



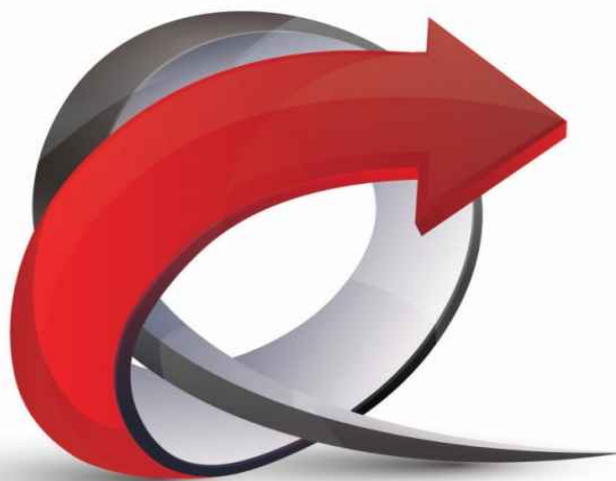
> 华为ICT认证系列丛书

华为技术有限公司 联合创作
武汉誉天互联科技有限责任公司

华为技术认证

HCNP路由交换 实验指南

华为技术有限公司 主编



 人民邮电出版社
POSTS & TELECOM PRESS

目 录

[封面](#)

[扉页](#)

[版权](#)

[序](#)

[前言](#)

[华为认证简介](#)

[华为路由交换产品介绍](#)

[eNSP介绍](#)

[使用说明](#)

[第1章 路由基础](#)

[1.1 访问控制列表](#)

[1.2 基本的路由策略配置](#)

[1.3 控制RIP 路由的发布及路由引入](#)

[第2章 OSPF](#)

[2.1 OSPF 基本配置](#)

[2.2 OSPF 邻居邻接关系](#)

[2.3 OSPF 链路状态数据库](#)

[2.4 OSPF Stub 区域](#)

[2.5 OSPF NSSA 区域](#)

[2.6 OSPF 虚链路](#)

[2.7 OSPF 网络类型](#)

[2.8 OSPF 路由聚合](#)

[2.9 OSPF 监测和调试](#)

[2.10 OSPF 缺省路由](#)

[2.11 OSPF 故障排除](#)

[第3章 BGP](#)

[3.1 BGP 邻居](#)

[3.2 BGP 认证功能](#)

[3.3 BGP 自动路由聚合](#)

[3.4 BGP 手动路由聚合](#)

[3.5 BGP 路径选择——Preferred Value](#)

[3.6 BGP 路径选择——Local Preference](#)

[3.7 BGP 路径选择——Next Hop](#)

[3.8 BGP 路径选择——AS Path](#)

[3.9 BGP 路径选择——MED](#)

[3.10 BGP 路径选择——Community](#)

[3.11 BGP 路由反射器](#)

[3.12 BGP 路由黑洞](#)

[3.13 BGP 联盟](#)

[3.14 BGP 路由过滤](#)

[3.15 BGP 路由引入](#)

[3.16 BGP 缺省路由](#)

[3.17 BGP 路由衰减](#)

[3.18 BGP 监测和调试](#)

[3.19 BGP 故障排除](#)

[第4章 IS-IS](#)

[4.1 IS-IS 基本配置](#)

[4.2 IS-IS 邻接关系](#)

[4.3 IS-IS 链路状态数据库](#)

[4.4 IS-IS DIS](#)

[4.5 IS-IS 开销值和协议优先级](#)

[4.6 IS-IS 路由聚合](#)

[4.7 IS-IS 缺省路由](#)

[4.8 IS-IS 路由引入](#)

[4.9 IS-IS 路由过滤](#)

[4.10 IS-IS 路由渗透](#)

[4.11 IS-IS 监测和调试](#)

[4.12 IS-IS 故障排除](#)

[第5章 IP组播](#)

[5.1 IP 组播的基本概念](#)

[5.2 IGMP](#)

[5.3 PIM-DM](#)

[5.4 PIM-SM](#)

[5.5 PIM-SM 的RP](#)

[5.6 RPF 校验](#)

[第6章 交换技术](#)

[6.1 观察和配置MAC 地址表](#)

[6.2 VLAN 基本配置](#)

[6.3 VLAN 间的通信](#)

[6.4 Mux VLAN](#)

[6.5 MSTP/RSTP 与STP 的兼容性](#)

[6.6 MSTP/RSTP 的保护功能](#)

[第7章 MPLS](#)

[7.1 MPLS 和LDP 基本配置](#)

[7.2 BGP/MPLS VPN 基本配置](#)

[第8章 其他](#)

[8.1 配置AAA](#)

[8.2 配置BFD](#)

8.3 综合实验1

8.4 综合实验2

华为ICT认证系列丛书
华为技术认证

HCNP路由交换实验指南

华为技术有限公司 主编

人民邮电出版社

北京

图书在版编目 (CIP) 数据

HCNP路由交换实验指南/华为技术有限公司主编.--北京:人民邮电出版社, 2014.12

(华为ICT认证系列丛书)

ISBN 978-7-115-36987-1

I.①H... II.①华... III.①计算机网络—路由选择—指南②计算机网络—信息交换机—指南 IV.①TN915.05-62

中国版本图书馆CIP数据核字 (2014) 第209107号

内容提要

本书基于eNSP搭建企业网络真实场景,给出大量的配置实例,将真实场景与配置实例紧密结合,使读者能够快捷、直观、深刻地掌握HCNP所需的知识,提高操作技能,增强实战经验。本书的主要内容包含两方面,一方面对《HCNA网络技术实验指南》涉及到的ACL、VLAN、RIP、OSPF、STP等知识点在复杂度和难度上进行了提升,另一方面是增加了HCNP所涉及的路由策略、BGP、IS-IS、IP组播、MPLS等新的知识点,特别适合于正在学习和备考HCNP,或者希望进一步提升对网络知识的理解及实际操作技能的读者朋友。

◆主编 华为技术有限公司

责任编辑 李静

责任印制 程彦红

◆人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京隆昌伟业印刷有限公司印刷

◆开本: 787×1092 1/16

印张: 38 2014年12月第1版

字数: 900千字 2014年12月北京第1次印刷

定价：89.00元

读者服务热线：（010）81055488 印装质量热线：（010）

81055316

反盗版热线：（010）81055315

序

作为全球领先的信息与通信解决方案供应商，华为以“丰富人们的沟通和生活”为愿景，利用在ICT领域的专业技术和经验，帮助不同地区的人们平等、自由地接入到信息社会，确保所有人都能享受到信息和通信服务的基本权利，消除数字鸿沟。我们提倡和致力于信息和通信技术的普及，增加教育机会并培养ICT人才。

为帮助广大 ICT 从业人员更好地学习信息和网络技术，华为技术有限公司于 2012年9月发布了业界首款免费的企业网络仿真软件平台 eNSP(Enterprise Network Simulation Platform)。这款仿真软件平台主要对企业网络路由器、交换机、防火墙、WLAN等网络设备进行软件仿真，具备仿真度高、界面友好、操作方便、版本更新及时等特点。eNSP一经推出就受到社会的广泛关注和欢迎，下载量已超过百万，迅速成为ICT从业人员学习信息和网络技术的首选工具。2014年5月出版的与eNSP配套使用的《HCNA网络技术实验指南》一书，更是让广大的读者朋友们体验到了利用 eNSP 学习信息和网络技术的高效性和趣味性。

此次出版的由华为技术有限公司与武汉誉天互联科技有限责任公司联合编写的《HCNP路由交换实验指南》一书，是《HCNA网络技术实验指南》的进阶，相信它一定能够进一步提高读者朋友们利用 eNSP 学习信息和网络技术的积极性，同时成为 HCNP备考者及时而得力的好帮手。如果说《HCNA网络技术实验指南》已经以一种新颖的方式为读者朋友们开启了一段利用 eNSP 学习探索信息和网络技术的知识旅程，那么本书呈现给大家的则是前进途中一幅幅妙趣横生、精彩动人的景象。在此，预祝读者朋友们“旅途”愉快！

figure_0004_0001

全球培训与认证部部长

华为企业业务集团

2014年8月

前言

本书共有63个实验，每个实验都包括“原理概述”、“实验目的”、“实验内容”、“实验拓扑”、“实验编址表”、“实验步骤”、“思考”等模块。读者应首先阅读“原理概述”和“实验目的”，了解本实验应该掌握的知识和技能，然后再进行实验操作。实验过程中请读者仔细阅读“实验步骤”中的说明，这些内容将很好地展示实验的思路和方法。最后的“思考”模块，可以启发读者进一步的思考，能够更加全面而深刻地理解相关的知识。

为了便于读者学习和练习，我们把每个实验项目都做成了独立的eNSP实验软件包，包括每个实验的最终配置和思考题答案等内容都放到网站上供读者下载和学习，您可以在本书的“使用说明”中找到相应的网址。

适用读者对象

本书的基本定位是华为HCNP认证的参考书，全方位涵盖了HCNP的知识点。本书适合以下几类读者。

华为路由器和交换机的用户

本书可帮助华为路由器和交换机的用户更加熟练地操作和使用华为设备，加深对网络技术的理解，通过实验模拟现网，丰富项目经验。

ICT从业人员

本书可作为工具用书，帮助ICT从业人员熟悉华为设备，具备快速配置华为路由器和交换机的能力。本书更有助于ICT从业人员获取

华为认证，提升在企业中的个人价值。

高校学生

本书可作为华为信息与网络技术学院的实验教材，也可作为计算机通信等相关专业学生的自学参考书。配合 eNSP 软件，本书可以帮助学生快速地熟悉华为网络设备的操作，理解和掌握信息和网络技术，使学生能够更快地积累企业网络实践经验，更早获得华为认证，在今后的职业生涯中有一个更好的起步。

信息和网络技术爱好者

本书可作为信息和网络技术爱好者的参考书籍，帮助爱好者了解华为产品和技术的特点，掌握华为产品和技术的应用，并为其技术探索活动提供有效的工具和指导。

本书主要内容

全书共分为8章，所有实验都以eNSP作为实验工具，并按照HCNP的知识点进行设计。

第1章：路由基础

本章回顾了最基本的路由知识，重点是增强读者对 ACL 和路由策略配置操作的熟练程度，因为在后续的实验中会经常用到这方面的知识和配置操作技能。

第2章：OSPF

本章聚焦于OSPF相关的各个知识点，具体包括OSPF基本配置，OSPF邻居邻接关系，OSPF 链路状态数据库，Stub 区域，NSSA 区域，虚链路，OSPF 网络类型，OSPF路由聚合，OSPF缺省路由，OSPF网络的监测、调试和排障。

第3章：BGP

本章聚焦于BGP相关的各个知识点，具体包括BGP邻居关系，BGP认证功能，BGP自动和手动路由聚合，各种常见的 BGP 路由属性

分析，路由反射器，路由黑洞问题，BGP联盟，BGP路由的过滤、引入和衰减，BGP缺省路由，BGP网络的监测、调试和排障。

第4章：IS-IS

本章聚焦于IS-IS相关的各个知识点，具体包括IS-IS基本配置，IS-IS邻接关系，IS-IS链路状态数据库，IS-IS DIS，IS-IS 的接口开销值和IS-IS 路由的协议优先级，IS-IS 路由的聚合、引入、过滤和渗透，IS-IS缺省路由，IS-IS网络的监测、调试和排障。

第5章：IP组播

本章聚焦于IP组播相关的各个知识点，具体包括IP组播的基本概念，IGMP介绍，PIM-DM 和PIM-SM，Rendezvous Point 以及RPF 校验。

第6章：交换技术

本章聚焦于交换技术的各个知识点，具体包括MAC地址表分析，VLAN基础知识回顾，常见的VLAN间通信技术，Mux VLAN，MSTP/RSTP与STP的兼容性，MSTP/RSTP的保护功能。

第7章：MPLS

本章只涉及到MPLS相关的基础知识，介绍了MPLS和LDP的基本配置方法，并以BGP/MPLS VPN 为例，展示了MPLS 的典型应用。

第8章：其他

作为全书的结尾，本章给出了两个综合实验，以帮助读者综合运用并检验从本书中所学的知识。这两个实验的网络架构和网络需求都较为复杂，目的是锻炼读者分析和配置中小型企业网络的综合能力。

鸣谢

本书由华为技术有限公司与武汉誉天互联科技有限责任公司联合编写，经过双方多位编写老师半年多时间的辛勤工作，严格审校、修改和完善，这本实验指南终于高质量完成并得以顺利出版。在此感谢

武汉誉天互联科技有限责任公司各位老师的付出和大力支持，感谢人民邮电出版社各位编辑老师，以及各位编委和审校人员的辛勤工作！

参与本书编写和审稿的老师虽然拥有多年ICT从业经验，但错漏之处在所难免，望读者朋友们不吝赐教，在此表示衷心的感谢。对于本书的任何意见和建议，敬请发送邮件至Learning@huawei.com。

以下是参与本书编写和技术审校的人员名单。（排名不分先后）

主编：涂文杰

编委人员：刘晶晶、江永红、宋新华、徐一鸣、阮维、龚腾、黄飞、管华、李海星、龚剑、张智勇、邹圣林

技术审校：江永红、徐一鸣、刘洋、杨瑞明、屠晓峰

华为认证简介

华为认证是华为公司凭借多年信息通信技术人才培养经验，以及对行业发展的深刻理解，基于 ICT（Information Communication Technology，信息通信技术）产业链人才个人职业发展生命周期，搭载华为“云-管-端”融合技术，推出的覆盖 IP、IT、CT 以及 ICT 融合技术领域的认证体系，是业界唯一的 ICT 全技术领域认证体系。

华为技术有限公司经过 20 多年在 ICT 行业培训和认证领域的积累，已经在全球形成了完整的培训认证体系，包括自有的培训中心、授权的培训中心以及与高校合作的教育项目，累计参加华为培训的人次已超过 300 万，培训与考试服务覆盖 160 多个国家。

对行业不同领域的人才，华为均有与之匹配的知识和技能培养解决方案，对其进行准确合理的能力评估。针对个人的职业发展历程，华为提供从工程师到资深工程师、专家、架构师层级，以及从单一的技术领域到 ICT 融合的职业技术认证体系。

如果希望全面了解华为认证培训相关信息，敬请访问华为培训认证主页（<http://support.huawei.com/learning>）；如果希望了解华为认证最新动态，敬请关注华为认证官方微博（<http://e.weibo.com/hwcertification>）；如果希望和广大用户一起进行技术问题的探讨，以及考试学习资料的分享，可通过华为官方论坛链接（<http://support.huawei.com/ecommunity/bbs>）点击进入华为认证版块。华为职业技术认证包含的内容如图 1 所示。

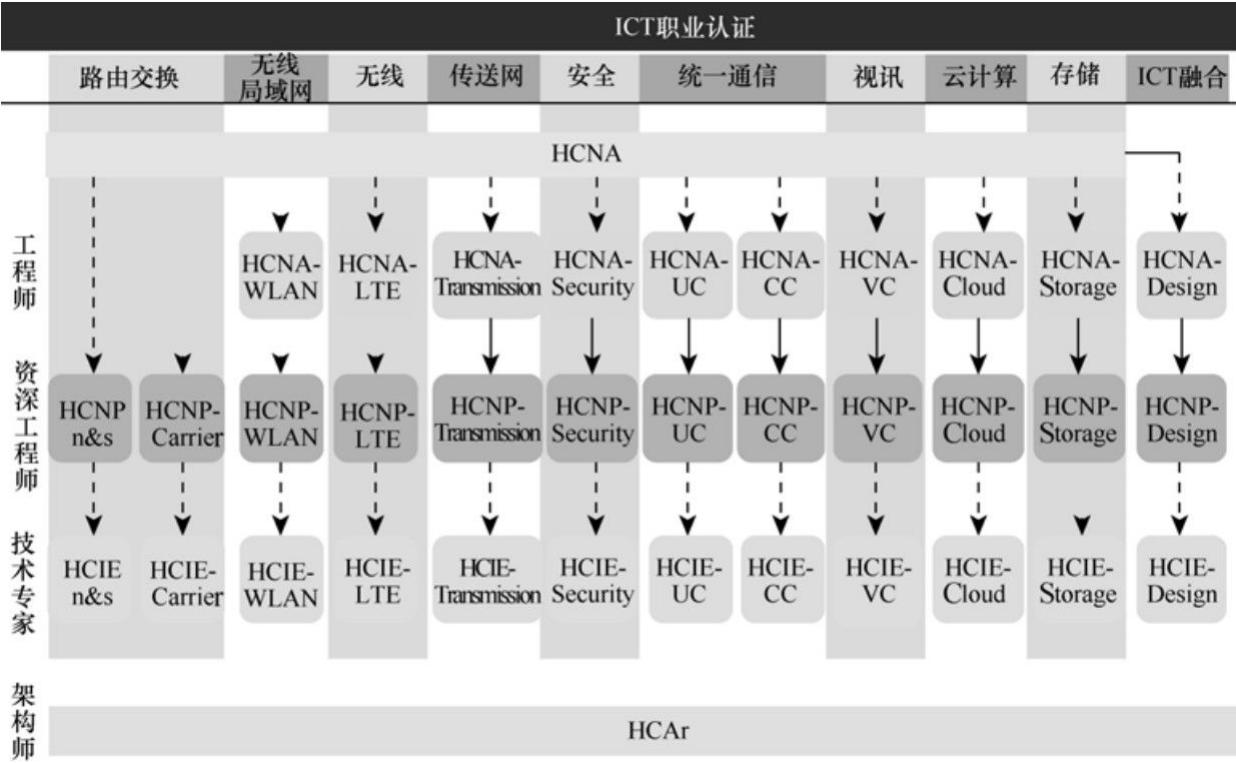


图1 华为职业技术认证的内容

华为路由交换产品介绍

AR系列路由器

2011年，华为技术有限公司发布了第三代企业接入路由器 AR G3 系列。该系列采用多核CPU及大容量交换网，是集安全、语音、互联、无线于一体的多业务的企业路由器，通过了北美权威机构的评测，性能是业界水平的两倍以上，从根本上为企业多业务环境的优质体验提供了保证。

AR G3 系列企业路由器一般部署在企业内部与外部网络的连接处，是内部与外部网络之间数据流的唯一出入口，它可将多种业务部署在同一设备上，极大地降低企业网络建设的初期投资与长期运维成本。用户可以根据企业用户规模选择不同规格的AR G3 路由器，作为出口网关设备。AR系列路由器如图2所示。

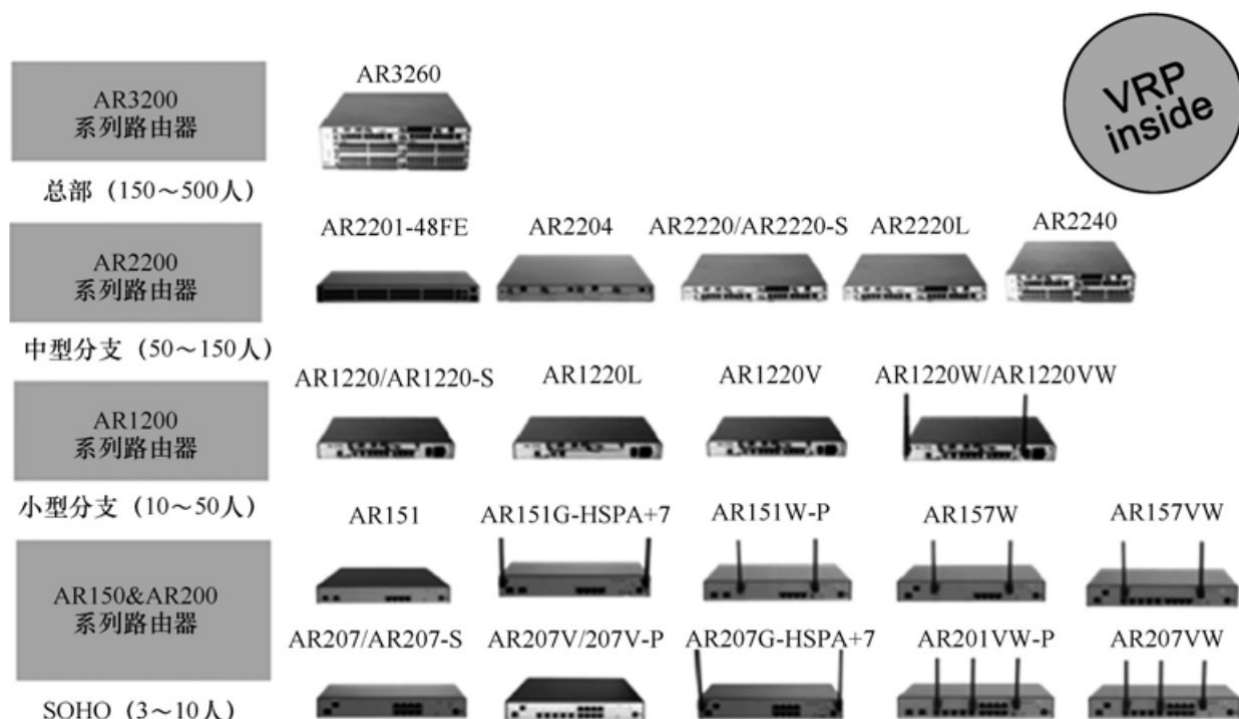


图2 AR系列路由器

Sx700系列交换机

华为技术有限公司于2010年6月推出了面向企业网络的Sx700系列交换机。Sx700系列交换机在提供高带宽、高性能服务的基础上，融合了可靠、安全、绿色环保等先进技术，具备强大的扩展性，可以满足企业网络的持续演进需求；在提高用户生产效率的同时，保证了网络最大正常的运行时间，从而降低了用户的总成本。基于新一代高性能硬件和统一的VRP平台，Sx700系列交换机特别适合于局域网络的部署和建设，以及数据中心的接入应用。在资质和认证方面，华为交换机在基本功能、节能减排、可靠性、互通性等方面都通过了北美权威评测机构的全方位测试和认证，是业界不可多得的高性价比网络产品。Sx700系列交换机如图3所示。

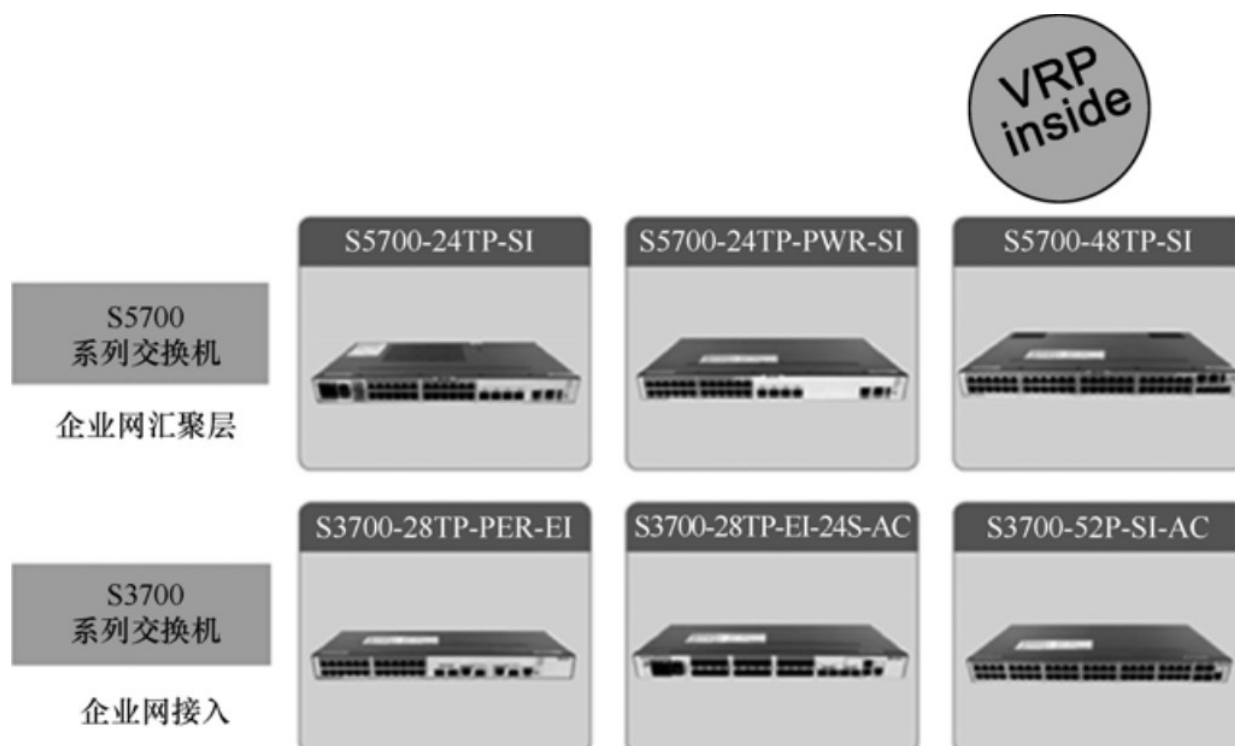


图3 Sx700系列交换机

企业网络解决方案

企业网络的构建可根据企业本身的规模大小选择不同的解决方案。例如，一个小型分支机构，可使用S3700作为接入层交换机，直接连接到作为网络出口的AR系列路由器。大中型企业网络通常需要分层设计，接入层可部署S3700交换机，实现对不同类型用户终端的接入；汇聚层可采用S5700交换机，下行通过千兆接口接入交换机，上行通过万兆光口连接核心层路由器；核心层可根据需求选择不同规格的AR路由器。华为数通产品企业网络解决方案场景如图4所示。

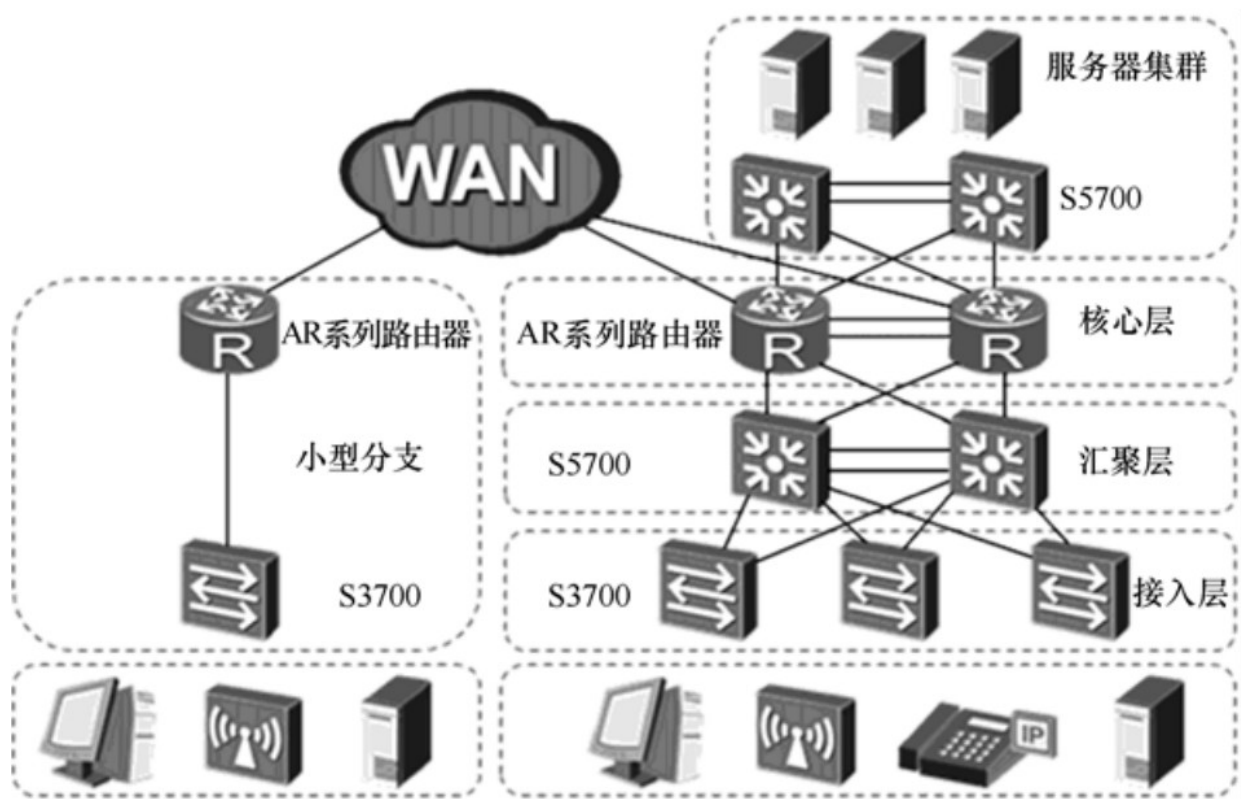


图4 华为数通产品企业网络解决方案场景

eNSP介绍

eNSP简介

eNSP（Enterprise Network Simulation Platform）是一款由华为技术有限公司自主开发的、免费的、可扩展的、图形化操作的网络仿真工具平台，主要是对企业网络路由器、交换机及相关物理设备进行软件仿真，完美呈现真实设备实景，支持大型网络模拟，让广大用户能够在没有真实设备的情况下模拟演练，学习和探索网络技术。

eNSP（企业网络仿真平台）具有仿真程度高、更新及时、界面友好、操作方便等特点。这款仿真软件运行的是与真实设备一样的VRP操作系统，能够最大程度地模拟真实设备环境。用户可以利用 eNSP 模拟工程开局和网络测试，高效地构建企业优质的 ICT网络。eNSP 还支持与真实设备对接，以及数据包的实时抓取，可以帮助用户深刻理解网络协议的运行原理，协助用户进行网络技术的钻研和探索。另外，用户还可以利用 eNSP 模拟华为认证相关的实验（如 R&S/Security/WLAN方向的HCNA/HCNP/HCIE实验），能够更轻松地获得华为认证，成就技术专家之路。

eNSP 的免费发布，为用户提供了近距离体验华为设备的机会。无论是操作数通产品、维护现网的技术工程师，还是教授网络技术的培训讲师，或者是想要获得华为认证、获得能力认可的在校学生，相信都可以因eNSP 而受益匪浅。

eNSP的特点

针对影响用户体验的主要问题，例如安装是否方便，仿真度是否够高，是否可视化操作，是否可更新等，eNSP 做到了扬各家之长，避各家之短，给用户带来极佳的操作体验，它具备以下几个特点。

1．人性化图形界面

eNSP全新的UI界面如图5所示。图形化界面不但美观，而且操作时可轻松上手，包括拓扑搭建和配置设备等。

2．设备图形化直观展示，支持插拔接口卡

在设备真实的图形化视图下，可将不同的接口卡拖曳到设备空槽位，单击电源开关即可启动或关闭设备，使用户对设备的感受更加直观。

3．多机互联，分布式部署

在4台服务器上最多可部署200台左右的模拟设备，并且实现互联，可以模拟大型复杂网络实验。

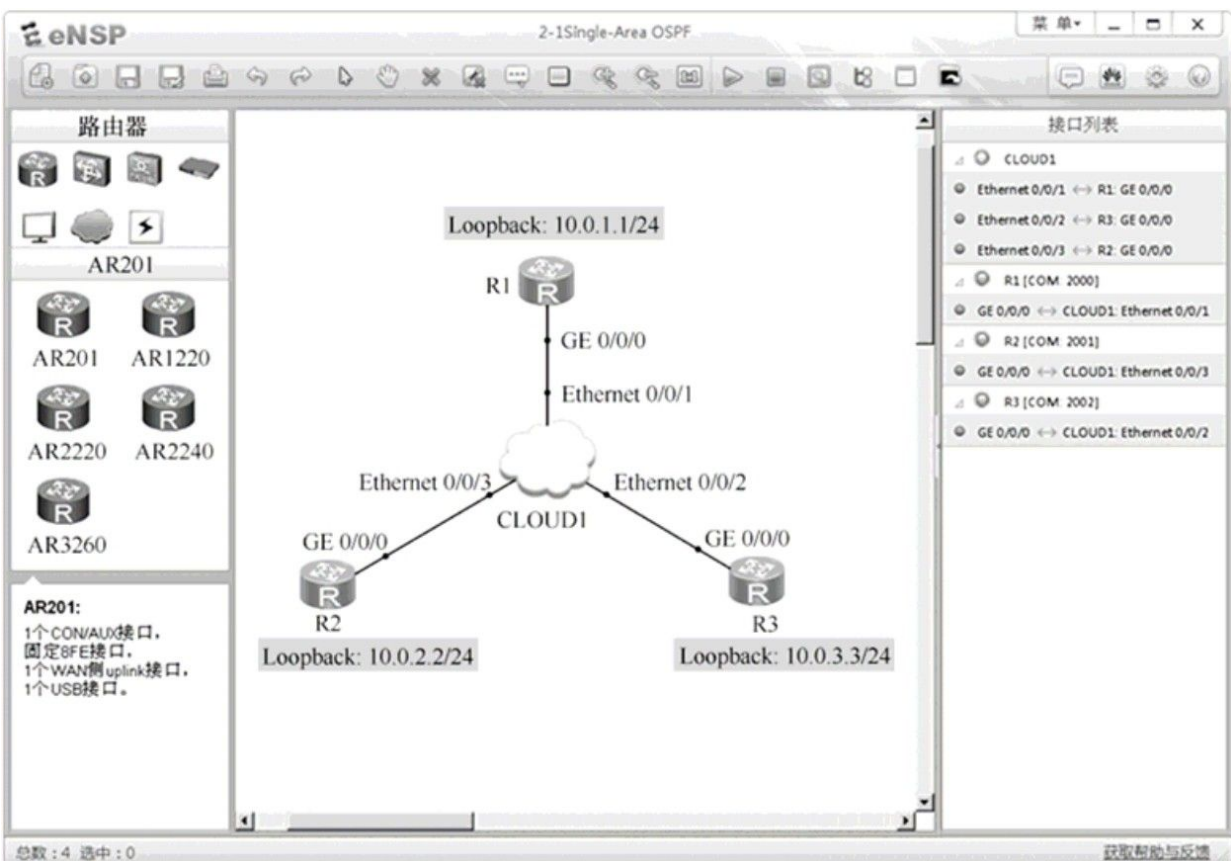


图5 eNSP全新的UI界面

4．高度仿真，实景再现，支持设备功能多

高度仿真的二层转发，运行华为通用路由平台VRP系统，支持对路由器、交换机各种特性的仿真和模拟（包括STP/RSTP/MSTP、Mux VLAN、SEP、GVRP 等各种协议），提供AR全系列仿真款型，支持NAT、防火墙、IPSec、SSLVPN、MQC、AC等功能。

5．不断增加的功能特性模拟

随着真实产品的升级更新，eNSP 将支持增加更多更新的功能特性与之对应。用户发现任何问题都可以通过华为官网论坛（<http://support.huawei.com/ecomcommunity>）进行反馈，华为技术有限公司有专人负责技术支持和疑问解答，亦可通过邮箱 eNetwork_tools@huawei.com 进行反馈，反馈的问题和建议将通过后续月度版本予以快速响应。

6．完全免费

软件完全免费，面向所有人群开放下载，用户可登录 <http://enterprise.huawei.com/cn/> 华为官方网站进行下载。

使用说明

本书所使用的eNSP软件版本为V100R002C00B210，请读者于官网上对应下载使用，避免因使用软件版本不符而造成实际实验操作与书中内容不一致。

受篇幅所限，在本书的实验命令输出信息中，凡是不与实验主题相关的部分都以省略号“.....”替代。

特别需要说明的是，在现实网络的部署中，公网IP地址和私网IP地址的使用范围和条件是有严格规定的。由于本书中的实验会用到大量的IP地址，稍有疏忽，便会记错并用错这些IP地址，为此，我们刻意淡化了公网和私网IP地址的严格区别，而只是从IP地址的易记性出发来选择实验所需的IP地址。

本书中每个实验的拓扑图、思考题答案及最终配置都以电子文件形式在网页上提供免费下载，详情请访问华为官方论坛链接（<http://support.huawei.com/ecomcommunity/bbs>），点击进入华为认证版块（二维码如图6所示）。



图6 华为官方论坛中华为认证板块的二维码

本书常用图标



第1章 路由基础

- 1.1 访问控制列表
- 1.2 基本的路由策略配置
- 1.3 控制RIP路由的发布及路由引入

1.1 访问控制列表

原理概述

访问控制列表（ACL：Access Control List）是一种常用的网络技术，它的基本功能是对经过网络设备的报文进行过滤处理。ACL是由permit和deny语句组成的一个有序规则的集合，它首先通过报文匹配过程来实现对报文的分类识别，然后根据报文的分类信息和相关的执行动作来判断哪些报文可以放行，哪些报文不能放行，从而实现对特定报文的过滤处理。除此之外，ACL 还有其他一些功能，这些功能常常被Route-Policy、QoS（Quality of Service）、IPSec（IP Security）、Firewall 等技术结合起来使用。

ACL的常见类型：基本ACL、高级ACL、二层ACL、用户自定义ACL等，其中应用最为广泛的是基本ACL和高级ACL。基本ACL可以根据源IP地址、报文分片标记和时间段信息来定义规则；高级ACL可以根据源/目的IP地址、TCP源/目的端口号、UDP 源/目的端口号、协议

号、报文优先级、报文大小、时间段等信息来定义规则。高级ACL可以比基本ACL定义出精细度更高的规则。

基本ACL、高级ACL、二层ACL、用户自定义ACL的编号范围分别为2000 ~ 2999、3000 ~ 3999、4000 ~ 4999、5000 ~ 5999。使用ACL时，ACL的类型应该与相应的编号范围保持一致。

实验目的

- 掌握基本ACL 和高级ACL 在安全策略中的应用
- 掌握基于时间信息的ACL 配置
- 掌握使用基本ACL 保护路由器的VTY 线路

实验内容

实验拓扑如图1-1所示，实验编址如表1-1所示。本实验模拟了一个简单的公司网络，基本组成：一台Ftp-Server，一台Web-Server，一台HR部门的终端PC-1，一台SALES部门的终端PC-2，一台IT部门的终端PC-3，一台路由器R1和一台交换机SW1。为了满足公司的安全需求，要求在R1上使用ACL对部门之间的互访，以及用户对服务器的访问进行控制。另外，只允许SW1的VLANIF 1 接口的IP地址作为源地址远程登录到R1，以实现对R1的远程控制和管理。

实验拓扑

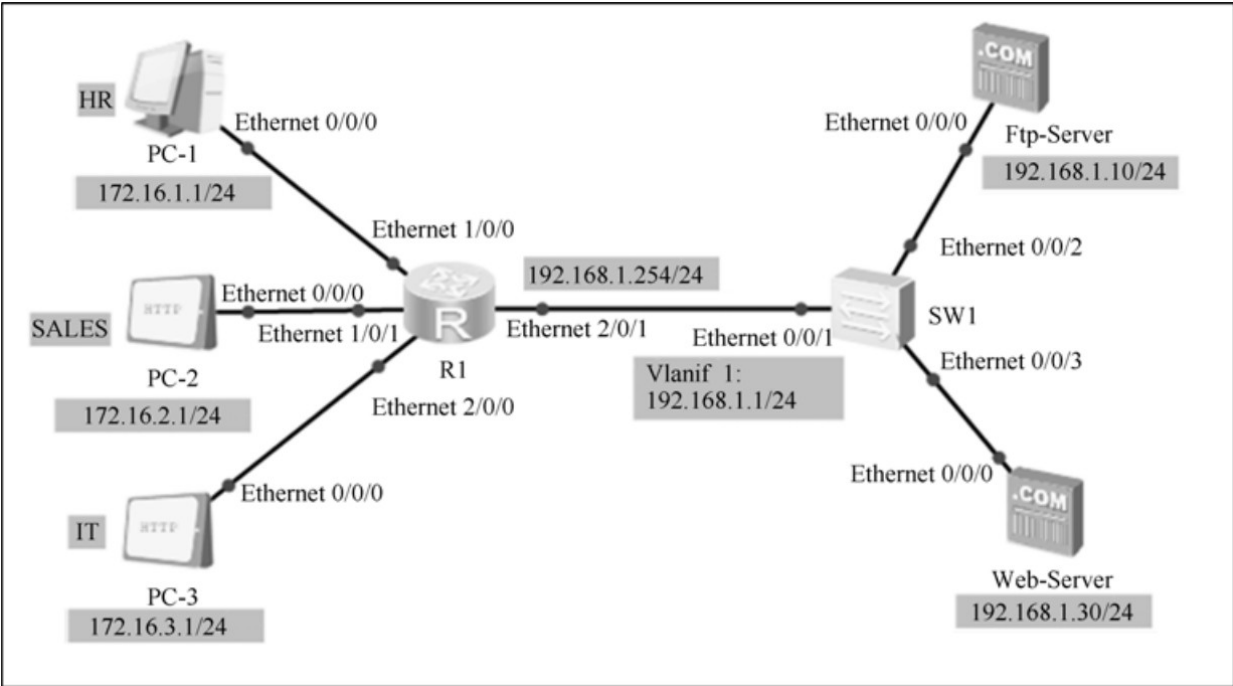


图1-1 实验拓扑图

实验编址表

表1-1 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR1220)	Ethernet 1/0/0	172.16.1.254	255.255.255.0	N/A
	Ethernet 1/0/1	172.16.2.254	255.255.255.0	N/A
	Ethernet 2/0/0	172.16.3.254	255.255.255.0	N/A
	Ethernet 2/0/1	192.168.1.254	255.255.255.0	N/A
SW1(S3700)	Vlanif 1	192.168.1.1	255.255.255.0	N/A
Web-Server	Ethernet 0/0/0	192.168.1.30	255.255.255.0	192.168.1.254
Ftp-Server	Ethernet 0/0/0	192.168.1.10	255.255.255.0	192.168.1.254
PC-1	Ethernet 0/0/0	172.16.1.1	255.255.255.0	172.16.1.254
PC-2	Ethernet 0/0/0	172.16.2.1	255.255.255.0	172.16.2.254
PC-3	Ethernet 0/0/0	172.16.3.1	255.255.255.0	172.16.3.254

实验步骤

1. 基本配置

根据图1-1和表1-1进行相应的基本配置，并使用ping命令检测R1与PC-1之间的连通性。

<R1>ping -c 1 172.16.1.1

PING 172.16.1.1: 56 data bytes, press CTRL_C to break


```
Reply from 172.16.1.1: bytes=56 Sequence=1 ttl=255 time=90 ms
--- 172.16.1.1 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
round-trip min/avg/max = 90/90/90 ms
```

其余直连网段的连通性测试过程在此省略。

使用eNSP软件设置Ftp-Server，用鼠标右键单击拓扑图中的Ftp-Server图标，选择“设置”选项，出现配置界面。注意，在配置之前不要启动设备。如图1-2所示，在“基础配置”页面设置IP地址、子网掩码、网关。本实验直接采用IP地址方式访问服务器，所以无需设置域名服务器。



图1-2 Ftp-Server“基础配置”页面

点击进入“服务器信息”页面，如图1-3所示，在左侧选择栏中选中“FtpServer”，设置端口号为21，设置文件根目录为F:\ftp。根目录下的ftptest.zip文件将作为测试文件在后续测试中使用，读者也可自行创建根目录名和测试文件。

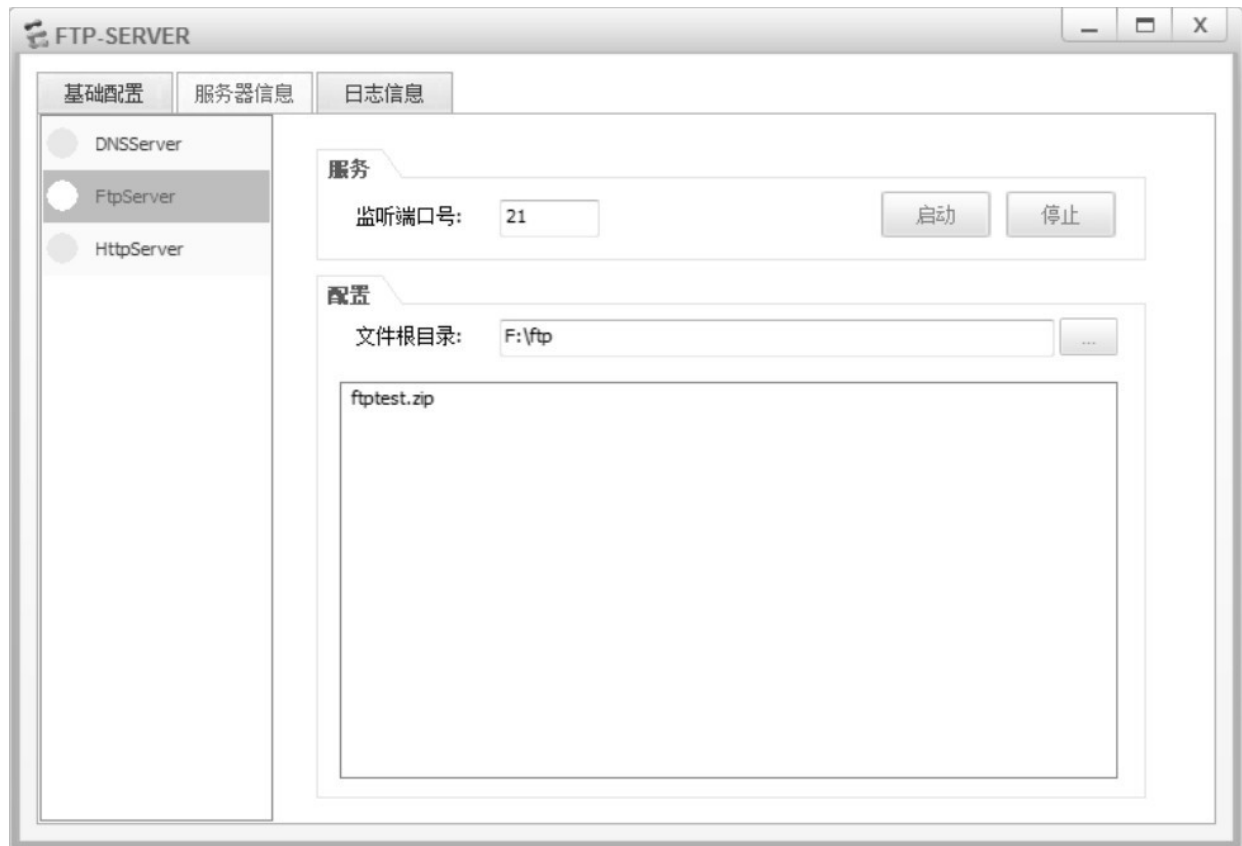


图1-3 Ftp-Server“服务器信息”页面

设置完成后，关闭设置窗口，右键点击 Ftp-Server 图标，选择“启动”选项，然后再次进入“服务器信息”页面，点击“启动”选项。如图1-4所示，“日志信息”页面中出现“FtpServer:FtpServerInfo: Listening!”提示，表明Ftp-Server已正常启动，此时关闭该窗口即可。

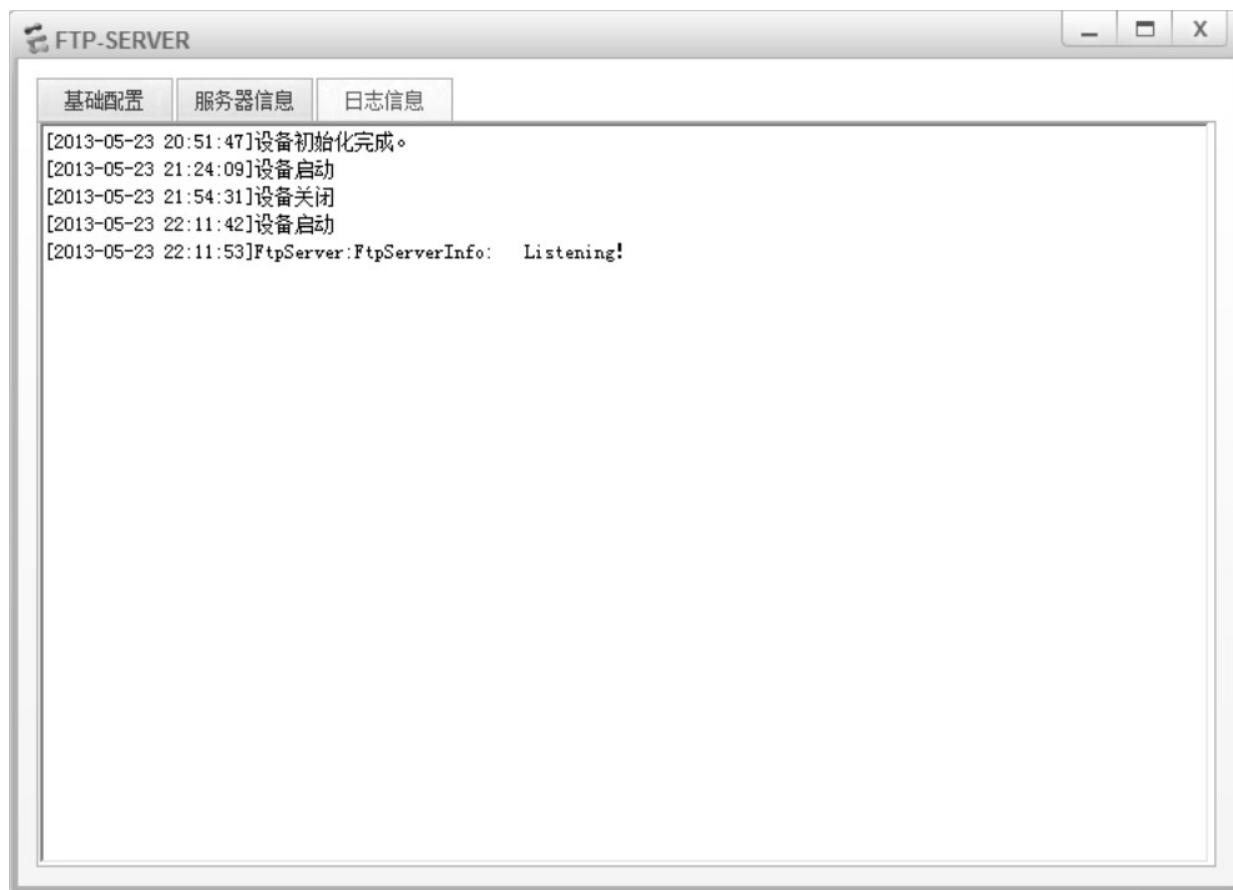


图1-4 Ftp-Server“日志信息”页面

接下来，使用eNSP软件设置Web-Server，它的“基础配置”与Ftp-Server的“基础配置”类似，请参照图1-5。

WEB-SERVER

基础配置 服务器信息 日志信息

Mac地址: 54-89-98-CF-11-3C (格式:00-01-02-03-04-05) 保存

IPv4配置

本机地址: 192 . 168 . 1 . 30 子网掩码: 255 . 255 . 255 . 0

网关: 192 . 168 . 1 . 254 域名服务器: . . .

PING测试

目的IPv4: 0 . 0 . 0 . 0 次数: 1 发送

本机状态: 设备关闭 ping 成功: 0 失败: 0

图1-5 Web-Server“基础配置”页面

然后，点击进入“服务器信息”页面，在左侧选择栏中选中“HttpServer”选项，设置端口号为 80，设置文件根目录为 F:\web。如图 1-6 所示，在根目录下创建文件 default.htm，文件内容为“Hello, Welcome to Huawei.com”。读者也可自行创建根目录名、文件名和文件内容。

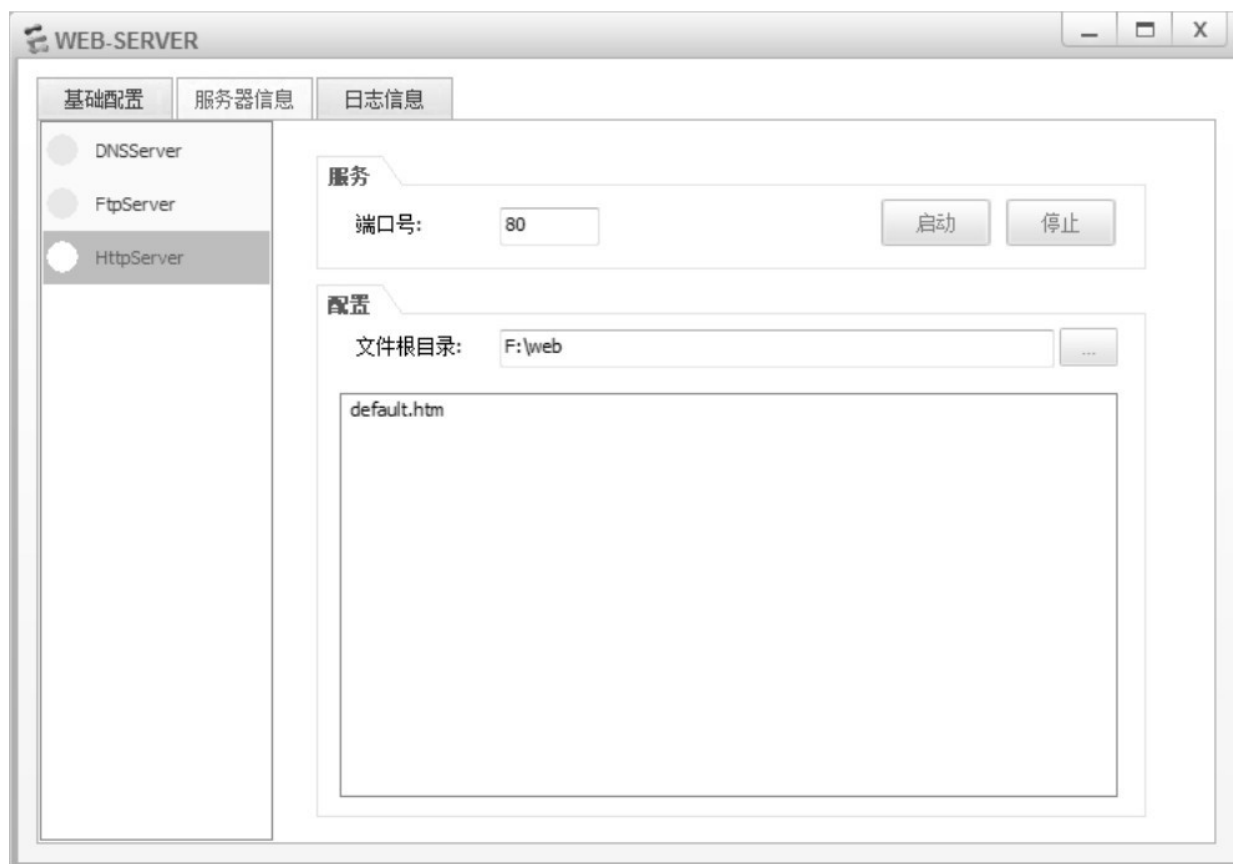


图1-6 Web-Server“服务器信息”页面

设置完成后，关闭窗口，右键点击Web-Server图标，选择“启动”选项，然后再次打开“服务器信息”页面，点击“启动”选项，“日志信息”页面中会出现“设备启动”的提示信息，表明Web-Server已经正常启动，如图1-7所示。



图1-7 Web-Server“日志信息”页面

使用eNSP软件设置终端PC-1，右键单击拓扑图中的PC-1图标，选择“设置”选项，出现配置界面，然后在“基础配置”页面下，进行如图1-8所示的配置，最后点击“应用”。PC-2和PC-3的配置过程完全类似，故在此省略。



图1-8 PC-1“基础配置”页面

2. 为各部门创建安全区域

公司希望使用华为AR系列路由器的域间防火墙特性来提高安全性，所以需要在R1上为HR、SALES、IT这3个部门分别创建安全区域。区域的名称和部门名称一致，HR区域的安全级别设置为12，SALES区域的安全级别设置为10，IT区域的安全级别设置为8。另外，还需要创建 Trust 区域，设置 Trust 区域的安全级别为 14，Ftp-Server 和Web-Server都属于Trust区域。AR系列路由器默认可以设置16种安全级别，取值范围为0 ~ 15，15保留给Local区域使用。

```
[R1]firewall zone HR
```

```
[R1-zone-HR]priority 12
```

```
[R1-zone-HR]firewall zone SALES
```

```
[R1-zone-SALES]priority 10
```

```
[R1-zone-SALES]firewall zone IT
```

```
[R1-zone-IT]priority 8
```

```
[R1-zone-IT]firewall zone trust
```

```
[R1-zone-trust]priority 14
```

将 R1 上连接不同部门的接口加入到相应部门的安全区域中，
Ethernet 2/0/1 接口加入到Trust 区域中。

```
[R1]interface Ethernet 1/0/0
```

```
[R1-Ethernet1/0/0]zone HR
```

```
[R1-Ethernet1/0/0]interface Ethernet 1/0/1
```

```
[R1-Ethernet1/0/1]zone SALES
```

```
[R1-Ethernet1/0/1]interface Ethernet 2/0/0
```

```
[R1-Ethernet2/0/0]zone IT
```

```
[R1-Ethernet2/0/0]interface Ethernet 2/0/1
```

```
[R1-Ethernet2/0/1]zone trust
```

使用命令display firewall zone 查看相应区域的优先级，区域内包含
接口名称、接口数量等信息。

```
[R1]display firewall zone
```

```
zone IT
```

```
priority is 8
```

```
interface of the zone is (total number 1): Ethernet2/0/0
```

```
zone SALES
```

```
priority is 10
```

```
interface of the zone is (total number 1): Ethernet1/0/1
```

```
zone HR
```

```
priority is 12
```

```
interface of the zone is (total number 1): Ethernet1/0/0
```

```
zone trust
```


priority is 14

interface of the zone is (total number 1): Ethernet2/0/1

zone Local

priority is 15

interface of the zone is (total number 0):

total number is : 5

从上面的显示信息可以看到，所有区域的配置工作已经完成。当把接口加入到相应的区域后，就可以实施基于安全区域的ACL了。

在配置AR系列路由器的防火墙特性时需要注意流量的方向。从较高安全级别区域去往较低安全级别区域的报文称为Outbound报文，从较低安全级别区域去往较高安全级别区域的报文称为Inbound报文。AR系列路由器的防火墙特性允许管理员在不同的区域之间进行报文的过滤处理。

3．禁止SALES部门和HR部门之间的互访

由于 SALES 和 HR 之间目前没有任何业务往来，为了保证部门信息安全，需要在R1上使用ACL来禁止这两个部门之间的互访。

启用SALES区域和HR区域的域间防火墙，命令中的HR和SALES的先后次序没有关系。

```
[R1]firewall interzone SALES HR
```

```
[R1-interzone-HR-SALES]firewall enable
```

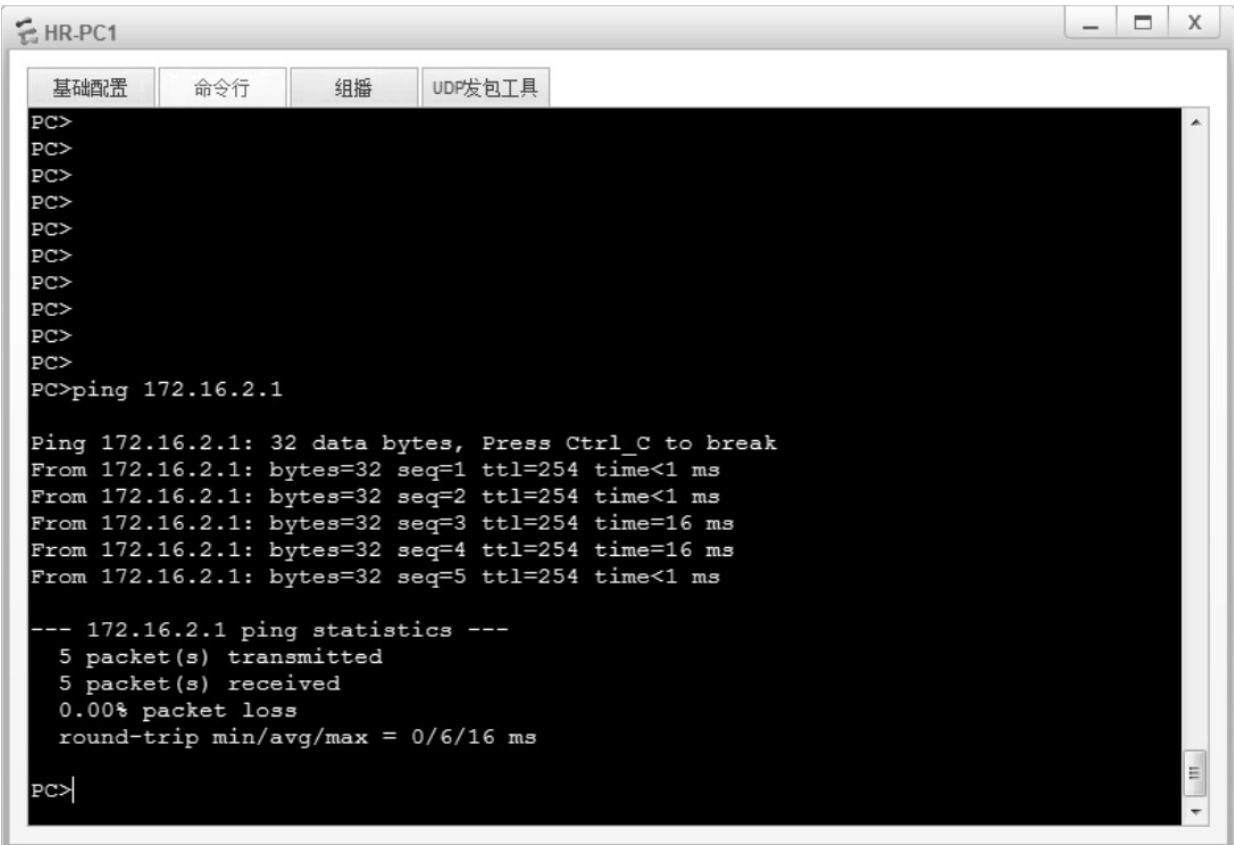
命令firewall enable 的作用是启用域间防火墙。缺省情况下，当域间防火墙启用之后，安全级别较高的区域能够访问安全级别较低的区域，并且应答的报文也能够返回到安全级别较高的区域，但是安全级别较低的区域无法访问安全级别较高的区域。注意，这条命令在这里只开启了HR区域和SALES区域之间的防火墙特性，不会对HR区域和SALES区域与其他区域之间的报文运动有任何影响。

使用命令display firewall interzone HR SALES查看区域间的默认策略。

```
[R1]display firewall interzone SALES HR
interzone HR SALES
firewall enable
packet-filter default deny inbound
packet-filter default permit outbound
```

可以看到，区域间的默认策略为Inbound报文被拒绝通过，而Outbound报文被允许通过。由于HR区域的安全级别为12，SALES区域的安全级别为10，所以从HR区域到SALES区域的报文是Outbound报文，而从SALES区域到HR区域的报文是Inbound报文。

接下来使用ping命令测试PC-1与PC-2之间的连通性。右键单击拓扑图中的PC-1，先选择“设置”选项，然后选择“命令行”选项，使用命令ping 172.16.2.1，结果如图1-9所示。



```
HR-PC1
基础配置 命令行 组播 UDP发包工具
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>ping 172.16.2.1

Ping 172.16.2.1: 32 data bytes, Press Ctrl_C to break
From 172.16.2.1: bytes=32 seq=1 ttl=254 time<1 ms
From 172.16.2.1: bytes=32 seq=2 ttl=254 time<1 ms
From 172.16.2.1: bytes=32 seq=3 ttl=254 time=16 ms
From 172.16.2.1: bytes=32 seq=4 ttl=254 time=16 ms
From 172.16.2.1: bytes=32 seq=5 ttl=254 time<1 ms

--- 172.16.2.1 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 0/6/16 ms

PC>
```

图1-9 在PC-1上检测PC-1与PC-2之间的连通性

从图1-9显示的信息可以判定，Outbound方向的报文是被放行的，返回的Inbound方向的报文也是被放行的。

右键单击拓扑图中的 PC-2，选择“设置”选项，然后在“基础配置”页面上进行ping测试，目的IP地址输入172.16.1.1，次数为5，点击“发送”，结果如图1-10所示。



图1-10 在PC-2上检测PC-2与PC-1之间的连通性

可以看到，结果显示为失败，其原因可以推断为Inbound方向的报文被拒绝通行。

为了禁止HR和SALES这两个部门之间的互访，管理员可以在它们之间使用ACL达到目的。由于默认SALES区域不能访问HR区域，因此，只需在Outbound方向上将HR去往SALES的报文全部过滤掉即可。

创建高级 ACL 3000 来定义从 HR 到 SALES 的报文，步长设置为 10。如果在配置ACL时没有给规则指定序列号，则起始序列号将为步长值，且后续序列号将以步长值的间隔进行累加递增。然后，在Outbound 方向上引用ACL 3000。

```
[R1]acl 3000
```

```
[R1-acl-adv-3000]step 10
```

```
[R1-acl-adv-3000]rule deny ip source 172.16.1.0 0.0.0.255 destination  
172.16.2.0 0.0.0.255
```

```
[R1-acl-adv-3000]firewall interzone SALES HR
```

```
[R1-interzone-HR-SALES]packet-filter 3000 outbound
```

配置完成后，在R1上使用命令display acl 3000 查看ACL 的配置。

```
[R1]display acl 3000
```

```
Advanced ACL 3000, 1 rule
```

```
Acl's step is 10
```

```
rule 10 deny ip source 172.16.1.0 0.0.0.255 destination 172.16.2.0  
0.0.0.255
```

可以看到，ACL 3000 中只有1 个规则，步长为10，规则序列号也为10。

查看SALES和HR的域间Firewall策略。

```
[R1]display firewall interzone SALES HR
```

```
interzone HR SALES
```

```
firewall enable
```

```
packet-filter default deny inbound
```

```
packet-filter default permit outbound
```

```
packet-filter 3000 outbound
```

可以看到，ACL 3000 已经被应用在SALES和HR 的域间Outbound方向上了。

在PC-1上重新测试与PC-2的连通性，如图1-11所示。

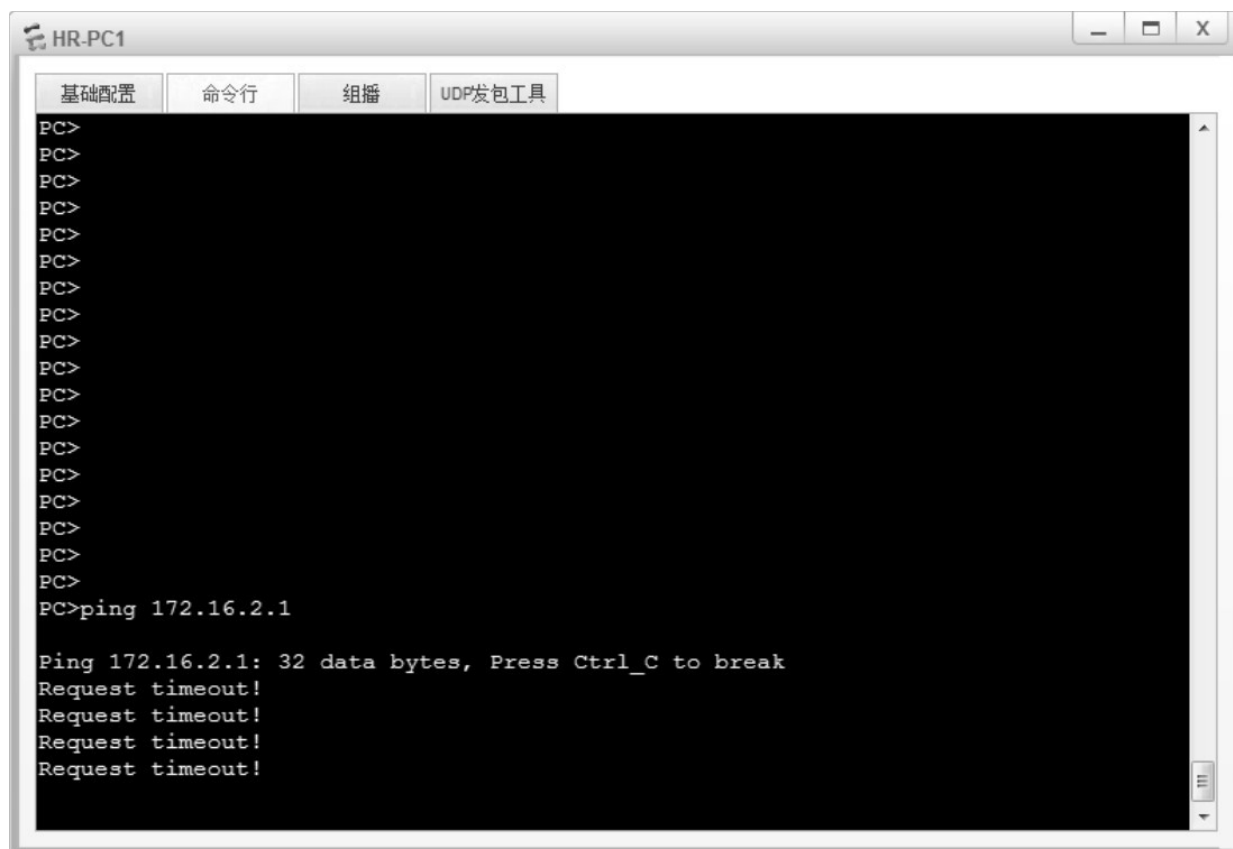


图1-11 在PC-1上重新测试PC-1与PC-2之间的连通性

可以看到，在PC-1上现在无法ping通PC-2，说明相应的安全需求已经得到实现。

4．实现对Web-Server和Ftp-Server访问的控制

接下来的安全需求：SALES 部门的用户可以访问公司的 Web-Server，但禁止访问Ftp-Server。

开启SALES区域和Trust区域间的防火墙。由于SALES区域的安全级别为10，Trust区域的安全级别为14，因此，访问流量的方向为Inbound方向。根据区域间防火墙默认规则，SALES部门的用户是无法访问Trust 区域中的服务器的。因此，创建ACL 3001，在Inbound方向上明确放行SALES区域访问Trust区域的Web-Server的报文，其他访问报文被默认规则拒绝通行，如此便可实现相应的安全需求。

```
[R1]firewall interzone SALES trust
```

[R1-interzone-trust-SALES]firewall enable

在 PC-2 的“客户端信息”页面中进行 Web 服务的测试，在地址栏中输入“http://192.168.1.30/default.htm”，发现无法访问Web-Server，如图 1-12所示。

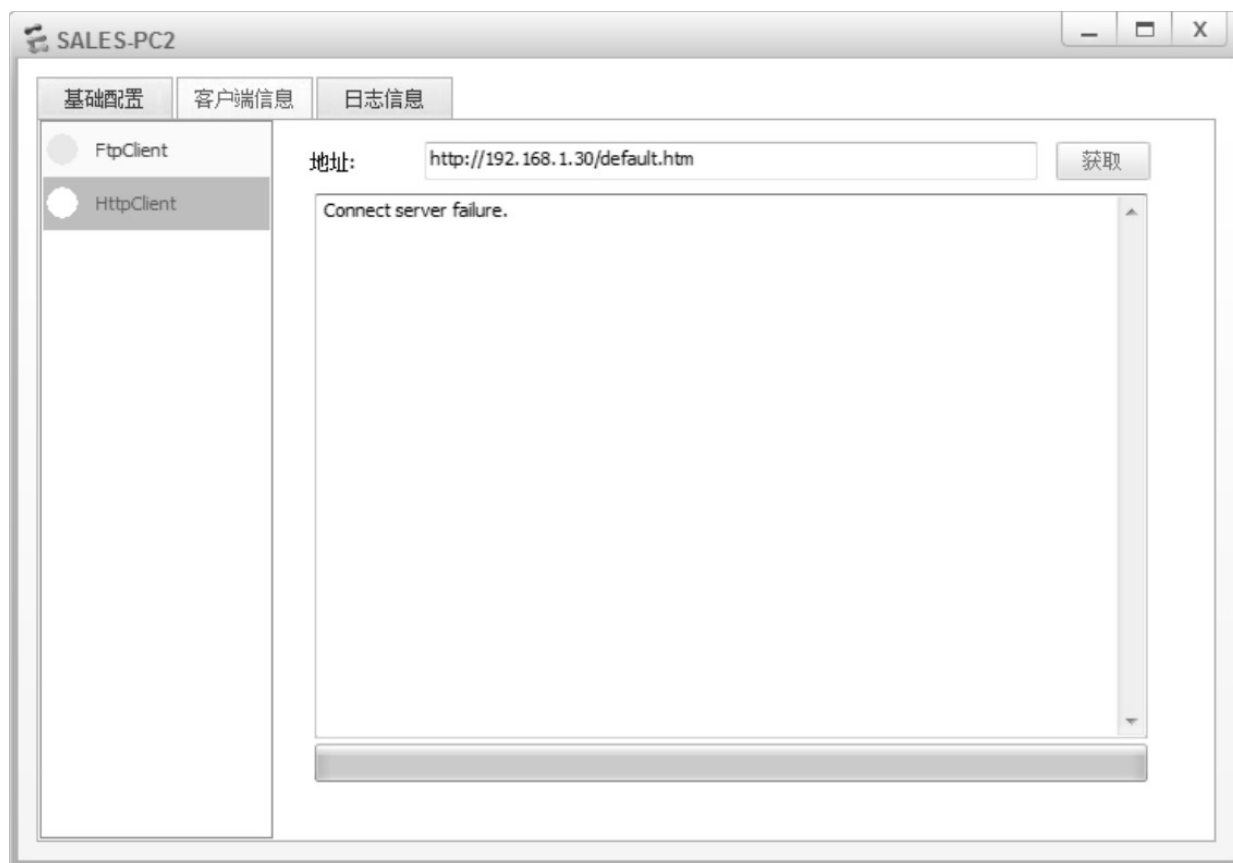


图1-12 PC-2“客户端信息”页面

创建ACL 3001，允许SALES部门的用户访问Web-Server，并应用在SALES和Trust的区域之间。

```
[R1]acl 3001
```

```
[R1-acl-adv-3001]rule 10 permit tcp source 172.16.2.0 0.0.0.255  
destination 192.168.1.30 0 destination-port eq 80
```

```
[R1-acl-adv-3001]firewall interzone SALES trust
```

```
[R1-interzone-trust-SALES]packet-filter 3001 inbound
```

配置完成后，再次测试PC-2能否访问Web-Server，然后测试能否访问Ftp-Server，测试结果分别如图1-13和图1-14所示。



图1-13 PC-2访问Web-Server



图1-14 PC-2访问Ftp-Server

可以看到，结果是PC-2可以访问Web-Server，但是无法访问Ftp-Server，说明安全需求已经得到满足。

现在有个新的安全需求：IT 部门的用户可以随时访问 Ftp-Server，但只能在每天的14：00 至 16：00 才能访问 Web-Server，另外还要求IT 部门的用户能够随时 ping 通Ftp-Server和Web-Server。

开启IT 和Trust 之间的域间防火墙。

```
[R1]firewall interzone IT trust
```

```
[R1-interzone-trust-IT]firewall enable
```

配置时间跨度为每天的14：00-16：00。

```
[R1]time-range access-web 14:00 to 16:00 daily
```

创建ACL 3003，放行IT 到 Trust 的Inbound 方向的FTP、HTTP、ICMP 的ECHO报文，步长设置为10。

```
[R1]acl 3003
[R1-acl-adv-3003]step 10
[R1-acl-adv-3003]rule permit tcp source 172.16.3.0 0.0.0.255
destination 192.168.1.30 0 destination-port eq 80 time-range access-web
[R1-acl-adv-3003]rule permit tcp source 172.16.3.0 0.0.0.255
destination 192.168.1.10 0 destination-port eq 21
[R1-acl-adv-3003]rule permit icmp source 172.16.3.0 0.0.0.255
destination 192.168.1.10 0
[R1-acl-adv-3003]rule permit icmp source 172.16.3.0 0.0.0.255
destination 192.168.1.30 0
```

配置完成后，在R1上查看ACL配置。

```
[R1]display acl 3003
Advanced ACL 3003, 4 rules
Acl's step is 10
rule 10 permit tcp source 172.16.3.0 0.0.0.255 destination
192.168.1.30 0 destination-port eq www time-range access-web (Inactive)
rule 20 permit tcp source 172.16.3.0 0.0.0.255 destination
192.168.1.10 0 destination-port eq ftp
rule 30 permit icmp source 172.16.3.0 0.0.0.255 destination
192.168.1.10 0
rule 40 permit icmp source 172.16.3.0 0.0.0.255 destination
192.168.1.30 0
```

可以看到，规则的序列号是以 10 为步长递增的。ACL 3003 的 rule 10 后面的Inactive 说明当前这条规则没有被激活，这是因为这条规则只在设备时间为time-range 所指定的范围时才会被激活。由于默认规则的存在，所以 IT 部门的用户只有在每天 14：00-16：00 才能访问 Web-Server，而访问 Ftp-Server则无时间上的限制。当 ACL 有多条规则时，

默认情况下路由器将按照规则的序列号从低到高依次将报文与规则进行匹配，一旦某个报文匹配上了某条规则，则按规则指定的动作进行处理，不会再继续往下匹配；对于没有匹配上任何规则的报文，则按默认规则进行处理。

将 ACL 3003 应用在 IT 区域和 Trust 区域之间的 Inbound 方向上，并查看配置情况。

```
[R1]firewall interzone IT trust
[R1-interzone-trust-IT]packet-filter 3003 inbound
[R1]display firewall interzone IT trust
interzone trust IT
firewall enable
packet-filter default deny inbound
packet-filter default permit outbound
packet-filter 3003 inbound
```

现在可以测试PC-3是否能访问Web-Server和Ftp-Server，以及是否能ping通这两个服务器。图1-15 显示了在PC-3 上ping Web-Server的结果，其他测试请读者自行完成。注意，在测试Ftp-Server的时候，文件传输模式请选择PORT模式；正确的结果是能ping通，能以 PORT 模式访问 Ftp-Server。如果时间点正好在 time-range 范围内，则能访问Web-Server，否则无法访问。



图1-15 在PC-3上ping Web-Server

5. 实现对设备的安全控制和管理

为了实现对R1 的安全控制和管理，现在只允许SW1 上的VLANIF 1 接口的IP 地址192.168.1.1能够作为源地址登录到R1。

在SW1 上创建VLANIF 1 接口，配置IP 地址为192.168.1.1/24，在R1 上配置VTY用户接口，允许远程主机通过Telnet管理R1。本例中的Telnet密码为huawei。

```
[SW1]interface Vlanif 1
```

```
[SW1-Vlanif1]ip add 192.168.1.1 24
```

```
[R1]user-interface vty 0 4
```

```
[R1-ui-vty0-4]authentication-mode password
```

```
Please configure the login password (maximum length 16):huawei
```

使用基本ACL对路由器的VTY终端进行保护，只允许源地址为192.168.1.1的报文访问R1的VTY终端。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]rule permit source 192.168.1.1 0
```

```
[R1-acl-basic-2000]user-interface vty 0 4
```

```
[R1-ui-vty0-4]acl 2000 inbound
```

在SW1上使用命令telnet，按照提示输入密码后，就可以登录到R1上了。

```
<SW1>telnet 192.168.1.254
```

```
Trying 192.168.1.254 ...
```

```
Press CTRL+K to abort
```

```
Connected to 192.168.1.254 ...
```

```
Login authentication
```

```
Password:
```

```
<R1>
```

现在，将SW1的VLANIF 1接口的IP地址修改为192.168.1.2。

```
[SW1]interface Vlanif 1
```

```
[SW1-Vlanif1]ip add 192.168.1.2 24
```

在SW1的用户视图下，再次测试是否能登录到R1。

```
<SW1>telnet 192.168.1.254
```

```
Trying 192.168.1.254 ...
```

```
Press CTRL+K to abort
```

```
Error: Can't connect to the remote host
```

测试结果显示，当SW1的VLANIF 1的IP地址更换后，便无法登录到R1。通过上面的方法，可以防止未被允许的IP地址登录到路由器，从而实现对路由器的安全控制和管理。

思考

ACL 中的通配符掩码（Wildcard Mask）的作用是什么？如果需要将172.16.16.0/24 ~ 172.16.31.0/24这个范围内的所有IP地址用一条ACL规则表示，那么通配符掩码应该是多少？

1.2 基本的路由策略配置

原理概述

路由策略 Route-Policy 的应用非常广泛。例如，它可以规定路由器在发布路由时只发布某些满足特定条件的路由，在接收路由时只接收某些满足特定条件的路由，在引入路由时只引入某些满足特定条件的路由，如此等等。

Route-Policy 由一个或多个节点（Node）构成，Node 之间是“或”的关系。每个Node都有一个编号，路由项按照Node编号由小到大的顺序通过各个Node。每个Node下可以有若干个if-match和apply子句（特殊情况下可以完全没有if-match和apply子句），if-match之间是“与”的关系。If-match子句用来定义匹配规则，即路由项通过当前Node所需要满足的条件，匹配对象是路由项的某些属性，比如路由前缀、Next Hop、Cost、路由优先级等；apply子句用来规定处理动作。

Route-Policy的每个 Node都有相应的 permit模式或 deny模式。如果是 permit模式，则当路由项满足该Node的所有If-match子句时，就被允许通过该Node的过滤并执行该Node的apply子句，不再进入下一个Node；如果路由项没有满足该Node的所有If-match子句，则会进入下一个Node继续进行过滤。如果是deny模式，则当路由项满足该Node的所有If-match子句时，就被拒绝通过该Node的过滤，这时apply子句不会被执行，并且不进入下一个Node；否则就进入下一个Node继续进行过滤。

实验目的

- 掌握Route-Policy的基本配置方法
- 掌握使用Route-Policy进行路由过滤
- 掌握使用Route-Policy进行OSPF 路由属性的修改

实验内容

实验拓扑如图1-16所示，实验编址如表1-2所示。本实验中，R2、R3、R4为某公司总部的路由器，R1为合作方的路由器，R1与R2和R4之间运行RIPv2，R3与R2和R4之间运行OSPF。R1的Loopback 1、Loopback 2、Loopback 3、Loopback 4分别用来模拟合作方内部的4个网段。网络管理员希望通过配置路由策略来实现R3去往192.168.1.0/24网段和 192.168.3.0/24 网段的流量经由路径 R3-R2-R1，而去往192.168.2.0/24 网段和192.168.4.0/24网段的流量经由路径R3-R4-R1，并且这两条路径互为备份。

实验拓扑

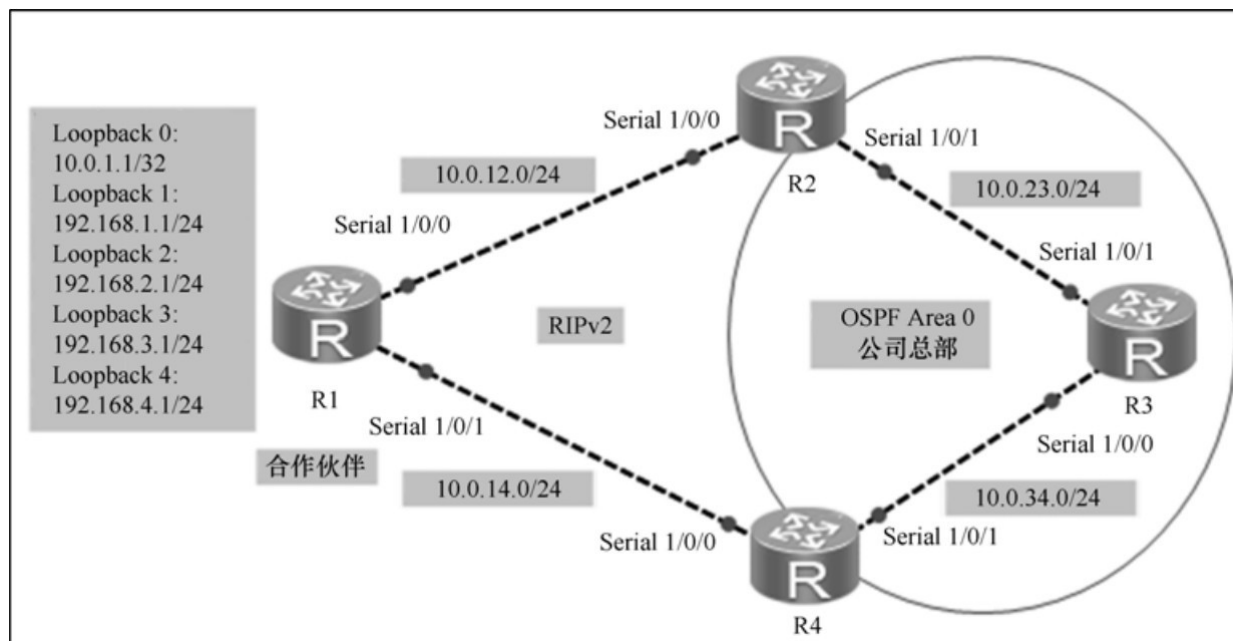


图1-16 基本的路由策略配置

实验编址表

表1-2 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	Serial 1/0/0	10.0.12.1	255.255.255.0	N/A
	Serial 1/0/1	10.0.14.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	192.168.1.1	255.255.255.0	N/A
	Loopback 2	192.168.2.1	255.255.255.0	N/A
	Loopback 3	192.168.3.1	255.255.255.0	N/A
	Loopback 4	192.168.4.1	255.255.255.0	N/A
R2(AR2220)	Serial 1/0/0	10.0.12.2	255.255.255.0	N/A
	Serial 1/0/1	10.0.23.1	255.255.255.0	N/A
R3(AR2220)	Serial 1/0/1	10.0.23.2	255.255.255.0	N/A
	Serial 1/0/0	10.0.34.2	255.255.255.0	N/A
R4(AR2220)	Serial 1/0/0	10.0.14.2	255.255.255.0	N/A
	Serial 1/0/1	10.0.34.1	255.255.255.0	N/A

实验步骤

1. 基本配置

根据图1-16和表1-2进行相应的基本配置，并使用ping命令检测R2与R1之间的连通性。

```
<R2>ping -c 1 10.0.12.1
```

```
PING 10.0.12.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.1: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
--- 10.0.12.1 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 60/60/60 ms
```

其余直连网段的连通性测试过程在此省略。

2. 搭建OSPF和RIP网络

R1与R2和R4之间运行RIPv2，R3与R2和R4之间运行OSPF。在R2和R4上将RIP路由引入到OSPF协议中。

```
[R1]rip
```

```
[R1-rip-1]version 2
```



```
[R1-rip-1]undo summary
[R1-rip-1]network 10.0.0.0
[R1-rip-1]network 192.168.1.0
[R1-rip-1]network 192.168.2.0
[R1-rip-1]network 192.168.3.0
[R1-rip-1]network 192.168.4.0
[R2]rip
[R2-rip-1]version 2
[R2-rip-1]undo summary
[R2-rip-1]network 10.0.0.0
[R2-rip-1]ospf 1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.23.1 0.0.0.0
[R3]ospf 1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.23.2 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 10.0.34.2 0.0.0.0
[R4]rip
[R4-rip-1]version 2
[R4-rip-1]undo summary
[R4-rip-1]network 10.0.0.0
[R4-rip-1]ospf 1
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.34.1 0.0.0.0
[R2]ospf 1
[R2-ospf-1]import-route rip 1
[R4]ospf 1
```

配置完成后，查看R3的IP路由表，检查R3是否接收到了RIP路由信息。

Route Flags: R - relay, D - download to fib

Routing Tables:Public									
Destinations : 19					Routes : 26				
Destination/Mask			Proto		PreCost		Flags		
NextHop		Interface							
10.0.1.1/32		O_ASE		150	1	D	10.0.34.1		
Serial1/0/0									
O_ASE		150	1	D	10.0.23.1		Serial1/0/1		
10.0.12.0/24		O_ASE		150	1	D	10.0.34.1		
Serial1/0/0									
O_ASE		150	1	D	10.0.23.1		Serial1/0/1		
10.0.14.0/24		O_ASE		150	1	D	10.0.34.1		
Serial1/0/0									
O_ASE		150	1	D	10.0.23.1		Serial1/0/1		
10.0.23.0/24		Direct		0	0	D	10.0.23.2		
Serial1/0/1									
.....									
127.255.255.255/32		Direct		0		0	D		
127.0.0.1		InLoopBack0							
192.168.1.0/24		O_ASE		150	1	D	10.0.34.1		
Serial1/0/0									

```

O_ASE 150 1 D 10.0.23.1 Serial1/0/1
192.168.2.0/24 O_ASE 150 1 D 10.0.34.1
Serial1/0/0
O_ASE 150 1 D 10.0.23.1 Serial1/0/1
192.168.3.0/24 O_ASE 150 1 D 10.0.34.1
Serial1/0/0
O_ASE 150 1 D 10.0.23.1 Serial1/0/1
192.168.4.0/24 O_ASE 150 1 D 10.0.34.1
Serial1/0/0
O_ASE 150 1 D 10.0.23.1 Serial1/0/1
255.255.255.255/32 Direct 0 0 D
127.0.0.1 InLoopBack0

```

可以看到，RIP进程中的路由已经被成功引进OSPF进程中。默认情况下，被引入到OSPF中的路由的Cost值为1，Cost Type为Type-2，协议优先级的值为150。还可以看到，由于在R2和R4上都进行了路由的引入，所以出现了路由冗余的现象，例如，从R3去往192.168.1.0/24时，下一跳可以是R4（10.0.34.1），也可以是R2（10.0.23.1）。

3. 使用Route-Policy对引入到OSPF进程的路由进行过滤和修改

默认情况下，引入路由的操作会将被引入协议的所有路由都引入到目标协议中；如果需要对引入的路由信息进行某些过滤处理，或者对引入的路由信息的某些属性进行修改，则可以使用Route-Policy。

现在要求从R3去往192.168.1.0/24和192.168.3.0/24这两个网段的流量经由路径R3-R2-R1，同时还要求这两个网段的路由在R2上被引入进OSPF时的Cost值为20，Cost Type为Type-1。另一方面，为了实现路由冗余，在R4上引入这两条路由时的Cost值为30，Cost Type为Type-

1。这样一来，当R3-R2-R1 这条路径失效时，便能使用R3-R4-R1这条备份路径。

为了实现上述需求，首先需要使用ACL将192.168.1.0/24和192.168.3.0/24这两个网段匹配出来。由于192.168.1.0/24和192.168.3.0/24的第3个字节都是奇数，所以可以通过如下的ACL通配符掩码来直接进行匹配。

```
[R2]acl 2000
```

```
[R2-acl-basic-2000]rule permit source 192.168.1.0 0.0.254.255
```

```
[R4]acl 2000
```

```
[R4-acl-basic-2000]rule permit source 192.168.1.0 0.0.254.255
```

创建Route-Policy，在R2上将192.168.1.0/24和192.168.3.0/24引入到OSPF时Cost设置为20，Cost Type设置为Type-1；在R4上将192.168.1.0/24和192.168.3.0/24引入到OSPF时Cost设置为30，Cost Type设置为Type-1。

```
[R2]route-policy import-ospf permit node 5
```

```
[R2-route-policy]if-match acl 2000
```

```
[R2-route-policy]apply cost 20
```

```
[R2-route-policy]apply cost-type type-1
```

```
[R4]route-policy import-ospf permit node 5
```

```
[R4-route-policy]if-match acl 2000
```

```
[R4-route-policy]apply cost 30
```

```
[R4-route-policy]apply cost-type type-1
```

在R2、R4上将RIP引入到OSPF时，应用Route-Policy。

```
[R2]ospf 1
```

```
[R2-ospf-1]import rip route-policy import-ospf
```

```
[R4]ospf 1
```

```
[R4-ospf-1]import rip route-policy import-ospf
```

配置完成后，查看R3的IP路由表。

[R3]display ip routing-table

Route Flags: R - relay, D - download to fib

RoutingTables:Public						
Destinations : 14 Routes : 14						
Destination/Mask		Proto	Pre	Cost	Flags	
NextHop	Interface					
10.0.23.0/24	Direct	0 0	D		10.0.23.2	
Serial1/0/1						
.....						
127.255.255.255/32	Direct	0 0	D			
127.0.0.1	InLoopBack0					
192.168.1.0/24	O_ASE	150 68	D		10.0.23.1	
Serial1/0/1						
192.168.3.0/24	O_ASE	150 68	D		10.0.23.1	
Serial1/0/1						
255.255.255.255/32	Direct	0 0	D			
127.0.0.1	InLoopBack0					

可以看到，R3 去往 192.168.1.0/24 和 192.168.3.0/24 的下一跳为 10.0.23.1，即R2。现在将R3 上的Serial 1/0/1 接口关闭，观察路径是否能切换到R3-R4-R1 上。

[R3]interface serial1/0/1

[R3-Serial1/0/1]shutdown

[R3]display ip routing-table

Route Flags: R - relay, D - download to fib

RoutingTables:Public						
Destinations : 10 Routes : 10						
Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
10.0.34.0/24	Direct	0	0	D	10.0.34.2	
Serial1/0/0						
.....						
127.255.255.255/32	Direct		0	0	D	
127.0.0.1	InLoopBack0					
192.168.1.0/24	O_ASE	150	78	D	10.0.34.1	
Serial1/0/0						
192.168.3.0/24	O_ASE	150	78	D	10.0.34.1	
Serial1/0/0						
255.255.255.255/32	Direct		0	0	D	
127.0.0.1	InLoopBack0					

可以看到，路径已切换到R3-R4-R1。重新开启R3 的Serial 1/0/1 接口，恢复网络路由选择。

```
[R3]interface serial1/0/1
```

```
[R3-Serial1/0/1]undo shutdown
```

现在要求从 R3 去往 192.168.2.0/24 和 192.168.4.0/24 这两个网段的流量经由路径R3-R4-R1，同时还要求这两个网段的路由在R4上被引进OSPF时的Cost值为20、Cost Type为Type-2。另一方面，为了实现路由冗余，在R2上引入这两条路由时的Cost值为30、Cost Type 为Type-2。这样一来，当R3-R4-R1 这条路径失效时，便能使用R3-R2-R1这条备份路径。

在R2、R4 上配置ACL 2001 , 匹配192.168.2.0/24 和192.168.4.0/24 这两个网段。

```
[R2]acl 2001
```

```
[R2-acl-basic-2001]rule permit source 192.168.2.0 0.0.254.255
```

```
[R4]acl 2001
```

```
[R4-acl-basic-2001]rule permit source 192.168.2.0 0.0.254.255
```

在R2、R4上添加新的策略Node。

```
[R2]route-policy import-ospf permit node 10
```

```
[R2-route-policy]if-match acl 2001
```

```
[R2-route-policy]apply cost 30
```

```
[R2-route-policy]apply cost-type type-2
```

```
[R4]route-policy import-ospf permit node 10
```

```
[R4-route-policy]if-match acl 2001
```

```
[R4-route-policy]apply cost 20
```

```
[R4-route-policy]apply cost-type type-2
```

配置完成后 , 查看R3的IP路由表。

```
<R3>display ip routing-table
```

Route Flags: R - relay, D - download to fib

RoutingTables:Public					
Destinations : 16		Routes : 16			
Destination/Mask	Proto	Pre	Cost	Flags	
NextHop	Interface				
10.0.23.0/24	Direct	0 0	D	10.0.23.2	
Serial1/0/1					
.....					

```

127.255.255.255/32  Direct      0  0  D      127.0.0.
InLoopBack0
192.168.1.0/24      O_ASE    150  68  D      10.0.23.1
Serial1/0/1
192.168.2.0/24      O_ASE    150  20  D      10.0.34.1
Serial1/0/0
192.168.3.0/24      O_ASE    150  68  D      10.0.23.1
Serial1/0/1
192.168.4.0/24      O_ASE    150  20  D      10.0.34.1
Serial1/0/0
255.255.255.255/32  Direct      0  0  D
127.0.0.1           InLoopBack0

```

可以看到，从R3去往192.168.2.0/24和192.168.4.0/24的流量会经由路径R3-R4-R1；当该路径失效后，会切换到R3-R2-R1，这个过程请读者自行进行验证。

细心的读者到这里可能已经发现，10.0.1.1/32 这条路由没有被引进 OSPF 协议中。为什么会出现这种现象呢？原因是 Route-Policy 存在一条默认规则：如果某条路由没有通过 Route-Policy 的任何 Node，则该条路由不会被引入。如果希望将 10.0.1.1/32 引进OSPF中，则需要在 Route-Policy中添加一个Node号最大的、模式为permit的Node，该Node下不需要定义任何if-match子句和apply子句，其含义是任何路由项都可以通过该Node。

```
[R2]route-policy import-ospf permit node 100
```

```
[R4]route-policy import-ospf permit node 100
```

配置完成后，查看R3的IP路由表。

```
[R3]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
10.0.1.1/32	O_ASE	150	1	D		10.0.23.1
Serial1/0/1						
O_ASE	150	1	D		10.0.34.1	Serial1/0/0
10.0.12.0/24	O_ASE	150	1	D		10.0.23.1
Serial1/0/1						
O_ASE	150	1	D		10.0.34.1	Serial1/0/0
10.0.14.0/24	O_ASE	150	1	D		10.0.23.1
Serial1/0/1						
O_ASE	150	1	D		10.0.34.1	Serial1/0/0
10.0.23.0/24	Direct	0	0	D		10.0.23.2
Serial1/0/1						
.....						
127.255.255.255/32	Direct		0	0		D
127.0.0.1	InLoopBack0					
192.168.1.0/24	O_ASE	150	68	D		10.0.23.1
Serial1/0/1						
192.168.2.0/24	O_ASE	150	20	D		10.0.34.1
Serial1/0/0						
192.168.3.0/24	O_ASE	150	68	D		10.0.23.1
Serial1/0/1						

```

192.168.4.0/24      O_ASE    150    20    D          10.0.34.1
Serial1/0/0
255.255.255.255/32  Direct      0        0        D
127.0.0.1          InLoopBack0

```

可以看到，此时R3已经能够接收到关于10.0.1.1/32的路由信息。另外，R1和R2互联的网段10.0.12.0/24以及R1和R4互联的网段10.0.14.0/24也被引进了OSPF中；由于这两个网段是互联网段，并没有承载业务，公司不希望把这两个网段引入到OSPF中，所以要求在路由引入的时候，过滤掉这两个网段。

要实现这个需求，还需要增加新的策略语句，将这两个互联网段明确拒绝引进OSPF协议中。使用前缀列表将这两个网段匹配出来，并在Route-Policy中增加新的Node，Node的模式为deny。

```

[R2]ip ip-prefix hcdp index 10 permit 10.0.12.0 24
[R2]ip ip-prefix hcdp index 20 permit 10.0.14.0 24
[R4]ip ip-prefix hcdp index 10 permit 10.0.12.0 24
[R4]ip ip-prefix hcdp index 20 permit 10.0.14.0 24
[R2]route-policy import-ospf deny node 15
[R2-route-policy]if-match ip-prefix hcdp
[R4]route-policy import-ospf deny node 15
[R4-route-policy]if-match ip-prefix hcdp

```

在R2上查看路由策略的配置情况。

```

[R2]display route-policy
Route-policy : import-ospf
  permit : 5 (matched counts: 40)
  Match clauses :
    if-match acl 2000
  Apply clauses :

```

```

        apply cost 20
        apply cost-type type-1
permit : 10 (matched counts: 40)
Match clauses :
    if-match acl 2001
Apply clauses :
    apply cost 30
    apply cost-type type-2
deny : 15 (matched counts: 6)
Match clauses :
    if-match ip-prefix hcdp

```

permit : 100 (matched counts: 34)

最后，在R3上查看IP路由表。

```
[R3]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public									
Destinations : 17 Routes : 18									
Destination/Mask		Proto		Pre		Cost		Flags	
NextHop	Interface								
10.0.1.1/32	O_ASE	150	1	D		10.0.34.1		Serial1/0/0	
	O_ASE	150	1	D		10.0.23.1		Serial1/0/1	
10.0.23.0/24	Direct	0	0	D		10.0.23.2		Serial1/0/1	
.....									
127.255.255.255/32	Direct		0	0		D		127.0.0.1	

InLoopBack0

192.168.1.0/24	O_ASE	150	68	D	10.0.23.1
Serial1/0/1					
192.168.2.0/24	O_ASE	150	20	D	10.0.34.1
Serial1/0/0					
192.168.3.0/24	O_ASE	150	68	D	10.0.23.1
Serial1/0/1					
192.168.4.0/24	O_ASE	150	20	D	10.0.34.1
Serial1/0/0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，关于10.0.12.0/24和10.0.14.0/24的路由信息已经消失，而10.0.1.1/32依然存在，表明需求已得到满足。

思考

定义ACL的规则时会用到permit一词，定义Route-Policy的Node时也会用到permit一词，这两处的permit各是什么含义？

1.3 控制RIP 路由的发布及路由引入

原理概述

RIP 协议是一种在现实网络中得到广泛应用的路由协议，相比于其他的路由协议，RIP最大的特点就是非常简单而且易于实现。

RIP协议中，每台RIP路由器都会定时向它的直连邻居发布它所知道的所有路由信息，同时又不断接收直连邻居发来的路由信息并以此更新自己的路由表，如此反复迭代，从而实现整个网络的路由收敛。

在实际中，人们常常会利用一些路由策略工具来对RIP路由器的路由发布加以控制，比如，可以利用Filter-Policy和ACL来对RIP路由器发

布的某些特定路由加以过滤。某些情况下，人们可能需要将一些外部路由信息引入到RIP进程中来，在这个过程中，同样可以利用一些路由策略工具来对引入到RIP进程中的路由进行过滤处理，从而实现某些特殊的网络需求。

实验目的

- 掌握使用Filter-Policy实现对RIP 路由发布的控制
- 加深对ACL 规则的匹配过程的理解

实验内容

实验拓扑如图1-17所示，实验编址如表1-3所示。本实验通过一个简单场景，介绍了如何使用Filter-Policy和ACL来控制RIP协议的路由发布，以及如何使用Route-Policy和ACL对引入到RIP中的路由进行过滤处理。

实验拓扑

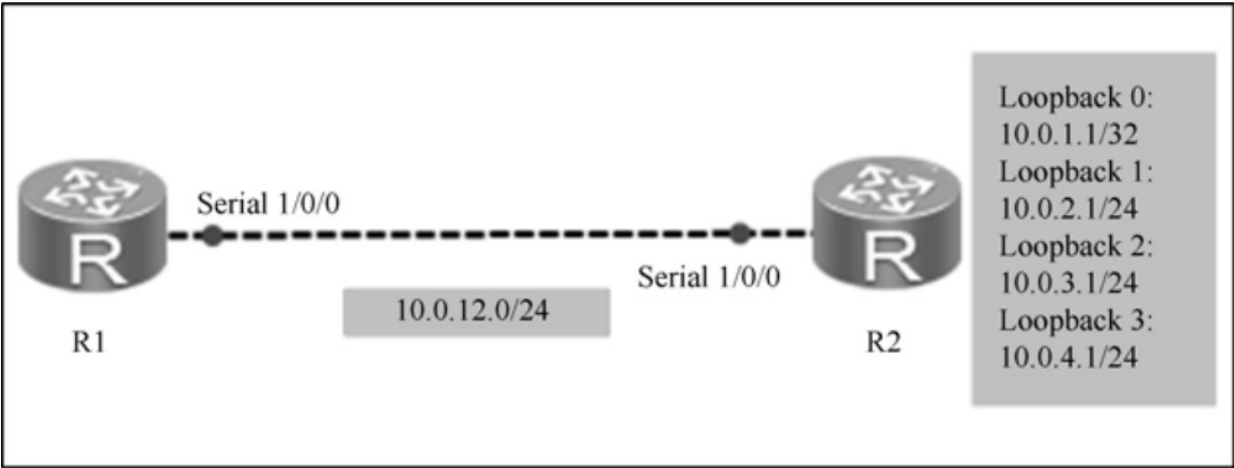


图1-17 控制RIP路由的发布及路由引入

实验编址表

表1-3 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	Serial 1/0/0	10.0.12.1	255.255.255.0	N/A
R2(AR2220)	Serial 1/0/0	10.0.12.2	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	10.0.2.1	255.255.255.0	N/A
	Loopback 2	10.0.3.1	255.255.255.0	N/A
	Loopback 3	10.0.4.1	255.255.255.0	N/A

实验步骤

1．基本配置

根据图1-17和表1-3进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 60/60/60 ms
```

2．使用Filter-Policy和ACL控制RIP协议的路由发布

在R1、R2上配置RIPv2路由协议，并在R2上引入所有Loopback接口的直连路由。

```
[R1]rip
```

```
[R1-rip-1]version 2
```

```
[R1-rip-1]undo summary
```

```
[R1-rip-1]network 10.0.0.0
```

```
[R2]rip
```

```
[R2-rip-1]version 2
```

```
[R2-rip-1]undo summary
```

[R2-rip-1]network 10.0.0.0

[R2-rip-1]import-route direct

查看R1的IP路由表。

[R1]display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 12 Routes : 12						
Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
10.0.12.0/24	Direct	0	0	D		10.0.12.1
Serial1/0/0						
10.0.12.1/32	Direct	0	0	D		127.0.0.1
Serial1/0/0						
10.0.12.2/32	Direct	0	0	D		10.0.12.2
Serial1/0/0						
10.0.12.255/32	Direct	0	0	D		127.0.0.1
Serial1/0/0						
10.0.1.1/32	RIP	100	1	D		10.0.12.2
Serial1/0/0						
10.0.2.0/24	RIP	100	1	D		10.0.12.2
Serial1/0/0						
10.0.3.0/24	RIP	100	1	D		10.0.12.2
Serial1/0/0						
10.0.4.0/24	RIP	100	1	D		10.0.12.2
Serial1/0/0						

```

127.0.0.0/8      Direct    0    0    D      127.0.0.1
InLoopBack0
127.0.0.1/32     Direct    0    0    D      127.0.0.1
InLoopBack0
127.255.255.255/32 Direct      0    0    D
127.0.0.1      InLoopBack0
255.255.255.255/32 Direct      0    0    D
127.0.0.1      InLoopBack0

```

可以看到，R1已经接收到关于R2上Loopback接口的路由。

现在，在R2上利用Filter-Policy和ACL，禁止R2向R1发布10.0.1.1/32和10.0.3.0/24这两条路由信息。

```

[R2]acl 2000
[R2-acl-basic-2000]rule deny source 10.0.1.0 0.0.254.255
[R2-acl-basic-2000]rule permit source any
[R2]rip
[R2-rip-1]filter-policy 2000 export Serial 1/0/0

```

配置完成后，查看R1的IP路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```

-----
Routing Tables: Public
Destinations : 10  Routes : 10

Destination/Mask    Proto      Pre  Cost           Flags
NextHop            Interface
10.0.12.0/24        Direct     0    0             D      10.0.12.1
Serial1/0/0

```


10.0.12.1/32	Direct	0	0	D	127.0.0.1
Serial1/0/0					
10.0.12.2/32	Direct	0	0	D	10.0.12.2
Serial1/0/0					
10.0.12.255/32	Direct	0	0	D	127.0.0.1
Serial1/0/0					
10.0.2.0/24	RIP	100	1	D	10.0.12.2
Serial1/0/0					
10.0.4.0/24	RIP	100	1	D	10.0.12.2
Serial1/0/0					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct		0	0	D
127.0.0.1	InLoopBack0				
255.255.255.255/32	Direct		0	0	D
127.0.0.1	InLoopBack0				

可以看到，R1的路由表中10.0.1.1/32和10.0.3.0/24这两条路由已经消失，其他路由仍然存在。

3．配置静态路由

在R1上配置静态路由，用以模拟外部网络。

```
[R1]ip route-static 1.1.1.1 255.255.255.255 NULL 0
```

```
[R1]ip route-static 1.1.1.0 255.255.255.0 NULL 0
```

```
[R1]ip route-static 1.1.1.0 255.255.255.128 NULL 0
```

```
[R1]ip route-static 1.1.0.0 255.255.0.0 NULL 0
```

```
[R1]ip route-static 1.0.0.0 255.0.0.0 NULL 0
```

配置完成后，查看R1的IP路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 15 Routes : 15						
Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
1.0.0.0/8	Static	60	0	D	0.0.0.0	NULL0
1.1.0.0/16	Static	60	0	D	0.0.0.0	NULL0
1.1.1.0/24	Static	60	0	D	0.0.0.0	NULL0
1.1.1.0/25	Static	60	0	D	0.0.0.0	NULL0
1.1.1.1/32	Static	60	0	D	0.0.0.0	NULL0
10.0.12.0/24	Direct	0	0	D	10.0.12.1	

Serial1/0/0

.....

可以看到，在R1上配置的静态路由已经生效。

4．使用Route-Policy和ACL对引入到RIP中的路由进行过滤处理

现在需要在R1上将静态路由引入到RIP进程中，但只允许1.1.0.0/16被引入。

使用ACL匹配1.1.0.0/16，规则号的步长设置为10。

[R1]acl 2000

[R1-acl-basic-2000]step 10

[R1-acl-basic-2000]rule permit source 1.1.0.0 0.0.255.255

配置Route-Policy。

[R1]route-policy import-rip permit node 10

```
[R1-route-policy]if-match acl 2000
引入静态路由，并应用Route-Policy对引入的静态路由进行控制。
[R1]rip
[R1-rip-1]import-route static route-policy import-rip
配置完成后，查看R2的IP路由表。
<R2>display ip routing-table
Route Flags: R - relay, D - download to fib
```

Routing Tables: Public						
Destinations : 22 Routes : 22						
Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
1.1.0.0/16	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.0/24	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.0/25	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.1/32	RIP	100	1	D		10.0.12.1
Serial1/0/0						
10.0.12.0/2	Direct	0	0	D		10.0.12.2
Serial1/0/0						
.....						

可以看到，1.1.0.0/16已被引入到RIP进程中，但是，1.1.1.0/24，1.1.1.0/25，1.1.1.1/32也被引进了RIP。这表明虽然1.1.0.0/16匹配了ACL 2000的规则，但同时1.1.1.0/24、1.1.1.0/25、1.1.1.1/32也匹配了

ACL 2000 的规则。造成这一问题的原因是ACL 中的通配符掩码使用不当。注意，通配符掩码中的“0”所对应的位是必须匹配的，而“1”所对应的位可忽略。

在R1 上查看ACL 2000。

```
[R1]display acl 2000
```

```
Basic ACL 2000, 1 rule
```

```
ACL's step is 10
```

```
rule 10 permit source 1.1.0.0 0.0.255.255 (4 times matched)
```

ACL 2000 的 rule 10 的意思是，只要路由项的前 16 比特为 0000000100000001 就算匹配，因此，1.1.1.0/24、1.1.1.0/25、1.1.1.1/32 也都是匹配这条规则的。若要唯一地匹配1.1.0.0/16，应该使用通配符掩码0.0.0.0。

在R1上重新配置ACL。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]undo rule 10
```

```
[R1-acl-basic-2000]rule 10 permit source 1.1.0.0 0.0.0.0
```

查看修改之后的ACL 2000。

```
[R1]display acl 2000
```

```
Basic ACL 2000, 1 rule
```

```
ACL's step is 10
```

```
rule 10 permit source 1.1.0.0 0 (1 times matched)
```

查看R2的IP路由表。

```
[R2]display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

Routing Tables: Public

		Destinations : 19		Routes : 19		
Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
1.1.0.0/16	RIP	100	1	D		10.0.12.1
Serial1/0/0						
10.0.12.0/24	Direct	0	0	D		10.0.12.2
Serial1/0/0						

.....

可以看到，现在R2只接收到了1.1.0.0/16这条路由，而1.0.0.0/8、1.1.1.0/24、1.1.1.0/25、1.1.1.1/32在被引入到RIP时已经被过滤掉了。

现在有个新的需求，就是将1.1.1.0/24和1.1.1.0/25也引入到RIP中。经过初步分析可知，ACL的规则中的通配符掩码应该为0.0.0.127。

在ACL 2000 中添加一条规则。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]rule permit source 1.1.1.0 0.0.0.127
```

配置完成后，查看ACL 2000。

```
[R1]display acl 2000
```

```
Basic ACL 2000, 2 rules
```

```
ACL's step is 10
```

```
rule 10 permit source 1.1.0.0 0 (10 times matched)
```

```
rule 20 permit source 1.1.1.0 0.0.0.127 (10 times matched)
```

查看R2的IP路由表。

```
<R2>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

Routing Tables: Public

		Destinations : 22		Routes : 22		
Destination/Mask		Proto	Pre	Cost	Flags	
NextHop	Interface					
1.1.0.0/16	RIP	100	1	D	10.0.12.1	
Serial1/0/0						
1.1.1.0/24	RIP	100	1	D	10.0.12.1	
Serial1/0/0						
1.1.1.0/25	RIP	100	1	D	10.0.12.1	
Serial1/0/0						
1.1.1.1/32	RIP	100	1	D	10.0.12.1	
Serial1/0/0						
10.0.12.0/24	Direct	0	0	D	10.0.12.2	
Serial1/0/0						

.....

可以看到，1.1.1.0/24和1.1.1.0/25都已被引进RIP中，然而，1.1.1.1/32也一起被引进RIP中。经过仔细分析发现，1.1.1.1/32这条路由也是匹配rule 20的。

修改ACL的配置，拒绝关于1.1.1.1/32的路由。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]rule deny source 1.1.1.1 0
```

配置完成后，查看R2的IP路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 22 Routes : 22

Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
1.1.0.0/16	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.0/24	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.0/25	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.1/32	RIP	100	1	D		10.0.12.1
Serial1/0/0						
10.0.12.0/24	Direct	0	0	D		10.0.12.2
Serial1/0/0						

.....

可以看到，路由1.1.1.1/32依然还在路由表中。

再次查看ACL 2000。

```
[R1]display acl 2000
```

```
Basic ACL 2000, 3 rules
```

```
Acl's step is 10
```

```
rule 10 permit source 1.1.0.0 0 (3 matches)
```

```
rule 20 permit source 1.1.1.0 0.0.0.127 (6 matches)
```

```
rule 30 deny source 1.1.1.1 0
```

可以看到，按照ACL的匹配顺序，1.1.1.1/32在匹配rule 30之前就已经匹配了rule 20，所以rule 30并没有起到过滤1.1.1.1/32的作用。正确的过程应该是让1.1.1.1/32先匹配到rule 30。现在，将rule 30删除，重新配置一个规则，并使规则的序列号小于20。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]undo rule 30
```

```
[R1-acl-basic-2000]rule 15 deny source 1.1.1.1 0
```

配置完成后，查看ACL，并查看R2的IP路由表。

```
[R1]display acl 2000
```

Basic ACL 2000, 3 rules

Acl's step is 10

```
rule 10 permit source 1.1.0.0 0 (5 matches)
```

```
rule 15 deny source 1.1.1.1 0 (1 matches)
```

```
rule 20 permit source 1.1.1.0 0.0.0.127 (11 matches)
```

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 21 Routes : 21						
Destination/Mask		Proto		Pre	Cost	Flags
NextHop	Interface					
1.1.0.0/16	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.0/24	RIP	100	1	D		10.0.12.1
Serial1/0/0						
1.1.1.0/25	RIP	100	1	D		10.0.12.1
Serial1/0/0						
10.0.12.0/24	Direct	0	0	D		10.0.12.2
Serial1/0/0						
.....						

可以看到，R2的IP路由表中已经没有了路由1.1.1.1/32，原因是在R1上引入路由时它已经被过滤掉了。

思考

通配符掩码和反掩码（ Inverted Subnet Mask ）实质上是同一个概念吗？

第2章 OSPF

- 2.1 OSPF基本配置
- 2.2 OSPF邻居邻接关系
- 2.3 OSPF链路状态数据库
- 2.4 OSPF Stub区域
- 2.5 OSPF NSSA区域
- 2.6 OSPF虚链路
- 2.7 OSPF网络类型
- 2.8 OSPF路由聚合
- 2.9 OSPF监测和调试
- 2.10 OSPF缺省路由
- 2.11 OSPF故障排除

2.1 OSPF 基本配置

原理概述

OSPF 是一种应用非常广泛的基于链路状态的动态路由协议，它具有区域（Area）化的层次结构，扩展性好，收敛速度快，适合部署在各种规模的网络上。

在OSPF中，每台路由器都必须有一个Router-ID来标示自己。为了使OSPF网络更加稳定可靠，路由器通常会启用Loopback接口，并配置特

定的IP地址，且将此地址作为自己的Router-ID。

OSPF协议定义了4种不同的网络类型，分别为广播网络（也称为Broadcast网络）、NBMA（Non-Broadcast Multi-Access）网络、点到点网络（也称为Point-to-Point网络，或P2P网络）和点到多点网络（也称为Point-to-Multipoint网络，或P2MP网络）。在广播网络和NBMA网络中，需要选举出DR（Designated Router）和BDR（Backup Designated Router）。DR和BDR是通过路由器接口的DR优先级来决定的，优先级最高的路由器将当选为DR，次之者当选为BDR；如果接口的DR优先级相同，则具有最高Router-ID的路由器将当选为DR，次之者当选为BDR。

实验目的

- 掌握OSPF 的基本配置
- 观察并理解DR/BDR 的选举过程
- 掌握OSPF 接口开销的修改方法
- 理解OSPF 被动接口的作用
- 掌握OSPF 认证功能的配置

实验内容

实验拓扑如图2-1所示，实验编址如表2-1所示。本实验模拟了一个企业网络场景，R1为企业总部的路由器，R2为地区总部A的路由器，R3为地区总部B的路由器，R4和R5分别为分支机构1和分支机构2的路由器。整个网络都运行OSPF，其中R1与R2和R3之间的链路位于区域0中，R2与R3之间的链路作为一条备份链路也位于区域0中，R2与R4之间的链路位于区域1中，R3与R5之间的链路位于区域2中。R4、R5、R1的Loopback 1接口模拟了不同分支机构内以及企业总部内的网络。通过正确的OSPF配置后，不同分支机构的网络之间以及分支机构与企业总部的网络之间都应实现正常通信。

实验拓扑

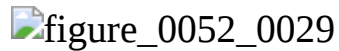
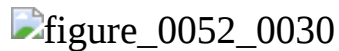


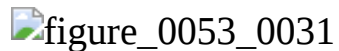
图2-1 OSPF基本配置

实验编址表

表2-1 实验编址



(续表)



实验步骤

1. 基本配置

根据图2-1和表2-1进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=90 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 90/90/90 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF路由协议

配置OSPF协议，其中R1、R2、R3之间的链路位于区域0，R2与R4之间的链路位于区域1，R3与R5之间的链路位于区域2，每台路由器均使用Loopback 0 接口的IP地址作为自己的Router-ID。

```
[R1]router id 1.1.1.1
```

```
[R1]ospf
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.13.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 1.1.1.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 11.11.11.11 0.0.0.0
```

```
[R2]router id 2.2.2.2
```

```
[R2]ospf
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.12.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.23.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]network 2.2.2.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]area 1
```

```
[R2-ospf-1-area-0.0.0.1]network 10.0.24.2 0.0.0.0
```

```
[R3]router id 3.3.3.3
```

```
[R3]ospf
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.13.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.23.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.0]network 3.3.3.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.0]area 2
```

```
[R3-ospf-1-area-0.0.0.2]network 10.0.35.3 0.0.0.0
```

```
[R4]router id 4.4.4.4
```

```
[R4]ospf
```

```
[R4-ospf-1]area 1
```

```
[R4-ospf-1-area-0.0.0.1]network 10.0.24.4 0.0.0.0
```

```
[R4-ospf-1-area-0.0.0.1]network 4.4.4.4 0.0.0.0
```

```
[R4-ospf-1-area-0.0.0.1]network 44.44.44.44 0.0.0.0
```

```
[R5]router id 5.5.5.5
```

```
[R5]ospf
```

```
[R5-ospf-1]area 2
```

```
[R5-ospf-1-area-0.0.0.2]network 10.0.35.5 0.0.0.0
```

```
[R5-ospf-1-area-0.0.0.2]network 5.5.5.5 0.0.0.0
```

```
[R5-ospf-1-area-0.0.0.2]network 55.55.55.55 0.0.0.0
```

配置完成后，查看R1、R2、R3上OSPF邻居的建立情况。

<R1>displayospf peer brief

OSPF Process 1 with Router ID 1.1.1.1

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	2.2.2.2	Full
0.0.0.0	GigabitEthernet0/0/2	3.3.3.3	Full

<R2>displayospf peer brief

OSPF Process 1 with Router ID 2.2.2.2

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	1.1.1.1	Full
0.0.0.0	Serial2/0/0	3.3.3.3	Full
0.0.0.1	Ethernet1/0/0	4.4.4.4	Full

<R3>displayospf peer brief

OSPF Process 1 with Router ID 3.3.3.3

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	1.1.1.1	Full
0.0.0.0	Serial2/0/0	2.2.2.2	Full
0.0.0.2	Ethernet1/0/0	5.5.5.5	Full

可以看到，OSPF邻居状态都为Full，表明各邻居关系都已成功建立。

查看R1的IP路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
-----
```

Routing Tables: Public

Destinations : 18 Routes : 19

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
1.1.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
2.2.2.2/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.12.2
3.3.3.3/32 GigabitEthernet0/0/2	OSPF	10	1	D	10.0.13.3
4.4.4.4/32 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.12.2
5.5.5.5/32 GigabitEthernet0/0/2	OSPF	10	2	D	10.0.13.3

10.0.12.0/24	Direct	0	0	D	10.0.12.1	
GigabitEthernet0/0/1						
10.0.12.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.13.0/24	Direct	0	0	D	10.0.13.1	
GigabitEthernet0/0/2						
10.0.13.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/2						
10.0.23.0/24	OSPF	10	1563	D	10.0.12.2	
GigabitEthernet0/0/1						
OSPF	10	1563	D	10.0.13.3	GigabitEthernet0/0/2	
10.0.24.0/24	OSPF	10	2	D	10.0.12.2	
GigabitEthernet0/0/1						
10.0.35.0/24	OSPF	10	2	D	10.0.13.3	
GigabitEthernet0/0/2						
11.11.11.0/24	Direct	0	0	D	11.11.11.11	LoopBack1
11.11.11.11/32	Direct	0	0	D	127.0.0.1	LoopBack1
44.44.44.44/32	OSPF	10	2	D	10.0.12.2	
GigabitEthernet0/0/1						
55.55.55.55/32	OSPF	10	2	D	10.0.13.3	
GigabitEthernet0/0/2						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0

可以看到，企业总部路由器R1已经获得了其他路由器的接口所在网段的路由。

查看R4的IP路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
1.1.1.1/32	OSPF	10	2	D	10.0.24.2 Ethernet1/0/0
2.2.2.2/32	OSPF	10	1	D	10.0.24.2 Ethernet1/0/0
3.3.3.3/32	OSPF	10	3	D	10.0.24.2 Ethernet1/0/0
4.4.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack0
5.5.5.5/32	OSPF	10	4	D	10.0.24.2 Ethernet1/0/0
10.0.12.0/24	OSPF	10	2	D	10.0.24.2 Ethernet1/0/0
10.0.13.0/24	OSPF	10	3	D	10.0.24.2 Ethernet1/0/0
10.0.23.0/24	OSPF	10	1563	D	10.0.24.2 Ethernet1/0/0
10.0.24.0/24	Direct	0	0	D	10.0.24.4 Ethernet1/0/0
10.0.24.4/32	Direct	0	0	D	127.0.0.1 Ethernet1/0/0
10.0.35.0/24	OSPF	10	4	D	10.0.24.2 Ethernet1/0/0
11.11.11.11/32	OSPF	10	2	D	10.0.24.2 Ethernet1/0/0

44.44.44.0/24	Direct	0	0	D	44.44.44.44	LoopBack1
44.44.44.44/32	Direct	0	0	D	127.0.0.1	LoopBack1
55.55.55.55/32	OSPF	10	4	D	10.0.24.2	Ethernet1/0/0
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0

可以看到，分支机构1的路由器R4已经获得了企业总部、地区总部以及分支机构2的所有网段的路由。R5的路由表请读者自行查看。至此，所有分支机构和企业总部及地区总部的路由已经实现了互通，相互可以进行通信了。

3．查看DR/BDR选举情况

在DR/BDR的选举过程中，首先比较的是路由器接口的DR优先级，优先级最高的路由器将被选为DR，次之者为BDR，其余的为DRother。DR优先级默认值为1，如果为0则代表不参与选举。如果接口的DR优先级相同，则比较路由器的Router-ID，数值最大的为DR，次之者为BDR，其余的为DRother。

在R1、R2上查看DR/BDR的选举情况。

```
<R1>display ospf interface
```

```
OSPF Process 1 with Router ID 1.1.1.1
```

```
Interfaces
```

```
Area: 0.0.0.0(MPLS TE not enabled)
```

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.12.1	Broadcast	BDR	1	1	10.0.12.2	
10.0.12.1						

10.0.13.1 10.0.13.1	Broadcast	BDR	1	1	10.0.13.3	
1.1.1.1	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0
11.11.11.11	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0

<R2>display ospf interface

OSPF Process 1 with Router ID 2.2.2.2

Interfaces

Area: 0.0.0.0 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.12.2	Broadcast	DR	1	1	10.0.12.2	10.0.12.1
2.2.2.2	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0
10.0.23.2	P2P	P-2-P	1562	1	0.0.0.0	0.0.0.0

Area: 0.0.0.1 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.24.2 10.0.24.2	Broadcast	BDR	1	1	10.0.24.4	

Loopback接口所在的网段都默认为是点到点网络，可以观察到，在点到点网络上是没有选举 DR/BDR 的。10.0.12.0/24 为以太网段，默认为是广播网络，所以需要选举DR/BDR。由于采用了默认配置，所以R1的GE 0/0/1 和R2的GE 0/0/0接口的DR 优先级的值都为 1，故需比较 Router-ID。因此，最终 R2 因为 Router-ID 较大而当选为10.0.12.0/24网段上的DR，R1则为BDR。另外，还可以看到，在10.0.24.0/24网段上，R2当选为该网段上的BDR，R4为DR。

现在，修改R1 的GE 0/0/1 的DR 优先级的值为2，希望使R1 成为10.0.12.0/24 网段的DR，R2成为BDR。

```
[R1]interface GigabitEthernet0/0/1
```

```
[R1-GigabitEthernet0/0/1]ospf dr-priority 2
```

在R1上重新查看DR/BDR的选举情况。

```
<R1>display ospf interface
```

OSPF Process 1 with Router ID 1.1.1.1

Interfaces

Area: 0.0.0.0 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.12.1	Broadcast	BDR	1	2	10.0.12.2	10.0.12.1
10.0.13.1	Broadcast	BDR	1	1	10.0.13.3	10.0.13.1
1.1.1.1	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0
11.11.11.11	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0

观察发现，R1并没有变为DR，这是因为为了维持OSPF网络的稳定性，DR/BDR的选举不具有抢占性。将R1的GE 0/0/1 接口先手动关闭，然后再开启，或者重启OSPF进程，才能使得DR/BDR重新进行选举。

下面重启OSPF进程。

```
<R1>reset ospf 1 process
```

在R1上重新查看DR/BDR的选举情况。

```
<R1>display ospf interface
```

OSPF Process 1 with Router ID 1.1.1.1

Interfaces

Area: 0.0.0.0 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.12.1	Broadcast	DR	1	2	10.0.12.1	10.0.12.2
10.0.13.1	Broadcast	BDR	1	1	10.0.13.3	10.0.13.1
1.1.1.1	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0
11.11.11.11	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0

可以看到，R1现在已成为了10.0.12.0/24网段上的DR。

4．配置OSPF的接口开销值

查看R2的路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 19 Routes : 19

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
1.1.1.1/32 GigabitEthernet0/0/0	OSPF	10	1	D	10.0.12.1
2.2.2.2/32	Direct	0	0	D	127.0.0.1 LoopBack0

3.3.3.3/32	OSPF	10	2	D	10.0.12.1	
GigabitEthernet0/0/0						
4.4.4.4/32	OSPF	10	1	D	10.0.24.4	Ethernet1/0/0
5.5.5.5/32	OSPF	10	3	D	10.0.12.1	
GigabitEthernet0/0/0						
10.0.12.0/24	Direct	0	0	D	10.0.12.2	
GigabitEthernet0/0/0						
10.0.12.2/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.13.0/24	OSPF	10	2	D	10.0.12.1	
GigabitEthernet0/0/0						
10.0.23.0/24	Direct	0	0	D	10.0.23.2	Serial2/0/0
10.0.23.2/32	Direct	0	0	D	127.0.0.1	Serial2/0/0
10.0.23.3/32	Direct	0	0	D	10.0.23.3	Serial2/0/0
10.0.24.0/24	Direct	0	0	D	10.0.24.2	Ethernet1/0/0
10.0.24.2/32	Direct	0	0	D	127.0.0.1	Ethernet1/0/0
10.0.35.0/24	OSPF	10	3	D	10.0.12.1	
GigabitEthernet0/0/0						
11.11.11.11/32	OSPF	10	1	D	10.0.12.1	
GigabitEthernet0/0/0						
44.44.44.44/32	OSPF	10	1	D	10.0.24.4	Ethernet1/0/0
55.55.55.55/32	OSPF	10	3	D	10.0.12.1	
GigabitEthernet0/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0

```
127.0.0.1/32    Direct    0    0    D    127.0.0.1    InLoopBack0
```

可以看到，R2 访问 R3 和 R5 的 Loopback 接口所在网段的路由的下一跳都为 R1（10.0.12.1），而不是R3，其原因相信读者都是很清楚的。

使用tracert命令来测试报文从R2到目的地址55.55.55.55所经过的路径。

```
<R2>tracert 55.55.55.55
```

```
tracert to 55.55.55.55(55.55.55.55), max hops: 30 ,packet length:
40,press CTRL_C to break
```

```
1 10.0.12.1 40 ms 30 ms 40 ms
```

```
2 10.0.13.3 30 ms 70 ms 40 ms
```

```
3 10.0.35.5 80 ms 100 ms 120 ms
```

可以看到，报文会经过R1（10.0.12.1）再到R3（10.0.13.3），然后到达R5。

在R2上查看接口的OSPF开销值。

```
<R2>display ospf interface
```

```
OSPF Process 1 with Router ID 2.2.2.2
```

```
Interfaces
```

```
Area: 0.0.0.0          (MPLS TE not enabled)
```

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.23.2	P2P	P-2-P	1562	1	0.0.0.0	0.0.0.0
10.0.12.2	Broadcast	BDR	1	1	10.0.12.1	10.0.12.2
2.2.2.2	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0

Area: 0.0.0.1 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.24.2	Broadcast	BDR	1	1	10.0.24.4	10.0.24.2

从上面的显示信息可以看到，R2 的Serial 2/0/0 接口（10.0.23.2）的Cost值为1562，GE 0/0/0 接口（10.0.12.2）的Cost 值为1。现在，将R2 的GE 0/0/0 接口的Cost 值修改为2000，将R3 的GE 0/0/0 接口的Cost 值也修改为2000。

```
[R2]interface GigabitEthernet0/0/0
```

```
[R2-GigabitEthernet0/0/0]ospf cost 2000
```

```
[R3]interface GigabitEthernet0/0/0
```

```
[R3-GigabitEthernet0/0/0]ospf cost 2000
```

修改完成后，在R2上重新查看接口的Cost值。

```
[R2]display ospf interface
```

OSPF Process 1 with Router ID 2.2.2.2

Interfaces

Area: 0.0.0.0 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.23.2	P2P	P-2-P	1562	1	0.0.0.0	0.0.0.0
10.0.12.2 10.0.12.2	Broadcast	BDR	2000	1	10.0.12.1	
2.2.2.2	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0

Area: 0.0.0.1 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.24.2	Broadcast	BDR	1	1	10.0.24.4	10.0.24.2

可以看到，R2 的GE 0/0/0 接口的Cost 值已修改为2000。再次在R2 上使用tracert命令来测试报文从R2到目的地址55.55.55.55所经过的路径。

```
<R2>tracert 55.55.55.55
```

```
traceroute to 55.55.55.55(55.55.55.55), max hops: 30 ,packet length:
40,press CTRL_C to break
```

```
1 10.0.23.3 30 ms 30 ms 50 ms
```

```
2 10.0.35.5 80 ms 50 ms 40 ms
```

可以看到，报文不再经过R1，而是直接经过R3（10.0.23.3）到达R5；这也验证了Cost越小，路由越优的原则。

5．配置OSPF被动接口

如果一个OSPF路由器的某一接口被配置为被动接口（Passive Interface），则该接口将不会发送和接收OSPF报文。例如，将R2的Ethernet 1/0/0接口配置为被动接口，如下。

```
[R2]ospf
```

```
[R2-ospf-1]silent-interface Ethernet1/0/0
```

配置完成后，设备会弹出如下信息。

```
Apr 11 2013 21:39:59-08:00 R2 %%01OSPF/3/NBR_CHG_DOWN(1)
[25]:Neighbor
```

```
event:neighbor state changed to Down. (ProcessId=1,
NeighborAddress=4.4.4.4,
```

NeighborEvent=KillNbr, NeighborPreviousState=Full,
NeighborCurrentState=Down)

Apr 11 2013 21:39:59-08:00 R2 %%01OSPF/3/NBR_DOWN_REASON(l)
[26]:Neighbor state

leaves full or changed to Down. (ProcessId=1, NeighborRouterId=4.4.4.4,

NeighborA

reaId=1,NeighborInterface=Ethernet0/0/1,NeighborDownImmediate

reason=NeighborDownDuetoKillNeighbor,NeighborDownPrimeReason=Pa
ssive

Interface Down, NeighborChangeTime=2013-04-11 21:39:59-08:00)

Apr 11 2013 21:40:03-08:00 R2 DS/4/DATASYNC_CFGCHANGE:OID

1.3.6.1.4.1.2011.5.25.191.3.1 configurations have been changed. The current
change number is 1, the change loop count is 0, and the maximum number
of records

is 4095.

从以上信息可知，R2与R4的邻居关系已经由Full状态变为了Down状
态。为了验证这一变化，查看此时R2的OSPF邻居建立情况。

<R2>display ospf peer brief

OSPF Process 1 with Router ID 2.2.2.2

Peer Statistic Information

Area Id	Interface	Neighbor id	State

0.0.0.0 GigabitEthernet0/0/0 1.1.1.1 Full

0.0.0.0 Serial2/0/0 3.3.3.3 Full

可以看到，R2未与R4建立OSPF邻居关系，但仍与R1和R3保持了正常的邻居关系。查看路由器 R2、R4 的 IP 路由表（请先恢复 R2 和 R3 的 GE 0/0/0 接口的 Cost值为1）。

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
1.1.1.1/32 GigabitEthernet0/0/0	OSPF	10	1	D	10.0.12.1
2.2.2.2/32	Direct	0	0	D	127.0.0.1 LoopBack0
3.3.3.3/32 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.12.1
5.5.5.5/32 GigabitEthernet0/0/0	OSPF	10	3	D	10.0.12.1
10.0.12.0/24 GigabitEthernet0/0/0	Direct	0	0	D	10.0.12.2
10.0.12.2/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1

10.0.13.0/24	OSPF	10	2	D	10.0.12.1	
GigabitEthernet0/0/0						
10.0.23.0/24	Direct	0	0	D	10.0.23.2	Serial2/0/0
10.0.23.2/32	Direct	0	0	D	127.0.0.1	Serial2/0/0
10.0.23.3/32	Direct	0	0	D	10.0.23.3	Serial2/0/0
10.0.24.0/24	Direct	0	0	D	10.0.24.2	Ethernet1/0/0
10.0.24.2/32	Direct	0	0	D	127.0.0.1	Ethernet1/0/0
10.0.35.0/24	OSPF	10	3	D	10.0.12.1	
GigabitEthernet0/0/0						
11.11.11.11/32	OSPF	10	1	D	10.0.12.1	
GigabitEthernet0/0/0						
55.55.55.55/32	OSPF	10	3	D	10.0.12.1	
GigabitEthernet0/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 7 Routes : 7

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

4.4.4.4/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.24.0/24	Direct	0	0	D	10.0.24.4	Ethernet1/0/0
10.0.24.4/32	Direct	0	0	D	127.0.0.1	Ethernet1/0/0
44.44.44.0/24	Direct	0	0	D	44.44.44.44	LoopBack1
44.44.44.44/32	Direct	0	0	D	127.0.0.1	LoopBack1
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0

可以看到，地区总部A的路由器R2上不再有分支机构1的网络的路由，分支机构1的路由器R4上也没有所有其他网络的路由了，只有直连网络的路由。

6．配置OSPF的认证功能

OSPF 的认证功能的配置可以是基于区域的，也可以是基于接口的。接下来，先在R1上配置基于区域0的认证功能，采用简单的明文方式。

```
[R1]ospf 1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]authentication-mode simple plain Huawei
```

配置完成后，在R1上查看OSPF邻居信息。

```
<R1>display ospf peer brief
```

```
OSPF Process 1 with Router ID 1.1.1.1
```

```
Peer Statistic Information
```

```
-----
-----
```

Area Id	Interface	Neighbor id	State

可以看到，R1现在没有任何OSPF邻居，这是因为R2、R3上还没有配置相匹配的认证功能。R2和R3都未能通过R1的认证，所以R1不会与R2和R3建立邻居关系。

在R2、R3上进行相应的认证功能的配置。

```
[R2]ospf 1
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]authentication-mode simple plain Huawei
```

```
[R3]ospf 1
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]authentication-mode simple plain Huawei
```

配置完成后，再在R1上查看OSPF邻居信息。

```
<R1>display ospf peer brief
```

```
OSPF Process 1 with Router ID 1.1.1.1
```

```
Peer Statistic Information
```

```
-----  
-----
```

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	2.2.2.2	Full
0.0.0.0	GigabitEthernet0/0/2	3.3.3.3	Full

可以看到，R1与R2、R1与R3之间的邻居关系已正常建立。

接下来，在R2 上配置基于Ethernet 1/0/0 接口的认证，并采用MD5 密文方式。

```
[R2]interface Ethernet1/0/0
```

```
[R2-Ethernet1/0/0]ospf authentication-mode md5 24 cipher Huawei
```

配置完成后，在R2上查看OSPF邻居信息。

```
<R2>display ospf peer brief
```

```
OSPF Process 1 with Router ID 2.2.2.2
```

```
Peer Statistic Information
```

```
-----  
-----
```

Area Id	Interface	Neighbor id	State
0.0.0.0	Serial2/0/0	3.3.3.3	Full
0.0.0.0	GigabitEthernet0/0/0	1.1.1.1	Full

可以看到，R2与R4没有建立起邻居关系，这是因为R4上还没有进行相匹配的认证功能的配置，所以R4未能通过R2的认证。

在R4上进行相应的认证功能的配置。

```
[R4]interface Ethernet 1/0/0
```

```
[R4-Ethernet1/0/0]ospf authentication-mode md5 24 cipher Huawei
```

配置完成后，重新在R2上查看OSPF邻居信息。

```
<R2>display ospf peer brief
```

```
OSPF Process 1 with Router ID 2.2.2.2
```


Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	Serial2/0/0	3.3.3.3	Full
0.0.0.0	GigabitEthernet0/0/0	1.1.1.1	Full
0.0.0.1	Ethernet1/0/0	4.4.4.4	Full

可以看到，现在R2与R4已经成功地建立了邻居关系。

在R2上查看认证配置情况。

```
<R2>display current-configuration | include authentication-mode
```

```
ospf authentication-mode md5 24 cipher Q+vy@C ~)5Y3IF$':[285sfn#
```

```
authentication-mode simple plain Huawei
```

可以看到，相对于 MD5 模式下的密文认证，简单的明文认证方式会直接将密码显示出来，安全性较低。

思考

按照OSPF的网络设计要求，不同普通区域（Area）之间的通信必须经由骨干区域（Area 0）中转才能实现。这种要求的出发点是什么？

2.2 OSPF 邻居邻接关系

原理概述

OSPF 网络中，路由器在发送任何链路状态信息之前，必须先建立起正确的 OSPF 邻居邻接关系。

OSPF 路由器是使用 Hello 报文来建立邻居关系的。OSPF 路由器会检查所收到的Hello 报文中的各种参数，如 Router-ID、Area-ID、认证信息、网络掩码、Hello 时间间隔等。如果这些参数和接收接口上配置的对应参数都一一保持一致，则邻居关系就会建立起来，否则就无法建立起邻居关系。

OSPF 路由器的邻居关系建立完成之后，下一步才是建立邻接关系。并不是所有的OSPF 邻居之间都可以建立邻接关系，这要取决于 OSPF 邻居之间的网络类型。例如，在点到点网络上，有效的OSPF邻居关系都可以进一步形成邻接关系。在广播型网络上，会选举DR和BDR；DR和BDR会与所有其他路由器都建立邻接关系，其他路由器都只与DR和BDR建立邻接关系。

实验目的

- 理解OSPF 邻居关系和OSPF 邻接关系的含义及差别
- 观察OSPF 邻居邻接关系的建立过程
- 观察OSPF 链路状态数据库的同步过程

实验内容

实验拓扑如图2-2所示，实验编址如表2-2所示。本实验模拟了一个跨国企业网络场景，国内集团总部的路由器R1、R2、R3组成了一个广播型网络，国外分公司1的路由器R4与国内集团总部核心路由器R1组成了一个点到点网络，国外分公司2的路由器R5与国内集团总部核心路由器R1组成了另一个点到点网络。通过实验，读者需要理解OSPF邻居关系和OSPF邻接关系的含义及差别，并且观察OSPF邻居邻接关系的建立过程以及OSPF 链路状态数据库（LSDB：Link State Database）的同步过程。

实验拓扑

figure_0062_0032

图2-2 OSPF邻居邻接关系

实验编址表

表2-2 实验编址

figure_0062_0033

(续表)

figure_0063_0034

实验步骤

1. 基本配置

根据图2-2和表2-2进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.123.2
```

```
PING 10.0.123.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.123.2: bytes=56 Sequence=1 ttl=255 time=400 ms
```

```
--- 10.0.123.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 400/400/400 ms
```

其余直连网段的连通性测试过程在此省略。

2. 配置OSPF路由协议

在每台路由器上进行OSPF协议的配置，其中R1、R2、R3之间的链路属于区域0，R1和R4之间的链路属于区域1，R1和R5之间的链路属于区域2。

```
[R1]ospf router-id 10.0.1.1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.123.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]area 1
```

```
[R1-ospf-1-area-0.0.0.1]network 10.0.14.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.1]area 2
```

```
[R1-ospf-1-area-0.0.0.2]network 10.0.15.0 0.0.0.255
```

```
[R2]ospf router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.123.0 0.0.0.255
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

```
[R3]ospf router-id 10.0.3.3
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.123.0 0.0.0.255
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

```
[R4]ospf router-id 10.0.4.4
```

```
[R4-ospf-1]area 1
```

```
[R4-ospf-1-area-0.0.0.1]network 10.0.14.0 0.0.0.255
```

```
[R4-ospf-1-area-0.0.0.1]network 10.0.4.4 0.0.0.0
```

```
[R5]ospf router-id 10.0.5.5
```

```
[R5-ospf-1]area 2
```

```
[R5-ospf-1-area-0.0.0.2]network 10.0.15.0 0.0.0.255
```

```
[R5-ospf-1-area-0.0.0.2]network 10.0.5.5 0.0.0.0
```

配置完成后，在 R1 上查看 OSPF 邻居建立情况，读者可自行在其他路由器上查看OSPF邻居建立情况。

```
<R1>display ospf peer brief
```

OSPF Process 1 with Router ID 10.0.1.1

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	10.0.2.2	Full
0.0.0.0	GigabitEthernet0/0/0	10.0.3.3	Full
0.0.0.1	Serial1/0/0	10.0.4.4	Full
0.0.0.2	Serial1/0/1	10.0.5.5	Full

可以看到，R1 的 OSPF 邻居状态都为 Full，说明邻居邻接关系已经成功建立。在R1上查看OSPF邻居状态的详细信息。

<R1>display ospf peer

OSPF Process 1 with Router ID 10.0.1.1

Neighbors

Area 0.0.0.0 interface 10.0.123.1(GigabitEthernet0/0/0)'s neighbors

Router ID: 10.0.2.2 Address: 10.0.123.2

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.123.3 BDR: 10.0.123.2 MTU: 0

Dead timer due in 26 sec

Retrans timer interval: 4

Neighbor is up for 00:09:00

Authentication Sequence: [0]

Router ID: 10.0.3.3 Address: 10.0.123.3

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.123.3 BDR: 10.0.123.2 MTU: 0

Dead timer due in 34 sec

Retrans timer interval: 0

Neighbor is up for 00:09:17

Authentication Sequence: [0]

Neighbors

Area 0.0.0.1 interface 10.0.14.1(Serial1/0/0)'s neighbors

Router ID: 10.0.4.4 Address: 10.0.14.4

State: Full Mode:Nbr is Master Priority: 1

DR: None BDR: None MTU: 0

Dead timer due in 64 sec

Retrans timer interval: 5

Neighbor is up for 00:02:47

Authentication Sequence: [0]

Neighbors

Area 0.0.0.2 interface 10.0.15.1(Serial1/0/1)'s neighbors

Router ID: 10.0.5.5 Address: 10.0.15.5

State: Exchange Mode:Nbr is Master Priority: 1

DR: None BDR: None MTU: 0

Dead timer due in 38 sec

Retrans timer interval: 0

Neighbor is up for 00:00:00

Authentication Sequence: [0]

可以看到，包含R1、R2、R3的广播网络已经完成了DR/BDR的选举，选举结果是10.0.123.3（R3）为DR，10.0.123.2（R2）为BDR。R1与R4之间、R1与R5之间的两个点到点网络都没有进行DR/BDR的选举。

在R1 上查看广播型网络的接口GE 0/0/0 和点到点网络的接口Serial 1/0/0 的详细情况。

```
<R1>display ospf interface GigabitEthernet0/0/0
```

OSPF Process 1 with Router ID 10.0.1.1

Interfaces

Interface: 10.0.123.1 (GigabitEthernet0/0/0)

Cost: 1 State:DROther Type:Broadcast MTU: 1500

Priority: 1

Designated Router: 10.0.123.3

Backup Designated Router: 10.0.123.2

Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1

Smart-discover: enable

```
<R1>display ospf interface serial1/0/0
```

OSPF Process 1 with Router ID 10.0.1.1

Interfaces

Interface: 10.0.14.1 (Serial1/0/0) --> 10.0.14.4

Cost: 48 State:P-2-P Type:P2P MTU: 1500

Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1

可以看到，广播网络接口和点到点网络接口默认的Hello时间间隔都为10s，失效时间都为40s。

3．观察OSPF邻居邻接关系的建立过程

首先观察在广播网络上 OSPF 邻居邻接关系的建立过程。为了在 R1 上清晰地观察到广播网络上 OSPF 邻居邻接关系的建立过程，请先关闭

R1 上的 Serial 1/0/0 和 Serial 1/0/1接口。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]shutdown
```

```
[R1-Serial1/0/0]interface Serial 1/0/1
```

```
[R1-Serial1/0/1]shutdown
```

然后，在R1上查看OSPF邻居状态。

```
<R1>display ospf peer brief
```

OSPF Process 1 with Router ID 10.0.1.1

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	10.0.2.2	Full
0.0.0.0	GigabitEthernet0/0/0	10.0.3.3	Full

可以看到，R1与R2、R3的邻居状态都是Full，说明已经建立好了邻接关系。

现在，在R1上重启OSPF进程，通过Debugging调试观察R1与R2之间的OSPF邻接关系的建立过程。

```
<R1>debugging ospf packet
```

```
<R1>reset ospf process
```

Jun 6 2013 20:12:40-05:13 R1 %%01OSPF/3/NBR_CHG_DOWN(l)
[32]:Neighbor event:neighbor state changed to Down. (ProcessId=256,
NeighborAddress=2.2.0.10, NeighborEvent=KillNbr,
NeighborPreviousState=Full, NeighborCurrentState=Down)

<R1>

Jun 6 2013 20:12:40-05:13 R1 %%01OSPF/3/NBR_DOWN_REASON(l)
[33]:Neighbor state leaves full or changed to Down. (ProcessId=256,
NeighborRouterId=2.2.0.10, NeighborAreaId=0,
NeighborInterface=GigabitEthernet0/0/0,NeighborDownImmediate
reason=Neighbor Down Due to Kill Neighbor,
NeighborDownPrimeReason=OSPF Process Reset,
NeighborChangeTime=2013-06-06 20:12:40-05:13)

.....

<R1>

Jun 6 2013 20:12:40-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[38]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent=HelloReceived,
NeighborPreviousState=Down, NeighborCurrentState=Init)

<R1>

Jun 6 2013 20:12:40-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[39]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent=2WayReceived,
NeighborPreviousState=Init, NeighborCurrentState=2Way)

<R1>

Jun 6 2013 20:12:40-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[40]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent=AdjOk?,
NeighborPreviousState=2Way, NeighborCurrentState=ExStart)

.....

