



雏鹰部落

中国·网络工程师集散地

BGP

目录

概述.....	2
BGP AS	3
BGP 邻居.....	4
BGP 更新源	6
BGP TTL.....	8
BGP AS_Path.....	8
BGP 路由表	11
BGP Synchronization	12
Path Attributes	14
BGP RIB-Failure	21
BGP最优路径选择.....	22
BGP基础实验.....	26
BGP 路由聚合	82
BGP 默认路由	102
BGP 路由过滤.....	104
BGP条件路由.....	111
BGP Peer Group.....	117
BGP Community	128
BGP Reflector(BGP反射器)	156
BGP Confederation(BGP联邦)	176
BGP 后门路由.....	195
BGP Dampening.....	205
BGP重分布进IGP.....	218

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。

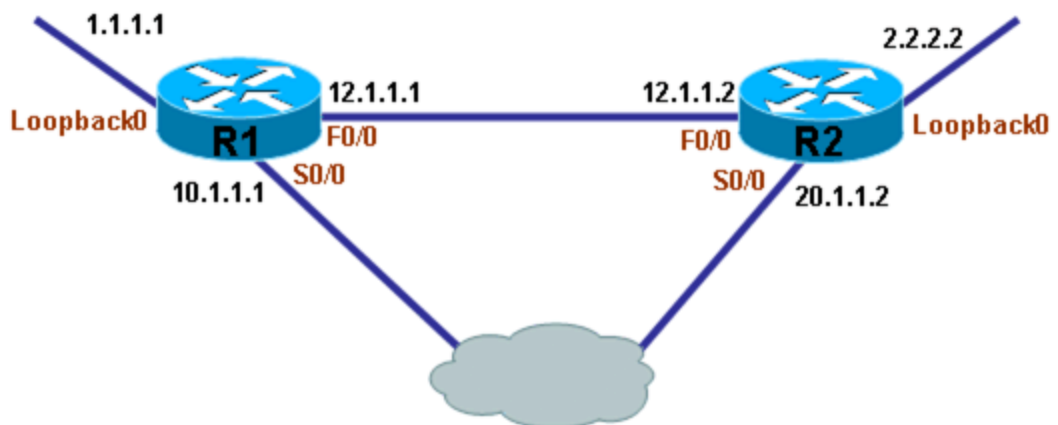
Established— 从对端 peer 收到了 keepalive，并开始交换数据，收到 keepalive 后，hold timer 都会被重置，如果收到 notification，就回到 idle 状态。

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 IP 12.1.1.2 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。

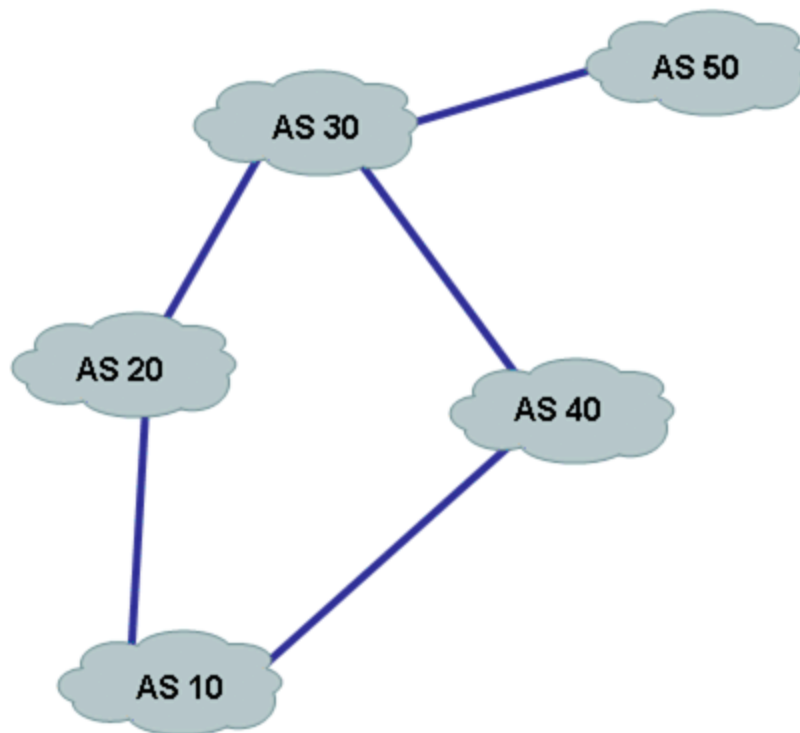
BGP TTL

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：

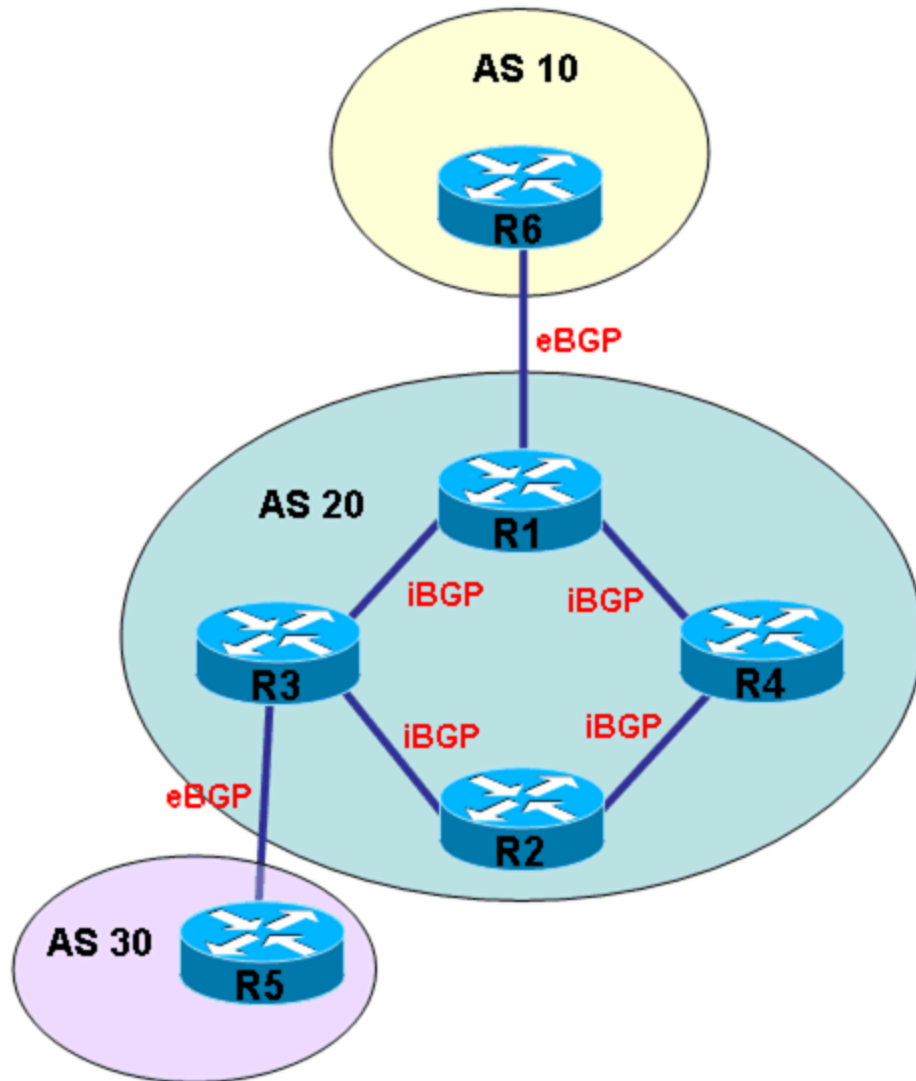


在上图中，当路由穿越各个 AS 时，所有发给 eBGP 邻居的路由，都会在 AS-path 中添加自己的 AS，自己的 AS 总是添加在 AS-path 的最前面。例如一条路由从 AS 10 被发往 AS 20，则 AS-path 为“10”，当 AS 20 将路由发往 AS 30 时，添加上自己的 AS 号码 20 之后，AS-path 变成“20, 10”，当 AS 30 将路由发往 AS 50 时，最终 AS 50 收到的路由的 AS-path 为“30, 20, 10”。当 AS 30 将路由发给 AS 40，AS 40 再将路由发给 AS 10 时，路由的 AS-path 为“40, 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。

