不再介绍计算方式。在接口上配置上 WFQ 之后，系统就会将接口所有可用带宽按每个流的 IP 优先级，公平地分给每一个流，并且，接口所有的流都是同时基于接口的全部可用带宽来分配的。CBWFQ 要对 WFQ 进行扩展和优化，就是要为特定的流量划分特定的带宽，让这些特定的流量在分配带宽时，只能从这些划分到的特定带宽中分配，而不是像 WFQ 一样从接口的全部可用带宽中分配。举个例子来解释 WFQ 和 CBWFQ 的区别，例如 A、B、C、D、E 共 5 个人分 100 斤大米，在使用 WFQ 时，是根据 A、B、C、D、E 这五个人的优先级，将 100 斤大米公平地分给五个人的，如果使用 CBWFQ，就可以规定将 80 斤大米按优先级只分配给 A、B、C 三个人，而再将 20 斤大米按优先级分配给 D、E 两个人，可以看出，使用 WFQ 时，竞争大米时，是所有人分配所有的大米，而使用 CBWFQ 时，是不同人群，分配不同数量的大米。

配置 CBWFQ 的方法为 MQC 形式，使用 class-map 匹配指定的流量，然后使用 policy-map 为指定的流量划分特定的带宽，这样之后，指定的流量就只能依靠 IP 优先级从特定的带宽中分配带宽。最后将 CBWFQ 应用到接口，需要注意的是，CBWFQ 只能用在接口的 out 方向。

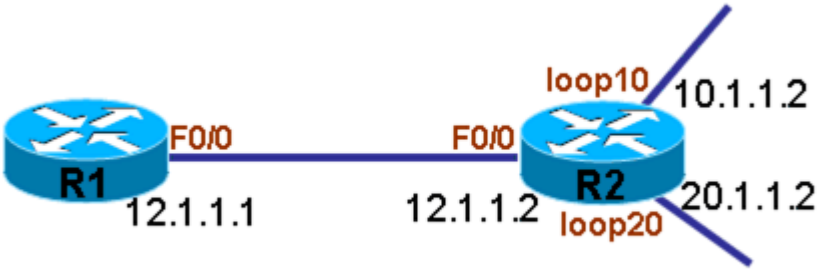
当划分特定的带宽给某类流量之后，这个带宽是绝对能够为此类流量保证的，而不会被其它流量所抢占，但是如果某类流量超过了被划分的带宽，那么超出的流量将实行尾丢弃。

某个接口的总带宽并不是全部都能被 WFQ 或者 CBWFQ 所使用，默认情况下，一个接口能使用的带宽最多不能超过 75%，所以接口总带宽的 75%才是可用总带宽，还有保留的 25%是不能使用的，但是可用总带宽是可以随意修改的。当配置 CBWFQ 时，默认没有分配带宽的流量，全部使用保留的 25%。

在配置 CBWFQ 时，接口必须处于默认的队列状态下，并且不支持以太网接口的子接口。

注： 一个 CBWFQ，最多可配置 64 类数据流。

配置 CBWFQ



说明：在 R2 上配置 CBWFQ，限制源地址为 10.1.1.0/24 的流量，从带宽 1000 Kbit 中分配带宽，限制源地址为 20.1.1.0/24 的流量，从带宽 2000 Kbit 中分配带宽，而其它流量从所有剩余带宽中分配。

1. 配置 ACL 匹配相应流量

（1）通过 ACL 匹配源地址为 10.1.1.0/24 的流量

r2(config)#access-list 10 permit 10.1.1.0 0.0.0.255

（2）通过 ACL 匹配源地址为 20.1.1.0/24 的流量

r2(config)#access-list 20 permit 20.1.1.0 0.0.0.255

2. 使用 class-map对流量分类

（1）通过 class-map 匹配源地址为 10.1.1.0/24的流量

r2(config)#class-map net10

r2(config-cmap)#match access-group 10

（2）通过 class-map 匹配源地址为 20.1.1.0/24的流量

r2(config)#class-map net20

r2(config-cmap)#match access-group 20

3. 使用 policy-map 为各流量划分带宽

（1）为源地址为 10.1.1.0/24 的流量划分带宽

r2(config)#policy-map CBW

r2(config-pmap)#class net10

r2(config-pmap-c)#bandwidth 1000

说明:命令 bandwidth 就是 CBWFQ 为某些流量划分带宽。

（2）为源地址为 20.1.1.0/24 的流量划分带宽

r2(config)#policy-map CBW

r2(config-pmap)#class net20

r2(config-pmap-c)#bandwidth 2000

（3）其它所有流量从所有剩余可用带宽中分配

说明:当从接口所有可用带宽中划分部分带宽给某类流量之后，那么剩余的可用带宽可以使用百分比的形式分给其它流量，但是需要注意，只有之前的流量也是使用百分比形式划分带宽时，才可以使用剩余可用带宽的方式划分带宽，所以此步只是举例，并不真实，除非将之前的配置改为百分比划分形式。

r2(config)#policy-map CBW

r2(config-pmap)#class class-default

r2(config-pmap-c)#bandwidth remaining percent 100

说明:要让此步配置生效，必须将前面配置改为同样使用百分比分配带宽的形式。

4. 在接口 F0/0 上应用 CBWFQ

（1）改变接口可用带宽总数（此步为可选配置）

r2(config)#int f0/0

r2(config-if)#max-reserved-bandwidth 90

说明:将接口 F0/0 的可用带宽总数从 75%更改到 90% 。

（2）在接口 F0/0 上应用 CBWFQ

说明:CBWFQ 只能应用在接口 out 方向

r2(config)#int f0/0

r2(config-if)#service-policy output CBW

5. 查看 CBWFQ

（1）查看接口 F0/0 的队列机制

r2#sh int f0/0

FastEthernet0/0 is administratively down, line protocol is down

Hardware is Gt96k FE, address is c000.0fec.0000 (bia c000.0fec.0000)

MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation ARPA, loopback not set

Keepalive set (10 sec)

Half-duplex, 10Mb/s, 100BaseTX/FX

ARP type: ARPA, ARP Timeout 04:00:00

Last input never, output never, output hang never

Last clearing of "show interface" counters never

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: Class-based queueing

Output queue: 0/1000/64/0 (size/max total/threshold/drops)

Conversations 0/0/256 (active/max active/max total)

Reserved Conversations 2/2 (allocated/max allocated)

Available Bandwidth 4500 kilobits/sec

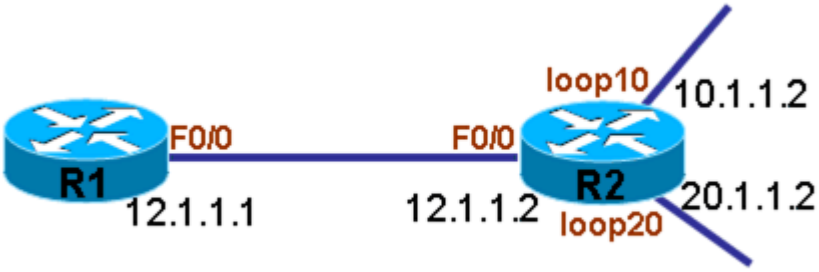
说明:可以看到接口上的队列为 Class-based queueing （CBWFQ）

Low Latency Queuing（LLQ）

在之前的队列技术中，没有办法保证重要的数据能够一直得到服务，并且一直都有足够的带宽，而且还要不阻断其它流量。比如对延迟和抖动较敏感的语音或视频数据，语音或视频在通信时，需要一直保持着足够的带宽，否则就会受到影响，对于这样的情况，就需要一种队列技术能够为特定的数据流保证特定量的带宽。LLQ 低延迟队列正是为对延迟和抖动较敏感的语音或视频流量设计的，LLQ 为特定的流量划分特定的带宽，划给特定流量的带宽是绝对能够保证的，无论接口有多繁忙，LLQ 中的流量是能够优先传送的，但是这些流量的带宽却不能超过所分配的带宽，如果超过了，也是没关系的，这些超过的流量只有在拥塞时才会被丢弃。

要将特定的流量分配到 LLQ 中，从而划分绝对保证的带宽，是通过 MQC 的方式来配置的，并且 LLQ 可以结合 CBWFQ，也就是说从接口全部可用带宽中，划出一部分给 LLQ 之后，其它带宽还可以分配给 CBWFQ 中各类数据流。在配置 LLQ 时，可以像 CBWFQ 一样使用具体数字，也可以使用百分比。需要注意，当从接口全部可用带宽中划走一分部给 LLQ 之后，剩下的带宽称为保留带宽（remaining bandwidth），可以将保留带宽以百分比的形式分配给 CBWFQ 中各类数据流。

配置 LLQ



说明:配置 R2，当接口 F0/0 的总带宽为 100 Mbit，且最大可用带宽为 80%，即 80 Mbit 时，使用 LLQ 保证源地址为 10.1.1.0/24 的流量的带宽为 30 Mbit，将全部可用带宽 80 Mbit 中剩下的 50 Mbit，拿出 50%，即 25 Mbit 给源地址为 20.1.1.0/24 的流量。

1. 配置 ACL 匹配相应流量

（1）通过 ACL 匹配源地址为 10.1.1.0/24 的流量

r2(config)#access-list 10 permit 10.1.1.0 0.0.0.255

（2）通过 ACL 匹配源地址为 20.1.1.0/24 的流量

r2(config)#access-list 20 permit 20.1.1.0 0.0.0.255

2. 使用 class-map对流量分类

（1）通过 class-map 匹配源地址为 10.1.1.0/24的流量

r2(config)#class-map net10

r2(config-cmap)#match access-group 10

（2）通过 class-map 匹配源地址为 20.1.1.0/24的流量

r2(config)#class-map net20

r2(config-cmap)#match access-group 20

3. 分配带宽

（1）使用 LLQ 分配 30 Mbit 给源地址为 10.1.1.0/24 的流量

r2(config)#policy-map band

r2(config-pmap)#class net10

r2(config-pmap-c)#priority percent 30

说明:命令 priority 为 LLQ 中的流量分配带宽，percent 后的百分比为接口总带宽的百分比，因为接口总带宽为 100 Mbit，所以 percent 30 就是 30 Mbit。

（2）使用 CBWFQ 将剩余可用带宽的 50%分配给源地址为 20.1.1.0/24 的流量

r2(config)#policy-map band

r2(config-pmap)#class net20

r2(config-pmap-c)#bandwidth remaining percent 50

说明:因为 LLQ 中的流量使用了 30 Mbit，全部可用带宽为 80 Mbit，所以剩余带宽为 50 Mbit，而 remaining percent 则是从分配给 LLQ 后剩余带宽 50 Mbit 中的 50%，即为 25 Mbit。

4. 应用队列到接口

（1）将接口的全部可用带宽改为 80Mbit

r2(config)#int f0/0

r2(config-if)#max-reserved-bandwidth 80

说明:将接口的全部可用带宽从默认的 75%更改到 80%,即 80 Mbit。

（2）应用队列到接口 F0/0

r2(config)#int f0/0

r2(config-if)#service-policy output band

说明:因为 LLQ 结合了 CBWFQ，只要有 CBWFQ，方向只能为 out。

5. 查看队列

（1）查看接口 F0/0 的队列机制

```
r2#sh int f0/0
FastEthernet0/0 is administratively down, line protocol is down
  Hardware is Gt96k FE, address is c000.0e48.0000 (bia c000.0e48.0000)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: Class-based queueing
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 1/1 (allocated/max allocated)
     Available Bandwidth 5000 kilobits/sec
```

说明:看到接口现有队列为 CBWFQ，并且是结合了 LLQ 的。

IP RTP （Real-Time Transport Protocol）

在使用 LLQ 时，可以做到为延迟和抖动较敏感的语音或视频数据绝对保证带宽，但是 LLQ 可以为任何数据流服务，不限制于任何协议。而 IP RTP 尽量只为对延迟要求较高的实时数据提供带宽保证，比如尽量只为语音提供带宽保证。受 RTP 保护的数据流，可以在任何流量之前优先传递，即使是 LLQ 和 RTP 同时出现的情况下，RTP 的流量是优先于 LLQ 传送的。并不是所有的流量都能受到 RTP 的保护，只有 UDP 目标端口号为 16384 至 32767 的数据才能得到保护，并且可以随意定义端口号范围，所以当为语音数据流保证带宽时，并不需要知道准确的语音端口号信息。

当 RTP 在为语音数据流提供带宽保证时，RTP 是不知道有多少条语音会话的，也就是说，所有的语音会话，是共享整个带宽的，比如划分了 100 Kbit 给语音会话，如果此时有 4 条语音会话，那么就是 100 Kbit / 4=25 Kbit，结果就是每个语音会话的带宽为 25 Kbit，而不是 100 Kbit。

RTP 可以和 WFQ 或 CBWFQ 配置于同一个出接口，但 RTP 只支持 Serial interfaces 和 Frame Relay PVC，如果配置于接口下，那么是在整个接口生效，如果配置于 PVC 下，则只在单独的 PVC 下生效，但是如果只需要配置于某条 PVC，需要在工具 map-class 下配置，并将 map-class 应用于 PVC 下。

注：当接口下同时出现 LLQ 和 RTP 时，RTP 的参数优先，例如 LLQ 和 RTP 为同一类数据流带宽保证时，LLQ 的带宽为 80 Kbit，RTP 的带宽为 60 Kbit，那么该流量的带宽则为 60 Kbit。

配置 RTP

1. 在接口下配置 RTP

（1）在路由器 Serial 1/0 接口下配置 RTP，为 UDP 端口号范围为 16384 到 16888 的流量保证带宽 200 Kbit。

Router(config)#int s1/0

Router(config-if)#ip rtp priority 16888 16383 200

说明:如果需要为 VoIP 保证带宽，请将端口号分别设置为 16384 16383。

2. 在 Frame Relay接口下配置 RTP

（1）在 map-class 下配置 RTP

Router(config)#map-class frame-relay VOIP

Router(config-map-class)#frame-relay cir 100000

Router(config-map-class)#frame-relay bc 1000

Router(config-map-class)#frame-relay be 0

Router(config-map-class)#frame-relay fragment 64

Router(config-map-class)#frame-relay ip rtp priority 16384 16383 100

说明:为 UDP 端口号范围为 16384 到 16383 的流量保证带宽 100 Kbit。配置 RTP 时，先配置 FRTS 和 FRF.12。

（2）将 map-class 应用于 Frame Relay 接口的 PVC 100 下

Router(config)#int s1/0

Router(config-if)#encapsulation frame-relay

Router(config-if)#frame-relay traffic-shaping

Router(config-if)#frame-relay interface-dlci 100

Router(config-fr-dlci)#class VOIP

拥塞避免（Congestion avoidance）

当网络发生拥塞之后，总是有数据是要被丢弃的，在默认情况下，拥塞之后，接口总是先丢弃最后到达的数据包，而将之前已经到达的先转发，这样一来，当网络中有多种流量出现时，就难免会将重要的流量丢弃，或者尽量让某些程序降级带宽而避免拥塞的发生，这些，都是拥塞避免（Congestion avoidance）需要做的事情。拥塞避免需要靠以下技术来完成：

Tail Drop

Weighted Random Early Detection (WRED)

WRED–Explicit Congestion Notification

Frame Relay Discard Dligible (DE)

Tail Drop

尾丢弃（Tail Drop）是接口的默认行为，当接口发生拥塞后，总是将最后到达的数据包丢弃，直到没有拥塞为止，因为最后到的数据就是引起网络拥塞的主要原因。在使用尾丢弃的情况下，是无法保证重要数据流优先传递的，所以尾丢弃不建议使用，但也无法配置。

Weighted Random Early Detection (WRED)

WRED 是基于 weight 的随机早侦测，工作思想和 WFQ 有相同之处，因为 WFQ 在工作时，是依靠流量的优先级来分配相应带宽的，而 WRED 却是依靠流量的优先级来分配相应的丢弃几率的。当网络中有多种数据时，在发生拥塞之后，人们总是希望先将优先级较低的相对不重要的数据丢弃，而优先保证重要数据的传递。WRED 正是迎合了人们的这种期望，在网络发生拥塞之后，总是先保证高优先级的重要数据的传递，而先丢弃普通的数据。

WRED 在网络发生拥塞之后，可根据数据包的 DSCP 或 IP 优先级来丢弃数据包，低优先级的数据总是比高优先级的数据先丢，从而保证重要数据的传递。在默认情况下，是根据数据包的 IP 优先级来决定如何丢弃的。

虽然说 WRED 是丢弃低优先级的数据包而保证高优先级的数据包，但是网络拥塞时，并不是总是先丢低优先级的，这是需要靠公式来计算的。思想为根据各优先级或 DSCP 设置的阀值，如果某优先级或 DSCP 的流量总是触及设定的阀值，那么该流量被丢弃的概率也就越大，所以如果低优先级不经常触及设置的阀值时，也有不被丢的可能。

在数据包被丢弃之后，如果是 TCP 流量，便可以调整窗口大小，从而降低速度，但是除 TCP 之外的其它流量便无能为力了。同时，也只有 TCP 才能够对丢弃的数据包进行重传，所以在使用 WRED 时，需要考虑这些问题。

WRED 在应用时，只能应用于接口下，或者和 WFQ 与 CBWFQ 一起使用，之所以不能和 PQ 一样的队列同时使用，是因为 PQ 或 LLQ 都有自己的保护和丢弃机制，WRED 对数据的操作没有太多意义。

配置 WRED

1. 在接口下配置 WRED

说明:可以选择开启基于 IP 优先级的 WRED 或基于 DSCP 的 WRED，默认基于 IP 优先级。

（1）在接口 F0/0 下开启基于 IP 优先级的 WRED

```
Router(config)#int f0/0

Router(config-if)#random-detect
```

说明:默认 WRED 基于 IP 优先级

（2）在接口 F0/1 下开启基于 DSCP 的 WRED

```
Router(config)#int f0/1

Router(config-if)#random-detect dscp-based
```

说明:指定 WRED 基于 DSCP 工作。

（3）查看 WRED

Router#sh queueing random-detect

Current random-detect configuration:

FastEthernet0/0

Queueing strategy: random early detection (WRED)

Random-detect not active on the dialer

Exp-weight-constant: 9 (1/512)

Mean queue depth: 0

class	Random drop	Tail drop	Minimum	Maximum	Mark
	pkts/bytes	pkts/bytes	thresh	thresh	prob
0	0/0	0/0	20	40	1/10
1	0/0	0/0	22	40	1/10
2	0/0	0/0	24	40	1/10
3	0/0	0/0	26	40	1/10
4	0/0	0/0	28	40	1/10
5	0/0	0/0	31	40	1/10
6	0/0	0/0	33	40	1/10
7	0/0	0/0	35	40	1/10
rsvp	0/0	0/0	37	40	1/10

FastEthernet0/1

Queueing strategy: random early detection (WRED)

Random-detect not active on the dialer

Exp-weight-constant: 9 (1/512)

Mean queue depth: 0

dscp	Random drop	Tail drop	Minimum thresh	Maximum thresh	Mark prob
	pkts/bytes	pkts/bytes			
af11	0/0	0/0	33	40	1/10
af12	0/0	0/0	28	40	1/10
af13	0/0	0/0	24	40	1/10
af21	0/0	0/0	33	40	1/10
af22	0/0	0/0	28	40	1/10
af23	0/0	0/0	24	40	1/10
af31	0/0	0/0	33	40	1/10
af32	0/0	0/0	28	40	1/10
af33	0/0	0/0	24	40	1/10
af41	0/0	0/0	33	40	1/10
af42	0/0	0/0	28	40	1/10
af43	0/0	0/0	24	40	1/10
cs1	0/0	0/0	22	40	1/10
cs2	0/0	0/0	24	40	1/10

cs3	0/0	0/0	26	40	1/10
cs4	0/0	0/0	28	40	1/10
cs5	0/0	0/0	31	40	1/10
cs6	0/0	0/0	33	40	1/10
cs7	0/0	0/0	35	40	1/10
ef	0/0	0/0	37	40	1/10
rsvp	0/0	0/0	37	40	1/10
default	0/0	0/0	20	40	1/10

Router#

2. 配置 CBWFQ 下的 WRED

（1）配置 WRED 在 CBWFQ 对所有流量生效

Router(config)#policy-map WWW

Router(config-pmap)#class class-default

Router(config-pmap-c)#bandwidth 1000000

Router(config-pmap-c)#random-detect

说明:对所有流量通过命令 bandwidth 配置了 CBWFQ，并通过命令 random-detect 开启了基于 IP 优先级的 WRED。

WRED—Explicit Congestion Notification

在使用 WRED 对过多的流量进行丢弃时，有时会造成不必要的麻烦。比如在使用非 TCP 进行通信时，被丢弃的数据包无法得到重传，然而即使是 TCP，虽然将数据丢弃之后，能够遏制高速流量，但当拥塞消失后，必定还会引起下一次拥塞，所以 WRED 只能治标而不能治本。

引起网络拥塞的原因就是数据源发送了过量的数据包，而 ECN（明确拥塞通告）就是通过发送警告，让数据源知道网络已经发生拥塞，从而可以降低自己发送数据包的速度。但是如果数据源不支持 ECN，那么所发送的流量照丢不误。

配置 ECN 时，必须开启 WRED，并且需要配合 WFQ 或 CBWFQ 使用。

在默认情况下，整形后的帧中继接口在收到 ECN 消息后，会将速度降到原有速度的一半，但是之后的速度也不会低于一半，只会在原有速度和一半之间浮动。这些参数可以调整。

配置 ECN

1. 在 CBWFQ 下配置 ECN

（1）在 CBWFQ 下为所有流量配置 ECN

```
Router(config)#policy-map EEE

Router(config-pmap)#class class-default

Router(config-pmap-c)#bandwidth percent 75

Router(config-pmap-c)#random-detect

Router(config-pmap-c)#random-detect ecn
```

说明:在 CBWFQ 下开启 ECN。

2. 查看配置

（1）查看开启 ECN 的 CBWFQ

```
Router#sh policy-map
  Policy Map EEE
    Class class-default
      Bandwidth 75 (%)
        exponential weight 9
        explicit congestion notification
        class    min-threshold    max-threshold    mark-probablity
        ------------------------------------------------

        0        -                -                1/10
        1        -                -                1/10
        2        -                -                1/10
        3        -                -                1/10
        4        -                -                1/10
        5        -                -                1/10
        6        -                -                1/10
        7        -                -                1/10
        rsvp     -                -                1/10
```

3. 更改 Frame Relay 下的 ECN 参数

（1）在收到 ECN 后，带宽的最低限制

Router(config)#map-class frame-relay FFF

Router(config-map-class)#frame-relay cir 100000

Router(config-map-class)#frame-relay bc 1000

Router(config-map-class)#frame-relay be 0

Router(config-map-class)#frame-relay mincir 6000

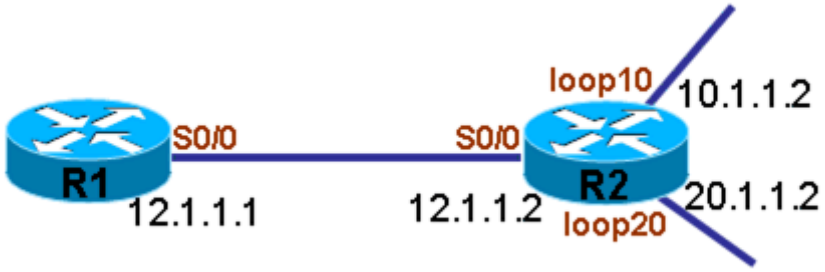
说明:在收到 ECN 后，带宽不会降到 6000 Bit 以下。

Frame Relay Discard Dligible (DE)

在 Frame Relay 网络中，数据包中标有 Discard Dligible (DE)字段，该字段告诉设备数据包的重要性，如果为 1，表示该数据包并不重要，在网络发生拥塞时可以优先被丢弃，如果为 0，则表示在将为 1 的数据包全部丢光的情况下，才可被丢弃，所有数据包的 DE 字段默认为 0。

可以将特定数据包的 DE 字段设置为 1。

配置 Frame Relay DE



说明:配置 R1，将去往 10.1.1.0/24 的数据包 DE 字段标为 1，在接口拥塞时，可以先丢弃。

1. 配置 ACL 匹配相应流量

（1）通过 ACL 匹配去往 10.1.1.0/24 的流量

r1(config)#access-list 100 permit ip any 10.1.1.0 0.0.0.255

2. 将指定流量的 DE 设置为 1

(1)设置 ACL 100 中的流是的 DE 为 1

r1(config)#frame-relay de-list 1 protocol ip list 100

说明:所有被 de-list 匹配到的流量的 DE 都会被设置为 1。

3. 应用策略到 PVC 下

（1）将 de-list 应用到 PVC 下

r1(config)#int s0/0

r1(config-if)#frame-relay de-group 1 102

4. 测试结果

（1）在 R1 上向 10.1.1.0/24 发送流量

r1#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/57 ms

r1#

（2）查看 PVC 下的结果

r1#sh fram pvc 102

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

input pkts 40 output pkts 40 in bytes 4160

out bytes 4160 dropped pkts 0 in pkts dropped 0

out pkts dropped 0 out bytes dropped 0

in FECN pkts 0 in BECN pkts 0 out FECN pkts 0

out BECN pkts 0 in DE pkts 0 out DE pkts 5

out bcast pkts 0 out bcast bytes 0

5 minute input rate 0 bits/sec, 1 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 00:05:34, last time pvc status changed 00:04:54

r1#

说明:可以看到 PVC 下匹配到的 DE 为 1 的数据包数量，当接口拥塞时，此类数据包将优先被丢弃。

链路优化（Link Efficiency Mechanisms）

在使用 QOS 保证重要数据的优先传递时，除了可以使用队列技术之外，还有就是在链路上做一些额外的优化，以获得更高的性能和更明显的效果。链路优化的技术有如下三种：

Link Efficiency Mechanisms

链路层技术，

有：

Multilink PPP (MLP)

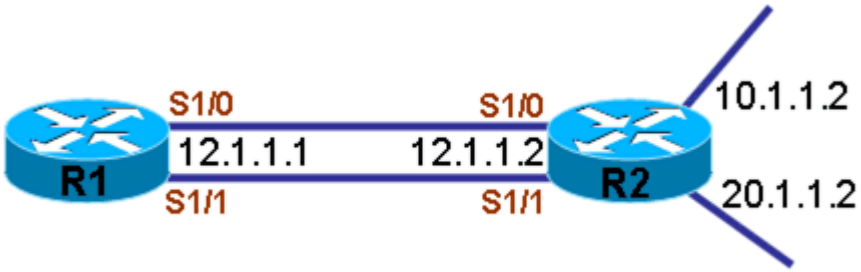
Frame Relay Fragmentation

Header Compression

CCIE 考试需要了解以上三种链路优化技术，下面分别来详细介绍。

Multilink PPP (MLP)

对于低于或等于 768kbps 的链路，由于带宽很低，延迟太大，将对语音或视频的通信带来麻烦。而 Multilink PPP (MLP)技术为了能够提高链路带宽，允许同时将多根低速链路捆绑成逻辑上单一的链路，称为 bundle，从而将数据流分散在多条链路上同时传输，以提高带宽，降低延迟，如下图所示：



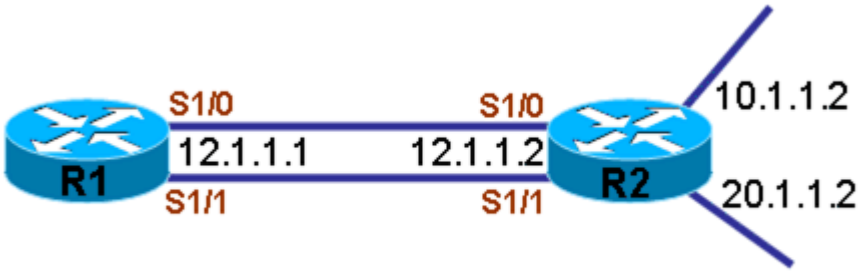
上图中，R1 和 R2 之间的两条低速串行链路便可捆绑成一条逻辑链路，在路由器之间传输数据流时，数据流将同时从两根链路上交叉传输，等于使用了两根链路，带宽得到了提高。将串行链路捆绑成 Multilink 时，需要 PPP 封装。

在将多条物理链路捆绑成一条逻辑链路后，还是可以实施各种 QOS 技术的，但是 QOS 必须应用到逻辑链路上，IP 地址也要配置到逻辑链路上。默认情况下，使

用 FIFO，但 LLQ，WFQ，CBWFQ 也可以使用，在使用其它 QOS 技术时，应该在配置 Multilink 之前就将 QOS 配置好。

在使用 Multilink 时，有个较为复杂的延迟计算，就是 fragment delay 单位为 ms，是用来控制多条链路交叉传输数据包的，这里不作出计算方式的详细介绍。

配置 Multilink PPP (MLP)



以上图为例，将 R1 与 R2 之间的两条链路 S1/0 和 S1/1 捆绑成一条逻辑链路，并在 R1 上实施 QOS，将 R1 到目标网络 10.1.1.0/24 的流量管制到 CIR 为 8000 bit，超过的流量则丢弃，R1 到目标网络 20.1.1.0/24 的流量正常通过。

1.配置 QOS

（1）通过 ACL 匹配到网络 10.1.1.0/24 的流量

r1(config)#access-list 100 permit ip any 10.1.1.0 0.0.0.255

（2）class-map 调用 ACL 中的流量

r1(config)#class-map net10

r1(config-cmap)#match access-group 100

（3）在 policy-map 中调用 class-map 的流量并管制带宽

r1(config)#policy-map band

r1(config-pmap)#class net10

r1(config-pmap-c)# police cir 8000 bc 1000 be 1000 conform-action transmit exceed-action drop

说明:配置中 CIR 为 8000 bit，超过的流量为 drop。

2.创建 Multilink

(1)在 R1 上创建 Multilink

r1(config)#int multilink 1

r1(config-if)#ip address 12.1.1.1 255.255.255.0

r1(config-if)#service-policy output band

r1(config-if)#ppp multilink fragment delay 10

说明:R1 上的 Multilink 号码为 1，并配置 IP 地址，设置数据包交叉延迟为 10ms。

3.将串口划入 Multilink

(1)将接口 S1/0 划入 Multilink

r1(config)#int s1/0

r1(config-if)#no ip address

r1(config-if)#encapsulation ppp

r1(config-if)#ppp multilink

r1(config-if)#ppp multilink group 1

r1(config-if)#no shutdown

(2)将接口 S1/1 划入 Multilink

r1(config)#int s1/1

r1(config-if)#no ip address

r1(config-if)#encapsulation ppp

r1(config-if)#ppp multilink

r1(config-if)#ppp multilink group 1

r1(config-if)#no shutdown

说明:R1 上的两条链路 S1/0 和 S1/1 封装成 PPP 后，并捆绑成单一逻辑链路。

4.配置 R2

说明:R2 配置与 R1 类似。

5.查看结果

（1）查看单条物理链路带宽

r1#show interfaces s1/0

Serial1/0 is up, line protocol is up

Hardware is M4T

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation PPP, LCP Open, multilink Open

Link is a member of Multilink bundle Multilink1, crc 16, loopback not

Set

说明:可以看到单条链路的带宽为 1544 Kbit，并且已经工作在 multilink

(2)查看 multilink

r1#sh ppp multilink

Multilink1, bundle name is r2

Endpoint discriminator is r2

Bundle up for 00:03:45, total bandwidth 3088, load 1/255

Receive buffer limit 24000 bytes, frag timeout 1000 ms

0/0 fragments/bytes in reassembly list

0 lost fragments, 0 reordered

0/0 discarded fragments/bytes, 0 lost received

0x1E received sequence, 0x1D sent sequence

Member links: 2 active, 0 inactive (max not set, min not set)

Se1/0, since 00:03:45, 1930 weight, 1496 frag size

Se1/1, since 00:03:30, 1930 weight, 1496 frag size

No inactive multilink interfaces

r1#

说明:可以看到 multilink 中拥有两条链路，因为单一链路带宽为 1544 Kbit，所以 multilink 的带宽为 3088。

（3）查看 multilink IP 信息

r1#sh interfaces multilink 1

Multilink1 is up, line protocol is up

Hardware is multilink group interface

Internet address is 12.1.1.1/24

MTU 1500 bytes, BW 3088 Kbit, DLY 100000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation PPP, LCP Open, multilink Open

Open: IPCP, CDPCP, loopback not set

Keepalive set (10 sec)

DTR is pulsed for 2 seconds on reset

说明:可以看到 multilink 拥有配置的 IP 地址。

6.测试 QOS

（1）测试 R1 到目标网络 20.1.1.0/24 的流量

r1#ping 20.1.1.2 size 800 repeat 10

Type escape sequence to abort.

Sending 10, 800-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

!!!!!!!!!!

Success rate is 100 percent (10/10), round-trip min/avg/max = 16/72/236

ms

r1#

说明:因为并没有对 R1 到目标网络 20.1.1.0/24 的流量进行管制，所以当数据包每个以 800 字节通过时，一切正常，并且速度正常。

（2）测试 R1 到目标网络 10.1.1.0/24 的流量

r1#ping 10.1.1.2 size 800 repeat 10

Type escape sequence to abort.

鸿鹄培训+考试群：301283867

Sending 10, 800-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!.!.!.!.!.

Success rate is 50 percent (5/10), round-trip min/avg/max = 156/188/252

ms

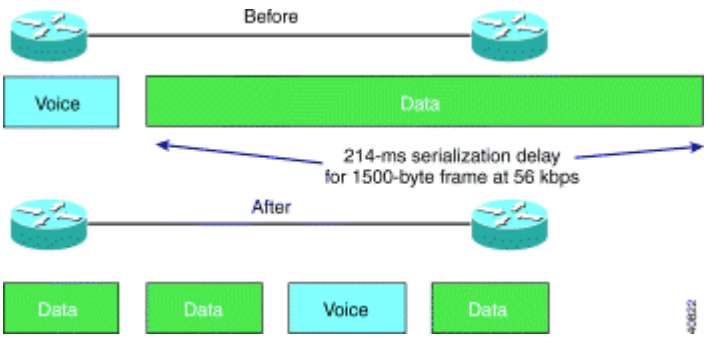
r1#

说明:因为 R1 到目标网络 10.1.1.0/24 的流量被管制为 8000bit 每秒，所以当数据包每个以 800 字节通过时，流量超出额定带宽，从而出现了超额流量被均衡地丢弃，multilink 上的 QOS 已生效。

Frame Relay Fragmentation

当不能接受延迟的敏感数据，如语音在低速的链路上传递时，可能会受到影响。一般情况下，语音需要低于 150 milliseconds (ms)的延迟，而语音在一个接口发送时建议低于 20ms 的延迟，但是当设备在将数据包放入接口进行处理时，这本身就需要花费一定的时间，如果数据包过大，那么将整个数据包放入接口处理可能就要花费过多的时间。

一个大小为 1500-byte 的数据包在进入一个速度为 64 kbps 的接口需要花的时间为 187 ms，很显然这会影响到语音的质量。如果接口上同时有普通数据和语音同时传输时，语音受到的影响则更为严重。因此，如果能够将数据包分割地更小来进行传递，这样就可以减少语音的抖动，如下图：



在帧中继接口上将数据包分割成更小的包以减少延迟，有以下两种技术：

FRF.11 Fragmentation

FRF.12 Fragmentation

FRF.11 已经不建议使用，FRF.12 Fragmentation 将语音和实时数据包分割的更小。FRF.12 在接口上将大的数据包分割成固定大小的更小的数据包再传递，在 FR 接口上，只有数据包大于配置好的大小，才会被分割。

在接口速率超过 768 kbps 时可能不需要 Fragmentation，但是肯定需要 RTP 或 LLQ 来保证语音流量的传递。不同速率的接口，都有一个建议的被分割的值：

Lowest Link Speed in Path	Recommended Fragmentation Size (for 10 ms Serialization)
56 Kbps	70 bytes
64 Kbps	80 bytes
128 Kbps	160 bytes
256 Kbps	320 bytes
512 Kbps	640 bytes
768 Kbps	1000 bytes
1536 Kbps	1600 bytes

配置 FRF.12 Fragmentation

说明:建议对联的两台路由器都配，且参数一致。

1. 在路由器上配置 FRF.12 Fragmentation

（1）在 map-class 下配置 FRF.12 Fragmentation

Router(config)#map-class frame-relay FFF

Router(config-map-class)#frame-relay fragment 640

说明:FRF.12 将数据包分割到 640 byte 每个包，默认不跟参数，大小为 53 byte。

(2)将 FRF.12 Fragmentation 用于帧中继接口下

Router(config)#int s1/0

Router(config-if)#frame-relay traffic-shaping

Router(config-if)#frame-relay class FFF

说明:map-class 用在哪，FRF.12 就在哪生效；traffic-shaping 必开，如果不开，FRF.12 不工作。

2. 查看 FRF.12

（1）查看接口下的 FRF.12

r1#sh frame-relay fragment

interface	dlci	frag-type	frag-size	in-frag	out-frag	dropped-frag
Serial0/1	112	end-to-end	80	0	0	0
Serial0/1	113	end-to-end	80	0	0	0
Serial0/1	114	end-to-end	80	0	0	0
Serial0/1	115	end-to-end	80	0	0	0
Serial0/1	116	end-to-end	80	0	0	0

r1#

说明:可以看到接口下每条 PVC 的 FRF.12 详细信息。

Header Compression

正常情况下，数据包除了真正的数据之外，还有一部分就是包头信息，为了能够更有效地利用带宽，可以在数据包被发送之前，将 IP 的包头压缩，以缩小多余的头部信息，从而传输更多的数据内容。

将 IP 数据包的包头压缩之后，可以减少过载，加快传输，头部压缩只能对 RTP 和 TCP 数据进行压缩。

配置数据包头部压缩，可以在接口下直接配置，也可能通过 MQC 基于 class 配置。在对数据进行压缩时，只支持 Frame Relay, HDLC 和 PPP 接口，ISDN 也可以。但在 Frame Relay 接口使用 IETF 封装时，不能对 RTP 进行压缩；压缩需要在对联的两台设备之间同时配置。

配置 Header Compression

1．HDLC 接口下配置 RTP 头部压缩

（1）在 HDLC 接口下配置 RTP 头部压缩

```
r1(config)#int s0/0
r1(config-if)#ip rtp header-compression
```

（2）查看压缩

```
r1#sh ip rtp header-compression
RTP/UDP/IP header compression statistics:
  Interface Serial0/0:
    Rcvd:    0 total, 0 compressed, 0 errors, 0 status msgs
         0 dropped, 0 buffer copies, 0 buffer failures
    Sent:    0 total, 0 compressed, 0 status msgs
         0 bytes saved, 0 bytes sent
    Connect: 16 rx slots, 16 tx slots,
         0 long searches, 0 misses 0 collisions, 0 negative cache hits
```

r1#

2. frame-relay接口下配置 RTP 头部压缩

（1）在 frame-relay 接口下配置 RTP 头部压缩

r1(config)#int s0/0

r1(config-if)#ip rtp header-compression

（2）查看压缩

r1(config)#int s0/0

r1(config-if)#encapsulation frame-relay

r1(config-if)#frame-relay ip rtp header-compression

3 HDLC 接口下配置 TCP 头部压缩

（1）在 HDLC 接口下配置 TCP 头部压缩

r1(config)#int s0/0

r1(config-if)#ip tcp header-compression

4. 配置 class-based 的头部压缩

（1）配置基于 class-based 的 RTP 头部压缩

r1(config)#policy-map COM

r1(config-pmap-c)#compression header ip rtp

说明:基于 class-based 的 TCP 头部压缩配置命令相似，不再重复。

（2）将策略应用于接口

r1(config)#int s0/0

r1(config-if)#service-policy output COM

说明:基于 class-based 的头部压缩只能用于接口 out 方向

返回目录

AutoQoS — VoIP

概述

在手工配置 QOS 为特定的流量提供带宽保证时，需要匹配特定的流量，再根据这些流量做相应的策略。这样的方式比较烦琐，并且对于不了解 QOS 的人来说，要为重要的流量，如语音流量提供带宽保证，可能容易出现人工配置错误。

对于一般流量的 QOS，可能由于需求各不相同，没有一个特定的规范，所以还是需要人工来配置。但是对于如语音或视频这样的特殊流量，因为它们有着固定的 QOS 需求，无论谁来配，只要满足这些要求即可，因此，思科在 IOS 中集成了固定的 QOS，为语音和视频自动配置符合语音的视频特定要求的 QOS，这就是 AutoQoS – VoIP。

在实施 AutoQoS – VoIP 时，有许多要注意的地方：

接口上不能事先配置 QOS，比如配置 service policies

Simple Network Protocol (SNMP) traps（会自动打开）

支持的接口有 Serial interfaces(如封装了 PPP，HDLC，以及 Frame Relay 的接口)，高版本的 IOS 也支持其它类型接口。

Frame Relay 接口在映射了 DLCI 时是不能配置的，且只支持 Frame Relay point-to-point 子接口。

对联的两台设备的接口都应该配置 AutoQoS – VoIP，并且速率应该相同，Frame

Relay 接口的 FRF.12 也要相同。

在配置 AutoQoS – VoIP 后，就不能再更改接口带宽了，即使改了，也还是原来的带宽，如果之前更改带宽，也不能随意更改。

在开启 AutoQoS – VoIP 后，会自动为 VoIP 流量标记，而且会使用 LLQ，PQ, RTP 也会实施。AutoQoS – VoIP 会自己在接口上为 VoIP 流量创建相应的 policy maps, class Maps 以及 ACLs。

注：带宽小于或等于 768 kpbs 的被认为是 low-speed links，大于 768 kpbs 的被认为是 high-speed links。

在普通网络实施 AutoQoS – VoIP，只需要一条命令 auto qos voip 即可。

在企业网络里实施 AutoQoS – VoIP 时，需要对流量有个发现和收集的过程，称为 Auto-Discovery，是使用 NBAR 来发现语音流量的。

注：在配置了 AutoQoS – VoIP 之后，若是关掉，那么之前的配置在某些 IOS 下还会保留，需要手工清除配置。

配置 AutoQoS — VoIP

1. 开启 HDLC 接口下的 AutoQoS — VoIP

（1）在 HDLC 接口下开启 AutoQoS– VoIP

```
Router(config)#int s0/0
Router(config-if)#encapsulation hdlc
Router(config-if)#bandwidth 1000
Router(config-if)#auto qos voip
```

说明:将带宽改为 1000Kbit，并开启 AutoQoS – VoIP，改带宽须在开启 AutoQoS –

VoIP 之前改好。

（2）查看 AutoQoS – VoIP

```
Router#show auto qos interface


Serial0/0 -
!
 interface Serial0/0
  service-policy output AutoQoS-Policy-UnTrust
Router#show auto qos
!
 policy-map AutoQoS-Policy-UnTrust
  class AutoQoS-VoIP-RTP-UnTrust
   priority percent 70
   set dscp ef
  class AutoQoS-VoIP-Control-UnTrust
   bandwidth percent 5
   set dscp af31
  class AutoQoS-VoIP-Remark
   set dscp default
  class class-default
   fair-queue
!
```

```
class-map match-any AutoQoS-VoIP-Remark
  match ip dscp ef
  match ip dscp cs3
  match ip dscp af31
!
class-map match-any AutoQoS-VoIP-Control-UnTrust
  match access-group name AutoQoS-VoIP-Control
!
class-map match-any AutoQoS-VoIP-RTP-UnTrust
  match protocol rtp audio
  match access-group name AutoQoS-VoIP-RTCP
!
ip access-list extended AutoQoS-VoIP-RTCP
  permit udp any any range 16384 32767
!
ip access-list extended AutoQoS-VoIP-Control
  permit tcp any any eq 1720
  permit tcp any any range 11000 11999
  permit udp any any eq 2427
  permit tcp any any eq 2428
  permit tcp any any range 2000 2002
  permit udp any any eq 1719
```

permit udp any any eq 5060

!

rmon event 33333 log trap AutoQoS description "AutoQoS SNMP traps for Voice Drops" owner AutoQoS

rmon alarm 33334 cbQosCMDropBitRate.1249.1251 30 absolute rising-threshold 1 33333 falling-threshold 0 owner AutoQoS

Serial0/0 -

!

interface Serial0/0

service-policy output AutoQoS-Policy-UnTrust

Router#

说明:可以看到接口 S0/0 已开启的 AutoQoS – VoIP，并且详细参数已生效。

2. 开启 Frame Relay 接口下的 AutoQoS — VoIP

(1)在 Frame Relay 子接口下开启 AutoQoS – VoIP

Router(config)#int s0/0

Router(config-if)#encapsulation frame-relay

Router(config-if)#exit

Router(config)#int s0/0.1 point-to-point

Router(config-subif)#frame-relay interface-dlci 102

Router(config-fr-dlci)#auto qos voip

说明: Frame Relay 只支持 point-to-point 子接口，并且只能在 DLCI 下开启。

（2）查看 AutoQoS – VoIP

```
Router#show auto qos
!
policy-map AutoQoS-Policy-UnTrust
 class AutoQoS-VoIP-RTP-UnTrust
  priority percent 70
  set dscp ef
 class AutoQoS-VoIP-Control-UnTrust
  bandwidth percent 5
  set dscp af31
 class AutoQoS-VoIP-Remark
  set dscp default
 class class-default
  fair-queue
!
class-map match-any AutoQoS-VoIP-Remark
 match ip dscp ef
 match ip dscp cs3
 match ip dscp af31
!
```

```
class-map match-any AutoQoS-VoIP-Control-UnTrust
 match access-group name AutoQoS-VoIP-Control
!
class-map match-any AutoQoS-VoIP-RTP-UnTrust
 match protocol rtp audio
 match access-group name AutoQoS-VoIP-RTCP
!
ip access-list extended AutoQoS-VoIP-RTCP
 permit udp any any range 16384 32767
!
ip access-list extended AutoQoS-VoIP-Control
 permit tcp any any eq 1720
 permit tcp any any range 11000 11999
 permit udp any any eq 2427
 permit tcp any any eq 2428
 permit tcp any any range 2000 2002
 permit udp any any eq 1719
 permit udp any any eq 5060
!
rmon event 33333 log trap AutoQoS description "AutoQoS SNMP traps for Voice
Drops" owner AutoQoS
    rmon    alarm    33336    cbQosCMDropBitRate.1399.1401    30    absolute
```

rising-threshold 1 33333 falling-threshold 0 owner AutoQoS

Serial0/0.1: DLCI 102 -

!

interface Serial0/0

frame-relay traffic-shaping

!

interface Serial0/0.1 point-to-point

frame-relay interface-dlci 102

class AutoQoS-FR-Se0/0-102

!

map-class frame-relay AutoQoS-FR-Se0/0-102

frame-relay cir 1544000

frame-relay bc 15440

frame-relay be 0

frame-relay mincir 1544000

service-policy output AutoQoS-Policy-UnTrust

Router#

说明:可以看到接口 S0/0.1 已开启的 AutoQoS – VoIP，并且详细参数已生效。

3. 配置企业网络的 AutoQoS — VoIP

（1）在接口 S0/0 下配置 Auto-Discovery

Router(config)#int s0/0

Router(config-if)#auto discovery qos

（2）在接口 S0/0 下配置 AutoQoS

Router(config)#int s0/0

Router(config-if)#auto qos voip

(3)查看 Auto-Discovery

Router#sh auto discovery qos

Serial0/0

AutoQoS Discovery enabled for applications

Discovery up time: 1 minutes, 6 seconds

AutoQoS Class information:

No AutoQoS data discovered

Router#

说明:因为没有语音流量，所有检测到的数据为空。

(4)查看 AutoQoS－VoIP

Router#show auto qos

!

policy-map AutoQoS-Policy-UnTrust

class AutoQoS-VoIP-RTP-UnTrust

priority percent 70

set dscp ef

class AutoQoS-VoIP-Control-UnTrust

```
 bandwidth percent 5
 set dscp af31
class AutoQoS-VoIP-Remark
 set dscp default
class class-default
 fair-queue
!
class-map match-any AutoQoS-VoIP-Remark
 match ip dscp ef
 match ip dscp cs3
 match ip dscp af31
!
class-map match-any AutoQoS-VoIP-Control-UnTrust
 match access-group name AutoQoS-VoIP-Control
!
class-map match-any AutoQoS-VoIP-RTP-UnTrust
 match protocol rtp audio
 match access-group name AutoQoS-VoIP-RTCP
!
ip access-list extended AutoQoS-VoIP-RTCP
 permit udp any any range 16384 32767
!
```

```
ip access-list extended AutoQoS-VoIP-Control
  permit tcp any any eq 1720
  permit tcp any any range 11000 11999
  permit udp any any eq 2427
  permit tcp any any eq 2428
  permit tcp any any range 2000 2002
  permit udp any any eq 1719
  permit udp any any eq 5060
!
rmon event 33333 log trap AutoQoS description "AutoQoS SNMP traps for Voice Drops" owner AutoQoS
    rmon alarm 33337 cbQosCMDropBitRate.1493.1495 30 absolute rising-threshold 1 33333 falling-threshold 0 owner AutoQoS

Serial0/0 -
!
interface Serial0/0
  service-policy output AutoQoS-Policy-UnTrust
Router#
```

说明:可以看到接口 S0/0 已开启的 AutoQoS – VoIP，并且详细参数已生效。

企业网络其它接口的 AutoQoS – VoIP 配置类似，不再举例。

RSVP

概述

Resource Reservation Protocol (RSVP) 被定义为 Signalling 技术，也就是 QOS 中的信号技术，RSVP 发出的信号，也就是一个主机或节点向网络中发出请求自己的数据流需要得到什么样的处理，也就是请求所需要的带宽。Signalling 是一个很有用的带宽请求技术，在端到端的 QOS 中起到非常重要的角色。

端到端的 QOS 要求网络路径中所有的设备（包括路由器和交换机，防火墙等等）都参与相同的 QOS 策略， RSVP 利用三层 IP 技术来穿透整个网络实施 Signalling，能实现这个目的，还有 IP 优先级技术，但是 IP 优先级用于区分服务，而 RSVP 一定会保证带宽。

RSVP 没有自己的路由协议，是根据当前已有的路由来决定路径的，如果路径改变了，那么 RSVP 请求也会重新计算，RSVP 并不能解决所有 QOS 问题，并且 RSVP 自己的申请过程也是要花时间的。

在应用程序申请到带宽后，RSVP 会做好记录的，发送的流量可以超过已申请到的带宽，但是只能保证在已申请到的带宽内，如果设置的空闲带宽足够够，就能提供超额带宽，如果不够了，也就会丢包。

RSVP with LLQ

RSVP 使用 WFQ 提供公平 QOS 服务，给别的流最低的优先级以保证其它流量，但 WFQ 不能保证语音的流量。

由于 LLQ 能为语音提供很好的带宽保证，所以要保证语音，RSVP 必须结合 LLQ，这样就可以将语音放入 LLQ 系统。

在 RSVP with LLQ 时，有以下一些限制：

不支持任何 tunnels

依靠 PQ，如果 LLQ 不支持，那么也不行

必须要支持 RSVP，WFQ 或 LLQ 要支持。

注：所有已经被 RSVP 匹配到的流量，是不会被其它技术所匹配的。RSVP 申请到的带宽，是单向的。

RSVP 在 Frame-relay 下时有以下限制：

不支持 GTS

不支持队列

非语音 RSVP 不支持

组播也不支持。

并且需要有：

RSVP

WFQ on the PVC

LLQ

Frame Relay Forum (FRF).12 on the interface

配置 RSVP 时，需要考虑：

一个用户流需要多少带宽，最多可用多少带宽，默认是可以用尽所有可用带宽

RSVP 所有会话可使用多少带宽，默认 75%，tunnel 是 100%

在 Frame-relay 下时，要为每个 DLCI 分配，命令 ip rsvp bandwidth 需要同时在主

接口和子接口下打开。

配置 RSVP

1. 开启 RSVP（接口下）

（1）在接口 F0/0 下开启 RSVP

router(config)#int f0/0

router(config-if)#ip rsvp bandwidth 1000 500

说明:定义 RSVP 可用的总带宽为 1000K，单个流可保护的带宽为 500K。默认 RSVP 可用总带宽为接口的 75%，并且单个流可全用光。

2. 配置 RSVP with LLQ

(1)配置 LLQ 可用的带宽参数

router(config)#ip rsvp pq-profile 200000 8000

说明:LLQ 流最大可用 200KB，Bc 为 8KB。

3. 开启 RSVP with WFQ(全局必配)

(1)在接口下开启 WFQ，并且指定 WFQ 可用的带宽比

router(config)#int f0/0

router(config-if)#fair-queue

router(config-if)#fair-queue 50

说明:WFQ 为必开，可用带宽比为 50%

4. 配置 RSVP path

（1）配置 RSVP path，也就是请求带宽

router(config)#ip rsvp sender 100.1.1.1 10.1.1.2 tcp 80 10000 12.1.1.2 f0/0 200 10

说明:配置为目标地址 100.1.1.1，源地址 10.1.1.2，且目标端口为 TCP 80，源端口为 10000，上一跳为 12.1.1.2，进口为 f0/0 的流量请求保留带宽 200Kbit，Bc 为 10Kbit。

5. 配置 RSVP 保留带宽

（1）配置 RSVP 为某流量保留带宽

router(config)#ip rsvp reservation 100.1.1.1 10.1.1.2 tcp 80 10000 50.1.1.1 f0/1 ff rate 200 10

说明:配置为目标地址 100.1.1.1，源地址 10.1.1.2，且目标端口为 TCP 80，源端口为 10000，下一跳为 12.1.1.2，出口为 f0/0 的流量保留单个流带宽 200Kbit，Bc 为 10Kbit。

6. 查看 RSVP

（1）查看 RSVP 配置情况

router#sh ip rsvp installed

RSVP: FastEthernet0/0 has no installed reservations

RSVP: FastEthernet0/1

BPS To From Protoc DPort Sport

200K 100.1.1.1 10.1.1.2 TCP 80 10000

router#

说明:可以看到 RSVP 保证带宽的情况。

7. 配置 Frame-relay下的 RSVP

(1)配置 map-class

router(config)#map-class frame-relay FFF

router(config-map-class)#frame-relay cir 1000000

router(config-map-class)#frame-relay bc 10000

router(config-map-class)#frame-relay mincir 500000

router(config-map-class)#frame-relay fragment 100

router(config-map-class)#frame-relay fair-queue

说明:配置 cir 为 1000Kbit，Bc 为 10Kbit,Mincir 为 500Kbit, FRF.12 为 100Byte，并开启 WFQ。

（2）配置主接口的 RSVP

router(config)#int s0/0

router(config-if)#encapsulation frame-relay

router(config-if)#frame-relay traffic-shaping

router(config-if)#ip rsvp bandwidth 1000 200

说明:在主接口上封装 frame-relay，并开启 FRTS，以及开启 RSVP。

（3）在子接口应用 RSVP

router(config)#int s0/0.1 multipoint

router(config-subif)#ip rsvp bandwidth 1000 200

router(config-subif)#frame-relay interface-dlci 102

router(config-fr-dlci)#class FFF

说明:在 PVC 102 上应用 RSVP，并且子接口同样需要配置 RSVP 参数。

交换机 QOS （Switching QOS）

概述

之前所介绍的 QOS，并不能完全在交换机中实施，在交换机中，有些与路由器不同方式的 QOS，并且不是所有型号的交换机都具有相同的实施方式，在此文档中，由于 CCIE R&S 指定考试型号为 3560，所以仅针对 3560 为基础进行介绍。

在交换机上，流量进入和出去，是需要分开考虑的，在整台交换机上，流量进入时，有两个队列，进入交换机的所有流量，都根据这两个队列来处理，然后再将流量发送到内部环，最终从交换机上的接口被发出。

在流量进入交换机时，应该被分配到哪一个队列，是靠 SRR 来分配的。两个进的队列中，其中有一个是 PQ，默认为队列 2，也就是此队列中的流量可以被优先处理。

不仅流量在进入交换机时，需要依靠队列来决定先后顺序，而且在离开交换机时，也需要靠队列来决定。而出去的队列，是基于每个交换机接口的，交换机上每一个接口在出去时都有 4 个队列，流量属于哪一个队列，依靠数据包的 CoS 值。在出口的 4 个队列中，也有一个队列是被优先处理的，默认为队列 1，也就是只有队列 1 的流量被转发完了，才能转发其它队列的流量。

由上可见，所有流量在进入接口时，总带宽是可能超过交换机内部环的，所以在出去时，很有可能会被出口队列所编排顺序。无论在流量进入交换机或离开交换机，都有可能超过交换机的负载，对于超载的流量，交换机是需要将其丢弃的，默认为尾丢弃（Tail Drop），而这里的尾丢弃不同于普通的尾丢弃，因为这里的尾丢弃是有一定算法的，被称为 Weighted Tail Drop (WTD)。

WTD 在丢弃算法中，控制了每个队列的最大容量和丢弃阀值，也就是说队列中的流量是否该被丢弃，完全根据队列的最大容量和丢弃阀值来决定的。

WTD 在每个队列中根据数据包的 CoS 值将其分配到不同的阀值，每个队列拥有 3 个阀值，范围为 1%到 100%，其中前面两个阀值是可以任意配置的，而第三个则不可以，因为第三个阀值固定为 100%。这样一来，当某个 CoS 值的流量超过相应队列总容量的相应阀值之后，也就意味着这该流是要被丢弃的。

SRR Shaping and Sharing

在出口上，4 个队列依靠 Shaping 和 Sharing 的方式来分配接口总带宽。要注意，进口只有 Sharing 的模式。

队列都有自己的 weight 值，当队列为 Sharing 模式时，就是所有 Sharing 模式的队列根据各自的 weight 值来分配相应的接口带宽，比如 4 个 Sharing 模式的队列 weight 值分别为 10,20,30,40，那么将所有 weight 值相加 10+20+30+40=100，weight 值为 10 的队列分到和带宽为 10/100，weight 值为 20 的队列分到和带宽为 20/100，weight 值为 30 的队列分到和带宽为 30/100，weight 值为 40 的队列分到和带宽为 40/100；当某个队列并没有流量传递时，那么这些空闲的带宽也同样按照各队列的 weight 值分配给每个队列使用。

而在队列为 Shaping 模式时，该队列依靠自己的 weight 值获得相应的带宽，如值为 3，则获得接口总带宽的 1/3，为 5 则获得接口总带宽的 1/5，并且该队列永远不能超过这个值，即使接口非常空闲也不能超过。无论是 Shaping 模式还是 Sharing 模式，分配到的带宽是保证的，但 Sharing 模式的队列在接口空闲时却可以完全利用。

当 Shaping 和 Sharing 同时存在于接口时，接口总带宽减去 Shaping 模式分配到的带宽后，剩余的带宽由 Sharing 模式分配。

前提配置

1. 开启 QOS

（1）全局开启多层交换 QOS 功能

switch(config)#mls qos

说明:配置任何 QOS 之前，必须开启 QOS 功能。

（2）查看多层交换 QOS 功能

switch#sh mls qos

QoS is enabled

QoS ip packet dscp rewrite is enabled

switch#

说明:QOS 功能已经开启。

2. 给接口配置 CoS

（1）信任数据原来的 COS 值（应在 trunk 上）：

```
switch(config)#int f0/1
switch(config-if)#mls qos trust cos
```

（2）查看接口 CoS 值

```
switch#sh mls qos interface f0/1
FastEthernet0/1
```

trust state: trust cos

trust mode: trust cos

trust enabled flag: ena

COS override: dis

default COS: 0

DSCP Mutation Map: Default DSCP Mutation Map

Trust device: none

qos mode: port-based

switch

说明:接口信任数据原来的 COS 值，默认接口 COS 值为 0。

（3）给接口配置新的 COS 值：

switch(config-if)#mls qos cos 5

（4）查看接口 CoS 值

switch#sh mls qos interface f0/1

FastEthernet0/1

trust state: trust cos

trust mode: trust cos

trust enabled flag: ena

COS override: dis

default COS: 5

DSCP Mutation Map: Default DSCP Mutation Map

Trust device: none

qos mode: port-based

switch#

说明:接口默认的 COS 值为 5。

（5）去除数据包原有的 COS 值

switch(config)#int f0/1

switch(config-if)#mls qos cos override

3. 配置映射：

说明:配置数据包各个参数之间的映射

（1）配置 CoS-to-DSCP map

switch(config)#mls qos map cos-dscp 5 10 15 20 25 30 35 40

（2）查看 CoS-to-DSCP map：

switch#sh mls qos maps cos-dscp

Cos-dscp map:

cos: 0 1 2 3 4 5 6 7

---

dscp: 5 10 15 20 25 30 35 40

switch#

（3）配置 ip-prec-to-dscp map

switch(config)#mls qos map ip-prec-dscp 11 12 13 14 15 16 17 18

(4)查看 ip-prec-to-dscp map

switch#sh mls qos maps ip-prec-dscp

IpPrecedence-dscp map:

ipprec: 0 1 2 3 4 5 6 7

---

dscp: 11 12 13 14 15 16 17 18

switch#

配置进口队列

1. 根据 COS 值将数据包放入进口队列和 WTD 阀值（全局配置）

（1）将 COS 值为 1、2、3 数据包放入队列 1 和阀值 1 中，将 COS 值为 4、5 数据包放入队列 2 和阀值 2 中

switch(config)#mls qos srr-queue input cos-map queue 1 1 2 3 1

switch(config)#mls qos srr-queue input cos-map queue 2 4 5 2

说明:默认为

Default CoS Input Queue Threshold Map

CoS Value	Queue ID-Threshold ID

0-4	1-1
5	2-1
6, 7	1-1

2. 在进口配置 WTD 的两个阀值（全局配置）

(1)配置队列 1 的 WTD 前两阀值分别为 50%和 70%

switch(config)#mls qos srr-queue input threshold 1 50 70

说明:默认 3 个 WTD 阀值都为 100%，值就是可用缓存的百分比，

3. 调整两个进口队列使用缓存的百分比（全局配置）

（1）配置两个进口队列使用缓存的百分比分别为 50%

switch(config)#mls qos srr-queue input buffers 50 50

说明:默认分别为 90 和 10

percentage1 percentage2, the range is 0 to 100.