<R1>

```
Jun 6 2013 20:12:45-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[45]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent=NegotiationDone,
NeighborPreviousState=ExStart, NeighborCurrentState=Exchange)
```

<R1>

```
Jun 6 2013 20:12:45-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[46]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent=ExchangeDone,
NeighborPreviousState=Exchange, NeighborCurrentState>Loading)
```

<R1>

```
Jun 6 2013 20:12:45-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[47]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent>LoadingDone,
NeighborPreviousState>Loading, NeighborCurrentState=Full)
```

从上面的显示信息可知，重启OSPF进程后，R1与R2的邻居关系由Full状态转到了Down状态。然后，当R1从R2收到Hello报文后，邻居关系由Down状态转变为了初始状态（Init）。Hello报文的参数协商完成后，R1与R2的邻居关系进入到了2-Way状态，2-Way状态表明双方已经成功建立了邻居关系。邻居关系建立之后，R1与R2进入到信息交换初始状态（ExStart）、信息交换状态（Exchange）以及信息加载状态（Loading），最终进入到Full状态。Full状态表明双方已成功建立了邻接关系。R1与R3的OSPF邻接关系的建立过程完全类似，这里不再赘述。

需要特别说明的是，OSPF路由器之间的邻居关系并不等于邻接关系。邻居关系建立后，还需完成链路状态信息的交换，然后才能建立起邻接关系。

下面通过实验来进一步说明OSPF路由器之间的邻居关系与邻接关系的区别。在广播网络中，DRothers之间不需要交换LSA（Link State

Advertisement) , DRothers是通过DR/BDR来获取整个广播网络的链路状态信息的, 所以DRothers之间不需要建立邻接关系, 只需要建立邻居关系即可。

在R1、R2、R3、SW组成的广播网络中, R3是DR, R2是BDR, 只有R1是DRothers, 所以不便于观察DRothers 之间的邻居关系。现在, 将R1的GE 0/0/0和R2 的GE 0/0/0接口优先级的值改为0, 放弃DR的选举, 使它们都成为DRothers, 以便观察它们之间的OSPF邻居关系。

```
[R1]interface GigabitEthernet0/0/0
```

```
[R1-GigabitEthernet0/0/0]ospf dr-priority 0
```

```
[R2]interface GigabitEthernet0/0/0
```

```
[R2-GigabitEthernet0/0/0]ospf dr-priority 0
```

重启R1和R2上的OSPF进程后, 先在DR路由器R3上查看OSPF的邻居建立情况。

```
<R3>display ospf peer
```

```
OSPF Process 1 with Router ID 10.0.3.3
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.123.3(GigabitEthernet0/0/0)'s neighbors
```

```
Router ID: 10.0.1.1      Address: 10.0.123.1
```

```
State: Full Mode:Nbr is Slave Priority: 0
```

```
DR: 10.0.123.3 BDR: None MTU: 0
```

```
Dead timer due in 34 sec
```

```
Retrans timer interval: 5
```

Neighbor is up for 00:26:08

Authentication Sequence: [0]

Router ID: 10.0.2.2 Address: 10.0.123.2

State: Full Mode:Nbr is Slave Priority: 0

DR: 10.0.123.3 BDR: None MTU: 0

Dead timer due in 33 sec

Retrans timer interval: 5

Neighbor is up for 00:26:04

Authentication Sequence: [0]

可以看到，R3为DR，网络中没有BDR，R3分别与R1和R2建立了邻接关系。

在路由器R1上查看OSPF邻居关系建立情况。

<R1>display ospf peer

OSPF Process 1 with Router ID 10.0.1.1

Neighbors

Area 0.0.0.0 interface 10.0.123.1(GigabitEthernet0/0/0)'s neighbors

Router ID: 10.0.2.2 Address: 10.0.123.2

State: 2-Way Mode:Nbr is Master Priority: 0

DR: 10.0.123.3 BDR: None MTU: 0

Dead timer due in 40 sec

Retrans timer interval: 5

Neighbor is up for 00:00:00

Authentication Sequence: [0]

Router ID: 10.0.3.3 Address: 10.0.123.3

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.123.3 BDR: None MTU: 0

Dead timer due in 30 sec

Retrans timer interval: 5

Neighbor is up for 00:30:03

Authentication Sequence: [0]

可以看到，R1与DR路由器R3建立的是邻接关系，状态为Full，而与DRothers路由器R2只建立了邻居关系，状态为2-Way。

在路由器R1上重启OSPF进程，通过Debugging调试观察OSPF邻居邻接关系的建立过程。

```
<R1>debugging ospf packet
```

```
<R1>reset ospf process
```

```
Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/3/NBR_CHG_DOWN(l)
[0]:Neighbor event:neighbor state changed to Down. (ProcessId=256,
NeighborAddress=2.2.0.10, NeighborEvent=KillNbr,
NeighborPreviousState=2Way, NeighborCurrentState=Down)
```

```
<R1>
```

```
Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/3/NBR_DOWN_REASON(l)
[1]:Neighbor state leaves full or changed to Down. (ProcessId=256,
NeighborRouterId=2.2.0.10, NeighborAreaId=0,
NeighborInterface=GigabitEthernet0/0/0,NeighborDownImmediate
```

reason=Neighbor Down Due to Kill Neighbor,
NeighborDownPrimeReason=OSPF Process Reset,
NeighborChangeTime=2013-06-23 20:52:58-05:13)

<R1>

Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/3/NBR_CHG_DOWN(l)
[2]:Neighbor event:neighbor state changed to Down. (ProcessId=256,
NeighborAddress=3.3.0.10, NeighborEvent=KillNbr,
NeighborPreviousState=Full, NeighborCurrentState=Down)

<R1>

Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/3/NBR_DOWN_REASON(l)
[3]:Neighbor state leaves full or changed to Down. (ProcessId=256,
NeighborRouterId=3.3.0.10, NeighborAreaId=0,
NeighborInterface=GigabitEthernet0/0/0,NeighborDownImmediate
reason=Neighbor Down Due to Kill Neighbor,
NeighborDownPrimeReason=OSPF Process Reset,
NeighborChangeTime=2013-06-23 20:52:58-05:13)

<R1>

Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[4]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent=HelloReceived,
NeighborPreviousState=Down, NeighborCurrentState=Init)

<R1>

Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[5]:Neighbor changes event: neighbor status changed.
(ProcessId=256,NeighborAddress=2.123.0.10,NeighborEvent=2WayReceiv
ed,NeighborPreviousState=Init,NeighborCurrentState=2Way)

<R1>

Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[6]:Neighbor changes event: neighbor status changed.

(ProcessId=256,NeighborAddress=3.123.0.10,NeighborEvent=HelloReceived,NeighborPreviousState=Down,NeighborCurrentState=Init)

<R1>

Jun 23 2013 20:52:58-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[7]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=3.123.0.10, NeighborEvent=2WayReceived,
NeighborPreviousState=Init, NeighborCurrentState=2Way)

<R1>

Jun 23 2013 20:52:59-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[8]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=2.123.0.10, NeighborEvent=AdjOk?,
NeighborPreviousState=2Way, NeighborCurrentState=2Way)

<R1>

Jun 23 2013 20:52:59-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[9]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=3.123.0.10, NeighborEvent=AdjOk?,
NeighborPreviousState=2Way, NeighborCurrentState=ExStart)

<R1>

Jun 23 2013 20:52:59-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[10]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=3.123.0.10, NeighborEvent=NegotiationDone,
NeighborPreviousState=ExStart, NeighborCurrentState=Exchange)

<R1>

Jun 23 2013 20:53:00-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[11]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=3.123.0.10, NeighborEvent=ExchangeDone,
NeighborPreviousState=Exchange, NeighborCurrentState>Loading)

<R1>


```
Jun 23 2013 20:53:00-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[12]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=3.123.0.10, NeighborEvent=LoadingDone,
NeighborPreviousState=Loading, NeighborCurrentState=Full)
```

从上面的显示信息可以看到，R1与R2只建立了OSPF邻居关系，处于2-Way状态，而R1与R3之间建立了邻接关系，处于Full状态。

接下来实验观察点到点网络中 OSPF 的邻居关系建立情况。开启 R1 上的两个串口Serial 1/0/0和Serial 1/0/1，然后关闭广播接口GE 0/0/0。关闭广播接口的目的是突出所要关注的实验现象，排除干扰因素。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]undo shutdown
```

```
[R1-Serial1/0/0]interface Serial 1/0/1
```

```
[R1-Serial1/0/1]undo shutdown
```

```
[R1-Serial1/0/1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]shutdown
```

然后，在R1上查看OSPF邻居建立情况。

```
<R1>display ospf peer brief
```

```
OSPF Process 1 with Router ID 10.0.1.1
```

```
Peer Statistic Information
```

```
-----
-----
```

Area Id	Interface	Neighbor id	State
0.0.0.1	Serial1/0/0	10.0.4.4	Full

0.0.0.2 Serial1/0/1 10.0.5.5 Full

可以看到，路由器R1与R4和R5已经分别建立了邻接关系，邻居状态为Full。

在路由器R1上重启OSPF进程，通过Debugging调试观察R1与R4之间的OSPF邻居邻接关系的建立过程。

```
<R1>debugging ospf packet
```

```
<R1>reset ospf process
```

```
Jun 24 2013 19:50:29-05:13 R1 %%01OSPF/3/NBR_CHG_DOWN(l)
[0]:Neighbor event:neighbor state changed to Down. (ProcessId=256,
NeighborAddress=4.4.0.10, NeighborEvent=KillNbr,
NeighborPreviousState=Full, NeighborCurrentState=Down)
```

```
<R1>
```

```
Jun 24 2013 19:50:29-05:13 R1 %%01OSPF/3/NBR_DOWN_REASON(l)
[1]:Neighbor state leaves full or changed to Down. (ProcessId=256,
NeighborRouterId=4.4.0.10, NeighborAreaId=16777216,
NeighborInterface=Serial1/0/0,NeighborDownImmediate reason=Neighbor
Down Due to Kill Neighbor, NeighborDownPrimeReason=OSPF Process
Reset, NeighborChangeTime=2013-06-24 19:50:29-05:13)
```

```
.....
```

```
<R1>
```

```
Jun 24 2013 19:50:38-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[9]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=4.14.0.10, NeighborEvent=HelloReceived,
NeighborPreviousState=Down, NeighborCurrentState=Init)
```

```
<R1>
```

Jun 24 2013 19:50:38-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[10]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=4.14.0.10, NeighborEvent=2WayReceived,
NeighborPreviousState=Init, NeighborCurrentState=ExStart)

<R1>

Jun 24 2013 19:50:38-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[11]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=4.14.0.10, NeighborEvent=NegotiationDone,
NeighborPreviousState=ExStart, NeighborCurrentState=Exchange)

<R1>

Jun 24 2013 19:50:38-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[12]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=4.14.0.10, NeighborEvent=ExchangeDone,
NeighborPreviousState=Exchange, NeighborCurrentState>Loading)

<R1>

Jun 24 2013 19:50:39-05:13 R1 %%01OSPF/4/NBR_CHANGE_E(l)
[13]:Neighbor changes event: neighbor status changed. (ProcessId=256,
NeighborAddress=4.14.0.10, NeighborEvent>LoadingDone,
NeighborPreviousState>Loading, NeighborCurrentState=Full)

从上面的显示信息可以看到，重启OSPF进程后，R1与R4的邻居关系由Full状态转变为Down状态。当R1收到R4发送的Hello报文后，邻居关系由Down状态转变为初始状态（Init）。接着，R1与R4便直接进入了信息交换初始状态（ExStart）、信息交换状态（Exchange），以及信息加载状态（Loading），最终成功建立了邻接关系，进入了Full状态。R1与R5的OSPF邻接关系的建立过程完全类似，这里不再赘述。

需要注意的是，R1与R4路由器没有经过2-Way状态，并且也不存在2-Way状态，说明点到点网络与广播网络中OSPF的邻接关系建立过程不是完全一样的。在点到点网络中，能够建立OSPF邻居关系的路由器一定会继续建立邻接关系。

4．观察OSPF链路状态数据库的同步过程

下面将通过查看报文的方式来简单观察一下OSPF邻接关系建立过程中链路状态数据库LSDB是如何同步的，这里仅以点到点网络为例进行实验。

在R1 的Serial 1/0/0 接口上查看报文，重启R1 上的OSPF 进程。

```
<R1>debugging ospf packet
```

```
<R1>reset ospf process
```

查看报文情况，如图2-3所示。

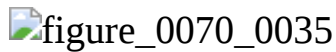


图2-3 在R1的Serial 1/0/0接口上查看报文

从图2-3中可以观察到OSPF协议的各种数据报文，它们反映了LSDB的同步过程，同时也反映了OSPF邻居邻接关系建立的过程：首先，R1（10.0.14.1）和R4（10.0.14.4）通过Hello 报文进行协商，然后通过数据库描述（DD：Database Description）报文、链路状态请求（LSR：Link State Request）报文、链路状态更新（LSU：Link State Update）报文等，最终实现了LSDB的同步，并建立起OSPF邻接关系。

下面来分析一下R1和R4相互发送的Hello报文，如图2-4和图2-5所示。

No.25（Hello报文）：

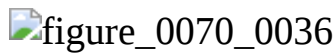


图2-4 R1发送的Hello报文

No.26（Hello报文）：

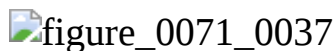


图2-5 R4发送的Hello报文

可以看到，Hello报文中包含了很多基本信息，例如，网络掩码为24位，Hello间隔时间为10s，路由器死亡时间间隔为40s，网络上没有DR和BDR。另外，R4发出的Hello报文中指出了活跃邻居为R1，这说明R1与R4成功建立了OSPF邻居关系。

接下来简单分析一下DD报文，如图2-6和图2-7所示。

No.27（DD报文）：

figure_0071_0038

图2-6 R1发送的DD报文

No.28（DD报文）：

figure_0071_0039

图2-7 R4发送的DD报文

可以发现，第27和第28两个报文为R1和R4首次交互的DD报文，其中I位、M位、MS位都设置为1。R1和R4都宣称自己是主路由器。这两个DD报文是不包含数据库摘要信息的。首次DD报文交互后，便可选举出Router-ID较大的R4为主路由器。

关于LSR、LSU、LSAck（Link State Acknowledge）等报文的详细内容，感兴趣的读者可自行去进行分析和研究。

思考

在OSPF广播型网络中的DR与BDR之间需要建立OSPF邻接关系吗？为什么？

2.3 OSPF 链路状态数据库

原理概述

OSPF 是一种基于链路状态的动态路由协议，每台 OSPF 路由器都会生成相关的LSA，并将这些LSA通告出去。路由器收到LSA后，会将它们存放在链路状态数据库LSDB中。

LSA 有多种不同的类型，不同类型的 LSA 的功能和作用是不同的，下面介绍几种常见的LSA。

Type-1 LSA（Router LSA）：每台路由器都会产生，用来描述路由器的直连链路状态和开销值。Type-1 LSA只能在所属区域内部泛洪，不能泛洪到其他区域。

Type-2 LSA（Network LSA）：它是由DR产生的，主要用来描述该DR所在网段的网络掩码以及该网段内有哪些路由器。Type-2 LSA 只能在所属区域内部泛洪，不能泛洪到其他区域。

Type-3 LSA（Network Summary LSA）：它是由ABR（Area Boundary Router）产生的，ABR 路由器将所连区域的Type-1 和Type-2 LSA转换为Type-3 LSA，用来描述区域间的路由信息。Type-3 LSA可以泛洪到整个AS（Autonomous System，自治域）内部，但不能泛洪到Totally Stub 区域和Totally NSSA（Not-So-Stubby Area）区域。

Type-4 LSA（ASBR Summary LSA）：它是由ASBR（Autonomous System Boundary Router）所在区域的ABR产生的，用来描述到ASBR的路由。Type-4 LSA 可以泛洪到整个AS 内部，但不能泛洪到Stub 区域、Totally Stub 区域、NSSA 区域和Totally NSSA区域中。

Type-5 LSA（AS External LSA）：它是由ASBR产生的，用来描述到AS 外部网络的路由。Type-5 LSA 可以泛洪到整个AS 内部，但不能泛洪到Stub 区域、Totally Stub 区域、NSSA 区域和Totally NSSA 区域中。

Type-6 LSA：用于OSPF 组播。

Type-7 LSA（NSSA LSA）：它是由NSSA 区域或Totally NSSA区域的NSSA ASBR产生的，用来描述到AS 外部的路由。Type-7 LSA只能出现在所属NSSA 区域或Totally NSSA区域内部。

实验目的

- 理解OSPF 中不同类型的LSA 的作用
- 熟悉OSPF 中不同类型的LSA 的泛洪范围
- 熟悉LSA 中重要字段的含义

实验内容

实验拓扑如图2-8所示，实验编址如表2-3所示。本实验模拟了一个企业总部与两个分支机构的网络场景，R2、R3、R5之间的链路属于区域0，R1与R2之间的链路属于区域1，R3 与R4 之间的链路属于区域2，R1和R4的Loopback 1 接口用来表示以后有合作伙伴加入时的网络。区域1为普通区域，区域2为NSSA区域。在区域0中，R5为DR，R2为BDR，R3为DRother。实验过程中会大量地展示和分析链路状态数据库，以加深读者对不同类型LSA的认识和理解。

实验拓扑

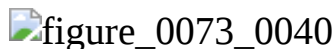
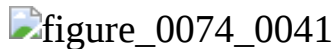
figure_0073_0040

图2-8 OSPF链路状态数据库

实验编址表

表2-3 实验编址

figure_0074_0041

实验步骤

1．基本配置

根据图2-8和表2-3进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 60/60/60 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF路由协议

在每台路由器上配置OSPF路由协议，R1与R2之间的链路属于区域1，R3与R4之间的链路属于区域2，R2、R3、R5之间的链路属于区域0，区域2是NSSA区域。

```
[R1]router id 10.0.1.1
```

```
[R1]ospf 10
```

```
[R1-ospf-10]area 1
```

```
[R1-ospf-10-area-0.0.0.1]network 10.0.12.0 0.0.0.255
```

```
[R1-ospf-10-area-0.0.0.1]network 10.0.1.1 0.0.0.0
```

```
[R2]router id 10.0.2.2
```


[R2]ospf 10

[R2-ospf-10]area 1

[R2-ospf-10-area-0.0.0.1]network 10.0.12.0 0.0.0.255

[R2-ospf-10-area-0.0.0.1]network 10.0.2.2 0.0.0.0

[R2-ospf-10-area-0.0.0.1]area 0

[R2-ospf-10-area-0.0.0.0]network 10.0.235.0 0.0.0.255

[R3]router id 10.0.3.3

[R3]ospf 10

[R3-ospf-10]area 2

[R3-ospf-10-area-0.0.0.2]nssa

[R3-ospf-10-area-0.0.0.2]network 10.0.34.0 0.0.0.255

[R3-ospf-10-area-0.0.0.2]network 10.0.3.3 0.0.0.0

[R3-ospf-10-area-0.0.0.2]area 0

[R3-ospf-10-area-0.0.0.0]network 10.0.235.0 0.0.0.255

[R4]router id 10.0.4.4

[R4]ospf 10

[R4-ospf-10]area 2

[R4-ospf-10-area-0.0.0.2]nssa

[R4-ospf-10-area-0.0.0.2]network 10.0.34.0 0.0.0.255

[R4-ospf-10-area-0.0.0.2]network 10.0.4.4 0.0.0.0

```
[R5]router id 10.0.5.5
```

```
[R5]ospf 10
```

```
[R5-ospf-10]area 0
```

```
[R5-ospf-10-area-0.0.0.0]network 10.0.235.0 0.0.0.255
```

```
[R5-ospf-10-area-0.0.0.0]network 10.0.5.5 0.0.0.0
```

配置完成后，在R2、R5的GE 0/0/0接口上修改接口优先级的值，使R5成为DR，R2成为BDR。

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]ospf dr-priority 50
```

```
[R5]interface GigabitEthernet 0/0/0
```

```
[R5-GigabitEthernet0/0/0]ospf dr-priority 100
```

在R2、R3、R5上重启OSPF进程，下面仅示意了R5的重启方法。

```
<R5>reset ospf process
```

```
Warning: The OSPF process will be reset. Continue? [Y/N]:y
```

在R3上查看OSPF的DR与BDR的选举情况。

```
<R3>display ospf peer
```

```
OSPF Process 10 with Router ID 10.0.3.3
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.235.3(GigabitEthernet0/0/0)'s neighbors
```

```
Router ID: 10.0.2.2      Address: 10.0.235.2
```

```
State: Full Mode:Nbr is Slave Priority: 50
```

DR: 10.0.235.5 BDR: 10.0.235.2 MTU: 0

Dead timer due in 31 sec

Retrans timer interval: 4

Neighbor is up for 00:00:52

Authentication Sequence: [0]

Router ID: 10.0.5.5 Address: 10.0.235.5

State: Full Mode:Nbr is Master Priority: 100

DR: 10.0.235.5 BDR: 10.0.235.2 MTU: 0

Dead timer due in 36 sec

Retrans timer interval: 4

Neighbor is up for 00:01:14

Authentication Sequence: [0]

Neighbors

Area 0.0.0.2 interface 10.0.34.3(GigabitEthernet0/0/1)'s neighbors

Router ID: 10.0.4.4 Address: 10.0.34.4

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.34.4 BDR: 10.0.34.3 MTU: 0

Dead timer due in 36 sec

Retrans timer interval: 5

Neighbor is up for 00:16:20

Authentication Sequence: [0]

可以看到，在R2、R3、R5组成的广播网络中，目前R5是DR，R2是BDR。接下来查看每台路由器上的路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0 0	D		127.0.0.1 LoopBack0
10.0.2.2/32 GigabitEthernet0/0/0	OSPF	10 1	D		10.0.12.2
10.0.3.3/32 GigabitEthernet0/0/0	OSPF	10 2	D		10.0.12.2
10.0.4.4/32 GigabitEthernet0/0/0	OSPF	10 3	D		10.0.12.2
10.0.5.5/32 GigabitEthernet0/0/0	OSPF	10 2	D		10.0.12.2
10.0.12.0/24 GigabitEthernet0/0/0	Direct	0 0	D		10.0.12.1
10.0.12.1/32 GigabitEthernet0/0/0	Direct	0 0	D		127.0.0.1

10.0.12.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.34.0/24	OSPF	10	3	D	10.0.12.2	
GigabitEthernet0/0/0						
10.0.235.0/24	OSPF	10	2	D	10.0.12.2	
GigabitEthernet0/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
192.168.1.0/24	Direct	0	0	D	192.168.1.1	LoopBack1
192.168.1.1/32	Direct	0	0	D	127.0.0.1	LoopBack1
192.168.1.255/32	Direct	0	0	D	127.0.0.1	LoopBack1
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 16 Routes : 16

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface

10.0.1.1/32	OSPF	10	1	D	10.0.12.1		
GigabitEthernet0/0/1							
10.0.2.2/32	Direct	0	0	D	127.0.0.1	LoopBack0	
10.0.3.3/32	OSPF	10	1	D	10.0.235.3		
GigabitEthernet0/0/0							
10.0.4.4/32	OSPF	10	2	D	10.0.235.3		
GigabitEthernet0/0/0							
10.0.5.5/32	OSPF	10	1	D	10.0.235.5		
GigabitEthernet0/0/0							
10.0.12.0/24	Direct	0	0	D	10.0.12.2		
GigabitEthernet0/0/1							
10.0.12.2/32	Direct	0	0	D	127.0.0.1		
GigabitEthernet0/0/1							
10.0.12.255/32	Direct	0	0	D	127.0.0.1		
GigabitEthernet0/0/1							
10.0.34.0/24	OSPF	10	2	D	10.0.235.3		
GigabitEthernet0/0/0							
10.0.235.0/24	Direct	0	0	D	10.0.235.2		
GigabitEthernet0/0/0							
10.0.235.2/32	Direct	0	0	D	127.0.0.1		
GigabitEthernet0/0/0							
10.0.235.255/32	Direct	0	0	D	127.0.0.1		
GigabitEthernet0/0/0							
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0	
127.0.0.1/32	Direct	0	0	0	D	127.0.0.1	InLoopBack0

127.255.255.255/32 Direct 0 0 0 D 127.0.0.1
InLoopBack0

255.255.255.255/32 Direct 0 0 0 D 127.0.0.1
InLoopBack0

<R3>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 16Routes : 16

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.235.2
10.0.2.2/32 GigabitEthernet0/0/0	OSPF	10	1	D	10.0.235.2
10.0.3.3/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.4.4/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.34.4
10.0.5.5/32 GigabitEthernet0/0/0	OSPF	10	1	D	10.0.235.5
10.0.12.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.235.2
10.0.34.0/24 GigabitEthernet0/0/1	Direct	0	0	D	10.0.34.3

10.0.34.3/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.34.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.235.0/24	Direct	0	0	D	10.0.235.3	
GigabitEthernet0/0/0						
10.0.235.3/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.235.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 18 Routes : 18

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

0.0.0.0/0	O_NSSA	150	1	D	10.0.34.3	
GigabitEthernet0/0/0						
10.0.1.1/32	OSPF	10	3	D	10.0.34.3	
GigabitEthernet0/0/0						
10.0.2.2/32	OSPF	10	2	D	10.0.34.3	
GigabitEthernet0/0/0						
10.0.3.3/32	OSPF	10	1	D	10.0.34.3	
GigabitEthernet0/0/0						
10.0.4.4/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.5.5/32	OSPF	10	2	D	10.0.34.3	
GigabitEthernet0/0/0						
10.0.12.0/24	OSPF	10	3	D	10.0.34.3	
GigabitEthernet0/0/0						
10.0.34.0/24	Direct	0	0	D	10.0.34.4	
GigabitEthernet0/0/0						
10.0.34.4/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.34.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.235.0/24	OSPF	10	2	D	10.0.34.3	
GigabitEthernet0/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

172.16.1.0/24	Direct	0	0	D	172.16.1.1	LoopBack1
172.16.1.1/32	Direct	0	0	D	127.0.0.1	LoopBack1
172.16.1.255/32	Direct	0	0	D	127.0.0.1	LoopBack1
255.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0

<R5>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 14 Routes : 14

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.235.2
10.0.2.2/32 GigabitEthernet0/0/0	OSPF	10	1	D	10.0.235.2
10.0.3.3/32 GigabitEthernet0/0/0	OSPF	10	1	D	10.0.235.3
10.0.4.4/32 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.235.3
10.0.5.5/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.12.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.235.2

10.0.34.0/24	OSPF	10	2	D	10.0.235.3	
GigabitEthernet0/0/0						
10.0.235.0/24	Direct	0	0	D	10.0.235.5	
GigabitEthernet0/0/0						
10.0.235.5/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.235.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direc	0	0	D	127.0.0.1	
InLoopBack0						
255.255.255.255/32	Direc	0	0	D	127.0.0.1	
InLoopBack0						

可以看到，每台路由器都已获得了非直连网络的路由条目。接下来使用ping命令检测连通性。

```
<R1>ping -a 10.0.1.1 10.0.4.4
```

```
PING 10.0.4.4: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.4.4: bytes=56 Sequence=1 ttl=253 time=150 ms
```

```
Reply from 10.0.4.4: bytes=56 Sequence=2 ttl=253 time=100 ms
```

```
Reply from 10.0.4.4: bytes=56 Sequence=3 ttl=253 time=70 ms
```

```
Reply from 10.0.4.4: bytes=56 Sequence=4 ttl=253 time=80 ms
```

```
Reply from 10.0.4.4: bytes=56 Sequence=5 ttl=253 time=90 ms
```

--- 10.0.4.4 ping statistics ---

5 packet(s) transmitted

5 packet(s) received

0.00% packet loss

round-trip min/avg/max = 70/98/150 ms

<R4>ping -a 10.0.4.4 10.0.5.5

PING 10.0.5.5: 56 data bytes, press CTRL_C to break

Reply from 10.0.5.5: bytes=56 Sequence=1 ttl=254 time=90 ms

Reply from 10.0.5.5: bytes=56 Sequence=2 ttl=254 time=60 ms

Reply from 10.0.5.5: bytes=56 Sequence=3 ttl=254 time=40 ms

Reply from 10.0.5.5: bytes=56 Sequence=4 ttl=254 time=60 ms

Reply from 10.0.5.5: bytes=56 Sequence=5 ttl=254 time=80 ms

--- 10.0.5.5 ping statistics ---

5 packet(s) transmitted

5 packet(s) received

0.00% packet loss

round-trip min/avg/max = 40/66/90 ms

可以看到，各个网段之间的通信是正常的。

区域1 是普通区域，区域2 是NSSA 区域，区域1 的R1 和区域2 的R4 都需要引入Loopback 1 接口所连接的外部网络路由。在R1 和R4 上使用Route-Policy 精确匹配Loopback 1 接口的直连路由并引入OSPF 进程。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]rule permit source 192.168.1.0 0.0.0.255
```

```
[R1-acl-basic-2000]route-policy 10 permit node 1
```

```
[R1-route-policy]if-match acl 2000
```

```
[R1-route-policy]ospf 10
```

```
[R1-ospf-10]import-route direct route-policy 10
```

```
[R4]acl 2000
```

```
[R4-acl-basic-2000]rule permit source 172.16.1.0 0.0.0.255
```

```
[R4-acl-basic-2000]route-policy 10 permit node 1
```

```
[R4-route-policy]if-match acl 2000
```

```
[R4-route-policy]ospf 10
```

```
[R4-ospf-10]import-route direct route-policy 10
```

配置完成后，在R5上查看由R1和R4引入的两条路由。

```
<R5>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 16 Routes : 16

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					

```

10.0.1.1/32    OSPF      10  2    D    10.0.235.2
GigabitEthernet0/0/0

.....

127.0.0.1/32   Direct    0  0    D    10.0.235.3
GigabitEthernet0/0/0

172.16.1.0/24   O_ASE    150  1    D    10.0.235.3
GigabitEthernet0/0/0

192.168.1.0/24   O_ASE    150  1    D    10.0.235.2
GigabitEthernet0/0/0

255.255.255.255/32 Direct    0  0    D    127.0.0.1
InLoopBack0

127.255.255.255/32 Direct    0  0    D    127.0.0.1
InLoopBack0

```

可以看到，在R5的路由表中，这两条路由都显示为O_ASE，且优先级与开销也都相同。不同之处是这两条路由的下一跳，因为它们是由不同的路由器发送给R5的。

3．查看Type-1 LSA，Type-2 LSA，Type-3 LSA

在区域0的R5上查看LSDB。

```
<R5>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.5.5
```

```
Link State Database
```

```
Area: 0.0.0.0
```

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence

Router	10.0.5.5	10.0.5.5	1548	48	8000000F	1
Router 1	10.0.3.3	10.0.3.3	1553	36	8000000D	
Router 1	10.0.2.2	10.0.2.2	1549	36	8000000C	
Network	10.0.235.5	10.0.5.5	1548	36	8000000A	0
Sum-Net	10.0.34.0	10.0.3.3	1437	28	80000003	1
Sum-Net	10.0.12.0	10.0.2.2	1446	28	80000003	1
Sum-Net	10.0.3.3	10.0.3.3	1437	28	80000003	0
Sum-Net	10.0.2.2	10.0.2.2	1450	28	80000003	0
Sum-Net	10.0.1.1	10.0.2.2	1404	28	80000003	1
Sum-Net	10.0.4.4	10.0.3.3	1398	28	80000003	1
Sum-Asbr	10.0.1.1	10.0.2.2	939	28	80000001	1

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
External	192.168.1.0	10.0.1.1	940	36	80000001	1
External	172.16.1.0	10.0.3.3	576	36	80000001	1

可以看到，R5的LSDB中共有5种LSA，它们分别是Router LSA（或称Type-1 LSA）、Network LSA（或称Type-2 LSA）、Sum-Net LSA（或称Type-3 LSA，Network Summary LSA）、Sum-Asbr LSA（或称Type-4 LSA，ASBR Summary LSA）和External LSA（或称Type-5 LSA，AS External LSA）。

在R5 上查看Router-ID 为10.0.2.2 产生的Router LSA的详细信息。

```
<R5>display ospf lsdb router 10.0.2.2
```

OSPF Process 10 with Router ID 10.0.5.5

Area: 0.0.0.0

Link State Database

Type : Router

Ls id : 10.0.2.2

Adv rtr : 10.0.2.2

Ls age : 1775

Len : 36

Options : ABR E

seq# : 8000000c

chksum : 0xe255

Link count : 1

*Link ID : 10.0.235.5

Data : 10.0.235.2

Link Type : TransNet

Metric : 1

下面解释一下显示信息中的部分参数的含义。

Type：显示信息中，Type 表示了LSA 的类型，这里表示的是Router LSA。不同类型的LSA的作用和泛洪范围是不相同的。Router LSA描述

了路由器的直连链路或接口，泛洪范围为所在区域的内部，以使本区域的其他路由器了解其直连链路或接口的状态信息。

Ls id：对于Router LSA，Ls id 就是产生该Router LSA的路由器的Router-ID。

Adv rtr：Adv rtr描述了LSA 是由哪台路由器产生的。对于Router LSA来讲，Adv rtr就是产生该Router LSA的路由器的Router-ID。

Seq#：每一条LSA都会维护一个Seq#（序列号），产生这条LSA的路由器默认会以30s的周期泛洪这条LSA，每次泛洪时，序列号就加1。LSA的序列号越大，表明这条LSA越新。

Chksum：chksum（校验和）用来校验LSA的完整性。所有的LSA都保存在路由器的LSDB中，每5min会计算一次。如果路由器收到了同一条LSA，且序列号相同，则会比较它们的校验和，校验和越大就被认为相应的LSA越新。

Ls age：Ls age 是指LSA的老化时间，用来表示LSA 已经存活了多长时间，最大值为3600s。当一台路由器产生一条LSA的时候，路由器会将LSA的老化时间设置为0。LSA在产生之后，无论是停留在路由器的LSDB内，还是在传递过程之中，老化时间都会不断增加。为了防止因LSA的过期而造成路由回馈，路由器会每隔30min泛洪自己产生的LSA。若序列号与校验和的比较都不能确定出最新的LSA时，则会比较老化时间。在LSDB 中，如果老化时间相差大于15min 以上，则Ls age 的值越小，说明LSA 越新；如果相差在15min内，则认为两条LSA是一样的。

在上面的显示信息中，Link count 以上的参数信息通常被称为 LSA 头部信息，Link count及以下部分为具体的链路描述信息。Link count标识了这条LSA 描述的链路信息的数量。对于P-2-P链路类型，Link ID是指链路上邻居接口的IP地址；对于TransNet链路类型，Link ID是指DR接口的IP地址。Data是指自身接口的IP地址，Link Type是指接口的链路类型，Metric是指路由器自己到达这条链路的Cost值。需要说明的是，

OSPF协议会把Broadcast和NBMA这两种具有多路访问能力的网络都认为是TransNet网络。

从上得知，R2 的Router LSA 描述了自己连接到了某个TransNet 网络，网络的DR接口的IP地址为10.0.235.5（R5），自己使用10.0.235.2连接到该网络中，且到达这个网络的Cost值为1。

Network LSA是由DR 产生的，它的主要作用是描述TransNet 网络的掩码信息和连接到TransNet网络的路由器的信息。在多路访问网络中，每台路由器都产生Network LSA是没有必要的，因为这会导致Network LSA的重复。

R5是TransNet网络的DR，在R5上查看它产生和发送的Network LSA的详细信息。

```
<R5>display ospf lsdb network 10.0.235.5
```

```
OSPF Process 10 with Router ID 10.0.5.5
```

```
Area: 0.0.0.0
```

```
Link State Database
```

```
Type      : Network
```

```
Ls id     : 10.0.235.5
```

```
Adv rtr   : 10.0.5.5
```

```
Ls age    : 1076
```

```
Len       : 36
```

```
Options   : E
```

```
seq#      : 8000000b
```

```
chksum    : 0x4cbe
```

Net mask : 255.255.255.0

Priority : Low

Attached Router 10.0.5.5

Attached Router 10.0.2.2

Attached Router 10.0.3.3

可以看到，这条Network LSA说明了TransNet 网络的掩码为255.255.255.0，连接到这个TransNet 网络的路由器有10.0.5.5（R5）、10.0.2.2（R2）、10.0.3.3（R3）。Network LSA中没有携带路径的开销，原因是Router LSA已经描述了自己到TransNet 网络的Cost 值。

在R2、R3、R5上查看区域0的LSDB。

<R2>display ospf lsdb

OSPF Process 10 with Router ID 10.0.2.2

Link State Database

Area: 0.0.0.0

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router 80000011	10.0.5.5 1	10.0.5.5	1068	48		
Router 8000000F	10.0.3.3 1	10.0.3.3	1076	36		
Router 1	10.0.2.2	10.0.2.2	1067	36	8000000D	
Network	10.0.235.5	10.0.5.5	1068	36	8000000D	0
Sum-Net	10.0.34.0	10.0.3.3	180	28	80000005	1

Sum-Net	10.0.12.0	10.0.2.2	172	28	80000005	1
Sum-Net 0	10.0.3.3	10.0.3.3	188	28	80000005	
Sum-Net 0	10.0.2.2	10.0.2.2	172	28	80000005	
Sum-Net 1	10.0.1.1	10.0.2.2	131	28	80000005	
Sum-Net 1	10.0.4.4	10.0.3.3	628	28	80000003	
Sum-Asbr 80000001	10.0.1.1 1	10.0.2.2	1005	28		

Area: 0.0.0.1

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router 1	10.0.2.2	10.0.2.2	132	48	80000009	
Router 80000009	10.0.1.1 1	10.0.1.1	1006	48		
Network	10.0.12.2	10.0.2.2	132	32	80000005	0
Sum-Net 2	10.0.34.0	10.0.2.2	1077	28	80000001	
Sum-Net 1	10.0.235.0	10.0.2.2	174	28	80000005	
Sum-Net 80000001	10.0.3.3 1	10.0.2.2	1078	28		

Sum-Net 1	10.0.5.5	10.0.2.2	128	28	80000005

Sum-Net 80000001	10.0.4.4 2	10.0.2.2	1078	28	

Sum-Asbr 80000001	10.0.3.3 1	10.0.2.2	1078	28	

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric

External 80000001	192.168.1.0 1	10.0.1.1	1007	36		

External 1	172.16.1.0	10.0.3.3	581	36	80000001	

<R3>displayospf lsdb

OSPF Process 10 with Router ID 10.0.3.3

Link State Database

Area: 0.0.0.0

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric

Router 80000011	10.0.5.5 1	10.0.5.5	1221	48		

Router 8000000F	10.0.3.3 1	10.0.3.3	1228	36		

Router 1	10.0.2.2	10.0.2.2	1221	36	8000000D	

Network	10.0.235.5	10.0.5.5	1221	36	8000000D	0

Sum-Net	10.0.34.0	10.0.3.3	333	28	80000005	1
Sum-Net	10.0.12.0	10.0.2.2	325	28	80000005	1
Sum-Net 0	10.0.3.3	10.0.3.3	340	28	80000005	
Sum-Net 0	10.0.2.2	10.0.2.2	325	28	80000005	
Sum-Net 1	10.0.1.1	10.0.2.2	284	28	80000005	
Sum-Net 1	10.0.4.4	10.0.3.3	779	28	80000003	
Sum-Asbr 80000001	10.0.1.1 1	10.0.2.2	1158	28		

Area: 0.0.0.2

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	777	48	8000000A	1
Router	10.0.4.4	10.0.4.4	731	48	8000000A	0
Network	10.0.34.4	10.0.4.4	778	32	80000004	0
Sum-Net 2	10.0.12.0	10.0.3.3	1226	28	80000001	
Sum-Net 1	10.0.235.0	10.0.3.3	340	28	80000005	
Sum-Net 80000001	10.0.2.2 1	10.0.3.3	1226	28		
Sum-Net 80000001	10.0.1.1 2	10.0.3.3	1226	28		

Sum-Net 80000001	10.0.5.5 1	10.0.3.3	1226	28	
NSSA 1	0.0.0.0	10.0.3.3	1230	36	80000001
NSSA 1	172.16.1.0	10.0.4.4	731	36	80000001

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External 8000000C	172.16.1.0 1	10.0.3.3	1054	36	
External 80000001	192.168.1.0 1	10.0.1.1	184	36	

<R5>display ospf lsdb

OSPF Process 10 with Router ID 10.0.5.5

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 1	10.0.5.5	10.0.5.5	683	48	80000011
Router 1	10.0.3.3	10.0.3.3	643	36	8000000F
Router	10.0.2.2	10.0.2.2	648	36	8000000D 1
Network	10.0.235.5	10.0.5.5	648	36	8000000D 0

Sum-Net	10.0.34.0	10.0.3.3	682	28	80000008	1
Sum-Net	10.0.12.0	10.0.2.2	132	28	80000001	1
Sum-Net 0	10.0.3.3	10.0.3.3	682	28	80000008	
Sum-Net 0	10.0.2.2	10.0.2.2	706	28	80000008	
Sum-Net	10.0.1.1	10.0.2.2	94	28	80000001	1
Sum-Net 1	10.0.4.4	10.0.3.3	642	28	80000008	
Sum-Asbr 80000001	10.0.1.1 1	10.0.2.2	1015	28		

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External 1	192.168.1.0	10.0.1.1	79	36	80000001
External 1	172.16.1.0	10.0.3.3	642	36	80000008

可以发现，R2、R3、R5的LSDB 中区域0 的Router LSA和Network LSA是完全一样的。

Router LSA和Network LSA可以完全描述本区域的网络拓扑，但这些LSA 不能泛洪到其他区域。当OSPF 网络包含多个区域时，通过Router LSA和Network LSA 就无法进行区域间路由的计算了。区域间路由的计算需要利用 Sum-Net LSA 来实现，ABR 路由器会将自己相连区域的Router LSA和Network LSA 转换为Sum-Net LSA，然后泛洪到其他区域。

R2同时连接了区域0和区域1，所以是一台ABR路由器。查看R2的LSDB。

```
<R2>display ospf lsdb
```

OSPF Process 10 with Router ID 10.0.2.2

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 8000000E	10.0.5.5 1	10.0.5.5	1592	48	
Router 8000000C	10.0.3.3 1	10.0.3.3	1551	36	
Router 8000000E	10.0.2.2 1	10.0.2.2	1556	36	
Network 8000000B	10.0.235.2 0	10.0.2.2	1556	36	
Sum-Net 1	10.0.34.0	10.0.3.3	1590	28	80000008
Sum-Net 1	10.0.12.0	10.0.2.2	1041	28	80000001
Sum-Net 80000008	10.0.3.3 0	10.0.3.3	1590	28	
Sum-Net 80000008	10.0.2.2 0	10.0.2.2	1613	28	

Sum-Net 80000001	10.0.1.1 1	10.0.2.2	1003	28
Sum-Net 80000008	10.0.4.4 1	10.0.3.3	1550	28
Sum-Asbr 80000006	10.0.1.1 1	10.0.2.2	1602	28

Area: 0.0.0.1

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router 1	10.0.2.2	10.0.2.2	995	48	80000011	
Router 80000006	10.0.1.1 1	10.0.1.1	1004	48		
Network	10.0.12.2	10.0.2.2	995	32	80000002	0
Sum-Net 2	10.0.34.0	10.0.2.2	1556	28	80000008	
Sum-Net 80000008	10.0.235.0 1	10.0.2.2	1613	28		
Sum-Net 80000008	10.0.3.3 1	10.0.2.2	1556	28		
Sum-Net 80000008	10.0.5.5 1	10.0.2.2	1565	28		
Sum-Net 80000008	10.0.4.4 2	10.0.2.2	1555	28		
Sum-Asbr 80000008	10.0.3.3 1	10.0.2.2	1556	28		

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External 80000001	192.168.1.0 1	10.0.1.1	143	36	
External 8000000C	172.16.1.0 1	10.0.3.3	1016	36	

可以看到，R2 的区域0 中有一条LinkState ID 为10.0.12.0 的Sum-Net LSA，它的AdvRouter为10.0.2.2。网段10.0.12.0/24本是属于区域1的网络，现在被ABR路由器R2 转换为Sum-Net LSA 并泛洪到了区域0 中。10.0.235.0/24 本是属于区域0 的网络，现在被ABR 路由器R2 转换为Sum-Net LSA并泛洪到了区域1 中。实际上，Sum-Net LSA是ABR利用自己相连区域的Router-LSA和Network-LSA来计算得到的路由信息的。

在R2 上查看LinkState ID 为10.0.12.0 的这条Sum-Net LSA的详细信息。

```
<R2>display ospf lsdb summary 10.0.12.0
```

OSPF Process 10 with Router ID 10.0.2.2

Area: 0.0.0.0

Link State Database

Type	:	Sum-Net
Ls id	:	10.0.12.0
Adv rtr	:	10.0.2.2
Ls age	:	703
Len	:	28
Options	:	E

seq# : 80000002
chksum : 0xf73f
Net mask : 255.255.255.0
Tos 0 metric : 1
Priority : Low

可以看到，这条LSA 的Type 为Sum-Net，Ls id 表明了目的网络地址为10.0.12.0，Net mask 表明了目的网络的掩码为255.255.255.0，Metric 表明了ABR 路由器R2 去往目的网络的Cost值为1。

在R5上查看LSDB，并查看路由表中关于10.0.12.0/24的路由信息。

<R5>display ospf lsdb

OSPF Process 10 with Router ID 10.0.5.5

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 80000010	10.0.5.5 1	10.0.5.5	1240	48	
Router 8000000E	10.0.3.3 1	10.0.3.3	1188	36	
Router 80000010	10.0.2.2 1	10.0.2.2	1195	36	
Network	10.0.235.5	10.0.5.5	1195	36	8000000D 0
Sum-Net	10.0.34.0	10.0.3.3	1227	28	8000000A 1

Sum-Net	10.0.12.0	10.0.2.2	680	28	80000003	1
Sum-Net	10.0.3.3	10.0.3.3	1227	28	8000000A	0
Sum-Net	10.0.2.2	10.0.2.2	1253	28	8000000A	0
Sum-Net	10.0.1.1	10.0.2.2	642	28	80000003	1

.....

<R5>display ip routing-table 10.0.12.0

Route Flags: R - relay, D - download to fib

Routing Table : Public

Summary Count : 1

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.12.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.235.2

可以看到，R5 的 LSDB 中存在 10.0.12.0 这条 Sum-Net LSA，R5 的路由表中关于10.0.12.0/24的这条路由信息表明R5去往10.0.12.0/24的Cost 为2。R5通过这条Sum-Net LSA得知网络中存在10.0.12.0/24网段，这个网段的AdvRouter为10.0.2.2（R2），R2自己到达10.0.12.0/24 的Cost 为1。R5和R2 同属于区域0，所以R5 可以通过Router LSA和Network LSA 计算出自己到R2的Cost 为1，因此，R5 可以计算出自己到10.0.12.0/24 的Cost值为1+1=2。

区域间的路由是根据Sum-Net LSA并结合Router LSA及Network-LSA计算出来的。对于某个区域的一台OSPF路由器来说，它无需了解其他区域的链路状态信息，但可以通过Sum-Net LSA并结合Router LSA及Network-LSA 计算出区域间路由；计算区域间路由时，采用的不再是链路状态算法，而是距离矢量算法。

在R2 上查看LinkState ID 为10.0.34.0/24 这条LSA 的信息。

```
<R2>display ospf lsdb summary 10.0.34.0
```

OSPF Process 10 with Router ID 10.0.2.2

Area: 0.0.0.0

Link State Database

Type	: Sum-Net
Ls id	: 10.0.34.0
Adv rtr	: 10.0.3.3
Ls age	: 442
Len	: 28
Options	: E
seq#	: 8000000b
chksum	: 0xe530
Net mask	: 255.255.255.0
Tos 0 metric	: 1
Priority	: Low

Area: 0.0.0.1

Link State Database

Type : Sum-Net
Ls id : 10.0.34.0
Adv rtr : 10.0.2.2
Ls age : 410
Len : 28
Options : E
seq# : 8000000b
chksum : 0xfc1a
Net mask : 255.255.255.0
Tos 0 metric : 2
Priority : Low

可以看到，10.0.34.0/24是属于区域2的网络，ABR路由器R3将关于10.0.34.0/24 的路由信息以Sum-Net LSA 的方式通告进了区域0，Cost为1。然后，ABR 路由器R2又继续将此信息以 Sum-Net LSA的方式通告进了区域0。

对于ABR 来说，如果在自己相连的某个区域的LSDB 中存在某条Sum-Net LSA，并且该Sum-Net LSA的AdvRouter 不是自己的Router-ID 时，就会将这条Sum-Net LSA的AdvRouter 修改为自己的Router-ID，并重新计算自己到达这条Sum-Net LSA 的Cost值，然后将之泛洪到与自己相连的其他区域中。

4．查看Type-4 LSA和Type-5 LSA

路由器可以通过Router LSA和Network LSA计算区域内的路由，可以通过Sum-Net LSA 并结合Router LSA和Network LSA计算区域间的路由，可以通过Sum-Asbr LSA 和External LSA计算AS 外部的路由。

R1 的 Loopback 1 是外部路由，被 ASBR 路由器 R1 引入到了 OSPF 网络中。查看R1的LSDB。

```
[R1]display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Link State Database
```

```
Area: 0.0.0.1
```

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router 1	10.0.2.2	10.0.2.2	828	48	80000015	
Router 8000000B	10.0.1.1 1	10.0.1.1	503	48		
Network	10.0.12.2	10.0.2.2	828	32	80000006	0
Sum-Net 2	10.0.34.0	10.0.2.2	1388	28	8000000C	
Sum-Net 8000000C	10.0.235.0 1	10.0.2.2	1445	28		
Sum-Net 8000000C	10.0.3.3 1	10.0.2.2	1388	28		
Sum-Net 8000000C	10.0.5.5 1	10.0.2.2	1397	28		

Sum-Net 8000000C	10.0.4.4 2	10.0.2.2	1387	28
Sum-Asbr 8000000C	10.0.3.3 1	10.0.2.2	1388	28

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External 80000001	192.168.1.0 1	10.0.1.1	503	36	
External 8000000C	172.16.1.0 1	10.0.3.3	1378	36	

可以看到，R1 的LSDB 中存在一条Type 为External，LinkState ID为192.168.1.0，AdvRouter为10.0.1.1的LSA。在R1上查看这条LSA的具体信息。

```
[R1]display ospf lsdb ase 192.168.1.0
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

Link State Database

```
Type           : External
Ls id           : 192.168.1.0
Adv rtr         : 10.0.1.1
Ls age          : 680
Len             : 36
Options         : E
```

seq# : 80000001
chksum : 0xda7f
Net mask : 255.255.255.0
TOS 0 Metric : 1
E type : 2
Forwarding Address : 0.0.0.0
Tag : 1
Priority : Low

可以看到，这条LSA的Type是External，AdvRouter为10.0.1.1（R1），这条LSA实际上是一条目的网络为192.168.1.0/24的AS外部路由，显示信息中的E Type（External Type）的值为2。

External LSA可以在整个AS内部泛洪（但不能泛洪到Stub区域、Totally Stub区域、NSSA区域和Totally NSSA区域中），在泛洪过程中其各个参数不会被改变。查看R2、R3、R4、R5的LSDB中是否也存在这条LSA。

```
<R2>display ospf lsdb ase 192.168.1.0
```

OSPF Process 10 with Router ID 10.0.2.2

Link State Database

Type : External
Ls id : 192.168.1.0
Adv rtr : 10.0.1.1
Ls age : 959

Len : 36
Options : E
seq# : 80000003
chksum : 0xd681
Net mask : 255.255.255.0
TOS 0 Metric : 1
E type : 2
Forwarding Address : 0.0.0.0
Tag : 1
Priority : Low

<R3>display ospf lsdb ase 192.168.1.0

OSPF Process 10 with Router ID 10.0.3.3

Link State Database

Type : External
Ls id : 192.168.1.0
Adv rtr : 10.0.1.1
Ls age : 967
Len : 36
Options : E
seq# : 80000003

chksum : 0xd681
Net mask : 255.255.255.0
TOS 0 Metric : 1
E type : 2
Forwarding Address : 0.0.0.0
Tag : 1
Priority : Low

<R4>display ospf lsdb ase 192.168.1.0

OSPF Process 10 with Router ID 10.0.4.4

<R5>display ospf lsdb ase 192.168.1.0

OSPF Process 10 with Router ID 10.0.5.5

Link State Database

Type : External
Ls id : 192.168.1.0
Adv rtr : 10.0.1.1
Ls age : 975
Len : 36
Options : E
seq# : 80000003
chksum : 0xd681

Netmas : 255.255.255.0

TOS 0 Metric : 1

E type : 2

Forwarding Address : 0.0.0.0

Tag : 1

Priority : Low

从上面的显示信息可以看到，R2、R3、R5 的LSDB 中都存在这条 External LSA，而且AdvRouter（10.0.1.1）等参数信息没有任何变化。需要注意的是，R4的LSDB中没有这条External LSA，这是因为R4处于NSSA区域中，而External LSA是不允许进入NSSA区域的。

R5 通过Link id 为192.168.1.0的External LSA得知，从自己去往192.168.1.0/24 是可以通过10.0.1.1（R1）到达的，并且知道从R1去往192.168.1.0/24的Cost为1。然而，R5并不知道从自己去往ASBR路由器R1的路由及Cost，所以R5还无法计算出从自己到达外部网络192.168.1.0/24 的路由及Cost。在OSPF 协议中，Sum-Asbr LSA 是用来描述去往ASBR的路由信息的。

查看R5 的LSDB 中LinkState ID 为10.0.1.1 的Sum-Asbr LSA的具体信息。

```
<R5>display ospf lsdb asbr 10.0.1.1
```

OSPF Process 10 with Router ID 10.0.5.5

Area: 0.0.0.0

Link State Database

Type : Sum-Asbr

Ls id : 10.0.1.1

Adv rtr : 10.0.2.2
Ls age : 1389
Len : 28
Options : E
seq# : 80000003
chksum : 0x57e7
Tos 0 metric : 1

可以看到，这条Sum-Asbr LSA 的AdvRouter是ABR 路由器R2 (10.0.2.2)，并且表明了从ABR路由器R2到ASBR路由器R1 (10.0.1.1) 的Cost值为1。

在R5 上使用display ospf abr-asbr 命令查看到达ABR 和ASBR 的Cost 值。

<R5>display ospf abr-asbr

OSPF Process 10 with Router ID 10.0.5.5

Routing Table to ABR and ASBR

RtType	Destination	Area	Cost	Nexthop	Type
Intra-area	10.0.2.2	0.0.0.0	1	10.0.235.2	ABR
Intra-area	10.0.3.3	0.0.0.0	1	10.0.235.3	
ABR/ASBR					
Inter-area	10.0.1.1	0.0.0.0	2	10.0.235.2	ASBR

可以看到，从R5到达ABR路由器R2的Cost值为1，从R5到达ASBR路由器R1的Cost值为2。由此可见，R5其实是通过Router LSA和Network LSA先计算出到达ABR路由器R2的Cost 值，然后加上Sum-Asbr LSA 所

表示的从ABR 路由器R2 到达ASBR路由器R1的Cost值，最终得出从自己到达ASBR路由器R1的Cost值。

Sum-Net LSA和Sum-Asbr LSA 的相同点是它们都由ABR 产生，并且其 AdvRouter在泛洪过程中会作相应的改变，不同点在于Sum-Net LSA 是用来计算区域间的路由的，而Sum-Asbr LSA 是用来计算到达ASBR 的路由的。如果网络中不存在ASBR，那就不会产生Sum-Asbr LSA，这也说明有External LSA存在时，才会有Sum-Asbr LSA。

在R1 上使用undo import-route 取消路由的引入。

[R1]ospf 10

[R1-ospf-10]undo import-route direct

在R5上查看LSDB。

<R5>display ospf lsdb

OSPF Process 10 with Router ID 10.0.5.5

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 80000013	10.0.5.5 1	10.0.5.5	1603	48	
Router 80000011	10.0.3.3 1	10.0.3.3	1614	36	
Router 8000000F	10.0.2.2 1	10.0.2.2	1605	36	

Network 8000000F	10.0.235.5 0	10.0.5.5	1603	36		
Sum-Net	10.0.34.0	10.0.3.3	718	28	80000007	1
Sum-Net	10.0.12.0	10.0.2.2	709	28	80000007	1
Sum-Net 0	10.0.3.3	10.0.3.3	725	28	80000007	
Sum-Net 0	10.0.2.2	10.0.2.2	709	28	80000007	
Sum-Net 1	10.0.1.1	10.0.2.2	669	28	80000007	
Sum-Net 80000005	10.0.4.4 1	10.0.3.3	1164	28		

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External 80000003	172.16.1.0 1	10.0.3.3	1116	36	

可以看到，LinkState ID为192.168.1.0的External LSA消失了，LinkState ID为10.0.1.1的Sum-Asbr LSA也随之消失了。

5．查看Type-7 LSA

NSSA 区域是不允许External LSA存在的，但NSSA 区域允许通过import-route 命令引入外部路由，那么如何来描述在NSSA区域中的AS 外部路由呢？NSSA区域引入的外部路由不能以External LSA的形式出现，取而代之的是使用NSSA LSA来描述NSSA区域中的AS 外部路由，且NSSA LSA 只能出现在NSSA 区域中。NSSA LSA 由NSSA区域的NSSAASBR 产生。

R4为NSSA区域的ASBR，查看R4的LSDB。

```
<R4>display ospf lsdb
```

OSPF Process 10 with Router ID 10.0.4.4

Link State Database

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	55	48	80000009	1
Router	10.0.4.4	10.0.4.4	47	48	80000009	1
Network	10.0.34.4	10.0.4.4	47	32	80000005	0
Sum-Net	10.0.12.0	10.0.3.3	37	28	80000004	2
Sum-Net	10.0.235.0	10.0.3.3	95	28	80000004	1
Sum-Net	10.0.2.2	10.0.3.3	37	28	80000004	1
Sum-Net	10.0.1.1	10.0.3.3	30	28	80000004	2
Sum-Net	10.0.5.5	10.0.3.3	17	28	80000004	1
NSSA	172.16.1.0	10.0.4.4	99	36	80000004	1
NSSA	0.0.0.0	10.0.3.3	37	36	80000004	1

可以看到，R4 为外部路由172.16.1.0 产生了相应的NSSA LSA。在R4上查看这条LSA的详细信息。

```
<R4>display ospf lsdb nssa 172.16.1.0
```

OSPF Process 10 with Router ID 10.0.4.4

Area: 0.0.0.2

Link State Database

Type : NSSA
Ls id : 172.16.1.0
Adv rtr : 10.0.4.4
Ls age : 222
Len : 36
Options : NP
seq# : 80000004
chksum : 0x3ea5
Net mask : 255.255.255.0
TOS 0 Metric : 1
E type : 2
Forwarding Address : 10.0.4.4
Tag : 1
Priority : Low

可以注意到，NSSA LSA的参数信息基本上和External LSA相同。

NSSA LSA 是特殊类型的LSA，只会出现在NSSA 区域中，不能泛洪到其他任何区域，那么其他区域的路由器又是如何计算去往NSSA LSA所表示的外部网络的路由呢？原来，NSSA 区域的ABR 会将NSSA LSA转换为External LSA，并泛洪到其他区域。

R3为NSSA区域的ABR路由器，在R3上查看LSDB信息。

```
<R3>display ospf lsdb
```

OSPF Process 10 with Router ID 10.0.3.3

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router 80000013	10.0.5.5 1	10.0.5.5	1730	48		
Router 80000011	10.0.3.3 1	10.0.3.3	1739	36		
Router 8000000F	10.0.2.2 1	10.0.2.2	1731	36		
Network 8000000F	10.0.235.5 0	10.0.5.5	1730	36		
Sum-Net	10.0.34.0	10.0.3.3	842	28	80000007	1
Sum-Net	10.0.12.0	10.0.2.2	834	28	80000007	1
Sum-Net 0	10.0.3.3	10.0.3.3	849	28	80000007	
Sum-Net 0	10.0.2.2	10.0.2.2	834	28	80000007	
Sum-Net 1	10.0.1.1	10.0.2.2	794	28	80000007	

Sum-Net 80000005	10.0.4.4 1	10.0.3.3	1289	28

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 8000000C	10.0.3.3 1	10.0.3.3	1288	48	
Router 8000000C	10.0.4.4 0	10.0.4.4	1242	48	
Network 0	10.0.34.3	10.0.3.3	1288	32	80000006
Sum-Net 2	10.0.12.0	10.0.3.3	1738	28	80000003
Sum-Net 1	10.0.235.0	10.0.3.3	849	28	80000007
Sum-Net 80000003	10.0.2.2 1	10.0.3.3	1738	28	

Sum-Net10.0.1.1 10.0.3.3 1738 28 80000003 2

Sum-Net10.0.5.5 10.0.3.3 1738 28 80000003 1

NSSA0.0.0.0 10.0.3.3 1741 36 80000003 1

NSSA172.16.1.0 10.0.4.4 1242 36 80000003 1

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric

External	172.16.1.0	10.0.3.3	1241	36	80000003	1

可以看到，由10.0.4.4 产生的NSSA LSA被R3 转换成了External LSA，并泛洪到其他区域。

思考

NSSA 区域的ABR 路由器是否会为NSSA LSA生成Network Summary LSA？

2.4 OSPF Stub 区域

原理概述

OSPF 协议定义了多种区域（Area）类型，其中比较常见的有Stub 区域和Totally Stub区域。区域的类型决定了在这个区域当中所存在的LSA的类型。

Stub 区域不允许Type-4和Type-5 LSA进入，该区域会通过Type-3 LSA所表示的缺省路由访问AS 外部目的地。Totally Stub 区域不仅不允许Type-4 和Type-5 LSA进入，同时也不允许Type-3 LSA进入，只允许表示缺省路由的Type-3 LSA 进入，并根据缺省路由来访问该区域以外的任何目的地。

Stub区域和Totally Stub区域的功能就是减少该区域中LSA的数量，从而缩小LSDB的规模，进而减少路由表中路由条目的数量，实现降低设备负担、增强网络稳定性、优化网络性能的目的。

配置Stub 和Totally Stub区域的时候需要注意以下几点：骨干区域（Area 0）不能被配置成为Stub区域或者Totally Stub区域，Virtual-link 不能通过Stub区域或者Totally Stub区域，Stub 区域或者Totally Stub 区域中不允许包含有ASBR 路由器。

实验目的

- 理解Stub 区域和Totally Stub 区域的作用与区别

- 掌握Stub 区域和Totally Stub 区域的配置方法

实验内容

实验拓扑如图2-9所示，实验编址如表2-4所示。本实验模拟了一个企业网络场景，R1、R2、R3为公司总部网络的路由器，R4、R5分别为企业分支机构1和分支机构2的路由器，并且都采用双上行方式与企业总部相连。整个网络都运行OSPF协议，R1、R2、R3之间的链路位于区域0，R4与R1、R4与R2之间的两条链路位于区域1，R5与R1、R5与R2之间的两条链路位于区域2，R3的Loopback 1 接口用来模拟企业外部网络。网络的最终需求是：不同分支机构通过不同的总部路由器访问总部网络及外网，并实现主备备份，即：R4与R1之间为分支机构1的主用链路，R4与R2之间为其备用链路；R5与R2之间为分支机构2的主用链路，R5与R1之间为其备用链路。另外，R4和R5的LSDB及路由表的规模应尽量小。

实验拓扑

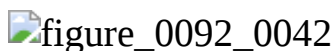
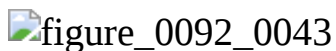
figure_0092_0042

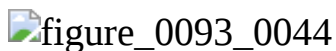
图2-9 OSPF Stub区域

实验编址表

表2-4 实验编址

figure_0092_0043

(续表)

figure_0093_0044

实验步骤

1. 基本配置

根据图2-9和表2-4进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=50 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 50/50/50 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF及路由引入

在每台路由器上配置OSPF协议，其中R1、R2、R3之间的链路位于区域0，R4与R1、R4与R2之间的链路位于区域1，R5与R1、R5与R2之间的链路位于区域2。

```
[R1]router id 1.1.1.1
```

```
[R1]ospf
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.13.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 1.1.1.1 0.0.0.0
```

[R1-ospf-1-area-0.0.0.0]area 1

[R1-ospf-1-area-0.0.0.1]network 10.0.14.1 0.0.0.0

[R1-ospf-1-area-0.0.0.1]area 2

[R1-ospf-1-area-0.0.0.2]network 10.0.15.1 0.0.0.0

[R2]router id 2.2.2.2

[R2]ospf

[R2-ospf-1-area-0.0.0.0]network 10.0.12.2 0.0.0.0

[R2-ospf-1-area-0.0.0.0]network 10.0.23.2 0.0.0.0

[R2-ospf-1-area-0.0.0.0]network 2.2.2.2 0.0.0.0

[R2-ospf-1-area-0.0.0.0]area 1

[R2-ospf-1-area-0.0.0.1]network 10.0.24.2 0.0.0.0

[R2-ospf-1-area-0.0.0.1]area 2

[R2-ospf-1-area-0.0.0.2]network 10.0.25.2 0.0.0.0

[R3]router id 3.3.3.3

[R3]ospf

[R3-ospf-1]area 0

[R3-ospf-1-area-0.0.0.0]network 10.0.13.3 0.0.0.0

[R3-ospf-1-area-0.0.0.0]network 10.0.23.3 0.0.0.0

[R3-ospf-1-area-0.0.0.0]network 3.3.3.3 0.0.0.0

[R4]router id 4.4.4.4


```
[R4]ospf
```

```
[R4-ospf-1]area 1
```

```
[R4-ospf-1-area-0.0.0.1]network 10.0.14.4 0.0.0.0
```

```
[R4-ospf-1-area-0.0.0.1]network 10.0.24.4 0.0.0.0
```

```
[R4-ospf-1-area-0.0.0.1]network 4.4.4.4 0.0.0.0
```

```
[R5]router id 5.5.5.5
```

```
[R5]ospf
```

```
[R5-ospf-1]area 2
```

```
[R5-ospf-1-area-0.0.0.2]network 10.0.15.5 0.0.0.0
```

```
[R5-ospf-1-area-0.0.0.2]network 10.0.25.5 0.0.0.0
```

```
[R5-ospf-1-area-0.0.0.2]network 5.5.5.5 0.0.0.0
```

配置完成后，查看R1、R2上的OSPF邻居建立情况。

```
<R1>display ospf peer brief
```

```
OSPF Process 1 with Router ID 1.1.1.1
```

```
Peer Statistic Information
```

```
-----
```

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	3.3.3.3	Full
0.0.0.0	GigabitEthernet0/0/1	2.2.2.2	Full
0.0.0.1	Ethernet1/0/0	4.4.4.4	Full

0.0.0.2 Ethernet1/0/1 5.5.5.5 Full

<R2>display ospf peer brief

OSPF Process 1 with Router ID 2.2.2.2

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	3.3.3.3	Full
0.0.0.0	GigabitEthernet0/0/1	1.1.1.1	Full
0.0.0.1	Ethernet1/0/0	4.4.4.4	Full
0.0.0.2	Ethernet1/0/1	5.5.5.5	Full

可以看到，邻居状态都是Full，说明邻居邻接关系都已成功建立。

查看R4的路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 16 Routes : 19

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
1.1.1.1/32	OSPF	10	1	D	10.0.14.1 Ethernet1/0/0

2.2.2.2/32	OSPF	10	1	D	10.0.24.2	Ethernet1/0/1
3.3.3.3/32	OSPF	10	2	D	10.0.24.2	Ethernet1/0/1
OSPF	10	2	D	10.0.14.1	Ethernet1/0/0	
4.4.4.4/32	Direct	0	0	D	127.0.0.1	LoopBack0
5.5.5.5/32	OSPF	10	2	D	10.0.24.2	Ethernet1/0/1
OSPF	10	2	D	10.0.14.1	Ethernet1/0/0	
10.0.12.0/24	OSPF	10	2	D	10.0.24.2	Ethernet1/0/1
OSPF	10	2	D	10.0.14.1	Ethernet1/0/0	
10.0.13.0/24	OSPF	10	2	D	10.0.14.1	Ethernet1/0/0
10.0.14.0/24	Direct	0	0	D	10.0.14.4	Ethernet1/0/0
10.0.14.4/32	Direct	0	0	D	127.0.0.1	Ethernet1/0/0
10.0.15.0/24	OSPF	10	2	D	10.0.14.1	Ethernet1/0/0
10.0.23.0/24	OSPF	10	2	D	10.0.24.2	Ethernet1/0/1
10.0.24.0/24	Direct	0	0	D	10.0.24.4	Ethernet1/0/1
10.0.24.4/32	Direct	0	0	D	127.0.0.1	Ethernet1/0/1
10.0.25.0/24	OSPF	10	2	D	10.0.24.2	Ethernet1/0/1
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0

可以看到，现在R4已经获得了所有其他网段的路由。由于R4采用了双出口设计，所以其中部分路由条目同时有两个下一跳，即通过R1或者R2都可以访问，处在负载均衡状态。

在R3 上配置路由引入，采用引入直连路由的方式将Loopback 1 接口所在网段引入到OSPF进程中，用它来模拟企业外部网络。

```
[R3]ospf
```

```
[R3-ospf-1]import-route direct
```

在R4上查看其路由表。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 17 Routes : 21

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
1.1.1.1/32	OSPF	10	1	D	10.0.14.1 Ethernet1/0/0
.....					
10.0.25.0/24	OSPF	10	2	D	10.0.24.2 Ethernet1/0/1
20.0.0.0/24	O_ASE	150	1	D	10.0.24.2 Ethernet1/0/1
O_ASE 150 1	D		10.0.14.1		Ethernet1/0/0
127.0.0.0/8	Direct	0	0	D	127.0.0.1 InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1 InLoopBack0

可以看到，此时R4已获得了该企业外部网络的路由，并且也是负载均衡方式。OSPF的外部路由在路由表中显示为O_ASE，其优先级的值为

150，远远大于普通OSPF内部路由优先级的值10。另外，我们也可以使用命令display ospf 1 routing来只查看OSPF路由表的信息。

```
<R4>display ospf 1 routing
```

OSPF Process 1 with Router ID 4.4.4.4

Routing Tables

Routing for Network

Destination	Cost	Type	NextHop	AdvRouter	Area
4.4.4.4/32	0	Stub	4.4.4.4	4.4.4.4	0.0.0.1

.....

Routing for ASEs

Destination	Cost	Type	Tag	NextHop	AdvRouter
20.0.0.0/24	1	Type2	1	10.0.24.2	3.3.3.3
20.0.0.0/24	1	Type2	1	10.0.14.1	3.3.3.3

Total Nets: 17

Intra Area: 3 Inter Area: 12 ASE: 2 NSSA: 0

显示信息表明，R4 拥有两条去往外部网络 20.0.0.0/24 的路由，下一跳分别是 R2（10.0.24.2）和R1（10.0.14.1），开销值都为1，类型为OSPF外部路由的默认类型2。注意，使用这种方式查看OSPF路由信息时，无法看到路由的优先级的值。

查看R4的LSDB。

```
<R4>display ospf lsdb
```

OSPF Process 1 with Router ID 4.4.4.4

Link State Database

Area: 0.0.0.1

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	4.4.4.4	4.4.4.4	250	60	8000000A	1
.....						
Sum-Net	10.0.23.0	1.1.1.1	251	28	80000003	2
Sum-Asbr	3.3.3.3	2.2.2.2	904	28	80000001	1
Sum-Asbr	3.3.3.3	1.1.1.1	904	28	80000001	1

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	10.0.13.0	3.3.3.3	906	36	80000001	1
External	20.0.0.0	3.3.3.3	906	36	80000001	1
External	10.0.23.0	3.3.3.3	906	36	80000001	1
External	3.3.3.3	3.3.3.3	906	36	80000001	1

可以看到，R4 的LSDB 中包含了若干条各种类型的LSA，在External LSA（即Type-5 LSA）中，存在一条LinkState ID 为20.0.0.0的LSA，通告路由器为R3。同时，在LSDB中还包含了两条LinkState ID为3.3.3.3的Type-4 LSA（Sum-Asbr LSA），通告路由器分别为R1 和R2，表示了两条去往ASBR R3 的路由。

此外，还可以注意到，在LSDB中，除了表示企业外部网络20.0.0.0的那条External LSA 之外，还存在着另外3 条External LSA，其原因是此前采取了直接引入直连路由的方式来引入外部路由，所以将R3上的所有直连网段的路由全部引入了进来。也就是说，现在R4可以通过两种

方式获得这3条路由（10.0.13.0/24，10.0.23.0/24，3.3.3.3/32），一种是在OSPF内部获得，一种是通过OSPF外部获得。在这种情况下，会首先比较两种不同方式下的路由优先级：OSPF内部路由优先级的值为10，而外部路由优先级的值为150，所以最终的选择结果应该是从内部获得该3条路由（注意，优先级的值越大，优先级越低）。

在R4上测试去往外部网络的连通性。

```
<R4>ping 20.0.0.1
```

```
PING 20.0.0.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 20.0.0.1: bytes=56 Sequence=1 ttl=254 time=220 ms
```

```
Reply from 20.0.0.1: bytes=56 Sequence=2 ttl=254 time=70 ms
```

```
Reply from 20.0.0.1: bytes=56 Sequence=3 ttl=254 time=50 ms
```

```
Reply from 20.0.0.1: bytes=56 Sequence=4 ttl=254 time=50 ms
```

```
Reply from 20.0.0.1: bytes=56 Sequence=5 ttl=254 time=60 ms
```

```
--- 20.0.0.1 ping statistics ---
```

```
5 packet(s) transmitted
```

```
5 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 50/90/220 ms
```

可以看到，R4与外部网络的通信是正常的。读者可自行验证，R5与外部网络的通信也是正常的。

3．配置Stub区域

当前情况下，两个分支机构在访问总部网络和外网时，是可以同时通过总部路由器R1和R2进行访问的。接下来的需求是，不同分支机构应通过不同的总部路由器访问总部网络及外网，并实现主备备份，即：R4与R1之间为分支机构1的主用链路，R4与R2之间为其备用链路；R5与R2之间为分支机构2的主用链路，R5与R1之间为其备用链路。

另外，R4和R5作为企业分支机构的路由器，只需要能够正常与总部网络和外网进行通信即可，没有必要获取及维护外网的明细路由。为此，可以将R4和R5各自所在的区域配置成为Stub 区域。配置成Stub 区域后，该区域内的路由器将不会接收区域外部路由，且ABR会在该区域中通告一条缺省路由，以供其访问区域外部网络。

配置Stub区域时必须注意，区域内的所有路由器都要配置stub命令，否则邻居关系无法正常建立。在配置过程中可以观察到，配置了stub命令的路由器与尚未配置stub命令的路由器的邻居关系处于down的状态。

```
[R1]ospf
```

```
[R1-ospf-1]area 1
```

```
[R1-ospf-1-area-0.0.0.1]stub
```

```
[R1-ospf-1-area-0.0.0.1]area 2
```

```
[R1-ospf-1-area-0.0.0.2]stub
```

```
[R2]ospf
```

```
[R2-ospf-1]area 1
```

```
[R2-ospf-1-area-0.0.0.1]stub
```

```
[R2-ospf-1-area-0.0.0.1]area 2
```

```
[R2-ospf-1-area-0.0.0.2]stub
```

```
[R4]ospf
```



```
[R4-ospf-1]area 1
```

```
[R4-ospf-1-area-0.0.0.1]stub
```

```
[R5]ospf
```

```
[R5-ospf-1]area 2
```

```
[R5-ospf-1-area-0.0.0.2]stub
```

配置完成后，查看R4的路由表及LSDB。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 17 Routes : 21

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
0.0.0.0/0	OSPF	10	2	D	10.0.14.1 Ethernet1/0/0

OSPF	10	2	D	10.0.24.2	Ethernet1/0/1

1.1.1.1/32

OSPF	10	1	D	10.0.14.1	Ethernet1/0/0

.....

```
<R4>display ospf lsdb
```

OSPF Process 1 with Router ID 4.4.4.4

Link State Database

Area: 0.0.0.1

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	4.4.4.4	4.4.4.4	665	60	80000007	1
Router	2.2.2.2	2.2.2.2	669	36	80000005	1
Router	1.1.1.1	1.1.1.1	697	36	80000006	1
Network	10.0.14.4	4.4.4.4	697	32	80000002	0
Network	10.0.24.4	4.4.4.4	665	32	80000002	0
Sum-Net	0.0.0.0	1.1.1.1	761	28	80000001	1
Sum-Net	0.0.0.0	2.2.2.2	729	28	80000001	1
Sum-Net	10.0.15.0	1.1.1.1	757	28	80000003	1
.....						

可以看到，现在 R4 的路由表中的外部路由条目已经消失了，取而代之的是一条缺省路由。同样，在R4的LSDB 中，已经没有了任何Type-5 LSA及Type-4 LSA条目，并且多了两条Type-3 LSA（Sum-Net LSA）。这两条Type-3 LSA的LinkState ID为0.0.0.0，说明是表示缺省路由的LSA，通告路由器分别为R1（1.1.1.1）和R2（2.2.2.2）。

在R5上也可以看到与上面相同的结果，读者可自行去查看R5的路由表及LSDB。

接下来，通过调整ABR路由器所通告的缺省路由的开销值来实现主备备份。在R2的区域1 中，配置命令default cost 10，表示将发送到该Stub 区域的Type-3 LSA的缺省路由开销值设为10。同样，在R1 的区域2 中，配置命令default cost 10。

```
[R2]ospf
```

```
[R2-ospf-1]area 1
```

```
[R2-ospf-1-area-0.0.0.1]default-cost 10
```

```
[R1]ospf
```

```
[R1-ospf-1]area 2
```

```
[R1-ospf-1-area-0.0.0.2]default-cost 10
```

配置完成后，查看R4、R5的LSDB。

```
<R4>display ospf lsdb
```

```
OSPF Process 1 with Router ID 4.4.4.4
```

```
Link State Database
```

```
Area: 0.0.0.1
```

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	4.4.4.4	4.4.4.4	701	60	8000000B	1
Router	2.2.2.2	2.2.2.2	703	36	8000000B	1
Router	1.1.1.1	1.1.1.1	703	36	8000000C	1
Network	10.0.14.4	4.4.4.4	703	32	80000005	0
Network	10.0.24.4	4.4.4.4	703	32	80000002	0
Sum-Net	0.0.0.0	2.2.2.2	511	28	80000004	10
Sum-Net	0.0.0.0	1.1.1.1	417	28	80000006	1
Sum-Net	10.0.15.0	2.2.2.2	1285	28	80000003	2

.....

<R5>display ospf lsdb

OSPF Process 1 with Router ID 5.5.5.5

Link State Database

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	2.2.2.2	2.2.2.2	1192	36	80000007	1
Router	1.1.1.1	1.1.1.1	548	36	8000000C	1
Router	5.5.5.5	5.5.5.5	548	60	8000000A	1
Network	10.0.15.5	5.5.5.5	548	32	80000005	0
Network	10.0.25.5	5.5.5.5	1192	32	80000003	0
Sum-Net	0.0.0.0	2.2.2.2	533	28	80000004	1
Sum-Net	0.0.0.0	1.1.1.1	482	28	80000003	10
Sum-Net	10.0.14.0	2.2.2.2	1296	28	80000002	2

.....

可以看到，R4 和R5 的LSDB 中相应的Type-3 LSA的开销值已经得到了修改。

再查看R4、R5的路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 20

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0	OSPF	10	2	D	10.0.14.1 Ethernet1/0/0
1.1.1.1/32	OSPF	10	1	D	10.0.14.1 Ethernet1/0/0
.....					

<R5>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 20

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0	OSPF	10	2	D	10.0.25.2 Ethernet1/0/1
1.1.1.1/32	OSPF	10	1	D	10.0.15.1 Ethernet1/0/0
.....					

可以看到，现在R4的路由表中的缺省路由的下一跳是
R1（10.0.14.1），R5的路由表中的缺省路由的下一跳是
R2（10.0.25.2）。

在R4、R5上使用tracert命令验证去往外网20.0.0.1的路径。

```
<R4>tracert 20.0.0.1
```

```
tracert to 20.0.0.1(20.0.0.1), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.14.1 110 ms 50 ms 40 ms
```

```
2 10.0.13.3 110 ms 80 ms 60 ms
```

```
<R5>tracert 20.0.0.1
```

```
tracert to 20.0.0.1(20.0.0.1), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.25.2 80 ms 50 ms 10 ms
```

```
2 10.0.23.3 80 ms 60 ms 70 ms
```

可以看到，R4和R5都选择了主用链路去访问外网。关于主备链路的切换过程，读者可自行去进行实验验证。

4 . 配置Totally Stub 区域

上面的实验已经基本上实现了该企业的网络需求。然而，在仔细观察了分支路由器R4 和R5的LSDB 后发现，LSDB 中存在着一些Type-3 LSA，即维护着一些域间路由信息。随着今后企业的发展，网络的扩容，这些Type-3 LSA的数量将大量增加，但本身又没有什么用处，从而成为路由器的不必要的负担。

解决这个问题一个有效方法是配置Totally Stub 区域。Totally Stub 区域是在Stub 区域的基础之上进一步拒绝接收除缺省路由之外的域间路由信息，即禁止Type-3 LSA进入该区域。配置Totally Stub 区域时，只需在stub 命令之后添加no-summary 选项，且只需在ABR上进行配置。

下面进行Totally Stub 区域的配置。注意，由于分支路由器R4 和R5 与总部之间是双出口设计，所以每个区域中都存在两台ABR。

```
[R1]ospf
```

```
[R1-ospf-1]area 1
```

```
[R1-ospf-1-area-0.0.0.1]stub no-summary
```

```
[R1-ospf-1]area 2
```

```
[R1-ospf-1-area-0.0.0.2]stub no-summary
```

```
[R2]ospf
```

```
[R2-ospf-1]area 1
```

```
[R2-ospf-1-area-0.0.0.1]stub no-summary
```

```
[R2-ospf-1]area 2
```

```
[R2-ospf-1-area-0.0.0.2]stub no-summary
```

配置完成后，以R4为例，查看此时R4的LSDB和路由表。

```
<R4>display ospf lsdb
```

```
OSPF Process 1 with Router ID 4.4.4.4
```

```
Link State Database
```

```
Area: 0.0.0.1
```

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	4.4.4.4	4.4.4.4	58	60	80000016	1
Router	2.2.2.2	2.2.2.2	64	36	8000000E	1

Router	1.1.1.1	1.1.1.1	95	36	8000000F	1
Network	10.0.14.4	4.4.4.4	93	32	80000002	0
Network	10.0.24.4	4.4.4.4	58	32	80000002	0
Sum-Net 10	0.0.0.0	2.2.2.2	66	28	80000006	
Sum-Net	0.0.0.0	1.1.1.1	96	28	80000008	1

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 8 Routes : 8

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0	OSPF	10	2	D	10.0.14.1 Ethernet1/0/0
4.4.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.14.0/24	Direct	0	0	D	10.0.14.4 Ethernet1/0/0
10.0.14.4/32	Direct	0	0	D	127.0.0.1 Ethernet1/0/0
10.0.24.0/24	Direct	0	0	D	10.0.24.4 Ethernet1/0/1
10.0.24.4/32	Direct	0	0	D	127.0.0.1 Ethernet1/0/1
127.0.0.0/8 InLoopBack0	Direct	0	0	D	127.0.0.1


```
127.0.0.1/32    Direct    0    0    D        127.0.0.1
InLoopBack0
```

可以看到，R4 的 LSDB 中只有两条表示缺省路由的 Type-3 LSA，没有任何其他 Type-3 LSA，路由表中也不存在任何域间路由，只有一条缺省路由。

以R4为例，测试企业分支机构路由器与企业总部路由器R1的环回接口所在网段的连通性，以及与外部网络的连通性。

```
<R4>ping 1.1.1.1
```

```
PING 1.1.1.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 1.1.1.1: bytes=56 Sequence=1 ttl=255 time=30 ms
```

```
Reply from 1.1.1.1: bytes=56 Sequence=2 ttl=255 time=30 ms
```

```
Reply from 1.1.1.1: bytes=56 Sequence=3 ttl=255 time=30 ms
```

```
Reply from 1.1.1.1: bytes=56 Sequence=4 ttl=255 time=50 ms
```

```
Reply from 1.1.1.1: bytes=56 Sequence=5 ttl=255 time=20 ms
```

```
--- 1.1.1.1 ping statistics ---
```

```
5 packet(s) transmitted
```

```
5 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/32/50 ms
```

```
<R4>ping 20.0.0.1
```

```
PING 20.0.0.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 20.0.0.1: bytes=56 Sequence=1 ttl=254 time=60 ms
```

Reply from 20.0.0.1: bytes=56 Sequence=2 ttl=254 time=50 ms

Reply from 20.0.0.1: bytes=56 Sequence=3 ttl=254 time=40 ms

Reply from 20.0.0.1: bytes=56 Sequence=4 ttl=254 time=80 ms

Reply from 20.0.0.1: bytes=56 Sequence=5 ttl=254 time=70 ms

--- 20.0.0.1 ping statistics ---

5 packet(s) transmitted

5 packet(s) received

0.00% packet loss

round-trip min/avg/max = 40/60/80 ms

可以看到，通信完全正常。至此，网络需求已经得到完全满足。

思考

OSPF网络中，ASBR能够配置在Stub区域中吗？骨干区域能否配置成Stub区域？为什么？

2.5 OSPF NSSA 区域

原理概述

OSPF 协议定义了 Stub 区域和 Totally Stub 区域这两种特殊的非骨干区域，为的是精简LSDB中LSA的数量，同时也精简路由表中的路由条目数量，实现优化设备和网络性能的目的。根据定义，Stub 区域或 Totally Stub 区域中是不允许存在 ASBR路由器的。

然而，在实际环境中，由于某种需要，有可能希望在Stub 区域或 Totally Stub 区域引入外部路由。为此，OSPF 又定义了NSSA 区域和 Totally NSSA区域，以此来进一步增强OSPF协议的适应和扩展能力。

NSSA 区域或Totally NSSA区域可以将外部路由以Type-7 LSA (NSSA LSA) 的方式引进本区域，这些Type-7 LSA将在本区域的ABR 路由器上被转换为Type-5 LSA (AS External LSA) 并泛洪到其他OSPF 区域中。Type-7 LSA只会出现在NSSA 区域或Totally NSSA区域中。

在其他方面，NSSA 区域和Totally NSSA 区域是与Stub 区域和Totally Stub区域完全一样的。NSSA 区域不允许Type-4和Type-5 LSA进入，该区域会通过Type-3 LSA所表示的缺省路由访问AS外部目的地。Totally NSSA区域不仅不允许Type-4和Type-5 LSA进入，同时也不允许Type-3 LSA 进入，只允许表示缺省路由的Type-3 LSA进入，并根据缺省路由来访问该区域以外的任何目的地。

实验目的

- 理解NSSA 区域和Totally NSSA区域的作用与区别
- 掌握NSSA 区域和Totally NSSA区域的配置方法
- 掌握修改NSSA 区域缺省路由开销值的方法

实验内容

实验拓扑如图2-10所示，实验编址如表2-5所示。本实验模拟了一个企业网络场景，路由器R1、R2、R3为企业总部网络路由器，R4为企业的分支机构的路由器。R1与R2、R1与R3之间的链路位于区域0，R4与R2、R4与R3之间的链路位于区域1。R1的所有Loopback接口用来模拟企业总部的非OSPF网络，R4的所有Loopback接口用来模拟企业分支机构的非OSPF网络。网络需求是：全网互通，且分支机构在访问总部网络时优先使用经由R2的路径，并尽量精简LSDB和路由表。

实验拓扑

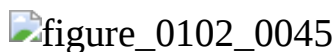
figure_0102_0045

图2-10 OSPF NSSA区域

实验编址表

表2-5 实验编址

figure_0103_0046

实验步骤

1．基本配置

根据图2-10和表2-5进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=20 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/20/20 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF及路由引入

在每台路由器上配置OSPF协议，其中R1与R2、R1与R3之间的链路属于区域0，R4与R2、R4与R3之间的链路属于区域1，R1和R4上所有Loopback接口都属于外部网络。如果采取引入直连路由的方式来引入Loopback接口的路由，将会导致R1、R4上所有的直连网段的路由全部被引入进来，在查看LSDB数据库时会发现R1和R4的所有直连网段

路由都将作为 Type-5 LSA 出现在 LSDB 中。因此，最好的方法是在R1和R4 路由器上利用Route-Policy 将Loopback 0 和Loopback 1 接口引入到OSPF 网络中。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]rule 5 permit source 10.0.1.1 0.0.0.0
```

```
[R1-acl-basic-2000]rule 10 permit source 10.0.11.11 0.0.0.0
```

```
[R1-acl-basic-2000]quit
```

```
[R1]route-policy 10 permit node 10
```

```
[R1-route-policy]if-match acl 2000
```

```
[R1-route-policy]quit
```

```
[R1]ospf 10 router-id 10.0.1.1
```

```
[R1-ospf-10]import-route direct route-policy 10
```

```
[R1-ospf-10]area 0
```

```
[R1-ospf-10-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R1-ospf-10-area-0.0.0.0]network 10.0.13.0 0.0.0.255
```

```
[R2]ospf 10 router-id 10.0.2.2
```

```
[R2-ospf-10]area 0
```

```
[R2-ospf-10-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

```
[R2-ospf-10-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R2-ospf-10-area-0.0.0.0]area 1
```

```
[R2-ospf-10-area-0.0.0.1]network 10.0.24.0 0.0.0.255
```

```
[R3]ospf 10 router-id 10.0.3.3
[R3-ospf-10]area 0
[R3-ospf-10-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R3-ospf-10-area-0.0.0.0]network 10.0.13.0 0.0.0.255
[R3-ospf-10-area-0.0.0.0]area 1
[R3-ospf-10-area-0.0.0.1]network 10.0.34.0 0.0.0.255
[R4]acl 2000
[R4-acl-basic-2000]rule 5 permit source 10.0.4.4 0.0.0.0
[R4-acl-basic-2000]rule 10 permit source 10.0.44.44 0.0.0.0
[R4-acl-basic-2000]quit
[R4]route-policy 10 permit node 10
[R4-route-policy]if-match acl 2000
[R4-route-policy]quit
[R4]ospf 10 router-id 10.0.4.4
[R4-ospf-10]import-route direct route-policy 10
[R4-ospf-10]area 1
[R4-ospf-10-area-0.0.0.1]network 10.0.24.0 0.0.0.255
[R4-ospf-10-area-0.0.0.1]network 10.0.34.0 0.0.0.255
```

配置完成后，查看R1的LSDB。

```
<R1>display ospf lsdb
```

OSPF Process 10 with Router ID 10.0.1.1

Link State Database

Area: 0.0.0.0

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router 1	10.0.3.3	10.0.3.3	1201	48	8000000C	
Router 1	10.0.2.2	10.0.2.2	1202	48	8000000C	
Router 1	10.0.1.1	10.0.1.1	1201	48	8000001B	
Network	10.0.13.3	10.0.3.3	1201	32	80000002	0
Network	10.0.12.2	10.0.2.2	1202	32	80000002	0
Sum-Net	10.0.34.0	10.0.3.3	1701	28	80000001	1
Sum-Net	10.0.34.0	10.0.2.2	1210	28	80000001	2
Sum-Net	10.0.24.0	10.0.3.3	1226	28	80000001	2
Sum-Net	10.0.24.0	10.0.2.2	1220	28	80000001	1
Sum-Asbr 1	10.0.4.4	10.0.3.3	1226	28	80000001	
Sum-Asbr 1	10.0.4.4	10.0.2.2	1210	28	80000001	

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	10.0.1.1	10.0.1.1	307	36	80000001	1

External	10.0.11.11	10.0.1.1	307	36	80000001	1
External	10.0.4.4	10.0.4.4	866	36	80000001	1
External	10.0.44.44	10.0.4.4	499	36	80000001	1

可以看到，R1的LSDB中有4条Typ-5 LSA（ External LSA ），同时还有两条LinkState ID为10.0.4.4、通告路由器分别为R2和R3的Type-4 LSA（ Sum-Asbr LSA ）。

查看R4的LSDB。

```
<R4>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.4.4
```

```
Link State Database
```

```
Area: 0.0.0.1
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	1321	36	8000000A	1
.....						
Sum-Net	10.0.2.2	10.0.3.3	1291	28	80000001	2
Sum-Asbr	10.0.1.1	10.0.2.2	1302	28	80000001	1
Sum-Asbr	10.0.1.1	10.0.3.3	1291	28	80000001	1

```
AS External Database
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	10.0.4.4	10.0.4.4	1335	36	80000001	1

External	10.0.44.44	10.0.4.4	762	36	80000001	1
External	10.0.1.1	10.0.1.1	2183	36	80000001	1
External	10.0.11.11	10.0.1.1	1284	36	80000001	1

可以看到，R4 的 LSDB 中也有 4 条 Type-5 LSA，同时还有两条 LinkState ID 为10.0.1.1、通告路由器分别为R2 和R3 的Type-4 LSA。

查看R1的路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
-----
```

Routing Tables: Public

Destinations : 17 Routes : 19

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0 0	D		127.0.0.1 LoopBack0
10.0.2.2/32 GigabitEthernet0/0/0	OSPF	10 1	D		10.0.12.2
10.0.3.3/32 GigabitEthernet0/0/1	OSPF	10 1	D		10.0.13.3
10.0.4.4/32 GigabitEthernet0/0/0	O_ASE	150 1	D		10.0.12.2
O_ASE	150 1	D	10.0.13.3		GigabitEthernet0/0/1

10.0.12.0/24	Direct	0	0	D	10.0.12.1	
GigabitEthernet0/0/0						
10.0.12.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.12.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.13.0/24	Direct	0	0	D	10.0.13.1	
GigabitEthernet0/0/1						
10.0.13.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.13.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.24.0/24	OSPF	10	2	D	10.0.12.2	
GigabitEthernet0/0/0						
10.0.34.0/24	OSPF	10	2	D	10.0.13.3	
GigabitEthernet0/0/1						
10.0.44.44/32	O_ASE	150	1	D	10.0.12.2	
GigabitEthernet0/0/0						
O_ASE	150	1	D	10.0.13.3	GigabitEthernet0/0/1	
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

可以看到，R1已经接收到了外部路由10.0.4.4/32和10.0.44.44/32。

查看R4的路由表。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 17 Routes : 19

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

10.0.1.1/32 GigabitEthernet0/0/1	O_ASE	150	1	D	10.0.24.2

O_ASE	150	1	D	10.0.34.3	GigabitEthernet0/0/0

10.0.2.2/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.24.2

10.0.3.3/32 GigabitEthernet0/0/0	OSPF	10	1	D	10.0.34.3

10.0.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack0

10.0.11.11/32 GigabitEthernet0/0/1	O_ASE	150	1	D	10.0.24.2

O_ASE	150	1	D	10.0.34.3	GigabitEthernet0/0/0

10.0.12.0/24 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.24.2

.....

可以看到，R4也已经接收到了外部路由10.0.1.1/32和10.0.11.11/32。

3．配置NSSA 和Totally NSSA区域

目前，企业内部的网络以及企业总部和企业分支的非OSPF网络都实现了互通。为了减小区域1内LSDB的规模，管理员决定将区域1配置为OSPF的特殊区域。由于区域1存在ASBR，如果配置为Stub区域，则会导致与外部网络无法正常通信，因此决定配置为NSSA区域。注意，在配置NSSA区域时，需要将区域内的所有路由器都配置为NSSA区域路由器，否则路由器之间无法形成邻居关系。

```
[R2]ospf 10
```

```
[R2-ospf-10]area 1
```

```
[R2-ospf-10-area-0.0.0.1]nssa
```

```
[R3]ospf 10
```

```
[R3-ospf-10]area 1
```

```
[R3-ospf-10-area-0.0.0.1]nssa
```

```
[R4]ospf 10
```

```
[R4-ospf-10]area 1
```

```
[R4-ospf-10-area-0.0.0.1]nssa
```

配置完成后，查看R4的LSDB。

```
<R4>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.4.4
```

```
Link State Database
```

Area: 0.0.0.1

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	36	36	80000005	1
.....						
Sum-Net 2	10.0.2.2	10.0.3.3	103	28	80000001	
NSSA	10.0.4.4	10.0.4.4	78	36	80000001	1
NSSA	10.0.44.44	10.0.4.4	53	36	80000001	1
NSSA	0.0.0.0	10.0.2.2	116	36	80000001	1
NSSA	0.0.0.0	10.0.3.3	103	36	80000001	1

可以看到，R4 的LSDB 中已经没有任何Type-4 LSA及Type-5 LSA，但是出现了两条LinkState ID为0.0.0.0的Type-7 LSA。R4自己引入的外部路由也生成了两条LinkState ID 分别为10.0.4.4和10.0.44.44 的Type-7 LSA，但并未生成Type-5 LSA。与原来相比，现在LSDB中LSA的数量得到了明显减少。

查看R4的路由表。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 16 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0 GigabitEthernet0/0/1	O_NSSA	150	1	D	10.0.24.2
O_NSSA	150	1	D	10.0.34.3	GigabitEthernet0/0/0
10.0.2.2/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.24.2
.....					

可以看到，R4 的路由表中出现了类型为 O_NSSA 的缺省路由，它代替了去往10.0.1.1/32和10.0.11.11/32的明细路由，且有两个下一跳，处于负载均衡状态。

使用nssa no-summary 命令还可以进一步阻止Type-3 LSA泛洪到NSSA区域1，使之成为一个Totally NSSA区域。

```
[R2]ospf 10
```

```
[R2-ospf-10]area 1
```

```
[R2-ospf-10-area-0.0.0.1]nssa no-summary
```

```
[R3]ospf 10
```

```
[R3-ospf-10]area 1
```

```
[R3-ospf-10-area-0.0.0.1]nssa no-summary
```

配置完成后，查看R4的LSDB。

```
<R4>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.4.4
```

```
Link State Database
```

Area: 0.0.0.1

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	8	36	8000000F	1
Router	10.0.4.4	10.0.4.4	2	48	80000021	1
Router	10.0.2.2	10.0.2.2	16	36	80000011	1
Network	10.0.24.4	10.0.4.4	17	32	80000002	0
Network	10.0.34.4	10.0.4.4	4	32	80000002	0
Sum-Net	0.0.0.0	10.0.2.2	29	28	80000001	1
Sum-Net	0.0.0.0	10.0.3.3	15	28	80000001	1
NSSA	10.0.4.4	10.0.4.4	544	36	80000001	1
NSSA	10.0.44.44	10.0.4.4	1149	36	80000001	1
NSSA	0.0.0.0	10.0.2.2	506	36	80000001	1
NSSA	0.0.0.0	10.0.3.3	505	36	80000001	1

可以看到，R4 的LSDB 中的Type-3 LSA也不存在了，取而代之的只是表示缺省路由的、分别由R2和R3通告的、LinkState ID 为0.0.0.0的Type-3 LSA，这进一步减小了LSDB的规模。

查看R4的路由表。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 12 Routes : 13

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.24.2
OSPF	10	2	D	10.0.34.3	GigabitEthernet0/0/0
10.0.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					

观察发现，R4 的路由表中原来的两条由Type-7 LSA生成的类型为O_NSSA 的缺省路由被两条由Type-3 LSA生成的类型为OSPF 的缺省路由代替了，这也说明了后者的路由优先级高于前者。

4．修改NSSA区域缺省路由开销值

目前，R4的路由表中拥有两条开销值均为2的、下一跳分别为R2和R3的缺省路由，所以这是一种负载均衡的状态。新的需求是：R4应优先使用经由R2的路径，同时以经由R3的路径作为备份。满足这一需求的方法是：增大R3向区域1通告的LinkState ID 为0.0.0.0 的Type-3 LSA的开销值。

```
[R3]ospf 10
```

```
[R3-ospf-10]area 1
```

```
[R3-ospf-10-area-0.0.0.1]default-cost 10
```

配置完成后，查看R4的LSDB。

```
<R4>display ospf lsdb
```


OSPF Process 10 with Router ID 10.0.4.4

Link State Database

Area: 0.0.0.1

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	173	36	80000006	1
Router	10.0.4.4	10.0.4.4	169	48	80000010	1
Router	10.0.2.2	10.0.2.2	186	36	80000006	1
Network	10.0.24.4	10.0.4.4	183	32	80000002	0
Network	10.0.34.4	10.0.4.4	169	32	80000002	0
Sum-Net	0.0.0.0	10.0.2.2	193	28	80000001	1
Sum-Net	0.0.0.0	10.0.3.3	12	28	80000002	10
NSSA	10.0.4.4	10.0.4.4	1313	36	80000001	1
NSSA	10.0.44.44	10.0.4.4	1313	36	80000001	1
NSSA	0.0.0.0	10.0.2.2	193	36	80000001	1
NSSA	0.0.0.0	10.0.3.3	180	36	80000001	1

可以看到，由R3 通告的、LinkState ID为0.0.0.0的Type-3 LSA的开销值变为了10，R3 通告的、LinkState ID 为0.0.0.0 的Type-7 LSA的开销值未发生改变。

查看R4的路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 12 Routes : 12

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.24.2
10.0.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					

可以看到，路由表中现在只有一条下一跳指向了 R2 的缺省路由了，原来的下一跳指向R3的缺省路由已经消失。

思考

在OSPF网络中，对于非骨干区域，如果能够将之合理地配置成Stub区域或NSSA区域，则可以缩减区域中路由器的路由表的规模。那么如何才能缩减骨干路由器的路由表规模呢？

2.6 OSPF 虚链路

原理概述

通常情况下，一个 OSPF网络的每个非骨干区域都必须与骨干区域通过 ABR路由器直接连接，非骨干区域之间的通信都需要通过骨干区域进行中转。但在现实中，可能会因为各种条件限制，导致非骨干区域与骨干区域无法直接连接，在这种情况下，可以使用 OSPF 虚链路（Virtual Link）来实现非骨干区域与骨干区域在逻辑上直接相连。

OSPF 协议还要求骨干区域必须是唯一且连续的，然而，由于发生故障等原因，骨干区域有可能出现被分割的情况。此时，同样可以使用虚链路来实现物理上被分割的骨干区域能够逻辑相连。

虚链路在网络中会穿越其他区域，因此可能会带来安全隐患，所以通常都会对虚链路进行认证功能的配置。虚链路认证其实是 OSPF 接口认证的一种，支持 MD5、HMAC-MD5、明文及Keychain等特性。

实验目的

- 理解OSPF 虚链路的应用场景
- 掌握OSPF 虚链路的配置方法
- 掌握OSPF 虚链路认证功能的配置方法

实验内容

实验拓扑如图2-11所示，实验编址如表2-6所示。本实验模拟了一个企业网络场景，全网运行OSPF，路由器R1、R2为公司总部路由器，R3为新建分公司的接入路由器，R4为分公司下面的分支机构的接入路由器。由于网络升级尚未完成，所以目前的区域划分是：R1与R2之间的链路位于区域0，R3与R1、R3与R2之间的链路位于区域1，R3与R4之间的链路位于区域2。网络需求是：使用虚链路技术，使得分支机构所属的区域2能够访问总部网络，且优先使用路径R4-R3-R1，并以R4-R3-R2路径作为备份。同时，总部路由器R1和R2之间的通信需要采用R1-R3-R2路径作为冗余备份。另外，为了提高安全性，对于所使用的虚链路应进行认证功能的配置。

实验拓扑

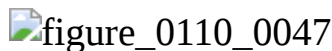
figure_0110_0047

图2-11 OSPF虚链路

实验编址表

表2-6 实验编址

figure_0110_0048

实验步骤

1．基本配置

根据图2-11和表2-6进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 10.0.13.3
```

```
PING 10.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=40 ms
```

```
--- 10.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 40/40/40 ms
```

其余直连网段的连通性测试过程在此省略。

2．搭建OSPF网络

在每台路由器上配置OSPF协议，其中R1与R2之间的链路位于区域0，R3与R1、R3与R2之间的链路位于区域1，R3与R4之间的链路位于区域2。

```
[R1]ospf 10 router-id 10.0.1.1
```

```
[R1-ospf-10]area 0
```

[R1-ospf-10-area-0.0.0.0]network 10.0.12.0 0.0.0.255

[R1-ospf-10-area-0.0.0.0]network 10.0.1.1 0.0.0.0

[R1-ospf-10-area-0.0.0.0]area 1

[R1-ospf-10-area-0.0.0.1]network 10.0.13.0 0.0.0.255

[R2]router 10 router-id 10.0.2.2

[R2-ospf-10]area 0

[R2-ospf-10-area-0.0.0.0]network 10.0.12.0 0.0.0.255

[R2-ospf-10-area-0.0.0.0]network 10.0.2.2 0.0.0.0

[R2-ospf-10-area-0.0.0.0]area 1

[R2-ospf-10-area-0.0.0.1]network 10.0.23.0 0.0.0.255

[R3]ospf 10 router-id 10.0.3.3

[R3-ospf-10]area 1

[R3-ospf-10-area-0.0.0.1]network 10.0.13.0 0.0.0.255

[R3-ospf-10-area-0.0.0.1]network 10.0.23.0 0.0.0.255

[R3-ospf-10-area-0.0.0.1]network 10.0.3.3 0.0.0.0

[R3-ospf-10-area-0.0.0.1]area 2

[R3-ospf-10-area-0.0.0.2]network 10.0.34.0 0.0.0.255

[R4]ospf 10 router-id 10.0.4.4

[R4-ospf-10]area 2

[R4-ospf-10-area-0.0.0.2]network 10.0.34.0 0.0.0.255

```
[R4-ospf-10-area-0.0.0.2]network 10.0.4.4 0.0.0.0
```

配置完成后，查看R3的OSPF邻居关系。

```
<R3>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.3.3
```

```
Peer Statistic Information
```


Area Id	Interface	Neighbor id	State
0.0.0.1	GigabitEthernet0/0/1	10.0.1.1	Full
0.0.0.1	GigabitEthernet0/0/2	10.0.2.2	Full
0.0.0.2	GigabitEthernet0/0/0	10.0.4.4	Full

可以看到，R3 的邻居关系都处于 Full 状态，表明各路由器之间已经成功建立了邻居关系。

查看R4的LSDB。

```
<R4>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.4.4
```

```
Link State Database
```

```
Area: 0.0.0.2
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence
Metric					

Router 0	10.0.4.4	10.0.4.4	732	48	80000004
Router 1	10.0.3.3	10.0.3.3	746	36	80000005
Network 80000002	10.0.34.4 0	10.0.4.4	746	32	

可以看到，R4的LSDB中没有区域0中关于10.0.1.1/32和10.0.2.2/32的LSA，也没有任何其他区域的LSA，仅仅只有本区域的Type-1 LSA和Type-2 LSA，这说明区域2中并没有ABR存在，即区域2并未与区域0相连，也无法与其他区域进行正常通信。

3．使用虚链路使区域2与区域0逻辑相连

接下来将使用虚链路使区域2与区域0在逻辑上相互连接起来，此时的区域1将作为区域2与区域0之间的传输区域。虚链路配置操作将在连接区域2与区域1的R3上，以及连接区域1与区域0的ABR路由器R1上进行。

在R3的区域1视图下，使用vlink-peer命令建立与R1的虚链路。

```
[R3]ospf 10
```

```
[R3-ospf-10]area 1
```

```
[R3-ospf-10-area-0.0.0.1]vlink-peer 10.0.1.1
```

同样，在R1的区域1视图下，使用vlink-peer命令建立与R3的虚链路。

```
[R1]ospf 10
```

```
[R1-ospf-10]area 1
```

```
[R1-ospf-10-area-0.0.0.1]vlink-peer 10.0.3.3
```

配置完成后，在R1上使用命令display ospf vlink查看虚链路信息。

```
[R1]display ospf vlink
```

OSPF Process 10 with Router ID 10.0.1.1

Virtual Links

Virtual-link Neighbor-id -> 10.0.3.3, Neighbor-State: Full

Interface: 10.0.13.1 (GigabitEthernet0/0/1)

Cost: 1 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

可以看到，R1与R3已经成功建立了虚链路，虚链路的状态为Full。

查看R4的LSDB。

```
<R4>display ospf lsdb
```

OSPF Process 10 with Router ID 10.0.4.4

Link State Database

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 80000004	10.0.4.4 0	10.0.4.4	1755	48	
Router 1	10.0.3.3	10.0.3.3	162	36	80000006

Network 80000002	10.0.34.4 0	10.0.4.4	1769	32
Sum-Net 80000001	10.0.13.0 1	10.0.3.3	162	28
Sum-Net 80000001	10.0.12.0 2	10.0.3.3	162	28
Sum-Net 80000001	10.0.2.2 1	10.0.3.3	162	28
Sum-Net 80000001	10.0.1.1 1	10.0.3.3	162	28
Sum-Net 80000001	10.0.23.0 1	10.0.3.3	162	28

可以看到，R4 的LSDB 中出现了由R3通告的、关于区域0 和区域1 的 Type-3 LSA，说明此时R4已经将R3作为连接区域2至区域0的ABR了。

测试R4与R1和R2的连通性。

```
<R4>ping -a 10.0.4.4 -c 1 10.0.1.1
```

```
PING 10.0.1.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.1.1: bytes=56 Sequence=1 ttl=254 time=50 ms
```

```
--- 10.0.1.1 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 50/50/50 ms
```

```
<R4>ping -a 10.0.4.4 -c 1 10.0.2.2
```

```
PING 10.0.2.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.2.2: bytes=56 Sequence=1 ttl=253 time=60 ms
```

```
--- 10.0.2.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 60/60/60 ms
```

可以看到，通信是正常的。

4．修改虚链路的开销值

通过R1与R3之间的虚链路，实现了区域2与区域0的逻辑相连。然而，区域1与区域0之间的ABR除了R1之外，还有R2。同样，也可以在R2与R3之间建立一条虚链路。

```
[R2]ospf 10
```

```
[R2-ospf-10]area 1
```

```
[R2-ospf-10-area-0.0.0.1]vlink-peer 10.0.3.3
```

```
[R3]ospf 10
```

```
[R3-ospf-10]area 1
```

```
[R3-ospf-10-area-0.0.0.1]vlink-peer 10.0.2.2
```

配置完成后，在R3上查看虚链路信息。

```
[R3]display ospf vlink
```

OSPF Process 10 with Router ID 10.0.3.3

Virtual Links

Virtual-link Neighbor-id -> 10.0.1.1, Neighbor-State: Full

Interface: 10.0.13.3 (GigabitEthernet0/0/1)

Cost: 1 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

Virtual Links

Virtual-link Neighbor-id -> 10.0.2.2, Neighbor-State: Full

Interface: 10.0.23.3 (GigabitEthernet0/0/2)

Cost: 1 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

可以看到，现在在R3与R1之间、R3与R2之间各存在一条虚链路，开销值均为1，那么当R4访问总部网络区域0时，就会出现负载均衡的情形。新的需求是：R4与区域0通信时优先选用经由R1的路径，并以经由R2的路径作为备份，实现方法是修改虚链路的开销值。

由于虚链路实际使用的路径是在传输区域内经过SPF（Short Path First）算法计算出的最优路径，虚链路的开销值其实就是OSPF协议在传输区域内所选用的物理路径的开销值，所以修改虚链路的开销值其实就是修改物理路径的OSPF开销值。

在R3 的GE 0/0/2 接口上修改OSPF 协议开销值。

```
[R3]interface GigabitEthernet 0/0/2
```

```
[R3-GigabitEthernet0/0/2]ospf cost 10
```

在R2 的GE 0/0/2 接口上完成同样的配置。

```
[R2]interface GigabitEthernet 0/0/2
```

```
[R2-GigabitEthernet0/0/2]ospf cost 10
```

配置完成后，在R3上查看虚链路信息。

```
[R3]display ospf vlink
```

```
OSPF Process 10 with Router ID 10.0.3.3
```

```
Virtual Links
```

```
Virtual-link Neighbor-id -> 10.0.1.1, Neighbor-State: Full
```

```
Interface: 10.0.13.3 (GigabitEthernet0/0/1)
```

```
Cost: 1 State: P-2-P Type: Virtual
```

```
Transit Area: 0.0.0.1
```

```
Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1
```

```
GR State: Normal
```

```
Virtual Links
```

```
Virtual-link Neighbor-id -> 10.0.2.2, Neighbor-State: Full
```

```
Interface: 10.0.23.3 (GigabitEthernet0/0/2)
```

```
Cost: 10 State: P-2-P Type: Virtual
```

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

可以看到，R3与R2之间的虚链路的开销值变为了10，R3与R1之间的虚链路的开销值保持为1。在这样的条件下，R4或R3都将通过经由R1的路径访问区域0，并以经由R2的路径作为备份。

5．使用虚链路作为区域0链路的冗余备份

目前，R1与R2之间只有单条链路连接，如果出现链路故障，就会导致区域0被分割的问题。为了解决这一问题，增强网络的可靠性，可以以区域1为传输区域，在R1与R2之间建立一条虚链路作为冗余备份。

```
[R1]ospf 10
```

```
[R1-ospf-10]area 1
```

```
[R1-ospf-10-area-0.0.0.1]vlink-peer 10.0.2.2
```

```
[R2]ospf 10
```

```
[R2-ospf-10]area 1
```

```
[R2-ospf-10-area-0.0.0.1]vlink-peer 10.0.1.1
```

配置完成后，在R1上查看虚链路信息。

```
[R1]display ospf vlink
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Virtual Links
```

```
Virtual-link Neighbor-id -> 10.2.2.2, Neighbor-State: Full
```

```
Interface: 10.0.13.1 (GigabitEthernet0/0/1)
```

Cost: 11 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

Virtual Links

Virtual-link Neighbor-id -> 10.0.3.3, Neighbor-State: Full

Interface: 10.0.13.1 (GigabitEthernet0/0/1)

Cost: 1 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

可以看到，R1与R2之间的虚链路的开销值为11。在R1上使用tracert命令测试访问10.0.2.2/32的路径。

```
<R1>tracert 10.0.2.2
```

```
traceroute to 10.0.2.2(10.0.2.2), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.12.2 30 ms 1 ms 10 ms
```

可以发现，此时R1与R2 之间的通信使用的仍是直连链路。关闭R1的GE 0/0/0 接口，模拟链路出现故障。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]shutdown
```

然后，再次用tracert命令进行测试。

```
<R1>tracert 10.0.2.2
```

```
traceroute to 10.0.2.2(10.0.2.2), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.13.3 20 ms 10 ms 10 ms
```

```
2 10.0.23.2 30 ms 20 ms 20 ms
```

可以看到，现在R1与R2采用了虚链路进行通信。

为进行后续实验，请重新打开R1 的GE 0/0/0 接口。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]undo shutdown
```

6．配置虚链路的认证功能

由于虚链路使用了其他传输区域的物理链路，所以通常应配置认证功能来增强安全性。以R1与R2之间的虚链路为例，在R1上区域1的视图下，使用命令vlink-peer 10.0.2.2 hmac-md5 1 plain huawei，其中hmac-md5 表示所选用的认证加密方式，1为key ID，plain huawei 表示以明文方式显示口令，口令为huawei。

```
[R1]ospf 10
```

```
[R1-ospf-10]area 1
```

```
[R1-ospf-10-area-0.0.0.1]vlink-peer 10.0.2.2 hmac-md5 1 plain huawei
```

配置完成后，在R1上观察虚链路信息。

```
<R1>display ospf vlink
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

Virtual Links

Virtual-link Neighbor-id -> 10.0.2.2, Neighbor-State: Down

Interface: 10.0.13.1 (GigabitEthernet0/0/1)

Cost: 11 State: P-2-P Type: Virtual

.....

可以看到，目前R1与R2之间的虚链路的状态为Down，说明虚链路建立失败，原因是R2还未进行相应的认证功能配置。

在R2上配置认证功能。

```
[R2]ospf 10
```

```
[R2-ospf-10]area 1
```

```
[R2-ospf-10-area-0.0.0.1]vlink-peer 10.0.1.1 hmac-md5 1 plain huawei
```

配置完成后，重新在R1上观察虚链路信息。

```
<R1>display ospf vlink
```

OSPF Process 10 with Router ID 1.1.1.1

Virtual Links

Virtual-link Neighbor-id -> 10.0.2.2, Neighbor-State: Full

Interface: 10.0.13.1 (GigabitEthernet0/0/1)

Cost: 11 State: P-2-P Type: Virtual

.....

可以看到，R1与R2之间的虚链路已经得到恢复。最后，请读者自行完成对其他虚链路的认证功能配置。

思考

本实验中，如何利用GRE Tunnel 实现区域2 与区域0 在逻辑上直接相连？

2.7 OSPF 网络类型

原理概述

OSPF协议定义了4种不同的网络类型，分别为广播网络（也称为Broadcast网络）、NBMA（Non-Broadcast Multi-Access）网络、点到点网络（也称为Point-to-Point网络，或P2P网络）和点到多点网络（也称为Point-to-Multipoint网络，或P2MP网络）。不同类型的网络上OSPF协议的工作机制会存在一些差别。例如，前两种类型的网络都要进行DR和BDR的选举，而后两种类型的网络不进行DR和BDR的选举，也不存在DR和BDR。读者可自行去全面地了解 and 比较这4种网络在运行OSPF协议时的异同点，这里不再赘述。

默认情况下，OSPF 协议会根据接口的链路层封装协议去自动设定接口的网络类型，以太网接口的默认网络类型为Broadcast，串行接口如果链路层封装协议是PPP协议（Point-to-Point Protocol）或HDLC（High-level Data Link Control Protocol）协议，则默认网络类型为点到点类型，ATM或帧中继（Frame-Relay）接口的默认网络类型为NBMA。需要强调的是，接口的网络类型是可以根据需要而进行人为修改的，这一特点使得OSPF协议具备了更多的灵活性和更强的适应能力，可以满足复杂网络中的各种不同需求。

实验目的

- 理解OSPF 不同网络类型的主要差别
- 掌握OSPF 不同网络类型的配置方法

实验内容

实验拓扑如图2-12所示，实验编址如表2-7所示。本实验模拟了一个简单的企业网络场景，R1 为公司总部的路由器，R2 为分支机构的路由器，R1 与 R2 通过帧中继交换机FRSW连接。R1到R2使用的PVC（Permanent Virtual Circuit）的DLCI（Data Link Connection Identifier）是 102，R2 到 R1 使用的 PVC 的 DLCI 是 201。R1 和 R2 运行OSPF协议，使用的OSPF网络类型为P2P。然后，公司新增了一个分支机构，R3为新分支机构的路由器，并与FRSW相连。R1到R3使用的PVC的DLCI是103，R3到R1使用的PVC的DLCI是301。原来运行的OSPF协议需要扩展到R3上，以实现公司全网互通。

实验拓扑

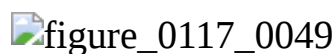
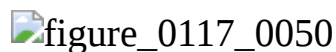


图2-12 OSPF网络类型

实验编址表

表2-7 实验编址



实验步骤

1．基本配置

首先，根据图2-12和表2-7进行一些基本的配置。如图2-13所示，在文本框内输入相应的FRSW 接口编号与DLCI，Interface 0 对应于Serial 0/0/0，Interface 1 对应于Serial 0/0/1，Interface 2 对应于Serial 0/0/2，配置完成之后点击OK。

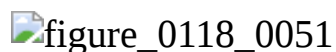


图2-13 配置帧中继交换机

接下来进行路由器的配置。注意，为了在帧中继网络中支持组播及广播报文，必须在配置帧中继映射时添加关键字Broadcast。如果不添加关键字Broadcast，则帧中继接口无法封装组播及广播的报文。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]link-protocol fr
```

```
[R1-Serial1/0/0]ip address 10.0.123.1 255.255.255.0
```

```
[R1-Serial1/0/0]fr interface-type dte
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.2 102 broadcast
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.3 103 broadcast
```

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]link-protocol fr
```

```
[R2-Serial1/0/0]ip address 10.0.123.2 255.255.255.0
```

```
[R2-Serial1/0/0]fr interface-type dte
```

```
[R2-Serial1/0/0]fr map ip 10.0.123.1 201 broadcast
```

配置完成之后，在R1上使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.123.2
```

```
PING 10.0.123.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.123.2: bytes=56 Sequence=1 ttl=255 time=20 ms
```

```
--- 10.0.123.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

0.00% packet loss

round-trip min/avg/max = 20/20/20 ms

2 . 配置OSPF点到点网络类型

在R1和R2上配置OSPF协议。

```
[R1]ospf 10
```

```
[R1-ospf-10]area 0
```

```
[R1-ospf-10-area-0.0.0.0]network 10.0.123.0 0.0.0.255
```

```
[R1-ospf-10-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R2]ospf 10
```

```
[R2-ospf-10]area 0
```

```
[R2-ospf-10-area-0.0.0.0]network 10.0.123.0 0.0.0.255
```

```
[R2-ospf-10-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

配置完成后，查看R1 的串行接口Serial 1/0/0 在OSPF 协议中的默认网络类型。

```
[R1]display ospf interface Serial 1/0/0
```

OSPF Process 10 with Router ID 10.0.1.1

Interfaces

Interface: 10.0.123.1 (Serial1/0/0)

Cost: 48 State:Waiting Type:NBMA MTU: 1500

Priority: 1

Designated Router: 10.0.123.1

Backup Designated Router: 0.0.0.0

Timers: Hello 30 , Dead 120 , Poll 120 , Retransmit 5 , Transmit Delay 1

可以看到，帧中继接口在 OSPF 协议中的默认网络类型为 NBMA，并且需要选择DR 和BDR，默认的Hello报文间隔为30s，Dead Timer的时间是Hello 报文间隔的4 倍。

由于企业总部目前只需与一个分支机构通信，所以决定在R1和R2 上将 Serial 1/0/0接口的网络类型修改为点到点类型。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]ospf network-type p2p
```

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]ospf network-type p2p
```

配置完成后，在R1上查看OSPF邻居状态和路由表。

```
<R1>display ospf peer brief
```

OSPF Process 10 with Router ID 10.0.1.1

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	Serial1/0/0	10.0.2.2	Full


```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations :11 Routes :11

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.2.2/32	OSPF	10	48	D	10.0.123.2 Serial1/0/0
10.0.123.0/24	Direct	0	0	D	10.0.123.1 Serial1/0/0
.....					

可以看到，R1与R2已经自动建立了OSPF邻接关系，并且R1的路由表中也拥有了去往R2的Loopback接口所在网段的路由信息。

在R1上使用命令display ospf interface Serial 1/0/0查看Serial 1/0/0接口上的OSPF信息。

<R1>display ospf interface Serial 1/0/0

OSPF Process 10 with Router ID 10.0.1.1

Interfaces

Interface: 10.0.123.1 (Serial1/0/0) --> 10.0.123.2

Cost: 48 State:P-2-P Type:P2P MTU: 1500

Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1

可以看到，R1 的 Serial 1/0/0 接口的网络类型现在已经变成修改后的点到点类型，Hello 报文间隔为10s，Dead Timer的时间也是Hello 报文间隔的4 倍。

在R1 上使用命令debugging ospf packet hello对OSPF Hello 报文的情况进行调试。

```
<R1>debugging ospf packet hello
```

```
Sep 19 2013 07:41:15.553.2-05:13 R1 RM/6/RMDEBUG: Source Address: 10.0.123.1
```

```
<R1>Sep 19 2013 07:41:15.553.3-05:13 R1 RM/6/RMDEBUG: Destination Address: 224.0.0.5
```

```
<R1>Sep 19 2013 07:41:15.553.4-05:13 R1 RM/6/RMDEBUG: Ver# 2, Type: 1 (Hello)
```

```
.....
```

```
<R1>Sep 19 2013 07:41:15.553.12-05:13 R1 RM/6/RMDEBUG: DR: 0.0.0.0
```

```
<R1>Sep 19 2013 07:41:15.553.13-05:13 R1 RM/6/RMDEBUG: BDR: 0.0.0.0
```

```
<R1>Sep 19 2013 07:41:15.553.14-05:13 R1 RM/6/RMDEBUG: # Attached Neighbors: 1
```

```
<R1>Sep 19 2013 07:41:15.553.15-05:13 R1 RM/6/RMDEBUG: Neighbor: 10.0.2.2
```

可以看到，Hello报文的目的是组播地址224.0.0.5，并且无需进行DR和BDR的选举。至此，R1和R2已成为了一个典型的OSPF点到点网络，路由器通过组播Hello报文自动发现邻居并建立邻接关系，且不需要DR和BDR。

现在，公司有了一个新增的分支机构，增加的路由器是R3。为了使R3也能够加入原来的OSPF网络，可在R1与R3之间再建立一条帧中继PVC，并将其OSPF网络类型配置为点到点类型。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.3 103 broadcast
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]link-protocol fr
```

```
[R3-Serial1/0/0]ip address 10.0.123.3 255.255.255.0
```

```
[R3-Serial1/0/0]fr interface-type dte
```

```
[R3-Serial1/0/0]fr map ip 10.0.123.1 301 broadcast
```

```
[R3-Serial1/0/0]ospf network-type p2p
```

```
[R3-Serial1/0/0]ospf 10
```

```
[R3-ospf-10]area 0
```

```
[R3-ospf-10-area-0.0.0.0]network 10.0.123.0 0.0.0.255
```

```
[R3-ospf-10-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

配置完成后，在R1上查看OSPF邻居信息。

```
<R1>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Peer Statistic Information
```

```
-----  
-----
```


Area Id	Interface	Neighbor id	State
0.0.0.0	Serial1/0/0	10.0.2.2	Full

观察发现，虽然R1与R2建立起了邻接关系，但R1与R3却不能建立邻接关系。原来，点到点网络的两端只能允许各有一个专门的接口；现在R1、R2、R3的Serial 1/0/0接口都配置成了点到点网络类型，这就意味着R1的Serial 1/0/0接口既要对应R1到R2这个点到点网络，又要对应R1到R3这个点到点网络，而这种情况是无法实现的。解决这一问题的方法之一是在R1上增加一个物理接口，其中一个接口对应R1到R2这个点到点网络，另一个接口对应R1到R3这个点到点网络；还有一个方法就是在R1的Serial 1/0/0接口下配置两个子接口，两个子接口使用不同网段的IP地址，并分别用来与R2和R3建立点到点网络。第一种方法显然会明显增加设备的成本，第二种方法会涉及到诸如重新编址等问题。其实，除了这些方法之外，采用NBMA网络类型应该更能有效地解决问题。

3．配置OSPF的NBMA及Broadcast网络类型

配置R1、R2、R3的Serial 1/0/0接口为NBMA类型。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]ospf network-type nbma
```

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]ospf network-type nbma
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]ospf network-type nbma
```

NBMA类型是帧中继串行接口运行OSPF时的默认网络类型。NB表示非广播，其含义是指NBMA接口不支持广播或组播报文；MA表示多路访问，在多路访问的网络中，OSPF是需要进行DR和BDR的选举的。

为了验证NBMA 网络的特点，可在R1，R2，R3 的Serial 1/0/0 接口配置帧中继映射时不添加关键字Broadcast，这样一来，即使OSPF希望通过组播形式发送Hello报文，链路层也无法对组播Hello报文进行封装，从而导致无法建立邻接关系。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.2 102
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.3 103
```

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]fr map ip 10.0.123.1 201
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]fr map ip 10.0.123.1 301
```

配置完成后，在R1上查看OSPF邻居信息。

```
[R1]display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Peer Statistic Information
```


Area Id	Interface	Neighbor id	State

可以看到，R1与R2、R1与R3都无法建立邻接关系，原因是此时每个接口都无法发送组播 OSPF Hello 报文。NBMA 网络类型不支持通过组播方式自动发现邻居，而需要通过手动配置来指定邻居，并通过单播 OSPF Hello 报文来建立邻接关系。

在R1、R2、R3上使用peer命令指定OSPF邻居。

```
[R1]ospf 10
```

```
[R1-ospf-10]peer 10.0.123.2
```

```
[R1-ospf-10]peer 10.0.123.3
```

```
[R2]ospf 10
```

```
[R2-ospf-10]peer 10.0.123.1
```

```
[R3]ospf 10
```

```
[R3-ospf-10]peer 10.0.123.1
```

配置完成后，查看R1的OSPF邻居信息。

```
<R1>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Peer Statistic Information
```

```
-----  
-----
```

Area Id	Interface	Neighbor id	State
0.0.0.0	Serial1/0/0	10.0.2.2	Full
0.0.0.0	Serial1/0/0	10.0.3.3	Full

-----可以看到，R1与R2、R1与R3现在已经成功建立起了邻接关系。

观察R1、R2、R3的路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations :11 Routes :11

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.3.3/32	OSPF	10	48	D	10.0.123.3 Serial1/0/0
10.0.123.0/24	Direct	0	0	D	10.0.123.1 Serial1/0/0
.....					

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 9 Routes : 9

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.2.2/32	Direct	0 0	D	127.0.0.1	LoopBack0
10.0.123.0/24	Direct	0 0	D	10.0.123.2	Serial1/0/0
10.0.123.1/32	Direct	0 0	D	10.0.123.1	Serial1/0/0
10.0.123.2/32	Direct	0 0	D	127.0.0.1	Serial1/0/0
10.0.123.255/32 Serial1/0/0	Direct	0 0	D	127.0.0.1	
127.0.0.0/8 InLoopBack0	Direct	0 0	D	127.0.0.1	
127.0.0.1/32 InLoopBack0	Direct	0 0	D	127.0.0.1	
127.255.255.255/32 InLoopBack0	Direct	0 0	D	127.0.0.1	
255.255.255.255/32 InLoopBack0	Direct	0 0	D	127.0.0.1	

<R3>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 10 Routes : 10

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

10.0.1.1/32	OSPF	10	48	D	10.0.123.1	Serial1/0/0
10.0.3.3/32	Direct	0	0	D	127.0.0.1	LoopBack0
.....						

观察发现，R1 和R3 相互都接收到了对方Loopback 0 的路由，但是R1 的路由表中没有去往R2的Loopback 0 的路由，R2 的路由表中没有去往R1和R3的Loopback 0 的路由，R3 的路由表中没有去往R2 的Loopback 0 的路由。

查看R1的OSPF邻居的详细信息。

```
[R1]display ospf peer
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.123.1(Serial1/0/0)'s neighbors
```

```
Router ID: 10.0.2.2      Address: 10.0.123.2
```

```
State: Full Mode:Nbr is Master Priority: 1
```

```
DR: 10.0.123.2 BDR: 10.0.123.1 MTU: 0
```

```
Dead timer due in 114 sec
```

```
Retrans timer interval: 0
```

```
Neighbor is up for 00:07:06
```

```
Authentication Sequence: [ 0 ]
```

```
Router ID: 10.0.3.3      Address: 10.0.123.3
```

```
State: Full Mode:Nbr is Master Priority: 1
```

DR: 10.0.123.3 BDR: 10.0.123.1 MTU: 0

Dead timer due in 98 sec

Retrans timer interval: 5

Neighbor is up for 00:06:56

Authentication Sequence: [0]

可以看到，R1 在Serial 1/0/0 接口上存在两个邻居，但是同时也有两个不同的DR。在多路访问的网络中，DR只能有一个，这说明网络存在故障。

目前，R1、R2、R3 的 Serial 1/0/0 接口都工作在 NBMA 模式下，需要选举 DR 和BDR，而R1、R2、R3的Serial 1/0/0 接口的DR 优先级的值都是1，因此Router-ID 最大的路由器将被选举为DR。但是，由于R2与R3之间缺少了PVC，导致R2和R3都认为各自的OSPF网络中只存在邻居R1，所以R2和R3都认为自己为DR，而R1为BDR。

在R1上使用命令display ospf interface Serial 1/0/0查看Serial 1/0/0接口的OSPF详细信息。

<R1>display ospf interface Serial 1/0/0

OSPF Process 10 with Router ID 10.0.1.1

Interfaces

Interface: 10.0.123.1 (Serial1/0/0)

Cost: 48 State:BDR Type:NBMA MTU: 1500

Priority: 1

Designated Router: 10.0.123.3

Backup Designated Router: 10.0.123.1

Timers: Hello 30 , Dead 120 , Poll 120 , Retransmit 5 , Transmit Delay 1

可以看到，R1认为R3是OSPF网络的DR，自己是BDR。尽管在R1的邻居表中显示R2也是DR，但是在R1的Serial 1/0/0 接口的详细信息中显示的DR却是R3，导致的结果是，R1和R3之间路由信息可以正常传递，但R1与R2之间的路由信息传递却出现了问题。

要解决这个问题，就必须确保 DR 有且只有一个。对于目前这个具有 Hub-Spoke结构的网络，应该保证Hub端R1成为DR路由器，Spoke端R2和R3成为DRother路由器。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]ospf dr-priority 10
```

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]ospf dr-priority 0
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]ospf dr-priority 0
```

配置完成后，在R1 上使用命令display ospf interface Serial 1/0/0 查看Serial 1/0/0接口的OSPF详细情况。

```
<R1>display ospf interface Serial 1/0/0
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Interfaces
```

```
Interface: 10.0.123.1 (Serial1/0/0)
```

```
Cost: 48   State:DR      Type:NBMA   MTU: 1500
```

```
Priority: 10
```


Designated Router: 10.0.123.1

Backup Designated Router: 0.0.0.0

Timers: Hello 30 , Dead 120 , Poll 120 , Retransmit 5 , Transmit Delay 1

可以看到 , 现在R1已经成为了DR。

查看R1、 R2、 R3的路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 12 Routes : 12

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.2.2/32	OSPF	10	48	D	10.0.123.2 Serial1/0/0
10.0.3.3/32	OSPF	10	48	D	10.0.123.3 Serial1/0/0
10.0.123.0/24	Direct	0	0	D	10.0.123.1 Serial1/0/0

.....

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations :11 Routes :11

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	OSPF	10	48	D	10.0.123.1 Serial1/0/0
10.0.2.2/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.3.3/32	OSPF	10	48	D	10.0.123.3 Serial1/0/0
10.0.123.0/24	Direct	0	0	D	10.0.123.2 Serial1/0/0

.....

<R3>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations :11 Routes :11

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	OSPF	10	48	D	10.0.123.1 Serial1/0/0
10.0.2.2/32	OSPF	10	48	D	10.0.123.2 Serial1/0/0
10.0.3.3/32	Direct	0	0	D	127.0.0.1 LoopBack0

.....

可以看到，现在每台路由器都获得了其他路由器的 Loopback 0 的路由，R2 去往10.0.3.3的下一跳是R3的10.0.123.3，R3去往R2的下一跳是R2的10.0.123.2。

测试R2 和R3 的Loopback 0 接口之间的连通性。

```
<R3>ping -a 10.0.3.3 10.0.2.2
```

```
PING 10.0.2.2: 56 data bytes, press CTRL_C to break
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
--- 10.0.2.2 ping statistics ---
```

```
5 packet(s) transmitted
```

```
0 packet(s) received
```

```
100.00% packet loss
```

可以看到，R2与R3的连通性存在问题。虽然路由表中彼此有了去往对方的路由，但网络仍然无法进行正常通信。

在R2 上测试去往R3 的Loopback 0 接口的下一跳10.0.123.3 是否可达。

```
<R2>ping 10.0.123.3
```

```
PING 10.0.123.3: 56 data bytes, press CTRL_C to break
```

```
Request time out
```

Request time out

Request time out

Request time out

Request time out

--- 10.0.123.3 ping statistics ---

5 packet(s) transmitted

0 packet(s) received

100.00% packet loss

可以看到，10.0.123.3对于R2是不可达的。

在R2上查看是否存在去往10.0.123.3的映射，在R3上查看是否存在去往10.0.123.2的映射。

<R2>display fr map-info

Map Statistics for interface Serial1/0/0 (DTE)

DLCI = 201, IP 10.0.123.1, Serial1/0/0

create time = 2013/09/19 07:39:13, status = ACTIVE

encapsulation = ietf, vlink = 3

<R3>display fr map-info

Map Statistics for interface Serial1/0/0 (DTE)

DLCI = 301, IP 10.0.123.1, Serial1/0/0

create time = 2013/09/19 07:39:19, status = ACTIVE

encapsulation = ietf, vlink = 3

可以看到，R2 的帧中继映射表中缺少了去往 10.0.123.3 的映射，R3 的帧中继映射表中缺少了去往10.0.123.2的映射。

为了解决R2和R3之间的互通性问题，可以在R2与R3之间增加一条PVC，使得网络成为全互联的状态。增加一条PVC不仅可以解决R2与R3之间的通信问题，而且还可以避免之前遇到的DR选举问题。需要说明的是，对于小型网络而言，增加PVC的数量使网络保持全互联状态的确是可行的做法，但是对于规模不断扩大的网络而言，这种方法所需增加的PVC数量会急剧增长，各种成本和工作量也会随之猛增，最终会变得理论上可行而实际上无法接受。

还有另外一种方法可以用来解决R2和R3之间的互通性问题，就是不增加新的PVC，而是在R2和R3上分别配置去往10.0.123.3和10.0.123.2的帧中继映射条目，即R2利用DLCI 201 这条PVC 经R1 中转去往10.0.123.3，R3 利用DLCI 301 这条PVC 经R1 中转去往10.0.123.2。

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]fr map ip 10.0.123.3 201
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]fr map ip 10.0.123.2 301
```

配置完成后，在R2、R3上查看映射信息。

```
[R2]display fr map-info
```

```
Map Statistics for interface Serial1/0/0 (DTE)
```

```
DLCI = 201, IP 10.0.123.1, Serial1/0/0
```

```
create time = 2013/09/19 07:39:13, status = ACTIVE
```

```
encapsulation = ietf, vlink = 3
```

```
DLCI = 201, IP 10.0.123.3, Serial1/0/0
```

create time = 2013/09/19 13:31:23, status = ACTIVE

encapsulation = ietf, vlink = 4

[R3]display fr map-info

Map Statistics for interface Serial1/0/0 (DTE)

DLCI = 301, IP 10.0.123.1, Serial1/0/0

create time = 2013/09/19 07:39:19, status = ACTIVE

encapsulation = ietf, vlink = 3

DLCI = 301, IP 10.0.123.2, Serial1/0/0

create time = 2013/09/19 13:31:40, status = ACTIVE

encapsulation = ietf, vlink = 4

可以看到，R2有了去往10.0.123.3的映射，R3有了去往10.0.123.2的映射。

测试R2 和R3 的Loopback 0 接口之间的连通性。

<R3>ping -a 10.0.3.3 10.0.2.2

PING 10.0.2.2: 56 data bytes, press CTRL_C to break

Reply from 10.0.2.2: bytes=56 Sequence=1 ttl=254 time=20 ms

Reply from 10.0.2.2: bytes=56 Sequence=2 ttl=254 time=20 ms

Reply from 10.0.2.2: bytes=56 Sequence=3 ttl=254 time=10 ms

Reply from 10.0.2.2: bytes=56 Sequence=4 ttl=254 time=30 ms

Reply from 10.0.2.2: bytes=56 Sequence=5 ttl=254 time=20 ms

--- 10.0.2.2 ping statistics ---

5 packet(s) transmitted

5 packet(s) received

0.00% packet loss

round-trip min/avg/max = 10/20/30 ms

可以看到，现在R2与R3之间实现了互通，R2和R3的Loopback 0接口之间可以正常通信了。至此，R1、R2、R3构成了一个典型的OSPF NBMA网络，它不支持通过组播OSPF Hello报文自动发现邻居并建立邻接关系，而需要通过手工指定邻居，同时网络还需要进行DR和BDR的选举。

我们还可以将R1、R2、R3组成的网络修改为OSPF Broadcast类型的网络，但是这有个前提，就是网络中的PVC必须支持广播。不过，先尝试一下在配置帧中继映射时不添加关键字Broadcast，看看情况会怎么样。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]ospf network-type broadcast
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.2 102
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.3 103
```

```
[R1-Serial1/0/0]ospf 10
```

```
[R1-ospf-10]undo peer 10.0.123.2
```

```
[R1-ospf-10]undo peer 10.0.123.3
```

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]ospf network-type broadcast
```

```
[R2-Serial1/0/0]fr map ip 10.0.123.1 201
```

```
[R2-Serial1/0/0]ospf 10
```

```
[R2-ospf-10]undo peer 10.0.123.1
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]ospf network-type broadcast
```

```
[R3-Serial1/0/0]fr map ip 10.0.123.1 301
```

```
[R3-Serial1/0/0]ospf 10
```

```
[R3-ospf-10]undo peer 10.0.123.1
```

配置完成后，在R1 上使用display ospf peer brief命令查看OSPF 邻居信息。

```
<R1>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Peer Statistic Information
```


Area Id	Interface	Neighbor id	State

可以看到，R1没有建立起任何邻居关系。

重新修改接R1、R2、R3的Serial 1/0/0 接口的帧中继映射，添加关键字Broadcast。

```
[R1-Serial1/0/0]fr map ip 10.0.123.2 102 broadcast
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.3 103 broadcast
```



```
[R2-Serial1/0/0]fr map ip 10.0.123.1 201 broadcast
```

```
[R3-Serial1/0/0]fr map ip 10.0.123.1 301 broadcast
```

配置完成后，在R1 上使用display ospf peer 命令来查看OSPF 邻居信息。

```
[R1]display ospf peer
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.123.1(Serial1/0/0)'s neighbors
```

```
Router ID: 10.0.2.2      Address: 10.0.123.2
```

```
State: Full Mode:Nbr is Master Priority: 0
```

```
DR: 10.0.123.1 BDR: None MTU: 0
```

```
Dead timer due in 32 sec
```

```
Retrans timer interval: 5
```

```
Neighbor is up for 00:00:09
```

```
Authentication Sequence: [ 0 ]
```

```
Router ID: 10.0.3.3      Address: 10.0.123.3
```

```
State: Full Mode:Nbr is Master Priority: 0
```

```
DR: 10.0.123.1 BDR: None MTU: 0
```

```
Dead timer due in 31 sec
```

```
Retrans timer interval: 5
```

```
Neighbor is up for 00:00:10
```

Authentication Sequence: [0]

可以看到，R1被选举成为了DR，而R2和R3的接口因为DR优先级的值已被设置为0，所以没有参加选举。

测试R2 和R3 的Loopback 0 接口之间的连通性。

```
<R3>ping -a 10.0.3.3 10.0.2.2
```

```
PING 10.0.2.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.2.2: bytes=56 Sequence=1 ttl=254 time=20 ms
```

```
Reply from 10.0.2.2: bytes=56 Sequence=2 ttl=254 time=20 ms
```

```
Reply from 10.0.2.2: bytes=56 Sequence=3 ttl=254 time=10 ms
```

```
Reply from 10.0.2.2: bytes=56 Sequence=4 ttl=254 time=20 ms
```

```
Reply from 10.0.2.2: bytes=56 Sequence=5 ttl=254 time=10 ms
```

```
--- 10.0.2.2 ping statistics ---
```

```
5 packet(s) transmitted
```

```
5 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 10/16/20 ms
```

可以看到，R2 与R3 之间可以互通，R2 和R3 的Loopback 0 接口之间可以正常通信。至此，R1、R2、R3 已构成了一个OSPF Broadcast 网络，它通过组播OSPF Hello 报文自动发现邻居并建立邻接关系，并且需要进行DR和BDR的选举。

4．配置OSPF的点到多点网络类型

接下来，将现在的网络类型修改配置为点到多点网络。点到多点类型与点到点类型非常相似，点到多点网络可以理解为由多个点到点网络组成，它通过组播OSPF Hello 报文自动发现邻居并建立邻接关系，不选举也不存在DR和BDR。

```
[R1]interface Serial 1/0/0
```

```
[R1-Serial1/0/0]ospf network-type p2mp
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.2 102 broadcast
```

```
[R1-Serial1/0/0]fr map ip 10.0.123.3 103 broadcast
```

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]ospf network-type p2mp
```

```
[R2-Serial1/0/0]undo ospf dr-priority
```

```
[R2-Serial1/0/0]undo fr map ip 10.0.123.3 201
```

```
[R2-Serial1/0/0]fr map ip 10.0.123.1 201 broadcast
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]ospf network-type p2mp
```

```
[R3-Serial1/0/0]undo ospf dr-priority
```

```
[R3-Serial1/0/0]undo fr map ip 10.0.123.2 301
```

```
[R3-Serial1/0/0]fr map ip 10.0.123.1 301 broadcast
```

上述配置完成后，在R1上查看OSPF邻居信息和路由表。

```
<R1>display ospf peer
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

Neighbors

Area 0.0.0.0 interface 10.0.123.1(Serial1/0/0)'s neighbors

Router ID: 10.0.2.2 Address: 10.0.123.2

State: Full Mode:Nbr is Master Priority: 1

DR: None BDR: None MTU: 0

Dead timer due in 97 sec

Retrans timer interval: 5

Neighbor is up for 00:01:14

Authentication Sequence: [0]

Router ID: 10.0.3.3 Address: 10.0.123.3

State: Full Mode:Nbr is Master Priority: 1

DR: None BDR: None MTU: 0

Dead timer due in 100 sec

Retrans timer interval: 5

Neighbor is up for 00:01:14

Authentication Sequence: [0]

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 12 Routes : 12

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.2.2/32	OSPF	10	48	D	10.0.123.2 Serial1/0/0
10.0.3.3/32	OSPF	10	48	D	10.0.123.3 Serial1/0/0
10.0.123.0/24	Direct	0	0	D	10.0.123.1 Serial1/0/0
.....					

可以看到，R1自动与R2和R3建立了OSPF邻接关系，不存在DR和BDR，R1的路由表中拥有去往R2 和R3 的Loopback 0 接口的路由。

在R2和R3上查看OSPF邻居信息和路由表。

<R2>display ospf peer brief

OSPF Process 10 with Router ID 10.0.2.2

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	Serial1/0/0	10.0.1.1	Full

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 12 Routes : 12

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	OSPF	10	48	D	10.0.123.1 Serial1/0/0
10.0.2.2/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.3.3/32	OSPF	10	96	D	10.0.123.1 Serial1/0/0
10.0.123.0/24	Direct	0	0	D	10.0.123.2 Serial1/0/0
.....					

<R3>display ospf peer brief

OSPF Process 10 with Router ID 10.0.3.3

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	Serial1/0/0	10.0.1.1	Full

<R3>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 12 Routes : 12

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	OSPF	10	48	D	10.0.123.1 Serial1/0/0
10.0.2.2/32	OSPF	10	96	D	10.0.123.1 Serial1/0/0
10.0.3.3/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					

测试R2 和R3 的Loopback 0 接口之间的连通性。

<R2>ping -a 10.0.2.2 10.0.3.3

PING 10.0.3.3: 56 data bytes, press CTRL_C to break

Reply from 10.0.3.3: bytes=56 Sequence=1 ttl=254 time=10 ms

Reply from 10.0.3.3: bytes=56 Sequence=2 ttl=254 time=10 ms

Reply from 10.0.3.3: bytes=56 Sequence=3 ttl=254 time=20 ms

Reply from 10.0.3.3: bytes=56 Sequence=4 ttl=254 time=20 ms

Reply from 10.0.3.3: bytes=56 Sequence=5 ttl=254 time=20 ms

--- 10.0.3.3 ping statistics ---

5 packet(s) transmitted

5 packet(s) received

0.00% packet loss

round-trip min/avg/max = 10/16/20 ms

可以看到，通信正常。至此，R1、R2、R3 构成了一个 OSPF 点到多点网络。最后需要指出的是，与其他OSPF网络类型相比，在Hub-Spoke的网络结构上应用点到多点网络类型是最为合适的，配置工作也最为简便。

思考

在Hub-Spoke的网络架构中，如果Hub端路由器的接口类型为点到多点类型，Spoke端路由器的接口类型为点到点类型，那么它们可以建立起邻接关系吗？为什么？

2.8 OSPF 路由聚合

原理概述

与RIP不同，OSPF不支持自动路由聚合，仅支持手动路由聚合。OSPF的路由聚合有两种机制：区域间路由聚合和外部路由聚合。区域间路由聚合必须配置在 ABR 路由器上，指的是ABR 在把与自己直接相连区域（Area）中的Type-1和Type-2 LSA转换成Type-3 LSA时，对生成的Type-3 LSA进行聚合。外部路由聚合必须配置在ASBR 路由器上，指的是ASBR 对Type-5 LSA进行聚合。

区域间路由聚合是 ABR 对与自己直接相连区域内的路由进行聚合，从而减少传播至与自己直接相连的其他区域的Type-3 LSA的数量。需要特别强调的是，区域间路由只能聚合由Type-1 LSA 或Type-2 LSA产生的路由；如果路由是由外部或其他区域传到本区域的（或者说路由是由Type-5 LSA或Type-3 LSA生成的），则对于这样的路由ABR是不能够进行聚合的。

外部路由聚合是指在ASBR路由器上针对引入OSPF网络的外部路由进行的聚合，目的是减少在 OSPF 网络中的 Type-5 LSA 的数量。外部路

由聚合必须在外部路由进入OSPF网络的ASBR上进行；外部路由进入OSPF网络后，在ABR上是无法对相应的Type-5 LSA进行聚合的。

对于NSSA区域，当该区域的ABR将Type-7 LSA转换为Type-5 LSA时，该ABR也可以充当ASBR的角色，并对Type-5 LSA进行聚合。需要注意的是，当NSSA区域存在多台ABR时，必须由Router-ID最大的ABR进行Type-7 LSA到Type-5 LSA的转换操作。NSSA区域的外部路由聚合有两种方式，一种是在NSSA区域的ASBR上直接对外部路由进行聚合，另一种是在NSSA区域中Router-ID最大的、负责将Type-7 LSA转成Type-5 LSA的ABR上进行聚合。

实验目的

- 理解OSPF 区域间路由聚合和外部路由聚合的概念和过程
- 掌握配置OSPF 区域间路由聚合和外部路由聚合的方法

实验内容

实验拓扑如图2-14所示，实验编址如表2-8所示。本实验模拟了一个企业网络场景，R1、R2、R3为公司总部网络路由器，R4为分支机构路由器，R5为外部非OSPF网络的路由器，SW1为公司总部内部的交换机，R1、R2、R3、R4与SW1运行OSPF协议。PC-1、PC-2、PC-3分别属于VLAN 2、VLAN 3、VLAN 4；SW1与R1之间的链路属于VLAN 5，且属于区域1。R1与R2、R1与R3之间的链路属于区域0，R2与R4、R3与R4之间的链路属于区域2。区域2是一个NSSA区域，R4使用静态路由去往R5的Loopback接口所模拟的外部网络。网络管理员需要在实现全网互通的前提下，尽可能地精简LSDB和优化路由表。

实验拓扑

figure_0131_0052

图2-14 OSPF路由聚合

实验编址表

表2-8 实验编址

figure_0132_0053

实验步骤

1．基本配置

根据图2-14和表2-8进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=300 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 300/300/300 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF及路由引入

OSPF协议的配置及NSSA区域的配置等过程在此省略。需要说明的是，SW1及每台路由器都使用了自己的Loopback 0 接口的IP 地址作为Router-ID。

在R4上配置去往外部网络的静态路由，并进行引入。

```
[R4]ip route-static 20.0.5.1 255.255.255.255 10.0.45.5
```

```
[R4]ip route-static 20.0.5.2 255.255.255.255 10.0.45.5
```

```
[R4]ip route-static 20.0.5.3 255.255.255.255 10.0.45.5
```

```
[R4]ospf 10
```

```
[R4-ospf-10]import-route static
```

配置完成后，在R1上查看邻居状态。

```
[R1]display ospf peer
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.12.1(GigabitEthernet0/0/1)'s neighbors
```

```
Router ID: 10.0.2.2      Address: 10.0.12.2
```

```
State: Full Mode:Nbr is Master Priority: 1
```

```
DR: 10.0.12.2 BDR: 10.0.12.1 MTU: 0
```

```
Dead timer due in 31 sec
```

```
Retrans timer interval: 5
```

```
Neighbor is up for 00:02:55
```

```
Authentication Sequence: [ 0 ]
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.13.1(GigabitEthernet0/0/2)'s neighbors
```

```
Router ID: 10.0.3.3      Address: 10.0.13.3
```

```
State: Full Mode:Nbr is Master Priority: 1
```

DR: 10.0.13.3 BDR: 10.0.13.1 MTU: 0

Dead timer due in 38 sec

Retrans timer interval: 5

Neighbor is up for 00:02:22

Authentication Sequence: [0]

Neighbors

Area 0.0.0.1 interface 10.0.16.1(GigabitEthernet0/0/0)'s neighbors

Router ID: 10.0.6.6 Address: 10.0.16.6

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.16.6 BDR: 10.0.16.1 MTU: 0

Dead timer due in 32 sec

Retrans timer interval: 0

Neighbor is up for 00:01:43

Authentication Sequence: [0]

可以看到 , R1与R2、 R3、 SW1的邻居关系状态都是Full。请读者自行查看其他邻居关系的状态。

查看R1的路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 26 Routes : 31

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.2.2/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.12.2
10.0.3.3/32 GigabitEthernet0/0/2	OSPF	10	1	D	10.0.13.3
10.0.4.4/32 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.12.2
OSPF	10	2	D	10.0.13.3	GigabitEthernet0/0/2
10.0.12.0/24 GigabitEthernet0/0/1	Direct	0	0	D	10.0.12.1
10.0.12.1/32 GigabitEthernet0/0/1	Direct	0	0	D	127.0.0.1
10.0.12.255/32 GigabitEthernet0/0/1	Direct	0	0	D	127.0.0.1
10.0.13.0/24 GigabitEthernet0/0/2	Direct	0	0	D	10.0.13.1
10.0.13.1/32 GigabitEthernet0/0/2	Direct	0	0	D	127.0.0.1
10.0.13.255/32 GigabitEthernet0/0/2	Direct	0	0	D	127.0.0.1

10.0.16.0/24	Direct	0	0	D	10.0.16.1
GigabitEthernet0/0/0					
10.0.16.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.16.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.24.0/24	OSPF	10	2	D	10.0.12.2
GigabitEthernet0/0/1					
10.0.34.0/24	OSPF	10	2	D	10.0.13.3
GigabitEthernet0/0/2					
10.0.45.0/24	OSPF	10	50	D	10.0.12.2
GigabitEthernet0/0/1					
OSPF	10	50	D	10.0.13.3	GigabitEthernet0/0/2
20.0.1.0/24	OSPF	1	2	D	10.0.16.6
GigabitEthernet0/0/0					
20.0.2.0/24	OSPF	10	2	D	10.0.16.6
GigabitEthernet0/0/0					
20.0.3.0/24	OSPF	10	2	D	10.0.16.6
GigabitEthernet0/0/0					
20.0.5.1/32	O_ASE	150	1	D	10.0.12.2
GigabitEthernet0/0/1					
O_ASE	150	1	D	10.0.13.3	GigabitEthernet0/0/2
20.0.5.2/32	O_ASE	150	1	D	10.0.12.2
GigabitEthernet0/0/1					
O_ASE	150	1	D	10.0.13.3	GigabitEthernet0/0/2

```

20.0.5.3/32    O_ASE  150  1    D    10.0.12.2
GigabitEthernet0/0/1

O_ASE  150  1    D    10.0.13.3  GigabitEthernet0/0/2

127.0.0.0/8    Direct    0  0    D    127.0.0.1      InLoopBack0

127.0.0.1/32   Direct    0  0    D    127.0.0.1      InLoopBack0

127.255.255.255/32 Direct    0  0    D    127.0.0.1
InLoopBack0

255.255.255.255/32 Direct    0  0    D    127.0.0.1
InLoopBack0

```

可以看到，R1已经接收到了所有的非直连路由，全网已经实现了互通。读者可自行查看其他设备上的路由表。

3．配置区域间路由聚合

在R2上查看LSDB。

```
<R2>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.2.2
```

```
Link State Database
```

```
Area: 0.0.0.0
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	170	48	80000015	1
Router	10.0.2.2	10.0.2.2	134	48	80000014	1
Router	10.0.1.1	10.0.1.1	143	60	80000038	1
Network	10.0.13.3	10.0.3.3	166	32	80000002	0

Network	10.0.12.2	10.0.2.2	134	32	80000002	0
Sum-Net	20.0.3.0	10.0.1.1	136	28	80000001	2
Sum-Net	20.0.2.0	10.0.1.1	136	28	80000001	2
Sum-Net	10.0.34.0	10.0.3.3	173	28	80000006	1
Sum-Net	10.0.34.0	10.0.2.2	123	28	80000001	2
Sum-Net	20.0.1.0	10.0.1.1	136	28	80000001	2
Sum-Net	10.0.24.0	10.0.2.2	185	28	80000001	1
Sum-Net	10.0.24.0	10.0.3.3	157	28	80000001	2
Sum-Net	10.0.16.0	10.0.1.1	189	28	80000004	1
Sum-Net	10.0.4.4	10.0.3.3	157	28	80000001	1
Sum-Net	10.0.4.4	10.0.2.2	123	28	80000001	1
Sum-Net	10.0.45.0	10.0.3.3	157	28	80000001	49
Sum-Net	10.0.45.0	10.0.2.2	123	28	80000001	49

Area: 0.0.0.2

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	157	36	8000000A	1
Router	10.0.4.4	10.0.4.4	119	72	80000020	1
Router	10.0.2.2	10.0.2.2	122	36	8000000A	1
Network	10.0.24.4	10.0.4.4	119	32	80000002	0
Network	10.0.34.4	10.0.4.4	157	32	80000002	0

Sum-Net	20.0.3.0	10.0.2.2	123	28	80000003	3
Sum-Net	20.0.3.0	10.0.3.3	137	28	80000001	3
Sum-Net	20.0.2.0	10.0.2.2	123	28	80000003	3
Sum-Net	20.0.2.0	10.0.3.3	138	28	80000001	3
Sum-Net	20.0.1.0	10.0.2.2	124	28	80000003	3
Sum-Net	20.0.1.0	10.0.3.3	138	28	80000001	3
Sum-Net	10.0.13.0	10.0.2.2	124	28	80000003	2
.....						

可以看到，目前 R2 为每一台 PC 所属的网络都单独维护了 Type-3 LSA (Sum-Net LSA)。

查看R2的路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
-----
```

Routing Tables: Public

Destinations : 24 Routes : 24

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.12.1

.....

```

10.0.45.0/24    OSPF      10  49    D    10.0.24.4
GigabitEthernet0/0/2

20.0.1.0/24    OSPF      10   3     D    10.0.12.1
GigabitEthernet0/0/1

20.0.2.0/24    OSPF      10   3     D    10.0.12.1
GigabitEthernet0/0/1

20.0.3.0/24    OSPF      10   3     D    10.0.12.1
GigabitEthernet0/0/1

20.0.5.1/32    O_NSSA    150   1     D    10.0.24.4
GigabitEthernet0/0/2

```

.....