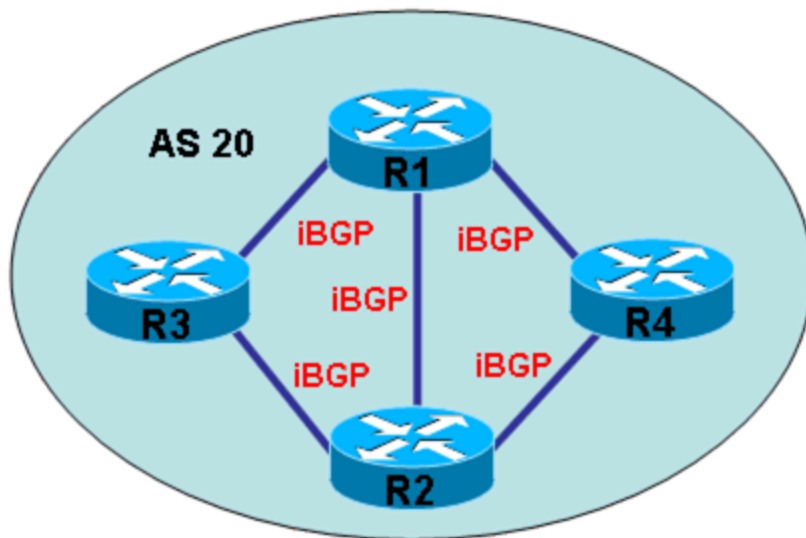
在 IGP 协议中，我们将路由协议分成两大类：距离矢量路由协议和链路状态路由协议，而 BGP 被划分为路径矢量路由协议（path-vector routing），路径矢量算法结合了距离矢量特性与 AS-path 防环特性。

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表,一张是存放普通路由的路由表,被称为 IGP 路由表,就平时我们使用命令 `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。

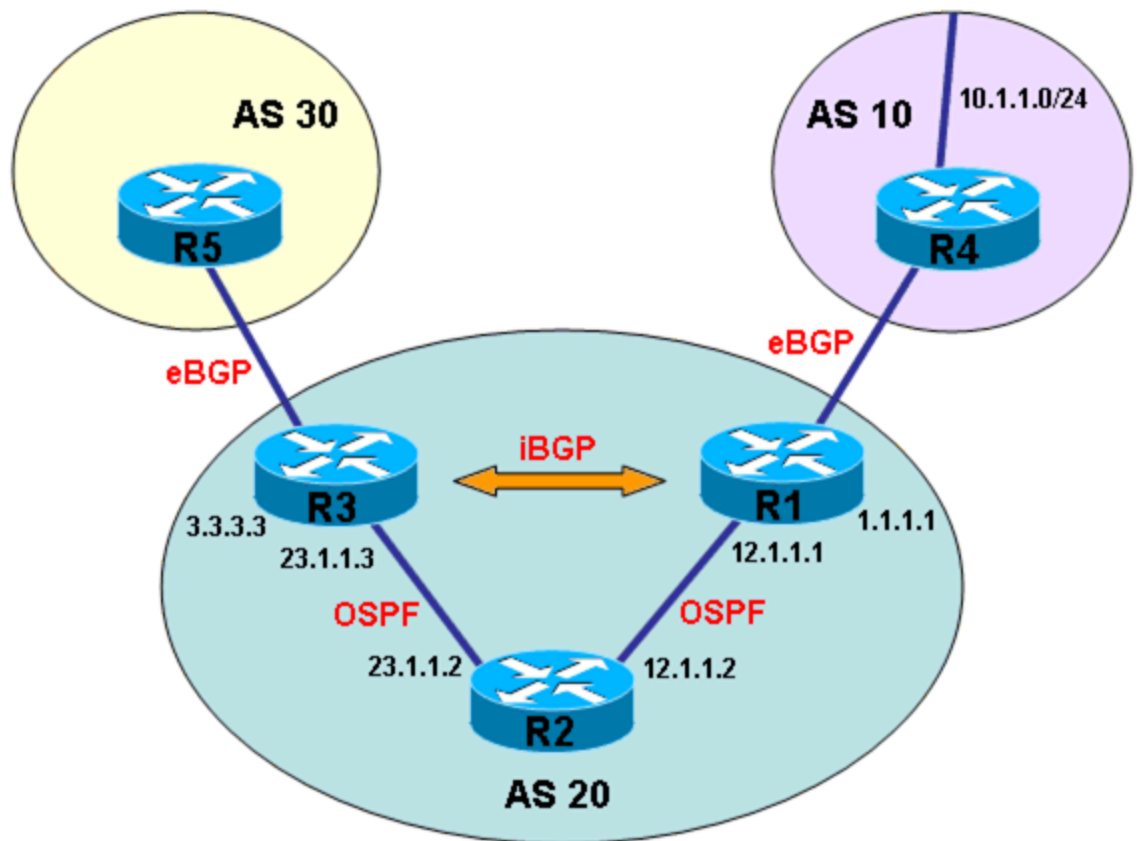
因为 BGP 的邻居类型分为两种：eBGP 和 iBGP，所以 BGP 路由的 AD 值也有区分，如果 BGP 的路由是从 eBGP 学习到的，AD 值为 20，可以发现，从 eBGP 邻居学习到的路由，将优于任何 IGP 协议；从 iBGP 学习到的路由的 AD 值为 200，同样可以发现，此类路由的优先级低于任何 IGP 协议。BGP 除了以上两种 AD 值之外，如果 BGP 路由是从本地手工导入的，即 BGP 本地路由，则 BGP 本地路由的 AD 值为 200，与 iBGP 路由的 AD 值相同，优先级低于任何 IGP 协议。

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 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。