(2)查看进口队列使用缓存的百分比

switch#sh mls qos input-queue

Queue : 1 2

---

buffers : 50 50

bandwidth : 4 4

priority : 0 10

threshold1: 100 100

threshold2: 100 100

switch#

4. 配置进口队列的 PQ：

可以配置进 Q 中的一个为 PQ（全局配置）

（1）配置队列 1 为 PQ，并且保证 20%的带宽

```
switch(config)#mls qos srr-queue input priority-queue 1 bandwidth 20
```

说明:配置队列 1 为 PQ，默认队列 2 为 PQ，总带宽的 20%可以保证，剩下的再由两个队列根据 weight 分配。

（2）查看接口队列

switch#sh mls qos input-queue

Queue : 1 2

---

buffers : 50 50

bandwidth : 4 4

priority : 20 0

鸿鹄培训+考试群：301283867

threshold1: 100 100

threshold2: 100 100

switch#

说明:队列 2 为 PQ，并且保证带宽 20%

5. 控制进口队列的可用带宽（全局配置）

(1)关闭 PQ 功能，并且分配两个队列的可用带宽比为 25%和 75%

switch(config)#mls qos srr-queue input priority-queue 1 bandwidth 0

switch(config)#mls qos srr-queue input bandwidth 25 75

说明:默认分别 4 、4

（2）查看进口队列

switch#sh mls qos input-queue

Queue : 1 2

---

buffers : 50 50

bandwidth : 25 75

priority : 0 0

threshold1: 100 100

threshold2: 100 100

switch#

说明:两个队列的可用带宽比为 25%和 75%

配置出口队列

说明:出口有 expedite queue，也就是 PQ，配置这些队列的方式为配置 queue-set，总共为 2 个 queue-set，再将接口放入相应 queue-set，即得到相应 queue-set 的配置参数。

如果 expedite queue 打开了，默认队列 1 将优于 SRR shaped 和 shared 分配的 weight

如果 expedite queue 关了，默认队列 1 将被 shaped 模式取代

如果 expedite queue 关了，并且没有配置 SRR，则默认队列 1 在 shared 模式。

1. 为队列分配内存

（1）为 4 个队列分配可用的内存比例

switch(config)#mls qos queue-set output 1 buffers 20 20 20 40

说明:四个值加起来必须是 100%，默认分配为 25、25、25、25

2. 为队列配置 WTD 阀值和队列最小内存占用率于最大内存占用率(全局配置)

（1）配置 queue-set 1,将队列 1 的前两个 WTD 阀值分别设置为 50%和 70%，最小内存占用率于最大内存占用率分别为 50%和 80%

switch(config)#mls qos queue-set output 1 threshold 1 50 70 60 80

说明:默认全部为 100%

3. 在出口队列中根据 COS 值将数据包放入相应队列和相应 WTD 阀值(全局配置)

（1）配置 Cos 4 5 到队列 1 和 WTD 阀值 2

switch(config)#mls qos srr-queue output cos-map queue 1 threshold 2 4 5

说明:默认为

Default CoS Output Queue Threshold Map

CoS Value	Queue ID-Threshold ID
0, 1	2-1
2, 3	3-1
4	4-1
5	1-1
6, 7	4-1

（2）查看 COS 值到相应队列和相应 WTD 阀值

switch#sh mls qos maps cos-output-q

Cos-outputq-threshold map:

cos: 0 1 2 3 4 5 6 7

---

queue-threshold: 2-1 2-1 3-1 3-1 1-2 1-2 4-1 4-1

switch#

4. 将接口放入相应 queue-set（接口配置）

（1）将接口 f0/1 放入 queue-set 2

```
switch(config)#int f0/1
switch(config-if)#queue-set 2
```

说明:默认在 queue-set 1

（2）查看接口所在 queue-set

```
switch#sh mls qos interface queueing
FastEthernet0/1
Egress Priority Queue : disabled
Shaped queue weights (absolute) :  25 0 0 0
Shared queue weights  :  25 25 25 25
The port bandwidth limit : 100  (Operational Bandwidth:100.0)
The port is mapped to qset : 2
```

5. 接口下开 PQ：

（1）将接口 f0/1 开启 PQ

```
switch(config)#int f0/1
switch(config-if)#priority-queue out
```

说明： 默认 PQ 是关闭，且默认 PQ 是队列 1，在传数据时，只有所有 PQ 中的数据全部传完，才能传其它队列的数据。

（2）查看接口 PQ

说明:只能使用命令 show running-config

配置 SRR 中的 shared 和 shaped

1. 在接口为 4 个队列分配 weights 值（使用 shared 分配）

(1)分配 weights 值

switch(config)#int f0/1

switch(config-if)#srr-queue bandwidth share 20 20 20 20

说明:默认 4 个队列的 weights 值分别为 25、25、 25、 25

（2）查看接口队列的 weights 值：

switch#sh mls qos interface f0/1 queueing

FastEthernet0/1

Egress Priority Queue : enabled

Shaped queue weights (absolute) : 25 0 0 0

Shared queue weights : 20 20 20 20

The port bandwidth limit : 100 (Operational Bandwidth:100.0)

The port is mapped to qset : 2

switch#

2. 配置接口的 shaped模式

(1)将接口 f0/1 的队列 1 模式设置为 shaped，且使用接口带宽的 1/4

switch(config)#int f0/1

switch(config-if)#srr-queue bandwidth shape 4 0 0 0

说明:队列 1 为 shaped 模式，且使用接口带宽的 1/4，即 25%，值为 0 的队列表示在 shared 模式。

（2）查看接口的队列模式

switch#sh mls qos interface f0/1 queueing

FastEthernet0/1

Egress Priority Queue : enabled

Shaped queue weights (absolute) : 4 0 0 0

Shared queue weights : 20 20 20 20

The port bandwidth limit : 100 (Operational Bandwidth:100.0)

The port is mapped to qset : 2

switch#

3. 配置接口的 shared模式

配置接口 4 个队列的 weight 值分别为 1、2、3、4

(1)switch(config)#int f0/1

switch(config-if)#srr-queue bandwidth share 1 2 3 4

说明:接口 4 个队列的 weight 值分别为 1、2、3、4，所以每个队列可用带宽分别为 1/(1+2+3+4), 2/(1+2+3+4), 3/(1+2+3+4), and 4/(1+2+3+4)

(2)查看接口的模式

switch#sh mls qos interface f0/1 queueing

FastEthernet0/1

Egress Priority Queue : enabled

Shaped queue weights (absolute) : 4 0 0 0

Shared queue weights : 1 2 3 4

The port bandwidth limit : 100 (Operational Bandwidth:100.0)

The port is mapped to qset : 2

switch#

说明:因为队列 1 为 shaped 模式，占用带宽 25%，所以队列 2、队列 3、队列 4 的合用带宽为 75%，结果分别为接口总带宽的 2/(2+3+4)×75、3/(2+3+4)×75、4/(2+3+4)×75

4. 限制出口带宽

说明:比如用户只付了相应带宽的钱，所以要限制接口的可用带宽

（1）限制接口的可用带宽为总带宽的 80%

```
switch(config)#int f0/1

switch(config-if)#srr-queue bandwidth limit 80
```

说明:范围是 10 to 90.，默认是 100%。

(2)查看接口可用带宽

```
switch#sh mls qos interface f0/1 queueing

FastEthernet0/1

Egress Priority Queue : enabled

Shaped queue weights (absolute) :  4 0 0 0

Shared queue weights  :  1 2 3 4

The port bandwidth limit : 80  (Operational Bandwidth:80.0)

The port is mapped to qset : 2


switch#
```

说明:接口可用带宽为 80%

配置 auto-QOS

1. 接口配置 Auto-QOS

（1）在接口 f0/1 开启 Auto-QOS

switch(config)#int f0/1

switch(config-if)#auto qos voip cisco-phone

或

switch(config-if)#auto qos voip trust

说明:开启了接口的 Auto-QOS，并自动为 cisco-phone 配置 QOS 策略，接口使用 CDP 发现 cisco-phone，

(2)查看接口 Auto-QOS

switch#show auto qos interface

FastEthernet0/1

auto qos voip cisco-phone

switch#

说明:接口已经启用 cisco-phone 的 Auto-QOS。

Security

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

Log 和 log-input 作用

概述

当路由器为用户转发了数据之后，如果管理员想查看路由器曾经为哪些用户转发过数据，在正常情况下，这是无法查证的。但是，可以通过接口配置 ACL，并且强制 ACL 记录下曾经转发过的用户记录，这样，就能从路由器得知哪些用户是发起过数据的，并且发送了多少数据，但是用户发出的数据内容，是无法记录的。

鸿鹄培训+考试群：301283867

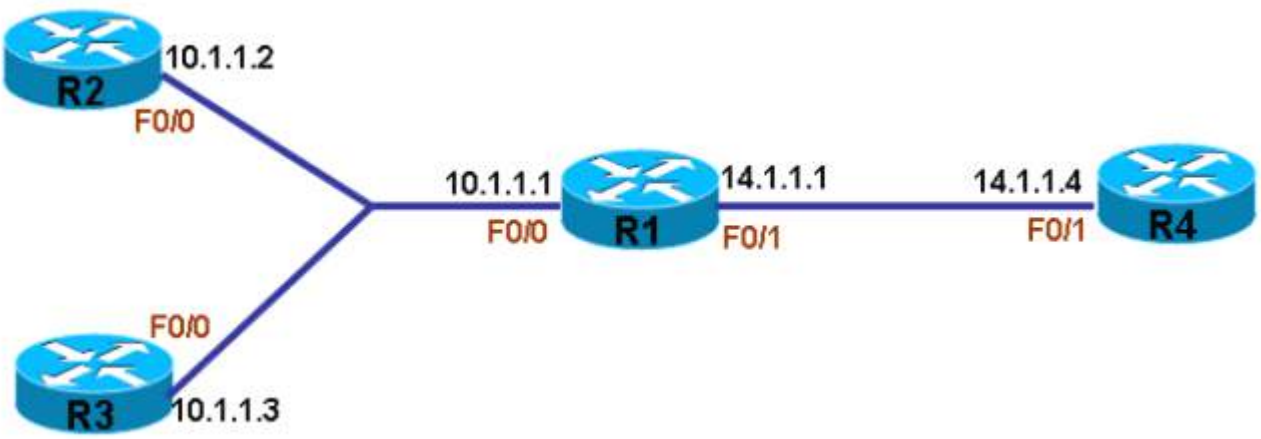
要达到以上目的，那在配置 ACL 时，使用 Log 和 log-input 的功能，并且将配置好的 ACL 用于接口上。

Log 和 log-input 的区别是：

Log 只能记录数据包通过时的源 IP 和目的 IP，

而 log-input 除了记录源 IP 和目的 IP 之外，还会记录源的 MAC 地址。

配置



1.配置 ACL 中的 Log

说明：配置路由器 R1，让其允许 R2 发来的数据包通过，但拒绝 R3 的数据包通过，并且记录下它们数据量。

（1）配置 ACL

说明：配置 ACL，允许 R2，拒绝 R3，分别使用 log 关键字

r1(config)#access-list 100 permit ip host 10.1.1.2 any log

r1(config)#access-list 100 deny ip host 10.1.1.3 any log

（2）应用 ACL

r1(config)#int f0/0

r1(config-if)#ip access-group 100 in

（3）测试结果

说明：从 R2 和 R3 分别 ping R4，查看 R1 上的 log

Oct 1 14:15:26: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 10.1.1.2 -> 14.1.1.4 (0/0), 5 packets

Oct 1 14:16:46: %SEC-6-IPACCESSLOGDP: list 100 denied icmp 10.1.1.3 -> 14.1.1.4 (0/0), 5 packet

说明：从 R1 上弹出的日志可以看出，R2 到 R4 的数据包是被放行了的，而 R3 到 R4 的数据包被丢弃了。

（4）查看 ACL 记录

r1#sh ip access-lists

Extended IP access list 100

10 permit ip host 10.1.1.2 any log (25 matches)

20 deny ip host 10.1.1.3 any log (5 matches)

说明：从 ACL 中也可以看出，R2 的流量被放行，R3 的流量被拒绝了。

2.配置 ACL 中 log-input

说明：配置路由器 R1，让其允许所有数据包通过，不仅记录下它们数据量，还将记录下源 MAC。

（1） 配置 ACL：

说明：配置 ACL，允许所有数据包通过，并且使用 log-input 关键字

r1(config)#access-list 130 permit ip an an log-input

（2）应用 ACL

r1(config)#int f0/0

r1(config-if)#ip access-group 130 in

（3）查看 R2 的源 MAC

r2#show interfaces f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is AmdFE, address is 0013.1a2f.1200 (bia 0013.1a2f.1200)

Internet address is 10.1.1.2/24

（4）从 R2 ping R4，查看 R1 上的 log

Oct 1 14:23:21: %SEC-6-IPACCESSLOGDP: list 130 permitted icmp 10.1.1.2 (FastEthernet0/0 0013.1a2f.1200) -> 14.1.1.4 (0/0), 1 packet

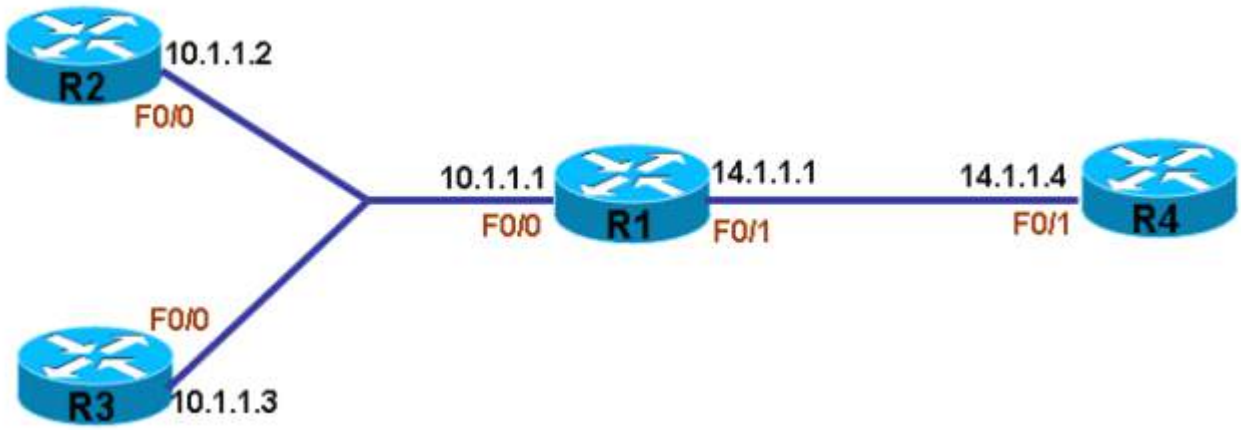
说明：从 R1 上弹出的日志可以看出，R2 到 R4 的数据包是被放行了的，并且还看到 R2 的源 MAC。

remark

概述

在配置 ACL 时，有时因为条目太多，ACL 结构复杂，事后可能很难辨别出每条 ACL 的作用分别是什么，在这种情况下，就可以在配置 ACL 时，给 ACL 中的条目写上标记，类似于说明文字，这通过 remark 来实现，remark 可以在条目的前一行，也可以在后一行，由自己决定，但 remark 不能和条目同一行。

配置



1.配置 ACL，并使用 remark

说明：如上图，在 R1 上配置 ACL 时，拒绝 R2，但允许 R3，还分别为每个条目写上注释。

r1(config)#access-list 100 remark Deny_R2 写上拒绝 R2 的注释

r1(config)#access-list 100 deny ip host 10.1.1.2 any

r1(config)#access-list 100 remark Permit_R3 写上拒绝 R3 的注释

r1(config)#access-list 100 permit ip host 10.1.1.3 any

2.查看结果

说明：在 ACL 中是无法查看注释的，但在 running-configuration 中可以看出。

access-list 100 remark Deny_R2

access-list 100 deny ip host 10.1.1.2 any

access-list 100 remark Permit_R3

access-list 100 permit ip host 10.1.1.3 any

Autocommand

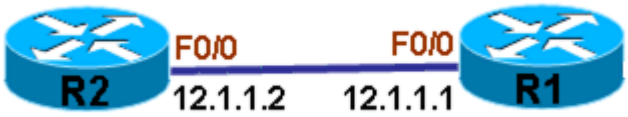
概述

有时，当客户的网络出现故障时，需要远程工程师协助或指导客户解决故障，这时就需要远程工程师 telnet 到客户的网络设备上，但是却并不希望远程工程师去直接更改用户设备的配置，在这种情况下，就可以在用户的设备上为远程工程师配置一个用户，通过这样的用户登陆设备之后，可以自动执行远程工程师想要执行的命令，从而达到了远程工程师查看设备配置的目的，又不违反修改配置的规矩。

要实现这样的功能，就可以在设备上配置 Autocommand 的功能，这样，当相应的用户 telnet 到设备时，就可以自动执行其想要的命令。

这样的 Autocommand 可以配置为所有 VTY 连上来的人执行，即配置在 VTY 接口下，也可以单独为某个用户执行，即配置在用户名之后。但这样的命令都只能执行一条。

配置



1.在 VTY 下为所有用户配置自动执行命令

（1）在 R2 上配置用户名和密码

r2(config)#username ccie password cisco

（2） 配置为所有 VTY 用户自动执行命令

说明：要为所有 VTY 用户执行命令，就配置在 VTY 接口下，这里配置自动执

行命令为 show ip interface brief

r2(config)#line vty 0 935

r2(config-line)#login local

r2(config-line)#autocommand show ip interface brief

（3）测试结果

说明：从 R1 telnet 到 R2，输入正确用户名和密码，即可看到命令执行后的输出

r1#telnet 12.1.1.2

Trying 12.1.1.2 ... Open

User Access Verification

Username: ccie

Password:

Interface IP-Address OK? Method Status Protocol

FastEthernet0/0 12.1.1.2 YES manual up up

FastEthernet0/1 unassigned YES unset administratively down down

Serial1/0 unassigned YES unset administratively down down

Serial1/1 unassigned YES unset administratively down down

Serial1/2 unassigned YES unset administratively down down

Serial1/3 unassigned YES unset administratively down down

[Connection to 12.1.1.2 closed by foreign host]

r1#

说明：从结果中可以看出，R1 telnet 到 R2 后，并不需要输入命令，就弹出之前命令的自动执行命令的输出结果，这样即达到了查看配置的目的，又不违反更改配置的规矩。

2.为单个用户配置自动执行命令

（1）配置用户名

r2(config)#username test password test

（2）为单个用户配置自动执行命令

说明：这里配置自动执行命令：show ip route

r2(config)#username test autocommand show ip route

（3）从 R1 上 telnet 到 R2 做测试

r1#telnet 12.1.1.2

Trying 12.1.1.2 ... Open

User Access Verification

Username: test

Password:

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

[Connection to 12.1.1.2 closed by foreign host]

r1#

说明：可以看到 R1 从 VTY 连上 R2 时，输入相应的 用户名和密码后，便自动执行了相应的命令，并且证明用户的命令优先于接口的命令。

Privilege Levels

概述

在 Cisco 设备中，将所有用户的操作权限分为 0-15 共 16 个等级，0 为最低等级，15 为最高等级。等级越高，能执行的命令就越多，权限就越大。要给用户赋于等级，可以在配置用户名或者密码时赋予。

在 Cisco 设备中，有一种最初级的模式，称为用户模式，即 User EXEC mode，默认表示为 Router>，在这个模式下，默认等级是 1，能执行的命令命令是相当少的，而需要注意的是，在这个模式下，永远只能执行等级为 1 的命令，如果要将等级提高到更高，就需要手动进入更高等级的模式，这个模式被称为特权模式，即 Privileged EXEC mode，表示为 Router#，通常此模式被称为 enable 模式。在没有

指定等级而进入 enable 模式后，默认等级为 15，也就表示可以完全控制设备。如果要进入比 1 级更高的模式而又不是 15 级，可以在进入 enable 模式时手工指定要进入的等级。除此之外，在创建本地用户数据库时，也可以为相应用户指定相应等级，如果不指定，默认用户等级为 1 级。

通过以上方法为相应用户或密码分配等级之后，他们所能执行的命令也只是相应等级范围内的，比如 5 级的用户是不能执行 15 级的命令的，但是可以手工赋予每个等级可以执行哪些命令，如让 5 级的用户能执行某 15 级的命令，如果 5 级用户能执行某 15 级的命令，那么 6-15 也都是可以执行的。

注：如果 15 级 Privileged EXEC mode 没有密码，默认只有本地终端直连可以登陆，VTY 是不能登陆的。

配置

1.查看 User EXEC mode 的默认等级

r1>show privilege

Current privilege level is 1

r1>

说明：从结果中看出，在 User EXEC mode 下，默认的等级为 1 级，也永远只能执行 1 级的命令。

2.查看 Privileged EXEC mode 的默认等级

（1） 不指定等级登陆 Privileged EXEC mode

r1>enable

r1#

（2）查看查看 Privileged EXEC mode 的默认等级

r1#show privilege

Current privilege level is 15

r1#

说明：从结果中看出，在 Privileged EXEC mode 下，默认的等级为 15 级

3.创建不同等级的密码

（1）创建一个 5 级的密码，为 cisco5

注：只有在创建 secret 密码时，才能指定等级，password 是不可以的。

r1(config)#enable secret level 5 cisco5

（2）创建一个 6 级的密码，为 cisco6

r1(config)#enable secret level 6 cisco6

4.测试 5 级的密码

（1）登陆 5 级的密码

r1>enable 5

Password: 输入 cisco5

r1#

（2）查看当前等级

r1#show privilege

Current privilege level is 5

r1#

说明：从结果中看出，当前等级为 5 级。

（3） 查看 show run 配置

r1#show run

^

% Invalid input detected at '^' marker.

r1#

说明：从结果中看出， 5 级用户是不能执行 show run 命令的。

5.测试 6 级的密码

（1）登陆 6 级的密码

r1>enable 6

Password: 输入 cisco6

r1#

（2）查看当前等级

r1#show privilege

Current privilege level is 6

r1#

说明：从结果中看出，当前等级为 6 级。

（4）查看 show run 配置

r1#show run

^

% Invalid input detected at '^' marker.

r1#

说明：从结果中看出， 6 级用户也是不能执行 show run 命令的。

6.赋予 5 级用户可以执行 show run 命令

说明：要赋予某个等级用户可以执行某个命令时，必须使用 15 级的等级来指定，还必须说明该等级是在哪个模式下执行命令的。

r1(config)#privilege exec level 5 show run

说明：赋予 5 级用户可以在 exec 执行 show run 命令

7.测试 5 级用户的命令

r1>enable 5

Password:

r1#show run

Building configuration...

Current configuration : 55 bytes

!

boot-start-marker

boot-end-marker

!

end

r1#

说明：可以看到,5 级用户已经可以执行 show run 命令，但要注意的是，该等级只能查看权限范围内能够配置的参数情况。

8.查看 6 级是否可以执行 show run

r1>enable 6

Password:

r1#

r1#

r1#sh run

Building configuration...

Current configuration : 55 bytes

!

boot-start-marker

boot-end-marker

!

!

end

r1#

说明：当一条命令赋予某个等级之后，比此等级更高的等级同样获得该命令的执行权。

9.创建默认等级的本地用户数据库

（1）配置用户名和密码

r1(config)#username aaa password bbb

（2）登陆已配置的用户名和密码

r1>login

Username: aaa

Password: 输入密码 bbb

r1>

(3)查看默认用户名的等级

R1>show privilege

Current privilege level is 1

R1>

说明：可以看到创建的用户默认是 1 级

10.创建等级为 15 的用户

（1）创建用户

R1(config)#username ccc privilege 15 password ddd

（2）登陆用户

r2>login

Username: ccc

Password:

r2#

（3） 查看该用户等级

R1#show privilege

Current privilege level is 15

R1#

说明：可以看到创建的用户是 15 级

基于时间的 ACL

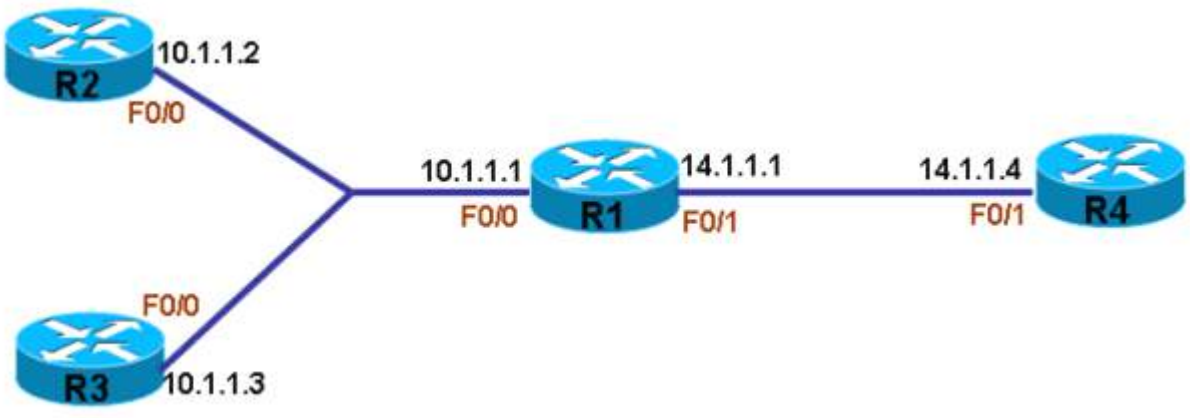
概述

一个很通常的需求，就是在某个公司里，有时希望限制员工在某个时间范围内才可以访问网页，即 HTTP 服务，或其它服务，在时间范围之外，就不能访问，那么这样的需求，就可以通过配置基于时间的 ACL 来实现。

要通过 ACL 来限制用户在规定的时间范围内访问特定的服务，首先设备上必须配置好正确的时间。在相应的时间要允许相应的服务，这样的命令，在配置 ACL 时，是正常配置的，但是，如果就将命令正常配置之后，默认是在所有时间内允许的，要做到在相应时间内允许，还必须为该命令加上一个时间限制，这样就使得这条 ACL 命令只在此时间范围内才能生效。而要配置这样的时间范围，是通过配置 time-range 来实现的，在 time-range 中定义好时间，再将此 time-range 跟在某 ACL 的条目之后，那么此条目就在该时间范围内起作用，其它时间是不起作用的。

在定义 time-range 时，常用的时间简单分为两种，第一种叫做绝对时间（absolute），即这个时间只生效一次，比如 2010 年 1 月 1 月 15:00；另一种时间叫做周期时间（periodic），即这个时间是会多次重复的，比如每周一，或者每周一到周五。

配置



前提：在 R1 路由器上需要提前配置好正确的时间，此步骤省略。

1.配置 time-range

r1(config)#time-range TELNET

r1(config-time-range)#periodic weekdays 9:00 to 15:00

说明：定义的时间范围为每周一到周五的 9:00 to 15:00

2.配置 ACL

说明：配置 R1 在上面的时间范围内拒绝 R2 到 R4 的 telnet，其它流量全部通过。

r1(config)#access-list 150 deny tcp host 10.1.1.2 any eq 23 time-range TELNET

r1(config)#access-list 150 permit ip any any

3.应用 ACL

r1(config)#int f0/0

r1(config-if)#ip access-group 150 in

4.测试时间范围内的流量情况

（1）查看当前 R1 的时间

r1#sh clock

14:34:33.002 GMT Thu Oct 1 2009

r1#

说明：当前时间为周四 14:34，即在所配置的时间范围内。

（2）测试 R2 向 R4 发起 telnet 会话

r2#telnet 14.1.1.4

Trying 14.1.1.4 ...

% Destination unreachable; gateway or host down

r2#

说明：可以看到，在规定的时间范围内，R2 向 R4 发起 telnet 会话是被拒绝的。

（3）测试除 telnet 外的其它流量

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：可以看到，在规定的时间范围内，除了 telnet 之外，其它流量不受限制。

（4）测试除 R2 之外的设备 telnet 情况

r3#telnet 14.1.1.4

Trying 14.1.1.4 ... Open

r4>

说明：可以看到，除 R2 之外，其它设备 telnet 并不受限制。

5.测试时间范围外的流量情况

（1）查看当前 R1 的时间

r1#sh clock

15:01:15.206 GMT Thu Oct 1 2009

r1#

说明：当前时间为周四 15:01，即在所配置的时间范围之外。

（2）测试 R2 向 R4 发起 telnet 会话

r2#telnet 14.1.1.4

鸿鹄培训+考试群：301283867

Trying 14.1.1.4 ... Open

r4>

说明：在时间范围之外，所限制的流量被放开。

Reflexive ACL

概述

在某些网络中，为了考虑安全性，不希望外网的用户主动向内网发起连接，因为怀疑这样的动作可能是攻击行为。但是内网用户主动向外部发起的连接，外网的回包可以进入内网。这样的需求，如果使用普通的 ACL 在外网进来的接口上拒绝所有数据包，这肯定是不行的，因为这样虽然保证外网不能访问内网了，安全目的达到了，但是内网主动向外网发起的连接，外网回包时也进不来了，所以这种普通 ACL 不可行。更好的方法就是，先拒绝所有外网主动向内网发起的连接，但是在内网主动向外网发起的连接中，作好记录，打好标记，等到外网回包时，能够让其顺利进入内网，这样即保证了外网不能主动访问内网，实现了安全，又保证了内网发起的连接，外网可以回应，也不妨碍通信。要实现这样的功能，就可以通过特殊的 ACL，即 Reflexive ACL 来实现。

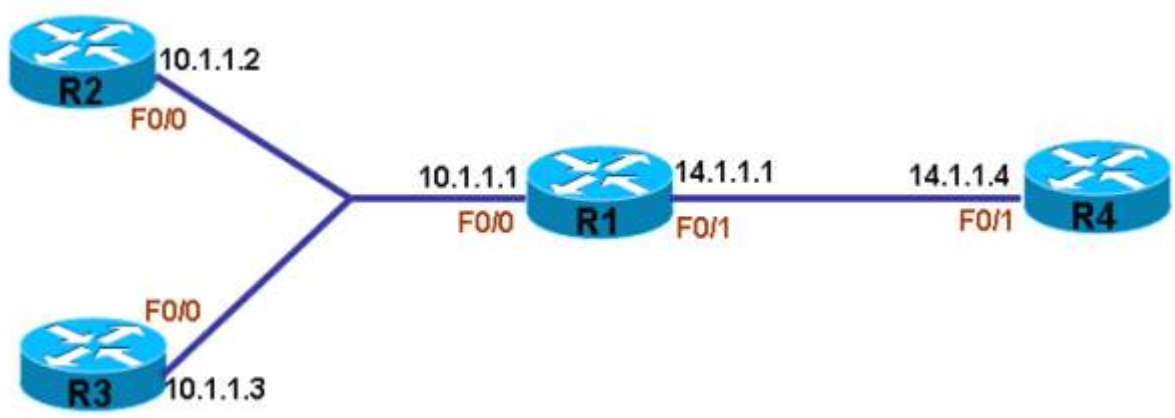
Reflexive ACL 就是根据以上所述，先拒绝外网向内网发送数据，然后允许内网向外网发送数据，但是在内网的数据发向外网时，这些数据的会话会被记录，被标记，等外网发回的数据和这些有记录的会话属于同一会话时，便可临时在进来的方向上打开缺口，让其返回，其它外网发来的不在记录中的数据，统统不能进入内网。所以根据这些原理，Reflexive ACL 需要有两个 ACL 来配合使用，一个 ACL 是用在外网到内网的方向，以拒绝外网的主动连接，另一个 ACL 是用在内网到外网的方向，用来检测内网有数据发向外网时，做上记录，等外网回包时，就在之前那个 ACL 中打开一个临时缺口，让外网的回包进入，这样就实现了之前所说的安全功能。

Reflexive ACL 有许多功能限制，只支持命名的扩展 ACL，并且思科官方文档会解释说此 ACL 在最后没有像通常那样隐含拒绝所有，所以请注意你使用的 IOS 是否隐含拒绝了所有数据通过。此 ACL 正因为外网数据在主动进入内网时被拒绝的，所以当内网数据出去时，这个会话会做好记录，会在进的方向给打开缺口，让其返回，这个被打开的缺口，在会话结束后，缺口被关闭。其实大家都知道，只有 TCP 的数据才存在会话，所以当 TCP 数据传完之后，会马目关闭缺口，但是对于

没有会话的 UDP，就不能使用上面的方法了，就软件根据 timeout 来判断数据是否传完，如果没有给 ACL 指定 timeout，默认使用全局 timeout，默认全局是 timeout 是 300 秒，在 timeout 结束后，缺口被关闭。正因为这些从内网发到外网的数据被记录了，只有返回的数据和记录中相符，才能进入内网，所以如果返回的数据不符，就进不来，因此，会话在中途端口号是不能更换的，一旦更换，就无法匹配记录了。而像 FTP 这样的会话，在中途要改变端口号，所以 FTP 在有 Reflexive ACL 时，不能很好的工作。

在 Reflexive ACL 拒绝外网数据进入内网时，外网是不能先向内网发起连接的，但是并不需要将所有数据都拒绝，在配置时，某些数据就可以放开，让它和正常数据一样没有限制，比如路由协议的数据。而在定义什么样的数据出去之后被记录，可以返回，也可以只选择特定的数据。当在定义这个需要被记录的数据时，使用 ACL 来匹配，只能写一条，如果写多条，除了第一条，其它统统无效。

配置



说明：R4 为外网，R2 和 R3 为内网。

1.配置拒绝外网主动访问内网

说明：拒绝外网主动访问内网，但是 ICMP 可以不受限制

（1）配置允许 ICMP 可以不用标记就进入内网，其它的必须被标记才返回

r1(config)#ip access-list extended come

r1(config-ext-nacl)#permit icmp any any 被允许的 ICMP 是不用标记即可进入内网的

r1(config-ext-nacl)#evaluate abc 其它要进入内网的，必须是标记为 abc 的

（2）应用 ACL

r1(config)#int f0/1

r1(config-if)#ip access-group come in

2.测试结果

（1）测试外网 R4 的 ICMP 访问内网

r4#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r4#

说明：可以看到，ICMP 是可以任意访问的

（2）测试外网 R4 telnet 内网

r4#telnet

r4#telnet 10.1.1.2

Trying 10.1.1.2 ...

% Destination unreachable; gateway or host down

r4#

说明：可以看到，除 ICMP 之外，其它流量是不能进入内网的。

（1） 测试内网 R2 的 ICMP 访问外网

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms

r2#

说明：可以看到，内网发 ICMP 到外网，也正常返回了

（2） 测试内网 R2 发起 telnet 到外网

r2#telnet 14.1.1.4

Trying 14.1.1.4 ...

% Connection timed out; remote host not responding

r2#

说明：可以看到，除 ICMP 之外，其它流量是不能通过的。

3.配置内网向外网发起的 telnet 被返回

说明：外网和内网之间的 ICMP 可以不受限制，外网不能 telnet 内网，但内网 telnet 外网时，需要配置记录，让其返回，根据上面的 ACL 配置，可以返回的，必须是标为 abc 的，所以在此为内网发向外网的 telnet 标为 abc，返回时，就会有缺口，因此内网能正常 telnet 外网，但外网不可主动 telnet 内网。

（1）配置内网出去时，telnet 被记录为 abc,将会被允许返回

r1(config)#ip access-list extended goto

r1(config-ext-nacl)#permit tcp any any eq telnet reflect abc timeout 60 telnet 已记为 abc

r1(config-ext-nacl)#permit ip any any

（2）应用 ACL

r1(config)#int f0/1

r1(config-if)#ip access-group goto out

4.测试结果

（1）查看 R2 到外网的 ICMP

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：ICMP 属正常

（3）查看内网向外网发起 telnet

r2#telnet 14.1.1.4

Trying 14.1.1.4 ... Open

r4>

说明：可以看出，此时内网发向外网的 telnet 因为被标记为 abc，所以在回来时，开了缺口，也就可以允许返回了。

（4）查看 ACL

r1#sh ip access-lists

Reflexive IP access list abc

permit tcp host 14.1.1.4 eq telnet host 10.1.1.2 eq 23395 (16 matches) (time left 33)

Extended IP access list come

10 permit icmp any any (86 matches)

20 evaluate abc

Extended IP access list goto

10 permit tcp any any eq telnet reflect abc

20 permit ip any any (20 matches)

r1#

说明：可以看到，有一条为 abc 的 ACL 为允许外网到内网的 telnet，正是由于内网发到外网的 telnet 被标记了，所以也自动产生了允许其返回的 ACL，并且后面跟有剩余时间。

Context-Based Access Control（CBAC）

概述

CBAC 要实现的功能，就是 Reflexive ACL 所要实现的功能，但是由于 Reflexive ACL 有许多不足之处，所以需要 CBAC 来更好的支持。

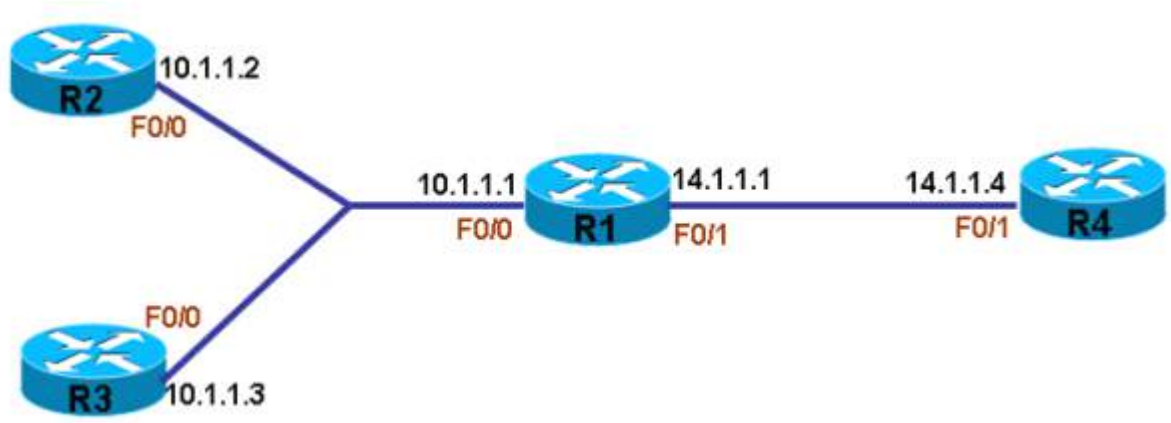
Reflexive ACL 不支持中途变换端口的协议，如 FTP，这在 CBAC 中，是可以支持的，Reflexive ACL

对于什么样的数据包需要允许返回，是需要写 ACL 来精确匹配的，并且只能写一条，

而 CBAC 可以直接写数据的协议名称，不用写 ACL 去匹配，CBAC 所写的协议，就是 OSI 第 7 层应用层的协议，所以很方便用户匹配数据，并且可以写多个协议。CBAC 在思科官方文档中也会说不支持 ICMP 这个协议，所以请注意你的 IOS，在实际中，支持 CBAC 的，都是支持 ICMP 的。

用户不希望某些数据能主动从外网发向内网，只有当数据是从内网发向外网时，被允许返回，对于这样的数据，应该一开始就拒绝从外网进入内网，然后从内网发向外网时，让 CBAC 记住这个会话，并且在从外网进入内网的接口上打开缺口，方便其返回时顺利进入内网。所以在使用 CBAC 时，就需要先写好一个 ACL，应用在外网进内网的接口上，用以拒绝某些数据的进入，当这些数据从内网发起外网时，CBAC 就在进入的接口上临时创建缺口，让其返回。这个用在拒绝外网进入内网的 ACL，必须是扩展 ACL。以路由器自身为源或目的的数据，不被 CBAC 所记录。CBAC 同样有超时，由自己定义，并且可以打开审记功能，产生的日志就可以被记录。

配置



说明：R4 为外网，R2 和 R3 为内网。

1.配置拒绝所有的数据包从外网进入内网

（1）在 R1 上配置 ACL 防止所有数据包进来：

鸿鹄培训+考试群：301283867

r1(config)#access-list 100 deny ip an any

（2）应用 ACL

r1(config)#int f0/1

r1(config-if)#ip access-group 100 in

2.测试结果

（1）使用 ICMP 和 telnet 测试外网访问内网

r4#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

U.U.U

Success rate is 0 percent (0/5)

r4#telnet 10.1.1.2

Trying 10.1.1.2 ...

% Destination unreachable; gateway or host down

r4#

说明：从结果中看出，外网向内网发起的 ICMP 和 telnet 均不能通过。

（2）使用 ICMP 和 telnet 测试内网访问外网

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#telnet 14.1.1.4

Trying 14.1.1.4 ...

% Connection timed out; remote host not responding

r2#

说明：从结果中看出，内网向外网发起的 ICMP 和 telnet 也不能通过。

3.配置 CBAC 允许相应协议被返回

（1）在 R1 上配置 CBAC 记录 telnet 会话，因此可以返回

r1(config)# ip inspect name ccie tcp audit-trail on timeout 60

r1(config)#int f0/1

r1(config-if)#ip inspect ccie out

说明：测试 IOS 没有单独的 telnet 协议，只能选整个 TCP 协议。

4.测试 CBAC 效果

（1）测试外网向内网发起 ICMP

r4#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

U.U.U

Success rate is 0 percent (0/5)

r4#

说明：外网向内网发起的 ICMP 是不能进入的。

（2）测试外网向内网发起 telnet

r4#telnet 10.1.1.2

鸿鹄培训+考试群：301283867

Trying 10.1.1.2 ...

% Destination unreachable; gateway or host down

r4#

说明：外网向内网发起的 telnet 是不能进入的。

（3）测试内网向外网发起 ICMP

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：因为 CBAC 只记录 TCP，所以 ICMP 没有被允许返回。

（4）测试内网向外网发起 telnet

r2#telnet 14.1.1.4

Trying 14.1.1.4 ... Open

r4>

说明：因为 CBAC 记录 TCP，所以 telnet 被允许返回。

（1）查看 CBAC 的会话记录

r1#sh ip inspect sessions

Established Sessions

Session 835AC510 (10.1.1.2:63276)=>(14.1.1.4:23) tcp SIS_OPEN

r1#

说明：内网到外网的 telnet 被 CBAC 成功记录

（2）查看 ACL 中 CBAC 打开的缺口

r1#sh ip access-lists

Extended IP access list 100

permit tcp host 14.1.1.4 eq telnet host 10.1.1.2 eq 20029 (13 matches)

10 deny ip any any (48 matches)

r1#

说明：可以看到，原本拒绝所有的 ACL，被 CBAC 临时创建了允许 telnet 返回的条目。

5.看日志

Oct 1 18:52:22: %FW-6-SESS_AUDIT_TRAIL: tcp session initiator (10.1.1.2:62155) sent 30 bytes -- responder (14.1.1.4:23) sent 32 bytes

r1#

说明：在 CBAC 检测时，审计打开，并且会产生日志。

Port to Application Mapping（PAM）

概述

常用的协议 HTTP 对应 TCP 端口号 80，常用的协议 telnet 对应 TCP 端口号 23，这些端口号，Cisco 设备也是遵守默认的端口号规则，而这些协议对应的端口号，是任何时候都是可以随意改动的。在正常情况下，看到 TCP 80，就会把它当成 HTTP 来处理，看到 TCP 23，就会当成 telnet 来处理。

当设备上开启了 CBAC 后，CBAC 是根据相应的协议来做好会话记录，从而在

ACL 为其返回的流量打开缺口的。比如已经配置 CBAC 检测 HTTP 协议，也就是说当设备检测到有 TCP 80 的数据通过时，就会记录下会话，并且为其打开缺口，而检测 TCP 其它端口号，是不会这么做的。但是，在某些特殊时候，如果你发起的 HTTP 会话，端口号已经被改变，如已经改成 1000，那么这个时候你发起的 HTTP 会话就是 TCP 1000，对于 TCP 1000 这样的数据，在已配置好的 CBAC 中，是不会为其记录并打开缺口的，如果要让 CBAC 也知道你现在所使用的 HTTP 已经不再是标准的端口号了，但还希望 CBAC 为你服务，那么就必须手动告诉设备你已经将 HTTP 改为了 TCP 1000。这个功能就是靠 PAM 来实现的。

原本的 CBAC 只支持协议的标准端口，但是要让 CBAC 支持非常规端口协议，就需要 PAM 来完成这个工作。

要让设备知道相应协议是用哪些端口号，可以配置协议和对应的端口号，如果没有人为配置，系统中也同样会存在这样的对应表。表里面提供三种信息，分别为：

系统定义的映射

用户定义的映射

主机映射

其中系统定义的就是标准的协议端口号，是不能更改的；

而用户定义的可以随意更改和删除，如果同时存在系统定义的和用户定义的，那么会优先使用用户定义的。

主机映射就是可以使用重复端口，即是基于每台主机的，比如定义某台主机 HTTP 使用 TCP 1000，而定义另外一台主机 FTP 使用 TCP 1000。

也可以为某个协议定义一个范围的端口号，通过多次输入命令实现。

配置

说明：在 R3 上配置 NAT，让 telnet 到 13.1.1.100 的结果被转到 telnet 34.1.1.4

1.在 R3 上配置 NAT

说明：配置让 telnet 到 13.1.1.100，目标端口为 1000 的，结果被转到 telnet 34.1.1.4

(1) 定义 NAT 方向

r3(config)#int f0/0

r3(config-if)#ip nat inside

r3(config)#int f0/1

r3(config-if)#ip nat outside

（2）配置映射

r3(config)#ip nat inside source static tcp 34.1.1.4 23 13.1.1.100 1000

说明：当 telnet 13.1.1.100 1000 时，结果为 telnet 到 34.1.1.4

2.测试 telnet 结果

（1）测试从 R2 telnet 13.1.1.100，目标端口为 1000

r2#telnet 13.1.1.100 1000

Trying 13.1.1.100, 1000 ... Open

r4>

说明：从结果中看出，当 R2 telnet 13.1.1.100，目标端口为 1000 时，结果为结果为 telnet 到 34.1.1.4。

3.配置 CBAC

说明：在 R1 上配置 CBAC，拒绝从 F0/1 进来的所有数据，但是记录 R2 发起的 telnet

数据，并允许其返回

（1）配置 ACL 拒绝所有数据进入

r1(config)#access-list 100 deny ip any any

r1(config-if)#ip access-group 100 in

（2）配置 CBAC 允许 telnet返回

r1(config)#ip inspect name ccie telnet timeout 100

r1(config)#int f0/1

r1(config-if)#ip inspect ccie out

4.测试 CBAC 结果

（1）测试 R2 telnet 13.1.1.100，目标端口为 1000 时，CBAC 是否能为其记录会话并打开缺口

r2#telnet 13.1.1.100 1000

Trying 13.1.1.100, 1000 ...