可以看到，目前R2的路由表中拥有每一台PC所属网络的路由。为了减少LSDB中Type-3 LSA的数量以及路由表中路由条目的数量，下面将进行区域间路由聚合。

在SW1上使用命令abr-summary配置区域间路由聚合。

```
[SW1-ospf-10-area-0.0.0.1]abr-summary 20.0.0.0 255.255.252.0
```

配置完成后，查看R1的LSDB及路由表。

```
<R1>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Link State Database
```

```
Area: 0.0.0.0
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence
Metric					

Router 80000009	10.0.3.3 1	10.0.3.3	1760	48
Router 80000009	10.0.2.2 1	10.0.2.2	1769	48
Router 8000000B	10.0.1.1 1	10.0.1.1	1336	60
Network 80000003	10.0.13.3 0	10.0.3.3	1337	32
Network 80000004	10.0.12.2 0	10.0.2.2	1364	32
Sum-Net 80000001	20.0.3.0 2	10.0.1.1	76	28
Sum-Net 80000001	20.0.2.0 2	10.0.1.1	76	28
Sum-Net 80000003	10.0.34.0 1	10.0.3.3	1759	28
Sum-Net 80000003	10.0.34.0 2	10.0.2.2	1719	28
Sum-Net 80000001	20.0.1.0 2	10.0.1.1	76	28
Sum-Net 80000003	10.0.24.0 1	10.0.2.2	1768	28

.....

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 25 Routes : 29

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					
20.0.1.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.16.6
20.0.2.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.16.6
20.0.3.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.16.6
.....					

可以看到，R1 在LSDB 中为每一台PC所属的网络都单独维护了Type-3 LSA，在路由表中为每一台 PC 所属的网络都单独维护了路由，这说明所配置的区域间路由聚合并没有产生作用，原因是只有在ABR上才能进行区域间路由聚合，而SW1并非ABR。

删除在SW1上所进行的配置。

```
[SW1]ospf 10
```

```
[SW1-ospf-10]area 1
```

```
[SW1-ospf-10-area-0.0.0.1]undo abr-summary 20.0.0.0 255.255.252.0
```

在ABR路由器R2的区域0中配置区域间路由聚合。

```
[R2]ospf 10
```

```
[R2-ospf-10]area 0
```

```
[R2-ospf-10-area-0.0.0.0]abr-summary 20.0.0.0 255.255.252.0
```

配置完成后，查看R4的LSDB及路由表。

```
<R4>display ospf lsdb
```

OSPF Process 10 with Router ID 10.0.4.4

Link State Database

Area: 0.0.0.2

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	673	36	80000006	1
Router	10.0.4.4	10.0.4.4	579	60	80000007	1
Router	10.0.2.2	10.0.2.2	673	36	80000006	1
Network	10.0.24.4	10.0.4.4	673	32	80000003	0
Network	10.0.34.4	10.0.4.4	673	32	80000003	0
Sum-Net	20.0.3.0	10.0.2.2	722	28	80000002	3
Sum-Net	20.0.3.0	10.0.3.3	714	28	80000002	3
Sum-Net	20.0.2.0	10.0.2.2	722	28	80000002	3
Sum-Net	20.0.2.0	10.0.3.3	714	28	80000002	3
Sum-Net	20.0.1.0	10.0.2.2	722	28	80000002	3
Sum-Net	20.0.1.0	10.0.3.3	714	28	80000002	3

Sum-Net	10.0.13.0	10.0.2.2	722	28	80000002	2

.....

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 28 Routes : 34

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

0.0.0.0/0 GigabitEthernet0/0/2	O_NSSA	150	1	D	10.0.24.2

O_NSSA	150	1	D	10.0.34.3	GigabitEthernet0/0/1

.....

10.0.45.255/32	Direct	0	0	D	127.0.0.1	Serial1/0/0

20.0.1.0/24 GigabitEthernet0/0/2	OSPF	10	4	D	10.0.24.2

OSPF	10	4	D	10.0.34.3	GigabitEthernet0/0/1

20.0.2.0/24 GigabitEthernet0/0/2	OSPF	10	4	D	10.0.24.2

OSPF	10	4	D	10.0.34.3	GigabitEthernet0/0/1

20.0.3.0/24 GigabitEthernet0/0/2	OSPF	10	4	D	10.0.24.2

```
OSPF      10   4   D   10.0.34.3  GigabitEthernet0/0/1
20.0.5.1/32      Static   60   0   RD   10.0.45.5   Serial1/0/0
.....
```

可以看到，R4 在LSDB 中为每一台PC所属的网络都单独维护了Type-3 LSA，在路由表中为每一台PC所属的网络都单独维护了路由，这说明在ABR路由器R2上所配置的区域间路由聚合也没有产生作用，原因是ABR 只能对与自己直接相连的区域进行区域间路由聚合。

删除在R2上所进行的配置。

```
[R2]ospf 10
```

```
[R2-ospf-10]area 0
```

```
[R2-ospf-10-area-0.0.0.0]undo abr-summary 20.0.0.0 255.255.252.0
```

在ABR路由器R1的区域1中配置区域间路由聚合。

```
[R1]ospf 10
```

```
[R1-ospf-10]area 1
```

```
[R1-ospf-10-area-0.0.0.1]abr-summary 20.0.0.0 255.255.252.0
```

配置完成后，查看R2的LSDB及路由表。

```
<R2>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.2.2
```

```
Link State Database
```

```
Area: 0.0.0.0
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence
Metric					

Router 80000009	10.0.3.3 1	10.0.3.3	1168	48
Router 80000009	10.0.2.2 1	10.0.2.2	1175	48
Router 8000000B	10.0.1.1 1	10.0.1.1	743	60
Network 80000003	10.0.13.3 0	10.0.3.3	745	32
Network 80000004	10.0.12.2 0	10.0.2.2	769	32
Sum-Net 80000003	10.0.34.0 1	10.0.3.3	1167	28
Sum-Net 80000003	10.0.34.0 2	10.0.2.2	1125	28
Sum-Net 80000003	10.0.24.0 1	10.0.2.2	1174	28
Sum-Net 80000002	10.0.24.0 2	10.0.3.3	1127	28
Sum-Net 80000001	20.0.0.0 2	10.0.1.1	289	28
Sum-Net 80000003	10.0.16.0 1	10.0.1.1	741	28
Sum-Net 80000002	10.0.4.4 1	10.0.2.2	1129	28
Sum-Net 80000002	10.0.4.4 1	10.0.3.3	1127	28

Sum-Net 80000002	10.0.45.0 49	10.0.2.2	1626	28

Sum-Net 80000002	10.0.45.0 49	10.0.3.3	1628	28

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 80000006	10.0.3.3 1	10.0.3.3	1126	36	
Router 80000007	10.0.4.4 1	10.0.4.4	1033	60	
Router 80000006	10.0.2.2 1	10.0.2.2	1125	36	
Network 80000003	10.0.24.4 0	10.0.4.4	1125	32	
Network 80000003	10.0.34.4 0	10.0.4.4	1126	32	
Sum-Net 80000002	10.0.13.0 2	10.0.2.2	1174	28	
Sum-Net 80000002	10.0.13.0 1	10.0.3.3	1167	28	
Sum-Net 3	20.0.0.0	10.0.2.2	288	28	80000001
Sum-Net 3	20.0.0.0	10.0.3.3	290	28	80000001

```
Sum-Net      10.0.12.0    10.0.2.2      1174  28    80000002
1
```

.....

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
-----
```

Routing Tables: Public

Destinations : 21 Routes : 21

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

10.0.1.1/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.12.1

.....

10.0.45.0/24 GigabitEthernet0/0/2	OSPF	10	49	D	10.0.24.4

20.0.0.0/22 GigabitEthernet0/0/1	OSPF	10	3	D	10.0.12.1

20.0.5.1/32 GigabitEthernet0/0/2	O_NSSA	150	1	D	10.0.24.4

.....

可以看到，现在R2的LSDB 中没有为每一台PC所属的网络单独维护Type-3 LSA，维护的是聚合后的Type-3 LSA；R2 的路由表中去往每一台PC所属的网络的明细路由也被聚合后的路由取代了。

4 . 配置外部路由聚合

通过前面的步骤，已经实现了区域间路由聚合。然而，以 Type-7 LSA 的形式进入OSPF 网络的外部路由仍然未被聚合，LSDB 仍然会为每一条外部路由单独维护一条LSA，路由表中也会为每一条这样的LSA产生明细路由。

在R4上使用命令asbr-summary配置外部路由聚合。

```
[R4-ospf-10]asbr-summary 20.0.5.0 255.255.255.252
```

配置完成后，查看R1的LSDB及路由表。

```
<R1>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Link State Database
```

```
Area: 0.0.0.0
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	1704	48	8000000A	1
.....						

```
AS External Database
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	20.0.5.0	10.0.3.3	193	36	80000001	2

```
<R1>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```
-----  
-----
```

Routing Tables: Public

Destinations : 23 Routes : 25

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					
20.0.3.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.16.6
20.0.5.0/30 GigabitEthernet0/0/2	O_ASE	150	2	D	10.0.13.3
O_ASE	150	2	D	10.0.12.2	GigabitEthernet0/0/1
127.0.0.0/8	Direct	0	0	D	127.0.0.1 InLoopBack0
.....					

可以看到，R1 的LSDB 中没有明细Type-5 LSA，只有聚合了的Type-5 LSA。R1 的路由表中也没有外部网络的明细路由，而只有聚合后的路由。

查看R2的LSDB及路由表。

```
<R2>display ospf lsdb
```

OSPF Process 10 with Router ID 10.0.2.2

Link State Database

.....

Area: 0.0.0.2

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.3.3	10.0.3.3	985	36	80000004	1
Router	10.0.4.4	10.0.4.4	984	72	8000000C	1
.....						
Sum-Net	10.0.16.0	10.0.3.3	993	28	80000002	2
NSSA	0.0.0.0	10.0.2.2	997	36	80000002	1
NSSA	0.0.0.0	10.0.3.3	993	36	80000002	1
NSSA	20.0.5.0	10.0.4.4	76	36	80000001	2

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	20.0.5.0	10.0.3.3	77	36	80000001	2

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 20 Routes : 20

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.12.1

.....

20.0.0.0/22	OSPF	10	3	D	10.0.12.1	
GigabitEthernet0/0/1						
20.0.5.0/30	O_NSSA	150	2	D	10.0.24.4	
GigabitEthernet0/0/2						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

可以看到，R2的LSDB中没有明细Type-5 LSA和Type-7 LSA，只有聚合后的Type-5 LSA 和Type-7 LSA，R2 的路由表中没有外部网络的明细路由，只有聚合后的路由。

5．在NSSA区域的ABR上配置外部路由聚合

由于区域 2 是 NSSA 区域，该区域的 ABR 路由器会将 Type-7 LSA 转换为 Type-5 LSA，并泛洪到区域0。

先删除R4上的路由聚合配置，然后在区域2的ABR路由器R2上配置外部路由聚合。

```
[R4-ospf-10]undo asbr-summary 20.0.5.0 255.255.255.252
```

```
[R2-ospf-10]asbr-summary 20.0.5.0 255.255.255.252
```

配置完成后，查看R1的LSDB。

```
<R1>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

Link State Database

.....

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	20.0.5.1	10.0.3.3	38	36	80000001	1
External	20.0.5.3	10.0.3.3	38	36	80000001	1
External	20.0.5.2	10.0.3.3	38	36	80000001	1

可以观察到，R1 的LSDB中针对每一条外部明细路由都有一条相应的 Type-5 LSA，这说明在 ABR 路由器 R2 上进行的外部路由聚合配置并未生效。原来，将 Type-7 LSA转换为Type-5 LSA 的是Router-ID 较大的 ABR 路由器R3，所以，在R2 上进行的外部路由聚合配置不能生效

查看R1上的路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 26 Routes : 31

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0 0	D	127.0.0.1	LoopBack0

.....

```

20.0.3.0/24    OSPF      10  2    D    10.0.16.6
GigabitEthernet0/0/0

20.0.5.1/32    O_ASE    150  1    D    10.0.12.2
GigabitEthernet0/0/1

O_ASE  150  1    D    10.0.13.3  GigabitEthernet0/0/2

20.0.5.2/32    O_ASE    150  1    D    10.0.12.2
GigabitEthernet0/0/1

O_ASE  150  1    D    10.0.13.3  GigabitEthernet0/0/2

20.0.5.3/32    O_ASE    150  1    D    10.0.12.2
GigabitEthernet0/0/1

O_ASE  150  1    D    10.0.13.3  GigabitEthernet0/0/2

127.0.0.0/8    Direct    0  0    D    127.0.0.1      InLoopBack0

127.0.0.1/32   Direct    0  0    D    127.0.0.1      InLoopBack0

127.255.255.255/32 Direct    0  0    D    127.0.0.1
InLoopBack0

255.255.255.255/32 Direct    0  0    D    127.0.0.1
InLoopBack0

```

可以看到，R1上外部网络的路由全部是明细路由。

现在，保留R2上的外部路由聚合配置，并在R3上配置外部路由聚合。

```
[R3-ospf-10]asbr-summary 20.0.5.0 255.255.255.252
```

配置完成后，查看R1的LSDB及路由表。

```
<R1>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.1.1
```


Link State Database

.....

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	20.0.5.0	10.0.3.3	11	36	80000001	2

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 24 Routes : 26

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					
20.0.3.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.16.6
20.0.5.0/30 GigabitEthernet0/0/2	O_ASE	150	2	D	10.0.13.3
127.0.0.0/8	Direct	0	0	D	127.0.0.1 InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1 InLoopBack0
127.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1

```
255.255.255.255/32   Direct    0    0    D    127.0.0.1
InLoopBack0
```

可以看到，R1 的LSDB 中没有明细的Type-5 LSA，只有AdvRouter为R3（10.0.3.3）的聚合后的Type-5 LSA，路由表中没有外部网络的明细路由，只有外部网络的聚合路由，下一跳为R3（10.0.13.3），这说明R3上外部路由聚合配置已经生效。

关闭R3 的GE 0/0/1 接口，模拟R3 发生了故障。

```
[R3-GigabitEthernet0/0/1]shutdown
```

再查看R1的LSDB。

```
<R1>display ospf lsdb
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Link State Database
```

```
.....
```

```
AS External Database
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
External	20.0.5.0	10.0.2.2	9	36	80000001	2

可以看到，R1 上聚合后的Type-5 LSA的AdvRouter 变成了R2（10.0.2.2）。

查看R1的路由表。

```
<R1>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

Routing Tables: Public

Destinations : 23 Routes : 23

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					
20.0.3.0/24 GigabitEthernet0/0/0	OSPF	10	2	D	10.0.16.6
20.0.5.0/30 GigabitEthernet0/0/1	O_ASE	150	2	D	10.0.12.2
127.0.0.0/8	Direct	0	0	D	127.0.0.1 InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1 InLoopBack0
127.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1
255.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1

可以看到，R1去往外部网络的聚合路由的下一跳变成了R2（10.0.12.2）。上面的实验说明，ABR路由器R2上的外部路由聚合配置，是对ABR路由器R3上的外部路由聚合配置的一个冗余备份。

思考

在OSPF Stub 区域的ABR 上能不能配置区域间路由聚合呢？

2.9 OSPF 监测和调试

原理概述

为了监测OSPF协议的工作状态，VRP系统提供了一系列的查询命令。熟练使用这些命令，可以全面地了解网络的运行情况。同时，VRP系统还提供了一系列的调试命令，用以详细地了解和调试OSPF的工作过程，并知道工作过程中各种事件的细节和关系。查询命令和调试命令的结合使用，有助于快速查找到网络的故障点和故障原因，提高查错排错的效率。

实验目的

- 掌握监测OSPF 工作状态的方法
- 掌握调试OSPF 工作过程的方法

实验内容

实验拓扑如图2-15所示，实验编址如表2-9所示。本实验模拟了一个企业网络场景，R1 和 R2 为公司总部网络的路由器，R3 为分支机构的路由器，R1、R2、R3 上都运行OSPF协议。R4为公司外部网络的路由器，使用缺省路由访问公司网络。R3使用静态路由访问R4的所有Loopback接口所模拟的外部网络，这些静态路由被引入到公司的OSPF网络时需要被聚合。R1与R2、R2与R3之间的接口上需要启用HMAC-MD5认证功能。

实验拓扑

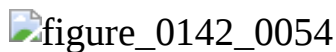
figure_0142_0054

图2-15 OSPF监测和调试

实验编址表

表2-9 实验编址

figure_0142_0055

实验步骤

1. 基本配置

根据图2-15和表2-9进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 60/60/60 ms
```

其余直连网段的连通性测试过程在此省略。

在R1、R2、R3上配置OSPF协议，配置静态路由，引入外部路由，配置外部路由聚合，并在相应的接口上配置认证功能。

```
[R1]ospf router-id 10.0.1.1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]ospf authentication-mode hmac-md5 1 plain  
huawei
```

```
[R2]ospf router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.12.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]area 1
```

```
[R2-ospf-1-area-0.0.0.1]network 10.0.23.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.1]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]ospf authentication-mode hmac-md5 1 plain  
huawei
```

```
[R2-GigabitEthernet0/0/0]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]ospf authentication-mode hmac-md5 1 plain huawei
```

```
[R3]ip route-static 10.0.100.1 255.255.255.255 10.0.34.4
```

```
[R3]ip route-static 10.0.100.2 255.255.255.255 10.0.34.4
```

```
[R3]ip route-static 10.0.100.3 255.255.255.255 10.0.34.4
```

```
[R3]ospf router-id 10.0.3.3
```

```
[R3-ospf-1]area 1
```

```
[R3-ospf-1-area-0.0.0.1]network 10.0.3.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.1]network 10.0.23.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.1]import-route static
```

```
[R3-ospf-1-area-0.0.0.1]quit
```

```
[R3-ospf-1]asbr-summary 10.0.100.0 255.255.255.252
```

```
[R3-ospf-1]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]ospf authentication-mode hmac-md5 1 plain huawei
```

2 . 监测OSPF的基本状态

完成上述配置后，在R2上使用display ospf peer命令查看OSPF 邻居的相关信息。

```
<R2>display ospf peer
```

```
OSPF Process 1 with Router ID 10.0.2.2
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.12.2(GigabitEthernet0/0/0)'s neighbors
```

```
Router ID: 10.0.1.1      Address: 10.0.12.1
```

```
State: Full Mode:Nbr is Slave Priority: 1
```

```
DR: 10.0.12.2 BDR: 10.0.12.1 MTU: 0
```

```
Dead timer due in 37 sec
```

```
Retrans timer interval: 5
```

```
Neighbor is up for 00:05:30
```

```
Authentication Sequence: [63]
```

```
Neighbors
```

```
Area 0.0.0.1 interface 10.0.23.2(Serial1/0/0)'s neighbors
```

Router ID: 10.0.3.3 Address: 10.0.23.3

State: Full Mode:Nbr is Master Priority: 1

DR: None BDR: None MTU: 0

Dead timer due in 40 sec

Retrans timer interval: 5

Neighbor is up for 00:05:59

Authentication Sequence: [63]

回显信息表明，R2已经与区域0的R1（10.0.1.1）以及区域1的R3（10.0.3.3）建立了邻接关系，状态为Full。回显信息中还出现了诸如邻居的接口地址，邻居的DR优先级，邻居在LSDB同步时的主从角色等参数。

在R2 上使用display ospf peer brief命令查看邻居的概要信息。

<R2>display ospf peer brief

OSPF Process 1 with Router ID 10.0.2.2

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	10.0.1.1	Full
0.0.0.1	Serial1/0/0	10.0.3.3	Full

可以看到，回显信息中包含了邻居所在的区域，邻居的连接接口，邻居的Router-ID和邻居关系的当前状态。

在R2 上使用display ospf interface 命令查看运行OSPF 协议的接口信息。

```
<R2>display ospf interface
```

OSPF Process 1 with Router ID 10.0.2.2

Interfaces

Area: 0.0.0.0 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.12.2	Broadcast	DR	1	1	10.0.12.2	10.0.12.1
10.0.2.2	P2P	P-2-P	0	1	0.0.0.0	0.0.0.0

Area: 0.0.0.1 (MPLS TE not enabled)

IP Address	Type	State	Cost	Pri	DR	BDR
10.0.23.2	P2P	P-2-P	48	1	0.0.0.0	0.0.0.0

可以看到，回显信息包含了接口的 IP 地址，接口的类型，接口的开销值，接口的DR优先级等参数。

在R2 上使用display ospf interface GigabitEthernet 0/0/0命令查看接口GE 0/0/0 的详细信息。

```
<R2>display ospf interface GigabitEthernet 0/0/0
```

OSPF Process 1 with Router ID 10.0.2.2

Interfaces

Interface: 10.0.12.2 (GigabitEthernet0/0/0)

Cost: 1 State:DR Type:Broadcast MTU: 1500

Priority: 1

Designated Router: 10.0.12.2

Backup Designated Router: 10.0.12.1

Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1

可以看到，回显信息包含了 GE 0/0/0 接口所连网段的 DR、BDR、MTU、Hello 时间间隔等参数。

在R2 上使用display ospf lsdb命令查看LSDB。

```
<R2>display ospf lsdb
```

OSPF Process 1 with Router ID 10.0.2.2

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.2.2	10.0.2.2	992	48	80000006	1
Router	10.0.1.1	10.0.1.1	1001	60	80000006	1
Network	10.0.12.2	10.0.2.2	992	32	80000002	0
Sum-Net 80000001	10.0.3.3 48	10.0.2.2	1029	28		
Sum-Net 48	10.0.23.0	10.0.2.2	1039	28	80000001	

Sum-Asbr 48	10.0.3.3	10.0.2.2	1029	28	80000001

Area: 0.0.0.1

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence

.....

Sum-Net 1	10.0.12.0	10.0.2.2	1039	28	80000001

.....

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence

External 80000001	10.0.100.0	10.0.3.3	1040	36	2

可以看到，R2 的LSDB 成功接收到了所有的LSA。display ospf lsdb 命令后面可以通过添加关键字asbr、ase、network、nssa和summary来查看相应类型的LSA的详细信息。

在R2 上使用display ospf lsdb ase 命令查看LSDB 中的Type-5 LSA的详细信息。

<R2>display ospf lsdb ase

OSPF Process 1 with Router ID 10.0.2.2

Link State Database

Type : External

Ls id : 10.0.100.0

Adv rtr : 10.0.3.3
Ls age : 581
Len : 36
Options : E
seq# : 80000001
chksum : 0xa2b1
Net mask : 255.255.255.252
TOS 0 Metric : 2
E type : 2
Forwarding Address : 0.0.0.0
Tag : 1
Priority : Low

可以看到，回显信息包含了Type-5 LSA（AS External LSA）的详细参数。

在R2上使用display ospf routing 命令查看OSPF 路由表。

<R2>display ospf routing

OSPF Process 1 with Router ID 10.0.2.2

Routing Tables

Routing for Network

Destination	Cost	Type	NextHop	AdvRouter	Area
10.0.2.2/32	0	Stub	10.0.2.2	10.0.2.2	0.0.0.0

10.0.12.0/24	1	Transit	10.0.12.2	10.0.2.2	0.0.0.0
10.0.23.0/24	48	Stub	10.0.23.2	10.0.2.2	0.0.0.1
10.0.1.1/32	1	Stub	10.0.12.1	10.0.1.1	0.0.0.0
10.0.3.3/32	48	Stub	10.0.23.3	10.0.3.3	0.0.0.1

Routing for ASEs

Destination	Cost	Type	Tag	NextHop	AdvRouter
10.0.100.0/30	2	Type2	1	10.0.23.3	10.0.3.3

Total Nets:6

Intra Area: 5 InterArea: 0 ASE: 1 NSSA: 0

可以看到，回显信息包含了所有OSPF路由条目的相关信息。

3．调试OSPF的工作过程

上述步骤中，通过VRP提供的各种信息查看命令，能够很好地监测OSPF的基本工作状态。如果需要了解OSPF的工作过程，需要用到各种调试命令。

在R1 上使用terminal debugging 命令开启debug 功能。

```
<R1>terminal debugging
```

由于调试功能会带来大量的信息输出，如果使用不当就会导致网络设备瘫痪，所以一定要谨慎使用。使用调试功能时需要尽可能的精确，通常情况下，应该避免使用诸如debugging ip packet或是debug nat all 等信息输出特别多的调试命令。

debugging ospf event 命令是一个常用的调试命令，用来查看OSPF 协议工作过程中的所有事件。

在R1 上使用debugging ospf event命令。

```
<R1>debugging ospf event
```

结果发现没有任何消息输出，这是因为OSPF此时工作在常态，并没有发生变化事件。下面使用 reset ospf process 命令重启 OSPF 进程来观察 OSPF 邻居关系的建立过程。

```
<R1>reset ospf process
```

Warning: The OSPF process will be reset. Continue? [Y/N]:y

Aug 15 2013 03:23:18.540.9-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802c Line: 2755 Level: 0x20

OSPF 1: Intf 10.0.12.1 Rcv InterfaceDown State DR -> Down.

Aug 15 2013 03:23:18.540.10-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802d Line: 3360 Level: 0x20

OSPF 1: Nbr 10.0.12.2 Rcv KillNbr State Full -> Down.

Aug 15 2013 03:23:18.540.11-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802c Line: 2755 Level: 0x20

OSPF 1: Intf 10.0.1.1 Rcv InterfaceDown State Point-to-Point -> Down.

Aug 15 2013 03:23:18.540.12-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802c Line: 1295 Level: 0x20

OSPF 1: Intf 10.0.12.1 Rcv InterfaceUp State Down -> Waiting.

Aug 15 2013 03:23:18.540.13-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802c Line: 1409 Level: 0x20

OSPF 1 Send Hello Interface Up on 10.0.12.1

Aug 15 2013 03:23:18.540.14-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802c Line: 1295 Level: 0x20

OSPF 1: Intf 10.0.1.1 Rcv InterfaceUp State Down -> Point-to-Point.

Aug 15 2013 03:23:18.540.15-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802c Line: 1409 Level: 0x20

OSPF 1 Send Hello Interface Up on 10.0.1.1

Aug 15 2013 03:23:23.600.1-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802d Line: 1136 Level: 0x20

OSPF 1: Nbr 10.0.12.2 Rcv HelloReceived State Down -> Init.

Aug 15 2013 03:23:23.600.2-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802d Line: 1732 Level: 0x20

OSPF 1: Nbr 10.0.12.2 Rcv 2WayReceived State Init -> 2Way.

Aug 15 2013 03:23:23.610.1-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802d Line: 1732 Level: 0x20

OSPF 1: Nbr 10.0.12.2 Rcv AdjOk? State 2Way -> ExStart.

Aug 15 2013 03:23:23.610.2-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802c Line: 2107 Level: 0x20

OSPF 1: Intf 10.0.12.1 Rcv BackupSeen State Waiting -> BackupDR.

Aug 15 2013 03:23:23.660.1-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802d Line: 1845 Level: 0x20

OSPF 1: Nbr 10.0.12.2 Rcv NegotiationDone State ExStart -> Exchange.

Aug 15 2013 03:23:23.690.1-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802d Line: 1957 Level: 0x20

OSPF 1: Nbr 10.0.12.2 Rcv ExchangeDone State Exchange -> Loading.

Aug 15 2013 03:23:23.710.1-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd017802d Line: 2356 Level: 0x20

OSPF 1: Nbr 10.0.12.2 Rcv LoadingDone State Loading -> Full.

可以看到，显示信息反映了R1与R2建立邻居邻接关系的每一步过程。

需要注意的，在获取了所需的调试输出信息后，应尽快使用undo debugging all 命令关闭所有的调试功能，以减轻设备负担。

<R1>undo debugging all

Info: All possible debugging has been turned off

另一个常用的调试命令是debugging ospf packet，通常携带hello、update等关键字以便对特定类型的数据包进行调试。

在R1 上使用debugging ospf packet hello 命令查看OSPF 协议的Hello 数据包。

<R1>debugging ospf packet hello

Aug 15 2013 03:41:14.330.1-05:13 R1 RM/6/RMDEBUG:

FileID: 0xd0178024 Line: 2236 Level: 0x20

OSPF 1: RECV Packet. Interface: GigabitEthernet0/0/0

Aug 15 2013 03:41:14.330.2-05:13 R1 RM/6/RMDEBUG: Source Address:
10.0.12.2

Aug 15 2013 03:41:14.330.3-05:13 R1 RM/6/RMDEBUG: Destination
Address: 224.0.0.5

Aug 15 2013 03:41:14.330.4-05:13 R1 RM/6/RMDEBUG: Ver# 2, Type: 1
(Hello)

Aug 15 2013 03:41:14.330.5-05:13 R1 RM/6/RMDEBUG: Length: 48,
Router: 10.0.2.2

Aug 15 2013 03:41:14.330.6-05:13 R1 RM/6/RMDEBUG: Area: 0.0.0.0,
Chksum: 0

Aug 15 2013 03:41:14.330.7-05:13 R1 RM/6/RMDEBUG: AuType: 02

Aug 15 2013 03:41:14.330.8-05:13 R1 RM/6/RMDEBUG:
Key(ascii):*****

Aug 15 2013 03:41:14.330.9-05:13 R1 RM/6/RMDEBUG: Net Mask:
255.255.255.0

Aug 15 2013 03:41:14.330.10-05:13 R1 RM/6/RMDEBUG: Hello Int: 10,
Option: _E_

Aug 15 2013 03:41:14.330.11-05:13 R1 RM/6/RMDEBUG: Rtr Priority: 1,
Dead Int: 40

Aug 15 2013 03:41:14.330.12-05:13 R1 RM/6/RMDEBUG: DR: 10.0.12.2

Aug 15 2013 03:41:14.330.13-05:13 R1 RM/6/RMDEBUG: BDR: 10.0.12.1

Aug 15 2013 03:41:14.330.14-05:13 R1 RM/6/RMDEBUG: # Attached
Neighbors: 1

Aug 15 2013 03:41:14.330.15-05:13 R1 RM/6/RMDEBUG: Neighbor:
10.0.1.1

这里再次强调，一旦获取了所需的调试输出信息后，应尽快关闭所有的调试功能。

```
<R1>undo debugging all
```

Info: All possible debugging has been turned off

思考

导致OSPF邻居邻接关系不能正常建立的原因通常有哪些？

2.10 OSPF 缺省路由

原理概述

OSPF 是目前企业网络中应用最为广泛的一种IGP（Interior Gateway Protocol）路由协议。企业的OSPF网络通常需要与ISP（Internet Service Provider）相连，通过ISP来访问整个外部网络。除非有某种特别的需求，通常情况下企业网络设备和ISP设备之间不会也没有必要运行某种动态路由协议来交换路由信息。企业网络设备无需知道和维护海量的外部网络的各种路由，而是可以通过利用缺省路由的方式来实现对外部网络的访问，这样既可以精简企业网络设备的路由表规模，同时，当外部网络发生故障时，企业内部的网络也不会因此而受到影响，从而增强了企业网络的稳定性。

在OSPF网络环境中，有两种方法可以动态地注入缺省路由。第一种方法是在ASBR上手动注入缺省路由，也就是ASBR向整个OSPF网络泛洪表示缺省路由的Type-5 LSA，其他路由器通过Type-5 LSA所表示的缺省路由来访问外部网络。第二种方法是在Stub区域或Totally Stub区域以及NSSA区域或Totally NSSA区域中，由ABR自动注入缺省路由，也就是ABR向该区域泛洪表示缺省路由的Type-3 LSA或Type-7 LSA，该区域内的路由器通过Type-3 LSA或Type-7 LSA所表示的缺省路由来访问该区域以外的任何目的地。

实验目的

- 理解和掌握向OSPF 网络手动注入缺省路由的方法
- 理解和掌握向OSPF 网络自动注入缺省路由的方法

实验内容

实验拓扑如图2-16所示，实验编址如表2-10所示。本实验模拟了一个企业网络场景，R1、R2、R3、R4为企业网络路由器，其中R3和R4为总部路由器，R1和R2分别为分支机构1 和分支机构2 的路由器。R1 和R2 的Loopback 1 接口用来模拟分支机构内部的网络，R5模拟了ISP的边界路由器，R5的所有Loopback接口模拟了各种外部网络。R3与R4之间的链路属于区域0，R1与R3之间的链路属于区域2，R2与R3之间的链路属于区域1。网络需求是要实现全网互通。

实验拓扑

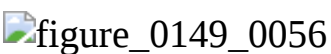
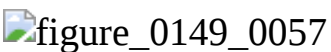


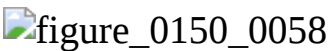
图2-16 OSPF缺省路由

实验编址表

表2-10 实验编址



（续表）



实验步骤

1．基本配置

根据图2-16和表2-10进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 10.0.13.3
```

```
PING 10.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=30 ms
```

```
--- 10.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 30/30/30 ms
```

其余直连网段的连通性测试过程在此省略。

2 . 配置OSPF路由协议

在路由器 R1、R2、R3、R4 上完成 OSPF 协议的配置，每台路由器均使用自己的Loopback 0 接口的IP 地址作为自己的Router-ID，R3 与R4 之间的链路位于区域0，R1与R3之间的链路位于区域2，R2与R3之间的链路位于区域1。

```
[R1]ospf 100 router-id 10.0.1.1
```

```
[R1-ospf-100]area 2
```

```
[R1-ospf-100-area-0.0.0.2]network 172.16.1.0 0.0.0.255
```

```
[R1-ospf-100-area-0.0.0.2]network 10.0.13.0 0.0.0.255
```

```
[R2]ospf 100 router-id 10.0.2.2
```

```
[R2-ospf-100]area 1
```

```
[R2-ospf-100-area-0.0.0.1]network 172.16.2.0 0.0.0.255
```

```
[R2-ospf-100-area-0.0.0.1]network 10.0.23.0 0.0.0.255
```

```
[R3]ospf 100 router-id 10.0.3.3
```

```
[R3-ospf-100]area 2
```

```
[R3-ospf-100-area-0.0.0.2]network 10.0.13.0 0.0.0.255
```

```
[R3-ospf-100-area-0.0.0.2]area 1
```

```
[R3-ospf-100-area-0.0.0.1]network 10.0.23.0 0.0.0.255
```

```
[R3-ospf-100-area-0.0.0.1]area 0
```

```
[R3-ospf-100-area-0.0.0.0]network 10.0.34.0 0.0.0.255
```

```
[R4]ospf 100 router-id 10.0.4.4
```

```
[R4-ospf-100]area 0
```

```
[R4-ospf-100-area-0.0.0.0]network 10.0.34.0 0.0.0.255
```

配置完成后，查看R4的LSDB。

```
[R4]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.4.4

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	973	36	80000004	1
Router	10.0.4.4	10.0.4.4	973	36	80000003	1
Network	10.0.34.4	10.0.4.4	974	32	80000002	0

Sum-Net	10.0.13.0	10.0.3.3	1017	28	80000001	1
Sum-Net	172.16.2.1	10.0.3.3	1017	28	80000001	1
Sum-Net	172.16.1.1	10.0.3.3	1017	28	80000001	1
Sum-Net	10.0.23.0	10.0.3.3	1017	28	80000001	1

可以看到，R4 的 LSDB 中包含了两条 LinkState ID 分别为 172.16.1.1 和 172.16.2.1 的 Sum-Net LSA，这说明 R4 知道去往两个分支机构的内部网络的路由了。

本实验中，R5 是 ISP 的边界路由器，R4 是企业边界路由器，显然，R5 需要拥有能够访问企业网络的能力，为此，可以在 R5 上配置一条静态缺省路由指向 R4。

```
[R5]ip route-static 0.0.0.0 0.0.0.0 202.103.68.2
```

3．向普通区域注入缺省路由

目前，企业内网已经实现了互通，但是企业网络现在还无法访问外部网络，原因是企业网络路由器上现在还缺少去往外部网络的路由。

首先，尝试在 R4 上配置一条静态缺省路由，然后使用 import-route static 命令将它引进整个 OSPF 网络。

```
[R4]ip route-static 0.0.0.0 0.0.0.0 202.103.68.1
```

```
[R4]ospf 100
```

```
[R4-ospf-100]import-route static
```

配置完成后，查看 R4 的 LSDB。

```
[R4]display ospf lsdb
```

```
OSPF Process 100 with Router ID 10.0.4.4
```

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	1384	36	80000004	1
Router	10.0.4.4	10.0.4.4	49	36	80000006	1
Network 0	10.0.34.4	10.0.4.4	1373	32	80000002	
Sum-Net 1	10.0.13.0	10.0.3.3	1410	28	80000001	
Sum-Net 80000001	172.16.2.1	10.0.3.3	1363	28		1
Sum-Net 80000001	172.16.1.1	10.0.3.3	1372	28		1
Sum-Net 1	10.0.23.0	10.0.3.3	1401	28	80000001	

可以看到，在R4的LSDB 中并没有出现LinkState ID为0.0.0.0 的LSA，这就意味着所配置的静态缺省路由并未被引进OSPF网络。为什么会出现这样的问题呢？原来，OSPF 网络规定了不允许通过import-route static 命令注入缺省路由。

删除引入静态缺省路由的配置。

```
[R4]ospf 100
```

```
[R4-ospf-100]undo import-route static
```

在R4上使用default-route-advertise命令注入一条缺省路由。注意，注入缺省路由的前提是该路由器上必须已经有了一条通过其他方法获得的缺省路由。

```
[R4]ospf 100
```

```
[R4-ospf-100]default-route-advertise
```

配置完成后，查看R4的LSDB。

```
[R4]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.4.4

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router 1	10.0.3.3	10.0.3.3	203	36	80000003	
Router	10.0.4.4	10.0.4.4	91	36	80000005	1
Network 80000001	10.0.34.4 0	10.0.4.4	202	32		
Sum-Net 80000001	10.0.13.0 1	10.0.3.3	222	28		
Sum-Net 80000001	172.16.2.1 1	10.0.3.3	189	28		
Sum-Net 80000001	172.16.1.1 1	10.0.3.3	183	28		

Sum-Net	10.0.23.0	10.0.3.3	233	28
80000001	1			

AS External Database

Type	LinkState ID	AdvRouter	Age	Len	Sequence
Metric					
External	0.0.0.0	10.0.4.4	43	36	80000001
1					

可以看到，R4的LSDB 中现在多了一条LinkState ID为0.0.0.0的Type-5 LSA（AS External LSA），它表示了一条去往外部网络的缺省路由。因为 Type-5 的泛洪范围是整个 OSPF 网络，这说明通过 default-route-advertise 命令注入缺省路由的方法已经生效了。

查看R1的LSDB。

[R1]display ospf lsdb

OSPF Process 100 with Router ID 10.0.1.1

Link State Database

Area: 0.0.0.2

Type	LinkState ID	AdvRouter	Age	Len	Sequence
Metric					
Router	10.0.3.3	10.0.3.3	1487	36	
80000004	1				
Router	10.0.1.1	10.0.1.1	1495	48	
80000005	1				
Network	10.0.13.3	10.0.3.3	1488	32	
80000002	0				

Sum-Net 80000001	10.0.34.0 1	10.0.3.3	1534	28	
Sum-Net 80000001	172.16.2.1 1	10.0.3.3	1486	28	
Sum-Net 80000001	10.0.23.0 1	10.0.3.3	1525	28	
Sum-Asbr 1	10.0.4.4	10.0.3.3	49	28	80000001

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External 1	0.0.0.0	10.0.4.4	50	36	80000001

观察发现，在R1的LSDB 中现在也存在一条表示缺省路由的Type-5 LSA。测试R1的内部网络与外部网络间的连通性。

```
<R1>ping -a 172.16.1.1 202.68.1.1
```

```
PING 202.68.1.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 202.68.1.1: bytes=56 Sequence=1 ttl=253 time=70 ms
```

```
Reply from 202.68.1.1: bytes=56 Sequence=2 ttl=253 time=50 ms
```

```
Reply from 202.68.1.1: bytes=56 Sequence=3 ttl=253 time=40 ms
```

```
Reply from 202.68.1.1: bytes=56 Sequence=4 ttl=253 time=40 ms
```

```
Reply from 202.68.1.1: bytes=56 Sequence=5 ttl=253 time=30 ms
```

```
--- 202.68.1.1 ping statistics ---
```

5 packet(s) transmitted

5 packet(s) received

0.00% packet loss

round-trip min/avg/max = 30/46/70 ms

可以看到，企业内部已经可以访问外部网络了。

但是，在实际场景中，如果R4与R5之间的链路出现了故障，导致R4的静态缺省路由失效，那么 R4 的路由表中就将失去这条缺省路由，进而导致通过 default-route-advertise发布的缺省路由也随之失效。而当链路恢复正常时，静态缺省路由又会出现现在路由表中，而表示这条缺省路由的Type-5 LSA又会被再次发布到OSPF 网络中。这样，如果这条链路不稳定，就会造成缺省路由和路由表的不稳定。

关闭R5 的GE 0/0/0 接口，以模拟R4 与R5 之间的链路故障。

```
[R5]interface GigabitEthernet 0/0/0
```

```
[R5-GigabitEthernet0/0/0]shutdown
```

查看R1的LSDB。

```
[R1]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.1.1

Link State Database

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 80000004	10.0.3.3 1	10.0.3.3	1630	36	

Router 80000005	10.0.1.1 1	10.0.1.1	1637	48
Network 80000002	10.0.13.3 0	10.0.3.3	1630	32
Sum-Net 80000001	10.0.34.0 1	10.0.3.3	1676	28
Sum-Net 80000001	172.16.2.1 1	10.0.3.3	1628	28
Sum-Net 80000001	10.0.23.0 1	10.0.3.3	1667	28
Sum-Asbr 80000001	10.0.4.4 1	10.0.3.3	191	28

可以看到，表示缺省路由的那条Type-5 LSA已经消失了。

将R5 的GE 0/0/0 接口重新打开。

```
[R5]interface GigabitEthernet 0/0/0
```

```
[R5-GigabitEthernet0/0/0]undo shutdown
```

查看R1的LSDB。

```
[R1]display ospf lsdb
```

```
OSPF Process 100 with Router ID 10.0.1.1
```

```
Link State Database
```

```
Area: 0.0.0.2
```

Type	LinkState ID	AdvRouter	Age	Len	Sequence
Metric					

Router 80000004	10.0.3.3 1	10.0.3.3	1671	36
Router 80000005	10.0.1.1 1	10.0.1.1	1678	48
Network 80000002	10.0.13.3 0	10.0.3.3	1671	32
Sum-Net 80000001	10.0.34.0 1	10.0.3.3	1717	28
Sum-Net 80000001	172.16.2.1 1	10.0.3.3	1669	28
Sum-Net 80000001	10.0.23.0 1	10.0.3.3	1708	28
Sum-Asbr 80000001	10.0.4.4 1	10.0.3.3	232	28

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External 1	0.0.0.0	10.0.4.4	16	36	80000001

观察发现，表示缺省路由的LinkState ID 为0.0.0.0 的Type-5 LSA 现在又出现在了R1的LSDB中。可见，动态注入的缺省路由是会受到链路故障影响的。为了避免链路不稳定所带来的这种影响，提高网络的可靠性，我们希望无论 R4 上是否已经存在缺省路由，R4都能够向整个OSPF网络注入缺省路由。为此，可以在使用default-route-advertise命令时添加关键字always。

```
[R4]ospf 100
```

```
[R4-ospf-100]default-route-advertise always
```

配置完成后，删除静态缺省路由，并在删除之后查看R4的LSDB。

```
[R4]undo ip route-static 0.0.0.0 0.0.0.0 202.103.68.1
```

```
[R4]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.4.4

Link State Database

Area: 0.0.0.0

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	22	36	80000005	1
Router 1	10.0.4.4	10.0.4.4	367	36	80000008	
Network 80000003	10.0.34.4 0	10.0.4.4	12	32		
Sum-Net 80000002	10.0.13.0 1	10.0.3.3	50	28		
Sum-Net 80000002	172.16.2.1 1	10.0.3.3	22	28		
Sum-Net 80000002	172.16.1.1 1	10.0.3.3	11	28		
Sum-Net 80000002	10.0.23.0 1	10.0.3.3	41	28		

AS External Database

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
External	0.0.0.0	10.0.4.4	151	36	80000001 1

可以看到，即使删除了缺省路由之后，LSDB 中依然存在LinkState ID 为0.0.0.0 的Type-5 LSA。也就是说，无论R4 上是否存在缺省路由，R4 都会向OSPF 网络注入缺省路由；无论外部网络拓扑是否发生了变化，缺省路由始终会出现在OSPF网络的各个路由表中。

4．向stub区域或Totally-Stub区域注入缺省路由

该OSPF网络中，区域1本是一个普通的非骨干区域，如果将它配置成Stub区域，由于Stub 区域是不允许Type-5 LSA 进入的，则该区域是不可能获得前面实验中注入的缺省路由的。实际上，在配置OSPF的Stub 区域时，Stub区域的ABR会自动生成表示缺省路由的Type-3 LSA，并将它泛洪进该区域。Totally Stub 区域有着类似的特性；Totally Stub 区域虽然不允许Type-3 LSA 进入，但是允许表示缺省路由的Type-3 LSA 进入。

配置区域1为Stub区域。

```
[R3]ospf 100
```

```
[R3-ospf-100]area 1
```

```
[R3-ospf-100-area-0.0.0.1]stub
```

```
[R2]ospf 100
```

```
[R2-ospf-100]area 1
```

```
[R2-ospf-100-area-0.0.0.1]stub
```

配置完成后，查看R2的LSDB。

```
[R2]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.2.2

Link State Database

Area: 0.0.0.1

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	54	36	80000005	1
Router	10.0.2.2	10.0.2.2	53	48	80000004	1
Network 0	10.0.23.3	10.0.3.3	55	32	80000001	
Sum-Net 80000001	0.0.0.0 1	10.0.3.3	106	28		
Sum-Net 80000001	10.0.34.0 1	10.0.3.3	106	28		
Sum-Net 80000001	10.0.13.0 1	10.0.3.3	106	28		
Sum-Net 80000001	172.16.1.1 1	10.0.3.3	106	28		

可以看到，在R2的LSDB 中没有Type-5 LSA，取而代之的是一条表示缺省路由的LinkState ID 为0.0.0.0 的Type-3 LSA。

将区域1配置成Totally-Stub区域。

```
[R3]ospf 100
```

```
[R3-ospf-100]area 1
```

```
[R3-ospf-100-area-0.0.0.1]stub no-summary
```


配置完成后，查看R2的LSDB。

```
[R2]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.2.2

Link State Database

Area: 0.0.0.1

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence
Router 1	10.0.3.3	10.0.3.3	9	36	80000006
Router 1	10.0.2.2	10.0.2.2	1	48	80000009
Network 80000002	10.0.23.3 0	10.0.3.3	1	32	
Sum-Net 80000001	0.0.0.0 1	10.0.3.3	146	28	

观察发现，表示缺省路由的LinkState ID为0.0.0.0的Type-3 LSA依然存在，但其他的Type-3 LSA都消失了。

5．向NSSA区域或Totally-NSSA区域注入缺省路由

NSSA区域或Totally NSSA区域也是不允许Type-5 LSA进入的。但是，在配置NSSA区域或Totally NSSA区域时，该区域的ABR会自动向该区域注入表示缺省路由的Type-7 LSA。

配置区域2为NSSA区域。

```
[R1]ospf 100
```

```
[R1-ospf-100]area 2
```

```
[R1-ospf-100-area-0.0.0.2]nssa
```

```
[R3]ospf 100
```

```
[R3-ospf-100]area 2
```

```
[R3-ospf-100-area-0.0.0.2]nssa
```

配置完成后，查看R1的LSDB。

```
[R1]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.1.1

Link State Database

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router	10.0.3.3	10.0.3.3	18	36	80000005	1
Router	10.0.1.1	10.0.1.1	17	48	80000004	1
Network 0	10.0.13.3	10.0.3.3	18	32	80000001	
Sum-Net 1	10.0.34.0	10.0.3.3	67	28	80000001	
Sum-Net 80000001	172.16.2.1 1	10.0.3.3	67	28		
Sum-Net 1	10.0.23.0	10.0.3.3	67	28	80000001	
NSSA	0.0.0.0	10.0.3.3	67	36	80000001	1

观察发现，NSSA 区域 2 拒绝了 Type-5 LSA，取而代之的是一条表示缺省路由的Type-7 LSA。

配置区域2 为Totally NSSA区域。

```
[R3]ospf 100
```

```
[R3-ospf-100]area 2
```

```
[R3-ospf-100-area-0.0.0.2]nssa no-summary
```

配置完成后，查看R1的LSDB。

```
[R1]display ospf lsdb
```

OSPF Process 100 with Router ID 10.0.1.1

Link State Database

Area: 0.0.0.2

Type Metric	LinkState ID	AdvRouter	Age	Len	Sequence	
Router 1	10.0.3.3	10.0.3.3	7	36	80000008	
Router	10.0.1.1	10.0.1.1	1	48	8000000D	1
Network 80000002	10.0.13.3 0	10.0.3.3	1	32		
Sum-Net 1	0.0.0.0	10.0.3.3	11	28	80000001	
NSSA	0.0.0.0	10.0.3.3	121	36	80000001	1

可以看到，和 NSSA 区域一样，进入 Totally NSSA 区域的是一条表示缺省路由的Type-7 LSA。

思考

OSPF ASBR 路由器发布的缺省路由能不能被Stub 区域中的路由器接收到？

2.11 OSPF 故障排除

原理概述

OSPF 是一种应用非常广泛的路由协议，掌握 OSPF 协议的故障诊断和排除方法显得尤为重要。

OSPF协议故障问题可以大致分为三类，第一类是涉及OSPF邻居关系的建立问题，第二类是涉及OSPF LSA 的泛洪问题，第三类是涉及OSPF路由的计算问题。第一类问题最为常见，所以也是本次实验所关注的问题。

实验目的

- 发现 and 解决区域号不匹配的问题
- 发现 and 解决Router-ID 冲突的问题
- 发现 and 解决掩码不匹配的问题
- 发现 and 解决Hello Timer时间不匹配的问题
- 发现 and 解决认证类型不匹配的问题

实验内容

实验拓扑如图2-17 所示，实验编址如表 2-11 所示。本实验模拟了一个企业的OSPF网络，其中R1与R2、R1与R3、R2与R4之间的链路属于区域0，R3与R4、R3与R5、R4与R6之间的链路属于区域1。显然，如果R1与R2之间的链路出现故障，则整个骨干区域会被分割，因此需要在

R3与R4之间建立一条虚链路作为区域0的一条备份链路。实验过程中，会人为地制造一些故障点，然后再一步一步地进行故障排除。

实验拓扑

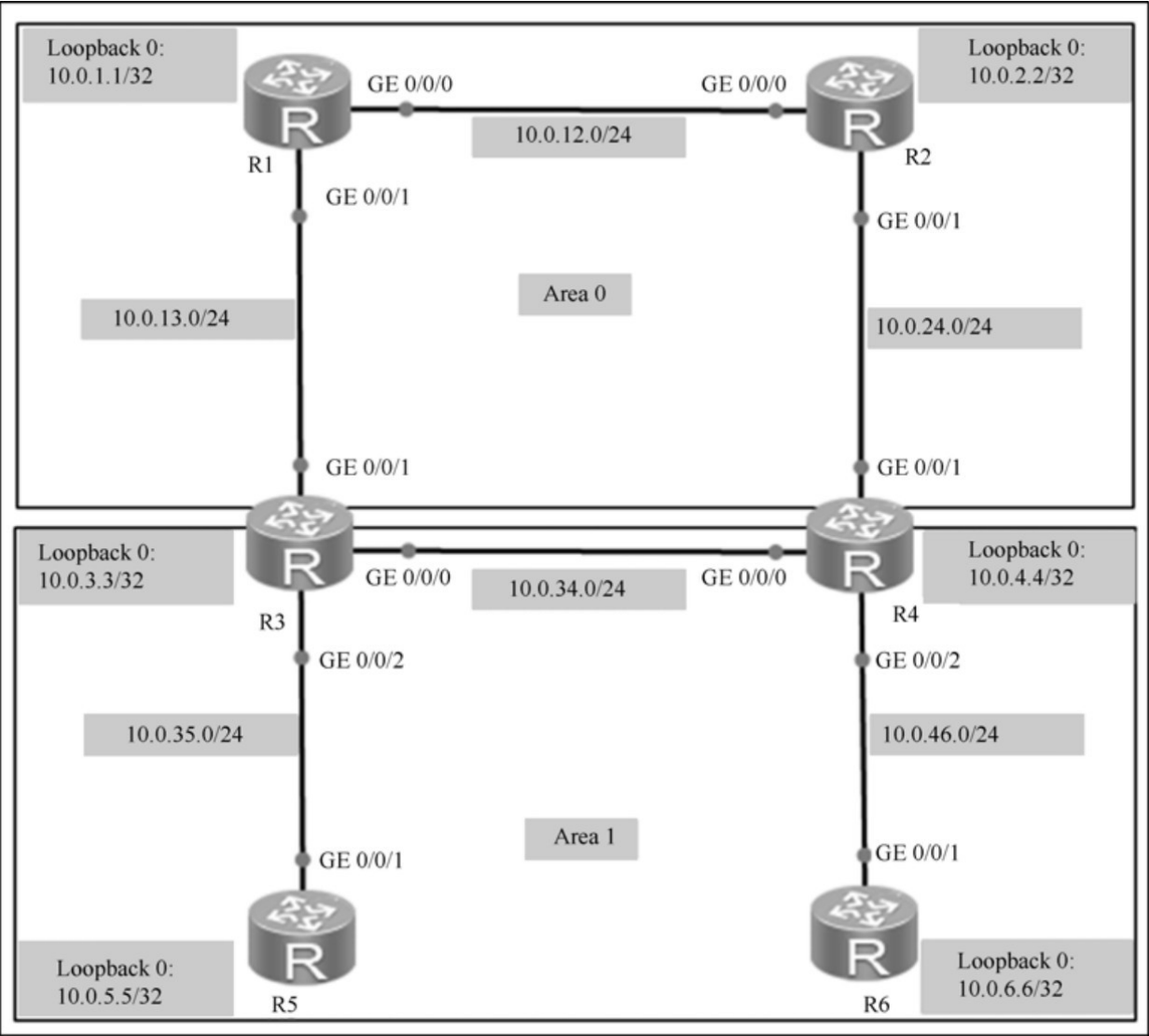


图2-17 OSPF故障排除

实验编址表

表2-11 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.24.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	Loopback 0	10.0.3.3	255.255.255.255	N/A
	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.35.3	255.255.255.0	N/A

(续表)

设备	接口	IP 地址	子网掩码	默认网关
R4(AR2220)	Loopback 0	10.0.4.4	255.255.255.255	N/A
	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	GE 0/0/2	10.0.46.4	255.255.255.0	N/A
R5(AR2220)	GE 0/0/1	10.0.35.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A
R6(AR2220)	GE 0/0/1	10.0.46.6	255.255.255.0	N/A
	Loopback 0	10.0.6.6	255.255.255.255	N/A

实验步骤

1 . 基本配置

根据图2-17和表2-11进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 10.0.13.3
```

```
PING 10.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=10 ms
```

```
--- 10.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 10/10/10 ms

其余直连网段的连通性测试过程在此省略。

2 . 配置OSPF路由协议

配置OSPF协议，其中R1与R2、R1与R3、R2与R4之间的链路属于区域0，R3与R4、R3与R5、R4与R6之间的链路属于区域1，R3与R4配置虚链路，每台路由器使用Loopback 0 接口的IP 地址作为自己的Router-ID。

```
[R1]router id 10.0.1.1
```

```
[R1]ospf 10
```

```
[R1-ospf-10]area 0
```

```
[R1-ospf-10-area-0.0.0.0]network 10.0.12.1 0.0.0.0
```

```
[R1-ospf-10-area-0.0.0.0]network 10.0.13.1 0.0.0.0
```

```
[R1-ospf-10-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R2]router id 10.0.2.2
```

```
[R2]ospf 10
```

```
[R2-ospf-10]area 0
```

```
[R2-ospf-10-area-0.0.0.0]network 10.0.12.2 0.0.0.0
```

```
[R2-ospf-10-area-0.0.0.0]network 10.0.24.2 0.0.0.0
```

```
[R2-ospf-10-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

[R3]router id 10.0.3.3

[R3]ospf 10

[R3-ospf-10]area 0

[R3-ospf-10-area-0.0.0.0]network 10.0.13.3 0.0.0.0

[R3-ospf-10-area-0.0.0.0]network 10.0.3.3 0.0.0.0

[R3-ospf-10-area-0.0.0.0]area 1

[R3-ospf-10-area-0.0.0.1]network 10.0.35.3 0.0.0.0

[R3-ospf-10-area-0.0.0.1]network 10.0.34.3 0.0.0.0

[R3-ospf-10-area-0.0.0.1]vlink-peer 10.0.4.4

[R4]router id 10.0.4.4

[R4]ospf 10

[R4-ospf-10]area 0

[R4-ospf-10-area-0.0.0.0]network 10.0.24.4 0.0.0.0

[R4-ospf-10-area-0.0.0.0]network 10.0.4.4 0.0.0.0

[R4-ospf-10-area-0.0.0.0]area 1

[R4-ospf-10-area-0.0.0.1]network 10.0.46.4 0.0.0.0

[R4-ospf-10-area-0.0.0.1]network 10.0.34.4 0.0.0.0

[R4-ospf-10-area-0.0.0.1]vlink-peer 10.0.3.3

[R5]router id 10.0.5.5

[R5]ospf 10


```
[R5-ospf-10]area 1
```

```
[R5-ospf-10-area-0.0.0.1]network 10.0.35.5 0.0.0.0
```

```
[R5-ospf-10-area-0.0.0.1]network 10.0.5.5 0.0.0.0
```

```
[R6]router id 10.0.6.6
```

```
[R6]ospf 10
```

```
[R6-ospf-10]area 1
```

```
[R6-ospf-10-area-0.0.0.1]network 10.0.46.6 0.0.0.0
```

```
[R6-ospf-10-area-0.0.0.1]network 10.0.6.6 0.0.0.0
```

配置完成后，在R2、R3、R4上查看OSPF邻居建立情况。

```
<R2>display ospf peer brief
```

OSPF Process 10 with Router ID 10.0.2.2

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	10.0.1.1	Full
0.0.0.0	GigabitEthernet0/0/1	10.0.4.4	Full


```
<R3>display ospf peer brief
```

OSPF Process 10 with Router ID 10.0.3.3

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	10.0.1.1	Full
0.0.0.1	GigabitEthernet0/0/0	10.0.4.4	Full
0.0.0.1	GigabitEthernet0/0/2	10.0.5.5	Full

<R4>display ospf peer brief

OSPF Process 10 with Router ID 10.0.4.4

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	10.0.2.2	Full
0.0.0.1	GigabitEthernet0/0/0	10.0.3.3	Full
0.0.0.1	GigabitEthernet0/0/2	10.0.6.6	Full

可以看到，邻居邻接关系都已成功建立。

在R5上查看OSPF路由信息。

<R5>display ip routing-table protocol ospf

Route Flags: R - relay, D - download to fib

Public routing table : OSPF

Destinations : 10 Routes : 10

OSPF routing table status :<Active>

Destinations : 10 Routes : 10

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.35.3
10.0.2.2/32 GigabitEthernet0/0/1	OSPF	10	3	D	10.0.35.3
10.0.3.3/32 GigabitEthernet0/0/1	OSPF	10	1	D	10.0.35.3
10.0.4.4/32 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.35.3
10.0.6.6/32 GigabitEthernet0/0/1	OSPF	10	3	D	10.0.35.3
10.0.12.0/24 GigabitEthernet0/0/1	OSPF	10	3	D	10.0.35.3
10.0.13.0/24 GigabitEthernet0/0/1	OSPF	10	2	D	10.0.35.3

```
10.0.24.0/24    OSPF    10   3    D    10.0.35.3
GigabitEthernet0/0/1

10.0.34.0/24    OSPF    10   2    D    10.0.35.3
GigabitEthernet0/0/1

10.0.46.0/24    OSPF    10   3    D    10.0.35.3
GigabitEthernet0/0/1
```

OSPF routing table status :<Inactive>

Destinations : 0 Routes : 0

可以看到，R5已经获得了所有网段的路由。

在R6上查看OSPF路由信息。

<R6>display ip routing-table protocol ospf

Route Flags: R - relay, D - download to fib

Public routing table :OSPF

Destinations : 10 Routes : 10

OSPF routing table status :<Active>

Destinations : 10 Routes : 10

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/1	OSPF	10	3	D	10.0.46.4

10.0.2.2/32	OSPF	10	2	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.3.3/32	OSPF	10	2	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.4.4/32	OSPF	10	1	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.5.5/32	OSPF	10	3	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.12.0/24	OSPF	10	3	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.13.0/24	OSPF	10	3	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.24.0/24	OSPF	10	2	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.34.0/24	OSPF	10	2	D	10.0.46.4
GigabitEthernet0/0/1					
10.0.35.0/24	OSPF	10	3	D	10.0.46.4
GigabitEthernet0/0/1					

OSPF routing table status :<Inactive>

Destinations : 0 Routes : 0

可以看到，R6也已获得了所有网段的路由。

查看R5和R6的LSDB。

<R5>display ospf lsdb

OSPF Process 10 with Router ID 10.0.5.5

Link State Database

Area: 0.0.0.1

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.5.5	10.0.5.5	67	48	80000005	0
Router	10.0.3.3	10.0.3.3	885	48	8000000B	1
Router	10.0.4.4	10.0.4.4	886	48	80000009	1
Router	10.0.6.6	10.0.6.6	1767	48	80000004	1
Network	10.0.46.6	10.0.6.6	1766	32	80000001	0
Network	10.0.35.5	10.0.5.5	105	32	80000002	0
Network	10.0.34.4	10.0.4.4	1212	32	80000001	0
Sum-Net	10.0.13.0	10.0.3.3	227	28	80000002	1
Sum-Net	10.0.24.0	10.0.4.4	321	28	80000002	1
Sum-Net	10.0.12.0	10.0.3.3	227	28	80000002	2
Sum-Net	10.0.12.0	10.0.4.4	321	28	80000002	2
Sum-Net	10.0.3.3	10.0.3.3	227	28	80000002	0
Sum-Net	10.0.2.2	10.0.4.4	321	28	80000002	1
Sum-Net	10.0.1.1	10.0.3.3	227	28	80000002	1
Sum-Net	10.0.4.4	10.0.4.4	321	28	80000002	0

<R6>display ospf lsdb

OSPF Process 10 with Router ID 10.0.6.6

Link State Database

Area: 0.0.0.1

Type	LinkState ID	AdvRouter	Age	Len	Sequence	Metric
Router	10.0.5.5	10.0.5.5	97	48	80000005	0
Router	10.0.3.3	10.0.3.3	913	48	8000000B	1
Router	10.0.4.4	10.0.4.4	912	48	80000009	1
Router	10.0.6.6	10.0.6.6	1790	48	80000004	1
Network	10.0.46.6	10.0.6.6	1791	32	80000001	0
Network	10.0.35.5	10.0.5.5	133	32	80000002	0
Network	10.0.34.4	10.0.4.4	1240	32	80000001	0
Sum-Net	10.0.13.0	10.0.3.3	255	28	80000002	1
Sum-Net	10.0.24.0	10.0.4.4	347	28	80000002	1
Sum-Net	10.0.12.0	10.0.4.4	347	28	80000002	2
Sum-Net	10.0.12.0	10.0.3.3	255	28	80000002	2
Sum-Net	10.0.3.3	10.0.3.3	255	28	80000002	0
Sum-Net	10.0.2.2	10.0.4.4	347	28	80000002	1
Sum-Net	10.0.1.1	10.0.3.3	255	28	80000002	1
Sum-Net	10.0.4.4	10.0.4.4	347	28	80000002	0

可以看到，R5和R6的LSDB的内容是完全相同的。

3．添加OSPF故障点

目前，企业网络的配置是正确的，通信也是正常的。为了模拟故障现象，接下来将人为地制造一些故障点。

故障点1：将路由器R5 的GE 0/0/1 接口通告进区域2。

```
[R5]ospf 10
```

```
[R5-ospf-10]area 2
```

```
[R5-ospf-10-area-0.0.0.2]network 10.0.35.5 0.0.0.0
```

说明：OSPF 协议规定，不同区域（Area）之间的边界是路由器而不是链路。如果R3 的GE 0/0/2 接口属于区域1，而R5的GE 0/0/1接口属于区域2，就意味着R3与R5之间的链路成了区域1与区域2的边界，这违背了OSPF关于区域划分的原则，也将导致R3与R5的邻居关系无法建立。

故障点2：将路由器R2 的Router-ID 修改为10.0.1.1，并重启OSPF 进程。

```
[R2]router id 10.0.1.1
```

```
[R2]quit
```

```
<R2>reset Ospf process
```

```
Warning: The OSPF process will be reset. Continue? [Y/N]:y
```

说明：Router-ID唯一地标识了OSPF路由器的身份，OSPF路由器之间通过交换Hello报文来协商成为邻居，而Hello报文中就包含Router-ID等信息。如果R2与R1的Router-ID发生了冲突，则它们的邻居关系是无法建立的。

故障点3：将R2 的GE 0/0/0 接口的IP 地址掩码改成255.255.255.128。

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]ip address 10.0.12.2 25
```


说明：将R2的GE 0/0/0接口的IP 地址掩码改成255.255.255.128 后，马上会有日志信息弹出，OSPF邻居状态变成Down。原来，Hello报文中携带了掩码信息，如果链路两端接口的掩码不匹配，则邻居关系无法建立。

故障点4：将R1 的GE 0/0/1 接口的Hello Timer的时间修改成100s。

```
[R1]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]ospf timer hello 100
```

说明：OSPF 协议依靠Hello 报文来建立和维护邻居关系，缺省情况下，Hello Timer的时间为10s，该信息也包含在Hello报文中。如果相邻路由器的Hello报文中的Hello Timer的时间不一致，则将导致双方无法建立邻居关系。

故障点5：将R4的GE 0/0/2 接口配置为简单明文认证类型，R6 的GE 0/0/1 接口配置为MD5加密认证类型。

```
[R4]interface GigabitEthernet 0/0/2
```

```
[R4-GigabitEthernet0/0/2]ospf authentication-mode simple
```

```
[R4-GigabitEthernet0/0/2]ospf authentication simple huawei
```

```
[R6]interface GigabitEthernet 0/0/1
```

```
[R6-GigabitEthernet0/0/1]ospf authentication-mode md5
```

```
[R6-GigabitEthernet0/0/1]ospf authentication md5 1 huawei
```

说明：在OSPF协商建立邻居关系的过程中，双方认证类型的不匹配（例如：一方使用简单明文认证，而另一方使用MD5加密认证）将导致邻居关系无法建立。

故障点6：配置R3与R4之间的虚链路时，使用不一致的认证类型。

```
[R3]ospf 10
```

```
[R3-ospf-10]area 1
```

```
[R3-ospf-10-area-0.0.0.1]vlink-peer 10.0.4.4 simple huawei hello 10 dead  
40
```

```
[R4]ospf 10
```

```
[R4-ospf-10]area 1
```

```
[R4-ospf-10-area-0.0.0.1]vlink-peer 10.0.3.3 md5 1 huawei hello 10 dead 40
```

说明：在虚链路上也会交换 Hello 报文，但与普通的 Hello 报文不一样，虚链路的Hello报文是单播报文而并非是组播报文。如果需要建立虚链路的双方所发送的Hello报文中的认证类型不匹配，则双方不能建立虚链路关系。

4．排除OSPF的邻居关系故障

现在，由于上述故障的存在，网络已无法进行正常的通信了。在开始排除故障之前，请读者先参照图2-12熟悉一下每台路由器有哪几个邻居。

在R5 上使用display ospf peer 命令查看邻居关系。

```
[R5]display ospf peer
```

```
OSPF Process 10 with Router ID 10.0.5.5
```

结果发现R5没有任何邻居。使用reset ospf counter 命令清空OSPF 计数器，然后使用display ospf error命令查看错误报文信息。

```
<R5>reset ospf counter
```

```
<R5>display ospf error
```

```
OSPF Process 10 with Router ID 10.0.5.5
```

OSPF error statistics

General packet errors:

0	:	IP: received my own packet	0	:	Bad packet
0	:	Bad version	0	:	Bad checksum
254	:	Bad area id	0	:	Drop on unnumbered interface
0	:	Bad virtual link	0	:	Bad authentication type
.....					

可以看到，在Bad area id 处显示有错误数据包计数。查看R5 的OSPF 配置情况。

```
[R5]ospf 10
```

```
[R5-ospf-10]display this
```

```
[V200R003C00]
```

```
#
```

```
ospf 10
```

```
area 0.0.0.1
```

```
network 10.0.5.5 0.0.0.0
```

```
area 0.0.0.2
```

```
network 10.0.35.5 0.0.0.0
```

```
#
```

```
Return
```

可以看到，R5 的OSPF 区域配置有误，R5 的GE 0/0/1 接口（10.0.35.5）本应该属于区域1，而现在是属于区域2。

修改R5 的GE 0/0/1 接口的区域配置。

```
[R5]ospf 10
```

```
[R5-ospf-10]area 1
```

```
[R5-ospf-10-area-0.0.0.1]network 10.0.35.5 0.0.0.0
```

修改之后，在R5上查看邻居关系是否能正常建立。

```
<R5>display ospf peer
```

```
OSPF Process 10 with Router ID 10.0.5.5
```

```
Neighbors
```

```
Area 0.0.0.1 interface 10.0.35.5(GigabitEthernet0/0/1)'s neighbors
```

```
Router ID: 10.0.3.3      Address: 10.0.35.3
```

```
State: Full Mode:Nbr is Slave Priority: 1
```

```
DR: 10.0.35.5 BDR: 10.0.35.3 MTU: 0
```

```
Dead timer due in 31 sec
```

```
Retrans timer interval: 5
```

```
Neighbor is up for 00:02:11
```

```
Authentication Sequence: [ 0 ]
```

可以看到，现在邻居关系已经正常，R5只有一个邻居R3，这与实际情况是相符的。

查看R3的邻居关系。

```
<R3>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.3.3
```

```
Peer Statistic Information
```


Area Id	Interface	Neighbor id	State
0.0.0.1	GigabitEthernet0/0/0	10.0.4.4	Full
0.0.0.1	GigabitEthernet0/0/2	10.0.5.5	Full

观察发现，R3上已建立了两个OSPF邻居关系，而实际上R3应该有3个邻居，目前R3与R1还没有建立起邻居关系。

查看R3接收到的错误数据包。

```
<R3>display ospf error
```

```
OSPF Process 10 with Router ID 10.0.3.3
```

```
OSPF error statistics
```

```
.....
```

```
HELLO packet errors:
```

0	: Netmask mismatch	23	: Hello timer
mismatch			
0	: Dead timer mismatch	0	: Virtual neighbor
unknown			

0 : NBMA neighbor unknown 0 : Invalid Source Address

.....

可以看到，在Hello timer mismatch 处有错误数据包计数，说明Hello Timer时间不匹配。

查看R1 的GE 0/0/1 接口以及R3 的GE 0/0/1 接口的OSPF 状态。

<R1>display ospf interface GigabitEthernet 0/0/1

OSPF Process 10 with Router ID 10.0.1.1

Interfaces

Interface: 10.0.13.1 (GigabitEthernet0/0/1)

Cost: 1 State:DR Type:Broadcast MTU: 1500

Priority: 1

Designated Router: 10.0.13.1

Backup Designated Router: 0.0.0.0

Timers: Hello 100 , Dead 400 , Poll 120 , Retransmit 5 , Transmit Delay 1

<R3>display ospf interface GigabitEthernet 0/0/1

OSPF Process 10 with Router ID 10.0.3.3

Interfaces

Interface: 10.0.13.3 (GigabitEthernet0/0/1)

Cost: 1 State:DR Type:Broadcast MTU: 1500

Priority: 1

Designated Router: 10.0.13.3

Backup Designated Router: 0.0.0.0

Timers: Hello 10 , Dead 40 , Poll 120 , Retransmit 5 , Transmit Delay 1

可以看到，这两个接口上配置的Hello Timer时间不匹配。修改R1的GE 0/0/1 接口的Hello Timer的值为10s，稍等片刻后，查看R1 的邻居关系。

```
[R1]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]ospf timer hello 10
```

```
<R1>display ospf peer brief
```

OSPF Process 10 with Router ID 10.0.1.1

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	10.0.3.3	Full

可以看到，现在R1与R3已经建立了正常的邻居关系，但是却发现，R1还没有与R2建立邻居关系。

在R2上查看是否接收到错误数据包。

```
<R2>display ospf error
```

OSPF Process 10 with Router ID 10.0.1.1

OSPF error statistics

General packet errors:

0	:	IP: received my own packet	0	:	Bad packet
0	:	Bad version	0	:	Bad checksum
0	:	Bad area id	0	:	Drop on unnumbered interface
0	:	Bad virtual link	0	:	Bad authentication type
0	:	Bad authentication key	0	:	Packet too small
0	:	Packet size> ip length	0	:	Transmit error
1	:	Interface down	0	:	Unknown neighbor
0	:	Bad net segment	0	:	Extern option mismatch
40	:	Router id confusion			
.....					

可以看到，在Router id confusion 处有错误数据包计数，显示Router-ID 有错误。

使用display ospf routing router-id 命令查看R1 和R2 的Router-ID。

```
<R1>display ospf routing router-id
```

OSPF Process 10 with Router ID 10.0.1.1

Router Type URT Routing Tables

RtType	Destination	Area	Cost	NextHop	Type
Intra-area	10.0.3.3	0.0.0.0	1	10.0.13.3	ABR


```
<R2>display ospf routing router-id
```

```
OSPF Process 10 with Router ID 10.0.1.1
```

```
Router Type URT Routing Tables
```

RtType	Destination	Area	Cost	Nexthop	Type
Intra-area	10.0.4.4	0.0.0.0	1	10.0.24.4	ABR

可以看到，R2与R1具有相同的Router-ID，发生了冲突。修改R2的Router-ID，并重启R2上的OSPF进程，然后查看R2上的邻居情况。

```
[R2]router id 10.0.2.2
```

```
[R2]quit
```

```
<R2>reset ospf process
```

```
Warning: The OSPF process will be reset. Continue? [Y/N]:y
```

```
<R2>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.2.2
```

```
Peer Statistic Information
```

```
-----  
-----  


| Area Id | Interface            | Neighbor id | State |
|---------|----------------------|-------------|-------|
| 0.0.0.0 | GigabitEthernet0/0/1 | 10.0.4.4    | Full  |

  
-----  
-----
```

可以看到，R2的Router-ID修改正确之后，R2与R1仍然没有建立起邻居关系。再次查看R2是否接收到错误数据包。

<R2>display ospf error

OSPF Process 10 with Router ID 10.0.2.2

OSPF error statistics

.....

HELLO packet errors:

18	: Netmask mismatch	0	: Hello timer mismatch
0	: Dead timer mismatch	0	: Virtual neighbor unknown
0	: NBMA neighbor unknown	0	: Invalid Source Address

.....

可以看到，在Netmask mismatch 处有错误数据包计数。查看R2 的接口信息。

<R2>display ip interface brief

*down: administratively down

.....

Interface	IP Address/Mask	Physical	Protocol
GigabitEthernet0/0/0	10.0.12.2/25	up	up
GigabitEthernet0/0/1	10.0.24.2/24	up	up
GigabitEthernet0/0/2	unassigned	down	down
LoopBack0	10.0.2.2/32	up	up(s)

NULL0 unassigned up up(s)

可以看到，R2 的GE 0/0/0 接口的子网掩码是255.255.255.128，导致了不能与R1 建立邻居关系。

修改R2 的GE 0/0/0 接口的子网掩码，使链路两端接口的掩码一致。

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]ip add 10.0.12.2 24
```

修改掩码后，可以看到如下的日志提示信息。

```
[R2-GigabitEthernet0/0/0]
```

```
Jul 31 2013 11:24:28-05:13 R2 %%01OSPF/4/NBR_CHANGE_E(l)
[8]:Neighbor changes event: neighbor status changed. (ProcessId=2560,
NeighborAddress=1.12.0.10, NeighborEvent=HelloReceived,
NeighborPreviousState=Down, NeighborCurrentState=Init)
```

```
[R2-GigabitEthernet0/0/0]
```

```
Jul 31 2013 11:24:28-05:13 R2 %%01OSPF/4/NBR_CHANGE_E(l)
[9]:Neighbor changes event: neighbor status changed. (ProcessId=2560,
NeighborAddress=1.12.0.10, NeighborEvent=2WayReceived,
NeighborPreviousState=Init, NeighborCurrentState=2Way)
```

```
[R2-GigabitEthernet0/0/0]
```

```
Jul 31 2013 11:24:28-05:13 R2 %%01OSPF/4/NBR_CHANGE_E(l)
[10]:Neighbor changes event: neighbor status changed. (ProcessId=2560,
NeighborAddress=1.12.0.10, NeighborEvent=AdjOk?,
NeighborPreviousState=2Way, NeighborCurrentState=ExStart)
```

```
[R2-GigabitEthernet0/0/0]
```

```
Jul 31 2013 11:24:28-05:13 R2 %%01OSPF/4/NBR_CHANGE_E(l)
[11]:Neighbor changes event: neighbor status changed. (ProcessId=2560,
```

NeighborAddress=1.12.0.10, NeighborEvent=NegotiationDone,
NeighborPreviousState=ExStart, NeighborCurrentState=Exchange)

[R2-GigabitEthernet0/0/0]

Jul 31 2013 11:24:33-05:13 R2 %%01OSPF/4/NBR_CHANGE_E(l)
[12]:Neighbor changes event: neighbor status changed. (ProcessId=2560,
NeighborAddress=1.12.0.10, NeighborEvent=ExchangeDone,
NeighborPreviousState=Exchange, NeighborCurrentState>Loading)

[R2-GigabitEthernet0/0/0]

Jul 31 2013 11:24:33-05:13 R2 %%01OSPF/4/NBR_CHANGE_E(l)
[13]:Neighbor changes event: neighbor status changed. (ProcessId=2560,
NeighborAddress=1.12.0.10, NeighborEvent>LoadingDone,
NeighborPreviousState>Loading, NeighborCurrentState=Full)

上面的信息说明了现在R2与R1已经建立起了邻居关系。

在R4上查看邻居情况。

<R4>display ospf peer brief

OSPF Process 10 with Router ID 10.0.4.4

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	10.0.2.2	Full
0.0.0.1	GigabitEthernet0/0/0	10.0.3.3	Full

可以看到，R4与R6还没有建立起邻居关系。查看R4是否接收到错误数据包。

```
<R4>display ospf error
```

```
OSPF Process 10 with Router ID 10.0.4.4
```

```
OSPF error statistics
```

```
General packet errors:
```

```
0      :  IP: received my own packet      983      : Bad packet
0      : Bad version                      0          : Bad checksum
0      : Bad area id                      0          : Drop on unnumbered
interface
0      : Bad virtual link                  983          : Bad authentication type
0      : Bad authentication key            0            : Packet too small
.....
```

可以看到，在Bad authentication type 处有错误数据包计数。

在R4 上查看GE 0/0/2 接口下的认证配置。

```
<R4>display current-configuration interface GigabitEthernet 0/0/2
```

```
interface GigabitEthernet0/0/2
```

```
ip address 10.0.46.4 255.255.255.0
```

```
ospf authentication-mode simple cipher
%%$%$9T|q3n=ZTV|afLWCm5P3UXE&%%$%$
```

在R6 上查看GE 0/0/1 接口下的认证配置。

```
<R6>display current-configuration interface GigabitEthernet 0/0/1
```

```
interface GigabitEthernet0/0/1
```

```
ip address 10.0.46.6 255.255.255.0
```

```
ospf authentication-mode md5 1 cipher %$%$cw(YOf; ~  
R$B2@7'5KCXMUZ4T%$%$
```

可以看到，R4 的GE 0/0/2 接口与R6的GE 0/0/1接口的认证类型配置不一致，导致无法建立邻居关系。修改R4 的GE 0/0/2 接口的认证类型。

```
[R4]interface GigabitEthernet 0/0/2
```

```
[R4-GigabitEthernet0/0/2]ospf authentication-mode md5
```

```
[R4-GigabitEthernet0/0/2]ospf authentication md5 1 huawei
```

在R4上查看邻居关系。

```
<R4>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.4.4
```

```
Peer Statistic Information
```


Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	10.0.2.2	Full
0.0.0.1	GigabitEthernet0/0/0	10.0.3.3	Full
0.0.0.1	GigabitEthernet0/0/2	10.0.6.6	Full

可以看到，R4与相邻的路由器都建立起了邻居关系。在R6上查看邻居建立情况。

```
<R6>display ospf peer brief
```

```
OSPF Process 10 with Router ID 10.0.6.6
```

```
Peer Statistic Information
```


Area Id	Interface	Neighbor id	State
0.0.0.1	GigabitEthernet0/0/1	10.0.4.4	Full

可以看到，R6的OSPF邻居关系是正常的。

测试R5 与R6 的Loopback 0 接口间的连通性。

```
<R5>ping -a 10.0.5.5 -c 1 10.0.6.6
```

```
PING 10.0.6.6: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.6.6: bytes=56 Sequence=1 ttl=253 time=30 ms
```

```
--- 10.0.6.6 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 30/30/30 ms
```

至此，故障点1至故障点5都已经得到了排除。

5．排除OSPF的虚链路故障

在R3 和R4 上使用display ospf vlink命令查看虚链路的建立情况。

```
<R3>display ospf vlink
```

OSPF Process 10 with Router ID 10.0.3.3

Virtual Links

Virtual-link Neighbor-id -> 10.0.4.4, Neighbor-State: Down

Interface: 10.0.34.3 (GigabitEthernet0/0/0)

Cost: 1 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

```
<R4>display ospf vlink
```

OSPF Process 10 with Router ID 10.0.4.4

Virtual Links

Virtual-link Neighbor-id -> 10.0.3.3, Neighbor-State: Down

Interface: 10.0.34.4 (GigabitEthernet0/0/0)

Cost: 1 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

可以看到，R3和R4之间的虚链路状态出现了问题，状态为Down。

在R3和R4上查看错误数据包信息。

<R3>display ospf error

OSPF Process 10 with Router ID 10.0.3.3

OSPF error statistics

General packet errors:

0	:	IP: received my own packet	608	:	Bad packet
0	:	Bad version	0	:	Bad checksum
312	:	Bad area id	0	:	Drop on unnumbered interface
1	:	Bad virtual link type	580	:	Bad authentication
0	:	Bad authentication key	0	:	Packet too small
.....					

<R4>display ospf error

OSPF Process 10 with Router ID 10.0.4.4

OSPF error statistics

General packet errors:

0	:	IP: received my own packet	1204	:	Bad packet
0	:	Bad version	0	:	Bad checksum

0	: Bad area id	0	: Drop on unnumbered interface
0	: Bad virtual link type	1201	: Bad authentication
3	: Bad authentication key	0	: Packet too small
.....			

可以看到，在Bad authentication type处有错误数据包计数。查看虚链路的配置以核实是否真的存在错误。

```
[R3]ospf 10
```

```
[R3-ospf-10]display this
```

```
[V200R003C00]
```

```
#
```

```
ospf 10
```

```
area 0.0.0.0
```

```
network 10.0.3.3 0.0.0.0
```

```
network 10.0.13.3 0.0.0.0
```

```
area 0.0.0.1
```

```
network 10.0.34.3 0.0.0.0
```

```
network 10.0.35.3 0.0.0.0
```

```
vlink-peer 10.0.4.4 simple huawei hello 10 dead 40
```

```
#
```

```
return
```

```
[R4]ospf 10
```

```
[R4-ospf-10]display this
```

```
[V200R003C00]
```

```
#
```

```
ospf 10
```

```
area 0.0.0.0
```

```
network 10.0.4.4 0.0.0.0
```

```
network 10.0.24.4 0.0.0.0
```

```
area 0.0.0.1
```

```
network 10.0.34.4 0.0.0.0
```

```
network 10.0.46.4 0.0.0.0
```

```
vlink-peer 10.0.3.3 md5 1 huawei hello 10 dead 40
```

```
#
```

```
return
```

可以看到，虚链路上配置的认证类型不匹配。修改R4的GE 0/0/0 接口的认证类型。

```
[R4]ospf 10
```

```
[R4-ospf-10]area 1
```

```
[R4-ospf-10-area-0.0.0.1]vlink-peer 10.0.3.3 simple huawei hello 10 dead 40
```

然后，在R4上查看虚链路的状态。

```
<R4>display ospf vlink
```

OSPF Process 10 with Router ID 10.0.4.4

Virtual Links

Virtual-link Neighbor-id -> 10.0.3.3, Neighbor-State: Full

Interface: 10.0.34.4 (GigabitEthernet0/0/0)

Cost: 1 State: P-2-P Type: Virtual

Transit Area: 0.0.0.1

Timers: Hello 10 , Dead 40 , Retransmit 5 , Transmit Delay 1

GR State: Normal

可以看到，R4与R3之间的虚链路邻居状态为Full，说明虚链路的邻居关系已经正常。至此，所有的故障点都已得到了排除。

思考

在一个OSPF网络中，如果两台路由器的Router-ID相同，但它们是不同区域（Area）中的内部路由器，那么这种情况会对OSPF协议的运行造成什么影响呢？如果它们是同一区域中的内部路由器，情况又会如何？

第3章 BGP

- 3.1 BGP邻居
- 3.2 BGP认证功能
- 3.3 BGP自动路由聚合
- 3.4 BGP手动路由聚合
- 3.5 BGP路径选择——Preferred Value
- 3.6 BGP路径选择——Local Preference
- 3.7 BGP路径选择——Next Hop
- 3.8 BGP路径选择——AS_Path
- 3.9 BGP路径选择——MED
- 3.10 BGP路径选择——Community
- 3.11 BGP路由反射器
- 3.12 BGP路由黑洞
- 3.13 BGP联盟
- 3.14 BGP路由过滤
- 3.15 BGP路由引入
- 3.16 BGP缺省路由
- 3.17 BGP路由衰减
- 3.18 BGP监测和调试
- 3.19 BGP故障排除

3.1 BGP 邻居

原理概述

路由协议通常分为内部网关协议（IGP: Interior Gateway Protocol）和外部网关协议（EGP: Exterior Gateway Protocol）两大类。一般来讲，IGP用于自治系统AS（Autonomous System）内部，EGP用于AS之间。最早的IGP是一种称为GGP（Gateway-to-Gateway Protocol）的路由协议，而最早的EGP是一种称为EGP（Interior Gateway Protocol，注意，它与类别名EGP同名，现已被废除）的路由协议。目前，常见的IGP包括RIP、OSPF、IS-IS等，而常见的EGP只有BGP（Border Gateway Protocol）。

早期发布的BGP 3个版本分别是BGP-1、BGP-2和BGP-3，这3个版本目前已停止使用，当前使用的版本是BGP-4（RFC4271）。BGP-4作为事实上的互联网外部路由协议标准，已被广泛应用于ISP（Internet Service Provider）之间。

BGP虽然是一种动态路由协议，但它实际上本身并不产生路由、不发现路由、不计算路由，其主要功能是完成最佳路由的选择并在BGP邻居之间进行最佳路由的传递。BGP选择了TCP作为其传输协议，端口号为179。

BGP支持无类域间路由CIDR（Classless Inter-Domain Routing），并且采用了触发增量更新方式，这大大地减少了BGP在传播路由信息时所占用的带宽，特别适用于在互联网上传播大量的路由信息。

BGP提供了丰富的路由属性（Attribute），通过对这些属性的操作和控制，BGP能够非常容易地实现丰富而灵活的路由策略。BGP还具有良好的扩展性，支持Multicast、VPN、IPv6等多种特性。

BGP 的邻居关系分为IBGP（ Internal BGP ）和EBGP（ External BGP ）两种：当两台BGP路由器位于同一AS时（ AS编号相同 ），它们的邻居关系为IBGP邻居关系；当两台BGP路由器位于不同的AS时（ AS编号不同 ），它们的邻居关系为EBGP邻居关系。BGP没有自动建立邻居关系的能力，邻居关系必须通过手动配置来建立。

实验目的

- 理解BGP 协议的应用场景
- 理解IBGP 与EBGP 邻居的概念
- 配置IBGP 与EBGP 邻居关系

实验内容

实验拓扑如图3-1所示，实验编址如表 3-1所示。R1与 R2属于同一个运营商网络，AS 编号为 100，R1 与 R2 之间的邻居关系为 IBGP 邻居关系。R3 属于另一个运营商网络，AS编号为200，R3与R2之间的邻居关系为EBGP邻居关系。本实验中，路由器将分别采用物理接口和Loopback接口来进行IBGP和EBGP邻居关系的建立。

实验拓扑

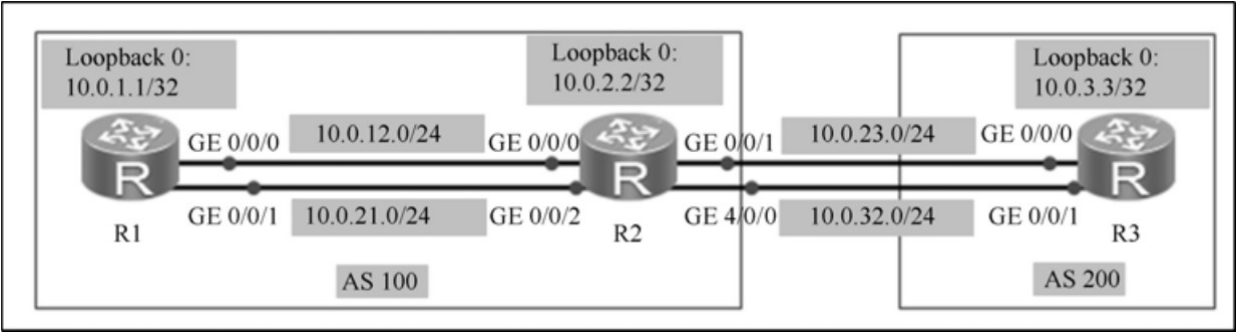


图3-1 BGP邻居

实验编址表

表3-1 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2200)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.21.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2200)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.23.2	255.255.255.0	N/A
	GE 0/0/2	10.0.21.2	255.255.255.0	N/A
	GE 4/0/0	10.0.32.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2200)	GE 0/0/0	10.0.23.3	255.255.255.0	N/A
	GE 0/0/1	10.0.32.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-1和表3-1进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=530 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 530/530/530 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置IBGP邻居

接下来将在R1和R2上使用直连物理接口来配置IBGP邻居关系。为了实现链路冗余，R1与R2之间部署了两条链路，当其中一条物理链路出现故障时，另一条物理链路可以提供连通性。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```



```
[R1-bgp]peer 10.0.12.2 as-number 100
```

```
[R1-bgp]peer 10.0.21.2 as-number 100
```

```
[R2]bgp 100
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.21.1 as-number 100
```

上述配置完成后，在R2上使用display bgp peer 命令，查看BGP邻居关系。

```
[R2]display bgp peer
```

```
BGP local router ID : 10.0.2.2
```

```
Local AS number : 100
```

```
Total number of peers : 2      Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.12.1	4	100	5	5	0	00:03:23
Established	0					
10.0.21.1	4	100	5	5	0	00:03:12
Established	0					

可以看到，R2 现在有两个BGP 邻居，分别使用了R1的GE 0/0/0和GE 0/0/1 接口地址来表示，AS编号为100，与R2自己的AS编号相同，因此R2与R1为IBGP邻居。当前邻居状态为Established，表示邻居关系已完全建立。

在R1 上将Loopback 0 接口地址通告到BGP 进程中。

```
[R1]bgp 100
```

```
[R1-bgp]network 10.0.1.1 32
```

上述配置完成后，在R2上查看BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.12.1	0	100	0	i
* i		10.0.21.1	0	100	0	i

可以看到，R2的BGP路由表中包含了两条去往10.0.1.1/32的路由，下一跳分别为10.0.12.1和10.0.21.1，这是因为R1与R2之间建立了两个IBGP邻居关系，BGP路由实现了冗余。

BGP 是运行在TCP 之上的，如果能让R1的Loopback 0 接口与R2的Loopback 0 接口建立起TCP 会话，并使用Loopback 0 接口的IP地址来建立BGP 邻居关系，则可以让R1和R2只维护一个BGP邻居关系即可。当R1与R2之间的一条链路出现故障时，TCP可以通过另外一条物理链路继续维持会话关系，这种方法在网络的稳定性方面和网络资源的节省上比直接使用物理接口来建立BGP邻居关系更具优势。

为了能使R1的Loopback 0 接口与R2的Loopback 0 接口建立起TCP 会话，需要在R1 和R2 上配置到达对方Loopback 0 接口的静态路由。

```
[R1]ip route-static 10.0.2.2 32 10.0.12.2
```

```
[R1]ip route-static 10.0.2.2 32 10.0.21.2
```

```
[R2]ip route-static 10.0.1.1 32 10.0.12.1
```

```
[R2]ip route-static 10.0.1.1 32 10.0.21.1
```

删除之前采用物理接口配置 IBGP 邻居的命令，并使用 Loopback 0 接口重新建立IBGP邻居关系。

```
[R1]bgp 100
```

```
[R1-bgp]undo peer 10.0.12.2
```

```
[R1-bgp]undo peer 10.0.21.2
[R1-bgp]peer 10.0.2.2 as-number 100
[R2]bgp 100
[R2-bgp]undo peer 10.0.12.1
[R2-bgp]undo peer 10.0.21.1
[R2-bgp]peer 10.0.1.1 as-number 100
```

上述配置完成后，在R1上查看BGP邻居关系。

```
[R1]display bgp peer
```

BGP local router ID : 10.0.1.1

Local AS number : 100

Total number of peers : 1 Peers in established state : 0

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State

PrefRcv

10.0.2.2	4	100	0	0	0	00:01:35	Active	0

可以看到，R1与R2的邻居关系停留在Active状态，而非Established，这说明R1与R2尚未正常建立起IBGP邻居关系。

在配置BGP邻居时所使用的IP地址，应该互为BGP报文的源IP地址和目的IP地址。默认情况下，BGP会使用去往邻居路由器的出接口的IP地址作为BGP报文的源地址。在上面的配置中，R2向R1发送BGP报文的源IP地址和R1上指定的邻居地址10.0.2.2不一致，从而导致了R1无法和R2正常建立BGP邻居关系。解决此问题的方法是通过命令来强制指定路由器发送BGP报文时所使用的源IP地址。

在R1上使用命令 peer 10.0.2.2 connect-interface LoopBack 0，指定R1使用自己的Loopback 0接口地址作为发送BGP报文时的源IP地址；R2上也需使用类似的命令。

```
[R1]bgp 100
```

```
[R1-bgp]peer 10.0.2.2 connect-interface LoopBack 0
```

```
[R2]bgp 100
```

```
[R2-bgp]peer 10.0.1.1 connect-interface LoopBack 0
```

上述配置完成后，在R2上查看BGP邻居关系。

```
[R2]display bgp peer
```

```
BGP local router ID : 10.0.2.2
```

```
Local AS number : 100
```

```
Total number of peers : 1      Peers in established state : 1
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.1.1	4	100	4	3	0	00:01:05
Established	1					

可以看到，R2现在只与10.0.1.1有一个IBGP邻居关系，状态为Established。

查看R2的BGP路由表。

```
[R2]display bgp routing-table
```

```
BGP Local router ID is 10.0.2.2
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 1
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	10.0.1.1/32	10.0.1.1	0	100	0	i

可以看到，R2的BGP路由表中只有一条去往10.0.1.1/32的路由，下一跳为10.0.1.1。

再查看R2的IP路由表。

```
[R2]display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```

-----
-----
Routing Tables:Public
Destinations : 18  Routes : 19

Destination/Mask    Proto Pre  Cost    Flags  NextHop
-----
Interface
10.0.1.1/32          Static 60     0        RD     10.0.12.1
GigabitEthernet0/0/0
    Static 60  0      RD  10.0.21.1  GigabitEthernet0/0/2
10.0.2.2/32          Direct 0     0        D      127.0.0.1    LoopBack0
.....

```

可以看到，R2去往10.0.1.1的路由有两条，下一跳分别为10.0.12.1和10.0.21.1，因此，当10.0.12.0/24这条链路不可用时，R1和R2之间的邻居关系不会受到影响，相应的BGP路由也不会受到影响。

总之，使用Loopback接口建立BGP邻居关系与使用物理接口来建立邻居关系相比较，前者具有更好的稳定性，且能够减少设备资源的开销。

3. 配置EBGP邻居

从前面的实验内容我们知道，使用物理接口来建立R1和R2的BGP邻居关系时，配置相对简单，并且能够实现邻居关系的冗余。但是，这样的配置会产生两个TCP会话和两个BGP邻居关系，当有路由需要彼此通告时，会通过这两个邻居关系分别进行通告，因而比较消耗设备资源，并且链路的不稳定也会导致BGP邻居关系的不稳定。

接下来，我们将在R2和R3上使用Loopback 0接口来建立EBGP邻居关系。

```

[R2]bgp 100
[R2-bgp]peer 10.0.3.3 as-number 200

```

```
[R3]bgp 200
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.2.2 as-number 100
```

上述配置完成后，在R3上查看BGP邻居关系。

```
[R3]display bgp peer
BGP local router ID : 10.0.3.3
Local AS number : 200
Total number of peers : 1      Peers in established state : 0
Peer    V    AS    MsgRcvd   MsgSent   OutQ   Up/Down   State
PrefRcv
10.0.2.2  4   100    0         0         0    00:01:58   Idle    0
```

可以看到，R2与R3的邻居状态一直停留在Idle状态，说明邻居关系未能正常建立。

我们知道，BGP邻居关系建立的前提条件是要能够建立起TCP会话，而目前R2和R3上都不存在去往对方Loopback 0接口的路由，因此无法建立TCP会话。为了解决这个问题，可以在R2和R3上配置到达对方Loopback 0接口的静态路由。

```
[R2]ip route-static 10.0.3.0 255.255.255.0 10.0.23.3
[R2]ip route-static 10.0.3.0 255.255.255.0 10.0.32.3
[R3]ip route-static 10.0.2.0 255.255.255.0 10.0.23.2
[R3]ip route-static 10.0.2.0 255.255.255.0 10.0.32.2
```

上述配置完成后，在R3上查看BGP邻居关系。

```
[R3]display bgp peer
BGP local router ID : 10.0.3.3
Local AS number : 200
Total number of peers : 1      Peers in established state : 0
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv

10.0.2.2	4	100	0	0	0	00:09:32	Active	0

可以看到，R2与R3之间的邻居关系一直停留在Active状态，说明邻居关系还是未能正常建立起来，其原因我们在前面已有解释。

```
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```

```
[R3-bgp]peer 10.0.2.2 connect-interface LoopBack 0
```

配置完成后，在R3上查看BGP邻居关系。

```
[R3]display bgp peer
```

```
BGP local router ID : 10.0.3.3
```

```
Local AS number : 200
```

```
Total number of peers : 1      Peers in established state : 0
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv

10.0.2.2	4	100	1	0	0	00:00:17	Idle	0

可以看到，现在R3的邻居状态是Idle，说明R2与R3之间的EBGP邻居关系仍然未能正常建立。

原来，在默认情况下，EBGP邻居之间在发送BGP报文时，TTL值为1，所以EBGP默认要求邻居之间必须物理直连。但是，当R2和R3使用Loopback 0接口建立邻居关系时，由于使用的不是物理直连的接口，所以TTL值会被多减一次，成为0，最终使得BGP报文会被丢弃，从而导致邻居关系无法建立。为解决这一问题，可以修改EBGP邻居发送BGP报文的TTL值，使报文的TTL值大于1。

在R2和R3上使用命令peer ebgp-max-hop 2，配置BGP报文的TTL值为2。

```
[R2-bgp]peer 10.0.3.3 ebgp-max-hop 2
```

```
[R3-bgp]peer 10.0.2.2 ebgp-max-hop 2
```

上述配置完成后，在R3上查看BGP邻居关系。

```
[R3-bgp]display bgp peer
```

```
BGP local router ID : 10.0.3.3
```

```
Local AS number : 200
```

```
Total number of peers : 1      Peers in established state : 1
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
10.0.2.2	4	100	3	3	0	00:01:47	Established	0

可以看到，R2与R3之间已经建立起了EBGP邻居关系。

需要说明的是，在实际场景中，通常使用Loopback接口来建立IBGP邻居关系，使用物理接口建立EBGP邻居关系。

思考

BGP协议选择了TCP协议作为其传输协议，这样做有什么好处？
OSPF协议也是以TCP协议作为其传输协议吗？

3.2 BGP 认证功能

原理概述

BGP是一种运行在AS之间的动态路由协议，具备强大的路径选择能力，这也使得BGP协议能够管理超大型网络。对于超大型网络来说，路由的稳定性和安全性尤为重要，因为它直接影响到超大型网络的稳定性和安全性。我们知道，路由协议的报文一般都是明文发送的，如果网络攻击者伪造路由更新报文，或者篡改路由更新报文，就会造成严重的网络安全问题。因此，在实际部署各种路由协议时，通常会配置认证功能，BGP更是如此。所谓认证，就是指路由器对路由信息来源的可靠性及路由信息本身的完整性进行检测的机制。

BGP支持简单的密码认证方式，也支持安全性更高的MD5认证方式。如果是MD5认证方式，路由器会根据BGP报文的某些字段和密钥计算出一个128比特的散列值，然后将 BGP 报文连同散列值发送给邻居。邻居路由器收到之后，会在本地基于接收到的BGP报文和相同的密钥再进行一次Hash运算。如果计算出的散列值与接收到的散列值相同，则认证通过，邻居关系能够正常建立；如果不同，则认证不通过，邻居关系就不会建立，且所收到的BGP报文会被丢弃。

实验目的

- 掌握基于单一密钥的BGP 认证功能的配置
- 掌握基于Keychain 的BGP 认证功能的配置

实验内容

实验拓扑如图3-2所示，实验编址如表3-2所示。本实验使用了3台路由器，R1和R2属于ISP-A的网络，AS编号为100，R3属于ISP-B的网络，AS编号为200。R1与R2建立IBGP邻居关系，R2与R3建立EBGP邻居关系。为了保证BGP邻居之间发送的BGP路由信息的完整性，决定在BGP邻居之间配置认证功能。

实验拓扑

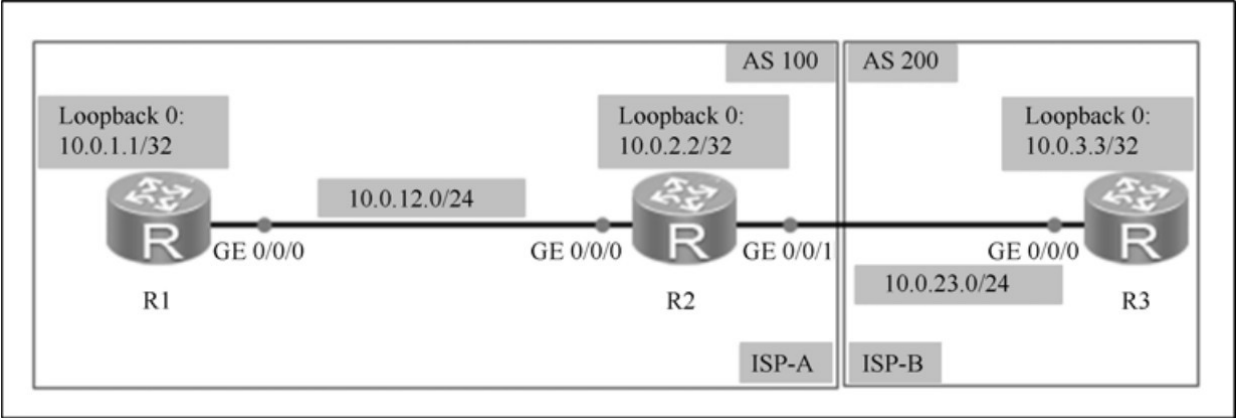


图3-2 BGP认证功能

实验编址表

表3-2 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.23.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/0	10.0.23.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-2和表3-2进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=430 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 430/430/430 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置BGP路由协议

R1、R2 和R3 的Router-ID分别为1.1.1.1、2.2.2.2和3.3.3.3，R1 与R2 属于AS 100，R3 属于AS 200，使用直连的物理接口的IP 地址来建立BGP 邻居关系。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 1.1.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 100
```

```
[R2]bgp 100
```

```
[R2-bgp]router-id 2.2.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.23.3 as-number 200
```

```
[R3]bgp 200
```

```
[R3-bgp]router-id 3.3.3.3
```

```
[R3-bgp]peer 10.0.23.2 as-number 100
```

上述配置完成后，在R2上查看BGP邻居关系。

```
[R2]display bgp peer
```

```
BGP local router ID : 2.2.2.2
```

```
Local AS number : 100
```

```
Total number of peers : 2      Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.12.1	4	100	10	10	0	00:08:24 Established
0						
10.0.23.3	4	200	9	11	0	00:07:21 Established
0						

可以看到，R2与R1和R3之间的邻居状态均为Established，说明邻居关系已正常建立。

3．配置基于单一密钥的BGP认证功能

经过上述步骤，BGP邻居关系已经建立，路由器之间可以传递BGP路由信息了。但是，因为没有配置BGP的认证功能，所以网络会存在很大的安全风险。

接下来，分别在R1、R2、R3上配置BGP认证功能。

```
[R1]bgp 100
```

```
[R1-bgp]peer 10.0.12.2 password simple Huawei
```

```
[R2]bgp 100
```

```
[R2-bgp]peer 10.0.12.1 password simple Huawei123
```

```
[R2-bgp]peer 10.0.23.3 password simple Huawei
```

```
[R3]bgp 200
```

```
[R3-bgp]peer 10.0.23.2 password cipher Huawei
```

配置完成后，查看R1的BGP邻居关系。

```
[R1]display bgp peer
```

```
BGP local router ID : 1.1.1.1
```

```
Local AS number : 100
```

```
Total number of peers : 1      Peers in established state : 0
```

```
Peer  V   AS  MsgRcvd  MsgSent  OutQ  Up/Down  State
```

```
PrefRcv
```

```
10.0.12.2  4  100  0        0        0    00:02:23  Connect    0
```

可以看到，R1与R2的邻居状态变成了Connect，说明R1与R2的邻居关系未能正常建立。

查看R2的BGP邻居关系。

```
[R2]display bgp peer
```

```
BGP local router ID : 2.2.2.2
```

```
Local AS number : 100
```

```
Total number of peers : 2      Peers in established state : 1
```

```
Peer  V   AS  MsgRcvd  MsgSent  OutQ  Up/Down  State
```

```
PrefRcv
```

```
10.0.12.1  4  100  0        0        0    00:02:34  Connect    0
```

```
10.0.23.3  4  200  7        8        0    00:05:11  Established 0
```

可以看到，R2与R3的邻居状态为Established，与R1的邻居状态为Connect，说明R2与R3建立了正常的邻居关系，但与R1未能建立起正常的邻居关系。

出现上述情况的原因是R1使用的密钥Huawei与R2使用的密钥Huawei123不一致，无法进行正确的认证，所以建立不了邻居关系。在

R2与R3之间，虽然R2使用的是Simple方式，R3使用的是Cipher方式，但这并不影响R2与R3之间的邻居关系的建立，这是由于Cipher与Simple两种方式的区别仅仅在于：设备在存储密钥时，使用Cipher方式后的密钥将会被加密，而Simple方式是明文显示的，可以查看到真实的密钥信息。

在R1、R2、R3 上的BGP 视图下使用display this 命令，查看当前的BGP 认证配置信息。

```
[R1-bgp]display this
bgp 100
  router-id 1.1.1.1
  peer 10.0.12.2 as-number 100
  peer 10.0.12.2 password simple Huawei
  #
  ipv4-family unicast
  undo synchronization
  peer 10.0.12.2 enable
  #
return
[R2-bgp]display this
bgp 100
  router-id 2.2.2.2
  peer 10.0.12.1 as-number 100
  peer 10.0.12.1 password simple Huawei123
  peer 10.0.23.3 as-number 200
  peer 10.0.23.3 password simple Huawei
  #
  ipv4-family unicast
```

```

undo synchronization
peer 10.0.12.1 enable
peer 10.0.23.3 enable
#
return
[R3-bgp]display this
bgp 200
  router-id 3.3.3.3
  peer 10.0.23.2 as-number 100
  peer          10.0.23.2          password          cipher
  %$%$IKUxMU_Lg5IDZm<Hq@(1QC0#%$%$
  #
  ipv4-family unicast
  undo synchronization
  peer 10.0.23.2 enable
  #
return

```

可以看到，在R2上配置的认证密钥是明文显示的，在R3上配置的认证密钥是密文显示的，并且R2上配置的密钥和R1上的不一致。

在R2上将错误的密钥进行更正，然后再次查看R2的BGP邻居关系。

```

[R2]bgp 100
[R2-bgp]peer 10.0.12.1 password simple Huawei
[R2]display bgp peer
BGP local router ID : 2.2.2.2
Local AS number : 100
Total number of peers : 2      Peers in established state : 2

```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
PrefRcv							
10.0.12.1		4	100	2	2	0	00:00:47
Established 0							
10.0.23.3		4	200	7	7	0	00:05:05
Established 0							

可以看到，密钥修正之后，R2与R1之间的邻居关系已经正常建立。

配置BGP认证功能时，通常会使用Cipher方式，使得存储于配置文件中的密钥不会以明文方式显示，密钥的安全性更高。

4. 配置基于Keychain的BGP认证

上面的实验中，BGP进行认证时使用的是一个固定的密钥，当需要变换密钥来增强安全性时，操作将非常繁琐，并且还会造成BGP连接的中断。为此，我们可以使用基于Keychain的认证方式来实现密钥的周期性更换，并且对众多的密钥进行集中管理。使用Keychain的方式可以定义密钥的存活期，但应保证设备的系统时间一致，避免认证失败。在实际场景中，设备通常会使用NTP（Network Time Protocol）协议来保证时间的同步。

在Keychain方式下定义密钥的存活期分为Absolute与Periodic两种模式。Absolute模式下，密钥Key的有效时间为一个绝对时间段；Periodic模式下，一个Key的有效时间为周期性的一段时间，分为Daily、Monthly、Weekly和Yearly等。以Daily为例，一个Key的有效时间为每一天的某一特定时间段。一个Keychain中可以有多个Key，最多可支持64个Key-ID。

Key 具有多个属性，包括 Key-ID、认证算法、Key-String 以及 Send-Time 和Receive-Time，其中 Send-Time 和 Receive-Time 用来定义

Keychain 中某个 Key-ID 的 Active 时间段。如果系统时间不在 Send-Time 或者 Receive-Time 时间内，则该 Key-ID 不会被使用。

接下来，在 R1 和 R2 上配置基于 Keychain 的认证功能，Key-ID 为 1，Key-String 为 huawei，选用 Periodic Daily 模式，每天 08:00 到 18:00 使用 Key-ID 1 对发送的 BGP 报文做 Hash 运算，每天 08:00 到 18:00 使用 Key-ID 1 对接收到的 BGP 报文进行认证。

```
[R1]keychain key mode periodic daily
[R1-keychain]key-id 1
[R1-keychain-keyid-1]algorithm md5
[R1-keychain-keyid-1]key-string huawei
[R1-keychain-keyid-1]send-time daily 08:00 to 18:00
[R1-keychain-keyid-1]receive-time daily 08:00 to 18:00
[R1-keychain-keyid-1]bgp 100
[R1-bgp]undo peer 10.0.12.2 password
[R1-bgp]peer 10.0.12.2 keychain key
[R2]keychain key mode periodic daily
[R2-keychain]key-id 1
[R2-keychain-keyid-1]algorithm md5
[R2-keychain-keyid-1]key-string huawei
[R2-keychain-keyid-1]send-time daily 08:00 to 18:00
[R2-keychain-keyid-1]receive-time daily 08:00 to 18:00
[R2-keychain-keyid-1]bgp 100
[R2-bgp]undo peer 10.0.12.1 password
[R2-bgp]peer 10.0.12.1 keychain key
配置完成后，在R1上使用命令查看BGP邻居关系。
[R1]display bgp peer
BGP local router ID : 1.1.1.1
```


Local AS number : 100

Total number of peers : 1 Peers in established state : 1

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
PrefRcv							
10.0.12.2	4	100	6	7	0	00:04:53	Established

0

可以看到，R1与R2的邻居关系已正常建立。

在两台路由器上使用Keychain认证时，应保证Keychain的名称、Key-ID、Algorithm和Key-String保持一致，任意一个参数不匹配都会导致认证失败。当一个Key的某个属性不完整或系统时间不在定义的时间段内，Key 会处于 Inactive 状态，不会被用来进行认证。另外，在一个Keychain 中，不同的 Key-ID 的 Send-Time 时间不能重叠，但Receive-Time时间可以重叠，保证在任何时间段内，BGP报文的发送方只使用一个Key-ID所对应的Key-String对发送的BGP报文进行Hash值计算，接收方对接收到的BGP报文将使用在Receive-Time为Active的且Key-ID相同的Key-String来进行认证。如果认证设备之间的系统时间不一致，但使用的Key-ID相同，并且在自身设备上Active的，认证也是可以通过的。

在R1 上使用display keychain key 命令来查看Keychain 的信息。

<R1>display keychain key

Keychain Information:

Keychain Name	: key
Timer Mode	: Daily periodic
Receive Tolerance(min)	: 0
TCP Kind	: 254
TCP Algorithm IDs	:

```

HMAC-MD5                : 5
HMAC-SHA1-12            : 2
HMAC-SHA1-20            : 6
MD5                      : 3
SHA1                    : 4
Number of Key IDs       : 1
Active Send Key ID      : 1
Active Receive Key IDs  : 01
Default sendKey ID      : Not configured
Key ID Information:
-----
Key ID                   : 1
Key string               :
%$%$)F]>$CBTi@rR*1.#M\_3e5vT%$%$ (cipher)
Algorithm                : MD5
SEND TIMER               :
    Start time           : 08:00
    End time             : 18:00
    Status                : Active
RECEIVE TIMER            :
    Start time           : 08:00
    End time             : 18:00
    Status                : Active

```

显示信息表明，Keychain 的名称为 key，密钥的数量为 1。显示信息还包括了处于活动状态的Key-ID信息。

在没有NTP来保证时间同步的情况下，尽管管理员可以手动调整时间以尽量保证时间的一致性，但这样做的精度很差。在这种情况下

下，可以使用下面的命令来配置接收容忍时间，避免由于时间不同步或者Key-ID的变更过程中存在的时间延迟而导致BGP报文认证失败的情况。接收容忍时间只对接收端的 Key 有效，其原理就是延长了Receive-Timer时间。

```
[R1-keychain]receive-tolerance infinite
```

Infinite表示容忍所有的时间延迟；也可以用某一具体的时间代替，单位是min，最大值是14400min。

为了避免在某一时刻没有活跃的Key-ID而导致BGP没有认证交互的情况，可以使用命令default send-key-id 指定一个缺省的发送Key-ID。一个Keychain 中最多只能有一个Key-ID配置为缺省的发送Key-ID。

当一个 Keychain 中有多个 Key-ID 时，可以合理地给不同的 Key-ID 设置不同的Send-Time和Receive-Time，实现密钥的无缝隙周期性更换，并且不会导致BGP邻居关系的中断。当然，如果密钥的更换存在时间缝隙，但缝隙不超过 180s（一个 HoldTime周期），BGP连接也是不会中断的。

思考

显然，Keychain认证方式相比于单一密钥认证方式来讲，安全性更高，特性也更丰富。那么为什么在实际的网络部署中还是可能会使用单一密钥认证方式呢？

3.3 BGP 自动路由聚合

原理概述

在大型网络中，路由条目通常多达成千上万条，甚至几十万条，这给路由设备带来的挑战是：如何存储并有效管理如此众多的路由信

息？

BGP 是一种无类路由协议，支持CIDR、VLSM（Variable Length Sub-network Mask）和路由聚合。路由聚合技术的使用，可以在一定程度上缩减路由条目的数量，同时还可以减轻路由震荡导致的网络不稳定的问题。BGP的路由聚合有两种方式，一种是自动路由聚合，一种是手动路由聚合。

自动路由聚合是在自然网络边界路由器上自动执行的。在默认情况下，BGP的自动路由聚合功能是关闭的，并且BGP不会自动聚合BGP邻居发送的路由以及使用network命令通告的路由。

使用BGP自动路由聚合时，需要进行严谨的IP地址规划。在一个地址规划杂乱无序的网络中，自动路由聚合可能会产生许多意想不到的问题。例如，在采用不连续子网规划的网络中，自动路由聚合可能会导致报文转发出现选路问题，或者是产生路由环路。

实验目的

- 理解BGP 自动路由聚合的概念
- 掌握BGP 自动路由聚合的配置

实验内容

实验拓扑如图3-3所示，实验编址如表3-3所示。本实验模拟了3个运营商网络，R1属于ISP-A，R2属于ISP-B，R3属于ISP-C。三台路由器都使用直连的物理接口IP地址来建立EBGP 邻居关系，R2 和R3的Loopback 1 接口用来模拟各自ISP 中的一个网段。R1、R2、R3 上将开启自动路由聚合功能，R3 的 Loopback 1 接口所在网段将使用network命令通告给BGP进程，R2的Loopback 1接口所在网段将被引入到BGP进程中，最终实现R2 的Loopback 1 与R3 的Loopback 1 之间可以互相通信。

实验拓扑

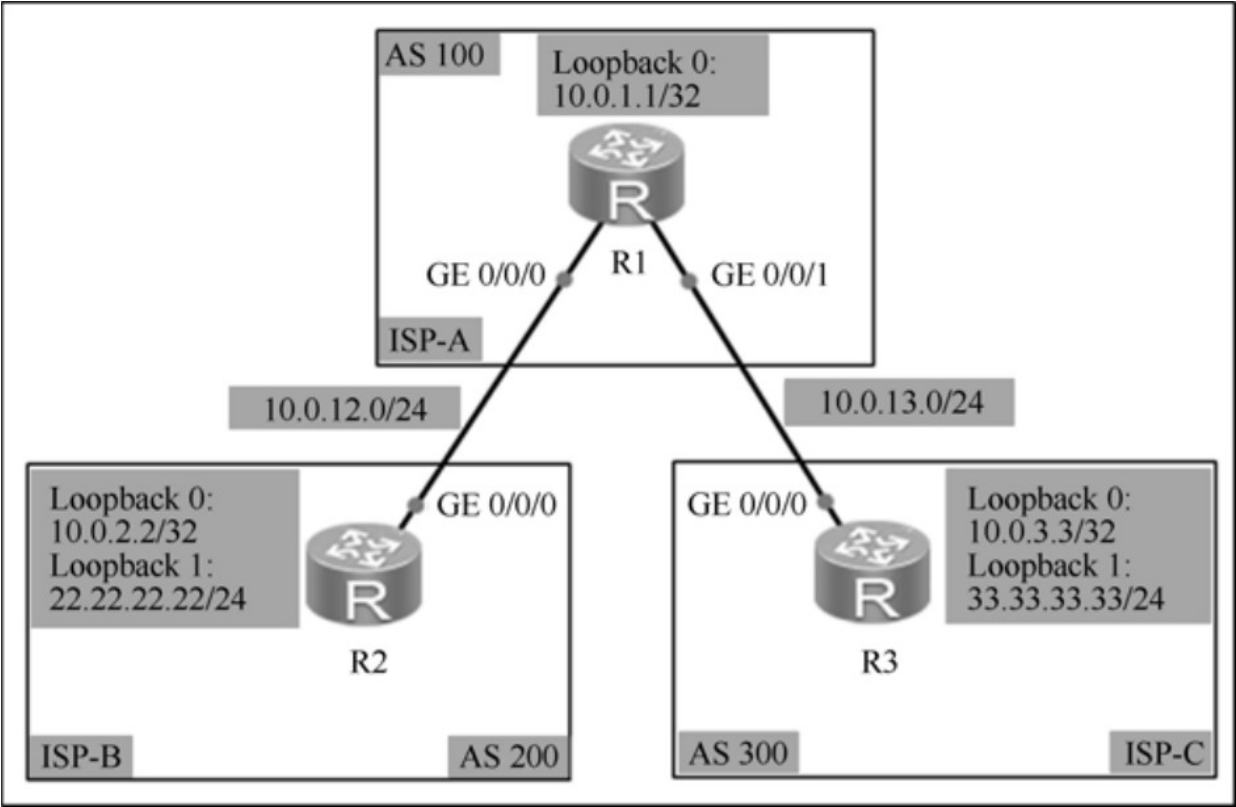


图3-3 BGP自动路由聚合

实验编址表

表3-3 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
	Loopback 1	22.22.22.22	255.255.255.0	N/A
R3(AR2220)	GE 0/0/0	10.0.13.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
	Loopback 1	33.33.33.33	255.255.255.0	N/A

实验步骤

1．基本配置

根据图3-3和表3-3进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=250 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 250/250/250 ms
```

其余直连网段的连通性测试过程在此省略。

2 . 配置BGP路由协议

配置 BGP 邻居关系，每台路由器均使用 Loopback 0 接口的 IP 地址作为自己的Router-ID。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 200
```

```
[R1-bgp]peer 10.0.13.3 as-number 300
```

```
[R2]bgp 200
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R3]bgp 300
```

```
[R3-bgp]router-id 10.0.3.3
```

```
[R3-bgp]peer 10.0.13.1 as-number 100
```

配置完成后，查看R1的BGP邻居关系。

```
[R1]display bgp peer
```

```
BGP local router ID : 10.0.1.1
```

```
Local AS number : 100
```

```
Total number of peers : 2      Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State
PrefRcv							
10.0.12.2		4	200	3	4	0	00:01:22
Established							
10.0.13.3		4	300	2	3	0	00:00:52
Established							

可以看到，R1与R2、R1与R3之间的邻居状态都为Established，表示邻居关系已正常建立。

3．开启BGP自动路由聚合功能

缺省情况下，华为设备的BGP自动路由聚合功能是关闭的，现在开启这一功能。

```
[R1]bgp 100
[R1-bgp]ipv4-family unicast
[R1-bgp-af-ipv4]summary automatic
[R2]bgp 200
[R2-bgp]ipv4-family unicast
[R2-bgp-af-ipv4]summary automatic
[R3]bgp 300
[R3-bgp]ipv4-family unicast
[R3-bgp-af-ipv4]summary automatic
```

当路由器的BGP自动路由聚合功能打开时，系统会有如下的提示信息。

Info: Automatic summarization is valid only for the routes imported through the import-route command.

这说明，BGP自动路由聚合只适用于通过路由引入方式引入的路由。

4．通告路由进入BGP中

使用network 命令，将R3 的Loopback 1 接口所在网段通告进入BGP 进程。

```
[R3]bgp 300
[R3-bgp]ipv4-family unicast
[R3-bgp-af-ipv4]network 33.33.33.0 24
```

然后，在R1、R2、R3 上使用display bgp routing-table 命令，查看BGP 路由表。

```
[R1]display bgp routing-table
BGP Local router ID is 10.0.1.1
Status codes:*- valid, >- best, d - damped,
             h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete
```

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 33.33.33.0/24	10.0.13.3	0		0	300i

```
[R2]display bgp routing-table
BGP Local router ID is 10.0.2.2
Status codes:*- valid, >- best, d - damped,
             h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete
```

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 33.33.33.0/24	10.0.12.1			0	100 300i

```
[R3]display bgp routing-table
BGP Local router ID is 10.0.3.3
Status codes:*- valid, >- best, d - damped,
             h - history, i - internal, s - suppressed, S - Stale
```


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

Total Number of Routes: 1

Network	NextHop	ME D	LocPrf	PrefVal	Path/Ogn
*> 33.33.33.0/24	0.0.0.0	0		0	i

可以看到，在R1、R2、R3的BGP路由表中，33.33.33.0/24并没有被聚合。R3将33.33.33.0/24通告给R1，R1再通告给R2，整个过程中该路由都没有被聚合。这就说明，使用network命令通告到BGP中的路由，在自然网络边界处，BGP不会进行自动聚合。

5. 引入外部路由到BGP协议中

在R2上使用import-route命令引入直连的路由。

```
[R2]bgp 200
```

```
[R2-bgp]ipv4-family unicast
```

```
[R2-bgp-af-ipv4]import-route direct
```

配置完成后，查看R1、R2、R3的BGP路由表。

```
[R1]display bgp routing-table
```

BGP Local router ID is 10.0.1.1

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 10.0.0.0	10.0.12.2			0	200?
*> 22.0.0.0	10.0.12.2			0	200?
*> 33.33.33.0/24	10.0.13.3	0		0	300i

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 10

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.0.0	127.0.0.1			0	?
s>	10.0.2.2/32	0.0.0.0		0	0	?
s>	10.0.12.0/24	0.0.0.0		0	0	?
*>	10.0.12.2/32	0.0.0.0		0	0	?
*>	22.0.0.0	127.0.0.1			0	?
s>	22.22.22.0/24	0.0.0.0		0	0	?
*>	22.22.22.22/32	0.0.0.0		0	0	?
*>	33.33.33.0/24	10.0.12.1			0	100 300i
*>	127.0.0.0	0.0.0.0		0	0	?
*>	127.0.0.1/32	0.0.0.0		0	0	?