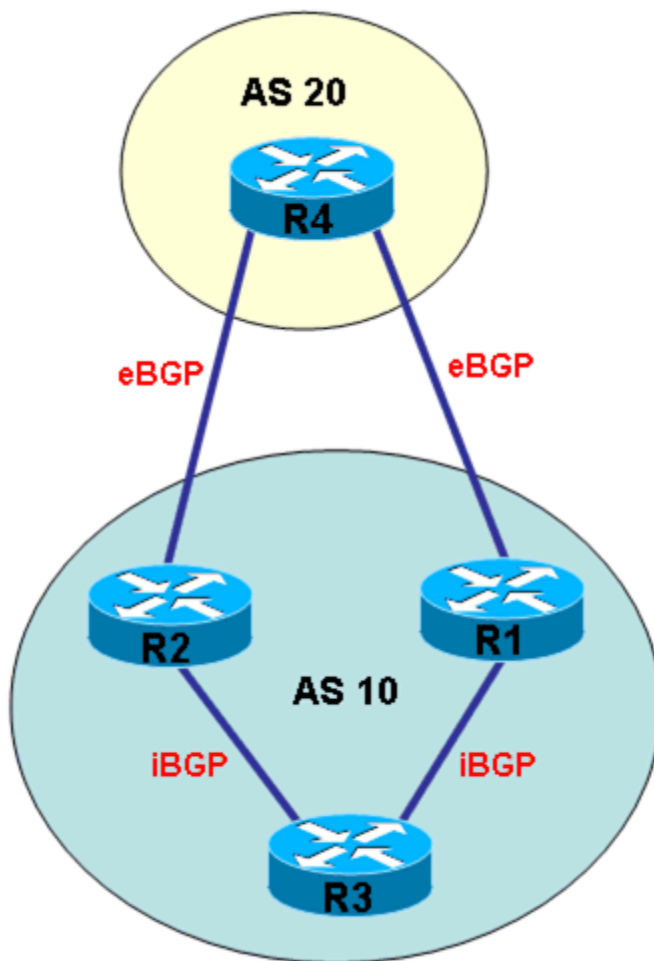
Local_Pref 属性在 BGP 邻居之间是自动传递的，只有在将路由发给 iBGP 时才会传递，而在发给 eBGP 时，是没有 Local_Pref 值的，一条路由的 Local_Pref 属性

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。

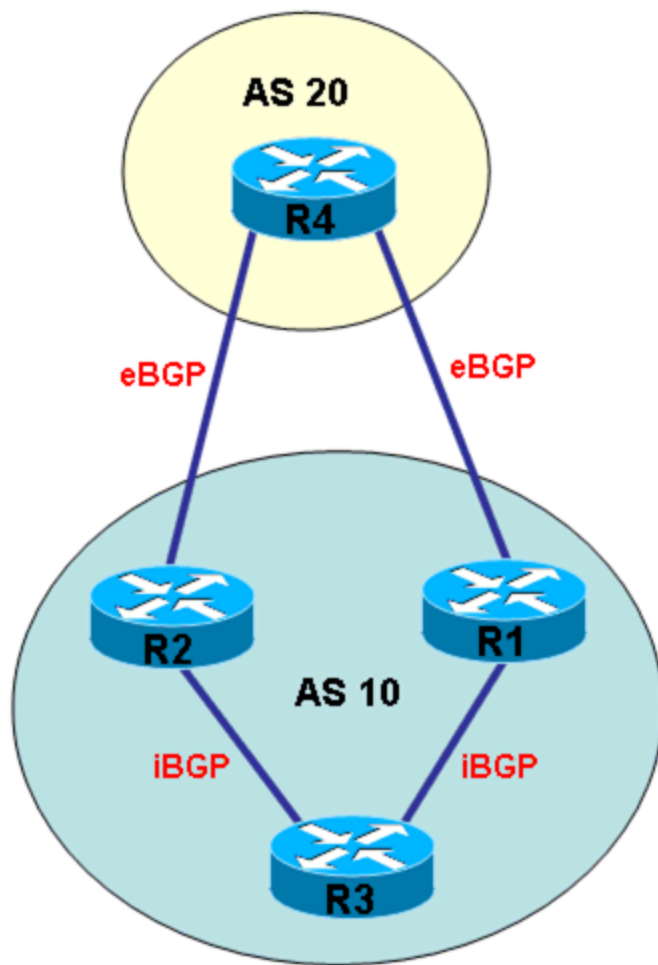
因为 R1 和 R2 在将路由发给 iBGP 邻居 R3 时会携带 Local_Pref 属性，所以 R3 同时比较 iBGP 邻居 R1 与 iBGP 邻居 R2 时，才合适使用 Local_Pref 属性，因为下一跳都是 iBGP 邻居，如果下一跳不都是 iBGP 邻居，并不建议修改 Local_Pref 属性来影响选路。

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。）

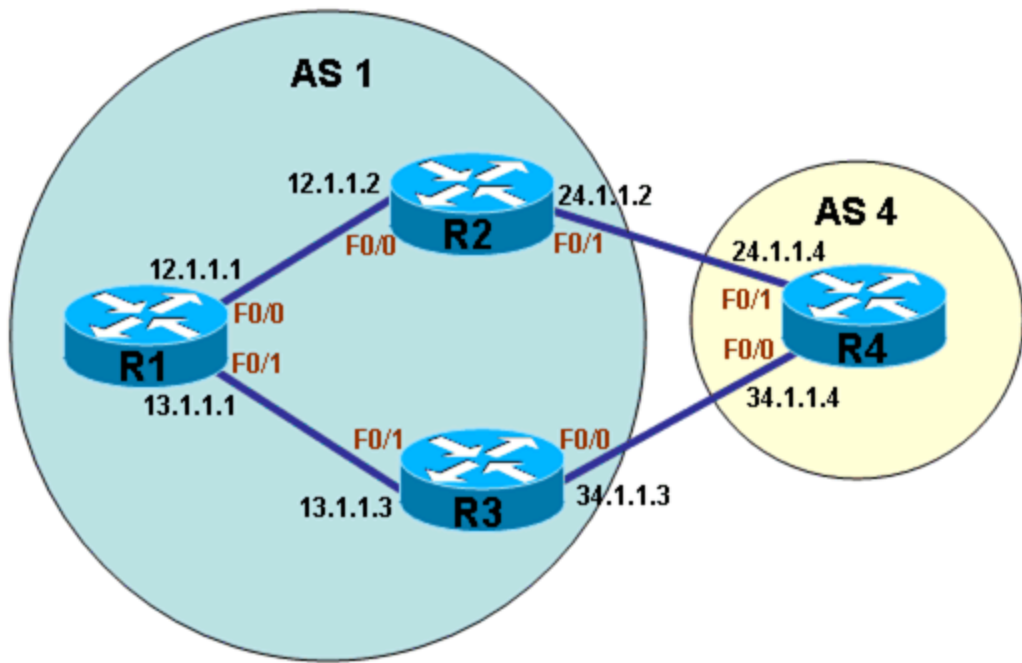
12. 最短 cluster list（如同 AS_PATH）

[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 Up/Down	V State/PfxRcd	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ
1.1.1.1 00:08:45	4 0	1	12	12	1	0	0
4.4.4.4 never	4 Idle	4	0	0	0	0	0

```
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

```
l#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
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 Up/Down	V State/PfxRcd	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ
1.1.1.1 00:02:08	4 1	1	7	7	4	0	0
4.4.4.4 00:00:08	4 2	4	7	5	4	0	0

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,
```

```
o 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 Up/Down	V State/PfxRcd	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ
2.2.2.2 00:00:27	4 2	1	57	55	10	0	0
3.3.3.3 00:03:45	4 1	1	28	36	10	0	0