% Connection timed out; remote host not responding

r2#

说明：从结果中看出，CBAC 并不会为端口号为 1000 的数据创建返回缺口，因为已配置的 CBAC 只记录 telnet，也就是标准 TCP 23 端口，而现在是 TCP 1000 端口，所以并不被关心。

5.配置 PAM

说明：已配置的 CBAC 只记录 telnet，也就是 TCP 为 23 端口的数据包，而现在的 telnet 为 TCP 端口号 1000，所以并没有被记录，因此配置 PAM，改变设备的默认 telnet 端口，应改为 1000，从而让 CBAC 根据此端口映射表作记录。

（1）配置 telnet 端口号为 1000

r1(config)#ip port-map telnet port tcp 1000

6.测试 CBAC 支持 PAM 非常规端口

（1）再次测试 R2 telnet 13.1.1.100，目标端口为 1000 时，CBAC 是否打开缺口

r2#telnet 13.1.1.100 1000

Trying 13.1.1.100, 1000 ... Open

r4>

说明：可以看出，CBAC 已经认为 telnet 为 TCP 端口号 1000，并成功为其打开缺口。

（2）查看 CBAC 中的会话

r1#sh ip inspect sessions

Established Sessions

Session 66C93DE4 (12.1.1.2:17502)=>(13.1.1.100:1000) telnet SIS_OPEN

r1#

说明：看到 CBAC 中成功理解 telnet 为端口号 1000。

7.定义范围端口给协议

（1）定义端口号 8001 到 8004 都给 HTTP

r1(config)#ip port-map http port 8001

r1(config)#ip port-map http port 8002

r1(config)#ip port-map http port 8003

r1(config)#ip port-map http port 8004

8.定义到主机映射

说明：可以让同一个端口被不同主机使用

（1）定义主机 10.1.1.1 的 HTTP 使用端口号 8000

r1(config)#access-list 10 permit 10.1.1.1

r1(config)#ip port-map http port 8000 list 10

（2）定义主机 20.1.1.1 的 FTP 使用端口号 8000

r1(config)#access-list 20 permit 20.1.1.1

r1(config)#ip port-map ftp port 8000 list 20

Lock-and-Key Security (Dynamic ACL)

概述

有一种特殊的需求，比如一台连接了内网和外网的路由器，某些时候想限制内网的用户访问外网，如果用户想要获得访问外网的权限，就必须通过认证。

路由器在最初阻挡用户的数据通过，在用户通过认证后，再放行。要实现这样的功能，可以使用 Dynamic ACL。Dynamic ACL 在一开始拒绝用户相应的数据包通过，当用户认证成功后，就临时放行该数据，但是在会话结束后，再将 ACL 恢复最初的配置。要定义 Dynamic ACL 什么时候恢复最初的配置，可以定义会话超时，即会话多久没有传数据，就断开，也可以定义绝对时间，即无论会话有没有结束，到了规定时间，也要断开。

Dynamic ACL 给用户提供的认证方法有多种，最常用的可以使用 AAA，本地用户数据库。只要用户提供了正确的用户名和密码，就可以获得网络访问权。用户要向路由器提供用户名和密码以获得认证，就必须 telnet 配置了 Dynamic ACL 的路由器，当 telnet 到路由器时，输入了正确的用户名和密码之后，认证就算通过。而要注意的是，用户输入的用户名，并不是普通的用户名，必须是具有访问功能的用户名，如果用户输入的用户名是普通的，那么等于普通的 telnet，并不能成为认

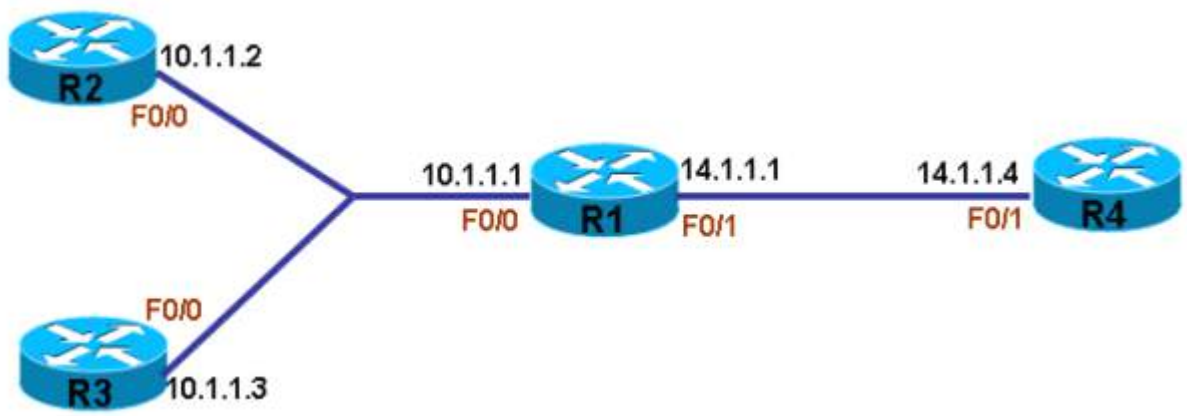
证，只有输入的用户名是具有访问功能的，才能通过认证并获得网络访问权。要赋即一个用户名网络访问权的功能，就需要配置 autocommand 来实现。

当使用 AAA 认证时，用户名要有 autocommand，也可以在配置本地用户数据库时为用户名添加，并且还可以加在 VTY 接口下。

会话结束的时间以分钟为单位，有空闲时间和绝对时间，如果两个时间同时配，空闲时间必须小于绝对时间，如果两个时间都不配，Dynamic ACL 打开的访问缺口必须手工清除。

因为 Dynamic ACL 在认证时是依靠用户 telnet 自己，所以一定要为用户打开 telnet 访问权限，某些数据也可以让其默认通过，比如路由协议的数据。认证通过之后，就可以访问相应的服务，在认证通过之后，具体可以访问哪些，需要配置 ACL 来定义，也就是配置了 dynamic 的条目，这样的动态条目只能配置一条，如果配多条动态 ACL，IOS 只认第一条，要注意 Dynamic ACL 只支持扩展的 ACL。

配置



说明：

R2 和 R3 为内网，R4 为外网，配置 R1，默认允许所有 telnet 通过，因为要使用 telnet 做认证，然后只有当认证通过之后，ICMP 才可以通过。

1.配置 Dynamic ACL

（1）配置默认不需要认证就可以通过的数据，如 telnet

r1(config)#access-list 100 permit tcp an an eq telnet

(2)配置认证之后才能通过的数据，如 ICMP，绝对时间为 2 分钟。

r1(config)#access-list 100 dynamic ccie timeout 2 permit icmp any any

（3）应用 ACL

r1(config)#int f0/0

r1(config-if)#ip access-group 100 in

2.测试访问

（1）测试内网 R2 telnet 外网 R4

r2#telnet 14.1.1.4

Trying 14.1.1.4 ... Open

r4>

说明：从结果中看出，telnet 不受限制。

（2）测试测试内网 R2 ping 外网 R4

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

U.U.U

Success rate is 0 percent (0/5)

r2#

说明：内网在没有认证之前，ICMP 是无法通过的。

3.配置本地用户数据库

r1(config)#username ccie password cisco

鸿鹄培训+考试群：301283867

4.配置所有人的用户名具有访问功能

r1(config)#line vty 0 181

r1(config-line)#login local

r1(config-line)#autocommand access-enable 这条必加

5.内网 R2 做认证

r2#telnet 10.1.1.1

Trying 10.1.1.1 ... Open

User Access Verification

Username: ccie

Password:

[Connection to 10.1.1.1 closed by foreign host]

r2#

说明：当 telnet 路由器认证成功后，是会被关闭会话的。

6.测试内网到外网的 ICMP 通信功能

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：认证通过之后，ICMP 被放行。

7.查看 ACL 状态

鸿鹄培训+考试群：301283867

```
r1#sh ip access-lists
Extended IP access list 100
    10 permit tcp any any eq telnet (105 matches)
    20 Dynamic ccie permit icmp any any
       permit icmp any any (5 matches)
r1#
```

说明：可以看到动态允许的流量已放行。

8.host 功能

有配置访问功能时，命令有

```
r1(config-line)#autocommand access-enable
r1(config-line)# autocommand   access-enable host
```

两种，并且后面可以跟时间，这里的时间为空闲时间，必须比之前的绝对时间要小，在配置访问功能时，如果没有加 host，那么内网一台主机通过认证之后，所有主机都能访问外网，加了 host，就变成谁通过了认证，谁才能访问外网。

TCP Intercept

概述

通常所说的 TCP 要比 UDP 可靠，是因为 TCP 是有会话的，是面向连接的，任何两点之间在通信时，都必须有一条会话，所有被丢失的数据包，都会重传。而 TCP 在建立这条会话之前，必须完成的任务是三次握手。

TCP 只有在完成了三次握手之后，才能建立会话。比如 R1 是客户端，要与服务器 R2 建立 TCP 会话，这三次握手的过程是：

第一次握手：R1 向 R2 发送一个数据包，并且携带一个序列号，如 100。

第二次握手：R2 向 R1 回一个数据包，先回答 R1 的序列号，为 100+1,结果是 101，然后也会向 R1 发出一个序列号，比如是 200，并等待对方的回答，直到 TCP 超时为止。

第三次握手：R1 根据 R2 提出的序列号，作出回答，200+1，结果为 201。

当这样的过程完成之后，即表示三次握手成功完成，即可建立 TCP 会话。

从上面的过程可以看到，如果 R1 在给服务器 R2 发起握手之后，服务器 R2 回应之后，如果 R1 并不作第三次握手的回应，那么服务器将启动一个等待计时器，直到超时为止。这样的握手被称为半开连接。因此，如果客户端向服务器发起大量的半开连接，就会导致服务器由于启动过多的计时器而耗尽系统资源，从而停止工作。

当一台网络上的正常服务器向用户提供服务时，如果用户对其进行上述的 TCP 攻击，将导致该服务器停止工作，所以就试图寻找一种方法来避免服务器遭受这样的 TCP 攻击。很显然，要避免这样的攻击，可以让服务器尽早的清除半开连接，从而避免握手数量的累加。也可以让服务器限制最大握手数量，即累加到一定数量的半开连接后，便停止客户端的握手请求。

这样的工作，可以在服务器上实现，而我们现在要做的，是在路由器上实现，因为当客户访问服务器时，会话是通过路由器的，所以中间的路由器可以开启监测功能，来监测这些握手会话，当某些握手长时间不完成时，便可认为是攻击，就向服务器发送重置该会话的请求，路由器也可以限制客户到达服务器的半开连接数量。

在这里，路由器要为服务器提供 TCP 保护，有两种工作模式。第一种是客户向服务器发起的握手请求被正常转发到服务器，但是路由器会监视这条会话，当超过规定时间，三次握手还没完成，那么路由器就向服务器发送重置该会话的请求，从而保护了服务器。这种模式被称为 watch 模式。第二种是当客户向服务器发起握手请求时，路由器收到这样的包之后，将其拦下，只有当三次握手完成之后，再将该会话交给服务器，让服务器和客户机之间正常连接，否则，路由器就清除该会话。

在配置 TCP Intercept 时，需要配置 ACL 要告诉路由器监测哪些 TCP 会话，如果 ACL 不配，路由器是不会采取任何动作的。

配置：

1.配置需要监视的 TCP

说明：通过配置 ACL 来实现

r1(config)#access-list 100 permit tcp any any

2.应用 ACL 到 TCP Intercept

r1(config)#ip tcp intercept list 100

3.配置 TCP Intercept 的模式

（1）配置 watch 模式

r1(config)#ip tcp intercept mode watch

（2）配置 intercept 模式

r1(config)#ip tcp intercept mode intercept

4.配置半开连接最长等待时间（默认 30 秒）

r1(config)#ip tcp intercept watch-timeout 60

说明：半开连接在 60 秒未完成三次握手，连接将被清除。

5.配置连接超时

说明：既然三次握手成功完成，TCP 会话也建立，而路由器也会默认跟踪该会话 24 小时，可以缩短该时间，比如 1 小时，可以理解为 TCP 会话建立后，多长时间不传数据，会话将断开。

r1(config)#ip tcp intercept connection-timeout 3600

6.定义总的未完成数

说明：定义半开连接数到多少就开始清除会话，有高低两个，是阀值，默认是 900 和 1100

r1(config)#ip tcp intercept max-incomplete low 800

r1(config)#ip tcp intercept max-incomplete high 1000

7.定义每分钟的未完成数，有高低两个，是阀值，默认是 900 和 1100

r1(config)#ip tcp intercept one-minute low 800

r1(config)#ip tcp intercept one-minute high 1000

8.配置丢弃模式

说明：当会话到达最大数量后，默认是清除最老（时间最长）的会话，也可以配置随机清除。

r1(config)#ip tcp intercept drop-mode oldest/random

Unicast Reverse Path Forwarding (uRPF)

概述

uRPF 被称为单播的反向路径转发，功能是让路由器具备防 IP 欺骗或 IP 伪造的能力。

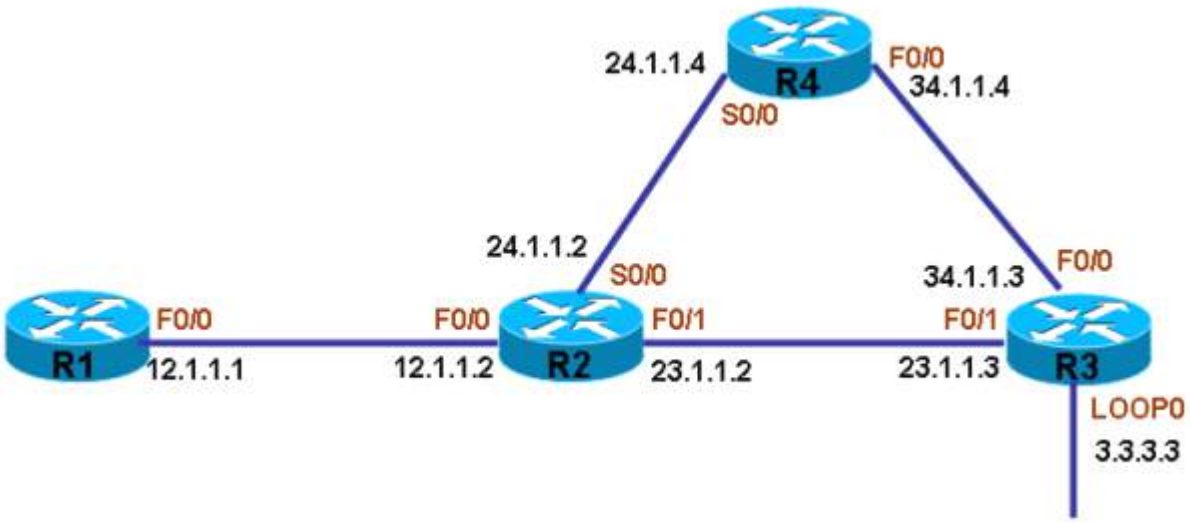
uRPF 所认为的 IP 伪造，是指某个 IP 的数据包的并不应该从某个接口进来，却从某个接口进来了，那么这样的数据包便认为是具有 IP 欺骗性质的，默认是被丢弃的。

当路由器从某个开启了 uRPF 的接口上收到数据包之后，都会检测该数据包的源 IP 地址，同时与路由表中的路由条目作对比，经过判断后，如果到达这个源 IP 的出口确实是这个开了 uRPF 的接口，则数据包被转发，否则被丢弃。比如路由器从接口 F0/0 收到一个源 IP 为 10.1.1.1 的数据包，那么就将 IP 地址 10.1.1.1 和路由表作对比，只要去往 10.1.1.1 的出口不是 F0/0（比如是 F0/1），那么该数据包被丢弃。

因为 uRPF 开启后，所有从此接口进入的数据包都要被检测，速度将会变慢，所以必须开启 CEF 后，才能开启 uRPF。uRPF 只能在 in 方向上开启，在做检查时，所有到源 IP 的最优路径都认为是可行的，EIGRP 非等价出口也算正常，并且即使是默认路由，也可通过检查。

在正常情况下，如果一个数据包无法通过 uRPF 检查，那么该数据包默认是丢弃的，但是有时因为特殊原因，可以让某些即使检查失败的数据包也能通过，要做到这一点，就可以在开启 uRPF 除加 ACL，其中检查失败的数据包，是丢弃还是放行，全由 ACL 来决定，ACL 允许，就放行，ACL 拒绝，就丢弃。这里的 ACL 和常用 ACL 一样配置，可以带 log 和 log-input 参数。

配置



说明：

R1 到任何网段的数据包都发向 12.1.1.2(即 R2)

R3 到任何网段的数据包都发向 34.1.1.4（即 R4）

R2 到 R3 的 loopback0 (3.3.3.3)都发向 23.1.1.3(即 R3)

1.确认网络路径

说明：先测试网络的路径走向

（1）查看 R3 的路由表

```
r3#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 34.1.1.4 to network 0.0.0.0

     34.0.0.0/24 is subnetted, 1 subnets
C       34.1.1.0 is directly connected, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
C       3.3.3.3 is directly connected, Loopback0
     23.0.0.0/24 is subnetted, 1 subnets
C       23.1.1.0 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 34.1.1.4
r3#
```

说明：从路由表中可以看出，R3 到任何网络都是发往 34.1.1.4(即 R4)的。

(2)从 R3 跟踪到 R1 的路径走向

r3#traceroute 12.1.1.1

Type escape sequence to abort.

Tracing the route to 12.1.1.1

1 34.1.1.4 0 msec 4 msec 0 msec

2 24.1.1.2 12 msec 12 msec 12 msec

3 12.1.1.1 12 msec * 8 msec

r3#

说明：从结果中看出，R3 到 R1，是先发往 R4，然后 R4 从 R2 的 s0/0 发过来，最后到 R1 的。

(3)查看 R2 到 R3 的 loopback0 (3.3.3.3)

r2#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-user static route

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

34.0.0.0/24 is subnetted, 1 subnets

S 34.1.1.0 [1/0] via 23.1.1.3

3.0.0.0/32 is subnetted, 1 subnets

S 3.3.3.3 [1/0] via 23.1.1.3

23.0.0.0/24 is subnetted, 1 subnets

C 23.1.1.0 is directly connected, FastEthernet0/1

24.0.0.0/24 is subnetted, 1 subnets

C 24.1.1.0 is directly connected, Serial0/0

12.0.0.0/24 is subnetted, 1 subnets

C 12.1.1.0 is directly connected, FastEthernet0/0

r2#

说明：从路由表中可以看出，R2 到 R3 的 loopback0 (3.3.3.3)是发往 23.1.1.3(即 R3)的。

（4）从 R2 跟踪到 loopback0 (3.3.3.3)的路径走向

r2#traceroute 3.3.3.3

Type escape sequence to abort.

Tracing the route to 3.3.3.3

1 23.1.1.3 0 msec * 0 msec

r2#

说明：可以看到，R2 到 R3 的 loopback0 (3.3.3.3)是直接从自己的 F0/1 发出去就到了。

（5）再看 R3 以 loopback0 (3.3.3.3)为源到 R1 的连通性

r3#ping 12.1.1.1 source loopback 0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

Packet sent with a source address of 3.3.3.3

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

r3#

说明：可以看到通信正常

（6） 再看 R3 以 loopback0 (3.3.3.3)为源到 R1 的路径

r3#traceroute 12.1.1.1 source loopback 0

Type escape sequence to abort.

Tracing the route to 12.1.1.1

1 34.1.1.4 0 msec 4 msec 0 msec

2 24.1.1.2 12 msec 12 msec 12 msec

3 12.1.1.1 12 msec * 8 msec

r3#

说明：在任何情况下到达 R1 都是同样的路径。

2.在 R2 上开启 uRPF

（1）在 R2 的 S0/0 上开启 uRPF

r2(config)#int s0/0

鸿鹄培训+考试群：301283867

r2(config-if)#ip verify unicast reverse-path

3.测试开启 uRPF 后的通信情况

（1）以 R3 的 loopback0 (3.3.3.3)为源，向 R1 发送数据

r3#ping 12.1.1.1 source loopback 0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

Packet sent with a source address of 3.3.3.3

.....

Success rate is 0 percent (0/5)

r3#

说明：可以看到，在 R2 的 S0/0 上开启 uRPF 之后，网络不通了。

4.查看 R2 的 uRPF 情况

r2#sh ip interface s0/0

IP verify source reachable-via RX, allow default

5 verification drops

0 suppressed verification drops

r2#

说明：可以看到，有 5 个数据包因为 uRPF 被丢弃，正是因为 R2 到 R3 的 loopback0 (3.3.3.3)是从 F0/1 出去的，所以以 R3 的 loopback0 (3.3.3.3)为源的数据包必须也从 F0/1 进来，所以从 S0/0 进来被 uRPF 检查失败，所以默认丢弃，如果不想丢弃，必须配置 ACL 允许。

5.以 R3 的 F0/0 为源，向 R1 发送数据

r3#ping 12.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r3#

说明：默认不指定源，即以 F0/0 发出数据，数据照样被丢弃，原因同上。

6.查看 R2 的 uRPF 情况

r2#sh ip interface s0/0

IP verify source reachable-via RX, allow default

10 verification drops

0 suppressed verification drops

r2#

说明：可以看到丢弃的数据包。

7.在 R2 上配置 ACL 允许以 R3 的 loopback0(3.3.3.3)为源的数据包即使 uRPF 检查失败也放行。

r2(config)#access-list 100 permit ip host 3.3.3.3 any

r2(config)#int s0/0

r2(config-if)#ip verify unicast reverse-path 100

8.测试以 R3 的 loopback0(3.3.3.3)为源的数据包通信情况

r3#ping 12.1.1.1 source loopback 0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

Packet sent with a source address of 3.3.3.3

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

r3#

说明：可以看到由于 uRPF 中的 ACL 允许此失败的数据包通过，所以网络正常。

9.再看以 R3 的 F0/0 为源，向 R1 发送数据的通信情况

r3#ping 12.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r3#

说明：因为 uRPF 中的 ACL 并没有允许以 R3 的 F0/0 为源数据检查失败也通过，所以数据包被丢弃。

AAA

概述

在 Cisco 设备中，许多接口，许多地方，为了安全，都需要开启认证服务，比如我们通常配置的 console 下的密码，进入 Privileged EXEC 模式的 enable 密码，VTY 线路下的密码，以及其它二层接口的认证密码等等。这些密码配置在哪里，那里就开启了相应的认证服务。并且这些认证服务方式是单一的，也没有备份认证方式，更重要的是，这些密码只能读取本地，却不可以通过读取远程服务器的认证方式来给自己提供认证服务。要想将一个接口的认证方式调用到另外一个接口，或者让某接口使用远程服务器的认证方式，那就需要 AAA 中的认证来实现。

AAA 中的认证可以给设备提供更多的认证方式，AAA 认证就相当于一个装有认证方式的容器一样，里面可以有多个认证方式，当这个容器被安装在某个接口，那么这个接口也就拥有了容器中所有的认证方式。而几乎所有的认证方式都可以被放入这个容器中，包含远程服务器的认证方式。

AAA 认证中，这个装有多个认证方式的容器，被称为列表，即 list，这个 list 加载到某个接口，那么这个接口就拥有 list 中所有的认证方式。List 中的多个认证方式是排列有序的，当前一个认证方式不可用时，才能使用第二个，以此类推。如果第一个可以使用，绝不会开启第二个，所以这样就提供了认证备份。那么何时 AAA 才认为第一个认证方式不可用呢？何时才使用第二个呢，这是当 AAA 询问一个认证方式时，如果认证数据库没有返回信息，即认证数据丢失或失败了，那么才使用下一个认证方式，如果某个认证方式是可用的，但用户提供了错误的用户名或密码，这样的情况是不会使用下一个认证方式的。

AAA 中的认证方式除了调用设备本地的认证方式之外，还可以调用远程服务器的认证方式，比如 radius、tacacs+。而一个 AAA list 中最多可以有四个认证方式，当第一个无法响应了，便使用第二个，第二个如果同样不响应，才使用第三个。但是如果前面的认证方式是在设备本地的，如本地用户数据库，当数据库中没有用户名时，IOS 可能并不认为是认证不响应，所以很难使用下一个认证方式，只有当前一个是远程服务器时，远程服务器不响应，才会使用下一个，这是 IOS 的不足之处。如果 list 中所有认证方式全部没有响应，那么 IOS 是不会让用户登陆的，除非最后有 none 指示放弃认证。

前面提到 AAA 的所有认证方式都应该放入 list 中，list 手动应用到某个接口上，这个接口就拥有了 list 中的认证方式，如果配置好的 list 没有应用到接口，那么是毫无意义的。一个 list 是有名字的，但是如果 list 的名字为 default，那么这个 list 不用手动应用到接口，因为默认情况下，名为 default 的 list 自动应用于所有相关接口，所以请谨慎创建名为 default 的 list。创建的 list，并不是所有接口都是可以用的，要看创建的是为什么样的接口使用的，比如是给登陆认证的，就要指定为 login，如果是给 dot1x 认证的，就要指定为 dot1x。所有支持 login 认证的方法有：

enable

krb5

krb5-telnet

line

local

local-case

none

group radius

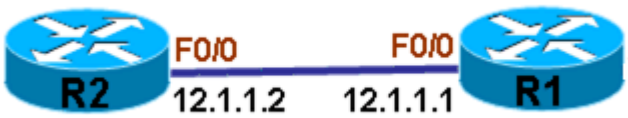
group tacacs+

group group-name

并且 login 中是不执行 autocommands 的。

AAA 的 list 可以调用多个认证方式，同样也可以调用远程服务器，比如 radius、tacacs+，不仅如此，还可以允许远程服务器有多个备份组，以便一台服务器坏了，另外一台可以继续顶替，要实现这样的服务器备份组，就必须配置 AAA 服务器组，在里面添加多台服务器的 IP 地址即可，并且配置了 AAA 的 Cisco 设备和远程服务器之间也可以应用密码认证。

配置：



1.开启 AAA

（1）开启 AAA

r1(config)#aaa new-model

（2）测试认证

r2#telnet 12.1.1.1

Trying 12.1.1.1 ... Open

User Access Verification

Username:

说明：当一台设备上开启 AAA 之后，所有 VTY 接口默认被加入 local 本地用户数据库认证。

2.为 VTY 使用 AAA 认证

（1）创建 list，并指定认证方法顺序为 locao 和 enable

r1(config)#aaa authentication login list1 local enable

说明：因为为 VTY 认证，所以认证关键字为 login

（2）在 VTY 使用 AAA 认证

r1(config)#line vty 0 935

r1(config-line)#login authentication list1

3.创建认证密码

（1）创建 enable 密码

r1(config)#enable secret cisco

4.测试认证

（1）测试使用的认证方法

r2#telnet 12.1.1.1

Trying 12.1.1.1 ... Open

User Access Verification

Username:

说明：在没有 local 认证的情况下，AAA 并没有跳到下一个 enable 认证，所以只有

远程服务器不响应，才认为认证为响应，才会使用下一个认证方式。

5.采用远程服务器认证

说明：将远程服务器认证做为第一个，当远程服务器不响应时，直接跳到下一个认证方式

（1）创建 AAA list，第一个为 tacacs+，第二个为 enable

r1(config)#aaa authentication login list2 group tacacs+ enable

r1(config)#line vty 0 935

r1(config-line)#login authentication list2

r1(config-line)#

（2）测试认证

r2#telnet 12.1.1.1

Trying 12.1.1.1 ... Open

User Access Verification

Password:

r1>

说明：在没有配置远程服务器时，AAA 认证认证无响应，所以切换到第二个认证方式 enable 认证。

6.测试更多认证

（1）配置第一个为 tacacs+，第二个为 enable，第三个为空，并且删除 enable 密码

r1(config)#no enable secret

r1(config)#aaa authentication login list3 group tacacs+ enable none

r1(config)#line vty 0 935

r1(config-line)#login authentication list3

r1(config-line)#

（2）测试认证

r2#telnet 12.1.1.1

Trying 12.1.1.1 ... Open

User Access Verification

Password:

说明：当不是远程服务器无响应的认证方式失败时，都不会跳到下一个认证方式。

7.配置远程服务器组：

（1）配置多个远程服务器

r1(config)#aaa group server tacacs+ ggg

r1(config-sg-tacacs+)#server 10.1.1.1

r1(config-sg-tacacs+)#server 20.1.1.1

r1(config-sg-tacacs+)#exit

（2）定义远程服务器密码

r1(config)#radius-server host 10.1.1.1 key cisco

8.更多提示

说明：AAA list 中，当第一项为服务器时，检测不可用，才会往后退，如果服务器后的认证方式为 local，并且下一项为 none 时，随便输入什么认证都无项，直接让用户登陆，所以请小心此类配置。

(1)配置第一项为服务器，第二项为 local，且紧跟 none

r1(config)#aaa authentication login list4 group tacacs+ local none

r1(config)#line vty 0 935

r1(config-line)#login authentication list4

(2)测试认证

r2#telnet 12.1.1.1

Trying 12.1.1.1 ... Open

User Access Verification

Username: abc

r1>

说明：可以看到，此类配置，随便输入任何认证，都为通过。

IP Source Tracker

概述

此功能是让路由器能够收集网络中可能正在遭受 DOS 攻击的主机，并且记录攻击源，创建必要的描述 DOS 攻击易用的信息，可以跟踪多个 IP。

记录日志的时间间隔是可以随意定义的，定义的最大主机数量也是可以定义的，并且这些信息全部可以输出到远程服务器，如 GRP 和 RSP，也只有高端系列 75，12000 才支持。

配置

1.配置跟踪的主机，可以配置多个主机。

r1(config)#ip source-track 100.10.0.1

2.配置产生日志的间隔，单位为分

r1(config)#ip source-track syslog-interval 2

3.阶段配置输出的时间间隔

r1(config)#ip source-track export-interval 60

4.配置最多记录的地址数量

r1(config)#ip source-track address-limit 3

Secure Shell (SSH)

概述

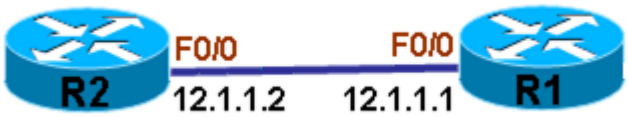
在对设备进行远程连接的方法中，最常用的是 telnet，而所有通过 telnet 会话传递的数据都是以明文（非加密）方式传递的，当这些数据被截取之后，很轻松就能读取原文意思，为了安全性，有一种在 telnet 会话之上的连接方法，将数据进行加密后传输，这就是 Secure Shell (SSH)。

SSH 共有两个版本，ver 1 和 ver 2，Cisco 设备在没有指定版本的情况下，默认就是指 ver 1。要支持 SSH，设备必须拥有支持 IPSec (DES or 3DES) 加密的 IOS，从 12.1(1)T 或之后都是可以的。除此之外，必须为设备配置主机名和域名，否则会报错。最关键的是必须配置 RSA key，配置之后，SSH 就自动打开了，否则不能开启。删除 RSA 使用命令 crypto key zeroize rsa，如果密码被删除，则表示 SSH 被禁用。

在 Cisco 设备上配置 SSH，SSH 分为 server 和 client 两种，server 就是提供 SSH 登陆的设备，而 client 就是发起 SSH 会话的设备，而 Cisco 设备无法单独开启 client，client 在配置 server 功能后自动开启，并且自身是不需要任何命令打开的，也没有特定命令能够打开 client。当 SSH 会话没有数据传递时，默认超时是 120 秒，即使是手工配置也不能超过这个值。并且 SSH 的最大连接数量就是 VTY 所允许的数量。

SSH 版本 2 的 RSA 至少是 768 位。

配置



1.配置双方主机名和域名

注： server 和 client 之间的域名是可以不一样的。

（1）配置 R1 的主机名和域名

router (config)#hostname r1

r1(config)#ip domain-name cisco.com

（2）配置 R2 的主机名和域名

router (config)#hostname r2

r2(config)#ip domain-name cisco.com

2.配置双方的 RSA key

注： 双方的 RSA key 位数必须一致。

(1)配置 R1 的 RSA key

r1(config)#crypto key generate rsa

The name for the keys will be: r1.cisco.com

Choose the size of the key modulus in the range of 360 to 2048 for your

General Purpose Keys. Choosing a key modulus greater than 512 may take

a few minutes.

How many bits in the modulus [512]: 1024

% Generating 1024 bit RSA keys ...[OK]

r1(config)#

(2)配置 R2 的 RSA key

r2(config)#crypto key generate rsa

The name for the keys will be: r2.cisco.com

Choose the size of the key modulus in the range of 360 to 2048 for your

General Purpose Keys. Choosing a key modulus greater than 512 may take

a few minutes.

How many bits in the modulus [512]: 1024

% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

*Mar 1 05:24:34.940: %SSH-5-ENABLED: SSH 1.99 has been enabled

r2(config)#

3.创建用户名和密码，client 通过此用户名和密码登陆

r1(config)#username ccie password cisco

4.在 VTY 下开启认证，并指定 SSH 可以登陆

r1(config)#line vty 0 181

r1(config-line)#login local

r1(config-line)#transport input ssh telnet

5.测试 SSH 登陆

r2#ssh -l ccie 10.1.1.1

Password:

r1>

说明：可以成功登陆

6.查看 SSH 版本：

r1#sh ip ssh

SSH Enabled - version 1.5

Authentication timeout: 120 secs; Authentication retries: 3

r1#

r2#sh ip ssh

SSH Enabled - version 1.99

Authentication timeout: 120 secs; Authentication retries: 3

r2#

说明：所有在 2.0 以下，都算是版本 1。

7.将 SSH 启用版本 2

（1）修改 SSH 为版本 2

r2(config)#ip ssh version 2

（2）查看版本：

r2#sh ip ssh

SSH Enabled - version 2.0

Authentication timeout: 120 secs; Authentication retries: 3

r2#

说明：已经改为版本 2。

注：版本 1 和版本 2 是可以互相登陆的。

8.指定源地址

说明：当使用 telnet 登陆远程时，可以指定源 IP 地址，在使用 SSH 时，需要在配置中修改源 IP 地址。

r2(config)#ip ssh source-interface Loopback0

说明：已将 SSH 的源 IP 地址改为 Loopback0 的地址。

9 同时开启两个 SSH 版本

R1 (config)# no ip ssh version

说明：加了此命令，表示 SSH 两个版本同时开启。

Intrusion Prevention System (IPS)

概述

在路由器连接了公司内网和外网的时候，通常希望路由器能够保护内网服务器抵御外网的攻击，这就需要配置路由器拒绝某些外网发向内网的可疑流量。但是，一个普通的网络操作人员，可能无法了解太多的可疑流量，也很难辨别什么样的流量算是可疑流量。基于以上原因，厂商就尽量将所有可疑流量的特征码写成一个文件，而管理员将此特征码文件导入路由器，让路由器根据特征码文件中的描述来拒

绝相应的流量，从而抵御外网的攻击和入侵。这就是 IPS（入侵防御系统）的工作，而厂商写的特征码文件，被称为 signatures（签名文件）。

SDF（签名定义文件）

signatures（签名文件）被称为 SDF，并且里面有描述攻击流量的条目数量。Cisco 设备上可用的 SDF 有两种：

1 默认的，也就是系统内置的签名（共 100 条）

2 厂商最新的，也就是使用路由器或 SDM 下载最新的签名文件

需要注意的是，如果要从厂商下载最新的 SDF，强烈建议使用 SDM 下载，如果从路由器下载，会有意想不到的问题。并且从路由器的 CLI 模式不能调整 IPS 的动作的，所以配置 IPS，建议使用 SDM。

如果要做二层透明 IPS，只支持以太网，并且签名是三层的，没有二层，组播也不支持。

厂商的签名文件现有以下三种：

attack-drop.sdf (83 条)，适用于内存 128MB 以下

128MB.sdf (300 条)，适用于内存 128 MB 或更多

256MB.sdf (500) ，适用于内存 256 MB 或更多。

这些文件可从 flash 直接加载进 IPS，但是 flash 被清除，SDF 也将清除，那么就使用内置的，

SDF 只能用于 12.4(9)Tx 或更早 IOS，主线 IOS 也支持。

系统内置的 SDF 是在 IOS 文件里面的，不需要下载。

IPS 检测所有经过自己的数据包，数据可以定义的，根据签名文件来判断，当检测到可疑流量时，可以先产生日志，或者发向 SDEE（也会有日志跳出），而产生的动作包含如下：

1 发送警报给日志服务器，或者集中管理器

2 丢包

3 重置连接

4 在规定时间内拒绝该源数据包

5 在规定时间内拒绝该连接数据

IPS 提供两种警报：syslog 和 SDEE，而 SDEE 其实是在运行的，但是并没处理 IPS 事件，除非开启 notification，并且要开 SDEE，必须开 http server。

在 IPS 检测流量时，也可以使用 ACL 要定义哪些流量需要检测，哪些不需要检测。命名和数字 ACL 都支持，但是 12.3(8)T，只支持标准数字 ACL。

当 IOS 中加载了 SDF 之后，要告诉设备从哪些接口检测进来还是出去的流量，都需要手工定义，可以两个方向同时定义。如果使用厂商 SDF，当在系统中出现错误时，默认使用内置 SDF，但可以禁止在厂商 SDF 失败时使用内置 SDF。

配置

1.安装内置 SDF

说明：因为有内置和厂商两种 SDF，如果要安装厂商的 SDF，也要先安装内置的。

（1）导入内置 SDF

```
r1(config)#ip ips sdf builtin
```

2.开启 IPS

（1）创建策略名

```
r1(config)#ip ips name ipp
```

（2）在接口上开启 IPS

```
r1(config)#int f0/0
r1(config-if)#ip ips ipp in
```

3.查看 IPS 结果

（1）查看 IPS 情况

r1#sh ip ips all

Configured SDF Locations:

flash:

Builtin signatures are enabled and loaded

Last successful SDF load time: 01:43:33 UTC Mar 1 2002

IPS fail closed is disabled

IPS deny-action ips-interface is false

Fastpath ips is enabled

Quick run mode is enabled

Event notification through syslog is enabled

Event notification through SDEE is disabled

Total Active Signatures: 135

Total Inactive Signatures: 0

Signature 50000:0 disable

Signature 50000:1 disable

Signature 50000:2 disable

Signature 1107:0 disable

IPS Rule Configuration

IPS name ipp

Interface Configuration

Interface FastEthernet0/0

Inbound IPS rule is ipp

Outgoing IPS rule is not set

r1#

说明：可以看到，已经加载了内置的 SDF，并且接口 F0/0 上也应用了进方向的 IPS。

4.将内置和厂商 SDF 结合使用

说明：可以同时使用两个 SDF，但内置必须先装载。

（1）加载厂商 SDF

R1# copy disk2:attack-drop.sdf ips-sdf

（2）保存为新的 SDF

R1# copy ips-sdf disk2:my-signatures.sdf

（3）加载新的 SDF

R1(config)# ip ips sdf location disk2:my-signatures.sdf

（4）在接口配置 IPS 的方法和之前一样

5.开启 SDEE 报警功能

（1）打开 SDEE

R1(config)# ip ips notify sdee

（2）配置条目数：最多 1000

R1(config)# ip sdee events 500

6.不加载内置 SDF

说明：在使用厂商 SDF 的情况下，默认失败后直接使用内置的，但可以在失败时也不使用内置 SDF

（1）关闭使用内置 SDF

R1(config)# no ip ips location in builtin

7.关闭丢包功能

说明：在 IPS 没有正常加载 SDF 时，可以选择丢弃所有数据包

R1(config)# ip ips fail closed

Zone-Based Policy Firewall

概述

在 Cisco IOS 版的路由器中，可以实施一种防火墙功能，被称为 Zone-Based Policy Firewall，也就是说这种防火墙是基于 zone 的，是基于区域的。既然是基于区域的防火墙，那么配置的防火墙策略都是在数据从一个区域发到另外一个区域时才生效，在同一个区域内的数据是不会应用任何策略的。而要配置这些策略，方法像使用 MQC 来配置 QOS 一样配置防火墙策略，但是两个的配置方法并不完全一致，因为双方的格式会有一些不同。

要完全理解 Zone-Based Policy Firewall 的工作，必须理解以下一些参数：

Zone

Zone 就是区域，因为区域化防火墙是基于区域的，策略也只能在区域间传递数据时才生效，在区域内是不生效的，所以我们就可以将需要使用策略的接口划入不同的区域，这样就可以应用我们想要的策略。但是，有时某些接口之间可能不需要彼此使用策略，那么这样的接口只要划入同一个区域，它们之间就可以任意互访了。Zone 是应用防火墙策略的最小单位，一个 zone 中可以包含一个接口，也可以包含多个接口。

Security Zones

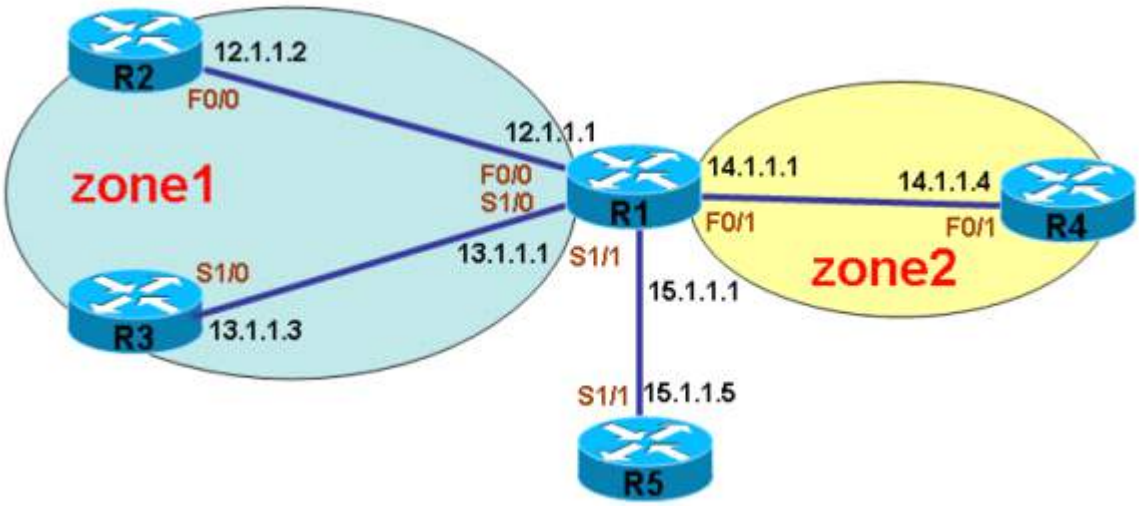
Security Zones 和上面的 zone 并没有区别，意思完全相同，只是因为 Security Zones 是指应用了策略的 zone，而且 Security Zones 应该是要包含接口的。

在配置好 Security Zones 之后，当一个接口属于 Security Zones 的成员时，所有发到此接口和从此接口发出去的（但发到路由器和路由器发给它的除外）数据是默认被丢弃的，也就是说区域间的接口默认是不能通信的，数据将全部被丢弃。

Virtual Interfaces 也可以作为 Security Zones 的成员，在将 Virtual Interfaces 划入 Security Zones 时，使用 interface virtual-template。

特别的是，如果一个接口不属于任何一个 zone，那么这个接口永远也不可以和任何 zone 的任何接口通信。

以下图解释通信过程：



R2 和 R3 属于区域 zone1

R4 属于区域 zone2

R5 不属于任何区域

默认的通信权限为：

R2 和 R3 可以自由通信，

Zone1 的 R2 和 R3 不能和 zone2 的 R4 互访，因为双方属于不区 zone，必须明确

配置策略允许。

而 R5 永远不能和任何路由器通信，因为它不属于任何区域，即使策略也做不到。

Zone-Pairs

因为在 Security Zones 之间的所有数据默认是全部被丢弃的，所以必须配置相应的策略来允许某些数据的通过。要注意，同区域的接口是不需要配置策略的，因为他们默认就是可以自由访问的，我们只需要在区域与区域之间配置策略，而配置这样的区域与区域之间的策略，必须定义从哪个区域到哪个区域，即必须配置方向，比如配置从 Zone1 到 Zone2 的数据全部被放行。可以看出，Zone1 是源区域，Zone2 是目的区域。配置一个包含源区域和目的区域的一组策略，这样的一个区域组，被称为 Zone-Pairs。因此可以看出，一个 Zone-Pairs，就表示了从一个区域到另一个区域的策略，而配置一个区域到另一个区域的策略，就必须配置一个 Zone-Pairs，并加入策略。

当配置了一个区域到另一个区域的策略后，如果策略动作是 inspect，则并不需要再为返回的数据配置策略，因为返回的数据是默认被允许的。

如果有两个 zone，并且希望在两个方向上都应用策略，比如 zone1 到 zone2 或 zone2 到 zone1,就必须配置两个 zone-pairs ,就是每个方向一个 zone-pairs。

有时可以将 self zone 即作源又作目的。self zone 是 system-defined zone，即系统定义的 zone，它是没有接口的。当 zone-pair 包含 self zone 时，被应用的策略只对发到路由器和路由器发出的数据生效，通过路由器中转的数据不生效。

路由器上最好有两个 zone 来做策略，其中不包含 self zone。

策略

当接口被划入不同的 zone 之后，想要相互通信，就必须配置策略，再将配置好的策略应用于 Zone-Pairs，因为一个 Zone-Pairs 就表示了一个区域到另一个区域的策略情况。

在为 zone 之间配置策略，使用的方法类似于用 MQC 配置 QOS，但格式会有略微的差异。配置策略的方法为先使用 Class Map 匹配出指定的数据，然后再利用 Policy Map 调用 Class Map 匹配到的数据，做出相应的策略动作，最终将 Policy Map

应用于 Zone-Pairs。下面分别针对 Zone-Based Policy Firewall 中的 Class Map 和 Policy Map 来做详细介绍。

3/4 层 Class Maps 和 Policy Maps

因为 Class Map 可以匹配 OSI 中第三层数据和第四层数据，也可以匹配第七层应用层数据，但是匹配三四层数据和匹配第七层数据是完全不一样的，我们把匹配第三四层数据的 Class Maps 和 Policy Maps 称为顶级 Class Maps 和顶级 Policy Maps。他们与普通 MQC 中的 Class Map 和 Policy Map 的区别在于，顶级 Class Maps 和顶级 Policy Maps 分别表示为 inspect class maps 和 inspect policy maps，而除了顶级 Class Maps 和顶级 Policy Maps 可以用在 zone-pair 中，其它统统不可以应用于 zone-pair 中。

顶级 Policy Maps 只能调用顶级 Class Maps，并且能执行的动作只有：drop, inspect, police, pass, service-policy, and urlfilter。而顶级 Class Maps 只能匹配 OSI 第三层数据和第四层数据，无法匹配第七层数据。

MQC 的 Police Maps 是在接口的，而 inspect policy Maps 是对 zone-pair 的，如果两个都配，zone-pair policer 是在接口进方向策略之后的，但在出策略之前。但两不冲突。

如果在 inspect policy Maps 中，默认没有被匹配到的数据是被丢弃。

7 层 Class Maps 和 Policy Maps

7 层 class maps 只能应用于 7 层 Policy Maps，而 7 层 Policy Maps 也只能调用 7 层 class maps,并且 7 层 Policy Maps 不能直接应用于 zone-pair 中，只能嵌套于 3/4 层 Policy Maps 中。如果 7 层 Policy Maps 嵌套在 3/4 层 Policy Maps 中，那么 3/4 层 Policy Maps 称为 parent policy，而 7 层 Policy Maps 称为 child policy。

7 层 Class Maps 和 Policy Maps 在配置时，必须指定协议名，比如 HTTP 协议，就配置 Class Maps 为 class-map type inspect http，Policy Maps 为 policy-map type inspect http。

当 7 层 Policy Maps 没有匹配到的数据，默认是要返回让顶层 Policy Maps 来处理的。

Parameter Maps

是用来定义动作和标准的，分别用在 policy map and 和 class map 中，有三种：

Inspect parameter map

URL Filter parameter map

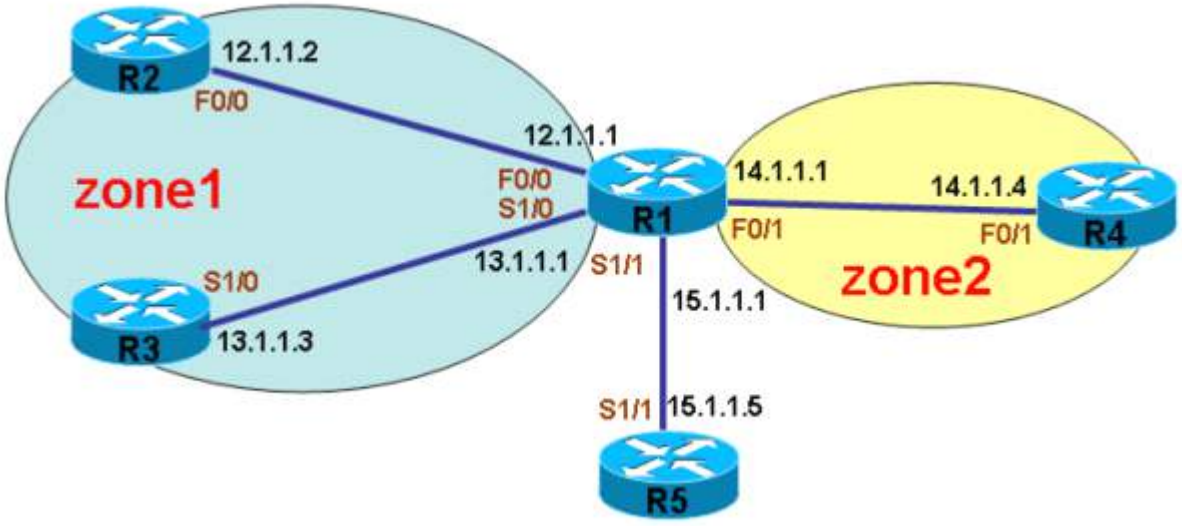
Protocol-specific parameter map

Inspect parameter map 是可选的，如果两级都有，低等级的有效。

URL Filter parameter map 在 URL 过滤时需要，在 34 层 policy MAP 中

Protocol-specific parameter map 只有 7 层 policy map 需要。

配置



1.测试默认通信：

说明：在没有配置防火墙的情况下，测试通信情况

（1）测试 R2 到 R3、R4、R5 的 ICMP 通信情况：

r2#ping 13.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 72/144/244 ms

r2#ping 15.1.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 15.1.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 72/147/368 ms

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/157/352 ms

r2#

说明：可以看到，默认没有配置防火墙的情况下，ICMP 畅通无阻。

（2）测试 R2 到 R4 的 telnet 情况：

r2#telnet 14.1.1.4

Trying 14.1.1.4 ... Open

r4>

说明：可以看到，默认没有配置防火墙的情况下，telnet 畅通无阻。

2.创建 security zone

（1）创建 zone1

r1(config)#zone security zone1

r1(config-sec-zone)#

（2）创建 zone2

r1(config)#zone security zone2

r1(config-sec-zone)#exi

3.将接口划入 zone

(1)将连 R2 和 R3 的接口划入 zone1

r1(config)#interface f0/0

r1(config-if)#zone-member security zone1

r1(config-if)#exit

r1(config)#int s1/0

r1(config-if)#zone-member security zone1

r1(config-if)#exit

(2)将连 R4 的接口划入 zone2

r1(config)#int f0/1

r1(config-if)#zone-member security zone2

r1(config-if)#exit

（3）查看结果

r1#sh zone security

zone self

Description: System defined zone

zone zone1

Member Interfaces:

FastEthernet0/0

Serial1/0

zone zone2

Member Interfaces:

FastEthernet0/1

r1#

说明：结果与配置一致。

4.测试通信

（1）测试 zone1 同区域的通信情况

r2#ping 13.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 56/188/408 ms

r2#

说明：在没有配置策略的情况下，同区域的通信不受限制

（2）测试不通区域的通信情况

r2#ping 15.1.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 15.1.1.5, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：从结果中看出，从 zone1 到 zone2，再到没有区域的网段，都是不通的。

5.创建 zone-pair

说明：创建 zone1 为源， zone2 为目的的 zone-pair

r1(config)#zone-pair security ccie source zone1 destination zone2

r1(config-sec-zone-pair)#exit

r1(config)#

6.配置策略

（1）配置 class-map 匹配 zone1 到 zone2 的流量

r1(config)#access-list 100 permit ip any host 14.1.1.4

r1(config)#class-map type inspect c1

鸿鹄培训+考试群：301283867

r1(config-cmap)#match access-group 100

说明：7 层 class-map 能匹配的协议很少，所以用 3/4 层

（2）配置 policy-map 允许 zone1 到 zone2 的流量

r1(config)#policy-map type inspect p11

r1(config-pmap)#class type inspect c1

r1(config-pmap-c)#pass

r1(config-pmap-c)#exit

说明：动作 pass 是不会创建返回流量的。

7.应用策略到 zone-pair

r1(config)#zone-pair security ccie source zone1 destination zone2

r1(config-sec-zone-pair)#service-policy type inspect ppp

r1(config-sec-zone-pair)#

8.测试通信情况

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：已经应用了策略，zone1 到 zone2 还是不通，因为动作 pass 没有创建返回的流量。

9.创建 zone2 到 zone1 返回流量

鸿鹄培训+考试群：301283867

（1）配置 class-map 匹配流量

r1(config)#access-list 101 permit ip any any

r1(config)#class-map type inspect c2

r1(config-cmap)#match access-group 101

r1(config-cmap)#exit

（2）配置 policy-map 允许流量

r1(config)#policy-map type inspect p22

r1(config-pmap)#class type inspect c2

r1(config-pmap-c)#pass

r1(config-pmap-c)#exit

（3）应用策略到返回流量

r1(config)#zone-pair security ccsp source zone2 destination zone1

r1(config-sec-zone-pair)#service-policy type inspect p22

10.测试 zone1 到 zone2 通信情况

（1）测试 R2 到 R4 的 ICMP 通信情况

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 36/112/360 ms

r2#

说明：ICMP 通信正常。

（2）测试 R2 到 R4 的 telnet 通信情况

r2#telnet 14.1.1.4

Trying 14.1.1.4 ... Open

r4>

(3)zone1 到非 zone 的通信情况

r2#ping 15.1.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 15.1.1.5, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：区域中和非区域是永远也不能通信的。

11.配置 3/4 层 class-map 直接匹配协议

说明：使用 3/4 层 class-map 直接匹配 telnet 协议，其它协议不管

（1）配置 class-map 匹配 telnet 协议

r1(config)#class-map type inspect c3

r1(config-cmap)#match protocol telnet

r1(config-cmap)#exit

（2）配置 policy-map 允许 telnet 协议

r1(config)#policy-map type inspect p33

r1(config-pmap)#class c3

r1(config-pmap-c)#inspect

r1(config-pmap-c)#

说明：这里的动作是 inspect，所以不用创建返回流量的允许策略。

（3）应用策略到 zone-pair

r1(config)#zone-pair security ccie source zone1 destination zone2

r1(config-sec-zone-pair)#service-policy type inspect p33

r1(config-sec-zone-pair)#exit

r1(config)#

12.测试通信情况

（1）测试 zone1 到 zone2 的 ICMP 通信情况

r2#ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：因为没有允许除 telnet 外的协议，所以 ICMP 不能通过。

（2）测试 telnet 通信情况

r2#telnet 14.1.1.4

Trying 14.1.1.4 ... Open

r4>

说明：因为策略明确允许 telnet 通过，所以 telnet 通信正常。

(3)测试到同区域的 ICMP

r2#ping 13.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 156/224/352 ms

r2#

说明：同区域中，不需要策略允许，所有通信正常。

（4）测试到非区域的通信情况

r4#telnet 12.1.1.2

Trying 12.1.1.2 ...

% Connection timed out; remote host not responding

r4#

说明：区域到不同区域的通信永远不能通过。

Control Plane Policing (CoPP)

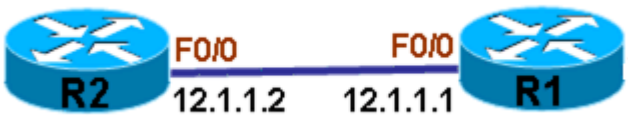
概述

Control Plane Policing (CoPP)被称为控制面板策略，控制面板策略这个特性让用户通过配置 QOS 过滤来管理控制面板中的数据包，从而保护路由器和交换机免受 DOS 的攻击，控制面板可以无论在流量多大的情况下都能管理数据包交换和协

议的状态情况。

在控制面板中，只能通过 MQC 配置常规的 QOS，并且 out 方向的 QOS 并不是所有 IOS 都支持，请自行检查，其中配置的 QOS 策略中，只有 drop 和 policy 两个动作可以使用。而且 NBAR 功能也不能很好的支持。当在控制面板中配置 QOS 后，不用在接口下应用该策略，因为控制面板下的策略对所有接口生效。

配置



1.测试 R2 到 R1 的数据流量

r2#ping 12.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：R1 在没有使用任何 QOS 的情况下，通信正常

2.配置 QOS

说明：这里配置 QOS 丢弃所有的包，以作测试用。

（1）配置匹配源自 R2 的数据

r1(config)#access-list 2 permit 12.1.1.2

r1(config)#class-map r2

鸿鹄培训+考试群：301283867

r1(config-cmap)#match access-group 2

r1(config-cmap)#exit

（2）配置丢弃源自 R2 的数据

r1(config)#policy-map copp

r1(config-pmap)#class r2

r1(config-pmap-c)#drop

3.将 QOS 应用于 COPP

r1(config)#control-plane

r1(config-cp)#service-policy input copp

4.测试到 R1 的通信

r2#ping 12.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：可以看到，并没有将所配置的 QOS 用于接口，只用在了控制面板下，所有接口都执行控制面板下的 QOS 策略，从而数据包被丢弃了，网络不通。

Switching

（提示：由于内容较多，阅读时，建议开启 文档结构图.）

目录

鸿鹄培训+考试群：301283867

交换机接口配置

因为交换机的特殊性，通常存在多个接口需要做相同的配置，如将多个接口划入相同 VLAN，这时就需要一种能够快速配置接口的方法。

对于 2 层交换机，所有的接口只能工作在二层，而对于三层交换机，接口除了可以工作在二层之外，并且还可以工作在三层，也就是说三层交换机的接口还可以配置 IP 地址，等同于路由器的接口。

配置

快速配置接口

1.快速对多个连续的接口做相同配置

（1）快速进入多个接口

Switch(config)#interface range f0/1-3

说明：同时进入接口 F0/1，F0/2，F0/3。

（2）配置接口参数

Switch(config)#interface range f0/1-3

Switch(config-if-range)#description ccie

说明：当同时进入多个接口后，所做的配置将对所有进入的接口生效。

（3）查看结果

```
Switch#sh run | b inter
vlan internal allocation policy ascending
!
!
interface FastEthernet0/1
 description ccie
!
interface FastEthernet0/2
 description ccie
!
interface FastEthernet0/3
 description ccie
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
```

说明：可以看到配置文件中，之前的配置对 F0/1，F0/2，F0/3 生效，其它没有进入的接口配置保存原状。

2.快速对多个不连续的接口做相同配置

（1）快速进入多个不连续接口

Switch(config)#interface range f0/1 - 2 , f0/4 , f0/6 - 7

说明：同时进入接口 F0/1，F0/2，F0/4，F0/6，F0/7。

在配置多个不连续接口时，请注意在连字符 - 前后都加上空格，这样可以保证在任何 IOS 版本中输入有效。

（2）配置接口参数

Switch(config)#interface range f0/1 - 2 , f0/4 , f0/6 - 7

Switch(config-if-range)#description cisco

说明：当同时进入多个接口后，所做的配置将对所有进入的接口生效。

（3）查看结果

Switch#sh run | b inter

vlan internal allocation policy ascending

!