[R3]display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.0.0	10.0.13.1			0	100 200?
*>	22.0.0.0	10.0.13.1			0	100 200?
*>	33.33.33.0/24	0.0.0.0		0	0	i

可以看到，在R1和R3的BGP路由表中，都出现了22.0.0.0，且没有显示掩码信息，这正是聚合后的一个A类自然网络。

与RIPv2路由协议类似，当开启了BGP自动路由聚合功能之后，R2的GE 0/0/0接口属于10.0.12.0/24网段，自然网络号是10.0.0.0/8，Loopback 1属于22.22.22.0/24网段，自然网络号是22.0.0.0/8，两个都是A类网络，但两个自然网络号不相同。R2的Loopback 1被引入到BGP之后，会被通告给R1，由于22.22.22.0/24的自然网络号为22.0.0.0/8，和R2上发送这个更新的物理接口GE 0/0/0所在的10.0.0.0/8这个自然网络号不同，所以R2位于自然网络的边界，因此当R2将22.22.22.0/24通告给R1时便进行了路由的自动聚合。

自动路由聚合对IP地址规划的要求是比较苛刻的，而在BGP网络环境中，IP地址的规划难以做到规整有序，所以在实际项目中，很少启用BGP自动路由聚合功能。读者只需了解，BGP的自动路由聚合功能开启后，会对哪些BGP路由进行自动聚合即可。

思考

在不连续的IP地址规划中，是否可以开启BGP自动路由聚合功能？为什么？

3.4 BGP 手动路由聚合

原理概述

BGP的路由聚合有两种方式，一种是自动路由聚合，一种是手动路由聚合。相对于自动路由聚合来讲，手动路由聚合具有更高的灵活性和可控性。

BGP手动路由聚合时，可以手动控制聚合路由的掩码长度，修改聚合路由属性等。手动路由聚合又有两种方法，一种是配置一条静态路由，然后用network命令进行通告；另一种是使用aggregate命令进行聚合。

采用第一种方法时，无法对通告的静态路由加以控制，并且明细路由仍然会被通告出去。如要抑制明细路由，则需使用 Route-Policy 来对明细路由进行过滤，实现起来配置命令较多，同时还会丢失明细路由的某些BGP属性。

采用第二种方法时，缺省情况下明细路由和聚合路由也都会被发送出去，但是可以通过关键字对全部或部分明细路由进行抑制，另外还可以对聚合路由的属性进行修改。和第一种方法相比，第二种方法对路由聚合的控制以及对路径选择的控制会更加灵活。

实验目的

- 掌握配置BGP 手动路由聚合的方法
- 熟悉aggregate 命令中关键字的作用

实验内容

实验拓扑如图3-4所示，实验编址如表3-4所示。R1属于AS 100，R2属于AS 200，R3和R4属于AS 300，R5和R6属于AS 400，每台路由器都使用自己的Loopback 0接口IP地址作为Router-ID，并且都使用直连物理接口建立邻居关系，整网运行BGP协议。在R5和R6上使用network命令通告Loopback 1至Loopback 7接口所在网络到BGP进程中，在R5上用aggregate命令聚合这些Loopback接口所在网络的路由，在R6上配置一条静态路由，并且使用network命令通告这条静态路由来实现路由聚合。实验过程中，在R5上还存在一些具体的路由控制需求，这些需求将采用aggregate命令结合一些关键字来实现。

实验拓扑

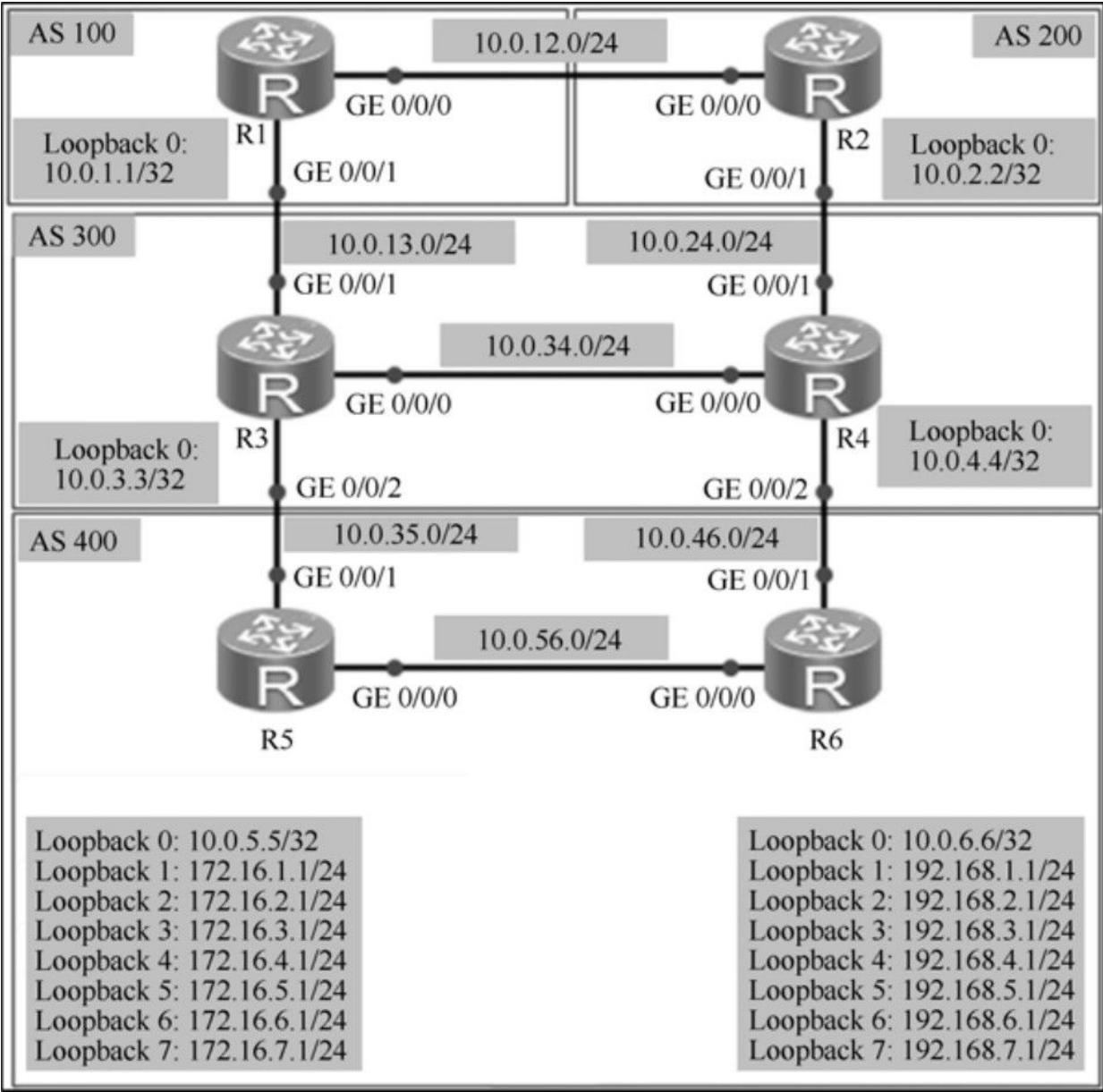


图3-4 BGP手动路由聚合

实验编址表

表3-4 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.24.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.35.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2220)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	GE 0/0/2	10.0.46.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
R5(AR2220)	GE 0/0/0	10.0.56.5	255.255.255.0	N/A
	GE 0/0/1	10.0.35.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A
	Loopback 1	172.16.1.1	255.255.255.0	N/A
	Loopback 2	172.16.2.1	255.255.255.0	N/A
	Loopback 3	172.16.3.1	255.255.255.0	N/A
	Loopback 4	172.16.4.1	255.255.255.0	N/A
	Loopback 5	172.16.5.1	255.255.255.0	N/A
	Loopback 6	172.16.6.1	255.255.255.0	N/A
	Loopback 7	172.16.7.1	255.255.255.0	N/A
R6(AR2220)	GE 0/0/0	10.0.56.6	255.255.255.0	N/A
	GE 0/0/1	10.0.46.6	255.255.255.0	N/A
	Loopback 0	10.0.6.6	255.255.255.255	N/A
	Loopback 1	192.168.1.1	255.255.255.0	N/A
	Loopback 2	192.168.2.1	255.255.255.0	N/A
	Loopback 3	192.168.3.1	255.255.255.0	N/A
	Loopback 4	192.168.4.1	255.255.255.0	N/A
	Loopback 5	192.168.5.1	255.255.255.0	N/A
	Loopback 6	192.168.6.1	255.255.255.0	N/A
	Loopback 7	192.168.7.1	255.255.255.0	N/A

实验步骤

1. 基本配置

根据图3-4和表3-4进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

PING 10.0.12.2: 56 data bytes, press CTRL_C to break

Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=390 ms

--- 10.0.12.2 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 390/390/390 ms

其余直连网段的连通性测试过程在此省略。

2 . 配置BGP路由协议

每台路由器都使用自己的Loopback 0 接口IP 地址作为Router-ID , 并且都使用直连物理接口建立BGP 邻居关系。R5和R6 的Loopback 1 至 Loopback 7 接口所在的网络将使用network命令通告到BGP进程中。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 200
```

```
[R1-bgp]peer 10.0.13.3 as-number 300
```

```
[R2]bgp 200
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.24.4 as-number 300
```

```
[R3]bgp 300
```

```
[R3-bgp]router-id 10.0.3.3
```

```
[R3-bgp]peer 10.0.13.1 as-number 100
```

```
[R3-bgp]peer 10.0.34.4 as-number 300
```

```
[R3-bgp]peer 10.0.35.5 as-number 400
```

```
[R4]bgp 300
```

```
[R4-bgp]router-id 10.0.4.4
```

```
[R4-bgp]peer 10.0.24.2 as-number 200
[R4-bgp]peer 10.0.34.3 as-number 300
[R4-bgp]peer 10.0.46.6 as-number 400
[R5]bgp 400
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.35.3 as-number 300
[R5-bgp]peer 10.0.56.6 as-number 400
[R5-bgp]network 172.16.1.0 255.255.255.0
[R5-bgp]network 172.16.2.0 255.255.255.0
[R5-bgp]network 172.16.3.0 255.255.255.0
[R5-bgp]network 172.16.4.0 255.255.255.0
[R5-bgp]network 172.16.5.0 255.255.255.0
[R5-bgp]network 172.16.6.0 255.255.255.0
[R5-bgp]network 172.16.7.0 255.255.255.0
[R6]bgp 400
[R6-bgp]router-id 10.0.6.6
[R6-bgp]peer 10.0.46.4 as-number 300
[R6-bgp]peer 10.0.56.5 as-number 400
[R6-bgp]network 192.168.1.0 255.255.255.0
[R6-bgp]network 192.168.2.0 255.255.255.0
[R6-bgp]network 192.168.3.0 255.255.255.0
[R6-bgp]network 192.168.4.0 255.255.255.0
[R6-bgp]network 192.168.5.0 255.255.255.0
[R6-bgp]network 192.168.6.0 255.255.255.0
[R6-bgp]network 192.168.7.0 255.255.255.0
```

上述配置完成后，在 R1 上查看 BGP 邻居关系，读者可自行查看其他路由器上的BGP邻居关系。

<R1>display bgp peer

BGP local router ID : 10.0.1.1

Local AS number : 100

Total number of peers : 2 Peers in established state : 2

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.12.2	4	200	8	10	0	00:07:01
Established	0					
10.0.13.3	4	300	8	9	0	00:06:14
Established	0					

可以看到，R1与它的所有对等体的邻居关系都已正常建立。

在R1上查看BGP路由表。

<R1>display bgp routing-table

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 28

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 172.16.1.0/24	10.0.13.3			0	300 400i
*	10.0.12.2		0	200 300	400i
*> 172.16.2.0/24	10.0.13.3			0	300 400i
*	10.0.12.2		0	200 300	400i
*> 172.16.3.0/24	10.0.13.3			0	300 400i
*	10.0.12.2		0	200 300	400i
*> 172.16.4.0/24	10.0.13.3			0	300 400i
*	10.0.12.2		0	200 300	400i

```

*> 172.16.5.0/24    10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 172.16.6.0/24    10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 172.16.7.0/24    10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 192.168.1.0      10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 192.168.2.0      10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 192.168.3.0      10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 192.168.4.0      10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 192.168.5.0      10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 192.168.6.0      10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i
*> 192.168.7.0      10.0.13.3          0      300 400i
*                10.0.12.2          0      200 300 400i

```

可以看到，R1已经接收到了BGP协议的明细路由。在R5上使用ping命令测试R5的Loopback 1 与R6 的Loopback 1 之间的连通性。

```
<R5>ping -c 1 -a 172.16.1.1 192.168.1.1
```

```
PING 192.168.1.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 192.168.1.1: bytes=56 Sequence=1 ttl=255 time=70
ms
```

```
--- 192.168.1.1 ping statistics ---
```

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 70/70/70 ms

可以看到，通信是正常的。

3. 配置BGP路由聚合

上述步骤中，R5 和R6的Loopback 1 至Loopback 7 接口所在的网络已经被通告到BGP进程中了，并且每台路由器都接收到了R5 和R6 所通告的明细路由。如果其中某条明细路由出现丢失或震荡的情况，则网络中所有其他路由器都将删除这条路由或发生路由表震荡问题。现在，我们将在R5和R6上配置路由聚合，将这种明细路由的变化隐藏在AS内部，不会对其他AS产生影响。

在R6上配置静态路由，然后使用network命令通告出去。

```
[R6]ip route-static 192.168.0.0 21 NULL 0
```

```
[R6]bgp 400
```

```
[R6-bgp]network 192.168.0.0 21
```

静态路由指向NULL 0的目的是防止网络中产生环路，这条静态路由仅仅是用来通告的。

在R5上使用aggregate命令进行通告。

```
[R5]bgp 400
```

```
[R5-bgp]aggregate 172.16.0.0 21
```

使用aggregate命令进行聚合，要求BGP路由表中至少要存在一条属于聚合后的路由的子网路由，否则聚合不会生效。

在R1上查看BGP路由表。

```
<R1>display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 32

	Network	NextHop	MED	LocPrf	PrefVa	Path/Ogn
*>	172.16.0.0/21	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	172.16.1.0/24	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	172.16.2.0/24	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	172.16.3.0/24	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	172.16.4.0/24	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	172.16.5.0/24	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	172.16.6.0/24	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	172.16.7.0/24	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	192.168.0.0/21	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	192.168.1.0	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	192.168.2.0	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300 400i	
*>	192.168.3.0	10.0.13.3			0	300 400i

```

*           10.0.12.2           0           200 300 400i
*> 192.168.4.0   10.0.13.3           0           300 400i
*           10.0.12.2           0           200 300 400i
*> 192.168.5.0   10.0.13.3           0           300 400i
*           10.0.12.2           0           200 300 400i
*> 192.168.6.0   10.0.13.3           0           300 400i
*           10.0.12.2           0           200 300 400i
*> 192.168.7.0   10.0.13.3           0           300 400i
*           10.0.12.2           0           200 300 400i

```

可以看到，R1的BGP路由表中虽然包含了聚合后的路由，但同时还包含有每个明细路由条目。若R5或R6的明细路由发生丢失或震荡，依然会导致路由表的波动，并没有解决路由不稳定的问题。

4. 使用No-Advertise关键字控制路由聚合

上述步骤中虽然配置了路由聚合，但明细路由还是被通告出去了。R6上使用了network通告静态路由的聚合方法，若想抑制明细路由，则需要配合使用Route-Policy来实现对明细路由的抑制。如果被抑制明细路由的数量比较多时，配置工作量就会比较大，而且维护起来不太方便，这种方法的扩展性和可维护性都比较差。

使用Aggregate命令的方法时，默认情况下也不会抑制明细路由，明细路由和聚合路由都将被发布出去。但是，这种特性在某些情况下也是有利的。例如，如果要求R5、R6将自己的明细路由以及聚合路由都通告给AS 300的路由器，即R3和R4，但是R3和R4只能将聚合路由通告给AS 100的路由器R1和AS 200的路由器R2时，网络管理员只需要在R5、R6上发布明细路由时给路由添加No-Advertise团体属性。

在R5、R6上使用前缀列表和Route-Policy给这些明细路由添加No-Advertise团体属性，并通告给R3、R4。

```
[R5]ip ip-prefix no-adver permit 172.16.1.0 24
```

```
[R5]ip ip-prefix no-adver permit 172.16.2.0 24
[R5]ip ip-prefix no-adver permit 172.16.3.0 24
[R5]ip ip-prefix no-adver permit 172.16.4.0 24
[R5]ip ip-prefix no-adver permit 172.16.5.0 24
[R5]ip ip-prefix no-adver permit 172.16.6.0 24
[R5]ip ip-prefix no-adver permit 172.16.7.0 24
[R5]ip ip-prefix no-adver permit 192.168.1.0 24
[R5]ip ip-prefix no-adver permit 192.168.2.0 24
[R5]ip ip-prefix no-adver permit 192.168.3.0 24
[R5]ip ip-prefix no-adver permit 192.168.4.0 24
[R5]ip ip-prefix no-adver permit 192.168.5.0 24
[R5]ip ip-prefix no-adver permit 192.168.6.0 24
[R5]ip ip-prefix no-adver permit 192.168.7.0 24
[R5]route-policy no-adver permit node 10
[R5-route-policy]if-match ip-prefix no-adver
[R5-route-policy]apply community no-advertise
[R5-route-policy]route-policy no-adver permit node 20
[R5-route-policy]bgp 400
[R5-bgp]peer 10.0.35.3 route-policy no-adver export
[R5-bgp]peer 10.0.35.3 advertise-community
[R6]ip ip-prefix no-adver permit 192.168.1.0 24
[R6]ip ip-prefix no-adver permit 192.168.2.0 24
[R6]ip ip-prefix no-adver permit 192.168.3.0 24
[R6]ip ip-prefix no-adver permit 192.168.4.0 24
[R6]ip ip-prefix no-adver permit 192.168.5.0 24
[R6]ip ip-prefix no-adver permit 192.168.6.0 24
[R6]ip ip-prefix no-adver permit 192.168.7.0 24
```

```

[R6]ip ip-prefix no-adver permit 172.16.1.0 24
[R6]ip ip-prefix no-adver permit 172.16.2.0 24
[R6]ip ip-prefix no-adver permit 172.16.3.0 24
[R6]ip ip-prefix no-adver permit 172.16.4.0 24
[R6]ip ip-prefix no-adver permit 172.16.5.0 24
[R6]ip ip-prefix no-adver permit 172.16.6.0 24
[R6]ip ip-prefix no-adver permit 172.16.7.0 24
[R6]route-policy no-adver permit node 10
[R6-route-policy]if-match ip-prefix no-adver
[R6-route-policy]apply community no-advertise
[R6-route-policy]route-policy no-adver permit node 20
[R6-route-policy]bgp 400
[R6-bgp]peer 10.0.46.4 route-policy no-adver export
[R6-bgp]peer 10.0.46.4 advertise-community

```

peer x.x.x.x advertise-community 命令是为了将团体属性传递给任何对等体（组），缺省情况下是不传递的。

配置完成后，可在R1、R2、R3、R4上查看BGP路由表，下面仅以R1、R3为例。

```
<R1>display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 4
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	172.16.0.0/21	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300	400i

```
*> 192.168.0.0/21 10.0.13.3 0 300 400i
* 10.0.12.2 0 200 300 400i
```

<R3>display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 18

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 172.16.0.0/21	10.0.35.5			0	400i
i	10.0.46.6	100	0		400i
*> 172.16.1.0/24	10.0.35.5	0		0	400i
*> 172.16.2.0/24	10.0.35.5	0		0	400i
*> 172.16.3.0/24	10.0.35.5	0		0	400i
*> 172.16.4.0/24	10.0.35.5	0		0	400i
*> 172.16.5.0/24	10.0.35.5	0		0	400i
*> 172.16.6.0/24	10.0.35.5	0		0	400i
*> 172.16.7.0/24	10.0.35.5	0		0	400i
*> 192.168.0.0/21	10.0.35.5			0	400i
i	10.0.46.6	0	100	0	400i
*> 192.168.1.0	10.0.35.5			0	400i
*> 192.168.2.0	10.0.35.5			0	400i
*> 192.168.3.0	10.0.35.5			0	400i
*> 192.168.4.0	10.0.35.5			0	400i
*> 192.168.5.0	10.0.35.5			0	400i
*> 192.168.6.0	10.0.35.5			0	400i
*> 192.168.7.0	10.0.35.5			0	400i

可以看到，R1只接收到了聚合路由，但R3既接收到了聚合路由也接收到了明细路由。

添加No-Advertise团体属性是通过路由策略告知对等体的，不要再将这些明细路由通告给其他任何BGP对等体。

5. 使用Detail-Suppressed关键字控制路由聚合

网络管理员还可以利用aggregate命令结合使用Detail-Suppressed关键字来实现对明细路由的抑制，只将聚合后的路由发送出去。

```
[R5]bgp 400
```

```
[R5-bgp]aggregate 172.16.0.0 255.255.248.0 detail-suppressed
```

配置完成后，可以在R1、R2、R3、R4、R6上查看BGP路由表，下面仅以R1、R3、R6为例。

```
<R1>display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 4
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	172.16.0.0/21	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300	400i
*>	192.168.0.0/21	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300	400i

```
<R3>display bgp routing-table
```

```
BGP Local router ID is 10.0.3.3
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

Total Number of Routes: 11

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	172.16.0.0/21	10.0.35.5			0	400i
	i	10.0.46.6	100	0	400i	
*>	192.168.0.0/21	10.0.35.5			0	400i

.....

<R6>display bgp routing-table

BGP Local router ID is 10.0.6.6

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 9

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>I	172.16.0.0/21	10.0.56.5	100		0	i
*>	192.168.0.0/21	0.0.0.0	0		0	i

.....

可以看到，R1、R3、R6只接收到R5通告的聚合路由，没有接收到明细路由。

6．使用Suppress-Policy关键字控制路由聚合

Aggregate 命令支持抑制全部明细路由，也可以支持仅抑制部分明细路由。如果需要R5将172.16.2.0/24、172.16.4.0/24、172.16.6.0/24这几条明细路由进行抑制，而将其他的明细路由和聚合路由通告出去，则网络管理员可以通过使用Suppress-Policy关键字配合Route-Policy来实现。前缀列表用来匹配哪些路由需要被抑制，Route-Policy用来调用前缀列表，配合Suppress-Policy实现最终需求。

```
[R5]ip ip-prefix sup_policy permit 172.16.2.0 24
```

```
[R5]ip ip-prefix sup_policy permit 172.16.4.0 24
```

```
[R5]ip ip-prefix sup_policy permit 172.16.6.0 24
```

```
[R5]route-policy sup_policy permit node 10
```

```
[R5-route-policy]if-match ip-prefix sup_policy
```

```
[R5-route-policy]bgp 400
```

```
[R5-bgp]undo peer 10.0.35.3 route-policy no-adver export
```

```
[R5-bgp]aggregate 172.16.0.0 21 suppress-policy sup_policy
```

配置完成后，可在R1、R2、R3、R4、R6上查看BGP路由表，下面
仅以R1、R3、R6为例。

```
<R1>display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 13
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	172.16.0.0/21	10.0.13.3			0	300 400i
*		10.0.12.2		0	200 300	400i
*>	172.16.1.0/24	10.0.13.3			0	300 400i
*>	172.16.3.0/24	10.0.13.3			0	300 400i
*>	172.16.5.0/24	10.0.13.3			0	300 400i
*>	172.16.7.0/24	10.0.13.3			0	300 400i
*>	192.168.0.0/21	10.0.13.3			0	300 400i

```
.....
```

```
<R3>display bgp routing-table
```

```
BGP Local router ID is 10.0.3.3
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

Total Number of Routes: 15

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	172.16.0.0/21	10.0.35.5			0	400i
i		10.0.46.6	100	0		400i
*>	172.16.1.0/24	10.0.35.5	0		0	400i
*>	172.16.3.0/24	10.0.35.5	0		0	400i
*>	172.16.5.0/24	10.0.35.5	0		0	400i
*>	172.16.7.0/24	10.0.35.5	0		0	400i
*>	192.168.0.0/21	10.0.35.5			0	400i

.....

<R6>display bgp routing-table

BGP Local router ID is 10.0.6.6

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 13

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	172.16.0.0/21	10.0.56.5	100		0	i
*>i	172.16.1.0/24	10.0.56.5	0	100	0	i
*>i	172.16.3.0/24	10.0.56.5	0	100	0	i
*>i	172.16.5.0/24	10.0.56.5	0	100	0	i
*>i	172.16.7.0/24	10.0.56.5	0	100	0	i
*>	192.168.0.0/21	0.0.0.0	0		0	i

.....

可以看到 , R1、 R3、 R6 的 BGP 表中都没有 172.16.2.0/24、
172.16.4.0/24、 172.16.6.0/24这几条被抑制了的明细路由。

7. 使用Attribute-Policy关键字控制路由聚合

关键字Attribute-Policy可用来设置聚合路由的属性。例如，在上述步骤中，R5通告的是属性没有经过任何修改的路由，其聚合路由的Origin属性是i。现在，通过Attribute-Policy将聚合路由的属性修改成Incomplete，在BGP路由表中显示为“?”。

```
[R5]route-policy att_policy permit node 10
```

```
[R5-route-policy]apply origin incomplete
```

```
[R5-route-policy]bgp 400
```

```
[R5-bgp]aggregate 172.16.0.0 255.255.248.0 attribute-policy att_policy
```

配置完成后，在R1上查看BGP路由表。

```
<R1>display bgp routing-table
```

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 18

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	172.16.0.0/21	10.0.13.3			0	300 400?
*		10.0.12.2		0	200 300 400?	
*>	172.16.1.0/24	10.0.13.3			0	300 400i

.....

可以看到，R1的BGP路由表中的聚合路由的Origin属性变成了“?”。

思考

在默认情况下，华为路由设备上的BGP自动路由聚合功能是开启的吗？

3.5 BGP 路径选择——Preferred Value

原理概述

当一台BGP路由器中存在多条去往同一目标网络的BGP路由时，BGP协议会对这些BGP路由的属性进行比较，以确定去往该目标网络的最优BGP路由，然后将该最优BGP路由与去往同一目标网络的其他协议路由进行比较，从而决定是否将该最优BGP路由放进IP路由表中。注意，路由器最终是根据IP路由表进行实际报文转发的。在对BGP路由属性进行比较时，BGP会遵循一定的先后次序进行比较，直到确定出一条最优路由为止。在BGP路由属性的比较过程中，首先要比较的就是路由信息首选值Preferred Value，也简称为PrefVal。

路由信息的首选值Preferred Value 的取值范围是0 ~ 65535，取值越大，优先级越高。缺省情况下，Preferred Value 取值为0；通过修改Preferred Value 的值，可以很方便地实现对路径选择的控制。Preferred Value 属性不会发送给任何BGP 邻居，仅作为本地路由器用来选择最佳BGP路径之用。

实验目的

- 理解BGP 路由信息首选值Preferred Value 的作用
- 掌握修改Preferred Value 属性的方法
- 掌握通过修改Preferred Value 属性来实现流量分担的方法

实验内容

实验拓扑如图3-5所示，实验编址如表3-5所示。本实验包含了4台路由器，R1、R2、R3 属于AS 100，R4属于AS 200。R1、R2、R3 之间运行RIPv2 协议，同时，所有路由器都运行BGP 协议，并通过各自的Loopback 0 接口建立BGP 邻居关系。R4 通告自己的Loopback 1 和Loopback 2 两个接口所在的网络到BGP 进程中，R3 通告自己的

Loopback 1接口所在的网络到BGP 进程中。最后，通过修改Preferred Value 的值，使得R3 的 Loopback 1 接口去往R4的 Loopback 1 和 Loopback 2 接口的报文分别通过R1 与R2进行转发，从而实现流量分担，并互为备份。

实验拓扑

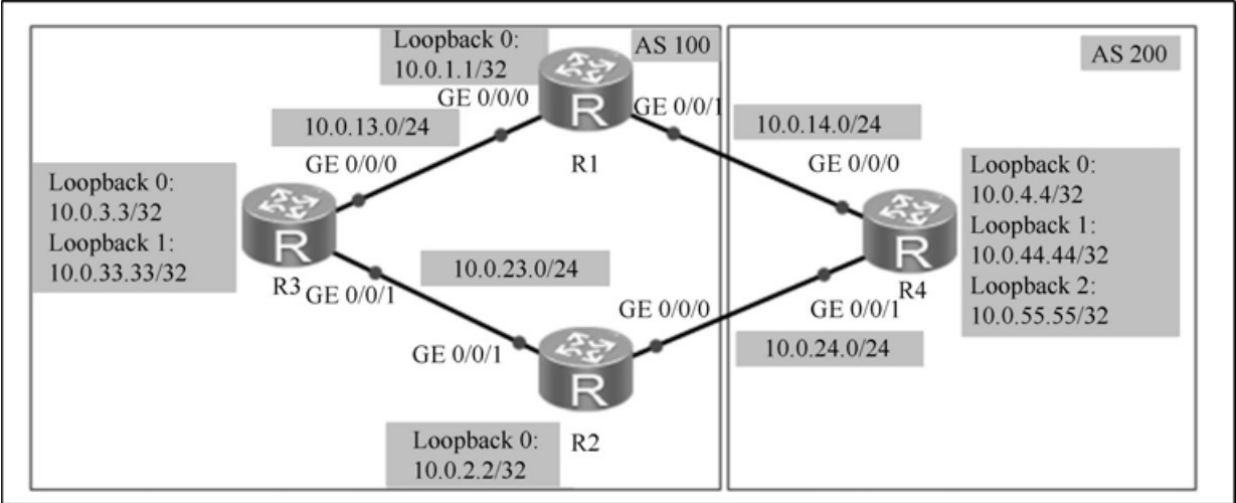


图3-5 BGP路径选择-Preferred Value

实验编址表

表3-5 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.13.1	255.255.255.0	N/A
	GE 0/0/1	10.0.14.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.24.2	255.255.255.0	N/A
	GE 0/0/1	10.0.23.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/0	10.0.13.3	255.255.255.0	N/A
	GE 0/0/1	10.0.23.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
	Loopback 1	10.0.33.33	255.255.255.255	N/A
R4(AR2220)	GE 0/0/0	10.0.14.4	255.255.255.0	N/A
	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
	Loopback 1	10.0.44.44	255.255.255.255	N/A
	Loopback 2	10.0.55.55	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-5和表3-5进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 10.0.13.3
```

```
PING 10.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=10 ms
```

```
--- 10.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 10/10/10 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置IGP和BGP路由协议

在AS 100 内采用RIPv2 协议配置IGP。

```
[R1]rip
```

```
[R1-rip-1]version 2
```

```
[R1-rip-1]network 10.0.0.0
```

```
[R2]rip
```

```
[R2-rip-1]version 2
```

```
[R2-rip-1]network 10.0.0.0
```

```
[R3]rip
```

```
[R3-rip-1]version 2
```

```
[R3-rip-1]network 10.0.0.0
```

在R1、R2、R4 上配置静态路由，保证R1的Loopback 0 接口和R4的Loopback 0接口之间，以及R2的Loopback 0 接口和R4的Loopback 0 接口之间能建立TCP 会话。


```
[R1]ip route-static 10.0.4.4 255.255.255.255 10.0.14.4
```

```
[R2]ip route-static 10.0.4.4 255.255.255.255 10.0.24.4
```

```
[R4]ip route-static 10.0.1.1 255.255.255.255 10.0.14.1
```

```
[R4]ip route-static 10.0.2.2 255.255.255.255 10.0.24.2
```

将每台路由器的Loopback 0接口IP地址作为自己的Router-ID，并且都采用Loopback 0接口来建立BGP邻居关系。R3通告自己的Loopback 1接口所在的网络到BGP进程中，R4 通告自己的Loopback 1 和Loopback 2 接口所在的网络到BGP 进程中。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.2.2 as-number 100
```

```
[R1-bgp]peer 10.0.2.2 connect-interface LoopBack0
```

```
[R1-bgp]peer 10.0.2.2 next-hop-local
```

```
[R1-bgp]peer 10.0.3.3 as-number 100
```

```
[R1-bgp]peer 10.0.3.3 connect-interface LoopBack0
```

```
[R1-bgp]peer 10.0.3.3 next-hop-local
```

```
[R1-bgp]peer 10.0.4.4 as-number 200
```

```
[R1-bgp]peer 10.0.4.4 ebgp-max-hop
```

```
[R1-bgp]peer 10.0.4.4 connect-interface LoopBack0
```

```
[R2]bgp 100
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.1.1 as-number 100
```

```
[R2-bgp]peer 10.0.1.1 connect-interface LoopBack0
```

```
[R2-bgp]peer 10.0.1.1 next-hop-local
```

```
[R2-bgp]peer 10.0.3.3 as-number 100
```

```
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack0
```

```
[R2-bgp]peer 10.0.3.3 next-hop-local
```

```

[R2-bgp]peer 10.0.4.4 as-number 200
[R2-bgp]peer 10.0.4.4 ebgp-max-hop
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack0
[R3]bgp 100
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.1.1 as-number 100
[R3-bgp]peer 10.0.1.1 connect-interface LoopBack0
[R3-bgp]peer 10.0.2.2 as-number 100
[R3-bgp]peer 10.0.2.2 connect-interface LoopBack0
[R3-bgp]network 10.0.33.33 255.255.255.255
[R4]bgp 200
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.1.1 as-number 100
[R4-bgp]peer 10.0.1.1 ebgp-max-hop
[R4-bgp]peer 10.0.1.1 connect-interface LoopBack0
[R4-bgp]peer 10.0.2.2 as-number 100
[R4-bgp]peer 10.0.2.2 ebgp-max-hop
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack0
[R4-bgp]network 10.0.44.44 255.255.255.255
[R4-bgp]network 10.0.55.55 255.255.255.255

```

配置完成后，在R1、R2上查看BGP邻居关系的建立情况。

```
[R1]display bgp peer
```

BGP local router ID : 10.0.1.1

Local AS number : 100

Total number of peers : 3 Peers in established state : 3

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State

PrefRcv

10.0.2.2	4	100	4	5	0	00:01:29
Established	2					
10.0.3.3	4	100	4	4	0	00:01:36
Established	1					
10.0.4.4	4	200	5	5	0	00:01:23
Established	2					

[R2]display bgp peer

BGP local router ID : 10.0.2.2

Local AS number : 100

Total number of peers : 3 Peers in established state : 3

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.1.1	4	100	5	5	0	00:02:34
Established	2					
10.0.3.3	4	100	5	5	0	00:02:34
Established	1					
10.0.4.4	4	200	6	6	0	00:02:28
Established	2					

可以看到，R1与R4、R2与R4的EBGP邻居关系已经成功建立，R1、R2、R3的IBGP邻居关系也已经成功建立，接下来查看R1、R2、R3的BGP路由表。

[R1]display bgp routing-table

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.33.33/32	10.0.3.3	0	100	0	i
*>	10.0.44.44/32	10.0.4.4	0		0	200i
* i		10.0.2.2	0	100	0	200i
*>	10.0.55.55/32	10.0.4.4	0		0	200i
* i		10.0.2.2	0	100	0	200i

[R2]display bgp routing-table

BGP Local router ID is 10.0.2.2

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.33.33/32	10.0.3.3	0	100	0	i
*>	10.0.44.44/32	10.0.4.4	0		0	200i
* i		10.0.1.1	0	100	0	200i
*>	10.0.55.55/32	10.0.4.4	0		0	200i
* i		10.0.1.1	0	100	0	200i

[R3]display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S – Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.33.33/32	0.0.0.0	0		0	i
*>i	10.0.44.44/32	10.0.1.1	0	100	0	200i

```

* i          10.0.2.2  0      100    0      200i
*>i 10.0.55.55/32      10.0.1.1  0      100    0      200i
* i          10.0.2.2  0      100    0      200i

```

可以看到，在 R3 的 BGP 路由表中，去往 10.0.44.44/32 和 10.0.55.55/32 网络的路由条目各有两条，但是最终 R3 优选的都是下一跳为 10.0.1.1 的路由。这两条 BGP 路由信息的下一跳不同，但路由信息首选值 PrefVal、本地优先级 LocPrf、路由生成方式、AS_Path 属性、Origin 属性、MED 属性、BGP 对等体类型（IBGP 邻居或 EBGP 邻居）等都是相同的，所以最终 BGP 选择了 Router-ID 较小的路由器 R1 发布的路由作为最佳路由。R1 和 R2 的 BGP 路由表中，去往 10.0.44.44/32 和 10.0.55.55/32 网络的路由也各有两条，但是最终优选的都是下一跳为 10.0.4.4 的路由。这两条 BGP 路由信息的下一跳不同，但路由信息首选值 PrefVal、本地优先级 LocPrf、路由生成方式、AS_Path 属性、Origin 属性、MED 属性等都是相同的，但是 BGP 对等体类型不同，所以最终 R1 和 R2 都选择了从 EBGP 邻居 R4 那里接收到的路由作为最佳路由。另外，无论是通过 EBGP 邻居还是 IBGP 邻居学习到的 BGP 路由条目，在 BGP 路由表中 PrefVal 都显示为缺省值 0。

3. 修改 Preferred Value

目前，R3 去往网络 10.0.44.44/32 和 10.0.55.55/32 的最佳下一跳都为 10.0.1.1，即 R1。我们可以在 R3 上使用 tracert 命令验证从 10.0.33.33/32 去往 10.0.44.44/32 和 10.0.55.55/32 的报文所经过的路径。

```
[R3]tracert -a 10.0.33.33 10.0.44.44
```

```
tracert to 10.0.44.44(10.0.44.44), max hops: 30 ,packet length:
40,press CTRL_C to break
```

```
1 10.0.13.1 10 ms 20 ms 10 ms
```

```
2 10.0.14.4 10 ms 30 ms 10 ms
```

```
[R3]tracert -a 10.0.33.33 10.0.55.55
```

traceroute to 10.0.55.55(10.0.55.55), max hops: 30 ,packet length: 40,press CTRL_C to break

1 10.0.13.1 10 ms 20 ms 10 ms

2 10.0.14.4 10 ms 30 ms 10 ms

可以看到，R3 的确是通过R1 去往AS 200 的，R2 没有分担任何流量。接下来，在R3 上进行Preferred Value 值的修改，使得R2 分担从R3 去往10.0.55.55/32 的流量。

```
[R3]ip ip-prefix 1 index 10 permit 10.0.55.55 32
```

```
[R3]route-policy 1 permit node 10
```

```
[R3-route-policy]if-match ip-prefix 1
```

```
[R3-route-policy]apply preferred-value 100
```

使用命令route-policy 1 permit node 20允许其他路由不做修改而被接收。

```
[R3]route-policy 1 permit node 20
```

在BGP视图下调用路由策略。

```
[R3-route-policy]bgp 100
```

```
[R3-bgp]peer 10.0.2.2 route-policy 1 import
```

上述配置完成后，查看R3的BGP路由表。

```
[R3]display bgp routing-table
```

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.33.33/32	0.0.0.0	0		0	i
*>i	10.0.44.44/32	10.0.1.1	0	100	0	200i

```

* i          10.0.2.2  0      100    0      200i
*>i 10.0.55.55/32      10.0.2.2  0      100    100    200i
* i          10.0.1.1  0      100    0      200i

```

可以发现，现在R3的BGP路由表中去往10.0.55.55/32的优选下一跳为10.0.2.2，即R2，PrefVal 的值为100，而前往10.0.44.44/32 的优选下一跳仍为10.0.1.1，即 R1。使用tracert命令测试报文转发的路径。

```
[R3]tracert -a 10.0.33.33 10.0.44.44
```

```

traceroute to 10.0.44.44(10.0.44.44), max hops: 30 ,packet length:
40,press CTRL_C to break

```

```
1 10.0.13.1 10 ms 10 ms 20 ms
```

```
2 10.0.14.4 10 ms 10 ms 30 ms
```

```
[R3]tracert -a 10.0.33.33 10.0.55.55
```

```

traceroute to 10.0.55.55(10.0.55.55), max hops: 30 ,packet length:
40,press

```

```
CTRL_C to break
```

```
1 10.0.23.2 10 ms 20 ms 1 ms
```

```
2 10.0.24.4 20 ms 30 ms 10 ms
```

可以看到，R3 的 Loopback1 接口发送数据去往 10.0.55.55 时，下一跳路由器为10.0.2.2，即R2，而发送数据去往10.0.44.44时，下一跳路由器为10.0.1.1，即R1，这样便达到了流量分担的目的。

在R1和R2上查看BGP路由表。

```
[R1]display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 5
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.33.33/32	10.0.3.3	0	100	0	i
*>	10.0.44.44/32	10.0.4.4	0		0	200i
* i		10.0.2.2	0	100	0	200i
*>	10.0.55.55/32	10.0.4.4	0		0	200i
* i		10.0.2.2	0	100	0	200i

[R2]display bgp routing-table

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.33.33/32	10.0.3.3	0	100	0	i
*>	10.0.44.44/32	10.0.4.4	0		0	200i
* i		10.0.1.1	0	100	0	200i
*>	10.0.55.55/32	10.0.4.4	0		0	200i
* i		10.0.1.1	0	100	0	200i

可以看到，R1 和R2 的BGP 路由表在修改了R3 上的Preferred Value 值之后没有发生变化，说明了Preferred Value 值只是作为本地路由器用来选择最佳BGP 路由之用，并不会传递给任何BGP邻居。

思考

本实验中，能否在R4 上配置路由策略对Preferred Value 进行修改，从而实现R3 经由R1去往10.0.44.44/32，经由R2去往10.0.55.55/32这样的负载分担效果呢？

3.6 BGP 路径选择——Local Preference

原理概述

当一台BGP路由器中存在多条去往同一目标网络的BGP路由时，BGP协议会对这些BGP路由的属性进行比较，以确定去往该目标网络的最优BGP路由。BGP首先比较的是路由信息的首选值（PrefVal），如果 PrefVal 相同，就会比较本地优先级（Local Preference，缩写为 LocPrf）属性。

Local Preference属性可以用于选择流量离开AS时的最佳路由，也就是控制流量从哪个出口离开AS。当BGP路由器通过不同的IBGP对等体接收到目标网络相同但下一跳不同的多条路由时，将优先选择Local Preference 值较高的路由。

Local Preference只在IBGP对等体之间进行通告，EBGP对等体之间传递BGP路由时，不携带Local Preference属性。默认情况下，本地使用network命令通告或者import命令引入到BGP中的路由的Local Preference值为空。当从IBGP对等体接收到的路由的Local Preference值为空时，接收路由器会使用100作为这条路由的Local Preference默认值，当从IBGP对等体接收到的路由的Local Preference值不为空时，接收路由器默认不做修改。

Local Preference 值是一个32比特的整数，取值范围为0 ~ 4294967295。

实验目的

- 理解Local Preference 属性的概念与作用
- 掌握修改Local Preference 属性的方法

实验内容

实验拓扑如图3-6 所示，实验编址如表3-6 所示。AS 100 为运营商网络，AS 200 为公司网络，R1 的Loopback 0与Loopback 1 接口用来分别模拟向公司提供服务的服务器A 和服务B，R4的Loopback 1 接口用来模拟公司的内部网络。R1 属于AS 100，R2、R3 和R4 属于AS 200，R1 与R2 和R3 采用直连物理接口建立EBGP 邻居关系，R2、R3、R4 之间采用各自的Loopback 0接口来建立IBGP邻居关系，同时R2、R3、R4 运行OSPF；通过修改Local Preference值，使得公司的内部网络访问服务器A时将使用R2作为出口，访问服务器B时将使用R3作为出口。

实验拓扑

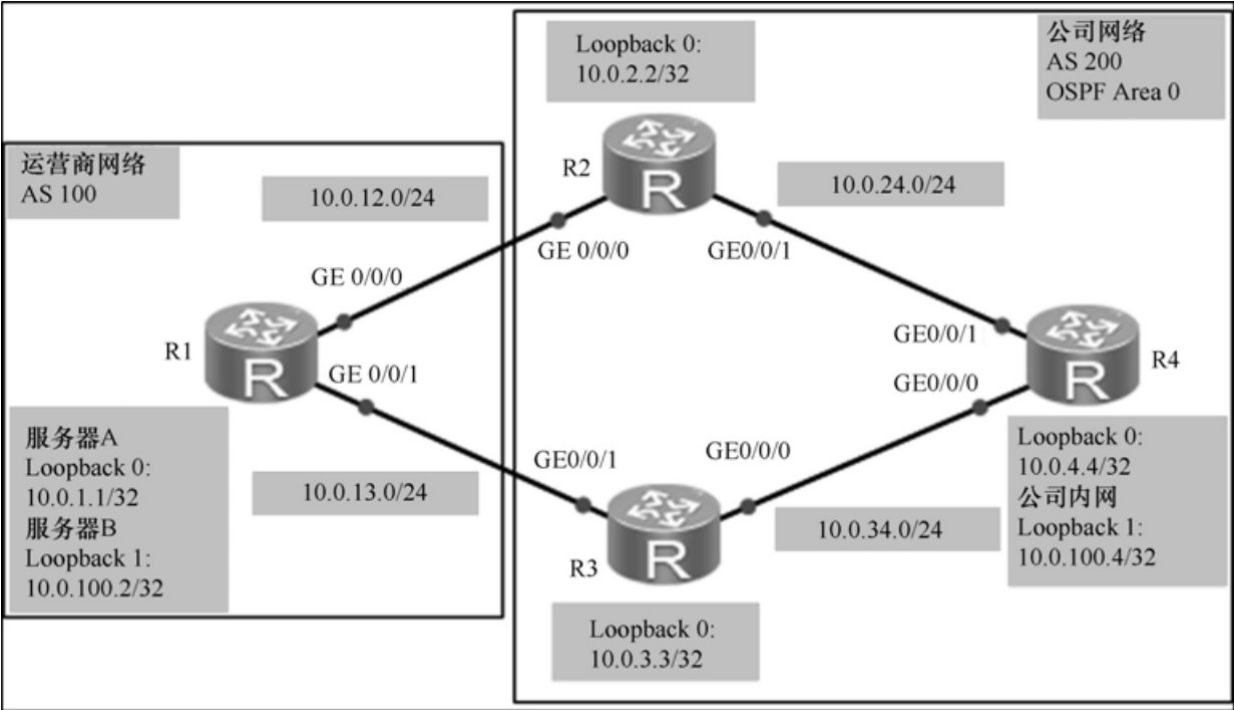


图3-6 BGP路径选择-Local Preference

实验编址表

表3-6 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR3260)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	10.0.100.2	255.255.255.255	N/A
R2(AR3260)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.24.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR3260)	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR3260)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
	Loopback 1	10.0.100.4	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-6和表3-6进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=80 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 80/80/80 ms
```

其余直连网段的连通性测试过程在此省略。

2．完成OSPF和BGP协议的基本配置

对AS 200 中的路由器进行OSPF 协议配置，所有路由器都属于区域0，且每台路由器都使用自己的Loopback 0 接口的IP 地址作为Router-ID。

```
[R2]ospf 1 router-id 10.0.2.2
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
[R3]ospf 1 router-id 10.0.3.3
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R4]ospf 1 router-id 10.0.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
```

配置完成后，在R4上使用display ospf peer 命令查看OSPF 邻居关系。

```
[R4]display ospf peer
                OSPF Process 1 with Router ID 10.0.4.4
                        Neighbors
Area 0.0.0.0 interface 10.0.34.4(GigabitEthernet0/0/0)'s neighbors
  Router ID: 10.0.3.3    Address: 10.0.34.3
    State: Full Mode:Nbr is Slave Priority: 1
    DR: 10.0.34.4 BDR: 10.0.34.3 MTU: 0
    Dead timer due in 40 sec
    Retrans timer interval: 5
    Neighbor is up for 00:02:04
    Authentication Sequence: [ 0 ]
                        Neighbors
```

Area 0.0.0.0 interface 10.0.24.4(GigabitEthernet0/0/1)'s neighbors

Router ID: 10.0.2.2 Address: 10.0.24.2

State: Full Mode:Nbr is Slave Priority: 1

DR: 10.0.24.4 BDR: 10.0.24.2 MTU: 0

Dead timer due in 34 sec

Retrans timer interval: 5

Neighbor is up for 00:02:22

Authentication Sequence: [0]

可以看到，邻居状态均为Full，说明R4与R2和R3已经成功建立了OSPF邻接关系。

接下来，在R1、R2、R3、R4上配置BGP协议。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 200
```

```
[R1-bgp]peer 10.0.13.3 as-number 200
```

```
[R1-bgp]network 10.0.1.1 32
```

```
[R1-bgp]network 10.0.100.2 32
```

```
[R2]bgp 200
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.3.3 as-number 200
```

```
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```

```
[R2-bgp]peer 10.0.3.3 next-hop-local
```

```
[R2-bgp]peer 10.0.4.4 as-number 200
```

```
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack 0
```

```
[R2-bgp]peer 10.0.4.4 next-hop-local
```

```
[R3]bgp 200
```

```

[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.13.1 as-number 100
[R3-bgp]peer 10.0.2.2 as-number 200
[R3-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R3-bgp]peer 10.0.2.2 next-hop-local
[R3-bgp]peer 10.0.4.4 as-number 200
[R3-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R3-bgp]peer 10.0.4.4 next-hop-local
[R4]bgp 200
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.2.2 as-number 200
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R4-bgp]peer 10.0.3.3 as-number 200
[R4-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R4-bgp]network 10.0.100.4 32

```

配置完成后，在R1 上使用display bgp peer 命令查看BGP 邻居关系。

```
[R1]display bgp peer
```

BGP local router ID : 10.0.1.1

Local AS number : 100

Total number of peers : 2 Peers in established state : 2

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.12.2	4	200	4	7	0	00:01:37
Established	1					
10.0.13.3	4	200	4	6	0	00:01:25
Established	1					

可以看到，邻居状态均为Established，表明R1与R2和R3已经成功建立了EBGP邻居关系。R2、R3、R4之间的IBGP邻居关系的查看在此省略。

3．观察BGP 路由信息的Local Preference 属性
在R4上查看BGP路由表。

```
[R4]display bgp routing-table
```

```
BGP Local router ID is 10.0.4.4
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 5
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.2.2	0	100	0	100i
* i		10.0.3.3	0	100	0	100i
*>i	10.0.100.2/32	10.0.2.2	0	100	0	100i
* i		10.0.3.3	0	100	0	100i
*>	10.0.100.4/32	0.0.0.0	0		0	i

可以看到，R4的BGP路由表中存在两条去往10.0.1.1/32网络的路由，下一跳分别为R2与R3，以及两条去往10.0.100.2/32网络的路由，下一跳还是分别为R2与R3，这些路由信息的Local Preference 值均为默认值100。R4 自己通告的10.0.100.4/32 网络的路由信息的Local Preference 值为空。

在R2上查看BGP路由表。

```
[R2]display bgp routing-table
```

```
BGP Local router ID is 10.0.2.2
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.12.1	0		0	100i
* i		10.0.3.3	0	100	0	100i
*>	10.0.100.2/32	10.0.12.1	0		0	100i
* i		10.0.3.3	0	100	0	100i
*>i	10.0.100.4/32	10.0.4.4	0	100	0	i

可以看到，R2从EBGP对等体R1接收到的10.0.1.1/32的路由信息的Local Preference值为空，而从IBGP对等体R3接收到的10.0.1.1/32的路由信息的Local Preference值为100。

由此可见，Local Preference属性不会通告给EBGP对等体，仅在AS内传递时才会通告。当从IBGP对等体接收到的路由的Local Preference值为空时，接收路由器会使用100作为这条路由的Local Preference默认值。

4. 修改Local Preference 值

从上面的实验内容已经看到，当从IBGP对等体接收到的路由的Local Preference值为空时，接收路由器会使用100作为这条路由的Local Preference默认值。接下来，我们将对Local Preference值进行修改，使得公司的内部网络发送数据到服务器B时以R3为出口。

先在R3上使用display default-parameter bgp命令查看BGP协议的默认参数。

```
[R3]display default-parameter bgp
  BGPversion                : 4
.....
  IBGP route-update-interval : 15s
  Default local-preference    : 100
```


Default MED : 0

.....

可以看到，R3 上BGP 的Local Preference 默认值为100。

在R4上查看BGP路由表。

```
[R4]display bgp routing-table
```

BGP Local router ID is 10.0.4.4

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.2.2	0	100	0	100i
* i		10.0.3.3	0	100	0	100i
*>i	10.0.100.2/32	10.0.2.2	0	100	0	100i
* i		10.0.3.3	0	100	0	100i
*>	10.0.100.4/32	0.0.0.0	0		0	i

可以看到，对于目的网络10.0.100.2/32，R4选择了下一跳为10.0.2.2的路由作为最佳路由，这是由于R4在对下一跳为10.0.2.2与10.0.3.3的路由信息进行比较时，二者的Preferred Value 属性、Local Preference 属性、路由生成方式、AS_Path 属性、Origin 属性、MED属性、BGP对等体类型等都是相同的，于是R4最终选择了Router-ID较小的路由器R2 发布的路由作为最佳路由。

为了使公司内部网络去往10.0.100.2/32的数据使用R3作为出口，可以在R3的BGP视图下使用 default local-preference 命令将 R3 的 BGP 默认 Local Preference 值修改为200。

```
[R3]bgp 200
```

```
[R3-bgp]default local-preference 200
```

修改之后，在R4和R2上查看BGP路由表。

```
[R4]display bgp routing-table
```

BGP Local router ID is 10.0.4.4

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.3.3	0	200	0	100i
*>i	10.0.100.2/32	10.0.3.3	0	200	0	100i
*>	10.0.100.4/32	0.0.0.0	0		0	i

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.12.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.3.3	0	200	0	100i
*		10.0.12.1	0	0	100i	
*>i	10.0.100.2/32	10.0.3.3	0	200	0	100i
*		10.0.12.1	0	0	100i	
*>i	10.0.100.4/32	10.0.4.4	0	100	0	i

从上面两个BGP路由表可以看到，R4去往10.0.1.1/32与10.0.100.2/32网络时，使用的是下一跳为R3（10.0.3.3）的路由，Local Preference 值为200；R4的BGP路由表中不再有下一跳为R2（10.0.2.2）的去往10.0.1.1/32与10.0.100.2/32的路由。

在R3上修改了Local Preference值后，R1把关于10.0.1.1/32与10.0.100.2/32的路由传递给EBGP对等体R2时，Local Preference值为空，R3把关于10.0.1.1/32与10.0.100.2/32的路由传递给IBGP对等体R2时，Local Preference值为200，于是R2会选择下一跳为R3（10.0.3.3）的路由作为自己去往10.0.1.1/32与10.0.100.2/32的最佳路由。BGP协议在向BGP对等体传递路由时只传递最佳路由，同时，由于IBGP的防环机制，BGP路由器不会将从IBGP对等体那里学到的路由再传递给别的IBGP对等体，因此，R2就不会再向R4传递关于10.0.1.1/32与10.0.100.2/32的路由了，最后的结果是，R4的BGP路由表中最终只存在去往10.0.1.1/32与10.0.100.2/32的下一跳为R3（10.0.3.3）的路由。

现在，在R4上使用tracert命令验证从10.0.100.4/32去往10.0.100.2/32的报文所经过的路径。

```
<R4>tracert -a 10.0.100.4 10.0.100.2
tracert to 10.0.100.2(10.0.100.2), max hops: 30 ,packet length:
40,press CTRL_C to break
 1 10.0.34.3 10 ms 10 ms 10 ms
 2 10.0.13.1 30 ms 10 ms 10 ms
```

可以看到，公司内部网络发送数据到服务器B（10.0.100.2）时是以R3为出口的。

5．使用Route-Policy修改Local Preference 值

在R4上使用tracert命令验证从10.0.100.4/32去往10.0.1.1/32的报文所经过的路径。

```
<R4>tracert -a 10.0.100.4 10.0.1.1
tracert to 10.0.1.1(10.0.1.1), max hops: 30 ,packet length: 40,press
CTRL_C to break
 1 10.0.34.3 10 ms 10 ms 20 ms
 2 10.0.13.1 10 ms 30 ms 10 ms
```

可以看到，公司内部网络发送数据到服务器A时也是以R3为出口的。为实现流量分担，现在使用Route-Policy对特定路由的Local Preference值进行修改，从而使得公司内部网络去往服务器A（10.0.1.1）的报文选择R2为出口。

```
[R2]ip ip-prefix 1 permit 10.0.1.1 32
```

```
[R2]route-policy 1 permit node 10
```

```
[R2-route-policy]if-match ip-prefix 1
```

```
[R2-route-policy]apply local-preference 500
```

使用命令route-policy 1 permit node 20允许其他路由不做修改被接收。

```
[R2]route-policy 1 permit node 20
```

在BGP视图下使用peer 10.0.12.1 route-policy 1 import命令在R2接收R1所传递的路由信息的import方向上调用路由策略。

```
[R2]bgp 200
```

```
[R2-bgp]peer 10.0.12.1 route-policy 1 import
```

配置完成后，在R2上查看BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPr	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.12.1	0	500	0	100i
*>i	10.0.100.2/32	10.0.3.3	0	200	0	100i
*		10.0.12.1	0	0	100i	
*>i	10.0.100.4/32	10.0.4.4	0	100	0	i

可以看到，在R2路由表中，对于目的网络10.0.1.1/32，只存在一条由EBGP对等体R1发送的、Local Preference 值被R2 修改为500 的路由。

在R4上查看BGP路由表。

```
[R4]display bgp routing-table
```

```
BGP Local router ID is 10.0.4.4
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 3
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.2.2	0	500	0	100i
*>i	10.0.100.2/32	10.0.3.3	0	200	0	100i
*>	10.0.100.4/32	0.0.0.0	0		0	i

可以看到，目前R4去往10.0.1.1/32和10.0.100.2/32的路由信息都只有一条，下一跳分别为 R2（10.0.2.2）和 R3（10.0.3.3）。R4 在接收从IBGP 对等体 R2 发来的关于10.0.1.1/32 这条路由时，其Local Preference 值未做任何修改，保持为500。

在R4上使用tracert命令验证从10.0.100.4/32去往10.0.1.1/32的报文所经过的路径。

```
<R4>tracert -a 10.0.100.4 10.0.1.1
```

```
traceroute to 10.0.100.1(10.0.100.1), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.24.2 10 ms 10 ms 10 ms
```

```
2 10.0.12.1 20 ms 10 ms 10 ms
```

可以看到，现在公司内部网络去往服务器A的报文以R2为出口。至此，流量分担的网络需求得到了实现。

思考

对于聚合后的BGP 路由，Local Preference 会有怎样的变化？

3.7 BGP 路径选择——Next Hop

原理概述

当一台BGP路由器中存在多条去往同一目标网络的BGP路由时，BGP协议会对这些BGP路由的属性进行比较，以确定出去往该目标网络的最优BGP路由，然后将该最优BGP路由与去往同一目标网络的其他协议路由进行比较，从而决定是否将该最优 BGP 路由放进IP路由表中。BGP路由属性的比较顺序为Preferred Value属性、Local Preference属性、路由生成方式、AS_Path属性、Origin属性、MED属性、BGP对等体类型等，如果前面这些路由属性都完全相同或在比较选择的过程中可被忽略，则将比较路由的 Next Hop属性。

Next Hop 属性记录了去往目标网络所对应的下一跳IP 地址。BGP 在比较Next Hop属性时，会优选去往Next Hop 属性中IP 地址的IGP 开销最小的路由。需要注意的是，如果一条BGP 路由的Next Hop 属性中的IP 地址不可达，则该条路由在BGP 路由表中不会被标记为可用路由，从而也就根本无法参与BGP路由协议的选路过程。

BGP 路由器在发布路由给EBGP 对等体时，该路由的Next Hop 的IP 地址会被自动修改，但发布路由给IBGP 对等体时，Next Hop的IP 地址不会被自动修改。为了满足不同网络环境的需求，当路由器发布路由给IBGP 对等体时，也可以手动修改Next Hop 的IP地址。

实验目的

- 理解Next Hop 属性的概念与作用
- 掌握修改Next Hop 属性的方法

● 理解Next Hop 属性对BGP 路由协议选路的影响

实验内容

实验拓扑如图3-7 所示，实验编址如表 3-7 所示。R1 属于 AS 100，R2、R3 和 R4属于AS 200。R1 的Loopback 1 接口模拟客户所在的网络，R4 的Loopback 1 接口模拟目标服务器所在的网络。所有的路由器都运行BGP，同时R2、R3和R4还运行OSPF。R1与R2和R3之间的EBGP邻居关系采用直连物理接口来建立，R2、R3、R4之间的IBGP邻居关系采用Loopback 0接口来建立。最终的目标是实现AS 100的客户与AS 200的服务器能够进行正常通信，并且不能出现非对称路由的现象。

实验拓扑

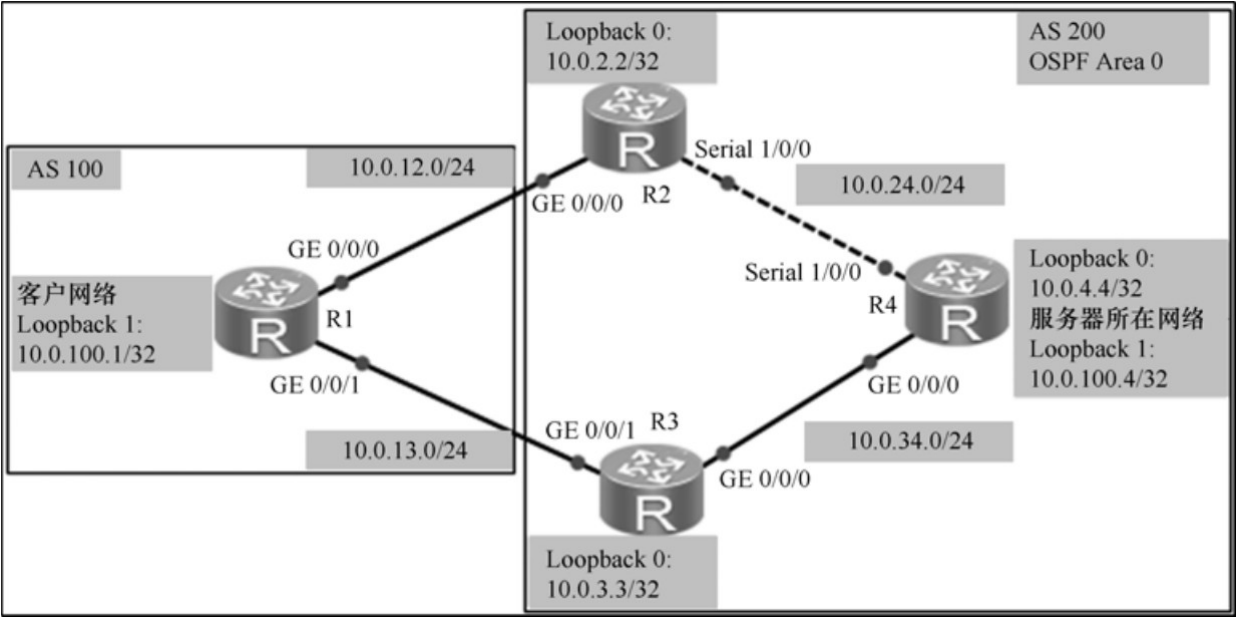


图3-7 BGP路径选择-Next Hop

实验编址表

表3-7 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR3260)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	Loopback 1	10.0.100.1	255.255.255.255	N/A
R2(AR3260)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	Serial 1/0/0	10.0.24.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A

(续表)

设备	接口	IP 地址	子网掩码	默认网关
R3(AR3260)	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR3260)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	Serial 1/0/0	10.0.24.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
	Loopback 1	10.0.100.4	255.255.255.255	N/A

实验步骤

1. 基本配置

根据图3-7和表3-7进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=100 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 100/100/100 ms
```

其余直连网段的连通性测试过程在此省略。

2. IGP和BGP路由协议配置

在AS 200 中的路由器上配置OSPF 协议，所有路由器都属于区域0，每台路由器都使用Loopback 0 接口的IP 地址作为Router-ID。


```
[R2]ospf 1 router-id 10.0.2.2
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
[R3]ospf 1 router-id 10.0.3.3
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R4]ospf 1 router-id 10.0.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
[R4-ospf-1-area-0.0.0.0]network 10.0.100.4 0.0.0.0
```

配置完成后，在R4上查看OSPF邻居关系。

```
[R4]display ospf peer
```

OSPF Process 1 with Router ID 10.0.4.4

Neighbors

Area 0.0.0.0 interface 10.0.34.4(GigabitEthernet0/0/0)'s neighbors

Router ID: 10.0.3.3 Address: 10.0.34.3

State: Full Mode:Nbr is Slave Priority: 1

DR: 10.0.34.4 BDR: 10.0.34.3 MTU: 0

Dead timer due in 34 sec

Retrans timer interval: 5

Neighbor is up for 00:00:27

Authentication Sequence: [0]

Neighbors

Area 0.0.0.0 interface 10.0.24.4(Serial1/0/0)'s neighbors

Router ID: 10.0.2.2 Address: 10.0.24.2

State: Full Mode:Nbr is Slave Priority: 1

DR: None BDR: None MTU: 0

Dead timer due in 36 sec

Retrans timer interval: 5

Neighbor is up for 00:01:14

Authentication Sequence: [0]

可以看到，邻居状态都为Full，表明R4与R2和R3均已成功建立了OSPF邻居关系。

下面进行BGP路由协议的配置。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 200
```

```
[R1-bgp]peer 10.0.13.3 as-number 200
```

```
[R1-bgp]network 10.0.100.1 255.255.255.255
```

```
[R2]bgp 200
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.3.3 as-number 200
```

```
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```

```
[R2-bgp]peer 10.0.4.4 as-number 200
```

```
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack 0
```

```
[R3]bgp 200
```

```
[R3-bgp]router-id 10.0.3.3
```

```
[R3-bgp]peer 10.0.13.1 as-number 100
```

```
[R3-bgp]peer 10.0.2.2 as-number 200
```

```

[R3-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R3-bgp]peer 10.0.4.4 as-number 200
[R3-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R4]bgp 200
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.2.2 as-number 200
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R4-bgp]peer 10.0.3.3 as-number 200
[R4-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R4-bgp]network 10.0.100.4 255.255.255.255

```

配置完成后，在R2上查看BGP邻居关系。其他路由器的BGP邻居关系的查看在此省略。

```
[R2]display bgp peer
```

```
BGP local router ID : 10.0.2.2
```

```
Local AS number : 200
```

```
Total number of peers : 3      Peers in established state : 3
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.3.3	4	200	8	9	0	00:05:17
Established	1					
10.0.4.4	4	200	5	7	0	00:02:24
Established	1					
10.0.12.1	4	100	71	64	0	00:54:28
Established	1					

可以看到，邻居状态均为Established，说明R2已经成功与R1、R3、R4建立起了BGP邻居关系。

3 . Next Hop 属性在路由传递过程中的变化情况

通过前面的步骤，网络的基本配置已经完成。接下来，在R1上测试R1的Loopback 1 接口与R4 的Loopback 1 接口之间的连通性。

```
<R1>ping -a 10.0.100.1 10.0.100.4
```

```
PING 10.0.100.4: 56 data bytes, press CTRL_C to break
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
--- 10.0.100.4 ping statistics ---
```

```
5 packet(s) transmitted
```

```
0 packet(s) received
```

```
100.00% packet loss
```

可以看到，客户网络并不能与服务器进行正常通信。

在R1上查看BGP路由表。

```
[R1]display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 3
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	0.0.0.0	0		0	i
*>	10.0.100.4/32	10.0.12.2			0	200i
*		10.0.13.3		0	200i	

可以看到，R1的BGP路由表中有两条去往10.0.100.4/32的路由信息，下一跳分别为R2与R3。R1 通告的10.0.100.1/32网络的Next Hop 为

0.0.0.0 , 即自己通告的BGP 路由信息的Next Hop 为0.0.0.0。

在R4上查看BGP路由表。

```
[R4]display bgp routing-table
```

BGP Local router ID is 10.0.4.4

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	10.0.100.1/32	10.0.12.1	0	100	0	100i
i		10.0.13.1	0	100	0	100i
*>	10.0.100.4/32	0.0.0.0	0		0	i

可以看到 , R4的BGP路由表中也有两条去往10.0.100.1/32网络的路由信息 , Next Hop 分别为10.0.12.1与10.0.13.1 , 但没有标记为可用 (valid) 。

在R4上查看IP路由表。

```
[R4]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
-----
                                Routing Tables: Public
                                Destinations : 15  Routes : 15

Destination/Mask    Proto    Pre  Cost    Flags  NextHop
-----
Interface
10.0.2.2/32         OSPF     10   48      D      10.0.24.2  Serial1/0/0
10.0.3.3/32         OSPF     10    1      D      10.0.34.3
GigabitEthernet0/0/0
```

10.0.4.4/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.24.0/24	Direct	0	0	D	10.0.24.4	Serial1/0/0
10.0.24.2/32	Direct	0	0	D	10.0.24.2	Serial1/0/0
10.0.24.4/32	Direct	0	0	D	127.0.0.1	Serial1/0/0
10.0.24.255/32	Direct	0	0	D	127.0.0.1	
Serial1/0/0						
10.0.34.0/24	Direct	0	0	D	10.0.34.4	
GigabitEthernet0/0/0						
10.0.34.4/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.34.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.100.4/32	Direct	0	0	D	127.0.0.1	
LoopBack1						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	
InLoopBack0						
127.0.0.1/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

可以发现，R4 的 IP 路由表中并没有去往 10.0.100.1/32 的路由信息，也没有去往10.0.12.1与10.0.13.1的路由信息。而在R4的BGP路由表中，虽有条去往10.0.100.1/32的路由信息，但没有标记为可用，说明R4认为这两条路由信息的下一跳都是不可达的。

在R2、R3上查看BGP路由表。

```
[R2]display bgp routing-table
```

```
BGP Local router ID is 10.0.2.2
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 3
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.12.1	0		0	100i
i		10.0.13.1	0	100	0	100i
*>i	10.0.100.4/32	10.0.4.4	0	100	0	i

```
[R3]display bgp routing-table
```

```
BGP Local router ID is 10.0.3.3
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 3
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.13.1	0		0	100i
i		10.0.12.1	0	100	0	100i
*>i	10.0.100.4/32	10.0.4.4	0	100	0	i

可以看到，R2的BGP路由表中有两条去往10.0.100.1/32的路由信息，其中Next Hop为10.0.12.1的路由信息标记为可用。根据前面的实验步骤得知，10.0.100.1/32路由在R1上的Next Hop为0.0.0.0，说明当10.0.100.1/32的路由信息在由R1传递至EBGP对等体R2的过程中，Next Hop属性会被自动修改为发送BGP报文的源地址，即10.0.12.1。而去往10.0.100.1/32的路由信息的Next Hop在R2与R4上均为10.0.12.1，说明10.0.100.1/32这条路由信息在由R2传递至IBGP对等体R4时，Next

Hop 属性不会自动被修改。R3上的现象与R2上的现象类似，这里不再赘述。

为了使R4的BGP路由表中去往10.0.100.1/32的路由信息标记为可用，并放进IP路由表中，必须使R4去往10.0.100.1/32的BGP路由信息中的Next Hop是可达的。实现这一要求的方法有两种：第一种方法是将EBGP对等体之间的链路通告进IGP网络；第二种方法是在R2和R3将路由信息传递给IBGP对等体R4时，使用发送BGP报文的源地址作为BGP路由的下一跳。在实际应用中，通常会使用第二种方法，本实验也将采用这种方法。

在R2 上使用 peer 10.0.4.4 next-hop-local 和 peer 10.0.3.3 next-hop-local 命令，使BGP路由信息传递给IBGP对等体R4和R3时，使用R2发送BGP报文的源地址作为BGP 路由的下一跳来代替原有的Next Hop。在R3 上也进行类似操作。

```
[R2-bgp]peer 10.0.3.3 next-hop-local
```

```
[R2-bgp]peer 10.0.4.4 next-hop-local
```

```
[R3-bgp]peer 10.0.2.2 next-hop-local
```

```
[R3-bgp]peer 10.0.4.4 next-hop-local
```

配置完成后，在R4上查看BGP路由表。

```
[R4]display bgp routing-table
```

BGP Local router ID is 10.0.4.4

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.3.3	0	100	0	100i
* i		10.0.2.2	0	100	0	100i


```
*> 10.0.100.4/32    0.0.0.0    0        0        i
```

可以看到，去往10.0.100.1/32的两条路由信息现在都标记为可用了。

在R1上测试R1的Loopback 1接口与R4的Loopback 1接口之间的连通性。

```
<R1>ping -c 1 -a 10.0.100.1 10.0.100.4
```

```
PING 10.0.100.4: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.100.4: bytes=56 Sequence=1 ttl=254 time=20 ms
```

```
--- 10.0.100.4 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/20/20 ms
```

可以看到，客户网络现在能够与服务器进行通信了。

4. Next Hop 属性对BGP路由协议选路的影响

虽然客户网络与服务器之间能够进行通信了，但实际上还存在一些问题。

在R1上使用tracert命令验证从10.0.100.1/32去往10.0.100.4/32的报文所经过的路径。

```
<R1>tracert -a 10.0.100.1 10.0.100.4
```

```
tracert to 10.0.100.4(10.0.100.4), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.12.2 20 ms 10 ms 20 ms
```

```
2 10.0.24.4 20 ms 20 ms 20 ms
```

可以看到，从R1去往10.0.100.4/32时使用的是经过R2的路径。

在R4上使用tracert命令验证从10.0.100.4/32去往10.0.100.1/32的报文所经过的路径。

```
<R4>tracert -a 10.0.100.4 10.0.100.1
```

```
tracert to 10.0.100.1(10.0.100.1), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.34.3 10 ms 20 ms 10 ms
```

```
2 10.0.13.1 20 ms 40 ms 20 ms
```

可以看到，从R4去往10.0.100.1/32时使用的是经过R3的路径。

通信双方的往返报文选用不同路径的现象称为不对称路由。对于某些特定应用，以及在部署了某些特别的安全设备和安全策略的情况下，不对称路由的存在可能会导致通信中断的现象。

在R4上查看BGP路由表。

```
[R4]display bgp routing-table
```

```
BGP Local router ID is 10.0.4.4
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 3
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.3.3	0	100	0	100i
* i		10.0.2.2	0	100	0	100i
*>	10.0.100.4/32	0.0.0.0	0		0	i

可以看到，去往10.0.100.1/32的两条路由信息均标记为可用，但BGP路由协议最终选择了Next Hop 属性为10.0.3.3 的路由信息。

根据BGP协议路由选择的策略，此时去往10.0.100.1/32的两条路由信息的PrefVal属性、LocPrf属性、路由生成方式、AS_Path属性、Origin属性、MED属性都相同，且两条路由信息都来自IBGP 对等体，所以需要比较两条路由信息中去往Next Hop 地址的IGP开销，并选择开销更小的路由。

Route Flags: R - relay, D - download to fib

Route Flags: R - relay, D - download to fib

Routing Tables: Public

```

Destinations : 16  Routes : 16
Destination/Mask  Proto    Pre   Cost    Flags   NextHop
Interface
  10.0.2.2/32    OSPF     10   48      D      10.0.24.2  Serial1/0/0
  10.0.3.3/32    OSPF          10   100      D        10.0.34.3
GigabitEthernet0/0/0
  10.0.4.4/32   Direct    0    0       D      127.0.0.1  LoopBack0
.....

```

可以看到，去往 10.0.3.3/32 的开销值已经变为 100，而去往 10.0.2.2/32 网络的开销值没有改变。

查看 R4 的 BGP 路由表。

```
[R4]display bgp routing-table
```

```
BGP Local router ID is 10.0.4.4
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 3
```

```

Network      NextHop  MED  LocPrf  PrefVal  Path/Ogn
*>i  10.0.100.1/32    10.0.2.2  0     100     0       100i
* i           10.0.3.3  0     100     0       100i
*>  10.0.100.4/32    0.0.0.0  0           0         i

```

可以看到，在 R4 的 BGP 路由表中，BGP 路由协议选择了 Next Hop 为 10.0.2.2 的路由作为去往 10.0.100.1/32 的最佳路由。

在 R4 上使用 tracert 命令验证从 10.0.100.4/32 去往 10.0.100.1/32 的报文所经过的路径。

```
<R4>tracert -a 10.0.100.4 10.0.100.1
```

```
tracert to 10.0.100.1(10.0.100.1), max hops: 30 ,packet length:
40,press CTRL_C to break
```

```
1 10.0.24.2 20 ms 10 ms 20 ms
```

```
2 10.0.12.1 30 ms 10 ms 30 ms
```

可以看到，从R4去往10.0.100.1/32时使用的是经过R2的路径。

在R1 上测试R1 的Loopback 1 接口与R4 的Loopback 1 接口之间的连通性。

```
<R1>ping -c 1 -a 10.0.100.1 10.0.100.4
```

```
PING 10.0.100.4: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.100.4: bytes=56 Sequence=1 ttl=255 time=20 ms
```

```
--- 10.0.100.4 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/20/20 ms
```

可以看到，客户与服务器之间可以正常通信，并且消除了非对称路由的现象。

为了避免非对称路由的情况，我们选择了修改R4 上的GE 0/0/0 接口的OSPF 协议的开销值。其实，修改R4 上Serial 1/0/0 接口的开销值为1（1为最小开销值）也可以实现同样的目的。总之，只有熟练掌握了BGP协议选择最佳路由的机制，才能在实际场景中灵活地使用BGP路由策略来控制流量的转发路径。

思考

BGP 在优选路径的过程中，会按一定的先后顺序来比较路由的属性。在比较 Next Hop属性之前，需要比较的是哪一种属性？

3.8 BGP 路径选择——AS Path

原理概述

当一台BGP路由器中存在多条去往同一目标网络的BGP路由时，BGP协议会对这些 BGP路由的属性进行比较，以确定去往该目标网络的最优 BGP路由。首先要比较的属性是 Preferred Value，然后是 Local Preference，再次是路由生成方式，如果在比较了这几个属性之后还是无法确定出最优路由，则将进行AS_Path属性的比较。

AS_Path属性顺序地记录了某条BGP路由所经过的AS信息。BGP路由器在向EBGP对等体通告路由时，会在该路由的AS_Path属性的最左端添加本地自治系统的AS编号。BGP在比较了AS_Path属性后，会优选AS_Path长度最短的那条路由。如果AS_Path的长度相等，则 BGP 会对下一个属性 Origin 进行比较。另外，AS_Path 还可以用来防止AS之间的路由环路。当路由器从EBGP邻居收到BGP路由时，如果该路由的AS_Path中包含了自己的AS编号，则该路由将会被直接丢弃。

类似于其他BGP路由属性，AS_Path属性也是可以被手动修改的。

实验目的

- 理解AS_Path 属性的概念
- 理解通过AS_Path 属性进行选路的机制
- 掌握修改AS_Path 属性的方法

实验内容

实验拓扑如图3-8所示，实验编址如表3-8所示。本实验模拟了一个运营商网络场景，所有路由器都运行 BGP 协议，R1 的 Loopback 0 接口用来模拟某一个用户网络10.0.1.1/32，R2 的Loopback 0 接口用来模拟另一个用户网络10.0.2.2/32。两个用户网络需要进行互相通信，但由于 AS 500 转发的流量太多，所以运营商要求 10.0.1.1/32 与10.0.2.2/32

之间的通信只能使用经由 R3、R4 的路径；如果这条路径发生了故障，才能使用经由AS 500 的路径。

实验拓扑

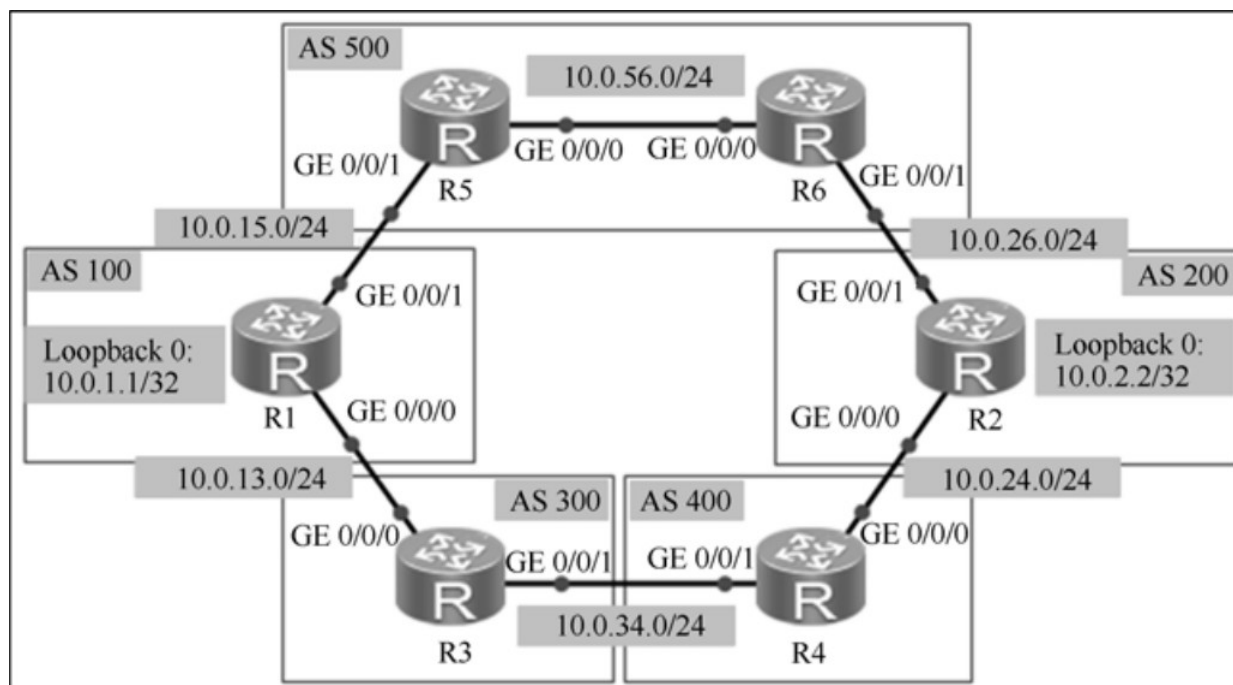


图3-8 BGP路径选择-AS_Path

实验编址表

表3-8 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.13.1	255.255.255.0	N/A
	GE 0/0/1	10.0.15.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.24.2	255.255.255.0	N/A
	GE 0/0/1	10.0.26.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/0	10.0.13.3	255.255.255.0	N/A
	GE 0/0/1	10.0.34.3	255.255.255.0	N/A
R4(AR2220)	GE 0/0/0	10.0.24.4	255.255.255.0	N/A
	GE 0/0/1	10.0.34.4	255.255.255.0	N/A
R5(AR2220)	GE 0/0/0	10.0.56.5	255.255.255.0	N/A
	GE 0/0/1	10.0.15.5	255.255.255.0	N/A
R6(AR2220)	GE 0/0/0	10.0.56.6	255.255.255.0	N/A
	GE 0/0/1	10.0.26.6	255.255.255.0	N/A

实验步骤

1．基本配置

根据图3-8和表3-8进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 10.0.13.3
```

```
PING 10.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=90 ms
```

```
--- 10.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 90/90/90 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置BGP路由协议

基本配置完成后，进行BGP协议的配置。

```
[R1]bgp 100
```

```
[R1-bgp]peer 10.0.13.3 as-number 300
```

```
[R1-bgp]peer 10.0.15.5 as-number 500
```

```
[R1-bgp]network 10.0.1.1 32
```

```
[R2]bgp 200
```

```
[R2-bgp]peer 10.0.24.4 as-number 400
```

```
[R2-bgp]peer 10.0.26.6 as-number 500
```

```
[R2-bgp]network 10.0.2.2 255.255.255.255
```

```
[R3]bgp 300
```

```
[R3-bgp]peer 10.0.13.1 as-number 100
```

```
[R3-bgp]peer 10.0.34.4 as-number 400
```

```
[R4]bgp 400
```



```
[R4-bgp]peer 10.0.24.2 as-number 200
```

```
[R4-bgp]peer 10.0.34.3 as-number 300
```

```
[R5]bgp 500
```

```
[R5-bgp]peer 10.0.15.1 as-number 100
```

```
[R5-bgp]peer 10.0.56.6 as-number 500
```

```
[R5-bgp]peer 10.0.56.6 next-hop-local
```

```
[R6]bgp 500
```

```
[R6-bgp]peer 10.0.26.2 as-number 200
```

```
[R6-bgp]peer 10.0.56.5 as-number 500
```

```
[R6-bgp]peer 10.0.56.5 next-hop-local
```

上述配置完成后，在R1上查看BGP邻居关系。在其余设备上查看BGP邻居关系的过程在此省略。

```
[R1]display bgp peer
```

```
BGP local router ID : 10.0.13.1
```

```
Local AS number : 100
```

```
Total number of peers : 2      Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.13.3	4	300	10	12	0	00:06:23 Established
1						
10.0.15.5	4	500	8	11	0	00:05:26 Established
1						

可以看到，R1已经与R3和R5建立了EBGP邻居关系。

3．观察AS_Path属性对BGP选路的影响

在R1上查看BGP路由表，观察10.0.1.1/32访问10.0.2.2/32的选路情况。

```
[R1]display bgp routing-table
```

BGP Local router ID is 10.0.13.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	0.0.0.0	0		0	i
*>	10.0.2.2/32	10.0.15.5			0	500 200i
*		10.0.13.3		0	300	400 200i

可以看到，R1的BGP路由表中存在两条去往10.0.2.2/32的路由，下一跳分别为R5（10.0.15.5）和R3（10.0.13.3），但是优选的是下一跳为R5的路由。这两条路由的PrefVal值均为0，LocPrf属性均为空，均不是本地生成的路由，但它们的AS_Path属性不同。观察发现，下一跳为R5的路由的AS_Path属性为500 200，所以长度为2，而下一跳为R3的路由的AS_Path属性为300 400 200，所以长度为3，于是，R1最终选择了下一跳为R5的路由，因为它的AS_Path长度较小。

在R1上使用tracert命令验证从10.0.1.1/32去往10.0.2.2/32的报文所经过的路径。

```
[R1]tracert -a 10.0.1.1 10.0.2.2
```

```
tracert to 10.0.2.2(10.0.2.2), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
 1 10.0.15.5 60 ms 1 ms 10 ms  
 2 10.0.56.6 40 ms 10 ms 20 ms  
 3 10.0.26.2 20 ms 20 ms 30 ms
```

可以看到，10.0.1.1/32访问10.0.2.2/32时的确选用了经过R5、R6的路径。

在R2上查看BGP路由表。

```
[R2]display bgp routing-table
BGP Local router ID is 10.0.24.2
Status codes: *- valid, >- best, d - damped,
              h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete
Total Number of Routes: 3
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.26.6			0	500 100i
*		10.0.24.4		0	400 300	100i
*>	10.0.2.2/32	0.0.0.0	0		0	i

可以看到，R2去往10.0.1.1/32的路由也有两条，优选的是下一跳为R6（10.0.26.6）的路由，其原因也是因为这条路由的AS_Path的长度较短。

在R2上使用tracert命令验证从10.0.2.2/32去往10.0.1.1/32的报文所经过的路径。

```
[R2]tracert -a 10.0.2.2 10.0.1.1
tracert to 10.0.1.1(10.0.1.1), max hops: 30 ,packet length: 40,press
CTRL_C to break
```

```
 1 10.0.26.6 30 ms 10 ms 10 ms
 2 10.0.56.5 10 ms 20 ms 10 ms
 3 10.0.15.1 30 ms 20 ms 20 ms
```

可以看到，10.0.2.2/32访问10.0.1.1/32时选用了经过R6、R5的路径。

4．修改AS_Path属性控制BGP选路

现在，假定R5和R6的流量负担太重，希望用户网络10.0.1.1/32与10.0.2.2/32之间的通信优先选用经由R3和R4的路径。

为了实现这个需求，最直接的做法是在R1上拒绝接收来自AS 500的关于10.0.2.2/32的路由信息，以及在R2上拒绝接收来自AS 500的关于10.0.1.1/32的路由信息。但是，如此一来，R1和R2的BGP路由表中将不再有经由AS 500去往对方的路由信息，当经由R3和R4之间的链路发生故障时，两个用户网络的通信就会中断。为此，可以采用修改AS_Path的方法来更好地实现上述需求。

使用 Route-Policy 对 R1 接收的来自 AS 500 的关于 10.0.2.2/32 的路由信息中的AS_Path属性进行修改。

```
[R1]ip ip-prefix as_path permit 10.0.2.2 32
[R1]route-policy as_path permit node 10
[R1-route-policy]if-match ip-prefix as_path
[R1-route-policy]apply as-path 500 500 additive
[R1-route-policy]route-policy as_path permit node 20
[R1-route-policy]bgp 100
[R1-bgp]peer 10.0.15.5 route-policy as_path import
配置完成后，在R1上查看BGP路由表。
```

```
[R1]display bgp routing-table
BGP Local router ID is 10.0.13.1
Status codes: *- valid, >- best, d - damped,
               h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete
```

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	0.0.0.0	0		0	i
*>	10.0.2.2/32	10.0.13.3			0	300 400 200i
*		10.0.15.5		0	500 500 500	200i

可以看到，现在R1优选了下一跳为10.0.13.3，即通过R3的路径，原因是现在经由R5的路由的AS_Path属性变为了500 500 500 200，长度为4。

在R1上使用tracert命令验证从10.0.1.1/32去往10.0.2.2/32的报文所经过的路径。

```
[R1]tracert -a 10.0.1.1 10.0.2.2
tracert to 10.0.2.2(10.0.2.2), max hops: 30 ,packet length: 40,press
CTRL_C to break
  1 10.0.13.3 80 ms 10 ms 10 ms
  2 10.0.34.4 30 ms 10 ms 10 ms
  3 10.0.24.2 40 ms 20 ms 20 ms
```

可以看到，从10.0.1.1/32去往10.0.2.2/32的报文选用了经由R3、R4的路径。

在R2上使用tracert命令验证从10.0.2.2/32去往10.0.1.1/32的报文所经过的路径。

```
[R2]tracert -a 10.0.2.2 10.0.1.1
tracert to 10.0.1.1(10.0.1.1), max hops: 30 ,packet length: 40,press
CTRL_C to break
  1 10.0.26.6 10 ms 20 ms 10 ms
  2 10.0.56.5 20 ms 30 ms 20 ms
  3 10.0.15.1 30 ms 20 ms 20 ms
```

可以看到，从10.0.2.2/32 去往10.0.1.1/32 的报文依旧选用的是经由AS 500 的路径。

为了实现从10.0.2.2/32去往10.0.1.1/32的报文同样选用经由R4、R3的路径，可以在R2 上修改来自AS 500 的关于10.0.1.1/32 的路由信息的AS_Path 属性。

```
[R2]ip ip-prefix as_path permit 10.0.1.1 32
```

```
[R2]route-policy as_path permit node 10
```

```
[R2-route-policy]if-match ip-prefix as_path
```

```
[R2-route-policy]apply as-path 300 500 500 100 overwrite
```

Warning: The AS-Path lists of routes to which this route-policy is applied will be overwritten. Continue? [Y/N]Y

注意，使用关键字 `overwrite`，意味着将用 300 500 500 100 覆盖路由信息原有的AS_Path属性，所以系统弹出了警告。输入Y选择继续。

创建路由策略的后续索引节点允许未被匹配的路由能够正常被接收。

```
[R2]route-policy as_path permit node 20
```

在R2的BGP视图下调用路由策略。

```
[R2-bgp]peer 10.0.26.6 route-policy as_path import
```

配置完成后，在R2上观察BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.24.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.24.4			0	400 300 100i
*		10.0.26.6		0	300 500 500 100i	
*>	10.0.2.2/32	0.0.0.0	0		0	i

可以看到，现在R2去往10.0.1.1/32网络时，选用的是下一跳为10.0.24.4，即经由R4和R3的路径。经由R6的路由的AS_Path属性已被修改为300 500 500 100，长度为4。注意，选用`overwrite`关键字，路由策略会使用配置的AS编号序列替换原有的AS_Path属性。

在R1上使用tracert命令验证从10.0.1.1/32去往10.0.2.2/32的报文所经过的路径。

```
[R1]tracert -a 10.0.1.1 10.0.2.2
tracert to 10.0.2.2(10.0.2.2), max hops: 30 ,packet length: 40,press
CTRL_C to break
```

```
1 10.0.13.3 50 ms 10 ms 30 ms
```

```
2 10.0.34.4 20 ms 20 ms 20 ms
```

```
3 10.0.24.2 20 ms 20 ms 20 ms
```

可以看到，从10.0.1.1/32去往10.0.2.2/32的报文选用了经由R3和R4的路径。

在R2上使用tracert命令验证从10.0.2.2/32去往10.0.1.1/32的报文所经过的路径。

```
[R2]tracert -a 10.0.2.2 10.0.1.1
tracert to 10.0.1.1(10.0.1.1), max hops: 30 ,packet length: 40,press
CTRL_C to break
```

```
1 10.0.24.4 20 ms 10 ms 10 ms
```

```
2 10.0.34.3 20 ms 20 ms 20 ms
```

```
3 10.0.13.1 30 ms 30 ms 30 ms
```

可以看到，从10.0.2.2/32去往10.0.1.1/32的报文也选用的是经由R4和R3的路径。

至此，在10.0.1.1/32与10.0.2.2/32之间进行通信时，往返路径均经过了R3和R4，实现了所需的路径控制，同时也实现了路径的冗余备份。

思考

路由器把路由传递给IBGP邻居时，是否会把自己的AS编号添加到AS_Path属性中？路由器从IBGP邻居那里接收到路由后，如果发现该路由的AS_Path属性中包括了自己的AS编号，那么该路由是否会被丢弃？

3.9 BGP 路径选择——MED

原理概述

当一台BGP路由器中存在多条去往同一目标网络的BGP路由时，BGP协议会对这些BGP路由的属性进行比较，以确定去往该目标网络的最优BGP路由。BGP路由属性的比较顺序为Preferred Value 属性、Local Preference属性、路由生成方式、AS_Path 属性、Origin属性、MED属性、BGP对等体类型等。

MED (MULTI_EXIT_DISC) 也称为多出口鉴别器，它是一个4字节的整数，取值范围为0 ~ 4294967295。缺省情况下，MED 的值为0，但通过命令default med value 可对其进行修改。MED的数值越小，表明相应的路由优先级越高，因此MED也常被称为Cost。MED属性的主要作用是用来控制来自邻居AS的流量从哪个入口进入到本AS中。

缺省情况下，只有去往同一目标网络的多条路由均来自同一个邻居AS时，BGP才会比较这些路由的MED值，但是，配置命令compare-different-as-med后，则会比较来自不同邻居AS的目标网络相同的BGP路由的MED值。注意，MED属性只会影响相邻两个AS，收到MED属性的AS不会把此属性再继续传递给别的AS。

实验目的

- 理解MED 属性对BGP 路径选择的影响
- 掌握修改MED 属性的方法
- 掌握通过修改MED 值实现流量分担的方法

实验内容

实验拓扑如图3-9所示，实验编址如表3-9所示。本实验包含了4个AS，所有的路由器都运行BGP，所有的BGP邻居关系都使用直连物理接口来建立。R1上的Loopback 1、Loopback 2、Loopback 3 接口用来分

别模拟 3 个网络 172.16.1.0/24、172.16.2.0/24、192.168.1.0/24，这3 个网络都被通告进BGP 进程。对于通信的需求是：从AS 200 去往 172.16.1.0/24 的数据流量需经由 R1 的 GE 0/0/2 接口进入 AS 100，从 AS 200 去往172.16.2.0/24 的数据流量需经由 R1 的 GE 0/0/0 接口进入 AS 100，从 AS 400 去往192.168.1.0/24 的数据流量需先通过R4，然后经由R1 的GE 0/0/1 接口进入AS 100，所有需求都应通过修改MED属性值来实现。

实验拓扑

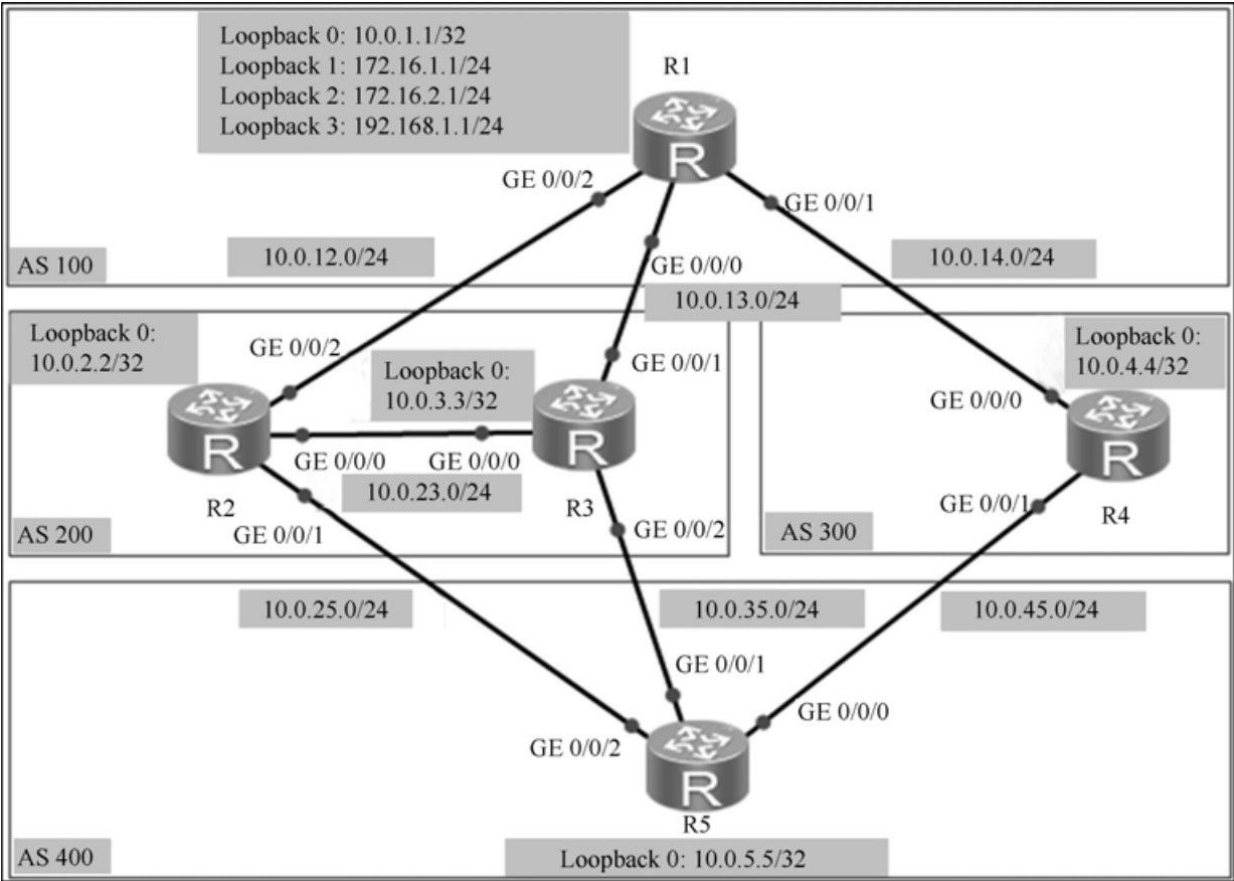


图3-9 BGP路径选择-MED

实验编址表

表3-9 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.13.1	255.255.255.0	N/A
	GE 0/0/1	10.0.14.1	255.255.255.0	N/A
	GE 0/0/2	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	172.16.1.1	255.255.255.0	N/A
	Loopback 2	172.16.2.1	255.255.255.0	N/A
	Loopback 3	192.168.1.1	255.255.255.0	N/A

(续表)

设备	接口	IP 地址	子网掩码	默认网关
R2(AR2220)	GE 0/0/0	10.0.23.2	255.255.255.0	N/A
	GE 0/0/2	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.25.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/0	10.0.23.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.35.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2220)	GE 0/0/0	10.0.14.4	255.255.255.0	N/A
	GE 0/0/1	10.0.45.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
R5(AR2220)	GE 0/0/0	10.0.45.5	255.255.255.0	N/A
	GE 0/0/1	10.0.35.5	255.255.255.0	N/A
	GE 0/0/2	10.0.25.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A

实验步骤

1. 基本配置

根据图3-9和表3-9进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=270 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

0.00% packet loss

round-trip min/avg/max = 270/270/270 ms

其余直连网段的连通性测试过程在此省略。

2 . 配置BGP路由协议

配置BGP路由协议，使用直连物理接口建立BGP邻居关系。

```
[R1]bgp 100
[R1-bgp]router-id 10.0.1.1
[R1-bgp]peer 10.0.12.2 as-number 200
[R1-bgp]peer 10.0.13.3 as-number 200
[R1-bgp]peer 10.0.14.4 as-number 300
[R1-bgp]network 172.16.1.0 24
[R1-bgp]network 172.16.2.0 24
[R1-bgp]network 192.168.1.0 24
[R1-bgp]network 10.0.1.1 32
[R2]bgp 200
[R2-bgp]router-id 10.0.2.2
[R2-bgp]peer 10.0.12.1 as-number 100
[R2-bgp]peer 10.0.23.3 as-number 200
[R2-bgp]peer 10.0.23.3 next-hop-local
[R2-bgp]peer 10.0.25.5 as-number 400
[R2-bgp]network 10.0.2.2 32
[R3]bgp 200
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.13.1 as-number 100
[R3-bgp]peer 10.0.23.2 as-number 200
[R3-bgp]peer 10.0.23.2 next-hop-local
[R3-bgp]peer 10.0.35.5 as-number 400
```

```

[R3-bgp]network 10.0.3.3 32
[R4]bgp 300
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.14.1 as-number 100
[R4-bgp]peer 10.0.45.5 as-number 400
[R4-bgp]network 10.0.4.4 32
[R5]bgp 400
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.25.2 as-number 200
[R5-bgp]peer 10.0.35.3 as-number 200
[R5-bgp]peer 10.0.45.4 as-number 300
[R5-bgp]network 10.0.5.5 32

```

配置完成后，查看每台路由器的BGP邻居关系。

```
[R1]display bgp peer
```

BGP local router ID : 10.0.1.1

Local AS number : 100

Total number of peers : 3 Peers in established state : 3

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.12.2	4	200	8	13	0	00:04:31
10.0.13.3	4	200	10	12	0	00:03:27
10.0.14.4	4	300	9	10	0	00:02:26

```
[R2]display bgp peer
```

BGP local router ID : 10.0.2.2

Local AS number : 200

Total number of peers : 3 Peers in established state : 3

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.12.1	4	100	12	8	0	00:04:50
Established	4					
10.0.23.3	4	200	9	9	0	00:03:39
Established	6					
10.0.25.5	4	400	12	9	0	00:01:42
Established	1					

[R3]display bgp peer

BGP local router ID : 10.0.3.3

Local AS number : 200

Total number of peers : 3 Peers in established state : 3

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.13.1	4	100	12	11	0	00:04:12
Established	4					
10.0.23.2	4	200	9	10	0	00:04:05
Established	5					
10.0.35.5	4	400	10	10	0	00:02:19
Established	2					

[R4]display bgp peer

BGP local router ID : 10.0.4.4

Local AS number : 300

Total number of peers : 2 Peers in established state : 2

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					

```

10.0.14.1    4    100    9          10      0      00:03:25
Established   6
10.0.45.5    4   400    10     11      0      00:02:27  Established   6
[R5]display bgp peer
BGP local router ID : 10.0.5.5
Local AS number : 400
Total number of peers : 3      Peers in established state : 3
Peer    V    AS    MsgRcvd    MsgSent    OutQ    Up/Down
State   PrefRcv
10.0.25.2  4   200    15      20      0      00:07:15  Established
7
10.0.35.3   4   200     9          10      0      00:02:53
Established   6
10.0.45.4   4   300     9          10      0      00:02:46
Established   6

```

可以看到，各路由器之间的BGP邻居关系都已经正常建立。

查看每台路由器上的BGP路由表。

```
[R1]display bgp routing-table
```

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 12

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	0.0.0.0	0		0	i
*>	10.0.2.2/32	10.0.12.2	0		0	200i
*		10.0.13.3		0	200i	

```

*> 10.0.3.3/32    10.0.12.2        0      200i
*           10.0.13.3  0      0      200i
*> 10.0.4.4/32    10.0.14.4  0      0      300i
*> 10.0.5.5/32    10.0.12.2        0      200 400i
*           10.0.13.3        0      200 400i
*           10.0.14.4        0      300 400i
*> 172.16.1.0/24   0.0.0.0    0      0      i
*> 172.16.2.0/24   0.0.0.0    0      0      i
*> 192.168.1.0/24  0.0.0.0    0      0      i

```

[R2]display bgp routing-table

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 15

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.12.1	0		0	100i
* i		10.0.23.3	0	100	0	100i
*>	10.0.2.2/32	0.0.0.0	0		0	i
*>i	10.0.3.3/32	10.0.23.3	0	100	0	i
*>	10.0.4.4/32	10.0.12.1			0	100 300i
*		10.0.25.5		0	400	300i
* i		10.0.23.3	100	0	100	300i
*>	10.0.5.5/32	10.0.25.5	0		0	400i
* i		10.0.23.3	0	100	0	400i
*>	172.16.1.0/24	10.0.12.1	0		0	100i
* i		10.0.23.3	0	100	0	100i

```
*> 172.16.2.0/24    10.0.12.1  0      0      100i
* i      10.0.23.3  0      100    0      100i
*> 192.168.1.0/24  10.0.12.1  0      0      100i
* i      10.0.23.3  0      100    0      100i
```

<R3>display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 15

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.13.1	0		0	100i
* i		10.0.23.2	0	100	0	100i
*>i	10.0.2.2/32	10.0.23.2	0	100	0	i
*>	10.0.3.3/32	0.0.0.0	0		0	i
*>	10.0.4.4/32	10.0.13.1			0	100 300i
*		10.0.35.5		0	400	300i
* i		10.0.23.2	100	0	100	300i
*>	10.0.5.5/32	10.0.35.5	0		0	400i
* i		10.0.23.2	0	100	0	400i
*>	172.16.1.0/24	10.0.13.1	0		0	100i
* i		10.0.23.2	0	100	0	100i
*>	172.16.2.0/24	10.0.13.1	0		0	100i
* i		10.0.23.2	0	100	0	100i
*>	192.168.1.0/24	10.0.13.1	0		0	100i
* i		10.0.23.2	0	100	0	100i

[R4]display bgp routing-table

BGP Local router ID is 10.0.4.4

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 15

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.14.1	0		0	100i
*		10.0.45.5		0	400 200	100i
*>	10.0.2.2/32	10.0.14.1			0	100 200i
*		10.0.45.5		0	400 200	i
*>	10.0.3.3/32	10.0.14.1			0	100 200i
*		10.0.45.5		0	400 200	i
*>	10.0.4.4/32	0.0.0.0	0		0	i
*>	10.0.5.5/32	10.0.45.5	0		0	400i
*		10.0.14.1		0	100 200	400i
*>	172.16.1.0/24	10.0.14.1	0		0	100i
*		10.0.45.5		0	400 200	100i
*>	172.16.2.0/24	10.0.14.1	0		0	100i
*		10.0.45.5		0	400 200	100i
*>	192.168.1.0/24	10.0.14.1	0		0	100i
*		10.0.45.5		0	400 200	100i

[R5]display bgp routing-table

BGP Local router ID is 10.0.5.5

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 22

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.25.2		0	200	100i
*		10.0.35.3		0	200	100i
*		10.0.45.4		0	300	100i
*>	10.0.2.2/32	10.0.25.2	0		0	200i
*		10.0.35.3		0		200i
*		10.0.45.4		0	300	100 200i
*>	10.0.3.3/32	10.0.25.2		0		200i
*		10.0.35.3	0	0		200i
*		10.0.45.4		0	300	100 200i
*>	10.0.4.4/32	10.0.45.4	0		0	300i
*		10.0.25.2		0	200	100 300i
*		10.0.35.3		0	200	100 300i
*>	10.0.5.5/32	0.0.0.0	0		0	i
*>	172.16.1.0/24	10.0.25.2			0	200 100i
*		10.0.35.3		0	200	100i
*		10.0.45.4		0	300	100i
*>	172.16.2.0/24	10.0.25.2			0	200 100i
*		10.0.35.3		0	200	100i
*		10.0.45.4		0	300	100i
*>	192.168.1.0/24	10.0.25.2			0	200 100i
*		10.0.35.3		0	200	100i
*		10.0.45.4		0	300	100i

可以看到，R2、R3、R4、R5上都接收到了R1的4个网段的路由信息。仔细观察发现，无论是通过EBGP邻居还是IBGP邻居接收到的路由条目，以及路由器自身产生的BGP路由条目，其MED字段的值均为0。如果接收到的路由条目经过了一个AS进行中转，那么MED值将会丢

失，设置为空。在BGP选择最佳路径时，MED值为空实际上等同于值为0。

3．控制来自同一AS的数据流量的最佳路径选择

目前，根据BGP选路机制中EBGP路由优于IBGP路由的原则，R2在去往R1的各Loopback 接口所表示的各个网络时，选择了R1的GE 0/0/2接口作为进入AS 100 的入口；R3 在去往R1 的各Loopback 接口所表示的各个网络时，选择了R1的GE 0/0/0 接口作为进入AS 100 的入口。这一结论可以通过使用tracert命令得到验证，如下。

```
<R2>tracert -a 10.0.2.2 172.16.1.1
```

```
tracert to 172.16.1.1(172.16.1.1), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.12.1 10 ms 10 ms 10 ms
```

```
<R2>tracert -a 10.0.2.2 172.16.2.1
```

```
tracert to 172.16.2.1(172.16.2.1), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.12.1 10 ms 10 ms 10 ms
```

```
<R3>tracert -a 10.0.3.3 172.16.1.1
```

```
tracert to 172.16.1.1(172.16.1.1), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.13.1 30 ms 1 ms 10 ms
```

```
<R3>tracert -a 10.0.3.3 172.16.2.1
```

```
tracert to 172.16.2.1(172.16.2.1), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.13.1 10 ms 10 ms 10 ms
```

可以看到，R2 去往 172.16.1.1 和 172.16.2.1 的下一跳为 10.0.12.1（R1 的 GE 0/0/2接口），R3 去往172.16.1.1和172.16.2.1的下一跳为10.0.13.1（R1的GE 0/0/0接口）。现在，AS 100的管理人员要求AS

200 访问172.16.1.0/24 网络的流量从R1的GE 0/0/2 接口进入AS 100，访问172.16.2.0/24 网络的流量从R1 的GE 0/0/0 接口进入AS 100。

在R1上使用前缀列表匹配要修改MED值的路由。

```
[R1]ip ip-prefix 1 permit 172.16.1.0 24
```

```
[R1]ip ip-prefix 2 permit 172.16.2.0 24
```

在R1 上创建Route-Policy 1，将172.16.1.0/24 的MED 配置为100，将172.16.2.0/24的MED配置为200。

```
[R1]route-policy 1 permit node 10
```

```
[R1-route-policy]if-match ip-prefix 1
```

```
[R1-route-policy]apply cost 100
```

```
[R1-route-policy]route-policy 1 permit node 20
```

```
[R1-route-policy]if-match ip-prefix 2
```

```
[R1-route-policy]apply cost 200
```

```
[R1-route-policy]route-policy 1 permit node 30
```

然后，在 R1 上创建 Route-Policy 2，将 172.16.2.0/24 的 MED 配置为 100，将172.16.1.0/24的MED配置为200。

```
[R1]route-policy 2 permit node 10
```

```
[R1-route-policy]if-match ip-prefix 2
```

```
[R1-route-policy]apply cost 100
```

```
[R1-route-policy]route-policy 2 permit node 20
```

```
[R1-route-policy]if-match ip-prefix 1
```

```
[R1-route-policy]apply cost 200
```

```
[R1-route-policy]route-policy 2 permit node 30
```

在R1 上配置peer 10.0.12.2 route-policy 1 export命令，使得R1 在传递路由给R2时调用Route-Policy 1，再配置peer 10.0.13.3 route-policy 2 export命令，使得R1 在传递路由给R3 时调用Route-Policy 2。

```
[R1-bgp]peer 10.0.12.2 route-policy 1 export
```

```
[R1-bgp]peer 10.0.13.3 route-policy 2 export
```

配置完成后，分别在R2和R3上查看BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 14

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.12.1	0		0	100i
* i		10.0.23.3	0	100	0	100i
*>	10.0.2.2/32	0.0.0.0	0		0	i
*>i	10.0.3.3/32	10.0.23.3	0	100	0	i
*>	10.0.4.4/32	10.0.12.1			0	100 300i
*		10.0.25.5		0	400	300i
* i		10.0.23.3	100	0	400	300i
*>	10.0.5.5/32	10.0.25.5	0		0	400i
* i		10.0.23.3	0	100	0	400i
*>	172.16.1.0/24	10.0.12.1	100		0	100i
*>i	172.16.2.0/24	10.0.23.3	100	100	0	100i
*		10.0.12.1	200		0	100i
*>	192.168.1.0/24	10.0.12.1	0		0	100i
* i		10.0.23.3	0	100	0	100i

```
[R3]display bgp routing-table
```

BGP Local router ID is 10.0.3.3

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 14

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.13.1	0		0	100i
* i		10.0.23.2	0	100	0	100i
*>i	10.0.2.2/32	10.0.23.2	0	100	0	i
*>	10.0.3.3/32	0.0.0.0	0		0	i
*>	10.0.4.4/32	10.0.13.1			0	100 300i
*		10.0.35.5		0	400	300i
* i		10.0.23.2	100	0	100	300i
*>	10.0.5.5/32	10.0.35.5	0		0	400i
* i		10.0.23.2	0	100	0	400i
*>i	172.16.1.0/24	10.0.23.2	100	100	0	100i
*		10.0.13.1	200		0	100i
*>	172.16.2.0/24	10.0.13.1	100		0	100i
*>	192.168.1.0/24	10.0.13.1	0		0	100i
* i		10.0.23.2	0	100	0	100i

在PrefVal属性、LocPrf属性、路由生成方式、AS_Path属性、Origin属性都相同的情况下，BGP会选择最小MED值的路由作为最优路由。可以看到，在R2的BGP路由表中，去往172.16.1.0/24的下一跳为10.0.12.1，MED值为100，也就是选择了R1的GE 0/0/2 接口作为进入AS 100 的入口；去往172.16.2.0/24 的下一跳为10.0.23.3，MED 值为100，也就是选择了R1 的GE 0/0/0 接口作为进入AS 100 的入口。

根据相同的原理，R3选择了去往目标网络172.16.1.0/24的下一跳为10.0.23.2，也就是选择了R1的GE 0/0/2接口作为进入AS 100 的入口；去往目标网络172.16.2.0/24 的下一跳为10.0.13.1，也就是选择了R1 的GE 0/0/0 接口作为进入AS 100 的入口。

在R2和R3上，使用tracert命令验证数据经过的路径情况。

```
<R2>tracert -a 10.0.2.2 172.16.1.1
```

```
tracert to 172.16.1.1(172.16.1.1), max hops: 30 ,packet length:  
40,press CTRL_C to break
```

```
1 10.0.12.1 10 ms 10 ms 30 ms
```

```
<R2>tracert -a 10.0.2.2 172.16.2.1
```

```
tracert to 172.16.2.1(172.16.2.1), max hops: 30 ,packet length:  
40,press CTRL_C to break
```

```
1 10.0.23.3 20 ms 10 ms 20 ms
```

```
2 10.0.13.1 20 ms 40 ms 10 ms
```

```
<R3>tracert -a 10.0.3.3 172.16.1.1
```

```
tracert to 172.16.1.1(172.16.1.1), max hops: 30 ,packet length:  
40,press CTRL_C to break
```

```
1 10.0.23.2 20 ms 1 ms 20 ms
```

```
2 10.0.12.1 20 ms 20 ms 30 ms
```

```
<R3>tracert -a 10.0.3.3 172.16.2.1
```

```
tracert to 172.16.2.1(172.16.2.1), max hops: 30 ,packet length:  
40,press CTRL_C to break
```

```
1 10.0.13.1 30 ms 20 ms 20 ms
```

可以看到，AS 200 访问 172.16.1.0/24 网络的流量和访问 172.16.2.0/24 网络的流量分别是从小R1的GE 0/0/2接口和GE 0/0/0 接口进入AS 100 的。至此，AS 100 的管理员的要求得到了满足。

查看R2、R3、R5上的关于172.16.1.0的BGP路由的详细信息。

```
[R2]display bgp routing-table 172.16.1.0
```

```
BGP local router ID : 10.0.2.2
```

```
Local AS number : 200
```

```
Paths: 1 available, 1 best, 1 select
```

BGP routing table entry information of 172.16.1.0/24:

From: 10.0.12.1 (10.0.1.1)

Route Duration: 00h22m48s

Direct Out-interface: GigabitEthernet0/0/2

Original nexthop: 10.0.12.1

Qos information : 0x0

AS-path 100, origin igp, MED 100, pref-val 0, valid, external, best, select, active, pre 255

Advertised to such 3 peers:

10.0.12.1

10.0.25.5

10.0.23.3

[R3]display bgp routing-table 172.16.1.0

BGP local router ID : 10.0.3.3

Local AS number : 200

Paths: 2 available, 1 best, 1 select

BGP routing table entry information of 172.16.1.0/24:

From: 10.0.23.2 (10.0.2.2)

Route Duration: 00h23m32s

Relay IP Nexthop: 0.0.0.0

Relay IP Out-Interface: GigabitEthernet0/0/0

Original nexthop: 10.0.23.2

Qos information : 0x0

AS-path 100, origin igp, MED 100, localpref 100, pref-val 0, valid, internal, best, select, active, pre 255

Advertised to such 2 peers:

10.0.35.5

10.0.13.1

BGP routing table entry information of 172.16.1.0/24:

From: 10.0.13.1 (10.0.1.1)

Route Duration: 00h23m31s

Direct Out-interface: GigabitEthernet0/0/1

Original nexthop: 10.0.13.1

Qos information : 0x0

AS-path 100, origin igp, MED 200, pref-val 0, valid, external, pre 255,
not preferred for MED

Not advertised to any peer yet

[R5]display bgp routing-table 172.16.1.0

BGP local router ID : 10.0.5.5

Local AS number : 400

Paths: 3 available, 1 best, 1 select

BGP routing table entry information of 172.16.1.0/24:

From: 10.0.25.2 (10.0.2.2)

Route Duration: 00h26m06s

Direct Out-interface: GigabitEthernet0/0/2

Original nexthop: 10.0.25.2

Qos information : 0x0

AS-path 200 100, origin igp, pref-val 0, valid, external, best, select,
active, pre 255

Advertised to such 3 peers:

10.0.25.2

10.0.45.4

10.0.35.3

BGP routing table entry information of 172.16.1.0/24:

From: 10.0.35.3 (10.0.3.3)

Route Duration: 00h01m24s

Direct Out-interface: GigabitEthernet0/0/1

Original nexthop: 10.0.35.3

Qos information : 0x0

AS-path 200 100, origin igp, pref-val 0, valid, external, pre 255, not preferred for router ID

Not advertised to any peer yet

BGP routing table entry information of 172.16.1.0/24:

From: 10.0.45.4 (10.0.4.4)

Route Duration: 00h25m26s

Direct Out-interface: GigabitEthernet0/0/0

Original nexthop: 10.0.45.4

Qos information : 0x0

AS-path 300 100, origin igp, pref-val 0, valid, external, pre 255, not preferred for router ID

Not advertised to any peer yet

可以观察到，R2和R3去往172.16.1.0/24 网络的路由都携带了MED属性，而R5去往172.16.1.0/24网络的路由，没有MED值，这说明BGP路由的MED属性只传递给邻居AS，邻居AS不会将收到的MED属性再传递给其他AS。

4．控制来自不同AS且去往同一目标网络的数据流量的最佳路径选择

查看R5的BGP路由表。

[R5]display bgp routing-table

BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 20

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.25.2			0	200 100i
.....						
*		10.0.45.4		0	300	100i
*>	192.168.1.0/24	10.0.25.2			0	200 100i
*		10.0.35.3		0	200	100i
*		10.0.45.4		0	300	100i

可以看到，在R5的BGP路由表中有多条去往192.168.1.0/24网络的路由。我们知道，通过PrefVal属性、LocPrf属性、路由生成方式、AS_Path属性、Origin属性的比较都无法选出最优路由时，BGP将会比较MED属性。但是，在默认情况下，BGP不会比较来自不同AS的路由的MED属性，所以R5无法通过比较MED属性选择出去往192.168.1.0/24网络的最优路由。为此，BGP会继续依次比较邻居类型、到达下一跳的IGP开销值等，最后的结果是，R5选择了Router-ID最小的路由器R2发布的路由作为最优路由。

现在，网络管理员希望AS 400去往192.168.1.0/24网络的流量经由R4，然后通过R1的GE 0/0/1接口进入AS 100，采用的方法是修改R2、R3和R4在传递关于192.168.1.0/24的路由信息给R5时的MED属性值。

在R2上使用前缀列表匹配路由192.168.1.0/24。

```
[R2]ip ip-prefix 1 permit 192.168.1.0 24
```

在R2上使用Route-Policy将192.168.1.0/24的MED值配置为200。

```
[R2]route-policy 1 permit node 10
```

```
[R2-route-policy]if-match ip-prefix 1
```

```
[R2-route-policy]apply cost 200
```

```
[R2-route-policy]route-policy 1 permit node 20
```

在R2 上配置peer 10.0.25.5 route-policy 1 export 命令，在传递路由给R5 时调用Route-Policy。

```
[R2]bgp 200
```

```
[R2-bgp]peer 10.0.25.5 route-policy 1 export
```

在R3上完成类似的配置。

```
[R3]ip ip-prefix 1 permit 192.168.1.0 24
```

```
[R3]route-policy 1 permit node 10
```

```
[R3-route-policy]if-match ip-prefix 1
```

```
[R3-route-policy]apply cost 200
```

```
[R3-route-policy]route-policy 1 permit node 20
```

在R3 上配置peer 10.0.35.5 route-policy 1 export 命令，在传递路由给R5 时调用Route-Policy。

```
[R3]bgp 200
```

```
[R3-bgp]peer 10.0.35.5 route-policy 1 export
```

在R4上完成类似的配置。

```
[R4]ip ip-prefix 1 permit 192.168.1.0 24
```

```
[R4]route-policy 1 permit node 10
```

```
[R4-route-policy]if-match ip-prefix 1
```

```
[R4-route-policy]apply cost 100
```

```
[R4-route-policy]route-policy 1 permit node 20
```

在R4 上配置peer 10.0.45.5 route-policy 1 export 命令，在传递路由给R5 时调用Route-Policy。

```
[R4]bgp 300
```

```
[R4-bgp]peer 10.0.45.5 route-policy 1 export
```

在R5上，配置compare-different-as-med命令，让R5强制比较来自不同AS且去往同一目标网络的路由的MED属性值。

```
[R5]bgp 400
```

```
[R5-bgp]compare-different-as-med
```

上述配置完成后，再次查看R5的BGP路由表。

```
[R5-bgp]display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 15

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.25.2			0	200 100i
						
*		10.0.35.3		0		200 100i
*>	192.168.1.0	10.0.45.4	100		0	300 100i
*		10.0.25.2	200	0		200 100i
*		10.0.35.3	200	0		200 100i

可以看到，R5去往192.168.1.0/24网络的路由的MED值均已修改，且选择了MED值最小的路由作为最优路由。使用tracert命令验证路径情况。

```
[R5-bgp]tracert -a 10.0.5.5 192.168.1.1
```

```
traceroute to 192.168.1.1(192.168.1.1), max hops: 30 ,packet length: 40,press CTRL_C to break
```

```
1 10.0.45.4 60 ms 10 ms 10 ms
```

```
2 10.0.14.1 40 ms 20 ms 10 ms
```

可以看到，从AS 400到192.168.1.0/24的流量现在经过了R4，并且经由R1的GE 0/0/1接口进入AS 100。

本实验表明，BGP路由中的MED属性可以用来控制流量经过哪个入口进入一个AS，并且MED只会传递给邻居AS，邻居AS不会再将接收到的MED传递给其他AS。缺省情况下，对于来自不同AS的前缀相同的路由，BGP不会比较它们的MED属性。

思考

聚合后的BGP路由还会携带MED属性吗？

3.10 BGP 路径选择——Community

原理概述

BGP路由的团体属性Community的主要作用是简化路由策略的实现过程。例如，可以将拥有相同团体属性的若干路由视为属于同一个团体，当需要对该团体中所有路由的某个特定属性进行修改时，就没必要逐一对每条路由单独进行修改，而是可以通过匹配相应的团体属性来自动实现所有路由的特定属性的修改。

团体属性是BGP路由的一种可选属性，路由器在向BGP对等体传递路由时，如果希望所传递的路由携带团体属性，则需要额外的配置。一条BGP路由可以拥有多个团体属性，团体属性的值规定为4个字节，通常用AA:NN的格式表示，其中前2个字节AA为AS号，后2个字节NN为团体编号。另外，团体属性的值也可表示为一个十六进制数或十进制数，范围是 0x00000000 ~ 0xFFFFFFFF 或 0 ~ 4294967295，其中，0（0x00000000）~ 65535（0x0000FFFF）和 4294901760（0xFFFF0000）~ 4294967295（0xFFFFFFFF）为预留给

RFC1997中定义了几个特殊的团体，也被称为Well-known团体，它们是Internet、No-Export、No-Advertise、No-Export-Subconfed。路由器接收到属于这些团体的路由时，将会直接执行相应的动作。

Internet团体属性没有一个特定的值，所有路由都默认为属于该团体，路由器可以向任何BGP对等体发布所收到的属于Internet团体的路由。

No-Export 团体属性的值为 4294967041 (0xFFFFFFFF01)。路由器接收到一条携带No-Export 团体属性的路由后，不会将它发布给 EBGP对等体，但可以发布给联盟 (Confederation) EBGP对等体。

No-Advertise 团体属性的值为 4294967042 (0xFFFFFFFF02)。路由器接收到一条携带No-Advertise团体属性的路由后，不会将它发布给任何BGP对等体。

No-Export-Subconfed 团体属性的值为 4294967043 (0xFFFFFFFF03)。路由器接收到一条携带 No-Export-Subconfed团体属性的路由后，不会将它发布给EBGP对等体，也不会将它发布给联盟EBGP对等体。

实验目的

- 理解团体属性的概念与作用
- 熟悉运用团体属性来控制路由传递的方法
- 理解No-Export、No-Advertise、No-Export-Subconfed 属性的区别

实验内容

实验拓扑如图3-10 所示，实验编址如表3-10 所示。本实验网络中，R1 属于AS 100，R2、R3和R4 属于AS 编号为200 的一个联盟，R5 属于AS 300。在联盟AS 200 中，R2和R4 属于成员AS 2001，R3 属于成员AS 2002，整个联盟内使用OSPF 作为IGP。全网路由器都使用直连接口建立BGP 邻居关系，R1上的Loopback 1 ~ Loopback 5 接口用来模拟AS 100需要传递的路由信息。网络管理员需要利用BGP 团体属性来实

现下面的需求：10.0.100.2/32 这条路由信息只能够被AS 200的路由器接收到，不能够被AS 300中的路由器接收到；10.0.100.3/32 这条路由信息只能够被成员AS 2001 的路由器接收到，不能被成员AS 2002 以及AS 300 的路由器接收到；10.0.100.4/32 这条路由信息只能够被R2 接收到，不能被其他路由器接收到；10.0.100.5/32 这条路由信息只能够被R2 和 R3接收到，不能被其他路由器接收到。

实验拓扑

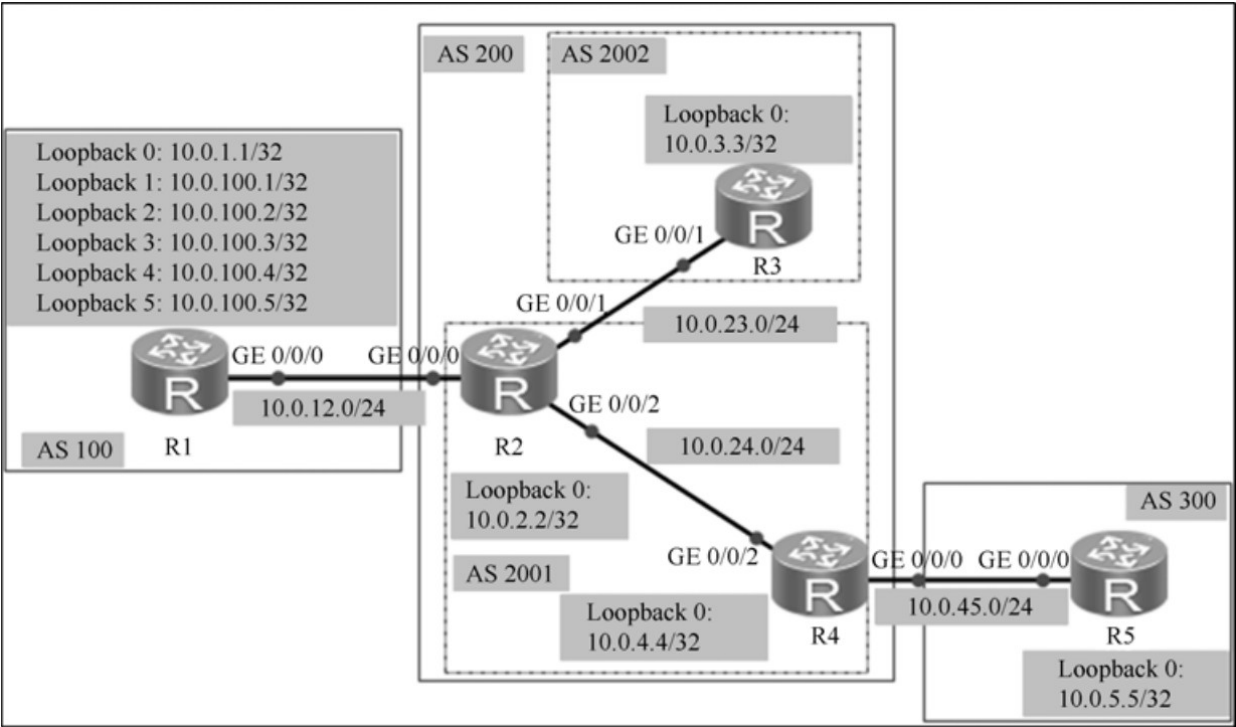


图3-10 BGP路径选择-Community

实验编址表

表3-10 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	10.0.100.1	255.255.255.255	N/A
	Loopback 2	10.0.100.2	255.255.255.255	N/A
	Loopback 3	10.0.100.3	255.255.255.255	N/A
	Loopback 4	10.0.100.4	255.255.255.255	N/A
	Loopback 5	10.0.100.5	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.23.2	255.255.255.0	N/A
	GE 0/0/2	10.0.24.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/1	10.0.23.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2220)	GE 0/0/0	10.0.45.4	255.255.255.0	N/A
	GE 0/0/2	10.0.24.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
R5(AR2220)	GE 0/0/0	10.0.45.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-10和表3-10进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=180 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 180/180/180 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置BGP路由协议

先在AS 200 内进行OSPF 协议的配置。

```
[R2]ospf 1 router-id 10.0.2.2
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
[R3]ospf 1 router-id 10.0.3.3
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R4]ospf 1 router-id 10.0.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
```

配置完成后，在R2上查看OSPF邻居信息。

```
[R2]display ospf peer
```

OSPF Process 1 with Router ID 10.0.2.2

Neighbors

Area 0.0.0.0 interface 10.0.23.2(GigabitEthernet0/0/1)'s neighbors

Router ID: 10.0.3.3 Address: 10.0.23.3

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.23.3 BDR: 10.0.23.2 MTU: 0

Dead timer due in 35 sec

Retrans timer interval: 5

Neighbor is up for 00:02:45

Authentication Sequence: [0]

Neighbors

Area 0.0.0.0 interface 10.0.24.2(GigabitEthernet0/0/2)'s neighbors

Router ID: 10.0.4.4 Address: 10.0.24.4
State: Full Mode:Nbr is Master Priority: 1
DR: 10.0.24.4 BDR: 10.0.24.2 MTU: 0
Dead timer due in 35 sec
Retrans timer interval: 5
Neighbor is up for 00:02:19
Authentication Sequence: [0]

可以看到，R2与R3和R4的OSPF邻居状态均为Full，表明邻居关系已成功建立。

接下来在每一台路由器上配置BGP 协议，其中R1 属于AS 100，R2和R4属于联盟AS 200 的成员AS 2001，R3 属于联盟AS 200 的成员AS 2002，R5 属于AS 300。

```
[R1]bgp 100
[R1-bgp]router-id 10.0.1.1
[R1-bgp]peer 10.0.12.2 as-number 200
[R1-bgp]network 10.0.100.1 32
[R1-bgp]network 10.0.100.2 32
[R1-bgp]network 10.0.100.3 32
[R1-bgp]network 10.0.100.4 32
[R1-bgp]network 10.0.100.5 32
[R2]bgp 2001
[R2-bgp]router-id 10.0.2.2
[R2-bgp]confederation id 200
[R2-bgp]confederation peer-as 2002
[R2-bgp]peer 10.0.23.3 as-number 2002
[R2-bgp]peer 10.0.23.3 next-hop-local
[R2-bgp]peer 10.0.24.4 as-number 2001
```

```

[R2-bgp]peer 10.0.24.4 next-hop-local
[R2-bgp]peer 10.0.12.1 as-number 100
[R3]bgp 2002
[R3-bgp]router-id 10.0.3.3
[R3-bgp]confederation id 200
[R3-bgp]confederation peer-as 2001
[R3-bgp]peer 10.0.23.2 as-number 2001
[R4]bgp 2001
[R4-bgp]router-id 10.0.4.4
[R4-bgp]confederation id 200
[R4-bgp]peer 10.0.24.2 as-number 2001
[R4-bgp]peer 10.0.24.2 next-hop-local
[R4-bgp]peer 10.0.45.5 as-number 300
[R5]bgp 300
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.45.4 as-number 200

```

配置完成后，在R2上查看BGP邻居信息。

```
[R2]display bgp peer
```

BGP local router ID : 10.0.2.2

Local AS number : 2001

Total number of peers : 3 Peers in established state : 3

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.12.1	4	100	6	5	0	00:03:27
Established	5					
10.0.23.3	4	2002	6	9	0	00:04:41
Established	0					

```
10.0.24.4    4    2001        6            8            0        00:04:01
Established    0
```

可以看到，R2与R1、R3、R4的BGP邻居状态均为Established。读者可自行在其他路由器上查看BGP邻居信息。

在R5上查看BGP路由表。

```
[R5]display bgp routing-table
```

```
BGP Local router ID is 10.0.5.5
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 5
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.45.4			0	200 100i
*>	10.0.100.2/32	10.0.45.4			0	200 100i
*>	10.0.100.3/32	10.0.45.4			0	200 100i
*>	10.0.100.4/32	10.0.45.4			0	200 100i
*>	10.0.100.5/32	10.0.45.4			0	200 100i

可以看到，R5已经接收到R1通告的所有路由信息。

3. 使用No_Export团体属性控制路由信息传递

接下来的需求是：10.0.100.2/32 这条路由信息只能够被联盟AS 200中的路由器接收到，而不能够被AS 300中的路由器接收到。为此，可以利用团体属性No_Export 来满足这个需求，因为路由器接收到一条携带 No-Export 团体属性的路由后，不会将它发布给EBGP对等体，但可以发布给联盟EBGP对等体。

在R1上使用前缀列表方法来匹配路由10.0.100.2/32。

```
[R1]ip ip-prefix 1 permit 10.0.100.2 32
```

在R1上创建Route-Policy。

```
[R1]route-policy 1 permit node 10
[R1-route-policy]if-match ip-prefix 1
[R1-route-policy]apply community no-export
[R1-route-policy]route-policy 1 permit node 20
```

在R1上调用Route-Policy。

```
[R1]bgp 100
[R1-bgp]peer 10.0.12.2 route-policy 1 export
```

在R5上查看BGP路由表。

```
[R5]display bgp routing-table
BGP Local router ID is 10.0.5.5
Status codes: *- valid, >- best, d - damped,
               h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete
```

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.45.4			0	200 100i
*>	10.0.100.2/32	10.0.45.4			0	200 100i
*>	10.0.100.3/32	10.0.45.4			0	200 100i
*>	10.0.100.4/32	10.0.45.4			0	200 100i
*>	10.0.100.5/32	10.0.45.4			0	200 100i

可以发现，R5仍然收到了10.0.100.2/32这条路由信息，说明某个地方存在问题。

在R2上查看携带BGP团体属性的路由信息。

```
[R2]display bgp routing-table community
```

Total Number of Routes: 0

可以看到，R2上没有任何携带BGP团体属性的路由信息。原来，在缺省情况下，路由器向BGP对等体传递路由信息时不会携带团体属

性。

在R1的BGP视图下增加如下的配置，使得R1在向R2传递BGP路由信息时携带团体属性。

```
[R1-bgp]peer 10.0.12.2 advertise-community
```

在 R2、R4 上增加同样的配置，使得 R2 在向 R3 和 R4 传递 BGP 路由信息时携带团体属性，并且R4在向R5传递BGP路由信息时也携带团体属性。

```
[R2-bgp]peer 10.0.23.3 advertise-community
```

```
[R2-bgp]peer 10.0.24.4 advertise-community
```

```
[R4-bgp]peer 10.0.45.5 advertise-community
```

配置完成后，在R5上查看BGP路由表。

```
[R5]display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.45.4			0	200 100i
*>	10.0.100.3/32	10.0.45.4			0	200 100i
*>	10.0.100.4/32	10.0.45.4			0	200 100i
*>	10.0.100.5/32	10.0.45.4			0	200 100i

可以看到，现在R5上已经没有了10.0.100.2/32这条路由信息。

在R2、R3和R4上查看携带团体属性的BGP路由表。

```
[R2]display bgp routing-table community
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Community
*> 10.0.100.2/32	10.0.12.1	0		0	no-export

[R3]display bgp routing-table community

BGP Local router ID is 10.0.3.3

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Community
*>i 10.0.100.2/32	10.0.23.2	0	100	0	no-export

[R4]display bgp routing-table community

BGP Local router ID is 10.0.4.4

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Community
*>i 10.0.100.2/32	10.0.24.2	0	100	0	no-export

可以看到，R2、R3和R4上10.0.100.2/32这条路由都携带了No-Export团体属性。

上面的实验表明，R5没有接收到10.0.100.2/32这条路由，但R2、R3和R4都接收到了这条路由，这是因为BGP路由带有No-Export团体属性时，不会传递给EBGP邻居，但在联盟内部的EBGP邻居之间是可以传递的。

4 . 使用No_Export_Subconfed团体属性控制路由信息传递

接下来的需求是：10.0.100.3/32 这条路由信息只能够被联盟AS 200的成员AS 2001中的路由器收到，而不能够被成员AS 2002 以及AS 300中的路由器收到。为此，可以利用团体属性 No_Export_Subconfed 来满足这个需求，因为路由器接收到一条携带No-Export-Subconfed团体属性的路由后，不会将它发布给EBGP对等体，也不会将它发布给联盟EBGP对等体。

在R1上使用前缀列表方法来匹配路由10.0.100.3/32。

```
[R1]ip ip-prefix 2 permit 10.0.100.3 32
```

在R1上创建新的路由策略的Node。

```
[R1]route-policy 1 permit node 11
```

```
[R1-route-policy]if-match ip-prefix 2
```

```
[R1-route-policy]apply community no-export-subconfed
```

配置完成后，在R2、 R3、 R4、 R5上查看BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.12.1	0		0	100i
*>	10.0.100.2/32	10.0.12.1	0		0	100i
*>	10.0.100.3/32	10.0.12.1	0		0	100i
*>	10.0.100.4/32	10.0.12.1	0		0	100i
*>	10.0.100.5/32	10.0.12.1	0		0	100i

可以看到，R2接收到了10.0.100.3/32这条路由信息。

[R3]display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.23.2	0	100	0	(2001)
100i						
*>i	10.0.100.2/32	10.0.23.2	0	100	0	(2001)
100i						
*>i	10.0.100.4/32	10.0.23.2	0	100	0	(2001)
100i						
*>i	10.0.100.5/32	10.0.23.2	0	100	0	(2001)
100i						

可以看到 , R3未接收到10.0.100.3/32这条路由信息。

[R4]display bgp routing-table

BGP Local router ID is 10.0.4.4

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.2/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.3/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.4/32	10.0.24.2	0	100	0	100i

```
*>i 10.0.100.5/32 10.0.24.2 0 100 0 100i
```

可以看到，R4接收到了10.0.100.3/32这条路由信息。

```
[R5]display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.45.4			0	200 100i
*>	10.0.100.4/32	10.0.45.4			0	200 100i
*>	10.0.100.5/32	10.0.45.4			0	200 100i

可以看到，R5未收到10.0.100.3/32这条路由信息。

在R2和R4上查看携带团体属性的BGP路由表。

```
[R2]display bgp routing-table community
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Community
*>	10.0.100.2/32	10.0.12.1	0		0	no-export
*>	10.0.100.3/32	10.0.12.1	0		0	no-export-

subconfed

可以看到，R2上10.0.100.3/32这条路由携带了No-Export-Subconfed团体属性。

```
[R4]display bgp routing-table community
```

BGP Local router ID is 10.0.4.4

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Community
*>i	10.0.100.2/32	10.0.24.2	0	100	0	no-export
*>i	10.0.100.3/32	10.0.24.2	0	100	0	no-export-

subconfed

可以看到，R4上10.0.100.3/32这条路由也携带了No-Export-Subconfed团体属性。

上面的实验表明，带有No-Export-Subconfed团体属性的BGP路由可以在联盟中的成员AS内部传递，但不会在成员AS之间传递，也不会EBGP邻居之间传递。

5. 使用No_Advertise团体属性控制路由信息传递

接下来的需求是：10.0.100.4/32这条路由信息只能够被R2接收到，而不能被其他路由器接收到。为此，可以利用团体属性No_Advertise来满足这个需求，因为路由器接收到一条携带No-Advertise团体属性的路由后，不会将它发布给任何BGP对等体。

在R1上使用前缀列表方法来匹配路由10.0.100.4/32。

```
[R1]ip ip-prefix 3 permit 10.0.100.4 32
```

在R1上创建新的路由策略的Node。

```
[R1]route-policy 1 permit node 12
```

```
[R1-route-policy]if-match ip-prefix 3
```

```
[R1-route-policy]apply community no-advertise
```

配置完成后，在R2、R3、R4和R5上查看BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.12.1	0		0	100i
*>	10.0.100.2/32	10.0.12.1	0		0	100i
*>	10.0.100.3/32	10.0.12.1	0		0	100i
*>	10.0.100.4/32	10.0.12.1	0		0	100i
*>	10.0.100.5/32	10.0.12.1	0		0	100i

可以看到 , R2接收到了10.0.100.4/32这条路由信息。

[R3]display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.23.2	0		100	0 (2001)
100i						
*>i	10.0.100.2/32	10.0.23.2	0		100	0 (2001)
100i						
*>i	10.0.100.5/32	10.0.23.2	0		100	0 (2001)
100i						

可以看到 , R3未接收到10.0.100.4/32这条路由信息。

[R4]display bgp routing-table

BGP Local router ID is 10.0.4.4

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.2/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.3/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.5/32	10.0.24.2	0	100	0	100i

可以看到 , R4未接收到10.0.100.4/32这条路由信息。

[R5]display bgp routing-table

BGP Local router ID is 10.0.5.5

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.45.4			0	200 100i
*>	10.0.100.5/32	10.0.45.4			0	200 100i

可以看到 , R5未接收到10.0.100.4/32这条路由。

在R2上查看携带团体属性的BGP路由表。

[R2]display bgp routing-table community

BGP Local router ID is 10.0.2.2

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Community
*>	10.0.100.2/32	10.0.12.1	0		0	no-export
*>	10.0.100.3/32	10.0.12.1	0		0	no-export-
						subconfed
*>	10.0.100.4/32	10.0.12.1	0		0	no-advertise

可以看到，R2上10.0.100.4/32这条路由携带了No_Advertise团体属性，因此这条路由不会被R2传递给任何BGP邻居。

6. 使用自定义团体属性控制路由信息传递

接下来的需求是：10.0.100.5/32这条路由信息只能够被R2和R3接收到，而不能被R4和R5接收到。经过分析可知，使用Well-known团体属性难以满足这样的需求，所以需要为路由信息添加自定义团体属性，然后通过匹配自定义团体属性来控制路由信息的传递。

在R1上使用前缀列表方法来匹配路由10.0.100.5/32。

```
[R1]ip ip-prefix 4 permit 10.0.100.5 32
```

在R1上创建新的路由策略的Node。

```
[R1]route-policy 1 permit node 13
```

```
[R1-route-policy]if-match ip-prefix 4
```

```
[R1-route-policy]apply community 100:1
```

配置完成后，在R2、R3、R4和R5上查看BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn

```
*> 10.0.100.1/32    10.0.12.1  0      0      100i
*> 10.0.100.2/32    10.0.12.1  0      0      100i
*> 10.0.100.3/32    10.0.12.1  0      0      100i
*> 10.0.100.4/32    10.0.12.1  0      0      100i
*> 10.0.100.5/32    10.0.12.1  0      0      100i
```

可以看到，R2接收到了10.0.100.5/32这条路由信息。

```
[R3]display bgp routing-table
```

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.23.2	0	100	0	(2001)
100i						
*>i	10.0.100.2/32	10.0.23.2	0	100	0	(2001)
100i						
*>i	10.0.100.5/32	10.0.23.2	0	100	0	(2001)
100i						

可以看到，R3接收到了10.0.100.5/32这条路由信息。

```
[R4]display bgp routing-table
```

BGP Local router ID is 10.0.4.4

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn


```
*>i 10.0.100.1/32 10.0.24.2 0 100 0 100i
*>i 10.0.100.2/32 10.0.24.2 0 100 0 100i
*>i 10.0.100.3/32 10.0.24.2 0 100 0 100i
*>i 10.0.100.5/32 10.0.24.2 0 100 0 100i
```

可以看到，R4接收到了10.0.100.5/32这条路由信息。

```
[R5]display bgp routing-table
```

```
BGP Local router ID is 10.0.5.5
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 2
```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 10.0.100.1/32	10.0.45.4			0	200 100i
*> 10.0.100.5/32	10.0.45.4			0	200 100i

可以看到，R5接收到了10.0.100.5/32这条路由。

由于R2、R3、R4和R5都接收到了10.0.100.5/32这条路由信息，这说明需求并没有得到实现。原来，自定义团体属性在缺省情况下没有相应的动作，仅仅是为路由信息添加了标识而已。

在R4上匹配自定义团体属性为100:1的路由。

```
[R4]ip community-filter 1 permit 100:1
```

然后在R4上创建Route-Policy。

```
[R4]route-policy 1 deny node 10
```

```
[R4-route-policy]if-match community-filter 1
```

```
[R4-route-policy]route-policy 1 permit node 20
```

R4在接收R2传递过来的路由时调用Route-Policy。

```
[R4]bgp 2001
```

```
[R4-bgp]peer 10.0.24.2 route-policy 1 import
```

配置完成后，在R2、R3、R4、R5上查看BGP路由表。

[R2]display bgp routing-table

BGP Local router ID is 10.0.2.2

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.12.1	0		0	100i
*>	10.0.100.2/32	10.0.12.1	0		0	100i
*>	10.0.100.3/32	10.0.12.1	0		0	100i
*>	10.0.100.4/32	10.0.12.1	0		0	100i
*>	10.0.100.5/32	10.0.12.1	0		0	100i

可以看到，R2接收到了10.0.100.5/32这条路由信息。

[R3]display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.23.2	0		100	0 (2001)
100i						
*>i	10.0.100.2/32	10.0.23.2	0		100	0 (2001)
100i						
*>i	10.0.100.5/32	10.0.23.2	0		100	0 (2001)
100i						

可以看到，R3接收到了10.0.100.5/32这条路由信息。

```
[R4]display bgp routing-table
```

BGP Local router ID is 10.0.4.4

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.1/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.2/32	10.0.24.2	0	100	0	100i
*>i	10.0.100.3/32	10.0.24.2	0	100	0	100i

可以看到，R4未接收到10.0.100.5/32这条路由。

```
[R5]display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.45.4			0	200 100i

可以看到，R5未接收到10.0.100.5/32这条路由信息。

在R2与R3上查看携带团体属性的BGP路由表。

```
[R2]display bgp routing-table community
```

BGP Local router ID is 10.0.2.2

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Community
*>	10.0.100.2/32	10.0.12.1	0		0	no-export
*>	10.0.100.3/32	10.0.12.1	0		0	no-export-
subconfed						
*>	10.0.100.4/32	10.0.12.1	0		0	no-advertise
*>	10.0.100.5/32	10.0.12.1	0		0	<100:1>

可以看到，R2上10.0.100.5/32这条路由携带了自定义的团体属性100 : 1。

[R3]display bgp routing-table community

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Community
*>i	10.0.100.2/32	10.0.23.2	0	100	0	no-export
*>i	10.0.100.5/32	10.0.23.2	0	100	0	<100:1>

可以看到，R3上10.0.100.5/32这条路由也携带了自定义的团体属性100 : 1。至此，所有的网络需求都得到了满足。

思考

带有No-Advertise团体属性的路由可以被传递给联盟EBGP对等体吗？

3.11 BGP 路由反射器

原理概述

缺省情况下，路由器从它的一个 IBGP 对等体那里接收到的路由条目不会被该路由器再传递给其他IBGP对等体，这个原则称为BGP水平分割原则，该原则的根本作用是防止 AS 内部的 BGP 路由环路。因此，在 AS 内部，一般需要每台路由器都运行 BGP协议并建立全互联的IBGP对等体关系，这样才能避免BGP路由黑洞等问题。对于有 n 个BGP路由器的AS来说，全互联的IBGP对等体关系将有 $n \times (n-1) \div 2$ 个。对于大型AS来说，数量众多的IBGP对等体关系将导致配置和维护的工作量都非常大，且人为出错的可能性也随之增加。

解决上述问题的方法之一就是使用 BGP 路由反射器。BGP 路由反射器的使用，可以在很大程度上减少大型AS中IBGP对等体关系的数量并简化相应的配置和维护工作。BGP路由反射器是AS内部IBGP网络环境中的一种特殊角色，其他的角色还有反射器的客户端和非客户端。一个反射器和它所有的客户端一起被统称为一个Cluster；客户端与它的反射器建立的是 IBGP 对等体关系；客户端之间无需建立 IBGP 对等体关系；非客户端和反射器建立的是 IBGP 对等体关系；非客户端之间需要建立全互连的 IBGP 对等体关系；非客户端和客户端之间无需建立 IBGP 对等体关系；一个 AS 内部可以有多个Cluster；一个Cluster中可以有多台反射器。另外，EBGP对等体之间是不存在BGP路由反射器的概念的。

BGP路由反射器在反射路由的时候遵循的原则是：从一个非客户端那里接收到的路由，反射器会将它只传递给所有的客户端；从一个客户端那里接收到的路由，反射器会将它传递给所有其他的客户端以及所有的非客户端；从EBGP对等体那里接收到的路由，反射器会将它传递给所有的客户端和非客户端。

实验目的

- 理解BGP 路由反射器的应用场景

- 理解BGP 路由反射器的工作原理
- 掌握BGP 路由反射器的基本配置方法

实验内容

实验拓扑如图3-11所示，实验编址如表3-11所示。本实验网络包含了两个AS，两个Cluster。R1、R2、R3 属于Cluster 1，R4、R5、R6属于Cluster 2，R7 不属于任何Cluster。在AS 100 内部，所有路由器都运行OSPF 协议作为IGP，并将各自的Loopback 0 接口宣告进OSPF 进程中，使得各路由器可以使用Loopback 0 接口来建立全互联的IBGP 对等体关系。然后，为了减少配置工作量，决定使用路由反射器，要求是：在Cluster 1 中，R1 为路由反射器，R2 和 R3 为其客户端；在Cluster 2 中，R4 为路由反射器，R5、R6为其客户端；R7为非客户端；R1与R8为EBGP对等体关系。

实验拓扑

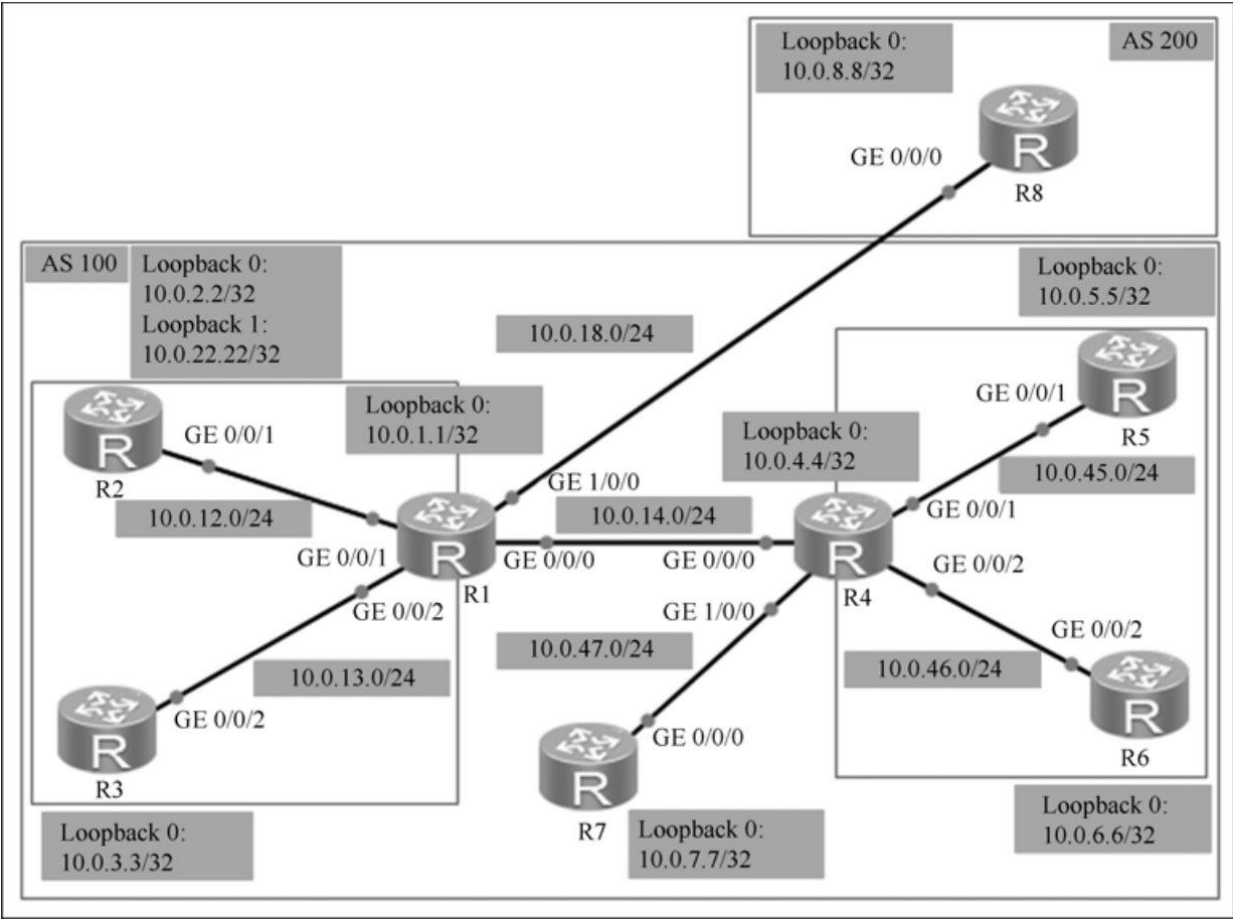


图3-11 BGP路由反射器

实验编址表

表3-11 实验编址

figure_0253_0085

实验步骤

1. 基本配置

根据图3-11和表3-11进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

<R1>ping -c 1 10.0.12.2

PING 10.0.12.2: 56 data bytes, press CTRL_C to break

Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=160 ms

--- 10.0.12.2 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 160/160/160 ms

其余直连网段的连通性测试过程在此省略。

2 . 配置OSPF路由协议

为了使AS 100 内部的路由器之间都能够使用Loopback 0 接口建立IBGP 对等体关系，需要在每台路由器（R8 除外）上配置 OSPF 路由协议，并将 Loopback 0 接口通告进OSPF进程。

```
[R1]ospf 1 router-id 10.0.1.1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.14.0 0.0.0.255
```

```
[R2]ospf 1 router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R3]ospf 1 router-id 10.0.3.3
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
```

```
[R4]ospf 1 router-id 10.0.4.4
```

```
[R4-ospf-1]area 0
```

```
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
```

```
[R4-ospf-1-area-0.0.0.0]network 10.0.14.0 0.0.0.255
```



```
[R4-ospf-1-area-0.0.0.0]network 10.0.45.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.46.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.47.0 0.0.0.255
[R5]ospf 1 router-id 10.0.5.5
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]network 10.0.5.5 0.0.0.0
[R5-ospf-1-area-0.0.0.0]network 10.0.45.0 0.0.0.255
[R6]ospf 1 router-id 10.0.6.6
[R6-ospf-1]area 0
[R6-ospf-1-area-0.0.0.0]network 10.0.6.6 0.0.0.0
[R6-ospf-1-area-0.0.0.0]network 10.0.46.0 0.0.0.255
[R7]ospf 1 router-id 10.0.7.7
[R7-ospf-1]area 0
[R7-ospf-1-area-0.0.0.0]network 10.0.7.7 0.0.0.0
[R7-ospf-1-area-0.0.0.0]network 10.0.47.0 0.0.0.255
```

3 . 配置BGP路由协议

配置BGP 路由协议，在AS 100 内部的每台路由器上使用Loopback 0 接口建立全互联的IBGP 对等体关系，并通告各自的Loopback 0 接口到BGP 进程中。

```
[R1]bgp 100
[R1-bgp]router-id 10.0.1.1
[R1-bgp]peer 10.0.2.2 as-number 100
[R1-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R1-bgp]peer 10.0.2.2 next-hop-local
[R1-bgp]peer 10.0.3.3 as-number 100
[R1-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R1-bgp]peer 10.0.3.3 next-hop-local
```

```
[R1-bgp]peer 10.0.4.4 as-number 100
[R1-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R1-bgp]peer 10.0.4.4 next-hop-local
[R1-bgp]peer 10.0.5.5 as-number 100
[R1-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R1-bgp]peer 10.0.5.5 next-hop-local
[R1-bgp]peer 10.0.6.6 as-number 100
[R1-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R1-bgp]peer 10.0.6.6 next-hop-local
[R1-bgp]peer 10.0.7.7 as-number 100
[R1-bgp]peer 10.0.7.7 connect-interface LoopBack 0
[R1-bgp]peer 10.0.7.7 next-hop-local
[R1-bgp]network 10.0.1.1 32
[R2]bgp 100
[R2-bgp]router-id 10.0.2.2
[R2-bgp]peer 10.0.1.1 as-number 100
[R2-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R2-bgp]peer 10.0.3.3 as-number 100
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R2-bgp]peer 10.0.4.4 as-number 100
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R2-bgp]peer 10.0.5.5 as-number 100
[R2-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R2-bgp]peer 10.0.6.6 as-number 100
[R2-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R2-bgp]peer 10.0.7.7 as-number 100
[R2-bgp]peer 10.0.7.7 connect-interface LoopBack 0
```

```
[R2-bgp]network 10.0.2.2 32
[R3]bgp 100
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.1.1 as-number 100
[R3-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R3-bgp]peer 10.0.2.2 as-number 100
[R3-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R3-bgp]peer 10.0.4.4 as-number 100
[R3-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R3-bgp]peer 10.0.5.5 as-number 100
[R3-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R3-bgp]peer 10.0.6.6 as-number 100
[R3-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R3-bgp]peer 10.0.7.7 as-number 100
[R3-bgp]peer 10.0.7.7 connect-interface LoopBack 0
[R3-bgp]network 10.0.3.3 32
[R4]bgp 100
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.1.1 as-number 100
[R4-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R4-bgp]peer 10.0.2.2 as-number 100
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R4-bgp]peer 10.0.3.3 as-number 100
[R4-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R4-bgp]peer 10.0.5.5 as-number 100
[R4-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R4-bgp]peer 10.0.6.6 as-number 100
```

```
[R4-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R4-bgp]peer 10.0.7.7 as-number 100
[R4-bgp]peer 10.0.7.7 connect-interface LoopBack 0
[R4-bgp]network 10.0.4.4 32
[R5]bgp 100
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.1.1 as-number 100
[R5-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R5-bgp]peer 10.0.2.2 as-number 100
[R5-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R5-bgp]peer 10.0.3.3 as-number 100
[R5-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R5-bgp]peer 10.0.4.4 as-number 100
[R5-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R5-bgp]peer 10.0.6.6 as-number 100
[R5-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R5-bgp]peer 10.0.7.7 as-number 100
[R5-bgp]peer 10.0.7.7 connect-interface LoopBack 0
[R5-bgp]network 10.0.5.5 32
[R6]bgp 100
[R6-bgp]router-id 10.0.6.6
[R6-bgp]peer 10.0.1.1 as-number 100
[R6-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R6-bgp]peer 10.0.2.2 as-number 100
[R6-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R6-bgp]peer 10.0.3.3 as-number 100
[R6-bgp]peer 10.0.3.3 connect-interface Loopback 0
```

```
[R6-bgp]peer 10.0.4.4 as-number 100
[R6-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R6-bgp]peer 10.0.5.5 as-number 100
[R6-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R6-bgp]peer 10.0.7.7 as-number 100
[R6-bgp]peer 10.0.7.7 connect-interface LoopBack 0
[R6-bgp]network 10.0.6.6 32
[R7]bgp 100
[R7-bgp]router-id 10.0.7.7
[R7-bgp]peer 10.0.1.1 as-number 100
[R7-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R7-bgp]peer 10.0.2.2 as-number 100
[R7-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R7-bgp]peer 10.0.3.3 as-number 100
[R7-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R7-bgp]peer 10.0.4.4 as-number 100
[R7-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R7-bgp]peer 10.0.5.5 as-number 100
[R7-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R7-bgp]peer 10.0.6.6 as-number 100
[R7-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R7-bgp]network 10.0.7.7 32
```

在R1和R8之间使用直连物理接口建立EBGP对等体关系，并通告R8的Loopback 0接口到BGP进程中。

```
[R1]bgp 100
[R1-bgp]peer 10.0.18.8 as-number 200
[R8]bgp 200
```

```
[R8-bgp]router-id 10.0.8.8
```

```
[R8-bgp]peer 10.0.18.1 as-number 100
```

```
[R8-bgp]network 10.0.8.8 32
```

查看 AS 100 内部的每台路由器上的 BGP 路由表（这里仅以 R2 为例），同时查看R8的BGP路由表。

```
<R2>display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 8

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
	i 10.0.1.1/32	10.0.1.1	0	100	0	i
*>	10.0.2.2/32	0.0.0.0	0		0	i
i	10.0.3.3/32	10.0.3.3	0	100	0	i
i	10.0.4.4/32	10.0.4.4	0	100	0	i
i	10.0.5.5/32	10.0.5.5	0	100	0	i
i	10.0.6.6/32	10.0.6.6	0	100	0	i
i	10.0.7.7/32	10.0.7.7	0	100	0	i
*>i	10.0.8.8/32	10.0.1.1	0	100	0	200i

```
<R8>dispaly bgp routing-table
```

BGP Local router ID is 10.0.8.8

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn

```
*> 10.0.1.1/32    10.0.18.1  0      0      100i
*> 10.0.8.8/32    0.0.0.0   0      0      i
```

可以看到，AS 100 内部的路由器都已经接收到了关于 10.0.8.8/32 的路由信息。R8只接收到了关于 10.0.1.1/32 的路由信息，而没有接收到关于 AS 100 内部其他路由器的Loopback 0的路由信息，这是因为AS 100 内部OSPF 路由协议的优先级要高于BGP 路由协议的优先级，于是R1 就不会将除了本地起源（即下一跳为 0.0.0.0）的路由之外的其他路由信息传递给R8。显然，这会导致R8 与AS 100 内部的路由器的互通问题。为了使R8 能够与AS 100 内部的所有路由器的Loopback 0 接口所在的网络进行通信，可以在R8上配置一条聚合的静态路由，下一跳为 10.0.18.1。

```
[R8]ip route-static 10.0.0.0 20 10.0.18.1
```

配置完成后，网络通信正常，但是整体配置工作量较大。

4．配置BGP路由反射器

对于大型网络来讲，使用路由反射器可以大大减少 IBGP 对等体关系的数量。路由反射器的使用，会明显减少配置工作量，人为出错的可能性也会大大降低。

下面将进行关于路由反射器的实验，首先清除之前各路由器上的BGP进程。在此需要提醒读者的是，在实际场景中如果这样操作，将会导致网络瘫痪一段时间。

以R1为例，清除原来的BGP进程。

```
[R1]undo bgp 100
```

```
Warning: All BGP configurations will be deleted. Continue? [Y/N]: y
```

R2和R3是路由反射器R1的客户端，它们只需和R1配置成IBGP对等体关系即可，R2 和R3 之间无需配置为IBGP 对等体关系。另外，将R2 的Loopback 1（10.0.22.22/32）接口通告进BGP进程。

```
[R2]bgp 100
```

```
[R2-bgp]router-id 10.0.2.2
[R2-bgp]peer 10.0.1.1 as-number 100
[R2-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R2-bgp]network 10.0.2.2 32
[R2-bgp]network 10.0.22.22 32
[R3]bgp 100
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.1.1 as-number 100
[R3-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R3-bgp]network 10.0.3.3 32
```

配置R1为R2和R3的路由反射器，配置Cluster-ID为1，配置R1与R4之间的IBGP对等体关系，配置R1与R8之间的EBGP对等体关系。

```
[R1]bgp 100
[R1-bgp]router-id 10.0.1.1
[R1-bgp]group in_1
[R1-bgp]peer 10.0.2.2 group in_1
[R1-bgp]peer 10.0.3.3 group in_1
[R1-bgp]peer in_1 reflect-client
[R1-bgp]peer in_1 next-hop-local
[R1-bgp]reflector cluster-id 1
[R1-bgp]peer 10.0.4.4 as-number 100
[R1-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R1-bgp]peer 10.0.4.4 next-hop-local
[R1-bgp]peer 10.0.18.8 as-number 200
```

R5和R6是路由反射器R4的客户端，它们只需和R4配置成IBGP对等体关系即可，R5和R6之间无需配置为IBGP对等体关系。

```
[R5]bgp 100
```



```
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.4.4 as-number 100
[R5-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R5-bgp]network 10.0.5.5 32
[R6]bgp 100
[R6-bgp]router-id 10.0.6.6
[R6-bgp]peer 10.0.4.4 as-number 100
[R6-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R6-bgp]network 10.0.6.6 32
```

配置R4为R5和R6的路由反射器，配置Cluster-ID为2，配置R4与R1之间的IBGP对等体关系，配置R4与R7之间的IBGP对等体关系。

```
[R4]bgp 100
[R4-bgp]router-id 10.0.4.4
[R4-bgp]group in_2
[R4-bgp]peer 10.0.5.5 group in_2
[R4-bgp]peer 10.0.6.6 group in_2
[R4-bgp]peer in_2 reflect-client
[R4-bgp]reflector cluster-id 2
[R4-bgp]peer 10.0.1.1 as-number 100
[R4-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R4-bgp]peer 10.0.7.7 as-number 100
[R4-bgp]peer 10.0.7.7 connect-interface LoopBack 0
```

R7是非客户端路由器，配置R7与R4之间的IBGP对等体关系。

```
[R7]bgp 100
[R7-bgp]router-id 10.0.7.7
[R7-bgp]peer 10.0.4.4 as-number 100
[R7-bgp]peer 10.0.4.4 connect-interface LoopBack 0
```

```
[R7-bgp]network 10.0.7.7 32
```

配置R8与R1之间的EBGP对等体关系。

```
[R8]bgp 200
```

```
[R8-bgp]router-id 10.0.8.8
```

```
[R8-bgp]peer 10.0.18.1 as-number 100
```

```
[R8-bgp]network 10.0.8.8 255.255.255.255
```

5. 验证路由反射器的反射原理

根据 IBGP 的水平分割原则，R1 从 IBGP 对等体 R2 接收到 BGP 路由条目10.0.22.22/32后，不会再传递给其他IBGP对等体，因此R3和R4就应该接收不到这条路由。当然，R1可以将此路由传递给EBGP对等体R8。

在R3、R4、R8上查看BGP路由表。

```
[R3]display bgp routing-table
```

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.3.3/32	0.0.0.0	0		0	i
*>i	10.0.8.8/32	10.0.1.1	0	100	0	200i
*>i	10.0.22.22/32	10.0.2.2	0	100	0	i

```
[R4]display bgp routing-table
```

BGP Local router ID is 10.0.4.4

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	10.0.5.5/32	10.0.5.5	0	100	0	i
i	10.0.6.6/32	10.0.6.6	0	100	0	i
i	10.0.7.7/32	10.0.7.7	0	100	0	i
*>i	10.0.8.8/32	10.0.1.1	0	100	0	200i
*>i	10.0.22.22/32	10.0.2.2	0	100	0	i

<R8>display bgp routing-table

BGP Local router ID is 10.0.8.8

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.8.8/32	0.0.0.0	0		0	i
*>	10.0.22.22/32	10.0.18.1			0	100i

观察发现，R3、R4、R8的BGP路由表中都存在关于10.0.22.22/32的路由信息。由此可见，R1将10.0.22.22/32这条路由传递给了R3和R4，不再受BGP水平分割原则的限制，同时，这条路由也被R1传递给了EBGP对等体R8。实验表明，BGP路由反射器从它的一个客户端接收到路由之后，会将该路由反射给它的其他客户端、非客户端，以及EBGP对等体。

在R5、R6、R7上查看BGP路由表。

[R5]display bgp routing-table

BGP Local router ID is 10.0.5.5

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.5.5/32	0.0.0.0	0		0	i
*>i	10.0.8.8/32	10.0.1.1	0	100	0	200i
*>i	10.0.22.22/32	10.0.2.2	0	100	0	i

[R6]display bgp routing-table

BGP Local router ID is 10.0.6.6

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.6.6/32	0.0.0.0	0		0	i
*>I	10.0.8.8/32	10.0.1.1	0	100	0	200i
*>i	10.0.22.22/32	10.0.2.2	0	100	0	i

[R7]display bgp routing-table

BGP Local router ID is 10.0.7.7

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.7.7/32	0.0.0.0	0		0	i

可以看到，R4将10.0.22.22/32这条路由传递给了R5和R6，但是没有传递给R7，说明路由反射器会把从非客户端收到的路由传递给客户端，但不会传递给其他非客户端。由于路由反射器认为非客户端之间

应该是存在 IBGP 对等体关系的，所以路由反射器和非客户端之间依然遵循水平分割原则。

路由反射器R4认为R1与R7之间应该存在IBGP对等体关系，所以没有将从非客户端R1接收到的BGP路由传递给R7。但实际上，R1与R7之间并没有被配置为IBGP对等体关系，这就导致了R7的BGP路由表中并没有关于10.0.22.22/32的路由。解决此问题的办法就是将R1和R7配置为IBGP对等体关系。

```
[R1]bgp 100
[R1-bgp]peer 10.0.7.7 as-number 100
[R1-bgp]peer 10.0.7.7 connect-interface LoopBack 0
[R1-bgp]peer 10.0.7.7 next-hop-local
[R7]bgp 100
[R7-bgp]peer 10.0.1.1 as-number 100
[R7-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R7-bgp]peer 10.0.1.1 next-hop-local
```

重新查看R7的BGP路由表。

```
[R7]display bgp routing-table
BGP Local router ID is 10.0.7.7
Status codes: *- valid, >- best, d - damped,
               h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete
```

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.7.7/32	0.0.0.0	0		0	i
*>i	10.0.8.8/32	10.0.1.1	0	100	0	200i
*>i	10.0.22.22/32	10.0.2.2	0	100	0	i

可以看到，R7现在接收到了10.0.22.22/32这条路由。

在R7上查看10.0.22.22/32这条路由的详细信息。

```
<R7>display bgp routing-table 10.0.22.22
```

```
BGP local router ID : 10.0.7.7
```

```
Local AS number : 100
```

```
Paths: 1 available, 1 best, 1 select
```

```
BGP routing table entry information of 10.0.22.22/32:
```

```
From: 10.0.1.1 (10.0.1.1)
```

```
Route Duration: 00h07m28s
```

```
.....
```

可以看到，R7上的10.0.22.22/32这条路由信息是从R1（10.0.1.1）传递过来的，而不是从R4传递过来的，这说明路由反射器和非客户端之间是遵循水平分割原则的。

在R1、R2、R3、R4上查看接收到的关于10.0.8.8/32的路由信息。