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

```
rl#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
```

```
0       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
```

```
0       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
```

```
0       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

0 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

0 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

0 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

rl#

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

0 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
- 0 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
- 0 100.1.1.0 [110/3] via 12.1.1.2, 00:01:10, FastEthernet0/0
- 3.0.0.0/32 is subnetted, 1 subnets
- 0 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
- 0 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
- 0 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 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,
```

```
o 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

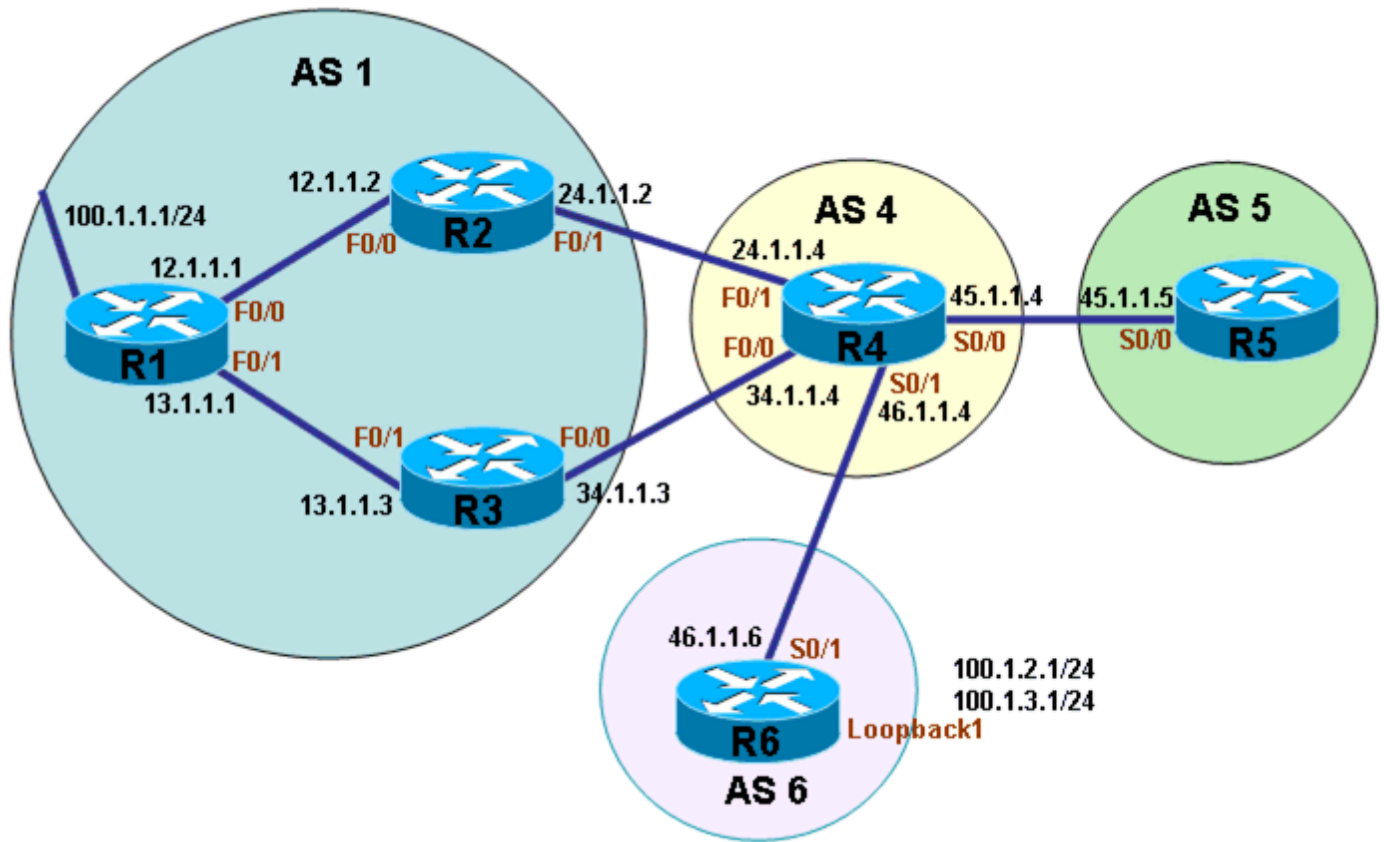
atomic-aggregate 属性。汇总路由是否使用 AS-SET，可以自由决定。因为 AS-SET 中可能包含多个 AS，但即使一个 AS-SET 中有多个 AS，但在计算 AS_Path 长度时，只被计算为 1 个 AS。

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
*> 44.4.4.0/24	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
```

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

```
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 nonexistent 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#			

*> 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

```
*> 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。

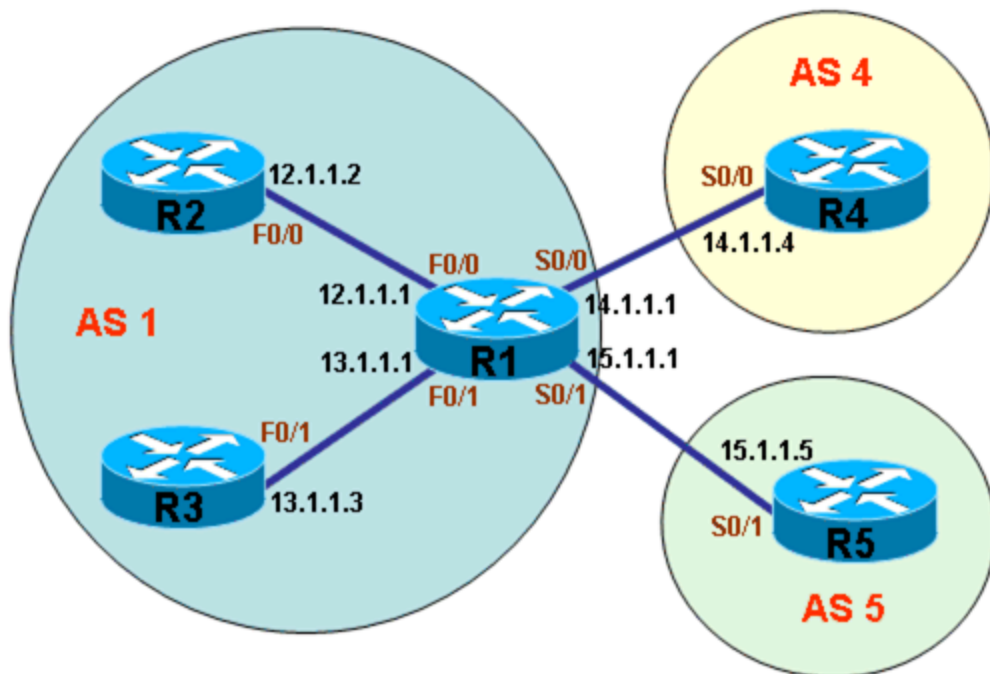
在使用普通方式配置 BGP 邻居时，假如配置一个特定的邻居需要 4 条命令，那么配置 10 个邻居就需要 40 条命令，在使用 Peer Group 时，创建 Peer Group 使用 1 条命令，再使用 4 条命令为 Peer Group 配置参数，最后再使用 10 条命令将 10 个邻居全部划入 Peer Group，可以看出，使用 Peer Group 配置 10 个邻居所使用的 15 条命令，远远少于使用普通方式的 40 条命令，从而体现出使用 Peer Group 对配置 BGP 工作量的简化是相当明显的。

除了可以对 Peer Group 配置各种参数外，各种可以为邻居配置的性质和策略，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 Up/Down	V State/PfxRcd	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ
2.2.2.2 00:03:01	4 1	1	8	9	5	0	0
3.3.3.3 00:02:18	4 1	1	7	8	5	0	0
4.4.4.4 00:01:15	4 1	4	6	7	4	0	0
5.5.5.5 00:00:28	4 1	5	5	6	4	0	0

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

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一同 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)