!

interface FastEthernet0/1

description cisco

!

interface FastEthernet0/2

description cisco

!

```
interface FastEthernet0/3
!
interface FastEthernet0/4
 description cisco
!
interface FastEthernet0/5
!
interface FastEthernet0/6
 description cisco
!
interface FastEthernet0/7
 description cisco
!
interface FastEthernet0/8
!
```

说明：可以看到配置文件中，之前的配置对 F0/1，F0/2，F0/4，F0/6，F0/7 生效，其它没有进入的接口配置保存原状。

3.接口宏（macro）定义

说明：当在交换机的日常管理中，可能需要多次对多个非连续的接口进行管理和配置，而当需要进入多个非连续接口时，输入的命令较为烦琐，所以为了方便，系统允许人工将多个接口定义成一个组，这个组成为宏（macro），当进入这个宏（macro），就等于进入了组中的所有接口，对宏（macro）做出的配置，将对组中的所有接口生效，即便交换机重启后，所定义的宏（macro）仍旧存在，可以多次使用，直到手工删除。

（1）定义宏（macro）

Switch(config)#define interface-range ccie f0/1 - 2 , f0/4 , f0/6 - 7

说明：将接口 F0/1，F0/2，F0/4，F0/6，F0/7 放入了宏 ccie 中，对宏 ccie 所做的配置将对 F0/1，F0/2，F0/4，F0/6，F0/7 生效。

（2）配置接口参数

Switch(config)#interface range macro ccie

Switch(config-if-range)#description abc

说明：对宏 ccie 所做的配置将对 F0/1，F0/2，F0/4，F0/6，F0/7 生效。

（3）查看结果

Switch#sh run | b inter

vlan internal allocation policy ascending

!

!

interface FastEthernet0/1

description abc

!

interface FastEthernet0/2

description abc

!

interface FastEthernet0/3

!

interface FastEthernet0/4

```
 description abc
!
interface FastEthernet0/5
!
interface FastEthernet0/6
 description abc
!
interface FastEthernet0/7
 description abc
!
interface FastEthernet0/8
!
```

说明：可以看到配置文件中，对宏 ccie 所做的配置将对 F0/1，F0/2，F0/4，F0/6，F0/7 生效，其它接口配置保存原状。

4.配置 SVI（switch virtual interface）接口

说明：三层交换机的物理接口既可以配置为 2 层接口，也可以配置为三层接口，对于这些配置，此文档将跳过不作详细介绍。

除了交换机物理接口外，交换机还可以将某个 VLAN 配置为 3 层接口，称为 SVI（switch virtual interface）接口，将 VLAN 配置为 3 层接口的作用在于为 VLAN 内的流量与外部流量提供 3 层路由转发功能，该 VLAN 内所有的流量都应该在同网段，而该 VLAN 的主机网关都应该指向 SVI 接口地址。此时的 SVI 接口，其实也等同于路由器的接口，但是 SVI 接口也只有在状态都为 UP 的时候，才能提供路由功能，一个状态为 down 的 SVI 接口是不能发送数据包的。要将 SVI 接口激活并且变成 UP 状态，必须将一个活动的物理接口划入该 VLAN，当某 VLAN 中没有活动物理接口时，该 VLAN 的 SVI 接口永远将处于 down 状态而不能转发数据。需要大家注

意的是，一个 Trunk 接口允许某个 VLAN 通过，就表示该 Trunk 接口属于该 VLAN，也就是说某个 VLAN 被一个活动的 Trunk 接口允许通过时，那么就说明该 VLAN 中存在活动的物理接口，因此，该 VLAN 的 SVI 接口可以变成 UP 状态，也就可以转发数据包。

（1）创建 SVI 接口，并配置 IP 地址

Switch(config)#vlan 2

Switch(config-vlan)#exit

Switch(config)#int vlan 2

Switch(config-if)#ip add 2.2.2.2 255.255.255.0

说明：创建 SVI 接口时，必须保证该 VLAN 已经在交换机上存在。

（2）查看状态

查看 VLAN：

Switch#sh vlan

VLAN Name Status Ports

---- ----------------------- -------- -------------------------------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/23, Fa0/24

Gi0/1, Gi0/2

2 VLAN0002 active

1002 fddi-default act/unsup

说明：可以看到，VLAN 2 中没有活动的物理接口存在。

查看 Trunk 所允许的 VLAN:

Switch#sh interfaces trunk

Switch#

说明：可以看到，交换机上没有任何 Trunk 接口，也表示 VLAN 2 中没有活动的物理接口存在。

查看 SVI 接口状态:

Switch#sh protocols vlan 2

Vlan2 is up, line protocol is down

Internet address is 2.2.2.2/24

Switch#

说明：和预期的一样，因为 VLAN 2 中没有任何活动物理接口，所以接口状态为 down，并不能提供数据转发。

（3）激活 SVI 接口

Switch(config)#int f0/1

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 2

Switch(config-if)#no shutdown

说明：将物理接口 f0/1 划入 VLAN 2，只要 f0/1 状态为 UP，则 VLAN 2 的 SVI 接口便能变为 UP。

（4）再次查看状态

查看 VLAN：

Switch#sh vlan

VLAN Name Status Ports

---- -------------------------------- --------- -------------------------------

1 default active Fa0/2, Fa0/3, Fa0/4, Fa0/5

Fa0/6, Fa0/7, Fa0/8, Fa0/9

Fa0/10, Fa0/11, Fa0/12, Fa0/13

Fa0/14, Fa0/15, Fa0/16, Fa0/17

Fa0/18, Fa0/19, Fa0/20, Fa0/21

Fa0/22, Fa0/23, Fa0/24, Gi0/1

Gi0/2

2 VLAN0002 active Fa0/1

说明：VLAN 2 中存在物理接口 F0/1。

查看物理接口 F0/1 的状态：

Switch#sh protocols f0/1

FastEthernet0/1 is up, line protocol is up

Switch#

说明：接口 F0/1 的状态为 UP。

Switch#sh protocols vlan 2

Vlan2 is up, line protocol is up

Internet address is 2.2.2.2/24

Switch#

说明：因为 VLAN 2 中存在活动的物理接口 F0/1，所以 VLAN 2 的 SVI 接口状态变成了 UP，并且能够提供数据转发。

（5）通过 Trunk 允许 VLAN 来控制 SVI 接口状态

Switch(config)#vlan 3

Switch(config-vlan)#exit

Switch(config)#vlan 4

Switch(config-vlan)#exit

Switch(config)#int vlan 3

Switch(config-if)#ip add 3.3.3.3 255.255.255.0

Switch(config-if)#exit

Switch(config)#int vlan 4

Switch(config-if)#ip add 4.4.4.4 255.255.255.0

Switch(config-if)#exit

Switch(config)#

说明：创建了 VLAN 3 和 VLAN4，并同时创建了相应的 SVI 接口。

查看 VLAN:

Switch#sh vlan

VLAN Name Status Ports

---- -------------------------- -------- -------------------------------

1 default active Fa0/2, Fa0/3, Fa0/4, Fa0/5

Fa0/6, Fa0/7, Fa0/8, Fa0/9

Fa0/10, Fa0/11, Fa0/12, Fa0/13

Fa0/14, Fa0/15, Fa0/16, Fa0/17

Fa0/18, Fa0/19, Fa0/20, Fa0/21

Fa0/22, Fa0/23, Fa0/24, Gi0/1

Gi0/2

2 VLAN0002 active Fa0/1

3 VLAN0003 active

4 VLAN0004 active

说明：VLAN 3 和 VLAN4 中没有任何物理接口。

查看 Trunk

Switch#sh int trunk

Switch#

说明：交换机上也没有任何 Trunk 接口。

查看 SVI 接口:

Switch#sh prot

Switch#sh protocols vlan 3

Vlan3 is up, line protocol is down

  Internet address is 3.3.3.3/24

Switch#sh protocols vlan 4

Vlan4 is up, line protocol is down

  Internet address is 4.4.4.4/24

Switch#

说明：可以看见，由于 VLAN3 和 VLAN4 中没有任何活动物理接口，所以 SVI 接口都为 down 状态。

激活 VLAN 3 的 SVI 接口:

Switch(config)#int f0/23

Switch(config-if)#switchport trunk encapsulation dot1q

Switch(config-if)#switchport mode trunk

Switch(config-if)#switchport trunk allowed vlan 3

Switch(config-if)#no shutdown

查看 Trunk:

Switch#sh interfaces trunk

Port	Mode	Encapsulation	Status	Native vlan
Fa0/23	on	802.1q	trunking	1

Port	Vlans allowed on trunk
Fa0/23	3

Port	Vlans allowed and active in management domain
Fa0/23	3

Port	Vlans in spanning tree forwarding state and not pruned
Fa0/23	3

Switch#

说明：可以看到 Trunk 接口 F0/23 允许 VLAN3 通过。

查看 SVI 接口:

Switch#sh protocols vlan 3

Vlan3 is up, line protocol is up

Internet address is 3.3.3.3/24

Switch#sh protocols vlan 4

Vlan4 is up, line protocol is down

Internet address is 4.4.4.4/24

Switch#

说明：因为 Trunk 允许 VLAN 3 通过，所以 VLAN 3 的 SVI 接口状态已变为 UP，而 VLAN 4 则仍旧为 down。

Switch(config)#int f0/23

Switch(config-if)#switchport trunk allowed vlan 3,4

Switch#sh interfaces trunk

Port	Mode	Encapsulation	Status	Native vlan
Fa0/23	on	802.1q	trunking	1

Port	Vlans allowed on trunk
Fa0/23	3-4

Port	Vlans allowed and active in management domain
Fa0/23	3-4

Port Vlans in spanning tree forwarding state and not pruned

Fa0/23 3-4

说明：可以看到 Trunk 接口 F0/23 允许 VLAN3 和 VLAN 4 通过。

Switch#sh protocols vlan 3

Vlan3 is up, line protocol is up

Internet address is 3.3.3.3/24

Switch#sh protocols vlan 4

Vlan4 is up, line protocol is up

Internet address is 4.4.4.4/24

Switch#

说明：因为 Trunk 允许 VLAN 3 和 VLAN4 通过，所以 VLAN 3 和 VLAN4 的 SVI 接口状态都已变为 UP。

交换机 MAC 地址表

交换机在转发数据时，需要根据 MAC 地址表来做出相应转发，如果目标主机的 MAC 地址不在表中，交换机将收到的数据包在所有活动接口上广播发送。当交换机上的接口状态变成 UP 之后，将动态从该接口上学习 MAC 地址，并且将学习到的 MAC 地址与接口相对应后放入 MAC 地址表。

交换机的 MAC 地址表除了动态学习之外，还可以静态手工指定，并且在指定 MAC 地址时，还可以指定在某个 VLAN 的某个接口收到相应的 MAC 后，将数据包作丢弃处理。

注：交换机上，一个接口可以对应多个 MAC 地址，地址的数量无上限，但不超过交换机所支持的 MAC 地址最大数量。

一个 MAC 地址可以同时出现在交换机的多个接口上，但此特性并不被所有型号的交换机支持，在某些型号的交换机上，一个 MAC 地址只能出现在一个接口上，如果出现在另外一个接口上，将会报错，并且数据转发也会出错。

1.查看交换机 MAC 地址表

（1）查看接口 F0/1 的 MAC 地址表

```
Switch#sh mac-address-table interface f0/1

          Mac Address Table

-----------------------------------


Vlan    Mac Address       Type        Ports

----    -----------       --------    -----

   2    0013.1a2f.0680    DYNAMIC     Fa0/1

Total Mac Addresses for this criterion: 1

Switch#
```

说明：交换机从 F0/1 上学习到了 MAC 地址 0013.1a2f.0680，并且说明是动态学习到的。

2.手工静态指定 MAC 地址

（1）手工静态指定 MAC 地址

```
Switch(config)#mac-address-table static 0013.1a2f.0680 vlan 1 interface f0/2
```

说明：指定 VLAN 1 的接口 F0/2 的 MAC 地址为 0013.1a2f.0680。

（2）查看接口 F0/2 的 MAC 地址表

```
Switch#sh mac-address-table interface f0/2
          Mac Address Table
-----------------------------------


Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0013.1a2f.0680    STATIC      Fa0/2
   1    0013.1a7f.a4a0    DYNAMIC     Fa0/2
Total Mac Addresses for this criterion: 2
Switch#
```

说明：接口 F0/2 上除了动态学习到的 MAC 地址之外，还有静态手工指定的地址。

（3）指定丢弃某个 MAC 地址

```
Switch(config)#mac-address-table static 0013.1a2f.0680 vlan 2 drop
```

说明：此配置将使源 MAC 为 0013.1a2f.0680 的数据包在 VLAN 2 被丢弃，但在别的 VLAN 通信正常。

3.MAC 地址老化时间（aging-time）

交换机在一个接口上学习到 MAC 地址之后，该 MAC 与接口的映射并不会永远被保存在 MAC 地址表中，除非是手工静态指定的。当一台主机从某个接口转移后，

交换机再将目标 MAC 为该主机的数据从该接口发出去是毫无意义的，所以 MAC 地址在 MAC 地址表中是有最大停留时间的，称为老化时间（aging-time），当相应 MAC 地址在超出老化时间后还没有数据传输时，该 MAC 地址将从表中被清除。默认的 MAC 地址老化时间为 300 秒（5 分钟）。

（1）修改 MAC 地址的老化时间

说明：只能针对 VLAN 作修改

Switch(config)#mac-address-table aging-time 60 vlan 1

说明：将 VLAN 1 的 MAC 地址老化时间改为 60 秒。

（2）查看 MAC 地址的老化时间

Switch#sh mac-address-table aging-time

Global Aging Time: 300

Vlan Aging Time

--- --------

1 60

2 300

3 300

4 300

Switch#

说明：可以看到，VLAN 1 的 MAC 地址老化时间为 60 秒，其它 VLAN 保存默认 300 秒。

交换机自身 MAC 地址

以太网中，每一个节点，都需要一个 MAC 地址，而以太网交换机可以与多个

终端连接，也就有多个节点，因此，交换机上也会有多个 MAC 地址存在，如交换机的每个接口都有一个 MAC 地址，包含物理接口和 SVI 接口。除此之外，还有一个 MAC 地址是用来表示整台交换机的。

注：都知道 2 层交换机的 VLAN 1 为管理 VLAN，一个表示整台交换机的 MAC 地址通常就是 VLAN 1 的 MAC 地址，但这种情况又需要根据交换机型号而定，并不适用于任何型号的交换机。

某些型号的交换机，所有 VLAN 的 SVI 接口 MAC 地址全部相同，但某些型号却是不同的，但是连续的。

1.查看交换机的 MAC 地址

（1）查看表示整台交换机的 MAC 地址

Switch#sh version

（输出被省略）

512K bytes of flash-simulated non-volatile configuration memory.

Base ethernet MAC Address : 00:1A:6C:6F:FB:00

Motherboard assembly number : 73-9897-06

Power supply part number : 341-0097-02

Motherboard serial number : CAT10475C57

Power supply serial number : AZS104407JE

Model revision number : D0

Motherboard revision number : A0

Model number : WS-C3560-24TS-S

System serial number : CAT1047RJNU

Top Assembly Part Number : 800-26160-02

Top Assembly Revision Number : C0

Version ID : V02

CLEI Code Number : COMMG00ARB

Hardware Board Revision Number : 0x01

Switch Ports Model SW Version SW Image

------ ----- ----- --------- ---------

* 1 26 WS-C3560-24TS 12.2(35)SE1 C3560-ADVIPSERVICESK

Configuration register is 0xF

Switch#

说明：表示整台交换机的 MAC 地址为 00:1A:6C:6F:FB:00。

（2）查看物理接口的 MAC 地址

Switch#sh int f0/1

FastEthernet0/1 is up, line protocol is up (connected)

Hardware is Fast Ethernet, address is 001a.6c6f.fb03 (bia 001a.6c6f.fb03)

（输出被省略）

Switch#sh int f0/2

FastEthernet0/2 is up, line protocol is up (connected)

Hardware is Fast Ethernet, address is 001a.6c6f.fb04 (bia 001a.6c6f.fb04)

（输出被省略）

Switch#sh int f0/3

FastEthernet0/3 is up, line protocol is up (connected)

Hardware is Fast Ethernet, address is 001a.6c6f.fb05 (bia 001a.6c6f.fb05)

（输出被省略）

说明：可以看到，物理接口的 MAC 地址是连续的，但无论什么型号的交换机，物理接口的 MAC 地址一定是不同的。

（3）查看 SVI 接口的 MAC 地址

Switch#sh int vlan 1

Vlan1 is up, line protocol is up

Hardware is EtherSVI, address is 001a.6c6f.fb40 (bia 001a.6c6f.fb40)

（输出被省略）

Switch#sh int vlan 2

Vlan2 is up, line protocol is up

Hardware is EtherSVI, address is 001a.6c6f.fb41 (bia 001a.6c6f.fb41)

（输出被省略）

Switch#sh int vlan 3

Vlan3 is up, line protocol is up

Hardware is EtherSVI, address is 001a.6c6f.fb42 (bia 001a.6c6f.fb42)

（输出被省略）

说明：可以看到，交换机 SVI 接口的 MAC 地址是连续的，但某些型号的交换机，所有 SVI 接口的 MAC 地址全部是相同的。

Trunk

在交换机上，可以将 access 接口划入各个 VLAN 中，不同 VLAN 的流量是不被交换机转发的。如果要让两个 access 接口互相通信，就必须将这两个接口划入相同的 VLAN 中。

当需要在交换机与交换机之间通信时，连接交换机的链路就可能需要为多个 VLAN 提供数据传输，这样在一条链路上提供多个 VLAN 数据传输的链路，就是 Trunk，进入 Trunk 的数据包被打上标记，写上相应的 VLAN 号，当传输到对端时，则被去掉标记，并且根据 VLAN 号将数据包转发到相应的 VLAN 中。需要说明，在 access 接口上的数据包，是没有 VLAN 号标记的，并且也不允许 VLAN 标记，如果一个 access 接口收到一个带有 VLAN 标记的数据包，是要将数据包丢弃的。

在 Trunk 上为数据包打标记是通过协议来完成的，目前有两种协议可以完成 VLAN 标记工作，分别是 Inter-Switch Link (ISL) 和 IEEE 802.1Q，其中 ISL 为思科私有协议。

当 Trunk 使用 ISL 封装时，将对进入 Trunk 的每个 VLAN 的数据包打上标记，当 ISL 收到一个没有标记的数据帧，直接丢弃。ISL 在原始以太网数据帧的基础上，

额外加上 26 字节的标记，但最多只支持 1000 个 VLAN，除此之外，ISL 还将对整个数据帧重新计算 FCS，在帧的最后插入 4 字节的新 FCS，也就是说，ISL 会在原始数据帧的基础上再加 30 字节，数据包结构如下：

可以看出，原始以太网帧的大小范围为 64-1518 字节，而 ISL 帧的大小范围为 94-1548 字节。当 ISL Trunk 收到数据帧后，直接去掉 ISL 标记和新 FCS 后，就可马上转发。

当 Trunk 使用 IEEE 802.1Q 封装时，将对除了 Native VLAN 之外的所有 VLAN 打上标记，如果 802.1Q 收到一个没有 VLAN 标记的数据帧，将其在 Native VLAN 内转发，所以请确保 Trunk 两头的 Native VLAN 号是一致的。802.1Q 在原始以太网帧中插入 4 字节的标记，支持 4096 个 VLAN，

数据包结构如下：

可以看出，原始以太网帧的大小范围为 64-1518 字节，而 802.1Q 帧的大小范围为 68-1522 字节。当 802.1Q Trunk 收到数据帧后，去掉 802.1Q 标记之外，还要重新计算 FCS 才有转发。

配置

说明：只有数据帧经过 Trunk 时，才会打上 VLAN 标记，而经过 access 接口的数据帧是没有标记的，当一个 access 接口属于某个 VLAN，那么从此接口收到的数据帧都被认为是此 VLAN 的数据，因此就可与该接口相同 VLAN 的主机通信。下面以上图配置。

1.在交换机上将 access 接口划入相应 VLAN

（1）在 SW1 上做相应配置

sw1(config)#vlan 10

```
sw1(config-vlan)#exit

sw1(config)#vlan 20

sw1(config-vlan)#exit


sw1(config)#int f0/1

sw1(config-if)#switchport mode access

sw1(config-if)#switchport access vlan 10

sw1(config-if)#no shutdown

sw1(config-if)#exit


sw1(config)#int f0/2

sw1(config-if)#switchport mode access

sw1(config-if)#switchport access vlan 20

sw1(config-if)#no shutdown

sw1(config-if)#exit


sw1(config)#int f0/23

sw1(config-if)#switchport mode access

sw1(config-if)#switchport access vlan 10

sw1(config-if)#no shutdown
```

（2）在 SW2 上做相应配置

sw2(config)#vlan 20

sw2(config-vlan)#exit

sw2(config)#int range f0/3 , f0/23

sw2(config-if-range)#switchport mode access

sw2(config-if-range)#switchport access vlan 20

sw2(config-if-range)#no shutdown

sw2(config-if-range)#exit

2.配置各路由器

（1）配置 R1

r1(config)#int f0/0

r1(config-if)#ip add 10.1.1.1 255.255.255.0

r1(config-if)#no sh

（2）配置 R2

r2(config)#int f0/0

r2(config-if)#ip add 10.1.1.2 255.255.255.0

r2(config-if)#no sh

（3）配置 R3

r3(config)#int f0/1

r3(config-if)#ip add 10.1.1.3 255.255.255.0

r3(config-if)#no sh

3.测试结果

（1）测试从 R1 到 R2 的连通性

r1#ping 10.1.1.2 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

.....

Success rate is 0 percent (0/10)

r1#

说明：因为从 R1 发送数据包到 10.1.1.2 时，数据包从 SW1 的接口 F0/1 进入，因为 F0/1 属于 VLAN 10，而 F0/2 属于 VLAN 20，所以 SW1 并不会将去往 10.1.1.2 的数据包从接口 F0/2 发出去，所以 R1 到 R2 的通信失败。

（2）测试从 R1 到 R3 的连通性

r1#ping 10.1.1.3 repeat 10

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

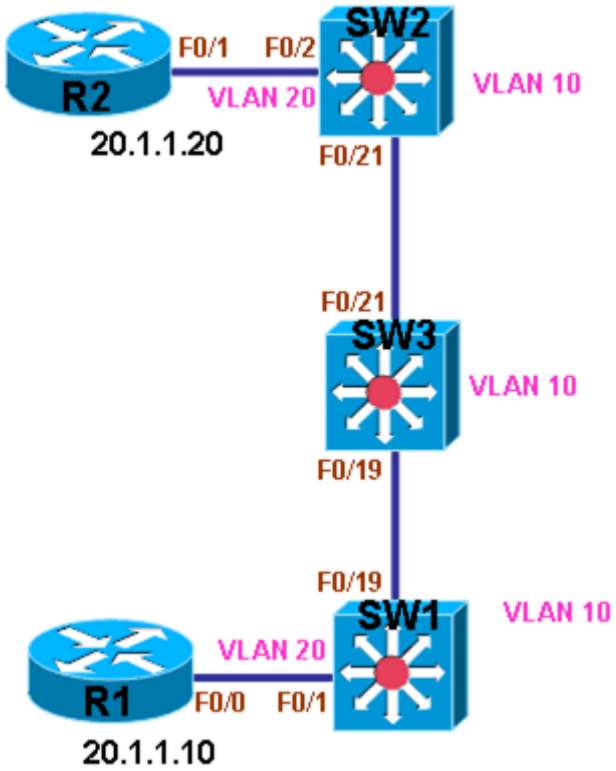
!!!!!!!!!!

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/2/4 ms

r1#

说明：因为从 R1 发送数据包到 10.1.1.3 时，数据包从 SW1 的接口 F0/1 进入，由于 F0/1 属于 VLAN 10，而 F0/23 也属于 VLAN 10，所以 SW1 将去往 10.1.1.3 的数据包从接口 F0/23 发出去，当 SW2 从 F0/23 收到数据包后，因为没有 VLAN 标记，所以认为数据包是属于 VLAN 20，便将数据包从 F0/3 发出去，最后 R3 收到数据包后，向 R1 回包，最终虽然 R1 和 R3 属于不同的 VLAN，但由于 access 接口没有 VLAN 标记，交换机并不认为是不同 VLAN，所以 R1 与 R3 的通信成功。

Trunk 重点实验



说明：以上图为例，配置实验，本实验在于说明，当一台交换机上的 VLAN 与另外一台交换机的相同 VLAN 通信时，如果中间还有交换机，当中间交换机上没有配置一个相同 VLAN 时，并且无论 Trunk 是否允许该 VLAN 通过，两边的交换机流量无法通过此 VLAN 进行通信，

1.配置交换机

（1）配置 SW1 的 VLAN 与 Trunk

sw1(config)#vlan

sw1(config)#vlan 10

sw1(config-vlan)#exit

sw1(config)#vlan 20

sw1(config-vlan)#exit

sw1(config)#int f0/19

sw1(config-if)#switchport trunk encapsulation dot1q

sw1(config-if)#switchport mode trunk

sw1(config-if)#no shutdown

sw1(config-if)#exit

sw1(config)#int f0/1

sw1(config-if)#switchport mode access

sw1(config-if)#switchport access vlan 20

sw1(config-if)#no shutdown

sw1(config-if)#exit

（2）配置 SW2 的 VLAN 与 Trunk

sw2(config)#vlan 10

sw2(config-vlan)#exit

sw2(config)#vlan 20

sw2(config-vlan)#exit

sw2(config)#int f0/21

sw2(config-if)#switchport trunk encapsulation dot1q

sw2(config-if)#switchport mode trunk

sw2(config-if)#no shutdown

sw2(config-if)#exit

sw2(config)#int f0/2

sw2(config-if)#switchport mode access

sw2(config-if)#switchport access vlan 20

sw2(config-if)#no shutdown

sw2(config-if)#exit

（3）配置 SW3 的 VLAN 与 Trunk

sw3(config)#vlan 10

sw3(config-vlan)#exit

sw3(config)#

sw3(config)#int range f0/19 , f0/21

sw3(config-if-range)#switchport trunk encapsulation dot1q

sw3(config-if-range)#switchport mode trunk

sw3(config-if-range)#no shutdown

2.配置 IP

（1）配置各设备的 IP 地址

SW1：

sw1(config)#int vlan 10

sw1(config-if)#ip address 10.1.1.1 255.255.255.0

sw1(config-if)#exit

sw1(config)#int vlan 20

sw1(config-if)#ip address 20.1.1.1 255.255.255.0

sw1(config-if)#exit

SW2：

sw2(config)#int vlan 10

sw2(config-if)#ip add 10.1.1.2 255.255.255.0

sw2(config-if)#exit

sw2(config)#int vlan 20

sw2(config-if)#ip add 20.1.1.2 255.255.255.0

sw2(config-if)#exi

R1：

r1(config)#int f0/0

r1(config-if)#ip add 20.1.1.10 255.255.255.0

r1(config-if)#no sh

R2：

r2(config)#int f0/1

r2(config-if)#ip add 20.1.1.20 255.255.255.0

r2(config-if)#no sh

3.测试通信

（1）测试 SW1 的 VLAN 10 到 SW2 的 VLAN 10 的连通性

SW1：

sw1#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

sw1#

说明：因为 SW1，SW2，SW3 都有 VLAN 10，所以 VLAN 10 从 SW1 到 SW2 是畅通的。

（2）测试 SW1 的 VLAN 20 到 SW2 的 VLAN 20 的连通性

sw1#ping 20.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

sw1#

说明：虽然 SW1，SW2 有 VLAN 20，但是 SW1 与 SW2 的 VLAN 20 通信需要穿越 SW3，而 SW3 却没有 VLAN 20，因此 SW3 在自身没有 VLAN 20 的情况下，是不允许 VLAN 20 的流量从自己经过的，所以 SW1 的 VLAN 20 到 SW2 的 VLAN 20 不通。

（3）测试 R1 到 R2 的连通性

r1#ping 20.1.1.20

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.20, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r1#

说明：虽然 R1 与 R2 都属于 VLAN 20，SW1 与 SW2 都有 VLAN 20，但是 SW3 却没有 VLAN 20，因此 SW3 在自身没有 VLAN 20 的情况下，是不允许 VLAN 20 的流量从自己经过的，所以 R1 到 R2 不通。

4.解决 VLAN 20 通信

说明：因为 SW1 与 SW2 中间的交换机 SW3 没有 VLAN 20，所以穿越 SW3 的 VLAN 20 的流量不能通过，解决方法为在 SW3 上创建 VLAN 20 即可。

（1）在 SW3 上创建 VLAN 20

sw3(config)#vlan 20

sw3(config-vlan)#exit

sw3(config)#exi

（2）测试 SW1 的 VLAN 20 到 SW2 的 VLAN 20 的连通性

sw1#ping 20.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

sw1#

说明：因为 SW1 与 SW2 中间的交换机 SW3 已经创建 VLAN 20，所以能够放行 VLAN 20 的流量，最终 SW1 的 VLAN 20 到 SW2 的 VLAN 20 通信正常。

（3）测试 R1 到 R2 的连通性

r1#ping 20.1.1.20

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 20.1.1.20, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#

说明：因为 SW1 与 SW2 中间的交换机 SW3 已经创建 VLAN 20，所以能够放行 VLAN 20 的流量，最终 R1 到 R2 的通信正常.

说明：所以基于以上结论，在多台交换机相连时，需要在跨交换机实现 VLAN 通信，即使是同 VLAN，也要解决好连通性问题。

DTP (Dynamic Trunking Protocol)

在需要使用 Trunk 链路时，通常是手工静态配置接口模式，并且手工指定 Trunk 封装协议。然而，当交换机与交换机的接口相连时，多数都需要配置为 Trunk 模式，而连接主机时，都需要配置为 access 模式，为了能够让交换机自动判断什么时候该将接口设置为 Trunk，因此开发出了动态 Trunk 配置协议（Dynamic Trunking Protocol），DTP 能够在需要将交换机接口配置为 Trunk 模式时，自动将接口配置为 Trunk，并自动选择 Trunk 封装协议，默认 ISL 优先。

DTP 采用协商的方式来决定是否将接口配置为 Trunk，可配置的接口模式，准确地讲，应该是 3 种，分别为 ON，desirable， auto，下面详细介绍各模式功能：

ON

其实就是手工静态配置为 Trunk，并且还会向对方主动发起 DTP 信息，要求对方也工作在 Trunk 模式，无论对方邻居在什么模式，自己永远工作在 Trunk 模式。

Desirable

此模式为 DTP 主动模式，工作在此模式的接口会主动向对方发起 DTP 信息，要求对方也工作在 Trunk 模式，如果对方回复同意工作在 Trunk 模式，则工作在 Trunk 模式，如果没有 DTP 回复，则工作在 access 模式。

Auto

此模式为 DTP 被动模式，工作在此模式的接口不会主动发起 DTP 信息，只会等待对方主动发起 DTP 信息，如果收到对方的 DTP 信息要求工作在 Trunk 模式，则自己回复对方同意工作在 Trunk 模式，最后的模式为 Trunk，如果 DTP 被动模式收不到 DTP 要求工作在 Trunk 的信息，则工作在 access 模式。

以上三种接口模式都会产生 DTP 信息，ON 和 desirable 是主动产生 DTP 信息，而 auto 是被动生产 DTP 信息，如果手工将接口配置成 Trunk 模式后，可以关闭 DTP 信息以节省资源，关闭 DTP 的模式为 nonegotiate。

注：

Access 模式不是 DTP 的一部分。

开启 DTP 协商的双方都必须在相同的 VTP 域内，否则协商不成功。

交换机的型号不同，默认的 DTP 模式会有所不同，3550 默认为 desirable 模式，

3560 默认为 auto 模式。

当收不到对方 DTP 回复时，则选择工作在 access 模式。

接口配置模式与最终工作模式对照表如下:

Port1 administrator mode	Port2 administrator mode	Port1&2 working mode
trunk	trunk	trunk
trunk	dynamic desirable	trunk
trunk	dynamic auto	trunk
trunk	access	fault
access	access	access
access	dynamic desirable	access
access	dynamic auto	access
dynamic desirable	dynamic desirable	trunk
dynamic desirable	dynamic auto	trunk
dynamic auto	dynamic auto	access

配置

1.配置 SW1 为 desirable，SW2 为 Trunk

（1）配置 DTP

sw1(config)#int f0/23

sw1(config-if)#switchport mode dynamic desirable

sw2(config)#int f0/23

sw2(config-if)#switchport trunk encapsulation dot1q

sw2(config-if)#switchport mode trunk

（2）查看结果

sw1#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic desirable

Operational Mode: trunk

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: On

（输出被省略）

sw1#

sw2#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: trunk

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: On

（输出被省略）

sw2#

说明：可以看到，双方接口的 DTP 协商是开启的，因为双方都会主动发起 DTP 要求对方工作在 trunk，所以最终双方的工作模式为 Trunk。

2.配置 SW1 为 desirable，SW2 为 auto

（1）配置 DTP

sw1(config)#int f0/23

sw1(config-if)#switchport mode dynamic desirable

sw2(config)#int f0/23

sw2(config-if)#switchport mode dynamic auto

（2）查看结果

sw1#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic desirable

Operational Mode: trunk

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: isl

Negotiation of Trunking: On

（输出被省略）

sw1#

sw2#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: trunk

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: isl

Negotiation of Trunking: On

（输出被省略）

sw2#

说明：可以看到，双方接口的 DTP 协商是开启的，因为 SW1 会主动发起 DTP 要求对方工作在 trunk，而 SW2 会同意工作在 Trunk，所以最终双方的工作模式为

Trunk，并且封装协议优选 ISL。

3.配置 SW1 为 auto，SW2 为 auto

（1）配置 DTP

sw1(config)#int f0/23

sw1(config-if)#switchport mode dynamic auto

sw2(config)#int f0/23

sw2(config-if)#switchport mode dynamic auto

（2）查看结果

sw1#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

（输出被省略）

sw1#

sw2#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic auto

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

（输出被省略）

sw2#

说明：可以看到，双方接口的 DTP 协商是开启的，但由于双方都不会主动发起 DTP 要求对方工作在 trunk，所以最终双方的工作模式为 access。

4.配置 SW1 为 desirable，SW2 为 Trunk，并且关闭 DTP（即为 nonegotiate）

（1）配置 DTP

sw1(config)#int f0/23

sw1(config-if)#switchport mo dynamic desirable

sw2(config)#int f0/23

sw2(config-if)#switchport trunk encapsulation dot1q

sw2(config-if)#switchport mode trunk

sw2(config-if)#switchport nonegotiate

（2）查看结果

sw1#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic desirable

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

（输出被省略）

sw1#

sw2#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: trunk

Operational Mode: trunk

Administrative Trunking Encapsulation: dot1q

Operational Trunking Encapsulation: dot1q

Negotiation of Trunking: Off

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

（输出被省略）

sw2#

说明：可以看到，SW1 的 DTP 协商是开启的，而 SW2 的 DTP 协商是关闭的，所以最终 SW1 的接口选择工作在 access 模式，而 SW2 的模式永远都为 Trunk。

5.配置双方都为 desirable，但 VTP 不在相同域内

（1）配置 DTP

sw1(config)#vtp domain ccie

sw1(config)#int f0/23

sw1(config-if)#switchport mode dynamic desirable

sw2(config)#vtp domain cisco

sw2(config)#int f0/23

sw2(config-if)#switchport mode dynamic desirable

（2）查看结果

sw1#sh vtp status

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 5

VTP Operating Mode : Server

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x04 0x98 0x3D 0x1A 0xA5 0x42 0xDC 0x34

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Local updater ID is 0.0.0.0 (no valid interface found)

sw1#

sw1#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic desirable

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

（输出被省略）

sw1#

sw2#sh vtp status

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 5

VTP Operating Mode : Server

VTP Domain Name : cisco

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x57 0x30 0x6D 0x7A 0x76 0x12 0x7B 0x40

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Local updater ID is 0.0.0.0 (no valid interface found)

sw2#

sw2#sh int f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic desirable

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

（输出被省略）

sw2#

说明：可以看到，双方的 DTP 协商都是开启的，并且模式都为 desirable，正常情况下，双方最终模式应为 trunk，然而，由于双方的 VTP 域名不同，所以 DTP 协商会失败，所以最终双方的工作模式为 access 模式。当双方 VTP 域名不匹配时，开启 DTP 协商的接口会有如下提示：

01:14:51: %LINK-3-UPDOWN: Interface FastEthernet0/23, changed state to up

01:14:51: %DTP-5-DOMAINMISMATCH: Unable to perform trunk negotiation on port Fa0/23 because of VTP domain mismatch.

VTP (VLAN Trunking Protocol)

在一个拥有多台交换机的交换网络中，通常会在多台交换机上配置相同的 VLAN，并且也会对多个接口做相同的配置。

对于需要对多个接口做相同的配置，通过快速接口配置，能够轻松实现，提高工作效率。而对于在多台交换机上做相同的 VLAN 配置，则通过 VTP 来实现。

VTP 为了在多台交换机上配置相同的 VLAN，通过将一台交换机的 VLAN 向其它交换机传播的方法来完成，其它交换机在接收到 VLAN 信息后，然后更新自己的 VLAN 数据库，以达到同步。

要将自己的 VLAN 信息发送到网络中，交换机上必须配置 Trunk，IEEE 802.1Q 和 ISL 都支持，通过 Trunk 相连的交换机便能收到对方发来的 VLAN 信息。

VTP 通过域来管理网络中的交换机，任何交换机发出的 VLAN 信息只能在一个域内传播，只有相同域的交换机才能接收此 VLAN 信息，并且根据接收到的 VLAN 信息更新自己的 VLAN 数据库。交换机是否在同一个域，是通过域名来分辨的，比如域名 ccie 与域名 ccie 属于同一个域，而域名 ccie 与域名 cisco 就属于不同的域。默认交换机的域名为空，但是最重点的，需要大家牢记的是，如果自己的域名为空，则表示与任何非空域名相同，也就是说如果对方有域名，而自己却没有域名，则自己和对方属于相同的域。

在 VTP 中，交换机分三种模式：Server、Client、Transparent，他们的功能分别如下：

Server：

可以创建，更改和删除 VLAN，可以更改任何 VTP 参数，可以将自己的 VLAN 信息向网络中发送，并且也会根据收到的 VLAN 信息来选择是否同步自己的 VLAN 数据库。

Client：

不能创建，更改和删除 VLAN，但是可以更改部分 VTP 参数，也可以将自己的 VLAN 信息向网络中发送，并且也会根据收到的 VLAN 信息来选择是否同步自己的 VLAN 数据库。

Transparent：

可以创建，更改和删除 VLAN，可以更改任何 VTP 参数，不会将自己的 VLAN 信息向网络中发送，但是会转发接收到其它交换机发来的 VLAN 信息，并且不会根据收到的 VLAN 信息来同步自己的 VLAN 数据库。

从上可以看出，Server 与 Client 的唯一区别在于，Server 可以随意修改自己的 VLAN 信息和 VTP 参数，而 Client 则不能，除此之外，其它完全相同。

Server 与 Transparent 的区别在于, Transparent 不会将自己的 VLAN 信息发送到网络中，并且也不会向别人同步自己的 VLAN 数据库。

所以最终的结论是，如果希望从网络中接收 VLAN 信息来同步自己的 VLAN 数据库，配置成 Server 与 Client 都可以实现，要将自己的 VLAN 信息发送到网络中，Server 与 Client 也都能实现。如果要具有修改 VLAN 数据库的权限，只有 Server 与 Transparent 能做到，Client 是不能自己更改 VLAN 数据库的。

Server 与 Client 发出的 VLAN 信息，都有一个 configuration revision 号码，每修改一次 VLAN 信息，configuration revision 号则加 1，如果做相同操作，configuration revision 号是不会有变化的。configuration revision 号越高（数字越大），则说明 VLAN 信息越新。

Server 与 Client 从网络中接收到 VLAN 信息后，是否根据此信息同步自己的 VLAN 数据库，则要将自己的 VLAN 信息与接收到的作对比，如果接收到的 VLAN 信息的 configuration revision 号比自己的大，则将自己的 VLAN 数据库与接收到的进行同步，如果 configuration revision 号比自己的小或者相等，则放弃同步。域中总是先使用 configuration revision 号码最高的 VLAN 信息

默认情况下，交换机的域名为空，无论是 Server 还是 Client，在空域名的情况下，是不会将自己的 VLAN 信息往外发的，但是在域名为空的情况下，无论收到任何 VLAN 信息，只要 configuration revision 号比自己的大，就会同步自己的 VLAN 数据库，并且添加上相同的域名。域名在配置之后，只能更改，但不能删除。如果网络中全是 Client，可想而知就不要配置域名了。

在谈及 VTP，不得不详细解释 VLAN，交换机所支持的 VLAN 数为 1-4094，VLAN 1-1005 称为 Normal VLAN，VLAN 1006 – 4094 称为 Extended VLAN。Normal VLAN（1-1005）是保存在 VLAN 数据库中的，也就是 vlan.dat，而 Extended VLAN(1006-4094)是保存在 startup-config 中的。Normal VLAN（1-1005）可以随意配置，而 Extended VLAN(1006-4094)只能在 VTP 模式为 Transparent 时才能配置。所以，VTP 只能将 Normal VLAN（1-1005）在网络中更新。当同时配置了 1-1005 的 VLAN 和 1006-4096 的 VLAN，在删除 vlan.dat 后，1-1005 的 VLAN 会被删除，但 1006-4096 的 VLAN 还在，如果删除了 startup-config，那么则会删除 1006-4096 的 VLAN，但不会影响 1-1005 的 VLAN。

VTP 现有两个版本，ver 1 和 ver 2，默认为 ver 1，因为 Transparent 会转发接收到其它交换机发来的 VLAN 信息，但是当自己的 VTP 版本为 ver 1 时，只有自己接收到的 VLAN 信息的域名和 VTP 版本与自己的相同，才会转发，但如果自己为 ver 2，则无论收到任何 VLAN 信息都会转发。

如果域中一台交换机开了 VTP ver 2，则应该全部都要打开，但是只有 Server 和 Transparent 才能更改 VTP 版本，而 Client 会根据收到的 VLAN 信息同步自己的 VTP 版本。

交换机还可以为 VTP 配置密码，当配置密码后，即使 VTP 域名相同，如果密码不同，也不能根据接收到的 VLAN 信息更新自己的 VLAN 数据库。要确认 VTP 密码是否相同，双方的 MD5 digest 值必须相同。

附:在交换机最新的 IOS 版本中,如果 3560 的 12.2(52)SE ,已经加入对 VTP version 3 的支持,最大的特点就是,可以在 VTP 信息中传递 Extended VLAN(1006-4094),但改为 Ver 3 之后,不能再切换到 Ver 1 和 Ver 2.

重点说明：

★ 交换机的配置信息保存在 nvram 存储器的 startup-config 文件中。

★ 而 Flash 中的文件 config.text 与 nvram 存储器的 startup-config 文件完全相同，删除任何一个，即同时删除两个。（注：此规则不完全适用于高端交换机）

★交换机的 Normal VLAN（1-1005）是保存在文件 vlan.dat 中，而 Extended VLAN(1006-4094)是保存在 nvram 存储器的文件 startup-config 中。

★ VTP 信息全部保存在 vlan.dat 中。

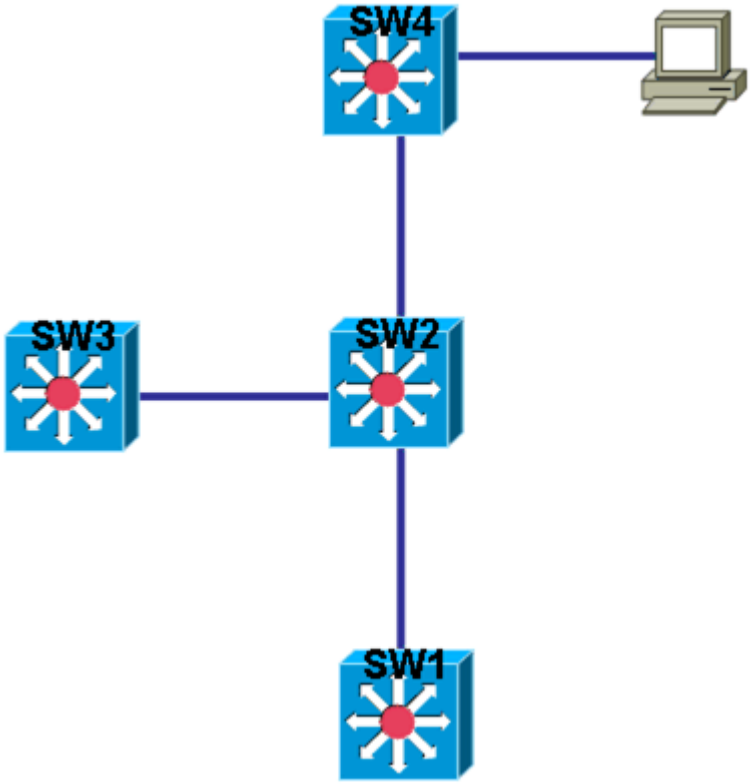
★ 当 VTP 模式为 Transparent 时，所有 VLAN 信息和 VTP 信息除了保存在 vlan.dat 中之外，还会保存在 nvram 存储器的 startup-config 中。

★ 当 VTP 模式为 Server 和 Client 时，所有 VLAN 信息和 VTP 信息只保存在 vlan.dat 中，不会保存在 nvram 存储器的 startup-config 中，所以 show running-config 时，也是看不到 VLAN 信息的。

★ 域名为空的交换机是不会发送任何 VTP 信息的。

★ 将模式改为 Transparent，可以清除所有 VTP 信息。

VTP Pruning



如上图所示，当交换机 SW1 收到 broadcast, multicast 以及 unknown unicast 后，会在所有 Trunk 上进行广播发送，最终结果造成 SW2 会转发给 SW3，也会转发给 SW4，而只有 SW4 上接有终端，也就是说只有 SW4 需要接收这些广播，对于 SW3，转发这些广播是毫无意义的，因为自己没有连接终端。

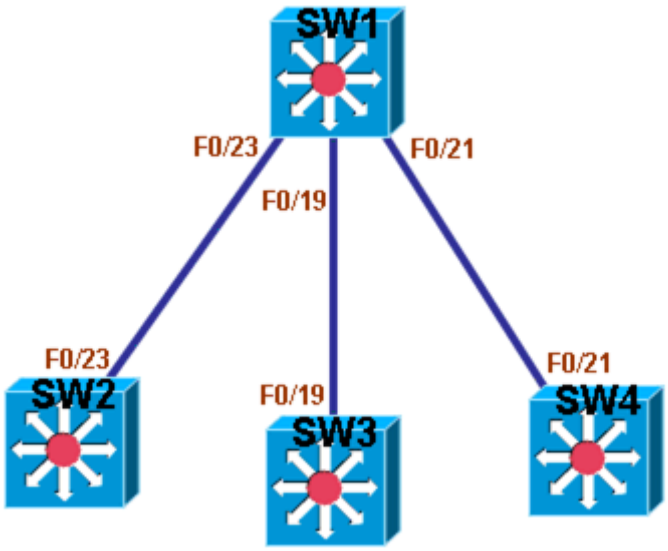
对于上述情况，当一台交换机在某 VLAN 进入广播发送数据时，流量应该只被发送到在此 VLAN 连接了终端的交换机，而对于没有连接终端的交换机，很明显，是没有必须接收这样的广播了。为了节省带宽，提高网络性能，VTP Pruning 限制交换机只将广播发送到连接了终端的交换机。如果上图中开启了 VTP Pruning，则 SW1 发出的广播只会被发送到 SW2，再转发到 SW4，而不会转发到 SW3。

在 Trunk 上，只有某 VLAN 允许被剪除，那么在此 VLAN 的广播才不会发到没有连接终端的交换机，如果不允许剪除，则广播照常。允许被剪除的 VLAN 范围是 2-1001，而 VLAN1 和 1002-1005 以及 1006-4094 是不能被剪除的，开启 VTP

Pruning 后，默认 VLAN2-1001 被剪除，但剪除的 VLAN 号可以在 Trunk 上随意定义。

VTP 模式为 Transparent 时，是不支持 VTP Pruning 的，但无论支持 VTP Ver 1 还是 Ver 2 都支持 VTP Pruning。

配置



说明：以上图为例，配置 VTP。第一部分为验证交换机文件系统，第二部分为验证 VTP。

第一部分 （验证交换机文件系统）

1.在 SW1 上配置 VTP

（1）创建 vlan 2000，vlan 3000

sw1(config)#vlan 2000

sw1(config-vlan)#exit

% Failed to create VLANs 2000

Extended VLAN(s) not allowed in current VTP mode.

%Failed to commit extended VLAN(s) changes.

sw1(config)#

说明：因为交换机默认为 Server 模式，所以不能创建 Extended VLAN(1006-4094)。

（2）在 VTPTransparent 下创建 vlan 2000，vlan 3000

sw1(config)#vtp domain ccie

sw1(config)#vtp mo transparent

Setting device to VTP TRANSPARENT mode.

sw1(config)#vlan 2000

sw1(config-vlan)#exit

sw1(config)#vlan 3000

sw1(config-vlan)#exit

sw1(config)#

说明：Vlan 2000 在 transparent 模式下创建成功。

(3)查看 VLAN

sw1#sh vlan

VLAN Name Status Ports

---- ---------- -------- ----------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/23, Fa0/24

Gi0/1, Gi0/2

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

2000 VLAN2000 active

3000 VLAN3000 active

（输出被省略）

sw1#

说明：Vlan 2000 在 transparent 模式下创建成功。

（4）在 SW1 上创建 VLAN 2-5，以及 VLAN 3000

sw1(config)#vlan 2

sw1(config-vlan)#exit

sw1(config)#vlan 3

sw1(config-vlan)#exit

sw1(config)#vlan 4

sw1(config-vlan)#exit

sw1(config)#vlan 5

sw1(config-vlan)#exit

(5)保存并查看

保存：

sw1#wr

Building configuration...

[OK]

sw1#

查看 VLAN：

sw1#sh vlan

VLAN Name Status Ports

---- -------------------------------- --------- -------------------------------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/23, Fa0/24

Gi0/1, Gi0/2

2 VLAN0002 active

3 VLAN0003 active

4 VLAN0004 active

5 VLAN0005 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

2000 VLAN2000 active

3000 VLAN3000 active

（输出被省略）

sw1#

查看 VTP：

sw1#sh vtp sta

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 9

VTP Operating Mode : Transparent

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x63 0xE7 0xF7 0x4B 0xFD 0xED 0x17 0xAA

Configuration last modified by 0.0.0.0 at 3-1-93 00:02:01

sw1#

说明：VLAN 创建成功，VTP 也修改成功。

（6）查看文件系统

sw1#dir flash:

Directory of flash:/

2 -rwx 7457899 Mar 1 1993 06:35:16 +00:00 c3550-ipservicesk9-mz.122-35.SE3.bin

3 -rwx 796 Mar 1 1993 00:02:44 +00:00 vlan.dat

4 -rwx 0 Mar 1 1993 05:57:14 +00:00 env_vars

```
5   -rwx         24    Mar 1 1993 05:57:14 +00:00   system_env_vars

6   -rwx        2416   Mar 1 1993 00:03:10 +00:00   config.text

7   -rwx         24    Mar 1 1993 00:03:10 +00:00   private-config.text


15998976 bytes total (8535040 bytes free)


sw1#dir nv

sw1#dir nvram:

Directory of nvram:/


  380   -rw-       2416                <no date>   startup-config

  381   ----         24                <no date>   private-config


393216 bytes total (390724 bytes free)

sw1#
```

说明：存在 VLAN 信息和 VTP 信息的 vlan.dat 已经生成；nvram 中的 startup-config 也已经生成，相应的 config.text 也已经生成。

（7）共享文件系统

```
sw1(config)#int vlan 1

sw1(config-if)#ip add 1.1.1.1 255.255.255.0
```

sw1(config)#tftp-server flash:vlan.dat

sw1(config)#tftp-server flash:config.text

sw1(config)#tftp-server nvram:startup-config

说明：交换机已经将 vlan.dat，config.text，startup-config 通过 TFTP 在网络中共享。

2.通过 SW2 验证 SW1 的 vlan.dat

（1）查看当前 VTP 和 VLAN

sw2#sh vtp status

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 5

VTP Operating Mode : Server

VTP Domain Name :

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Local updater ID is 1.1.1.2 on interface VI1 (lowest numbered VLAN interface found)

sw2#

sw2#sh vlan

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4
			Fa0/5, Fa0/6, Fa0/7, Fa0/8
			Fa0/9, Fa0/10, Fa0/11, Fa0/12
			Fa0/13, Fa0/14, Fa0/15, Fa0/16
			Fa0/17, Fa0/18, Fa0/19, Fa0/20
			Fa0/21, Fa0/22, Fa0/24, Gi0/1
			Gi0/2
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

（输出被省略）

sw2#

说明：SW2 的 VLAN 和 VTP 为默认配置。

（2）复制 SW1 的 vlan.dat

sw2(config)#int vlan 1

sw2(config-if)#ip add 1.1.1.2 255.255.255.0

sw2#copy tftp: flash:

Address or name of remote host []? 1.1.1.1

Source filename []? vlan.dat

Destination filename [vlan.dat]?

Accessing tftp://1.1.1.1/vlan.dat...

Loading vlan.dat from 1.1.1.1 (via Vlan1): !

[OK - 796 bytes]

796 bytes copied in 0.032 secs (24875 bytes/sec)

sw2#

说明：SW1 的 vlan.dat 已经被 SW2 复制，接下来可以验证 vlan.dat 中的内容。

（3）查看 SW2 复制的 SW1 的 vlan.dat

sw2#dir flash:

Directory of flash:/

2 -rwx 7457899 Mar 1 1993 06:33:13 +00:00 c3550-ipservicesk9-mz.122-35.SE3.bin

3 -rwx 796 Mar 1 1993 00:10:41 +00:00 vlan.dat

4 drwx 0 Mar 1 1993 02:51:43 +00:00 test

7 -rwx 0 Mar 1 1993 01:52:09 +00:00 system_env_vars

8 -rwx 0 Mar 1 1993 01:52:09 +00:00 env_vars

15998976 bytes total (8538624 bytes free)

sw2#

说明：可以看到 vlan.dat 与 SW1 的 vlan.dat 相同。

(4)在 SW2 上使用 SW1 的 vlan.dat

说明：因为 SW1 的 vlan.dat 已经复制到 SW2 的 flash 中，所以重启 SW2 后，便可读取其中的内容。

重启 SW2 后，查看 VLAN 信息和 VTP 信息：

查看 VLAN 信息：

Sw2#sh vlan

VLAN Name Status Ports

---- ----------------------- -------- -------------------------------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/23, Fa0/24

Gi0/1, Gi0/2

2 VLAN0002 active

3 VLAN0003 active

4 VLAN0004 active

5 VLAN0005 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

（输出被省略）

Sw2#

查看 VTP 信息：

Sw2#sh vtp sta

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 9

VTP Operating Mode : Transparent

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x63 0xE7 0xF7 0x4B 0xFD 0xED 0x17 0xAA

Configuration last modified by 0.0.0.0 at 3-1-93 00:02:01

Sw2#

说明：可以验证，vlan.dat 中只有 1-1005 的 VLAN，并且 VTP 信息保存在 vlan.dat 中。

3.通过 SW3 验证 SW1 的 startup-config

（1）查看 SW3 当前的 startup-config

sw3#dir nvram:

Directory of nvram:/

382 -rw- 0 <no date> startup-config

383 ---- 0 <no date> private-config

393216 bytes total (393164 bytes free)

sw3#

说明：SW3 当前的 startup-config 为空。

（2）复制 SW1 的 startup-config

sw3(config)#int vlan 1

sw3(config-if)#ip add 1.1.1.3 255.255.255.0

sw3#copy tftp: flash:

Address or name of remote host [1.1.1.1]?