```
[R1]display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 5
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	10.0.2.2/32	10.0.2.2	0	100	0	i
i	10.0.3.3/32	10.0.3.3	0	100	0	i
i	10.0.7.7/32	10.0.7.7	0	100	0	i
*>	10.0.8.8/32	10.0.18.8	0		0	200i
*>i	10.0.22.22/32	10.0.2.2	0	100	0	i

```
[R2]display bgp routing-table
```

```
BGP Local router ID is 10.0.2.2
```

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.2.2/32	0.0.0.0	0		0	i
*>i	10.0.8.8/32	10.0.1.1	0	100	0	200i
*>	10.0.22.22/32	0.0.0.0	0		0	i

[R3]display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.3.3/32	0.0.0.0	0		0	i
*>i	10.0.8.8/32	10.0.1.1	0	100	0	200i
*>i	10.0.22.22/32	10.0.2.2	0	100	0	i

[R4]display bgp routing-table

BGP Local router ID is 10.0.4.4

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	10.0.5.5/32	10.0.5.5	0	100	0	i
i	10.0.6.6/32	10.0.6.6	0	100	0	i

```
i 10.0.7.7/32      10.0.7.7  0    100    0          i
*>i 10.0.8.8/32     10.0.1.1  0    100    0        200i
*>i 10.0.22.22/32   10.0.2.2  0    100    0          i
```

可以看到，R1从EBGP对等体R8接收到关于10.0.8.8/32的路由之后，将这条路由传递给了R2、R3、R4，说明路由反射器会把从EBGP对等体接收到的路由传递给它的客户端和非客户端。

6 . BGP路由反射器的防环原理

在前面的配置中，R1 上使用了命令peer in_1 reflect-client。这条命令的含义是指定BGP对等体组in_1中的路由器（即R2和R3）为R1的客户端，从相反的角度来说，也就是R1被指定成为BGP对等体组in_1中的路由器（即R2和R3）的路由反射器。

在R1、R3、R4上查看10.0.22.22/32这条路由的具体属性。

```
[R1]display bgp routing-table 10.0.22.22
```

```
BGP local router ID : 10.0.1.1
```

```
Local AS number : 100
```

```
Paths: 1 available, 1 best, 1 select
```

```
BGP routing table entry information of 10.0.22.22/32:
```

```
RR-client route.
```

```
From: 10.0.2.2 (10.0.2.2)
```

```
Route Duration: 00h25m36s
```

```
Relay IP Nexthop: 10.0.12.2
```

```
Relay IP Out-Interface: GigabitEthernet0/0/1
```

```
Original nexthop: 10.0.2.2
```

```
Qos information : 0x0
```

```
AS-path Nil, origin igp, MED 0, localpref 100, pref-val 0, valid,
internal, best, select, active, pre 255, IGP cost 1
```

```
Advertised to such 5 peers:
```


10.0.18.8

10.0.3.3

10.0.2.2

10.0.4.4

10.0.7.7

[R3]display bgp routing-table 10.0.22.22

BGP local router ID : 10.0.3.3

.....

select, active, pre 255, IGP cost 2

Originator: 10.0.2.2

Cluster list: 0.0.0.1

Not advertised to any peer yet

[R4]display bgp routing-table 10.0.22.22

BGP local router ID : 10.0.4.4

.....

select, active, pre 255, IGP cost 2

Originator: 10.0.2.2

Cluster list: 0.0.0.1

Advertised to such 2 peers:

10.0.5.5

10.0.6.6

可以观察到，在R1、R3、R4上关于10.0.22.22/32的路由的属性是有所区别的。在R3和R4上关于此路由多了Originator和Cluster List这两个属性。Originator属性的作用是防止路由在反射器和客户端/非客户端之间出现环路。路由第一次被反射的时候，反射器会将Originator属性加入这条路由中，用BGP Router-ID表示，用来标识这条路由的起源路由器。如果路由中已经存在Originator属性，则反射器不会创建新的

Originator。当其他BGP对等体接收到这条路由时，将对收到的Originator和本地的BGP Router-ID 进行比较，如果两者相同，BGP对等体将会忽略掉这条路由，不做处理。Originator 属性可以传递给其他的Cluster，路由在AS内传递时该属性不会丢失。

Cluster List 属性可用来防止Cluster间的路由环路。当路由反射器在客户端之间或客户端与非客户端之间反射路由时，会将自己的Cluster-ID 添加到Cluster List 中。路由反射器接收到 BGP 路由后会去检查其中的 Cluster List，如果发现自己的 Cluster-ID 位于Cluster List 中，则表明出现了路由环路，因而会忽略该路由。AS 内的每台路由反射器都采用了一个唯一的4个8位组来标识Cluster-ID，如果Cluster中包含了多台路由反射器，则必须以手工的方式为每台路由反射器配置Cluster-ID。

在R5上查看10.0.22.22/32这条路由的具体属性。

```
[R5]display bgp routing-table 10.0.22.22
```

```
BGP local router ID : 10.0.5.5
```

```
.....
```

```
Originator: 10.0.2.2
```

```
Cluster list: 0.0.0.2, 0.0.0.1
```

```
Not advertised to any peer yet
```

可以看到，Cluster List 中含有两个Cluster-ID：0.0.0.2 和0.0.0.1，这两个Cluster-ID都是在配置路由反射器时定义的。

在R8上查看10.0.22.22/32这条路由的具体属性。

```
<R8>display bgp routing-table 10.0.22.22
```

```
BGP local router ID : 10.0.8.8
```

```
Local AS number : 200
```

```
Paths: 1 available, 1 best, 1 select
```

```
BGP routing table entry information of 10.0.22.22/32:
```

```
From: 10.0.18.1 (10.0.1.1)
```

Route Duration: 02h23m28s

Direct Out-interface: GigabitEthernet0/0/0

Original nexthop: 10.0.18.1

Qos information : 0x0

AS-path 100, origin igp, pref-val 0, valid, external, best, select, active, pre 255

Not advertised to any peer yet

可以看到，R8 上的10.0.22.22/32 这条路由没有Cluster-ID 和Cluster List 属性信息，说明Cluster-ID 和Cluster List 属性不会通告给EBGP 对等体。

思考

一个Cluster中的客户端能作为另一个Cluster中的反射器吗？能作为另一个Cluster中的客户端吗？

3.12 BGP 路由黑洞

原理概述

在BGP 网络中，报文穿越Transit AS 时，有可能会被Transit AS 中未运行BGP 协议的路由器接收到。由于这样的路由器没有AS间的BGP 路由信息，报文有可能会被直接丢弃，然后路由器会向报文的源 IP 地址发送 ICMP Unreachable 消息。然而，由于这样的路由器上没有运行BGP协议，很可能导致该路由器上也不存在去往报文的源IP地址的路由，从而使得ICMP Unreachable消息也无法被发送出去。如此一来，报文就无声无息地消失在了这样的路由器上。这种现象被形象地称为BGP路由黑洞。

解决BGP路由黑洞问题的方法之一是采用IBGP与IGP的同步机制。同步机制要求：路由器在接收到一条IBGP对等体发送来的路由后，必须检查自己的IGP路由表，只有在自己的IGP路由表中也存在关于这条路由的信息时，才会将该BGP路由发布给EBGP对等体。为此，可以将BGP协议引入IGP协议中，让没有运行BGP协议的路由器也能够获得BGP路由。然而，这种方法实现起来并不容易，同时也有很多缺点，因为IGP一般都不具备管理和维护大量被引入的BGP路由的能力，BGP路由的不稳定情况也会影响到IGP协议，另外，IGP协议的路由策略和控制工具也远没有BGP协议那样丰富。

解决BGP 路由黑洞问题还有许多其他方法，例如，可以让Transit AS 中的每台路由器都运行BGP 协议，并建立全互联的IBGP 邻居关系；可以使用GRE（Generic Routing Encapsulation）隧道技术，在Transit AS 中的BGP 对等体之间建立逻辑上的连接，使得报文的路径在逻辑上不经过未运行 BGP 协议的路由器；还可以使用 MPLS 技术，使得报文在Transit AS 内部不通过IP 协议进行传输，从而避免在未运行BGP 协议的路由器上由于没有目标网络的IP路由而将报文丢弃的情况。

实验目的

- 理解BGP 路由黑洞的概念和成因
- 理解IBGP 与IGP 的同步机制
- 理解使用IBGP 全互联方式解决BGP 路由黑洞问题的原理
- 理解使用GRE 隧道解决BGP 路由黑洞问题的原理
- 掌握上述解决BGP 路由黑洞问题的配置方法

实验内容

实验拓扑如图3-12 所示，实验编址如表3-12 所示。本实验网络中，假定AS 20 为运营商网络，AS 10 和 AS 30 分别为企业分公司 A 和分公司 B 的网络，R1 和 R5 的Loopback 0接口分别模拟了分公司A 的

内部网络和分公司B 的内部网络。R1、R2、R4、R5 运行 BGP 协议，AS 20 内部使用 OSPF 协议作为 IGP。网络需求是：实现分公司 A 的内部网络与分公司B的内部网络之间的正常通信。

实验拓扑

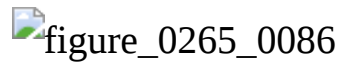
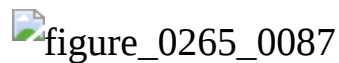


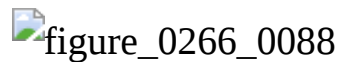
图3-12 BGP路由黑洞

实验编址表

表3-12 实验编址



(续表)



实验步骤

1. 基本配置

根据图3-12和表3-12进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=120 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 120/120/120 ms
```

其余直连网段的连通性测试过程在此省略。

2. 配置OSPF和BGP路由协议

在AS 20 中配置OSPF 协议作为IGP 协议。

```
[R2]ospf router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
[R2-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
[R3]ospf router-id 10.0.3.3
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R4]ospf router-id 10.0.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
[R4-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
```

配置完成后，在R3上查看OSPF邻居信息。

```
[R3]display ospf peer
```

OSPF Process 1 with Router ID 10.0.3.3

Neighbors

Area 0.0.0.0 interface 10.0.34.3(GigabitEthernet0/0/0)'s neighbors

Router ID: 10.0.4.4 Address: 10.0.34.4

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.34.4 BDR: 10.0.34.3 MTU: 0

Dead timer due in 35 sec

Retrans timer interval: 5

Neighbor is up for 00:02:06

Authentication Sequence: [0]

Neighbors

Area 0.0.0.0 interface 10.0.23.3(GigabitEthernet0/0/1)'s neighbors

Router ID: 10.0.2.2 Address: 10.0.23.2

State: Full Mode:Nbr is Slave Priority: 1

DR:10.0.23.3 BDR:10.0.23.2 MTU:0

Dead timer due in 40 sec

Retrans timer interval: 5

Neighbor is up for 00:02:49

Authentication Sequence: [0]

可以看到，R3与R2、R3与R4的邻居状态均为Full，说明OSPF邻接关系已经成功建立。

在R1、R2、R4、R5上配置BGP协议。

```
[R1]bgp 10
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 20
```

```
[R1-bgp]network 10.0.1.1 255.255.255.255
```

```
[R2]bgp 20
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 10
```

```
[R2-bgp]peer 10.0.4.4 as-number 20
```

```
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack 0
```

```
[R2-bgp]peer 10.0.4.4 next-hop-local
```

```
[R4]bgp 20
```

```
[R4-bgp]router-id 10.0.4.4
```

```
[R4-bgp]peer 10.0.2.2 as-number 20
```

```
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack 0
```

```
[R4-bgp]peer 10.0.2.2 next-hop-local
```

```
[R4-bgp]peer 10.0.45.5 as-number 30
```

```
[R5]bgp 30
```

```
[R5-bgp]router-id 10.0.5.5
```

```
[R5-bgp]peer 10.0.45.4 as-number 20
```

```
[R5-bgp]network 10.0.5.5 255.255.255.255
```

配置完成后，在R2上查看BGP邻居信息。

```
[R2]display bgp peer
```

```
BGP local router ID : 10.0.2.2
```

```
Local AS number : 20
```

```
Total number of peers : 2      Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.4.4	4	20	12	14	0	00:09:42 Established 1
10.0.12.1	4	10	14	14	0	00:11:13 Established 1

可以看到，R2与R1、R2与R4的BGP邻居关系已成功建立。读者可自行查看R4与R5的BGP邻居关系状态。

在R1上查看BGP路由表。

```
[R1]display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 2
```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 10.0.1.1/32	0.0.0.0	0		0	i
*> 10.0.5.5/32	10.0.12.2			0	20 30 i

可以看到，R1的BGP路由表中拥有关于10.0.1.1/32和10.0.5.5/32的路由信息。

在R5上查看BGP路由表。

```
[R5]display bgp routing-table
```


BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.45.4		0	20	10 i
*>	10.0.5.5/32	0.0.0.0	0	0		i

可以看到，R5的BGP路由表中也拥有关于10.0.1.1/32和10.0.5.5/32的路由信息。

3 . BGP路由黑洞问题

目前，R1和R5 的BGP 路由表中均存在去往对方Loopback 0 接口所在网络的路由。在R1上测试10.0.1.1/32与10.0.5.5/32之间的连通性。

```
<R1>ping -a 10.0.1.1 10.0.5.5
```

```
PING 10.0.5.5: 56 data bytes, press CTRL_C to break
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
--- 10.0.5.5 ping statistics ---
```

```
5 packet(s) transmitted
```

```
0 packet(s) received
```

```
100.00% packet loss
```

结果发现，R1无法与R5进行通信。

在R1上使用tracert命令验证从10.0.1.1/32去往10.0.5.5/32的报文经过的路径。

```
<R1>tracert -a 10.0.1.1 10.0.5.5
```

```
tracert 10.0.5.5(10.0.5.5), max hops: 30 ,packet length:
40,press CTRL_C to break
```

1 10.0.12.2 20 ms 10 ms 10 ms

2 * * *

观察发现，报文只到达了R2，未能到达R3。

在R2上查看IP路由表。

```
[R2]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 16 Routes : 16

Interface	Destination/Mask	Proto	Pre	Cost	Flags	NextHop
GigabitEthernet0/0/0	10.0.1.1/32	EBGP	255	0	D	10.0.12.1
	10.0.2.2/32	Direct	0	0	D	LoopBack0
	10.0.3.3/32	OSPF	10	1	D	10.0.23.3
GigabitEthernet0/0/1	10.0.4.4/32	OSPF	10	2	D	10.0.23.3
	10.0.5.5/32	IBGP	255	0	RD	10.0.4.4
GigabitEthernet0/0/1	10.0.12.0/24	Direct	0	0	D	10.0.12.2
GigabitEthernet0/0/0						

可以看到，R2拥有去往10.0.5.5/32接口所在网络的路由信息，下一跳为10.0.4.4。进行路由递归查找后，可知 R2 去往 10.0.4.4/32 的下一跳为 10.0.23.3，即 R3。这说明R2上存在去往10.0.5.5/32的路由，并且下一跳是可达的。

在R3上查看IP路由表。

```
[R3]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 13		Routes : 13				
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
10.0.2.2/32	OSPF	10	1	D	10.0.23.2	
GigabitEthernet0/0/1						
10.0.3.3/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.4.4/32	OSPF	10	1	D	10.0.34.4	
GigabitEthernet0/0/0						
10.0.23.0/24	Direct	0	0	D	10.0.23.3	
GigabitEthernet0/0/1						
10.0.23.3/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.23.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.34.0/24	Direct	0	0	D	10.0.34.3	
GigabitEthernet0/0/0						

10.0.34.3/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.34.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

观察发现，R3的IP路由表中并没有关于10.0.1.1及10.0.5.5的路由信息。当R3接收到从R2转发过来的源地址为10.0.1.1、目的地址为10.0.5.5的报文时，会在自己的IP路由表中查找关于10.0.5.5的路由信息。由于现在R3在自己的IP路由表中查不到关于10.0.5.5的路由信息，所以R3会直接将报文丢弃，然后向源地址10.0.1.1发送ICMP Destination Unreachable消息。但是，此时R3的IP路由表中也没有关于10.0.1.1的路由信息，导致ICMP Destination Unreachable消息也无法被发送。这样一来，从10.0.1.1去往10.0.5.5的报文就在R3上无声无息地消失了，这就是所谓的BGP路由黑洞问题。

4．采用IBGP全互联方式解决BGP路由黑洞问题

为了使R3也能拥有去往10.0.1.1/32和10.0.5.5/32的路由信息，以便解决BGP路由黑洞问题，可以让R3也运行BGP协议，并让R2、R3、R4建立全互联的IBGP邻居关系。

```
[R3]bgp 20
```

```
[R3-bgp]router-id 10.0.3.3
```

```

[R3-bgp]peer 10.0.2.2 as-number 20
[R3-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R3-bgp]peer 10.0.4.4 as-number 20
[R3-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R2]bgp 20
[R2-bgp]peer 10.0.3.3 as-number 20
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R2-bgp]peer 10.0.3.3 next-hop-local
[R4]bgp 20
[R4-bgp]peer 10.0.3.3 as-number 20
[R4-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R4-bgp]peer 10.0.3.3 next-hop-local
配置完成后，在R3上查看BGP邻居信息。

```

```

[R3]display bgp peer
BGP local router ID : 10.0.3.3
Local AS number : 20
Total number of peers : 2      Peers in established state : 2

```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.2.2	4	20	4	0	00:01:23	Established
10.0.4.4	4	20	3	0	00:00:37	Established

可以看到，R3与R2、R3与R4已经成功建立起了IBGP邻居关系。
在R3上查看BGP路由表。

```

[R3]display bgp routing-table
BGP Local router ID is 10.0.3.3

```

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.2.2	0	100	0	10 i
*>i	10.0.5.5/32	10.0.4.4	0	100	0	30 i

观察发现，R3现在已经学习到了关于10.0.1.1/32和10.0.5.5/32的路由信息。

在R1上测试10.0.1.1与10.0.5.5之间的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.5.5
```

```
PING 10.0.5.5: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.5.5: bytes=56 Sequence=1 ttl=252 time=50 ms
```

```
--- 10.0.5.5 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 50/50/50 ms
```

可以看到，10.0.1.1/32与10.0.5.5/32可以互通了，BGP路由黑洞问题得到了解决。

5. 采用GRE隧道方式解决BGP路由黑洞问题

解决BGP路由黑洞问题还可以通过使用GRE隧道技术来实现。对于本实验网络中的BGP路由黑洞问题，可以在R2与R4之间建立GRE隧道，使得在逻辑上R2与R4直接相连。当报文到达未配置BGP协议的R3时，报文的源IP地址和目的IP地址已经是由GRE隧道重新封装后的可达地址，这样就可以避免BGP路由黑洞现象的出现。

首先，在R3的BGP视图下关闭BGP进程。

```
[R3]bgp 20
```

```
[R3-bgp]shutdown
```

Warning: All BGP peer sessions will be interrupted. Continue? [Y/N]:y

然后在R2 上使用interface Tunnel 0/0/0 命令创建Tunnel 接口。

```
[R2]interface Tunnel 0/0/0
```

使用ip address 10.0.100.2 24 命令为Tunnel 接口配置IP 地址。

```
[R2-Tunnel0/0/0]ip address 10.0.100.2 24
```

使用tunnel-protocol gre 命令配置Tunnel 接口的封装协议为GRE 协议。

```
[R2-Tunnel0/0/0]tunnel-protocol gre
```

使用source 10.0.23.2 命令定义隧道的源端，即GRE 协议封装的新的源IP 地址。

```
[R2-Tunnel0/0/0]source 10.0.23.2
```

使用destination 10.0.34.4命令定义隧道的目的端，即GRE协议封装的新的目的IP地址。

```
[R2-Tunnel0/0/0]destination 10.0.34.4
```

在完成隧道的配置后，在R2上创建静态路由，并修改静态路由的协议优先级的值为1。

```
[R2]ip route-static 10.0.4.4 32 10.0.100.4 preference 1
```

创建静态路由并修改协议优先级的值为1（注意，协议优先级的值越小，则优先级越高），是为了保证去往10.0.4.4/32的路由的下一跳为10.0.100.4，也就是R4的Tunnel 0/0/0接口的地址，从而使得所有去往10.0.4.4/32的报文都会被GRE协议封装后进行传输。

在R4上完成同样的配置。

```
[R4]interface Tunnel 0/0/0
```

```
[R4-Tunnel0/0/0]ip address 10.0.100.4 24
```

```
[R4-Tunnel0/0/0]tunnel-protocol gre
```

```
[R4-Tunnel0/0/0]source 10.0.34.4
```

```
[R4-Tunnel0/0/0]destination 10.0.23.2
```

```
[R4]ip route-static 10.0.2.2 32 10.0.100.2 preference 1
```

在R3的GE 0/0/0 接口上查看报文（见图3-13），并在R1 上测试10.0.1.1和10.0.5.5之间的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.5.5
```

```
PING 10.0.5.5: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.5.5: bytes=56 Sequence=1 ttl=252 time=90 ms
```

```
--- 10.0.5.5 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 90/90/90 ms
```

figure_0271_0089

图3-13 R3的GE 0/0/0接口的报文情况

可以看到，10.0.1.1/32与10.0.5.5/32实现了互通，即分公司A的内部网络与分公司B的内部网络能够进行正常的通信了，BGP路由黑洞问题得到了解决。从图3-13中可知，原来的IP报文被GRE协议进行了重新封装，添加了一个新的IP报文头，新的源IP地址为10.0.23.2，新的目的IP地址为10.0.34.4。

思考

解决BGP路由黑洞问题时，什么情况下更适合采用IBGP全互联的方法，什么情况下更适合采用GRE隧道方法？

3.13 BGP 联盟

原理概述

BGP路由反射器可以用来减少大型AS中IBGP邻居关系的数量和简化IBGP邻居关系的管理和维护，BGP联盟（Confederation）也可以用来实现类似的目的。

一个BGP联盟是一个具有内部层次结构的AS。一个BGP联盟由若干个子AS（Subautonomous System）组成，子AS也称为成员AS（Member Autonomous System）。对于一个BGP联盟，其成员AS内部的路由器之间需要建立全互联的IBGP邻居关系或使用BGP路由反射器，而成员AS之间需建立EBGP邻居关系。从联盟外的EBGP对等体来看，整个联盟无异于一个普通的AS，联盟内部的结构对于联盟外的EBGP对等体来说是完全透明的。

每一个联盟都有一个联盟号（Confederation ID），它其实就是一个普通的AS编号。联盟中的成员AS通常使用BGP协议预留的私有AS编号，但也可以使用非预留的AS编号。联盟内各成员AS可以使用相同的IGP协议，也可使用不同的IGP协议。

实验目的

- 理解BGP联盟的概念和作用
- 掌握配置BGP联盟的基本方法

实验内容

实验拓扑如图3-14所示，实验编址如表3-13所示。本实验模拟了一个企业网络场景，AS 100为分公司网络，AS 200为公司总部网络，所有的路由器都运行BGP协议，R1的Loopback 1接口模拟了分公司的内部网络。现在，要求公司总部网络需要拥有去往分公司的内部网络的BGP路由，但网络管理员发现公司总部网络的路由器数量较多，建立全互联IBGP邻居关系需要进行大量的配置工作，特别是公司总部网络后续扩展以后，配置工作的繁杂程度将会变得无法接受。因此，网络管理员决定使用BGP联盟技术来优化公司总部的网络架构：公司总部

网络被视为一个BGP联盟，AS编号为200，R2属于成员AS 2001，R3 和 R4 属于成员AS 2002，R5 和R6 属于成员AS 2003。

实验拓扑

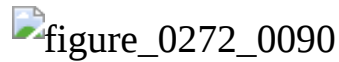
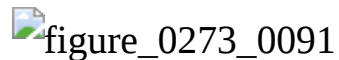


图3-14 BGP联盟

实验编址表

表3-13 实验编址



实验步骤

1．基本配置

根据图3-14和表3-13进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=150 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 150/150/150 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF路由协议

在AS 200 内部的路由器上配置OSPF 协议作为IGP 协议。

```
[R2]ospf 1 router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.25.0 0.0.0.255
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
[R3]ospf 1 router-id 10.0.3.3
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R4]ospf 1 router-id 10.0.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
[R4-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R5]ospf 1 router-id 10.0.5.5
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]network 10.0.5.5 0.0.0.0
[R5-ospf-1-area-0.0.0.0]network 10.0.25.0 0.0.0.255
[R5-ospf-1-area-0.0.0.0]network 10.0.56.0 0.0.0.255
[R6]ospf 1 router-id 10.0.6.6
[R6-ospf-1]area 0
[R6-ospf-1-area-0.0.0.0]network 10.0.6.6 0.0.0.0
[R6-ospf-1-area-0.0.0.0]network 10.0.56.0 0.0.0.255
```

配置完成后，在R2 上使用display ospf peer 命令查看OSPF 邻居信息。

```
[R2]display ospf peer
                OSPF Process 1 with Router ID 10.0.2.2
                  Neighbors
Area 0.0.0.0 interface 10.0.23.2(GigabitEthernet0/0/1)'s neighbors
  Router ID: 10.0.3.3    Address: 10.0.23.3
    State: Full Mode:Nbr is Master Priority: 1
```

DR: 10.0.23.3 BDR: 10.0.23.2 MTU: 0

Dead timer due in 29 sec

Retrans timer interval: 5

Neighbor is up for 00:04:55

Authentication Sequence: [0]

Neighbors

Area 0.0.0.0 interface 10.0.25.2(GigabitEthernet0/0/2)'s neighbors

Router ID: 10.0.5.5 Address: 10.0.25.5

State: Full Mode:Nbr is Master Priority: 1

DR: 10.0.25.5 BDR: 10.0.25.2 MTU: 0

Dead timer due in 36 sec

Retrans timer interval: 5

Neighbor is up for 00:03:28

Authentication Sequence: [0]

可以看到，R2与R3、R2与R5都已经成功建立起了OSPF邻接关系。读者可自行在其他路由器上查看OSPF邻居信息。

3．配置BGP路由协议

建立R1与R2的EBGP对等体关系；在AS 200内部建立全互联的IBGP对等体关系。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 200
```

```
[R1-bgp]network 10.0.100.1 255.255.255.0
```

```
[R2]bgp 200
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.3.3 as-number 200
```

```
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R2-bgp]peer 10.0.3.3 next-hop-local
[R2-bgp]peer 10.0.4.4 as-number 200
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R2-bgp]peer 10.0.4.4 next-hop-local
[R2-bgp]peer 10.0.5.5 as-number 200
[R2-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R2-bgp]peer 10.0.5.5 next-hop-local
[R2-bgp]peer 10.0.6.6 as-number 200
[R2-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R2-bgp]peer 10.0.6.6 next-hop-local
[R3]bgp 200
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.2.2 as-number 200
[R3-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R3-bgp]peer 10.0.4.4 as-number 200
[R3-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R3-bgp]peer 10.0.5.5 as-number 200
[R3-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R3-bgp]peer 10.0.6.6 as-number 200
[R3-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R4]bgp 200
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.2.2 as-number 200
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R4-bgp]peer 10.0.3.3 as-number 200
[R4-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```

```
[R4-bgp]peer 10.0.5.5 as-number 200
[R4-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R4-bgp]peer 10.0.6.6 as-number 200
[R4-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R5]bgp 200
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.2.2 as-number 200
[R5-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R5-bgp]peer 10.0.3.3 as-number 200
[R5-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R5-bgp]peer 10.0.4.4 as-number 200
[R5-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R5-bgp]peer 10.0.6.6 as-number 200
[R5-bgp]peer 10.0.6.6 connect-interface LoopBack 0
[R6]bgp 200
[R6-bgp]router-id 10.0.6.6
[R6-bgp]peer 10.0.2.2 as-number 200
[R6-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R6-bgp]peer 10.0.3.3 as-number 200
[R6-bgp]peer 10.0.3.3 connect-interface LoopBack 0
[R6-bgp]peer 10.0.4.4 as-number 200
[R6-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R6-bgp]peer 10.0.5.5 as-number 200
[R6-bgp]peer 10.0.5.5 connect-interface LoopBack 0
```

配置完成后，在R2上使用display bgp peer 命令查看BGP 邻居信息。

```
[R2]display bgp peer
```

BGP local router ID : 10.0.2.2

Local AS number : 200

Total number of peers : 5 Peers in established state : 5

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.3.3	4	200	11	14	0	00:09:06 Established 0
10.0.4.4	4	200	9	12	0	00:07:41 Established 0
10.0.5.5	4	200	8	11	0	00:06:26 Established 0
10.0.6.6	4	200	6	8	0	00:04:58 Established 0
10.0.12.1	4	100	14	13	0	00:11:47 Established 1

可以看到，R2的BGP邻居关系状态均为Established。读者可自行在其他路由器上查看BGP 邻居关系。需要说明的是，现在AS 200 内部的IBGP 邻居关系数量众多，每台路由器都需要维护4个IBGP邻居关系。

在R2上查看BGP路由表。

```
<R2>display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 10.0.100.0/24	10.0.12.1	0		0	100i

可以看到，R2的BGP路由表中已经拥有了关于10.0.100.0/24的路由信息。

在R6上查看BGP路由表。

```
<R6>display bgp routing-table
```

```
BGP Local router ID is 10.0.6.6
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 1
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.0/24	10.0.2.2	0	100	0	100i

可以看到，R6的BGP路由表中也已经拥有了关于10.0.100.0/24的路由信息。读者可自行查看AS 200 内部其他路由器上的BGP 路由表，并会发现每台路由器都已接收到了关于10.0.100.0/24的路由信息。

4 . 配置BGP联盟

接下来将把AS 200 视为一个BGP 联盟，并根据图3-13 进行相应的配置，从而减少AS 200 内部IBGP 邻居关系的数量。

在R2 上使用命令undo bgp 200 删除当前BGP 进程。

```
[R2]undo bgp 200
```

```
Warning: All BGP configurations will be deleted. Continue? [Y/N]: y
```

使用bgp 2001 命令配置R2 所属的成员AS 编号，并启动BGP 协议进程。

```
[R2]bgp 2001
```

```
[R2-bgp]router-id 10.0.2.2
```

在BGP 视图下使用confederation id 200 命令配置R2 所属的联盟ID。

```
[R2-bgp]confederation id 200
```

使用confederation peer-as 2002 2003命令指明R2的联盟EBGP邻居所属的成员AS编号。该命令只能配置在存在联盟EBGP邻居的BGP路由器上。


```
[R2-bgp]confederation peer-as 2002 2003
```

使用peer 10.0.23.3 as-number 2002和peer 10.0.25.5 as-number 2003命令，与成员AS 2002 中的路由器R3，以及成员AS 2003 中的路由器R5 建立联盟EBGP 邻居关系，且指明在发送路由信息时将把Next Hop属性修改为自己。

```
[R2-bgp]peer 10.0.23.3 as-number 2002
```

```
[R2-bgp]peer 10.0.23.3 next-hop-local
```

```
[R2-bgp]peer 10.0.25.5 as-number 2003
```

```
[R2-bgp]peer 10.0.25.5 next-hop-local
```

使用peer 10.0.12.1 as-number 100 命令与AS 100 中的路由器R1 建立EBGP 邻居关系。

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

在R3、R4、R5、R6上完成类似的配置。

```
[R3]undo bgp 200
```

Warning: All BGP configurations will be deleted. Continue? [Y/N]: y

```
[R3]bgp 2002
```

```
[R3-bgp]router-id 10.0.3.3
```

```
[R3-bgp]confederation id 200
```

```
[R3-bgp]confederation peer-as 2001
```

```
[R3-bgp]peer 10.0.23.2 as-number 2001
```

```
[R3-bgp]peer 10.0.34.4 as-number 2002
```

```
[R4]undo bgp 200
```

Warning: All BGP configurations will be deleted. Continue? [Y/N]: y

```
[R4]bgp 2002
```

```
[R4-bgp]router-id 10.0.4.4
```

```
[R4-bgp]confederation id 200
```

```
[R4-bgp]peer 10.0.34.3 as-number 2002
```

```
[R5]undo bgp 200
```

Warning: All BGP configurations will be deleted. Continue? [Y/N]: y

```
[R5]bgp 2003
```

```
[R5-bgp]router-id 10.0.5.5
```

```
[R5-bgp]confederation id 200
```

```
[R5-bgp]confederation peer-as 2001
```

```
[R5-bgp]peer 10.0.25.2 as-number 2001
```

```
[R5-bgp]peer 10.0.56.6 as-number 2003
```

```
[R6]undo bgp 200
```

Warning: All BGP configurations will be deleted. Continue? [Y/N]: y

```
[R6]bgp 2003
```

```
[R6-bgp]router-id 10.0.6.6
```

```
[R6-bgp]confederation id 200
```

```
[R6-bgp]peer 10.0.56.5 as-number 2003
```

至此，关于联盟的配置工作就结束了。在R2上查看BGP邻居信息。

```
[R2]display bgp peer
```

BGP local router ID : 10.0.2.2

Local AS number : 2001

Total number of peers : 3 Peers in established state : 3

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.12.1	4	100	3	2	0	00:00:19
Established	1					
10.0.23.3	4	2002	16	18	0	00:14:25
Established	0					

```
10.0.25.5    4    2003        13        15        0        00:11:55
Established    0
```

可以看到，R2与EBGP对等体R1，与联盟EBGP对等体R3和R5的邻居关系状态都为Established。

在R3上查看BGP邻居信息。

```
[R3]display bgp peer
```

```
BGP local router ID : 10.0.3.3
```

```
Local AS number : 2002
```

```
Total number of peers : 2    Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.23.2	4	2001	21	20	0	00:18:13
Established	1					
10.0.34.4	4	2002	10	11	0	00:08:59
Established	0					

可以看到，R3与R2的联盟EBGP邻居关系，R3与R4的IBGP邻居关系状态都为Established。读者可自行在其他路由器上查看BGP邻居关系。

在R2上查看BGP路由表。

```
[R2]display bgp routing-table
```

```
BGP Local router ID is 10.0.2.2
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 1
```

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 10.0.100.0/24	10.0.12.1	0		0	100i

可以看到，R2的BGP路由表中已经拥有了关于10.0.100.1/24的路由信息。

在R6上查看BGP路由表。

```
[R6]display bgp routing-table
```

```
BGP Local router ID is 10.0.6.6
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 1
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.100.0/24	10.0.25.2	0	100	0	(2001)

```
100i
```

可以看到，R6也接收到了关于10.0.100.1/24的路由信息。读者可自行查看联盟AS 200 内部其他路由器上的 BGP 路由表，并会发现每台路由器都已接收到了关于10.0.100.0/24的路由信息。

若公司总部网络规模需要扩大，则一般只需要在相应的成员 AS 中添加路由器并进行相关的配置即可，配置工作量远远小于不使用BGP联盟时的情形。

思考

联盟的内部可以使用BGP反射器吗？

3.14 BGP 路由过滤

原理概述

BGP 路由可以携各种各样的路由属性，例如 Preferred Value 属性、Local Preference 属性、AS_Path 属性、Origin 属性、MED 属性、

Next Hop 属性、团体属性等。路由属性的丰富性可以为实现路由过滤、路由引入等路由策略和控制提供非常有利的条件。

实验目的

- 掌握利用BGP 路由属性AS_Path 进行路由过滤的方法
- 掌握利用BGP 路由属性Community进行路由过滤的方法
- 掌握利用BGP 路由属性Next Hop 进行路由过滤的方法

实验内容

实验拓扑如图3-15 所示，实验编址如表 3-14 所示。本实验网络中，AS 100 模拟了企业总部，AS 200、AS 300、AS 400、AS 500 分别模拟了企业的分支机构1、分支机构2、分支机构3、分支机构4。网络需求是：各个分支机构都需要与企业总部进行通信，同时要求分支机构1（AS 200）不能接收其他分支机构的路由；分支机构2（AS 300）不能将自己的路由信息通告给其他分支机构；分支机构4（AS 500）不能接收分支机构3（AS 400）的路由。这些需求都需要针对BGP 路由的某些属性进行路由过滤来实现。

实验拓扑

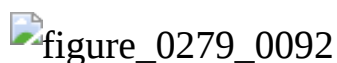
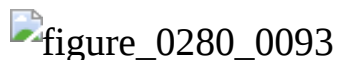


图3-15 BGP路由过滤

实验编址表

表3-14 实验编址



实验步骤

1．基本配置

根据图3-15和表3-14进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=30 ms
--- 10.0.12.2 ping statistics ---
    1 packet(s) transmitted
    1 packet(s) received
    0.00% packet loss
    round-trip min/avg/max = 30/30/30 ms
```

其余直连网段的连通性测试过程在此省略。

2 . 配置BGP路由协议

配置BGP路由协议，每台路由器均使用直连物理接口建立BGP邻居关系，并通告自己的Loopback接口到BGP进程中。

```
[R1]bgp 100
[R1-bgp]router-id 10.0.1.1
[R1-bgp]peer 10.0.12.2 as-number 100
[R1-bgp]peer 10.0.12.2 next-hop-local
[R1-bgp]peer 10.0.13.3 as-number 200
[R1-bgp]peer 10.0.14.4 as-number 300
[R1-bgp]network 10.0.1.1 255.255.255.255
[R2]bgp 100
[R2-bgp]router-id 10.0.2.2
[R2-bgp]peer 10.0.12.1 as-number 100
[R2-bgp]peer 10.0.12.1 next-hop-local
[R2-bgp]peer 10.0.25.5 as-number 400
[R2-bgp]peer 10.0.26.6 as-number 500
[R2-bgp]network 10.0.2.2 255.255.255.255
[R3]bgp 200
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.13.1 as-number 100
```

```
[R3-bgp]network 10.0.3.3 255.255.255.255
[R3-bgp]network 192.168.1.0 255.255.255.0
[R4]bgp 300
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.14.1 as-number 100
[R4-bgp]network 10.0.4.4 255.255.255.255
[R4-bgp]network 192.168.2.0 255.255.255.0
[R5]bgp 400
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.25.2 as-number 100
[R5-bgp]network 10.0.5.5 255.255.255.255
[R5-bgp]network 192.168.3.0 255.255.255.0
[R6-bgp]bgp 500
[R6-bgp]router-id 10.0.6.6
[R6-bgp]peer 10.0.26.2 as-number 100
[R6-bgp]network 10.0.6.6 255.255.255.255
[R6-bgp]network 192.168.4.0 255.255.255.0
```

配置完成后，在R1上使用ping命令检测R1的Loopback 0接口与R2的Loopback 0接口之间的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.2.2
PING 10.0.2.2: 56 data bytes, press CTRL_C to break
  Reply from 10.0.2.2: bytes=56 Sequence=1 ttl=255 time=30 ms
--- 10.0.2.2 ping statistics ---
  1 packet(s) transmitted
  1 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 30/30/30 ms
```

读者可自行检查各路由器Loopback接口之间的联通性，并会发现整个网络已实现了互相连通。

3．利用AS_Path属性进行路由过滤

目前的情况是，每台路由器都接收到了其他路由器的Loopback接口的路由信息，然而公司要求分支机构1（AS 200）是不能接收其他分支机构的路由的，但允许与企业总部（AS 100）进行通信。为了实现这一需求，可以利用AS_Path 属性来进行路由过滤，即只允许 AS_Path 列表中只存在 AS 100 的路由才能被 R3 接收。为此，可以使用as-path-filter结合正则表达式来对BGP路由的AS_Path属性进行匹配，实现路由的过滤。

```
[R3]ip as-path-filter 1 permit 100$
```

```
[R3]bgp 200
```

```
[R3-bgp]peer 10.0.13.1 as-path-filter 1 import
```

配置完成后，在R3上查看BGP路由表。

```
[R3]display bgp routing-table
```

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.13.1	0		0	100i
*>	10.0.2.2/32	10.0.13.1			0	100i
*>	10.0.3.3/32	0.0.0.0	0		0	i
*>	192.168.1.0	0.0.0.0	0		0	i

可以看到，AS 200 中的R3 上已经没有了涉及到分支机构2（AS 300）、分支机构3（AS 400）和分支机构4（AS 500）的路由了。

4 . 利用Community属性进行路由过滤

公司还要求分支机构2 (AS 300) 不能将自己的路由信息通告给其他分支机构 , 但需要将自己的路由信息通告给企业总部 (AS 100) 。为此 , 可利用团体属性中的No-Export来方便而有效地实现这一需求。

```
[R4]route-policy 1 permit node 10
```

```
[R4-route-policy]apply community no-export
```

```
[R4-route-policy]bgp 300
```

```
[R4-bgp]peer 10.0.14.1 route-policy 1 export
```

```
[R4-bgp]peer 10.0.14.1 advertise-community
```

```
[R1]bgp 100
```

```
[R1-bgp]peer 10.0.12.2 advertise-community
```

配置完成后 , 在R1和R2上查看BGP路由表。

```
[R1]display bgp routing-table
```

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 10

	Network	NextHo	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	0.0.0.0	0		0	i
*>i	10.0.2.2/32	10.0.12.2	0	100	0	i
*>	10.0.3.3/32	10.0.13.3	0		0	200i
*>	10.0.4.4/32	10.0.14.4	0		0	300i
*>i	10.0.5.5/32	10.0.12.2	0	100	0	400i
*>i	10.0.6.6/32	10.0.12.2	0	100	0	500i
*>	192.168.1.0	10.0.13.3	0		0	200i
*>	192.168.2.0	10.0.14.4	0		0	300i

*>i 192.168.3.0 10.0.12.2 0 100 0 400i

*>i 192.168.4.0 10.0.12.2 0 100 0 500i

[R2]display bgp routing-table

BGP Local router ID is 10.0.2.2

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 10

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.12.1	0	100	0	i
*>	10.0.2.2/32	0.0.0.0	0		0	i
*>i	10.0.3.3/32	10.0.12.1	0	100	0	200?
*>i	10.0.4.4/32	10.0.12.1	0	100	0	300i
*>	10.0.5.5/32	10.0.25.5	0		0	400i
*>	10.0.6.6/32	10.0.26.6	0		0	500i
*>i	192.168.1.0	10.0.12.1	0	100	0	200?
*>i	192.168.2.0	10.0.12.1	0	100	0	300i
*>	192.168.3.0	10.0.25.5	0		0	400i
*>	192.168.4.0	10.0.26.6	0		0	500i

在R5和R6上查看BGP路由表。

[R5]display bgp routing-table

BGP Local router ID is 10.0.5.5

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 8

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn

```

*> 10.0.1.1/32    10.0.25.2          0      100i
*> 10.0.2.2/32    10.0.25.2  0          0      100i
*> 10.0.3.3/32    10.0.25.2          0      100 200i
*> 10.0.5.5/32    0.0.0.0    0          0          i
*> 10.0.6.6/32    10.0.25.2          0      100 500i
*> 192.168.1.0    10.0.25.2          0      100 200i
*> 192.168.3.0    0.0.0.0    0          0          i
*> 192.168.4.0    10.0.25.2          0      100 500i

```

[R6]display bgp routing-table

BGP Local router ID is 10.0.6.6

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 8

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 10.0.1.1/32	10.0.26.2			0	100i
*> 10.0.2.2/32	10.0.26.2	0		0	100i
*> 10.0.3.3/32	10.0.26.2			0	100 200i
*> 10.0.5.5/32	10.0.26.2			0	100 400i
*> 10.0.6.6/32	0.0.0.0	0		0	i
*> 192.168.1.0	10.0.26.2			0	100 200i
*> 192.168.3.0	10.0.26.2			0	100 400i
*> 192.168.4.0	0.0.0.0	0		0	i

可以看到，除了企业总部路由器R1和R2外，其他分支机构的路由器都未接收到涉及分支机构2的路由信息。

5．利用Next Hop 属性进行路由过滤

公司还要求分支机构4 (AS 500) 不能接收分支机构3 (AS 400) 的路由 , 为此 , 可以利用Next Hop 属性进行路由过滤来实现这一需求。

```
[R2]ip ip-prefix 1 permit 10.0.25.5 32
[R2]route-policy 1 deny node 10
[R2-route-policy]if-match ip next-hop ip-prefix 1
[R2-route-policy]route-policy 1 permit node 20
[R2-route-policy]bgp 100
[R2-bgp]peer 10.0.26.6 route-policy 1 export
```

配置完成后 , 查看R6的BGP路由表。

```
[R6]display bgp routing-table
BGP Local router ID is 10.0.6.6
Status codes: *- valid, >- best, d - damped,
               h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete
Total Number of Routes: 6
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.26.2			0	100i
*>	10.0.2.2/32	10.0.26.2	0		0	100i
*>	10.0.3.3/32	10.0.26.2			0	100 200i
*>	10.0.6.6/32	0.0.0.0	0		0	i
*>	192.168.1.0	10.0.26.2			0	100 200i
*>	192.168.4.0	0.0.0.0	0		0	i

可以看到 , AS 500 中的R6 已经成功地拒绝了涉及AS 400 的路由 , 但并没有影响到其他路由的接收。至此 , 网络需求得到了完全的满足。

思考

相比于其他路由协议，基于BGP协议的路由过滤方法显得非常丰富而灵活，其根本原因是什么？

3.15 BGP 路由引入

原理概述

在多协议混合的网络环境中，不同的路由协议使用的协议报文各不相同，就好比说着不同的语言。默认情况下，不同的路由协议相互间独立工作，互不沟通，互不干扰。也就是说，在默认情况下，一种路由协议无法从别的路由协议那里获取到任何路由信息。如果一种路由协议需要从别的路由协议那里获取路由信息，则可以使用路由引入的技术。例如，可以将RIP、OSPF、IS-IS等IGP协议的动态路由，以及直连路由和静态路由引入到BGP协议的进程中，同时，在引入的过程还可以根据需要实施相应的路由策略和控制。

实验目的

- 理解BGP 路由引入的概念
- 掌握BGP 路由引入的配置方法

实验内容

实验拓扑如图3-16所示，实验编址如表3-15所示。本实验模拟了一个企业网络场景，公司原来有3 台BGP 路由器R2、R3、R4，均属于AS 100，且都采用了直连物理接口建立全互联的IBGP 邻居关系，并通告了各自的用来模拟公司内部网络的Loopback 0接口。后来公司有了两个合作伙伴A和B，合作伙伴A的路由器R1运行的是RIP协议，合作伙伴B的路由器R5 运行的是OSPF 协议，R1 和R5 的Loopback 0 接口模拟了各自的内部网络。网络需求是：通过将RIP路由和OSPF路由引入进

BGP进程，实现全网互联互通；合作伙伴使用缺省路由来访问公司的内部网络和其他合作伙伴的内部网络。

实验拓扑

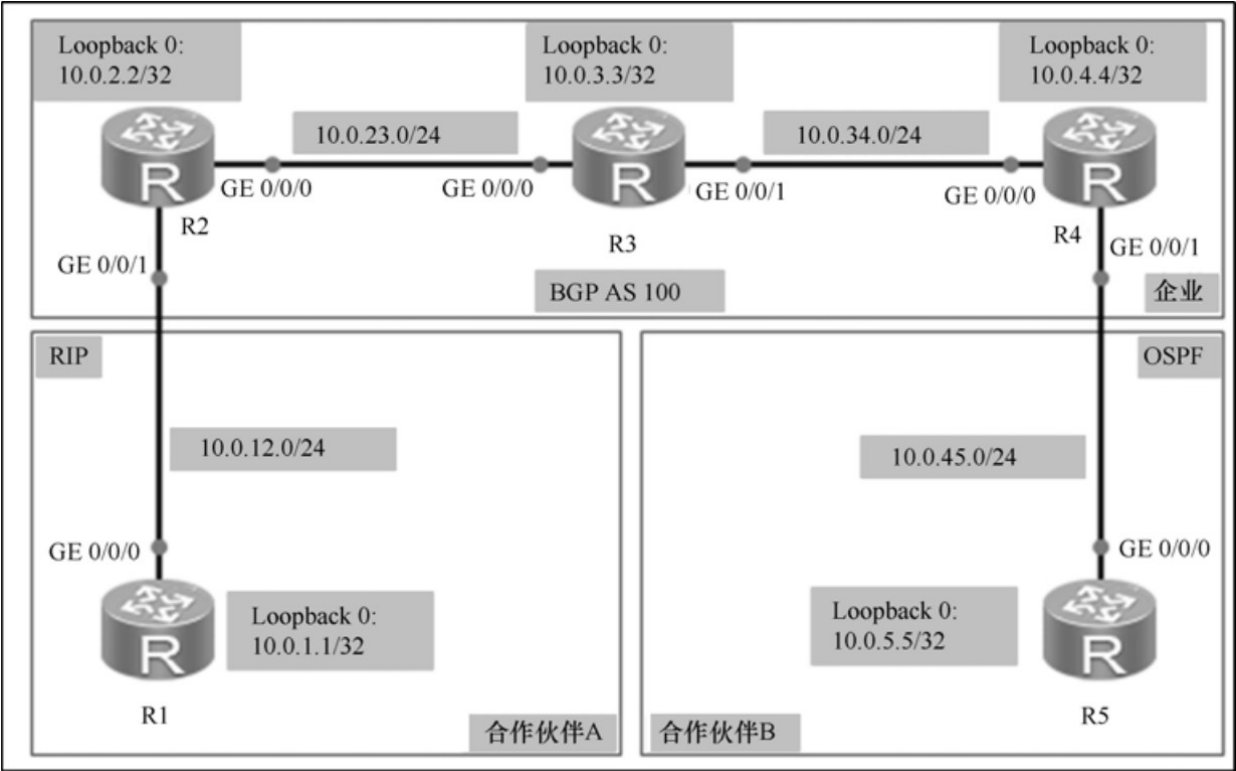


图3-16 BGP路由引入

实验编址表

表3-15 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.23.2	255.255.255.0	N/A
	GE 0/0/1	10.0.12.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/0	10.0.23.3	255.255.255.0	N/A
	GE 0/0/1	10.0.34.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2220)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.45.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
R5(AR2220)	GE 0/0/0	10.0.45.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A

实验步骤

1. 基本配置

根据图3-16和表3-15进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=100 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 100/100/100 ms
```

其余直连网段的连通性测试过程在此省略。

2. 配置BGP路由协议

由于 R2、R3、R4 都使用直连物理接口来建立 IBGP 邻居关系，因此，为了让 R2与R4能够建立TCP会话，可以配置如下的两条静态路由。

```
[R2]ip route-static 10.0.34.0 255.255.255.0 10.0.23.3
```

```
[R4]ip route-static 10.0.23.0 255.255.255.0 10.0.34.3
```

在R2、R3、R4上配置BGP路由协议。

```
[R2]bgp 100
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.23.3 as-number 100
```

```
[R2-bgp]peer 10.0.23.3 next-hop-local
```

```
[R2-bgp]peer 10.0.34.4 as-number 100
```

```
[R2-bgp]peer 10.0.34.4 next-hop-local
```

```
[R2-bgp]network 10.0.2.2 255.255.255.255
```

```
[R3]bgp 100
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.23.2 as-number 100
[R3-bgp]peer 10.0.34.4 as-number 100
[R3-bgp]network 10.0.3.3 255.255.255.255
[R4]bgp 100
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.23.2 as-number 100
[R4-bgp]peer 10.0.23.2 next-hop-local
[R4-bgp]peer 10.0.34.3 as-number 100
[R4-bgp]peer 10.0.34.3 next-hop-local
[R4-bgp]network 10.0.4.4 255.255.255.255
```

3 . 配置RIP和OSPF路由协议

在R1和R2上配置RIP协议。

```
[R1]rip 1
[R1-rip-1]version 2
[R1-rip-1]network 10.0.0.0
[R2]rip 1
[R2-rip-1]version 2
[R2-rip-1]network 10.0.0.0
```

在R4和R5上配置OSPF协议。

```
[R4]ospf 1 router-id 10.0.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.45.0 0.0.0.255
[R5]ospf 1 router-id 10.0.5.5
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]network 10.0.5.5 0.0.0.0
```


可以看到，合作伙伴A的路由器R1上没有去往公司内部和合作伙伴B的路由信息。

查看R3的IP路由表。

```
[R3]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 13 Routes : 13						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
10.0.2.2/32	IBGP	255	0	RD	10.0.23.2	
GigabitEthernet0/0/0						
10.0.3.3/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.4.4/32	IBGP	255	0	RD	10.0.34.4	
GigabitEthernet0/0/1						
10.0.23.0/24	Direct	0	0	D	10.0.23.3	
GigabitEthernet0/0/0						
10.0.23.3/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.23.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.34.0/24	Direct	0	0	D	10.0.34.3	
GigabitEthernet0/0/1						
10.0.34.3/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						

```

10.0.34.255/32      Direct      0    0    D      127.0.0.1
GigabitEthernet0/0/1
127.0.0.0/8        Direct      0    0    D      127.0.0.1    InLoopBack0
127.0.0.1/32       Direct      0    0    D      127.0.0.1    InLoopBack0
127.255.255.255/32 Direct      0    0    D      127.0.0.1
InLoopBack0
255.255.255.255/32 Direct      0    0    D      127.0.0.1
InLoopBack0

```

可以看到，公司内部路由器R3上没有去往合作伙伴网络的路由信息。

查看R5的IP路由表。

[R5]display ip routing-table

Route Flags: R - relay, D - download to fib

```

-----
Routing Tables: Public
Destinations : 8    Routes : 8
Destination/Mask    Proto    Pre    Cost    Flags    NextHop
Interface
10.0.5.5/32        Direct    0    0    D      127.0.0.1    LoopBack0
10.0.45.0/24       Direct      0    0    D      10.0.45.5
GigabitEthernet0/0/0
10.0.45.5/32       Direct      0    0    D      127.0.0.1
GigabitEthernet0/0/0
10.0.45.255/32     Direct      0    0    D      127.0.0.1
GigabitEthernet0/0/0
127.0.0.0/8        Direct    0    0    D      127.0.0.1    InLoopBack0

```

```

127.0.0.1/32    Direct    0  0  D    127.0.0.1    InLoopBack0
127.255.255.255/32    Direct        0  0  D    127.0.0.1
InLoopBack0
255.255.255.255/32    Direct        0  0  D    127.0.0.1
InLoopBack0

```

可以看到，合作伙伴B的路由器R5上没有去往公司内部和合作伙伴A的路由信息。

4．引入RIP路由

在R2上进行配置，将RIP路由引入BGP进程中（通过配置路由策略，只引入R1的Loopback 0 接口所在网段的路由信息），同时在RIP进程中下发缺省路由。

```

[R2]ip ip-prefix 1 permit 10.0.1.1 32
[R2]route-policy 1 permit node 10
[R2-route-policy]if-match ip-prefix 1
[R2-route-policy]bgp 100
[R2-bgp]import-route rip 1 route-policy 1
[R2-bgp]rip 1
[R2-rip-1]default-route originate

```

配置完成后，查看R2的BGP路由表。

```

[R2]display bgp routing-table
BGP Local router ID is 10.0.2.2
Status codes: *- valid, >- best, d - damped,
               h - history, i - internal, s - suppressed, S - Stale
Origin : i - IGP, e - EGP, ? - incomplete

```

Total Number of Routes: 4

```

      Network      NextHop  MED   LocPrf  PrefVal  Path/Ogn
*>  10.0.1.1/32    0.0.0.0    1         0         ?

```

```
*> 10.0.2.2/32    0.0.0.0    0        0        i
*>i 10.0.3.3/32    10.0.23.3  0      100     0        i
*>i 10.0.4.4/32    10.0.34.4  0      100     0        i
```

可以看到，R2的BGP路由表中获得了关于10.0.1.1/32的路由，其下一跳为0.0.0.0，表示是本地生成的；Path/Ogn 处显示的是“？”，说明这是一条引入的路由。默认情况下，路由被引入到BGP进程中时，其MED的值都会被自动设置为1，但可以根据需要进行修改，比如下面将其修改为5。

```
[R2-bgp]import-route rip 1 med 5 route-policy 1
```

配置完成后，查看R2的BGP路由表。

```
[R2]display bgp routing-table
```

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	0.0.0.0	5		0	?
*>	10.0.2.2/32	0.0.0.0	0		0	i
*>i	10.0.3.3/32	10.0.23.3	0	100	0	i
*>i	10.0.4.4/32	10.0.34.4	0	100	0	i

可以看到，关于10.0.1.1/32的路由信息的MED值已被修改为5。

查看R1的IP路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
-----
```

Routing Tables: Public						
Destinations :11 Routes :11						
Destination/Mask	Proto		Pre	Cost	Flags	NextHop
Interface						
0.0.0.0/0	RIP		100	1	D	10.0.12.2
GigabitEthernet0/0/0						
10.0.1.1/32	Direct	0 0	D	127.0.0.1	LoopBack0	
.....						

可以看到，R1上有一条缺省路由，它是由R2通过RIP协议下发的，下一跳为R2（10.0.12.2）。

5. 引入OSPF路由

在R4上进行配置，将OSPF路由引入到BGP进程中（通过配置路由策略，只引入R5的Loopback 0接口所在网段的路由信息），同时在OSPF进程中下发缺省路由。

```
[R4]ip ip-prefix 1 index 10 permit 10.0.5.5 32
```

```
[R4]route-policy 1 permit node 10
```

```
[R4-route-policy]if-match ip-prefix 1
```

```
[R4-route-policy]bgp 100
```

```
[R4-bgp]import-route ospf 1 route-policy 1
```

```
[R4-bgp]ospf 1
```

```
[R4-ospf-1]default-route-advertise always
```

配置完成后，查看R4的BGP路由表。

```
[R4]display bgp routing-table
```

```
BGP Local router ID is 10.0.4.4
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.23.2	5	100	0	?
*>i	10.0.2.2/32	10.0.23.2	0	100	0	i
*>i	10.0.3.3/32	10.0.34.3	0	100	0	i
*>	10.0.4.4/32	0.0.0.0	0		0	i
*>	10.0.5.5/32	0.0.0.0	1		0	?

可以看到，R4 的BGP 路由表中拥有了R5的Loopback 0 接口所在网段的路由信息，其下一跳为0.0.0.0，MED值为1。同时，R4 的BGP路由表中也拥有了R1的Loopback 0接口所在网段的路由信息，其下一跳为R2 (10.0.23.2)。

至此，公司网络中已经拥有了关于两个合作伙伴的路由，而合作伙伴也都有了去往外部网络的缺省路由，所以全网实现了互联互通。

在R1上使用ping命令检测两个合作伙伴之间的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.5.5
```

```
PING 10.0.5.5: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.5.5: bytes=56 Sequence=1 ttl=252 time=40 ms
```

```
--- 10.0.5.5 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 40/40/40 ms
```

可以看到，合作伙伴 A的内部网络是可以与合作伙伴 B的内部网络进行正常通信的。

思考

BGP能否通过路由引入命令引入静态缺省路由进BGP进程？

3.16 BGP 缺省路由

原理概述

和许多其他路由协议一样，BGP 协议也支持缺省路由的使用。在实际的网络场景中，适当而灵活地运用BGP缺省路由，可以大大地简化繁杂的路由问题，有利于网络的优化。

在BGP网络中，一台路由器可以向它的一个BGP对等体发布一条下一跳为自己的缺省路由，也可以使用network命名向整个AS通告一条下一跳为自己的缺省路由，另外，还可以根据需要在BGP路由器上手工配置静态缺省路由。

实验目的

- 理解BGP 缺省路由的使用环境
- 掌握BGP 缺省路由的配置方法

实验内容

实验拓扑如图3-17 所示，实验编址如表 3-16 所示。本实验网络中，R1、R2、R3、R4 属于 AS 100，R5 属于 AS 200。R1 与 R2、R3、R4 采用直连物理接口建立 IBGP 邻居关系，并通告自己的 Loopback 0 接口地址到 BGP 进程中。R5 与 R2、R3、R4采用直连物理接口建立 EBGP 邻居关系，并通告自己的 Loopback 0 接口地址到 BGP 进程中。注意，R5 在通告自己的 Loopback 0 接口时携带了 No-Advertise 团体属性，以此方式来模拟出 R5 能够接收到 R1 的 Loopback 0 的路由但 R1 接收不到 R5 的Loopback 0 的路由的情形，从而导致 R1 和 R5 的 Loopback 0 接口之间无法正常通信。实验的要求是：利用缺省路由的方法来解决 R1 和 R5 的 Loopback 0 接口之间的互通问题。

实验拓扑

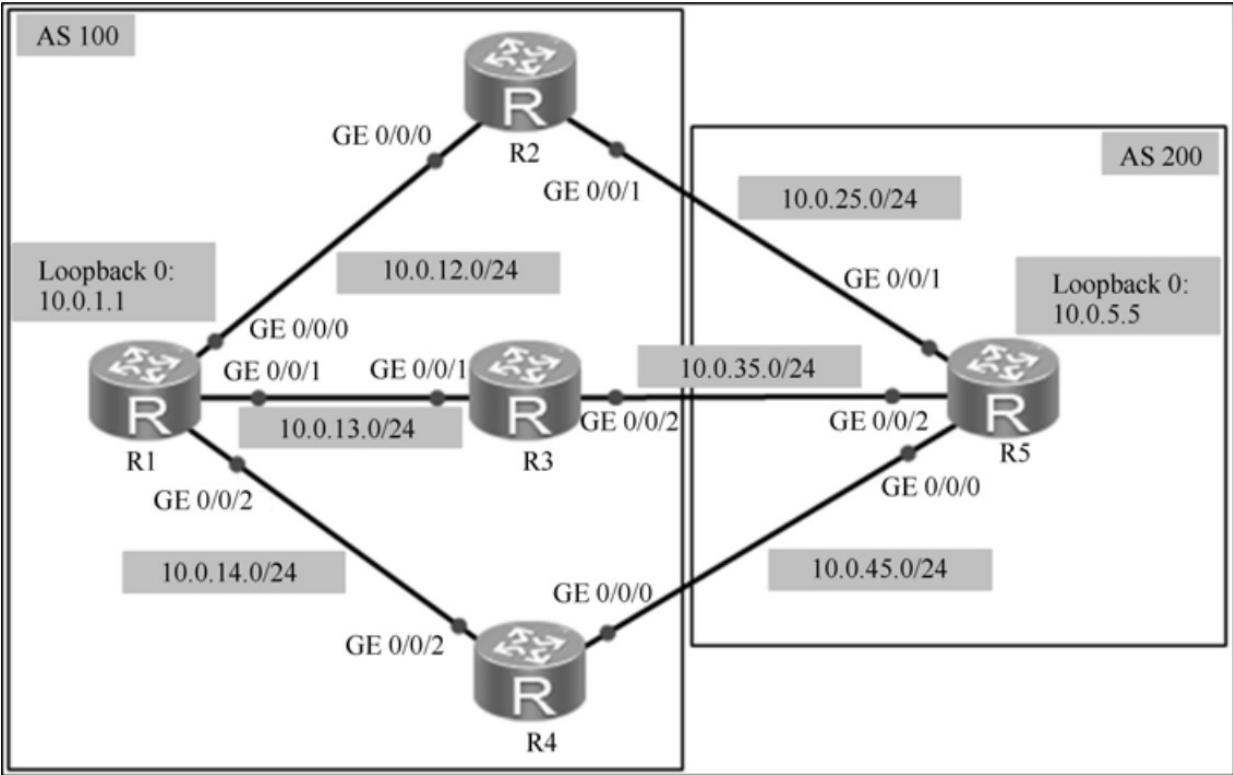


图3-17 BGP缺省路由

实验编址表

表3-16 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR3260)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	GE 0/0/2	10.0.14.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR3260)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.25.2	255.255.255.0	N/A
R3(AR3260)	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.35.3	255.255.255.0	N/A
R4(AR3260)	GE 0/0/0	10.0.45.4	255.255.255.0	N/A
	GE 0/0/2	10.0.14.4	255.255.255.0	N/A
R5(AR3260)	GE 0/0/0	10.0.45.5	255.255.255.0	N/A
	GE 0/0/1	10.0.25.5	255.255.255.0	N/A
	GE 0/0/2	10.0.35.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-17和表3-16进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=160 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 160/160/160 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置BGP路由协议

配置BGP路由协议，R1与R2、R3、R4采用直连物理接口建立IBGP邻居关系，并通告自己的Loopback 0 接口地址到BGP 进程中。R5与R2、R3、R4 采用直连物理接口建立EBGP 邻居关系，并通告自己的Loopback 0 接口地址到BGP 进程中。R5 在通告自己的Loopback 0 接口时携带了No-Advertise 团体属性。

```
[R1]bgp 100
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.12.2 as-number 100
```

```
[R1-bgp]peer 10.0.13.3 as-number 100
```

```
[R1-bgp]peer 10.0.14.4 as-number 100
```

```
[R1-bgp]network 10.0.1.1 255.255.255.255
```

```
[R2]bgp 100
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.25.5 as-number 200
```

```
[R3]bgp 100
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.13.1 as-number 100
[R3-bgp]peer 10.0.35.5 as-number 200
[R4]bgp 100
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.14.1 as-number 100
[R4-bgp]peer 10.0.45.5 as-number 200
[R5]route-policy 1 permit node 10
[R5-route-policy]apply community no-advertise
[R5]bgp 200
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.25.2 as-number 100
[R5-bgp]peer 10.0.25.2 route-policy 1 export
[R5-bgp]peer 10.0.25.2 advertise-community
[R5-bgp]peer 10.0.35.3 as-number 100
[R5-bgp]peer 10.0.35.3 route-policy 1 export
[R5-bgp]peer 10.0.35.3 advertise-community
[R5-bgp]peer 10.0.45.4 as-number 100
[R5-bgp]peer 10.0.45.4 route-policy 1 export
[R5-bgp]peer 10.0.45.4 advertise-community
[R5-bgp]network 10.0.5.5 255.255.255.255
```

显然，通过上述配置，R5能够从R2、R3、R4那里接收到3条关于R1的Loopback 0 接口的路由信息，但R1接收不到任何关于R5的Loopback 0 接口的路由信息。因此，R1 和R5 的Loopback 0 接口之间无法进行正常的通信。

3．向BGP对等体发布下一跳为本地路由器的缺省路由

在R4上向BGP对等体R1发布一条下一跳为R4自己的缺省路由。

```
[R4]bgp 100
```

```
[R4-bgp]peer 10.0.14.1 default-route-advertise
```

配置完成后，查看R1的BGP路由表。

```
<R1>display bgp routing-table
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 2
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	0.0.0.0	10.0.14.4	0	100	0	i
*>	10.0.1.1/32	0.0.0.0	0		0	i

可以看到，R1的BGP路由表中已经拥有了一条缺省路由，下一跳为R4（10.0.14.4）。这说明，无论R4的本地路由表中有无缺省路由，都可以向对等体发送一条下一跳为自己的缺省路由。

在R1上测试R1和R5的Loopback 0接口之间的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.5.5
```

```
PING 10.0.5.5: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.5.5: bytes=56 Sequence=1 ttl=254 time=50 ms
```

```
--- 10.0.5.5 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 50/50/50 ms
```

可以看到，R1和R5的Loopback 0接口之间通信正常。

4．使用network命令在AS内通告缺省路由

在R3的BGP进程中直接使用network命令通告缺省路由。

```
[R3]bgp 100
```

```
[R3-bgp]network 0.0.0.0 0.0.0.0
```

Info: The network does not exist.

提示信息“The network does not exist”说明，使用network 命令通告路由时，被通告的路由必须在本地路由表中是已经存在的。

因此，在R3 上手动配置一条缺省路由，下一跳指向NULL 0。

```
[R3]ip route-static 0.0.0.0 0.0.0.0 NULL 0
```

再次在R3的BGP进程中通告缺省路由。

```
[R3]bgp 100
```

```
[R3-bgp]network 0.0.0.0 0.0.0.0
```

配置完成后，查看R1的BGP路由表。

```
<R1>display bgp routing-table
```

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	0.0.0.0	10.0.13.3	0	100	0	i
* i		10.0.14.4	0	100	0	i
*>	10.0.1.1/32	0.0.0.0	0		0	i

可以看到，R1的BGP路由表中现在有了两条缺省路由，下一跳分别是R3（10.0.13.3）和R4（10.0.14.4），其中第一条是R3使用network命令通告的缺省路由，第二条是之前R4专门向R1发布的缺省路由。这两条缺省路由的下一跳不同，但路由信息首选值Preferred Value、本地优先级Local Preference、路由生成方式、AS_Path属性、Origin属

性、MED属性、BGP对等体类型等都是相同的，所以最终R1选择了Router-ID较小的路由器R3发布的缺省路由作为最佳缺省路由。

查看R5的BGP路由表。

```
<R5>display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 5

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	0.0.0.0	10.0.35.3	0	0	100i	
*>	10.0.1.1/32	10.0.25.2		0	100i	
*		10.0.35.3		0	100i	
*		10.0.45.4		0	100i	
*>	10.0.5.5/32	0.0.0.0	0	0		i

观察发现，R5也接收到了R3通告的那条缺省路由，但目前这条缺省路由在R5上不起任何作用，所以可以在R3上配置路由策略，使这条缺省路由不会传递给R5。

```
[R3]ip ip-prefix 1 permit 0.0.0.0 0
```

```
[R3]route-policy 1 deny node 10
```

```
[R3-route-policy]if-match ip-prefix 1
```

```
[R3-route-policy]route-policy 1 permit node 20
```

```
[R3-route-policy]bgp 100
```

```
[R3-bgp]peer 10.0.35.5 route-policy 1 export
```

配置完成后，再次查看R5的BGP路由表。

```
<R5>display bgp routing-table
```

BGP Local router ID is 10.0.5.5

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.25.2		0	100i	
*		10.0.35.3		0	100i	
*		10.0.45.4		0	100i	
*>	10.0.5.5/32	0.0.0.0	0	0		i

可以看到，R5上的那条缺省路由已经消失了。

在R1上使用tracert命令验证从10.0.1.1/32去往10.0.5.5/32的报文的转发路径。

```
<R1>tracert -a 10.0.1.1 10.0.5.5
```

```
traceroute to 10.0.5.5(10.0.5.5), max hops: 30 ,packet length: 40,pres  
CTRL_C to break
```

```
1 10.0.13.3 20 ms 1 ms 1 ms
```

```
2 10.0.35.5 30 ms 20 ms 20 ms
```

可以看到，报文是经过R3转发的，说明报文在R1上是选用了R3通告的那条缺省路由发出的。

5．手动添加一条静态缺省路由

查看R1的IP路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 15 Routes : 15

```

Destination/Mask  Proto Pre  Cost  Flags NextHop      Interface
0.0.0.0/0                IBGP  255    0    RD    10.0.13.3
GigabitEthernet0/0/1
10.0.1.1/32  Direct 0   0   D   127.0.0.1  LoopBack0
.....

```

可以看到，R1 的 IP 路由表中缺省路由的下一跳为 R3 (10.0.13.3)，该缺省路由是通过IBGP协议从R3那里学习来的。

下面，在 R1 上手动配置一条静态缺省路由，下一跳指向 R2 (10.0.12.2)。

```
[R1]ip route-static 0.0.0.0 0.0.0.0 10.0.12.2
```

配置完成后，查看R1的IP路由表。

```
<R1>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```

-----
Routing Tables: Public
Destinations : 15  Routes : 15

Destination/Mask    Proto    Pre  Cost  Flags  NextHop
Interface
0.0.0.0/0            Static      60    0    RD     10.0.12.2
GigabitEthernet0/0/0
10.0.1.1/32  Direct    0   0   D     127.0.0.1  LoopBack0
.....

```

可以看到，R1的IP路由表中的缺省路由发生了改变，新配置的静态缺省路由取代了原来的 IBGP 缺省路由，这是因为静态路由的协议优先级比 IBGP 路由的协议优先级要高（注意，协议优先级的值越小，优先级越高）。

在R1上使用tracert命令验证从10.0.1.1/32去往10.0.5.5/32的报文的转发路径。

```
<R1>tracert -a 10.0.1.1 10.0.5.5
```

```
tracert to 10.0.5.5(10.0.5.5), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.12.2 20 ms 10 ms 10 ms
```

```
2 10.0.25.5 20 ms 20 ms 10 ms
```

可以看到，报文是经过R2转发的，说明报文在R1上是选用了静态缺省路由发出的。

思考

实验步骤4中，如果R3不把发往R5的缺省路由过滤掉，可能会出现什么不良后果呢？

3.17 BGP 路由衰减

原理概述

相信读者已经非常清楚，路由的不稳定性对于网络、特别是大型网络来说具有极大的危害性。虽然路由聚合的方法能够减轻路由的不稳定性对整个网络的负面影响，但在很多情况下，路由聚合方法本身却是难以实现的。

顺便说明一下，常见的路由不稳定性包括了路由震荡（Route Oscillation）和路由抖动（Route Flapping）。虽然这两个术语经常被混用，但其实严格来讲二者是有区别的：前者的不稳定性具有时间上的周期性，后者的不稳定性具有时间上的随机性。

为了减轻路由的不稳定性对整个网络的负面影响，除了可以使用路由聚合的方法之外，还可以使用路由衰减（Route Dampening）的方

法。具有路由衰减功能的BGP 路由器在接收到一条不稳定的路由后，会酌情考虑是否将这条路由通告给EBGP邻居。

路由衰减方法会给接收到的不稳定的路由记上惩罚性点数（Penalty），比如，路由每抖动一次（不管是从可达到不可达的变化，还是从不可达到可达的变化，都算抖动一次），便追加记上500点，也就是Penalty值增加500点。显然，某条路由抖动越频繁，则这条路由的Penalty值就越大，但规定最大不能超过Ceiling Value。同时，Penalty值又会随时间自动逐渐减小，减少的速度为每HalfLife Time 的时间减少一半。如果某条未被抑制的路由的Penalty值上升超过了Suppress-Limit，该路由就会开始被抑制，也就是不能被使用，也不能被通告给EBGP邻居；如果被抑制的路由的Penalty值降低低于了Reuse Value，该路由的抑制状态就会被解除，也就是又可以被使用和被通告给EBGP邻居了。

特别强调一下，路由衰减机制只会对EBGP邻居产生影响，对于IBGP邻居不起任何作用。另外，不同路由器厂商在实现BGP路由衰减功能时，所使用的术语及一些参数的默认值等都存在一定的差别。

实验目的

- 理解BGP 路由衰减的基本原理和过程
- 掌握BGP 路由衰减参数的配置方法

实验内容

实验拓扑如图3-18所示，实验编址如表3-17所示。本实验网络中，R1和R2属于AS 100，R3 属于 AS 200。所有路由器都运行 BGP 协议，并且都使用直连接口来建立BGP 邻居关系。所有路由上都开启了路由衰减功能，R2 的Loopback 1接口所在的网段路由用来模拟一条不稳定的路由。实验的主要内容是观察路由衰减的现象并熟悉与路由衰减有关的各种参数。

实验拓扑

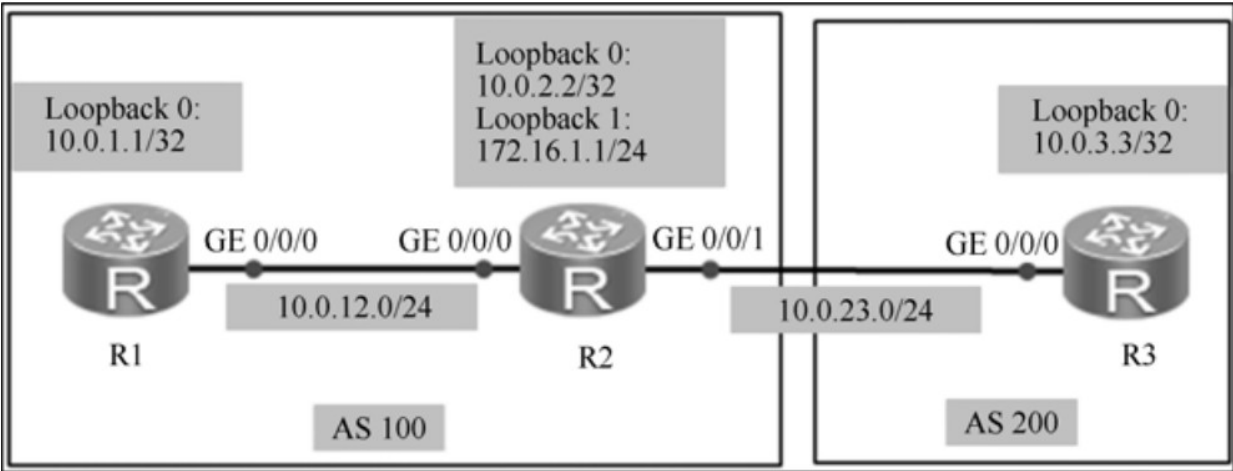


图3-18 BGP路由衰减

实验编址表

表3-17 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR3260)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR3260)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.23.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
	Loopback 1	172.16.1.1	255.255.255.0	N/A
R3(AR3260)	GE 0/0/0	10.0.23.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-18和表3-17进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
  Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=80 ms
--- 10.0.12.2 ping statistics ---
  1 packet(s) transmitted
  1 packet(s) received
```

0.00% packet loss

round-trip min/avg/max = 80/80/80 ms

R2与R3之间的连通性测试过程在此省略。

2 . 配置BGP路由协议

在每台路由器上配置BGP路由协议，EBGP邻居关系和IBGP邻居关系都使用直连物理接口来建立。

```
[R1]bgp 100
[R1-bgp]router-id 10.0.1.1
[R1-bgp]peer 10.0.12.2 as-number 100
[R1-bgp]network 10.0.1.1 32
[R2]bgp 100
[R2-bgp]router-id 10.0.2.2
[R2-bgp]peer 10.0.12.1 as-number 100
[R2-bgp]peer 10.0.12.1 next-hop-local
[R2-bgp]peer 10.0.23.3 as-number 200
[R2-bgp]network 10.0.2.2 32
[R2-bgp]network 172.16.1.0 24
[R3]bgp 200
[R3-bgp]router-id 10.0.3.3
[R3-bgp]peer 10.0.23.2 as-number 100
[R3-bgp]network 10.0.3.3 32
```

配置完成后，查看R1和R3的BGP路由表。

```
[R1]display bgp routing-table
```

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	0.0.0.0	0		0	i
*>i	10.0.2.2/32	10.0.12.2	0	100	0	i
*>i	10.0.3.3/32	10.0.12.2	0	100	0	200i
*>i	172.16.1.0/24	10.0.12.2	0	100	0	i

[R3]display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.23.2			0	100i
*>	10.0.2.2/32	10.0.23.2	0		0	100i
*>	10.0.3.3/32	0.0.0.0	0		0	i
*>	172.16.1.0/24	10.0.23.2	0		0	100i

可以看到，R1 和 R3 都已经学习到了关于 R2 的 Loopback 1 接口所在网段的路由信息。

在R1上测试10.0.1.1与172.16.1.1之间的连通性。

<R1>ping -c 1 -a 10.0.1.1 172.16.1.1

PING 172.16.1.1: 56 data bytes, press CTRL_C to break

Reply from 172.16.1.1: bytes=56 Sequence=1 ttl=255 time=20 ms

--- 172.16.1.1 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 20/20/20 ms

在R3上测试10.0.3.3与172.16.1.1之间的连通性。

```
<R3>ping -c 1 -a 10.0.3.3 172.16.1.1
```

```
PING 172.16.1.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 172.16.1.1: bytes=56 Sequence=1 ttl=255 time=20 ms
```

```
--- 172.16.1.1 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

round-trip min/avg/max = 20/20/20 ms

可以看到，通信完全正常。

3．配置默认参数下的BGP路由衰减功能

在R1、R2、R3的BGP视图下使用dampening命令开启路由衰减功能。

```
[R1]bgp 100
```

```
[R1-bgp]dampening
```

```
[R2]bgp 100
```

```
[R2-bgp]dampening
```

```
[R3]bgp 200
```

```
[R3-bgp]dampening
```

配置完成后，使用命令display bgp routing-table dampening parameter查看BGP衰减的配置参数。

```
[R1]display bgp routing-table dampening parameter
```

```
Maximum Suppress Time(in second) : 3973
```

```
Ceiling Value : 16000
```

```
Reuse Value : 750
```

```
HalfLife Time(in second) : 900
```

Suppress-Limit : 2000

回显信息中包括了与路由衰减相关的几个参数的默认值，其中 Reuse Value 为 750点，Suppres-Limit 为2000点，Ceiling Value 为16000点，HalfLife Time 为900s，Maximum Suppress Time 为3973s。

在R2 上的 Loopback 1 接口下交替使用 shutdown 命令和 undo shutdown 命令，模拟R2 的Loopback 1 接口不稳定的情况，然后在R1 和R3 上查看BGP 路由表。

```
[R1]display bgp routing-table
```

BGP Local router ID is 10.0.1.1

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	0.0.0.0	0	0		i
*>i	10.0.2.2/32	10.0.12.2	0	100	0	i
*>i	10.0.3.3/32	10.0.12.2	0	100	0	200i
*>i	172.16.1.0/24	10.0.12.2	0	100	0	i

```
[R3]display bgp routing-table
```

BGP Local router ID is 10.0.3.3

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.23.2		0	100i	
*>	10.0.2.2/32	10.0.23.2	0	0	100i	

```
*> 10.0.3.3/32    0.0.0.0    0        0        i
    d 172.16.1.0/24 10.0.23.2  0        0        100i
```

仔细观察会发现，当R2的Loopback 1 接口所在网段的路由发生抖动后，R3 的BGP路由表中对应的路由条目的状态码（Status Code）变为了d（damped），表示路由处于抑制的状态。然而在R1的BGP路由表中，相应的路由条目并无异常，原因是BGP路由衰减对IBGP邻居不起作用。

在R3上测试10.0.3.3与172.16.1.1之间的连通性。

```
<R3>ping -a 10.0.3.3 172.16.1.1
```

```
PING 172.16.1.1: 56 data bytes, press CTRL_C to break
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
Request time out
```

```
--- 172.16.1.1 ping statistics ---
```

```
5 packet(s) transmitted
```

```
0 packet(s) received
```

```
100.00% packet loss
```

可以看到，此时10.0.3.3与172.16.1.1之间无法通信，说明被抑制的BGP路由是不可用的。

在R1和R3上使用命令display bgp routing-table flap-info查看BGP路由抖动信息。

```
[R1]display bgp routing-table flap-info
```

```
Total Number of Routes: 0
```

```
[R3]display bgp routing-table flap-info
```

```
BGP Local router ID is 10.0.3.3
```


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

Total Number of Routes: 1

Network	From	Flaps	Duration	Reuse	Path/Origin
d 172.16.1.0/24	10.0.23.2	4	00:03:49	00:26:41	100i

可以看到，在R1的路由抖动信息表中，并未看到任何路由抖动的情况。在R3的路由抖动信息表中，可以看到172.16.1.0/24这条路由已经抖动了4次；Duration字段显示了这条路由已经被抑制了3min49s；Reuse字段显示按当前的情况计算（如果该路由不再抖动），还需要等待26min41s才能解除抑制。

4．根据需求修改与BGP衰减有关的参数

前面的路由衰减实验过程中，所使用的参数值都是系统的默认值。实际上，这些参数的值往往需要根据具体的情况进行修改。例如，就172.16.1.1/24这条路由来说，如果抖动的原因已经找到，并且故障也已排除，那么依据系统默认参数值，则该路由还需要被抑制大概26min，直到Reuse字段倒计为0的时候才能解除抑制。

在R3上配置HalfLife Time 为1min 即60s，Resue Value 为100点，Suppress-Limit为200点，Ceiling Value 为1001。

```
[R3]bgp 200
```

```
[R3-bgp]dampening 1 100 200 1001
```

配置完成后，使用display bgp routing-table dampening parameter命令查看BGP衰减的配置参数。

```
[R3]display bgp routing-table dampening parameter
```

```
Maximum Suppress Time(in second) : 199
```

```
Ceiling Value : 1001
```

```
Reuse Value : 100
```

HalfLife Time(in second) : 60

Suppress-Limit : 200

可以看到，有关参数的值已经发生了改变。另外需要说明的是，Maximum suppress Time这个参数的值是根据所配置的参数值和有关的数学公式进行计算而得到的。

重新进行实验，模拟路由抖动的故障，并查看路由器R3上的路由抖动信息。

```
[R3]display bgp routing-table flap-info
```

BGP Local router ID is 10.0.3.3

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

	Network	From	Flaps	Duration	Reuse	Path/Origin
d	172.16.1.0/24	10.0.23.2	2	00:01:23	00:02:45	100i

可以看到，172.16.1.0/24这条路由已经抖动了两次，已经被抑制了1min23s；如果不再抖动，大概再过 2min45s 就可以解除抑制了。显然，Reuse 字段的倒计时时间远远小于之前的二十几min。

等待大约3min（注意，这段时间不要再产生路由抖动）后，查看R3的BGP路由表。

```
[R3]display bgp routing-table
```

BGP Local router ID is 10.0.3.3

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 4

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn

```
*> 10.0.1.1/32    10.0.23.2          0      100i
*> 10.0.2.2/32    10.0.23.2  0          0      100i
*> 10.0.3.3/32    0.0.0.0    0          0          i
*> 172.16.1.0/24   10.0.23.2  0          0      100i
```

可以看到，此时路由已经完全恢复了正常。

在R3上测试10.0.3.3与172.16.1.1之间的连通性。

```
<R3>ping -c 1 -a 10.0.3.3 172.16.1.1
```

```
PING 172.16.1.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 172.16.1.1: bytes=56 Sequence=1 ttl=255 time=10 ms
```

```
--- 172.16.1.1 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 10/10/10 ms
```

可以看到，通信正常，说明172.16.1.0/24这条路由的确已经被解除了抑制。

思考

Reuse Value、Suppress-Limit、Ceiling Value 这3个参数值的大小顺序是怎样的？

3.18 BGP 监测和调试

原理概述

为了监测 BGP 协议的工作状态，VRP 系统提供了一系列的查询命令。熟练使用这些命令，可以全面地了解网络的运行情况。同时，VRP系统还提供了一系列的调试命令，用以详细地了解和调试BGP的

工作过程，并知道工作过程中各种事件的细节和关系。查询命令和调试命令的结合使用，有助于快速查找到网络的故障点和故障原因，提高查错排错的效率。

实验目的

- 掌握监测BGP 工作状态的方法
- 掌握调试BGP 工作过程的方法

实验内容

实验拓扑如图3-19 所示，实验编址如表 3-18 所示。本实验网络中，R1、R2、R3都运行BGP 协议，R1 属于AS 100，R2 和R3 属于AS 200，AS 200 内运行OSPF 协议。R1 和R3的Loopback 1接口分别用来模拟AS 100 和AS 200中的两个内部网络，这两个内部网络都需要被通告进BGP进程。R1与R2的EBGP邻居关系采用直连物理接口来建立，R2与R3 的IBGP 邻居关系采用Loopback 0 接口来建立。另外，R1 在将路由信息传递给R2时需要添加自定义团体属性，且R1和R2的GE 0/0/0接口需要配置简单的密码认证功能。实验过程中会使用一些监测和调试命令来了解网络的运行状态和工作过程。

实验拓扑

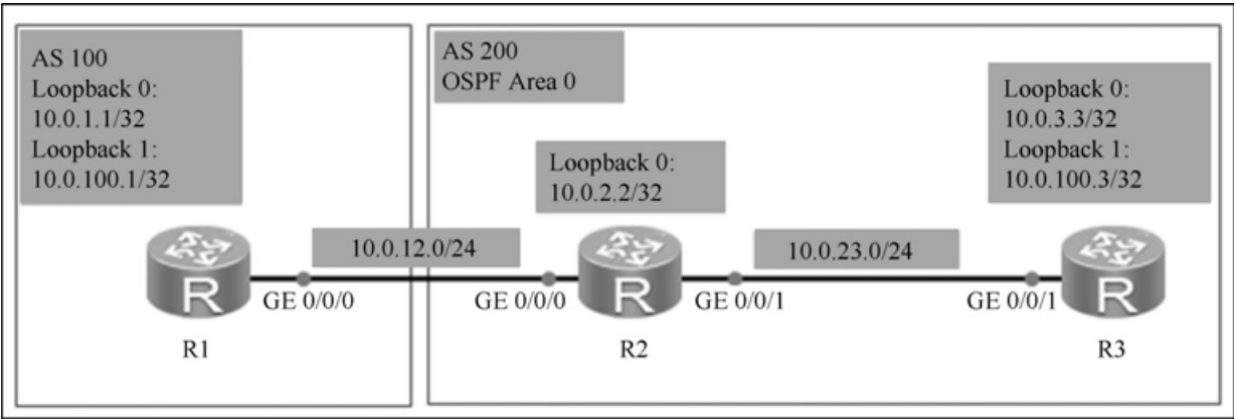


图3-19 BGP监测和调试

实验编址表

表3-18 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR3260)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	10.0.100.1	255.255.255.255	N/A
R2(AR3260)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.23.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR3260)	GE 0/0/1	10.0.23.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
	Loopback 1	10.0.100.3	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-19和表3-18进行相应的基本配置，并使用ping命令检测R2与R1之间的连通性。

```
<R2>ping -c 1 10.0.12.1
```

```
PING 10.0.12.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.1: bytes=56 Sequence=1 ttl=255 time=530 ms
```

```
--- 10.0.12.1 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 530/530/530 ms
```

R2与R3之间的连通性测试过程在此省略。

2．配置OSPF和BGP路由协议

在AS 200 内配置OSPF 协议作为IGP。

```
[R2]ospf 1 router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.23.2 0.0.0.0
```

```
[R3]ospf 1 router-id 10.0.3.3
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.23.3 0.0.0.0
```

配置完成后，在所有路由器上进行BGP协议的配置。

```
[R1]bgp 100
```

```
[R1-bgp]peer 10.0.12.2 as-number 200
```

```
[R1-bgp]peer 10.0.12.2 route-policy 1 export
```

```
[R1-bgp]peer 10.0.12.2 advertise-community
```

```
[R1-bgp]peer 10.0.12.2 password simple huawei
```

```
[R1-bgp]network 10.0.100.1 32
```

```
[R1-bgp]route-policy 1 permit node 10
```

```
[R1-route-policy]apply community 100:1
```

```
[R2]bgp 200
```

```
[R2-bgp]peer 10.0.12.1 as-number 100
```

```
[R2-bgp]peer 10.0.12.1 password simple huawei
```

```
[R2-bgp]peer 10.0.3.3 as-number 200
```

```
[R2-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```

```
[R2-bgp]peer 10.0.3.3 next-hop-local
```

```
[R3]bgp 200
```

```
[R3-bgp]peer 10.0.2.2 as-number 200
```

```
[R3-bgp]peer 10.0.2.2 connect-interface LoopBack 0
```

```
[R3-bgp]network 10.0.100.3 32
```

3．监测BGP协议的基本状态

BGP 协议的运行主要分为 3 个部分，即邻居的建立、路由的发布和策略控制。对BGP协议的监测和调试一般也是按照这个顺序来逐步进行的。

在R2 上使用display bgp peer 命令查看BGP 邻居信息。

```

[R2]display bgp peer
BGP local router ID : 10.0.12.2
Local AS number : 200
Total number of peers : 2          Peers in established state : 2
Peer      V    AS    MsgRcvd    MsgSent    OutQ    Up/Down
State      PrefRcv
10.0.3.3   4   200   19        21         0    00:16:50 Established    1
10.0.12.1  4   100   21        21         0    00:18:26 Established
1

```

从上面的回显信息中可以看到当前路由器的Router-ID及AS编号，当前路由器有哪些BGP邻居，邻居所属的AS编号，BGP协议版本号，当前路由器发送和接收的BGP消息数量，BGP邻居关系已经建立了多长时间以及目前的状态等。

在R2上使用display bgp peer 10.0.3.3 verbose命令查看BGP邻居10.0.3.3的详细信息。

```

[R2]display bgp peer 10.0.3.3 verbose
BGP Peer is 10.0.3.3, remote AS 200
Type: IBGP link
BGP version 4, Remote router ID 10.0.23.3
Update-group ID: 1
BGP current state: Established, Up for 00h21m09s
BGP current event: KATimerExpired
BGP last state: OpenConfirm
BGP Peer Up count: 1
Received total routes: 1
Received active routes total: 0
Advertised total routes: 1

```

Port: Local - 179 Remote - 50277
Configured : Connect-retry Time: 32 sec
Configured : Active Hold Time: 180 sec Keepalive Time:60 sec
Received : Active Hold Time: 180 sec
Negotiated : Active Hold Time: 180 sec Keepalive Time:60 sec
Peer optional capabilities:
Peer supports bgp multi-protocol extension
Peer supports bgp route refresh capability
Peer supports bgp 4-byte-as capability
Address family IPv4 Unicast: advertised and received
Received: Total 24 messages
Update messages 1
Open messages 1
KeepAlive messages 22
Notification messages 0
Refresh messages 0
Sent:Total 26 messages
Update messages 1
Open messages 3
KeepAlive messages 22
Notification messages 0
Refresh messages 0
Authentication type configured: None
Last keepalive received: 2013/09/20 11:14:01 UTC-05:13
.....

从上面的回显信息中可以看到关于 R2的 BGP邻居 R3 (10.0.3.3) 的许多详细情况 , 例如发送和接收的BGP消息的统计信息 , 定时器的

时间信息，配置的认证类型等。

当BGP邻居关系成功建立之后，一般还需要查看BGP协议所获得的BGP路由信息。在R2上使用display bgp routing-table 命令查看BGP路由表。

```
[R2]display bgp routing-table
```

```
BGP Local router ID is 10.0.12.2
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 2
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.12.1	0		0	100i
*>i	10.0.100.3/32	10.0.3.3	0	100	0	i

可以看到，R2接收到了关于10.0.100.1/32和10.0.100.3/32的路由信息。

为了进一步了解BGP路由信息的传递过程，还可以查看从特定邻居那里接收或者传递给特定邻居的路由信息。在R2上使用display bgp routing-table peer 10.0.3.3 advertised-routes命令查看R2传递给BGP邻居R3 (10.0.3.3) 的路由信息。

```
[R2]display bgp routing-table peer 10.0.3.3 advertised-routes
```

```
BGP Local router ID is 10.0.12.2
```

```
Status codes:*- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 1
```

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.100.1/32	10.0.2.2	0	100	0	100i

可以看到，R2向R3传递了关于10.0.100.1/32的路由信息。

团体属性是一种常用的控制 BGP 路由信息传递的路由属性，在 R2 上使用 display bgp routing-table community 命令可以专门查看R2 中带有团体属性的路由信息。

```
[R2]display bgp routing-table community
```

```
BGP Local router ID is 10.0.12.2
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

```
Total Number of Routes: 1
```

Network	NextHop	MED	LocPrf	PrefVal	Community
*> 10.0.100.1/32	10.0.12.1	0		0	<100:1>

可以看到，R2的BGP路由表中10.0.100.1/32这条路由的团体属性值为100:1。

4．调试BGP协议的工作过程

在R2 上使用terminal debugging 命令开启调试功能。

```
<R2>terminal debugging
```

由于调试命令一般会产生大量的输出信息，所以如果使用不当就可能会导致网络出现故障，因此，使用调试功能时需要尽可能做到有的放矢，尽量避免使用诸如debugging ip packet这样的会产生非常大量的输出信息的调试命令。

在R2 使用调试命令debugging bgp event，看看有什么输出。

```
<R2>debugging bgp event
```

```
Sep 20 2013 11:28:02.103.1-05:13 R2 RM/6/RMDEBUG:
```

```
BGP.public: 10.0.3.3 Current event is RecvKeepAliveMessage.
```

```
Sep 20 2013 11:28:02.123.1-05:13 R2 RM/6/RMDEBUG:
```

```
BGP.Public: 10.0.3.3 Current event is KATimerExpired.
```

Sep 20 2013 11:28:25.663.1-05:13 R2 RM/6/RMDEBUG:

BGP.public: 10.0.12.1 Current event is RecvKeepAliveMessage.

Sep 20 2013 11:28:26.93.1-05:13 R2 RM/6/RMDEBUG:

BGP.Public: 10.0.12.1 Current event is KATimerExpired.

<R2>undo debugging all

Info: All possible debugging has been turned off

可以看到，该命令输出了当前 BGP 协议的事件，路由器正在稳定地收发 BGP KeepAlive报文。观察和分析表明，当前的BGP协议工作在正常的稳定状态。

前面提到，为避免大量的信息输出，一般不直接使用debugging bgp packet 命令。但是，结合适当的关键字来使用该命令却是一种很好的做法，这样既能减少信息输出量，又能更有针对性地查找问题。常见的关键字有 Keepalive、Open、Route-Refresh、Update、Send、Receive等。例如，同时使用关键字Update和Receive，就可以只对接收到的BGP Update报文进行调试。

在R2上调试接收的Update报文。

<R2>debugging bgp update receive

结果发现，没有任何回显信息，这是因为BGP协议在路由信息未发生改变时并不会发送路由更新报文。为了看到回显信息，可以在R2上使用refresh bgp all import命令来强行刷新接收到的路由。

<R2>debugging bgp update receive

<R2>refresh bgp all import

Sep 20 2013 11:31:25.603.1-05:13 R2 RM/6/RMDEBUG:

BGP.Public: Recv UPDATE from 10.0.3.3 with following destinations :

Update message length : 56

MP_reach : AFI/SAFI 1/1

Origin : IGP

AS Path :

Next Hop : 10.0.3.3

Local Pref : 100

MED : 0

10.0.100.3/32,

<R2>

Sep 20 2013 11:31:25.623.1-05:13 R2 RM/6/RMDEBUG:

BGP.Public: Recv UPDATE from 10.0.12.1 with following destinations

:

Update message length : 62

MP_reach : AFI/SAFI 1/1

Origin : IGP

AS Path : 100

Next Hop : 10.0.12.1

MED : 0

Community :<100:1>

10.0.100.1/32,

<R2>undo debugging all

Info: All possible debugging has been turned off

可以看到，现在的回显信息中包含了R2从R3那里接收到的Update报文以及从R1那里接收到的Update报文的详细信息。

思考

BGP邻居关系建立失败的原因通常有哪些？

3.19 BGP 故障排除

原理概述

BGP协议排障的大致思路是，首先检查BGP邻居关系是否正常，然后检查BGP路由是否正确。如有必要，再检查BGP协议与其他路由协议的协同方面是否存在问题。

由于 BGP 是基于 TCP 会话连接的，所以在检查BGP 邻居关系时，必须确认 TCP连接没有问题。另外，还需要注意BGP对等体双方的参数是否一致，EBGP对等体之间的跳数限制等问题。

如果BGP邻居关系正常，则下一步就是查看和分析BGP路由是否正确。和其他的路由协议一样，路由故障诊断的基本手段就是检查路由表，并对实际的路由条目和预期的路由条目进行比较，从中发现故障线索。

如果BGP路由正常，但仍然出现通信故障，则有可能是BGP与其他路由协议之间的协同出了问题，最常见的问题就是因IGP与IBGP不同步而产生了BGP路由黑洞。

实验目的

- 掌握排除BGP 邻居关系故障的方法
- 掌握排除BGP 路由故障的方法
- 掌握排除BGP 路由黑洞故障的方法

实验内容

实验拓扑如图3-20所示，实验编址如表3-19所示。本实验网络中，R1、R2、R4、R5 运行BGP 协议。R1属于AS 10，R2、R3、R4属于AS 100，R5属于AS 50，AS 100内使用OSPF 协议作为IGP，所有BGP 邻居关系都使用Loopback 0 接口来建立。R1的Loopback 1接口和R5的Loopback 1 接口模拟了两个需要进行通信的网络。实验过程中，会人为地制造一些故障点，然后再一步一步地进行故障排除。

实验拓扑

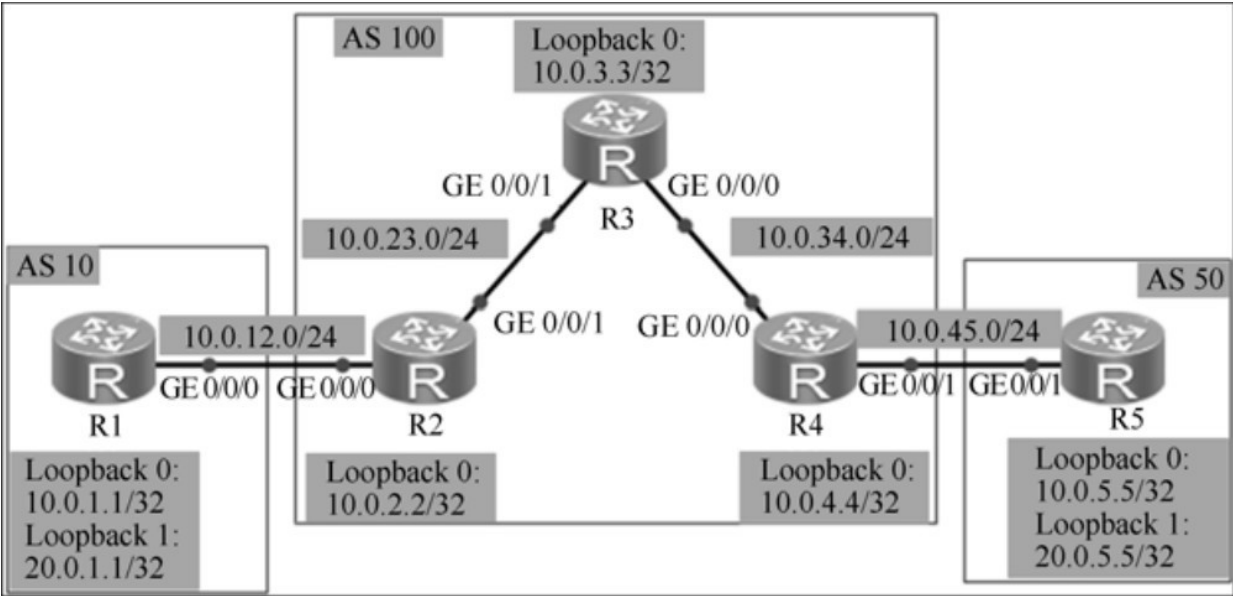


图3-20 BGP故障排除

实验编址表

表3-19 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	20.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.23.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.23.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2220)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.45.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
R5(AR2220)	GE 0/0/1	10.0.45.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A
	Loopback 1	20.0.5.5	255.255.255.255	N/A

实验步骤

1．基本配置

根据图3-20和表3-19进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=180 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 180/180/180 ms
```

其余直连网段的连通性测试过程在此省略。

在AS 100 中配置OSPF 协议作为IGP（配置过程在此省略），配置完成之后，在R3上使用display ospf peer 命令查看OSPF 邻居信息。

```
[R3]display ospf peer
```

```
OSPF Process 1 with Router ID 10.0.3.3
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.34.3(GigabitEthernet0/0/0)'s neighbors
```

```
Router ID: 10.0.4.4    Address: 10.0.34.4
```

```
State: Full Mode:Nbr is Master Priority: 1
```

```
DR: 10.0.34.4 BDR: 10.0.34.3 MTU: 0
```

```
Dead timer due in 37 sec
```

```
Retrans timer interval: 5
```

```
Neighbor is up for 00:00:12
```

```
Authentication Sequence: [ 0 ]
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.23.3(GigabitEthernet0/0/1)'s neighbors
```

```
Router ID: 10.0.2.2    Address: 10.0.23.2
```

```
State: Full Mode:Nbr is Slave Priority: 1
```

```
DR: 10.0.23.3 BDR: 10.0.23.2 MTU: 0
```

Dead timer due in 36 sec

Retrans timer interval: 5

Neighbor is up for 00:00:57

Authentication Sequence: [0]

可以看到，R3与R2和R4已经建立起了正常的OSPF邻接关系。

2．配置BGP路由协议并设置故障点

在配置BGP协议的过程中，人为制造一些故障点：R1 与R2 的认证密码不匹配；没有解除 R1 与 R2 的 EBGP 关系多跳限制；R1 向 R2 传递的路由携带了团体属性No-Advertise；R2向R4传递路由时未修改下一跳地址；R5的BGP对等体R4的AS编号出现了错误；R5缺少去往10.0.4.4/32的路由。

```
[R1]bgp 10
```

```
[R1-bgp]router-id 10.0.1.1
```

```
[R1-bgp]peer 10.0.2.2 as-number 100
```

```
[R1-bgp]peer 10.0.2.2 connect-interface LoopBack 0
```

```
[R1-bgp]peer 10.0.2.2 password simple huawei
```

```
[R1-bgp]peer 10.0.2.2 route-policy 1 export
```

```
[R1-bgp]peer 10.0.2.2 advertise-community
```

```
[R1-bgp]network 20.0.1.1 255.255.255.255
```

```
[R1-bgp]route-policy 1 permit node 10
```

```
[R1-route-policy]apply community no-advertise
```

```
[R1-route-policy]ip route-static 10.0.2.2 255.255.255.255 10.0.12.2
```

```
[R2]bgp 100
```

```
[R2-bgp]router-id 10.0.2.2
```

```
[R2-bgp]peer 10.0.1.1 as-number 10
```

```
[R2-bgp]peer 10.0.1.1 connect-interface LoopBack 0
```

```
[R2-bgp]peer 10.0.1.1 password simple huawei1
```



```
[R2-bgp]peer 10.0.4.4 as-number 100
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R2-bgp]peer 10.0.4.4 advertise-community
[R2-bgp]peer 10.0.4.4 password simple huawei
[R2-bgp]ip route-static 10.0.1.1 255.255.255.255 10.0.12.1
[R4]bgp 100
[R4-bgp]router-id 10.0.4.4
[R4-bgp]peer 10.0.2.2 as-number 100
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R4-bgp]peer 10.0.2.2 next-hop-local
[R4-bgp]peer 10.0.2.2 password simple huawei
[R4-bgp]peer 10.0.5.5 as-number 50
[R4-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R4-bgp]peer 10.0.5.5 ebgp-max-hop
[R4-bgp]peer 10.0.5.5 password simple huawei
[R4-bgp]ip route-static 10.0.5.5 255.255.255.255 10.0.45.5
[R5]bgp 50
[R5-bgp]router-id 10.0.5.5
[R5-bgp]peer 10.0.4.4 as-number 1000
[R5-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R5-bgp]peer 10.0.4.4 ebgp-max-hop
[R5-bgp]peer 10.0.4.4 password simple huawei
[R5-bgp]network 20.0.5.5 255.255.255.255
```

3 . 查找并排除BGP邻居关系故障

配置完成后，在R1上查看BGP邻居信息。

```
[R1]display bgp peer
```

BGP local router ID : 10.0.1.1

Local AS number : 10

Total number of peers : 1 Peers in established state : 0

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State

10.0.2.2	4	100	0	0	0	00:12:10	Connect

可以看到，R1与R2的BGP邻居关系停留在Connect状态，说明有问题存在。

在R1上打开调试功能。

<R1>debugging bgp all

<R1>terminal debugging

<R1>

Aug 20 2013 13:56:38.2.1-05:13 R1 RM/6/RMDEBUG:

BGP_TIMER: CR Timer Expired for Peer 10.0.2.2

<R1>

Aug 20 2013 13:56:38.2.2-05:13 R1 RM/6/RMDEBUG:

BGP.public: 10.0.2.2 Current event is CRTimerExpired.

<R1>

Aug 20 2013 13:56:38.2.3-05:13 R1 RM/6/RMDEBUG:

BGP.Public: 10.0.2.2 State is changed from CONNECT to CONNECT.

可以看到，输出提示CR Timer（重连计时器）超时，说明TCP连接无法建立。

在R1上使用display tcp status 命令查看TCP状态。

[R1]display tcp status

TCPCB	Tid/Soid	Local Add:port	Foreign Add:port	VPNID
b4a9db80	6/1	0.0.0.0:23	0.0.0.0:0	23553
				Listening

```
b4a9de08 164/1 0.0.0.0:179 10.0.2.2:0 0 Listening *
b4a9df4c 164/31 10.0.1.1:49428 10.0.2.2:179 0
```

Syn_Sent *

可以看到，R1的179端口处于监听状态，R1向目标10.0.2.2:179发送了SYN请求，但却没有接收到SYN_ACK。由于BGP协议配置中涉及TCP连接问题的只有对BGP对等体进行认证这一个特性，所以不妨在R1和R2上检查一下BGP配置情况。

```
[R1]bgp 10
```

```
[R1-bgp]display this
```

```
bgp 10
```

```
router-id 10.0.1.1
```

```
peer 10.0.2.2 as-number 100
```

```
peer 10.0.2.2 connect-interface LoopBack0
```

```
peer 10.0.2.2 password simple huawei
```

```
#
```

```
ipv4-family unicast
```

```
undo synchronization
```

```
network 20.0.1.1 255.255.255.255
```

```
peer 10.0.2.2 enable
```

```
peer 10.0.2.2 route-policy 1 export
```

```
peer 10.0.2.2 advertise-community
```

```
[R2]bgp 100
```

```
[R2-bgp]display this
```

```
bgp 100
```

```
router-id 10.0.2.2
```

```
peer 10.0.1.1 as-number 10
```

```
peer 10.0.1.1 connect-interface LoopBack0
```

```

peer 10.0.1.1 password simple huawei1
peer 10.0.4.4 as-number 100
peer 10.0.4.4 connect-interface LoopBack0
peer 10.0.4.4 password simple huawei
#
ipv4-family unicast
    undo synchronization
    peer 10.0.1.1 enable
    peer 10.0.4.4 enable

```

可以看到，R1上和R2上配置的密钥是不一致的。

将R2上配置的密钥修改为huawei。

```
[R2-bgp]peer 10.0.1.1 password simple huawei
```

配置完成后，在R1上查看BGP邻居关系。

```
[R1]display bgp peer
```

```
BGP local router ID : 10.0.1.1
```

```
Local AS number : 10
```

```
Total number of peers : 1      Peers in established state : 0
```

```
Peer    V    AS    MsgRcvd    MsgSent    OutQ Up/Down    State
```

```
PrefRcv
```

```
10.0.2.2  4   100   1         0         0    00:00:03  Idle   0
```

可以看到，修改密钥后，R1 与R2仍未建立起正常的BGP邻居关系，邻居关系状态处于Idle。

在R1上再次打开调试功能。

```
<R1>debugging bgp all
```

```
<R1>terminal debugging
```

```
<R1>
```

```
Aug 20 2013 14:07:02.522.10-05:13 R1 RM/6/RMDEBUG:
```

BGP.Public: 10.0.2.2 State is changed from OPENCONFIRM to IDLE.

<R1>

Aug 20 2013 14:07:02.522.11-05:13 R1 RM/6/RMDEBUG:

BGP peer: 10.0.2.2, SockID: 40, Read Sock API: Return value -9

从回显信息中可以看到，R2的状态从OpenConfirm回到了Idle，这种情况一般发生在TCP连接中断或传输出错的条件下。在实验条件下，一般不会出现线路拥塞丢包等问题，所以推测应该还是BGP 配置问题导致了传输错误。

检查R1的BGP配置情况。

```
[R1]bgp 100
```

```
[R1-bgp]display this
```

```
bgp 10
```

```
router-id 10.0.1.1
```

```
peer 10.0.2.2 as-number 100
```

```
peer 10.0.2.2 connect-interface LoopBack0
```

```
peer 10.0.2.2 password simple huawei
```

```
#
```

```
ipv4-family unicast
```

```
undo synchronization
```

```
network 20.0.1.1 255.255.255.255
```

```
peer 10.0.2.2 enable
```

```
peer 10.0.2.2 route-policy 1 export
```

```
peer 10.0.2.2 advertise-community
```

可以看到，R1 希望采用Loopback 0 接口与R2 建立EBGP 邻居关系，但是在配置的过程中却没有添加ebgp-max-hop功能。

检查R2的BGP配置情况。

```

[R2-bgp]display this
bgp 100
  router-id 10.0.2.2
  peer 10.0.1.1 as-number 10
  peer 10.0.1.1 connect-interface LoopBack0
  peer 10.0.1.1 password simple huawei
  peer 10.0.4.4 as-number 100
  peer 10.0.4.4 connect-interface LoopBack0
  peer 10.0.4.4 password simple huawei
#
ipv4-family unicast
  undo synchronization
  peer 10.0.1.1 enable
  peer 10.0.4.4 enable

```

可以看到，R2 也希望采用Loopback 0 接口与R1建立EBGP 邻居关系，但是在配置的过程中也没有添加ebgp-max-hop功能。

添加ebgp-max-hop功能如下，也就是解除R1与R2的EBGP关系的多跳限制。

```

[R1-bgp]peer 10.0.2.2 ebgp-max-hop
[R2-bgp]peer 10.0.1.1 ebgp-max-hop

```

配置完成后，在R1上查看BGP邻居信息。

```
[R1]display bgp peer
```

```
BGP local router ID : 10.0.1.1
```

```
Local AS number : 10
```

```
Total number of peers : 1      Peers in established state : 1
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State		PrefRcv				

```
10.0.2.2    4    100    2          4          0          00:00:08
Established    0
```

可以看到，R1与R2成功建立起了EBGP邻居关系，状态为Established。

在R2上查看BGP邻居信息。

```
[R2]display bgp peer
```

```
BGP local router ID : 10.0.2.2
```

```
Local AS number : 100
```

```
Total number of peers : 2    Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.1.1	4	10	4	3	0	00:01:37 Established

1

10.0.4.4	4	100	23	24	0	00:21:33 Established	0

可以看到，R2上的BGP邻居关系均已成功建立，状态均为Established。

在R4上查看BGP邻居信息。

```
[R4]display bgp peer
```

```
BGP local router ID : 10.0.4.4
```

```
Local AS number : 100
```

```
Total number of peers : 2    Peers in established state : 1
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	
10.0.2.2	4	100	26	26	0	00:24:09 Established	0
10.0.5.5	4	50	0	0	0	00:21:37 Connect	0

可以看到，R4与R5的邻居关系存在问题。

在R5上查看BGP邻居信息。

[R5]display bgp peer

BGP local router ID : 10.0.5.5

Local AS number : 50

Total number of peers : 1 Peers in established state : 0

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State

PrefRcv

10.0.4.4	4	1000	0	0	0	00:24:56	Idle	0

可以看到，R5与R4的邻居关系存在问题。

在R5上打开调试功能。

<R5>debugging bgp all

<R5>terminal debugging

<R5>

Aug 20 2013 14:15:50.955.1-05:13 R5 RM/6/RMDEBUG:

BGP_TIMER: CR Timer Expired for Peer 10.0.4.4

<R5>

Aug 20 2013 14:15:50.955.2-05:13 R5 RM/6/RMDEBUG:

BGP.public: 10.0.4.4 Current event is CRTimerExpired.

<R5>

Aug 20 2013 14:15:50.955.3-05:13 R5 RM/6/RMDEBUG:

BGP.public: 10.0.4.4 Current event is Start.

可以看到，输出提示CR Timer超时，说明应该是TCP 连接问题。

在R5上查看TCP状态。

[R5]display tcp status

TCPCB	Tid/SoId	Local Add:port	Foreign Add:port	VPNID

State

b4a5d41c	164/4	0.0.0.0:0	0.0.0.0:0	0	Closed

*


```

        b4a5d050    6/1      0.0.0.0:23      0.0.0.0:0      23553
Listening
        b4a5d2d8    164/1     0.0.0.0:179     10.0.4.4:0      0
Listening *
        b4a5d6a4    164/0     10.0.5.5:179    10.0.4.4:49265   0
Syn_Rcvd *
        b4a5d560    164/0     10.0.5.5:179    10.0.4.4:49814   0
Syn_Rcvd *
        b4a5d92c    164/0     10.0.5.5:179    10.0.4.4:50227   0
Syn_Rcvd *

```

可以观察到，R5正在监听179端口，状态处于Syn_Rcvd，这说明R5接收到了R4的SYN请求，但是R5并未向R4发送SYN请求，也没有回复SYN_ACK。出现这样的情况，很可能是因为R5不知道如何去往目标R4（10.0.4.4）。

在R5上使用display ip routing-table 命令查看IP 路由表。

```
[R5]display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```

-----
Routing Tables: Public
          Destinations : 9      Routes : 9
Destination/Mask    Proto Pre  Cost    Flags NextHop
-----
Interface
  10.0.5.5/32        Direct 0   0      D     127.0.0.1 LoopBack0
  10.0.45.0/24        Direct 0   0      D     10.0.45.5
GigabitEthernet0/0/1

```

```

10.0.45.5/32      Direct 0    0      D      127.0.0.1
GigabitEthernet0/0/1
10.0.45.255/32    Direct 0    0      D      127.0.0.1
GigabitEthernet0/0/1
20.0.5.5/32      Direct 0    0      D      127.0.0.1  LoopBack1
127.0.0.0/8      Direct 0    0      D      127.0.0.1  InLoopBack0
127.0.0.1/32     Direct 0    0      D      127.0.0.1  InLoopBack0
127.255.255.255/32 Direct 0    0      D      127.0.0.1
InLoopBack0
255.255.255.255/32 Direct 0    0      D      127.0.0.1
InLoopBack0

```

可以看到，R5的IP路由表中的确没有去往10.0.4.4/32的路由。
在R5上配置去往R4的静态路由。

```
[R5]ip route-static 10.0.4.4 255.255.255.255 10.0.45.4
```

配置完成后，在R5上查看BGP邻居信息。

```
[R5]display bgp peer
```

```
BGP local router ID : 10.0.5.5
```

```
Local AS number : 50
```

```
Total number of peers : 1      Peers in established state : 0
```

```

Peer   V   AS   MsgRcvd   MsgSent   OutQ   Up/Down   State
PrefRcv
10.0.4.4  4  1000    0        0        0    00:00:11   Idle    0

```

结果发现R5与R4还是未能建立起正常的BGP邻居关系。

在R5上打开调试功能。

```
<R5>debugging bgp all
```

```
<R5>terminal debugging
```

```
<R5>
```

Aug 20 2013 14:18:31.35.3-05:13 R5 RM/6/RMDEBUG:

BGP.Public: Send NOTIFICATION MSG to peer 10.0.4.4

Err/SubErr: 2/2 (OPEN Message Error/Bad Peer AS)

Error data 410400000064.

回显信息中出现了“Bad Peer AS”，说明R5的对等体的AS编号有错。

在R5上查看BGP的配置情况。

```
[R5]bgp 50
```

```
[R5-bgp]display this
```

```
bgp 50
```

```
router-id 10.0.5.5
```

```
peer 10.0.4.4 as-number 1000
```

```
peer 10.0.4.4 ebgp-max-hop 255
```

```
peer 10.0.4.4 connect-interface LoopBack0
```

```
peer 10.0.4.4 password simple huawei
```

```
#
```

```
ipv4-family unicast
```

```
undo synchronization
```

```
network 20.0.5.5 255.255.255.255
```

```
peer 10.0.4.4 enable
```

可以看到，100错成了1000。改错如下。

```
[R5-bgp]undo peer 10.0.4.4
```

```
[R5-bgp]peer 10.0.4.4 as-number 100
```

```
[R5-bgp]peer 10.0.4.4 connect-interface LoopBack0
```

```
[R5-bgp]peer 10.0.4.4 ebgp-max-hop
```

```
[R5-bgp]peer 10.0.4.4 password simple huawei
```

在R4和R5上查看BGP邻居信息。

[R4]display bgp peer

BGP local router ID : 10.0.4.4

Local AS number : 100

Total number of peers : 2 Peers in established state : 2

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.2.2	4	100	48	49	0	00:46:51 Established 0
10.0.5.5	4	50	4	4	0	00:01:37 Established

1

[R5]display bgp peer

BGP local router ID : 10.0.5.5

Local AS number : 50

Total number of peers : 1 Peers in established state : 1

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.4.4	4	100	5	6	0	00:03:48 Established 0

可以看到，R4与R5的BGP邻居关系已经正常了。

4．查找并排除BGP路由故障

至此，所有的BGP邻居关系已经正常了，接下来需要查找并排除BGP路由故障。

在R1上查看BGP路由表。

[R1]display bgp routing-table

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	20.0.1.1/32	0.0.0.0	0		0	i
*>	20.0.5.5/32	10.0.2.2			0	100 50i

可以看到，R1正常学习到了关于20.0.5.5/32的路由。

在R5上查看BGP路由表。

[R5]display bgp routing-table

BGP Local router ID is 10.0.5.5

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	20.0.5.5/32	0.0.0.0	0		0	i

可以看到，R5并没有学习到关于20.0.1.1/32的路由。

在R4上查看BGP路由表。

[R4]display bgp routing-table

BGP Local router ID is 10.0.4.4

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	20.0.5.5/32	10.0.5.5	0		0	50i

可以看到，R4也没有学习到关于20.0.1.1/32的路由。

在R2上查看BGP路由表。

[R2]display bgp routing-table

BGP Local router ID is 10.0.2.2

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	20.0.1.1/32	10.0.1.1	0	0	10i	
*>i	20.0.5.5/32	10.0.4.4	0	100	0	50i

可以看到，R2已经正常学习到了关于20.0.1.1/32的路由。在R2上，关于20.0.1.1/32的路由只有一条，并且是可用的状态，那么在正常情况下 R2 应该会把这条路由传递给R4，但R4现在并没有接收到这条路由，说明可能是该路由在传递过程中遇到了问题，并且可以怀疑是不是因为路由控制方面的原因所导致。

查看R2的BGP配置情况。

```
[R2]bgp 100
```

```
[R2-bgp]display this
```

```
bgp 100
```

```
router-id 10.0.2.2
```

```
peer 10.0.1.1 as-number 10
```

```
peer 10.0.1.1 ebgp-max-hop 255
```

```
peer 10.0.1.1 connect-interface LoopBack0
```

```
peer 10.0.1.1 password simple huawei
```

```
peer 10.0.4.4 as-number 100
```

```
peer 10.0.4.4 connect-interface LoopBack0
```

```
peer 10.0.4.4 password simple huawei
```

```
#
```

```
ipv4-family unicast
```

```
undo synchronization
```

```
peer 10.0.1.1 enable
```

```
peer 10.0.4.4 enable
```

可以看到，R2上没有调用任何路由策略来控制路由的发布。

查看R4的BGP配置情况。

```
[R4-bgp]display this
```

```
bgp 100
```

```
router-id 10.0.4.4
```

```
peer 10.0.2.2 as-number 100
```

```
peer 10.0.2.2 connect-interface LoopBack0
```

```
peer 10.0.2.2 password simple huawei
```

```
peer 10.0.5.5 as-number 50
```

```
peer 10.0.5.5 ebgp-max-hop 255
```

```
peer 10.0.5.5 connect-interface LoopBack0
```

```
peer 10.0.5.5 password simple huawei
```

```
#
```

```
ipv4-family unicast
```

```
undo synchronization
```

```
peer 10.0.2.2 enable
```

```
peer 10.0.2.2 next-hop-local
```

```
peer 10.0.5.5 enable
```

可以看到，R4上也未调用任何路由策略来控制路由的接收。我们知道，两个BGP对等体之间的路由传递过程是可以受到团体属性影响的，所以，不妨在R2上使用display bgp routing-table community 命令查看一下是否存在携带团体属性的路由。

```
[R2]display bgp routing-table community
```

```
BGP Local router ID is 10.0.2.2
```

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Community
*> 20.0.1.1/32	10.0.1.1	0	0		no-advertise

可以看到，正是20.0.1.1/32 这条路由携带了团体属性，并且是No-Advertise，这应该就是R4学习不到这条路由的真正原因。

在R2上配置Route-Policy，将从R1处接收到的路由条目的团体属性进行清除。

```
[R2]ip community-filter 1 permit no-advertise
```

```
[R2]route-policy 1 permit node 10
```

```
[R2-route-policy]apply comm-filter 1 delete
```

```
[R2-route-policy]bgp 100
```

```
[R2-bgp]peer 10.0.1.1 route-policy 1 import
```

在R2上查看BGP路由所携带的团体属性是否被清除了。

```
[R2]display bgp routing-table community
```

Total Number of Routes: 0

可以看到，团体属性已被清除，R2上现在不存在任何携带团体属性的路由了。

在R5上查看BGP路由表。

```
[R5]display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 20.0.5.5/32	0.0.0.0	0		0	i

可以看到，R5仍未接收到关于20.0.1.1/32的路由。

在R4上查看BGP路由表。

[R4]display bgp routing-table

BGP Local router ID is 10.0.4.4

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	20.0.1.1/32	10.0.1.1	0	100	0	10i
*>	20.0.5.5/32	10.0.5.5	0		0	50i

可以看到，R4能够接收到关于10.0.1.1/32的路由，同时也发现，这条路由不是最优的，下一跳为10.0.1.1。

在R4上查看IP路由表。

[R4]display ip routing-table

Route Flags: R - relay, D - download to fib

```

-----
                                     Routing Tables: Public
                                Destinations : 16  Routes : 16

Destination/Mask    Proto    Pre  Cost    Flags    NextHop
-----
Interface
10.0.2.2/32         OSPF     10    2        D        10.0.34.3
GigabitEthernet0/0/0

```

10.0.3.3/32	OSPF	10	1	D	10.0.34.3
GigabitEthernet0/0/0					
10.0.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.5.5/32	Static	60	0	RD	10.0.45.5
GigabitEthernet0/0/1					
10.0.23.0/24	OSPF	10	2	D	10.0.34.3
GigabitEthernet0/0/0					
10.0.34.0/24	Direct	0	0	D	10.0.34.4
GigabitEthernet0/0/0					
10.0.34.4/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.34.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.45.0/24	Direct	0	0	D	10.0.45.4
GigabitEthernet0/0/1					
10.0.45.4/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.45.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
20.0.5.5/32	EBGP	255	0	RD	10.0.5.5
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

```
255.255.255.255/32   Direct    0      0      D      127.0.0.1
```

InLoopBack0

可以看到，R4的IP路由表中根本就没有去往10.0.1.1/32的路由。

在默认情况下，路由器从EBGP邻居学习到的路由在被传递给自己的IBGP邻居时是不修改下一跳属性的，而这可能就是产生上述问题的原因。

在R2上查看BGP的配置情况。

```
[R2-bgp]display this
```

```
bgp 100
```

```
router-id 10.0.2.2
```

```
peer 10.0.1.1 as-number 10
```

```
peer 10.0.1.1 ebgp-max-hop 255
```

```
peer 10.0.1.1 connect-interface LoopBack0
```

```
peer 10.0.1.1 password simple huawei
```

```
peer 10.0.4.4 as-number 100
```

```
peer 10.0.4.4 connect-interface LoopBack0
```

```
peer 10.0.4.4 password simple huawei
```

```
#
```

```
ipv4-family unicast
```

```
undo synchronization
```

```
peer 10.0.1.1 enable
```

```
peer 10.0.1.1 route-policy 1 import
```

```
peer 10.0.4.4 enable
```

```
peer 10.0.4.4 advertise-community
```

可以看到，配置情况表明，R2在向R4传递路由信息时的确不会将路由的下一跳修改成自己。

修改配置如下。

```
[R2-bgp]peer 10.0.4.4 next-hop-local
```

在R5上查看BGP路由表。

```
[R5]display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 2

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	20.0.1.1/32	10.0.4.4			0	100 10i
*>	20.0.5.5/32	0.0.0.0	0		0	i

可以看到，R5现在正常接收到了关于20.0.1.1/32的路由。

5. 查找并排除其他故障

至此，R1 和R5 都已拥有了去往对方Loopback 1的路由。在R1 上检测20.0.1.1/32与20.0.5.5/32之间的连通性。

```
<R1>ping -a 20.0.1.1 20.0.5.5
```

PING 20.0.5.5: 56 data bytes, press CTRL_C to break

Request time out

Request time out

Request time out

Request time out

Request time out

--- 20.0.5.5 ping statistics ---

5 packet(s) transmitted

0 packet(s) received

100.00% packet loss

可以看到，R1 的Loopback 1 接口和R5 的Loopback 1 接口之间无法进行正常的通信，怀疑应该是BGP路由黑洞所致。

在R1上使用tracert命令检测从20.0.1.1/32去往20.0.5.5/32的每一跳信息。

```
<R1>tracert -a 20.0.1.1 20.0.5.5
```

```
tracert to 20.0.5.5(20.0.5.5), max hops: 30 ,packet length:
40,press CTRL_C to break
```

1 10.0.12.2 120 ms 10 ms 10 ms

2 * * *

可以看到，最后能够回复 ICMP Unreachable 消息的是 R2（10.0.12.2），说明报文离开 R2 之后可能掉进了 BGP 路由黑洞，并且推测黑洞应该在 R3 上。

在R3上查看IP路由表。

```
[R3]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 13 Routes : 13

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.2.2/32	OSPF	10	1	D	10.0.23.2
GigabitEthernet0/0/1					
10.0.3.3/32	Direct	0	0	D	127.0.0.1
10.0.4.4/32	OSPF	10	1	D	10.0.34.4
GigabitEthernet0/0/0					

10.0.23.0/24	Direct	0	0	D	10.0.23.3
GigabitEthernet0/0/1					
10.0.23.3/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.23.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.34.0/24	Direct	0	0	D	10.0.34.3
GigabitEthernet0/0/0					
10.0.34.3/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.34.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R3的IP路由表中并没有关于20.0.1.1/32和20.0.5.5/32的路由，这就证实了BGP路由黑洞的存在。接下来可使用GRE隧道方式来解决路由黑洞的问题。

```
[R2]interface Tunnel 0/0/0
[R2-Tunnel0/0/0]ip address 100.0.24.2 24
[R2-Tunnel0/0/0]tunnel-protocol gre
[R2-Tunnel0/0/0]source 10.0.23.2
```

```
[R2-Tunnel0/0/0]destination 10.0.34.4
```

```
[R2-Tunnel0/0/0]ip route-static 10.0.4.4 32 100.0.24.4 preference 1
```

```
[R4]interface Tunnel 0/0/0
```

```
[R4-Tunnel0/0/0]ip address 100.0.24.4 24
```

```
[R4-Tunnel0/0/0]tunnel-protocol gre
```

```
[R4-Tunnel0/0/0]source 10.0.34.4
```

```
[R4-Tunnel0/0/0]destination 10.0.23.2
```

```
[R4-Tunnel0/0/0]ip route-static 10.0.2.2 32 100.0.24.2 preference 1
```

配置完成后，在R1上再次检测20.0.1.1/32与20.0.5.5/32之间的连通性。

```
<R1>ping -c 1 -a 20.0.1.1 20.0.5.5
```

```
PING 20.0.5.5: 56 data bytes, press CTRL_C to break
```

```
Reply from 20.0.5.5: bytes=56 Sequence=1 ttl=252 time=30 ms
```

```
--- 20.0.5.5 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 30/30/30 ms
```

可以看到，通信正常。至此，所有的故障都已得到了排除。

思考

华为路由设备上，IBGP与IGP的同步功能在默认情况下是开启的吗？

第4章 IS-IS

4.1 IS-IS基本配置

4.2 IS-IS邻接关系

4.3 IS-IS链路状态数据库

4.4 IS-IS DIS

4.5 IS-IS开销值和协议优先级

4.6 IS-IS路由聚合

4.7 IS-IS缺省路由

4.8 IS-IS路由引入

4.9 IS-IS路由过滤

4.10 IS-IS路由渗透

4.11 IS-IS监测和调试

4.12 IS-IS故障排除

4.1 IS-IS 基本配置

原理概述

和OSPF路由协议一样，IS-IS也是一个应用非常广泛的IGP路由协议，很多ISP网络、特别是大型的ISP网络都部署了IS-IS路由协议。

RIP、OSPF 等许多IGP 都是针对IP（Internet Protocol）这个网络层协议而开发的路由协议，但IS-IS 最初是针对CLNP（Connection-Less Network Protocol）这个网络层协议而开发的路由协议。后来，进行扩展后的 IS-IS 既能够支持 CLNP，也能够支持 IP，这样的 IS-IS 协议被称为 Integrated IS-IS 协议。目前，通常情况下所说的 IS-IS 都是指 Integrated IS-IS 协议。

IS-IS 协议最初是由ISO（International Organization for Standardization）对其进行标准化工作的，所以 IS-IS 协议中有许多 ISO 的特殊用语，例如，主机（Host）被称为末端系统（End System），简称ES；路由器（Router）被称为中间系统（Intermediate System），简称IS；ES与IS之间的信息沟通协议被称为ES-IS协议，而IS与IS之间用来交换路由信息的协议被称为IS-IS协议。

IS-IS协议与OSPF协议非常相似。例如，它们都是基于链路状态的路由协议，都需要建立和维护链路状态数据库（LSDB），都使用Hello报文来建立和维护邻居/邻接关系，都具有区域化和层次化的结构，如此等等，这里就不再赘述了。

另一方面，IS-IS协议与OSPF协议又存在许多差别。例如，OSPF区域的分界位于路由器上，而IS-IS区域的分界位于链路上；OSPF协议支持点到点、点到多点、NBMA、Broadcast这4种类型的网络，而IS-IS协议只支持点到点和Broadcast这两种类型的网络，如此等等，这里不再赘述。

运行IS-IS协议的路由器（简称为IS-IS路由器）必须有一个被称为NET（Network Entity Title）的网络地址，即使是在IP 环境下也是如此。NET 也称为网络实体名，长度为8到20个字节，其格式可以多种多样。通常，在IP环境下NET的格式为：区域ID（1个字节）+ 系统ID（6个字节）+ SEL（1个字节）。例如，4A.2000.00E0.008C.00就是一个NET，其中的每一位都是一个十六进制数字，4A是区域ID，2000.00E0.008C是系统ID，末尾的00 是SEL。SEL 是NSAP（Network Service Access Point）Selector的简称，NET中的SEL总是为00。总之，

一个IS-IS路由器的网络实体名NET中包含了该路由器所属区域的ID，以及在这个区域中该路由器的身份识别标志，即系统ID。

另外，需要特别说明的是，在本章及本书中，所有图、表、和文字中涉及的IS-IS区域的区域ID指的都是十六进制数。例如，当描述某IS-IS路由器属于区域20时，这里的20指的是十六进制的20，相当于是十进制的32。

实验目的

- 理解网络实体名NET 的结构和含义
- 掌握IS-IS 协议的基本配置方法

实验内容

实验拓扑如图4-1所示，实验编址如表4-1所示。本实验模拟了一个简单的企业网络场景，路由器R1、R2、R3 的Loopback 0 接口分别模拟了企业内部的不同网络。网络需求是：全网运行 IS-IS 协议，实现企业内部不同网络的互通，并且各路由器接口都需要配置认证功能以保证网络的基本安全性。

实验拓扑

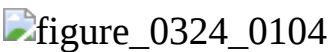
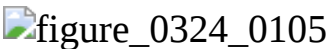


图4-1 IS-IS基本配置

实验编址表

表4-1 实验编址



实验步骤

1．基本配置

根据图4-1和表4-1进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=110 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 110/110/110 ms
```

R2与R3之间的连通性测试过程在此省略。

2．配置IS-IS路由协议

配置IS-IS协议首先要在系统视图下使用命令isis创建IS-IS进程。如果不指明IS-IS进程号，则进程号默认为1。

```
[R1]isis
```

然后，在IS-IS视图下使用network-entity命令配置路由器的网络实体名，即指定系统的区域ID和系统ID。

```
[R1-isis-1]network-entity 10.0000.0000.0001.00
```

与OSPF协议的配置不同，配置IS-IS时，路由器上需要运行IS-IS的接口必须使用isis enable 命令逐一进行IS-IS 协议的使能。

```
[R1-isis-1]quit
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable
```

```
[R1-GigabitEthernet0/0/0]interface LoopBack 0
```

```
[R1-LoopBack0]isis enable
```

至此，路由器R1上的IS-IS基本配置工作已告完成。在R2、R3上也进行类似的配置。

```
[R2]isis
```

```
[R2-isis-1]network-entity 10.0000.0000.0002.00
```

```
[R2-isis-1]quit
```

```
[R2]interface GigabitEthernet0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis enable
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]isis enable
```

```
[R2-GigabitEthernet0/0/1]interface LoopBack 0
```

```
[R2-LoopBack0]isis enable
```

```
[R3]isis
```

```
[R3-isis-1]network-entity 10.0000.0000.0003.00
```

```
[R3-isis-1]quit
```

```
[R3]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]isis enable
```

```
[R3-GigabitEthernet0/0/1]interface LoopBack 0
```

```
[R3-LoopBack0]isis enable
```

配置完成后，在R2 上使用命令display isis peer 查看IS-IS 邻居信息。

```
<R2>display isis peer
```

Peer information for IS-IS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type
PRI					

0000.0000.0001	GE0/0/0	0000.0000.0001.01	Up	14s	
L1(L1L2) 64					
0000.0000.0001	GE0/0/0	0000.0000.0001.01	Up	14s	
L2(L1L2) 64					
0000.0000.0003	GE0/0/1	0000.0000.0003.01	Up	8s	L1(L1L2)
64					
0000.0000.0003	GE0/0/1	0000.0000.0003.01	Up	8s	L2(L1L2)
64					

Total Peer(s): 4

可以看到，R2共有4条邻居信息，分别与系统ID为0000.0000.0001的邻居建立了Level-1邻接关系及Level-2邻接关系，与系统ID为0000.0000.0003的邻居建立了Level-1邻接关系及Level-2邻接关系。4条邻居信息的状态均为Up，表示IS-IS邻接关系已正常建立。在R1和R3上查看邻居信息会得到与上面一致的结论，这里不再赘述。

由于系统ID不易于管理和维护时的识别和认读，因此可以在IS-IS视图下使用命令is-name为系统设置一个动态主机名。

```
[R1-isis-1]is-name R1
```

```
[R2-isis-1]is-name R2
```

```
[R3-isis-1]is-name R3
```

配置完成后，重新在R2上查看IS-IS邻居信息。

```
<R2>display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	R1.01	Up	14s	L1(L1L2)	64

R1	GE0/0/0	R1.01	Up	14s	L2(L1L2)	64

R3	GE0/0/1	R3.01	Up	8s	L1(L1L2)	64

R3	GE0/0/1	R3.01	Up	8s	L2(L1L2)	64

Total Peer(s): 4

可以看到，系统ID 现在已经被所配置的动态主机名替换，Circuit ID中的系统ID 部分也已经被动态主机名替换。

查看R1的路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations :11 Routes :11

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.2.2/32 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.2
10.0.3.3/32 GigabitEthernet0/0/0	ISIS-L1	15	20	D	10.0.12.2
10.0.12.0/24	Direct	0	0	D	10.0.12.1 GigabitEthernet0/0/0
10.0.12.1/32	Direct	0	0	D	127.0.0.1 GigabitEthernet0/0/0
10.0.12.255/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.23.0/24 GigabitEthernet0/0/0	ISIS-L1	15	20	D	10.0.12.2
127.0.0.0/8	Direct	0	0	D	127.0.0.1 InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1 InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1 InLoopBack0
255.255.255.255/32	Direct	0	0	D	127.0.0.1 InLoopBack0

可以看到，R1 已经获得了其他路由器的Loopback 0 接口的路由，以及其他非直连网段的路由。在R2和R3上查看路由表后，可以发现同样的结果，这里不再赘述。至此，企业网络已经实现了全网互通。

3．配置IS-IS认证功能

与OSPF协议一样，IS-IS也能够支持使用诸如明文、MD5及keychain等方式的认证功能，这里使用基于接口的MD5认证功能来保证网络的基

本安全性。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis authentication-mode md5 plain Huawei
```

在 R1 的 GE 0/0/0 接口进行完上述配置之后，系统输出日志信息，提示 R1 与邻居 0000.0000.0002 的邻接关系由于 Hold Timer 超时而变为 Down 状态。

```
[R1-GigabitEthernet0/0/0]
```

```
Sep 21 2013 08:52:09-05:13 R1
```

```
%%01ISIS/4/PEER_DWN_HLDTMR_EXPR(l)[3]:ISIS 256 neighbor  
0000.0000.0002 was Down on interface GE0/0/0 because hold timer  
expired. The Hello packet was received at 08:52:00 last time; the maximum  
interval for sending Hello packets was 1242890240; the local router sent  
3825205248 Hello packets and received 67108864 packets; the type of the  
Hello packet was Lan Level-1; CPU usage was 486539264%.
```

```
[R1-GigabitEthernet0/0/0]
```

```
Sep 21 2013 08:52:09-05:13 R1 %%01ISIS/4/ADJ_CHANGE_LEVEL(l)  
[4]:The neighbor of ISIS was changed. (IsisProcessId=256,  
Neighbor=0000.0000.0002, InterfaceName=GE0/0/0, CurrentState=down,  
ChangeType=L1_HOLDTIMER_EXPIRED, Level=Level-1)
```

```
[R1-GigabitEthernet0/0/0]
```

```
Sep 21 2013 08:52:12-05:13 R1
```

```
%%01ISIS/4/PEER_DWN_HLDTMR_EXPR(l)[5]:ISIS 256 neighbor  
0000.0000.0002 was Down on interface GE0/0/0 because hold timer  
expired. The Hello packet was received at 08:52:08 last time; the maximum  
interval for sending Hello packets was 1846804480; the local router sent  
3825205248 Hello packets and received 83886080 packets; the type of the  
Hello packet was Lan Level-2; CPU usage was 486539264%.
```

```
[R1-GigabitEthernet0/0/0]
```



```
Sep 21 2013 08:52:12-05:13 R1 %%%01ISIS/4/ADJ_CHANGE_LEVEL(l)
[6]:The neighbor of ISIS was changed. (IsisProcessId=256,
Neighbor=0000.0000.0002, InterfaceName=GE0/0/0, CurrentState=down,
ChangeType=L2_HOLDTIMER_EXPIRED, Level=Level-2)
```

在R2上查看IS-IS邻居信息。

```
<R2>display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	---	Init	23s	L1(L1L2)	64
R1	GE0/0/0	---	Init	26s	L2(L1L2)	64
R3	GE0/0/1	R3.01	Up	8s	L1(L1L2)	64
R3	GE0/0/1	R3.01	Up	7s	L2(L1L2)	64

Total Peer(s): 4

可以看到，R2与R1的邻居状态现在处于Init。通过上面的系统输出日志可以知道，邻接关系当初变为Down的原因是Hold Timer 超时，这是因为启用IS-IS 的接口认证功能后，接口会对Level-1和Level-2的Hello报文进行认证。

为了让R1和R2 重新建立起邻接关系，R2 的GE 0/0/0 接口也需要配置相应的认证功能。

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis authentication-mode md5 plain Huawei
```

配置完成之后，系统输出日志信息，提示R2与邻居0000.0000.0001的邻接关系发生了改变，R2与R1重新建立起了邻接关系。

```
[R2-GigabitEthernet0/0/0]
```

```
Sep 21 2013 09:26:34-05:13 R2 %%01ISIS/4/ADJ_CHANGE_LEVEL(l)
[0]:The neighbor of ISIS was changed. (IsisProcessId=256,
Neighbor=0000.0000.0001, InterfaceName=GE0/0/0, CurrentState=up,
ChangeType=NEW_L1_ADJ, Level=Level-1)
```

```
[R2-GigabitEthernet0/0/0]
```

```
Sep 21 2013 09:26:35-05:13 R2 %%01ISIS/4/ADJ_CHANGE_LEVEL(l)
[1]:The neighbor of ISIS was changed. (IsisProcessId=256,
Neighbor=0000.0000.0001, InterfaceName=GE0/0/0, CurrentState=up,
ChangeType=NEW_L2_ADJ, Level=Level-2)
```

在R2上查看IS-IS邻居信息。

```
[R2]display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	R1.01	Up	9s	L1(L1L2)	64
R1	GE0/0/0	R1.01	Up	9s	L2(L1L2)	64
R3	GE0/0/1	R3.01	Up	8s	L1(L1L2)	64
R3	GE0/0/1	R3.01	Up	7s	L2(L1L2)	64

Total Peer(s): 4

可以看到，R2与R1的Level-1邻接关系及Level-2邻接关系的状态均为Up，说明R2与R1重新建立起了邻接关系。

继续R2和R3上接口认证功能的配置。

```
[R2]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]isis authentication-mode md5 plain huawei
```

```
[R3]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]isis authentication-mode md5 plain huawei
```

配置完成后，在R3上观察IS-IS邻居信息。

```
[R3]display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R2	GE0/0/1	R3.01	Up	27s	L1(L1L2)	64

R2	GE0/0/1	R3.01	Up	27s	L2(L1L2)	64

Total Peer(s): 2

可以看到，R3与R2已经重新建立起了邻接关系。

思考

OSPF网络中，区域0专门用来表示骨干区域。IS-IS网络中也是这样的吗？

4.2 IS-IS 邻接关系

原理概述

在IS-IS协议中，路由器的IS-IS接口有3种不同的类型或级别：Level-1、Level-2、Level-1-2。Level-1 接口只能发送和接收IS-IS Level-1 Hello 消息，Level-2 接口只能发送和接收 IS-IS Level-2 Hello 消息，Level-1-2 接口同时能发送和接收 IS-IS Level-1 和Level-2 Hello 消息。

相应地，IS-IS 路由器也有 3 种不同的类型或级别：Level-1、Level-2、Level-1-2。如果一台路由器的所有IS-IS接口都是Level-1接口，则这种路由器称为Level-1路由器；如果一台路由器的所有IS-IS接口都是Level-2接口，则这种路由器称为Level-2路由器；如果一台路由器既有Level-1接口，又有Level-2接口，或者该路由器拥有Level-1-2接口，则这种路由器称为Level-1-2路由器。默认情况下，路由器的IS-IS接口都为Level-1-2接口，因此，IS-IS路由器在默认情况下都是Level-1-2路由器。根据IS-IS协议的设计思想，Level-1路由器部署在IS-IS区域内，Level-2路由器部署在IS-IS区域之间，Level-1-2路由器部署在Level-1路由器与Level-2路由器之间。

在 IS-IS 协议中，路由器之间的邻接关系分为两种类型或级别：通过交换 Level-1 Hello 消息而建立的邻接关系称为Level-1 邻接关系，通过交换Level-2 Hello消息而建立的邻接关系称为Level-2邻接关系。显然，两台路由器之间可以同时具有Level-1邻接关系和Level-2邻接关系。

在 IS-IS 协议中，Level-1 邻接关系只能够在区域 ID 相同的路由器之间建立，而Level-2 邻接关系的建立则无需考虑区域ID是否相同。所有建立了Level-2 邻接关系的路由器，即所有相连的Level-1-2路由器与Level-2路由器共同构成了IS-IS的骨干区域。另外需要说明的是，Level-1-2 路由器既能够与拥有相同区域 ID 的 Level-1 路由器建立Level-1邻接关系，又能够与Level-2路由器建立Level-2邻接关系。

OSPF和IS-IS都是基于链路状态的路由协议。在OSPF协议中，描述链路状态及路由信息的报文称为LSA；在IS-IS协议中，描述链路状态及路由信息的报文称为LSP（Link State PDU，或Link State Packet）。注意，LSP 也有两种类型或级别：Level-1 LSP和Level-2 LSP。

实验目的

- 理解IS-IS 协议中路由器级别和接口级别的含义及关系
- 掌握修改IS-IS 路由器级别的方法
- 掌握修改IS-IS 路由器接口级别的方法
- 掌握查看IS-IS 邻接关系的方法

实验内容

实验拓扑如图4-2所示，实验编址如表4-2所示。本实验模拟了一个企业网络场景，R1、R2、R3为公司部门A的路由器，R5和R6为公司部门B的路由器，R4为连接公司部门A和部门B的骨干路由器，全网运行IS-IS。R4属于区域20，部门A属于区域10，部门B属于区域30。R1 和R2的Loopback 0 接口模拟了部门A 的内部网络，R6的Loopback 0接口模拟了部门B 的内部网络。网络需求是：全网互通，并需要通过修改路由器的级别以及接口级别来减少路由器的资源开销及减少网络中不必要的流量，实现优化整个网络的目的。

实验拓扑

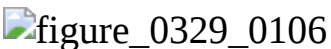
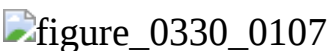
figure_0329_0106

图4-2 IS-IS邻接关系

实验编址表

表4-2 实验编址

figure_0330_0107

实验步骤

1．基本配置

根据图4-2和表4-2进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=20 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/20/20 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置IS-IS路由协议并查看IS-IS邻接关系

在每台路由器上进行IS-IS协议的基本配置。

```
[R1]isis 1
```

```
[R1-isis-1]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-1]is-name R1
```

```
[R1-isis-1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable
```

```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]isis enable
```

[R1-GigabitEthernet0/0/1]interface LoopBack 0

[R1-LoopBack0]isis enable

[R2]isis 1

[R2-isis-1]network-entity 10.0000.0000.0002.00

[R2-isis-1]is-name R2

[R2-isis-1]interface GigabitEthernet 0/0/0

[R2-GigabitEthernet0/0/0]isis enable

[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/2

[R2-GigabitEthernet0/0/2]isis enable

[R2-GigabitEthernet0/0/2]interface LoopBack 0

[R2-LoopBack0]isis enable

[R3]isis 1

[R3-isis-1]network-entity 10.0000.0000.0003.00

[R3-isis-1]is-name R3

[R3-isis-1]interface GigabitEthernet 0/0/0

[R3-GigabitEthernet0/0/0]isis enable

[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R3-GigabitEthernet0/0/1]isis enable

[R3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2

[R3-GigabitEthernet0/0/2]isis enable

[R4]isis 1

[R4-isis-1]network-entity 20.0000.0000.0004.00

[R4-isis-1]is-name R4

[R4-isis-1]interface GigabitEthernet 0/0/0

[R4-GigabitEthernet0/0/0]isis enable

[R4-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R4-GigabitEthernet0/0/1]isis enable

[R5]isis 1

[R5-isis-1]network-entity 30.0000.0000.0005.00

[R5-isis-1]is-name R5

[R5-isis-1]interface GigabitEthernet 0/0/0

[R5-GigabitEthernet0/0/0]isis enable

[R5-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R5-GigabitEthernet0/0/1]isis enable

[R6]isis 1

[R6-isis-1]network-entity 30.0000.0000.0006.00

[R6-isis-1]is-name R6

[R6-isis-1]interface GigabitEthernet 0/0/0

[R6-GigabitEthernet0/0/0]isis enable

[R6-GigabitEthernet0/0/0]interface LoopBack 0

[R6-LoopBack0]isis enable

配置完成后，在R1上测试R1的Loopback 0接口（10.0.1.1）与R6的Loopback 0接口（10.0.6.6）之间的连通性。

<R1>ping -c 1 -a 10.0.1.1 10.0.6.6

PING 10.0.6.6: 56 data bytes, press CTRL_C to break

Reply from 10.0.6.6: bytes=56 Sequence=1 ttl=252 time=20 ms

--- 10.0.6.6 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 20/20/20 ms

可以看到，部门A的内部网络已经可以与部门B的内部网络进行通信了，全网已经实现了互通。在R3上查看IS-IS邻居信息。

<R3>display isis peer

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R4	GE0/0/0	R4.01	Up	7s	L2(L1L2)	64
R1	GE0/0/1	R3.02	Up	22s	L1(L1L2)	64
R1	GE0/0/1	R3.02	Up	22s	L2(L1L2)	64

R2 GE0/0/2 R2.02 Up 9s L1(L1L2) 64

R2 GE0/0/2 R2.02 Up 9s L2(L1L2) 64

Total Peer(s): 5

可以看到，R3与R1既建立了Level-1邻接关系，又建立了Level-2邻接关系；R3与R2既建立了Level-1邻接关系，又建立了Level-2邻接关系；R3与R4只建立了Level-2邻接关系。

由于在默认情况下IS-IS路由器都是Level-1-2路由器，且R1、R2、R3同属于区域10，所以R3能够与R1和R2建立Level-1邻接关系，也能够建立Level-2邻接关系。由于R4的区域ID为20，与R3的区域ID不同，所以R3无法与R4建立Level-1邻接关系，但可以建立Level-2邻接关系。

3．修改IS-IS路由器的级别

在R1 上使用display isis lsdb命令查看IS-IS 协议的链路状态数据库（LSDB）。

<R1>display isis lsdb

Database information for ISIS(1)

Level-1 Link State Database

LSPID ATT/P/OL	Seq Num	Checksum	Holdtime	Length

R1.00-00* 1/0/0	0x0000000e	0xdd1b	1112	115
R2.00-00 1/0/0	0x0000000f	0xd00c	1111	115

R2.01-00	0x00000004	0x9ee8	1095	55	0/0/0
R2.02-00 0/0/0	0x00000001	0xd5b1	1094	55	
R3.00-00 1/0/0	0x00000013	0x3d4a	1113	115	
R3.02-00	0x00000001	0xa6e0	1102	55	0/0/0

Total LSP(s): 6

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	
ATT/P/OL					

R1.00-00*	0x0000000c	0x5d6f	1117	151	0/0/0
R2.00-00	0x0000000e	0x506b	1116	151	0/0/0
R2.01-00	0x00000003	0xa0e7	1095	55	0/0/0
R2.02-00	0x00000001	0xd5b1	1085	55	0/0/0
R3.00-00	0x0000001e	0xf3f3	1118	162	0/0/0
R3.02-00	0x00000001	0xa6e0	1098	55	0/0/0
R4.00-00	0x0000000d	0x68b7	1115	99	0/0/0
R4.01-00	0x00000003	0xea97	1109	55	0/0/0

R4.02-00	0x00000001	0x2061	1109	55	0/0/0
R5.00-00	0x0000000f	0x5cd2	1114	111	0/0/0
R5.01-00	0x00000001	0x4c33	1114	55	0/0/0
R6.00-00	0x00000009	0x40b4	1120	100	0/0/0

Total LSP(s): 12

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R1同时为Level-1和Level-2分别维护了一个LSDB。Level-1的LSDB中有R1所属区域的LSP，Level-2的LSDB中不仅有R1所属区域的LSP，还有其他区域的LSP。

在R1 上使用display isis route 命令查看IS-IS 路由表。

<R1>display isis route

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

0.0.0.0/0	10	NULL	GE0/0/1	10.0.13.3	A/-/-/-
10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/L/-

GE0/0/1 10.0.13.3

10.0.13.0/24 10 NULL GE0/0/1 Direct D/-/L/-

10.0.2.2/32 10 NULL GE0/0/0 10.0.12.2 A/-/L/-

10.0.12.0/24 10 NULL GE0/0/0 Direct D/-/L/-

10.0.1.1/32 0 NULL Loop0 Direct D/-/L/-

10.0.34.0/24 20 NULL GE0/0/1 10.0.13.3 A/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

ISIS(1) Level-2 Forwarding Table

IPV4 Destination IntCost ExtCost ExitInterface NextHop Flags

10.0.6.6/32 40 NULL GE0/0/1 10.0.13.3 A/-/-/-

10.0.23.0/24 20 NULL

10.0.13.0/24 10 NULL GE0/0/1 Direct D/-/L/-

10.0.2.2/32 10 NULL

10.0.56.0/24 40 NULL GE0/0/1 10.0.13.3 A/-/-/-

10.0.12.0/24 10 NULL GE0/0/0 Direct D/-/L/-

10.0.45.0/24 30 NULL GE0/0/1 10.0.13.3 A/-/-/-

10.0.1.1/32 0 NULL Loop0 Direct D/-/L/-

10.0.34.0/24 20 NULL

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R1同时为Level-1和Level-2分别维护了一张IS-IS路由表，其中Level-2路由表中非本区域路由的下一跳均为R3（10.0.13.3）。

在R1 上使用display ip routing-table 命令查看路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 18

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0 0	D		127.0.0.1 LoopBack0
10.0.2.2/32 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.2
10.0.6.6/32 GigabitEthernet0/0/1	ISIS-L2	15	40	D	10.0.13.3
10.0.12.0/24	Direct	0 0	D		10.0.12.1 GigabitEthernet0/0/0
10.0.12.1/32	Direct	0 0	D		127.0.0.1 GigabitEthernet0/0/0
10.0.12.255/32 GigabitEthernet0/0/0	Direct	0 0	D		127.0.0.1

```

10.0.13.0/24    Direct    0  0  D    10.0.13.1  GigabitEthernet0/0/1
10.0.13.1/32    Direct    0  0  D    127.0.0.1  GigabitEthernet0/0/1
10.0.13.255/32  Direct    0  0  D    127.0.0.1
GigabitEthernet0/0/1

10.0.23.0/24    ISIS-L1   15  20  D    10.0.12.2
GigabitEthernet0/0/0

ISIS-L1   15  20  D    10.0.13.3  GigabitEthernet0/0/1

10.0.34.0/24    ISIS-L1   15  20  D    10.0.13.3
GigabitEthernet0/0/1

10.0.45.0/24    ISIS-L2   15  30  D    10.0.13.3
GigabitEthernet0/0/1

10.0.56.0/24    ISIS-L2   15  40  D    10.0.13.3
GigabitEthernet0/0/1

127.0.0.0/8     Direct    0  0  D    127.0.0.1   InLoopBack0
127.0.0.1/32    Direct    0  0  D    127.0.0.1   InLoopBack0
127.255.255.255/32 Direct    0  0  D    127.0.0.1   InLoopBack0
255.255.255.255/32 Direct    0  0  D    127.0.0.1   InLoopBack0

```

可以看到，R1 去往所属区域的其他目标网络的路由均是由IS-IS Level-1 路由表提供的，而去往非R1 所属区域的网络的路由，则是由IS-IS Level-2 路由表提供的，且去往这些目标网络的路由的下一跳均为R3（10.0.13.3）。也就是说，R1去往其他区域的各个网络的路由是可以得到简化的，即利用缺省路由来代替，R1完全没有必要为Level-2单独维护一个LSDB和一张IS-IS路由表。因此，可以在R1上的IS-IS进程视图下使用is-level level-1命令将R1修改为Level-1路由器，从而让R1停止为Level-2维护LSDB和路由表，实现减少系统开销并优化网络的目的。

[R1]isis

[R1-isis-1]is-level level-1

配置完成后，系统会输出日志，提示由于R1的IS-IS Level 发生了改变，IS-IS 模块失效，邻接关系断开。然后IS-IS模块又重新恢复工作，并重新与R2和R3建立起Level-1邻接关系。

[R1-isis-1]

Aug 11 2013 23:09:35-05:1info: IS Level Changed, Reset3

R1 %%01ISIS/4/START_DISABLE_ISIS(l)[0]:ISIS 256 disabled all ISIS modules. ting ISIS...