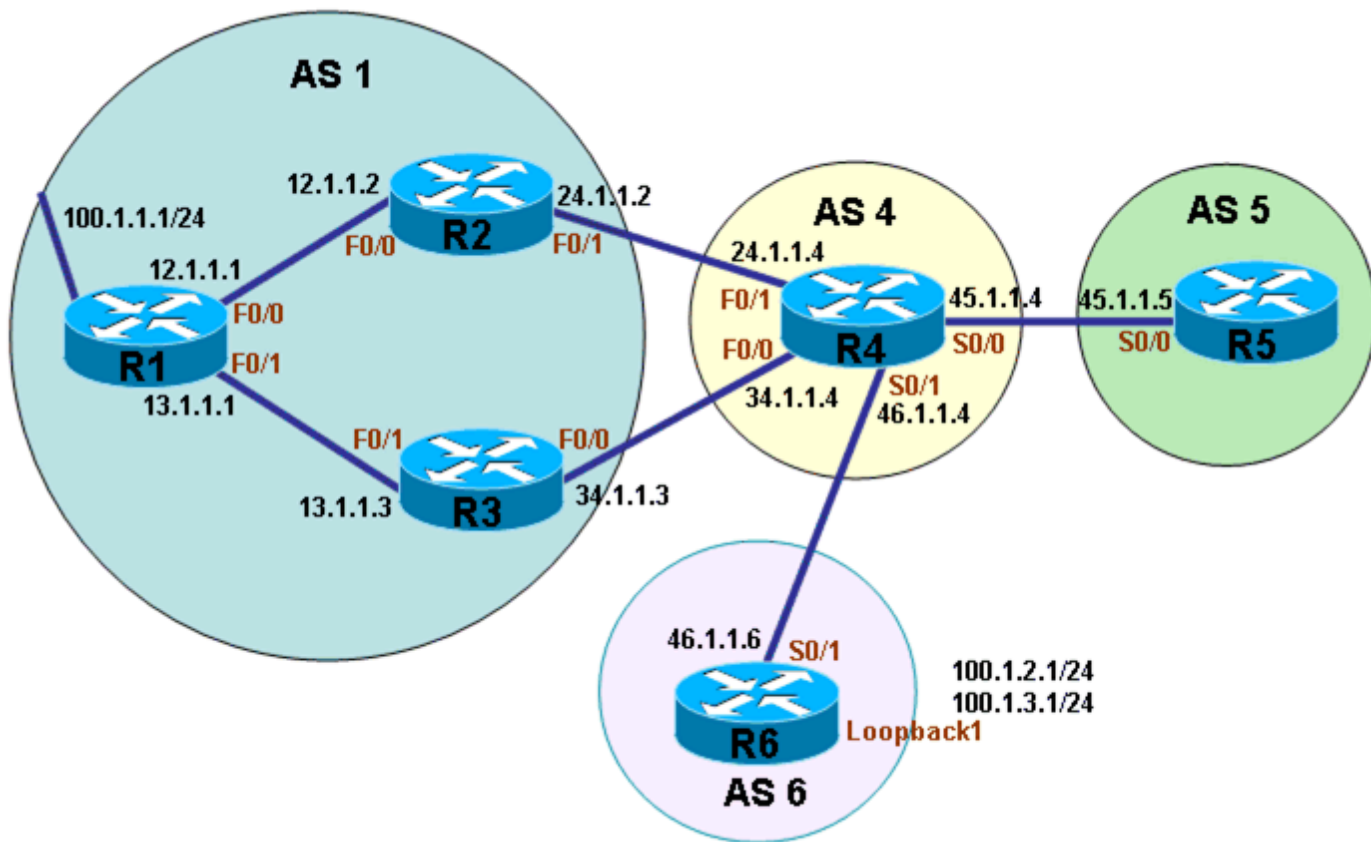
扩展 Community 为 MPLS 中的 VRF 路由传递所定义的，在 MPLS 中有所解释。

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

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#			

*> 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)#match 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
*>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
*> 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
*>i100.1.1.0/24	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