Source filename [startup-config]?

Destination filename [startup-config]?

Accessing tftp://1.1.1.1/startup-config...

Loading startup-config from 1.1.1.1 (via Vlan1): !

[OK - 2416 bytes]

2416 bytes copied in 0.088 secs (27455 bytes/sec)

sw3#

说明：SW1 的 startup-config 已经被 SW3 复制，接下来可以验证 startup-config 中的内容。

（3）在 SW3 上导入复制的 SW1 的 startup-config

sw3#copy flash:startup-config running-config

Destination filename [running-config]?

Failed to generate persistent self-signed certificate.

Secure server will use temporary self-signed certificate.

2416 bytes copied in 0.416 secs (5808 bytes/sec)

sw1#

说明：因为使用了 SW1 的 startup-config，所以主机名也变成了 SW1。

（4）查看 VLAN 与 VTP 信息

查看 VLAN 信息：

sw1#sh vlan

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4
			Fa0/5, Fa0/6, Fa0/7, Fa0/8
			Fa0/9, Fa0/10, Fa0/11, Fa0/12
			Fa0/13, Fa0/14, Fa0/15, Fa0/16
			Fa0/17, Fa0/18, Fa0/19, Fa0/20
			Fa0/21, Fa0/22, Fa0/23, Fa0/24
			Gi0/1, Gi0/2
2	VLAN0002	active	
3	VLAN0003	active	

4 VLAN0004 active

5 VLAN0005 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

2000 VLAN2000 active

3000 VLAN3000 active

（输出被省略）

sw1#

查看 VTP：

sw1#sh vtp status

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 9

VTP Operating Mode : Transparent

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x63 0xE7 0xF7 0x4B 0xFD 0xED 0x17 0xAA

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

sw1#

说明：SW3 上除了拥有 VLAN 1-1005 外，1006-4094 的 VLAN 也存在，说明在 Transparent 模式下，VLAN 信息不仅保存在 vlan.dat 中，还保存在 startup-config 中，并且 VTP 也成功保存在 startup-config 中。

4.通过 SW4 验证 SW1 的 config.text

（1）从 SW4 复制 SW1 的 config.text

sw4(config)#int vlan 1

sw4(config-if)#ip address 1.1.1.4 255.255.255.0

sw4#copy tftp: flash:

Address or name of remote host []? 1.1.1.1

Source filename []? config.text

Destination filename [config.text]?

Accessing tftp://1.1.1.1/config.text...

Loading config.text from 1.1.1.1 (via Vlan1): !

[OK - 2416 bytes]

2416 bytes copied in 0.052 secs (46462 bytes/sec)

sw4#

说明： SW1 的 vlan.dat 已经被 SW4 复制，接下来可以验证 config.text 中的内容。

（2）SW4 上使用 SW1 的 config.text

说明：因为 SW4 上没有保存配置文件，但拥有了 SW1 的 config.text，所以重启后，就会读取 config.text 的配置，重启后，SW1 的 config.text 内容就被验证

重启 SW4，查看结果：

查看 VLAN：

sw1#sh vlan

VLAN Name Status Ports

---- -------------------------------- --------- -------------------------------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/23, Fa0/24

Gi0/1, Gi0/2

2 VLAN0002 active

3 VLAN0003 active

4 VLAN0004 active

5 VLAN0005 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

2000 VLAN2000 active

3000 VLAN3000 active

（输出被省略）

sw1#

查看 VTP：

sw1#sh vtp status

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 9

VTP Operating Mode : Transparent

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x63 0xE7 0xF7 0x4B 0xFD 0xED 0x17 0xAA

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

sw1#

说明：因为使用了 SW1 的 config.text，所以主机名也变成了 SW1。并且 VLAN 与 VTP 与 SW1 完全相同，说明 config.text 与 startup-config 完全相同。

5.在 SW1 上验证 VLAN 存放位置

（1）在 SW1 上删除 startup-config

说明：由于删除 vlan.dat 是没有用的，因为 Transparent 会将所有 VLAN，如 VLAN 1006-4094 存放在 startup-config 中，即使删了 vlan.dat，所有内容还存在，所以直接删除 startup-config 来测试：

sw1#erase nvram:

Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]

[OK]

Erase of nvram: complete

sw1#

sw1#dir nvram:

Directory of nvram:/

382 -rw- 0 <no date> startup-config

383 ---- 0 <no date> private-config

393216 bytes total (393164 bytes free)

sw

说明：startup-config 已经为空，因为已被删除。

（2）重启 SW1 后查看结果：

查看 VLAN：

Switch#sh vlan

VLAN Name Status Ports

---- ---------- -------- ----------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/23, Fa0/24

Gi0/1, Gi0/2

2 VLAN0002 active

3 VLAN0003 active

4 VLAN0004 active

5 VLAN0005 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

（输出被省略）

Switch#

查看 VTP：

Switch#sh vtp status

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 9

VTP Operating Mode : Transparent

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

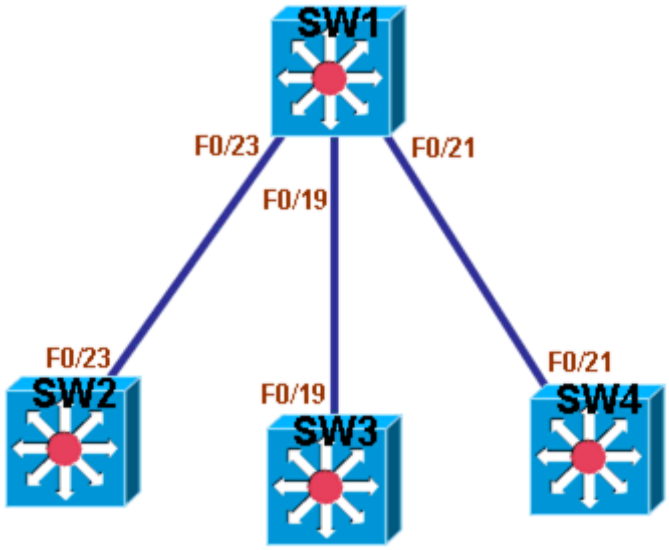
MD5 digest : 0x63 0xE7 0xF7 0x4B 0xFD 0xED 0x17 0xAA

Configuration last modified by 0.0.0.0 at 3-1-93 00:02:01

Switch#

说明： 因为 VTP 信息和 VLAN 1-1005 存放在 vlan.dat 中，所以删除了 startup-config,只是删除了 VLAN 1006-4094，而 VTP 信息和 VLAN 1-1005 仍旧存在。

第二部分（验证 VTP）



说明：还是以上图为例，验证 VTP

1.关闭交换机上所有端口

（1）在所有交换机上关闭所有端口

int range f0/1 - 24

shutdown

2.查看默认 VTP

（1）所有交换机上，默认 VTP 如下：

switch#sh vtp status

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 5

VTP Operating Mode : Server

VTP Domain Name :

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Local updater ID is 0.0.0.0 (no valid interface found)

switch#

说明：默认 VTP 域名为空，且默认模式为 Server。

3.配置 SW1 的 VTP

（1）在 SW1 上配置 VLAN

sw1(config)#vlan 3

sw1(config-vlan)#exi

sw1(config)#vlan 5

sw1(config-vlan)#exi

sw1(config)#vlan 7

sw1(config-vlan)#exi

sw1(config)#vlan 9

sw1(config-vlan)#exit

sw1(config)#vtp domain ccie

说明：SW1 上的 VLAN 为 3 5 7 9 ，全部是奇数，VTP 域名为 ccie。

（2）查看 SW1 的 VTP 信息

sw1#sh vtp status

VTP Version : 2

Configuration Revision : 4

Maximum VLANs supported locally : 1005

Number of existing VLANs : 9

VTP Operating Mode : Server

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x4C 0x22 0xDD 0xCA 0x61 0xA4 0x7C 0x65

Configuration last modified by 0.0.0.0 at 3-1-93 00:04:19

Local updater ID is 0.0.0.0 (no valid interface found)

sw1#

说明：现在 SW1 的 VTP 模式为 Server，域名为 ccie，Configuration Revision 为 4。

（3）查看 SW1 的 VLAN 信息

sw1#sh vlan

VLAN Name Status Ports

---- ----------------------------- -------- -------------------------------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/23, Fa0/24

Gi0/1, Gi0/2

3 VLAN0003 active

5 VLAN0005 active

7 VLAN0007 active

9 VLAN0009 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

（输出被省略）

sw1#

说明：SW1 上的 VLAN 为 1 3 5 7 9，全部奇数。

4.配置 SW2 的 VTP

（1）在 SW2 上配置 VLAN

sw2(config)#vlan 2

sw2(config-vlan)#exi

sw2(config)#vlan 4

sw2(config-vlan)#exi

sw2(config)#vlan 6

sw2(config-vlan)#exi

sw2(config)#vlan 8

sw2(config-vlan)#exi

sw2(config)#vlan 10

sw2(config-vlan)#exi

sw2(config)#vlan 12

sw2(config-vlan)#exit

sw2(config)#vtp domain ccie

sw2(config)#vtp mode client

说明：SW2 上的 VLAN 为 2 4 6 8 10 12 ，全部是偶数，VTP 域名为 ccie，并且模式为 Client。

（2）查看 SW2 的 VTP 信息

查看 VTPsw2#sh vtp status

VTP Version : 2

Configuration Revision : 6

Maximum VLANs supported locally : 1005

Number of existing VLANs : 11

VTP Operating Mode : Client

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x5E 0x0C 0x19 0x2B 0xC3 0x13 0x05 0x4F

Configuration last modified by 0.0.0.0 at 3-1-93 00:05:49

sw2#

说明：现在 SW2 的 VTP 模式为 Client，域名为 ccie，Configuration Revision 为 6。

（3）查看 SW2 的 VLAN 信息

sw2#sh vlan

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4
			Fa0/5, Fa0/6, Fa0/7, Fa0/8
			Fa0/9, Fa0/10, Fa0/11, Fa0/12
			Fa0/13, Fa0/14, Fa0/15, Fa0/16
			Fa0/17, Fa0/18, Fa0/19, Fa0/20
			Fa0/21, Fa0/22, Fa0/23, Fa0/24
			Gi0/1, Gi0/2
2	VLAN0002	active	
4	VLAN0004	active	
6	VLAN0006	active	
8	VLAN0008	active	

10 VLAN0010 active

12 VLAN0012 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

（输出被省略）

sw2#

说明：SW2 上的 VLAN 为 2 4 6 8 10 12 ，全部是偶数。

5.验证 VTP

（1）开启 SW1 与 SW2 之间的 Trunk 链路：

SW1：

sw1(config)#int ran f0/23

sw1(config-if-range)#switchport trunk encapsulation dot1q

sw1(config-if-range)#switchport mode trunk

sw1(config-if-range)#no shut

SW2：

sw2(config)#int f0/23

sw2(config-if)#switchport trunk encapsulation dot1q

sw2(config-if)#switchport mode trunk

sw2(config-if)#no shutdown

说明：SW1 与 SW2 的 Trunk 已连通，VTP 即将同步。

（2）查看 VTP 结果

SW1：

sw1#sh vtp status

VTP Version : 2

Configuration Revision : 6

Maximum VLANs supported locally : 1005

Number of existing VLANs : 11

VTP Operating Mode : Server

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x5E 0x0C 0x19 0x2B 0xC3 0x13 0x05 0x4F

Configuration last modified by 0.0.0.0 at 3-1-93 00:05:49

Local updater ID is 0.0.0.0 (no valid interface found)

sw1#

sw1#sh vlan

VLAN Name Status Ports

---- ----------------------------- -------- -------------------------------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/24, Gi0/1

Gi0/2

2 VLAN0002 active

4 VLAN0004 active

6 VLAN0006 active

8 VLAN0008 active

10 VLAN0010 active

12 VLAN0012 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

（输出被省略）

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

sw1#

SW2：

sw2#sh vtp status

VTP Version : 2

Configuration Revision : 6

Maximum VLANs supported locally : 1005

Number of existing VLANs : 11

VTP Operating Mode : Client

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x5E 0x0C 0x19 0x2B 0xC3 0x13 0x05 0x4F

Configuration last modified by 0.0.0.0 at 3-1-93 00:05:49

sw2#

sw2#sh vlan

VLAN Name Status Ports

---- ---------- -------- ----------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/24, Gi0/1

Gi0/2

2 VLAN0002 active

4 VLAN0004 active

6 VLAN0006 active

8 VLAN0008 active

10 VLAN0010 active

12 VLAN0012 active

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

（输出被省略）

sw2#

说明：从结果中可以看出，VTP 模式为 Server 的 SW1 已经将自己的 VLAN 信息与 VTP 模式为 Client 的 SW2 同步，因为 SW1 的 Configuration Revision 为 4，而 SW2 的 Configuration Revision 为 6，所以无论 Server 与 Client，在收到 VTP 信息后，只要 Configuration Revision 比自己的大，则将自己的与收到的同步。

6.验证 VTP 空域名

（1）查看 SW3 的 VTP 信息和 VLAN 信息

查看 VTP 信息：

sw3#sh vtp sta

VTP Version : 2

Configuration Revision : 0

Maximum VLANs supported locally : 1005

Number of existing VLANs : 5

VTP Operating Mode : Server

VTP Domain Name :

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00

Local updater ID is 0.0.0.0 (no valid interface found)

sw3#

查看 VLAN 信息:

sw3#sh vlan

VLAN Name Status Ports

---- ---------- -------- ----------

1 default active Fa0/1, Fa0/2, Fa0/3, Fa0/4

Fa0/5, Fa0/6, Fa0/7, Fa0/8

Fa0/9, Fa0/10, Fa0/11, Fa0/12

Fa0/13, Fa0/14, Fa0/15, Fa0/16

Fa0/17, Fa0/18, Fa0/19, Fa0/20

Fa0/21, Fa0/22, Fa0/23, Fa0/24

Gi0/1, Gi0/2

1002 fddi-default act/unsup

1003 token-ring-default act/unsup

1004 fddinet-default act/unsup

1005 trnet-default act/unsup

（输出被省略）

sw3#

说明：可以看到，SW3 的 VTP 域名为空，并且没有手工配置的 VLAN。

（2）开启 SW1 与 SW3 之间的 Trunk 链路：

SW1：

sw1(config)#int f0/19

sw1(config-if)#switchport trunk encapsulation dot1q

sw1(config-if)#switchport mode trunk

sw1(config-if)#no shutdown

SW2：

sw3(config)#int f0/19

sw3(config-if)#switchport trunk encapsulation dot1q

sw3(config-if)#switchport mode trunk

sw3(config-if)#no shutdown

说明：SW1 与 SW3 的 Trunk 已连通，VTP 即将同步。

（3）查看 SW3 的 VTP 信息：

sw3#sh vtp status

VTP Version : 2

Configuration Revision : 6

Maximum VLANs supported locally : 1005

Number of existing VLANs : 11

VTP Operating Mode : Server

VTP Domain Name : ccie

VTP Pruning Mode : Disabled

VTP V2 Mode : Disabled

VTP Traps Generation : Disabled

MD5 digest : 0x5E 0x0C 0x19 0x2B 0xC3 0x13 0x05 0x4F

Configuration last modified by 0.0.0.0 at 3-1-93 00:05:49

Local updater ID is 0.0.0.0 (no valid interface found)

sw3#

说明：因为 SW3 的 VTP 域名为空，而 SW1 的 VTP 域名为 ccie，在域名为空的情况下，无论收到任何 VLAN 信息，只要 configuration revision 号比自己的大，就会同步自己的 VLAN 数据库，并且添加上相同的域名，所以空域名的 SW3 在收到 VTP 更新之后，同步了自己的信息。所以请谨慎使用空域名交换机。

VTP Pruning 与 VTP Ver 2

1.在 SW1 上开启 VTP Pruning 与 VTP Ver 2

(1) 在 SW1 上开启 VTP Pruning 与 VTP Ver 2

sw1(config)#vtp pruning

Pruning switched on

sw1(config)#vtp version 2

2 在 SW4 上查看结果

（1）查看 SW4 上的 VTP Pruning 与 VTP Ver 2

sw4#sh vtp status

VTP Version : 2

Configuration Revision : 10

Maximum VLANs supported locally : 1005

Number of existing VLANs : 11

VTP Operating Mode : Server

VTP Domain Name : ccie

VTP Pruning Mode : Enabled

VTP V2 Mode : Enabled

VTP Traps Generation : Disabled

MD5 digest : 0xCA 0x78 0x25 0x9B 0x99 0x9B 0xE7 0x72

Configuration last modified by 1.1.1.1 at 3-1-93 00:35:06

Local updater ID is 1.1.1.4 on interface Vl1 (lowest numbered VLAN interface found)

sw4#

说明：VTP 域中，只要在一台上开启 VTP Pruning 与 VTP Ver 2，其它交换机全部开启，但只有 Server 和 Transparent 才能更改 VTP 版本，而 Transparent 是不支持 VTP Pruning 的。

3.更改 Pruning VLAN

说明：默认剪除 VLAN 2-1001，但可随意更改

（1）在 SW1 的 Trunk 上更改 Pruning VLAN

sw1(config)#int f0/21

sw1(config-if)#switchport trunk pruning vlan remove 10

sw1(config-if)#exit

(2)查看 Pruning VLAN

sw1#sh int f0/21 switchport

（输出被省略）

Trunking VLANs Enabled: ALL

Pruning VLANs Enabled: 2-9,11-1001

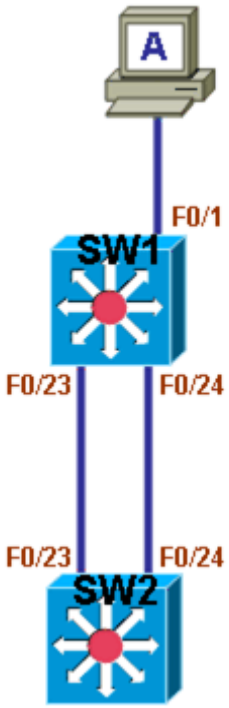
Capture Mode Disabled

（输出被省略）

sw1#

说明：可以看到，VLAN 10 已经从 Pruning VLAN 中移除，只剩 VLAN 2-9,11-1001。

STP（Spanning-Tree Protocol）



在上图所示的网络环境中，当交换机之间连有多条链路时，将存在一定的问题，如 SW1 的 MAC 地址表中会显示接口 F0/1 与主机 A 相对应，而当数据发往 SW2 后，SW2 的 MAC 地址表则记录接口 F0/23 与主机 A 相对应，当 SW2 再次将流量从接口 F0/24 发回 SW1 时，SW1 的 MAC 地址表又会记录接口 F0/24 与主机 A 相对应。

因此可以看出，当交换机之间存在多条活动链路时，交换机将从不正常的接口上学习到 MAC 地址，导致 MAC 地址表的不正确与不稳定，并且还会导致重复的数据包在网络中传递，引起广播风暴，使网络不稳定。

为了防止交换机之间由于多条活动链路而导致的网络故障，必须将多余的链路置于非活动状态，即不转发用户数据包，而只留下单条链路作为网络通信，当唯一的活动链路不能工作时，再启用非活动链路，从而达到网络的冗余性。要实现此功能，需要依靠生成树协议（STP）来完成，STP 将交换网络中任何两个点之间的多余链路置于 Blocking（关闭）状态，而只留一条活动链路，当使用中的活动链路失效时，立即启用被 Block 的链路，以此来提供网络的冗余效果。

STP 并非思科私有协议，STP 为 IEEE 标准协议，并且有多个协议版本，版本与协议号的对应关系如下：

Common Spanning Tree (CST) = IEEE 802.1D

Rapid Spanning Tree Protocol (RSTP) = IEEE 802.1w

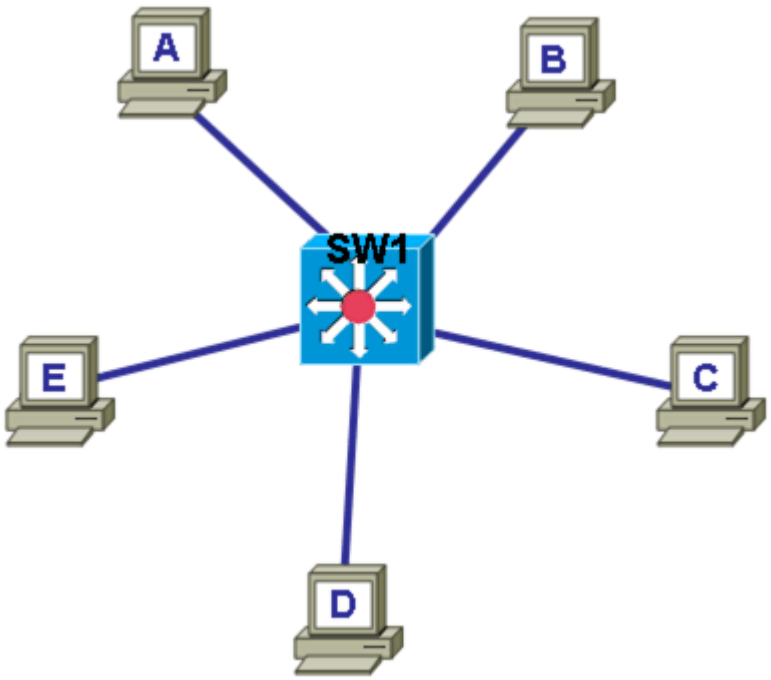
Per-VLAN Spanning-Tree plus (PVST+) = Per-VLAN EEE 802.1D

Rapid PVST+ = Per-VLAN IEEE 802.1w

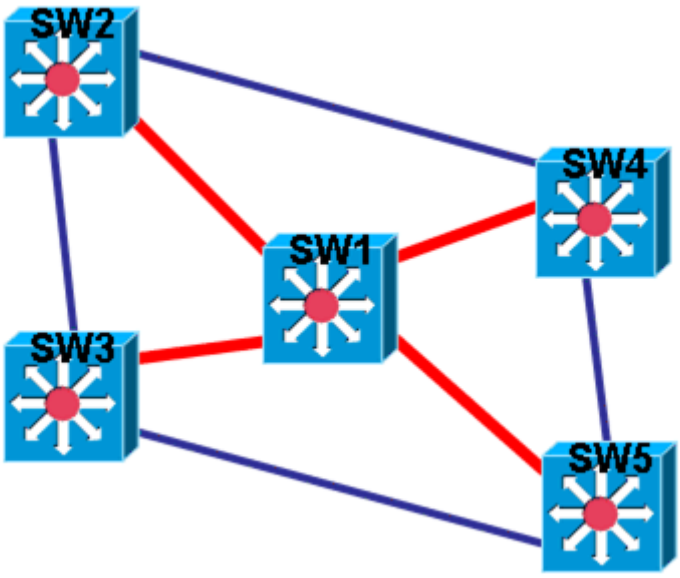
Multiple Spanning Tree Protocol (MSTP) = IEEE 802.1s

下面来详细介绍 STP 协议：

请观察如下网络环境：



在如上所示的网络环境中，不难看出，当所有主机都使用单条链路与一台核心相连时，只要不再增加其它额外设备与链路，就不可能存在环路。交换机就当相于 Hub 一样连接了多台主机，而这样的网络结构，被称为 hub-spoke 网络结构，只要主机与 Hub 是连通的，那么就表示主机之间是连通的。基于此原因，STP 借助了 hub-spoke 网络结构无环的网络思想，将一个拥有多台交换机通过多条链路相连的网络，通过 Block 掉任意两点之间多余的链路而只留下单条链路，最终修整出一个 hub-spoke 的网络环境，创造一个无环的交换网络。

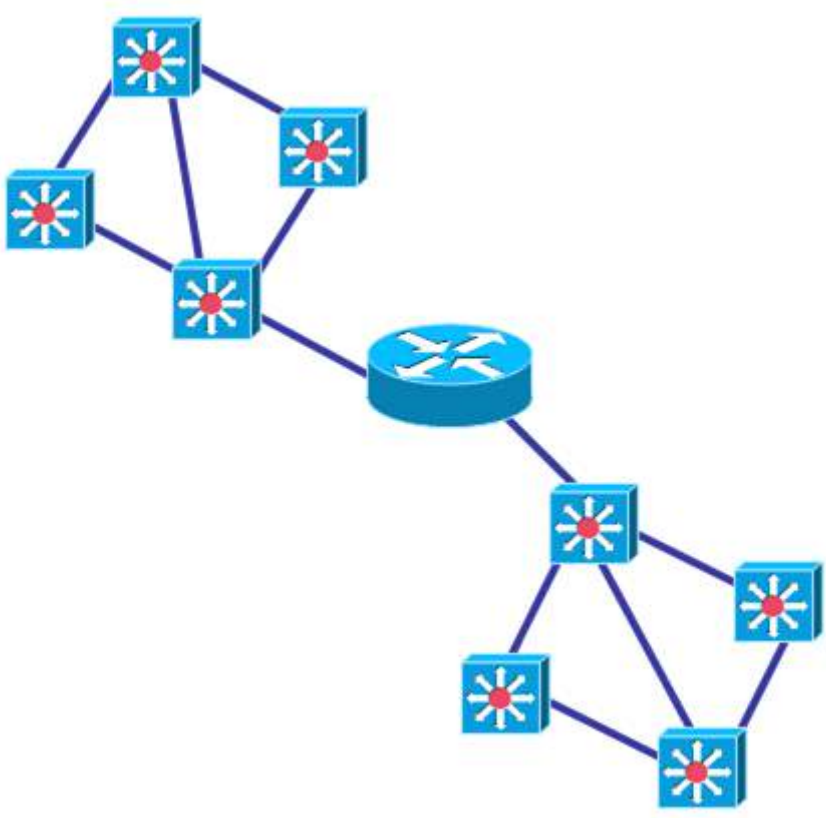


在上图的交换网络中，由于存在多台交换机，并且交换机之间有多条冗余链路，因此，只要在网络中找一台交换机充当核心，也就是相当于 hub-spoke 网络中的 Hub，而其它交换机则留出一条活动链路到核心交换机即可，其它链路全部被 block，

当留出的活动链路失效之后，再启用 block 链路作为备份。上图中 SW1 被选作交换网络中的核心，而其它交换机则只留一条活动链路到核心交换机，只要其它交换机与核心交换机是通的，就证明交换机之间一定是通的。图中红色的连路表示被留出的普通交换机到核心交换机的活动链路，蓝色链路表示被 block 掉的链路，只要红色链路是通的，就表示整个网络都是通的，当某条红色链路断掉以后，只要启用相应的蓝色链路代替即可，也就实现了网络的冗余功能。

通过上述的解释，STP 要构建出无环的交换网络，就必须在网络中选出一台交换机做为核心交换机，STP 称其为 Root，也就是根，功能相当于 hub-spoke 网络中的 Hub。其它不是 Root 的交换机则需要留出一条活动链路去往根交换机，因为只要普通交换机到根是通的，到其它交换机也就是通的。

需要说明的是，只有在一个三层网络中，广播能够到达的范围内，才需要进行相同的 STP 计算与选举，也就是一个广播域内独立选举 STP：



上图中，因为网络被路由器分割成两个广播域，所以在两个网段中，需要进行

独立的 STP 计算与选举。

STP 在计算与选举时，只会留下唯一一条活动链路，将其它所有多余链路全部 block，所以 STP 要确定两点之间是否存在多条链路，因为只有两点之间有多条链路时，才有链路需要被 block。要确认两点之间网络是否通畅，只要发送数据作个测试即可得到答案，而要确认两点之间是否有多条链路，方法还是发送数据作个测试就能得到答案。当然，要测试两点之间是否有多条链路，需要发送特殊的数据来做测试，比如给数据包都做上相同的标记，然后发出去，如果交换机同时从多个接口收到相同标记的数据包，很显示，交换机与发送者之间就是存在多条链路的，因此需要靠 STP 计算来断开多余链路。

STP 在发送数据包测试网络是否有多条链路，是靠发送 bridge protocol data units (BPDUs)来完成的，同台交换机发出去的 BPDU 都被做上了相同的标记，只要任何交换机从多个接口收到相同标记的 BPDU，就表示网络中有冗余链路，因此需要 STP 断开多余链路。BPDU 数据包里面有以下信息：

根交换机的 bridge ID。

发送交换机的 bridge ID 。

到根交换机的 Path Cost。

发送接口以及优先级。

Hello、forward delay、max-age 时间。

同台交换机发出的 BPDU，bridge ID 都是一样的，因为是用来标识自己的，其中 bridge ID 由两部分组成：Bridge 优先级和 MAC 地址，默认优先级为 32678。

交换机上的每个端口也是有优先级的，默认为 128，范围为 0-255。

注：在 STP 协议中，所有优先级数字越小，表示优先级越高，数字越大，优先级越低。

STP 在计算网络时，需要在网络中选举出根交换机（Root）根端口（Root Port），以及指定端口(Designated Port)，才能保证网络的无环，选举规则分别如下：

根交换机（Root）

在同一个三层网络中需要选举，即一个广播域内要选举，并且一个网络中只能选举一台根交换机。Birdge-ID 中优先级最高（即数字最小）的为根交换机，优先级范围为 0-65535，如果优先级相同，则 MAC 地址越小的为根交换机。

根端口（Root Port）

所有非根交换机都要选举，非根交换机上选举的根端口就是普通交换机去往根交换机的唯一链路，选举规则为 到根交换机的 Path Cost 值最小的链路，如果多条链路到达根交换机的 Path Cost 值相同，则选举上一跳交换机 Bridge-ID 最小的链路，如果是经过的同一台交换机，则上一跳交换机 Bridge-ID 也是相同的，再选举对端端口优先级最小的链路，如果到达对端的多个端口优先级相同，最后选举交换机对端端口号码最小的链路。

指定端口(Designated Port)

在每个二层网段都要选举，也就是在每个冲突域需要选举，简单地理解为每条连接交换机的物理线路的两个端口中，有一个要被选举为指定端口，每个网段选举指定端口后，就能保证每个网段都有链路能够到达根交换机，选举规则和选举根端口一样，即：到根交换机的 Path Cost 值最小的链路，如果多条链路到达根交换机的 Path Cost 值相同，则选举上一跳交换机 Bridge-ID 最小的链路，如果是经过的同一台交换机，则上一跳交换机 Bridge-ID 也是相同的，再选举对端端口优先级最小的链路，如果到达对端的多个端口优先级相同，最后选举交换机对端端口号码最小的链路。

在 STP 选出根交换机，根端口以及指定端口后，其它所有端口全部被 Block，为了防止环路，所以 Block 端口只有在根端口或指定端口失效的时候才有可能被启用。

交换机上的端口，根据端口的带宽不同，Path Cost 值也不同，以下参数为标准：

10 Mb/s：100

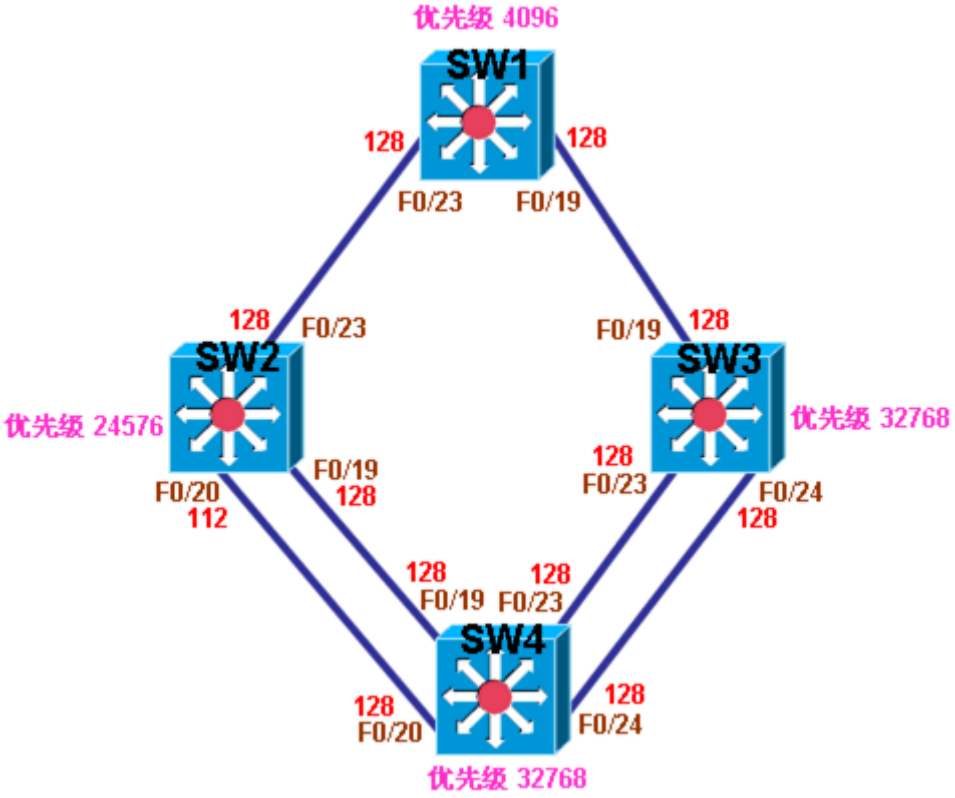
100 Mb/s：19

1000 Mb/s：4

10000 Mb/s：2

可以看出，带宽越高，被选为根端口和指定端口的几率就越大，所以经过 STP 选举后，活动的链路总是性能最好的，其它被 Block 掉的端口，将在活动端口失效时被启用。

以下图为例来看 STP 计算：



上图的网络环境中，运行 STP 后，则选举如下角色：（所有链路为 100 Mb/s，即 Path Cost 值为 19）

根交换机（Root）

因为 4 台交换机的优先级分别为 SW1（4096），SW2（24576），SW3（32768），SW4（32768），选举优先级最高的（数字最低的）为根交换机，所以 SW1 被选为根交换机，如果优先级相同，则比较 MAC 地址。

根端口（Root Port）

根端口需要在除 SW1 外的非根交换机上选举。

SW2 上从端口 F0/23 到达根的 Path Cost 值为 19，从 F0/19 和 F0/20 到达根的 Path Cost 值都为 19×3=57。因此，F0/23 被选为根端口。

SW3 上从端口 F0/19 到达根的 Path Cost 值为 19，从 F0/23 和 F0/24 到达根的 Path Cost 值都为 19×3=57。因此，F0/19 被选为根端口。

SW4 上从所有端口到达根的 Path Cost 值都为 19×2=38，所以从比较 Path Cost 值，无法选出根端口，接下来比较上一跳交换机 Bridge-ID，也就是比较 SW2 与 SW3 的 Bridge-ID，所以选择往 SW2 的方向，然而通过端口 F0/19 和 F0/20 都可以从 SW2 到达根交换机，所以接下来比较端口 F0/19 和 F0/20 对端交换机端口的优先级，因为 SW2 的 F0/19 端口优先级为 128，而 F0/20 的端口优先级为 112，所以 SW4 选择连接 SW2 的 F0/20 的端口为根端口，即 SW4 的 F0/20 为根端口，如果此步还选不出，SW4 将根据对端端口号做出决定，也就是 F0/19 和 F0/20，数字小的为根端口，也就是 F0/19。

指定端口(Designated Port)

每个网段（每个冲突域），或理解为每条线路都要选举指定端口。

在根交换机 SW1 连接 SW2 的网段与连接 SW3 的网段中，当然是根自己的端

口离自己最近，所以这两个网段中，选举根交换机上的端口为指定端口，因此，根交换机上所有的端口都应该是指定端口。

在 SW3 连接 SW4 的两个网段中，同样也是 SW3 上的两个端口离根交换机最近，所以在这两个网段中，选举 SW3 上的端口为指定端口。

在 SW2 连接 SW4 的两个网段中，同样也是 SW2 上的两个端口离根交换机最近，所以在这两个网段中，选举 SW2 上的端口为指定端口。

注：根交换机上所有的端口最终都为指定端口。

其它既不是根端口，也不是指定端口的落选的端口，就是 SW4 上的 F0/19，F0/23，F0/24，都将被 STP 放入 Blocking 状态，不为用户提供数据转发，以此来防止环路。最终的网络，构建出了任何两点之间，都是单链路的环境，不会有环路，当使用中的链路失效时，Blocking 的端口可以代替原端口。上图的 STP 选举结果如下：

根交换机（Root）

SW1

根端口（Root Port）

SW2：F0/23　SW3：F0/19　SW4：F0/20

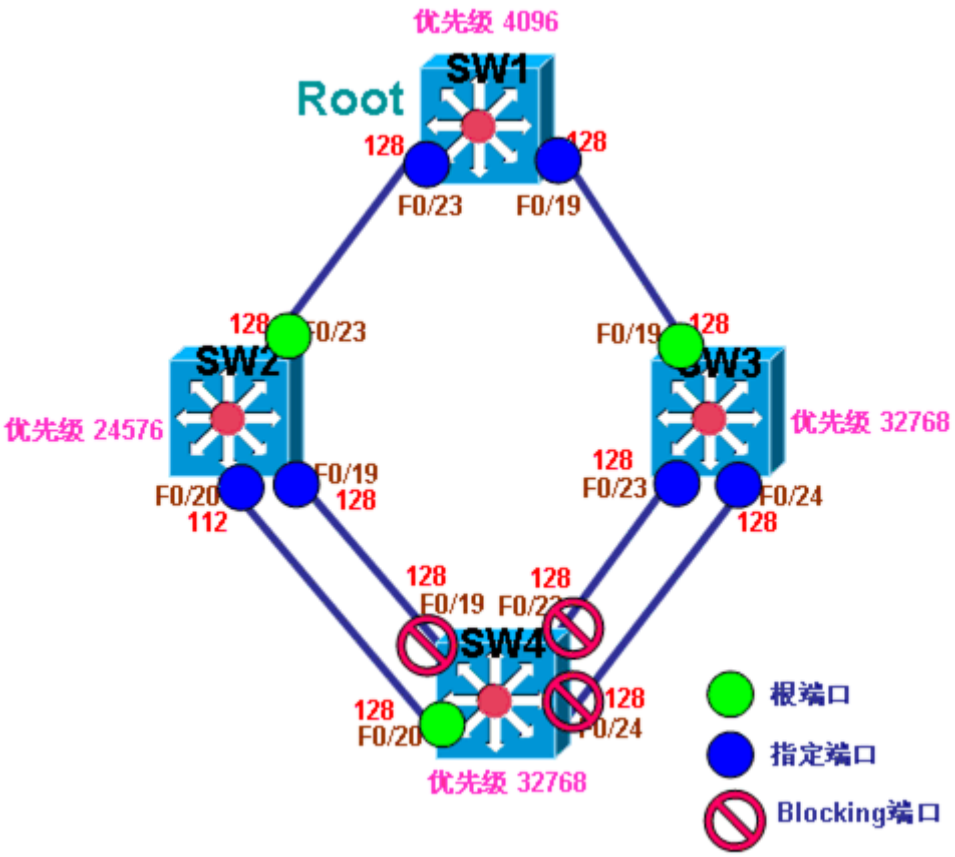
指定端口(Designated Port)

SW1：F0/19，F0/23　SW2：F0/19，F0/20　SW3：F0/23，F0/24

Blocking 端口

SW4：F0/19，F0/23，F0/24

结果图如下：



注：一个端口，在 STP 中只能处于一种角色，不可能是两种角色。

在交换机启动后，端口要过渡到转发状态，需要经历以下的状态：

1 从 initialization（初始化）到 blocking

2 从 blocking 到 listening 或 disabled

3 从 listening 到 learning 或 disabled

4 从 learning 到 forwarding 或 disabled

被 Disabled 的接口相当于关闭了，每个状态有如下功能：

Blocking

丢弃所有收到的数据帧，不学习 MAC 地址，能收 BPDU 但不发 BPDU。

Listening

丢弃所有收到的数据帧，不学习 MAC 地址，能收 BPDP 的处理 BPDU，并进行 STP 计算。

Learning

丢弃所有收到的数据帧，会学习 MAC 地址，能收 BPDU 和处理 BPDU。

Forwarding

也就是正常转发状态，能转发收到的数据帧，能学习 MAC 地址，接收并处理 BPDU。

Disabled

丢弃所有收到的数据帧，不学习 MAC 地址，能收 BPDU，除此之外不会再做其它的。

当交换机启动后，都认为自己是根交换机，然后从所有接口向网络中发送 BPDU，称为 configuration BPDU，所以 configuration BPDU 是根交换机发出的。当交换机收到更优 Bridge-ID 的 configuration BPDU，会将它从自己所有接口转发出去，并保存在接口，如果收到差的 configuration BPDU，则全部丢掉，所以在交换网络中，只有根交换机的 BPDU 在转发，其它普通交换机的 BPDU 不会出现在网络中。

根交换机的 BPDU 会在每个 hello 时间往网络中发送一次，hello 时间默认为 2 秒钟，也就是交换机的 BPDU 会在每 2 秒钟往网络中发送一次，如果普通交换机在 max-age 时间内没有收到根交换机的 BPDU，则认为根交换机已经失效，便开始重新选举 BPDU，默认 max-age 时间为 20 秒，即 10 倍 hello 时间。

除了 hello 时间和 max-age 时间外，还有一个 forward delay 时间，默认为 15 秒，接口在经过 Listening 和 Learning 状态时，都会分别停留一个 forward delay 时间，也就是说接口从 Listening 状态到 Learning 状态，最后变成转发状态，需要经过两个 forward delay 时间共计 30 秒。

因为 STP 有多个版本，不同版本的 STP，在操作和运行上，会有所不同，但是需要说明，无论什么版本的 STP，对根交换机，根端口以及指定端口的选举规则完全是一样的，下面分别详细介绍各版本的运行过程：

Common Spanning Tree (CST)

CST 的协议号为 IEEE 802.1D，如果交换机运行在 CST，交换机只进行一次 STP 计算，无论交换机上有多少个 VLAN，所有流量都会走相同的路径。

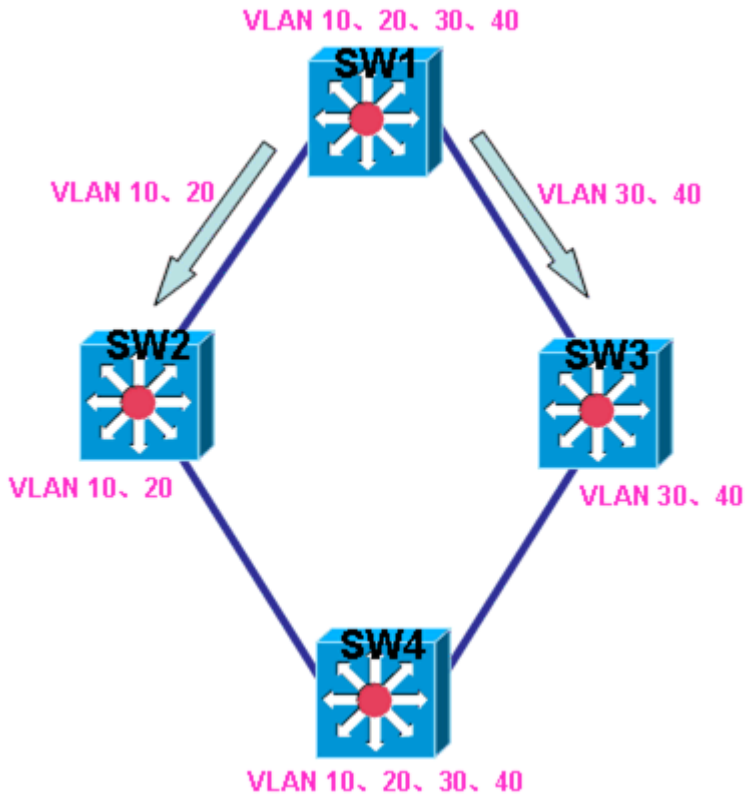
Rapid Spanning Tree Protocol (RSTP)

RSTP 是快速 STP，协议号为 IEEE 802.1w，在运行 CST 时，端口状态 blocking、listening、disabled 都不发送数据，RSTP 将这三个状态归为一个状态，discarding 状态。其次之外就是 learning 和 forwarding 状态，所以 RSTP 端口状态为 discarding、learning 和 forwarding。

当运行 CST 时，如果根交换机失效了，那么需要等待 10 个 hello 时间，也就是 20 秒收不到根交换机 BPDU 才能发现，再将 block 的端口过滤到 forwarding 状态，还需要经过两个 forward delay 时间共计 30 秒，所以 CST 在网络出现故障时，要经过 50 秒才能启用 block 端口，而 RSTP 则只需要在 3 个 hello 时间，即 6 秒收不到根交换机 BPDU，便认为根交换机已经失去连接，就立刻启用 discarding 状态的接口，RSTP 在根交换机失效后，并不会进行完整的 STP 计算，会在该启用备用端口时立即启用，因此网络收敛速度快，RSTP 会在低于 1 秒的时间内恢复网络。

Per-VLAN Spanning-Tree plus (PVST+)

PVST+是思科自己的协议，在之前有一个 PVST，但由于 PVST 只能支持 ISL Trunk，所以思科为了扩展 PVST 支持 IEEE 802.1Q，诞生了 PVST+，在多数三层交换机，如 3550、3560 及以上型号，默认运行的 STP 版本为 PVST+。PVST+是基于 CST（IEEE 802.1D）运行的，但运行了 PVST+的交换机并不像 CST 那样只进行一次 STP 计算，PVST+会在每个 VLAN 进行一次 STP 计算，也就是会根据 VLAN 数的不同，计算 STP 的次数也不同，并且每个 VLAN 的 STP 信息是单独保存的。请看下图：



在上图的网络中，各台交换机上都有 VLAN 10，VLAN 20，VLAN 30，VLAN 40，在运行 CST 的情况下，因为只进行一次 STP 计算，所以 SW1 到 SW4 的流量要么从 SW2 走，要么从 SW3 走，在这种情况下，流量只能走同一条路径，而另一条路径完全被空闲而得不到利用。

当在上图的网络中运行 PVST+后，因为 PVST+会在每个 VLAN 进行不同的 STP 计算，称为 STP 实例（instance），所以可以控制每个 VLAN 流量的路径走向。上图中，就可以通过 PVST+控制 SW1 的 VLAN10 和 VLAN20 从连接 SW2 的接口到达 SW4，控制 SW1 的 VLAN 30 和 VLAN 40 从连接 SW3 的接口到达 SW4，这样之后，将不同的 VLAN 流量分担到不同的路径，即实现了负载均衡，也通过 STP 避免了环路。

重点说明：

PVST+ 只支持 128 个实例(instance)，如果交换机上配置的 VLAN 数超过 128 个，那么 128 个以外的 VLAN 将没有 STP 在运行，所以此时剩余的 VLAN 将出现环路。可以单独在特定的 VLAN 上打开或关闭 STP 功能，即使一台没有运行 STP 的交换机或没有运行 STP 的 VLAN，在收到 BPDU 时，也会转发的，所以在对单个 VLAN 进行开启或关闭 STP 时，请确保交换机能够计算出无环的网络，否则网络将出现预想不到的故障。

在 PVST+可以配置全局关闭某 VLAN 的 STP，如关闭 VLAN 10 的 STP

no spanning-tree vlan 10，恢复使用命令 spanning-tree vlan 10

Extended System ID

默认交换机的 Bridge-ID 的优先级为 32768，当开启 Extended System ID 功能后，每个 VLAN 的默认的 Bridge-ID 优先级就不再是 32768 了，需要再加上 VLAN 号码，如 VLAN 1 的 Bridge-ID 优先级就是 32768+1=32769，VLAN 8 的 Bridge-ID 优先级就是 32768+8=32776。

如果网络中即有开启了 Extended System ID 功能的交换机，也有关闭的，那么关闭 Extended System ID 功能的交换机有更大的机会成为根交换机，因为自己默认的优先级就比其它开启了 Extended System ID 功能的优先级更高（数字更小）。

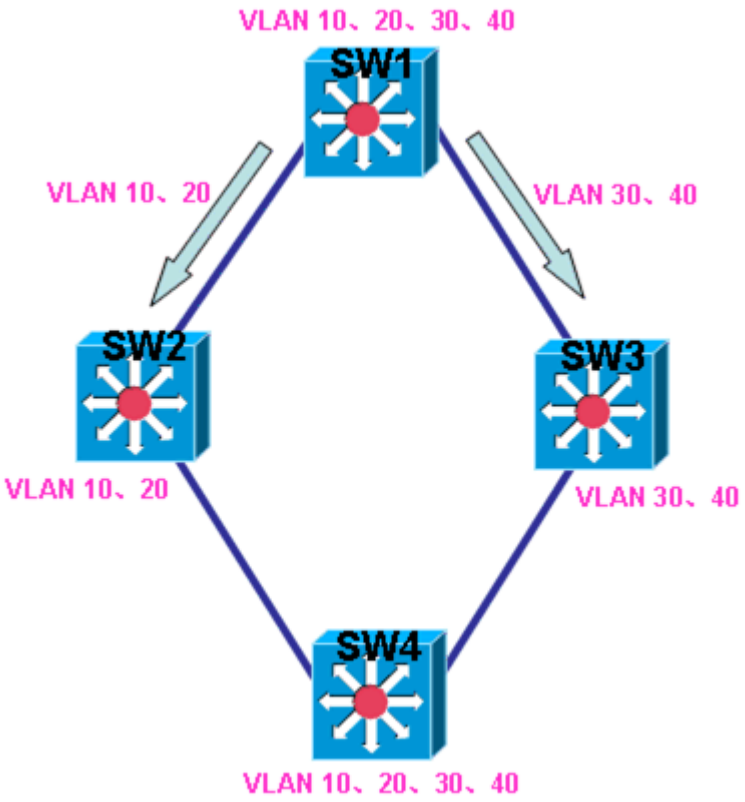
返回目录

Rapid PVST+

Rapid PVST+就是具有 RSTP 特性的 PVST+，是像 RSTP 一样基于 IEEE 802.1w 运行的，其它所有运行与规则与 PVST+完全相同，不再做详细介绍。

Multiple Spanning Tree Protocol (MSTP)

MSTP 的协议号为 IEEE 802.1s，因为在交换机存在多个 VLAN 时，CST 会将所有流量放在单条路径中传输，而 PVST+则可以通过为每个 VLAN 运行一个 STP 实例，从而将不同 VLAN 的流量放在不同的路径上传输。但正是由于 PVST+为每个 VLAN 都运行了一个 STP 实例，可能会多达 128 个 STP 实例，所以 PVST+会极其消耗系统资源。比如交换机上有 20 个 VLAN，而 PVST+会维护 20 个 STP 实例，但是这 20 个 VLAN 的流量也许只需要被分担到几条不同路径上，那就只需要维护几个 STP 实例即可，而并不需要维护 20 个 STP 实例。MSTP 正因为这个原因，将需要进行相同 STP 计算的 VLAN 映射到同一个 STP 实例中，即无论有多少个 VLAN，只要实际需要多少条不同的路径，就根据需要的路径维护相同的 STP 实例数，从而大大节省系统资源，如下图：



还是以此图为例，因为各台交换机上都有 VLAN 10，VLAN 20，VLAN 30，VLAN 40，为了能够在 SW1 上让不同 VLAN 的流量从不同的路径到达 SW4，所以可以运行 PVST+，将流量分担到不同的路径上，即 SW1 通过 PVST+将 VLAN10 和 VLAN20

的流量从连接 SW2 的接口到达 SW4，将 VLAN 30 和 VLAN 40 的流量从连接 SW3 的接口到达 SW4，但 PVST+维护了 4 个 STP 实例，才达到此效果，不难看出，其实网络中只有两个不同的路径，VLAN 10 和 VLAN 20 的路径完全是相同的，VLAN 30 和 VLAN 40 的路径也是完全相同的，此时，MSTP 就可以通过将相同的 VLAN 映射到同一个 STP 实例，如将 VLAN 10 和 VLAN 20 映射到一个实例，再将 VLAN 30 和 VLAN 40 映射到另外一个实例，总共只有两个 STP 实例，既像 PVST+那样实现了负载均衡的效果，也节省了系统资源。

MSTP 是在 RSTP 的基础之上运行的，所以具有快速收敛的功能，但不能不运行 RSTP 时运行 MSTP，RSTP 是随着开启 MSTP 时自动开启的。MSTP 最多支持 65 个 STP 实例，但是映射到实例的 VLAN 数量是没有限制的。默认所有 VLAN 都在实例 0。

MSTP 还需要通过分区域管理，即 region，交换机要在同一 region 进行相同的 STP 计算，必须 region name 和 revision number 一致，最重要的是 VLAN 和实例的映射也要一致，否则 STP 计算出来的网络，将不是你想要的网络，一个 VLAN 只能被映射到一个实例，一个网络可以有多个 MSTP revision，便于各自独立。

拓朴变更

当网络中的链路出现变化时，也就需要进行新的 STP 计算，并且由于交换机的 MAC 地址在表中的老化时间默认为 300 秒（5 分钟），所以当原有的链路发生变化后，MAC 地址与接口的对应关系也会发生变化，因此不能再等 5 分钟才更新，所以基于拓朴变化的因素，还需要将 MAC 地址的老化时间设置的更短，此动作在 STP 拓朴变更时，会自动更改为 forward_delay 的时间。

当网络链路发生变化后，必须进行新的 STP 计算，但是在正常的 STP 状态下，只有根交换机才能往网络里发送 BPDU，称为 configuration BPDU，而普通交换机只有接收 configuration BPDU 的权限，并不能向网络中发送 BPDU。但是当交换机检测到链路变化时，可以通知网络中的根交换机，此时可以发送一种特殊的 BPDU，叫做 topology change notification (TCN)，也就是 TCN BPDU。TCN BPDU 是用来告诉根交换机网络链路有变化，因此 TCN BPDU 只能从根端口发出去，如果接收者不是根交换机，则必须回复一个确认消息，这个消息是一个设置了 TCA 位的 configuration BPDU，然后自己再从根端口向根发送 TCN BPDU，直到根收到为止，当根收到 TCN BPDU 后，需要回复该 BPDU，方式为发送一个设置了 TC 位的

configuration BPDU。

其中，TCN 是一种特殊的 BPDU，而 TCA 只是设置了 TCA 位的 configuration BPDU，TC 也只是设置了 TC 位的 configuration BPDU。最终 STP 网络中，出现了两种 BPDU，即 TCN BPDU 和 configuration BPDU。

注：

★在配置 STP 时，Bridge-ID 的优先级，端口优先级，hello 时间，max-age 时间，forward delay 时间都可以手工修改，而 Bridge-ID 的优先级必须为 4096 的整数倍，端口优先级必须为 16 的整数倍。