[R1-isis-1]

Aug 11 2013 23:09:35-05:13 R1

%%01ISIS/4/PEER_DWN_SYS_DISABLE(l)[1]:ISIS 256 neighbor 0000.0000.0002 was Down on interface GE0/0/0 because ISIS was disabled. The Hello packet was received at 23:07:20 last time; the maximum interval for sending Hello packets was 3657891840; the local router sent 1980628992 Hello packets and received 134217728 packets; the type of the Hello packet was Lan Level-1.

[R1-isis-1]

Aug 11 2013 23:09:35-05:13 R1

%%01ISIS/4/PEER_DWN_SYS_DISABLE(l)[2]:ISIS 256 neighbor 0000.0000.0002 was Down on interface GE0/0/0 because ISIS was disabled. The Hello packet was received at 23:07:44 last time; the maximum interval for sending Hello packets was 3657891840; the local router sent 2148401152 Hello packets and received 134217728 packets; the type of the Hello packet was Lan Level-2.

[R1-isis-1]

Aug 11 2013 23:09:35-05:13 R1

%%01ISIS/4/PEER_DWN_SYS_DISABLE(l)[3]:ISIS 256 neighbor

0000.0000.0003 was Down on interface GE0/0/1 because ISIS was disabled. The Hello packet was received at 23:07:01 last time; the maximum interval for sending Hello packets was 3657891840; the local router sent 1980628992 Hello packets and received 134217728 packets; the type of the Hello packet was Lan Level-1.

[R1-isis-1]

Aug 11 2013 23:09:35-05:13 R1

%%01ISIS/4/PEER_DWN_SYS_DISABLE(l)[4]:ISIS 256 neighbor 0000.0000.0003 was Down on interface GE0/0/1 because ISIS was disabled. The Hello packet was received at 23:07:18 last time; the maximum interval for sending Hello packets was 3657891840; the local router sent 1745747968 Hello packets and received 134217728 packets; the type of the Hello packet was Lan Level-2.

[R1-isis-1]

Aug 11 2013 23:09:35-05:13 R1 %%01ISIS/4/START_ENABLE_ISIS(l)[5]:ISIS 256 enabled all ISIS modules.

[R1-isis-1]

Aug 11 2013 23:09:35-05:13 R1 %%01ISIS/4/ADJ_CHANGE_LEVEL(l)[6]:The neighbor of ISIS was changed. (IsisProcessId=256, Neighbor=0000.0000.0002, InterfaceName=GE0/0/0, CurrentState=up, ChangeType=NEW_L1_ADJ, Level=Level-1)

[R1-isis-1]

Aug 11 2013 23:09:35-05:13 R1 %%01ISIS/4/ADJ_CHANGE_LEVEL(l)[7]:The neighbor of ISIS was changed. (IsisProcessId=256, Neighbor=0000.0000.0003, InterfaceName=GE0/0/1, CurrentState=up, ChangeType=NEW_L1_ADJ, Level=Level-1)

在R1上查看IS-IS协议的LSDB。

[R1]display isis lsdb

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R1.00-00*	0x00000011	0xe31c	1194	115	0/0/0
R2.00-00	0x00000011	0xcc0e	1176	115	1/0/0
R2.01-00	0x00000005	0x9ce9	1174	55	0/0/0
R2.02-00	0x00000001	0xd5b1	983	55	0/0/0
R3.00-00	0x00000014	0x3b4b	1173	115	1/0/0
R3.02-00	0x00000002	0xa4e1	1172	55	0/0/0

Total LSP(s): 6

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到 , R1现在只为Level-1维护了LSDB。

在R1上查看IS-IS路由表。

[R1]display isis route

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

```

-----
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
-----
-----

```

```

0.0.0.0/0        10      NULL    GE0/0/0        10.0.12.2  A/-/-/-
GE0/0/1          10.0.13.3
10.0.23.0/24     20      NULL    GE0/0/0        10.0.12.2  A/-/-/-
GE0/0/1          10.0.13.3
10.0.13.0/24     10      NULL    GE0/0/1        Direct    D-/L/-
10.0.2.2/32      10      NULL    GE0/0/0        10.0.12.2  A/-/-/-
10.0.12.0/24     10      NULL    GE0/0/0        Direct    D-/L/-
10.0.1.1/32      0       NULL    Loop0          Direct    D-/L/-
10.0.34.0/24     20      NULL    GE0/0/1        10.0.13.3  A/-/-/-

```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R1现在只为Level-1维护了路由表，且缺省路由的下一跳指向了10.0.12.2与10.0.13.3。

在R1上查看IP路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```

-----
-----

```

Routing Tables: Public

Destinations : 15 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.2
ISIS-L1 15 10 D	10.0.13.3	GigabitEthernet0/0/1			
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.2.2/32 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.2
10.0.12.0/24	Direct	0	0	D	10.0.12.1 GigabitEthernet0/0/0
10.0.12.1/32	Direct	0	0	D	127.0.0.1 GigabitEthernet0/0/0
10.0.12.255/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.13.0/24	Direct	0	0	D	10.0.13.1 GigabitEthernet0/0/1
10.0.13.1/32	Direct	0	0	D	127.0.0.1 GigabitEthernet0/0/1
10.0.13.255/32 GigabitEthernet0/0/1	Direct	0	0	D	127.0.0.1
10.0.23.0/24 GigabitEthernet0/0/0	ISIS-L1	15	20	D	10.0.12.2
ISIS-L1 15 20 D	10.0.13.3	GigabitEthernet0/0/1			
10.0.34.0/24 GigabitEthernet0/0/1	ISIS-L1	15	20	D	10.0.13.3
127.0.0.0/8	Direct	0	0	D	127.0.0.1 InLoopBack0

```

127.0.0.1/32    Direct    0   0   D    127.0.0.1    InLoopBack0
127.255.255.255/32 Direct    0   0   D    127.0.0.1    InLoopBack0
255.255.255.255/32 Direct    0   0   D    127.0.0.1    InLoopBack0

```

可以看到，R1的路由表中只存在用于访问R1所在IS-IS区域内的网络路由和用于访问其他 IS-IS 区域的网络的缺省路由，缺省路由下一跳为 R2（10.0.12.2）与 R3（10.0.13.3）。由于访问其他IS-IS区域的网络的流量是经区域内的Level-1-2路由器出去的，所以可以将Level-1-2路由器R2也修改为Level-1路由器，避免R2向区域内下发缺省路由以至于可能使去往其他IS-IS区域的流量先到达R2，而不是先到达R3。这样，网络就可以得到进一步的优化。

```
[R2]isis
```

```
[R2-isis-1]is-level level-1
```

在R1和R2上查看缺省路由。

```
[R1]display ip routing-table 0.0.0.0
```

Route Flags: R - relay, D - download to fib

```
-----
Routing Table : Public

```

```
Summary Count : 1
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
0.0.0.0/0	ISIS-L1	15	10	D	10.0.13.3	GigabitEthernet0/0/1

```
[R2]display ip routing-table 0.0.0.0
```

Route Flags: R - relay, D - download to fib

Routing Table : Public

Summary Count : 1

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
0.0.0.0/0	ISIS-L1	15	10	D	10.0.23.3	GigabitEthernet0/0/2

观察发现，R1和R2的缺省路由的下一跳都只指向了R3。

最后，将R4配置为Level-2路由器，将R6配置为Level-1路由器。

[R4]isis

[R4-isis-1]is-level level-2

[R6]isis

[R6-isis-1]is-level level-1

这样一来，各路由器上就不会再维护没有必要的LSDB和IS-IS路由表了。

4．修改IS-IS路由器接口的级别

目前，虽然各路由器上不会再维护没有必要的LSDB与IS-IS路由表了，设备开销得到了节省，但是，在许多链路上还存在优化的空间。

在R3 上使用debugging isis adjacency interface GigabitEthernet 0/0/0命令针对R3的GE 0/0/0接口启用IS-IS邻接关系的调试工具。在获得调试输出后使用undo debugging all命令关闭调试工具。

<R3>debugging isis adjacency interface GigabitEthernet0/0/0

<R3>terminal debugging

Info: Current terminal debugging is on.

<R3>

Aug 11 2013 22:10:51.593.1-05:13 R3 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2679)

<R3>

Aug 11 2013 22:10:51.593.2-05:13 R3 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

<R3>

Aug 11 2013 22:10:56.773.1-05:13 R3 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-2 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2731)

<R3>

Aug 11 2013 22:10:56.773.2-05:13 R3 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L2 Hello on GE0/0/0, to SNPA 0180.c200.0015.
(IS15_6963)

<R3>undo debugging all

Info: All possible debugging has been turned off

可以看到，R3和R4由于区域ID不同，所以仅建立了Level-2邻接关系。
但是，R3的GE 0/0/0 接口依旧在同时发送Level-1 和Level-2 的Hello 消

息以尝试建立Level-1 和Level-2邻接关系。为了让该接口不再发送Level-1的Hello消息以减小链路系统与系统开销，可以在R3的GE 0/0/0接口视图下，使用命令isis circuit-level level-2修改GE 0/0/0 接口的IS-IS级别为Level-2。

```
[R3-GigabitEthernet0/0/0]isis circuit-level level-2
```

配置此命令后，R3 的GE 0/0/0 接口的链路会立刻进入Down状态，然后恢复。系统会输出日志信息，显示R3将重新与R4建立Level-2邻接关系。

```
Aug 11 2013 22:20:58-05:13 R3 %%01ISIS/4/ADJ_CHANGE_LEVEL(l)
[0]:The neighbor of ISIS was changed. (IsisProcessId=256,
Neighbor=0000.0000.0004, InterfaceName=GE0/0/0, CurrentState=down,
ChangeType=L2_CIRCUIT_DOWN, Level=Level-2)
```

```
[R3-GigabitEthernet0/0/0]
```

```
Aug 11 2013 22:20:58-05:13 R3
%%01ISIS/4/PEER_DOWN_CIRC_DOWN(l)[1]:ISIS 256 neighbor
0000.0000.0004 was Down because interface GE0/0/0 was down. The Hello
packet was received at 22:20:55 last time; the maximum interval for sending
Hello packets was 438763520; the local router sent 2583691264 Hello
packets and received 150994944 packets; the type of the Hello packet was
Lan Level-2.
```

```
[R3-GigabitEthernet0/0/0]
```

```
Aug 11 2013 22:20:58-05:13 R3 %%01ISIS/4/ADJ_CHANGE_LEVEL(l)
[2]:The neighbor of ISIS was changed. (IsisProcessId=256,
Neighbor=0000.0000.0004, InterfaceName=GE0/0/0, CurrentState=up,
ChangeType=NEW_L2_ADJ, Level=Level-2)
```

在R3 上针对GE 0/0/0 接口重新启用IS-IS 邻接关系的调试工具。

```
<R3>debugging isis adjacency interface GigabitEthernet 0/0/0
```

```
<R3>terminal debugging
```


Info: Current terminal debugging is on.

<R3>

Aug 11 2013 22:26:31.673.1-05:13 R3 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-2 IIH encode cache to send IIH, GE0/0/0.
(IS15_2731)

<R3>

Aug 11 2013 22:26:31.673.2-05:13 R3 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L2 Hello on GE0/0/0, to SNPA 0180.c200.0015.
(IS15_6963)

<R3>undo debugging all

Info: All possible debugging has been turned off

可以看到，R3 的GE 0/0/0 接口不再发送Level-1 的Hello 消息了。

在R3的GE 0/0/1 接口和GE 0/0/2 接口，以及R5的GE 0/0/0接口和GE 0/0/1 接口完成类似的配置。

```
[R3]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]isis circuit-level level-1
```

```
[R3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[R3-GigabitEthernet0/0/2]isis circuit-level level-1
```

```
[R5]interface GigabitEthernet 0/0/0
```

```
[R5-GigabitEthernet0/0/0]isis circuit-level level-1
```

```
[R5-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R5-GigabitEthernet0/0/1]isis circuit-level level-2
```

至此，Level-1-2路由器R3和R5不会再发送没有必要的Hello消息，减少了链路带宽的消耗，进一步优化了网络。

在R1上测试10.0.1.1/32与10.0.6.6/32之间的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.6.6
```

```
PING 10.0.6.6: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.6.6: bytes=56 Sequence=1 ttl=252 time=30 ms
```

```
--- 10.0.6.6 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 30/30/30 ms
```

可以看到，网络经过优化之后，网络的互通性未受到任何影响。

思考

IS-IS中的Level-1路由器、Level-2路由器、Level-1-2路由器分别类似于OSPF中的哪种路由器？

4.3 IS-IS 链路状态数据库

原理概述

一个OSPF链路状态数据库是若干条LSA的集合。与此相似，一个IS-IS链路状态数据库是若干条LSP的集合。与OSPF链路状态数据库不同，IS-IS链路状态数据库有Level-1和Level-2之分。

在IS-IS协议中，每一条LSP都有一个剩余生存时间、一个序列号和一个校验和。LSP的剩余生存时间是由最大生存时间（默认为1200s）开始逐渐递减的。当一条LSP的剩余生存时间递减至0时，仍然会在链路状态数据库中继续保留60s（称为ZeroAgeLifetime），然后才会被删除。LSP的始发路由器会周期性地刷新LSP，刷新时间间隔为900s减去不超过25%的随机量。

LSP的序列号是一个32bit的整数，初始值为1，每次刷新时都会递增1。与OSPF中的LSA一样，同一条LSP，其序列号越大，表示该LSP越新，路由器总是将最新的LSP放入其链路状态数据库中。如果序列号递增至最大值时，则无法被继续刷新，但其剩余生存时间会递减至0，然后会被从链路状态数据库中删除。

LSP的校验和用于检验LSP是否在传输过程中受到损坏。当路由器收到一条包含错误的校验和的LSP时，会将其直接丢弃。

实验目的

- 理解IS-IS 链路状态数据库的内容
- 掌握查看IS-IS 链路状态数据库的方法

实验内容

实验拓扑如图4-3所示，实验编址如表4-3所示。本实验模拟了一个简单的企业网络场景，Level-1路由器R1和Level-1-2路由器R2为公司部门A的网络设备，Level-2路由器R3为公司的骨干路由器。整个网络都运行IS-IS协议，R1和R2属于IS-IS区域10，R3属于IS-IS区域20，R1的Loopback 0接口模拟了部门A的内部网络，R3的Loopback 0接口模拟了公司服务器所在的网络。实验内容的重点是观察和分析R1、R2、R3上的IS-IS链路状态数据库。

实验拓扑

 figure_0339_0108

图4-3 IS-IS链路状态数据库

实验编址表

表4-3 实验编址

figure_0340_0109

实验步骤

1．基本配置

根据图4-3和表4-3进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=30 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 30/30/30 ms
```

R2与R3之间的连通性测试过程在此省略。

2．配置IS-IS路由协议

在R1、R2、R3上配置IS-IS协议，其中R1为Level-1路由器，R2为Level-1-2路由器，R3为Level-2路由器。

[R1]isis

[R1-isis-1]is-level level-1

[R1-isis-1]is-name R1

[R1-isis-1]network-entity 10.0000.0000.0001.00

[R1-isis-1]interface LoopBack 0

[R1-LoopBack0]isis enable

[R1-LoopBack0]interface GigabitEthernet 0/0/0

[R1-GigabitEthernet0/0/0]isis enable

[R2]isis

[R2-isis-1]is-level level-1-2

[R2-isis-1]is-name R2

[R2-isis-1]network-entity 10.0000.0000.0002.00

[R2-isis-1]interface GigabitEthernet 0/0/0

[R2-GigabitEthernet0/0/0]isis enable

[R2-GigabitEthernet0/0/0]isis circuit-level level-1

[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R2-GigabitEthernet0/0/1]isis enable

[R2-GigabitEthernet0/0/1]isis circuit-level level-2

[R3]isis

[R3-isis-1]is-name R3

```
[R3-isis-1]is-level level-2
```

```
[R3-isis-1]network-entity 20.0000.0000.0003.00
```

```
[R3-isis-1]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]isis enable
```

```
[R3-GigabitEthernet0/0/1]interface LoopBack 0
```

```
[R3-LoopBack0]isis enable
```

配置完成后，在R2上查看IS-IS邻居信息。

```
[R2]display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	R2.01	Up	20s	L1	64

R3	GE0/0/1	R2.02	Up	25s	L2	64

Total Peer(s): 2

可以看到，R2与R1建立了Level-1邻接关系，与R3建立了Level-2邻接关系。

在R1上以10.0.1.1为源，使用ping命令测试与10.0.3.3之间的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.3.3
```

```
PING 10.0.3.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.3.3: bytes=56 Sequence=1 ttl=254 time=30 ms
```

--- 10.0.3.3 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 30/30/30 ms

可以看到，部门A的内部网络与服务器所在网络之间的通信是正常的，全网实现了互通。

3．查看Level-1路由器的链路状态数据库

在Level-1路由器R1上查看IS-IS链路状态数据库。

[R1]display isis lsdb

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R1.00-00*	0x00000005	0x353d	571	88	0/0/0

R2.00-00	0x00000007	0x36af	784	76	1/0/0

R2.01-00	0x00000001	0xa4e5	590	55	0/0/0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以观察到，链路状态数据库中包含了3 条LSP，以及相应的LSP ID、序列号（Seq Num）、校验和（Checksum）、生存时间（Holdtime）、长度（Length）等属性。因为 R1是Level-1路由器，所以它只为Level-1维护了一个链路状态数据库。第一条LSP的LSP ID为R1.00-00*，R1.00-00*中的R1为动态主机名。如果没有配置动态主机名时，相应的位置就是系统ID。R1.00-00*中前面的00是伪节点标识，00表示此LSP是由真实节点而非伪节点生成的。R1.00-00*中后面的00为分片号，当LSP的长度太长时，LSP会被分片，分片号的作用是为了重组被分片的LSP。R1.00-00*中的“*”表示此LSP是于本地生成的。

在R1 上使用display isis lsdb verbose 命令查看IS-IS 链路状态数据库的详细信息。

```
[R1]display isis lsdb verbose
```

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

0000.0000.0001.00-00*	0x00000006	0x333e	1128	88	0/0/0
SOURCE	R1.00				
HOST NAME	R1				
NLPID	IPv4				
AREA ADDR	10				

INTF ADDR 10.0.1.1

INTF ADDR 10.0.12.1

NBR ID R2.01 COST: 10

IP-Internal 10.0.1.1 255.255.255.255 COST: 0

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

0000.0000.0002.00-00 0x00000008 0x34b0 1179 76
1/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPv4

AREA ADDR 10

INTF ADDR 10.0.12.2

INTF ADDR 10.0.23.2

NBR ID R2.01 COST: 10

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

0000.0000.0002.01-00 0x00000002 0xa2e6 1179 55
0/0/0

SOURCE R2.01

NLPID IPv4

NBR ID R2.00 COST: 0

NBR ID R1.00 COST: 0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，第一条LSP 是本地生成的，LSP ID 中包含了系统ID，系统是一个真实节点而非伪节点。Source为动态主机名附伪节点标识，Host Name 为动态主机名，NLP ID为该LSP 所支持的网络协议，此处为IPv4，表明此LSP 工作在IPv4 网络中。Area Addr为该LSP 的区域地址（即区域ID），此处为10。INTF Addr为接口地址，描述了生成此LSP 的路由器所拥有的接口的IP 地址，此处为10.0.1.1 和10.0.12.1。NBR ID 为邻居的系统ID附伪节点标识，Cost为去往邻居的开销值。IP-Internal 为区域内IP路由信息，描述网络前缀和掩码，以及Cost信息。需要注意的是，第三条LSP是伪节点产生的。

4．查看Level-1-2路由器的链路状态数据库

在Level-1-2路由器R2上查看IS-IS链路状态数据库。

```
[R2]display isis lsdb
```

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	

ATT/P/OL

R1.00-00	0x00000006	0x333e	913	88	0/0/0
R2.00-00*	0x00000008	0x34b0	966	76	1/0/0

R2.01-00* 0x00000002 0xa2e6 966 55 0/0/0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R2.00-00*	0x00000009	0x7717	966	100	0/0/0

R2.02-00*	0x00000002	0xd3b2	965	55	0/0/0

R3.00-00	0x00000004	0xbb80	405	88	0/0/0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R2为Level-1和Level-2分别维护了一份链路状态数据库，另外，其中的Level-1链路状态数据库中的LSP条目与Level-1路由器R1的链路状态数据库中的LSP条目完全相同，这表明Level-1链路状态数据库在R1和R2上完成了同步。

在R2 上使用命令display isis lsdb level-1 verbose查看Level-1 链路状态数据库的详细信息。

[R2]display isis lsdb level-1 verbose

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

0000.0000.0001.00-00	0x00000006	0x333e	807	88	0/0/0

SOURCE R1.00

HOST NAME R1

NLPID IPv4

AREA ADDR 10

INTF ADDR 10.0.1.1

INTF ADDR 10.0.12.1

NBR	ID	R2.01	COST: 10

IP-Internal	10.0.1.1	255.255.255.255	COST: 0

IP-Internal	10.0.12.0	255.255.255.0	COST: 10

0000.0000.0002.00-00*	0x00000008	0x34b0	60	76	1/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPv4

AREA ADDR 10

INTF ADDR 10.0.12.2

INTF ADDR 10.0.23.2

NBR ID R2.01 COST: 10

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

0000.0000.0002.01-00* 0x00000002 0xa2e6 860 55
0/0/0

SOURCE R2.01

NLPID IPv4

NBR ID R2.00 COST: 0

NBR ID R1.00 COST: 0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

观察发现，在R2的Level-1链路状态数据库中，除了用于标识本地生成的LSP的“*”之外，内容上与R1的Level-1链路状态数据库完全相同。

在R2 上使用命令display isis lsdb level-2 verbose查看Level-2 链路状态数据库的详细信息。

[R2]display isis lsdb level-2 verbose

Database information for ISIS(1)

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

0000.0000.0002.00-00* 0x00000009 0x7717 531 100 0/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPv4

AREA ADDR 10

INTF ADDR 10.0.12.2

INTF ADDR 10.0.23.2

NBR ID R2.02 COST: 10

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

IP-Internal 10.0.1.1 255.255.255.255 COST: 10

IP-Internal 10.0.23.0 255.255.255.0 COST: 10

0000.0000.0002.02-00* 0x00000002 0xd3b2 531 55 0/0/0

SOURCE R2.02

NLPID IPv4

NBR ID R2.00 COST: 0

NBR ID R3.00 COST: 0

0000.0000.0003.00-00 0x00000005 0xb981 19 88 0/0/0

SOURCE R3.00

HOST NAME R3

```

NLPID          IPv4
AREA ADDR      20
INTF ADDR      10.0.23.3
INTF ADDR      10.0.3.3
NBR   ID       R2.02                COST: 10
IP-Internal    10.0.23.0  255.255.255.0    COST: 10
IP-Internal    10.0.3.3  255.255.255.255    COST: 0

```

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

观察发现，在Level-2链路状态数据库中，LSP的格式与在Level-1链路状态数据库中并没有区别。Level-1链路状态数据库与Level-2链路状态数据库的最主要区别在于：Level-1链路状态数据库中的LSP的区域ID彼此都相同，而Level-2链路状态数据库中的LSP的区域ID彼此可以不同。

5．查看Level-2路由器的链路状态数据库

在Level-2路由器R3上查看IS-IS链路状态数据库。

```
[R3]display isis lsdb
```

```
Database information for ISIS(1)
```

```
-----
```

```
Level-2 Link State Database
```

LSPID ATT/P/OL	Seq Num	Checksum	Holdtime	Length	

R2.00-00	0x0000000a	0x7518	1162	100	0/0/0
R2.02-00	0x00000003	0xd1b3	1162	55	0/0/0
R3.00-00*	0x00000005	0xb981	670	88	0/0/0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，Level-2路由器R3只为Level-2维护了一份链路状态数据库，其中的LSP 条目与R2中的Level-2链路状态数据库中的LSP条目相同。

在R3上查看IS-IS链路状态数据库的详细信息。

[R3]display isis lsdb verbose

Database information for ISIS(1)

Level-2 Link State Database					
LSPID ATT/P/OL	Seq Num	Checksum	Holdtime	Length	

0000.0000.0002.00-00	0x0000000a	0x7518	1088	100	0/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPv4

AREA ADDR 10

INTF ADDR 10.0.12.2

INTF ADDR 10.0.23.2

NBR ID R2.02 COST: 10

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

IP-Internal 10.0.1.1 255.255.255.255 COST: 10

IP-Internal 10.0.23.0 255.255.255.0 COST: 10

0000.0000.0002.02-00 0x00000003 0xd1b3 1088 55
0/0/0

SOURCE R2.02

NLPID IPv4

NBR ID R2.00 COST: 0

NBR ID R3.00 COST: 0

0000.0000.0003.00-00* 0x00000005 0xb981 596 88
0/0/0

SOURCE R3.00

HOST NAME R3

NLPID IPv4

AREA ADDR 20

INTF ADDR 10.0.23.3

INTF ADDR 10.0.3.3

NBR ID R2.02 COST: 10

IP-Internal 10.0.23.0 255.255.255.0 COST: 10

IP-Internal 10.0.3.3 255.255.255.255 COST: 0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R3的Level-2链路状态数据库与R2的Level-2链路状态数据库完全相同。

思考

为什么说RIP 是一种基于Road Sign 的路由协议，而IS-IS 是一种基于Road Map 的路由协议？

4.4 IS-IS DIS

原理概述

OSPF协议支持4种网络类型，IS-IS协议只支持两种网络类型，即广播网络和点到点网络。与 OSPF 协议相同，IS-IS 协议在广播网络中会将网络视为一个伪节点（Pseudonode，简称PSN），并选举出一台DIS（Designated IS）路由器来代行这个伪节点的职责。DIS 的作用与OSPF 的DR 类似，可以减少不必要的LSP 泛洪。注意，与OSPF协议中的DR选举不同，DIS的选举是抢占性的。另外，DIS还有Level-1和

Level-2之分，同一网络的Level-1 DIS 和Level-2 DIS可能是同一台路由器，也可能是不同的路由器。在点到点网络中，IS-IS协议不选举DIS。

注意，选举出DIS后，广播网络中的路由器仍然需要与所有的邻居建立邻接关系，而不仅仅是与 DIS 建立邻接关系。在广播网络中，DIS 会周期性（默认为 10s）地发送携带CSNP（Complete Sequence Number PDU）消息的组播帧来实现链路状态数据库之间的同步，其中Level-1DIS使用的组播MAC 地址为0180.C200.0014，Level-2 DIS使用的组播MAC地址为0180.C200.0015。

选举DIS的过程是自动进行的，选举的依据是比较同一网络中路由器接口的DIS优先级，其次是比较接口的MAC 地址。在接口的DIS 优先级相同的情况，MAC 地址较大者将成为DIS。

路由器的 IS-IS 接口都拥有一个 Level-1 DIS 优先级和一个 Level-2 DIS 优先级，取值范围都是 0 到 127，默认值都是 64。IS-IS 接口所发出的 Level-1 Hello 报文中携带了Level-1 DIS优先级的值，Level-2 Hello报文中携带了Level-2 DIS优先级的值。注意，如果DIS优先级的值为0，并不表示不参与DIS的选举，而只是表示DIS优先级最低。

最后需要说明的是，在OSPF 协议中，除了有DR，还有Backup DR（BDR）。但是在IS-IS 协议中，只有DIS，没有Backup DIS。

实验目的

- 理解IS-IS 协议中DIS 的作用和选举方法
- 理解IS-IS 接口的DIS 优先级的概念
- 掌握通过修改DIS 优先级来控制DIS 选举结果的方法

实验内容

实验拓扑如图4-4所示，实验编址如表4-4所示。本实验模拟了一个简单的企业网络场景，R1、R2、R3、R4分别连接着公司部门A、B、C、

D ; R1、 R2、 R3、 R4分别为Level-1-2、 Level-1-2、 Level-1、 Level-2路由器。网络需求是：必须让R1成为Level-1 DIS , R2 成为Level-2 DIS。注意，本实验中路由器接口的MAC 地址是随机生成的，读者自行实验时，应以实际获取的MAC地址为准。

实验拓扑

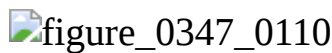
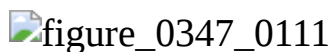
figure_0347_0110

图4-4 IS-IS DIS

实验编址表

表4-4 实验编址

figure_0347_0111

实验步骤

1．基本配置

根据图4-4和表4-4进行相应的基本配置，并使用ping命令检测R1与R4之间的连通性。

```
<R1>ping -c 1 10.0.1.4
```

```
PING 10.0.1.4: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.1.4: bytes=56 Sequence=1 ttl=255 time=90 ms
```

```
--- 10.0.1.4 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

round-trip min/avg/max = 90/90/90 ms

其余路由器之间的连通性测试过程在此省略。

2 . 配置IS-IS路由协议

在每台路由器上配置IS-IS协议，其中R1和R2为Level-1-2路由器，R3为Level-1路由器，R4为Level-2路由器。

```
[R1]isis 1
```

```
[R1-isis-1]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-1]is-name R1
```

```
[R1-isis-1]quit
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable 1
```

```
[R2]isis 1
```

```
[R2-isis-1]network-entity 10.0000.0000.0002.00
```

```
[R2-isis-1]is-name R2
```

```
[R2-isis-1]quit
```

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis enable 1
```

```
[R3]isis 1
```

```
[R3-isis-1]network-entity 10.0000.0000.0003.00
```

```
[R3-isis-1]is-level Level-1
```

```
[R3-isis-1]is-name R3
```

```
[R3-isis-1]quit
```

```
[R3]interface GigabitEthernet 0/0/0
```

```
[R3-GigabitEthernet0/0/0]isis enable 1
```

```
[R4]isis 1
```

```
[R4-isis-1]network-entity 10.0000.0000.0004.00
```

```
[R4-isis-1]is-level Level-2
```

```
[R4-isis-1]is-name R4
```

```
[R4-isis-1]quit
```

```
[R4]interface GigabitEthernet 0/0/0
```

```
[R4-GigabitEthernet0/0/0]isis enable 1
```

配置完成后，在R1上查看IS-IS邻居信息。

```
[R1]display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R2	GE0/0/0	R1.01	Up	28s	L1(L1L2)	64
R3	GE0/0/0	R1.01	Up	24s	L1	64
R2	GE0/0/0	R4.01	Up	24s	L2(L1L2)	64
R4	GE0/0/0	R4.01	Up	7s	L2	64

Total Peer(s): 4

可以看到，R1与相邻路由器的IS-IS邻接关系是正常的，其中R1与R2分别建立了Level-1邻接关系和Level-2邻接关系，R1与R3建立了Level-1邻接关系，R1与R4建立了Level-2邻接关系。

3．查看默认选举的DIS

在每台路由器上使用display isis interface GigabitEthernet 0/0/0 命令查看GE 0/0/0接口的IS-IS协议信息。

```
[R1]display isis interface GigabitEthernet 0/0/0
```

Interface information for ISIS(1)

Interface	Id	IPv4.State	IPv6.State	MTU	Type
DIS					
GE0/0/0	001	Up	Down	1497	L1/L2 Yes/No

```
[R2]display isis interface GigabitEthernet 0/0/0
```

Interface information for ISIS(1)

Interface	Id	IPv4.State	IPv6.State	MTU	Type
DIS					
GE0/0/0	001	Up	Down	1497	L1/L2 No/No

```
[R3]display isis interface GigabitEthernet 0/0/0
```

Interface information for ISIS(1)

Interface	Id	IPv4.State	IPv6.State	MTU	Type
DIS					

```
GE0/0/0    001    Up    Down    1497    L1/L2    No/No
```

```
[R4]display isis interface GigabitEthernet 0/0/0
```

```
Interface information for ISIS(1)
```

```
-----
```

Interface	Id	IPv4.State	IPv6.State	MTU	Type
DIS					

GE0/0/0	001	Up	Down	1497	L1/L2 No/Yes

可以看到，在使用接口的缺省DIS 优先级的情况下，能够发送和接收Level-1 Hello报文的接口中，R1 的GE 0/0/0 接口的MAC 地址最大，因此R1 被选举为Level-1 DIS；能够发送和接收Level-2 Hello 报文的接口中，R4的GE 0/0/0接口的MAC地址最大，因此R4 被选举为Level-2 DIS。

在每台路由器上使用display isis interface GigabitEthernet 0/0/0 verbose命令查看GE 0/0/0 接口的IS-IS 详细信息。

```
[R1]display isis interface GigabitEthernet 0/0/0 verbose
```

```
Interface information for ISIS(1)
```

```
-----
```

Interface	Id	IPv4.State	IPv6.State	MTU	Type
DIS					

GE0/0/0	001	Up	Down	1497	L1/L2 Yes/No

```
Circuit MT State          :Standard
```

```
Description              :HUAWEI,AR  
Series,GigabitEthernet0/0/0 Interface
```

```
SNPA Address              : 00e0-fc03-d86c
```



```

IP Address                : 10.0.1.1
IPv6 Link Local Address   :
IPv6 Global Address(es)   :
Csnp Timer Value          :L1  10      L2  10
Hello Timer Value         :L1  10      L2  10
DISHello Timer Value      :L1  3       L2  3
Hello Multiplier Value    :L1  3       L2  3
LSP-Throttle Timer        :L12      50
Cost                      :L1  10      L2  10
IPv6 Cost                 :L1  10      L2  10
Priority                  :L1  64      L2  64
Retransmit Timer Value    :L12      5
Bandwidth-Value           :Low  1000000000  High  0
Static Bfd                :NO
Dynamic Bfd               :NO
Fast-Sense Rpr            :NO

```

```
[R2]display isis interface GigabitEthernet 0/0/0 verbose
```

```
Interface information for ISIS(1)
```

```
-----
```

Interface	Id	IPv4.State	IPv6.State	MTU	Type
DIS					

GE0/0/0	001	Up	Down	1497	L1/L2	No/No
Circuit MT State			:Standard			
Description			:HUAWEI,AR			
Series,GigabitEthernet0/0/0 Interface						
SNPA Address			: 00e0-fc03-5d75			
IP Address			: 10.0.1.2			
IPv6 Link Local Address			:			
IPv6 Global Address(es)			:			
Csnp Timer Value			:L1	10	L2	10
Hello Timer Value			:L1	10	L2	10
DIS Hello Timer Value			:L1	3	L2	3
Hello Multiplier Value			:L1	3	L2	3
LSP-Throttle Timer			:L12	50		
Cost			:L1	10	L2	10
IPv6 Cost			:L1	10	L2	10
Priority			:L1	64	L2	64
Retransmit Timer Value			:L12	5		
Bandwidth-Value			:Low	1000000000	High	0
Static Bfd			:NO			
Dynamic Bfd			:NO			
Fast-Sense Rpr			:NO			

[R3]display isis interface GigabitEthernet 0/0/0 verbose

Interface information for ISIS(1)

Interface	Id	IPv4.State	IPv6.State	MTU	Type

GE0/0/0	001	Up	Down	1497	L1/L2 No/No

Circuit MT State :Standard

Description :HUAWEI,AR
Series,GigabitEthernet0/0/0 Interface

SNPA Address : 00e0-fc03-a0f9

IP Address : 10.0.1.3

IPv6 Link Local Address :

IPv6 Global Address(es) :

Csnp Timer Value :L1 10 L2 10

Hello Timer Value :L1 10 L2 10

DIS Hello Timer Value :L1 3 L2 3

Hello Multiplier Value :L1 3 L2 3

LSP-Throttle Timer :L12 50

Cost :L1 10 L2 10

IPv6 Cost :L1 10 L2 10

Priority :L1 64 L2 64

Retransmit Timer Value :L12 5

Bandwidth-Value :Low 1000000000 High 0

Static Bfd :NO

Dynamic Bfd :NO

Fast-Sense Rpr :NO

[R4]display isis interface GigabitEthernet 0/0/0 verbose

Interface information for ISIS(1)

Interface	Id	IPv4.State	IPv6.State	MTU	Type
DIS					

GE0/0/0	001	Up	Down	1497	L1/L2	No/Yes

Circuit MT State :Standard

Description :HUAWEI,AR
Series,GigabitEthernet0/0/0 Interface

SNPA Address : 00e0-fc03-d86d

IP Address : 10.0.1.4

IPv6 Link Local Address :

IPv6 Global Address(es) :

Csnp Timer Value :L1 10 L2 10

Hello Timer Value :L1 10 L2 10

DIS Hello Timer Value :L1 3 L2 3

Hello Multiplier Value	:L1 3 L2 3
LSP-Throttle Timer	:L12 50
Cost	:L1 10 L2 10
IPv6 Cost	:L1 10 L2 10
Priority	:L1 64 L2 64
Retransmit Timer Value	:L12 5
Bandwidth-Value	:Low 1000000000 High 0
Static Bfd	:NO
Dynamic Bfd	:NO
Fast-Sense Rpr	:NO

可以看到，R1、R2、R3、R4 的SNPA（Sub-Netowrk Point of Attachment）地址分别为00e0-fc03-d86c、00e0-fc03-5d75、00e0-fc03-a0f9、00e0-fc03-d86d（SNPA地址在这里指的就是MAC地址），而Level-1 DIS优先级和Level-2 DIS优先级的值都为默认值64。R1 被选举为Level-1 DIS，R4 被选举为Level-2 DIS。需要注意的是，R1 和 R2 都是Level-1-2 路由器，它们会使用Level-1 DIS优先级参与到Level-1 的DIS 选举中，同时还会使用Level-2 DIS优先级参与到Level-2 的DIS 选举中。

IS-IS协议会将广播型网络本身抽象成一个伪节点，伪节点并不实际存在，它只是一个逻辑上的概念，广播型网络中的路由器都认为自己和伪节点存在邻接关系，并通过产生相应的LSP来描述自己和这个伪节点之间的链路状态。广播型网络中的DIS充当了伪节点的角色并代行伪节点的职责；DIS 路由器会代替抽象的伪节点产生PSN LSP（伪节点LSP），用以描述哪些路由器与伪节点相连。PSN LSP 与OSPF 中的Type-2 LSA非常相似。

在R1上查看IS-IS的链路状态数据库。

<R1>display isis lsdb

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R1.00-00*	0x0000000f	0xc753	1182	74	1/0/0
R1.01-00*	0x00000005	0x9760	1179	66	0/0/0
R2.00-00	0x0000000c	0xf525	1158	74	1/0/0
R3.00-00	0x0000000a	0x180c	1160	74	0/0/0

Total LSP(s): 5

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R1.00-00*	0x0000000e	0x130e	1182	74	0/0/0
R2.00-00	0x0000000a	0x43de	752	74	0/0/0
R4.00-00	0x00000007	0xa87a	751	72	0/0/0

R4.01-00 0x00000006 0x6290 1160 66 0/0/0

Total LSP(s): 4

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，Level-1 链路状态数据库中有一条LSP ID 为R1.01-00*的LSP，这就是一条Level-1 PSN LSP，同时也说明R1是Level-1 DIS；在Level-2 链路状态数据库中有一条LSP ID为R4.01-00的LSP，这就是一条Level-2 PSN LSP，同时说明R4是Level-2 DIS。

在R1 上使用display isis lsdb is-name R1 verbose 命令查看R1 生成的LSP的详细信息。

<R1>display isis lsdb is-name R1 verbose

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0000.0000.0001.00-00*	0x00000010	0xc554	705	74
1/0/0				

SOURCE R1.00

HOST NAME R1

NLPID IPv4

AREA ADDR 10.0000

INTF ADDR 10.0.1.1

NBR ID R1.01 COST: 10

IP-Internal 10.0.1.0 255.255.255.0 COST: 10

0000.0000.0001.01-00* 0x00000006 0x9561 705 66
0/0/0

SOURCE R1.01

NLPID IPv4

NBR ID R1.00 COST: 0

NBR ID R2.00 COST: 0

NBR ID R3.00 COST: 0

Total LSP(s): 2

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0000.0000.0001.00-00* 0x0000000f 0x110f 705 74
0/0/0

SOURCE R1.00

HOST NAME R1

NLPID IPv4

AREA ADDR 10.0000

INTF ADDR 10.0.1.1

NBR ID R4.01 COST: 10

IP-Internal 10.0.1.0 255.255.255.0 COST: 10

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，在R1的Level-1 的LSDB 中LSP ID 为0000.0000.0001.00-00*的LSP 由R1 自己产生，这条LSP 描述了R1 和伪节点R1.01 之间的链路状态信息，其中Area Addr描述了R1 自己所在的区域ID 为10.0000，INTF Addr 描述了R1 自己与伪节点R1.01 相连的接口IP 地址为10.0.1.1，NBR ID 描述了邻居是伪节点R1.01，Cost 描述了R1 自己到伪节点R1.01的开销值为10，IP-Internal描述了R1和伪节点R1.01之间的网络前缀和掩码以及开销值信息。

伪节点只是一个逻辑上的概念，用来表示一个广播型网络本身，而DIS路由器是连接到这个广播型网络的一台路由器，DIS和伪节点是两个不同的概念，只是DIS代行了伪节点的职责而已。

观察还发现，在R1 的Level-1 LSDB 中还有一条LSP ID 为0000.0000.0001.01-00*的LSP，这条LSP 其实就是由R1 代替伪节点R1.01 产生的一条Level-1 PSN LSP，其中的NBR ID 描述了R1.01 这个伪节点同时连接了R1、R2、R3这3 台路由器，而Cost则说明了R1.01这个伪节点到R1、R2、R3的开销值都为0。注意，在广播网络上，路由器到伪节点的开销值默认为10，而伪节点到路由器的开销值为0。

R2 既不是Level-1 DIS , 也不是Level-2 DIS , 在R2 上使用display isis lsdb is-name R2 verbose 命令查看R2 生成的LSP 的详细信息。

```
<R2>display isis lsdb is-name R2 verbose
```

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0000.0000.0002.00-00*	0x0000000e	0xf127	1151	74

1/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPv4

AREA ADDR 10.0000

INTF ADDR 10.0.1.2

NBR ID R1.01 COST: 10

IP-Internal 10.0.1.0 255.255.255.0 COST: 10

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0000.0000.0002.00-00*	0x0000000e	0x3be2	1151	74

0/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPv4

AREA ADDR 10.0000

INTF ADDR 10.0.1.2

NBR	ID	R4.01	COST: 10

IP-Internal	10.0.1.0	255.255.255.0	COST: 10

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R2生成了两条LSP，第一条描述了自己与伪节点R1.01的关系，第二条描述了自己与伪节点R4.01的关系。

4．修改DIS优先级来控制DIS选举结果

根据需求，R2 应当成为Level-2 DIS。这一需求很容易通过修改接口的Level-2 DIS优先级来实现。

在R2的GE 0/0/0 接口的视图下使用isis dis-priority 127 Level-2命令修改GE 0/0/0接口的Level-2 DIS优先级的值为127。

```
[R2-GigabitEthernet0/0/0]isis dis-priority 127 Level-2
```

在R2 上使用命令display isis interface GigabitEthernet 0/0/0 verbose 查看GE 0/0/0接口的IS-IS协议的详细信息。

```
[R2]display isis interface GigabitEthernet 0/0/0 verbose
```

Interface information for ISIS(1)

Interface Type	DIS	Id	IPv4.State	IPv6.State	MTU
GE0/0/0	001	Up	Down	1497	L1/L2 No/Yes
Circuit MT State			: Standard		
Description			: HUAWEI,AR Series,GigabitEthernet0/0/0 Interface		
SNPA Address			: 00e0-fc03-5d75		
IP Address			: 10.0.1.2		
IPv6 Link Local Address			:		
IPv6 Global Address(es)			:		
Csnp Timer Value			: L1 10 L2 10		
Hello Timer Value			: L1 10 L2 10		
DIS Hello Timer Value			: L1 3 L2 3		
Hello Multiplier Value			: L1 3 L2 3		

LSP-Throttle Timer	:	L12	50		
Cost	:	L1	10	L2	10
IPv6 Cost	:	L1	10	L2	10
Priority	:	L1	64	L2	127
Retransmit Timer Value	:	L12	5		
Bandwidth-Value	:	Low	1000000000	High	0
Static Bfd	:	NO			
Dynamic Bfd	:	NO			
Fast-Sense Rpr	:	NO			

可以看到，R2的GE 0/0/0接口的Level-2 DIS优先级的值已被修改成127，Level-1 DIS优先级的值依旧为64，接口信息中DIS 属性为No/Yes。说明R2 现在是Level-2 DIS。注意，与OSPF协议不同，DIS优先级修改之后，优先级更高的路由器会迅速抢占DIS的角色。

思考

如果一个路由器的DIS优先级的值为0，那它可能会成为DIS吗？

4.5 IS-IS 开销值和协议优先级

原理概述

IS-IS 协议为路由器的每个 IS-IS 接口定义并维护了一个 Level-1 开销值和一个Level-2开销值。开销值可以在接口上或者全局上手动配置，也可以使用Auto-Cost自动计算确定。开销值的优先顺序为：接口上手动配置的开销值，全局上手动配置的开销值，Auto-Cost方式自动计算确定的开销值。

采用Auto-Cost计算确定接口的开销值时，如果开销值类型为Wide，则接口开销值 = (参考带宽 ÷ 接口带宽) × 10；如果开销值类型为Narrow，则接口开销值为与接口带宽绑定的固定值。开销值类型为Narrow时，接口带宽分为几个档次，依次为小于等于10MB、大于10MB小于等于100MB、大于100MB小于等于155MB、大于155MB小于等于622MB、大于622MB小于等于2.5GB，大于2.5GB，而相应的接口开销值分别为60、50、40、30、20、10。在没有任何配置的情况下，IS-IS 开销类型默认为Narrow，且所有带宽档次的接口默认开销值均为10。

任何一条路由都有相应的协议优先级，例如，直连路由的协议优先级的值为0，OSPF内部路由的协议优先级的值为10，静态路由的协议优先级的值为60，RIP路由的协议优先级的值为100，OSPF ASE (AS External) 路由的协议优先级的值为150，EIGP 路由的协议优先级的值为255，IBGP路由的协议优先级的值为255，IS-IS路由的协议优先级的值为15。注意，路由的协议优先级的值越小，路由的优先级越高。

与许多动态路由协议一样，IS-IS 也拥有一系列的计时器，其中的Hello Timer 是用来控制IS-IS Hello 报文发送的时间间隔的。

实验目的

- 掌握修改IS-IS 开销值的方法
- 掌握修改IS-IS 协议优先级的方法
- 掌握修改IS-IS Hello Timer设定值的方法

实验内容

实验拓扑如图4-5所示，实验编址如表4-5所示。本实验模拟了一个简单的企业网络场景，R1、R2、R3、R4 均为Level-1 IS-IS路由器，R1 为企业分支机构的路由器，R4的Loopback 0 接口、Loopback 1 接口以及Loopback 2 接口分别模拟了企业总部的3 台服务器A、B、C。网络需求是：企业分支机构访问服务器A的报文通过R2转发，企业分支机构

访问服务器B 和服务器C 的报文通过R3转发。此外，为了减少链路上IS-IS Hello报文带来的带宽开销，Hello Timer的设定值需要被修改增大。

实验拓扑

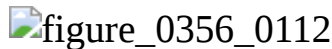
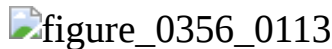
figure_0356_0112

图4-5 IS-IS开销值和协议优先级

实验编址表

表4-5 实验编址

figure_0356_0113

实验步骤

1．基本配置

根据图4-5和表4-5进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=70 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 70/70/70 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置IS-IS路由协议

在每台路由器上配置IS-IS协议。注意，各路由器均为Level-1路由器。

```
[R1]isis
```

```
[R1-isis-1]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-1]is-level level-1
```

```
[R1-isis-1]is-name R1
```

```
[R1-isis-1]quit
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable
```

```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]isis enable
```

```
[R2]isis
```

```
[R2-isis-1]network-entity 10.0000.0000.0002.00
```

```
[R2-isis-1]is-level level-1
```

```
[R2-isis-1]is-name R2
```

```
[R2-isis-1]quit
```

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis enable
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```


[R2-GigabitEthernet0/0/1]isis enable

[R3]isis

[R3-isis-1]network-entity 10.0000.0000.0003.00

[R3-isis-1]is-level level-1

[R3-isis-1]is-name R3

[R3-isis-1]quit

[R3]interface GigabitEthernet 0/0/0

[R3-GigabitEthernet0/0/0]isis enable

[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R3-GigabitEthernet0/0/1]isis enable

[R4]isis

[R4-isis-1]network-entity 10.0000.0000.0004.00

[R4-isis-1]is-level level-1

[R4-isis-1]is-name R4

[R4-isis-1]quit

[R4]interface GigabitEthernet 0/0/0

[R4-GigabitEthernet0/0/0]isis enable

[R4-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R4-GigabitEthernet0/0/1]isis enable

[R4-GigabitEthernet0/0/1]interface LoopBack 0

```
[R4-LoopBack0]isis enable
```

```
[R4-LoopBack0]interface LoopBack 1
```

```
[R4-LoopBack1]isis enable
```

```
[R4-LoopBack1]interface LoopBack 2
```

```
[R4-LoopBack2]isis enable
```

配置完成后，在R1上查看IS-IS邻居信息。

```
[R1]display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R2	GE0/0/0	R1.01	Up	24s	L1	64
R3	GE0/0/1	R1.02	Up	27s	L1	64

Total Peer(s): 2

可以看到，R1与R2和R3成功建立了Level-1邻接关系。读者可自行在其他路由器上查看邻居信息情况。

3．修改IS-IS开销值

在R1 上使用display isis route 命令查看IS-IS 路由表。

```
[R1]display isis route
```

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPv4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags	

10.0.100.2/32	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-	
GE0/0/1	10.0.13.3					
10.0.24.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-	
10.0.100.1/32	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-	
GE0/0/1	10.0.13.3					
10.0.13.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-	
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-	
10.0.100.3/32	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-	
GE0/0/1	10.0.13.3					
10.0.34.0/24	20	NULL	GE0/0/1	10.0.13.3	A/-/-/-	

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R1去往10.0.100.1/32、10.0.100.2/32、10.0.100.3/32的路由采用了负载均衡方式，分别以R2和R3为下一跳。

在R1上查看GE 0/0/0接口的IS-IS协议详细信息。

[R1]display isis interface GigabitEthernet 0/0/0 verbose

Interface information for ISIS(1)

Interface Type	DIS	Id	IPv4.State	IPv6.State	MTU	
GE0/0/0	001	Up	Down	1497	L1/L2	Yes/No
Circuit MT State			:Standard			
Description			:HUAWEI,AR			
Series,GigabitEthernet0/0/0 Interface						
SNPA Address			: 00e0-fc03-f7d1			
IP Address			: 10.0.12.1			
IPv6 Link Local Address :						
IPv6 Global Address(es)			:			
Csnp Timer Value			: L1 10 L2 10			
Hello Timer Value			: L1 10 L2 10			
DIS Hello Timer Value			: L1 3 L2 3			
Hello Multiplier Value			: L1 3 L2 3			
LSP-Throttle Timer			: L12 50			
Cost			: L1 10 L2 10			
IPv6 Cost			: L1 10 L2 10			
Priority			: L1 64 L2 64			
Retransmit Timer Value			: L12 5			

Bandwidth-Value : Low 1000000000 High 0

Static Bfd : NO

Dynamic Bfd : NO

Fast-Sense Rpr : NO

可以看到，R1 的GE 0/0/0 接口的IS-IS Level-1 和Level-2 的开销值均为10。

为了使R1访问10.0.100.1/32、10.0.100.2/32、10.0.100.3/32的报文都通过R3转发，可以在R1 的GE 0/0/0 接口上使用isis cost 50 level-1 命令修改Level-1 的开销值为50。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis cost 50 level-1
```

配置完成后，在R1 上查看GE 0/0/0 接口的IS-IS 协议详细信息。

```
[R1]display isis interface GigabitEthernet 0/0/0 verbose
```

Interface information for ISIS(1)

Interface	Id	IPv4.State	IPv6.State	MTU
GE0/0/0	001	Up	Down	1497
L1/L2 Yes/No				
Circuit MT State		:Standard		
Description		:HUAWEI,AR		
Series,GigabitEthernet0/0/0 Interface				
SNPA Address		: 00e0-fc03-f7d1		

IP Address	: 10.0.12.1
IPv6 Link Local Address :	
IPv6 Global Address(es)	:
Csnp Timer Value	: L1 10 L2 10
Hello Timer Value	: L1 10 L2 10
DIS Hello Timer Value	: L1 3 L2 3
Hello Multiplier Value	: L1 3 L2 3
LSP-Throttle Timer	: L12 50
Cost	: L1 50 L2 10
IPv6 Cost	: L1 10 L2 10
Priority	: L1 64 L2 64
Retransmit Timer Value	: L12 5
Bandwidth-Value	: Low 1000000000 High 0
Static Bfd	: NO
Dynamic Bfd	: NO
Fast-Sense Rpr	: NO

可以看到 , R1 的GE 0/0/0 接口的Level-1 开销值已经变为了50。

在R1上查看IS-IS路由表。

[R1]display isis route

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPv4 Destination		IntCost	ExtCost	ExitInterface	
NextHop	Flags				

10.0.100.2/32	20	NULL	GE0/0/1	10.0.13.3	A/-/-/-
10.0.24.0/24	30	NULL	GE0/0/1	10.0.13.3	A/-/-/-
10.0.100.1/32	20	NULL	GE0/0/1	10.0.13.3	A/-/-/-
10.0.13.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-
10.0.12.0/24	50	NULL	GE0/0/0	Direct	D/-/L/-
10.0.100.3/32	20	NULL	GE0/0/1	10.0.13.3	A/-/-/-
10.0.34.0/24	20	NULL	GE0/0/1	10.0.13.3	A/-/-/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，现在R1去往10.0.100.1/32、10.0.100.2/32、10.0.100.3/32的路由的下一跳均为R3。

在R1上使用tracert命令验证去往10.0.100.1/32、10.0.100.2/32、10.0.100.3/32的报文所经过的路径。

[R1]tracert 10.0.100.1

traceroute to 10.0.100.1(10.0.100.1), max hops: 30 ,packet length: 40,press CTRL_C to break

1 10.0.13.3 90 ms 60 ms 30 ms

2 10.0.34.4 70 ms 20 ms 10 ms

[R1]tracert 10.0.100.2

traceroute to 10.0.100.2(10.0.100.2), max hops: 30 ,packet length: 40,press CTRL_C to break

1 10.0.13.3 40 ms 1 ms 20 ms

2 10.0.34.4 20 ms 20 ms 10 ms

[R1]tracert 10.0.100.3

traceroute to 10.0.100.3(10.0.100.3), max hops: 30 ,packet length: 40,press CTRL_C to break

1 10.0.13.3 10 ms 10 ms 10 ms

2 10.0.34.4 30 ms 10 ms 20 ms

可以看到，R1去往各服务器的报文均选择了经由R3的路径。

4．修改IS-IS协议优先级

接下来，为了使R1访问服务器A的报文选择经由R2的路径，可配置如下的静态路由。

[R1]ip route-static 10.0.100.1 32 10.0.12.2

在R1上查看路由表。

[R1]display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 15 Routes : 15

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

10.0.12.0/24 GigabitEthernet0/0/0	Direct	0	0	D	10.0.12.1

.....

10.0.34.0/24 GigabitEthernet0/0/1	ISIS-L1	15	20	D	10.0.13.3

10.0.100.1/32 GigabitEthernet0/0/1	ISIS-L1	15	20	D	10.0.13.3

10.0.100.2/32 GigabitEthernet0/0/1	ISIS-L1	15	20	D	10.0.13.3

.....

可以看到，路由表中R1去往10.0.100.1/32的路由信息依旧是通过IS-IS协议所获得的，这是由于IS-IS协议的协议优先级的值为15，而静态路由的协议优先级的值为60。注意，路由的协议优先级的值越小，路由的优先级就越大。

在R1上使用tracert命令验证从R1去往10.0.100.1/32的报文所经过的路径。

[R1]tracert 10.0.100.1

traceroute to 10.0.100.1(10.0.100.1), max hops: 30 ,packet length: 40,press CTRL_C to break

```
1 10.0.13.3 80 ms 40 ms 20 ms
```

```
2 10.0.34.4 40 ms 10 ms 10 ms
```

可以看到，R1访问服务器A时依旧使用的是经由R3的路径。

在R1 上使用display default-parameter isis命令查看IS-IS 协议的默认参数。

```
[R1]display default-parameter isis
```

```
Default Configurations For Process
```

```
-----  
  
Cost-Style                               :Narrow  
  
.....  
  
LSP-Refresh-Interval<s>                 :   900  
  
Preference                               :   IPv4 15 IPv6 15  
  
SPF-Intelli Timer<s,ms,ms>              : Max 5 Init 50 Incr 200  
  
.....
```

可以看到，IS-IS对于IPv4的协议优先级的值默认为15。接下来，在R1的IS-IS视图下使用preference 70 命令修改R1 的IS-IS 协议优先级的值为70。

```
[R1]isis
```

```
[R1-isis-1]preference 70
```

配置完成后，在R1上查看路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 15 Routes : 15

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

10.0.12.0/24 GigabitEthernet0/0/0	Direct	0	0	D	10.0.12.1

.....

10.0.34.0/24 GigabitEthernet0/0/1	ISIS-L1	15	20	D	10.0.13.3

10.0.100.1/32 GigabitEthernet0/0/0	Static	60	0	RD	10.0.12.2

10.0.100.2/32 GigabitEthernet0/0/1	ISIS-L1	15	20	D	10.0.13.3

.....

可以看到，现在R1访问10.0.100.1/32的报文使用的是静态路由了，经由R2转发。

在R1上使用tracert命令验证从R1去往10.0.100.1/32的报文所经过的路径。

[R1]tracert 10.0.100.1

tracert to 10.0.100.1(10.0.100.1), max hops: 30 ,packet length: 40,press
CTRL_C to break

```
1 10.0.12.2 70 ms 50 ms 20 ms
```

```
2 10.0.24.4 50 ms 20 ms 10 ms
```

可以看到，现在R1访问服务器A时选择了经由R2的路径。

在R1上使用tracert命令验证从R1去往10.0.100.2/32和10.0.100.3/32的报文所经过的路径。

```
[R1]tracert 10.0.100.2
```

```
traceroute to 10.0.100.2(10.0.100.2), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.13.3 10 ms 10 ms 10 ms
```

```
2 10.0.34.4 10 ms 20 ms 10 ms
```

```
[R1]tracert 10.0.100.3
```

```
traceroute to 10.0.100.3(10.0.100.3), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.13.3 10 ms 10 ms 10 ms
```

```
2 10.0.34.4 20 ms 20 ms 10 ms
```

可以看到，R1去往服务器B和服务器C时依旧选择的是经由R3的路径。

5．修改IS-IS Hello Timer的设定值

为了减少在链路上发送IS-IS Hello 报文的频率，可以人为地增大IS-IS Hello 报文的时间间隔。在R1 上查看GE 0/0/0 接口的IS-IS 协议详细信息。

```
[R1]display isis interface GigabitEthernet 0/0/0 verbose
```

```
Interface information for ISIS(1)
```

Interface Type	DIS	Id	IPv4.State	IPv6.State	MTU
GE0/0/0	001	Up	Down	1497 L1/L2	Yes/No

Circuit MT State : Standard

Description : HUAWEI,AR
Series,GigabitEthernet0/0/0 Interface

SNPA Address : 00e0-fc03-f7d1

IP Address : 10.0.12.1

IPv6 Link Local Address :

IPv6 Global Address(es) :

Csnp Timer Value : L1 10 L2 10

Hello Timer Value : L1 10 L2 10

DIS Hello Timer Value : L1 3 L2 3

Hello Multiplier Value : L1 3 L2 3

.....

可以看到，默认情况下IS-IS 接口的Level-1和Level-2 Hello报文时间间隔均为10s，但是DIS 接口的Level-1和Level-2 Hello 报文时间间隔均为3s（自动取相应值的三分之一，并取整）。注意，R1 的GE 0/0/0 接口现在就是Level-1 DIS接口。

在R1上使用debugging isis adjacency GigabitEthernet 0/0/0命令查看GE 0/0/0接口的Hello报文发送情况。

<R1>debugging isis adjacency interface GigabitEthernet0/0/0

Aug 21 2013 03:30:47.605.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2679)

<R1>

Aug 21 2013 03:30:47.605.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

<R1>

Aug 21 2013 03:30:50.605.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2679)

<R1>

Aug 21 2013 03:30:50.605.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

<R1>

Aug 21 2013 03:30:53.605.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2679)

<R1>

Aug 21 2013 03:30:53.605.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

<R1> undo debugging all

可以看到，由于R1的GE 0/0/0 接口是Level-1 DIS 接口，所以其Hello 报文的时间间隔为3s。

在R1的GE 0/0/0 接口视图下使用isis timer hello 30 level-1 命令修改GE 0/0/0 接口发送IS-IS Level-1 Hello报文的时间间隔为30s。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis timer hello 30 level-1
```

配置完成后，在R1 上查看GE 0/0/0 接口的IS-IS 协议详细信息。

```
[R1]display isis interface GigabitEthernet 0/0/0 verbose
```

Interface information for ISIS(1)

Interface	Id		IPv4.State	IPv6.State	MTU	Type	DIS
GE0/0/0	001	Up	Down	1497 L1/L2	Yes/No		

Circuit MT State : Standard

Description : HUAWEI,AR
Series,GigabitEthernet0/0/0 Interface

SNPA Address : 00e0-fc03-f7d1

IP Address : 10.0.12.1

IPv6 Link Local Address :

IPv6 Global Address(es) :

Csnp Timer Value : L1 10 L2 10

Hello Timer Value : L1 30 L2 10

DIS Hello Timer Value : L1 10 L2 3

Hello Multiplier Value : L1 3 L2 3

.....

可以看到，Level-1 Hello时间间隔修改为30s 后，DIS 的Level-1 Hello时间间隔自动变化10s。

在R1 上使用debugging isis adjacency interface GigabitEthernet 0/0/0 命令查看GE 0/0/0接口发送Hello的情况。

```
<R1>debugging isis adjacency interface GigabitEthernet0/0/0
```

```
<R1>
```

Aug 21 2013 03:38:54.205.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2679)

```
<R1>
```

Aug 21 2013 03:38:54.205.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

```
<R1>
```

Aug 21 2013 03:39:04.205.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2679)

```
<R1>
```


Aug 21 2013 03:39:04.205.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

<R1>

Aug 21 2013 03:39:14.205.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH encode cache to send IIH, GE0/0/0.
(IS15_2679)

<R1>

Aug 21 2013 03:39:14.205.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

<R1>undo debugging all

可以看到，现在R1 的GE 0/0/0 接口的Level-1 Hello 报文时间间隔为
10s。

查看R1 的GE 0/0/1 接口的Level-1 Hello 报文时间间隔。

<R1>display isis interface GigabitEthernet0/0/1 verbose

Interface information for ISIS(1)

Interface	Id		IPv4.State	IPv6.State	MTU	Type
DIS						

GE0/0/1	002	Up	Down	1497	L1/L2	No/No

Circuit MT State : Standard

Description : HUAWEI,AR Series,GigabitEthernet0/0/1 Interface

SNPA Address : 00e0-fc03-d8d6

IP Address : 10.0.13.1

IPv6 Link Local Address :

IPv6 Global Address(es) :

Csnp Timer Value : L1 10 L2 10

Hello Timer Value : L1 10 L2 10

DIS Hello Timer Value : L1 3 L2 3

Hello Multiplier Value : L1 3 L2 3

.....

可以看到，R1 并不是接口GE 0/0/1所在链路的DIS 路由器，GE 0/0/1的Hello 报文的时间间隔仍为10s。

使用命令debugging isis adjacency interface GigabitEthernet 0/0/1 查看GE 0/0/1 发送Hello报文的情况。

```
<R1>debugging isis adjacency interface GigabitEthernet0/0/1
```

```
Sep 27 2013 20:30:16.672.1-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/1.  
(IS15_2679)
```

```
<R1>
```

```
Sep 27 2013 20:30:16.672.2-05:13 R1 ISIS/6/ISIS:
```

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/1, to SNPA 0180.c200.0014.
(IS15_6941)

<R1>

Sep 27 2013 20:30:26.682.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/1.
(IS15_2679)

<R1>

Sep 27 2013 20:30:26.682.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/1, to SNPA 0180.c200.0014.
(IS15_6941)

<R1>

Sep 27 2013 20:30:36.682.1-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/1.
(IS15_2679)

<R1>

Sep 27 2013 20:30:36.682.2-05:13 R1 ISIS/6/ISIS:

ISIS-1-ADJ: Sending Lan L1 Hello on GE0/0/1, to SNPA 0180.c200.0014.
(IS15_6941)

可以看到，R1 的GE 0/0/1 接口发送Level-1 Hello 报文的时间间隔的确为10s。

思考

直连路由的协议优先级的值，静态路由的协议优先级的值，RIP 路由的协议优先级的值，OSPF路由的协议优先级的值，BGP路由的协议优先级的值，IS-IS路由的协议优先级的值各是多少？

4.6 IS-IS 路由聚合

原理概述

与 OSPF 协议相同，IS-IS 也能够通过路由聚合来减少路由条目。不同的是，OSPF 只能够在 ABR 和 ASBR 路由器上进行路由聚合，而 IS-IS 路由器能否进行路由聚合以及对什么样的路由才能进行聚合取决于路由器的类型及路由的类型。

在 IS-IS 网络中，Level-1 路由器只维护 Level-1 链路状态数据库，只能对相应的 Level-1 的直连路由进行聚合，并将聚合后的路由以 Level-1 LSP 的形式发送给其他路由器；Level-2 路由器只维护 Level-2 链路状态数据库，只能对相应的 Level-2 的直连路由进行聚合，并将聚合后的路由以 Level-2 LSP 的形式发送给其他路由器；Level-1-2 路由器分别维护了 Level-1 和 Level-2 链路状态数据库，Level-1-2 路由器能够将 Level-1 路由表中的路由（不必一定是直连路由）进行聚合后以 Level-1 LSP 的形式发送给其他路由器，将 Level-2 路由表中的路由（不必一定是直连路由）进行聚合后以 Level-2 LSP 的形式发送给其他路由器，并且还能够将 Level-1 路由表中的路由（不必一定是直连路由）聚合后以 Level-2 LSP 的形式发送给其他路由器。

实验目的

- 理解 IS-IS 网络中路由聚合的条件和类型
- 掌握配置 IS-IS 路由聚合的方法

实验内容

实验拓扑如图 4-6 所示，实验编址如表 4-6 所示。本实验模拟了一个简单的企业网络场景，R1 和 R2 为公司总部的路由器，R3 为公司分支机构的路由器。R1 为 Level-1 路由器，R2 为 Level-1-2 路由器，R3 为 Level-2 路由器。R1 和 R2 属于 IS-IS 区域 10，R3 属于 IS-IS 区域 20。R1 和 R3 的 Loopback 接口分别用来模拟公司总部的各个网络和公司分支机构的各

个的网络。网络需求是：全网互通，并配置路由聚合以精简路由表中的路由条目。

实验拓扑

figure_0365_0114

图4-6 IS-IS路由聚合

实验编址表

表4-6 实验编址

figure_0365_0115

(续表)

figure_0366_0116

实验步骤

1. 基本配置

根据图4-6和表4-6进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=100 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

0.00% packet loss

round-trip min/avg/max = 100/100/100 ms

R2与R3之间的连通性测试过程在此省略。

2 . 配置IS-IS路由协议

配置IS-IS协议，其中R1为Level-1路由器，R2为Level-1-2路由器，R3为Level-2路由器。

```
[R1]isis
```

```
[R1-isis-1]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-1]is-level level-1
```

```
[R1-isis-1]is-name R1
```

```
[R2]isis
```

```
[R2-isis-1]network-entity 10.0000.0000.0002.00
```

```
[R2-isis-1]is-name R2
```

```
[R3]isis
```

```
[R3-isis-1]network-entity 20.0000.0000.0003.00
```

```
[R3-isis-1]is-level level-2
```

```
[R3-isis-1]is-name R3
```

在R1、R2、R3的各接口上使能IS-IS。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable
```

[R1-GigabitEthernet0/0/0]interface LoopBack 0

[R1-LoopBack0]isis enable

[R1-LoopBack0]interface LoopBack 1

[R1-LoopBack1]isis enable

[R1-LoopBack1]interface LoopBack 2

[R1-LoopBack2]isis enable

[R2]interface GigabitEthernet 0/0/0

[R2-GigabitEthernet0/0/0]isis enable

[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R2-GigabitEthernet0/0/1]isis enable

[R3]interface GigabitEthernet 0/0/1

[R3-GigabitEthernet0/0/1]isis enable

[R3-GigabitEthernet0/0/1]interface LoopBack 0

[R3-LoopBack0]isis enable

[R3-LoopBack0]interface LoopBack 1

[R3-LoopBack1]isis enable

[R3-LoopBack1]interface LoopBack 2

[R3-LoopBack2]isis enable

配置完成后，在R2上查看IS-IS邻居信息。

[R2]display isis peer

Peer information for ISIS(10)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	R1.01	Up	8s	L1	64
R3	GE0/0/1	R3.01	Up	7s	L2	64

Total Peer(s): 2

可以看到，R2与R1和R3已经成功建立起了IS-IS邻接关系。

在R1 上使用display isis route 命令查看IS-IS 路由表。

<R1>display isis route

Route information for ISIS(1)

IPv4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags
0.0.0.0/0	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.1.3/32	0	NULL	Loop2	Direct	D/-/L/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

10.0.1.2/32	0	NULL	Loop1	Direct	D/-/L/-

10.0.1.1/32	0	NULL	Loop0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

观察发现，R1的IS-IS路由表中没有R3上的Loopback接口的明细路由，
但有一条下一跳是R2（10.0.12.2）的缺省路由，该缺省路由是Level-1-
2路由器R2发布的。

3．在Level-1路由器上进行路由聚合

在R2 上使用display isis route 命令查看IS-IS 路由表。

```
[R2]display isis route
```

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPv4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-

10.0.1.3/32	10	NULL	GE0/0/0	10.0.12.1	A/-/L/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

10.0.1.2/32	10	NULL	GE0/0/0	10.0.12.1	A/-/L/-

10.0.1.1/32 10 NULL GE0/0/0 10.0.12.1 A/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

ISIS(1) Level-2 Forwarding Table

IPv4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.3.3/32	10	NULL	GE0/0/1	10.0.23.3	A/-/-/-

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-

10.0.3.2/32	10	NULL	GE0/0/1	10.0.23.3	A/-/-/-

10.0.3.1/32	10	NULL	GE0/0/1	10.0.23.3	A/-/-/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R2的Level-1路由表中拥有公司总部的各个网络（10.0.1.1/32，10.0.1.2/32，10.0.1.3/32）的明细路由，Level-2路由表中拥有公司分支机构的各个网络（10.0.3.1/32，10.0.3.2/32，10.0.3.3/32）的明细路由。

在R1的IS-IS视图下使用summary 10.0.1.0 255.255.255.252命令对公司总部的各个网络进行路由聚合。

```
[R1-isis-1]summary 10.0.1.0 255.255.255.252
```

在R2 上使用display isis route level-1 命令查看IS-IS 的Level-1 路由表。

```
[R2]display isis route level-1
```

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPv4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-
10.0.1.3/32	10	NULL	GE0/0/0	10.0.12.1	A/-/L/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-
10.0.1.2/32	10	NULL	GE0/0/0	10.0.12.1	A/-/L/-
10.0.1.1/32	10	NULL	GE0/0/0	10.0.12.1	A/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

观察发现，R2的Level-1路由表并没有什么变化，依然存在10.0.1.1/32、10.0.1.2/32和10.0.1.3/32这3条明细路由。原来，在默认情况下，IS-IS总是将聚合后的路由以Level-2 LSP 的形式传递给其他路由器，而R1 是一个Level-1 路由器，无法以Level-2 LSP 的形式传递聚合后的路由，所以之前的路由聚合命令是产生不了任何效果的。

在R1的IS-IS 视图下使用summary 10.0.1.0 255.255.255.252 level-1命令，将聚合后的路由以Level-1 LSP的形式传递给其他路由器。

```
[R1-isis-1]summary 10.0.1.0 255.255.255.252 level-1
```

查看R2上的Level-1路由表。

```
[R2]display isis route level-1
```

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPv4 Destination		IntCost	ExtCost	ExitInterface	
NextHop	Flags				

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D-/L/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D-/L/-
10.0.1.0/30	10	NULL	GE0/0/0	10.0.12.1	A-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

现在看到，R2的Level-1路由表中，已经不再有10.0.1.1/32、10.0.1.2/32、10.0.1.3/32这3条明细路由了，取而代之的是一条10.0.1.0/30的聚合路由，说明在R1上配置的路由聚合已经生效。

在R3上查看IS-IS路由表。

```
[R3]display isis route
```

Route information for ISIS(1)

ISIS(1) Level-2 Forwarding Table

IPv4 Destination NextHop	Flags	IntCost	ExtCost	ExitInterface		

10.0.3.3/32	0	NULL	Loop0	Direct	D/-/L/-	
10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-	
10.0.3.2/32	0	NULL	Loop1	Direct	D/-/L/-	
10.0.3.1/32	0	NULL	Loop2	Direct	D/-/L/-	
10.0.12.0/24	20	NULL	GE0/0/1	10.0.23.2	A/-/-/-	
10.0.1.0/30	20	NULL	GE0/0/1	10.0.23.2	A/-/-/-	

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R3的IS-IS路由表中没有10.0.1.1/32、10.0.1.2/32、10.0.1.3/32这3条明细路由了，只有它们的聚合路由，这再一次说明了在R1上配置的路由聚合已经生效。

4．在Level-2路由器上进行路由聚合

接下来将在Level-2路由器R3上将Level-2路由进行聚合。

在R3的IS-IS视图下使用summary 10.0.3.0 255.255.255.252 level-2命令进行路由聚合，将聚合后的路由以Level-2 LSP的形式传递给其他路由器。

```
[R3-isis-1]summary 10.0.3.0 255.255.255.252 level-2
```

配置完成后，在R2上使用display isis route level-2 命令查看Level-2 路由表。

```
[R2]display isis route level-2
```

Route information for ISIS(1)

ISIS(1) Level-2 Forwarding Table

IPv4 Destination		IntCost	ExtCost	ExitInterface	
NextHop	Flags				

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-
10.0.3.0/30	10	NULL	GE0/0/1	10.0.23.3	A/-/-/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R2 的 Level-2 路由表中不再有公司分支机构的各个网络（10.0.3.1/32，10.0.3.2/32，10.0.3.3/32）的明细路由了，取而代之的是一条聚合路由，这说明在 R3 上配置的路由聚合已经生效。

5．在Level-1-2路由器上进行路由聚合

从前面的实验我们可以知道，在Level-1路由器上对Level-1直连路由进行聚合，在Level-2路由器上对Level-2直连路由进行聚合，两种方式的

聚合都可以减少路由条目。另外还知道，对Level-1路由进行聚合后，还可以减少Level-2路由条目。

接下来将在 Level-1-2 路由器 R2 上进行路由聚合。进行聚合之前，先取消以前在R1和R3上进行的路由聚合配置。

```
[R1-isis-1]undo summary 10.0.1.0 255.255.255.252 level-1
```

```
[R3-isis-1]undo summary 10.0.3.0 255.255.255.252 level-2
```

在R2上查看Level-1路由表。

```
[R2]display isis route level-1
```

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPv4 Destination		IntCost	ExtCost	ExitInterface	Flags				
NextHop	Flags								

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D-/L/-				
10.0.1.3/32	10	NULL	GE0/0/0	10.0.12.1	A-/L/-				
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D-/L/-				
10.0.1.2/32	10	NULL	GE0/0/0	10.0.12.1	A-/L/-				
10.0.1.1/32	10	NULL	GE0/0/0	10.0.12.1	A-/L/-				

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R2的Level-1路由表中有10.0.1.1/32、10.0.1.2/32、10.0.1.3/32这3条明细路由。

在 R2 上使用 display isis lsdb is-name R2 level-2 verbose 命令，查看 R2 生成的Level-2 LSP的详细信息。

```
[R2]display isis lsdb is-name R2 level-2 verbose
```

```
Database information for ISIS(1)
```

```
-----
```

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

```
-----  
-----
```

0000.0000.0002.00-00*	0x00000016	0x36b5	550	116	0/0/0

SOURCE	R2.00

HOST NAME	R2

NLPID	IPv4

AREA ADDR	10

INTF ADDR	10.0.23.2

INTF ADDR	10.0.12.2

NBR	ID	R3.01	COST: 10

IP-Internal	10.0.23.0	255.255.255.0	COST: 10

IP-Internal	10.0.12.0	255.255.255.0	COST: 10
IP-Internal	10.0.1.1	255.255.255.255	COST: 10
IP-Internal	10.0.1.2	255.255.255.255	COST: 10
IP-Internal	10.0.1.3	255.255.255.255	COST: 10

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，这条Level-2 LSP对Level-1 路由表中的3条关于10.0.1.1/32、10.0.1.2/32、10.0.1.3/32的明细路由进行了描述。

在R2的IS-IS 视图下使用summary 10.0.1.0 255.255.255.252 level-2命令进行路由聚合。

```
[R2-isis-1]summary 10.0.1.0 255.255.255.252 level-2
```

在 R2 上使用 display isis lsdb is-name R2 level-2 verbose 命令，查看 R2 生成的Level-2 LSP的详细信息。

```
[R2]display isis lsdb is-name R2 level-2 verbose
```

Database information for ISIS(1)

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

0000.0000.0002.00-00*	0x00000016	0x36b5	550	116	0/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPv4

AREA ADDR 10

INTF ADDR 10.0.23.2

INTF ADDR 10.0.12.2

NBR ID R3.01 COST: 10

IP-Internal 10.0.23.0 255.255.255.0 COST: 10

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

IP-Internal 10.0.1.0 255.255.255.252 COST: 10

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R2 现在生成的 Level-2 LSP 不再分别具体地描述关于 10.0.1.1/32、10.0.1.2/32、10.0.1.3/32 的路由，因为这3 条路由已经被聚合了，这条Level-2 LSP描述的是聚合后的路由。

在R3上查看IS-IS路由表。

[R3]display isis route

Route information for ISIS(1)

ISIS(1) Level-2 Forwarding Table

IPv4 Destination		IntCost	ExtCost	ExitInterface		
NextHop	Flags					
10.0.3.3/32	0	NULL	Loop0	Direct	D/-/L/-	
10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-	
10.0.3.2/32	0	NULL	Loop1	Direct	D/-/L/-	
10.0.3.1/32	0	NULL	Loop2	Direct	D/-/L/-	
10.0.12.0/24	20	NULL	GE0/0/1	10.0.23.2	A/-/-/-	
10.0.1.0/30	20	NULL	GE0/0/1	10.0.23.2	A/-/-/-	

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R3的IS-IS路由表中现在拥有了关于10.0.1.0/30的聚合路由，原因是R2将描述这条聚合路由的Level-2 LSP传递给了Level-2 路由器R3。

思考

能否在Level-1-2路由器上对Level-2路由进行聚合？

4.7 IS-IS 缺省路由

原理概述

IS-IS有两种缺省路由，第一种缺省路由是由Level-1路由器在特定的条件下自动产生的，它的下一跳是离它最近的（Cost最小）Level-1-2路由器；第二种缺省路由是在IS-IS路由器上使用default-route-advertise命令产生并发布的。

实验目的

- 理解IS-IS 中缺省路由的种类
- 掌握在IS-IS 协议中发布缺省路由的方法

实验内容

实验拓扑如图4-7所示，实验编址如表4-7所示。本实验模拟了一个企业网络场景，R1和R2是公司A总部的路由器，R3为公司A的分支机构的路由器，R4是公司B的路由器。R1、R2、R3运行IS-IS协议，其中R1为Level-1路由器，R2为Level-1-2路由器，R3 为Level-2 路由器，R1 的Loopback 0 接口和R3的Loopback 0 接口分别模拟了公司总部的内部网络和分支机构的内部网络，R4 的Loopback 0 接口模拟了公司B 的内部网络。这两个公司之间有业务合作需要网络互通，R4 通过静态路由访问公司 A 的总部和分支机构。R2作为公司A的出口路由器，使用缺省路由访问公司B。公司A的网络管理员需要在R2上通过IS-IS协议发布缺省路由，使得公司A总部的内部网络和分支机构的内部网络能够与公司B的内部网络进行互访，且当R2与R4的链路出现故障时，缺省路由的发布将自动停止。

实验拓扑

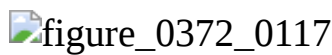
figure_0372_0117

图4-7 IS-IS缺省路由

实验编址表

表4-7 实验编址

figure_0373_0118

实验步骤

1．基本配置

根据图4-7和表4-7进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=570 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 570/570/570 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置IS-IS路由协议

在R1、R2、R3上配置IS-IS协议。

```
[R1]isis
```

```
[R1-isis-1]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-1]is-name R1
```

```
[R1-isis-1]is-level level-1
```

[R2]isis

[R2-isis-1]network-entity 10.0000.0000.0002.00

[R2-isis-1]is-name R2

[R3]isis

[R3-isis-1]network-entity 20.0000.0000.0003.00

[R3-isis-1]is-name R3

[R3-isis-1]is-level level-2

[R1]interface GigabitEthernet 0/0/0

[R1-GigabitEthernet0/0/0]isis enable

[R1-GigabitEthernet0/0/0]interface LoopBack 0

[R1-LoopBack0]isis enable

[R2]interface GigabitEthernet 0/0/0

[R2-GigabitEthernet0/0/0]isis enable

[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1

[R2-GigabitEthernet0/0/1]isis enable

[R3]interface GigabitEthernet 0/0/0

[R3-GigabitEthernet0/0/0]isis enable

[R3-GigabitEthernet0/0/0]interface LoopBack 0

[R3-LoopBack0]isis enable

查看R2的IS-IS邻居信息。

<R2>display isis peer

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	R1.01	Up	8s	L1	64

R3	GE0/0/1	R3.01	Up	9s	L2	64

Total Peer(s): 2

可以看到，R2与R1建立了Level-1邻接关系，与R3建立了Level-2邻接关系。在R2上查看IP路由表。

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 15Routes : 15

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop

10.0.1.1/32 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.1

10.0.3.3/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.23.3

```
10.0.12.0/24  Direct    0    0    D    10.0.12.2
GigabitEthernet0/0/0
```

.....

可以看到，R2 已经接收到了R1 和R3 的Loopback 0 接口的路由信息。

由于公司A与公司B有业务往来，所以公司A的网络与公司B的网络需要互通。R2 作为公司 A 的网络出口，可在其上配置静态缺省路由来访问公司 B 的内部网络（10.0.4.4/32），而公司B可在R4上配置静态路由以访问公司A的内部网络（10.0.1.1/32和10.0.3.3/32）。

```
[R2]ip route-static 0.0.0.0 0 10.0.24.4
```

```
[R4]ip route-static 10.0.1.1 32 10.0.24.2
```

```
[R4]ip route-static 10.0.3.3 32 10.0.24.2
```

测试R2与公司B的内部网络之间的连通性。

```
<R2>ping -c 1 10.0.4.4
```

```
PING 10.0.4.4: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.4.4: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
--- 10.0.4.4 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 60/60/60 ms
```

可以看到，现在R2与公司B的内部网络10.0.4.4/32是互通的。

3．观察自动生成的缺省路由

在R1上使用ping命令测试10.0.1.1与10.0.4.4/32之间的连通性，即公司A总部的内部网络与公司B的内部网络的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.4.4
```

```
PING 10.0.4.4: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.4.4: bytes=56 Sequence=1 ttl=254 time=110 ms
```

```
--- 10.0.4.4 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 110/110/110 ms
```

可以看到，公司A总部的内部网络与公司B的内部网络是互通的。

在R1上查看路由表。

```
<R1>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```
-----  
-----
```

```
Routing Tables: Public
```

```
Destinations : 10Routes : 10
```

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.2

```
10.0.1.1/32  Direct    0    0    D        127.0.0.1  LoopBack0
.....
```

可以看到，R1的路由表中有一条IS-IS缺省路由，下一跳为R2（10.0.12.2）。另外，路由表中是没有R3的Loopback 0（10.0.3.3/32）的明细路由的。

在IS-IS网络中，Level-1路由器只有本区域的路由信息，所有连接骨干区域的Level-1-2路由器会在自己的Level-1 LSP中设置ATT位（Attached-bit）为1，本区域的Level-1路由器收到来自不同Level-1-2路由器的且ATT位为1的Level-1 LSP后，会比较哪台Level-1-2路由器离自己最近（Cost值最小），并自动产生一条缺省路由指向这个最近的Level-1-2路由器。Level-1路由器需要去往目的地为本区域以外的任何地方时，只需使用这条缺省路由即可。

查看R1的IS-IS LSDB。

```
<R1>display isis lsdb
```

```
Database information for ISIS(1)
```

```
-----
```

```
Level-1 Link State Database
```

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R1.00-00*	0x00000007	0x1859	918	88	0/0/0
R1.01-00*	0x00000003	0xb1d9	918	55	0/0/0
R2.00-00	0x00000008	0x12c	962	88	1/0/0

```
Total LSP(s): 3
```

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，在R1的IS-IS LSDB中，有一条由Level-1-2路由器R2产生的ATT位被设置为1的Level-1 LSP。所以，R1自动生成了一条指向R2的缺省路由。

注意，如果Level-1-2路由器连接到IS-IS骨干区域的链路发生故障，则该Level-1-2路由器将不会再把自己产生的Level-1 LSP的ATT位设置为1，而是设置为0。关闭R2的GE 0/0/1接口，看看会发生什么。

```
[R2]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]shutdown
```

然后在R2上查看IS-IS邻居信息。

```
<R2>display isis peer
```

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime	Type

PRI

R1	GE0/0/0	R1.01	Up	8s	L1 64

Total Peer(s): 1

可以看到，R2现在只有一个Level-1邻居R1，没有任何Level-2邻居。R2不再与骨干区域相连。

在R1上查看IS-IS LSDB信息。

```
<R1>display isis lsdb
```

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R1.00-00*	0x00000007	0x1859	780	88	0/0/0
R1.01-00*	0x00000003	0xb1d9	780	55	0/0/0
R2.00-00	0x0000000a	0x2ae8	1146	72	0/0/0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

观察发现，R2 产生的Level-1 LSP的ATT 位现在的确变为0 了。这样一来，R1 是无法自动生成一条指向R2的缺省路由的。

在R1上查看IS-IS路由表。

<R1>display isis route

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.12.0/24 10 NULL GE0/0/0 Direct D/-/L/-

10.0.1.1/32 0 NULL Loop0 Direct D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

可以看到，R1的IS-IS路由表中没有了缺省路由，也无其他区域的路由。

为进行后续实验，请保持R2的GE 0/0/1接口为关闭状态。

4．手动向R1发布缺省路由

我们知道，本来R1能够通过自己自动生成的缺省路由来访问公司B的内部网络10.0.4.4/32，但是，当R2与Level-2路由器R3的邻接关系出现问题时，R2产生的Level-1 LSP中的ATT位将变为0，R1便因此不能生成缺省路由了，这就导致R1无法继续访问公司B的内部网络。为了解决这个问题，可以在R2上手动强制向R1发布缺省路由。

在R2的IS-IS视图下使用default-route-advertise命令强制R2发布缺省路由。

```
[R2]isis 1
```

```
[R2-isis-1]default-route-advertise
```

配置完成后，在R1上查看IS-IS路由表。

```
<R1>display isis route
```

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

10.0.1.1/32	0	NULL	Loop0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

结果发现，R1 的 IS-IS 路由表中还是没有缺省路由。原来，default-route-advertise命令在默认情况下只向Level-2邻接关系的路由器发布缺省路由，如果需要向Level-1邻接关系的路由器发布缺省路由，则需要使用default-route-advertise level-1 命令。

[R2]isis 1

[R2-isis-1]default-route-advertise level-1

在R1上查看IS-IS路由表。

<R1>display isis route

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table					

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

```

-----
-----
0.0.0.0/0      10      NULL  GE0/0/0      10.0.12.2    A/-/-/-
10.0.12.0/24   10      NULL  GE0/0/0      Direct      D/-/L/-
10.0.1.1/32    0       NULL  Loop0        Direct      D/-/L/-

```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

可以看到，R1的IS-IS路由表中现在拥有了一条下一跳为10.0.12.2的缺省路由，R1因此可以继续访问公司B的内部网络了。

接下来，关闭R2 的GE 0/0/2 接口，模拟R2 与公司B 之间的链路发生了故障。

```
[R2]interface GigabitEthernet 0/0/2
```

```
[R2-GigabitEthernet0/0/2]shutdown
```

查看R2的IP路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```

-----
-----
Routing Tables: Public

```

```
Destinations : 8      Routes : 8
```

```

Destination/Mask    Proto    Pre  Cost    Flags  NextHop
Interface

```

10.0.1.1/32	ISIS-L1	15	10	D	10.0.12.1	
GigabitEthernet0/0/0						
10.0.12.0/24	Direct	0	0	D	10.0.12.2	
GigabitEthernet0/0/0						
10.0.12.2/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.12.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

可以看到，R2的路由表中没有缺省路由存在。

然后在R1上查看IS-IS路由表。

```
<R1>display isis route
```

```
Route information for ISIS(1)
```

```
-----
```

```
ISIS(1) Level-1 Forwarding Table
```

```
-----
```

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags


```

-----
0.0.0.0/0      10    NULL  GE0/0/0      10.0.12.2    A/-/-/-
10.0.12.0/24   10    NULL  GE0/0/0      Direct      D/-/L/-
10.0.1.1/32    0     NULL  Loop0        Direct      D/-/L/-

```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

观察发现，R1的IS-IS路由表中依然拥有下一跳为10.0.12.2的缺省路由。原来，在默认情况下，路由器使用default-route-advertise命令发布缺省路由的原则是：无论自己的IP路由表中是否存在缺省路由，都会向建立了IS-IS邻接关系的路由器发布缺省路由。

现在，虽然R1拥有一条指向R2的缺省路由，但是R2上手动配置的静态缺省路由是没有进入R2的路由表的，所以此时公司A的内部网络10.0.1.1/32与公司B的内部网络10.0.4.4/32 仍是无法进行通信的。为了解决这个问题，可以在R2上使用default-route-advertise match default level-1命令使得R2只有在IP路由表中拥有缺省路由的情况下，才会向Level-1邻接关系的路由器发布缺省路由。

```
[R2]isis 1
```

```
[R2-isis-1]default-route-advertise match default level-1
```

在R1上查看IS-IS路由表。

```
<R1>display isis route
```

```
Route information for ISIS(1)
```

```

-----
ISIS(1) Level-1 Forwarding Table

```

```
-----
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
-----
```

```
10.0.12.0/24      10      NULL    GE0/0/0      Direct  D-/L/-
```

```
10.0.1.1/32      0       NULL    Loop0        Direct  D-/L/-
```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R1的IS-IS路由表中现在不再有缺省路由了。

当R2与R4之间的链路恢复正常后，查看R2的IP路由表。

```
[R2]interface GigabitEthernet 0/0/2
```

```
[R2-GigabitEthernet0/0/2]undo shutdown
```

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
Routing Tables: Public
```

```
Destinations : 12Routes : 12
```

```
Destination/Mask  Proto    Pre  Cost    Flags  NextHop
Interface
```

```
0.0.0.0/0        Static   60   0       RD    10.0.24.4  GigabitEthernet0/0/2
```

```
10.0.1.1/32      ISIS-L1  15  10    D    10.0.12.1
GigabitEthernet0/0/0
```

.....

可以看到，R2的路由表中有了手动配置的静态缺省路由信息。

查看R1的IS-IS路由表。

```
<R1>display isis route
```

```
Route information for ISIS(1)
```

```
-----
```

```
ISIS(1) Level-1 Forwarding Table
```

```
-----
```

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

```
-----
```

0.0.0.0/0	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

10.0.1.1/32	0	NULL	Loop0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R1接收了来自R2重新发布的缺省路由。

5．手动向R3发布缺省路由

目前，当R2和R3之间的链路出现故障时，R1仍然能够获得缺省路由来访问公司B的内部网络10.0.4.4/3，而当R2与R4之间的链路出现故障时，R1上将不再拥有没有意义的缺省路由。

为了在R3上也实现同样的效果，重新开启R2的GE 0/0/1接口。

```
[R2]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]undo shutdown
```

在R2的IS-IS视图下使用default-route-advertise match default level-1-2命令使得R2在拥有缺省路由时，同时向Level-1邻接的路由器和Level-2邻接的路由器发布缺省路由，也即同时向R1和R3发布缺省路由，以使公司A总部的内部网络和分支机构的内部网络都能访问公司B的内部网络。

```
[R2]isis 1
```

```
[R2-isis-1]default-route-advertise match default level-1-2
```

在R3上查看IS-IS路由表。

```
<R3>display isis route
```

Route information for ISIS(1)

ISIS(1) Level-2 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

0.0.0.0/0	10	NULL	GE0/0/0	10.0.23.2	A/-/-/-

10.0.3.3/32	0	NULL	Loop0	Direct	D/-/L/-
10.0.23.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-
10.0.12.0/24	20	NULL	GE0/0/0	10.0.23.2	A/-/-/-
10.0.1.1/32	20	NULL	GE0/0/0	10.0.23.2	A/-/-/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

可以看到，R3拥有了下一跳为R2（10.0.23.2）的缺省路由。

在R3上使用ping命令测试分支机构的内部网络10.0.3.3/32与公司B的内部网络10.0.4.4/32之间的连通性。

```
<R3>ping -c 1 -a 10.0.3.3 10.0.4.4
```

```
PING 10.0.4.4: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.4.4: bytes=56 Sequence=1 ttl=254 time=20 ms
```

```
--- 10.0.4.4 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/20/20 ms
```

可以看到，公司A的分支机构的内部网络与公司B的内部网络之间的通信是正常的。

接下来，关闭R2的GE 0/0/2 接口，模拟公司A 与公司B 之间的链路出现了故障。

```
[R2]interface GigabitEthernet 0/0/2
```

```
[R2-GigabitEthernet0/0/2]shutdown
```

查看R2的路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 12 Routes : 12

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.1
10.0.3.3/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.23.3
10.0.12.0/24 GigabitEthernet0/0/0	Direct	0	0	D	10.0.12.2
10.0.12.2/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.12.255/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.23.0/24 GigabitEthernet0/0/1	Direct	0	0	D	10.0.23.2

10.0.23.2/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.23.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

可以看到，R2的路由表中不再有缺省路由。

在R1上查看IS-IS路由表。

<R1>display isis route

Route information for ISIS(1)

ISIS(1) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

0.0.0.0/0	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

```
10.0.1.1/32    0    NULL  Loop0    Direct  D/-/L/-
```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R1的IS-IS路由表中仍然拥有下一跳为10.0.12.2的IS-IS缺省路由。

在R3上查看IS-IS路由表。

```
<R3>display isis route
```

Route information for ISIS(1)

ISIS(1) Level-2 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.3.3/32	0	NULL	Loop0	Direct	D/-/L/-

10.0.23.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

10.0.12.0/24	20	NULL	GE0/0/0	10.0.23.2	A/-/-/-

10.0.1.1/32	20	NULL	GE0/0/0	10.0.23.2	A/-/-/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R3的IS-IS路由表中是没有缺省路由的。

读者或许会觉得有些奇怪，既然R2的GE 0/0/2接口关闭了，R2 的路由表中不再有缺省路由，那么R2就不会向R1和R3发布缺省路由，但是为什么R3的IS-IS路由表中没有缺省路由，而R1的IS-IS路由表中却有缺省路由呢？答案是：R1是由于接收到R2的ATT 比特位置1 的Level-1 LSP后，自己生成了指向R2 的缺省路由。

在R1上使用ping命令测试公司A总部的内部网络10.0.1.1/32与分支机构的内部网络10.0.3.3/32的连通性。

```
<R1>ping -c 1 -a 10.0.1.1 10.0.3.3
```

```
PING 10.0.3.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.3.3: bytes=56 Sequence=1 ttl=254 time=20 ms
```

```
--- 10.0.3.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/20/20 ms
```

可以看到，通信正常。仔细分析会发现，在R1上，公司A总部的内部网络去往公司A的分支机构的内部网络使用的是缺省路由；在R3上，公司A的分支机构的内部网络去往公司A总部的内部网络使用的是明细路由。

重新启用R2的GE 0/0/2接口后，公司A 总部的内部网络和分支机构的内部网络都能通过IS-IS缺省路由与公司B的内部网络进行通信。至此，所有的网络需求都得到了实现。

思考

Level-1路由器会选择本区域中离它最近（Cost最小）的Level-1-2路由器作为本区域的出口路由器。如果Level-1路由器与两个Level-1-2路由器的距离相等（Cost相同），那该如何抉择呢？

4.8 IS-IS 路由引入

原理概述

IS-IS网络能够引入其他路由协议的路由和其他IS-IS协议进程的路由。默认情况下，IS-IS总是以Level-2路由类型引入外部路由。但是，通过手动配置，也可以以Level-1路由类型引入外部路由。IS-IS协议在引入外部路由时，可以手动配置引入路由的开销值，并可以使用Route-Policy对引入的路由进行过滤。

实验目的

- 掌握在IS-IS网络中引入外部路由的方法
- 理解Cost类型Internal和External的区别
- 掌握使用Route-Policy控制引入路由的方法

实验内容

实验拓扑如图4-8所示，实验编址如表4-8所示。本实验模拟了一个企业网络场景，B公司是A公司的业务合作伙伴，A公司在R1和R4上运行OSPF协议，其中R4的Loopback 0和Loopback 1接口代表了两个不同的业务网段。B公司在R1、R2、R3上运行IS-IS协议，R1和R2属于IS-IS区域10，R3属于IS-IS区域20，R1为Level-1路由器，R2为Level-1-2路由器，R3为Level-2路由器。R3的Loopback 0接口模拟了B公司的外部网络，所以该接口不要使能IS-IS。为了实现两个公司的业务对接，B公司决定在R1上引入A公司R4的Loopback 0这个业务网段的路由。注意，本实验并不涉及将IS-IS路由引入到OSPF网络中，感兴趣的读者可以自行去完成，以实现网络的互通。

实验拓扑

figure_0382_0119

图4-8 IS-IS路由引入

实验编址表

表4-8 实验编址

figure_0382_0120

(续表)

figure_0383_0121

实验步骤

1. 基本配置

根据图4-8和表4-8进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 60/60/60 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF和IS-IS路由协议

在R1和R4上配置OSPF协议，其中R1的Router-ID为10.0.1.1，R4的Router-ID为10.0.4.4。

```
[R1]router id 10.0.1.1
```

```
[R1]ospf 1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.14.0 0.0.0.255
```

```
[R4]router id 10.0.4.4
```

```
[R4]ospf 1
```

```
[R4-ospf-1]area 0
```

```
[R4-ospf-1-area-0.0.0.0]network 10.0.14.0 0.0.0.255
```

```
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
```

```
[R4-ospf-1-area-0.0.0.0]network 20.0.4.4 0.0.0.0
```

配置完成后，在R4上查看OSPF邻居信息。

```
[R4]display ospf peer
```

```
OSPF Process 1 with Router ID 10.0.4.4
```

```
Neighbors
```

```
Area 0.0.0.0 interface 10.0.14.4(GigabitEthernet0/0/0)'s neighbors
```

```
Router ID: 10.0.1.1      Address: 10.0.14.1
```

State: Full Mode:Nbr is Slave Priority: 1

DR: 10.0.14.4 BDR: 10.0.14.1 MTU: 0

Dead timer due in 36 sec

Retrans timer interval: 5

Neighbor is up for 00:02:44

Authentication Sequence: [0]

可以看到，R4与R1已经成功建立了OSPF邻接关系。

在R1、R2、R3上配置IS-IS协议，其中R1为Level-1路由器，R2为Level-1-2路由器，R3为Level-2路由器。需要注意的是，R3的Loopback 0接口模拟的是公司B的外部网络，不应进入IS-IS进程。

```
[R1]isis 10
```

```
[R1-isis-10]is-level level-1
```

```
[R1-isis-10]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-10]is-name R1
```

```
[R1-isis-10]quit
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable 10
```

```
[R2]isis 10
```

```
[R2-isis-10]network-entity 10.0000.0000.0002.00
```

```
[R2-isis-10]is-name R2
```

```
[R2-isis-10]quit
```

```
[R2]interface GigabitEthernet0/0/0
[R2-GigabitEthernet0/0/0]isis enable 10
[R2-GigabitEthernet0/0/0]interface GigabitEthernet0/0/1
[R2-GigabitEthernet0/0/1]isis enable 10
[R3]isis 10
[R3-isis-10]is-level level-2
[R3-isis-10]network-entity 20.0000.0000.0003.00
[R3-isis-10]is-name R3
[R3-isis-10]quit
[R3]interface GigabitEthernet 0/0/1
[R3-GigabitEthernet0/0/1]isis enable 10
```

3 . 引入外部直连路由

在R3 上引入直连的Loopback 0 接口的路由。

```
[R3]isis 10
[R3-isis-10]import-route direct
```

在R3上查看IS-IS路由表。

```
<R3>display isis route
```

```
Route information for ISIS(10)
```

```
-----
```

```
ISIS(10) Level-2 Forwarding Table
```

```

-----
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
-----

```

```

10.0.23.0/24      10       NULL    GE0/0/1       Direct   D-/L/-
10.0.12.0/24      20       NULL    GE0/0/1       10.0.23.2  A/-/-/-

```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

ISIS(10) Level-2 Redistribute Table

```

-----
Type      IPV4 Destination      IntCost  ExtCostTag
-----
-----
D         10.0.3.3/32             0        0

```

Type: D-Direct, I-ISIS, S-Static, O-OSPF, B-BGP, R-RIP, U-UNR

可以看到，R3 拥有IS-IS Level-2 路由表，同时还维护了Level-2 重分发
表，用以保存引入的外部路由。默认情况下，IS-IS以Level-2路由类型
来引入外部路由。10.0.23.0/24和10.0.12.0/24是两条IS-IS内部路由，它
们的IntCost参数值代表了这两条路由在IS-IS路由表中的Cost值，也代
表在IP路由表中的Cost值；对于IS-IS内部路由来说，ExtCost的值始终
为NULL。对于外部路由10.0.3.3/32来说，IntCost的值代表了这条路由
以Internal开销类型引入进 IS-IS 时配置的 Cost 值，而 ExtCost 的值则代
表了这条路由以 External开销类型引入进IS-IS时配置的Cost值。默认情
况下，被引入进IS-IS的路由的IntCost和ExtCost的值均为0。另外需要注
意的是，默认情况下，外部路由被引入IS-IS时，开销类型为External。

在R2上查看IS-IS路由表。

```
<R2>display isis route
```

Route information for ISIS(10)

ISIS(10) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

ISIS(10) Level-2 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.3.3/32	10	0	GE0/0/1	10.0.23.3	A/-/-/-

10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，R2的Level-2路由表中已经有了关于10.0.3.3/32的路由，但R2并没有维护Level-2重分发表，这是因为IS-IS重分发表只存在于引入IS-IS外部路由的路由器上，被引入的外部路由在发布给其他IS-IS路由器时与IS-IS内部路由没有什么区别。另外还可以发现，10.0.3.3/32这条路由的IntCost值为10，正好是R2与R3之间链路的开销值，ExtCost的值依然还是引入时的默认值0。

在R2上查看IP路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 12 Routes : 12

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.3.3/32 GigabitEthernet0/0/1	ISIS-L2	15	74	D	10.0.23.3
10.0.12.0/24	Direct	0	0	D	10.0.12.2 GigabitEthernet0/0/0
.....					

可以看到，R2的IP路由表中的10.0.3.3/32这条路由的Cost值为74。注意，IS-IS在引入外部路由时的Cost分为External和Internal两种不同的类型，Cost类型不同，在IP路由表中计算Cost值的方法也不同。10.0.3.3/32这条路由被引入IS-IS时的Cost类型是默认类型External。对于External类型的Cost，在IP路由表中的Cost值的计算方法为：64 + IS-

IS 路由表中的 IntCost + IS-IS 路由表中的 ExtCost。所以，R2 的 IP 路由表中的 10.0.3.3/32 这条路由的 Cost 值应该为：64 + 10 + 0 = 74。

在 R1 上查看 IS-IS 路由表。

```
<R1>display isis route
```

Route information for ISIS(10)

ISIS(10) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

0.0.0.0/0	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，Level-1 路由器 R1 只维护 Level-1 路由表，且会使用缺省路由经 Level-1-2 路由器 R2 来访问 10.0.3.3/32。

在 R1 上使用 ping 命令测试与 10.0.3.3/32 间的连通性。

```
<R1>ping -c 1 10.0.3.3
```

PING 10.0.3.3: 56 data bytes, press CTRL_C to break

Reply from 10.0.3.3: bytes=56 Sequence=1 ttl=254 time=30 ms

--- 10.0.3.3 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 30/30/30 ms

可以看到，现在整个 IS-IS 网络都可以访问 R3 的 Loopback 0 接口所模拟的外部网络了。

4 . 引入外部OSPF路由

在R1上以Cost类型为Internal的方式将OSPF协议引入到IS-IS协议中，引入时的IntCost值根据公司要求配置为30。

```
[R1]isis 10
```

```
[R1-isis-10]import-route ospf 1 cost-type internal cost 30
```

在R1上查看IS-IS路由表。

```
[R1]display isis route
```

Route information for ISIS(10)

ISIS(10) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

```

-----
0.0.0.0/0      10    NULL  GE0/0/0      10.0.12.2  A/-/-/-
10.0.23.0/24   20    NULL  GE0/0/0      10.0.12.2  A/-/-/-
10.0.12.0/24   10    NULL  GE0/0/0      Direct  D/-/L/-

```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

可以看到，关于10.0.4.4/32和20.0.4.4/32的路由并未进入R1的IS-IS路由表。原来，默认情况下，IS-IS协议总是以Level-2路由类型来引入外部路由的，而R1是Level-1路由器，只维护Level-1路由表。接下来，手动指定以Level-1路由类型引入OSPF的路由。

```
[R1]isis 10
```

```
[R1-isis-10]import-route ospf 1 level-1 cost-type internal cost 30
```

在R1上查看IS-IS路由表。

```
<R1>display isis route
```

```
Route information for ISIS(10)
```

```
-----
ISIS(10) Level-1 Forwarding Table
```

```
-----
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
-----
```

0.0.0.0/0	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

ISIS(10) Level-1 Redistribute Table

```

-----
Type          IPV4 Destination  IntCost  ExtCost Tag
-----
D      10.0.14.0/24      30      NULL
O      10.0.4.4/32       30      NULL
O      20.0.4.4/32       30      NULL

```

Type: D-Direct, I-ISIS, S-Static, O-OSPF, B-BGP, R-RIP, U-UNR

可以看到，OSPF网络中的所有路由已进入R1的Level-1重分发表中，并且IntCost值为30，而ExtCost值为NULL。

在R2上查看IS-IS路由表。

<R2>display isis route

Route information for ISIS(10)

```

-----
ISIS(10) Level-1 Forwarding Table

```

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.14.0/24	40	NULL	GE0/0/0	10.0.12.1	A/-/L/-
10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-
10.0.4.4/32	40	NULL	GE0/0/0	10.0.12.1	A/-/L/-
20.0.4.4/32	40	NULL	GE0/0/0	10.0.12.1	A/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

ISIS(10) Level-2 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.3.3/32	10	0	GE0/0/1	10.0.23.3	A/-/-/-
10.0.23.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

可以看到，在R2的IS-IS路由表中，关于10.0.14.0/24、10.0.4.4/32和20.0.4.4/32的路由进入了Level-1路由表，没有进入Level-2路由表。

查看R2的IP路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 14 Routes : 14

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.3.3/32 GigabitEthernet0/0/1	ISIS-L2	15	74	D	10.0.23.3
10.0.4.4/32 GigabitEthernet0/0/0	ISIS-L1	15	40	D	10.0.12.1
10.0.12.0/24 GigabitEthernet0/0/0	Direct	0	0	D	10.0.12.2
10.0.12.2/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.12.255/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.14.0/24 GigabitEthernet0/0/0	ISIS-L1	1	40	D	10.0.12.1
10.0.23.0/24 GigabitEthernet0/0/1	Direct	0	0	D	10.0.23.2

```

10.0.23.2/32    Direct    0   0   D    127.0.0.1
GigabitEthernet0/0/1

10.0.23.255/32  Direct    0   0   D    127.0.0.1
GigabitEthernet0/0/1

20.0.4.4/32     ISIS-L1   15  40   D    10.0.12.1
GigabitEthernet0/0/0

127.0.0.0/8     Direct    0   0   D    127.0.0.1    InLoopBack0

127.0.0.1/32    Direct    0   0   D    127.0.0.1    InLoopBack0

127.255.255.255/32 Direct    0   0   D    127.0.0.1
InLoopBack0

255.255.255.255/32 Direct    0   0   D    127.0.0.1
InLoopBack0

```

可以看到，R2的IP路由表中已经有了所有的OSPF外部路由。另外，这些外部路由是以Internal开销类型被引入的，引入时它们的IntCost被配置为30，所以在R2的IS-IS路由表中它们的IntCost的值为40（30加上R1与R2的链路开销值10），因此，在R2的IP路由表中，它们的Cost的值也都是40。

5．使用Route-Policy控制路由的引入

通过前面的实验步骤，所有的OSPF路由都已经被引入进IS-IS网络了。然而，真正的需求是：只能引入 R4 的 Loopback 0 接口的路由，即 10.0.4.4/32。为此，可以使用Route-Policy来对引入的路由进行控制。

在R1上配置Route-Policy。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]rule permit source 10.0.4.4 0
```

```
[R1-acl-basic-2000]route-policy 1 permit node 10
```



```
[R1-route-policy]if-match acl 2000
```

在R1上配置引入OSPF路由时调用Route-Policy。

```
[R1]isis 10
```

```
[R1-isis-10]import-route ospf 1 route-policy 1 cost 30 cost-type internal  
level-1
```

在R1上查看IS-IS路由表。

```
<R1>display isis route
```

Route information for ISIS(10)

ISIS(10) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

0.0.0.0/0	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-

10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

ISIS(10) Level-1 Redistribute Table

Type	PV4 Destination	IntCost	ExtCost	Tag

O	10.0.4.4/32	30	NULL	

Type: D-Direct, I-ISIS, S-Static, O-OSPF, B-BGP, R-RIP, U-UNR

可以看到，R1的Level-1重分发表中只有关于10.0.4.4/32的路由，这就说明需求已经得到了实现。

思考

IS-IS路由的最大Cost值默认是多少？

4.9 IS-IS 路由过滤

原理概述

在IS-IS网络中，有的时候需要使用Filter-Policy工具来对IS-IS路由进行过滤。这里所说的过滤，是指路由器在将自己的IS-IS路由表中的某些IS-IS路由纳入进自己的IP路由表的过程，一些满足了过滤条件的IS-IS路由将被限制纳入进IP路由表中。

需要注意的是，Filter-Policy 进行过滤的并非是生成那些 IS-IS 路由的 LSP，所以Filter-Ploicy进行路由过滤之后，路由器中的IS-IS链路状态数据库和IS-IS路由表都不会受到任何影响。

实验目的

- 理解IS-IS 路由过滤的工作原理
- 掌握配合使用Filter-Policy和Route-Policy实现IS-IS 路由过滤的方法

实验内容

实验拓扑如图4-9所示，实验编址如表4-9所示。本实验模拟了一个企业网络场景，所有路由器都运行 IS-IS 协议，且都为 Level-2 路由器。

R1、R2、R4 属于区域 10，R3属于区域20。R1、R2、R4 的Loopback 0 接口用来模拟不同的内部网络，R3 的Loopback 1 和Loopback 2 接口用来模拟两个不同的业务网段。网络需求是：R1 的Loopback 0访问R3的 Loopback 1 业务网段的流量由R2 转发，如果R1与R2 之间的链路出现了故障，则由R4 转发；R1 的Loopback 0 访问Loopback 2业务网段的流量由R4 转发，如果R1与R4之间的链路出现了故障，则由R2转发。

实验拓扑

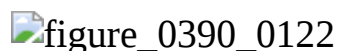
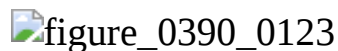


图4-9 IS-IS路由过滤

实验编址表

表4-9 实验编址



实验步骤

1．基本配置

根据图4-9和表4-9进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=1310 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 1310/1310/1310 ms

其余直连网段的连通性测试过程在此省略。

2 . 配置IS-IS路由协议

在每台路由器上配置IS-IS协议。注意，所有的路由器均为Level-2路由器。

```
[R1]isis 10
```

```
[R1-isis-10]is-level level-2
```

```
[R1-isis-10]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-10]is-name R1
```

```
[R1-isis-10]quit
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable 10
```

```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]isis enable 10
```

```
[R1-GigabitEthernet0/0/1]interface LoopBack 0
```

```
[R1-LoopBack0]isis enable 10
```

```
[R2]isis 10
```

```
[R2-isis-10]is-level level-2
```

```
[R2-isis-10]network-entity 10.0000.0000.0002.00
```

```
[R2-isis-10]is-name R2
```

```
[R2-isis-10]quit
```

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis enable 10
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]isis enable 10
```

```
[R2-GigabitEthernet0/0/1]interface LoopBack 0
```

```
[R2-LoopBack0]isis enable 10
```

```
[R3]isis 10
```

```
[R3-isis-10]is-level level-2
```

```
[R3-isis-10]network-entity 20.0000.0000.0003.00
```

```
[R3-isis-10]is-name R3
```

```
[R3-isis-10]quit
```

```
[R3]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]isis enable 10
```

```
[R3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[R3-GigabitEthernet0/0/2]isis enable 10
```

```
[R3-GigabitEthernet0/0/2]interface LoopBack 1
```

```
[R3-LoopBack1]isis enable 10
```

```
[R3-LoopBack1]interface LoopBack 2
```

```
[R3-LoopBack2]isis enable 10

[R4]isis 10

[R4-isis-10]is-level level-2

[R4-isis-10]network-entity 10.0000.0000.0004.00

[R4-isis-10]is-name R4

[R4-isis-10]quit

[R4]interface GigabitEthernet 0/0/1

[R4-GigabitEthernet0/0/1]isis enable 10

[R4-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2

[R4-GigabitEthernet0/0/2]isis enable 10

[R4-GigabitEthernet0/0/2]interface LoopBack 0

[R4-LoopBack0]isis enable 10
```

配置完成后，在R1上查看IS-IS路由表。

```
[R1]display isis route
```

```
Route information for ISIS(10)
```

```
-----
```

```
ISIS(10) Level-2 Forwarding Table
```

```
-----
```

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.14.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-
10.0.4.4/32	10	NULL	GE0/0/1	10.0.14.4	A/-/-/-
10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
172.16.2.1/32	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
GE0/0/1	10.0.14.4				
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-
10.0.2.2/32	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-
172.16.1.1/32	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
GE0/0/1	10.0.14.4				
10.0.1.1/32	0	NULL	Loop0	Direct	D/-/L/-
10.0.34.0/24	20	NULL	GE0/0/1	10.0.14.4	A/-/-/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

可以看到，R1 的 IS-IS 路由表中拥有去往区域 20 的业务网段 172.16.1.1/32 和 172.16.2.1/32，以及去往区域 10 的内部网络 10.0.2.2/32 和 10.0.4.4/32 的路由。

在 R1 上查看 IP 路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----
-----
```

Routing Tables: Public

Destinations : 17 Routes : 19

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					
127.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1
172.16.1.1/32 GigabitEthernet0/0/0	ISIS-L2	15	20	D	10.0.12.2
ISIS-L2 15 20	D	10.0.14.4	GigabitEthernet0/0/1		
172.16.2.1/32 GigabitEthernet0/0/0	ISIS-L2	15	20	D	10.0.12.2
ISIS-L2 15 20	D	10.0.14.4	GigabitEthernet0/0/1		
255.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1

观察发现，R1的IP路由表中去往172.16.1.1/32和172.16.2.1/32的路由来自于IS-IS路由表，并且采用了负载均衡方式。

查看R2的IP路由表。

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 19 Routes : 20

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/0	ISIS-L2	15	10	D	10.0.12.1
.....					
127.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1
172.16.1.1/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.23.3
172.16.2.1/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.23.3
255.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1

可以看到，R2的IP路由表中拥有去往172.16.1.1/32和172.16.2.1/32的路由，使用的也是IS-IS路由。

查看R4的IP路由表。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 18

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.14.1
.....					
127.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1
172.16.1.1/32 GigabitEthernet0/0/2	ISIS-L2	15	10	D	10.0.34.3
172.16.2.1/32 GigabitEthernet0/0/2	ISIS-L2	15	10	D	10.0.34.3
255.255.255.255/32 InLoopBack0	Direct	0	0	D	127.0.0.1

可以看到，R4的IP路由表中拥有去往172.16.1.1/32和172.16.2.1/32的路由，使用的也是IS-IS路由。

3．使用Filter-Policy实现路由过滤

目前的状态是，R1 上的内部网络Loopback 0 可以经由R2和R4 以负载均衡的方式访问 R3 上的业务网段 Loopback 1，同时可以经由 R2 和 R4 以负载均衡的方式访问 R3上的业务网段Loopback 2。而网络需求是：R1 上的内部网络Loopback 0只能经由R2访问R3 上的业务网段Loopback 1，同时只能经由R4访问R3 上的业务网段Loopback 2。为此，可以使用Filter-Policy来实现这样的需求。

在R2 上使用Filter-Policy过滤掉去往R3 上的Loopback 2（172.16.2.1/32）这个业务网段的路由。

```
[R2]acl 2000
```

```
[R2-acl-basic-2000]rule deny source 172.16.2.1 0
```

```
[R2-acl-basic-2000]rule permit source any
```

```
[R2-acl-basic-2000]isis 10
```

```
[R2-isis-10]filter-policy 2000 import
```

配置完成后，查看R2的IP路由表。

```
<R2>display ip routing-table
```

Route Flags:R - relay,D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 16 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/0	ISIS-L2	15	10	D	10.0.12.1
10.0.2.2/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.4.4/32 GigabitEthernet0/0/0	ISIS-L2	15	20	D	10.0.12.1
ISIS-L2	15	20	D	10.0.23.3	GigabitEthernet0/0/1
10.0.12.0/24 GigabitEthernet0/0/0	Direct	0	0	D	10.0.12.2
10.0.12.2/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.12.255/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1

10.0.14.0/24		ISIS-L2	15	20	D	10.0.12.1
GigabitEthernet0/0/0						
10.0.23.0/24	Direct	0	0	D	10.0.23.2	
GigabitEthernet0/0/1						
10.0.23.2/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.23.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.34.0/24		ISIS-L2	15	20	D	10.0.23.3
GigabitEthernet0/0/1						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
172.16.1.1/32		ISIS-L2	15	10	D	10.0.23.3
GigabitEthernet0/0/1						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

可以看到，R2的IP路由表中去往172.16.2.1/32的路由条目已经消失了。

查看R2的IS-IS路由表。

<R2>display isis route

Route information for ISIS(10)

ISIS(10) Level-2 Forwarding Table

```
-----
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
-----
```

```
-----
10.0.14.0/24      20       NULL    GE0/0/0        10.0.12.1  A/-/-/-
10.0.23.0/24      10       NULL    GE0/0/1        Direct    D/-/L/-
172.16.2.1/32     10       NULL    GE0/0/1        10.0.23.3  -/-/-/-
172.16.1.1/32     10       NULL    GE0/0/1        10.0.23.3  A/-/-/-
```

.....

可以看到，R2的IS-IS路由表中关于172.16.2.1/32的这个路由条目依然存在，并没有被过滤掉。

在R2上查看R3产生的LSP的详细信息。

[R2]display isis lsdb is-name R3 verbose

Database information for ISIS(10)

```
-----
Level-2 Link State Database
```

```
LSPID          Seq Num      Checksum    Holdtime    Length    ATT/P/OL
-----
```

```
-----
0000.0000.0003.00-00  0x000000013  0x2409    1002      131      0/0/0
```

```
SOURCE      R3.00
```

```
HOST NAME    R3
```

```

NLPID      IPv4
AREA ADDR  20
INTF ADDR  10.0.23.3
INTF ADDR  10.0.34.3
INTF ADDR  172.16.1.1
INTF ADDR  172.16.2.1

NBR  ID      R3.01          COST: 10
NBR  ID      R3.02          COST: 10

IP-Internal  10.0.23.0  255.255.255.0    COST: 10
IP-Internal  10.0.34.0  255.255.255.0    COST: 10
IP-Internal  172.16.1.0  255.255.255.255    COST: 0
IP-Internal  172.16.2.0  255.255.255.255    COST: 0

0000.0000.0003.01-00  0x00000007    0xbdc3  1002    55    0/0/0
.....

```

可以看到，R3产生的关于172.16.2.1/32的LSP仍然是保存在R2的LSDB中的，并没有被过滤掉。

对于IS-IS来说，Filter-Policy过滤工具实现的是对IS-IS路由表的过滤，过滤掉的路由条目是不能进入IP路由表的(请特别注意，报文的实际转发是基于IP路由表的)，但并不会对生成这些路由的LSP进行过滤。一个基于链路状态的路由协议，必须保证同一个区域中的所有路由器上的链路状态数据库信息保持一致。例如，对于本实验来说，如果R2上对某条LSP进行了过滤，同区域的其他路由器上还是会存在这条LSP，这就将导致链路状态数据库信息不一致，违背了基于链路状态的路由

协议的基本规则。所以，我们看到在R2的IS-IS路由表中以及R2的链路状态数据中仍然存在关于172.16.2.1/32这条路由的信息。

在R4 上使用Filter-Policy过滤掉去往R3上Loopback 1 (172.16.1.1/32) 这个业务网段的路由。

```
[R4]acl 2000

[R4-acl-basic-2000]rule deny source 172.16.1.1 0

[R4-acl-basic-2000]rule permit source any

[R4-acl-basic-2000]isis 10

[R4-isis-10]filter-policy 2000 import
```

配置完成后，查看R4的IP路由表。

```
[R4]display ip routing-table

Route Flags: R - relay, D - download to fib
```

Routing Tables: Public

Destinations : 16 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.14.1
10.0.2.2/32 GigabitEthernet0/0/1	ISIS-L2	15	20	D	10.0.14.1
ISIS-L2 15 20	D	10.0.34.3	GigabitEthernet0/0/2		

10.0.4.4/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.12.0/24		ISIS-L2	15	20	D	10.0.14.1
GigabitEthernet0/0/1						
10.0.14.0/24	Direct	0	0	D	10.0.14.4	
GigabitEthernet0/0/1						
10.0.14.4/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.14.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						
10.0.23.0/24		ISIS-L2	15	20	D	10.0.34.3
GigabitEthernet0/0/2						
10.0.34.0/24	Direct	0	0	D	10.0.34.4	
GigabitEthernet0/0/2						
10.0.34.4/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/2						
10.0.34.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/2						
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						
172.16.2.1/32		ISIS-L2	15	10	D	10.0.34.3
GigabitEthernet0/0/2						
255.255.255.255/32	Direct	0	0	D	127.0.0.1	
InLoopBack0						

可以看到，R4的IP路由表中已经没有了关于172.16.1.1/32的路由条目。

4．配合使用Filter-Policy和Route-Policy

观察R1的IP路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 19

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					
127.255.255.255/32	Direct	0	0	D	127.0.0.1 InLoopBack0
172.16.1.1/32	ISIS-L2	15	20	D	10.0.12.2 GigabitEthernet0/0/0
ISIS-L2	15	20	D	10.0.14.4	GigabitEthernet0/0/1
172.16.2.1/32	ISIS-L2	15	20	D	10.0.12.2 GigabitEthernet0/0/0
ISIS-L2	15	20	D	10.0.14.4	GigabitEthernet0/0/1
255.255.255.255/32	Direct	0	0	D	127.0.0.1 InLoopBack0

观察发现，R1的IP路由表中去往172.16.1.1/32和172.16.2.1/32的路由都有两个下一跳，分别是R2（10.0.12.2）和R4（10.0.14.4），它们都来自IS-IS路由表。

查看R1的IS-IS路由表。

```
[R1]display isis route
```

```
Route information for ISIS(10)
```

```
-----
```

```
ISIS(10) Level-2 Forwarding Table
```

```
-----
```

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

```
-----
```

```
-----
```

10.0.14.0/24	10	NULL	GE0/0/1	Direct	D/-/L/-

10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-

172.16.2.1/32	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-

GE0/0/	10.0.14.4				

10.0.2.2/32	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-

172.16.1.1/32	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-

GE0/0/1	10.0.14.4				

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

10.0.1.1/32	0	NULL	Loop0	Direct	D/-/L/-

10.0.4.4/32	10	NULL	GE0/0/1	10.0.14.4	A/-/-/-
10.0.34.0/24	20	NULL	GE0/0/1	10.0.14.4	A/-/-/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
U-Up/Down Bit Set

可以看到，R1的IS-IS路由表中去往172.16.1.1/32和172.16.2.1/32的路由均有两个下一跳，分别是R2（10.0.12.2）和R4（10.0.14.4）。

从上面的实验可知，R2和R4虽然过滤掉了关于172.16.1.1/32和172.16.2.1/32这两条路由条目，但是R2和R4仍然会将产生这两条路由的LSP继续进行泛洪，其结果是，R1上的内部网络访问R3的两个业务网段的报文依然会在R1上使用负载均衡的方式来进行转发。

由于网络需求是：R1 上的内部网络Loopback 0只能经由R2访问R3 上的业务网段Loopback 1，同时只能经由R4 访问R3 上的业务网段Loopback 2，所以在R1 上进行路由过滤时，不能只考虑路由的前缀，同时还必须考虑到路由的下一跳。为此，可以配合使用Filter-Policy和Route-Policy来实现这种比较特别的路由过滤。

定义ACL 2000，匹配路由下一跳10.0.12.2；定义ACL 2001，匹配路由下一跳10.0.14.4。

```
[R1]acl 2000
```

```
[R1-acl-basic-2000]rule permit source 10.0.12.2 0
```

```
[R1-acl-basic-2000]acl 2001
```

```
[R1-acl-basic-2001]rule permit source 10.0.14.4 0
```

定义 ACL 2002，匹配路由前缀 172.16.1.1/32；定义 ACL 2003，匹配路由前缀172.16.2.1/32。

```
[R1]acl 2002
```

```
[R1-acl-basic-2002]rule permit source 172.16.1.1 0
```

```
[R1-acl-basic-2002]acl 2003
```

```
[R1-acl-basic-2003]rule permit source 172.16.2.1 0
```

定义如下的Route-Policy。

```
[R1]route-policy isis-filter deny node 10
```

```
[R1-route-policy]if-match ip next-hop acl 2000
```

```
[R1-route-policy]if-match acl 2003
```

```
[R1-route-policy]route-policy isis-filter deny node 20
```

```
[R1-route-policy]if-match ip next-hop acl 2001
```

```
[R1-route-policy]if-match acl 2002
```

```
[R1-route-policy]route-policy isis-filter permit node 30
```

在R1的IS-IS进程视图下配合使用Filter-Policy和Route-Policy。

```
[R1]isis 10
```

```
[R1-isis-10]filter-policy route-policy isis-filter import
```

配置完成后，查看R1的路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 17 Routes : 17

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
10.0.2.2/32 GigabitEthernet0/0/0	ISIS-L2	15	10	D	10.0.12.2
10.0.4.4/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.14.4
10.0.12.0/24 GigabitEthernet0/0/0	Direct	0	0	D	10.0.12.1
10.0.12.1/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.12.255/32 GigabitEthernet0/0/0	Direct	0	0	D	127.0.0.1
10.0.14.0/24 GigabitEthernet0/0/1	Direct	0	0	D	10.0.14.1
10.0.14.1/32 GigabitEthernet0/0/1	Direct	0	0	D	127.0.0.1
10.0.14.255/32 GigabitEthernet0/0/1	Direct	0	0	D	127.0.0.1
10.0.23.0/24 GigabitEthernet0/0/0	ISIS-L2	15	20	D	10.0.12.2
10.0.34.0/24 GigabitEthernet0/0/1	ISIS-L2	15	20	D	10.0.14.4
127.0.0.0/8	Direct	0	0	D	127.0.0.1 InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1 InLoopBack0

```

127.255.255.255/32  Direct    0  0    D    127.0.0.1
InLoopBack0

172.16.1.1/32      ISIS-L2  15  20    D    10.0.12.2
GigabitEthernet0/0/0

172.16.2.1/32      ISIS-L2  15  20    D    10.0.14.4
GigabitEthernet0/0/1

255.255.255.255/32  Direct    0  0    D    127.0.0.1
InLoopBack0

```

现在，R1的IP路由表中关于172.16.1.1/32这条路由的下一跳为R2（10.0.12.2），关于172.16.2.1/32这条路由的下一跳为R4（10.0.14.4）。另外，R2、R4的Loopback 0接口所在网段的路由信息也在R1的IP路由表中。

查看R1的IS-IS路由表。

```
[R1]display isis route
```

```
Route information for ISIS(10)
```

```
-----
```

```
ISIS(10) Level-2 Forwarding Table
```

```
-----
```

```
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
```

```
-----
-----
```

```

10.0.14.0/24      10       NULL    GE0/0/1       Direct   D/-/L/-
10.0.23.0/24      20       NULL    GE0/0/0       10.0.12.2 A/-/-/-
172.16.2.1/32     20       NULL    GE0/0/0       10.0.12.2 A/-/-/-

```

```

GE0/0/1      10.0.14.4
10.0.2.2/32   10    NULL  GE0/0/0      10.0.12.2  A/-/-/-
172.16.1.1/32  20    NULL  GE0/0/0      10.0.12.2  A/-/-/-
GE0/0/1      10.0.14.4
10.0.12.0/24   10    NULL  GE0/0/0      Direct  D-/L/-
10.0.1.1/32    0     NULL  Loop0        Direct  D-/L/-
10.0.4.4/32    10    NULL  GE0/0/1      10.0.14.4  A/-/-/-
10.0.34.0/24   20    NULL  GE0/0/1      10.0.14.4  A/-/-/-

```

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

观察发现，R1的IS-IS路由表中仍然存在有关172.16.1.1/32、172.16.2.1/32、10.0.2.2/32和10.0.4.4/32的路由信息，其中去往172.16.1.1/32和172.16.2.1/32的路由仍然还是负载均衡的方式。

目前，业务分流的需求已经得到了实现，即R1去往172.16.1.1/32时的下一跳是R2，R1去往172.16.2.1/32时的下一跳是R4。然而，有这样一个问题存在，就是R1与R2或R4的链路出现故障时，相应的业务会中断。为了解决这个问题，可以在R1上配置两条缺省路由，分别把下一跳指向R2和R4；在R2、R4上分别配置一条缺省路由，下一跳都指向R3。这样一来，一旦R1与R2或R4的链路出现故障时，缺省路由就能够保证业务流量不会中断。

```
[R1]ip route-static 0.0.0.0 0.0.0.0 10.0.12.2
```

```
[R1]ip route-static 0.0.0.0 0.0.0.0 10.0.14.4
```

```
[R2]ip route-static 0.0.0.0 0.0.0.0 10.0.23.3
```

```
[R4]ip route-static 0.0.0.0 0.0.0.0 10.0.34.3
```

至此，网络需求得以完全实现。

思考

本实验中，Filter-policy都使用在Import方向。对于IS-IS协议来说，Filter-Policy能使用在Export方向吗？为什么？

4.10 IS-IS 路由渗透

原理概述

在IS-IS网络中，所有的Level-2和Level-1-2路由器构成了一个连续的骨干区域。Level-1区域必须且只能与骨干区域相连，不同的Level-1区域之间不能直接相连。Level-1区域内的路由信息会通过Level-1-2路由器通报给Level-2区域，即Level-1-2路由器会将学习到的Level-1 路由信息封装进Level-2 LSP，并将此Level-2 LSP 传递给其他Level-2和Level-1-2路由器。因此，Level-1-2和Level-2路由器是知道整个IS-IS路由域（IS-IS Routing Domain）的路由信息的。另一方面，为了减小路由表的规模，在缺省情况下，Level-2和Level-1-2路由器并不会将自己知道的路由域中其他Level-1区域以及骨干区域的路由信息通报给Level-1区域。这样一来，Level-1路由器只能通过缺省路由来访问本区域以外的任何目的地。

通常情况下，Level-1路由器只通过缺省路由来访问本区域以外的目的地，但是如果需求特殊，则这种方式或许不能被接受。例如，如果要求一个Level-1 路由器必须经由最优路经（也就是总的开销值最小）访问其他某个区域的目的地时，使用缺省路由就很可能无法满足需求。在这种情况下，Level-1路由器需要知道并使用其他区域中的目的地的明细路由，而不是盲目地使用缺省路由。

IS-IS路由渗透指的是Level-1-2和Level-2路由器将自己知道的其他Level-1区域以及Level-2区域的路由信息通报给指定的Level-1区域的过程。在

这个过程中，还可以利用ACL、路由策略、Tag标记等方式把需要渗透的路由筛选出来，实现精细化路由渗透。

实验目的

- 理解IS-IS 路由渗透的概念和工作机制
- 掌握在IS-IS 网络中控制路由渗透的方法

实验内容

实验拓扑如图4-10所示，实验编址如表4-10所示。本实验模拟了一个企业网络场景，所有路由器都运行IS-IS协议，其中R1为Level-1路由器，R2和R3为Level-1-2路由器，R4为Level-2 路由器。R1和R4使用了Loopback 0接口来模拟内部网络。网络需求是：首先实现R1和R4的Loopback 0接口的互通，然后通过修改接口开销值的方法让R1选择一条次优路径去往R4，最后通过配置路由渗透的方法让R1选择一条最优路径去往R4。

实验拓扑

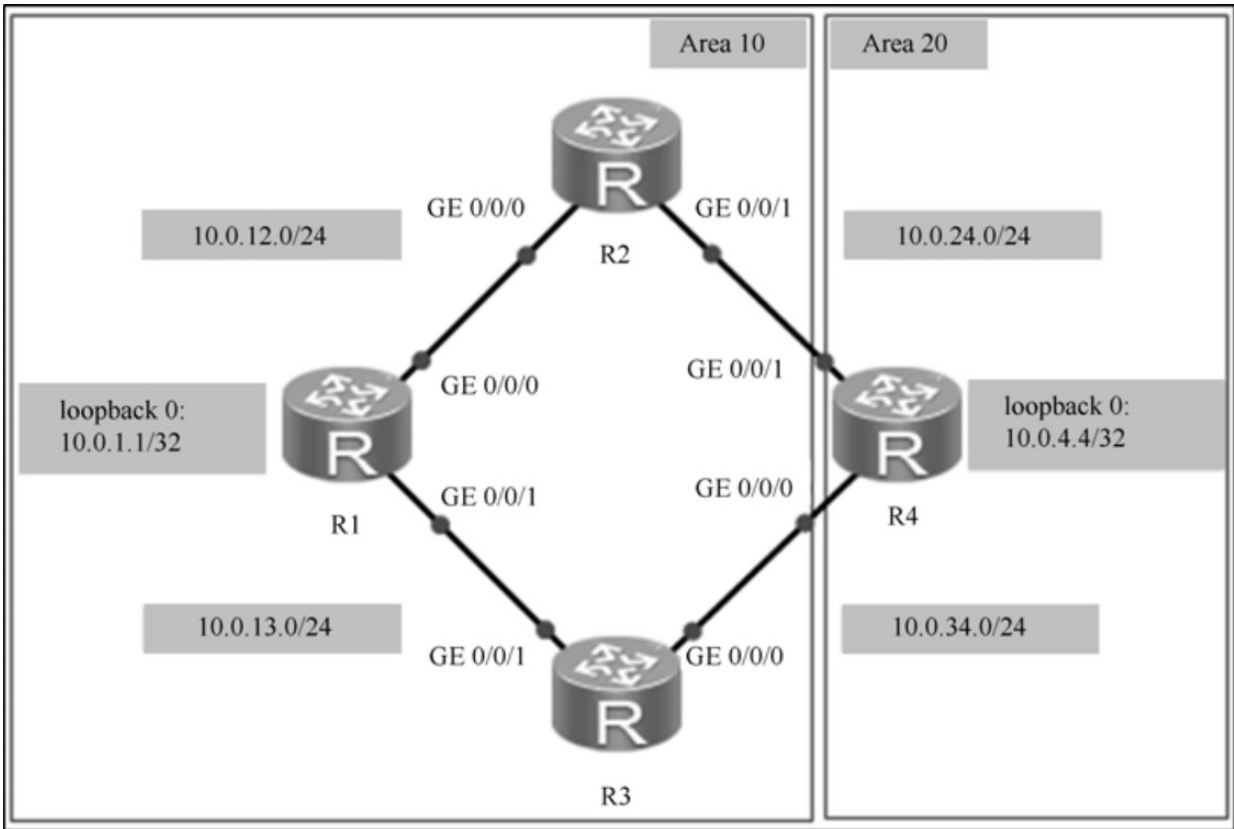


图4-10 IS-IS路由渗透

实验编址表

表4-10 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	NET: 10.0000.0000.0001.00			
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.24.2	255.255.255.0	N/A
	NET: 10.0000.0000.0002.00			
R3(AR2220)	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	NET: 10.0000.0000.0003.00			
R4(AR2220)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
	NET: 20.0000.0000.0004.00			

实验步骤

1．基本配置

根据图4-10和表4-10进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=80 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 80/80/80 ms
```

其余直连网段的连通性测试过程在此省略。

2 . 配置IS-IS路由协议

在每台路由器上配置IS-IS协议，其中R1为Level-1路由器，R2和R3为Level-1-2路由器，R4为Level-2路由器。

```
[R1]isis
```

```
[R1-isis-1]network-entity 10.0000.0000.0001.00
```

```
[R1-isis-1]is-level level-1
```

```
[R2]isis
```

```
[R2-isis-1]network-entity 10.0000.0000.0002.00
```

```
[R3]isis
```

```
[R3-isis-1]network-entity 10.0000.0000.0003.00
```

```
[R4]isis
```

```
[R4-isis-1]network-entity 20.0000.0000.0004.00
```

```
[R4-isis-1]is-level level-2
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable
```

```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]isis enable
```

```
[R1-GigabitEthernet0/0/1]interface LoopBack 0
```

```
[R1-LoopBack0]isis enable
```

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis enable
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]isis enable
```

```
[R3]interface GigabitEthernet 0/0/0
```

```
[R3-GigabitEthernet0/0/0]isis enable
```

```
[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]isis enable
```

```
[R4]interface GigabitEthernet 0/0/0
```

```
[R4-GigabitEthernet0/0/0]isis enable
```

```
[R4-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R4-GigabitEthernet0/0/1]isis enable
```

```
[R4-GigabitEthernet0/0/1]interface LoopBack 0
```

```
[R4-LoopBack0]isis enable
```

配置完成后，查看R1的路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----
```

Routing Tables: Public

Destinations : 14 Routes : 15

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					

```

0.0.0.0/0      ISIS-L1  15  10    D    10.0.13.3
GigabitEthernet0/0/1

ISIS-L1  15  10    D    10.0.12.2  GigabitEthernet0/0/0

10.0.1.1/32  Direct    0  0    D    127.0.0.1  LoopBack0

.....

```

可以看到，R1 的路由表中不存在关于R4 的Loopback 0 接口所在网段的路由信息，但是拥有自动生成的缺省路由，下一跳分别为R2（10.0.12.2）和R3（10.0.13.3）。

查看R2的路由表。

```
[R2]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```

-----
-----

```

Routing Tables: Public

Destinations : 14 Routes : 14

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/0	ISIS-L1	15	10	D	10.0.12.1
10.0.4.4/32 GigabitEthernet0/0/1	ISIS-L2	15	10	D	10.0.24.4
10.0.12.0/24	Direct	0	0	D	10.0.12.2 GigabitEthernet0/0/0
.....					

可以看到，R2 的路由表中拥有R1 和R4 的Loopback 0 接口所在网段的
路由信息。

查看R3的路由表。

[R3]display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 14 Routes : 14

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/1	ISIS-L1	15	10	D	10.0.13.1
10.0.4.4/32 GigabitEthernet0/0/0	ISIS-L2	15	10	D	10.0.34.4
10.0.12.0/24 GigabitEthernet0/0/1	ISIS-L1	15	20	D	10.0.13.1
.....					

可以看到，R3 的路由表中拥有R1 和R4 的Loopback 0 接口所在网段的
路由信息。

查看R4的路由表。

[R4]display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 14 Routes : 15

Destination/Mask Interface	Proto	Pre	Cost	Flags	NextHop
10.0.1.1/32 GigabitEthernet0/0/0	ISIS-L2	15	20	D	10.0.34.3
ISIS-L2 15 20	D		10.0.24.2		GigabitEthernet0/0/1
10.0.4.4/32	Direct	0 0	D		127.0.0.1 LoopBack0
.....					

可以看到，R4 的路由表中拥有R1的Loopback 0 接口所在网段的路由信息，且有两个下一跳，分别是R2（10.0.24.2）和R3（10.0.34.3）。

从上面检查各路由器的路由表情况可知，R1 和R4 的Loopback 0 接口所在网段已经可以互通了，前者访问后者使用的是缺省路由，并采用了负载均衡方式，后者访问前者使用的是明细路由，也采用了负载均衡方式。另外还发现，在Level-1-2路由器上，Level-1路由会被渗透进Level-2区域，而Level-2路由不会被渗透进Level-1区域。

3．配置IS-IS路由渗透

接下来，修改R1的GE 0/0/0 接口的Cost 值为10，R1 的GE 0/0/1 接口的Cost 值为20，R2 的GE 0/0/1 接口的Cost 值为30，R3 的GE 0/0/0 接口的Cost 值为10。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis cost 10
```



```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]isis cost 20
```

```
[R2]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]isis cost 30
```

```
[R3]interface GigabitEthernet 0/0/0
```

```
[R3-GigabitEthernet0/0/0]isis cost 10
```

配置完成后，在R1上使用tracert命令验证从10.0.1.1/32到10.0.4.4/32的报文所经过的路径。

```
<R1>tracert -a 10.0.1.1 10.0.4.4
```

```
traceroute to 10.0.4.4(10.0.4.4), max hops: 30 ,packet length: 40,press  
CTRL_C to break
```

```
1 10.0.12.2 20 ms 10 ms 10 ms
```

```
2 10.0.24.4 40 ms 20 ms 10 ms
```

可以看到，报文所经过的中转路由器是R2，这是因为修改了Cost值之后，Level-1路由器R1选择了离它最近（Cost值最小）的Level-1-2路由器R2作为它自动生成的缺省路由的下一跳，从10.0.1.1到10.0.4.4的报文就是根据这条缺省路由发送出去的。但是，分析表明，从R1经R2到R4的总的开销值是40，而从R1经R3到R4的总的开销值是30，所以现在从R1去往R4的报文选择的是一条次优路径。

解决上述次优路径问题的方法之一，便是在Level-1-2路由器R2和R3上配置路由渗透。

```
[R2-isis-1]import-route isis level-2 into level-1
```

```
[R3-isis-1]import-route isis level-2 into level-1
```

配置完成后，查看R1的路由表。

```
[R1]display ip routing-table
```

Route Flags: R - relay, D - download to fib

```
-----  
-----  
  
Routing Tables: Public
```

```
Destinations : 15   Routes : 15
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	nterface
0.0.0.0/0	ISIS-L1	15	10	D	10.0.12.2	GigabitEthernet0/0/0
10.0.1.1/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.4.4/32	ISIS-L1	15	30	D	10.0.13.3	GigabitEthernet0/0/1
10.0.12.0/24	Direct	0	0	D	10.0.12.1	GigabitEthernet0/0/0
.....						