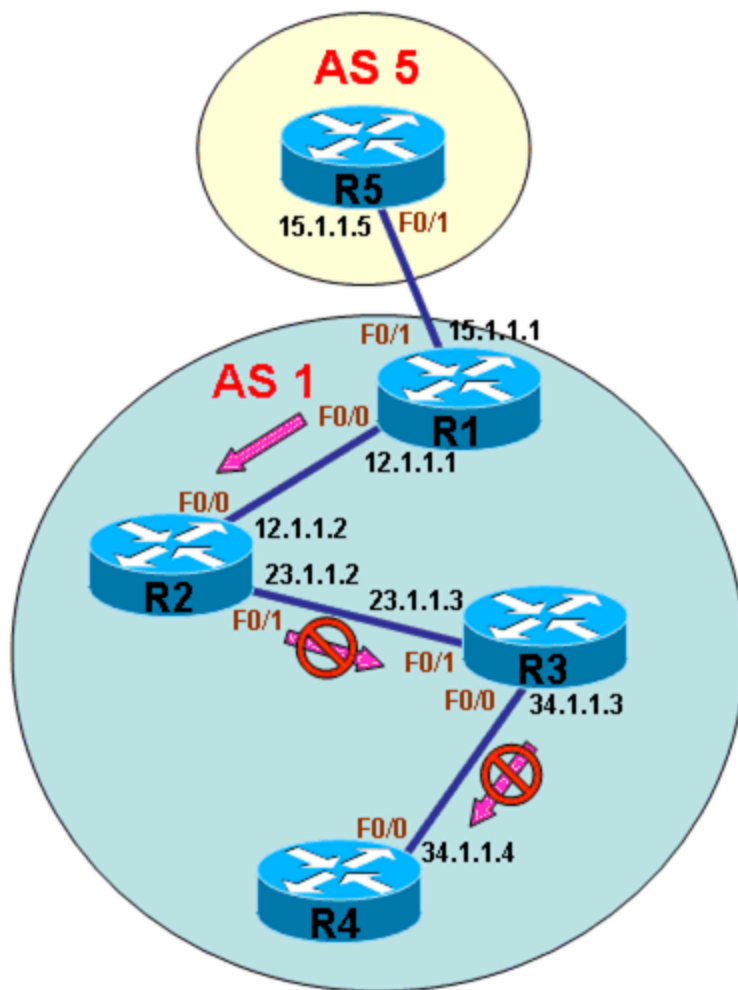
```
*> 55.5.5.0/24      4.4.4.4              0 4 5 i
*> 66.6.6.0/24      0.0.0.0              0      32768 i
*> 100.1.1.0/24     4.4.4.4              0 4 1 i
*> 100.1.2.0/24     0.0.0.0              0      32768 i
*> 100.1.3.0/24     0.0.0.0              0      32768 i

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。

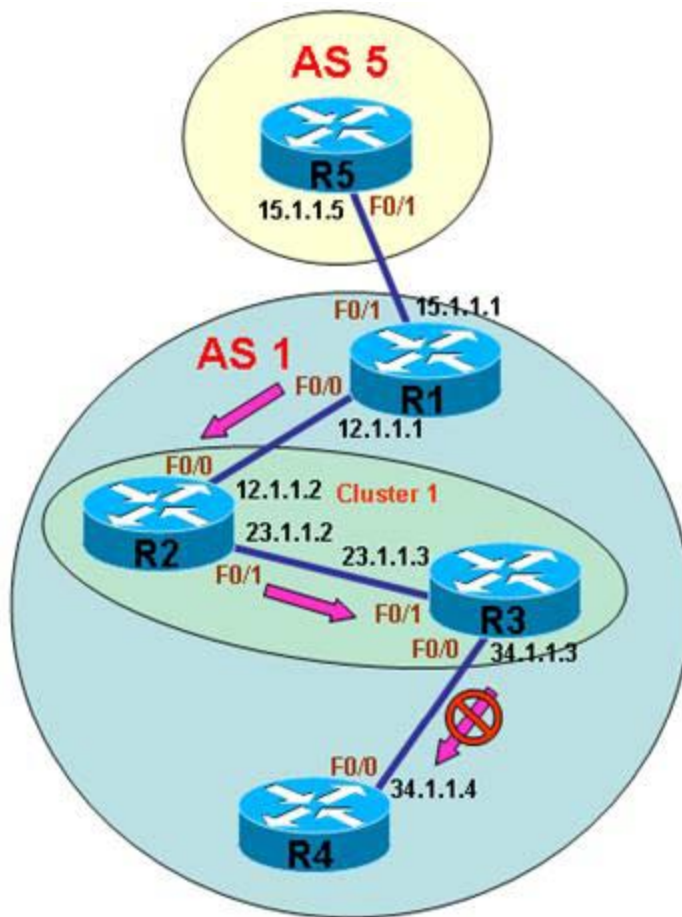
BGP Reflector 可以将自己的任何 BGP 路由反射给自己的 client，从而可以突破 iBGP 路由传递的限制，具体规则为：

★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。

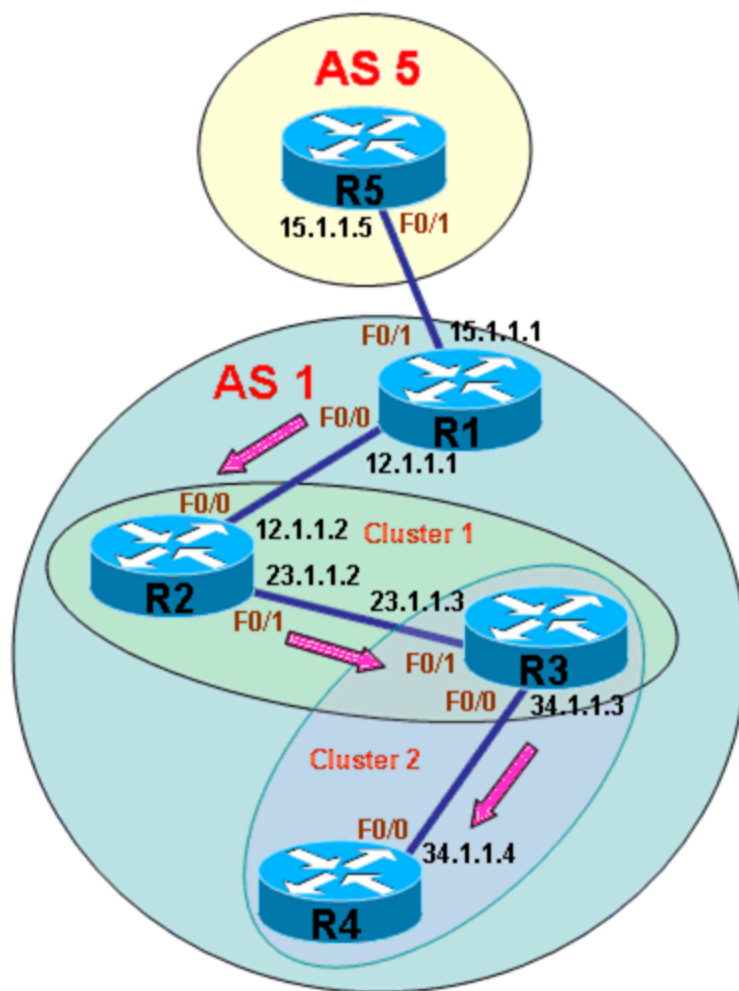
★从 client 学习到的路由将发送给所有 client 和所有非 client，也就是发给所有邻居。

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：



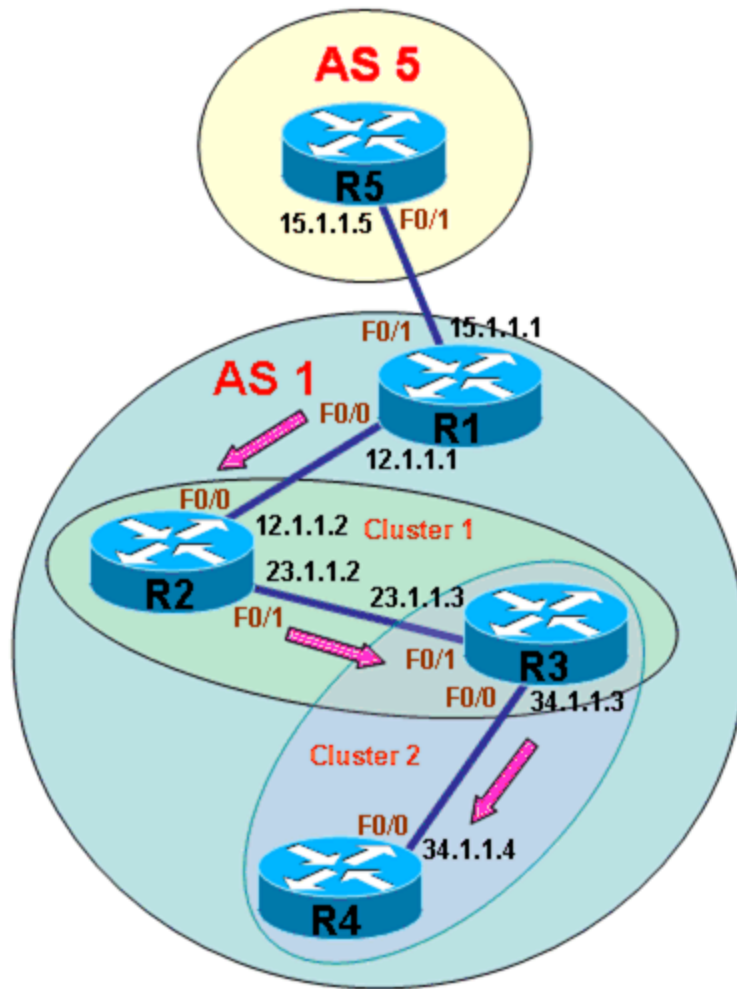
在 Cluster 1 中, R2 为 Reflector, R3 为 client, 因为 R3 并不知道自己是 client, 所以又将 R3 配置成了 Cluster 2 中的 Reflector, 并将 R4 配置成为 Cluster 2 中的 client。最终因为 Reflector R2 收到了 R1 的路由, 将所有路由发给 client R3, 又因为 R3 也是 Reflector, 再将路由发给 client R4, 并将所有从 R4 的路由也发送给 R2, 最后所有路由器都拥有全网的路由。

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, 所以当在一个 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

(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
*>i44.4.4.0/24	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：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
*> 44.4.4.0/24	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#