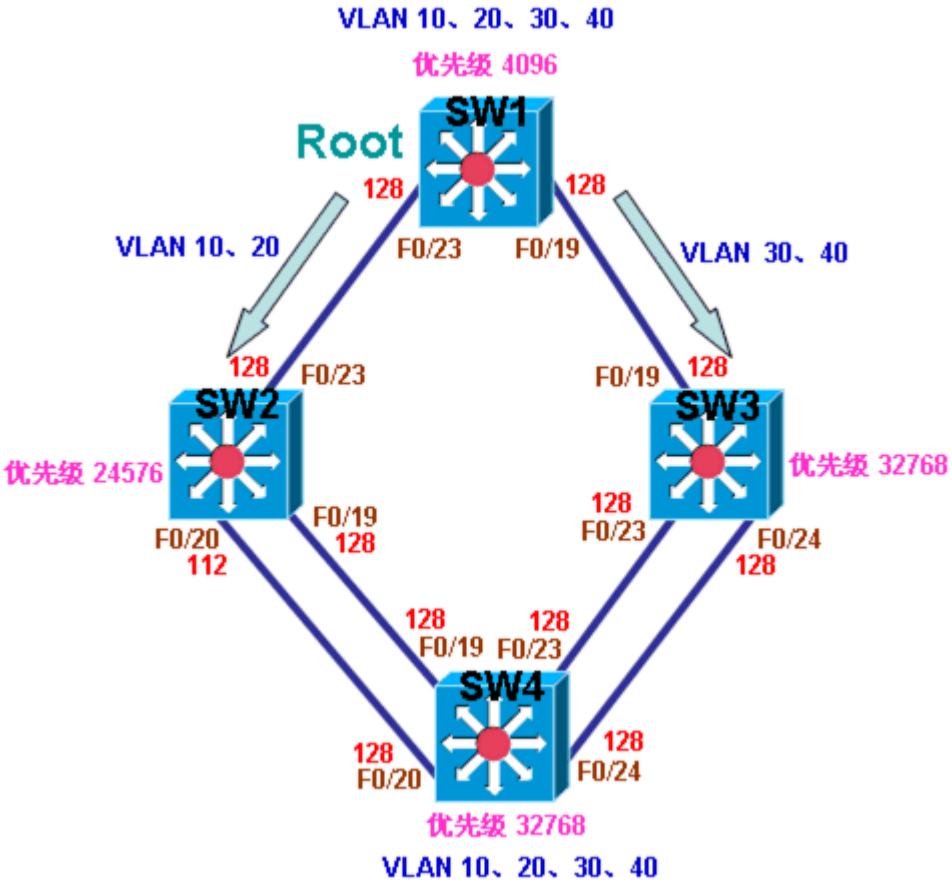
★在修改时，PVST+可以基于每个 VLAN 修改，而 MSTP 则只能基于实例，而不能基于 VLAN，因为一个实例会有多个 VLAN。

可以通过命令来强制指定某台交换机为根交换机，当使用命令强制指定某交换机为根后，此交换机将通过修改一个比当前根交换机更高优先级的 Bridge-ID，以此来抢夺根交换机的角色，如果命令再到别的交换机上输入，那么那台交换机将再次抢夺根交换机的角色，因为它可以修改自己的 Bridge-ID 比当前根更高的优先级，所以此命令最后在网络中的哪台交换机上输入后，哪台交换机就能成为根交换机，但是也有个限度，因为交换机的 Bridge-ID 不能自动改的比 1 小，又不能改 MAC 地址，所以如果需要修改优先级到 1 以下才能抢夺根交换机的角色，那么此命令将提示错误。

注：链路的全双工与半双工，在 STP 中，被分为不同的链路类型，如果是全双工（full-duplex），叫做 point-to-point(P2p)，如果是半双工，叫做（half-duplex）。接口下可以手工更改：spanning-tree link-type point-to-point。

配置

配置 PVST+



说明：以上图为例，配置 PVST+，默认交换机上都配置有 VLAN 10，VLAN 20，VLAN 30，VLAN 40，要求控制 SW1 与 SW4 之间的流量路径为 VLAN 10 和 VLAN 20 从 SW1—SW2—SW4，VLAN 30 和 VLAN 40 从 SW1—SW3—SW4。

注：默认为 PVST+，所以 STP 版本不用改。

1.配置各交换机优先级（只能为 4096 的整数倍）

（1）配置 SW1 在所有 VLAN 的优先级为 4096

sw1(config)#spanning-tree vlan 10-40 priority 4096

（2）配置 SW2 在所有 VLAN 的优先级 24576

sw2(config)#spanning-tree vlan 10-40 priority 24576

（3）配置 SW3 在所有 VLAN 的优先级 32768

sw3(config)#spanning-tree vlan 10-40 priority 32768

（4）配置 SW4 在所有 VLAN 的优先级 32768

sw4(config)#spanning-tree vlan 10-40 priority 32768

2.配置 SW2 的 F0/20 的端口优先级（必须为 16 的整数倍）

（1）在所有 VLAN 将 SW2 的 F0/20 的端口优先级配置为 112

sw2(config)#int f0/20

sw2(config-if)#spanning-tree vlan 10-40 port-priority 112

3.查看根交换机

（1）查看根交换机 SW1

说明：因为现在 4 个 VLAN 的配置是一样的，结果也是一样的，所以只提供一个 VLAN 的结果：

sw1#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4106 (priority 4096 sys-id-ext 10)

Address 001a.6c6f.fb00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface Role Sts Cost Prio.Nbr Type

---------- ---- --- -------- ------- --------------------------

Fa0/23 Desg FWD 19 128.25 P2p

（输出被省略）

sw1#

说明：从结果中看出，SW1 手工配置的优先级为 4096，但由于 Extended System ID 功能，所以优先级加上了 VLAN 号码 10，结果优先级变为 4106，因为优先级在网络中数字最小，所以自己就是当前网络的根交换机。

4.查看根端口

（1）查看 SW2 的根端口

sw2#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 19

Port 23 (FastEthernet0/23)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 24586 (priority 24576 sys-id-ext 10)

Address 0013.805c.9d00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Desg	FWD	19	128.19	P2p
Fa0/20	Desg	FWD	19	112.20	P2p
Fa0/23	Root	FWD	19	128.23	P2p

（输出被省略）

sw2#

说明：因为 SW2 上从端口 F0/23 到达根的 Path Cost 值为 19，从 F0/19 和 F0/20 到达根的 Path Cost 值都为 19×3=57。因此，F0/23 被选为根端口。

（2）查看 SW3 的根端口

sw3#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 19

Port 21 (FastEthernet0/19)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)

Address 001a.a256.f300

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 15

Interface Role Sts Cost Prio.Nbr Type

---------- ---- --- -------- ------- ------------------------

Fa0/19 Root FWD 19 128.21 P2p

Fa0/23 Desg FWD 19 128.25 P2p

Fa0/24 Desg FWD 19 128.26 P2p

（输出被省略）

sw3#

说明：因为 SW3 上从端口 F0/19 到达根的 Path Cost 值为 19，从 F0/23 和 F0/24 到达根的 Path Cost 值都为 19×3=57。因此，F0/19 被选为根端口。

（3）查看 SW4 的根端口

sw4#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 38

Port 22 (FastEthernet0/20)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Altn	BLK	19	128.21	P2p
Fa0/20	Root	FWD	19	128.22	P2p
Fa0/23	Altn	BLK	19	128.25	P2p
Fa0/24	Altn	BLK	19	128.26	P2p

（输出被省略）

sw4#

说明：因为 SW4 上从所有端口到达根的 Path Cost 值都为 19×2=38，所以从比较 Path Cost 值，无法选出根端口，接下来比较上一跳交换机 Bridge-ID，也就是比较 SW2 与 SW3 的 Bridge-ID，所以选择往 SW2 的方向，然而通过端口 F0/19 和 F0/20 都可以从 SW2 到达根交换机，所以接下来比较端口 F0/19 和 F0/20 对端交换机端口的优先级，因为 SW2 的 F0/19 端口优先级为 128，而 F0/20 的端口优先级为 112，所以 SW4 选择连接 SW2 的 F0/20 的端口为根端口，即 SW4 的 F0/20 为根端口

5.查看指定端口

（1）查看 SW1 的指定端口

sw1#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4106 (priority 4096 sys-id-ext 10)

Address 001a.6c6f.fb00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Desg	FWD	19	128.21	P2p
Fa0/23	Desg	FWD	19	128.25	P2p

（输出被省略）

SW1#

说明：在根交换机 SW1 连接 SW2 的网段与连接 SW3 的网段中，当然是根自己的端口离自己最近，所以这两个网段中，选举根交换机上的端口为指定端口，因此，根交换机上所有的端口都应该是指定端口。

（2）查看 SW2 的指定端口

sw2#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 19

Port 23 (FastEthernet0/23)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 24586 (priority 24576 sys-id-ext 10)

Address 0013.805c.9d00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Desg	FWD	19	128.19	P2p
Fa0/20	Desg	FWD	19	112.20	P2p
Fa0/23	Root	FWD	19	128.23	P2p

（输出被省略）

Sw2#

说明：在 SW2 连接 SW4 的两个网段中，同样也是 SW2 上的两个端口离根交换机最近，所以在这两个网段中，选举 SW2 上的端口为指定端口。

（3）查看 SW2 的指定端口

sw3#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 19

Port 21 (FastEthernet0/19)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)

Address 001a.a256.f300

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface Role Sts Cost Prio.Nbr Type

---------- ---- --- -------- ------- ------------------------

Fa0/19 Root FWD 19 128.21 P2p

Fa0/23 Desg FWD 19 128.25 P2p

Fa0/24 Desg FWD 19 128.26 P2p

（输出被省略）

Sw3#

说明：在 SW3 连接 SW4 的两个网段中，同样也是 SW3 上的两个端口离根交换机最近，所以在这两个网段中，选举 SW3 上的端口为指定端口。

（4）查看 SW2 的指定端口

sw4#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 38

Port 22 (FastEthernet0/20)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Altn	BLK	19	128.21	P2p
Fa0/20	Root	FWD	19	128.22	P2p
Fa0/23	Altn	BLK	19	128.25	P2p
Fa0/24	Altn	BLK	19	128.26	P2p

（输出被省略）

sw4#

说明：除了根端口和指定端口，其它的都为落选端口，也就是 SW4 上的 F0/19，F0/23，F0/24，都将被 STP 放入 Blocking 状态，不为用户提供数据转发，以此来防止环路

6.调整 VLAN 30 和 VLAN 40 的路径为 SW1—SW3—SW4。

说明：因为默认 4 个 VLAN 相同配置，所以全部和 VLAN 10 一样，路径为 SW1—SW2—SW4，现只对 VLAN 30 和 VLAN 40 做修改，以调整路径为 SW1—SW3—SW4。

（1）修改 SW3 在 VLAN 30 和 VLAN 40 的 Bridge-ID 优先级

说明：因为选举根端口和指定端口的第一步为比较到根的 Path Cost 值，第二步为比较上一跳 Bridge-ID，而 SW4 从 SW2 到 SW1 和从 SW3 到 SW1 的 Path Cost

值全部是一样的，所以可以选择修改 SW3 在 VLAN 30 和 VLAN 40 的 Bridge-ID 优先级来做调整：

sw3(config)#spanning-tree vlan 30,40 priority 20480

说明：SW3 在 VLAN 30 和 VLAN 40 的 Bridge-ID 优先级必须比 SW2 的 Bridge-ID 优先级小，才能将 VLAN 30 与 VLAN 40 的流量引过来。

7.查看修改后的 VLAN 30 与 VLAN 40 的路径

说明：因为 VLAN 30 与 VLAN 40 相同配置，所以只查看一个 VLAN 即可。

（1）查看 SW4 上 VLAN 10 与 VLAN 30 的路径对比

sw4#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 38

Port 22 (FastEthernet0/20)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Altn	BLK	19	128.21	P2p
Fa0/20	Root	FWD	19	128.22	P2p
Fa0/23	Altn	BLK	19	128.25	P2p
Fa0/24	Altn	BLK	19	128.26	P2p

VLAN0030

Spanning tree enabled protocol ieee

Root ID Priority 4126

Address 001a.6c6f.fb00

Cost 38

Port 25 (FastEthernet0/23)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32798 (priority 32768 sys-id-ext 30)

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

```
Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Fa0/19           Altn BLK 19        128.21   P2p
Fa0/20           Altn BLK 19        128.22   P2p
Fa0/23           Root FWD 19        128.25   P2p
Fa0/24           Altn BLK 19        128.26   P2p
```

（输出被省略）

```
sw4#
```

说明：可以看到，SW4 的 VLAN 10 还是保持原来的路径 SW4—SW2—SW1，而 VLAN 30 的路径已经变成 SW4—SW3—SW1 并且 VLAN 30 的根端口为 F0/23。

8.调整 STP 参数

（1）调整 SW4 在 VLAN 30 的根端口为 F0/24

说明：因为 SW4 在 VLAN 30 从 F0/23 和 F0/24 到达根的 Path Cost 值都为 19×2=38，所以从比较 Path Cost 值，无法选出根端口，接下来比较上一跳交换机 Bridge-ID，由于都是 SW3，所以 Bridge-ID 相同，接下来比较 F0/23 和 F0/24 对端交换机端口的优先级，但对方优先级都为 128，所以最后选择了本地端口号码小的，即 F0/23 比 F0/24 小，F0/23 被选为根端口，我们现在通过修改本地 F0/24 对端设备的端口优先级来调整路径，也就是修改 SW3 的 F0/24 的优先级：

```
sw3(config)#int f0/24
sw3(config-if)#spanning-tree vlan 30 port-priority 112
```

说明：端口优先级为 16 的整数倍。

（2）查看 SW4 在 VLAN 30 的根端口

sw4#sh spanning-tree vlan 30

VLAN0030

Spanning tree enabled protocol ieee

Root ID Priority 4126

Address 001a.6c6f.fb00

Cost 38

Port 26 (FastEthernet0/24)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32798 (priority 32768 sys-id-ext 30)

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Altn	BLK	19	128.21	P2p
Fa0/20	Altn	BLK	19	128.22	P2p
Fa0/23	Altn	BLK	19	128.25	P2p
Fa0/24	Root	FWD	19	128.26	P2p

sw4#

说明：因为选举时，比较对方的端口优先级，成功调整了路径，此时的根端口已变为 F0/24。

（3）修改 SW4 在 VLAN 10 的 hello 时间为 3 秒，max-age 为 25 秒，forward delay 为 10 秒

sw4(config)#spanning-tree vlan 10 hello-time 3

sw4(config)#spanning-tree vlan 10 max-age 30

sw4(config)#spanning-tree vlan 10 forward-time 10

（4）查看 SW4 在 VLAN 10 的 hello 时间，max-age，forward delay

sw4#sh spanning-tree

（输出被省略）

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 001a.6c6f.fb00

Cost 38

Port 22 (FastEthernet0/20)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)

Address 001e.14cf.0980

Hello Time 3 sec Max Age 30 sec Forward Delay 10 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Altn	BLK	19	128.21	P2p
Fa0/20	Root	FWD	19	128.22	P2p
Fa0/23	Altn	BLK	19	128.25	P2p
Fa0/24	Altn	BLK	19	128.26	P2p

VLAN0020

Spanning tree enabled protocol ieee

Root ID Priority 4116

Address 001a.6c6f.fb00

Cost 38

Port 22 (FastEthernet0/20)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32788 (priority 32768 sys-id-ext 20)

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Altn	BLK	19	128.21	P2p
Fa0/20	Root	FWD	19	128.22	P2p
Fa0/23	Altn	BLK	19	128.25	P2p
Fa0/24	Altn	BLK	19	128.26	P2p

（输出被省略）

sw4#

说明：可以看到，修改的时间只对 VLAN 10 生效，VLAN 20 还是保持原状，PVST+可以单独修改每个 VLAN 的参数。

9.强制指定根与备份根

（1）指定 SW2 为 VLAN 10 的根

sw2(config)#spanning-tree vlan 10 root primary

（2）在 SW2 查看 VLAN 10 的根

sw2#sh spanning-tree vlan 10

VLAN0010

Spanning tree enabled protocol ieee

Root ID Priority 4106

Address 0013.805c.9d00

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4106 (priority 4096 sys-id-ext 10)

Address 0013.805c.9d00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 15

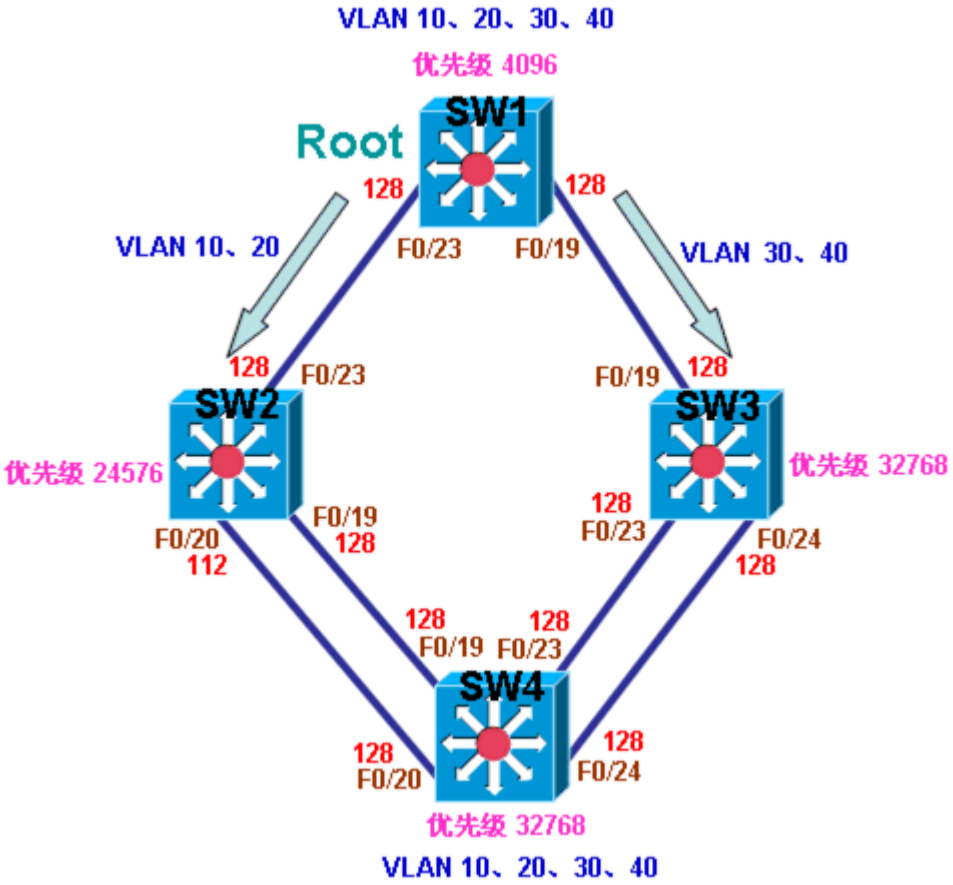
Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Desg	FWD	19	128.19	P2p
Fa0/20	Desg	FWD	19	112.20	P2p
Fa0/23	Desg	FWD	19	128.23	P2p

sw2#

重点说明：当使用命令强制指定某交换机为根后，此交换机将通过修改一个比当前根交换机更高优先级的 Bridge-ID，以此来抢夺根交换机的角色，如果命令再到别

的交换机上输入，那么那台交换机将再次抢夺根交换机的角色，因为它可以修改自己的 Bridge-ID 比当前根更高的优先级，所以此命令最后在网络中的哪台交换机上输入后，哪台交换机就能成为根交换机，但是也有个限度，因为交换机的 Bridge-ID 不能自动改的比 1 小，又不能改 MAC 地址，所以如果需要修改优先级到 1 以下才能抢夺根交换机的角色，那么此命令将提示错误。

配置 MSTP



1.配置 MSTP

（1）改变所有交换机的 STP 模式为 MSTP

Sw1(config)#spanning-tree mode mst

Sw2(config)#spanning-tree mode mst

Sw3(config)#spanning-tree mode mst

Sw4(config)#spanning-tree mode mst

（2）映射 VLAN 到实例

sw1(config)#spanning-tree mst configuration

sw1(config-mst)#name ccie

sw1(config-mst)#revision 1

sw1(config-mst)#instance 1 vlan 10,20

sw1(config-mst)#instance 2 vlan 30,40

说明：其它交换机配置和 SW1 配置完全相同，必须 region name 和 revision number 完全相同，否则属于不同的 region。

2.控制 VLAN 10 和 VLAN 20（实例 1）的路径为 SW1—SW2—SW4，VLAN 30 和 VLAN 40（实例 2）的路径为 SW1—SW3—SW4。

(1)配置 SW1 为实例 1 和实例 2 的根交换机

sw1(config)#spanning-tree mst 1 root primary

sw1(config)#spanning-tree mst 2 root primary

（2）控制 SW4 在实例 1 连 SW2 的端口 Path Cost 值为 10

sw4(config)#int range f0/19-20

sw4(config-if-range)#spanning-tree mst 1 cost 10

（3）控制 SW4 在实例 2 连 SW3 的端口 Path Cost 值为 10

sw4(config)#int ran f0/23-24

sw4(config-if-range)#spanning-tree mst 2 cost 10

3.查看 STP 状态

（1）查看根交换机

sw1#sh spanning-tree

（输出被省略）

MST1

Spanning tree enabled protocol mstp

Root ID Priority 24577

Address 001a.6c6f.fb00

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 24577 (priority 24576 sys-id-ext 1)

Address 001a.6c6f.fb00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Interface Role Sts Cost Prio.Nbr Type

---------- ---- --- -------- ------- --------------------------

Fa0/19 Desg FWD 200000 128.21 P2p

Fa0/23 Desg FWD 200000 128.25 P2p

MST2

Spanning tree enabled protocol mstp

Root ID Priority 24578

Address 001a.6c6f.fb00

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 24578 (priority 24576 sys-id-ext 2)

Address 001a.6c6f.fb00

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Interface Role Sts Cost Prio.Nbr Type

---------- ---- --- -------- ------- --------------------------

Fa0/19 Desg FWD 200000 128.21 P2p

Fa0/23 Desg FWD 200000 128.25 P2p

sw1#

说明：可以看到 SW1 已经成为实例 1 和实例 2 的根交换机。

（2）查看 SW4 的路径

sw4#sh spanning-tree

（输出被省略）

MST1

Spanning tree enabled protocol mstp

Root ID Priority 24577

Address 001a.6c6f.fb00

Cost 200010

Port 21 (FastEthernet0/19)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Root	FWD	10	128.21	P2p
Fa0/20	Altn	BLK	10	128.22	P2p
Fa0/23	Altn	BLK	200000	128.25	P2p
Fa0/24	Altn	BLK	200000	128.26	P2p

MST2

Spanning tree enabled protocol mstp

Root ID Priority 24578

Address 001a.6c6f.fb00

Cost 200010

Port 25 (FastEthernet0/23)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32770 (priority 32768 sys-id-ext 2)

Address 001e.14cf.0980

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/19	Altn	BLK	200000	128.21	P2p
Fa0/20	Altn	BLK	200000	128.22	P2p
Fa0/23	Root	FWD	10	128.25	P2p
Fa0/24	Altn	BLK	10	128.26	P2p

sw4#

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

说明：可以看到，实例 1 与实例 2 的流量已经分担到两条不同的路径上，既实现了与 PVST+相同的负载效果，也节省了系统资源，因为只有两个 STP 实例，而 PVST+要 4 个 STP 实例。

Spanning-Tree Feature

Port Fast

因为一个默认情况下的交换机端口，在交换机启动后，由于 STP 的原因，端口状态需要从 initialization（初始化）到 blocking，从 blocking 到 listening，从 listening 到 learning，从 learning 到 forwarding，其中经历了两个 forward delay，也就是说一个端口在交换机启动后，至少需要 30 秒后才能够为用户提供数据转发。对于一个连接了主机或服务器的端口，进行 STP 计算是毫无必要的，因为此类端口即使直接转发数据，也不会造成环路，并且 30 秒的时间对于需要立即传递数据的主机或服务器来说，是漫长的，因此，此类端口可以配置为跳过 STP 的计算，从而直接过渡到 forwarding 状态。

此类端口通常称为边缘端口，在思科交换机上，通过配置 Port Fast 功能，便可以使接口跳过 STP 的计算，从而直接过渡到 forwarding 状态。

access 接口和 Trunk 接口都可以配置 Port Fast 功能。如果将交换机连交换机的接口变成 Port Fast，则是制造环路。

当开启了 Port Fast 功能的接口，如果在接口上收到 BPDU 后，就认为对端连接的是交换机，而并非主机或服务器，因此默认在接口收到 BPDU 后会立即关闭该接口的 Port Fast 功能。

配置

1.在接口下配置 Port Fast

（1）将 SW2 的端口 F0/23 和 F0/24 改成三层接口

说明：因为如果 SW2 的端口是二层接口，那么就会向 SW1 发送 BPDU，最终会造成 SW1 由于收到 BPDU 而关闭 Port Fast 功能，所以就无法验证 Port Fast。

sw2(config)#int ran f0/23 - 24

sw2(config-if-range)#no switchport

说明：禁止从端口上向 SW1 发送 BPDU。

（2）在 SW1 的 F0/23 和 F0/24 上开启 Port Fast

说明：access 和 trunk 接口模式都可以配置

sw1(config)#int ran f0/23 - 24

sw1(config-if-range)#switchport mode access

sw1(config-if-range)#spanning-tree portfast

说明：将端口变为静态 access，再开 portfast。（无论什么模式的接口都可以开启 Port Fast）

（3）验证 Port Fast

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 enabled

sw1#

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 enabled

sw1#

说明：端口 F0/23 和 F0/24 已经开启 portfast 功能。

（4）在 SW2 的端口 F0/23 向 SW1 发送 BPDU

sw2(config)#int f0/23

sw2(config-if)#switchport

说明：只要将 SW2 的端口 F0/23 变成二层端口，便可以从此端口向外发送 BPDU。

（5）查看 SW1 的端口的 portfast 状态：

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 disabled

sw1#

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 enabled

sw1#

说明：可以看见，SW1 的端口 F0/23，在收到 BPDU 后，portfast 功能自动丢失。

2.在全局模式下配置 Port Fast（只能对 access 接口生效）

（1）将 SW2 的端口 F0/23 和 F0/24 改成三层接口

sw2(config)#int ran f0/23 - 24

sw2(config-if-range)#no switchport

(2)将 SW1 的端口配置为 access

sw1(config)#int ran f0/23 - 24

sw1(config-if-range)#switchport mode dynamic desirable

说明：因为对方是三层端口，在本地配置 DTP 后，会自动形成 access 模式。

（3）查看 SW1 的端口状态

sw1#show interfaces f0/23 switchport

Name: Fa0/23

Switchport: Enabled

Administrative Mode: dynamic desirable

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

（输出被省略）

sw1#

sw1#show interfaces f0/24 switchport

Name: Fa0/24

Switchport: Enabled

Administrative Mode: dynamic desirable

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: On

（输出被省略）

sw1#

说明：DTP 已经将本地端口变为 access 模式。

(4)在 SW1 全局开启 Port Fast

sw1(config)#spanning-tree portfast default

（5）查看 SW1 上端口的 Port Fast 状态

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 enabled

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 enabled

sw1#

说明：SW1 上的 access 端口受全局配置影响，已经变成 Port Fast 端口。

（6）验证同上，省略

BPDU Guard

因为开启了 Port Fast 功能的端口，会跳过 STP 的计算，从而直接过渡到 forwarding 状态。当端口连接的是主机或服务器，这样的操作不会有任何问题，但如果连接的是交换机，就会收到 BPDU，就证明在此接口开启 Port Fast 功能是错误的配置。为了杜绝此类错误配置，BPDU Guard 功能可以使端口在收到 BPDU 时，立即被 shutdown 或进入 err-disabled 状态。

BPDU Guard 可以在接口下或全局开启，但操作会有所不同。

如果 BPDU Guard 是全局开启，则只对 portfast 端口有影响，当 portfast 端口收到 BPDU 后，会 shutdown 此端口，需要注意，某些型号的交换机会将接口 error-disabled。

如果 BPDU Guard 是接口下开启，将对任何端口有影响，无论是正常端口还是 portfast 端口；当端口收到 BPDU 后，会变成 error-disabled 状态。

配置

1.在全局模式下配置 BPDU Guard（只对 Port Fast 端口有影响）

（1）将 SW2 的端口 F0/23 和 F0/24 改成三层接口

sw2(config)#int ran f0/23 - 24

sw2(config-if-range)#no switchport

说明：禁止从端口上向 SW1 发送 BPDU。

（2）将 SW1 的端口 F0/23 配置为 Port Fast，F0/24 为正常端口

sw1(config)#int f0/23

sw1(config-if)#switchport mode access

sw1(config-if)#spanning-tree portfast

sw1(config)#int f0/24

sw1(config-if)#switchport mode access

（3）查看 SW1 的端口 F0/23 和 F0/24 的状态

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 enabled

sw1#

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 disabled

sw1#

sw1#sh protocols f0/23

FastEthernet0/23 is up, line protocol is up

sw1#sh protocols f0/24

FastEthernet0/24 is up, line protocol is up

sw1#

说明：SW1 的端口 F0/23 已经变成 Port Fast 状态，而 F0/24 为正常端口，并且两个端口都为正常 UP 状态。

（4）在 SW1 全局开启 BPDU Guard（只对 Port Fast 端口有影响）

sw1(config)#spanning-tree portfast bpduguard default

（5）在 SW2 的端口 F0/23 和 F0/24 向 SW1 发送 BPDU，测试 BPDU Guard

sw2(config)#int ran f0/23 - 24

sw2(config-if-range)#switchport

说明：只要将 SW2 的端口 F0/23 和 F0/24 变成二层端口，便可以从此端口向外发送 BPDU。

（6）查看 SW1 的端口状态：

sw1#sh spanning-tree interface f0/23 portfast

no spanning tree info available for FastEthernet0/23

sw1#

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 disabled

sw1#

sw1#sh protocols f0/23

FastEthernet0/23 is down, line protocol is down

sw1#

sw1#sh protocols f0/24

FastEthernet0/24 is up, line protocol is up

sw1#sh int f0/23

FastEthernet0/23 is down, line protocol is down (err-disabled)

（输出被省略）

sw1#

说明：可以看见，SW1 的 portfast 端口 F0/23 收到 BPDU 后，受到 BPDU Guard 的影响，端口被 shutdown，并且变成 error-disabled，（某些型号不会），而全局 BPDU Guard 不能影响非 portfast 端口，所以 F0/24 还是正常状态。

2.在接口模式下配置 BPDU Guard（将对任何端口生效）

（1）将 SW2 的端口 F0/23 和 F0/24 改成三层接口

sw2(config)#int ran f0/23 - 24

sw2(config-if-range)#no switchport

说明：禁止从端口上向 SW1 发送 BPDU。

（2）将 SW1 的端口 F0/23 配置为 portfast，将 F0/24 配置为正常端口

sw1(config)#int f0/23

sw1(config-if)#switchport mode access

sw1(config-if)#spanning-tree portfast

sw1(config)#int f0/24

sw1(config-if)#switchport mode access

（3）在 SW1 的端口 F0/23 和 F0/24 开启 BPDU Guard

sw1(config)#int ran f0/23 - 24

sw1(config-if-range)#spanning-tree bpduguard enable

（4）查看 SW1 的端口 F0/23 和 F0/24 的状态

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 enabled

sw1#

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 disabled

sw1#

sw1#sh protocols f0/23

FastEthernet0/23 is up, line protocol is up

sw1#

sw1#sh protocols f0/24

FastEthernet0/24 is up, line protocol is up

sw1#

说明：SW1 上的端口 F0/23 为 portfast 状态，F0/24 为正常状态，并且状态都为 UP。

（5）在 SW2 的端口 F0/23 和 F0/24 向 SW1 发送 BPDU，测试 BPDU Guard

sw2(config)#int range f0/23 - 24

sw2(config-if-range)#switchport

（6）查看 SW1 的端口状态：

sw1#sh spanning-tree interface f0/23 portfast

no spanning tree info available for FastEthernet0/23

sw1#

sw1#sh spanning-tree interface f0/24 portfast

no spanning tree info available for FastEthernet0/24

sw1#

sw1#sh int f0/23

FastEthernet0/23 is down, line protocol is down (err-disabled)

（输出被省略）

sw1#sh int f0/24

FastEthernet0/24 is down, line protocol is down (err-disabled)

（输出被省略）

sw1#

说明：SW1 的端口在收到 BPDU 后，受到 BPDU Guard 的影响，无论是正常端口还是 portfast 端口，都被 err-disabled。

BPDU Filtering

BPDU Filtering 可以过滤掉在接口上发出或收到的 BPDU，这就相当于关闭了接口的 STP，将会有引起环路的可能。

BPDU Filtering 的配置同样也分两种，可以在接口下或在全局模式开启，但是不同的模式开启，会有不同效果。

如果 BPDU Filtering 是全局开启的，则只能在开启了 portfast 的接口上过滤 BPDU，并且只能过滤掉发出的 BPDU，并不能过滤收到的 BPDU，因为 BPDU Filtering 的设计目的是当交换机端口上连接的是主机或服务器时，就没有必要向对方发送 BPDU，所以要过滤掉 BPDU，但如果连接的是交换机，则会收到 BPDU，而且会引起环路，所以这样的情况，配置 BPDU Filtering 就是错误的。而当一个开启了 portfast 功能的接口，在开启了 BPDU Filtering 后，如果还能收到 BPDU，则 BPDU Filtering 特性会丢失，因此，还会造成接口 portfast 特性的丢失。

如果是在接口模式下开启的，则可以过滤掉任何接口收到和发出的 BPDU。(此理论为重点)

配置

1.在全局模式下配置 BPDU Filtering（只能过滤 portfast 上的 BPDU）

（1）将 SW2 的端口 F0/23 和 F0/24 改成三层接口

sw2(config)#int ran f0/23 - 24

sw2(config-if-range)#no switchport

说明：禁止从端口上向 SW1 发送 BPDU。

（2）将 SW1 的端口 F0/23 配置为 PortFast，将 F0/24 配置为正常端口，但开启 BPDU Guard

sw1(config)#int f0/23

sw1(config-if)#switchport mode access

sw1(config-if)#spanning-tree portfast

sw1(config)#int f0/24

sw1(config-if)#switchport mode access

sw1(config-if)#spanning-tree bpduguard enable

（3）查看 SW1 的端口 F0/23 和 F0/24 的状态

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 enabled

sw1#

sw1#sh int f0/24

FastEthernet0/24 is up, line protocol is up (connected)

（输出被省略）

sw1#

说明：SW1 的端口 F0/23 为 Port Fast 状态，F0/24 为正常状态，并且状态为 UP。

（4）在 SW1 上全局开启 BPDU Filtering（只能过滤 portfast 上的 BPDU）

sw1(config)#spanning-tree portfast bpdufilter default

（5）在 SW2 的端口 F0/23 和 F0/24 向 SW1 发送 BPDU，测试 BPDU Filtering

sw2(config)#int range f0/23 - 24

sw2(config-if-range)#switchport

说明：只要将 SW2 的端口 F0/23 和 F0/24 变成二层端口，便可以从此端口向外发送 BPDU。

（6）查看 SW1 的端口状态：

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 disabled

sw1#

sw1#sh int f0/24

FastEthernet0/24 is down, line protocol is down (err-disabled)

（输出被省略）

sw1#

说明：因为 F0/24 是普通端口，全局配置的 BPDU Filtering 不能过滤普通端口上的 BPDU，所以收到了 BPDU 后，但由于 BPDU Guard，最后端口被 err-disabled。

而 F0/23 是开启了 portfast 功能的接口，在开启了 BPDU Filtering 后，如果还能收到 BPDU，则 BPDU Filtering 特性会丢失，因此，造成了端口 F0/23 的 portfast 特性丢失。

2.在接口模式下配置 BPDU Filtering（将对任何端口生效）

（1）将 SW2 的端口 F0/23 和 F0/24 改成三层接口

sw2(config)#int ran f0/23 - 24

sw2(config-if-range)#no switchport

说明：禁止从端口上向 SW1 发送 BPDU。

（2）将 SW1 的端口 F0/23 配置为 portfast，将 F0/24 配置为正常端口

sw1(config)#int f0/23

sw1(config-if)#switchport mode access

sw1(config-if)#spanning-tree portfast

sw1(config)#int f0/24

sw1(config-if)#switchport mode access

（3）在 SW1 的端口 F0/24 开启 BPDU Guard

sw1(config)#int ran f0/23 - 24

sw1(config-if-range)#spanning-tree bpduguard enable

（4）查看 SW1 的端口 F0/23 和 F0/24 的状态

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 enabled

sw1#

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 disabled

sw1#

sw1#sh protocols f0/23

FastEthernet0/23 is up, line protocol is up

sw1#

sw1#sh protocols f0/24

FastEthernet0/24 is up, line protocol is up

sw1#

说明：SW1 上的端口 F0/23 为 portfast 状态，F0/24 为正常状态，并且状态都为 UP。

（5）在 SW1 的端口 F0/23 和 F0/24 下开启 BPDU Filtering

sw1(config)#int range f0/23 - 24

sw1(config-if-range)#spanning-tree bpdufilter enable

（6）在 SW2 的端口 F0/23 和 F0/24 向 SW1 发送 BPDU，测试 BPDU Filtering

sw2(config)#int range f0/23 - 24

sw2(config-if-range)#switchport

(7) 查看 SW1 的端口状态：

sw1#sh spanning-tree interface f0/23 portfast

VLAN0001 enabled

sw1#

sw1#sh spanning-tree interface f0/24 portfast

VLAN0001 disabled

sw1#

sw1#sh protocols f0/23

FastEthernet0/23 is up, line protocol is up

sw1#sh protocols f0/24

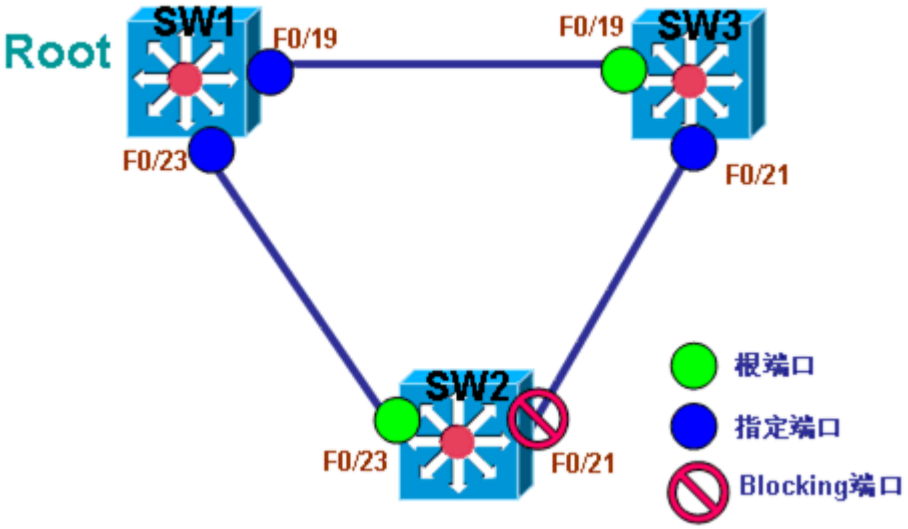
FastEthernet0/24 is up, line protocol is up

sw1#

说明：接口下开启的 BPDU Filtering，过滤掉了正常端口下的 BPDU，也过滤掉了 portfast 端口下的 BPDPU。

UplinkFast

以下图为例来解释 UplinkFast 的功能与作用：



在如上图一个运行 STP 的网络环境中，SW1 被选为根交换机，SW2 与 SW3 为普通交换机，其中 SW3 上的两个端口都为转发状态；SW2 上的端口 F0/23 为转发状态，而 F0/21 却为 Blocking 状态，因此无论 SW2 去往根交换机 SW1 还是去往 SW3，都只能从 F0/23 走。

当 SW2 的端口 F0/23 失效后，那么 SW2 去往 SW1 和 SW3 的唯一出口也就断掉了，如果 SW2 要启用 Blocking 的端口 F0/21，在 CST 下必须等待一个 max-age 时间（20 秒），再加两个 forward delay（共 30 秒），总共 50 秒的时间后，才能启用 Blocking 端口，即使是 RSTP，也有可能要等待 6 秒才能启用 Blocking 端口。

对于 SW2 来说，自己的端口 F0/23 断掉后，完全可以立刻检测出来，并且完全可以立刻启用 Blocking 的端口 F0/21，从而缩短网络的故障恢复时间。开启了 UplinkFast 交换机就能够在检测到交换机上直连的转发状态的接口失效后，立即启用 Blocking 的端口，提供网络快速收敛和恢复的功能。

很明显，UplinkFast 要真正起到作用，交换机上必须有 Blocking 的端口存在才行，因为根交换机上所有的接口最终都会变成指定端口，所以 UplinkFast 在根交换机上开启是毫无意义的，UplinkFast 只适合在非根交换机，即普通交换机上开启，普通交换机有时又称接入交换机。

★UplinkFast 只能在交换机上全局开启，不能针对 VLAN 单独开启，也不支持 MSTP 模式。

★UplinkFast 恢复网络的时间大约在 1 到 5 秒。

配置

1 在非根交换机上开启 UplinkFast

(1)在交换机 SW2 上开启 UplinkFast

Sw2(config)#spanning-tree uplinkfast

(2) 在交换机 SW3 上开启 UplinkFast

sw3(config)#spanning-tree uplinkfast

2 查看 UplinkFast

(1)查看交换机 SW2 的 UplinkFast

sw2#sh spanning-tree uplinkfast

UplinkFast is enabled

Station update rate set to 150 packets/sec.

UplinkFast statistics

Number of transitions via uplinkFast (all VLANs) : 3

Number of proxy multicast addresses transmitted (all VLANs) : 0

Name Interface List

---------- ----------

VLAN0001 Fa0/23(fwd), Fa0/21

sw2#

说明：SW2 在直连活动链路 F0/23 失效后，可将 F0/21 恢复。

(2)查看交换机 SW3 的 UplinkFast

sw3#sh spanning-tree uplinkfast

UplinkFast is enabled

Station update rate set to 150 packets/sec.

UplinkFast statistics

Number of transitions via uplinkFast (all VLANs) : 0

Number of proxy multicast addresses transmitted (all VLANs) : 0

Name Interface List

---------- ----------

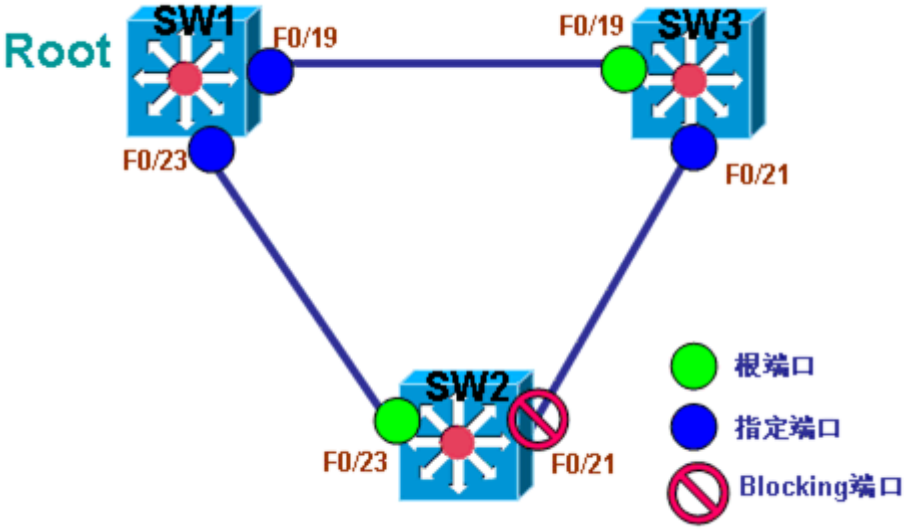
VLAN0001 Fa0/19(fwd)

sw3#

说明：SW3 在直连活动链路 F0/19 失效后，没有可恢复的端口。

BackboneFast

还是以下图为例来解释 BackboneFast 的功能与作用：



还是与 UplinkFast 同样的网络环境， 当 SW2 上直连的活动链路 F0/23 断掉之后，SW2 可以立刻检测出来，并且通过 UplinkFast 功能可以立即启用 Blocking 的端口 F0/21，提供网络快速收敛和恢复的功能。

交换机 SW3 无论去往根交换机 SW1 还是去往 SW2，都只能从 F0/19 走，如果 SW3 上的 F0/19 中断了，那么 SW3 去往 SW1 和 SW2 的唯一出口也就断掉了，此时的 SW3 不能与外界通信，正常情况下，需要等待 SW2 将自己的端口 F0/21 从 Blocking 状态过渡到转发状态后，才能为 SW3 提供一条新的通路。然而，即使是开启了 UplinkFast 功能的 SW2，也并不能在 SW3 的 F0/19 中断后立刻检测到，因为 UplinkFast 只能检测到自己直连的链路中断，并不能检测到远程链路中断。要在 SW3 的 F0/19 中断时，让其它所有交换机都检测到，从而为中断链路的交换机打开一条新的通路，需要在网络中所有交换机上开启 BackboneFast 功能。

因为正常网络中，除了根交换机，其它交换机不能发出 BPDU，所以 SW3 不能发送 BPDU，如果网络中出现了除根交换机发出的 configuration BPDU 之外的其它优先级更低的 BPDU，则称为 Inferior BPDU，当交换机收到 Inferior BPDU 时，默认是丢弃处理。

如果 SW3 上开启了 BackboneFast 功能，则当自己使用中的链路中断时，并且自己又没有 Blocking 状态的端口可以立即启用来代替活动链路，在这种情况下，就

可以以自己为根交换机，向网络中发出 Inferior BPDU，inferior BPDU 表示一台交换机即是根交换机，又是普通交换机，inferior BPDU 用来宣靠自己的链路中断，已经与网络失去联系。当其它非根交换机，当 SW2 在 Blocking 端口收到 Inferior BPDU 后，如果自己也开启了 BackboneFast 功能，就会将 Blocking 端口变成转发状态，还会向根交换机发出一个 root link query (RLQ)根链路查询，并且开启了 BackboneFast 功能的根交换机会做出回应，时此 SW2 就可以立刻启用自己 Blocking 状态的端口 F0/21，为 SW3 连接网络提供一条新的通路。

从上述可以看出，要网络中所有的交换机都能理解 Inferior BPDU，要全部配合工作，为链路中断的交换机开劈一条新通路，就必须在网络中所有交换机上都开启 BackboneFast 功能。

所以在开启 BackboneFast 功能时，需要在网络中所有交换机上开启，不能针对 VLAN 单独开启，也不支持 MSTP 模式。BackboneFast 是对 UplinkFast 的强化与补充。

配置

1.在非根交换机上开启 BackboneFast

(1)在交换机 SW1 上开启 BackboneFast

sw1(config)#spanning-tree backbonefast

(2) 在交换机 SW2 上开启 BackboneFast

sw2(config)#spanning-tree backbonefast

(3) 在交换机 SW3 上开启 BackboneFast

sw3(config)#spanning-tree backbonefast

2.查看 BackboneFast

(1)查看交换机 SW1 的 BackboneFast

sw1#sh spanning-tree backbonefast

BackboneFast is enabled

BackboneFast statistics

Number of transition via backboneFast (all VLANs) : 0

Number of inferior BPDUs received (all VLANs) : 0

Number of RLQ request PDUs received (all VLANs) : 0

Number of RLQ response PDUs received (all VLANs) : 0

Number of RLQ request PDUs sent (all VLANs) : 0

Number of RLQ response PDUs sent (all VLANs) : 0

sw1#

(2)查看交换机 SW2 的 BackboneFast

sw2#sh spanning-tree backbonefast

BackboneFast is enabled

BackboneFast statistics

Number of transition via backboneFast (all VLANs) : 0

Number of inferior BPDUs received (all VLANs) : 0

Number of RLQ request PDUs received (all VLANs) : 0

Number of RLQ response PDUs received (all VLANs) : 0

Number of RLQ request PDUs sent (all VLANs) : 0

Number of RLQ response PDUs sent (all VLANs) : 0

sw2#

(3)查看交换机 SW3 的 BackboneFast

sw3#sh spanning-tree backbonefast

BackboneFast is enabled

BackboneFast statistics

Number of transition via backboneFast (all VLANs) : 0

Number of inferior BPDUs received (all VLANs) : 0

Number of RLQ request PDUs received (all VLANs) : 0

Number of RLQ response PDUs received (all VLANs) : 0

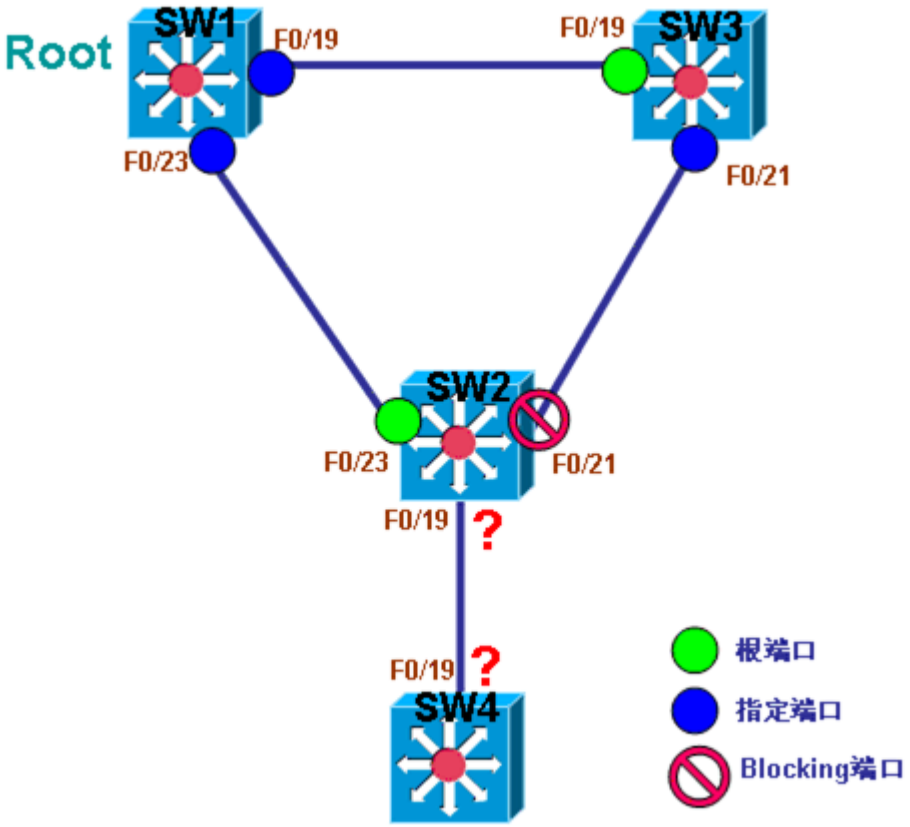
Number of RLQ request PDUs sent (all VLANs) : 0

Number of RLQ response PDUs sent (all VLANs) : 0

sw3#

Root Guard

以下图为例来解释 Root Guard 的功能与作用：



在上图中，交换机 SW1，SW2 与 SW3 为网络中运行正常的交换机，其中 SW1 被选为根交换机，当其它交换机之间要通信，都必须选出一个去往根交换机的端口，也就是根端口，所以当 SW2 与 SW3 承认 SW1 为网络中的根交换机时，SW2 便将连接 SW1 的端口 F0/23 选为根端口，SW3 将连 SW1 的端口 F0/19 选为根端口，此后网络通信正常。

考虑到网络的合理性与稳定性，将 SW1 选为根交换机是最佳选择，如果要将其它交换机选为根交换机或网络需要变动，都会引起不必要的麻烦。由于可以任意将一台交换机接入网络，而新接入的交换机，有很大的可能会因为自己拥有更高的

Bridge-ID 而抢夺当前根交换机的角色，这样就会引起网络的麻烦。上图中新加入的交换机 SW4，如果拥有比当前根交换机 SW1 更高的 Bridge-ID，就会抢夺根交换机的角色。如果 SW2 要承认 SW4 为根交换机，就必须将连接 SW4 的端口 F0/19 变成根端口，.如果 SW2 将端口 F0/19 中断或者阻塞，都将禁止 SW4 对网络的影响。所以只要控制好连接新加入交换机的端口角色，就能够阻止对方成为根交换机。

特性 Root Guard 正是利用上述原因，控制 SW2 用来连接新加入交换机的那个端口的角色，可以决定是否让其影响当前网络。开启了 Root Guard 功能的端口，如果在此端口上连接的新交换机试图成为根交换机，那么此端口并不会成为根端口，相反，此端口将进入 inconsistent (blocked) 状态，从而防止新加入交换机抢占根角色来影响网络。

注：

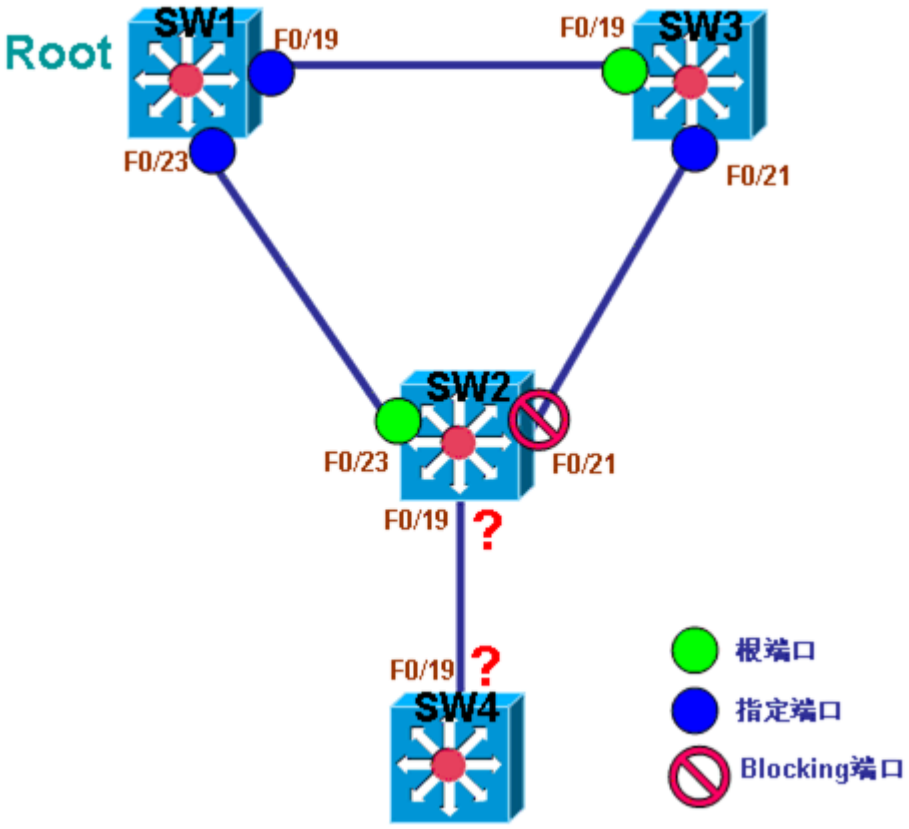
★运行 MSTP 时，开启了 Root guard 的端口强制成为指定端口。

★开启 Root guard 的端口在哪个 vlan，Root guard 就对哪些 vlan 生效。

★不能在需要被 UplinkFast, 使用的端口上开启 Root Guard。

★Root Guard 在可能连接新交换机的端口上开启。

配置



1.开启 Root guard

（1）在 SW2 上连接新交换机的端口 F0/19 上开启 Root guard

sw2(config-if)#spanning-tree guard root

2.查看 Root guard

（1）查看 SW2 上的 Rootguard

sw2#sh spanning-tree detail

（输出被省略）

Port 19 (FastEthernet0/19) of VLAN0001 is forwarding

Port path cost 19, Port priority 128, Port Identifier 128.19.