可以看到，现在R1的路由表中已经拥有了关于10.0.4.4/32的路由信息，下一跳为R3（10.0.13.3）。虽然缺省路由还仍然存在，并且缺省路由的下一跳仍然指向的是R2，但是根据路由匹配原则，从R1去往R4的报文已经不再选择这条缺省路由来发送了。至此，路经次优问题已得到了解决，网络需求已得到了完全满足。

思考

举例说明，可以利用IS-IS路由渗透来满足什么样的网络需求。

4.11 IS-IS 监测和调试

原理概述

为了监测IS-IS协议的工作状态，VRP系统提供了一系列的查询命令。熟练使用这些命令，可以全面地了解网络的运行情况。同时，VRP系统还提供了一系列的调试命令，用以详细地了解和调试IS-IS的工作过程，并知道工作过程中各种事件的细节和关系。查询命令和调试命令的结合使用，有助于快速查找到网络的故障点和故障原因，提高查错排错的效率。

实验目的

- 掌握监测IS-IS 协议工作状态的方法
- 掌握调试IS-IS 协议工作过程的方法

实验内容

实验拓扑如图4-11 所示，实验编址如表 4-11 所示。本实验模拟了一个企业网络场景，IS-IS区域20的R1和R2为公司总部的网络路由器，区域10的R3为公司分支机构的网络路由器。R1、R2、R3 都运行IS-IS 协议，R1、R2、R3的Loopback 0 接口模拟了公司内部的各个网络。R1为Level-1路由器，R2为Level-1-2路由器，R3为Level-2路由器。R4为公司外部网络路由器，不运行IS-IS路由协议，且通过静态缺省路由来访问公司的各个内部网络。R3通过静态路由访问R4的各个Loopback接口所在的网络，并将各静态路由聚合之后引入进IS-IS 网络。为了安全起见，R2 的Serial 1/0/0 接口和R3的Serial 1/0/0 接口上需要配置认证功能。在整个网络的配置和运行过程中，需要对网络的状态进行适当的监测和调试。

实验拓扑

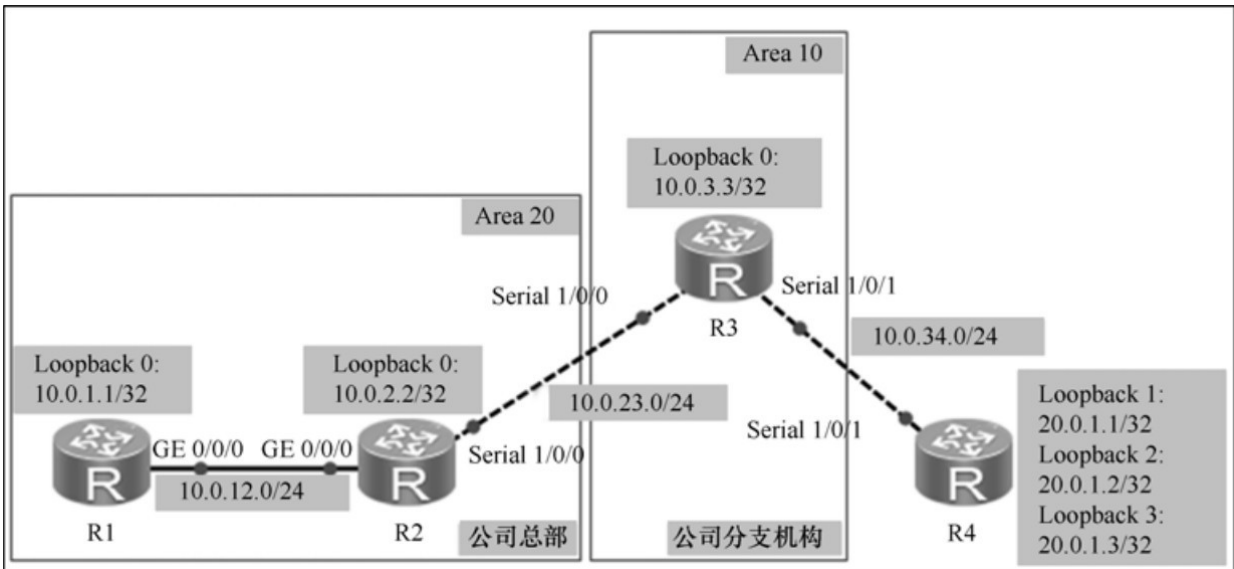


图4-11 IS-IS监测和调试

实验编址表

表4-11 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	NET: 20.0000.0000.0001.00			
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	Serial 1/0/0	10.0.23.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
	NET: 20.0000.0000.0002.00			
R3(AR2220)	Serial 1/0/0	10.0.23.3	255.255.255.0	N/A
	Serial 1/0/1	10.0.34.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
	NET: 10.0000.0000.0003.00			
R4(AR2220)	Serial 1/0/1	10.0.34.4	255.255.255.0	N/A
	Loopback 1	20.0.1.1	255.255.255.255	N/A
	Loopback 2	20.0.1.2	255.255.255.255	N/A
	Loopback 3	20.0.1.3	255.255.255.255	N/A

实验步骤

1. 基本配置

根据图4-11和表4-11进行相应的基本配置，并使用ping命令检测R2与R3之间的连通性。

```
<R2>ping -c 1 10.0.23.3
```

```
PING 10.0.23.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.23.3: bytes=56 Sequence=1 ttl=255 time=10 ms
```

```
--- 10.0.23.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 10/10/10 ms
```

其余直连网段的连通性测试过程在此省略。

2．IS-IS及相关内容的配置

在R1上配置IS-IS协议。

```
[R1]isis 10
```

```
[R1-isis-10]is-level level-1
```

```
[R1-isis-10]is-name R1
```

```
[R1-isis-10]network-entity 20.0000.0000.0001.00
```

```
[R1-isis-10]quit
```

```
[R1]interface LoopBack 0
```

```
[R1-LoopBack0]isis enable 10
```

```
[R1-LoopBack0]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable 10
```

在R2 上配置IS-IS 协议，并在其Serial 1/0/0 接口上配置IS-IS 认证功能。

```
[R2]isis 10
```

```
[R2-isis-10]is-level level-1-2
```

```
[R2-isis-10]is-name R2
```

```
[R2-isis-10]network-entity 20.0000.0000.0002.00
```

```
[R2-isis-10]quit
```

```
[R2]interface loopback 0
```

```
[R2-LoopBack0]isis enable 10
```

```
[R2-LoopBack0]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis enable 10
```

```
[R2-GigabitEthernet0/0/0]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]isis enable 10
```

```
[R2-Serial1/0/0]isis authentication-mode md5 plain huawei
```

在R3 上配置IS-IS 协议，在其Serial 1/0/0 接口上配置IS-IS 认证功能，并配置去往R4的各个Loopback接口的静态路由，然后将这些静态路由聚合后引入进IS-IS进程。

```
[R3]isis 10
```

```
[R3-isis-10]is-level level-2
```

```
[R3-isis-10]is-name R3
```

```
[R3-isis-10]network-entity 10.0000.0000.0003.00
```

```
[R3-isis-10]import-route static
```

```
[R3-isis-10]summary 20.0.1.0 255.255.255.0
```

```
[R3-isis-10]quit
```

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]isis enable 10
```

```
[R3-Serial1/0/0]isis authentication-mode md5 plain huawei
```

```
[R3-Serial1/0/0]interface LoopBack 0
```

```
[R3-LoopBack0]isis enable 10
```

```
[R3-LoopBack0]quit
```

```
[R3]ip route-static 20.0.1.1 32 10.0.34.4
```

```
[R3]ip route-static 20.0.1.2 32 10.0.34.4
```

```
[R3]ip route-static 20.0.1.3 32 10.0.34.4
```

在R4上配置静态缺省路由。

```
[R4]ip route-static 0.0.0.0 0.0.0.0 10.0.34.3
```

3 . 监测IS-IS运行的基本状态

在R1 上使用命令display isis brief查看IS-IS 协议的概要信息。

```
[R1]display isis brief
```

ISIS Protocol Information for ISIS(10)

SystemId: 0000.0000.0001 System Level:L1

Area-Authentication-mode: NULL

Domain-Authentication-mode: NULL

Ipv6 is not enabled

ISIS is in invalid restart status

ISIS is in protocol hot standby state: Real-Time Backup

Interface: 10.0.1.1(Loop0)

Cost:L1 0 L2 0 Ipv6 Cost:L1 0 L2 0

State: IPV4 Up IPV6 Down

Type:P2P MTU: 1500

Priority: L1 64 L2 64

Timers: Csnp:L12 10 Retransmit:L12 5 Hello: 10

Hello Multiplier: 3 LSP-Throttle Timer:L12 50

Interface: 10.0.12.1(GE0/0/0)

Cost:L1 10 L2 10 IPv6 Cost:L1 10 L2 10

State: IPv4 Up IPv6 Down

Type:BROADCAST MTU:1497

Priority: L1 64 L2 64

Timers: Csnp:L1 10 L2 10 Retransmit:L12 5 Hello:L1 10L2 10

Hello Multiplier:L13 L23 LSP-Throttle Timer: L1250

可以看到，回显信息中包含了R1上的一些IS-IS基本状态和参数，许多参数在前面的实验中都已进行了介绍，这里不再赘述。

IS-IS 协议中，最基本而重要的就是邻接关系的建立。接下来，在 R2 上使用命令display isis peer 查看R2 的邻居信息。

```
<R2>display isis peer
```

Peer information for ISIS(10)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	R1.01	Up	9s	L1	64
R3	S1/0/0	0000000001	Up	23s	L2	--

可以看到，邻居信息中包含了邻居的系统 ID，建立邻接关系所使用的本地接口及Circuit ID，以及邻接关系的状态标志、HoldTime、邻接级别（Level-1 或Level-2）等内容。该命令可以添加关键字Interface来查看指定接口所建立的IS-IS邻接关系，也可以添加IS-IS进程号来查看指定IS-IS进程的邻接关系，还可以添加关键字Verbose来获得邻居的详细信息。

在R2 上使用命令display isis peer interface Serial 1/0/0 verbose查看与R2 的Serial 1/0/0接口相连的邻居的详细信息。

```
<R2>display isis peer interface Serial 1/0/0 verbose
```

Peer information for ISIS(10)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

```
-----  
-----  
R3          S1/0/0      00000000002  Up    27s    L2      --
```

```
MT IDs supported          : 0(UP)
```

```
Local MT IDs              : 0
```

```
Area Address(es)         : 10
```

```
Peer IP Address(es)      : 10.0.23.3
```

```
Uptime                   : 00:00:42
```

```
Adj Protocol             : IPV4
```

```
Restart Capable          :YES
```

```
Suppressed Adj           :NO
```

```
Peer System Id           : 0000.0000.0003
```

```
Total Peer(s): 1
```

可以看到，回显信息中包含了邻居R3的各种详细信息。

在R2 上使用命令display isis interface 监测其IS-IS 接口的状态。

```
<R2>display isis interface
```

```
Interface information for ISIS(10)
```

```
-----  
Interface      Id      IPv4.State      IPv6.State  MTU  
Type  DIS  
Loop0      001    Up      Down      1500    L1/L2    --
```

GE0/0/0	001	Up	Down	1497	L1/L2	No/No
S1/0/0	002	Up	Down	1500	L1/L2	--

可以看到，回显信息中包含了R2的Loopback 0 接口、GE 0/0/0 接口和Serial 1/0/0接口的基本状态。通过添加关键字，此命令还可以查看指定接口的详细情况。

在R2 上使用命令display isis interface Serial 1/0/0 verbose 查看Serial 1/0/0接口的详细情况。

```
<R2>display isis interface Serial 1/0/0 verbose
```

```
Interface information for ISIS(10)
```

```
-----
```

Interface	Id		IPv4.State	IPv6.State	MTU	Type	DIS
S1/0/0	002	Up	Down	1500	L1/L2	--	

```
Circuit MT State           : Standard
```

```
Description                : HUAWEI,AR Series,Serial1/0/0
Interface
```

```
SNPA Address               : 0000-0000-0000
```

```
IP Address                  : 10.0.23.2
```

```
IPv6 Link Local Address    :
```

```
IPv6 Global Address(es)    :
```

```
Csnp Timer Value           : L12    10
```

```
Hello Timer Value          :      10
```

```
DIS Hello Timer Value      :
```

```

Hello Multiplier Value      :      3
Cost                        :  L1  10  L2  10
Ipv6 Cost                  :  L1  10  L2  10
Retransmit Timer Value     :  L12  5
LSP-Throttle Timer         :  L12  50
Bandwidth-Value            :  Low  2048000 High    0
Static Bfd                 : NO
Dynamic Bfd                : NO
Fast-Sense Rpr             : NO
Extended-Circuit-Id Value  :  0000000002

```

当 IS-IS 网络的邻居邻接关系建立完成之后，主要需要观察和监测的是链路状态数据库和IS-IS路由信息。例如，R2是一台Level-1-2路由器，会同时维护Level-1和Level-2 LSDB，显示LSDB 的整体内容可使用命令display isis lsdb。

```
<R2>display isis lsdb
```

```
Database information for ISIS(10)
```

```
-----
```

```
Level-1 Link State Database
```

```

LSPID      Seq Num    Checksum  Holdtime  Length  ATT/P/OL

```

```
-----
-----
```

```

R1.00-00    0x0000000d    0x5efd    728    88    0/0/0

```

R1.01-00	0x00000007	0xa9dd	728	55	0/0/0
R2.00-00*	0x0000000e	0xb3b7	597	104	1/0/0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

R2.00-00*	0x00000014	0x5b6e	597	116	0/0/0
R3.00-00	0x0000000e	0x4403	709	88	0/0/0
R3.00-01	0x00000001	0x4b7b	1149	65	0/0/0

Total LSP(s): 3

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),
ATT-Attached, P-Partition, OL-Overload

另外，此命令还可以通过添加IS-IS进程号来查看指定IS-IS进程的LSDB，添加LSP ID来查看指定的LSP，添加Local或邻居的Is-Name来查看本地生成的或是从指定邻居学习到的LSP的信息，添加Level-1或Level-2来单独查看Level-1或Level-2的LSDB，添加verbose来查看LSP的详细信息。

例如，可以在R2上查看由R1生成的LSP的详细情况。

<R2>display isis lsdb is-name R1 verbose

Database information for ISIS(10)

Level-1Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0000.0000.0001.00-00	0x0000000e	0x5cfe	495	88

0/0/0

SOURCE R1.00

HOST NAME R1

NLPID IPv4

AREA ADDR 20

INTF ADDR 10.0.1.1

INTF ADDR 10.0.12.1

NBR	ID	R1.01	COST: 10

IP-Internal	10.0.1.1	255.255.255.255	COST: 0

IP-Internal	10.0.12.0	255.255.255.0	COST: 10

除了 LSDB 外，IS-IS 路由表也是需要重点观察和监测的内容。在 R2 上使用命令display isis route 查看IS-IS 路由表。

<R2>display isis route

Route information for ISIS(10)

ISIS(10) Level-1 Forwarding Table

IPv4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.23.0/24	10	NULL	S1/0/0	Direct	D-/L/-

10.0.2.2/32	0	NULL	Loop0	Direct	D-/L/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D-/L/-

10.0.1.1/32	10	NULL	GE0/0/0	10.0.12.1	A-/L/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

ISIS(10) Level-2 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

10.0.3.3/32	10	NULL	S1/0/0	10.0.23.3	A/-/-

10.0.23.0/24	10	NULL	S1/0/0	Direct	D-/L/-

10.0.2.2/32	0	NULL	Loop0	Direct	D-/L/-

10.0.12.0/24	10	NULL	GE0/0/0	Direct	D-/L/-

20.0.1.0/24 10 0 S1/0/0 10.0.23.3 A/-/-/-

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

此命令还可以通过添加关键字 Verbose 来获得更详细的信息。在 R2 上使用命令 display isis route verbose 20.0.1.1 查看关于 20.0.1.1 的路由详情。

<R2>display isis route verbose 20.0.1.1

Route information for ISIS(10)

ISIS(10) Level-2 Forwarding Table

IPv4 Dest : 20.0.1.0/24 Int.Cost : 10 Ext.Cost: 0

Admin Tag : - Src Count : 1 Flags :A/-/-/-

Priority :Low

NextHop :10.0.23.3 Interface :S1/0/0 ExitIndex : 0x80000001

Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,

U-Up/Down Bit Set

命令 display isis error 可以用来查看 IS-IS 进程中的错误统计信息。在 R2 上使用命令 display isis error 查看错误统计信息。

<R2>display isis error

Statistics of error packets for ISIS(10)

LSP packet errors:

Longer LSP	:0	Smaller LSP	:0
Mismatched Level	:0	Invalid Sysid	:0
Zero Sequence Number	:0	Illegal IS Type	:0
Zero Checksum	:0	Incorrect Checksum	:0
Bad Authentication	:0	Bad Auth Count	:0
More Protocol TLV	:0	Bad Nbr TLV	:0
Bad Extended IS TLV	:0	Bad IF Addr TLV	:0
Bad Reach TLV	:0	Bad Inter Domain TLV	:0
Mismatched Area Id(L1)	:0	Bad TLV Length	:0
Bad Alias TLV	:0	Bad Area TLV	:0
Bad SRLG TLV	:0	Unknown Adjacency	:0
Bad Protocol ID	:0	Bad Version	:0
Zero Lifetime	:4	Bad Ext Reach TLV	:0
Bad TE Router ID TLV	:0	Bad TE Sub TLV	:0

Hello packet errors:

Bad Packet Length	:0	Reserved CircType	:0
Repeated System ID	:0	Bad Circuit Type	:0
Longer packet	:0	More Area Addr	:0

Longer Area Addr	:0	Bad Area Addr TLV	:0
More IF Addr	:0	Bad Formatted IF TLV	:0
More Nbr SNPA(LAN)	:0	Invalid Sysid	:0
Bad TLV Length	:0	Zero Holding Time	:0
Unusable IP Addr	:0	Repeated IPv4 Addr	:0
Mismatched Area Addr(L1):0		Mismatched Proto	:0
SNPA Conflicted(LAN)	:0	Mismatched Level	:0
Mismatched Max Area Addr:0		Bad Authentication	:0
More Auth TLV	:0	3-Way Option Error(P2P)	:0
No Area Addr TLV	:0	Bad Protocol ID	:0
Bad Version	:0	Invalid IPv6 Addr	:0
More IPv6 IF Addr	:0	Duplicate IPv6 Addr	:0
More Optional Checksum	:0	Bad Optional Checksum	:0

命令display isis spf-log和命令display isis spf-tree可以用来查看与SPF算法相关的信息，读者可自行执行这些命令并观察和分析相应的回显信息。

4．调试IS-IS协议的工作过程

上述步骤中，通过VRP提供的各种信息查看命令，能够很好地监测IS-IS的基本工作状态。如果需要了解IS-IS的工作过程，则需要用到各种调试命令。

在R1 上使用terminal debugging 命令开启调试功能。

```
<R1>terminal debugging
```

在R1 上使用命令debugging isis event 来查看IS-IS 协议工作过程中的所有事件。

```
<R1>debugging isis event
```

观察发现，结果是没有任何输出信息，这是因为 IS-IS 协议此时工作稳定，并没有发生变化事件。下面在R1 上使用reset isis all 命令重启IS-IS 进程来观察邻接关系的建立过程及SPF算法的工作过程。

```
<R1>reset isis all
```

Warning: The ISIS process(es) will be reset. Continue?[Y/N]y

```
ISIS-10-SPF-NODE: [L1-MT0] Destroy node(NodeID=0000.0000.0001.00, DIST=0, Nexthops=0, NBRs=1, Parents=0, Flags=Tree)(IS09_1018)
```

```
Aug 28 2013 11:26:18.337.2-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-SPF-LINK: [L1-MT0] Destroy
```

```
link(LinkID=0000.0000.0001.00->0000.0000.0001.01, OldCost=10, NewCost=10, Flags=0, Tree)(IS09_1119)
```

```
Aug 28 2013 11:26:18.337.3-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-SPF-NODE: [L1-MT0] Clear DIRECT flag on node(NodeID=0000.0000.0001.01, DIST=10, Nexthops=0, NBRs=2, Parents=1, Flags=Tree/Isolated)(IS09_1018)
```

```
Aug 28 2013 11:26:18.337.4-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-SPF-NODE: [L1-MT0] Destroy node(NodeID=0000.0000.0001.01, DIST=10, Nexthops=0, NBRs=2, Parents=1, Flags=Tree/Isolated) (IS09_1018)
```

Aug 28 2013 11:26:18.337.5-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-LINK: [L1-MT0] Destroy

link(LinkID=0000.0000.0001.01->0000.0000.0002.00, OldCost=0, NewCost=0, AttAdjs=1, Flags=1, Tree)(IS09_1119)

Aug 28 2013 11:26:18.337.6-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-LINK: [L1-MT0] Destroy

link(LinkID=0000.0000.0001.01->0000.0000.0001.00, OldCost=0, NewCost=0, Flags=0, Back)(IS09_1119)

Aug 28 2013 11:26:18.337.7-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-NODE: [L1-MT0] Destroy node(NodeID=0000.0000.0002.00, DIST=10, Nexthops=1, NBRs=1, Parents=1, Flags=Tree)(IS09_1018)

Aug 28 2013 11:26:18.337.8-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-LINK: [L1-MT0] Destroy

link(LinkID=0000.0000.0002.00->0000.0000.0001.01, OldCost=10, NewCost=10, Flags=0, Back)(IS09_1119)

Aug 28 2013 11:26:18.337.9-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-ADJ: The ADJ state changed from Up to Down.
(0000.0000.0002 on GigabitEthernet0/0/0, SysType=L1, State=Up, Prot=IPv4)(IS09_936)

Aug 28 2013 11:26:18.337.10-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-ADJ: [L1-MT0] Delete BCASTADJ(AdjID=0000.0000.0002) (IS09_4250)

Aug 28 2013 11:26:18.337.11-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-LINK: [L1-MT0] Deleting RMT

LINK(LinkID=0000.0000.0002.00->0000.0000.0001.01, Cost=10)
(IS09_2598)

Aug 28 2013 11:26:18.357.1-05:13 R1 ISIS/6/ISIS:

ISIS-10-AREA: Loading manual configured Areas.(IS03_219)

Aug 28 2013 11:26:18.357.2-05:13 R1 ISIS/6/ISIS:

ISIS-10-AREA: Install one Area [20] for L1 MT0 1/0(IS03_444)

Aug 28 2013 11:26:18.357.3-05:13 R1 ISIS/6/ISIS:

ISIS-10-AREA: Install one Area [20] for L2 MT0 1/1(IS03_444)

Aug 28 2013 11:26:18.357.4-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-NODE: [L1-MT0] Create node(NodeID=0000.0000.0001.00,
DIST=0,

Nexthops=0, NBRs=0, Parents=0, Flags=Tree)(IS09_1018)

Aug 28 2013 11:26:18.367.1-05:13 R1 ISIS/6/ISIS:

ISIS-10-ISPF-ADJ: The ADJ state changed from Null to Up.
(0000.0000.0002 on GigabitEthernet0/0/0, SysType=L1, State=Null,
Prot=IPv4)(IS09_936)

Aug 28 2013 11:26:18.367.2-05:13 R1 ISIS/6/ISIS:

ISIS-10-ISPF-ADJ: [MT0] ADJ MT usage change from 1 to 2.
(0000.0000.0002 on GigabitEthernet0/0/0, SysType=L1, State=Up,
Prot=IPv4)(IS09_936)

Aug 28 2013 11:26:18.367.3-05:13 R1 ISIS/6/ISIS:

ISIS-10-ISPF-ADJ: NextHop Change (0000.0000.0002 on
GigabitEthernet0/0/0,

SysType=L1, State=Up, Prot=IPv4)(IS09_936)

Aug 28 2013 11:26:18.407.1-05:13 R1 ISIS/6/ISIS:

ISIS-10-DEC-ISPF: MT 0 is scheduled to run ISPF calculation.(IS09_7567)

Aug 28 2013 11:26:18.407.2-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-EVE: [L1-MT0] Process node to inherit nexthop. (Nodes=0)
(IS09_8111)

Aug 28 2013 11:26:18.407.3-05:13 R1 ISIS/6/ISIS:

ISIS-10-SPF-EVE: [L1-MT0] Inform nodes change to Area and Route.
(Nodes=0)(IS09_8995)

Aug 28 2013 11:26:18.407.4-05:13 R1 ISIS/6/ISIS:

ISIS-10-DEC-ISPF: All Phases of MT 0 ISPF Completed.(IS09_7624)

Aug 28 2013 11:26:18.407.5-05:13 R1 ISIS/6/ISIS:

ISIS-10-DEC-ISPF: All MT SPF calculations were complete in process 10.
(IS10_13498)

有的时候，只观察 IS-IS 协议的事件是不够的，还需要对某些特定的数据报文或某些特别的属性进行调试。例如，可以使用命令 debugging isis adjacency 来查看邻接关系建立过程中报文的发送和接收的情况。

<R1>terminal debugging

<R1>debugging isis adjacency

<R1>reset isis all

Warning: The ISIS process(es) will be reset. Continue?[Y/N]y

Aug 28 2013 11:47:59.827.1-05:13 R1 ISIS/6/ISIS:

ISIS-10-set white list to product OperFlag = 2.(IS01_299)

Aug 28 2013 11:47:59.827.2-05:13 R1 ISIS/6/ISIS:

ISIS-10-ADJ: Sending Padding-Hello PDUs .(IS15_4803)

Aug 28 2013 11:47:59.827.3-05:13 R1 ISIS/6/ISIS:

ISIS-10-ADJ: Refresh level-1 IHH enconde cache, GE0/0/0.(IS15_2671)

Aug 28 2013 11:47:59.827.4-05:13 R1 ISIS/6/ISIS:

ISIS-10-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.
(IS15_6941)

Aug 28 2013 11:47:59.827.5-05:13 R1 ISIS/6/ISIS:

ISIS-10-ADJ: Circuit State Up Success.(IS15_260)

Aug 28 2013 11:47:59.847.1-05:13 R1 ISIS/6/ISIS:

ISIS-10-ADJ: Rxed Lan L1 Hello on GE0/0/0, from SNPA 00e0.fc03.12f2.
(IS15_1864)

Aug 28 2013 11:47:59.847.2-05:13 R1 ISIS/6/ISIS:

ISIS-10-set white list to product OperFlag = 1.(IS01_299)

.....

命令debugging isis authentication-error可用来调试IS-IS协议认证方面的问题。例如，将R2的Serial 1/0/0接口的认证密码修改为“huawei”，然后在R2上执行此命令。

```
[R2]interface Serial 1/0/0
```

```
[R2-Serial1/0/0]isis authentication-mode md5 plain huawei
```

```
<R2>terminal debugging
```

```
<R2>debugging isis authentication-error
```

Aug 28 2013 11:54:12.696.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-UTL: Invalid Info in the received PDU(IS39_701)

Aug 28 2013 11:54:12.696.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-UTL: Check fails for PDU type 17(IS39_352)

Aug 28 2013 11:54:22.706.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-UTL: Invalid Info in the received PDU(IS39_701)

Aug 28 2013 11:54:22.706.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-UTL: Check fails for PDU type 17(IS39_352)

调试完毕后，如果将密码重新修改为huawei，会发现此命令将不会产生任何输出信息，表明认证方面不存在问题。

针对CSNP（Complete Sequence Number PDU）和PSNP（Partial Sequence Number PDU）信息的调试命令是debugging isis snp-packet。在R2上使用此命令查看接口上发送和接收CSNP和PSNP报文的情况。

<R2>debugging isis snp-packet

Aug 28 2013 12:00:24.96.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-SNP: CSNP Range from 0000.0000.0000.00-00 to ffff.ffff.ffff.ff-ff. (IS36_424)

Aug 28 2013 12:00:24.96.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-SNP: Rxed L1 CSNP From 0000.0000.0001 (GE0/0/0).(IS36_271)

Aug 28 2013 12:00:25.486.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-SNP: Rxed L2 PSNP From 0000.0000.0003 (S1/0/0).(IS36_271)

Aug 28 2013 12:00:25.496.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-PSNP: Sending L2 PSNP on S1/0/0.(IS36_1433)

Aug 28 2013 12:00:29.486.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-SNP: Sending CSNP on Interface S1/0/0.(IS36_2051)

Aug 28 2013 12:00:29.486.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-CSNP: Sending L2 CSNP on S1/0/0.(IS36_2091)

Aug 28 2013 12:00:29.486.3-05:13 R2 ISIS/6/ISIS:

ISIS-10-SNP: Sending CSNP on Interface S1/0/0.(IS36_2051)

在R3 上取消静态路由的引入，并在R2 上使用debugging isis update-packet 命令调试IS-IS协议更新数据报文信息。

```
<R2>debugging isis update-packet
```

```
[R3]isis 10
```

```
[R3-isis-10]undo import-route static
```

```
<R2>
```

Aug 28 2013 12:42:01.136.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-UPDT: RxL2 Lsp 0000.0000.0003.00-00, Seq 0x35, ht 1196. From SNPA 00 on S1/0/0.

Aug 28 2013 12:42:01.136.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-UPDT: Because of no nbr ,Not Flooding L2 Lsp 0000.0000.0003.00-00 GE0/0/0.

可以看到，R2 从 Serial 1/0/0 接口接收到了 Level-2 LSP，但是这条 LSP 并没有从GE 0/0/0 接口泛洪出去，原因是GE 0/0/0 接口上的邻居 R1是Level-1 路由器，R2 与R1建立的是Level-1邻接关系。

最后需要提醒的是，一旦调试任务完成，应立即使用undo debugging all 命令关闭所有的调试功能。

思考

通常有哪些因素会影响到IS-IS邻接关系的正常建立？

4.12 IS-IS 故障排除

原理概述

IS-IS协议故障问题可以大致分为两类，第一类涉及IS-IS邻接关系的建立问题，第二类涉及LSP及路由计算问题。查找和排除故障问题时，应该先从第一类问题切入，因为如果邻接关系无法正常建立，路由器之间就无法正确地传递LSP并进行正确的路由计算。通常，邻接关系的故障起因于NET地址中区域ID的错误、路由器类型（Level-1，Level-2，Level-1-2）或接口的级别（Level-1，Level-2，Level-1-2）错误、认证方面的错误，以及IP地址错误等。

实验目的

- 理解IS-IS 协议故障排除的思路
- 掌握查找和排除IS-IS 邻接关系故障的基本方法
- 掌握查找和排除IS-IS LSP及路由故障的基本方法

实验内容

实验拓扑如图4-12所示，实验编址如表4-12所示。本实验模拟了一个简单的企业网络场景，R1和R2位于公司总部的网络，R3位于分支机构的网络。R1为Level-1路由器，R2 为Level-1-2 路由器，R3 为Level-2 路由器，R1 和R2 的Loopback 0 接口模拟了公司总部的两个内部网络，R3 的Loopback 0 接口模拟了分支机构的内部网络。所有路由器都运行IS-

IS路由协议，路由器之间都配置了认证功能，密码为Huawei。网络需求是：排除网络故障，实现全网互通。

实验拓扑

figure_0414_0128

图4-12 IS-IS故障排除

实验编址表

表4-12 实验编址

figure_0414_0129

实验步骤

1．基本配置

根据图4-12和表4-12进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=10 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 10/10/10 ms
```

R2与R3之间的连通性测试过程在此省略。

2 . 配置IS-IS路由协议并设置故障点

在R1上配置IS-IS协议及认证功能。

```
[R1]isis 10
```

```
[R1-isis-10]is-level level-1
```

```
[R1-isis-10]network-entity 20.0000.0000.0001.00
```

```
[R1-isis-10]is-name R1
```

```
[R1-isis-10]quit
```

```
[R1]interface LoopBack 0
```

```
[R1-LoopBack0]isis enable 10
```

```
[R1-LoopBack0]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis enable 10
```

```
[R1-GigabitEthernet0/0/0]isis authentication-mode md5 plain Huawei
```

```
[R2]isis 10
```

```
[R2-isis-10]network-entity 10.0000.0000.0002.00
```

```
[R2-isis-10]is-name R2
```

```
[R2-isis-10]area-authentication-mode md5 plain Huawei
```

```
[R2-isis-10]undo import-route isis level-1 into level-2
```

```
[R2-isis-10]quit
```

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis enable 10
[R2-GigabitEthernet0/0/0]isis circuit-level level-2
[R2-GigabitEthernet0/0/0]interface Serial 1/0/0
[R2-Serial1/0/0]isis enable 10
[R3]isis 10
[R3-isis-10]is-level level-2
[R3-isis-10]network-entity 20.0000.0000.0003.00
[R3-isis-10]is-name R3
[R3-isis-10]area-authentication-mode md5 plain Huawei
[R3-isis-10]quit
[R3]interface LoopBack 0
[R3-LoopBack0]isis enable 10
[R3-LoopBack0]interface Serial 1/0/0
[R3-Serial1/0/0]ip address 10.0.32.3 255.255.255.0
[R3-Serial1/0/0]isis enable 10
[R3-Serial1/0/0]isis silent
```

3．查找并排除IS-IS邻接关系故障

首先，在R2上查看R2的邻居信息。

```
<R2>display isis peer
```

```
<R2>
```

可以看到，R2与R1、R2与R3都未能建立起邻接关系。

在R1上使用debugging isis adjacency 命令观察R1与R2建立邻接关系的过程。

```
<R1>terminal debugging
```

```
Info: Current terminal debugging is on.
```

```
<R1>debugging isis adjacency
```

```
Aug 29 2013 07:14:13.366.2-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.  
(IS15_6941)
```

```
Aug 29 2013 07:14:23.366.1-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.  
(IS15_2679)
```

```
Aug 29 2013 07:14:23.366.2-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.  
(IS15_6941)
```

```
Aug 29 2013 07:14:33.366.1-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.  
(IS15_2679)
```

```
Aug 29 2013 07:14:33.366.2-05:13 R1 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.  
(IS15_6941)
```

```
Aug 29 2013 07:14:43.366.1-05:13 R1 ISIS/6/ISIS:
```

ISIS-10-ADJ: Use level-1 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2679)

可以看到，R1 的GE 0/0/0 接口每隔10s 就会发送一个Level-1 Hello 报文，但是却一直没有接收到R2 发送的Level-1 Hello 报文，这说明R2 的配置可能存在问题。

在R2上使用debugging isis adjacency interface GigabitEthernet 0/0/0命令进行观察。

```
<R2>terminal debugging
```

Info: Current terminal debugging is on.

```
<R2>debugging isis adjacency interface GigabitEthernet 0/0/0
```

Aug 29 2013 07:20:11.719.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Sending Lan L2 Hello on GE0/0/0, to SNPA 0180.c200.0015.
(IS15_6963)

Aug 29 2013 07:20:21.719.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Use level-2 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2731)

Aug 29 2013 07:20:21.719.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Sending Lan L2 Hello on GE0/0/0, to SNPA 0180.c200.0015.
(IS15_6963)

Aug 29 2013 07:20:31.719.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Use level-2 IIH enconde cache to send IIH, GE0/0/0.
(IS15_2731)

可以看到，R2 的GE 0/0/0 接口只是在持续发送Level-2 Hello 报文，并未接收到R1发送的Level-1 Hello 报文。至此，可以断定R1的GE 0/0/0

接口和R2的GE 0/0/0接口在级别上（Level-1或Level-2）存在不匹配的情况。

分别在R1和R2 上使用命令display isis brief | include System Level 来查看路由器的级别。

```
[R1]display isis brief | include System Level
```

```
ISIS Protocol Information for ISIS(10)
```

```
-----
```

```
SystemId: 0000.0000.0001   System Level:L1
```

```
[R2]display isis brief | include System Level
```

```
ISIS Protocol Information for ISIS(10)
```

```
-----
```

```
SystemId: 0000.0000.0002   System Level:L2
```

由上可知，R1是Level-1路由器，R2是Level-1-2路由器。之前的调试信息反映出R2 的GE 0/0/0 接口并没有发送Level-1 Hello 报文，所以应该怀疑R2的GE 0/0/0 接口级别配置有问题。

在R2 上使用display isis interface GigabitEthernet 0/0/0 命令查看接口的级别。

```
<R2>display isis interface GigabitEthernet 0/0/0
```

```
Interface information for ISIS(10)
```

```
-----
```

Interface	Id	IPV4.State	IPV6.State	MTU	Type
DIS					

GE0/0/0	001	Up	Down	1497	L2	No

可以看到，R2 的GE 0/0/0 接口的级别为Level-2。至此，已经清楚R1 的GE 0/0/0接口和R2 的GE 0/0/0 接口的级别不匹配。

修改R2 的GE 0/0/0 接口的级别为Level-1-2。

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis circuit-level level-1-2
```

观察R2的调试输出。

```
<R2>debugging isis adjacency interface GigabitEthernet 0/0/0
```

```
Aug 29 2013 08:02:25.679.2-05:13 R2 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.  
(IS15_6941)
```

```
Aug 29 2013 08:02:26.89.1-05:13 R2 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Use level-2 IIH enconde cache to send IIH, GE0/0/0.  
(IS15_2731)
```

```
Aug 29 2013 08:02:26.89.2-05:13 R2 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Sending Lan L2 Hello on GE0/0/0, to SNPA 0180.c200.0015.  
(IS15_6963)
```

```
Aug 29 2013 08:02:33.259.1-05:13 R2 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Rxed Lan L1 Hello on GE0/0/0, from SNPA 00e0.fc03.51a9.  
(IS15_1864)
```

```
Aug 29 2013 08:02:33.259.2-05:13 R2 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Area Mismatch.Lan L1 Hello from 0000.0000.0001, on
```

GE0/0/0.(IS01_2383)

Aug 29 2013 08:02:33.259.3-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Hello PDU Dropped.(IS21_5266)

可以看到，R2的GE0/0/0接口现在同时在向R1发送Level-1和Level-2Hello报文，并且也接收到了R1发送的Level-1 Hello报文。然而，调试信息显示R2直接丢弃了R1发送的Level-1Hello报文，并显示了区域不匹配的提示。网络设计方案中，R1和R2应该同属于区域 10，说明R1或R2的NET配置可能有误。

<R1>display isis lsdb verbose

Database information for ISIS(10)

Level-1Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length	ATT/P/OL

0000.0000.0001.00-00*	0x00000006	0x17e7	772	74	0/0/0

SOURCE R1.00

HOST NAME R1

NLPID IPv4

AREA ADDR 20

INTF ADDR 10.0.12.1

INTF ADDR 10.0.1.1

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

IP-Internal 10.0.1.1 255.255.255.255 COST: 0

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到 , R1的NET被错误地配置成了20.0000.0000.0001.00。

查看R2的Level-1的LSDB的详细信息。

<R2>display isis lsdb level-1 verbose

Database information for ISIS(10)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0000.0000.0002.00-00*	0x0000000c	0xbb5f	598

93 0/0/0

SOURCE R2.00

HOST NAME R2

Auth: ***** Len: 16 Type:MD5

NLPID IPv4

AREA ADDR 10

INTF ADDR 10.0.23.2

INTF ADDR 10.0.12.2

IP-Internal 10.0.23.0 255.255.255.0 COST: 10

IP-Internal 10.0.12.0 255.255.255.0 COST: 10

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R2的NET配置是正确的。

在R1上修改NET的配置。

```
[R1]isis 10
```

```
[R1-isis-10]undo network-entity 20.0000.0000.0001.00
```

```
[R1-isis-10]network-entity 10.0000.0000.0001.00
```

修改完成后，在R2上观察调试输出。

```
<R2>debugging isis adjacency
```

```
Aug 29 2013 08:20:25.739.2-05:13 R2 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Sending Lan L1 Hello on GE0/0/0, to SNPA 0180.c200.0014.  
(IS15_6941)
```

```
Aug 29 2013 08:20:26.129.1-05:13 R2 ISIS/6/ISIS:
```

```
ISIS-10-ADJ: Use level-2 IIH encode cache to send IIH, GE0/0/0.  
(IS15_2731)
```

```
Aug 29 2013 08:20:26.129.2-05:13 R2 ISIS/6/ISIS:
```

ISIS-10-ADJ: Sending Lan L2 Hello on GE0/0/0, to SNPA 0180.c200.0015.
(IS15_6963)

Aug 29 2013 08:20:28.459.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Use p2p IIH enconde cache to send IIH, S1/0/0.(IS15_2587)

Aug 29 2013 08:20:28.459.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Sending P2P Hello, on S1/0/0.(IS15_6984)

Aug 29 2013 08:20:28.469.1-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Rxed Lan L1 Hello on GE0/0/0, from SNPA 00e0.fc03.51a9.
(IS15_1864)

Aug 29 2013 08:20:28.469.2-05:13 R2 ISIS/6/ISIS:

ISIS-10-ADJ: Store last IIH successfully on GE0/0/0, from systemId
0000.0000.0001.(IS01_3264)

Aug 29 2013 08:20:28.469.3-05:13 R2 ISIS/6/ISIS:

ISIS-10-IIH: Set L1 pseudo IIH on GE0/0/0 for NBR 0000.0000.0001 as
1(IS21_4818)

可以看到，R2 的GE 0/0/0 接口发送和接收Level-1 Hello 报文的情况是
正确的。

查看R2的IS-IS邻居信息。

<R2>display isis peer

Peer information for ISIS(10)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI


```
0000.0000.0001  GE0/0/0    ---          Init    29s    L1    64
```

```
Total Peer(s): 1
```

可以看到，R1虽然出现在了R2的邻居信息中，但R2与R1的邻接状态一直停留在Init状态。

在R1上观察IS-IS邻居信息。

```
[R1]display isis peer
```

```
[R1]
```

可以看到，R1的邻居信息为空，说明R1与R2仍没有建立邻接关系。

通过之前的调试过程可知，R1 和R2 之间都在发送Level-1 Hello 报文，然而，双向的Level-1邻接关系却仍然无法建立，这说明Hello报文交互过程中还是存在某些问题。

在R1 上使用命令display isis error查看IS-IS 错误消息。

```
[R1]display isis error
```

```
Statistics of error packets for ISIS(10)
```

```
-----
```

```
LSP packet errors:
```

```
Longer LSP           : 0    Smaller LSP           : 0
```

```
.....
```

```
SNPA Conflicted(LAN)  :0    Mismatched Level      :0
```

```
Mismatched Max Area Addr: 0    Bad Authentication    : 66
```

```
More AuthT LV         : 0    3-Way Option Error(P2P) : 0
```

.....

可以看到，可能是R1和R2 之间的认证方面出现了问题。使用display isis brief 命令查看R1的IS-IS协议的基本情况。

<R1>display isis brief

ISIS Protocol Information for ISIS(10)

SystemId: 0000.0000.0001 System Level:L1

Area-Authentication-mode: NULL

Domain-Authentication-mode: NULL

.....

可以看到，R1的IS-IS区域认证模式为NULL。

查看R2的IS-IS协议的基本情况。

<R2>display isis brief

ISIS Protocol Information for ISIS(10)

SystemId: 0000.0000.0002 System Level:L12

Area-Authentication-mode: MD5

Domain-Authentication-mode: NULL

.....

可以看到，R2的IS-IS区域认证模式为MD5。

在R2 上使用display current-configuration | include area-authentication 命令查看IS-IS区域认证的密钥。

```
<R2>display current-configuration | include area-authentication  
area-authentication-mode md5 plain Huawei
```

可以看到，在R2上的确启用了IS-IS的MD5区域认证。

在R1上也配置区域认证，并将密码设置为Huawei。

```
[R1]isis 10  
[R1-isis-10]area-authentication-mode md5 plain Huawei
```

配置完成后，在R1上查看IS-IS邻居信息。

```
[R1]display isis peer
```

```
[R1]
```

结果发现，R1依旧没有与R2建立起邻接关系。

在R2上查看IS-IS邻居信息。

```
[R2]display isis peer
```

Peer information for ISIS(10)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

0000.0000.0001	GE0/0/0	---	Init	27s	L1	64

Total Peer(s): 1

可以看到，R1与R2的邻接关系仍然为Init状态。

分别查看R1 和R2 的GE 0/0/0 接口的配置情况。

```
<R1>display current-configuration interface GigabitEthernet 0/0/0
```

```
[V200R003C00]
```

```
interface GigabitEthernet0/0/0
```

```
ip address 10.0.12.1 255.255.255.0
```

```
isis enable 10
```

```
isis authentication-mode md5 plain Huawei
```

```
#
```

```
Return
```

```
[R2]display current-configuration interface GigabitEthernet 0/0/0
```

```
[V200R003C00]
```

```
interface GigabitEthernet0/0/0
```

```
ip address 10.0.12.2 255.255.255.0
```

```
isis enable 10
```

```
#
```

```
return
```

可以看到，R2 的GE 0/0/0 接口并没有启用接口认证，而R1的GE 0/0/0 接口是启用了接口认证的。

在R2 的GE 0/0/0 接口上添加相应的认证配置。

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis authentication-mode md5 plain Huawei
```

配置完成后，在R2上查看IS-IS邻居信息。

```
<R2>display isis peer
```

Peer information for ISIS(10)

System Id	Interface	Circuit Id	State	HoldTime	Type	PRI

R1	GE0/0/0	R1.01	Up	8s	L1	64

Total Peer(s): 1

可以看到，R2与R1现在已经成功建立起了邻接关系，然而，R2与R3的邻接关系还没有建立起来。

在R3 上使用debugging isis adjacency 命令观察R3 与R2 建立邻接关系的过程。

```
<R3>terminal debugging
```

Info: Current terminal debugging is on.

```
<R3>debugging isis adjacency
```

Aug 29 2013 11:20:47.820.1-05:13 R3 ISIS/6/ISIS:

ISIS-10-ADJ: Hello PDU Dropped.(IS21_5266)

Aug 29 2013 11:20:57.830.1-05:13 R3 ISIS/6/ISIS:

ISIS-10-ADJ: Hello PDU Dropped.(IS21_5266)

Aug 29 2013 11:21:07.830.1-05:13 R3 ISIS/6/ISIS:

ISIS-10-ADJ: Hello PDU Dropped.(IS21_5266)

可以看到，R3并未发送任何Hello报文，而且总是丢弃收到的Hello报文。接口不发送Hello报文，很可能是因为接口没有使能IS-IS。

检查R3 的Serial 1/0/0 接口是否使能了IS-IS。

```
<R3>display current-configuration interface Serial 1/0/0
```

```
[V200R003C00]
```

```
interface Serial1/0/0
```

```
link-protocol ppp
```

```
ip address 10.0.32.3 255.255.255.0
```

```
isis enable 10
```

```
isis silent
```

```
#
```

```
return
```

观察发现，R3 的Serial 1/0/0 接口虽然使能了IS-IS，但被配置成了抑制状态。

在R3 上删除抑制Serial 1/0/0 接口的配置。

```
[R3]interface Serial 1/0/0
```

```
[R3-Serial1/0/0]undo isis silent
```

配置完成后，在R3上观察调试输出。

```
<R3>debugging isis adjacency
```

```
Info: Current terminal debugging is on.
```

```
Aug 29 2013 11:38:07.650.2-05:13 R3 ISIS/6/ISIS:
```

ISIS-10-ADJ: Sending P2P Hello, on S1/0/0.(IS15_6984)

Aug 29 2013 11:38:08.250.1-05:13 R3 ISIS/6/ISIS:

ISIS-10-ADJ: Rxed P2P Hello, on S1/0/0.(IS15_1373)

Aug 29 2013 11:38:08.250.2-05:13 R3 ISIS/6/ISIS:

ISIS-10-ADJ: Rxed P2P IIH contains No usable Ipv4 Address.(IS15_1651)

Aug 29 2013 11:38:08.250.3-05:13 R3 ISIS/6/ISIS:

ISIS-10-ADJ: Hello PDU Dropped.(IS21_5266)

可以看到，R3 的Serial 1/0/0 接口开始发送Hello 报文了，但是对于接收到的Hello报文，调试信息提示它包含有不可用的IPv4地址。

查看R2 和R3 的Serial 1/0/0 接口的IP 地址。

<R2>display ip interface brief | include Serial1/0/0

*down: administratively down

^down: standby

(l): loopback

(s): spoofing

The number of interface that is UP in Physical is 4

The number of interface that is DOWN in Physical is 3

The number of interface that is UP in Protocol is 4

The number of interface that is DOWN in Protocol is 3

Interface	IP Address/Mask	Physical	Protocol
Serial1/0/0	10.0.23.2/24	up	up

<R3>display ip interface brief | include Serial1/0/0

*down: administratively down

^down: standby

(l): loopback

(s): spoofing

The number of interface that is UP in Physical is 4

The number of interface that is DOWN in Physical is 3

The number of interface that is UP in Protocol is 4

The number of interface that is DOWN in Protocol is 3

Interface	IP Address/Mask	Physical	Protocol
Serial1/0/0	10.0.32.3/24	up	up

观察发现，R3 的Serial 1/0/0 接口的IP 地址与R2的Serial 1/0/0 接口的IP 地址不在同一个网段。

修改R3 的Serial 1/0/0 接口的IP 地址。

[R3]interface Serial 1/0/0

[R3-Serial1/0/0]ip address 10.0.23.3 24

配置完成后，在R2上查看邻居信息。

[R2]display isis peer

Peer information for ISIS(10)

System Id	Interface	Circuit Id	State	HoldTime	Type
PRI					

```

-----
R1      GE0/0/0  R2.01    Up    29s    L2    64
R3      S1/0/0   0000000002  Up    24s    L2      --

```

Total Peer(s): 2

可以看到，现在R2与R1和R3都成功建立起了邻接关系。

4．查找并排除LSDB及路由故障

目前，各路由器之间的邻接关系是正常的，接下来需要了解路由现状。

在R1上查看IS-IS路由表。

[R1]display isis route

Route information for ISIS(10)

```

-----
ISIS(10) Level-1 Forwarding Table

```

```

-----
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
-----

```

```

0.0.0.0/0         10       NULL    GE0/0/0        10.0.12.2  A/-/-/-
10.0.23.0/24      20       NULL    GE0/0/0        10.0.12.2  A/-/-/-
.....

```

观察发现，R1 的IS-IS 路由表中竟然没有去往本区域中R2的Loopback 0 接口所模拟的内部网络 10.0.2.2/32 的路由。由于路由是根据 LSP 来计算得到的，所以可以先在R1的LSDB查看一下由R2生成的LSP的情况。

```
[R1]display isis lsdb is-name R2 verbose
```

Database information for ISIS(10)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0000.0000.0002.00-00	0x00000010	0x1837	1024	107
1/0/0				

.....

NBR	ID	R1.01	COST: 10

IP-Internal	10.0.12.0	255.255.255.0	COST: 10

IP-Internal	10.0.23.0	255.255.255.0	COST: 10

Total LSP(s): 1

.....

可以看到，R1的LSDB中并没有描述10.0.2.2/32网络的LSP，所以可能是LSP在传递过程中发生了问题。

在R2 上使用命令display isis lsdb is-name R2 verbose。

```
<R2>display isis lsdb is-name R2 verbose
```

Database information for ISIS(10)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0000.0000.0002.00-00*	0x0000001e	0xd320	1148
107 1/0/0			

.....

NBR	ID	R1.01	COST: 10

IP-Internal	10.0.12.0	255.255.255.0	COST: 10

IP-Internal	10.0.23.0	255.255.255.0	COST: 10

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0000.0000.0002.00-00*	0x0000000d	0x728	1148	88
0/0/0				

.....

NBR	ID	R3.00	COST: 10
IP-Internal	10.0.12.0	255.255.255.0	COST: 10
IP-Internal	10.0.23.0	255.255.255.0	COST: 10

Total LSP(s): 1

.....

可以看到，R2 的Level-1 LSDB 和Level-2 LSDB 中也都没有描述 10.0.2.2/32 网络的LSP，这说明R2根本就没有产生关于10.0.2.2/32的 LSP，究其原因，原来是R2的Lookback 0接口没有使能IS-IS。

```
[R2]interface LoopBack 0
```

```
[R2-LoopBack0]isis enable 10
```

配置完成后，在R2 上观察LSDB 中由自己生成的Level-1 LSP的详情。

```
[R2]display isis lsdb is-name R2 level-1 verbose
```

Database information for ISIS(10)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0000.0000.0002.00-00*	0x00000011	0x5439	1165
123 1/0/0			

.....

IP-Internal 10.0.23.0 255.255.255.0 COST: 10

IP-Internal 10.0.2.2 255.255.255.255 COST: 0

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R2的LSDB中拥有了自己生成的描述10.0.2.2/32网络的LSP。

查看R1的IS-IS的路由表。

<R1>display isis route

Route information for ISIS(10)

ISIS(10) Level-1 Forwarding Table

IPV4 Destination	IntCost	ExtCost	ExitInterface	NextHop	Flags

0.0.0.0/0	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.23.0/24	20	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.2.2/32	10	NULL	GE0/0/0	10.0.12.2	A/-/-/-
10.0.12.0/24	10	NULL	GE0/0/0	Direct	D/-/L/-

```
10.0.1.1/32    0    NULL  Loop0    Direct  D/-/L/-
```

```
.....
```

可以看到，R1的IS-IS路由表中现在拥有了去往10.0.2.2/32网络的路由。

在R3上观察Level-2的IS-IS路由表。

```
<R3>display isis route level-2
```

```
Route information for ISIS(10)
```

```
-----
```

```
ISIS(10) Level-2 Forwarding Table
```

```
-----
```

```
IPV4 Destination  IntCost  ExtCost  ExitInterface  NextHop  Flags
```

```
-----  
-----
```

```
10.0.3.3/32      0      NULL   Loop0        Direct   D/-/L/-
```

```
10.0.23.0/24     10     NULL   S1/0/0       Direct   D/-/L/-
```

```
10.0.2.2/32      10     NULL   S1/0/0       10.0.23.2  A/-/-/-
```

```
10.0.12.0/24     20     NULL   S1/0/0       10.0.23.2  A/-/-/-
```

```
Flags: D-Direct, A-Added to URT, L-Advertised in LSPs, S-IGP Shortcut,
```

```
U-Up/Down Bit Set
```

观察发现，R3的Level-2的IS-IS路由表中竟然没有去往10.0.1.1/32网络的路由。查看R3的LSDB中由R2生成的LSP的详情。

[R3]display isis lsdb is-name R2 verbose

Database information for ISIS(10)

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0000.0000.0002.00-00	0x000000019	0xede6	1111	116

0/0/0

SOURCE R2.00

HOST NAME R2

NLPID IPV4

AREA ADDR 10

INTF ADDR 10.0.12.2

INTF ADDR 10.0.23.2

INTF ADDR 10.0.2.2

NBR	ID	R3.00	COST: 10

IP-Internal	10.0.12.0	255.255.255.0	COST: 10

IP-Internal	10.0.23.0	255.255.255.0	COST: 10

IP-Internal	10.0.2.2	255.255.255.255	COST: 0

Total LSP(s): 1

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以观察到，R3的LSDB中没有描述10.0.1.1/32网络的LSP。查看R2的LSDB中由R1生成的LSP的详情。

<R2>display isis lsdb is-name R1 verbose

Database information for ISIS(10)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0000.0000.0001.00-00	0x00000006	0xdd21	790	107
0/0/0				

.....

NBR	ID	R2.01	COST: 10

IP-Internal	10.0.1.1	255.255.255.255	COST: 0

IP-Internal	10.0.12.0	255.255.255.0	COST: 10

.....

可以看到，在R2的LSDB中拥有描述10.0.1.1/32网络的LSP。由于10.0.1.1/32网络是R2通过Level-1邻接关系从R1处获得的，所以很可能是R2上的IS-IS路由渗透出现了问题，导致了R3未能从R2获得关于10.0.1.1/32网络的LSP。

查看R2的IS-IS配置情况。

```
[R2]isis 10
```

```
[R2-isis-10]display this
```

```
[V200R003C00]
```

```
#
```

```
isis 10
```

```
network-entity 10.0000.0000.0002.00
```

```
is-name R2
```

```
undo import-route isis level-1 into level-2
```

```
area-authentication-mode md5 plain huawei
```

```
#
```

```
return
```

观察发现，原来R2上关闭了Level-1向Level-2的路由渗透功能。

在R2上开启相应的路由渗透。

```
[R2]isis 10
```

```
[R2-isis-10]import-route isis level-1 into level-2
```

然后，查看R3的LSDB中由R2生成的LSP的详情。

```
[R3]display isis lsdb is-name R2 verbose
```

```
Database information for ISIS(10)
```

```
-----
```

Level-2 Link State Database

LSPID ATT/P/OL	Seq Num	Checksum	Holdtime	Length

0000.0000.0002.00-00 0/0/0	0x00000019	0xede6	1111	116
.....				
IP-Internal	10.0.2.2	255.255.255.255	COST: 0	
IP-Internal	10.0.1.1	255.255.255.255	COST: 10	
Total LSP(s): 1				
.....				

可以看到，R3的LSDB中现在拥有了描述10.0.1.1/32网络的LSP。如果观察R3的IS-IS路由表，就会发现其中存在去往10.0.1.1/32网络的路由。至此，所有的故障都得以排除，实现了全网互通的网络需求。

思考

IS-IS的区域认证和接口认证的关系是什么？

第5章 IP组播

5.1 IP组播的基本概念

5.2 IGMP

5.3 PIM-DM

5.4 PIM-SM

5.5 PIM-SM的RP

5.6 RPF校验

5.1 IP 组播的基本概念

原理概述

IANA（Internet Assigned Numbers Authority）将IP地址分成了A、B、C、D、E 5类，其中的D类为组播IP地址，范围是224.0.0.0 ~ 239.255.255.255。关于IP地址的分类，以及关于单播IP地址、组播IP地址、广播IP地址的详细描述，请读者自行学习和了解。

一个IP报文，其目的地址如果是单播IP地址，则称为单播IP报文；如果是组播IP地址，则称为组播IP报文；如果是广播IP地址，则称为广播IP报文。发送IP报文时，如果发送的是单播IP报文，则这样的发送方式称为IP单播方式，简称IP单播；如果发送的是组播IP报文，则这样的发送方式称为IP组播方式，简称IP组播；如果发送的是广播IP报文，则这样的发送方式称为IP广播方式，简称IP广播。

IP 单播是一种点到点的通信模式，而 IP 组播则是一种点到多点的通信模式。一个发送者需要同时向多个接收者发送完全相同的信息时，如果采用单播方式，则网络需要传输大量的报文，相比之下，采用组播方式可以大大减少需要传输的报文数量，从而可以节约大量的网络资源。随着Internet的不断发展，电子商务、网络会议、视频点播、远程教学等服务大量兴起，这些服务大多符合点到多点的模式，特别适合于IP组播的应用。

在组播方式下，组播报文将沿着组播路由协议建立的树型路由从信息源传递到众多的终端用户。在这个过程中，只有该组播组的成员才能收到并处理该组播组的报文，而对于不是该组播组的成员，要么不能收到该组播组的报文，要么收到后直接丢弃。

虽然 IP 广播也是一种点到多点的通信模式，但相比之下，IP 组播总的来说更具优势。例如，IP组播是可以跨越网段的，而IP广播只能限制在一个网段内。另外，IP组播也比IP广播具有更好的安全性。

实验目的

- 理解IP 组播的基本原理和应用场景
- 观察IP 单播、组播、广播现象
- 掌握组播源的配置方法

实验内容

实验拓扑如图5-1所示，实验编址如表5-1所示。本实验模拟了一个简单的公司网络，R1是公司的一台网关设备，PC-4为网络管理员所使用的终端，直接连接到R1上，公司其他员工的终端通过交换机 S1 与 R1 相连。现在，公司需要通过在组播服务器Source-1 上播放视频对员工进行内部培训，除了少数人事部门的员工（使用的终端是PC-3）不需要观看此视频外，其他员工均需要观看学习。管理员需要通过测试和比较，在单播、广播、组播中选择一种最为合适的方式传输该视频数据。

实验拓扑

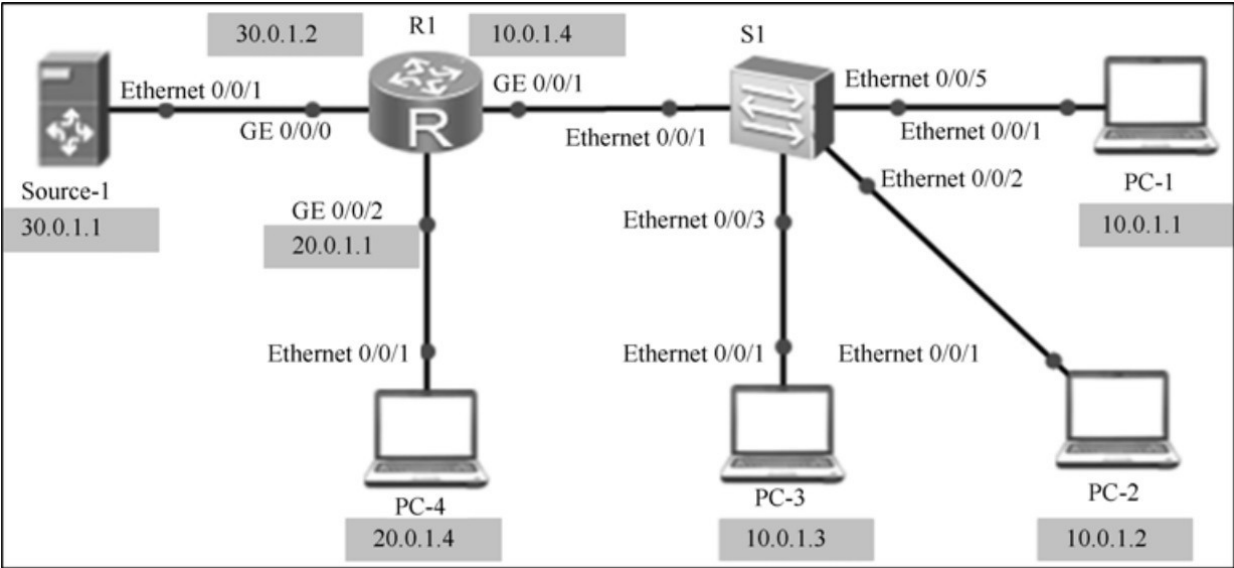


图5-1 IP组播的基本概念

实验编址表

表5-1 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2200)	GE 0/0/0	30.0.1.2	255.255.255.0	N/A
	GE 0/0/1	10.0.1.4	255.255.255.0	N/A
	GE 0/0/2	20.0.1.1	255.255.255.0	N/A
Source-1	Ethernet 0/0/1	30.0.1.1	255.255.255.0	30.0.1.2
PC-1	Ethernet 0/0/1	10.0.1.1	255.255.255.0	10.0.1.4
PC-2	Ethernet 0/0/1	10.0.1.2	255.255.255.0	10.0.1.4
PC-3	Ethernet 0/0/1	10.0.1.3	255.255.255.0	10.0.1.4
PC-4	Ethernet 0/0/1	20.0.1.4	255.255.255.0	10.0.1.4

实验步骤

1. 基本配置

根据图5-1和表5-1进行相应的基本配置，并使用ping命令检测R1与PC-1之间的连通性。

```
<R1>ping -c 1 10.0.1.1
```

```
PING 10.0.1.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.1.1: bytes=56 Sequence=1 ttl=128 time=70 ms
```

```
--- 10.0.1.1 ping statistics ---
```

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 70/70/70 ms

其余直连网段的连通性测试过程在此省略。

2. 观察单播方式

单播报文的目的IP地址只能标识一个唯一的接收者，只有该接收者才能收到并处理该IP报文；其他接收者要么不能收到此报文，要么收到后也不会进行处理，而是直接丢弃。

在R1上使用PC-1的单播IP地址10.0.1.1作为目的地址进行ping操作，然后分别在PC-1、PC-2和PC-3的Ethernet 0/0/1接口查看报文的接收情况，如图5-2、图5-3和图5-4所示。

<R1>ping 10.0.1.1

PING 10.0.1.1: 56 data bytes, press CTRL_C to break

Reply from 10.0.1.1: bytes=56 Sequence=1 ttl=128 time=20 ms

Reply from 10.0.1.1: bytes=56 Sequence=2 ttl=128 time=20 ms

Reply from 10.0.1.1: bytes=56 Sequence=3 ttl=128 time=20 ms

Reply from 10.0.1.1: bytes=56 Sequence=4 ttl=128 time=1 ms

Reply from 10.0.1.1: bytes=56 Sequence=5 ttl=128 time=20 ms

--- 10.0.1.1 ping statistics ---

5 packet(s) transmitted

5 packet(s) received

0.00% packet loss

round-trip min/avg/max = 1/16/20 ms

Filter: icmp		Expression... Clear Apply Save				
No.	Time	Source	Destination	Protocol	Length	Info
7	13.0420000	10.0.1.4	10.0.1.1	ICMP	98	Echo (ping) request
8	13.0420000	10.0.1.1	10.0.1.4	ICMP	98	Echo (ping) reply
10	13.5410000	10.0.1.4	10.0.1.1	ICMP	98	Echo (ping) request
11	13.5410000	10.0.1.1	10.0.1.4	ICMP	98	Echo (ping) reply
12	14.0250000	10.0.1.4	10.0.1.1	ICMP	98	Echo (ping) request
13	14.0250000	10.0.1.1	10.0.1.4	ICMP	98	Echo (ping) reply
14	14.5240000	10.0.1.4	10.0.1.1	ICMP	98	Echo (ping) request
15	14.5240000	10.0.1.1	10.0.1.4	ICMP	98	Echo (ping) reply
16	15.0230000	10.0.1.4	10.0.1.1	ICMP	98	Echo (ping) request
17	15.0230000	10.0.1.1	10.0.1.4	ICMP	98	Echo (ping) reply

图5-2 PC-1的Ethernet 0/0/1接口的报文情况

Filter: icmp		Expression... Clear Apply Save				
No.	Time	Source	Destination	Protocol	Length	Info

图5-3 PC-2的Ethernet 0/0/1接口的报文情况

Filter: icmp		Expression... Clear Apply Save				
No.	Time	Source	Destination	Protocol	Length	Info

图5-4 PC-3的Ethernet 0/0/1接口的报文情况

可以观察到，PC-2和PC-3没有收到ICMP消息，只有PC-1收到了。如果3个用户都需要接收到ping包，则需要在R1上分别使用3台PC的单播IP地址发送3次ping包。

如果多个用户需要获得相同的信息，那么在单播方式下，网络中需要传输的报文数量将和用户数量成正比。用户数量越多，网络中包含相同信息的报文数量就越多，这样既浪费网络设备的CPU资源又浪费网络的带宽资源。

3．观察广播方式

一个广播IP地址标识了某确定网段内的所有网络设备，该网段内每个网络设备都会收到并处理该网段的广播报文。

在R1上配置RIPv1协议，使R1通过广播方式发送RIP报文，然后在PC-1、PC-2和PC-3的Ethernet 0/0/1接口查看报文接收情况，如图5-5、图5-6和图5-7所示。

```
[R1]rip 1
```

```
[R1-rip-1]network 10.0.0.0
```

Filter: rip		Expression... Clear Apply				
No.	Time	Source	Destination	Protocol	Length	Info
3	4.13400000	10.0.1.4	255.255.255.255	RIPv1	66	Response
18	34.18000000	10.0.1.4	255.255.255.255	RIPv1	66	Response
31	61.19900000	10.0.1.4	255.255.255.255	RIPv1	66	Response
45	89.23300000	10.0.1.4	255.255.255.255	RIPv1	66	Response
59	116.25200000	10.0.1.4	255.255.255.255	RIPv1	66	Response
72	144.27000000	10.0.1.4	255.255.255.255	RIPv1	66	Response
89	177.31100000	10.0.1.4	255.255.255.255	RIPv1	66	Response
103	206.32700000	10.0.1.4	255.255.255.255	RIPv1	66	Response

图5-5 PC-1的Ethernet 0/0/1接口的报文情况

Filter: rip		Expression... Clear Apply				
No.	Time	Source	Destination	Protocol	Length	Info
7	12.58900000	10.0.1.4	255.255.255.255	RIPv1	66	Response
21	39.60800000	10.0.1.4	255.255.255.255	RIPv1	66	Response
34	67.64200000	10.0.1.4	255.255.255.255	RIPv1	66	Response
48	94.66100000	10.0.1.4	255.255.255.255	RIPv1	66	Response
62	122.67900000	10.0.1.4	255.255.255.255	RIPv1	66	Response
78	155.72000000	10.0.1.4	255.255.255.255	RIPv1	66	Response
92	184.73600000	10.0.1.4	255.255.255.255	RIPv1	66	Response
109	219.77400000	10.0.1.4	255.255.255.255	RIPv1	66	Response

图5-6 PC-2的Ethernet 0/0/1接口的报文情况

Filter: rip		Expression... Clear Apply				
No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	10.0.1.4	255.255.255.255	RIPv1	66	Response
15	27.01900000	10.0.1.4	255.255.255.255	RIPv1	66	Response
28	55.05300000	10.0.1.4	255.255.255.255	RIPv1	66	Response
42	82.07200000	10.0.1.4	255.255.255.255	RIPv1	66	Response
56	110.09000000	10.0.1.4	255.255.255.255	RIPv1	66	Response
72	143.13100000	10.0.1.4	255.255.255.255	RIPv1	66	Response
86	172.14700000	10.0.1.4	255.255.255.255	RIPv1	66	Response
103	207.18500000	10.0.1.4	255.255.255.255	RIPv1	66	Response

图5-7 PC-3的Ethernet 0/0/1接口的报文情况

可以看到，PC-1、PC-2和PC-3都收到了R1通过广播方式发送的RIP报文。

广播方式只在同一个网段中才有效，不能跨越网段。另外，广播方式是无法区分接收者的，这对于信息的安全性和服务的有偿性而言都是一个问题。

4. 观察组播方式

使用组播方式时，只有加入到该组播组的成员才能收到并处理该组播组的报文。对于不是该组播组的成员，要么不能收到该组播组的报文，要么收到后直接丢弃。

在R1上完成组播的基本配置，包括在全局模式下开启组播功能，在GE 0/0/0接口下开启组播功能，在GE 0/0/1和GE 0/0/2接口下开启组播功能及IGMP功能。关于组播的配置命令在后续实验中会有详细讲解，本次实验中读者只需按照步骤进行配置，通过观察实验现象理解组播的基本原理即可。

```
[R1]multicast routing-enable
[R1]interface GigabitEthernet 0/0/0
[R1-GigabitEthernet0/0/0]pim dm
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]pim dm
[R1-GigabitEthernet0/0/1]igmp enable
[R1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R1-GigabitEthernet0/0/2]pim dm
[R1-GigabitEthernet0/0/2]igmp enable
```

组播服务器 Source-1 需要使用 VLC 播放视频的方式来发送组播。首先，在网上下载并安装一个VLC软件，然后，在eNSP软件的主界面中点击右上方工具栏的设置按钮，在“工具设置”页面中设置VLC软件路径，如图5-8所示。



图5-8 eNSP工具页面

打开组播服务器Source-1的配置界面，在“基础配置”页面中配置IP地址、掩码等，然后点击“应用”，如图5-9所示。

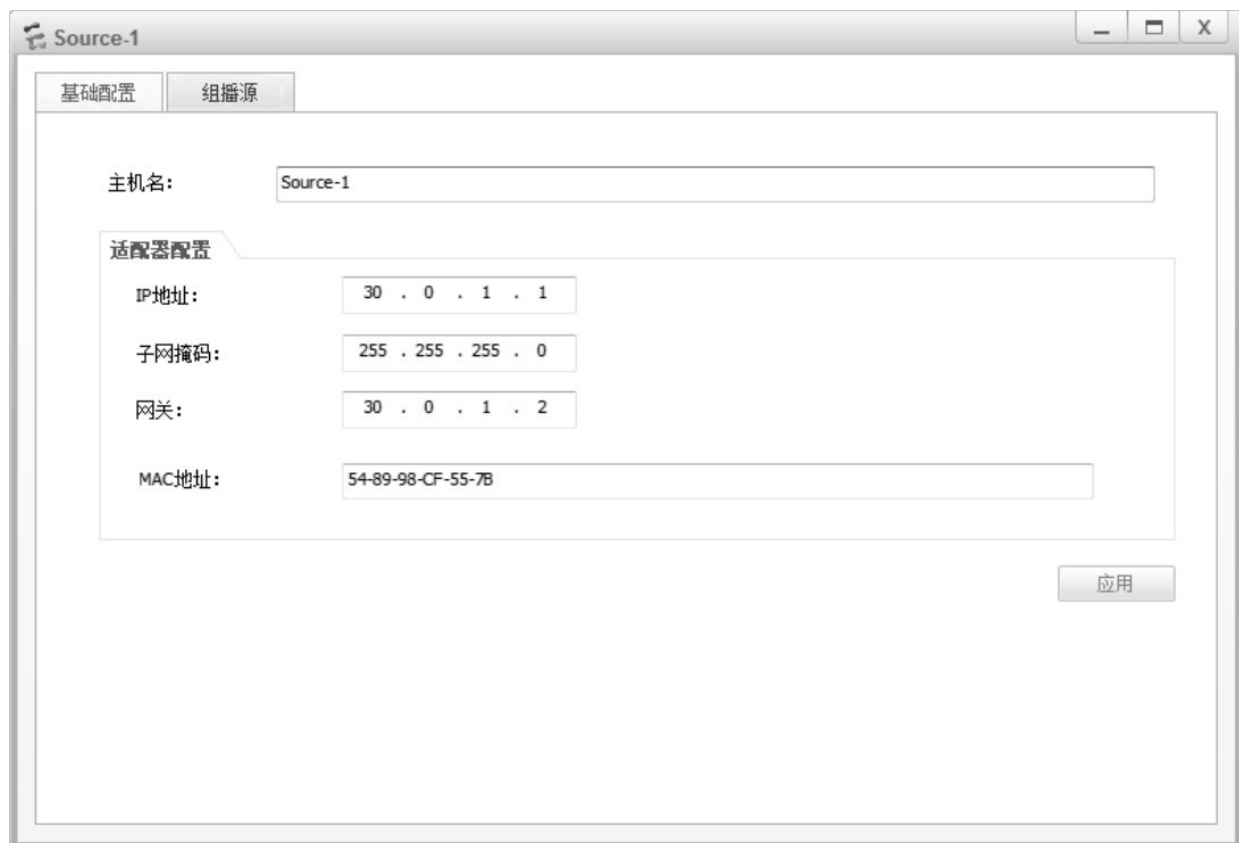


图5-9 Source-1“基础配置”页面

接下来，在“组播源”页面中配置组播组MAC地址01-00-5E-01-01-01和组播组IP地址224.1.1.1，如图5-10所示。

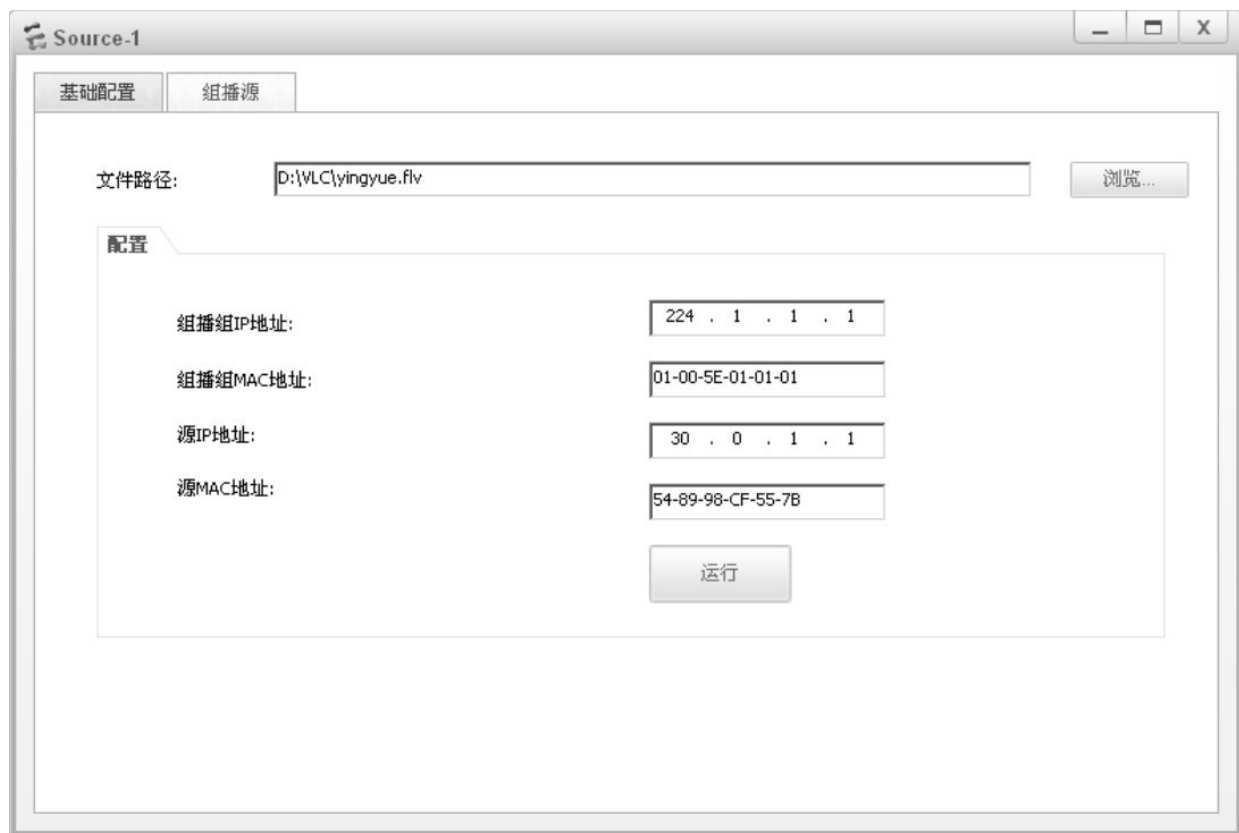


图5-10 Source-1“组播源”页面

在Source-1的设置界面中选择一个视频文件的路径后，点击“运行”播放视频。

在PC-4上使用IGMPv2加入224.1.1.1组播组，启动VLC，如图5-11所示。



图5-11 PC-4组播页面

可以观察到，在 PC-4 上已能播放组播源的视频了。另一方面，如果 PC-1 和 PC-2 不加入该组播组，则即使启动 VLC 也无视频播放出来，同时，在 PC-1 和 PC-2 的 Ethernet 0/0/1 接口抓取不到任何 UDP 的视频数据。

当 PC-1 和 PC-2 加入组播组 224.1.1.1 后，重新启动 VLC，则观察到可播放视频。此时，只有 PC-3 没加入该组播组，在 PC-3 上启动 VLC 也无视频播放。

至此，网络管理员应该可以得出结论了，即：应该选择 IP 组播的方式传输视频数据，以对公司的部分员工进行内部培训。

思考

IP 组播与通常所说的二层组播有什么关系？

5.2 IGMP

原理概述

IGMP (Internet Group Management Protocol , 因特网组管理协议) 是TCP/IP协议簇中负责组播成员管理的协议 , 其作用是在用户主机和与其直连的组播路由器之间建立和维护组播组成员关系。通过在用户主机和与其直连的组播路由器上配置和运行 IGMP , 可以实现主机动态地加入和离开组播组 , 以及组播路由器对本地网络中组播成员信息的动态管理。IGMP有3个版本 , 分别是IGMPv1、IGMPv2和IGMPv3。

IGMPv1主要基于查询和响应机制来完成组播组的管理。主机通过发送Report消息加入到某组播组 , 主机离开组播组时不发送离开报文 , 离开后再收到路由器发送的查询消息时不反馈Report消息 , 待维护组成员关系的定时器超时后 , 路由器会自动删除该主机的成员记录。

IGMPv2与IGMPv1基本相似 , 主要的不同点在于IGMPv2具有某些报文抑制机制 , 可以减少不必要的IGMP重复报文 , 从而节省网络带宽资源。另外 , 主机离开组播组时 , 会主动向路由器发送离开报文。

IGMPv1和IGMPv2报文中都只能携带组播组的信息 , 不能携带组播源的信息 , 所以主机只能选择加入某个组 , 而不能选择组播源 , 这一问题在IGMPv3中得到了解决。运行IGMPv3时 , 主机不仅能够选择组 , 还能根据需要选择组播源。主机发送的IGMPv3报文中可以包含多个组记录 , 每个组记录中可以包含多个组播源。

实验目的

- 理解IGMP的基本工作原理和应用场景
- 掌握在PC 和路由器上配置IGMP的方法
- 了解IGMP不同版本的区别

实验内容

实验拓扑如图5-12所示，实验编址如表5-2所示。本实验模拟了一个简单的公司网络，组播服务器Source-1存放了一些娱乐视频，组播服务器Source-2存放了一些学习视频，PC-1、PC-2、PC-3分别代表了公司人事部员工、市场部员工、研发部员工所使用的电脑。人事部员工只想看娱乐视频，且不在意带宽资源问题，所以在PC-1上配置的是IGMPv1；市场部员工希望尽量节约带宽资源，所以需要在PC-2上配置IGMPv2；研发部员工不允许观看娱乐视频，所以需要在PC-3上配置IGMPv3。除了终端电脑外，R1和R2上也需要进行相应的IGMP配置。

实验拓扑

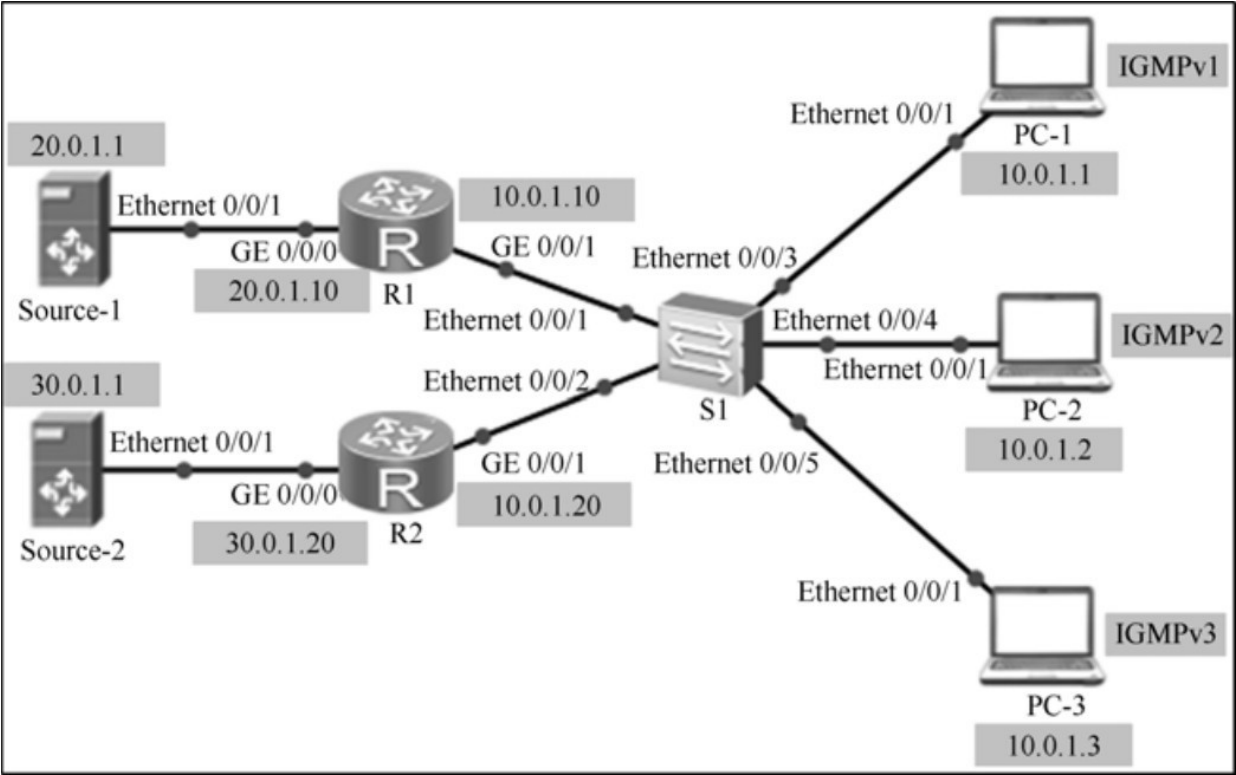


图5-12 IGMP

实验编址表

表5-2 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	20.0.1.10	255.255.255.0	N/A
	GE 0/0/1	10.0.1.10	255.255.255.0	N/A
R2(AR2220)	GE 0/0/0	30.0.1.20	255.255.255.0	N/A
	GE 0/0/1	10.0.1.20	255.255.255.0	N/A
Source-1	Ethernet 0/0/1	20.0.1.1	255.255.255.0	20.0.1.10
Source-2	Ethernet 0/0/1	30.0.1.1	255.255.255.0	30.0.1.20
PC-1	Ethernet 0/0/1	10.0.1.1	255.255.255.0	N/A
PC-2	Ethernet 0/0/1	10.0.1.2	255.255.255.0	N/A
PC-3	Ethernet 0/0/1	10.0.1.3	255.255.255.0	N/A

实验步骤

1．基本配置

根据图5-12和表5-2进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.1.20
```

```
PING 10.0.1.20: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.1.20: bytes=56 Sequence=1 ttl=255 time=100 ms
```

```
--- 10.0.1.20 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 100/100/100 ms
```

其余直连网段的连通性测试过程在此省略。

配置组播服务器 Source-1 的组播组 IP 地址为 224.1.1.1，组播组 MAC 地址为01-00-5E-01-01-01，如图5-13所示。

配置

组播组IP地址:	224 . 1 . 1 . 1
组播组MAC地址:	01-00-5E-01-01-01
源IP地址:	20 . 0 . 1 . 1
源MAC地址:	54-89-98-CF-9F-6A

运行

图5-13 配置Source-1

配置组播服务器 Source-2 的组播组 IP 地址为 224.1.1.1，组播组 MAC 地址为01-00-5E-01-01-01，如图5-14所示。

配置

组播组IP地址:	224 . 1 . 1 . 1
组播组MAC地址:	01-00-5E-01-01-01
源IP地址:	30 . 0 . 1 . 1
源MAC地址:	54-89-98-CF-C1-4F

运行

图5-14 配置Source-2

2．配置组播协议

由于公司需要通过组播方式发送视频，所以路由器需要开启组播功能，相应接口下也需要配置组播协议。此处照配即可，后续实验会详细讲解有关的配置命令。

```
[R1]multicast routing-enable
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]pim dm
```

```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]pim dm
```

```
[R2]multicast routing-enable
```

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]pim dm
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]pim dm
```

3 . 配置IGMPv1

由于公司员工需要在PC上使用IGMP加入相应的组播组，接收并观看组播视频，因此路由器用户侧的接口同样需要开启IGMP功能以处理IGMP消息。由于PC-1使用IGMPv1 播放来自Source-1 的娱乐视频，所以需要在R1的GE 0/0/1 接口配置IGMP，并修改版本为IGMPv1（注：缺省情况下为IGMPv2）。

```
[R1]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]igmp enable
```

```
[R1-GigabitEthernet0/0/1]igmp version 1
```

配置完成后，查看R1上的IGMP接口信息。

```
<R1>display igmp interface
```

```
Interface information of VPN-Instance: public net
```

```
GigabitEthernet0/0/1(10.0.1.10):
```

```
IGMP is enabled
```

```
Current IGMP version is 1
```

```
IGMP state: up
```

```
IGMP group policy: none
```

```
IGMP limit:-
```

```
Value of query interval for IGMP (negotiated):-
```

```
Value of query interval for IGMP (configured): 60 s
```

Value of other querier timeout for IGMP: 0 s

Value of maximum query response time for IGMP:-

Querier for IGMP: 10.0.1.10 (this router)

可以看到，R1 的接口GE 0/0/1 已经成功开启了IGMP功能，且版本为IGMPv1，查询消息的发送间隔为60s。

在R1上打开IGMP的Debug功能。

```
<R1>debugging igmp report
```

```
<R1>debugging igmp event
```

```
<R1>terminal monitor
```

```
<R1>terminal debugging
```

配置完成后，在PC-1上配置IGMPv1并加入组播组224.1.1.1。

```
<R1>
```

```
Sep 24 2013 11:53:15.216.1-05:13 R1 IGMP/7/REPORT:(public net):  
Received v1 report for group 224.1.1.1 on interface  
GigabitEthernet0/0/1(10.0.1.10) and source address is 10.0.1.1 (G081939)
```

可以看到，在R1 上收到了来自PC-1 的IGMP Report 消息，版本为v1，加入的组播组是224.1.1.1。在R1上查看IGMP组信息。

```
<R1>display igmp group
```

Interface group report information of VPN-Instance: public net

GigabitEthernet0/0/1(10.0.1.10):

Total 1 IGMP Group reported

Group Address	Last Reporter	Uptime	Expires
224.1.1.1	10.0.1.1	00:01:43	00:00:27

可以看到，R1上关于224.1.1.1这个组播组已经有了成员10.0.1.1。此时，使用Source-1播放组播视频，在PC-1上运行VLC软件即可观看视频。

在R1 上打开IGMP Debug 离组消息。

<R1>debugging igmp leave

在 PC-1 上的组播配置界面点击“离开”，然后查看 R1 上的Debug消息，发现 R1没有收到任何离组消息，这是因为运行IGMPv1时，组成员在离组时是不会发送离组消息的。路由器在3倍查询周期（180s）内如果没有收到某个组成员的Report消息，则认为该成员已经离组。

<R1>

Sep 24 2013 10:35:55.253.1-05:13 R1 IGMP/7/EVENT:(public net): Group(224.1.1.1) timeout, deleting record on interface GigabitEthernet0/0/1(10.0.1.10) (G016722)

<R1>

Sep 24 2013 10:35:55.253.2-05:13 R1 IGMP/7/EVENT:(public net): Deleting group(224.1.1.1) on interface GigabitEthernet 0/0/1(10.0.1.10) (G018254)

<R1>

Sep 24 2013 10:35:55.253.3-05:13 R1 IGMP/7/EVENT:(public net): Processing Aux Prune Alert for (*, 224.1.1.1) on interface GigabitEthernet0/0/1(10.0.1.10) (G011695)

4．配置IGMPv2

由于PC-2 使用IGMPv2加入组播组，所以需要在R1的GE 0/0/1和R2的GE 0/0/1接口配置IGMPv2。R1的GE 0/0/1接口之前已经配置了IGMPv1，所以只需修改版本号即可。

[R1]interface GigabitEthernet 0/0/1

[R1-GigabitEthernet0/0/1]igmp version 2

[R2]interface GigabitEthernet 0/0/1

[R2-GigabitEthernet0/0/1]igmp enable

配置完成后，在R1、R2上查看IGMP接口信息。

[R1]display igmp interface

Interface information of VPN-Instance: public net

GigabitEthernet0/0/1(10.0.1.10):

IGMP is enabled

Current IGMP version is 2

IGMP state: up

IGMP group policy: none

IGMP limit:-

Value of query interval for IGMP (negotiated):-

Value of query interval for IGMP (configured): 60 s

Value of other querier timeout for IGMP: 0 s

Value of maximum query response time for IGMP: 10 s

Querier for IGMP: 10.0.1.10 (this router)

[R2]display igmp interface

Interface information of VPN-Instance: public net

GigabitEthernet0/0/1(10.0.1.20):

IGMP is enabled

Current IGMP version is 2

IGMP state: up

IGMP group policy: none

IGMP limit:-

Value of query interval for IGMP (negotiated):-

Value of query interval for IGMP (configured): 60 s

Value of other querier timeout for IGMP: 0 s

Value of maximum query response time for IGMP: 10 s

Querier for IGMP: 10.0.1.10

可以看到，R2 的GE 0/0/1 接口运行的是IGMPv2，这是因为在接口下开启IGMP功能后，默认运行的是v2版本。

此外，上面的显示信息还表明IGMP的查询器（Querier for IGMP）为10.0.1.10，即R1。同一个网段上有多个组播路由器时，每个组播路由器都能从别的组播路由器和主机那里收到成员关系报告消息，但是只需要一台路由器发送成员资格查询消息，所以，这就需要一个路由器选举机制来确定一台路由器作为查询器。在IGMPv1中，查询器的选择由组播路由协议决定（后续实验将会介绍）；IGMPv2对此进行了简化，规定IP地址最小的将成为查询器。在PC-2的Ethernet 0/0/1接口查看报文情况，如图5-15和图5-16所示。

Filter: igmp Expression... Clear Apply Save						
No.	Time	Source	Destination	Protocol	Length	Info
26	46.519000	10.0.1.10	224.0.0.1	IGMPv2	60	Membership Query, general
58	106.704000	10.0.1.10	224.0.0.1	IGMPv2	60	Membership Query, general
91	166.734000	10.0.1.10	224.0.0.1	IGMPv2	60	Membership Query, general
123	226.732000	10.0.1.10	224.0.0.1	IGMPv2	60	Membership Query, general

图5-15 PC-2的Ethernet 0/0/1接口的报文情况

+	Frame 26: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
+	Ethernet II, Src: HuaweiTe_03:c5:ac (00:e0:fc:03:c5:ac), Dst: IPv4mcast_00:00:01 (01:00:5e:00:00:01)
+	Internet Protocol Version 4, Src: 10.0.1.10 (10.0.1.10), Dst: 224.0.0.1 (224.0.0.1)
+	Internet Group Management Protocol
	[IGMP Version: 2]
	Type: Membership Query (0x11)
	Max Response Time: 10.0 sec (0x64)
	Header checksum: 0xee9b [correct]
	Multicast Address: 0.0.0.0 (0.0.0.0)

图5-16 IGMP查询报文

可以观察到，IGMPv2的查询报文发送者为R1（10.0.1.10）。

在R1上打开IGMP的Debug功能。

<R1>debugging igmp report

<R1>debugging igmp leave

配置完成后，在PC-2上使用IGMPv2加入组播组224.1.1.1。

<R1>

Sep 24 2013 15:16:32.683.1-05:13 R1 IGMP/7/REPORT:(public net):
Received v2 report for group 224.1.1.1 on interface
GigabitEthernet0/0/1(10.0.1.10) and source address is 10.0.1.2 (G082908)

可以看到，R1 上收到了来自PC-2（10.0.1.2）的IGMPv2 Report 报文。在Source-1选择一个视频流，使用组播地址224.1.1.1进行播放，然后在PC-2上启动VLC即可接收并观看视频。

在PC-2的组播设置界面点击“离开”。

<R1>

```
Sep 24 2013 12:32:58.936.1-05:13 R1 IGMP/7/LEAVE:(public net):  
LEAVE received on interface GigabitEthernet0/0/1(10.0.1.10) has  
destination address(224.1.1.1), is not equal to all-router-multicast-  
address(224.0.0.2) (G083097)
```

<R1>

```
Sep 24 2013 12:32:58.936.2-05:13 R1 IGMP/7/LEAVE:(public net):  
Ignoring LEAVE received for non-member group(224.1.1.1) on interface  
GigabitEthernet0/0/1(10.0.1.10) (G083161)
```

<R1>

```
Sep 24 2013 12:32:58.936.3-05:13 R1 IGMP/7/LEAVE:(public net):  
Sending first last member query for group(224.1.1.1) on interface  
GigabitEthernet0/0/1(10.0.1.10) (G081113)
```

可以看到，R1收到了PC-2发送的离组消息。路由器收到离组消息后，会立即向网络中发送相应的特定组查询消息；如果在查询的最大响应时间（默认为1s）内没有收到关于该组的报告，则会再重复发送一次。如果两次查询后仍没有收到该组的成员报告，则认为该组播组的所有成员都已离开了。

5．配置IGMPv3

由于研发部员工只能观看、学习视频，不允许接收来自Source-1的娱乐视频，所以PC-3需要使用IGMPv3来选择组播源，拒绝来自Source-1的组播流。

在R1 和R2 的GE 0/0/1 接口下修改IGMP版本为v3。

[R1]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]igmp version 3
[R2]interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1]igmp version 3

在PC-3上配置IGMPv3，配置组播源IP起始地址为20.0.1.1，选择EXCLUDE模式，如图5-17所示。



图5-17 在PC-3上配置IGMPv3

配置完成后，在R1和R2 的GE 0/0/1 接口查看报文情况，然后在图5-17所示的界面中点击“发送”，显示如图5-18和图5-19所示。

Source	Destination	Protocol	Length	Info
10.0.1.3	224.0.0.22	IGMPv3	58	Membership Report / Join group 224.1.1.1, for source not {20.0.1.1}
10.0.1.10	224.0.0.1	IGMPv3	60	Membership Query, general

图5-18 R1的GE 0/0/1接口的报文情况

```

Frame 4: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface 0
Ethernet II, Src: HuaweiTe_cf:92:63 (54:89:98:cf:92:63), Dst: IPv4mcast_00:00:16 (01:00:5e:00:00:16)
Internet Protocol Version 4, Src: 10.0.1.3 (10.0.1.3), Dst: 224.0.0.22 (224.0.0.22)
Internet Group Management Protocol
  [IGMP Version: 3]
  Type: Membership Report (0x22)
  Header checksum: 0xe5f9 [correct]
  Num Group Records: 1
  Group Record : 224.1.1.1 Mode Is Exclude
    Record Type: Mode Is Exclude (2)
    Aux Data Len: 0
    Num Src: 1
    Multicast Address: 224.1.1.1 (224.1.1.1)
    Source Address: 20.0.1.1 (20.0.1.1)

```

图5-19 PC-3发送的IGMPv3消息

可以观察到，R1和R2都收到了来自PC-3发送的IGMPv3消息，模式为Exclude，组播地址为224.1.1.1，组播源地址为20.0.1.1，其含义就是拒绝接收来自20.0.1.1发送的组播组地址为224.1.1.1的信息。

6. 观察IGMP版本兼容性

IGMP的兼容性是指较高版本的组播路由器可以兼容较低版本的主机，例如，v2版本的组播路由器可以正确处理v1主机的加入，v3版本的组播路由可以正确处理v1和v2主机的加入。组播路由器收到较低版本主机的IGMP加入报文后，会自动降低至并工作在相应的主机版本。

工作在v2或v3的组播路由器收到IGMPv1主机发送的Report报文时，会自动把该组的兼容模式降到v1版本。

查看R1的IGMP版本。

```
<R1>display igmp interface
```

```
Interface information of VPN-Instance: public net
```

```
GigabitEthernet0/0/1(10.0.1.10):
```

```
IGMP is enabled
```

```
Current IGMP version is 3
```

```
IGMP state: up
```

```
.....
```

可以看到，R1上现在运行的是IGMPv3。在PC-1上使用IGMPv1加入组224.1.1.1，在PC-2上使用IGMPv2加入组224.1.1.1，然后在R1上再

查看IGMP信息。

```
<R1>display igmp interface
```

Interface information of VPN-Instance: public net

GigabitEthernet0/0/1(10.0.1.10):

IGMP is enabled

Current IGMP version is 3

IGMP state: up

IGMP group policy: none

IGMP limit:-

Value of query interval for IGMP (negotiated): 60 s

Value of query interval for IGMP (configured): 60 s

Value of other querier timeout for IGMP: 0 s

Value of maximum query response time for IGMP: 10 s

Querier for IGMP: 10.0.1.10 (this router)

Total 2 IGMP Groups reported

```
<R1>display igmp group
```

Interface group report information of VPN-Instance: public net

GigabitEthernet0/0/1(10.0.1.10):

Total 2 IGMP Groups reported

Group Address	Last Reporter	Uptime	Expires
224.1.1.1	10.0.1.2	00:01:59	00:00:20

可以看到，R1可以接收并处理v2和v3版本的IGMP消息，并使得PC-1和PC-2成功地加入组播组。同样，如果路由器配置IGMPv2，则主机使用IGMPv1和IGMPv2都是可以成功加入组播组的。

现在，把R1的IGMP版本改为v1。

```
[R1]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]igmp version 1
```

然后，PC-2使用IGMPv2加入组224.1.1.1后，在R1上查看IGMP信息。

```
<R1>display igmp interface
```

Interface information of VPN-Instance: public net

GigabitEthernet0/0/1(10.0.1.10):

IGMP is enabled

Current IGMP version is 1

IGMP state: up

.....

```
<R1>display igmp group
```

```
<R1>
```

可以看到，R1运行IGMPv1时，PC-2使用IGMPv2是无法加入组播组的，R1上IGMP Group为空。同样，如果路由器运行的是IGMPv2，则主机使用IGMPv3是无法加入组播组的。

另外，需要注意的是，工作在IGMPv3的组播路由器收到v2版本的Report报文时，会自动把该组的兼容模式降到v2。但是，在这种情况下，路由器会忽略IGMPv3的BLOCK报文、IGMPv3的TO_IN报文以及IGMPv3的TO_EX报文的源列表，即抑制了IGMPv3对组播源的选择功能。所以，在本实验中，为了让PC-3能正常使用IGMPv3功能，需要在PC-1和PC-2上都修改使用IGMPv3加入组播组。

思考

IGMP是一个路由协议吗？

5.3 PIM-DM

原理概述

树（Tree）和图（Graph）是计算机科学领域中两个常用的概念，前者具有层次化结构，而后者没有。从组播的角度来看，网络可以抽象为Tree，也称为组播树（Multicast Tree）；从单播的角度来看，网络可以抽象为Graph。组播树又可以分为两大类，一类称为Source-Based Tree，另一类称为Group-Shared Tree。

一棵Source-Based Tree是由组播源和组成员共同决定的。例如，对于一个特定的组播组，如果组成员的分布已经确定了，则组播源的位置不同将会导致形成不同的Source-Based Tree。同样，对于一个特定的组播组，如果组播源的位置已经确定了，则组成员的不同分布也将会导致产生不同的Source-Based Tree。一棵Group-Shared Tree仅仅是由组成员的分布完全决定的，而与组播源的位置无关。

PIM-DM（Protocol Independent Multicast Dense Mode）和PIM-SM（Protocol Independent Multicast Sparse Mode）是两个常见的组播路由协议，前者基于Source-Based Tree，后者基于Group-Shared Tree。

PIM-DM主要采用扩散-剪枝的方式来转发组播数据流。对于组播组成员稀少的网络，PIM-DM会产生大量的剪枝报文，而如果网络规模较大，则扩散-剪枝的周期就会比较长，因此PIM-DM一般适合于规模较小、组播组成员比较密集的网络。

PIM-DM首先假设网络中的每个子网都存在至少一个组成员，并将组播数据包从组播源扩散到网络中的所有路由器，然后，对于实际上没有组成员的分支进行剪枝操作。所谓剪枝（Prune），就是路由器向上游节点发送剪枝消息，通知上游节点不用再转发组播数据到该分支。上游节点收到剪枝消息后，会将相应的接口从其组播转发表项（S，G）中删除，只保留包含组成员的分支，这样便可减少网络资源的消耗。另外，各个被剪枝的节点同时还提供了超时机制，当剪枝超时后（默认为210s）将重新开始扩散-剪枝过程。被裁剪的分支如果临

时有组播数据转发需求，也可以使用嫁接（Graft）机制主动请求恢复组播数据的转发。

周期性的扩散-剪枝行为是PIM-DM的一个重要特征，通过这样的行为，PIM-DM可以构建并动态地维护一棵从组播源到组成员的单向无环的SPT（Shortest Path Tree）。SPT是以组播源为根、组播组成员为枝叶的从组播源到组成员的一棵最短路径树，此树也就是组播数据的转发路径。组播数据的转发中会出现上游接口和下游接口这两个概念，路由器收到组播数据的接口称为上游接口，转发组播数据的接口称为下游接口。

在PIM-DM网络中，路由器需要周期性地发送Hello消息来发现邻居并维护PIM邻居关系。此外，Hello消息还有一个重要的作用：路由器会通过比较Hello消息中携带的优先级和IP地址，为多路由器网段选举出DR（Designated Router），并以它作为IGMPv1中的查询器。

实验目的

- 理解PIM-DM 的应用场景
- 掌握PIM-DM 的基本配置
- 理解PIM-DM 中剪枝和嫁接的原理
- 理解PIM-DM 中的Assert 机制

实验内容

实验拓扑如图5-20所示，实验编址如表5-3所示。本实验模拟了一个公司网络场景，包含了5台路由器、一台交换机、一台组播服务器Source-1和3台终端电脑。所有的路由器都运行OSPF，并且都位于区域0。管理员需要在网络中部署PIM-DM，从而实现以组播的方式向员工播放培训视频。

实验拓扑

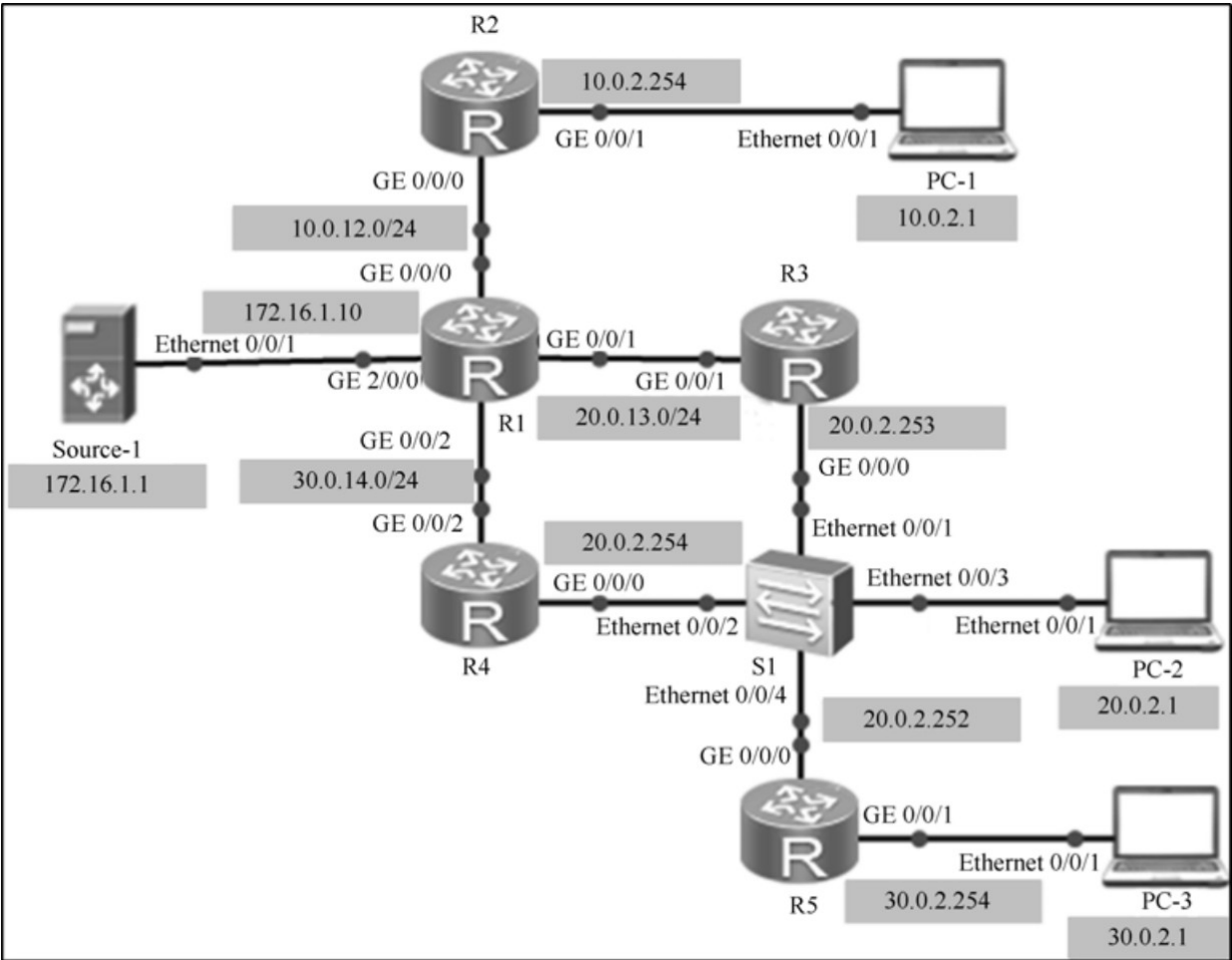


图5-20 PIM-DM

实验编址表

表5-3 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2200)	GE 2/0/0	172.16.1.10	255.255.255.0	N/A
	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	20.0.13.1	255.255.255.0	N/A
	GE 0/0/2	30.0.14.1	255.255.255.0	N/A
R2(AR2200)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.2.254	255.255.255.0	N/A
R3(AR2200)	GE 0/0/0	20.0.2.253	255.255.255.0	N/A
	GE 0/0/1	20.0.13.3	255.255.255.0	N/A
R4(AR2200)	GE 0/0/0	20.0.2.254	255.255.255.0	N/A
	GE 0/0/2	30.0.14.4	255.255.255.0	N/A
R5(AR2200)	GE 0/0/0	20.0.2.252	255.255.255.0	N/A
	GE 0/0/1	30.0.2.254	255.255.255.0	N/A
Source-1	Ethernet 0/0/1	172.16.1.1	255.255.255.0	172.16.1.10
PC-1	Ethernet 0/0/1	10.0.2.1	255.255.255.0	10.0.2.254
PC-2	Ethernet 0/0/1	20.0.2.1	255.255.255.0	20.0.2.254
PC-1	Ethernet 0/0/1	30.0.2.1	255.255.255.0	30.0.2.254

实验步骤

1．基本配置

根据图5-20和表5-3进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 20.0.13.3
```

```
PING 20.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 20.0.13.3: bytes=56 Sequence=1 ttl=255 time=20 ms
```

```
--- 20.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 20/20/20 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置IGP

在每台路由器上配置OSPF路由协议，并通告直连网段。

```
[R1]ospf 1
```

```
[R1-ospf-1]area 0
[R1-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
[R1-ospf-1-area-0.0.0.0]network 20.0.13.0 0.0.0.255
[R1-ospf-1-area-0.0.0.0]network 30.0.14.0 0.0.0.255
[R1-ospf-1-area-0.0.0.0]network 172.16.1.0 0.0.0.255
[R2]ospf 1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.2.0 0.0.0.255
[R3]ospf 1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 20.0.13.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 20.0.2.0 0.0.0.255
[R4]ospf 1
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 20.0.2.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 30.0.14.0 0.0.0.255
[R5]ospf 1
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]network 20.0.2.0 0.0.0.255
[R5-ospf-1-area-0.0.0.0]network 30.0.2.0 0.0.0.255
```

配置完成后，在R1上查看OSPF邻居建立情况。读者可自行查看其他路由器上OSPF邻居建立情况。

```
<R1>display ospf peer brief
```

```
      R1 OSPF Process 1 with Router ID 172.16.1.10
```

```
      Peer Statistic Information
```



```

20.0.2.0/24      OSPF      10    2      D      20.0.13.3
GigabitEthernet0/0/1
      OSPF      10    2      D      30.0.14.4  GigabitEthernet0/0/2
20.0.13.0/24      Direct    0      0      D      20.0.13.1
GigabitEthernet0/0/1
20.0.13.1/32      Direct    0      0      D      127.0.0.1
GigabitEthernet0/0/1
20.0.13.255/32    Direct    0      0      D      127.0.0.1
GigabitEthernet0/0/1
30.0.2.0/24      OSPF      10    3      D      20.0.13.3
GigabitEthernet0/0/1
      OSPF      10    3      D      30.0.14.4  GigabitEthernet0/0/2
30.0.14.0/24      Direct    0      0      D      30.0.14.1
GigabitEthernet0/0/2

```

.....

可以看到，R1已经收到了其他网段的路由信息。至此，网络已通过OSPF实现了互联互通。

3．配置PIM-DM

在所有路由器上开启组播功能，并在每台路由器的每个接口下配置命令pim dm。

```

[R1]multicast routing-enable
[R1]interface GigabitEthernet 0/0/0
[R1-GigabitEthernet0/0/0]pim dm
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]pim dm
[R1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R1-GigabitEthernet0/0/2]pim dm

```

```
[R1-GigabitEthernet0/0/2]interface GigabitEthernet 2/0/0
[R1-GigabitEthernet2/0/0]pim dm
[R2]multicast routing-enable
[R2]interface GigabitEthernet 0/0/0
[R2-GigabitEthernet0/0/0]pim dm
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1]pim dm
[R3]multicast routing-enable
[R3]interface GigabitEthernet 0/0/0
[R3-GigabitEthernet0/0/0]pim dm
[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R3-GigabitEthernet0/0/1]pim dm
[R4]multicast routing-enable
[R4]interface GigabitEthernet 0/0/0
[R4-GigabitEthernet0/0/0]pim dm
[R4-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/2
[R4-GigabitEthernet0/0/2]pim dm
[R5]multicast routing-enable
[R5]interface GigabitEthernet 0/0/0
[R5-GigabitEthernet0/0/0]pim dm
[R5-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R5-GigabitEthernet0/0/1]pim dm
```

配置完成后，在R1上查看PIM邻居关系的建立情况。读者可自行查看其他路由器的PIM邻居关系的建立情况。

```
<R1>display pim neighbor
  VPN-Instance: public net
  Total Number of Neighbors = 3
```


Neighbor	Interface	Uptime	Expires	Dr-Priority	BFD-Session
10.0.12.2	GE0/0/0	00:02:15	00:01:29	1	N
20.0.13.3	GE0/0/1	00:01:48	00:01:26	1	N
30.0.14.4	GE0/0/2	00:01:25	00:01:19	1	N

可以看到，R1已经与R2、R3、R4成功地建立了PIM邻居关系。

在R2、R3、R4、R5的用户侧接口下使能IGMP。

```
[R2]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]igmp enable
```

```
[R3]interface GigabitEthernet 0/0/0
```

```
[R3-GigabitEthernet0/0/0]igmp enable
```

```
[R4]interface GigabitEthernet 0/0/0
```

```
[R4-GigabitEthernet0/0/0]igmp enable
```

```
[R5]interface GigabitEthernet 0/0/1
```

```
[R5-GigabitEthernet0/0/1]igmp enable
```

查看R2的IGMP接口信息。读者可自行查看其他路由器的IGMP接口信息。

```
<R2>display igmp interface
```

```
Interface information of VPN-Instance: public net
```

```
GigabitEthernet0/0/1(10.0.2.254):
```

```
IGMP is enabled
```

```
Current IGMP version is 2
```

```
IGMP state: up
```

```
IGMP group policy: none
```

```
IGMP limit:-
```

```
Value of query interval for IGMP (negotiated):-
```

```
Value of query interval for IGMP (configured): 60 s
```

Value of other querier timeout for IGMP: 0 s

Value of maximum query response time for IGMP: 10 s

Querier for IGMP: 10.0.2.254 (this router)

可以看到，R2 的GE 0/0/1 接口的IGMP功能已经使能。

在组播服务器Source-1上使用组播地址224.1.1.1播放视频，当路由器接收到组播源发送的组播数据后便会自动生成组播路由。查看R1的组播路由表。

```
<R1>display pim routing-table
```

```
VPN-Instance: public net
```

```
Total 0 (*, G) entry; 1 (S, G) entry
```

```
(172.16.1.1, 224.1.1.1)
```

```
Protocol: pim-dm, Flag: LOC ACT
```

```
UpTime: 00:04:50
```

```
Upstream interface: GigabitEthernet2/0/0
```

```
Upstream neighbor: NULL
```

```
RPF prime neighbor: NULL
```

```
Downstream interface(s) information:
```

```
Total number of downstreams: 1
```

```
1: GigabitEthernet0/0/0
```

```
Protocol: pim-dm, UpTime: 00:03:51, Expires: never
```

可以看到，组播源地地址为 172.16.1.1，组播组的 IP 地址为 224.1.1.1，R1的上游接口为GE 2/0/0。

Source-1使用组播地址224.1.1.1播放视频后，由于PC-1并没有在第一时间加入该组播组观看视频，所以R2没有收到IGMP加入消息，于是R2会认为自己没有连接任何组成员。因此，R2在收到R1发送的组播数据包后，会向R1发送剪枝消息。R1收到剪枝消息后，会立即停止向R2

发送组播数据包。在R2 的GE 0/0/0 接口查看剪枝消息的数据包，如图5-21所示。

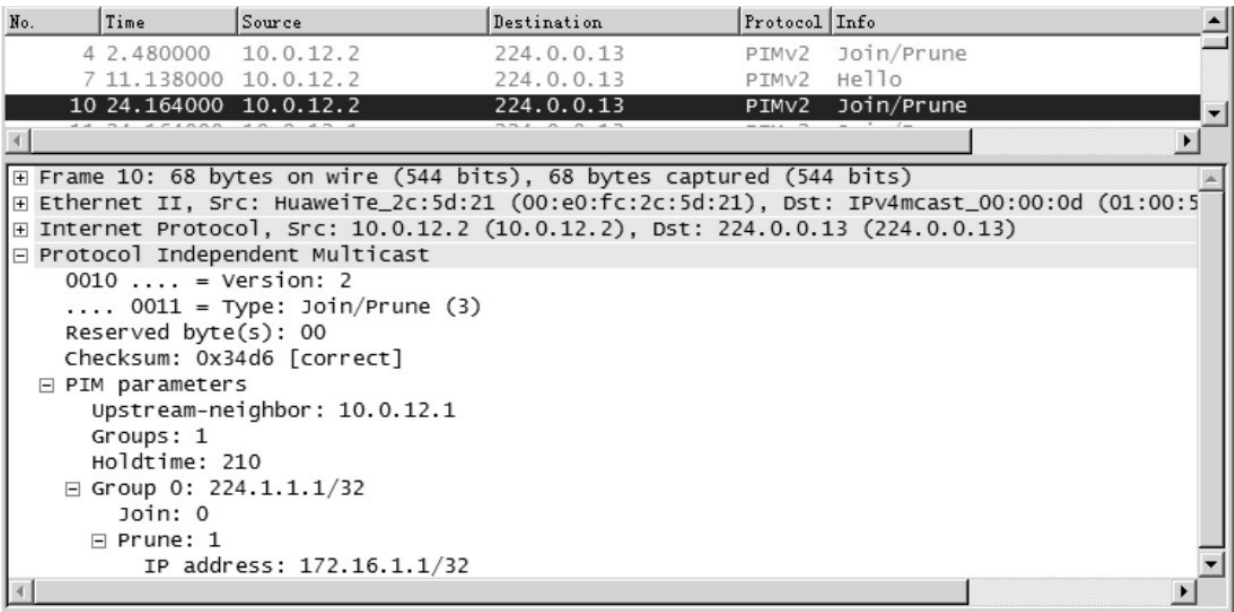


图5-21 R2的GE 0/0/0接口的报文情况

接下来，让PC-1使用IGMPv2加入组播组224.1.1.1，如图5-22所示；R2收到PC-1的加入消息后，会立即发送Graft消息给R1，R1收到后会立即开始重新转发组播数据包给R2（如图5-23所示），R2再转发给PC-1。在PC-1上启动VLC，便可观看视频了。



图5-22 PC-1使用IGMPv2加入组播组224.1.1.1

No.	Time	Source	Destination	Protocol	Length	Info
6523	119.294000	10.0.12.2	10.0.12.1	PIMv2	68	Graft
6524	119.309000	10.0.12.1	10.0.12.2	PIMv2	68	Graft-Ack
6525	119.309000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
6526	119.340000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
6527	119.372000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
6528	119.387000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
6529	119.403000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
6530	119.434000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
6531	119.465000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
6532	119.481000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0

图5-23 Graft/Graft-A消息和组播视频流

从图5-23中可以看到，R2发送了Graft消息后，R1立即回应了一个Graft-Ack消息，然后就开始向R2转发UDP的组播数据流。

4．观察PIM-DM中DR的选举

为了便于观察 PIM-DM 中的 DR 选举，假定 PC-2 将使用 IGMPv1 加入组播组224.1.1.1，为此，在R3和R4上修改IGMP为版本1。

```
[R3]interface GigabitEthernet 0/0/0
```

```
[R3-GigabitEthernet0/0/0]igmp version 1
```

```
[R4]interface GigabitEthernet 0/0/0
```

```
[R4-GigabitEthernet0/0/0]igmp version 1
```

由于PC-2通过交换机同时连接到了R3和R4，为了避免收到重复的IGMP查询消息，R3和R4之间需要选举出一个查询器，查询报文仅由查询器发送。IGMPv1中查询器的选举由组播路由协议决定，PIM-DM选举出来的DR即为IGMPv1中的查询器。

```
<R3>display pim interface
```

```
VPN-Instance: public net
```

Interface	State	NbrCnt	HelloInt	DR-Pri	DR-Address
GE0/0/0	up	2	30	1	20.0.2.254
GE0/0/1	up	1	30	1	20.0.13.3 (local)

```
<R4>display pim interface
```

```
VPN-Instance: public net
```

Interface	State	NbrCnt	HelloInt	DR-Pri	DR-Address
GE0/0/0	up	2	30	1	20.0.2.254 (local)
GE0/0/2	up	1	30	1	30.0.14.4 (local)

从上面的显示信息可以看到，PIM-DM选举出的DR为R4（优先级一样时，IP地址较大者为DR）。

接下来，验证一下IGMPv1的查询路由器是否就是PIM-DM选举出来的DR。在R3和R4上查看关于查询器（Querier）的情况。

```
<R3>display igmp interface
```

```
Interface information of VPN-Instance: public net
```

```
GigabitEthernet0/0/0(20.0.2.253):
```

```
.....
```

```
Value of maximum query response time for IGMP:-
```

```
Querier for IGMP: 20.0.2.254
```

```
<R4>display igmp interface
```

```
Interface information of VPN-Instance: public net
```

```
GigabitEthernet0/0/0(20.0.2.254):
```

```
.....
```

```
Value of maximum query response time for IGMP:-
```

```
Querier for IGMP: 20.0.2.254 (this router)
```

可以看到，IGMPv1的查询器就是PIM-DM选举出来的DR。

5．观察PIM-DM中的Assert机制

由于R3和R4从上游接收到组播报文后，都会向下游网络转发该组播报文，这样就会导致下游节点R5收到两份完全相同的组播报文。为了避免这种情况的发生，PIM-DM采用了Assert机制来选定一个唯一的转发者，即：对于一个特定的组播组，如果同一网段上存在多个上游路由器，则这些上游路由器中到组播源的路径开销最小者将被选举为转发者；若开销相同，则IP地址最大的路由器将被选举为转发者。只有转发者才能向该网段转发相应的组播数据报文，其他落选路由器应裁剪掉对应的接口，禁止向该网段转发相应的组播数据报文。

在PC-3上使用IGMPv2加入组播组224.1.1.1，在R5的GE 0/0/0接口查看报文情况，如图5-24所示。

No.	Time	Source	Destination	Protocol	Length	Info
10	9.25100000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
11	9.25100000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
12	9.25100000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
13	9.25100000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
14	9.26700000	20.0.2.253	224.0.0.13	PIMv2	60	Assert
15	9.26700000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
16	9.26700000	20.0.2.254	224.0.0.13	PIMv2	60	Assert
17	9.29800000	20.0.2.254	224.0.0.13	PIMv2	60	Assert
18	9.29800000	20.0.2.253	224.0.0.13	PIMv2	68	Join/Prune
19	9.31400000	20.0.2.253	224.0.0.13	PIMv2	68	Join/Prune
20	9.37600000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0

图5-24 R5的GE 0/0/0接口报文情况

从图 5-24 可以看到，R5 收到了来自 R3（20.0.2.253）和 R4（20.0.2.254）发送的Assert报文。由于R3和R4去往组播源的路径开销相同，所以R4凭借有较大的IP地址而成为了转发者。图5-25显示的是R4转发给R5的组播视频流。

No.	Time	Source	Destination	Protocol	Length	Info
18	9.29800000	20.0.2.253	224.0.0.13	PIMv2	68	Join/Prune
19	9.31400000	20.0.2.253	224.0.0.13	PIMv2	68	Join/Prune
20	9.37600000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
21	9.50100000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0
22	9.57900000	172.16.1.1	224.1.1.1	UDP	1370	Source port: avt-profile-2 Destination port: 0

图5-25 R4转发给R5的组播视频流

```
<R5>display pim routing-table
```

```
VPN-Instance: public net
```

```
Total 0 (*, G) entry; 1 (S, G) entry
```

```
(172.16.1.1, 224.1.1.1)
```

```
Protocol: pim-dm, Flag: ACT
```

```
UpTime: 00:19:50
```

```
Upstream interface: GigabitEthernet0/0/0
```

```
Upstream neighbor: 20.0.2.254
```

```
RPF prime neighbor: 20.0.2.254
```

```
.....
```

可以看到，R5的上游节点地址为20.0.2.254，即R4。

思考

PIM-DM中Assert机制的作用是什么？

5.4 PIM-SM

原理概述

PIM-SM是一种基于Group-Shared Tree的组播路由协议，与PIM-DM不同，它适合于组播组成员分布广泛而稀疏的大型网络。Group-Shared Tree分为两种：一种被称为Steiner Tree，另一种被称为Rendezvous Point Tree（简称RPT），PIM-SM采用的组播树是RPT。

RPT 是一棵以汇聚点RP（Rendezvous Point）路由器为根，以直连有组成员的路由器为叶子的组播树。RP 是一个组播供求信息的汇聚中心，它需要处理组播源端 DR（Designated Router）发送的组播注册消息及用户端DR 发送的组播加入消息。

RP是PIM-SM网络中的一台至关重要的路由器，网络中的所有PIM路由器都必须知道谁是RP。当网络中出现活跃的组播源（组播源向某组播组发送第一个组播数据）时，组播源端DR会将此组播数据封装在注册消息中并以单播形式发往RP，RP收到此消息后会立即创建相应的（S，G）组播路由表项。当网络中出现活跃的组播用户（用户主机通过IGMP加入某组播组G）时，用户端DR会向RP发送加入组播组G的消息，在该消息去往RP的路径上的每台路由器都创建（*，G）表项，由此便生成了一棵以RP为根的RPT。当网络中有活跃的组播用户时，组播报文先被封装在单播报文中从组播源发往RP，然后RP再将组播报文沿RPT转发给组播用户。若网络中没有活跃的组播用户时，组播数据到达RP后就停止了，不会再向下转发。

显然，RPT 并非是一棵SPT（Shortest Path Tree），经由RP 的转发路径可能不是从组播源到组播用户之间的最短路径。为了提高组播转发效率，PIM-SM 在实际部署时，通常都会允许从RPT切换到SPT。

PIM路由器会通过PIM-Hello消息来发现PIM邻居、协调各项协议参数、维护邻居关系。PIM-Hello消息的目的IP地址是组播地址224.0.0.13（表示同一网段中的所有PIM路由器），源地址为发送接口的IP地址，TTL值为1。另外，PIM-Hello的一个重要作用就是用来选举DR。在PIM-SM中，DR分为两种：组播源网段中的DR称为组播源端DR，它负责向RP发送组播注册消息；组成员网段中的DR称为用户端DR，它负责向RP发送组播加入消息。

实验目的

- 理解PIM-SM的应用场景
- 掌握PIM-SM的基本配置
- 理解PIM-SM中RPT到SPT的切换
- 理解组播源端DR和用户端DR的作用

实验内容

实验拓扑如图5-26所示，实验编址如表5-4所示。本实验模拟了一个公司网络场景，包含了6台路由器、3台交换机、一台组播服务器Source-1和两台终端电脑。所有路由器都运行OSPF，并且都位于区域0。管理员需要在网络中部署PIM-SM，从而实现以组播的方式向员工播放培训视频。

实验拓扑

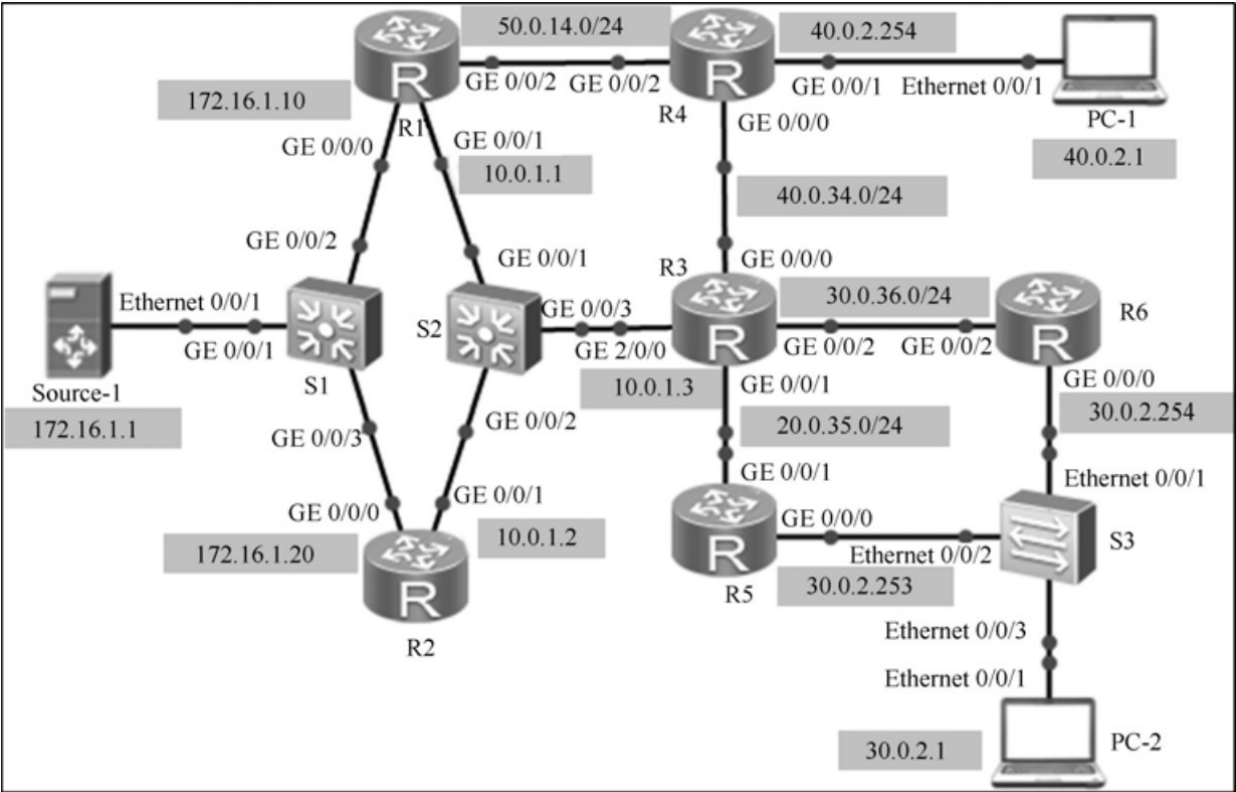


图5-26 PIM-SM

实验编址表

表5-4 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2200)	GE 0/0/0	172.16.1.10	255.255.255.0	N/A
	GE 0/0/1	10.0.1.1	255.255.255.0	N/A
	GE 0/0/2	50.0.14.1	255.255.255.0	N/A

(续表)

设备	接口	IP 地址	子网掩码	默认网关
R2(AR2200)	GE 0/0/0	172.16.1.20	255.255.255.0	N/A
	GE 0/0/1	10.0.1.2	255.255.255.0	N/A
R3(AR2200)	GE 2/0/0	10.0.1.3	255.255.255.0	N/A
	GE 0/0/0	40.0.34.3	255.255.255.0	N/A
	GE 0/0/1	20.0.35.3	255.255.255.0	N/A
	GE 0/0/2	30.0.36.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2200)	GE 0/0/0	40.0.34.4	255.255.255.0	N/A
	GE 0/0/1	40.0.2.254	255.255.255.0	N/A
	GE 0/0/2	50.0.14.4	255.255.255.0	N/A
R5(AR2200)	GE 0/0/0	30.0.2.253	255.255.255.0	N/A
	GE 0/0/1	20.0.35.5	255.255.255.0	N/A
R6(AR2200)	GE 0/0/0	30.0.2.254	255.255.255.0	N/A
	GE 0/0/2	30.0.36.6	255.255.255.0	N/A
Source-1	Ethernet 0/0/1	172.16.1.1	255.255.255.0	172.16.1.10
PC-1	Ethernet 0/0/1	40.0.2.1	255.255.255.0	40.0.2.254
PC-2	Ethernet 0/0/1	30.0.2.1	255.255.255.0	30.0.2.254

实验步骤

1. 基本配置

根据图5-26和表5-4进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.1.2
```

```
PING 10.0.1.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.1.2: bytes=56 Sequence=1 ttl=255 time=80 ms
```

```
--- 10.0.1.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 80/80/80 ms
```

其余直连网段的连通性测试过程在此省略。

配置组播服务器 Source-1 的组播 IP 地址为 224.1.1.1，组播 MAC 地址为01-00-5E-01-01-01，如图5-27所示。

配置

组播组IP地址:	224 . 1 . 1 . 1
组播组MAC地址:	01-00-5E-01-01-01
源IP地址:	172 . 16 . 1 . 1
源MAC地址:	54-89-98-CF-00-78

运行

图5-27 配置Source-1

2 . 配置IGP

在每台路由器上配置OSPF协议。

```
[R1]ospf 1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 50.0.14.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 172.16.1.0 0.0.0.255
```

```
[R2]ospf 1
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.1.0 0.0.0.255
```

```
[R2-ospf-1-area-0.0.0.0]network 172.16.1.0 0.0.0.255
```

```
[R3]ospf 1
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.1.0 0.0.0.255
```

```
[R3-ospf-1-area-0.0.0.0]network 20.0.35.0 0.0.0.255
```

```
[R3-ospf-1-area-0.0.0.0]network 30.0.36.0 0.0.0.255
```

```
[R3-ospf-1-area-0.0.0.0]network 40.0.34.0 0.0.0.255
```

```
[R4]ospf 1
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 40.0.34.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 40.0.2.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 50.0.14.0 0.0.0.255
[R5]ospf 1
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]network 20.0.35.0 0.0.0.255
[R5-ospf-1-area-0.0.0.0]network 30.0.2.0 0.0.0.255
[R6]ospf 1
[R6-ospf-1]area 0
[R6-ospf-1-area-0.0.0.0]network 30.0.2.0 0.0.0.255
[R6-ospf-1-area-0.0.0.0]network 30.0.36.0 0.0.0.255
```

配置完成后，查看R1的路由表。读者可自行查看其他路由器的路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 19 Routes : 20						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
10.0.1.0/24	Direct	0	0	D	10.0.1.1	
GigabitEthernet0/0/1						
10.0.1.1/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/1						

10.0.1.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.3.3/32	OSPF	10	1	D	10.0.1.3
GigabitEthernet0/0/1					
20.0.35.0/24	OSPF	10	2	D	10.0.1.3
GigabitEthernet0/0/1					
30.0.2.0/24	OSPF	10	3	D	10.0.1.3
GigabitEthernet0/0/1					
30.0.36.0/24	OSPF	10	2	D	10.0.1.3
GigabitEthernet0/0/1					
40.0.2.0/24	OSPF	10	2	D	50.0.14.4
GigabitEthernet0/0/2					
40.0.34.0/24	OSPF	10	2	D	50.0.14.4
GigabitEthernet0/0/2					
OSPF	10	2	D	10.0.1.3	GigabitEthernet0/0/1
50.0.14.0/24	Direct	0	0	D	50.0.14.1
GigabitEthernet0/0/2					

.....

可以看到，R1已经获得了所有网段的路由信息。至此，网络已经通过OSPF实现了互通。

3．配置PIM-SM

在所有路由器上开启组播功能，并在每台路由器的每个接口下配置命令pim sm，除此之外，还需要在R5和R6的GE 0/0/0接口以及R4的GE 0/0/1接口下使能IGMP。

```
[R1]multicast routing-enable
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]pim sm
```

```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]pim sm
[R1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R1-GigabitEthernet0/0/2]pim sm
[R2]multicast routing-enable
[R2]interface GigabitEthernet 0/0/0
[R2-GigabitEthernet0/0/0]pim sm
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1]pim sm
[R3]multicast routing-enable
[R3]interface GigabitEthernet 0/0/0
[R3-GigabitEthernet0/0/0]pim sm
[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R3-GigabitEthernet0/0/1]pim sm
[R3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R3-GigabitEthernet0/0/2]pim sm
[R3-GigabitEthernet0/0/2]interface GigabitEthernet 2/0/0
[R3-GigabitEthernet2/0/0]pim sm
[R4]multicast routing-enable
[R4]interface GigabitEthernet 0/0/0
[R4-GigabitEthernet0/0/0]pim sm
[R4-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R4-GigabitEthernet0/0/1]pim sm
[R4-GigabitEthernet0/0/1]igmp enable
[R4-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R4-GigabitEthernet0/0/2]pim sm
[R5]multicast routing-enable
```

```
[R5]interface GigabitEthernet 0/0/0
[R5-GigabitEthernet0/0/0]pim sm
[R5-GigabitEthernet0/0/0]igmp enable
[R5-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R5-GigabitEthernet0/0/1]pim sm
[R6]multicast routing-enable
[R6]interface GigabitEthernet 0/0/0
[R6-GigabitEthernet0/0/0]pim sm
[R6-GigabitEthernet0/0/0]igmp enable
[R6-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/2
[R6-GigabitEthernet0/0/2]pim sm
```

选择R3为RP节点，并在每台路由器上手工配置R3为静态RP。

```
[R1]pim
[R1-pim]static-rp 10.0.3.3
[R2]pim
[R2-pim]static-rp 10.0.3.3
[R3]pim
[R3-pim]static-rp 10.0.3.3
[R4]pim
[R4-pim]static-rp 10.0.3.3
[R5]pim
[R5-pim]static-rp 10.0.3.3
[R6]pim
[R6-pim]static-rp 10.0.3.3
```

配置完成后，查看R3的PIM邻居信息。读者可自行查看其他路由器的PIM邻居信息。

```
<R3>display pim neighbor
```

VPN-Instance: public net

Total Number of Neighbors = 5

Neighbor	Interface	Uptime	Expires	Dr-Priority	BFD-Session
40.0.34.4	GE0/0/0	00:03:44	00:01:30	1	N
20.0.35.5	GE0/0/1	00:02:20	00:01:25	1	N
30.0.36.6	GE0/0/2	00:01:37	00:01:37	1	N
10.0.1.1	GE2/0/0	00:03:58	00:01:25	1	N
10.0.1.2	GE2/0/0	00:03:57	00:01:20	1	N

可以看到，R3与所有相邻的路由器都已成功建立了PIM邻居关系。

4. 用户端DR与组播源端DR

本网络中，PC-2、R5、R6处于同一网段。如果PC-2希望加入组播组，则PIM-SM需要在R5和R6之间选举出一台用户端DR来发送组播加入消息，从而避免RP接收到重复的加入消息。选举DR时，首先比较DR优先级（缺省情况下优先级的值为1），若优先级一样则比较接口IP地址的大小，IP地址较大者将成为DR。

在R5、R6上查看DR信息。

<R5>display pim interface

VPN-Instance: public net

Interface	State	NbrCnt	HelloInt	DR-Pri	DR-Address
GE0/0/0	up	1	30	1	30.0.2.254
GE0/0/1	up	1	30	1	20.0.35.5 (local)

<R6>display pim interface

VPN-Instance: public net

Interface	State	NbrCnt	HelloInt	DR-Pri	DR-Address
GE0/0/0	up	1	30	1	30.0.2.254 (local)

GE0/0/2 up 1 30 1 30.0.36.6 (local)

可以看到，R6（30.0.2.254）现在是PC-2所在网段的DR。在PC-2上使用IGMP加入组播组224.1.1.1，在R3的GE 0/0/1和GE 0/0/2接口查看报文情况。如图5-28所示，可以看到，R3仅仅从GE 0/0/2接口收到了来自DR路由器R6的组播加入消息，而GE 0/0/1接口没有收到任何组播加入消息。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	30.0.36.6	224.0.0.5	OSPF	82	Hello Packet
2	0.46800000	30.0.36.3	224.0.0.13	PIMv2	72	Hello
3	7.23800000	30.0.36.3	224.0.0.5	OSPF	82	Hello Packet
4	7.44100000	30.0.36.6	224.0.0.13	PIMv2	68	Join/Prune
5	9.26600000	30.0.36.6	224.0.0.5	OSPF	82	Hello Packet
6	18.15800000	30.0.36.3	224.0.0.5	OSPF	82	Hello Packet
7	18.51700000	30.0.36.6	224.0.0.5	OSPF	82	Hello Packet

图5-28 R3的GE 0/0/2接口的报文情况

在R3上查看PIM路由表。

```
<R3>display pim routing-table
```

VPN-Instance: public net

Total 1 (*, G) entry; 0 (S, G) entry

(*, 224.1.1.1)

RP: 10.0.3.3 (local)

Protocol: pim-sm, Flag: WC

UpTime: 00:01:31

Upstream interface: Register

Upstream neighbor: NULL

RPF prime neighbor: NULL

Downstream interface(s) information:

Total number of downstreams: 1

1: GigabitEthernet0/0/2

Protocol: pim-sm, UpTime: 00:01:31, Expires: 00:02:59

可以看到，R3 在收到加入消息后，其组播路由表中生成了（*，G）的组播路由条目，下游接口为GE 0/0/2，形成了从RP（R3）到R6的一棵RPT。

另一方面，组播源Source-1、R1、R2处于同一网段，PIM-SM需要在R1和R2之间选举出组播源端DR来向RP发送注册消息。在R1、R2上查看DR信息。

```
<R1>display pim interface
```

```
VPN-Instance: public net
```

Interface	State	NbrCnt	HelloInt	DR-Pri	DR-Address
GE0/0/0	up	1	30	1	172.16.1.20
GE0/0/1	up	2	30	1	10.0.1.3
GE0/0/2	up	1	30	1	50.0.14.4

```
<R2>display pim interface
```

```
VPN-Instance: public net
```

Interface	State	NbrCnt	HelloInt	DR-Pri	DR-Address
GE0/0/0	up	1	30	1	172.16.1.20 (local)
GE0/0/1	up	2	30	1	10.0.1.3

可以看到，IP地址较大的R2（172.16.1.20）目前是组播源端DR。接下来，我们可以通过修改优先级的方法来强制让R1成为组播源端DR。

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]pim hello-option dr-priority 2
```

配置完成后，在R1上查看DR信息。

```
<R1>display pim interface
```

```
VPN-Instance: public net
```

Interface	State	NbrCnt	HelloInt	DR-Pri	DR-Address
GE0/0/0	up	1	30	2	172.16.1.10 (local)


```
<R4>display pim routing-table
VPN-Instance: public net
Total 1 (*, G) entry; 0 (S, G) entry
(*, 224.1.1.1)
  RP: 10.0.3.3
  Protocol: pim-sm, Flag: WC
  UpTime: 00:00:03
  Upstream interface: GigabitEthernet0/0/0
  Upstream neighbor: 40.0.34.3
  RPF prime neighbor: 40.0.34.3
  Downstream interface(s) information:
    Total number of downstreams: 1
    1: GigabitEthernet0/0/1
  Protocol: igmp, UpTime: 00:00:03, Expires:-
```

可以看到，R4上生成了（*，224.1.1.1）的组播路由，上游接口是连接到RP路由器的GE 0/0/0 接口。

在组播源发送组播地址为224.1.1.1的组播报文，当PC-1接收到组播报文后，在R4上查看组播路由表。

```
<R4>display pim routing-table
VPN-Instance: public net
Total 1 (*, G) entry; 1 (S, G) entry
(*, 224.1.1.1)
  RP: 10.0.3.3
  Protocol: pim-sm, Flag: WC
  UpTime: 00:01:35
  Upstream interface: GigabitEthernet0/0/0
  Upstream neighbor: 40.0.34.3
```

```
RPF prime neighbor: 40.0.34.3
Downstream interface(s) information:
Total number of downstreams: 1
1: GigabitEthernet0/0/1
Protocol: igmp, UpTime: 00:01:35, Expires:-
(172.16.1.1, 224.1.1.1)
RP: 10.0.3.3
Protocol: pim-sm, Flag: RPT SPTACT
UpTime: 00:00:55
Upstream interface: GigabitEthernet0/0/2
Upstream neighbor : 50.0.14.1
RPF prime neighbor : 50.0.14.1
Downstream interface(s) information:
Total number of downstreams: 1
1: GigabitEthernet0/0/1
Protocol: pim-sm, UpTime: 00:00:55, Expires:-
```

可以看到，R4在接收到组播数据后生成了（172.16.1.1，224.1.1.1）的组播路由，且上游接口切换到了GE 0/0/2。切换过程中，R4会向R1发送组播加入消息，要求从R1接收组播数据，同时R4也会向R3发送剪枝消息，使R3停止向自己转发组播数据。

缺省情况下，PIM-SM路由器会在收到第一个组播数据包后立即进行从RPT到SPT的切换。如果不希望发生切换，则可使用配置命令来禁止切换。另外，也可以设定切换阈值，实现有条件的切换。在用户端DR上配置了切换阈值后，只有当组播报文的速率超过阈值时，用户端DR才会发起切换。下面给出禁止切换的示例。

```
[R4]pim
```

```
[R4-pim]spt-switch-threshold infinity
```

配置完成后，在PC-1上使用IGMP加入组播组224.1.1.1，在Source-1上发送组播地址为224.1.1.1的组播报文，然后查看R4的组播路由表。

```
<R4>display pim routing-table
```

```
VPN-Instance: public net
```

```
Total 1 (*, G) entry; 1 (S, G) entry
```

```
(*, 224.1.1.1)
```

```
RP: 10.0.3.3
```

```
Protocol: pim-sm, Flag: WC
```

```
UpTime: 00:00:28
```

```
Upstream interface: GigabitEthernet0/0/0
```

```
Upstream neighbor: 40.0.34.3
```

```
RPF prime neighbor: 40.0.34.3
```

```
Downstream interface(s) information:
```

```
Total number of downstreams: 1
```

```
1: GigabitEthernet0/0/1
```

```
Protocol: igmp, UpTime: 00:00:28, Expires:-
```

```
(172.16.1.1, 224.1.1.1)
```

```
RP: 10.0.3.3
```

```
Protocol: pim-sm, Flag: ACT
```

```
UpTime: 00:00:18
```

```
Upstream interface: GigabitEthernet0/0/0
```

```
Upstream neighbor : 40.0.34.3
```

```
RPF prime neighbor : 40.0.34.3
```

```
Downstream interface(s) information:
```

```
Total number of downstreams: 1
```

```
1: GigabitEthernet0/0/1
```

```
Protocol: pim-sm, UpTime: 00:00:18, Expires:-
```

可以看到，当配置禁止切换后，R4生成的（172.16.1.1，224.1.1.1）的组播路由的上游接口没有发生切换。

6．配置PIM-Silent接口

通常情况下，路由器直连用户主机的接口上是需要配置PIM的，但是这样的配置同时也存在着安全隐患：当恶意主机发送大量PIM-Hello报文时，有可能导致路由器瘫痪。为了避免上述情况的发生，可以在路由器直连用户主机的接口上配置 PIM Silent，禁止该接口接收和转发任何PIM协议报文，同时，此接口上的组播转发功能及IGMP功能都不受影响。

配置R4的GE 0/0/1接口为PIM-Silent接口。

```
[R4]interface GigabitEthernet 0/0/1
```

```
[R4-GigabitEthernet0/0/1]pim silent
```

配置完成后，查看R4的PIM接口详细信息，并在R4的GE 0/0/1接口查看数据包的发送，如图5-31所示。

```
[R4]display pim interface GigabitEthernet 0/0/1 verbose
```

```
VPN-Instance: public net
```

```
Interface: GigabitEthernet0/0/1, 40.0.2.254
```

```
PIM version: 2
```

```
.....
```

```
PIM hello override interval (configured): 2500 ms
```

```
PIM Silent: enabled
```

```
PIM neighbor tracking (negotiated): disabled
```

```
.....
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
2	9.36000000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
3	18.70500000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
4	20.03100000	40.0.2.254	224.0.0.1	IGMPv2	60	Membership Query, general
5	28.06500000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
6	37.40900000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
7	46.76900000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
8	56.12900000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
9	65.47400000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
10	74.83400000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
11	79.99700000	40.0.2.254	224.0.0.1	IGMPv2	60	Membership Query, general
12	84.19400000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
13	93.53800000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet
14	102.89800000	40.0.2.254	224.0.0.5	OSPF	78	Hello Packet

图5-31 R4的GE 0/0/1接口的报文情况

可以看到，R4 的GE 0/0/1 接口的Silent 功能已经使能，此接口不再发送PIM-Hello报文。

思考

PIM-DM 协议和PIM-SM 协议都属于PIM（Protocol-Independent Multicast）协议。那么Protocol-Independent在这里究竟是什么意思呢？

5.5 PIM-SM 的RP

原理概述

一个PIM-SM网络中可以存在一个或多个RP。一个RP可以对应若干个组播组，负责这些组播组的注册消息的处理、加入消息的处理以及组播数据的转发，但是同一个组播组只能对应一个RP。RP是PIM-SM网络的核心，网络中的路由器必须知道RP的地址。

RP有两种形式：静态RP和动态RP。静态RP是由人工选定的，PIM网络中的所有PIM路由器上都需要逐一进行配置；通过配置，每台路由器便知道了静态RP的地址。动态RP的确定过程相对比较复杂一些：在PIM网络中人工选定并配置若干台PIM路由器，使得它们成为C-RP(Candidate-RP)，RP将从C-RP中选举产生。如果C-RP只有一个，则

RP就是这个C-RP；如果有多个C-RP，则优先级最高者（优先级数值越小优先级越高，缺省值是0）将会被选举为RP；如果通过优先级比较无法选举出RP，则依靠Hash算法算出的数值来决定RP，数值最大者将成为RP（Hash算法参数：组地址、掩码长度、C-RP地址）；如果通过Hash数值也无法确定出RP，则拥有最高IP地址的C-RP将成为RP。选定和配置C-RP时，还必须同时选定和配置C-BSR（Candidate-Bootstrap Router），并由C-BSR选举产生出一个BSR。如果有多个C-BSR，则拥有最高优先级的C-BSR将成为BSR；如果根据优先级无法确定出BSR，则拥有最高IP地址的C-BSR将成为BSR。BSR是PIM-SM网络的管理核心，它负责收集网络中C-RP发出的Advertisement宣告信息，并计算出与每个组播组对应的RP，然后将RP的信息发布到整个PIM-SM网络中。

在传统的PIM-SM网络中，每个组播组只能映射到一个RP，当网络负载较大以及流量分布不合理时，可能导致RP拥塞或者网络资源严重浪费的情况。解决上述问题的一个方案便是配置Anycast RP：在同一个PIM-SM网络中设置多个具有相同环回地址的RP，组播源和组播用户分别选择距离自己最近的RP进行RPT的创建，从而实现分担和优化组播流量的目的。

实验目的

- 理解RP的作用
- 掌握静态和动态RP的配置方法
- 理解Anycast RP的应用场景
- 掌握Anycast RP的配置方法

实验内容

实验拓扑如图5-32所示，实验编址如表5-5所示。本实验网络包含了6台路由器、两台组播服务器和两台终端电脑。全网运行OSPF，并

通过 PIM-SM 来实现组播服务。网络管理员需要配置静态RP、动态RP、Anycas RP，以便加深对PIM-SM 网络行为的理解和认识。

实验拓扑

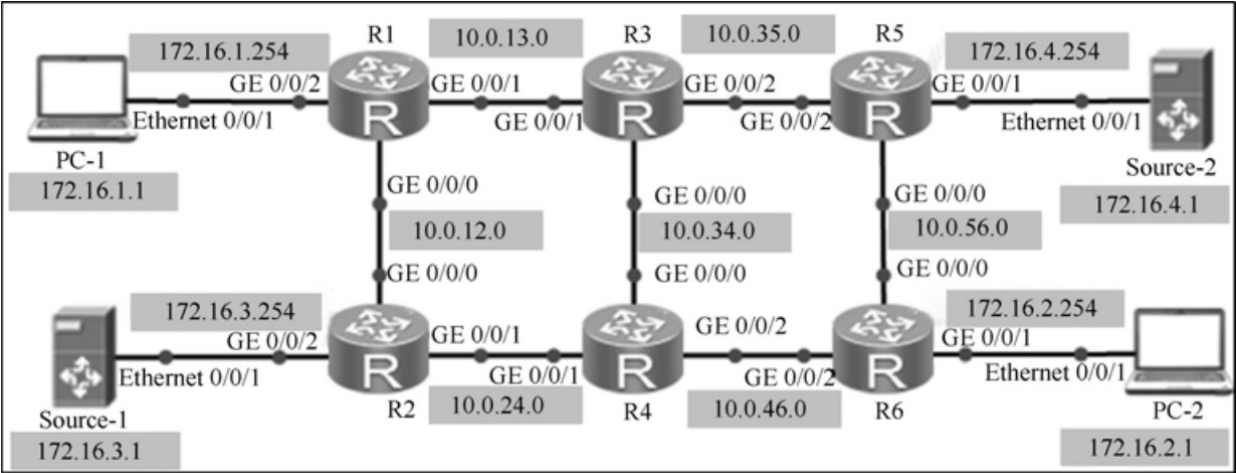


图5-32 PIM-SM的RP

实验编址表

表5-5 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2200)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	GE 0/0/2	172.16.1.254	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	10.0.11.11	255.255.255.255	N/A
R2(AR2200)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.24.2	255.255.255.0	N/A
	GE 0/0/2	172.16.3.254	255.255.255.0	N/A
R3(AR2200)	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.35.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2200)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	GE 0/0/2	10.0.46.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A

(续表)

设备	接口	IP 地址	子网掩码	默认网关
R5(AR2200)	GE 0/0/0	10.0.56.5	255.255.255.0	N/A
	GE 0/0/1	172.16.4.254	255.255.255.0	N/A
	GE 0/0/2	10.0.35.5	255.255.255.0	N/A
R6(AR2200)	GE 0/0/0	10.0.56.6	255.255.255.0	N/A
	GE 0/0/1	172.16.2.254	255.255.255.0	N/A
	GE 0/0/2	10.0.46.6	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
	Loopback 1	10.0.6.6	255.255.255.255	N/A
Source-1	Ethernet 0/0/1	172.16.3.1	255.255.255.0	172.16.3.254
Source-2	Ethernet 0/0/1	172.16.4.1	255.255.255.0	172.16.4.254
PC-1	Ethernet 0/0/1	172.16.1.1	255.255.255.0	172.16.1.254
PC-2	Ethernet 0/0/1	172.16.2.1	255.255.255.0	172.16.2.254

实验步骤

1. 基本配置

根据图5-32和表5-5进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 10.0.13.3
```

```
PING 10.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=230 ms
```

```
--- 10.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 230/230/230 ms
```

其余直连网段的连通性测试过程在此省略。

配置组播服务器 Source-1 的组播 IP 地址为 224.1.1.1，组播 MAC 地址为01-00-5E-01-01-01，如图5-33所示。

配置

组播组IP地址:	224 . 1 . 1 . 1
组播组MAC地址:	01-00-5E-01-01-01
源IP地址:	172 . 16 . 3 . 1
源MAC地址:	54-89-98-CF-77-5F
运行	

图5-33 配置Source-1

配置组播服务器 Source-2 的组播 IP 地址为 225.1.1.1，组播 MAC 地址为01-00-5E-02-02-02，如图5-34所示。

配置

组播组IP地址:	225 . 1 . 1 . 1
组播组MAC地址:	01-00-5E-02-02-02
源IP地址:	172 . 16 . 4 . 1
源MAC地址:	54-89-98-CF-A7-51
运行	

图5-34 配置Source-2

2 . 配置IGP

在每台路由器上配置OSPF协议。

```
[R1]ospf 1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.11.11 0.0.0.0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
[R1-ospf-1-area-0.0.0.0]network 172.16.1.0 0.0.0.255
[R2]ospf 1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 172.16.3.0 0.0.0.255
[R3]ospf 1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R3-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.35.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R4]ospf 1
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
[R4-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.46.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R5]ospf 1
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]network 10.0.35.0 0.0.0.255
[R5-ospf-1-area-0.0.0.0]network 10.0.56.0 0.0.0.255
[R5-ospf-1-area-0.0.0.0]network 172.16.4.0 0.0.0.255
[R6]ospf 1
[R6-ospf-1]area 0
[R6-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```


10.0.13.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.13.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.35.0/24	OSPF	10	2	D	10.0.13.3
GigabitEthernet0/0/1					
10.0.24.0/24	OSPF	10	2	D	10.0.12.2
GigabitEthernet0/0/0					
10.0.46.0/24	OSPF	10	3	D	10.0.12.2
GigabitEthernet0/0/0					
OSPF	10	3	D	10.0.13.3	GigabitEthernet0/0/1
10.0.12.0/24	Direct	0	0	D	10.0.12.1
GigabitEthernet0/0/0					
10.0.12.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.12.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.34.0/24	OSPF	10	2	D	10.0.13.3
GigabitEthernet0/0/1					
10.0.56.0/24	OSPF	10	3	D	10.0.13.3
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

```

172.16.1.0/24      Direct      0    0      D      172.16.1.2
GigabitEthernet0/0/2
172.16.1.254/32    Direct      0    0      D      127.0.0.1
GigabitEthernet0/0/2
172.16.1.255/32    Direct      0    0      D      127.0.0.1
GigabitEthernet0/0/2
172.16.2.0/24      OSPF        10    4      D      10.0.12.2
GigabitEthernet0/0/0
    OSPF    10    4      D      10.0.13.3  GigabitEthernet0/0/1
172.16.3.0/24      OSPF        10    2      D      10.0.12.2
GigabitEthernet0/0/0
172.16.4.0/24      OSPF        10    3      D      10.0.13.3
GigabitEthernet0/0/1
255.255.255.255/32 Direct      0    0      D      127.0.0.1
InLoopBack0

```

可以看到，R1已经获得了所有网段的路由信息。至此，网络已经通过OSPF实现了互通。

3．配置PIM-SM和静态RP

在所有路由器上开启组播功能，并在每台路由器的每个接口下配置命令pim sm，除此之外，还需要在R1的GE 0/0/2和R6的GE 0/0/1接口下使能IGMP。

```

[R1]multicast routing-enable
[R1]interface GigabitEthernet 0/0/0
[R1-GigabitEthernet0/0/0]pim sm
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]pim sm
[R1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2

```



```
[R1-GigabitEthernet0/0/2]pim sm
[R1-GigabitEthernet0/0/2]igmp enable
[R2]multicast routing-enable
[R2]interface GigabitEthernet 0/0/0
[R2-GigabitEthernet0/0/0]pim sm
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1]pim sm
[R2-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R2-GigabitEthernet0/0/2]pim sm
[R3]multicast routing-enable
[R3]interface GigabitEthernet 0/0/0
[R3-GigabitEthernet0/0/0]pim sm
[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R3-GigabitEthernet0/0/1]pim sm
[R3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R3-GigabitEthernet0/0/2]pim sm
[R4]multicast routing-enable
[R4]interface GigabitEthernet 0/0/0
[R4-GigabitEthernet0/0/0]pim sm
[R4-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R4-GigabitEthernet0/0/1]pim sm
[R4-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R4-GigabitEthernet0/0/2]pim sm
[R5]multicast routing-enable
[R5]interface GigabitEthernet 0/0/0
[R5-GigabitEthernet0/0/0]pim sm
[R5-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```

[R5-GigabitEthernet0/0/1]pim sm
[R5-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R5-GigabitEthernet0/0/2]pim sm
[R6]multicast routing-enable
[R6]interface GigabitEthernet 0/0/0
[R6-GigabitEthernet0/0/0]pim sm
[R6-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R6-GigabitEthernet0/0/1]pim sm
[R6-GigabitEthernet0/0/1]igmp enable
[R6-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R6-GigabitEthernet0/0/2]pim sm

```

配置完成后，查看 R1 的 PIM 邻居信息。读者可自行查看其他路由器的 PIM 邻居信息。

```

<R1>display pim neighbor
  VPN-Instance: public net
  Total Number of Neighbors = 2

```

Neighbor	Interface	Uptime	Expires	Dr-Priority	BFD-Session
10.0.12.2	GE0/0/0	00:47:25	00:01:23	1	N
10.0.13.3	GE0/0/1	00:46:04	00:01:40	1	N

可以看到，R1与R2和R3都已成功建立了PIM邻居关系。接下来，在每台路由器上手工配置R1（10.0.11.11）为静态RP。

```

[R1]interface loopback 1
[R1-LoopBack1]pim sm
[R1-LoopBack1]pim
[R1-pim]static-rp 10.0.11.11
[R2]pim

```

```
[R2-pim]static-rp 10.0.11.11
```

```
[R3]pim
```

```
[R3-pim]static-rp 10.0.11.11
```

```
[R4]pim
```

```
[R4-pim]static-rp 10.0.11.11
```

```
[R5]pim
```

```
[R5-pim]static-rp 10.0.11.11
```

```
[R6]pim
```

```
[R6-pim]static-rp 10.0.11.11
```

配置完成后，在R1和R2上查看RP信息。

```
<R1>display pim rp-info
```

```
VPN-Instance: public net
```

```
PIM SM static RP Number:1
```

```
Static RP: 10.0.11.11 (local)
```

```
<R2>display pim rp-info
```

```
VPN-Instance: public net
```

```
PIM SM static RP Number:1
```

```
Static RP: 10.0.11.11
```

可以看到，R1（10.0.11.11）已经成为了静态RP。

4．配置动态RP

选定并配置R1和R6为C-RP，R1使用Loopback 0为RP接口，R6使用Loopback 1为RP接口。同时，选定并配置R3和R4为C-BSR，R3和R4都使用Loopback 0作为C-BSR接口。

```
[R1]interface Loopback 0
```

```
[R1-LoopBack0]pim sm
```

```
[R1-LoopBack0]pim
```

```
[R1-pim]c-rp Loopback 0
```

```
[R6]interface Loopback 1
[R6-LoopBack1]pim sm
[R6-LoopBack1]pim
[R6-pim]c-rp LoopBack 1
[R3]interface Loopback 0
[R3-LoopBack0]pim sm
[R3-LoopBack0]pim
[R3-pim]c-bsr LoopBack 0
[R4]interface Loopback 0
[R4-LoopBack0]pim sm
[R4-LoopBack0]pim
[R4-pim]c-bsr LoopBack 0
```

配置完成后，在R1上查看RP信息和BSR信息。

```
<R1>display pim rp-info
  VPN-Instance: public net
  PIM-SM BSR RP Number:2
  Group/MaskLen: 224.0.0.0/4
    RP: 10.0.1.1(local)
    Priority: 0
    Uptime: 01:54:42
    Expires: 00:02:26
  Group/MaskLen: 224.0.0.0/4
    RP: 10.0.6.6
    Priority: 0
    Uptime: 01:54:42
    Expires: 00:02:26
  PIM SM static RP Number:1
```

```
Static RP: 10.0.11.11 (local)
<R1>display pim rp-info 224.1.1.1
VPN-Instance: public net
BSR RP Address is: 10.0.6.6
Priority: 0
Uptime: 02:12:13
Expires: 00:01:59
Static RPAddress is: 10.0.11.11
RP mapping for this group is: 10.0.6.6
<R1>display pim bsr-info
VPN-Instance: public net
Elected AdminScoped BSR Count: 0
Elected BSR Address: 10.0.4.4
Priority: 0
Hash mask length: 30
State: Accept Preferred
Scope: Not scoped
Uptime: 01:55:18
Expires: 00:01:29
C-RP Count: 2
```

可以看到，R1和R6的RP优先级的值在缺省情况下都为0。当静态RP和动态RP同时存在时，动态RP优先。在优先级和Hash值的掩码长度相同的情况下，IP地址较大的C-RP（R6）被选为了RP。另外，C-BSR优先级相同的情况下，IP地址较大的R4成为了BSR。

通过修改优先级，可以控制RP的选举。

```
[R6]pim
```

```
[R6-pim]c-rp priority 10
```

配置完成后，重新在R1上查看RP的信息。

```
<R1>display pim rp-info 224.1.1.1
```

```
VPN-Instance: public net
```

```
BSR RP Address is: 10.0.1.1
```

```
Priority: 0
```

```
Uptime: 00:27:59
```

```
Expires: 00:01:32
```

```
Static RP Address is: 10.0.11.11
```

```
RP mapping for this group is: 10.0.1.1 (local host)
```

可以看到，当R6的RP优先级的值调整为10（数值越小优先级越高）时，优先级较高的R1成为了RP。

5．配置Anycast RP

本网络中，若R1为RP，那么当Source-2发送组播数据，PC-2接收时，组播源端DR（R5）产生的注册消息和用户端DR（R6）产生的加入消息都要发送给远处的R1，另外，组播数据也要经历R5-R1-R6的绕行路径，浪费了链路带宽和路由器的CPU资源。在这种情况下，配置Anycast RP便是一个不错的解决方案。

在R1和R6上配置Anycast RP。

```
[R1]pim
```

```
[R1-pim]anycast-rp 10.0.1.1
```

```
[R1-pim-anycast-rp-10.0.1.1]local-address 10.0.11.11
```

```
[R1-pim-anycast-rp-10.0.1.1]peer 10.0.6.6
```

```
[R6]interface loopback 0
```

```
[R6-LoopBack0]pim sm
```

```
[R6-LoopBack0]pim
```

```
[R6-pim]undo c-rp loopback 1
```

```
[R6-pim]undo c-rp priority
```

```
[R6-pim]c-rp loopback 0
[R6-pim]anycast-rp 10.0.1.1
[R6-pim-anycast-rp-10.0.1.1]local-address 10.0.6.6
[R6-pim-anycast-rp-10.0.1.1]peer 10.0.11.11
```

配置完成后，在R1和R6上查看RP信息。

```
<R1>display pim rp-info 224.1.1.1
```

VPN-Instance: public net

BSR RP Address is: 10.0.1.1

Priority: 0

Uptime: 00:02:20

Expires: 00:01:59

Static RP Address is: 10.0.11.11

RP mapping for this group is: 10.0.1.1 (local host)

```
<R6>display pim rp-info 225.1.1.1
```

VPN-Instance: public net

BSR RP Address is: 10.0.1.1

Priority: 0

Uptime: 00:02:56

Expires: 00:02:23

Static RP Address is: 10.0.11.11

RP mapping for this group is: 10.0.1.1 (local host)

可以看到，RP为10.0.1.1，R1和R6都可充当RP。接下来将要验证，组播注册消息和加入消息会由就近的RP来处理。

在R1和R6上打开Debugging功能。

```
<R1>debugging pim register
```

```
<R1>debugging pim join-prune
```

```
<R1>terminal monitor
```

<R1>terminal debugging

<R6>debugging pim register

<R6>debugging pim join-prune

<R6>terminal monitor

<R6>terminal debugging

配置完成后，让Source-1发送224.1.1.1组播流量，PC-1接收，查看注册消息和加入消息的收发情况。

<R1>

Sep 25 2013 18:36:22.634.1-05:13 R1 PIM/7/REG:(public net): PIM ver 2 REG receiving 172.16.3.254 -> 10.0.1.1 on GigabitEthernet0/0/0 (S01758)

<R1>

Sep 25 2013 18:36:22.634.2-05:13 R1 PIM/7/REG:(public net): Border bit: false, Null bit: false (S01769)

<R1>

Sep 25 2013 18:36:22.634.3-05:13 R1 PIM/7/REG:(public net): Encapsulated ip src: 172.16.3.1, dst: 224.1.1.1, len: 20 (S01787)

<R1>

Sep 25 2013 18:36:22.634.4-05:13 R1 PIM/7/REG:(public net): Receiving register message with source 172.16.3.1, group address 224.1.1.1 (S213262)

<R1>

Sep 25 2013 18:36:22.634.5-05:13 R1 PIM/7/REG:(public net): Receiving register from 172.16.3.254, Store the Source DR address (S213467)

<R1>

Sep 25 2013 18:36:22.634.6-05:13 R1 PIM/7/REG:(public net): PIM
ver 2 RSP sending 10.0.1.1 -> 172.16.3.254 on GigabitEthernet0/0/0
(S01624)

.....