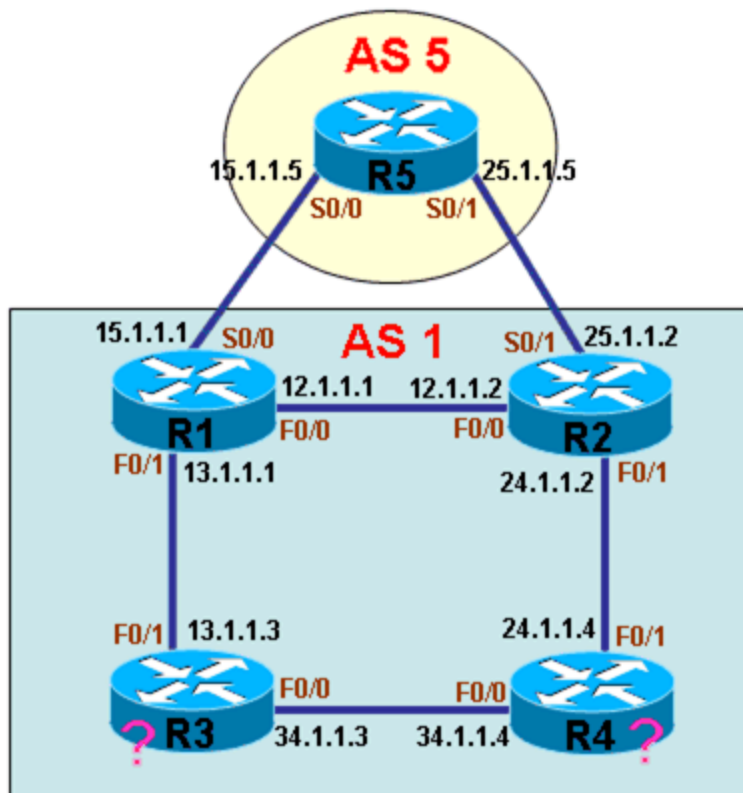
说明：R3 已经将 cluster ID 改为 0.0.0.123。

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)。

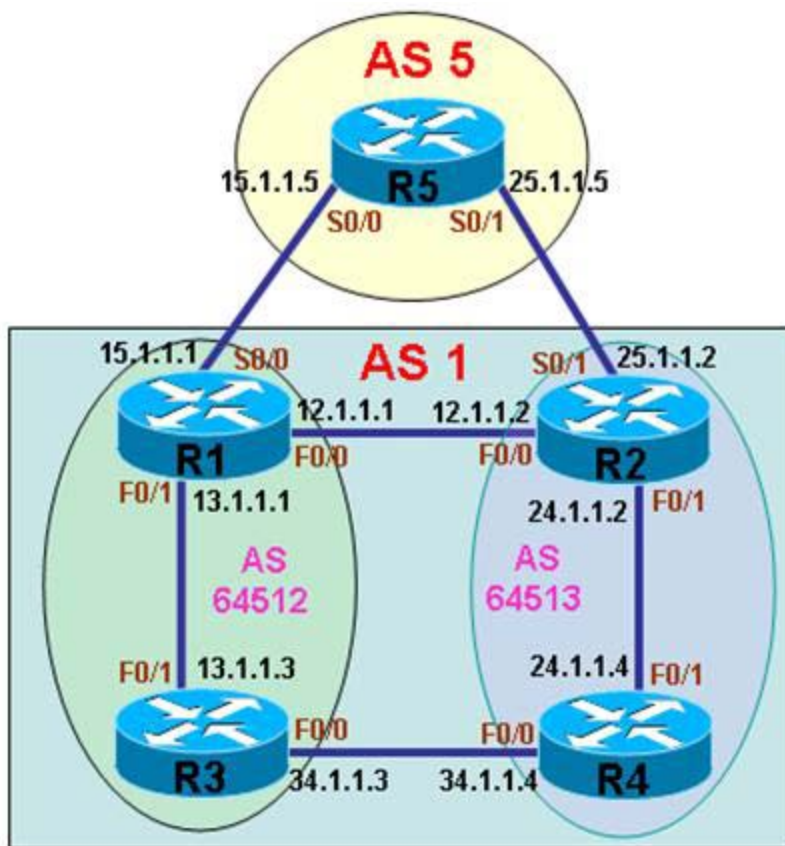
因为只有从 iBGP 邻居收到的路由才不能转发给其它 iBGP 邻居，而从 eBGP 邻居收到的路由可以转发给任何邻居，包括 iBGP 邻居，所以在拥有多个路由器的大型 AS 中，BGP Confederation 采用在 AS 内部建立多个子 AS 的方法，从而将一个大的 AS 分割成多个小型 AS，让 AS 内部拥有足够数量的 eBGP 邻居关系来解决路由限制问题。

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。

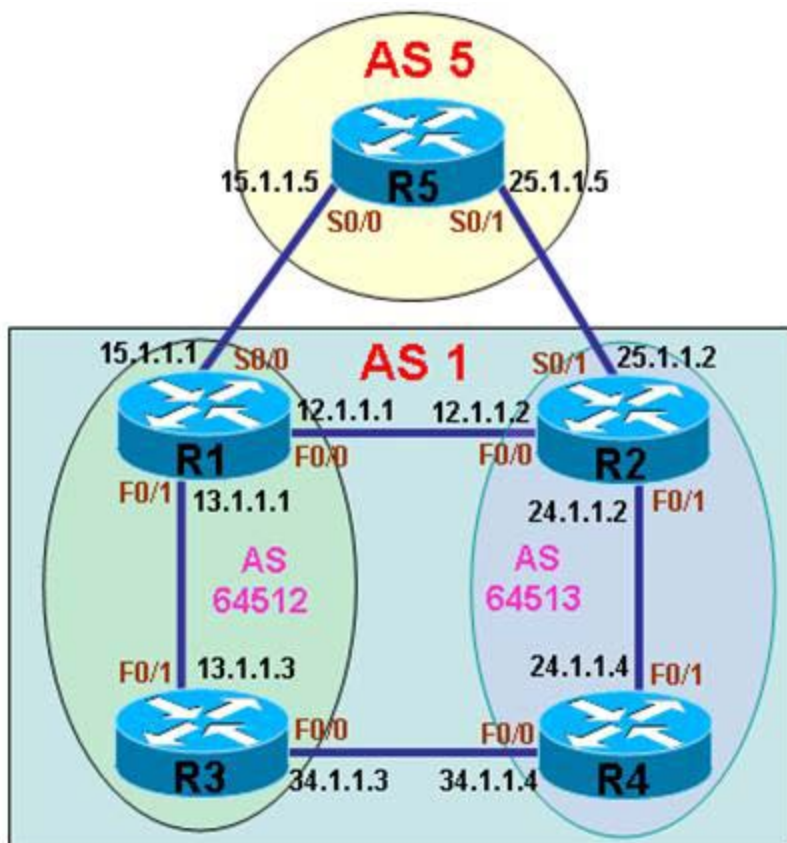
在使用 BGP Confederation 在 AS 内部创建子 AS 时，建议使用私有 AS 号码，范围是 64512~65534，所有 BGP Confederation 内部的子 AS，对于外界都是不可见的，如上图，R1 与 R2 在 AS 1 中分别为 AS 64512 和 AS 64513，但是对于 R5 来说，R1 和 R2 都为 AS 1 的，而 AS 64512 和 AS 64513 对于 R5 来说是透明的，外界并不知道 AS 内部是否创建了 BGP Confederation，对于子 AS 的号码只在 AS 内部传递路由时才会添加到 AS_Path 中去，在出 AS 时，这些子 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 Up/Down	V State/PfxRcd	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ
1.1.1.1 00:17:30	4 4	1	25	26	6	0	0
2.2.2.2 00:07:08	4 4	1	15	16	6	0	0

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 Up/Down	V State/PfxRcd	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ
2.2.2.2 00:06:27	4 3	64513	13	13	6	0	0
3.3.3.3 00:03:14	4 1	64512	8	11	6	0	0
5.5.5.5 00:17:51	4 1	5	26	25	6	0	0

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,
```

```
o 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