Designated root has priority 16385, address 007d.618d.0300

Designated bridge has priority 32769, address 0013.805c.4b00

Designated port id is 128.19, designated path cost 19

Timers: message age 0, forward delay 0, hold 0

Number of transitions to forwarding state: 1

Link type is point-to-point by default

Root guard is enabled on the port

BPDU: sent 204, received 0

（输出被省略）

sw2#

说明：可以看到，F0/19 已经开启 Root guard

3.测试 Rootguard

(1)配置 SW4 为根

sw4(config)#spanning-tree vlan 1 priority 4096

说明：给 SW4 配置一个更高优先级的 Bridge-ID，以此来抢夺根交换机的角色。

（2）查看 SW2 的状态

当开启了 Root guard 的端口对方如果要成为根交换机，则会有如下提示，并且接口被放入 inconsistent (blocked) 状态：

sw2#

01:18:56: %SPANTREE-2-ROOTGUARD_BLOCK: Root guard blocking port FastEthernet0/19 on VLAN0001.

sw2#

（3）查看被放入 inconsistent (blocked) 状态的端口：

sw2#sh spanning-tree inconsistentports

Name	Interface	Inconsistency
VLAN0001	FastEthernet0/19	Root Inconsistent

Number of inconsistent ports (segments) in the system : 1

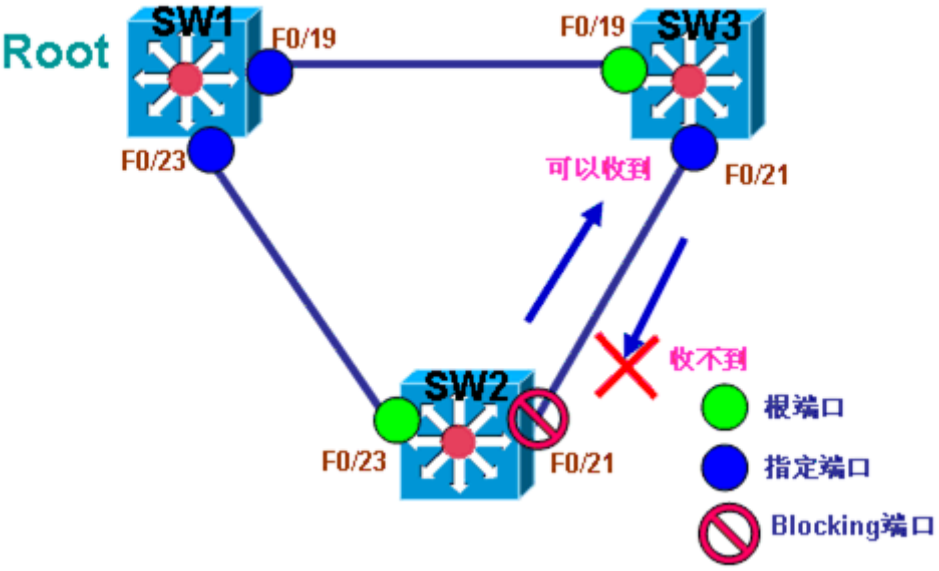
sw2#

说明：由于端口 F0/19 开启了 Root guard，而对端要成为根交换机，所以此端口被放入 inconsistent (blocked) 状态的端口。

Loop Guard

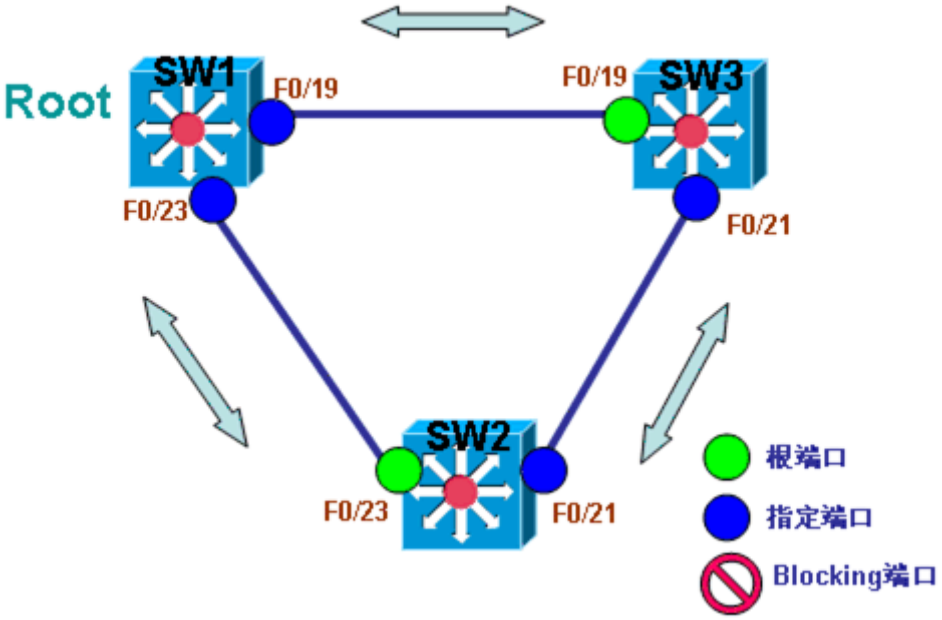
在交换网络中，当两点之间存在多条冗余链路时，就会因为重复的数据包在网络中传递，引起广播风暴，并且还会造成交换机 MAC 地址表错误，使网络不稳定，因此造成环路。所以需要借助 STP 来阻塞网络中两点之间多余的链路，而只留一条活动链路，即为转发状态，其它多余链路变为 Blocking 状态，当转发状态的链路中断时，再启用 Blocking 状态的端口。

当 STP 运行时，只有两点之间存在多条冗余链路时，才会阻塞多余链路而只留一条活动链路。如果 STP 认为两点之间并没有多条链路，也就不会产生环路，那么就不会有端口被 Blocking。因为 STP 在判断两点之间是否有多条链路，是靠发送 BPDU，如果从多个端口收到同一台交换机的 BPDU，则认为与那个点之间有多条链路，所以会阻塞多余链路而只留一条。如果只从一个端口收到同台交换机的 BPDU，或者是没有收到重复 BPDU，则认为网络是无环的，也就没有端口被 Blocking，其它不需要被 Blocking 的端口，都会被变为指定端口。



在上图中的网络环境中，如果交换机所有端口收发数据的功能正常，则交换机就能够靠收发 BPDU 来检测出网络中的多余链路，就会将其 Block，从而避免环路。但是当网络中出现单向链路故障时，也就是某个端口只能收数据而不能发数据，或

者只能发数据而不能收数据，这时网络会出现意想不到的麻烦。如上图，由于 SW2 的端口 F0/21 出现单向链路故障，导致从 F0/21 发出去的数据包能被 SW3 收到，而 SW3 发的数据包却不能被 SW2 收到，此时造成的结果是，SW3 认为网络是正常的，又由于 SW3 拥有更高优先级的 Bridge-ID，所以 SW3 上 F0/19 为根端口，F0/21 为指定端口，SW3 上所有端口都是转发状态，而没有 Blocking 的端口。但是由于 SW2 不能收到 SW3 发来的数据包，也就不能从 SW3 收到 BPDU，最终 SW2 只能从 F0/23 收到数据包，所以 SW2 认为网络是无环的，因此做出了一个错误的决定，就是在 STP 计算结束后，认为网络无环，而将原本应该被 Block 的端口 F0/21 变为指定端口，造成 SW 上 F0/23 和 F0/21 同时为转发状态。不难看出，此时，网络中所有的交换机端口都为转发状态，结果如下：



最终造成网络中所有的交换机端口都为转发状态，流量在所有端口上被转发，引起广播风暴，出现环路。此结果是非常严重的。

单向链路故障不仅会使 Blocking 状态的端口错误地变成指定端口，还会造成根端口错误地变成指定端口。

对于上述问题，可以通过 Loop Guard 来解决，开启了 Loop Guard 的端口在收

不到 BPDU 的情况下，并不会认为网络是无环的，并不会错误地将端口变成指定端口，而是将收不到 BPDU 的端口变成 loop-inconsistent 状态，此状态等同于 blocking 状态。

Loop Guard 可以全局开启，也可以在接口下开启，但不建议在全局开启，请在相应接口下开启。什么端口最需要开，很明显，当然是被 blocking 的端口，但并不完全正确，准确答案是在所有非指定端口开启，其实就是根端口和 blocking 端口。

当在接口开启后 Loop Guard，接口所在的所有 VLAN 都会生效，如果是接口是 trunk，哪个 VLAN 没有收到 BPDU，接口就会在哪个 VLAN 被 blocking。在 EtherChannel 上是对整条生效。

只有交换机上的 blocking 端口和根端口才需要开启 Loop Guard。如果一个网络中所有交换机没有 blocking 的端口，就表示此网络无环，所以就不需要开 Loop Guard。并且根交换机上所有端口都是指定端口，所以在根上开 Loop Guard 是没有意义的。

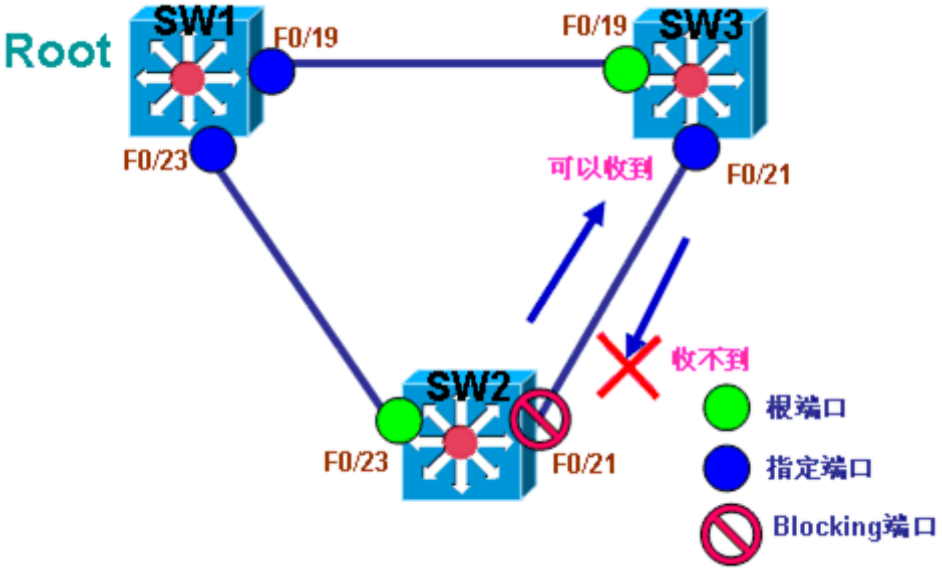
注：

★PortFast 的接口不能开启 Loop Guard。

★Root guard 和 Loop Guard 不能同时开。

★Root guard 支持 PVST+,，rapid PVST+， MSTP。

配置



1.开启 LoopGuard

（1）在 SW2 的根端口与 Blocking 端口开启 Loop Guard

sw2(config)#int f0/21

sw2(config-if)#spanning-tree guard loop

sw2(config)#int f0/23

sw2(config-if)#spanning-tree guard loop

说明：在所有根端口与 Blocking 端口开启 Loop Guard

2.查看 LoopGuard

(1)查看开启了 Loop Guard 的端口

sw2#sh spanning-tree detail

（输出被省略）

Port 21 (FastEthernet0/21) of VLAN0001 is blocking

Port path cost 19, Port priority 128, Port Identifier 128.21.

Designated root has priority 16385, address 007d.618d.0300

Designated bridge has priority 24577, address 0013.8065.bd80

Designated port id is 128.21, designated path cost 19

Timers: message age 3, forward delay 0, hold 0

Number of transitions to forwarding state: 0

Link type is point-to-point by default

Loop guard is enabled on the port

BPDU: sent 3, received 3917

（输出被省略）

sw2#

说明：可以看到，F0/21 已经开启 Loop Guard

3.测试 Loop Guard

(1)过滤掉 SW3 从 F0/21 发往 SW2 的 BPDU

sw3(config)#int f0/21

sw3(config-if)#spanning-tree bpdufilter enable

说明：让 SW2 开启了 Loop Guard 的端口 F0/21 收不到 BPDU。

（2）查看 SW2 的 Loop Guard 状态

当 SW2 开启了 Loop Guard 的端口收不到 BPDU 时，会有如下提示：

sw2#

02:16:28: %SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking port FastEthernet0/21 on VLAN0001.

sw2#

（3）查看被放入 inconsistent (blocked) 状态的端口：

sw2#sh spanning-tree inconsistentports

Name Interface Inconsistency

---------- ---------- ----------

VLAN0001 FastEthernet0/21 Loop Inconsistent

Number of inconsistent ports (segments) in the system : 1

sw2#

说明：由于端口 F0/21 开启了 Loop Guard，所有在收不到 BPDU 时，此端口被放入 inconsistent (blocked) 状态的端口。

EtherChannel

当在两台交换机之间连接多条线路来增加带宽时，由于 STP 的原因，最终会阻断其它多余的线路而只留下一条活动链路来转发数据，因此，在两台交换机之间连接多条线路，并不能起到增加带宽的作用。为了能够让两台交换机之间连接的多条线路同时提供数据转发以达到增加带宽的效果，可以通过 EtherChannel 来实现。

EtherChannel 将交换机上的多条线路捆绑成一个组，相当于逻辑链路，组中活动的物理链路同时提供数据转发，可以提高链路带宽。当组中有物理链路断掉后，那么流量将被转移到剩下的活动链路中去，只要组中还有活动链路，用户的流量就不会中断。

EtherChannel 只支持对 Fast Ethernet 接口或 Gigabit Ethernet 接口的捆绑，对于 10M 的接口还不支持。一个 EtherChannel 组中，最多只能有 8 个接口为用户转发数据。

在两台交换机之间连接多条链路时，如果只有一边交换机做了 EtherChannel 捆绑，而另一边不做捆绑，那么接口会工作在异常状态，而不能正常转发流量。所以，必须同时在两边交换机都做 EtherChanne 捆绑。

为了让两边交换机的接口都工作在 EtherChannel 组中，可以通过手工强制指定接口工作在组中，也可以通过协议自动协商。如果是手工强制指定，则不需要协议，自动协议的协议有以下两种：

Port Aggregation Protocol (PAgP)

Link Aggregation Control Protocol (LACP)

无论是手工指定，还是通过协议协商，交换机双方都必须采取相同的方式和协议，否则将导致接口异常。

EtherChannel 自动协商协议 PAgP 为思科专有，只有在双方交换机都为思科交换机时，才可以使用，而 LACP 为 IEEE 协议，任何交换机，只要支持 EtherChannel 的都可以使用该协议。

当将接口使用 PAgP 作为协商协议时，有以下两种模式可供选择：

Auto

只接收 PAgP 协商消息，并做出回应同意工作在 EtherChannel 下，并不主动发

出 PAgP 协商，属于被动状态。

Desirable

主动发送 PAGP 协商消息，主动要求对方工作在 EtherChannel 下，属于主动模式。

如果两边交换机都是 Desirable 模式，则可以协商成功，如果两边都是 Auto 模式，则不能工作在 EtherChannel。

当将接口使用 LACP 作为协商协议时，有以下两种模式可供选择：

Passive

只接收 LACP 协商消息，并做出回应同意工作在 EtherChannel 下，并不主动发出 LACP 协商，属于被动状态。

Active

主动发送 LACP 协商消息，主动要求对方工作在 EtherChannel 下，属于主动模式。

如果两边交换机都是 Active 模式，则可以协商成功，如果两边都是 Passive 模式，则不能工作在 EtherChannel。

在配置 EtherChannel 时，除了在接口上配置以上两种协议来自动协商外，还可以强制让接口工作在 EtherChannel 而不需要协商，配置为 ON 模式即可，如果配置 ON，则两边都必须配置为 ON，否则不能转发数据。

下表为配置 EtherChannel 的模式总结：

模式	协议	描述
ON	无	手工静态强制接口工作在 EtherChanne 下。
Auto	PAGP	只接收 PAgP 协商消息，并做出回应同意工作在 EtherChanne 下，并不主动发出 PAgP 协商。
Desirable	PAGP	主动发送 PAGP 协商消息，主动要求对方工作在 EtherChanne 下。
Passive	LACP	只接收 LACP 协商消息，并做出回应同意工作在 EtherChanne 下，并不主动发出 LACP 协商。
Active	LACP	主动发送 LACP 协商消息，主动要求对方工作在 EtherChanne 下。

当配置 PAGP 时，可以使用关键字 non-silent，如果不指定 non-silent，默认为 silent。

Silent 表示即使不能从对端设备收到 PAGP 协商数据，也使物理接口工作在 EtherChannel 组中，思科建议接口连接服务器或分析仪时使用。
non-silent 表示只有在和对方协商成功之后，才使物理接口工作在 EtherChannel 组中。也就是说只有双方都支持 PAGP 的情况下，才才使物理接口工作在 EtherChannel 组中。

因为三层交换机的接口即可以工作在二层模式，也可以工作在三层模式，所以 EtherChannel 捆绑后的逻辑接口也有二层和三层之分。

当将接口 EtherChannel 捆绑后，会自动生成逻辑接口，称为 port-channel 接口，port-channel 接口与 EtherChannel 组的号码相同，但范围是 1-48。当使用二层接口时，在物理接口下配置参数后，port-channel 接口将读取物理接口下的参数，但必须组成的所有接口都做相同的配置；在 port-channel 接口下做的配置也会自动在物理接口下生效。当使用三层接口时，必须先将物理接口变成三层接口后，再做捆绑，因为 port-channel 接口是不能在二层与三层之间转换的，配置三层接口，应该到 port-channel 接口下做的配置，而不应该直接配置物理接口。

如果是使用 2 层 EtherChannel，那么组中第一个正常工作的口接口的 MAC 地址就是 port-channel 接口接口的 MAC 地址。

注：

★在配置 EtherChannel 组时，需要定义组号码，但不要配置超过 48 个组。

★两边交换机的 EtherChannel 组号码可以采用不同号码。

★PAGP 组中不能配超过 8 个接口。

★LACP 中不能超过 16 个接口，但只有 8 个活动接口。

★两个协议可以配置在同台交换机上，但不能配置在同一个组中。

★组中的接口不能是 SPAN 的目标接口和安全接口以及 802.1x 端口。

★将接口配置为 2 层时，全部必须在相同 VLAN，如果是 trunk，native vlan 必须相同。

★配好 EtherChannel 组后后，在 port-channel 下配的参数会对所有物理接口生效，但对单个物理接口配置的只对单物理接口生效。

★多个接口捆绑成单条 EtherChannel 后，在 STP 中，被当作单条链路来计算，同时 Path Cost 值会和原物理链路有所不同。

EtherChannel Load Balancing

当将多个接口捆绑成 EtherChannel 组之后，流量将同时从多个接口上被发出去，称为 Load Balancing,即负载均衡，对于流量以什么样的负载均衡方式从 EtherChannel 组中的多个接口上发出去，可以有以下几种方式：

Souce-MAC

基于源 MAC，默认为此模式，不同源主机，流量可能从不同的接口被发出去，但相同源主机肯定走相同接口。

Source-and-Destination MAC

同时基于源和目标 MAC，流量从主机 A 到主机 B，从主机 A 到主机 C 以及从主机 C 到主机 B 都可能走不同的接口。

Source-IP

基于源 IP，不能源 IP 的流量可能走不同接口，相同 IP 则走相同接口。

Destination-IP

基于目的 IP，到不同目标 IP 的流量，会走不同接口，不同主机发往相同 IP 的流量会走相同接口。

Source-and-Destination IP

同时基于源和目标 IP，流量从主机 A 到主机 B，从主机 A 到主机 C 以及从主机 C 到主机 B 都可能走不同的接口。

注：并不是所有型号的交换机所有 IOS 都支持所有负载方式，需要视 IOS 版本而定。

在交换机之间通过 EtherChannel 捆绑了多条链路后，默认执行基于源 MAC 的负载均衡，而每条链路的流量比例却是固定的，也就是说，你只能改变 EtherChannel 负载均衡方式，但却改不了每条物理链路上的流量比例，接口上的流量比例，执行以下标准：

Number of Ports in the EtherChannel	Load Balancing
8	1:1:1:1:1:1:1:1
7	2:1:1:1:1:1:1
6	2:2:1:1:1:1
5	2:2:2:1:1
4	2:2:2:2
3	3:3:2
2	4:4

配置

1.配置 2 层 EtherChannel

(1)配置 SW1

```
sw1(config)#int range f0/23 - 24

sw1(config-if-range)#channel-group 12 mode desirable
```

说明：在接口 F0/23-24 下选用 PAGP 配置 EtherChannel

（2）配置 SW2

```
sw2(config)#int range f0/23-24
```

sw2(config-if-range)#channel-group 12 mode desirable

说明：在接口 F0/23-24 下选用 PAGP 配置 EtherChannel

（3）查看 EtherChannel

sw1#show etherchannel summary

Flags: D - down P - in port-channel

I - stand-alone s - suspended

H - Hot-standby (LACP only)

R - Layer3 S - Layer2

U - in use f - failed to allocate aggregator

u - unsuitable for bundling

w - waiting to be aggregated

d - default port

Number of channel-groups in use: 1

Number of aggregators: 1

Group Port-channel Protocol Ports

------+-------------+-----------+-----------------------------------
----

12 Po12(SU) PAgP Fa0/23(P) Fa0/24(P)

sw1#

说明：可以看到，已捆绑的接口为 2 层接口，并且所有物理接口都工作在 EtherChannel 下。

（4）在 port-channel 接口下配置接口

sw1(config)#int port-channel 12

sw1(config-if)#switchport mode access

sw1(config-if)#switchport access vlan 10

说明：port-channel 接口下将接口划入 VLAN。

（5）查看 port-channel 接口 MAC 地址

F0/23：

sw1#sh int f0/23

FastEthernet0/23 is up, line protocol is up (connected)

Hardware is Fast Ethernet, address is 007d.618d.0317 (bia 007d.618d.0317)

（输出被省略）

sw1#

F0/24

sw1#sh int f0/24

FastEthernet0/24 is up, line protocol is up (connected)

Hardware is Fast Ethernet, address is 007d.618d.0318 (bia 007d.618d.0318)

（输出被省略）

sw1#

port-channel：

sw1#sh int port-channel 12

Port-channel12 is up, line protocol is up (connected)

Hardware is EtherChannel, address is 007d.618d.0318 (bia 007d.618d.0318)

（输出被省略）

sw1#

说明：port-channel 使用了接口 F0/24 下的 MAC 地址，说明接口 F0/24 先工作正常。

2.配置 3 层 EtherChannel

(1)配置 SW1

sw1(config)#int range f0/23 - 24

sw1(config-if-range)#no switchport

sw1(config-if-range)#channel-group 12 mode active

sw1(config)#int port-channel 12

sw1(config-if)#ip address 10.1.1.1 255.255.255.0

说明：配置 3 层 EtherChannel，需要先将物理接口变成 3 层接口后，才能正常配置，IP 地址必须在 port-channel 下配置。

（2）配置 SW2

```
sw2(config)#int range f0/23 - 24

sw2(config-if-range)#no switchport

sw2(config-if-range)#channel-group 12 mo active


sw2(config)#int port-channel 12

sw2(config-if)#ip address 10.1.1.2 255.255.255.0
```

说明：配置 3 层 EtherChannel，需要先将物理接口变成 3 层接口后，才能正常配置，IP 地址必须在 port-channel 下配置。

（3）查看 EtherChannel

```
sw1#sh eth summary

Flags:  D - down        P - in port-channel

        I - stand-alone s - suspended

        H - Hot-standby (LACP only)

        R - Layer3      S - Layer2

        U - in use      f - failed to allocate aggregator

        u - unsuitable for bundling

        w - waiting to be aggregated
```

```
            d - default port



    Number of channel-groups in use: 1

    Number of aggregators:           1


    Group  Port-channel  Protocol    Ports

    ------+-------------+-----------+-------------------------------
------------

    12     Po12(RU)         LACP       Fa0/23(P)    Fa0/24(P)


    sw1#
```

说明：可以看到，已捆绑的接口为 3 层接口，并且所有物理接口都工作在 EtherChannel 下。

（4）测试 port-channel 连通性

```
    sw1#ping 10.1.1.2


    Type escape sequence to abort.

    Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

    !!!!!

    Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

sw1#

说明：port-channel 正常工作在 3 层。

3.配置 Load Balancing

（1）配置基于目标 MAC 的负载均衡

sw1(config)#port-channel load-balance dst-mac

说明：开启了基于目标 MAC 的负载均衡，默认为基于源 MAC，其它负载方式，可自行配置。

（2）查看 EtherChannel Load Balancing

sw1#sh etherchannel load-balance

EtherChannel Load-Balancing Configuration:

dst-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:

Non-IP: Destination MAC address

IPv4: Destination MAC address

sw1#

说明：可以看到，EtherChannel 已经基于 MAC 的负载均衡。

附：当配置 PAGP 时，可以选择配置 non-silent，默认为 silent，配置如下：

```
sw1(config)#int range f0/23 - 24

sw1(config-if-range)#channel-group 12 mode desirable non-silent
```

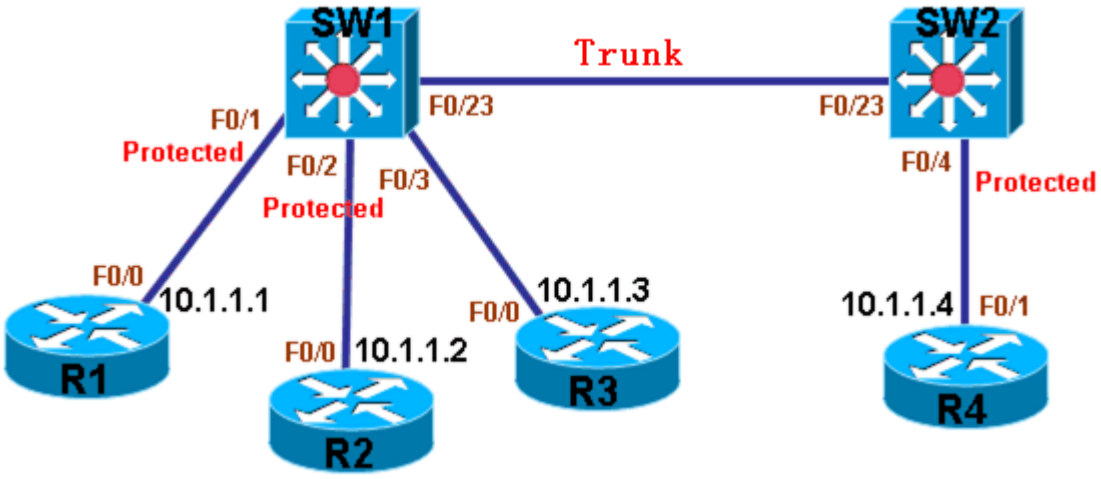
Protected Port

在某些特殊需求下，需要禁止同台交换机上相同 VLAN 的主机之间通信，但又不能将这些不能通信的主机划到不同 VLAN，因为还需要和 VLAN 中的其它主机通信，只是不能和部分主机通信。要限制交换机上相同 VLAN 的主机通信，通过将交换机上的接口配置成 Protected Port 来实现，如果交换机上某个 VLAN 有三个接口，其中有两个是 Protected Port，有一个是正常端口，那么两个 Protected Port 之间是不能通信的，但是 Protected Port 与正常端口之间的流量还是保持正常，而不受任何限制。

Protected Port 可以拒绝 unicast，broadcast 以及 multicast 在这些端口之间通信，也就是说 Protected Port 与 Protected Port 之间没有任何流量发送。Protected Port 只在单台交换机上有效，也就是说只有单台交换机上的 Protected Port 与 Protected Port 之间是不能通信的，但是不同交换机的 Protected Port 与 Protected Port 之间通信还是保持正常。

配置 Protected Port 时，可以在物理接口和 EtherChannel 上配置，如果是配在 EtherChannel 上，那么配置将对 EtherChannel 中的所有物理接口生效。

配置



说明：以上图为例，配置 protected port，SW1 的 F0/1，F0/2，F0/3 以及 SW2 的 F0/4 都在 VLAN 10 中。

1.配置交换机

（1）配置 SW1

```
sw1(config)#vlan 10

sw1(config-vlan)#exit


sw1(config)#int range f0/1 - 3

sw1(config-if-range)#switchport mode access

sw1(config-if-range)#switchport access vlan 10


sw1(config)#int f0/23

sw1(config-if)#switchport trunk encapsulation dot1q

sw1(config-if)#switchport mode trunk
```

（2）配置 SW2

```
sw2(config)#vlan 10
sw2(config-vlan)#exit


sw2(config)#int f0/4
sw2(config-if)#switchport mode access
sw2(config-if)#switchport access vlan 10


sw2(config)#int f0/23
sw2(config-if)#switchport trunk encapsulation dot1q
sw2(config-if)#switchport mode trunk
```

2.配置路由器

（1）配置 R1

```
r1(config)#int f0/0
r1(config-if)#ip add 10.1.1.1 255.255.255.0
```

（2）配置 R2

```
r2(config)#int f0/0
r2(config-if)#ip add 10.1.1.2 255.255.255.0
```

（3）配置 R3

```
r3(config)#int f0/0
r3(config-if)#ip add 10.1.1.3 255.255.255.0
```

（4）配置 R4

```
r4(config)#int f0/1

r4(config-if)#ip add 10.1.1.4 255.255.255.0
```

3.测试正常情况下的通信

（1）测试 R1 到 R2 的连通性

```
r1#ping 10.1.1.2


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#
```

说明：因为没有配置 protected port，所以 R1 到 R2 通信正常。

（2）测试 R1 到 R3 的连通性

```
r1#ping 10.1.1.3


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#
```

说明：因为没有配置 protected port，所以 R1 到 R3 通信正常。

（3）测试 R1 到 R4 的连通性

```
r1#ping 10.1.1.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
r1#
```

说明：因为没有配置 protected port，所以 R1 到 R4 通信正常。

3.配置 protected port

（1）在 SW1 上将 F0/1 和 F0/2 配置为 protected port

```
sw1(config)#int f0/1
sw1(config-if)#switchport protected

sw1(config)#int f0/2
sw1(config-if)#switchport protected
```

（2）在 SW2 上将 F0/4 配置为 protected port

```
sw2(config)#int f0/4
sw2(config-if)#switchport protected
```

4.测试配置了 protected port 的网络通信

（1）测试 R1 到同台交换机的正常端口 F0/3 的连通性

```
r1#ping 10.1.1.3
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

r1#
```

说明：因为 protected port 与正常端口之间的通信不受影响，所以 R1 到 R3 通信正常。

（2）测试 R1 到同台交换机的 protected port F0/2 的连通性

```
r1#ping 10.1.1.2


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r1#
```

说明：因为同台交换机上 protected port 与 protected port 之间的流量被拒绝，所以 R1 到 R2 通信失败。

（3）测试 R1 到远程交换机 SW2 的 protected port F0/4 的连通性

```
r1#ping 10.1.1.4


Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#
```

说明：因为只有单台交换机上的 Protected Port 与 Protected Port 之间是不能通信的，但是不同交换机的 Protected Port 与 Protected Port 之间通信还是保持正常，所以 R1 到 R4 的通信正常。

（4）在交换机上查看 Protected Port

```
sw1#sh int f0/1 switchport

（输出被省略）

Protected: true

Unknown unicast blocked: disabled

Unknown multicast blocked: disabled

Appliance trust: none

sw1#
```

说明：可以看到交换机上接口的 Protected Port 功能已经开启。

Port Blocking

默认情况下，交换机收到未知目标 MAC 的流量，也就是目标 MAC 地址不在 MAC 地址表中的流量，会将此流量在所有接口上泛洪。用户可以选择在交换机接口上拒绝泛洪未知目标 MAC 的流量，配置可以对 unicast 和 multicast 生效，但不能限制广播流量。

接口上默认是没有 Port Blocking 配置的。

配置 Port Blocking 时，可以在物理接口和 EtherChannel 上配置，如果是配在 EtherChannel 上，那么配置将对 EtherChannel 中的所有物理接口生效。

配置

1.在接口上配置 Port Blocking

（1）配置 Port Blocking 限制 unicast

```
sw1(config)#int f0/1
sw1(config-if)#switchport block unicast
```

（2）配置 Port Blocking 限制 multicast

```
sw1(config)#int f0/1
sw1(config-if)#switchport block multicast
```

(3)查看 Port Blocking

```
sw1#sh interfaces f0/1 switchport
（输出被省略）
Unknown unicast blocked: enabled
Unknown multicast blocked: enabled
Appliance trust: none
sw1#
```

说明：可以看到，交换机接口上已经开启拒绝泛洪未知目标 MAC 的单播流量和组播流量，并且两个可以同时开启。

Port Security

交换机在转发数据包时，需要根据数据包的目标 MAC 地址来决定出口，因此，交换机会将 MAC 地址与相对应的接口记录在一张表中，以供转发数据包使用，这张表就是 MAC 地址表。在正常情况下，MAC 地址表允许一个接口可以与多个 MAC 地址相对应，只要接口上有相应的 MAC 地址，那么数据包就可以从这个接口发出去。一个接口上对应着什么样的 MAC 地址，一个接口允许多少个 MAC 地址与之相对应，这都影响到交换机对数据的转发。为了让用户对交换机的 MAC 地址表有更高的控制权限，交换机接口上的 Port Security 功能提供更多的安全保护。

Port Security 可以控制交换机上特定的接口与特定的 MAC 地址的对应关系，也可以限制接口上最大的 MAC 地址数量。

具有 Port Security 功能的接口，被称为 secure port，secure port 接口上通过控制数据包的源 MAC 地址来控制流量，绝不会转发预先定义好的 MAC 地址之外的流量。准确地说，是 secure port 只转发合法的流量，对于违规的流量，是不放行的。区别是否违则，有以下两种情况：

1．当接口上 MAC 地址数量达到最大允许数量后，还有更多的 MAC 要访问，就算违规。

2．一个 secure port 接口上的合法 MAC 在另外一个 secure port 接口上访问，也算违规。

被 Port Security 允许的 MAC 地址，就是合法的 MAC 地址，称为安全 MAC 地址（Secure MAC Addresses），secure port 接口只放行源 MAC 为安全 MAC 地址的数据包。

要在 secure port 接口上定义安全 MAC 地址，有以下几种方法：

静态手工配置

手工添加MAC 地址与接口的对应关系，会保存在地址表和running configuration 中。

动态学习

将接口上动态学习到的 MAC 地址作为安全 MAC 地址，但此 MAC 地址只保存在 MAC 地址表中，交换机重启后将丢失。

Sticky secure MAC addresses

为了结合静态手工配置与动态学习 MAC 地址的优势，Sticky 将动态学习到的 MAC 地址作为安全 MAC 地址，并且将结果保存到 running configuration 中。

一个 secure port 接口上可以允许的 MAC 地址数量是系统可支持的最大 MAC 地址数量。对于违规的流量，可以采取以下四个可执行的动作：

Protect

只丢弃不允许 MAC 地址的流量，其它合法流量正常，但不会通知有流量违规了。

Restric

只丢弃不允许 MAC 地址的流量，其它合法流量正常，但会有通知，发送 SNMP trap，并会记录 syslog。

Shutdown

（默认模式） 将接口变成 error-disabled 并 shut down，并且接口 LED 灯会关

闭，也会发 SNMP trap，并会记录 syslog。

shutdown vlan

相应 VLAN 变成 error-disabled ，但接口不会关，也会发 SNMP trap，并会记录 syslog。

注：

★当一个 secure port 接口上的 MAC 地址在另外一个 secure port 接口出现后，就算违规，而违规动作是对出现重复地址的接口实施的，是为了防止攻击。

★当接口被 error-disabled 后，要恢复，请在接口上使用命令：shutdown 后 no shutdown。

以下是来自思科官方的模式与结果对应表：

Security Violation Mode Actions

Violation Mode	Traffic is forwarded[1]	Sends SNMP trap	Sends syslog message	Displays error message[2]	Violation counter increments	Shuts down port
protect	No	No	No	No	No	No
restrict	No	Yes	Yes	No	Yes	No
shutdown	No	Yes	Yes	No	Yes	Yes
shutdown vlan	No	Yes	Yes	No	Yes	No[3]

注：

★默认接口上 Port Security 是关闭的，Port Security 默认只允许 1 个安全 MAC 地址。

★只能在静态 access 接口和静态 trunk 接口上配 Port Security，不能在 dynamic 接口上配。

★Port Security 接口不能是 SPAN 的目标接口，不能在 EtherChannel 中。

Port Security Aging Time （Port Security MAC 地址老化时间）

在正常接口下动态学习到的 MAC 地址，在老化时间到了之后，交换机会将它从 MAC 地址表中删除。

而对于 Port Security 接口下的 MAC 地址，如果是通过安全命令静态手工添加的，则不受 MAC 地址老化时间的限制，也就是说通过安全命令静态手工添加的 MAC 在 MAC 地址表中永远不会消失。而即使在 Port Security 接口下动态学习到的 MAC 地址，也永远不会消失。

基于上述原因，有时限制了 Port Security 接口下的最大 MAC 地址数量后，当相应的地址没有活动了，为了腾出空间给其它需要通信的主机使用，则需要让 Port Security 接口下的 MAC 地址具有老化时间，也就是说需要交换机自动将安全 MAC 地址删除。

对于在 Port Security 接口下设置 MAC 地址的老化时间，分两种类型：absolute 和 inactivity，其中 absolute 表示绝对时间，即无论该 MAC 地址是否在通信，超过老化时间后，立即从表中删除；inactivity 为非活动时间，即该 MAC 地址在没有流量的情况下，超过一定时间后，才会从表中删除。

配置 MAC 地址老化时间的单位是分钟，范围是 0-1440，对于 sticky 得到的 MAC 地址，不受老化时间限制，并且不能更改。

配置

1.查看当前路由器的 MAC 地址

（1）查看 R1 的接口 F0/0 的 MAC 地址

r1#sh int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is AmdFE, address is 0013.1a85.d160 (bia 0013.1a85.d160)

Internet address is 10.1.1.1/24

（输出被省略）

r1#

说明：R1 的接口 F0/0 的 MAC 地址为 0013.1a85.d160

（2）查看 R2 的接口 F0/0 的 MAC 地址

r2#sh int f0/0

FastEthernet0/0 is up, line protocol is up

Hardware is AmdFE, address is 0013.1a2f.1200 (bia 0013.1a2f.1200)

（输出被省略）

R2：

说明：R2 的接口 F0/0 的 MAC 地址为 0013.1a2f.1200

2.配置交换机的 port-security

（1）配置 F0/1

```
sw1(config)#int f0/1
sw1(config-if)#switchport mode access
sw1(config-if)#switchport port-security
sw1(config-if)#switchport port-security maximum 1
sw1(config-if)#switchport port-security mac-address 0013.1a85.d160
sw1(config-if)#switchport port-security violation shutdown
```

说明：将接口静态配置成 access 后，再开启 port-security，允许最大地址数量为 1，默认也是为 1，定义的最大地址数量值不能比已学到的 MAC 地址少，否则无效。手工静态指定的安全 MAC 地址为 0013.1a85.d160，在违规后采取动作 shutdown。

（2）查看 F0/1 的配置

```
sw1#sh run int f0/1
Building configuration...

Current configuration : 136 bytes
!
interface FastEthernet0/1
 switchport mode access
```

switchport port-security

switchport port-security mac-address 0013.1a85.d160

end

sw1#

说明：因为默认允许的最大地址数量为 1，所有不显示出来。

（3）配置 F0/2

sw1(config)#int f0/2

sw1(config-if)#switchport mode access

sw1(config-if)#switchport port-security

sw1(config-if)#switchport port-security maximum 2

sw1(config-if)#switchport port-security mac-address sticky

sw1(config-if)#switchport port-security violation shutdown

说明：将接口静态配置成 access 后，再开启 port-security，允许最大地址数量为 2，指定安全 MAC 地址的方式为 sticky，在违规后采取动作 shutdown。

（4）查看 F0/2 的配置

sw1#sh run int f0/2

Building configuration...

Current configuration : 224 bytes

```
!
interface FastEthernet0/2
 switchport mode access
 switchport port-security maximum 2
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0013.1a2f.1200
end

sw1#
```

说明：因为指定安全 MAC 地址的方式为 sticky，所以此接口连接的 R2 上的 MAC 地址 0013.1a2f.1200 已经被载入配置中。

3.测试 port-security

（1）测试 R1 以合法 MAC 地址访问 R2

```
r1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
r1#
```

说明：因为 R1 以源 MAC0013.1a85.d160 访问 R2，交换机的接口 F0/1 认为 0013.1a85.d160 是安全 MAC，所以 R1 访问 R2 成功。

（2）测试交换机的 F0/1 上的 port-security 违规

r1(config)#int f0/0

r1(config-if)#standby 1 ip 10.1.1.10

说明：因为交换机的 F0/1 允许的最大 MAC 地址数量为 1，而 R1 已经有了一个 MAC 地址，在接口上配置 HSRP 之后，还会产生一个虚拟 MAC 地址，所以这个 HSRP 虚拟 MAC 地址就是第 2 个 MAC 地址，而第 2 个 MAC 地址在交换机的 F0/1 上出现就算是违规。

（3）查看交换机上 port-security 违规后的现象

sw1#

01:39:48: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/1, putting Fa0/1 in

err-disable state

01:39:48: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC

address 0000.0c07.ac01 on port FastEthernet0/1.

01:39:49: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to

down

01:39:50: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down

sw1#

sw1#sh int f0/1

FastEthernet0/1 is down, line protocol is down (err-disabled)

（输出被省略）

sw1#

说明：当交换机的 port-security 违规后,会出现以上 log 提示，并且查看交换机的接口为 err-disabled 状态，并且被 shutdown。

（4）测试交换机上 port-security 另一种违规

r2(config)#int f0/0

r2(config-if)#mac-address 0013.1a85.d160

说明：将 R1 的 MAC 地址 0013.1a85.d160 添加到 R2 的接口上，因为一个 secure port 接口上的合法 MAC 在另外一个 secure port 接口上访问，也算违规。

（5）查看交换机上 port-security 违规后的现象

sw1#

01:46:27: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/2, putting Fa0/2 in

err-disable state

01:46:27: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC

address 0013.1a85.d160 on port FastEthernet0/2.

01:46:28: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to

down

01:46:29: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to down

sw1#

sw1#sh int f0/1

FastEthernet0/1 is up, line protocol is up (connected)

（输出被省略）

sw1#

sw1#sh int f0/2

FastEthernet0/2 is down, line protocol is down (err-disabled)

（输出被省略）

sw1#

说明：可以看见，当一个 secure port 接口上的 MAC 地址在另外一个 secure port

接口出现后，就算违规，而违规动作是对出现重复地址的接口实施的，是为了防止攻击。

4.配置 Port Security MAC 地址老化时间

（1）配置 Port Security MAC 地址老化时间

```
sw1(config)#int f0/1
sw1(config-if)#switchport port-security aging time 1
sw1(config-if)#switchport port-security aging type inactivity
```

说明：配置 Port Security MAC 地址老化时间为 1 分钟，并且相应 MAC 在 1 分钟没有流量的情况下被删除。但此配置只对接口下动态学习到的 MAC 地址生效。

（2）配置手工静态指定的 MAC 地址的老化时间

```
sw1(config)#int f0/1
sw1(config-if)#switchport port-security aging static
```

说明：配置手工静态指定的 MAC 地址在 1 分钟没有流量的情况下被删除。

IP Source Guard

默认情况下，交换机在二层接口上转发数据时，只查看数据包的 MAC 地址，并不会查看数据包的 IP 地址，如果要让交换机根据数据包的 IP 或者同时根据 IP 与 MAC 做出转发决定，有多种方法可以实现。如根据 IP 转发，可以通过在接口上应

用 Port ACL 来实现，如果要根据 MAC 转发，可以通过应用 port-security 来实现，但无论是 Port ACL 还是 port-security，都存在一些局限性，下面介绍一种扩展性较高的安全防护特性- IP Source Guard，IP Source Guard 可以根据数据包的 IP 地址或 IP 与 MAC 地址做出转发决定，如果数据包的 IP 或 MAC 是不被允许的，那么数据包将做丢弃处理。

因为 IP Source Guard 需要根据数据包的 IP 或者同时根据 IP 与 MAC 做出转发决定，所以 IP Source Guard 在工作时，需要有一张 IP 和 MAC 的转发表，在这张表中，明确记录着哪些 IP 是可以转发的，哪些 MAC 可以被转发，其它不能被转发的统统丢弃。这表转发表称为 IP source binding table，并且只能被 IP source guard 使用。而 IP source binding table 表，只有在交换机上开启 DHCP snooping 功能后，才会生成。

IP Source Guard 的这张转发表的条目可以自动学习，也可以手工静态添加，如果是自动学习，是靠 DHCP snooping 功能学习的，所以只有客户端是通过 DHCP 请求获得地址，并且 DHCP 服务器的回复是经过交换机时，才能被 DHCP snooping 学习到。

当在交换机上同时开启了 IP Source Guard 与 DHCP snooping 后，交换机将在开启了的接口上拒绝所有流量通过，只放行 DHCP 流量，并且会自动应用一条 ACL 到接口上，也只有 ACL 允许的 IP 才能通过，这条 ACL 无法在正常配置中查看，只能通过表的方式查看。因为默认可以允许 DHCP 通过，所以主机的 DHCP 请求可以帮助他们从 DHCP 服务器获得地址，当 DHCP 服务器向主机提供地址时，这个信息在穿越交换机时，IP 信息会被记录，并且被自动生成的 ACL 允许转发，这样以后，只有主机从 DHCP 服务器自动获得的 IP 可以通过交换机，而其它 IP 的流量都是被拒绝的，因此可以看出，IP Source Guard 和 DHCP snooping 的配合使用，可以防止一个主机使用其它主机的 IP 地址来攻击网络，因为只有 DHCP 获得的地址能够被交换机转发，其它接口即使配置了相同 IP，都会被 IP Source Guard 拒绝放行。

注：IP Source Guard 自动生成的 ACL 优先于任何 Router ACLs 和 VLAN map。

IP source binding table 的条目除了靠 DHCP snooping 自动学习之外，还可以手工静态配置。

当 IP source binding table 表的内容有任何变化时，自动生成的 ACL 也会自动

改变，自动生成的 ACL 总是与 IP source binding table 表中的条目同步。如果 IP source binding table 表是空的，那么 ACL 将拒绝所有流量通过，也就是最初的状态。

因为 IP Source Guard 可以根据数据包的 IP 地址做出转发，也可以同时根据 IP 与 MAC 地址做出转发，如果要开启同时根据 IP 与 MAC 做出转发，还必须在接口上开启 port security 功能，port security 帮助识别流量的 MAC 地址，并将其添加到 source binding table 表中，最后被 ACL 设置为允许。

注：

★IP source guard 只能在二层接口上开启，并且不支持 EtherChannel。

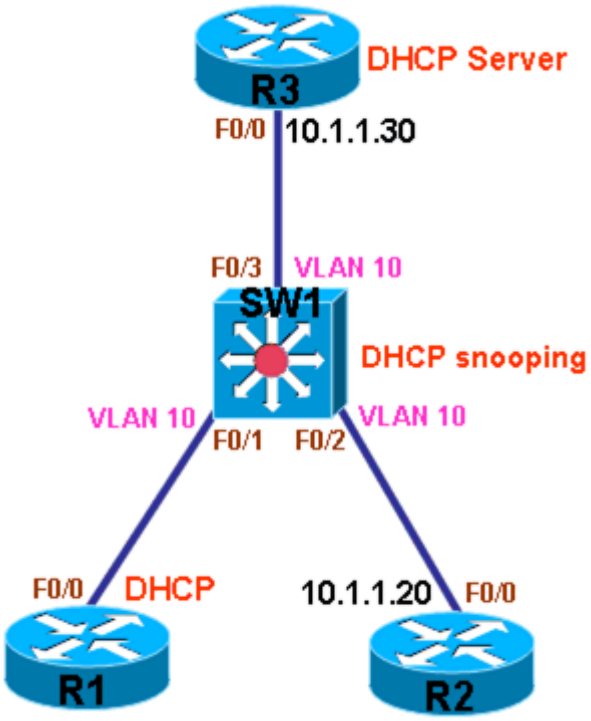
★当 IP source guard 根据 IP 转发在接口上开启，这个接口所在的 VLAN 必须开启 DHCP snooping，

★如果是在 trunk 上开启，DHCP snooping 应该在所有 VLAN 上开启，过滤也对所有 VLAN 生效。

★如果在 trunk 上打开或关闭某一个 VLAN DHCP snooping 的，可能会出问题。

★当 IP source guard 根据 IP 和 MAC 转发在接口上开启，那么 DHCP snooping 和 port security 必须同时开启。

配置



1.网络初始

（1）配置交换机

sw1(config)#int range f0/1 - 3

sw1(config-if-range)#switchport mode access

sw1(config-if-range)#switchport access vlan 10

说明：已经将 R1、R2、R3 划到同一 VLAN 10。

（2）测试 R2 到 R3 的连通性

r2#ping 10.1.1.30

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.30, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：因为 R2 与 R3 在同一 VLAN 10，所以通信正常。

2.在 SW1 上配置 IP Source Guard

（1）在交换机上开启 DHCP snooping

sw1(config)#ip dhcp snooping

sw1(config)#ip dhcp snooping vlan 10

sw1(config)#int f0/3

sw1(config-if)#ip dhcp snooping trust

说明：开启 IP Source Guard，交换机上必须先开启 DHCP snooping，并且将 DHCP Server 的接口设置为 trust 接口。

（2）在接口下开启基于 IP 的 IP Source Guard

sw1(config)#int range f0/1 - 2

sw1(config-if-range)# ip verify source

3.查看结果

（1）查看 IP source binding table

sw1#sh ip source binding

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface

Total number of bindings: 0

说明：初始状态下，IP source binding table 为空，意味着 ACL 会拒绝所有流量通过，只有 DHCP 能通过。

(2)查看自动生成的 ACL

sw1#

##

sw1#sh ip verify source

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
Fa0/1	ip	active	deny-all		10
Fa0/2	ip	active	deny-all		10

sw1#

说明：因为初始状态下，IP source binding table 为空，所以 ACL 会拒绝所有流量通过，只有 DHCP 能通过。

（3）在 R1 开启 DHCP 自动获得地址

r1(config)#int f0/0

r1(config-if)#ip address dhcp

注：因为开启了 DHCP snooping，交换机会在 DHCP 请求从插入 option 82，所以 DHCP server 要接收此数据包:

R3：

r3(config)#int f0/0

r3(config-if)#ip dhcp relay information trusted

（4）查看 R1 DHCP 自动获得的地址

r1#sh protocols f0/0

FastEthernet0/0 is up, line protocol is up

Internet address is 10.1.1.4/24

r1#

说明：R1 从 DHCP Server 获得的地址为 10.1.1.4。

（5）查看 IP source binding table

sw1#sh ip source binding

MacAddress IpAddress Lease(sec) Type VLAN Interface

---------- ---------- -------- ---------- ---- ----------

00:13:1A:85:D1:60 10.1.1.4 86348 dhcp-snooping 10 FastEth

ernet0/1

Total number of bindings: 1

sw1#

说明：因为 R1 从 DHCP Server 获得的地址为 10.1.1.4，所以 DHCP snooping 将该地址记录在 IP source binding table 中。

（6）查看自动生成的 ACL

```
sw1#sh ip verify source
Interface  Filter-type  Filter-mode  IP-address      Mac-address       Vlan
---------  -----------  -----------  --------------  ----------------  ---------
Fa0/1      ip           active       10.1.1.4                          10
Fa0/2      ip           active       deny-all                          10
sw1#
```

说明：因为 R1 从 DHCP Server 获得的地址被 DHCP snooping 记录在 IP source binding table 中，所以自动被 ACL 允许通过。

（7）测试 R1 到 R3 的连通性

r1#ping 10.1.1.30

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.30, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#

说明：因为 R1 的地址被 DHCP snooping 记录在 IP source binding table 中，并且自动被 ACL 允许通过，所以 R1 到 R3 通信正常。

（8）测试 R2 到 R3 的连通性

r2#ping 10.1.1.30

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.30, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：因为 IP source binding table 通过 DHCP snooping 只记录了 R1 从 DHCP 自动获得的地址，但表中没有 R2 的地址，因此 R2 的地址默认被拒绝，所以 R2 到 R3 通信失败。

4.手工添加 IP source binding table

（1）手工添加 R2 的 IP 地址到 IP source binding table 中

sw1(config)#ip source binding 0013.1a2f.1200 vlan 10 10.1.1.20 interface f0/2

说明：手工添加要，要同时指定 MAC 地址，IP 地址，VLAN 号，接口。

（2）查看 IP source binding table

sw1#sh ip verify source

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
Fa0/1	ip	active	10.1.1.4		10
Fa0/2	ip	active	10.1.1.20		10

sw1#

说明：R2 的 IP 地址已经被手工添加到 IP source binding table。

（3）查看自动生成的 ACL

sw1#sh ip verify source

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
Fa0/1	ip	active	10.1.1.4		10
Fa0/2	ip	active	10.1.1.20		10

sw1#

说明：自动 ACL 已经允许 IP source binding table 表中的地址通过。

（4）测试 R2 到 R3 的连通性

r2#ping 10.1.1.30

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.30, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

r2#

说明：因为 R2 的 IP 地址已经被手工添加到 IP source binding table 中，并且被自动 ACL 允许通过，所以 R2 到 R3 通信成功。

5.开启基于 IP 与 MAC 的 IP Source Guard

(1)在接口上开启基于 IP 与 MAC 的 IP Source Guard

sw1(config)#int f0/1

sw1(config-if)#ip verify source port-security

(2)开启 port-security

sw1(config)#int f0/1

sw1(config-if)#switchport port-security

说明：启基于 IP 与 MAC 的 IP Source Guard，必须开启 port-security

（3）查看 IP source binding table

sw1#sh sou

sw1#sh ip source binding

MacAddress IpAddress Lease(sec) Type VLAN Interface

---------- ---------- -------- ---------- ---- ----------

00:13:1A:85:D1:60 10.1.1.4 86034 dhcp-snooping 10 FastEthernet0/1

00:13:1A:2F:12:00 10.1.1.20 infinite static 10 FastEthernet0/2

Total number of bindings: 2

sw1#

说明：可以看到，IP source binding table 中不仅有 IP 地址记录，还有了 MAC 地址记录，只有当数据包的 IP 和 MAC 同时匹配时，才能被放行。

（4）查看自动生成的 ACL

sw1#sh ip verify source

Interface Filter-type Filter-mode IP-address Mac-address Vlan

--------- ---------- ---------- ---------- ---------- ----

Fa0/1 ip-mac active 10.1.1.4 00:13:1A:85:D1:60 10

Fa0/2 ip active 10.1.1.20 10

sw1#

说明：自动 ACL 已经与 IP source binding table 表中的内容同步。

（5）测试 R1 到 R3 的连通性

r1#ping 10.1.1.30

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.30, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r1#

说明：因为 R1 的 IP 与 MAC 在 IP source binding table 中被记录，并且被自动 ACL 允许通过，所以 R1 到 R3 通信正常。

Security with ACL

在三层交换机上，支持多种类型的 ACL，有支持 IP 的 ACL 还有支持 MAC 的 Ethernet ACL。

IP ACL 只过滤 IPv4 流量，而 Ethernet ACL 过滤除 IP 之外的流量，也就是非 IP 流量。

根据 ACL 不同的应用，可以分成三种：Port ACL、Router ACL，VLAN ACL（VLAN map）

Port ACL

是应用在二层接口上的，并没有任何特别之处，将任何 ACL 应用到二层接口后，就称为 Port ACL，但在二层接口只支持 in 方向，一个接口只能使用一条 ACL，IP

或 MAC 的 ACL 都可以应用到二层接口，但不不能在应用到 EtherChannel。

Router ACL

和路由器接口上应用的 ACL 没有区别，将任何 ACL 应用到三层接口后，就称为 Router ACL，但只能是 IP ACL，不能是 MAC ACL。Router ACL 在 in 和 out 方向上都可以使用，每个接口每个方向只能使用一条 ACL。

VLAN ACL（VLAN map）

是使用在 VLAN 与 VLAN 之间的 ACL，相同 VLAN 也是可以过滤的，只要流量是进入或离开指定的 VLAN，都会被过滤，可以同时控制二层与三层流量。准确地讲，VLAN ACL 并不是一个 ACL，只是一个能调用 ACL 到 VLAN 的技术，只要被调用的 ACL 支持什么功能，那么 VLAN ACL 就支持什么功能。当需要控制 IPv4 流量时，VLAN ACL 需要调用 IP ACL，而其它流量则需要靠调用 MAC ACL。当应用 VLAN ACL 后，进入或离开 VLAN 的流量都会被检测，无论是通过二层转发的还是三层转发的。而 VLAN ACL 是不能定义方向的，所以所有经过指定 VLAN 的流都都会被过滤到。

VLAN ACL 根据所调用的 ACL 匹配到的流量，来做出是转发还是丢弃的动作，当被调用的 ACL 匹配到流量后，默认动作是转发，可以改为丢弃，而没有被匹配到的流量，默认也全部丢弃。

以上三种方式的 ACL 可以同时使用，但 Port ACL 优先于任何 ACL。

说明：

★Port ACL 支持标准，扩展 ACL 和 MAC ACL，也就是支持所有类型 ACL。

★接口上可同时应用 IP 和 MAC ACL，而 MAC ACL 是不能过滤 IP 流量的。

★Router ACL 可用在 SVI，三层接口，或 3 层 EtherChannel，每个接口每个方向只能使用一个。

★在配置 IP ACL 时，可以使用数字，也可以使用名字，而命名 ACL 的好处是可以删除单条 ACL 语句，而数字 ACL 则不可以，只能删除整条 ACL。

★交换机上不支持 Dynamic ACL 和 Reflexive ACL。

★VLAN ACL 是不能定义方向的，也就相当于会在两个方向上同时生效，所以在配置 VLAN ACL 时，所以请注意所写的 ACL 一定要考虑来回两个方向，否则只匹配到单向的流量，另外一方向可能被丢掉。

★一个 VLAN ACL 可以用于多个 VLAN，但一个 VLAN 只能使用一个 VLAN ACL。

★被 ACL 拒绝的 ICMP，ACL 将向源发送 ICMP-unreachable。

配置

1.配置 MAC ACL

说明:在 VLAN 或 2 层接口上过滤非 IP 流量，3 层接口上不能使用 MAC ACL。

（1）定义 MAC ACL：

sw1(config)#mac access-list extended ccie

sw1(config-ext-macl)#deny host 0001.0001.0001 host 0002.0002.0002 netbios

sw1(config-ext-macl)#permit any any

说明：拒绝源 MAC 为 0001.0001.0001 发送到 MAC 为 0002.0002.0002 的 netbios 流量，并允许其它所有，MAC ACL 匹配的协议为非 IP 协议，协议可以不指定。

（2）应用 MAC ACL

sw1(config)#int f0/1

sw1(config-if)#mac access-group ccie in

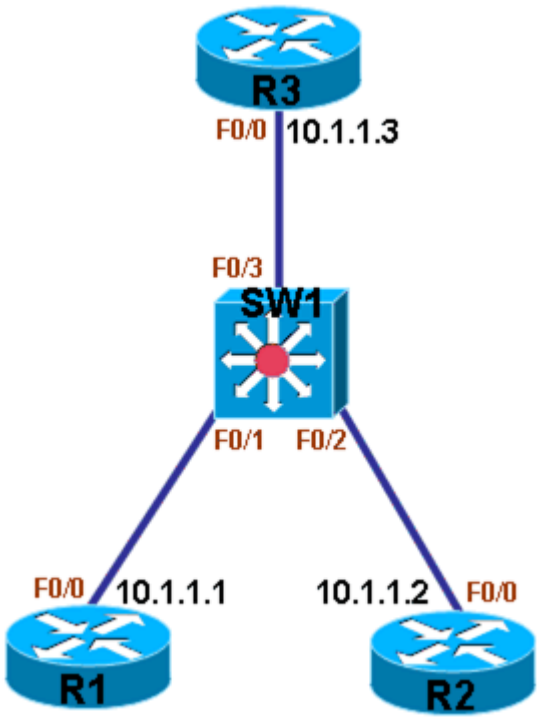
说明：在二层接口 F0/1 上使用，并且只能应用于 in 方向。因为 MAC ACL 只能过滤非 IP 流量，所以难以用实验来验证效果。

Port ACL

2.配置 Port ACL

说明：将任何 ACL 应用于二层接口，便称为 Port ACL。

以下图为例，在 SW1 上配置 Port ACL 拒绝 R3 去往 R1 的流量，并放行其它所有流量。



（1）测试在没有配置 PortACL 之前的网络连通性

r3#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r3#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

r3#

说明：在没有配置 Port ACL 之前，R3 到 R1 与 R2 的通信正常。

（2）在 SW1 上配置 PortACL（只能用于 in 方向）

sw1(config)#access-list 100 deny ip host 10.1.1.3 host 10.1.1.1

sw1(config)#access-list 100 permit ip any any

sw1(config)#int f0/3

http://itcto.taobao.com, 最小的投入，最大的回报，最全、最低价 IT 课程

鸿鹄培训+考试群：301283867

sw1(config-if)#ip access-group 100 in

（3）测试配置 Port ACL 之后的网络连通性

r3#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r3#

r3#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

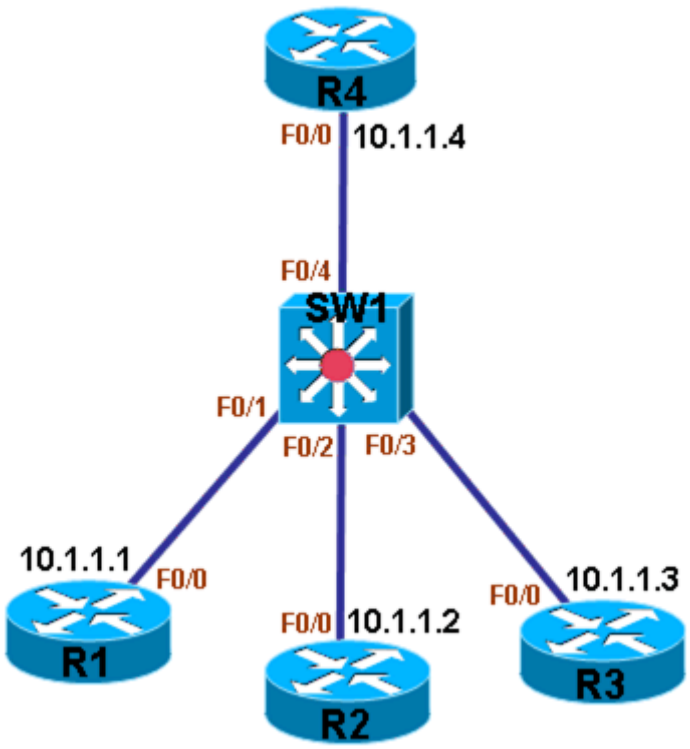
r3#

说明：可以看到，Port ACL 拒绝了 R3 去往 R1 的流量，并放行其它流量。

Router ACL

VLAN ACL

3.配置 VLAN ACL（VLAN map）



说明：以上图为例，所有接口在 VLAN1，配置 VLAN ACL，设置 R4 到 R1 的为默认动作，设置 R4 到 R2 的为丢弃动作，其它流量不作设置。

（1）测试在没有配置 VLAN ACL 之前的网络连通性

r4#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r4#

r4#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

r4#

r4#ping 10.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r4#

说明：在没有配置 VLAN ACL 之前，所有通信正常。

（2）创建各 ACL

sw1(config)#access-list 111 permit ip host 10.1.1.4 host 10.1.1.1

sw1(config)#access-list 111 permit ip host 10.1.1.1 host 10.1.1.4

sw1(config)#access-list 112 permit ip host 10.1.1.4 host 10.1.1.2

sw1(config)#access-list 112 permit ip host 10.1.1.2 host 10.1.1.4

说明：分别匹配 R4 到 R1 的流量，R4 到 R2 的流量，因为 VLAN ACL 没有方向，也就是在两个方向生效，所以请注意所写的 ACL 一定要考虑来回两个方向，否则只匹配到单向的流量，另外一方向可能被丢掉。

（3）配置 VLAN ACL

sw1(config)#vlan access-map ccie 10

sw1(config-access-map)#match ip address 111

sw1(config-access-map)#exit

sw1(config)#vlan access-map ccie 20

sw1(config-access-map)#match ip address 112

sw1(config-access-map)#action drop

sw1(config-access-map)#exit

说明：设置 ACL 111 的流量为默认动作，ACL 112 的流量被明确丢弃，其它不匹配。

（4）应用 VLAN ACL

sw1(config)#vlan filter ccie vlan-list 1

说明：将 VLAN ACL 应用于 VLAN 1，也可以应用于多个 VLAN，但不能设置方向。

（5）测试配置 VLAN ACL 之后的网络连通性

r4#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r4#

r4#ping 10.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r4#

r4#ping 10.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r4#

说明：R4 去往 R1 的流量是默认动作，证明默认动作是允许转发的，而 R4 去往 R2 的流量被明确丢弃了，最后其它没有设置的流量，全部是被隐含拒绝的。

Storm Control

在默认情况下，交换机在接口上收到任何数据包，将尽全力转发，只有在硬件性能不足的情况下，才会丢弃数据包。在某些时候，由于协议错误，配置错误或人为攻击，导致网络流量增大时，将影响网络的性能，在这种情况下，需要在交换机上限制流量占用接口的带宽，则可以使用 Storm control 来实现。

Storm control 可以在交换机接口上限制 broadcast,,multicast,以及 unicast 的流量带宽，在接口上开启了 Storm control 后，Storm control 便开始监控流量从接口到交换机总线的速度，并统计每秒通过的数据包，将当前流量的速度与预先配置好的阀值作比较，阀值分为上限（rising suppression level）和下限（falling suppression level），当流量的速度达到上限的阀值后，流量就会被 block，直到流量低于下限后，才会恢复正常。

在配置阀值时，可以使用以下标准来衡量带宽：

使用接口总带宽的百分比。

每秒通过的数据包个数（PPS-Packets Per Second）。

每秒通过的 Bit 数（Bps-Bit Per Second）

注：使用接口带宽百分比时，100 percent 表示不限制，0.0 表示流量全部丢弃。

默认情况下，接口上是没有流量限制的。在配置阀值时，上限（rising suppression level））必须配置，而下限（falling suppression level）却可以不配置，在没有配置下限时，下限将采用和上限相同的值。

当配置了 storm-control 限制 multicast 时,如果 multicast 的流量超过上限,那么所有的 multicast 流量都会被丢弃，其中包含如 OSPF，EIGRP 的流量都会被丢弃，但是 BPDU，CDP 的流量不会被丢弃。

配置 storm-control 时，可以在物理接口和 EtherChannel 上配置，如果是配在 EtherChannel 上，那么配置将对 EtherChannel 中的所有物理接口生效。

在配置 storm-control 时，可以设置流量达到上限后，采取相应的处理动作，可配置的动作分为 Shutdown 和 Trap，Shutdown 是在流量达到上限后，将接口陷入 error-disable 状态，Trap 是在流量达到上限后，产生一条 SNMP Trap 消息，而默认的动作是丢弃流量而不生产 SNMP Trap 消息。

配置

说明：在配置 BPS 和 PPS 作为标准时，可以使用 K，M 以及 G 为单位。

1.在交换机接口上配置 storm-control

（1）在交换机上开启 storm-control，并定义上限和下限：

```
sw1(config)#int f0/1

sw1(config-if)# storm-control unicast level pps 100 80
```

说明：定义的上限为 PPS 100，下限为 PPS 80。

（2）定义流量达到上限后，采限的动作：

```
sw1(config)#int f0/1

sw1(config-if)#storm-control action trap
```

说明：定义的违规动作为生产 SNMP Trap。

（3）查看接口 storm-control 状态：

正常状态：

```
sw1#sh storm-control unicast
Interface  Filter State   Upper        Lower        Current
---------  -------------  -----------  -----------  ----------
Fa0/1      Forwarding        100 pps       80 pps       10 pps
sw1#
```

说明：接口流量正常的状态，流量都被放行。

被丢的状态：

```
sw1#sh storm-control unicast
Interface  Filter State   Upper        Lower        Current
---------  -------------  -----------  -----------  ----------
Fa0/1      Blocking          100 pps       80 pps      273 pps
sw1#
```

说明：接口流量达到上限的状态，流量都被丢弃。

SPAN and RSPAN

交换机在收到数据包后，将根据数据包的目标 MAC 地址来做出转发决定，只有与目标 MAC 地址对应的接口才能收到数据包，交换机并不会将数据包转发到不相关的接口上。

当网络管理者需要监控网络中的流量时，装有监控软件的主机接到交换机上之后，并不能像预期那样能够收到所要监控的流量，除非流量是原本就要发送给自己的，或者是广播流量。对于装有监控软件的主机想要从交换机上接收到其它流量，

就必须依靠交换机的协助，通过交换机将其它正常流量复制一份发送到接有监控主机的接口即可。

要让交换机将正常流量复制下来并发送到相关端口，需要靠 SPAN（Switched Port Analyzer）来实现。SPAN 允许将交换机的任意端口或任意 VLAN 上的流量复制之后发送到其它任何端口上。

因为 SPAN 要将某端口或 VLAN 的流量复制一份发送到其它端口，所以 SPAN 需要明确源和目的，SPAN 只复制从源收到的流量，然后只发送到目的。

SPAN 可以将某些接口或某些 VLAN 的流量复制下来，所以 SPAN 的源可以是物理接口，也可以是 VLAN，并且可以定义多个物理接口或多个 VLAN。复制的流量可以是接收到的，可以是发送出去的，也可以是双向的，默认为双向流量。而 SPAN 的目的有时只能是物理接口，有时只能是 VLAN，需要视情况而定。

并不是经过交换机的任何流量都能被 SPAN 复制，某些流量是不能被复制的，如三层流量需要被交换机路由到源 VLAN 的流量是不能被复制的，也就是说需要交换机查路由表将流量发送到源 VLAN 的流量是不能被复制的，但是从源 VLAN 被路由到外面去的流量还是可以被复制的。

当物理接口或 VLAN 被 SPAN 定义为源之后，源端口或源 VLAN 的流量是不会受到任何影响的，但 SPAN 的目标端口除了接收 SPAN 的流量外，不能再接收其它任何正常的流量。因此在 SPAN 的源和目标在同台交换机与不同交换机时，操作是不一样的，当 SPAN 的源和目的在同台交换机上时，被称为 Local SPAN，即 SPAN，而当 SPAN 的源和目的在不同交换机上时，被称为 Remote SPAN，即 RSPAN。

因为 RSPAN 是跨越了多台交换机的，而目标端口除了接收 SPAN 的流量外，不能再接收其它任何正常的流量，所以在为 RSPAN 定义目标时，不能将目标定义为物理接口，因为连接交换机的物理接口通常还有其它流量传播，所以在实施 RSPAN 时，必须将 SPAN 复制的流量通过发送到某个 VLAN，然后从 Trunk 上传到目标交换机，这个 VLAN 就是 RSPAN VLAN，从源到目标的每台交换机都应该配置 RSPAN VLAN。

在配置 RSPAN 时，只需要在源交换机和目标交换机上配置即可，如果中间还有交换机，中间的交换机只需要配置 RSPAN VLAN，而不需要配置其它任何参数。在源交换机上，将 SPAN 的源定义为物理接口或 VLAN，且必须将目的定义为 RSPAN VLAN，不能定义为物理接口。在目标交换机上将 SPAN 的源定义为 RSPAN VLAN，并且将目的定义为物理接口，目标交换机从 RSPAN VLAN 中收到流量后，将转发到目标接口。

在某些 IOS 版本中，配置 RSPAN 时，源交换机在定义目标时，不仅需要将 RSPAN VLAN 指定为目标，并且还需要指定一个 reflector-port 端口，系统是将流量发送到 reflector-port 端口后，再由 reflector-port 端口发送到 RSPAN VLAN，最终发送到目标交换机。而当配置一个接口为 reflector-port 端口后，这个接口就不能正常使用了。

在配置 SPAN 时，会有以下一些限制条件：

★交换机上支持最多两个 SPAN 会话。

★最多可以有 64 个目标端口，而源端口无上限。

★3 层接口也可以作为源或目的。

★源和目标的速率要一致。

★源端口可以是 EtherChannel, Fast Ethernet, Gigabit Ethernet 以及其它接口

★源也可以是 access port, trunk port, routed port, voice VLAN port

★如果一个目标端口在源 VLAN 中，则会被源排除在外。

★当源端口是 trunk 时，那就是所有 VLAN 的流量都被复制，但可以过滤某些 VLAN，配置时，

★就只有在 list 中的 vlan 的流量才会被复制。

★当一个接口变成 SPAN 的目标端口后，所有配置丢失，关闭 SPAN 后，则配置恢复。

★如果目标端口在 EtherChannel 组中，将从组中消失。

★目标不能是安全端口，也不能是源。

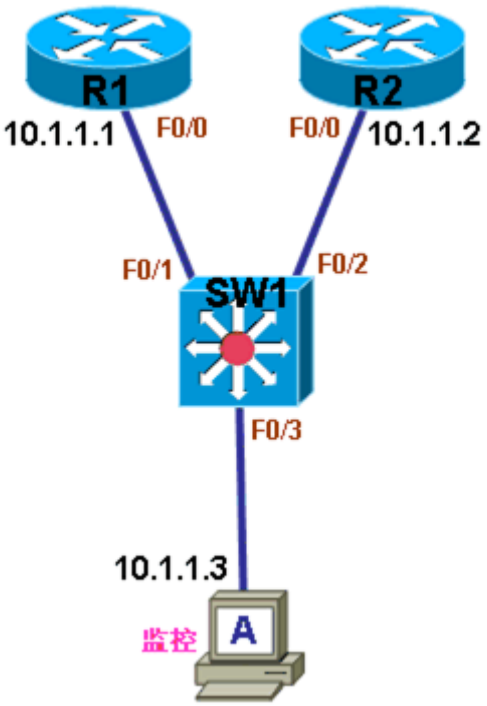
★目标也不能是 EtherChannel group 或正常 VLAN。

★一个目标端口不能成为两个会话的目标。

★目标端口不会转发 SPAN 流量之外的任何流量。

★在配置源时，多个源可以一条命令配完，也可以分多条命令配置。

配置 SPAN



1.测试网络连通性

（1）测试 R2 到 R1 的网络连通性

r2#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：因为交换机为正常状态，所以 R2 到 R1 的通信正常。

（2）测试 R2 到主机 A 的网络连通性

r2#ping 10.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：因为交换机为正常状态，所以 R2 到主机 A 的通信正常。

2.在 SW1 上配置 SPAN

（1）指定接口 F0/1 为 SPAN 源

sw1(config)#monitor session 1 source interface f0/1

说明：指定连接 R1 的接口 F0/1 为 SPAN 源接口，并且监控双向流量（默认为

双向）。

（2）指定接口 F0/3 为 SPAN 目的

sw1(config)#monitor session 1 destination interface f0/3

说明：指定连接主机 A 的接口 F0/3 为 SPAN 目的。

3.测试配置 SPAN 后的网络连通性

（1）测试 R2 到 R1 的网络连通性

r2#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

r2#

说明：因为 SPAN 不影响源接口的通信，所以 R2 到 R1 的通信正常。

（2）测试 R2 到主机 A 的网络连通性

r2#ping 10.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

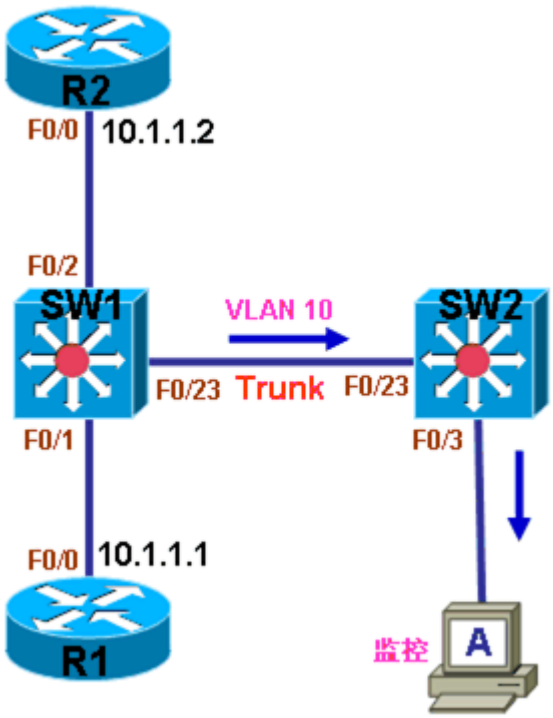
.....

Success rate is 0 percent (0/5)

r2#

说明：因为 SPAN 的目标端口除了接收 SPAN 的流量外，不能再接收其它任何正常的流量，所以 R2 到 R1 的通信失败。

配置 RSPAN



1.测试网络连通性

（1）测试 R2 到 R1 的网络连通性

r2#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

r2#

说明：因为交换机为正常状态，所以 R2 到 R1 的通信正常。

2.交换机上配置 RSPAN

（1）在两台交换机上配置 RSPAN VLAN 10

SW1：

sw1(config)#vlan 10

sw1(config-vlan)#remote-span

SW2：

sw2(config)#vlan 10

sw2(config-vlan)#remote-span

说明：VLAN 号可任意配置，只需将其定义为 RSPAN VLAN 即可。

（2）在源交换机上指定 RSPAN 源

sw1(config)#monitor session 1 source interface f0/1

说明：指定连接 R1 的接口 F0/1 为 RSPAN 源接口，并且监控双向流量（默认为双向）。

（3）在源交换机上指定 RSPAN 目的

sw1(config)#monitor session 1 destination remote vlan 10 reflector-port f0/2

说明：RSPAN 的目的只能为 RSPAN VLAN，并且此 IOS 版本需要指定 reflector-port，将连接 R2 的接口 F0/2 指定为 reflector-port。

（4）在目标交换机上指定 RSPAN 源

sw2(config)#monitor session 1 source remote vlan 10

说明：目标交换机上的 RSPAN 源只能为 RSPAN VLAN。

（5）在目标交换机上指定 RSPAN 目的

sw2(config)#monitor session 1 destination interface f0/3

说明：目标交换机上的 RSPAN 目的是物理端口。

3.测试配置 RSPAN 后的网络连通性

（1）测试 R2 到 R1 的网络连通性

r2#ping 10.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

r2#

说明：虽然 SPAN 不影响源接口的通信，但是当配置一个接口为 reflector-port 端口后，这个接口就不能正常使用了，所以 R2 到 R1 的通信失败。

4.更多 SPAN 配置命令

（1）配置过滤 Trunk 上的 VLAN

sw1(config)#monitor session 1 filter vlan 1 - 3 , 5

说明：当将 Trunk 接口配置为 SPAN 源时，Trunk 上所有的 VLAN 流量都会被复制，以上配置为只复制 Trunk 上 VLAN 1，VLAN2，VLAN3，VLAN5 的流量。

UDLD (UniDirectional Link Detection)

在交换机没有使用任何模块的接口上，如果接口出现故障，如物理故障，或不能发送数据与接收数据，自己能够快速察觉；而当接口上使用了模块后，如光纤模块，当模块上出现故障后，交换机并不能保证在任何时候都能察觉。当交换机的接口如果自己不能发送或接收数据，而对方却能发送或接收数据时，这可能会引起 STP 环路，这样的故障被称为单向链路故障，而交换机上的特性 UDLD 则是用来专门检测此类单向链路故障的。

UDLD 使用一层协议来做单向链路检测，开启了 UDLD 的接口会向外发送 Udld

hello，可以理解为交换机之间的心跳，收到 Udld hello 的交换机必须向邻居回复，如果超过一定时间没有回复，那么就认为单向链路故障已经出现，就会采取相应措施。

UDLD 的运行分为两种模式：normal (默认) 和 aggressive。

Normal 模式只能检测光纤上的单向链路故障，而 aggressive 模式不仅能够检测光纤上的单向链路故障，还能检测双绞线上的单向链路故障。当使用 ormal 模式时，检测到单向链路故障后，接口没有变化，而使用 aggressive 模式时，检测到单向链路故障后，接口会被 disable。

当交换机一端支持 UDLD，而另一端不支持 UDLD，这样的链路连在一起，是不能做 UDLD 检测的，所以在配置 UDLD 时，必须相连的链路两端都要配置 UDLD，并且要配置成相同的模式。

配置

1.配置 UDLD

（1）在所有接口上开启 UDLD（必须为光纤接口）

```
sw1(config)#udld enable
```

或

```
sw1(config)#udld aggressive
```

（2）查看 UDLD：

```
sw1#show udld
```

说明：没有光纤口，没有结果。

（3）在接口下开启 UDLD（接口下的配置将覆盖全局模式下的配置）

sw1(config)#int f0/8

sw1(config-if)#udld port aggressive

（4）查看接口下的 UDLD

sw1#show udld f0/8

Interface Fa0/8

Port enable administrative configuration setting: Enabled / in aggressive mode

Port enable operational state: Enabled / in aggressive mode

Current bidirectional state: Unknown

Current operational state: Link down

Message interval: 7

Time out interval: 5

No neighbor cache information stored

sw1#

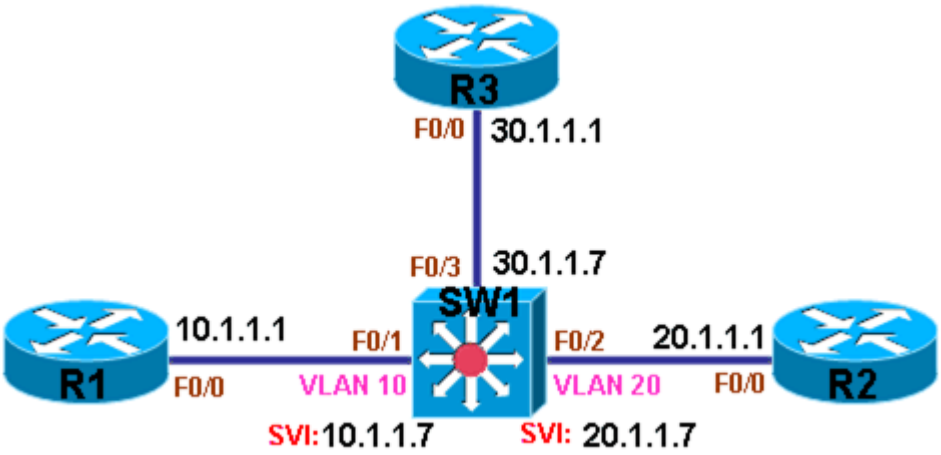
说明：可以看到，接口下已经开启 UDLD。

2.恢复被 UDLD 关闭的接口

（1）恢复被 UDLD 关闭的接口

先 Shut 再 no shut

Fallback Bridging



从上图中的网络环境可以看出，当 R1，R2 和 R3 使用 IP 协议互相通信时，只需要让 SW1 支持 IP 路由转发功能即可，方法为在 SW1 上输入命令 ip routing。

当 R1，R2 和 R3 不使用 IP 协议互相通信时，如使用 DECnet.等非 IP 协议通信时，交换机并不能为其提供 VLAN 与 VLAN 之间的数据转发以及 VLAN 与三层接口之间的数据转发。要让这些在不同 VLAN 的主机能够以非 IP 协议通信，交换机必须将这些需要通信的 VLAN 或三层接口配置到同一个组中即可，这个组就是 VLAN bridging，通常称为 fallback bridging。

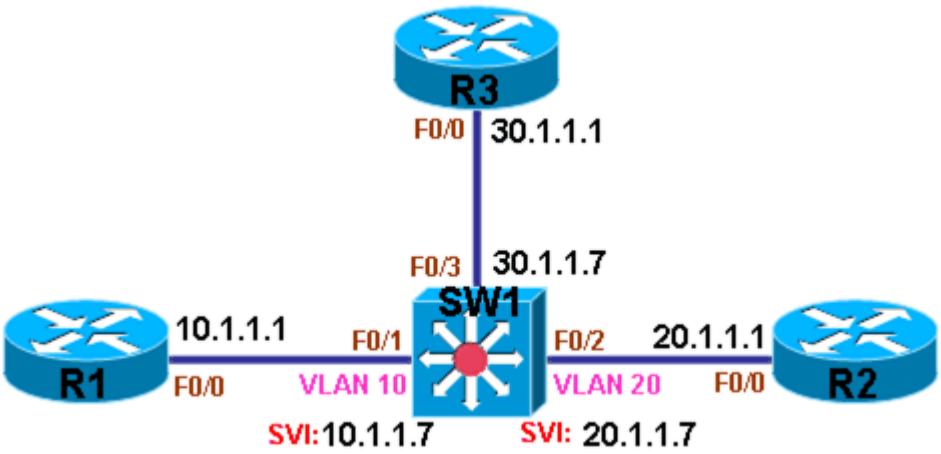
fallback bridging 为交换机上 VLAN 与 VLAN 之间以及 VLAN 与三层接口之间提供数据转发，需要通信的 VLAN 或三层接口，需要配置到同一个组中。可以看出，既然使用非 IP 协议进行通信，交换机上无论是 VLAN，还是三层接口，都是不需要配置 IP 地址的。

★交换机上最多可配置 32 个 fallback bridging 组。

★一个接口或 VLAN 只能属于一个组。

★fallback bridging 除了不能转发 IP 协议外， Address Resolution Protocol (ARP), reverse ARP (RARP), LOOPBACK, and Frame Relay ARP 都可以转发。

配置



1.在交换机上配置 fallback bridging

（1）在交换机上开启 fallback bridging

```
sw1(config)#bridge 10 protocol vlan-bridge
```

说明：创建 fallback bridging 组，组号为 10。

（2）将相应接口与 VLAN 划入 fallback bridging 组中

```
sw1(config)#int vlan 10
sw1(config-if)#bridge-group 10

sw1(config)#int vlan 20
sw1(config-if)#bridge-group 10
```

```
sw1(config)#int f0/3
sw1(config-if)#bridge-group 10
```

说明：将 VLAN 10，VLAN 20，F0/3 划入组 10 中，VLAN 10，VLAN 20 与 F0/3 可以使用非 IP 协议进行通信。

2.查看 fallback bridging

（1）查看 fallback bridging

```
sw1#show bridge group

Bridge Group 10 is running the VLAN Bridge compatible Spanning Tree protocol

    Port 31 (Vlan10) of bridge group 10 is forwarding
    Port 32 (Vlan20) of bridge group 10 is forwarding
    Port 30 (FastEthernet0/3) of bridge group 10 is forwarding

sw1#
```

说明：VLAN 10，VLAN 20，F0/3 已经划入组 10 中。

（2）查看 fallback bridging 转发状态

```
sw1#sh bridge
Br Group    Mac Address      State     Type       Ports
-------    ----------      -----     ---        -----
  10       0013.1a2f.0380   Forward   DYNAMIC    Fa0/3
  10       0013.1a2f.1200   Forward   DYNAMIC    Vl20 Fa0/2
  10       0013.1a85.d160   Forward   DYNAMIC    Vl10 Fa0/1
  10       0013.805c.4b17   Forward   DYNAMIC    Vl10 Fa0/23
sw1#
```

说明：因为是以太网交换机，即使不使用 IP 协议，也会有 MAC 地址，fallback bridging 开启后，交换机也需要根据主机的 MAC 地址做出转发决定，并且可以手工定义哪些 MAC 被丢弃或被转发。

3.手工定义 fallback bridging 转发表

（1）定义转发表

```
sw1(config)#bridge 10 address 0001.0002.0003 discard

sw1(config)#bridge 10 address 0004.0005.0006 forward
```

说明：定义 MAC 地址为 0001.0002.0003 的流量被丢弃，MAC 地址为 0004.0005.0006 的流量被转发。

(2)查看 fallback bridging 的地址表

```
sw1#sh bridge

00:11:32: %SYS-5-CONFIG_I: Configured from console by console

Br Group    Mac Address       State      Type       Ports

-------    ----------       -----      ---        -----

  10        0001.0002.0003    Discard    STATIC      -

  10        0004.0005.0006    Forward    STATIC      -

  10        0013.1a2f.0380    Forward    DYNAMIC     Fa0/3

  10        0013.1a2f.1200    Forward    DYNAMIC     Vl20 Fa0/2

  10        0013.1a85.d160    Forward    DYNAMIC     Vl10 Fa0/1

  10        0013.805c.4b17    Forward    DYNAMIC     Vl10 Fa0/23

sw1#
```

说明：存在动态学习的，根据交换机 MAC 地址为准，也有手工静态配置的。

IEEE 802.1x（DOT1X）Authentication

简单的说，IEEE 802.1x 是一种认证技术，是对交换机上的 2 层接口所连接的主机做认证，当主机接到开启了 IEEE 802.1x 认证的接口上，就有可能被认证，否则就有可能被拒绝访问网络。在接口上开启 IEEE 802.1x 认证后，在没有通过认证之前，只有 IEEE 802.1x 认证消息，CDP，以及 STP 的数据包能够通过。

因为主机接到开启了 IEEE 802.1x 认证的接口后，需要通过认证才能访问网络，要通过认证，只要输入合法的用户名和密码即可。交换机收到用户输入的账户信息后，要判断该账户是否合法，就必须和用户数据库做个较对，如果是数据库中存在的，则认证通过，否则认证失败，最后拒绝该用户访问网络。

交换机提供给 IEEE 802.1x 认证的数据库可以是交换机本地的，也可以是远程服务器上的，这需要通过 AAA 来定义，如果 AAA 指定认证方式为本地用户数据库（Local），则读取本地的账户信息，如果 AAA 指定的认证方式为远程服务器，则读取远程服务器的账户信息，AAA 为 IEEE 802.1x 提供的远程服务器认证只能是 RADIUS 服务器，该 RADIUS 服务器只要运行 Access Control Server Version 3.0（ACS 3.0）或更高版本即可。

当开启 IEEE 802.1x 后,并且连接的主机支持 IEEE 802.1x 认证时，将得出如下结果：

★如果认证通过，交换机将接口放在配置好的 VLAN 中，并放行主机的流量。

★如果认证超时，交换机则将接口放入 guest vlan。

★如果认证不通过，但是定义了失败 VLAN，交换机则将接口放入定义好的失败 VLAN 中。

★如果服务器无响应，定义放行，则放行。

注：不支持 IEEE 802.1x 认证的主机，也会被放到 guest vlan 中。

提示：

当交换机使用 IEEE 802.1x 对主机进行认证时，如果主机通过了认证，交换机还可以根据主机输入的不同账户而将接口划入不同的 VLAN，此方式称为 IEEE 802.1x 动态 VLAN 认证技术，并且需要在 RADIUS 服务器上做更多的设置。本文档并不对 IEEE 802.1x 动态 VLAN 认证技术做更多的介绍，如有需要，本文档将补充对 IEEE 802.1x 动态 VLAN 认证技术的详细介绍与配置说明。

当主机认证失败后，交换机可以让主机多次尝试认证，称为重认证（re-authentication），在交换机上开启 re-authentication 功能即可，默认是关闭的。并且还可以配置认证时间间隔，默认 60 秒，默认可以尝试 2 次重认证。

如果要手工对某接口重认证，在 enable 模式输入命令 dot1x re-authenticate interface

在交换机接口上开启认证后，只要接口从 down 状态到 up 状态，就需要再次认证。

dot1x port-control auto 接口开认证

对于开启了认证的接口，分为两种状态，unauthorized（未认证的）和 authorized（认证过的）。

接口的状态，可以手工强制配置，接口可选的配置状态分以下 3 种：

force-authorized：就是强制将接口直接变认证后的状态，即 authorized 状态。

force-unauthorized：就是强制将接口直接变没有认证的状态，即 unauthorized 状态。

Auto：就是正常认证状态，主机通过认证，则接口在 authorized 状态，认证失败，则在 unauthorized 状态。.

注：当交换机接口从 up 到 down，或者主机离开了发送 logoff，都将合接口重新变成 unauthorized 状态。

开启了 IEEE 802.1x 认证的接口除了有状态之外，还有主机模式，称为 Host Mode，分两种模式：single-host 和 multiple-host。

在 single-host 模式（默认），表示只有一台主机能连上来。

在 multiple-hosts 模式，表示可以有多台主机连上来，并且一台主机认证通过后，所有主机都可以访问网络。

当认证超时或主机不支持认证时，接口将被划到 guest VLAN ，当认证失败时，将被划失败 VLAN，也就是受限制的 VLAN（restricted VLAN），guest VLAN 和 restricted VLAN 可以定义为同一个 VLAN，并且每接配置的。其实即使划入这个 VLAN 后，也会告诉客户是认证通过，要不然会得不到 DHCP。

注：

★IEEE 802.1x 认证只能配置在静态 access 模式的接口上。

★正常工作在 IEEE 802.1x 的接口被称为 port access entity (PAE) authenticator。

★在认证超时或主机不支持认证时，才会将接口划入 guest-vlan，在 IOS 12.2(25)SE 之前，是不会将支持认证但认证失败的接口划入 guest-vlan 的，如果要开启 guest-vlan supplicant 功能，要全局配置 dot1x guest-vlan supplicant。

配置

1.定义认证方式

（1）定义 IEEE 802.1x 使用本地账户数据库认证

sw1(config)#aaa new-model

sw1(config)#aaa authentication dot1x default local

（2）定义 IEEE 802.1x 使用 RADIUS 服务器认证

sw1(config)#aaa new-model

sw1(config)#aaa authentication dot1x default group radius

（3）定义 RADIUS 服务器

sw1(config)#radius-server host 10.1.1.1 auth-port 1645 acct-port 1646 key cisco

说明：定义 RADIUS 服务器地址为 10.1.1.1，密码为 cisco，认证端口 UDP 1645，默认为 1812，accounting 端口为 1646，默认为 1813.。

2.开启 IEEE 802.1x 认证

（1）全局开启 IEEE 802.1x 认证

sw1(config)#dot1x system-auth-control

说明：必须在全局开启 IEEE 802.1x 认证。

（2）在接口下开启 IEEE 802.1x 认证

说明：接口模式必须为静态 access

sw1(config)#int f0/1

sw1(config-if)#swi

sw1(config-if)#switchport mo

sw1(config-if)#switchport mode acce

sw1(config-if)#switchport mode access

sw1(config-if)#dot1x port-control auto

说明：将接口设置为 auto 状态，即正常认证状态。

（3）定义 guest VLAN 和 restricted VLAN

sw1(config)#vlan 10

sw1(config-vlan)#exit

sw1(config)#vlan 20

sw1(config-vlan)#exit

sw1(config)#int f0/1

sw1(config-if)#dot1x guest-vlan 10

sw1(config-if)#dot1x auth-fail vlan 20

说明：先在交换机上配置好 VLAN，然后定义 guest-vlan 和 restricted VLAN（auth-fail vlan），两个 VLAN 可以设置为同一个。

3.查看配置

（1）查看接口配置信息

sw1#sh run int f0/1

Building configuration...

Current configuration : 153 bytes

!

interface FastEthernet0/1

 switchport mode access

```
 dot1x pae authenticator

 dot1x port-control auto

 dot1x guest-vlan 10

 dot1x auth-fail vlan 20

end


sw1#
```

说明：可以看到，将接口配置为 IEEE 802.1x 后，接口自动设置为 dot1x pae authenticator。

（2）查看接口 IEEE 802.1x 状态

```
sw1#sh dot1x interface f0/1

Dot1x Info for FastEthernet0/1

-----------------------------------

PAE                     = AUTHENTICATOR

PortControl             = AUTO

ControlDirection        = Both

HostMode                = SINGLE_HOST

ReAuthentication        = Disabled

QuietPeriod             = 60

ServerTimeout           = 30

SuppTimeout             = 30

ReAuthPeriod            = 3600 (Locally configured)
```

```
ReAuthMax                = 2
MaxReq                   = 2
TxPeriod                 = 30
RateLimitPeriod          = 0
Auth-Fail-Vlan           = 20
Auth-Fail-Max-attempts   = 3
Guest-Vlan               = 10


sw1#
```

说明：以上是 IEEE 802.1x 的默认参数。

4.设置其它 IEEE 802.1x参数

（1）开启重认证功能（默认关闭）

```
sw1(config)#int f0/1
sw1(config-if)#dot1x reauthentication、
```

（2）重认证次数（默认 2 次）

```
sw1(config)#int f0/1
sw1(config-if)#dot1x max-reauth-req 3
```

（3）在划到限制 VLAN 之前，可以尝试几次输入（默认是 3 次）

```
sw1(config)#int f0/1
sw1(config-if)#dot1x auth-fail max-attempts 2
```

（4）设置主机模式（默认 Single-host）

```
sw1(config)#int f0/1
```

```
sw1(config-if)#dot1x host-mode multi-host
```

(5)查看配置

```
sw1#sh dot1x interface f0/1

Dot1x Info for FastEthernet0/1

-----------------------------------

PAE                       = AUTHENTICATOR

PortControl               = AUTO

ControlDirection          = Both

HostMode                  = MULTI_HOST

ReAuthentication          = Enabled

QuietPeriod               = 60

ServerTimeout             = 30

SuppTimeout               = 30

ReAuthPeriod              = 3600 (Locally configured)

ReAuthMax                 = 3

MaxReq                    = 2

TxPeriod                  = 30

RateLimitPeriod           = 0

Auth-Fail-Vlan            = 20

Auth-Fail-Max-attempts    = 2

Guest-Vlan                = 10
```

```
sw1#
```

说明：以上的配置已经显示。

5.强制接口为 authorized状态

(1)直接将接口 F0/2 配置为 authorized 状态

```
sw1(config)#int f0/2
sw1(config-if)#switchport mode access
sw1(config-if)#dot1x port-control force-authorized
```

（2）查看接口 F0/2 的配置

```
sw1#sh run int f0/2
Building configuration...

Current configuration : 82 bytes
!
interface FastEthernet0/2
 switchport mode access
 dot1x pae authenticator
end

sw1#
```

说明：可以看到，接口只有 dot1x pae authenticator，没有 auto 字样

（3）查看接口 F0/2 的状态

```
sw1#sh dot1x interface f0/2
```

```
Dot1x Info for FastEthernet0/2

-----------------------------------

PAE                       = AUTHENTICATOR

PortControl               = FORCE_AUTHORIZED

ControlDirection          = Both

HostMode                  = SINGLE_HOST

ReAuthentication          = Disabled

QuietPeriod               = 60

ServerTimeout             = 30

SuppTimeout               = 30

ReAuthPeriod              = 3600 (Locally configured)

ReAuthMax                 = 2

MaxReq                    = 2

TxPeriod                  = 30

RateLimitPeriod           = 0


sw1#
```

说明：接口已经为 authorized 状态。

(4)直接将接口 F0/3 配置为 authorized 状态

```
sw1(config)#int f0/3

sw1(config-if)#switchport mode access

sw1(config-if)#dot1x pae authenticator
```

说明：命令 dot1x pae authenticator 也将接口直接设置为 authorized 状态。

(5)查看接口 F0/3 的配置

```
sw1#sh run int f0/3
Building configuration...

Current configuration : 82 bytes
!
interface FastEthernet0/3
 switchport mode access
 dot1x pae authenticator
end

sw1#
```

说明：可以看到，命令 dot1x pae authenticator 等同命令 dot1x port-control force-authorized。

（6）查看接口 F0/2 的状态

```
sw1#sh dot1x interface f0/3
Dot1x Info for FastEthernet0/3
-----------------------------------
PAE                     = AUTHENTICATOR
PortControl             = FORCE_AUTHORIZED
ControlDirection        = Both
```

```
HostMode                = SINGLE_HOST

ReAuthentication        = Disabled

QuietPeriod             = 60

ServerTimeout           = 30

SuppTimeout             = 30

ReAuthPeriod            = 3600 (Locally configured)

ReAuthMax               = 2

MaxReq                  = 2

TxPeriod                = 30

RateLimitPeriod         = 0


sw1#
```

说明：接口已经为 authorized 状态。

6.guest-vlan supplicant

说明：在认证超时或主机不支持认证时，才会将接口划入 guest-vlan，在 IOS 12.2(25)SE 之前，是不会将支持认证但认证失败的接口划入 guest-vlan 的，如果要开启 guest-vlan supplicant 功能，做如下配置：

```
sw1(config)#dot1x guest-vlan supplicant
```

说明：某些 IOS 版本，此命令是隐藏命令，需要手工全部输入。

7.主机设置网卡 IEEE 802.1x 认证

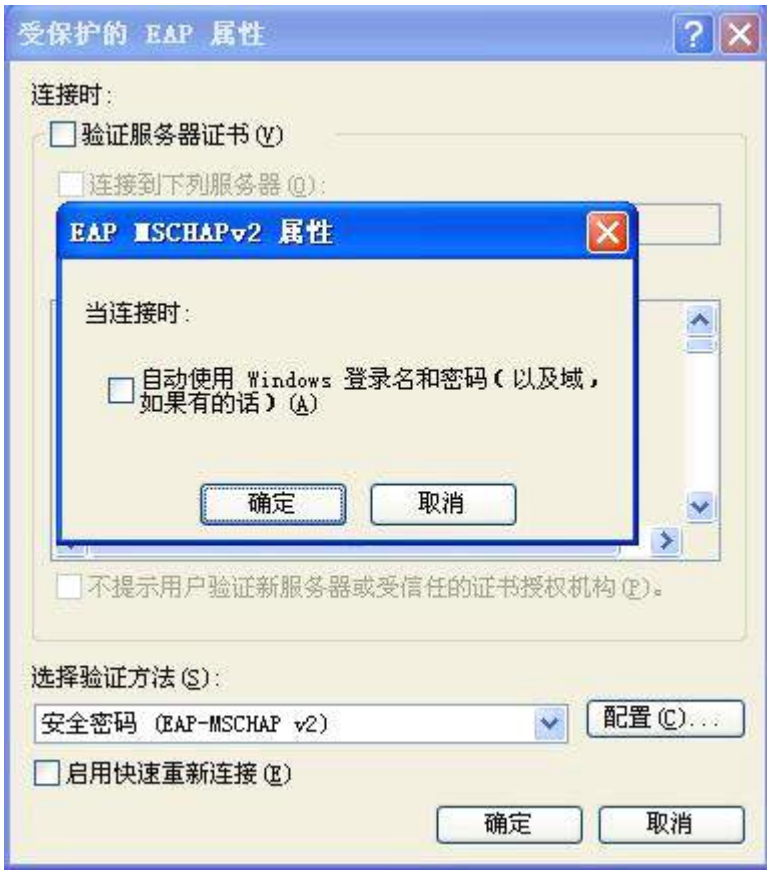
说明：下面以 XP 操作系统的主机说明如何开启网卡的 IEEE 802.1x 认证

（1）点击本地连接的“属性”，出现如下窗口：

说明：勾选“启用此网络的 IEEE 802.1x 验证”，并选择 EAP 类型为“受保护的 EAP (PEAP)”，然后点“属性”，出现如下窗口：

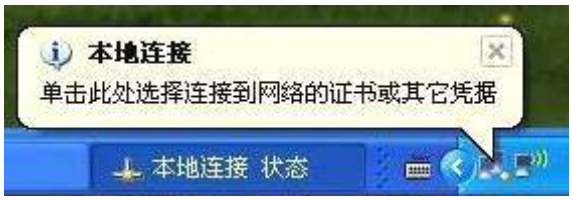
说明：请不要勾选“验证服务器证书”，选择验证方法为“安全密码(EAP-MSCHAP v2)”，然后点击“配置”，出现如下窗口：

鸿鹄培训+考试群：301283867



说明：请不要勾选“自动使用 Windows 登录名和密码（以及域，如果有的话）”。

（2）当将此网卡连上开启 IEEE 802.1x 认证的交换机端口后，如出现如下提示：



说明：点击对话框，会弹出输入用户名和密码的窗口，如下图：

说明：只要在此对话框中输入正确的用户名和密码，即可通过 IEEE 802.1x 认证。

交换机故障恢复管理

交换机密码恢复

通常情况下，可以为交换机设置 enable 密码来提供安全，在没有 enable 密码的情况下，无法对交换机修改任何配置，因此，在忘记 enable 密码的时候，就意味着无法改动交换机信息。但是，如果能够物理上接触到交换机，便可以通过某些方法来清除已经配置的 enable 密码，甚至是交换机的所有配置。

Cisco 交换机有个叫做 nvram:的存储器，类似于 PC 机的硬盘，断电后数据不会丢失，交换机的配置都会存放在 nvram:里面，配置文件名为 startup-config，只要

配置保存过，nvram:里面都会出现 startup-config 这个文件，每当交换机启动时，都会读取 nvram:里面的 startup-config 文件，如果有密码，我们就无法进入 enable 模式，在这里需要提示，在 cisco 低型号的交换机中，如 3750 以下（包括 3750），nvram:里面的 startup-config 文件会同时在 flash:存储器里再生成一份名为 config.text 的文件，要注意的是这两个文件属同一文件，删除任何一个，两个同时丢失;在高型号的交换机中，如 4500 系列以上（包括 45 系列），只有 nvram:，并没有名为 flash:的这个存储器，所以就没有 config.text 这个文件，与 flash:存储器相对应的存储器叫做 bootflash:，在这里面除了 IOS 文件，没有别的。

所以我们在交换机有 enable 密码的情况下，要清除密码，只需要让交换机不读取 nvram:中的 startup-config 即可，由于上述原因，需要分两种情况来进行操作。

操作步骤：

第一种情况（3750 以下系列）

（1）通常在交换机前面板（正视有端口的一面）左侧，有一个名为“mode”的按钮，用手按住此按钮不放，然后拔掉交换机电源线，过 5 秒钟再次插回该电源线，如果是 3550，这时会看到交换机所有端口都会亮，等到交换机第一个端口熄灭后，则把按按钮的手放开；如果是 3560，则端口灯不会亮，但左侧面板的灯会闪烁，等到第一个灯变为绿色不闪时，放开按按钮的手，（其它型号交换机类同），这时会看到登陆交换机的界面如下

switch:

（2）接下来我们要改掉 nvram:里面的 startup-config 文件，让交换机找不到此配置文件，便认为没有密码，但我们只须改 flash:下的 config.text 即可。

switch: flash_init （初始化 flash 文件系统）

switch: load_helper （装载并初始化辅助映像 ROM 中的最小 IOS 映像）

switch: dir flash: （显示 flash 文件内的文件和目录）

switch: rename flash:config.text flash:config.old （修改配置文件名，以便跳过密码）

switch: boot （重启）

（3）这时我们不需要密码便可顺利进入交换机 enable 模式，但此时交换机并未装载任何配置，此时我们在 enable 模式便可把配置文件名改回 config.text：

Swith#rename flash:config.old flash:config.text

然后再将配置导入内存正常运行

Switch#copy startup-config running-config

最后再改掉之前忘记的 enable 密码即可，并且之前的配置也没有丢失。

Switch(config)#no enable secret

第二种情况（4500 以上系列）

(1)将交换机的电源直接拔掉，并反复按 ctrl C 两个组合键，直到出现

rommon 1 >

模式为止。

(2)输入 confreg:

rommon 1 > confreg

(3)照以下情况选择

Configuration Summary :

=> load ROM after netboot fails

=> console baud: 9600

=> autoboot from: commands specified in 'BOOT' environment variable

do you wish to change the configuration? y/n [n]: y （重要）

enable "diagnostic mode"? y/n [n]: n

enable "use net in IP bcast address"? y/n [n]: n

disable "load ROM after netboot fails"? y/n [n]: n

enable "use all zero broadcast"? y/n [n]: n

enable "break/abort has effect"? y/n [n]: n

enable "ignore system config info"? y/n [n]: y （重要）

change console baud rate? y/n [n]: n

change the boot characteristics? y/n [n]: n

Configuration Summary :

=> load ROM after netboot fails

=> ignore system config info

=> console baud: 9600

=> autoboot from: commands specified in 'BOOT' environment variable

do you wish to save this configuration? y/n [n]: y （重要）

You must reset or power cycle for new configuration to take effect

rommon 2 > reset （最后重启）

(4)此时可不需要密码便能进入交换机 enable 模式，但交换机此时也是按默认的引导方式从 CF 卡中启动系统，如果没有 CF 卡，那么再次重启交换机，将会再进

入 rommon 1>这个模式，所以必须现在将交换机引导模式改为 bootflash:中的 IOS 启动，

如果 bootflash:中只有一个 IOS，则输入：

Switch(config)#boot system bootflash:

如果 bootflash:中有两个文件名分别为 a.bin 和 b.bin 的 IOS，如想从 a.bin 引导，则输入：

Switch(config)#boot system bootflash:a.bin

(5)最后保存所有改动信息：

Switch#write

交换机密码恢复管理

即使是一个规模较大的公司，可能需要的路由器设备是少数的，可以将路由器放在机房的机柜里，锁上机房门和机柜门，一般人是碰不到路由器设备的。但是如果由于公司人数众多，或者由于场地的原因，或者再由于别的原因，如双绞线的有效传输范围为 100 米，也就表示主机离交换机的距离不能超过 100 米，在这些情况下，可能需要将交换机放置在离员工近距离的地方，有时选择直接放在员工办公的 Office 房间里。因为我们知道，当交换机放置在一个能让人物理接触到的位置时，就可以通过绕过密码的方式对交换机进行配置修改，比如网络管理员在交换机上对网络做过某些访问限制，或者 QOS 带宽限制，那么就有人会尝试着接触交换机修改其中的配置，这对于网络管理员来说，是不容易发现的。

因为无论是路由器还是交换机，都可以通过物理接触来绕过密码，从而修改配置信息，而由于交换机的特殊性，所以交换机系统集成了一个特殊的功能，就是可以开启或关闭对交换机密码恢复功能，如果此功能打开，则可以通过物理接触来绕过密码，如果关闭，则交换机的全部配置信息会自动全部删除，在因为交换机被别人误动而造成配置全部丢失，作为网络管理员，是有足够的理由发现的。

配置管理密码恢复

1.查看当前的密码恢复功能

（1）查看默认的交换机密码恢复功能

Switch#sh version

Cisco IOS Software, C3560 Software (C3560-ADVIPSERVICESK9-M), Version 12.2(35)SE1, RELEASE SOFTWARE (fc1)

Copyright (c) 1986-2006 by Cisco Systems, Inc.

Compiled Tue 19-Dec-06 10:54 by antonino

Image text-base: 0x00003000, data-base: 0x01362CA0

（输出被省略）

cisco WS-C3560-24TS (PowerPC405) processor (revision D0) with 122880K/8184K bytes of memory.

Processor board ID CAT1047RJNU

Last reset from power-on

1 Virtual Ethernet interface

24 FastEthernet interfaces

2 Gigabit Ethernet interfaces

The password-recovery mechanism is disabled.

（输出被省略）

Switch#

说明：可以看到，3560 默认的密码恢复功能是关闭的，此特性会因为交换机型号的不同，默认会有所不同，如 3550 是默认打开的。

2.测试关闭了密码恢复功能的效果

（1）为交换机配置 enable 密码

Switch(config)#enable secret cisco

（2）保存配置

Switch#wr

Building configuration...

[OK]

Switch#

（3）通过物理接触来做密码恢复

说明：密码恢复方法，请参见上一节

在做密码恢复时，交换机会出现以下提示：

Switch#

Base ethernet MAC Address: 00:1a:6c:6f:fb:00

Xmodem file system is available.

The password-recovery mechanism is disabled.

Initializing Flash...

flashfs[0]: 30 files, 3 directories

flashfs[0]: 0 orphaned files, 0 orphaned directories

flashfs[0]: Total bytes: 32514048

flashfs[0]: Bytes used: 11811840

flashfs[0]: Bytes available: 20702208

flashfs[0]: flashfs fsck took 13 seconds.

...done Initializing Flash.

Boot Sector Filesystem (bs) installed, fsid: 3

done.

The password-recovery mechanism has been triggered, but

is currently disabled. Access to the boot loader prompt

through the password-recovery mechanism is disallowed at

this point. However, if you agree to let the system be

reset back to the default system configuration, access

to the boot loader prompt can still be allowed.

Would you like to reset the system back to the default configuration (y/n)?

说明：提示信息说明交换机的密码恢复功能已被关闭，并给出提问，如果回答 n，则等于什么都没有做，交换机将做正常启动，如下：

Would you like to reset the system back to the default configuration (y/n)?n

Boot process continuing...

Loading "flash:c3560-advipservicesk9-mz.122-35.SE1.bin"...@@@@@@@@

（4）回答 y

说明：如果上一问中回答 y，则像上述所说一样，配置将被全部清除，以下是回答 y 后，进行配置密码恢复的结果：

Would you like to reset the system back to the default configuration (y/n)?y

The system has been interrupted, and the config file

has been deleted. The following command will finish

loading the operating system software:

boot

switch:

switch: flash_init

Initializing Flash...

...The flash is already initialized.

Setting console baud rate to 9600...

switch: load_helper

switch: dir flash:

Directory of flash:/

2 -rwx 8450865 <date> c3560-advipservicesk9-mz.122-35.SE1.bin

3 drwx 832 <date> crashinfo_ext

26 -rwx 24 <date> private-config.text

7 drwx 832 <date> crashinfo

20706304 bytes available (11807744 bytes used)

switch:

说明：可以看到，回答 y，交换机已经自动删除配置，而进入之后会发现 flash:中根本就已经没有了 config.text 这个文件。被人误清除了配置的交换机，管理员是应该有责任发现的。