```

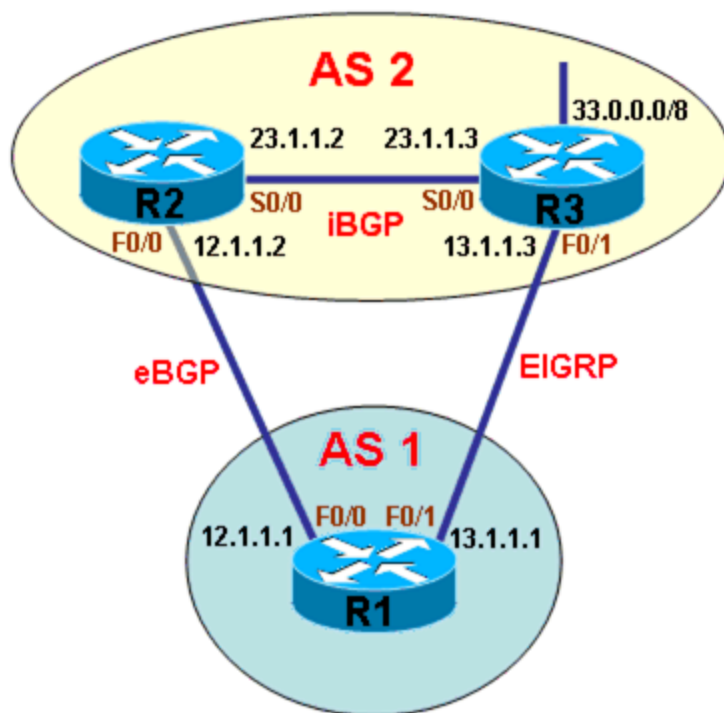
    *>i          1. 1. 1. 1          0    100      0 (64513)
i
    *> 33. 3. 3. 0/24    0. 0. 0. 0          0          32768 i
    *  44. 4. 4. 0/24    4. 4. 4. 4          0    100      0 (64513)
i
    *>i          1. 1. 1. 1          0    100      0 (64513)
i
    *  55. 5. 5. 0/24    4. 4. 4. 4          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



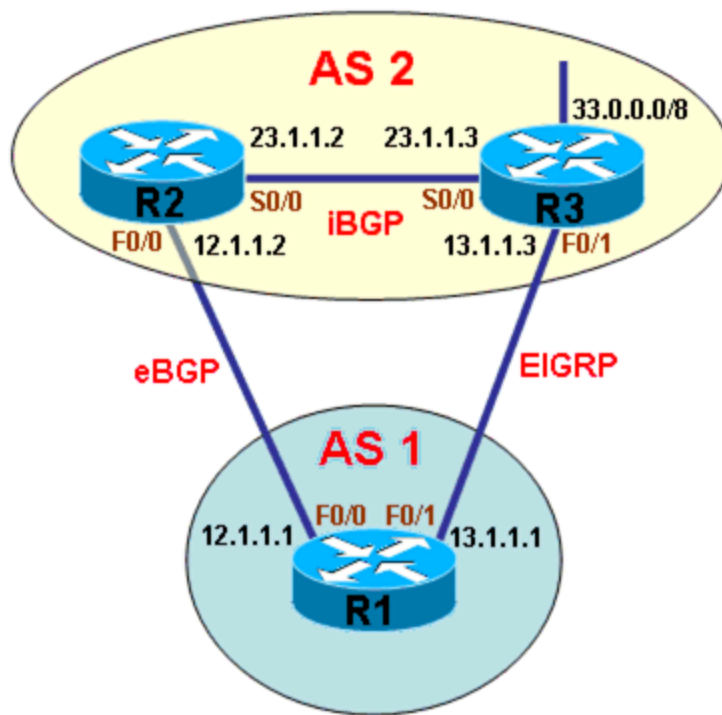
如上图网络环境中，AS 1 与 AS 2 存在多种业务流量，AS 1 中的路由器 R1 与 AS 2 中的路由器 R2 建立 eBGP 邻居关系，当 R2 将 AS 2 内的路由传递给 R1 之后，便可以实现 AS 1 与 AS 2 的互通，R1 将所有去往 AS 2 的流量全部交给 R2。

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

rl#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的减少，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。

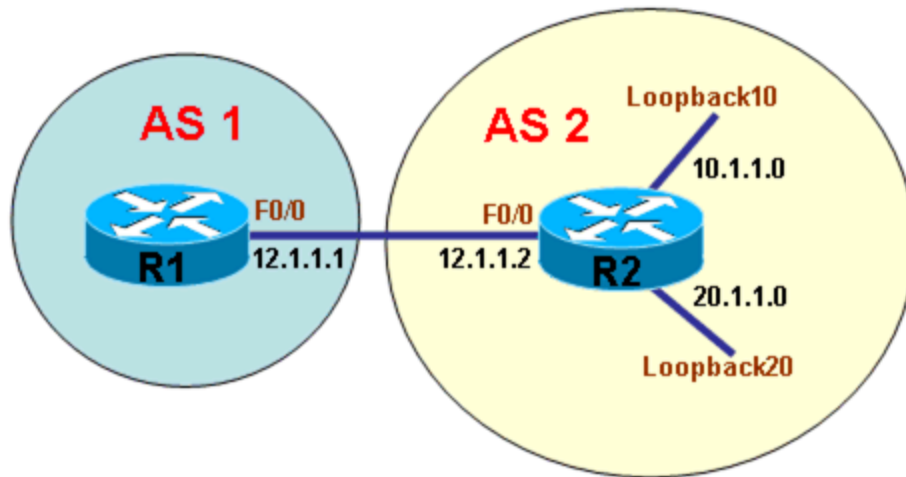
注：

★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，但不能针对特定邻居配置。

★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

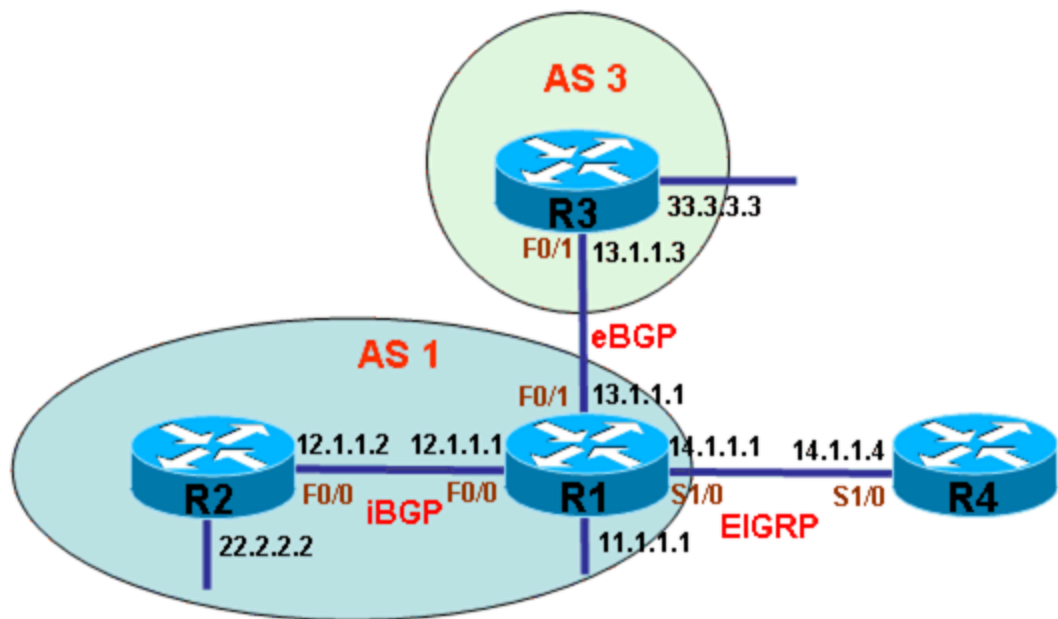
r1#

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，很可能会导致 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   RT0   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。