<R1>

Sep 25 2013 18:36:42.134.3-05:13 R1 PIM/7/JP:(public net): Group:
224.1.1.1/32 --- 1 join 0 prune (P013107)

<R1>

Sep 25 2013 18:36:42.134.4-05:13 R1 PIM/7/JP:(public net): Join:
172.16.3.1/32 S (P013117)

.....

<R6>

Sep 25 2013 18:36:25.175.1-05:13 R6 PIM/7/REG:(public net): PIM
ver 2 REG receiving 10.0.11.11 -> 10.0.6.6 on GigabitEthernet0/0/2
(S01758)

<R6>

Sep 25 2013 18:36:25.175.2-05:13 R6 PIM/7/REG:(public net): Border
bit: false, Null bit: false (S01769)

<R6>

Sep 25 2013 18:36:25.175.3-05:13 R6 PIM/7/REG:(public net):
Encapsulated ip src: 172.16.3.1, dst: 224.1.1.1, len: 20 (S01787)

.....

从上面的显示信息中可以看到源端 DR (172.16.3.254) 发送给 R1
的注册消息，以及用户端DR (R1) 发送给R1的组播加入消息。另外还
可以看到，R1将注册报文重新封装后发送给了Anycast RP的对等体
R6，以便共享组播源信息。

当Source-2发送225.1.1.1组播数据，PC-2接收时，类似于上面的实验观察将表明，组播注册消息和加入消息都会就近发送给R6处理，读者可自行进行实验验证。

思考

PIM-SM中，组播源是如何进行注册的？

5.6 RPF 校验

原理概述

所谓 RPF（Reverse Path Forwarding）校验，就是指在基于 Source-Based Tree的组播网络（例如PIM-DM网络）中路由器通过查找去往组播源的最优单播路由来判断所收到的组播数据是否来源于“正确的”上游接口。某一路由器去往某一组播源的最优单播路由所对应的出接口称为该路由器上关于该组播源的RPF接口。一台路由器从某一接口收到一个组播数据后，如果发现该接口不是相应组播源的RPF接口，就意味着 RPF 校验失败，所收到的组播数据将被丢弃；如果发现该接口正是相应组播源的RPF接口，就表明RPF校验通过，所收到的组播数据将被进行后续处理。

正是因为有了 RPF 校验机制，基于 Source-Based Tree 的组播网络中所生成的组播树才能是一棵 SPT（Shortest Path Tree），同时，RPF 校验机制也防止了组播数据在转发过程中出现重复报文及流量环路的情况。另外，RPF 校验过程中所使用的单播路由可以来源于任何一种单播路由协议，并不依赖于某一特定的单播路由协议。

当然，为了某些特殊的需要，RPF接口也是可以被人修改的。如果路由器上配置了组播静态路由，则RPF校验将首先依据组播静态路由而非单播路由。通过配置组播静态路由，可以在当前路由器上为特定

的组播源人为指定一个RPF。组播静态路由只在所配置的路由器上才有效，不会以任何方式传递给其他路由器。

实验目的

- 理解RPF 的原理和作用
- 掌握组播静态路由的配置方法

实验内容

实验拓扑如图5-35所示，实验编址如表5-6所示。本实验网络包含了4台路由器、一台交换机、两台组播服务器和两台终端电脑，全网运行OSPF，并且部署了PIM-DM。组播服务器 Source-1 存储的是学习视频，Source-2 存储的是电影视频，PC-1 需要从Source-1接收学习视频，PC-2需要从Source-2接收电影视频。网络管理员需要在R4上配置组播静态路由，以实现组播流量分布的优化。

实验拓扑

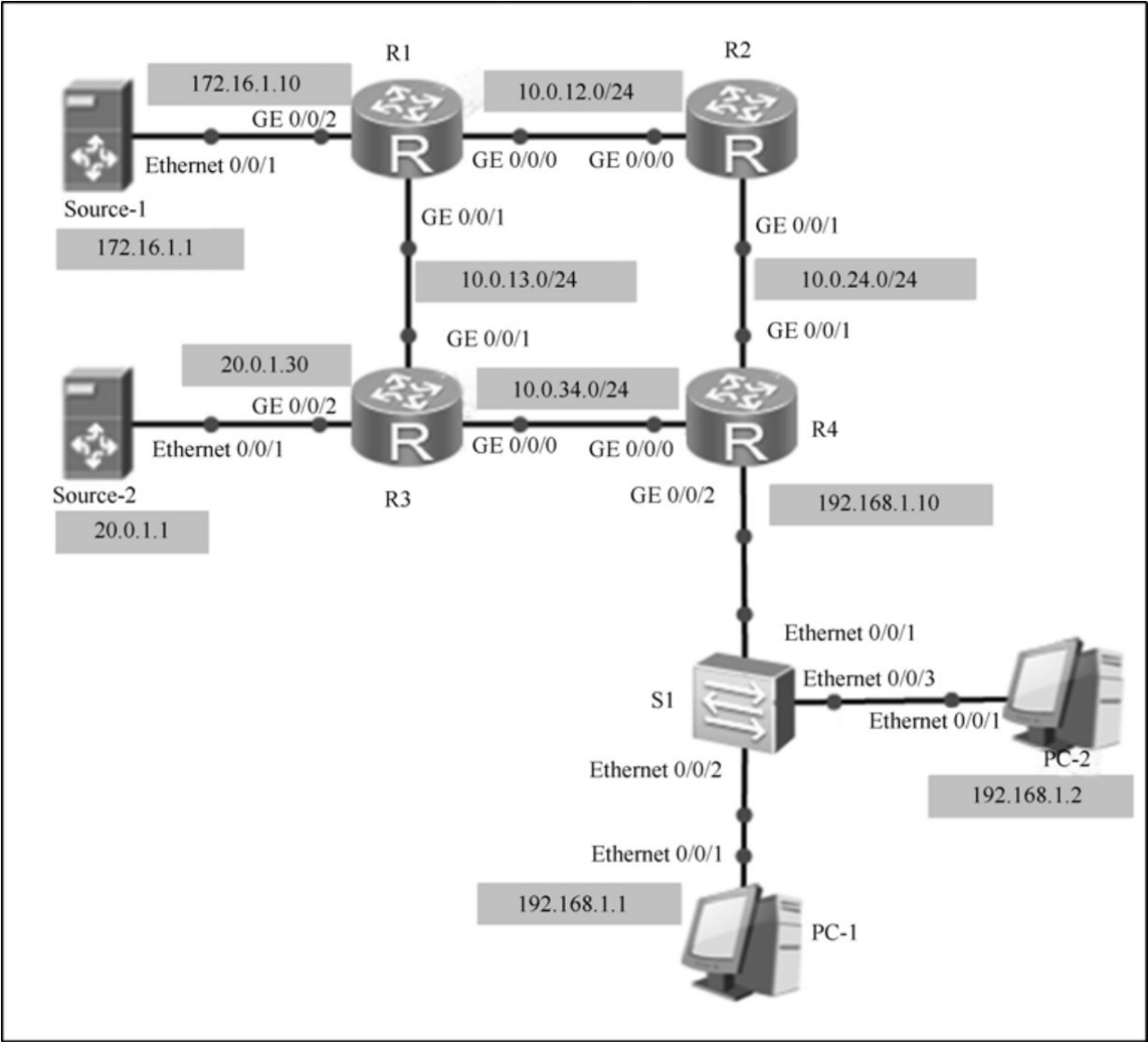


图5-35 RPF校验

实验编址表

表5-6 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2200)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	GE 0/0/2	172.16.1.10	255.255.255.0	N/A
R2(AR2200)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.24.2	255.255.255.0	N/A

(续表)

设备	接口	IP 地址	子网掩码	默认网关
R3(AR2200)	GE 0/0/0	10.0.34.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	20.0.1.30	255.255.255.0	N/A
R4(AR2200)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	GE 0/0/2	192.168.1.10	255.255.255.0	N/A
Source-1	Ethernet 0/0/1	172.16.1.1	255.255.255.0	172.16.1.10
Source-2	Ethernet 0/0/1	20.0.1.1	255.255.255.0	20.0.1.30
PC-1	Ethernet 0/0/1	192.168.1.1	255.255.255.0	192.168.1.10
PC-2	Ethernet 0/0/1	192.168.1.2	255.255.255.0	192.168.1.10

实验步骤

1．基本配置

根据图5-35和表5-6进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=50 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 50/50/50 ms
```

其余直连网段的连通性测试过程在此省略。

配置组播服务器 Source-1 的组播 IP 地址为 224.1.1.1，组播 MAC 地址为01-00-5E-01-01-01，如图5-36所示。

配置

组播组IP地址:	224 . 1 . 1 . 1
组播组MAC地址:	01-00-5E-01-01-01
源IP地址:	172 . 16 . 1 . 1
源MAC地址:	54-89-98-CF-B7-61

运行

图5-36 配置Source-1

配置组播服务器 Source-2 的组播 IP 地址为 225.1.1.1，组播 MAC 地址为01-00-5E-02-02-02，如图5-37所示。

配置

组播组IP地址:	225 . 1 . 1 . 1
组播组MAC地址:	01-00-5E-02-02-02
源IP地址:	20 . 0 . 1 . 1
源MAC地址:	54-89-98-CF-77-69

运行

图5-37 配置组播服务器Source-2

2 . 配置IGP

在每台路由器上配置OSPF协议。

```
[R1]ospf 1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 172.16.1.0 0.0.0.255
```

```
[R2]ospf 1
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R3]ospf 1
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 20.0.1.0 0.0.0.255
[R4]ospf 1
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.24.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 192.168.1.0 0.0.0.255
```

配置完成后，查看R1的路由表。读者可自行查看其他路由器的路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 17 Routes : 18

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.12.0/24	Direct	0	0	D	10.0.12.1
GigabitEthernet0/0/0					

10.0.12.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.12.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.13.0/24	Direct	0	0	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.13.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.13.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.24.0/24	OSPF	10	2	D	10.0.12.2
GigabitEthernet0/0/0					
10.0.34.0/24	OSPF	10	2	D	10.0.13.3
GigabitEthernet0/0/1					
20.0.1.0/24	OSPF	10	2	D	10.0.13.3
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
172.16.1.0/24	Direct	0	0	D	172.16.1.10
GigabitEthernet0/0/2					
172.16.1.10/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					

172.16.1.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
192.168.1.0/24	OSPF	10	3	D	10.0.12.2
GigabitEthernet0/0/0					
OSPF	10	3	D	10.0.13.3	GigabitEthernet0/0/1
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R1已经获得了所有网段的路由信息。至此，网络已经通过OSPF实现了互通。

3．配置PIM-DM

在所有路由器上开启组播功能，并在每台路由器的每个接口下配置命令 `pim dm`，除此之外，还需要在R4的GE 0/0/2接口下使能IGMP。

```
[R1]multicast routing-enable
[R1]interface GigabitEthernet 0/0/0
[R1-GigabitEthernet0/0/0]pim dm
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R1-GigabitEthernet0/0/1]pim dm
[R1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R1-GigabitEthernet0/0/2]pim dm
[R2]multicast routing-enable
[R2]interface GigabitEthernet 0/0/0
[R2-GigabitEthernet0/0/0]pim dm
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1]pim dm
[R3]multicast routing-enable
[R3]interface GigabitEthernet 0/0/0
```

```

[R3-GigabitEthernet0/0/0]pim dm
[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R3-GigabitEthernet0/0/1]pim dm
[R3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R3-GigabitEthernet0/0/2]pim dm
[R4]multicast routing-enable
[R4]interface GigabitEthernet 0/0/0
[R4-GigabitEthernet0/0/0]pim dm
[R4-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R4-GigabitEthernet0/0/1]pim dm
[R4-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[R4-GigabitEthernet0/0/2]pim dm
[R4-GigabitEthernet0/0/2]igmp enable

```

配置完成后，查看 R1 的 PIM 邻居信息。读者可自行查看其他路由器的 PIM 邻居信息。

```

<R1>display pim neighbor
  VPN-Instance: public net
  Total Number of Neighbors = 2
  Neighbor    Interface    Uptime    Expires    Dr-Priority    BFD-
Session
  10.0.12.2    GE0/0/0    00:36:04    00:01:30    1              N
  10.0.13.3    GE0/0/1    00:35:34    00:01:41    1              N

```

可以看到，R1与R2和R3都已成功建立了PIM邻居关系。

4．RPF校验过程

由于网络部署了PIM-DM，所以R1在接收到Source-1发送的组播数据后，会通过它的每个PIM-DM接口转发组播数据。类似地，R2和R3

也会将接收到的组播数据通过每个 PIM-DM 接口继续转发，这样一来，R4 就会接收到两份来自不同接口的相同的组播数据。

在R4上打开Debug功能。

```
<R4>debugging pim join-prune
```

```
<R4>terminal monitor
```

```
<R4>terminal debugging
```

配置完成后，先让PC-1加入组播组224.1.1.1，然后在Source-1上发送组播地址为224.1.1.1的组播视频流。

```
<R4>
```

```
Sep 25 2013 20:39:53.47.1-05:13 R4 PIM/7/JP:(public net): PIM ver 2  
JP sending 10.0.34.4 -> 224.0.0.13 on GigabitEthernet0/0/0 (P013156)
```

```
<R4>
```

```
Sep 25 2013 20:39:53.47.2-05:13 R4 PIM/7/JP:(public net): Upstream  
10.0.34.3, Groups 1, Holdtime 210 (P013160)
```

```
<R4>
```

```
Sep 25 2013 20:39:53.47.3-05:13 R4 PIM/7/JP:(public net): Group:  
224.1.1.1/32 --- 0 join 1 prune (P013169)
```

```
<R4>
```

```
Sep 25 2013 20:39:53.47.4-05:13 R4 PIM/7/JP:(public net): Prune:  
172.16.1.1/32 (P013179)
```

```
<R4>
```

```
Sep 25 2013 20:39:53.47.5-05:13 R4 PIM/7/JP:(public net): PIM ver 2  
JP sending 10.0.24.4 -> 224.0.0.13 on GigabitEthernet0/0/1 (P013156)
```

```
<R4>
```

```
Sep 25 2013 20:39:53.47.6-05:13 R4 PIM/7/JP:(public net): Upstream  
10.0.24.2, Groups 1, Holdtime 180 (P013160)
```

```
<R4>
```

Sep 25 2013 20:39:53.47.7-05:13 R4 PIM/7/JP:(public net): Group: 224.1.1.1/32 --- 0 join 1 prune (P013169)

<R4>

Sep 25 2013 20:39:53.47.8-05:13 R4 PIM/7/JP:(public net): Prune: 172.16.1.1/32 (P013179)

<R4>

Sep 25 2013 20:39:53.47.9-05:13 R4 PIM/7/JP:(public net): PIM ver 2 JP sending 10.0.24.4 -> 224.0.0.13 on GigabitEthernet0/0/1 (P013156)

<R4>

Sep 25 2013 20:39:53.47.10-05:13 R4 PIM/7/JP:(public net): Upstream 10.0.24.2, Groups 1, Holdtime 180 (P013160)

<R4>

Sep 25 2013 20:39:53.47.11-05:13 R4 PIM/7/JP:(public net): Group: 224.1.1.1/32 --- 0 join 1 prune (P013169)

<R4>

Sep 25 2013 20:39:53.47.12-05:13 R4 PIM/7/JP:(public net): Prune: 172.16.1.1/32 (P013179)

可以看到，R4向R2发送了裁剪消息，使R2不再向R4转发该组播组的数据包。后来R4又收到了来自R2的裁剪消息，因为如果R4将来自R3的组播数据包又转发给R2，R2再转发给R1，这样就会形成组播环路，所以R2与R4之间互相发送裁剪消息，避免了重复包与组播环路的问题。

路由器之所以会发送裁剪消息避免重复包和环路问题，是由于PIM-DM具有RPF校验功能。路由器如果从非RPF接口收到了组播数据包，就会立即从该接口发送裁剪消息。在R4上可以观察到关于组播源172.16.1.1的RPF接口。

<R4>display multicast rpf-info 172.16.1.1

VPN-Instance: public net

RPF information about source: 172.16.1.1

RPF interface: GigabitEthernet0/0/0, RPF neighbor: 10.0.34.3

Referenced route/mask: 172.16.1.0/24

Referenced route type: unicast

Route selection rule: preference-preferred

Load splitting rule: disable

可以看到，对于R4来说，关于组播源172.16.1.1的RPF接口为GE 0/0/0，RPF邻居为R3（10.0.34.3）。RPF校验将依据如下顺序确定出RPF接口：组播静态路由、协议优先级的值最小的路由、Cost最小的路由、下一跳IP地址最大的路由。

在R4上查看单播路由表（注：R4上现在还没有配置组播静态路由）。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```

-----
-----
                                Routing Tables: Public
                                Destinations : 17  Routes : 18
Destination/Mask      Proto      Pre      Cost      Flags      NextHop
Interface
    172.16.1.0/24      OSPF        10       3         D         10.0.34.3
GigabitEthernet0/0/0
    OSPF      10    3    D    10.0.24.2  GigabitEthernet0/0/1
.....

```

可以看到，R4去往组播源172.16.1.1的路由有两条，它们的协议优先级和Cost都是一样的。根据RPF校验规则，确定出来的R4的RPF接

口应该是GE 0/0/0，因为该接口对应的下一跳IP 地址（10.0.34.3）大于接口GE 0/0/1 对应的下一跳IP 地址（10.0.24.2）。最后,通过裁剪，组播转发路径将会是：Source-1 > R1 > R3 > R4 > PC-1。

在R2上查看RPF接口。

```
<R2>display multicast rpf-info 172.16.1.1
```

```
VPN-Instance: public net
```

```
RPF information about source: 172.16.1.1
```

```
RPF interface: GigabitEthernet0/0/0, RPF neighbor: 10.0.12.1
```

```
Referenced route/mask: 172.16.1.0/24
```

```
Referenced route type: unicast
```

```
Route selection rule: preference-preferred
```

```
Load splitting rule: disable
```

从上面的显示信息可知，R2 上关于组播源172.16.1.1的RPF 接口是GE 0/0/0，RPF邻居为R1（10.0.12.1）。

每台路由器上对于一个特定的组播源都只有唯一一个RPF接口，如果来自该组播源的组播数据包不是从这个RPF接口收到的话，将会被直接丢弃。

5．配置组播静态路由

本网络中，Source-1使用组播地址224.1.1.1，Source-2使用组播地址225.1.1.1，二者同时发送组播视频流量，PC-1 加入组播组224.1.1.1，PC-2 加入组播组225.1.1.1。在这样的情况下，两个组播源发送的组播数据都会通过R3转发给R4。为了减轻R3的一部分负担，管理员可以在R4上配置组播静态路由，使得来自Source-1的组播流量由R2转发给R4。

在R4上配置组播静态路由，修改关于组播源172.16.1.1的RPF接口。

```
[R4]ip rpf-route-static 172.16.1.0 24 10.0.24.2
```

配置完成后，在R4上查看关于组播源172.16.1.1的RPF接口。

```
[R4]display multicast rpf-info 172.16.1.1
```

VPN-Instance: public net

RPF information about source: 172.16.1.1

RPF interface: GigabitEthernet0/0/1, RPF neighbor: 10.0.24.2

Referenced route/mask: 172.16.1.0/24

Referenced route type: mstatic

Route selection rule: preference-preferred

Load splitting rule: disable

可以看到，R4 上关于组播源172.16.1.1 的RPF 接口已经变成了GE 0/0/1。

让PC-1加入组播组224.1.1.1，PC-2加入组播组225.1.1.1后，让Source-1和Source-2同时发送组播视频，并在R4 的GE 0/0/1 接口查看报文情况，如图5-38 所示。

No.	Time	Source	Destination	Protocol	Length	Info
15	11.2000000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
16	11.2320000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
17	11.2780000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
18	11.3100000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
19	11.3560000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
20	11.3880000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
21	11.4340000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
22	11.4810000	172.16.1.1	224.1.1.1	UDP	1370	Source port:
23	11.5280000	172.16.1.1	224.1.1.1	UDP	1370	Source port:

图5-38 R4的GE 0/0/1接口报文情况

可以看到，R4的GE 0/0/1接口接收到的是组播地址为224.1.1.1的UDP组播数据流。在R4 的GE 0/0/0 接口查看报文情况，如图5-39 所示。

No.	Time	Source	Destination	Protocol	Length	Info
12	14.1650000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
13	14.2120000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
14	14.2430000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
15	14.2740000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
16	14.3050000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
17	14.3360000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
18	14.3680000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
19	14.3830000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
20	14.4140000	20.0.1.1	225.1.1.1	UDP	1370	Source port:
21	14.4300000	20.0.1.1	225.1.1.1	UDP	1370	Source port:

图5-39 R4的GE 0/0/0接口报文情况

可以看到，R4的GE 0/0/0接口接收到的是组播地址为225.1.1.1的UDP组播数据流。至此，学习视频流量的转发路径为：Source-1 > R1 > R2 > R4 > PC-1，电影视频流量的转发路径为：Source-2 > R3 > R4 > PC-2，R3上的流量负担得以减轻，实现了组播流量分布的优化。

思考

比较RPT（Rendezvous Point Tree）与SPT（Shortest Path Tree）的主要异同点。

第6章 交换技术

6.1 观察和配置MAC地址表

6.2 VLAN基本配置

6.3 VLAN间的通信

6.4 Mux VLAN

6.5 MSTP/RSTP与STP的兼容性

6.6 MSTP/RSTP的保护功能

6.1 观察和配置MAC 地址表

原理概述

MAC地址表是交换机的一个核心组成部分，交换机主要是根据MAC地址表来进行帧的转发的。交换机对帧的转发操作行为一共有3种：泛洪（Flooding）、转发（Forwarding）和丢弃（Discarding）。关于这3种转发操作行为的具体含义在此不再赘述。

在不涉及VLAN的情况下，交换机的转发原理可以概括地描述为：

（1）如果进入交换机的是一个单播帧，则交换机会去 MAC 地址表中查找这个帧的目的 MAC 地址，如果查不到这个 MAC 地址，则交换机将对该帧执行泛洪操作；如果查到了这个 MAC 地址，则比较这个 MAC 地址在 MAC 地址表中对应的端口是不是这个帧进入交换机的那个端口，如果不是，则交换机将对该帧执行转发操作，如果是，则交换机将对该帧执行丢弃操作。（2）如果进入交换机的是一个广播帧，则交换机不会去查 MAC 地址表，而是直接对该帧执行泛洪操作。（3）如果进入交换机的是一个组播帧，则交换机的处理行为比较复杂，超出了这里的学习范围，所以在此不作描述。

交换机具有转发帧的能力，同时还具有 MAC 地址学习能力。当一个帧进入交换机后，交换机会检查这个帧的源 MAC 地址，并将该 MAC 地址与这个帧进入交换机的那个端口进行映射，然后将这个映射关系作为一个动态地址表项存放在 MAC 地址表中。

MAC 地址表是一张动态的表，每个表项在创建或刷新时，都会设定并维护一个默认是 300s 的生存期（也称为老化周期）。一个 MAC 地址表项如果超过了生存期，则该表项会立即被自动清除。

MAC 地址表中的表项分为动态表项和静态表项，前者是交换机通过动态学习过程创建的，后者是通过手工配置创建的。静态表项不存在生存期的概念，并且其优先级高于动态表项：对于一个特定的 MAC 地址，如果手工配置了关于它的静态表项，则 MAC 地址表中将不会再出现关于它的动态表项。

如果两台主机之间通过交换机相连，那么其中一台主机在向另一台主机发送数据帧的时候，会首先在自己的 ARP 缓存表中查找目标主机的 MAC 地址。如果 ARP 缓存表中不存在目标主机的 MAC 地址，则源主机将以广播帧的形式发送 ARP 请求报文来获取目标主机的 MAC 地址，目标主机接收到该 ARP 请求报文后，会以单播帧的形式回应一个 ARP 回复报文，告知自己的 MAC 地址。源主机在获取了目标主机的 MAC 地址后，一方面可利用该 MAC 地址向目标主机发起通信，另一方面会将目标主机的 IP 地址和 MAC 地址建立一个映射关系，并将此映射关系作为一个条目存放在自己的 ARP 缓存表中。ARP 缓存表也是一张动态的表，关于其动态机制这里就不再赘述了。

实验目的

- 理解 MAC 地址表的基本作用和动态特性
- 掌握静态 MAC 地址表项的创建方法

实验内容

实验拓扑如图6-1所示，实验编址如表6-1所示。本实验网络的结构非常简单，只包含一台交换机SW1和三台终端电脑PC-1、PC-2和PC-3。网络管理员需要为终端电脑配置固定的IP地址，然后观察交换机的MAC地址表在终端电脑的通信过程中所发生的各种变化。另外，管理员还需要在SW1上练习使用手动方式创建静态的MAC地址表项。

实验拓扑

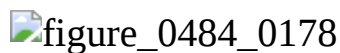
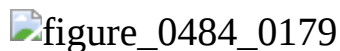
figure_0484_0178

图6-1 观察和配置MAC地址表

实验编址表

表6-1 实验编址

figure_0484_0179

实验步骤

1．基本配置

根据图6-1 和表 6-1 进行相应的 IP 地址配置，同时设置 PC-1 的 MAC 地址为00-01-00-01-00-01，PC-2 的 MAC 地址为 00-02-00-02-00-02，PC-3 的 MAC 地址为00-03-00-03-00-03。然后，在PC-1上使用ping命令检测PC-1和PC-2之间的连通性，如图6-2所示。

PC-1和PC-3之间以及PC-2和PC-3之间的连通性测试过程在此省略。

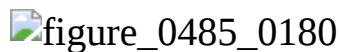
figure_0485_0180

图6-2 在PC-1上测试PC-1和PC-2之间的连通性

2．观察正常状态时的MAC地址表

由于在上述步骤中使用了ping命令测试3台PC之间的连通性，所以交换机应该已经通过动态学习过程而在MAC地址表中创建了关于PC-1、PC-2、PC-3的MAC地址的表项。为了便于观察MAC地址表项的创建过程，可以先用命令把SW1通过动态学习而创建的地址表项清空。

```
[SW1]undo mac-address dynamic
```

然后，在SW1上查看MAC地址表。

```
[SW1]display mac-address
```

可以看到，MAC地址表中现在没有任何内容。

接下来，查看SW1上MAC地址表项的老化周期。

```
[SW1]display mac-address aging-time
```

```
Aging time: 300 seconds
```

可以看到，默认情况下，MAC地址表项的老化周期为300s。

为了便于接下来的实验观察，避免MAC表项因超过了生存期而被清除掉，可以使用命令将老化周期设置为0，这样便关闭了MAC地址表的老化功能，地址表项不会再因超过生存期而被清除。

```
[SW1]mac-address aging-time 0
```

交换机只在接收到数据帧时，才会使用所收到的数据帧的源MAC地址来创建相应的地址表项。如图6-3所示，在PC-1上使用ping命令访问PC-2，这样一来，SW1便会分别接收到以PC-1的MAC地址为源地址和以PC-2的MAC地址为源地址的数据帧。

然后，在SW1上查看MAC地址表。

```
[SW1]display mac-address
```

```
MAC address table of slot 0:
```

MAC Address LSP/LSR-ID	VLAN/ VSI/SI	PEVLAN	CEVLAN	Port	Type

MAC-Tunnel

0001-0001-0001	1	-	-	GE0/0/1	dynamic 0/-

0002-0002-0002	1	-	-	GE0/0/2	dynamic 0/-

Total matching items on slot 0 displayed = 2

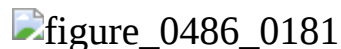


图6-3 在PC-1上ping PC-2

可以看到，SW1分别为MAC地址0001-0001-0001和MAC地址0002-0002-0002创建了地址表项，每个表项包含了MAC地址、VLAN、端口编号、类型等信息。

由于此时SW1尚未收到以PC-3的MAC地址为源地址的数据帧，所以MAC地址表中还没有关于PC-3的MAC地址表项。接下来，在PC-1上使用ping命令访问PC-3，以触发PC-3发送数据帧，如图6-4所示。

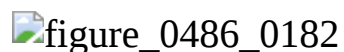


图6-4 在PC-1上ping PC-3

然后，在SW1上查看MAC地址表。

```
[SW1]display mac-address
```

MAC address table of slot 0:

MAC Address	VLAN/	PEVLAN	CEVLAN	Port	Type	
LSP/LSR-ID						
VSI/SI MAC-Tunnel						

0001-0001-0001	1	-	-	GE0/0/1	dynamic	0/-
0002-0002-0002	1	-	-	GE0/0/2	dynamic	0/-
0003-0003-0003	1	-	-	GE0/0/3	dynamic	0/-

Total matching items on slot 0 displayed = 3

可以看到，MAC地址表中现在已经增加了关于PC-3的MAC地址表项。

3．观察MAC地址冲突时的MAC地址表

接下来，修改PC-3的MAC地址为00-02-00-02-00-02，如图6-5所示，以此来模拟PC-3与PC-2产生MAC地址冲突的情况。

figure_0487_0183

图6-5 修改PC-3的MAC地址

然后，在PC-1上使用ping命令访问PC-3，如图6-6所示。

figure_0488_0184

图6-6 在PC-1上ping PC-3

从图6-6中可以看到，PC-3没有任何回应，此时PC-1与PC-3无法进行正常的通信。

在PC-1上查看ARP缓存表，如图6-7所示。

figure_0488_0185

图6-7 查看PC-1上的ARP缓存表

可以看到，此时PC-1的ARP缓存表中，10.0.1.3（PC-3）对应的MAC地址依旧为00-03-00-03-00-03。因此，PC-1发往PC-3的报文其实是封装在目的MAC地址为00-03-00-03-00-03的帧中，当PC-3接收到该帧时，发现该帧的目的MAC地址00-03-00-03-00-03与自己的MAC地址00-02-00-02-00-02不匹配，于是会直接将该帧丢弃。

在PC-1上使用命令`arp -d`清空ARP缓存表，并使用命令`arp -a`来确认ARP缓存表已被清空，如图6-8所示。

figure_0489_0186

图6-8 清空PC-1上的ARP缓存表

从图6-8中可以看到，现在PC-1的ARP缓存表已被清空。在PC-1上使用ping命令访问PC-3，如图6-9所示。

figure_0489_0187

图6-9 在PC-1上ping PC-3

可以看到，现在PC-1可以ping通PC-3了。在SW1上查看MAC地址表。

```
[SW1]display mac-address
```

MAC address table of slot 0:

MAC Address	VLAN/	PEVLAN	CEVLAN	Port	Type	
LSP/LSR-ID						
VSI/SI MAC-Tunnel						

0001-0001-0001	1	-	-	GE0/0/1	dynamic	0/-
0002-0002-0002	1	-	-	GE0/0/3	dynamic	0/-
0003-0003-0003	1	-	-	GE0/0/3	dynamic	0/-

Total matching items on slot 0 displayed = 3

观察发现，此时在 SW1 的 MAC 地址表中，关于 MAC 地址 0002-0002-0002 的表项的端口编号已经由原来的 GE 0/0/2 变为了 GE 0/0/3，这是因为以 PC-3 为源的数据帧经过 SW1 时，0002-0002-0002 这一表项所对应的端口编号被刷新为连接 PC-3 的 GE 0/0/3。

在 PC-1 上使用 ping 命令访问 PC-2，触发 PC-2 发送数据帧，如图 6-10 所示。

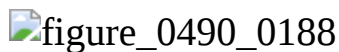
figure_0490_0188

图 6-10 在 PC-1 上 ping PC-2

然后，查看 SW1 上的 MAC 地址表。


```
[SW1]display mac-address
```

MAC address table of slot 0:

MAC Address	VLAN/	PEVLAN	CEVLAN	Port	Type	
LSP/LSR-ID						

VSI/SI MAC-Tunnel

0001-0001-0001	1	-	-	GE0/0/1	dynamic	0/-
0002-0002-0002	1	-	-	GE0/0/2	dynamic	0/-
0003-0003-0003	1	-	-	GE0/0/3	dynamic	0/-

Total matching items on slot 0 displayed = 3

观察发现，此时在SW1 的MAC 地址表中，关于 MAC 地址0002-0002-0002的表项的端口编号又由原来的GE 0/0/3 变为了GE 0/0/2，这是因为以PC-2 为源的数据帧经过SW1 时，0002-0002-0002 这一表项所对应的端口编号被刷新为连接PC-2 的GE 0/0/2。

PC-2与PC-3的MAC地址目前是相同的，处于冲突的状态。当不断地有以PC-2为源的帧和以PC-3为源的帧通过SW1时，SW1就需要频繁地刷新0002-0002-0002这个表项所对应的端口编号，从而产生常说的 MAC 地址表翻转现象，耗费大量的系统资源，并且可能会导致通信异常或通信数据丢失的现象。

在PC-2和PC-3上分别使用ping命令访问PC-1，以此来模拟产生MAC地址冲突的设备同时有通信流量需要经过SW1的情形，如图6-11和图6-12所示。

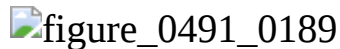
figure_0491_0189

图6-11 在PC-2上ping PC-1

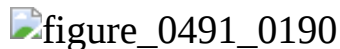
figure_0491_0190

图6-12 在PC-3上ping PC-1

查看SW1，发现系统日志输出了如下的警告信息。

```
Aug 8 2013 16:09:33-08:00 SW1 L2IFPPI/4/MFLPVLANALARM:OID
1.3.6.1.4.1.2011.5.25.
```

```
160.3.7 MAC move detected, VlanId = 1, MacAddress = 0002-0002-0002,
Original-Port
```

```
= GE0/0/3, Flapping port = GE0/0/2. Please check the network accessed to
flappin
```

```
g port.
```

上面的显示信息表明，系统已经出现了MAC地址表翻转现象，要求用户进行相应的检查。

4．配置静态MAC地址表项

MAC地址表项可以通过手工配置来创建，所创建的表项称为静态表项。例如，针对PC-3，可以手工配置一个MAC地址为0003-0003-0003、对应VLAN为1，对应端口编号为GE 0/0/3的MAC地址表项。

```
[SW1]mac-address static 3-3-3 GigabitEthernet 0/0/3 vlan 1
```

配置完成后，在SW1上查看MAC地址表。

[SW1]display mac-address

MAC address table of slot 0:

MAC Address LSP/LSR-ID	VLAN/	PEVLAN	CEVLAN	Port	Type

VSI/SI MAC-Tunnel

0003-0003-0003	1	-	-	GE0/0/3	
static	-				

Total matching items on slot 0 displayed = 1

MAC address table of slot 0:

MAC Address LSP/LSR-ID	VLAN/	PEVLAN	CEVLAN	Port	Type

VSI/SI MAC-Tunnel

0001-0001-0001	1	-	-	GE0/0/1	dynamic	0/-
0002-0002-0002	1	-	-	GE0/0/3	dynamic	0/-

Total matching items on slot 0 displayed = 2

可以看到，此时MAC地址表中多出了一个关于MAC地址为0003-0003-0003的表项，类型为static，而原有的MAC地址为0003-0003-0003、类型为dynamic的表项不再存在，这是因为静态表项的优先级要高于动态表项。

在SW1上为PC-1和PC-2也创建静态MAC地址表项。

```
[SW1]mac-address static 1-1-1 GigabitEthernet 0/0/1 vlan 1
```

```
[SW1]mac-address static 2-2-2 GigabitEthernet 0/0/2 vlan 1
```

在PC-1上使用ping命令访问PC-2和PC-3，如图6-13和图6-14所示。

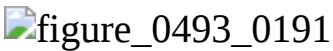
figure_0493_0191

图6-13 在PC-1上ping PC-2

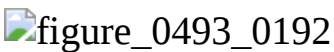
figure_0493_0192

图6-14 在PC-1上ping PC-3

可以看到，此时PC-1能够与PC-2正常通信，但与PC-3无法通信。

在SW1上观察MAC地址表。

```
[SW1]display mac-address
```

MAC address table of slot 0:

MAC Address LSP/LSR-ID	VLAN/	PEVLAN	CEVLAN	Port	Type

VSI/SI MAC-Tunnel

0001-0001-0001	1	-	-	GE0/0/1	static	-
0002-0002-0002	1	-	-	GE0/0/2	static	-
0003-0003-0003	1	-	-	GE0/0/3	static	-

Total matching items on slot 0 displayed = 3

可以看到，此时MAC 地址表中的3个表项的类型均为static，并且没有因为PC-1访问了PC-3 而使得0002-0002-0002 表项的接口由GE 0/0/2 变更为GE 0/0/3。

接下来，将PC-3的MAC地址修改为正确的00-03-00-03-00-03，然后再在PC-1上使用ping命令访问PC-3，如图6-15所示。

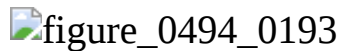
figure_0494_0193

图6-15 在PC-1上ping PC-3

从图6-15中可以看到，此时PC-1能够与PC-3进行正常的通信了。

思考

创建静态的MAC地址表项来解决MAC地址冲突问题时有什么缺点？

6.2 VLAN 基本配置

原理概述

关于VLAN（Virtual Local Area Network：虚拟局域网）的基本概念和作用，相信读者已经有了一定的了解和认识，所以这里不再赘述。这里只简要回顾一下交换机VLAN端口的3种类型。

交换机的VLAN端口可以分为Access、Trunk和Hybrid 3种类型。Access端口是交换机上用来直接连接用户终端的端口，它只允许属于该端口的缺省VLAN的帧通过。Access端口发往用户终端的帧一定不带VLAN标签。Trunk端口是交换机上用来连接其他交换机的端口，它可以允许属于多个VLAN的帧通过。Hybrid端口是交换机上既可以连接用户终端，又可以连接其他交换机的端口。Hybrid端口也可以允许属于多个VLAN的帧通过，并且可以在出端口的方向上将某些VLAN帧的标签剥掉。

实验目的

- 理解Access端口、Trunk端口和Hybrid端口的作用与区别
- 掌握基本的VLAN配置方法

实验内容

实验拓扑如图6-16所示，实验编址如表6-2所示。本实验模拟了一个简单的公司网络场景，SW1和SW2为楼层交换机，PC-1和PC-3属于公司的部门A，PC-2和PC-4属于公司的部门B，PC-5属于部门A和部门B的上级部门C。在网络规划中，部门A属于VLAN 10，部门B属于VLAN 20，部门C属于VLAN 30。公司希望通过VLAN的划分和配置，使各部门内部之间可以互相通信，部门A和部门B均能够与部门C进行通信，但是部门A与部门B之间不能互相通信。

实验拓扑

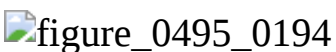
figure_0495_0194

图6-16 VLAN基本配置

实验编址表

表6-2 实验编址

figure_0496_0195

实验步骤

1．基本配置

根据图6-16和表6-2配置各终端电脑的IP地址及掩码，图6-17只示意了PC-1的配置过程。

figure_0496_0196

图6-17 PC-1的配置过程

2．创建VLAN

在SW1 上使用命令vlan 分别创建VLAN 10、VLAN 20 和VLAN 30。

```
[SW1]vlan 10
```

```
[SW1-vlan10]vlan 20
```

```
[SW1-vlan20]vlan 30
```

配置完成后，在SW1 上使用命令display vlan查看当前交换机VLAN 的相关信息。

```
[SW1]display vlan
```

```
The total number of vlans is : 4
```

```
-----  
-----
```

U:Up; D:Down; TG:Tagged; UT:Untagged;

MP:Vlan-mapping; ST:Vlan-stacking;

#:ProtocolTransparent-vlan; *:Management-vlan;

VID	Type	Ports

1	common UT:	GE0/0/1(U) GE0/0/2(U) GE0/0/3(D) GE0/0/4(D)

		GE0/0/5(D) GE0/0/6(D) GE0/0/7(D) GE0/0/8(D)

		GE0/0/9(D) GE0/0/10(D) GE0/0/11(D) GE0/0/12(D)

		GE0/0/13(D) GE0/0/14(D) GE0/0/15(D) GE0/0/16(D)

		GE0/0/17(D) GE0/0/18(D) GE0/0/19(D) GE0/0/20(D)

		GE0/0/21(D) GE0/0/22(D) GE0/0/23(D) GE0/0/24(U)

10	common	

20	common	

30	common	

VID	Status	Property	MAC-LRN	Statistics	Description

.....

可以看到，SW1 上已经创建了VLAN 1、VLAN 10、VLAN 20 和
VLAN 30。默认情况下，VLAN 1 无需手工创建就会自动存在，并且

所有端口都默认属于 VLAN 1。从上面的显示信息还可以看到，目前 VLAN 10、VLAN 20、VLAN 30 尚未被分配任何端口。

在SW2 上使用命令vlan batch一次性批量创建VLAN 10、VLAN 20 和 VLAN 30。

```
[SW2]vlan batch 10 20 30
```

配置完成后，在SW2上查看VLAN的相关信息。

```
[SW2]display vlan
```

```
The total number of vlans is : 4
```

```
-----  
-----
```

```
U:Up;      D:Down;      TG:Tagged;      UT:Untagged;
```

```
MP:Vlan-mapping;      ST:Vlan-stacking;
```

```
#:ProtocolTransparent-vlan; *:Management-vlan;
```

```
-----  
-----
```

```
VID   Type   Ports
```

```
-----  
-----
```

```
1   common UT:  GE0/0/1(U)  GE0/0/2(U)  GE0/0/3(U)  GE0/0/4(D)
```

```
GE0/0/5(D)  GE0/0/6(D)  GE0/0/7(D)  GE0/0/8(D)
```

```
GE0/0/9(D)  GE0/0/10(D) GE0/0/11(D) GE0/0/12(D)
```

```
GE0/0/13(D) GE0/0/14(D) GE0/0/15(D) GE0/0/16(D)
```

GE0/0/17(D) GE0/0/18(D) GE0/0/19(D) GE0/0/20(D)

GE0/0/21(D) GE0/0/22(D) GE0/0/23(D) GE0/0/24(U)

10 common

20 common

30 common

VID	Status	Property	MAC-LRN	Statistics	Description

.....

可以看到，此时SW2上已经成功创建了相应的VLAN。

3．配置Access端口并划分VLAN

交换机上虽然创建了相应的VLAN，但由于终端设备无法识别和处理 Tagged帧，所以还需要配置Access 端口。接下来，在SW1 的接口视图下配置GE 0/0/1 和GE 0/0/2 为Access端口。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port link-type access
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port link-type access
```

然后，在接口视图下分别将GE 0/0/1 和GE 0/0/2 划入VLAN 10 和VLAN 20。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port default vlan 10
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port default vlan 20
```

配置完成后，在SW1上查看VLAN的相关信息。

```
<SW1>display vlan
```

The total number of vlans is : 4

U:Up; D:Down; TG:Tagged; UT:Untagged;

MP:Vlan-mapping; ST:Vlan-stacking;

#:ProtocolTransparent-vlan; *:Management-vlan;

VID Type Ports

1 common UT:GE0/0/3(D) GE0/0/4(D) GE0/0/5(D) GE0/0/6(D)

GE0/0/7(D) GE0/0/8(D) GE0/0/9(D) GE0/0/10(D)

GE0/0/11(D) GE0/0/12(D) GE0/0/13(D) GE0/0/14(D)

GE0/0/15(D) GE0/0/16(D) GE0/0/17(D) GE0/0/18(D)

GE0/0/19(D) GE0/0/20(D) GE0/0/21(D) GE0/0/22(D)

GE0/0/23(D) GE0/0/24(U)

10 common UT:GE0/0/1(U)

```
20 common UT:GE0/0/2(U)
```

```
30 common
```

VID	Status	Property	MAC-LRN	Statistics	Description

.....

可以看到，此时GE 0/0/1端口和GE 0/0/2端口已分别被划入VLAN 10和VLAN 20。

在SW2上进行类似的配置。

```
[SW2]interface GigabitEthernet 0/0/1
```

```
[SW2-GigabitEthernet0/0/1]port link-type access
```

```
[SW2-GigabitEthernet0/0/1]port default vlan 10
```

```
[SW2-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW2-GigabitEthernet0/0/2]port link-type access
```

```
[SW2-GigabitEthernet0/0/2]port default vlan 20
```

```
[SW2-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
```

```
[SW2-GigabitEthernet0/0/3]port link-type access
```

```
[SW2-GigabitEthernet0/0/3]port default vlan 30
```

配置完成后，在SW2上查看VLAN的相关信息。

```
[SW2]display vlan
```

```
The total number of vlans is : 4
```

U:Up; D:Down; TG:Tagged; UT:Untagged;

MP:Vlan-mapping; ST:Vlan-stacking;

#:ProtocolTransparent-vlan; *:Management-vlan;

VID	Type	Ports

1	common	UT:GE0/0/4(U) GE0/0/5(D) GE0/0/6(D) GE0/0/7(D) GE0/0/8(D) GE0/0/9(D) GE0/0/10(D) GE0/0/11(D) GE0/0/12(D) GE0/0/13(D) GE0/0/14(D) GE0/0/15(D) GE0/0/16(D) GE0/0/17(D) GE0/0/18(D) GE0/0/19(D) GE0/0/20(D) GE0/0/21(D) GE0/0/22(D) GE0/0/23(D) GE0/0/24(U)

10 common UT:GE0/0/1(U)

20 common UT:GE0/0/2(U)

30 common UT:GE0/0/3(U)

VID	Status	Property	MAC-LRN	Statistics	Description

.....

可以看到，SW2上相应的端口都已被划分到相应的VLAN之中。

4．配置Trunk端口实现跨交换机通信

在PC-1上使用ping命令测试与PC-3间的连通性，如图6-18所示。

figure_0499_0197

图6-18 在PC-1上ping PC-3

从图6-18中可以看到，目前PC-1与PC-3还无法进行通信，这是由于PC-1和PC-3虽然属于同一个VLAN 20，但是二者间的通信需要跨越不同的交换机，为此，还需要将SW1和SW2之间的链路配置为Trunk链路，并允许携带VLAN 20标签的帧通过。

首先，在SW1上查看VLAN的相关信息。

```
[SW1]display vlan
```

```
The total number of vlans is : 4
```

```
-----  
-----
```

```
U:Up;      D:Down;    TG:Tagged;    UT:Untagged;
```

```
MP:Vlan-mapping;      ST:Vlan-stacking;
```

```
#:ProtocolTransparent-vlan; *:Management-vlan;
```

```
-----  
-----
```

```
VID  Type  Ports
```

```
-----  
-----
```

```
1   common UT:GE0/0/3(D) GE0/0/4(D) GE0/0/5(D) GE0/0/6(D)
```

```
GE0/0/7(D) GE0/0/8(D) GE0/0/9(D) GE0/0/10(D)
```

```
GE0/0/11(D) GE0/0/12(D) GE0/0/13(D) GE0/0/14(D)
```

GE0/0/15(D) GE0/0/16(D) GE0/0/17(D) GE0/0/18(D)

GE0/0/19(D) GE0/0/20(D) GE0/0/21(D) GE0/0/22(D)

GE0/0/23(D) GE0/0/24(U)

10 common UT:GE0/0/1(U)

20 common UT:GE0/0/2(U)

.....

可以看到，此时SW1 的GE 0/0/24 端口是以Untagged 的形式被划分至 VLAN 1 中的，此端口目前只能转发属于VLAN 1 的帧，而PC-1 和 PC-3 是属于VLAN 10 的，所以PC-1和PC-3目前还不能进行通信。

在SW1 的GE 0/0/24 接口下配置该端口为Trunk 端口。

```
[SW1]interface GigabitEthernet 0/0/24
```

```
[SW1-GigabitEthernet0/0/24]port link-type trunk
```

然后，使用命令port trunk allow-pass vlan让GE 0/0/24 端口允许VLAN 10、VLAN 20、VLAN 30 的帧通过。注意，默认情况下，Trunk 端口只允许VLAN 1 的帧通过。

```
[SW1]interface GigabitEthernet 0/0/24
```

```
[SW1-GigabitEthernet0/0/24]port trunk allow-pass vlan 10 20 30
```

在SW2上完成类似的配置。

```
[SW2]interface GigabitEthernet 0/0/24
```

```
[SW2-GigabitEthernet0/0/24]port link-type trunk
```

```
[SW2-GigabitEthernet0/0/24]port trunk allow-pass vlan 10 20 30
```

配置完成后，在SW1上查看VLAN的相关信息。

```
[SW1]display vlan
```

The total number of vlans is : 4

U:Up; D:Down; TG:Tagged; UT:Untagged;

MP:Vlan-mapping; ST:Vlan-stacking;

#:ProtocolTransparent-vlan; *:Management-vlan;

VID Type Ports

1 common UT:GE0/0/3(D) GE0/0/4(D) GE0/0/5(D) GE0/0/6(D)

GE0/0/7(D) GE0/0/8(D) GE0/0/9(D) GE0/0/10(D)

GE0/0/11(D) GE0/0/12(D) GE0/0/13(D) GE0/0/14(D)

GE0/0/15(D) GE0/0/16(D) GE0/0/17(D) GE0/0/18(D)

GE0/0/19(D) GE0/0/20(D) GE0/0/21(D) GE0/0/22(D)

GE0/0/23(D) GE0/0/24(U)

10 common UT:GE0/0/1(U)

TG:GE0/0/24(U)


```
20 common UT:GE0/0/2(U)
```

```
TG:GE0/0/24(U)
```

```
30 common TG:GE0/0/24(U)
```

```
VID Status Property MAC-LRN Statistics Description
```

```
.....
```

可以看到，SW1的GE 0/0/24端口现在以Tagged的形式被划分进了VLAN 10、VLAN 20 和VLAN 30 中。在SW2 上也可看到同样的结果，此处不再赘述。

在PC-1上再次使用ping命令测试与PC-3间的连通性，如图6-19所示。

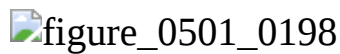
figure_0501_0198

图6-19 在PC-1上测试PC-1和PC-3间的连通性

从图6-19中可以看到，现在PC-1与PC-3可以进行正常的通信了。

5．使用Hybrid接口实现不同VLAN间的通信

实验进行到这里时，相同VLAN的终端之间已经可以实现相互通信了。然而，根据要求，部门A和部门B还需要能够与部门C实现通信。

在PC-1上使用ping命令测试与PC-5间的连通性，如图6-20所示。

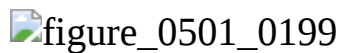
figure_0501_0199

图6-20 PC-1和PC-5之间的连通性测试

从图6-20中可以看到，此时PC-1与PC-5还无法实现通信。下面将采用Hybrid端口的方法来解决这个问题。

在SW1连接终端的接口视图下，先使用undo命令删除之前的Access端口配置命令。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]undo port default vlan
```

```
[SW1-GigabitEthernet0/0/1]undo port link-type
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]undo port default vlan
```

```
[SW1-GigabitEthernet0/0/2]undo port link-type
```

接下来，使用port link-type hybrid命令修改端口为Hybrid端口。注意，默认情况下，交换机的端口就是Hybrid端口。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port link-type hybrid
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port link-type hybrid
```

接下来，使用命令port hybrid untagged vlan 指示端口需要将携带相应VLAN 标签的帧以Untagged的形式进行发送。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port hybrid untagged vlan 10 30
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port hybrid untagged vlan 20 30
```

接下来，使用命令port hybrid pvid vlan 指示端口对收到的Untagged 帧添加相应的VLAN标签。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port hybrid pvid vlan 10
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port hybrid pvid vlan 20
```

最后，在 SW1 与 SW2 相连的端口，修改端口类型为 Hybrid，并将端口以 Tagged方式加入进VLAN 10，VLAN 20，VLAN 30。

```
[SW1]interface GigabitEthernet 0/0/24
```

```
[SW1-GigabitEthernet0/0/24]port link-type hybrid
```

```
[SW1-GigabitEthernet0/0/24]port hybrid tagged vlan 10 20 30
```

在SW2上也进行类似的操作配置。

```
[SW2]interface GigabitEthernet 0/0/1
```

```
[SW2-GigabitEthernet0/0/1]port link-type hybrid
```

```
[SW2-GigabitEthernet0/0/1]port hybrid untagged vlan 10 30
```

```
[SW2-GigabitEthernet0/0/1]port hybrid pvid vlan 10
```

```
[SW2-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW2-GigabitEthernet0/0/2]port link-type hybrid
```

```
[SW2-GigabitEthernet0/0/2]port hybrid untagged vlan 20 30
```

```
[SW2-GigabitEthernet0/0/2]port hybrid pvid vlan 20
```

```
[SW2-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
```

```
[SW2-GigabitEthernet0/0/3]port link-type hybrid
```

```
[SW2-GigabitEthernet0/0/3]port hybrid untagged vlan 10 20 30
```

```
[SW2-GigabitEthernet0/0/3]port hybrid pvid vlan 30
```

```
[SW2-GigabitEthernet0/0/3]interface GigabitEthernet 0/0/24
```

```
[SW2-GigabitEthernet0/0/24]port link-type hybrid
```

```
[SW2-GigabitEthernet0/0/24]port hybrid tagged vlan 10 20 30
```

配置完成后，在SW1上查看VLAN的相关信息。

```
<SW1>display vlan
```

The total number of vlans is : 4

```
-----  
-----
```

U:Up; D:Down; TG:Tagged; UT:Untagged;

MP:Vlan-mapping; ST:Vlan-stacking;

#:ProtocolTransparent-vlan; *:Management-vlan;

```
-----  
-----
```

VID Type Ports

```
-----  
-----
```

1 common UT:GE0/0/1(U) GE0/0/2(U) GE0/0/3(D) GE0/0/4(D)

GE0/0/5(D) GE0/0/6(D) GE0/0/7(D) GE0/0/8(D)

```

GE0/0/9(D)  GE0/0/10(D)  GE0/0/11(D)  GE0/0/12(D)
GE0/0/13(D)  GE0/0/14(D)  GE0/0/15(D)  GE0/0/16(D)
GE0/0/17(D)  GE0/0/18(D)  GE0/0/19(D)  GE0/0/20(D)
GE0/0/21(D)  GE0/0/22(D)  GE0/0/23(D)  GE0/0/24(U)

10  common UT:GE0/0/1(U)
TG:GE0/0/24(U)

20  common UT:GE0/0/2(U)
TG:GE0/0/24(U)

30  common UT:GE0/0/1(U)  GE0/0/2(U)
TG:GE0/0/24(U)

```

```

VID  Status  Property  MAC-LRN Statistics Description
.....

```

可以看到，此时GE 0/0/1 已经以Untagged 的形式被划分至VLAN 10 和 VLAN 30，对于 VLAN 10 和 VLAN 30 的帧，都会剥离其 VLAN 标签后发送；GE 0/0/2 已经以Untagged 的形式被划分至VLAN 20 与VLAN 30，对于VLAN 20 和VLAN 30 的帧，都会剥离其VLAN标签后发送。在SW2上可以看到类似的结果，此处不再赘述。

在PC-1上使用ping命令分别测试与PC-3、PC-5的连通性，如图6-21所示。

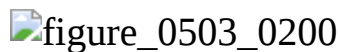
figure_0503_0200

图6-21 PC-1与PC-3、PC-5之间的连通性测试

从图6-21中可以看到，现在PC-1既能够与属于同一VLAN的PC-3进行通信，也能够与属于不同VLAN的PC-5进行通信。

在PC-1上使用ping命令测试与PC-4的连通性，如图6-22所示。

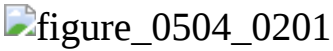
figure_0504_0201

图6-22 PC-1与PC-4之间的连通性测试

从图6-22中可以看到，部门A的PC-1不能与部门B的PC-4进行通信。至此，所有的需求都得到了实现。

思考

Hybrid端口能否实现Trunk端口的功能？如果能，应该如何实现？

6.3 VLAN 间的通信

原理概述

通常情况下，如果不采用一些特殊的方法（如采用Hybrid端口的方法），不同的VLAN之间是不能够进行二层（数据链路层）通信的，这也是VLAN技术的基本出发点；一般地，VLAN之间的通信是需要第三层（网络层）才能实现的。

实现VLAN间的三层通信的方法有很多，最为传统的方法是使用路由器。除此之外，常用的方法还有很多，例如，在交换机上使用VLANIF接口，在交换机上使用VLAN聚合方法等。

VLANIF接口只是一个逻辑意义上的三层接口。采用VLANIF接口的方法时，每一个VLAN都对应了交换机上的一个VLANIF接口，不同的VLAN对应了不同的VLANIF接口，并且每个VLAN中的终端设备的网关地址就是所对应的VLANIF接口的IP地址。显然，使用VLANIF接口方法的一个主要缺点就是比较耗费IP地址资源，这是因为每一个不

同的VLAN都必须对应一个不同的VLANIF接口，而每个不同的VLANIF接口都必须配置一个不同的IP地址。

VLAN间的通信也可以通过使用VLAN聚合的方法来实现。VLAN聚合使用了两种类型的VLAN，分别称为Sub-VLAN和Super-VLAN。VLAN聚合的方法可以节省大量的IP地址资源，这是因为一个Super-VLAN需要配置一个VLANIF接口，并为该VLANIF接口配置一个IP地址，但该Super-VLAN下的各个Sub-VLAN都无需再配置VLANIF接口。

实验目的

- 理解VLAN 之间二层通信和三层通信的区别
- 掌握VLANIF 接口的配置方法
- 掌握VLAN 聚合的配置方法

实验内容

实验拓扑如图6-23所示，实验编址如表6-3所示。本实验模拟了一个简单的公司网络场景，SW1和SW3为楼层交换机，SW2为核心交换机，PC-1和PC-3是部门A的终端电脑，PC-2 和PC-4 是部门B 的终端电脑。根据网络规划，部门A 属于VLAN 2，部门B 属于VLAN 3。网络管理员需要分别使用VLANIF 接口的方法和VLAN 聚合的方法来实现部门A与部门B之间的通信。

实验拓扑

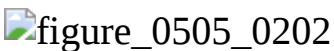
figure_0505_0202

图6-23 VLAN间的通信

实验编址表

表6-3 实验编址

figure_0506_0203

实验步骤

1．基本配置

根据图6-23和表6-3进行相应的基本配置，PC-1的配置过程示意如图6-24所示。

figure_0506_0204

图6-24 PC-1的配置过程

在SW1 和SW3 上创建VLAN 2 和VLAN 3，并将相应的端口划分至对应的VLAN。

```
[SW1]vlan batch 2 3
```

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port link-type access
```

```
[SW1-GigabitEthernet0/0/1]port default vlan 2
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port link-type access
```

```
[SW1-GigabitEthernet0/0/2]port default vlan 3
```

```
[SW3]vlan batch 2 3
```

```
[SW3]interface GigabitEthernet 0/0/1
```

```
[SW3-GigabitEthernet0/0/1]port link-type access
```

```
[SW3-GigabitEthernet0/0/1]port default vlan 2
```

```
[SW3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```



```
[SW3-GigabitEthernet0/0/2]port link-type access
```

```
[SW3-GigabitEthernet0/0/2]port default vlan 3
```

在SW1、SW2和SW3上完成Trunk端口的配置，并允许所有VLAN的帧通过Trunk链路。

```
[SW1]interface GigabitEthernet 0/0/23
```

```
[SW1-GigabitEthernet0/0/23]port link-type trunk
```

```
[SW1-GigabitEthernet0/0/23]port trunk allow-pass vlan all
```

```
[SW3]interface GigabitEthernet 0/0/24
```

```
[SW3-GigabitEthernet0/0/24]port link-type trunk
```

```
[SW3-GigabitEthernet0/0/24]port trunk allow-pass vlan all
```

```
[SW2]interface GigabitEthernet 0/0/23
```

```
[SW2-GigabitEthernet0/0/23]port link-type trunk
```

```
[SW2-GigabitEthernet0/0/23]port trunk allow-pass vlan all
```

```
[SW2-GigabitEthernet0/0/23]interface GigabitEthernet 0/0/24
```

```
[SW2-GigabitEthernet0/0/24]port link-type trunk
```

```
[SW2-GigabitEthernet0/0/24]port trunk allow-pass vlan all
```

配置完成后，在PC-1上使用ping命令测试与PC-3的连通性，如图6-25所示。

 figure_0507_0205

图6-25 PC-1与PC-3之间的连通性测试

奇怪的是，此时 PC-1 并不能与 PC-3 进行通信。究其原因，原来是忘记了在 SW2上创建VLAN 2 和VLAN 3。

在SW2 上创建VLAN 2 和VLAN 3。

```
[SW2]vlan batch 2 3
```

再次在PC-1上使用ping命令测试与PC-3的连通性，如图6-26所示。

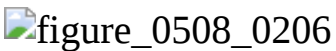
figure_0508_0206

图6-26 PC-1与PC-3之间的连通性测试

可以看到，现在PC-1与PC-3能够正常通信了。

在PC-1上使用ping命令测试与PC-2、PC-4的连通性，如图6-27所示。

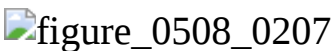
figure_0508_0207

图6-27 PC-1与PC-2、PC-4之间的连通性测试

可以看到，由于PC-1与PC-2和PC-4不属于同一VLAN，所以它们之间的通信现在还无法实现。

2．使用VLANIF接口实现VLAN间的通信

在SW2 上为VLAN 2 创建VLANIF 接口，并为VLANIF 接口配置IP 地址，该IP地址应该是属于VLAN 2 的终端电脑的网关地址。

```
[SW2]interface Vlanif 2
```

```
[SW2-Vlanif2]ip address 10.0.1.100 24
```

在SW2 上为VLAN 3 创建VLANIF 接口，并为VLANIF 接口配置IP 地址，该IP地址应该是属于VLAN 3 的终端电脑的网关地址。

```
[SW2]interface Vlanif 3
```

```
[SW2-Vlanif3]ip address 10.0.2.100 24
```

配置完成后，在PC-1上使用ping命令测试与PC-3、PC-2以及PC-4的连通性，如图6-28所示。

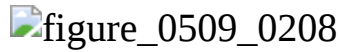
figure_0509_0208

图6-28 PC-1与PC-3、PC-2以及PC-4之间的连通性测试

可以看到，属于不同VLAN的终端之间现在能够正常通信了。

3．使用VLAN聚合实现VLAN间的通信

为了减少VLANIF接口的数目，节省IP地址的使用，我们还可以使用VLAN聚合的方法来实现不同VLAN间的通信。

首先，在SW2上清除掉之前已经配置的VLANIF接口（注意，Sub-VLAN不能拥有VLANIF接口），并创建VLAN 4；VLAN 4将会作为Super-VLAN。

```
[SW2]undo interface Vlanif 2
```

```
[SW2]undo interface Vlanif 3
```

```
[SW2]vlan 4
```

Super-VLAN是不能包含任何物理端口的，但目前SW2的G0/0/23端口和G0/0/24端口已经作为Trunk端口被划分进了所有VLAN，所以，需要将G0/0/23端口和G0/0/24端口从VLAN 4中移除，方法如下。

```
[SW2]interface GigabitEthernet 0/0/23
```

```
[SW2-GigabitEthernet0/0/23]undo port trunk allow-pass vlan 4
```

```
[SW2-GigabitEthernet0/0/23]interface GigabitEthernet 0/0/24
```

```
[SW2-GigabitEthernet0/0/24]undo port trunk allow-pass vlan 4
```

然后，配置VLAN 4 为Super-VLAN，并将VLAN 2 与VLAN 3 作为Sub-VLAN 划分进Super-VLAN。

```
[SW2]vlan 4
```

```
[SW2-vlan4]aggregate-vlan
```

```
[SW2-vlan4]access-vlan 2 3
```

配置完成后，在SW2 上为VLAN 4 创建VLANIF 接口，并配置该接口的IP 地址为10.0.0.100，然后开启ARP代理功能。

```
[SW2]interface Vlanif 4
```

```
[SW2-Vlanif4]ip address 10.0.0.100 16
```

```
[SW2-Vlanif4]arp-proxy inner-sub-vlan-proxy enable
```

最后，将所有终端电脑的网关修改设置为Super-VLAN的VLANIF接口的IP地址，即10.0.0.100（请读者自行完成此操作）。

测试PC-1与PC-3、PC-2、PC-4的连通性，如图6-29所示。

figure_0510_0209

图6-29 PC-1与PC-3、PC-2、PC-4之间的连通性测试

可以看到，属于不同VLAN的终端之间可以进行正常的通信了。

思考

使用VLAN聚合方法与使用VLANIF接口方法实现不同VLAN间的通信各有什么优点？

6.4 Mux VLAN

原理概述

在实际的企业网络环境中，往往需要所有的终端用户都能够访问某些特定的服务器，而用户之间的访问控制规则则比较复杂。在这样的场景下，使用普通VLAN划分的方法往往是很难满足需求的，通常的解决方法是使用Mux VLAN。

Mux VLAN 拥有一个Principal VLAN，即主VLAN，同时拥有多个与主VLAN 关联的Subordinate VLAN，即从VLAN。从VLAN 又有两种类型，一种是Separate VLAN，即隔离型从VLAN，另一种是Group VLAN，即互通型从VLAN。任何从VLAN 中的设备都能够与主VLAN 中的设备进行通信。除此之外，互通型从VLAN 中的设备只能与本互通型从VLAN中的设备进行通信，不能与其他互通型从VLAN 中的设备进行通信，也不能与隔离型从VLAN 中的设备进行通信；隔离型从VLAN 中的设备不能与互通型从VLAN中的设备进行通信，也不能与其他隔离型从VLAN中以及本隔离型从VLAN中的设备进行通信。

另外需要说明的是，交换机上加入Mux VLAN 的端口只能允许一个VLAN 的帧通过，允许多个VLAN 的帧通过的端口是不能被加入到Mux VLAN 中的。

实验目的

- 理解Mux VLAN 的应用场景
- 掌握Mux VLAN 的配置方法

实验内容

实验拓扑如图6-30所示，实验编址如表6-4所示。本实验模拟了一个公司网络，PC-1和PC-2为部门A的终端电脑，PC-3和PC-4为访客的终端电脑，Server-1为公司服务器。网络需求是：访客只能与服务器进行通信，不能与其他访客以及部门A的终端进行通信。部门A的终端可以与服务器进行通信，也可以与部门A的其他终端进行通信。网络管

理员需要分别使用 Hybrid 端口的方法和 Mux VLAN 的方法来实现上述需求。

实验拓扑

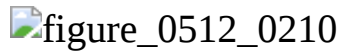
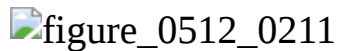
figure_0512_0210

图6-30 Mux VLAN

实验编址表

表6-4 实验编址

figure_0512_0211

实验步骤

1．基本配置

根据图6-30 和表 6-4 进行相应的基本配置，PC-1 的配置过程示意如图6-31所示。

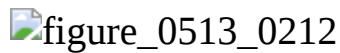
figure_0513_0212

图6-31 PC-1的配置过程

在PC-1上使用Ping命令测试与PC-2的连通性，如图6-32所示。

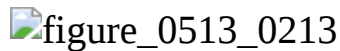
figure_0513_0213

图6-32 PC-1与PC-2之间的连通性测试

其他设备之间的连通性测试过程在此省略。

2．使用Hybrid端口实现网络需求

基于网络需求，可以将Server-1 划分至VLAN 10，部门A 划分至VLAN 20，访客A 划分至VLAN 30，访客B 划分至VLAN 40。

在SW1上创建相应的VLAN。

```
[SW1]vlan batch 10 20 30 40
```

对于SW1 的GE 0/0/1 端口和GE 0/0/2 端口，配置端口类型为Hybrid，并要求端口对于收到的Untagged 帧添加VLAN 20 的标签后进行转发，且在发送属于VLAN 10 和VLAN 20 的帧之前进行去标签处理。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port link-type hybrid
```

```
[SW1-GigabitEthernet0/0/1]port hybrid untagged vlan 10 20
```

```
[SW1-GigabitEthernet0/0/1]port hybrid pvid vlan 20
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port link-type hybrid
```

```
[SW1-GigabitEthernet0/0/2]port hybrid untagged vlan 10 20
```

```
[SW1-GigabitEthernet0/0/2]port hybrid pvid vlan 20
```

对于 SW1 的 GE 0/0/5 端口，配置端口类型为 Hybrid，并要求端口对于收到的Untagged帧添加VLAN 10的标签后进行转发，且在发送属于VLAN 10、VLAN 20、VLAN 30 和VLAN 40 的帧之前进行去标签处理。

```
[SW1]interface GigabitEthernet 0/0/5
```

```
[SW1-GigabitEthernet0/0/5]port link-type hybrid
```

```
[SW1-GigabitEthernet0/0/5]port hybrid untagged vlan 10 20 30 40
```

```
[SW1-GigabitEthernet0/0/5]port hybrid pvid vlan 10
```

对于 SW1 的 GE 0/0/3 端口，配置端口类型为 Hybrid，并要求端口对于收到的Untagged 帧添加VLAN 30 的标签后进行转发，且在发送属于VLAN 10 和VLAN 30 的帧之前进行去标签处理。

```
[SW1]interface GigabitEthernet 0/0/3
```

```
[SW1-GigabitEthernet0/0/3]port link-type hybrid
```

```
[SW1-GigabitEthernet0/0/3]port hybrid untagged vlan 10 30
```

```
[SW1-GigabitEthernet0/0/3]port hybrid pvid vlan 30
```

对于 SW1 的 GE 0/0/4 端口，配置端口类型为 Hybrid，并要求端口对于收到的Untagged 帧添加VLAN 40 的标签后进行转发，且在发送属于VLAN 10 和VLAN 40 的帧之前进行去标签处理。

```
[SW1]interface GigabitEthernet 0/0/4
```

```
[SW1-GigabitEthernet0/0/4]port link-type hybrid
```

```
[SW1-GigabitEthernet0/0/4]port hybrid untagged vlan 10 40
```

```
[SW1-GigabitEthernet0/0/4]port hybrid pvid vlan 40
```

配置完成后，在PC-3上使用ping命令测试与PC-1、PC-4、Server-1的连通性，如图6-33所示。

 figure_0514_0214

图6-33 PC-3与PC-1、PC-4、Server-1之间的连通性测试

从图6-33中可以看到，PC-3只能够与Server-1进行通信。

在PC-1上使用ping命令测试与PC-2、Server-1、PC-3之间的连通性，如图6-34所示。

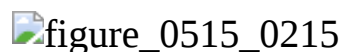


图6-34 PC-1与PC-2、Server-1、PC-3之间的连通性测试

从图6-34中可以看到，PC-1能够与同属部门A的PC-2进行通信，也能够与Server-1进行通信，但无法和PC-3进行通信。至此，网络的需求完全得到了满足。

3．使用Mux VLAN 实现网络需求

使用Hybrid端口的方法虽然能够实现网络需求，但是，为了隔离不同访客间的互相访问，就需要为每一个访客规划一个VLAN，同时SW1上连接Server-1的端口的配置也要进行相应的修改。显然，当访客数量快速增长时，这种方法会变得非常笨拙。下面使用Mux VLAN 方法来解决这个问题。

将VLAN 10 作为Mux VLAN 中的主VLAN，将部门A 所属的VLAN 20 作为互通型从VLAN，将所有访客均划分至VLAN 30，并将VLAN 30 作为隔离型从VLAN。由于 SW1 上加入 Mux VLAN 的端口仅能够允许一个 VLAN 的帧通过，所以需要将加入Mux VLAN 的端口类型修改为Access。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port link-type access
```

```
[SW1-GigabitEthernet0/0/1]port default vlan 20
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port link-type access
```

```
[SW1-GigabitEthernet0/0/2]port default vlan 20
```

```
[SW1-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
```

```
[SW1-GigabitEthernet0/0/3]port link-type access
```

```
[SW1-GigabitEthernet0/0/3]port default vlan 30
```

```
[SW1-GigabitEthernet0/0/3]interface GigabitEthernet 0/0/4
```

```
[SW1-GigabitEthernet0/0/4]port link-type access
```

```
[SW1-GigabitEthernet0/0/4]port default vlan 30
```

```
[SW1-GigabitEthernet0/0/4]interface GigabitEthernet 0/0/5
```

```
[SW1-GigabitEthernet0/0/5]port link-type access
```

```
[SW1-GigabitEthernet0/0/5]port default vlan 10
```

在VLAN 10的视图下设置VLAN 10为主VLAN，设置VLAN 20为互通型从VLAN，设置VLAN 30 为隔离型从VLAN。

```
[SW1]vlan 10
```

```
[SW1-vlan10]mux-vlan
```

```
[SW1-vlan10]subordinate group 20
```

```
[SW1-vlan10]subordinate separate 30
```

然后，在所有加入了Mux VLAN 的端口下使能Mux VLAN 功能。

```
[SW1]interface GigabitEthernet 0/0/1
```

```
[SW1-GigabitEthernet0/0/1]port mux-vlan enable
```

```
[SW1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[SW1-GigabitEthernet0/0/2]port mux-vlan enable
```

```
[SW1-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
```

```
[SW1-GigabitEthernet0/0/3]port mux-vlan enable
```

```
[SW1-GigabitEthernet0/0/3]interface GigabitEthernet 0/0/4
```

```
[SW1-GigabitEthernet0/0/4]port mux-vlan enable
```

```
[SW1-GigabitEthernet0/0/4]interface GigabitEthernet 0/0/5
```

```
[SW1-GigabitEthernet0/0/5]port mux-vlan enable
```

接下来，在PC-3上使用ping命令测试与PC-1、PC-4、Server-1的连通性，如图6-35所示。

figure_0516_0216

图6-35 PC-3与PC-1、PC-4、Server-1之间的连通性测试

从图6-35中可以看到，PC-3无法与部门A的PC-1进行通信，也无法与属于同一个VLAN的访客B（PC-4）进行通信，只能够与属于主VLAN的Server-1进行通信。

在PC-1上使用ping命令测试与PC-2、Server-1、PC-3的连通性，如图6-36所示。

figure_0517_0217

图6-36 PC-1与PC-2、Server-1、PC-3之间的连通性测试

从图6-36中可以看到，PC-1无法与访客A（PC-3）进行通信，但可以与属于同一部门A的PC-2进行通信，也可以与属于主VLAN的Server-1进行通信。至此，网络的需求完全得到了满足。

思考

与使用Hybrid端口的方法相比较，Mux VLAN方法在实现二层隔离时有哪些优点？

6.5 MSTP/RSTP 与STP 的兼容性

原理概述

MSTP（Multiple STP）协议和RSTP（Rapid STP）协议都可以向下兼容STP（Spanning Tree Protocol）协议。运行MSTP/RSTP协议的交换机会根据收到的BPDU版本号信息自动判断与之相连的交换机的运行模式。如果收到的是STP BPDU，MSTP/RSTP交换机就会自动按照STP模式来运行。一个运行在STP模式的交换机在收到MSTP/RSTP的报文后会直接丢弃。

对于运行RSTP/MSTP的交换机，如果某个端口与运行STP的交换机直连，则该端口会自动将其工作模式迁移到STP模式，然后向外发送配置BPDU报文从而保证设备之间的互通。但是在华为的交换机上，如果运行STP的设备被关机或移走，那么MSTP/RSTP交换机的端口无法自动迁移回RSTP/MSTP模式，此时需要在相应的端口上执行Mcheck操作，将端口手动迁移回RSTP/MSTP模式。

实验目的

- 理解MSTP/RSTP与STP的兼容性原理和应用场景

实验内容

实验拓扑如图6-37所示。本实验模拟了一个企业网络场景，公司A的内部网络是由5台交换机组成的局域网，S1、S2、S3和S4组成环形网络，S5通过集线器HUB1与S4相连，所有交换机运行的是RSTP生成树协议。由于公司A跟公司B有业务需要进行合作，公司B的交换机S6通过集线器HUB1与公司A的网络相连，公司B的交换机运行的是STP协议，因此S4和S5会自动降为STP模式。合作期间公司A进行网络优化，所有交换机都运行MSTP协议，但仍然需要能够兼容S6。当两家公司的合作完成之后，S6撤离公司A的网络，S4和S5需要恢复为原来的MSTP模式。

实验拓扑

 figure_0518_0218

图6-37 MSTP/RSTP与STP的兼容性

实验步骤

1．配置RSTP

根据图6-37 进行相应的基本配置，在 S1，S2，S3，S4，S5 上配置生成树模式为RSTP，并配置S1为根交换机。

```
[S1]stp mode rstp
```

```
[S1]stp priority 8192
```

```
[S2]stp mode rstp
```

```
[S3]stp mode rstp
```

```
[S4]stp mode rstp
```

```
[S5]stp mode rstp
```

配置完成后，查看交换机上的生成树模式，此处仅以S1、S4为例。

```
[S1]display stp interface GigabitEthernet 0/0/1
```

```
-----[CIST Global Info][Mode RSTP]-----
```

```
CIST Bridge           :8192 .4c1f-cced-0f0e
```

```
Config Times          :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
```

```
Active Times          :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
```

```
CIST Root/ERPC        :8192 .4c1f-cced-0f0e /0
```

```
CIST RegRoot/IRPC     :8192 .4c1f-cced-0f0e /0
```

```
CIST RootPortId       :0.0
```

.....

```
[S4]display stp interface GigabitEthernet 0/0/2
```

```
-----[CIST Global Info][Mode RSTP]-----
```

```
CIST Bridge                :32768.4c1f-ccc3-18a0
```

```
Config Times                :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
```

```
Active Times                :Hello 2s MaxAge 20s FwDly 15s MaxHop 20
```

```
CIST Root/ERPC              :8192 .4c1f-cced-0f0e / 20001
```

```
CIST RegRoot/IRPC           :32768.4c1f-ccc3-18a0 / 0
```

```
CIST RootPortId             :128.24
```

.....

可以看到，公司A的交换机都运行在RSTP模式下，且S1为根交换机。为了进一步加快收敛速度，配置S3和S4的Ethernet 0/0/1端口为边缘端口。

```
[S3]interface Ethernet 0/0/1
```

```
[S3-Ethernet0/0/1]stp edged-port enable
```

```
[S4]interface Ethernet 0/0/1
```

```
[S4-Ethernet0/0/1]stp edged-port enable
```

2．实现RSTP与STP的兼容

S6通过集线器HUB1接入到公司A的网络。S6运行的是STP协议。配置S6的生成树模式为STP。

```
[S6]stp mode stp
```

在S4 和S5 上查看Ethernet 0/0/2 端口的生成树模式。

```
[S4]display stp interface Ethernet 0/0/2
```

```
-----[CIST Global Info][Mode RSTP]-----
```

```
CIST Bridge           :32768.4c1f-ccc3-18a0
```

```
.....
```

```
Last TC occurred      :Ethernet0/0/2
```

```
----[Port2(Ethernet0/0/2)][FORWARDING]----
```

```
Port Protocol        :Enabled
```

```
.....
```

```
Protection Type      :None
```

```
Port STP Mode        :STP
```

```
Port Protocol Type   :Config=auto /Active=dot1s
```

```
.....
```

```
[S5]display stp interface Ethernet 0/0/2
```

```
-----[CIST Global Info][Mode RSTP]-----
```

```
CIST Bridge          :32768.4c1f-cc06-1497
```

```
.....
```

```
Last TC occurred     :Ethernet0/0/2
```

```
----[Port2(Ethernet0/0/2)][FORWARDING]----
```

```
Port Protocol        :Enabled
```

.....

Protection Type :None

Port STP Mode :STP

Port Protocol Type :Config=auto / Active=dot1s

.....

可以看到，S4和S5全局的生成树模式依然是RSTP，但与S6相连的端口的生成树模式已经变为了STP。

在S6上查看STP生成树信息。

<S6>display stp interface Ethernet 0/0/2

-----[CIST Global Info][Mode STP]-----

CIST Bridge :32768.4c1f-cc0a-349f

Config Times :Hello 2s MaxAge 20s FwDly 15s MaxHop 20

Active Times :Hello 2s MaxAge 20s FwDly 15s MaxHop 20

CIST Root/ERPC :8192 .4c1f-cced-0f0e / 20002

CIST RegRoot/IRPC :32768.4c1f-cc0a-349f / 0

.....

可以看到，S6上生成树模式为STP，且根交换机为S1，所以S6已经加入到整个交换网络的生成树当中，RSTP兼容了STP。

如果公司网络内发生链路故障，比如S4与S2之间的链路Down掉，就会造成生成树端口状态发生迁移。使用命令display stp brief 查看端口状态。


```
[S4]interface GigabitEthernet 0/0/2
```

```
[S4-GigabitEthernet0/0/2]shutdown
```

```
[S4-GigabitEthernet0/0/2]display stp brief
```

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	NONE
0	Ethernet0/0/2	DESI	DISCARDING	NONE
0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE

```
[S4]display stp brief
```

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	NONE
0	Ethernet0/0/2	DESI	LEARNING	NONE
0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE

```
[S4]display stp brief
```

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	NONE
0	Ethernet0/0/2	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE

可以看到，S4 的Ethernet 0/0/1 以及GE 0/0/1 这两个运行RSTP 的端口，使用RSTP的P/A机制由Discarding状态快速进入到了Forwarding状态。而与S6相连，运行STP协议的 Ethernet 0/0/2 端口还停留在Discarding 状态，需要经历 Learning 状态后再到Forwarding状态，所

以当运行STP的交换机加入到RSTP网络中后，会造成生成树网络的收敛时间变慢。

3．实现MSTP与STP的兼容

公司A根据业务需求优化网络，配置所有交换机运行MSTP协议。

```
[S1]stp mode mstp
```

```
[S2]stp mode mstp
```

```
[S3]stp mode mstp
```

```
[S4]stp mode mstp
```

```
[S5]stp mode mstp
```

配置完成后，查看S4、S5与S6相连端口的生成树状态，此处以S4为例。

```
[S4]display stp interface Ethernet 0/0/2
```

```
-----[CIST Global Info][Mode MSTP]-----
```

```
CIST Bridge                               :32768.4c1f-ccc3-18a0
```

```
.....
```

```
Last TC occurred                           :GigabitEthernet0/0/2
```

```
----[Port2(Ethernet0/0/2)][FORWARDING]----
```

```
Port Protocol                             :Enabled
```

```
.....
```

```
Protection Type                           :None
```

```
Port STP Mode                             :STP
```

Port Protocol Type :Config=auto /Active=dot1s

.....

可以看到，S4 的全局生成树模式已经变为MSTP，但是Ethernet 0/0/2端口的生成树模式却是STP，可见MSTP兼容了STP，兼容现象与RSTP兼容STP现象一致，这里不再赘述。

4．交换机端口迁移

公司A与公司B的合作结束后，S6交换机撤离公司A的网络。此时，为了提高网络的运行效率，需要恢复S4和S5的生成树模式为MSTP。

在 S6 上关闭 Ethernet 0/0/2 端口，在 S4 和 S5 上使用命令 display stp interface Ethernet 0/0/2 查看端口协议状态。

```
[S6]interface Ethernet0/0/2
```

```
[S6-Ethernet0/0/2]shutdown
```

```
[S4]display stp interface Ethernet 0/0/2
```

```
-----[CIST Global Info][Mode MSTP]-----
```

```
CIST Bridge :32768.4c1f-ccc3-18a0
```

.....

```
----[Port2(Ethernet0/0/2)][FORWARDING]----
```

```
Port Protocol :Enabled
```

.....

```
Protection Type :None
```

```
Port STP Mode :STP
```

Port Protocol Type :Config=auto /Active=dot1s

.....

[S5]display stp interface Ethernet 0/0/2

-----[CIST Global Info][Mode MSTP]-----

CIST Bridge :32768.4c1f-cc06-1497

.....

----[Port2(Ethernet0/0/2)][FORWARDING]----

Port Protocol :Enabled

.....

Protection Type :None

Port STP Mode :STP

Port Protocol Type :Config=auto /Active=dot1s

.....

可以看到，虽然S6已经撤离了MSTP网络，但是S4和S5的Ethernet 0/0/2端口协议模式仍是STP，无法自动迁移回MSTP模式，造成公司A的生成树网络无法实现快速收敛。此时需要在S4和S5的Ethernet 0/0/2端口使用命令stp mcheck，使端口模式从STP迁移回MSTP。

[S4]interface Ethernet0/0/2

[S4-Ethernet0/0/2]stp mcheck

[S5]interface Ethernet0/0/2

[S5-Ethernet0/0/2]stp mcheck

配置完成后，重新查看S4 和S5 的Ethernet 0/0/2 端口协议模式。

```
[S4]display stp interface Ethernet 0/0/2
```

```
-----[CIST Global Info][Mode MSTP]-----
```

```
CIST Bridge                               :32768.4c1f-ccc3-18a0
```

```
.....
```

```
----[Port2(Ethernet0/0/2)][FORWARDING]----
```

```
Port Protocol                             :Enabled
```

```
.....
```

```
Protection Type                           :None
```

```
Port STP Mode                             :MSTP
```

```
.....
```

```
[S5]display stp interface Ethernet 0/0/2
```

```
-----[CIST Global Info][Mode MSTP]-----
```

```
CIST Bridge                               :32768.4c1f-cc06-1497
```

```
.....
```

```
----[Port2(Ethernet0/0/2)][FORWARDING]----
```

```
Port Protocol                             :Enabled
```

```
.....
```

```
Protection Type                           :None
```

```
Port STP Mode                             :MSTP
```

Port Protocol Type

:Config=auto /Active=dot1s

.....

可以看到，S4 和S5的Ethernet 0/0/2端口协议模式已经恢复为MSTP，提高了公司A的整个交换网络的运行效率。

思考

CIST的英文全称是什么？它的含义是什么？

6.6 MSTP/RSTP 的保护功能

原理概述

在RSTP或MSTP交换网络中，为了防止恶意攻击或临时环路的产生，可配置保护功能来增强网络的健壮性和安全性。

BPDU 保护：在交换设备上，通常将直接与用户终端或文件服务器等非交换设备相连的端口配置为边缘端口，边缘端口一般不会收到BPDU。如果有人伪造BPDU恶意攻击交换设备，边缘端口接收到BPDU后，交换设备会自动将边缘端口设置为非边缘端口，并重新进行生成树计算，从而引起网络震荡。交换设备上启动了BPDU保护功能后，如果边缘端口收到 BPDU，那么边缘端口将被关闭，但是边缘端口属性不变，同时通知网管系统。被关闭的边缘端口只能由网络管理员手动恢复。如果需要被关闭的边缘端口自动恢复，可以配置端口自动恢复功能，并设置延迟时间。

根保护：由于维护人员的错误配置或网络中的恶意攻击，网络中的合法根交换机有可能会收到优先级更高的 BPDU，使得合法根交换机失去根交换机的地位，从而引起网络拓扑结构的错误变动。这种不合法的拓扑变化，可能会导致原来应该通过高速链路的流量被牵引到低速链路上，造成网络拥塞。对于启用了根保护功能的指定端口，其端口角色不能成为根端口。一旦启用根保护功能的指定端口收到优先级更

高的BPDU时，端口将进入Discarding状态，不再转发报文。在经过一段时间（通常为两倍的Forward Delay）后，如果端口一直没有再收到优先级更高的BPDU，端口会自动恢复到正常的Forwarding状态。

环路保护：在运行RSTP或MSTP协议的网络中，根端口和其他阻塞端口的状态是依靠上游交换设备不断发来的BPDU进行维持的。当由于链路拥塞或者单向链路故障导致这些端口收不到来自上游交换设备的BPDU时，交换设备就会重新选择根端口。原先的根端口会转变为指定端口，而原先的阻塞端口会迁移到转发状态，从而造成交换网络中可能产生环路。在启动了环路保护功能后，如果根端口或Alternate端口长时间收不到来自上游的BPDU，则会向网管发送通知信息。如果是根端口则进入Discarding状态，阻塞端口则会一直保持在阻塞状态，不转发报文，从而不会在网络中形成环路。直到根端口或Alternate端口收到BPDU后，端口状态才恢复到Forwarding状态。

防止TC-BPDU攻击：交换设备在接收到TC BPDU报文后，会执行MAC地址表项和ARP表项的删除操作。如果有人伪造TC BPDU报文恶意攻击交换设备，交换设备在短时间内会收到很多TC BPDU报文，频繁的删除操作会给设备造成很大的负担，给网络的稳定性会带来很大隐患。启用防TC-BPDU报文攻击功能后，可以配置交换设备在单位时间内处理TC BPDU报文的次数。如果在单位时间内，交换设备收到的TC BPDU报文数量大于配置的阈值，交换设备只会处理阈值指定的次数。对于其他超出阈值的TCN BPDU报文，定时器到期后设备只对其统一处理一次。这样可以避免频繁地删除MAC地址表项和ARP表项，从而达到保护设备的目的。

实验目的

- 理解MSTP/RSTP 保护功能的工作原理
- 理解MSTP/RSTP 保护功能的应用场景
- 掌握MSTP/RSTP 保护功能的配置方法

实验内容

实验拓扑如图6-38所示，实验编址如表6-5所示。本实验网络中，4台交换机组成了一个环形网络，交换机运行RSTP或MSTP。为了增强网络的健壮性和安全性，需要配置MSTP/RSTP的保护功能。

实验拓扑

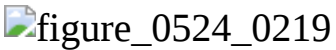
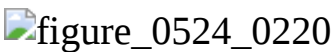
figure_0524_0219

图6-38 MSTP/RSTP的保护功能

实验编址表

表6-5 实验编址

figure_0524_0220

实验步骤

1．配置RSTP/MSTP

根据图6-38 和表 6-5 进行相应的基本配置，在所有交换机上配置生成树模式为RSTP，并配置优先级使S1为主根交换机，S2为备份根交换机。

```
[S1]stp mode rstp
```

```
[S1]stp priority 4096
```

```
[S2]stp mode rstp
```

```
[S2]stp priority 8192
```

```
[S3]stp mode rstp
```

```
[S4]stp mode rstp
```


使用display stp brief 观察交换机的各个端口状态。

<S1>display stp brief

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/1	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/2	DESI	FORWARDING	NONE

<S2>display stp brief

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE
0	GigabitEthernet0/0/2	DESI	FORWARDING	NONE

<S3>display stp brief

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/1	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/2	ROOT	FORWARDING	NONE

<S4>display stp brief

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/1	ALTE	DISCARDING	NONE
0	GigabitEthernet0/0/2	ROOT	FORWARDING	NONE

可以看到，S1 为根交换机，S4 的GE 0/0/1 端口状态为Discarding。

配置S3 和S4 的Ethernet 0/0/1 端口为边缘端口。

```
[S3]interface Ethernet 0/0/1
```

```
[S3-Ethernet0/0/1]stp edged-port enable
```

```
[S4]interface Ethernet 0/0/1
```

```
[S4-Ethernet0/0/1]stp edged-port enable
```

在PC-1上使用ping命令检测与PC-2之间的连通性。

```
<PC-1>ping 10.1.1.2
```

```
Ping 10.1.1.2: 32 data bytes, Press Ctrl_C to break
```

```
From 10.1.1.2: bytes=32 seq=1 ttl=128 time=141 ms
```

```
From 10.1.1.2: bytes=32 seq=2 ttl=128 time=125 ms
```

```
From 10.1.1.2: bytes=32 seq=3 ttl=128 time=125 ms
```

```
From 10.1.1.2: bytes=32 seq=4 ttl=128 time=125 ms
```

```
From 10.1.1.2: bytes=32 seq=5 ttl=128 time=109 ms
```

```
--- 10.0.1.2 ping statistics ---
```

```
5 packet(s) transmitted
```

```
5 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 109/125/141 ms
```

可以看到，PC-1和PC-2正常通信。

由于保护功能在 RSTP和MSTP中完全一样，所以本实验中仅以RSTP为例进行演示。

2．配置BPDU保护

交换机S3和S4为接入交换机，需要接入用户终端，在RSTP中把连接终端的端口配置为边缘端口，并且为了防止边缘端口收到不合法的BPDU 后网络重新收敛，在 S3和S4上配置BPDU保护功能。

在系统视图下使用命令stp bpdu-protection启用交换设备边缘端口的BPDU 保护功能。缺省情况下，交换设备的BPDU保护功能处于禁用状态。

```
[S3]stp bpdu-protection
```

```
[S4]stp bpdu-protection
```

为了演示边缘端口收到BPDU 的效果，把S3 的GE 0/0/2 端口配置为边缘端口。

```
[S3]interface GigabitEthernet 0/0/2
```

```
[S3-GigabitEthernet0/0/2]stp edged-port enable
```

```
Sep 26 2013 15:22:01-08:00 S3 %%01MSTP/4/BPDU_PROTECTION(1)
[4]:This edged-port GigabitEthernet0/0/2 that enabled BPDU-Protection
will be shutdown, because it received BPDU packet!
```

```
Sep 26 2013 15:22:01-08:00 S3 %%01PHY/1/PHY(1)[5]:
GigabitEthernet0/0/2: change status to down
```

可以看到，当配置了BPDU 保护后，S3 的GE 0/0/2 端口接收到了根交换机的BPDU后被关闭，并弹出日志提示。

可以使用命令error-down auto-recovery cause bpdu-protection interval 30 设置端口自动恢复为Up 的延时为30s。当端口被关闭后，删掉GE 0/0/2端口的边缘端口配置，30s后端口会自动Up并弹出日志提示。

[S3]interface GigabitEthernet 0/0/2

[S3-GigabitEthernet0/0/2]undo stp edged-port

[S3-GigabitEthernet0/0/2]undo shutdown

[S3-GigabitEthernet0/0/2]quit

[S3]error-down auto-recovery cause bpdu-protection interval 30

[S3]interface GigabitEthernet 0/0/2

[S3-GigabitEthernet0/0/2]stp edged-port enable

Sep 26 2013 15:40:14-08:00 S3 %%01MSTP/4/BPDU_PROTECTION(1)
[54]:This edged-port GigabitEthernet0/0/2 that enabled BPDU-Protection
will be shutdown, because it received BPDU packet!

Sep 26 2013 15:40:14-08:00 S3
%%01ERRDOWN/4/ERRDOWN_DOWNNOTIFY(1)[55]:Notify interface
to change status to error-down. (InterfaceName=GigabitEthernet0/0/2,
Cause=bpdu-protection)

Sep 26 2013 15:40:14-08:00 S3 ERRDOWN/4/ErrordownOccur:OID
1.3.6.1.4.1.2011.5.25.257.2.1 Error-down occurred. (Ifindex=29,
Ifname=GigabitEthernet0/0/2, Cause=bpdu-protection)

Sep 26 2013 15:40:16-08:00 S3 %%01PHY/1/PHY(1)[56]:
GigabitEthernet0/0/2: change status to down

Sep 26 2013 15:40:21-08:00 S3 DS/4/DATASYNC_CFGCHANGE:OID
1.3.6.1.4.1.2011.5.25.191.3.1 configurations have been changed. The
current change number is 17, the change loop count is 0, and the maximum
number of records is 4095.

Sep 26 2013 15:40:44-08:00 S3
%%01ERRDOWN/4/ERRDOWN_DOWNRECOVER(1)[59]:Notify
interface to recover state from error-down.
(InterfaceName=GigabitEthernet0/0/2)

Sep 26 2013 15:40:44-08:00 S3 ERRDOWN/4/ErrordownRecover:OID 1.3.6.1.4.1.2011.5.25.257.2.2 Error-down recovered. (Ifindex=29, Ifname=GigabitEthernet0/0/2, Cause=bpdu-protection, RecoverType=auto recovery)

Sep 26 2013 15:40:46-08:00 S3 %%01PHY/1/PHY(l)[60]: GigabitEthernet0/0/2: change status to up

可以看到，GE 0/0/2 端口从Down 到Up 所经历的时间是自动恢复的延时时间30s。但由于此时该接口下仍有stp edged-port enable命令，故端口一直处于up 和down 的状态切换。撤销端口下的边缘端口配置。

```
[S3]interface GigabitEthernet 0/0/2
```

```
[S3-GigabitEthernet0/0/2]undo stp edged-port
```

3 . 配置根保护

根保护是指定端口上的特性。当端口的角色是指定端口时，配置根保护功能才能生效。若在其他类型的端口上配置根保护功能，根保护功能不会生效。

使用display stp brief 命令，查看S1、 S2、 S3、 S4 的指定端口。

```
<S1>display stp brief
```

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/1	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/2	DESI	FORWARDING	NONE

```
<S2>display stp brief
```

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE

```
0 GigabitEthernet0/0/2 DESI FORWARDING NONE
```

```
<S3>display stp brief
```

MSTID	Port	Role	STP State	Protection

0	Ethernet0/0/1	DESI	FORWARDING	BPDU

0	GigabitEthernet0/0/1	DESI	FORWARDING	NONE

0	GigabitEthernet0/0/2	ROOT	FORWARDING	NONE

```
<S4>display stp brief
```

MSTID	Port	Role	STP State	Protection

0	Ethernet0/0/1	DESI	FORWARDING	BPDU

0	GigabitEthernet0/0/1	ALTE	DISCARDING	NONE

0	GigabitEthernet0/0/2	ROOT	FORWARDING	NONE

在指定端口上配置根保护。

```
[S1]interface GigabitEthernet 0/0/1
```

```
[S1-GigabitEthernet0/0/1]stp root-protection
```

```
[S1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[S1-GigabitEthernet0/0/2]stp root-protection
```

```
[S2]interface GigabitEthernet 0/0/2
```

```
[S2-GigabitEthernet0/0/2]stp root-protection
```

```
[S3]interface GigabitEthernet 0/0/1
```

```
[S3-GigabitEthernet0/0/1]stp root-protection
```

接下来，修改S4的优先级优于S1。

```
[S4]stp priority 0
```

交换机的优先级的值越小，优先级越大，成为根交换机的可能性也就越大。

在S4 上使用display stp命令查看根交换机信息。

```
<S4>display stp
```

```
-----[CIST Global Info][Mode RSTP]-----
```

```
CIST Bridge           : 4096 .4c1f-cc3d-1546
```

```
Config Times          : Hello 2s MaxAge 20s FwDly 15s MaxHop 20
```

```
Active Times          : Hello 2s MaxAge 20s FwDly 15s MaxHop 20
```

```
CIST Root/ERPC        : 4096 .4c1f-cc3d-1546 / 0
```

```
CIST RegRoot/IRPC     : 4096 .4c1f-cc3d-1546 / 0
```

```
.....
```

可以看到，S4已经认为自己是根桥。

在S2 和S3 上使用display stp命令查看根交换机信息。

```
<S2>display stp
```

```
-----[CIST Global Info][Mode RSTP]-----
```

```
CIST Bridge           : 32768.4c1f-cc92-8651
```

```
Config Times          : Hello 2s MaxAge 20s FwDly 15s  
MaxHop 20
```

Active Times : Hello 2s MaxAge 20s FwDly 15s
MaxHop 20

CIST Root/ERPC : 8192 .4c1f-ccad-2f19 / 1

CIST RegRoot/IRPC : 32768.4c1f-cc92-8651 / 0

.....

<S3>display stp

-----[CIST Global Info][Mode RSTP]-----

CIST Bridge : 32768.4c1f-cc8b-e4c0

Config Times : Hello 2s MaxAge 20s FwDly 15s
MaxHop 20

Active Times : Hello 2s MaxAge 20s FwDly 15s
MaxHop 20

CIST Root/ERPC : 8192 .4c1f-ccad-2f19 / 20000

CIST RegRoot/IRPC : 32768.4c1f-cc8b-e4c0 / 0

.....

可以看到，S2、S3仍然认为S1是根交换机。

在S2 和S3 上使用命令display stp brief 查看端口状态信息。

<S2>display stp brief

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE
0	GigabitEthernet0/0/2	DESI	DISCARDING	ROOT

<S3>display stp brief

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	BPDU
0	GigabitEthernet0/0/1	DESI	DISCARDING	ROOT
0	GigabitEthernet0/0/2	ROOT	FORWARDING	NONE

可以看到，与S4相连的指定端口状态变为了Discarding。

删除S4上的优先级配置。

[S4]undo stp priority

4．配置环路保护

如果由于链路拥塞或者单向链路故障导致根端口收不到来自上游交换设备的BPDU，交换设备会重新选择根端口。原先的根端口会转变为指定端口，而原先的阻塞端口会迁移到转发状态，从而造成交换网络中环路的生产。

在S2 的GE 0/0/2 端口下配置stp bpdu-filter enable。

[S2]interface GigabitEthernet 0/0/2

[S2-GigabitEthernet0/0/2]stp bpdu-filter enable

这样一来，S4由于收不到来自上游的BPDU，就重新选择根端口。观察所有交换机的所有端口的STP信息。

<S1>display stp brief

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/1	DESI	FORWARDING	ROOT

```
0 GigabitEthernet0/0/2 DESI FORWARDING ROOT
```

```
<S2>display stp brief
```

MSTID	Port	Role	STP State	Protection

0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE

0	GigabitEthernet0/0/2	DESI	FORWARDING	ROOT

```
<S3>display stp brief
```

MSTID	Port	Role	STP State	Protection

0	Ethernet0/0/1	DESI	FORWARDING	BPDU

0	GigabitEthernet0/0/1	DESI	FORWARDING	ROOT

0	GigabitEthernet0/0/2	ROOT	FORWARDING	NONE

```
<S4>display stp brief
```

MSTID	Port	Role	STP State	Protection

0	Ethernet0/0/1	DESI	FORWARDING	BPDU

0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE

0	GigabitEthernet0/0/2	DESI	FORWARDING	NONE

可以看到，所有交换机的端口都进入了Forwarding状态。

在PC-1上使用ping命令检测与PC-2的连通性。

```
<PC>ping 10.1.1.2
```

```
Ping 10.1.1.2: 32 data bytes, Press Ctrl_C to break
```

```
Request timeout!
```

Request timeout!

Request timeout!

From 10.1.1.2: bytes=32 seq=4 ttl=128 time=31 ms

Request timeout!

--- 10.1.1.2 ping statistics ---

5 packet(s) transmitted

1 packet(s) received

80.00% packet loss

round-trip min/avg/max = 0/31/31 ms

可以看到，PC-1在成功发送和接收了一个报文后，其他报文都超时了，这是因为网络中已经产生了环路。

恢复S2的GE 0/0/2端口，在S4的GE 0/0/1和GE 0/0/2端口配置环路保护功能，然后在S2的GE 0/0/2端口上配置stp bpdu-filter enable。

```
[S2]interface GigabitEthernet 0/0/2
```

```
[S2-GigabitEthernet0/0/2]undo stp bpdu-filter
```

```
[S4]interface GigabitEthernet 0/0/1
```

```
[S4-GigabitEthernet0/0/1]stp loop-protection
```

```
[S4-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[S4-GigabitEthernet0/0/2]stp loop-protection
```

```
[S2]interface GigabitEthernet 0/0/2
```

```
[S2-GigabitEthernet0/0/2]stp bpdu-filter enable
```

在S4上查看STP的状态信息。

```
<S4>display stp brief
```

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	BPDU
0	GigabitEthernet0/0/1	ROOT	FORWARDING	LOOP
0	GigabitEthernet0/0/2	DESI	DISCARDING	LOOP

可以看到，S4 的GE 0/0/1 端口成为了根端口，GE 0/0/2 端口虽然成为了指定端口，但是处于Discarding状态，不转发数据，这样就避免了环路。

再次在PC-1上使用ping命令测试与PC-2的连通性。

```
<PC>ping 10.1.1.2
```

```
Ping 10.1.1.2: 32 data bytes, Press Ctrl_C to break
```

```
From 10.1.1.2: bytes=32 seq=1 ttl=128 time=31 ms
```

```
From 10.1.1.2: bytes=32 seq=2 ttl=128 time=16 ms
```

```
From 10.1.1.2: bytes=32 seq=3 ttl=128 time=15 ms
```

```
From 10.1.1.2: bytes=32 seq=4 ttl=128 time=32 ms
```

```
From 10.1.1.2: bytes=32 seq=5 ttl=128 time=31 ms
```

```
--- 10.1.1.2 ping statistics ---
```

```
5 packet(s) transmitted
```

```
5 packet(s) received
```

```
0.00% packet loss
```

round-trip min/avg/max = 15/25/32 ms

可以看到，PC-1与PC-2之间可以进行正常通信。

5．配置TC-BPDU保护

启用TC-BPDU保护功能后，可以配置交换设备处理TC类型BPDU报文的最大速度，以避免频繁地删除MAC地址表项和ARP表项，从而达到保护交换设备的目的。单位时间的取值与RSTP的Hello Time一致，可以通过执行命令stp timer hello 进行配置。

使用命令stp tc-protection，开启交换设备对TC类型BPDU报文的保护功能，缺省情况下，交换设备的TC保护功能处于关闭状态。

```
[S1]stp tc-protection
```

```
[S1]stp tc-protection threshold 2
```

```
[S2]stp tc-protection
```

```
[S2]stp tc-protection threshold 2
```

```
[S3]stp tc-protection
```

```
[S3]stp tc-protection threshold 2
```

```
[S4]stp tc-protection
```

```
[S4]stp tc-protection threshold 2
```

上面所进行的配置的含义是：交换设备在单位时间（与RSTP Hello时间间隔一致）内，允许在收到TC-BPDU报文后立即进行地址表项删除操作的最大次数为两次。

思考

配置了根保护的端口，可以再配置环路保护吗？

第7章 MPLS

7.1 MPLS和LDP基本配置

7.2 BGP/MPLS VPN基本配置

7.1 MPLS 和LDP 基本配置

原理概述

MPLS（Multi-Protocol Label Switching，多协议标签交换）技术的出现，极大地推动了互联网的发展和应用。例如，利用MPLS技术，可以有效而灵活地部署VPN（Virtual Private Network，虚拟专用网）、TE（Traffic Engineering，流量工程）和QoS（Quality of Service，服务质量）。目前，MPLS技术主要应用在运营商网络之中。

在MPLS网络中，位于网络边缘的路由器称为LER（Label Edge Router），网络内部的路由器称为LSR（Label Switch Router），MPLS报文经过的路径称为LSP（Label Switched Path）。一条LSP总是起于一台被称为Ingress的LER，止于另一台被称为Egress的LER，中间经过若干台被称为Transit的LSR。LSP具有单向性，且有静态LSP和动态LSP之分，静态LSP需要人工进行固定的标签分配，动态LSP需要利用诸如LDP（Label Distribution Protocol，标签分发协议）这样的协议进行动态标签分配。

实验目的

- 掌握MPLS 和LDP 的基本配置方法
- 观察MPLS 标签转发过程

实验内容

实验拓扑如图7-1所示，实验编址如表7-1所示。本网络全网运行OSPF，并且需要部署MPLS和LDP，创建R1与R3之间的LSP。

实验拓扑

figure_0533_0221

图7-1 MPLS和LDP基本配置

实验编址表

表7-1 实验编址

figure_0533_0222

(续表)

figure_0534_0223

实验步骤

1. 基本配置

根据图7-1和表7-1进行相应的基本配置，并使用ping命令检测R2与R1之间的连通性。

```
<R2>ping -c 1 10.0.12.1
```

```
PING 10.0.12.1: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.1: bytes=56 Sequence=1 ttl=255 time=60 ms
```

```
--- 10.0.12.1 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

0.00% packet loss

round-trip min/avg/max = 60/60/60 ms

R2与R3之间的连通性测试过程在此省略。

2 . 配置OSPF路由协议

在每台路由器上配置OSPF协议。

```
[R1]ospf router-id 10.0.1.1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R2]ospf router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
```

```
[R3]ospf router id 10.0.3.3
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

配置完成后，查看R1的OSPF路由表。

```
[R1]display ospf routing
```


OSPF Process 1 with Router ID 10.0.1.1

Routing Tables

Routing for Network

Destination	Cost	Type	NextHop	AdvRouter	Area
10.0.1.1/32	0	Stub	10.0.1.1	10.0.1.1	0.0.0.0
10.0.12.0/24	1	Transit	10.0.12.1	10.0.1.1	0.0.0.0
10.0.2.2/32	1	Stub	10.0.12.2	10.0.2.2	0.0.0.0
10.0.3.3/32	2	Stub	10.0.12.2	10.0.3.3	0.0.0.0
10.0.23.0/24	2	Transit	10.0.12.2	10.0.2.2	0.0.0.0

Total Nets: 5

Intra Area: 5 Inter Area: 0 ASE: 0 NSSA: 0

可以看到，R1 的OSPF 路由表中存在去往R2的Loopback 0 和R3 的 Loopback 0 的路由，以及去往10.0.23.0/24网段的路由。

3．配置MPLS协议

配置MPLS 协议时，首先需要配置LSR ID。以R1 为例，在系统视图下配置R1 的LSR ID为10.0.1.1。

```
[R1]mpls lsr-id 10.0.1.1
```

然后，在系统视图下，使用mpls命令全局启用MPLS，并进入到MPLS 视图。

```
[R1]mpls
```

Info: Mpls starting, please wait... OK!

在全局启用MPLS之后，还需要在转发MPLS报文的接口上使用mpls命令使能接口的MPLS功能。

```
[R1-mpls]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]mpls
```

配置完成后，在R1 上使用命令display mpls lsp查看LSP 的信息。

```
[R1]display mpls lsp
```

```
[R1]
```

可以看到，R1上现在还不存在任何LSP。

4．配置静态LSP

接下来，手动建立一条从R1到R3的静态LSP。

在R1上配置从R1到R3的静态LSP的Ingress，并进行标签的分配。

```
[R1]static-lsp ingress R1toR3 destination 10.0.3.3 32 nexthop 10.0.12.2  
out-label 102
```

在R2上配置从R1到R3的静态LSP的Transit，并进行标签的分配。

```
[R2]mpls lsr-id 10.0.2.2
```

```
[R2]mpls
```

```
[R2-mpls]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]mpls
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]mpls
```

```
[R2-GigabitEthernet0/0/1]quit
```

```
[R2]static-lsp transit R1toR3 incoming-interface GigabitEthernet 0/0/0 in-label 102 nexthop 10.0.23.3 out-label 203
```

在R3上配置从R1到R3的静态LSP的Egress，并进行标签的分配。

```
[R3]mpls lsr-id 10.0.3.3
```

```
[R3]mpls
```

```
[R3-mpls]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]mpls
```

```
[R3-GigabitEthernet0/0/1]quit
```

```
[R3]static-lsp egress R1toR3 incoming-interface GigabitEthernet0/0/1 in-label 203
```

配置完成后，在R1上查看LSP信息。

```
<R1>display mpls lsp
```

```
-----  
-----  
LSP Information: STATIC LSP  
-----  
-----
```

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.3.3/32	NULL/102	-/GE0/0/0	

可以看到，R1上已经拥有了去往R3（10.0.3.3/32）的静态LSP，且在本地的In标签为NULL，说明R1是该LSP的Ingress。

在R2和R3上也可以查看到同样的信息。

```
[R2]display mpls lsp
```

```
-----  
-----  
LSP Information: STATIC LSP
```

```
-----  
-----  
FEC                In/OutLabel      In/Out IF          Vrf Name  
-/-                102/203      GE0/0/0/GE0/0/1
```

```
<R3>display mpls lsp
```

```
-----  
-----  
LSP Information: STATIC LSP
```

```
-----  
-----  
FEC                In/OutLabel      In/Out IF          Vrf Name  
-/-                203/NULL     GE0/0/1/-
```

在R1 上使用tracert lsp ip 命令验证去往10.0.3.3/32 的MPLS 报文所经过的路径。

```
<R1>tracert lsp ip 10.0.3.3 32
```

```
LSP Trace Route FEC: IPV4 PREFIX 10.0.3.3/32 , press CTRL_C to  
break.
```

```
TTL  Replier      Time    Type  Downstream
```

```
0                               Ingress    10.0.12.2/[102 ]
1 10.0.12.2    250ms  Transit    10.0.23.3/[203 ]
2 10.0.3.3      230ms  Egress
```

从上面的显示信息中可以看到报文在进行MPLS转发过程中使用的标签，以及各路由器在该LSP中的角色。

在R3 上使用tracert lsp ip 命令验证去往10.0.1.1/32 的MPLS 报文所经过的路径。

```
<R3>tracert lsp ip 10.0.1.1 32
```

```
Error: The specified LSP does not exist.
```

可以看到，系统提示LSP并不存在，这也正好说明了LSP具有单向性。

接下来，手动配置从R3去往R1的静态LSP。

```
[R3]static-lsp ingress R3toR1 destination 10.0.1.1 32 nexthop 10.0.23.2
out-label 101
```

```
[R2]static-lsp transit R3toR1 incoming-interface GigabitEthernet 0/0/1 in-
label 101 nexthop 10.0.12.1 out-label 201
```

```
[R1]static-lsp egress R3toR1 incoming-interface GigabitEthernet0/0/0 in-
label 201
```

配置完成后，在R3 上使用tracert lsp ip 命令验证去往10.0.1.1/32的MPLS报文所经过的路径。

```
[R3]tracert lsp ip 10.0.1.1 32
```

```
LSP Trace Route FEC: IPV4 PREFIX 10.0.1.1/32 , press CTRL_C to
break.
```

TTL	Replier	Time	Type	Downstream
0			Ingress	10.0.23.2/[101]
1	10.0.23.2	80ms	Transit	10.0.12.1/[201]
2	10.0.1.1	110ms	Egress	

从上面的显示信息中可以看到报文在进行MPLS转发过程中使用的标签，以及各路由器在该LSP中的角色。

5．利用LDP动态分发标签并建立LSP

首先，在R1、R2、R3上删除之前创建的静态LSP。

```
[R1]undo static-lsp ingress R1toR3
```

```
[R1]undo static-lsp egress R3toR1
```

```
[R2]undo static-lsp transit R1toR3
```

```
[R2]undo static-lsp transit R3toR1
```

```
[R3]undo static-lsp egress R1toR3
```

```
[R3]undo static-lsp ingress R3toR1
```

在R1上使用mpls ldp命令全局启用LDP，然后在接口上使用同样的命令使能LDP。

```
[R1]mpls ldp
```

```
[R1-mpls-ldp]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]mpls ldp
```

在R2、R3上也进行同样的配置。

```
[R2]mpls ldp
```

```
[R2-mpls-ldp]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]mpls ldp
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]mpls ldp
```

```
[R3]mpls ldp
```

```
[R3-mpls-ldp]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]mpls ldp
```

需要注意的是，必须先完成MPLS协议的配置，然后才能够进行上面LDP的配置。

在R1 上使用display mpls ldp interface 命令查看启用了LDP 的接口。

```
[R1]display mpls ldp interface
```

LDP Interface Information in Public Network

Codes:LAM(Label Advertisement Mode), IFName(Interface name)

A '*' before an interface means the entity is being deleted.

```
-----  
-----
```

```
IFNameStatus LAM TransportAddress HelloSent/Rcv
```

```
-----  
-----
```

```
GE0/0/0Active DU 10.0.1.1 17/0
```

```
-----  
-----
```

可以看到，R1 的GE 0/0/0 接口启用了LDP，并且标签分发方式为DU 方式。

在R1、R2、R3 上使用display mpls ldp session命令查看LDP 会话信息。

[R1]display mpls ldp session

LDP Session(s) in Public Network

Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)

A '*' before a session means the session is being deleted.

PeerID	Status	LAM	SsnRole	SsnAge	KA Sent/Rcv
10.0.2.2:0	Operational	DU	Passive	0000:00:04	20/20

TOTAL: 1 session(s) Found.

[R2]display mpls ldp session

LDP Session(s) in Public Network

Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)

A '*' before a session means the session is being deleted.

PeerID	Status	LAM	SsnRole	SsnAge	KASent/Rcv

10.0.1.1:0	Operational	DU	Active	0000:00:19	78/78

10.0.3.3:0	Operational	DU	Passive	0000:00:18	75/75

TOTAL: 2 session(s) Found.

<R3>display mpls ldp session

LDP Session(s) in Public Network

Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)

A '*' before a session means the session is being deleted.

PeerID	Status	LAM	SsnRole	SsnAge	KASent/Rcv

10.0.2.2:0	Operational	DU	Active	0000:00:25	104/105

TOTAL: 1 session(s) Found.

可以看到，R1和R2之间、R2和R3之间的LDP会话状态为Operational，表示会话已成功建立。

在R1、R2、R3 上使用命令display mpls lsp查看LSP 信息。

[R1]display mpls lsp

LSP Information: LDP LSP

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.2.2/32	NULL/3	-/GE0/0/0	
10.0.2.2/32	1024/3	-/GE0/0/0	
10.0.1.1/32	3/NULL	-/-	
10.0.3.3/32	NULL/1025	-/GE0/0/0	
10.0.3.3/32	1025/1025	-/GE0/0/0	

[R2]display mpls lsp

LSP Information: LDP LSP

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.2.2/32	3/NULL	-/-	
10.0.1.1/32	NULL/3	-/GE0/0/0	

10.0.1.1/32 1024/3 -/GE0/0/0

10.0.3.3/32 NULL/3 -/GE0/0/1

10.0.3.3/32 1025/3 -/GE0/0/1

<R3>display mpls lsp

LSP Information: LDP LSP

FEC	In/OutLabel	In/Out IF	Vrf Name
10.0.3.3/32	3/NULL	-/-	
10.0.1.1/32	NULL/1024	-/GE0/0/1	
10.0.1.1/32	1024/1024	-/GE0/0/1	
10.0.2.2/32	NULL/3	-/GE0/0/1	
10.0.2.2/32	1025/3	-/GE0/0/1	

可以看到，LDP为R1去往R3以及R3去往R1均动态地建立了LSP，从R1去往R3方向的标签顺序为NULL/1025、1025/3、3/NULL，从R3去往R1方向的标签顺序为NULL/1024、1024/3、3/NULL。

在R1上验证去往10.0.3.3/32的MPLS报文所经过的路径。

<R1>tracert lsp ip 10.0.3.3 32

LSP Trace Route FEC: IPV4 PREFIX 10.0.3.3/32 ,press CTRL_C to break.

TTL	Replier	Time	Type	Downstream

0			Ingress	10.0.12.2/[1025]
1	10.0.12.2	50ms	Transit	10.0.23.3/[3]
2	10.0.3.3	70ms	Egress	

可以看到，报文在R1上出发时被赋予了标签1025，经过R2时，标签被替换为3。

在R3上验证去往10.0.1.1/32的MPLS报文所经过的路径。

```
<R3>tracert lsp ip 10.0.1.1 32
```

LSP Trace Route FEC: IPV4 PREFIX 10.0.1.1/32 , press CTRL_C to break.

TTL	Replier	Time	Type	Downstream
0			Ingress	10.0.23.2/[1024]
1	10.0.23.2	50ms	Transit	10.0.12.1/[3]
2	10.0.1.1	80ms	Egress	

可以看到，报文在R3上出发时被赋予了标签1024，经过R2时，标签被替换为3。

分别在R1 和R3 上使用ping lsp ip 命令测试连通性。

```
<R1>ping lsp -c 1 ip 10.0.3.3 32
```

LSP PING FEC: IPV4 PREFIX 10.0.3.3/32/: 100 data bytes, press CTRL_C to break

Reply from 10.0.3.3: bytes=100 Sequence=1 time=10 ms

--- FEC: IPV4 PREFIX 10.0.3.3/32 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 10/10/10 ms

<R3>ping lsp -c 1 ip 10.0.1.1 32

LSP PING FEC: IPV4 PREFIX 10.0.1.1/32/: 100 data bytes, press
CTRL_C to break

Reply from 10.0.1.1: bytes=100 Sequence=1 time=20 ms

--- FEC: IPV4 PREFIX 10.0.1.1/32 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 20/20/20 ms

可以看到，R1与R3之间可以通过MPLS的LSP进行报文的转发。

思考

为什么标签转发比IP转发的效率更高？

7.2 BGP/MPLS VPN 基本配置

原理概述

BGP/MPLS VPN 有时也简称为MPLS L3 VPN，它是MPLS 最为广泛的应用之一。BGP/MPLS VPN 主要部署在运营商网络中。

在BGP/MPLS VPN 网络中，路由器被分为3 类：PE 路由器（ Provider Edge Router ）、 P 路由器（ Provider Router ）和CE 路由器（ Customer Edge Router ）。 P 路由器为BGP/MPLS VPN网络内部的路由器，通常只需要运行IGP、 MPLS和LDP。 PE路由器为BGP/MPLS VPN网络的边缘路由器，用于连接客户的CE设备，通常需要运行MP-BGP（ Muti-Protocol BGP ）、 IGP、 MPLS 和LDP，并为不同的VPN 客户配置VPN 实例（ VPN Instance ）。 CE为客户的边缘设备，用于连接PE，其上仅需要配置PE-CE连通性。

传统的BGP只能维护单一路由表的路由信息，无法为地址重叠的不同客户直接提供服务，同时也难以对不同客户的数据实施隔离，所以在BGP/MPLS VPN 中使用了MP-BGP，它可以通过使用 VPNv4 地址族来区分不同客户的网络层地址信息，并使用VPN实例区分不同VPN客户的路由及流量。

在BGP/MPLS VPN 中，每个VPN 实例为相应的VPN 客户单独维护了一张路由和转发表，称为VRF（ VPN Routing and Forwarding Table ），不同的VPN 实例间的路由是不能够互通的。在PE上，通过将连接CE的接口绑定至VPN实例，就可以区分不同VPN客户的路由。当PE将VPN路由传递至对端PE后，对端PE将使用VPN实例的RD（ Route Distinguisher ）与VPN Target属性来区分VPN路由并将其分配至对应的VPN实例。

在BGP/MPLS VPN 中，BGP 扩展团体属性VPN Target 用来控制VPN 路由的发布和接受。对于一个VPN 实例，其Export Target 与Import Target 相互对应。一般情况下，对端PE上VPN实例的Export Target应与本地Import Target相同；本地VPN实例的Export Target 应与对端PE的Import Target 相同。

通常，在BGP/MPLS VPN 中， P 路由器无需运行BGP，也无需知道关于VPN 的任何信息。 PE上的MP-BGP会为VPN路由分配相应的标签值（ VPN标签 ），作为内层标签， LDP分配的标签会作为外层标签。当VPN流量沿LSP经过P路由器时， P路由器只会进行外层标签的交换。

当流量抵达对端PE时，对端PE会根据内层标签判断出流量所属的VPN。

在BGP/MPLS VPN中，PE-CE连通性的方式决定了客户如何使自己的路由进入VPN实例。通常，可以使用BGP在CE与PE间建立EBGP连接来实现PE-CE的连通，也可以使用静态路由方式或其他动态路由协议来实现这一目的。

实验目的

- 理解PE、P 和CE 设备的作用与区别
- 掌握MPLS/BGP VPN 的基本配置方法

实验内容

实验拓扑如图7-2 所示，实验编址如表7-2 所示。本实验中，假定AS 100 为运营商网络，使用OSPF 作为IGP，并运行BGP/MPLS VPN。R4 和R6 为公司A 的两个站点的CE 路由器，且使用Loopback 0 接口来模拟各自内部的网络，R4 和R6 的PE-CE连通性均使用 BGP 来实现。R5 和 R7 为公司 B 的两个站点的 CE 路由器，且使用Loopback 0 接口来模拟各自内部的网络，R5 的 PE-CE 连通性使用了静态路由方式，R7 的 PE-CE 连通性使用了 OSPF 协议来实现。网络的最终需求是: 公司 A 和公司 B 分别属于不同的 VPN，同一公司内部的网络可以相互通信，不同公司的网络之间不能通信。

实验拓扑

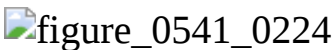


图7-2 BGP/MPLS VPN基本配置

实验编址表

表7-2 实验编址

figure_0541_0225

(续表)

figure_0542_0226

实验步骤

1．基本配置

根据图7-2和表7-2进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=200 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 200/200/200 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置运营商网络的OSPF路由协议

在AS 100 内配置OSPF 协议作为IGP，各路由器均属于区域0，且使用Loopback 0接口IP地址作为Router-ID。

```
[R1]ospf router-id 10.0.1.1
```



```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
```

```
[R2]ospf router-id 10.0.2.2
```

```
[R2-ospf-1]area 0
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.12.0 0.0.0.255
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
```

```
[R2-ospf-1-area-0.0.0.0]network 10.0.2.2 0.0.0.0
```

```
[R3]ospf router-id 10.0.3.3
```

```
[R3-ospf-1]area 0
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
```

```
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

配置完成后，在R2上查看OSPF邻居建立情况。

```
[R2]display ospf peer brief
```

OSPF Process 1 with Router ID 10.0.2.2

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	10.0.1.1	Full

0.0.0.0	GigabitEthernet0/0/1	10.0.3.3	Full

可以看到，R2与R1及R3的OSPF邻居状态为Full，表明邻居关系已成功建立。

3．配置运营商网络的MPLS协议与LDP

在 AS 100 内配置 MPLS 协议与 LDP，各路由器使用 Loopback 0 接口地址作为LSR-ID。

```
[R1]mpls lsr-id 10.0.1.1
```

```
[R1]mpls
```

```
[R1-mpls]mpls ldp
```

```
[R1-mpls-ldp]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]mpls
```

```
[R1-GigabitEthernet0/0/0]mpls ldp
```

```
[R2]mpls lsr-id 10.0.2.2
```

```
[R2]mpls
```

```
[R2-mpls]mpls ldp
```

```
[R2-mpls-ldp]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]mpls
```

```
[R2-GigabitEthernet0/0/0]mpls ldp
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]mpls
```

```
[R2-GigabitEthernet0/0/1]mpls ldp
```

```
[R3]mpls lsr-id 10.0.3.3
```

```
[R3]mpls
```

```
[R3-mpls]mpls ldp
```

```
[R3-mpls-ldp]interface GigabitEthernet 0/0/1
```

```
[R3-GigabitEthernet0/0/1]mpls
```

```
[R3-GigabitEthernet0/0/1]mpls ldp
```

配置完成后，在R2上查看LDP会话建立情况。

```
[R2]display mpls ldp session
```

LDP Session(s) in Public Network

Codes: LAM(Label Advertisement Mode), SsnAge Unit(DDDD:HH:MM)

A '*' before a session means the session is being deleted.

```
-----  
-----  
PeerID      Status  LAM   SsnRole  SsnAge  KASent/Rcv
```

```
-----  
-----  
10.0.1.1:0   Operational  DU    Active   0000:00:01  8/8
```

```
10.0.3.3:0   Operational  DU    Passive  0000:00:00  4/4
```


TOTAL: 2 session(s) Found.

可以看到，LDP会话状态为Operational，会话成功建立。

在R1上查看LSP信息。

[R1]display mpls lsp

LSP Information: LDP LSP

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.2.2/32	NULL/3	-/GE0/0/0	
10.0.2.2/32	1024/3	-/GE0/0/0	
10.0.1.1/32	3/NULL	-/-	
10.0.3.3/32	NULL/1025	-/GE0/0/0	
10.0.3.3/32	1025/1025	-/GE0/0/0	

可以看到，MPLS网络已经为R1，R2和R3的Loopback接口路由建立了相应的LSP。

4．配置PE设备间的MP-BGP

首先，在R1上建立R1与R3的IBGP邻居关系。

[R1]bgp 100

```
[R1-bgp]peer 10.0.3.3 as-number 100
```

```
[R1-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```

```
[R1-bgp]peer 10.0.3.3 next-hop-local
```

然后，使用ipv4-family vpnv4 命令进入VPNv4 视图。

```
[R1-bgp]ipv4-family vpnv4
```

```
[R1-bgp-af-vpnv4]
```

接下来，在VPNv4视图下启用与对等体交换VPNv4路由信息的能力。

```
[R1-bgp-af-vpnv4]peer 10.0.3.3 enable
```

最后，允许与对等体交换路由信息时携带BGP团体属性。

```
[R1-bgp-af-vpnv4]peer 10.0.3.3 advertise-community
```

在R3上完成同样的配置。

```
[R3]bgp 100
```

```
[R3-bgp]peer 10.0.1.1 as-number 100
```

```
[R3-bgp]peer 10.0.1.1 connect-interface LoopBack 0
```

```
[R3-bgp]peer 10.0.1.1 next-hop-local
```

```
[R3-bgp]ipv4-family vpnv4
```

```
[R3-bgp-af-vpnv4]peer 10.0.1.1 enable
```

```
[R3-bgp-af-vpnv4]peer 10.0.1.1 advertise-community
```

配置完成后，在R1上查看BGP邻居关系。

```
[R1]display bgp peer
```

BGP local router ID : 10.0.1.1

Local AS number : 100

Total number of peers : 1 Peers in established state : 1

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
10.0.3.3	4	100	4	6	0	00:02:03	Established	0

可以看到，R1与R3之间的BGP邻居状态为Established，表明BGP邻居关系已成功建立。

5．在PE上创建VPN实例并与接口进行绑定

首先，在R1 的系统视图下使用ip vpn-instance vpna 命令为公司A 创建名为vpna的VPN实例，并进入实例视图。然后，在实例视图下使用ipv4-family命令启用VPN实例的 IPv4 地址族，并进入 IPv4 地址族的视图。接下来，在 IPv4 地址族视图下使用route-distinguisher 300:1命令配置RD 为300:1。最后，使用vpn-target 100:1 both命令配置Import与Export方向的VPN-Target团体属性。

```
[R1]ip vpn-instance vpna
```

```
[R1-vpn-instance-vpna]ipv4-family
```

```
[R1-vpn-instance-vpna-af-ipv4]route-distinguisher 300:1
```

```
[R1-vpn-instance-vpna-af-ipv4]vpn-target 100:1 both
```

在R1连接公司A的GE 0/0/1接口的接口视图下，使用ip binding vpn-instance vpna命令将GE 0/0/1 接口与VPN 实例vpna 进行绑定。

```
[R1]interface GigabitEthernet 0/0/1
```

```
[R1-GigabitEthernet0/0/1]ip binding vpn-instance vpna
```

Info: All IPv4 related configurations on this interface are removed!

Info: All IPv6 related configurations on this interface are removed!

注意，绑定后接口的IP地址等信息将被删除，需要重新配置。

```
[R1-GigabitEthernet0/0/1]ip add 10.0.14.1 255.255.255.0
```

同样，在R1上为公司B创建名为vpnb的VPN实例，RD为300:2，VPN-Target为100:2，绑定接口为GE 0/0/2。

```
[R1]ip vpn-instance vpb
```

```
[R1-vpn-instance-vpb]ipv4-family
```

```
[R1-vpn-instance-vpb-af-ipv4]route-distinguisher 300:2
```

```
[R1-vpn-instance-vpb-af-ipv4]vpn-target 100:2 both
```

```
[R1-vpn-instance-vpb-af-ipv4]interface GigabitEthernet 0/0/2
```

```
[R1-GigabitEthernet0/0/2]ip binding vpn-instance vpb
```

```
[R1-GigabitEthernet0/0/2]ip add 10.0.15.1 255.255.255.0
```

在R3上也完成相应的配置。

```
[R3]ip vpn-instance vpa
```

```
[R3-vpn-instance-vpa]ipv4-family
```

```
[R3-vpn-instance-vpa-af-ipv4]route-distinguisher 300:1
```

```
[R3-vpn-instance-vpa-af-ipv4]vpn-target 100:1 both
```

```
[R3-vpn-instance-vpa-af-ipv4]ip vpn-instance vpb
```

```
[R3-vpn-instance-vpb]ipv4-family
```

```
[R3-vpn-instance-vpnb-af-ipv4]route-distinguisher 300:2
[R3-vpn-instance-vpnb-af-ipv4]vpn-target 100:2 both
[R3-vpn-instance-vpnb-af-ipv4]interface GigabitEthernet 0/0/0
[R3-GigabitEthernet0/0/0]ip binding vpn-instance vpna
[R3-GigabitEthernet0/0/0]ip add 10.0.36.3 255.255.255.0
[R3-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/2
[R3-GigabitEthernet0/0/2]ip binding vpn-instance vpnb
[R3-GigabitEthernet0/0/2]ip add 10.0.37.3 255.255.255.0
```

6 . 为公司A配置基于BGP的PE-CE连通性

在公司A的CE设备R4上进行BGP配置，建立与PE设备R1的EBGP邻居关系。

```
[R4]bgp 10
[R4-bgp]peer 10.0.14.1 as-number 100
[R4-bgp]network 10.0.4.4 32
```

在PE 设备R1 的BGP 视图下使用ipv4-family vpn-instance vpna 命令进入VPN 实例vpna的视图，然后与R4建立BGP邻居关系。

```
[R1]bgp 100
[R1-bgp]ipv4-family vpn-instance vpna
[R1-bgp-vpna]peer 10.0.14.4 as-number 10
```

配置完成后，在R4上查看BGP邻居状态。

```
[R4]display bgp peer
```


BGP local router ID : 10.0.14.4

Local AS number : 10

Total number of peers : 1 Peers in established state : 1

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
10.0.14.1	4	100	2	4	0	00:00:44	Established	0

可以看到，R4与10.0.14.1的BGP邻居状态为Established，表明邻居关系已成功建立。

在R1 上使用display bgp peer 命令查看BGP 邻居状态。

[R1]display bgp peer

BGP local router ID : 10.0.1.1

Local AS number : 100

Total number of peers : 1 Peers in established state : 1

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
10.0.3.3	4	100	36	36	0	00:31:50	Established	0

可以看到，在R1上使用display bgp peer命令并不能查看到与10.0.14.4的邻居信息，正确的方法是使用命令display bgp vpnv4 vpn-instance vpna peer 查看VPN 实例vpna的BGP邻居状态。

[R1]display bgp vpnv4 vpn-instance vpna peer

BGP local router ID : 10.0.1.1

Local AS number : 100

VPN-Instance vpna, Router ID 10.0.1.1:

Total number of peers : 1 Peers in established state : 1

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down	State	PrefRcv
10.0.14.4	4	10	6	5	0	00:03:29	Established	1

可以看到，R1与10.0.14.4的邻居状态为Established，表明BGP邻居关系已成功建立。

在R3和R6上完成同样的配置。

```
[R3]bgp 100
```

```
[R3-bgp]ipv4-family vpn-instance vpna
```

```
[R3-bgp-vpna]peer 10.0.36.6 as-number 20
```

```
[R6]bgp 20
```

```
[R6-bgp]peer 10.0.36.3 as-number 100
```

```
[R6-bgp]network 10.0.6.6 32
```

在R1上使用display bgp vpnv4 vpn-instance vpna routing-table命令查看VPN实例vpna的BGP路由表。

```
<R1>display bgp vpnv4 vpn-instance vpna routing-table
```

BGP Local router ID is 10.0.1.1

Status codes: * - valid,> - best,d - damped,

h - history, i - internal, s - suppressed,S -Stale

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

VPN-Instance vpna,Router ID10.0.1.1:

Total Number of Routes: 2

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*> 10.0.4.4/32	10.0.14.4	0		0	10 i
*>i 10.0.6.6/32	10.0.3.3	0	100	0	20 i

可以看到，VPN实例vpna仅仅拥有10.0.4.4/32和10.0.6.6/32的路由。

在R1 上使用display mpls lsp命令查看LSP 信息。

<R1>display mpls lsp

LSP Information: BGP LSP

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.4.4/32	1026/NULL	-/-	vpna

LSP Information: LDP LSP

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.2.2/32	NULL/3	-/GE0/0/0	

10.0.2.2/32	1024/3	-/GE0/0/0
10.0.3.3/32	NULL/1025	-/GE0/0/0
10.0.3.3/32	1025/1025	-/GE0/0/0
10.0.1.1/32	3/NULL	-/-

可以看到，表中出现了BGP LSP 的信息，FEC为10.0.4.4/32，In 标签为1026，Out标签为NULL，VRF Name 为vpna。In 标签1026应该是由MP-BGP 协议分配的内层标签，仅用于区分路由信息所属的VRF。

在R3上查看LSP信息。

[R3]display mpls lsp

```

-----
-----

```

LSP Information: BGP LSP

```

-----
-----

```

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.6.6/3	1026/NULL	-/-	vpna

```

-----
-----

```

LSP Information: LDP LSP

```

-----
-----

```

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.1.1/32	NULL/1024	-/GE0/0/1	

10.0.1.1/32	1024/1024	-/GE0/0/1
10.0.2.2/32	NULL/3	-/GE0/0/1
10.0.2.2/32	1025/3	-/GE0/0/1
10.0.3.3/32	3/NULL	-/-

可以看到，对于FEC 10.0.6.6/32，MP-BGP 分配的In 标签为1026。

在R4上以10.0.4.4为源，使用ping命令测试与10.0.6.6的连通性。

```
<R4>ping -c 1 -a 10.0.4.4 10.0.6.6
```

```
PING 10.0.6.6: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.6.6: bytes=56 Sequence=1 ttl=252 time=50 ms
```

```
--- 10.0.6.6 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 50/50/50 ms
```

可以看到，R4与R6能够正常通信，实现了公司A的VPN网络的互联互通。

7. 为公司B配置基于静态路由及OSPF协议的PE-CE连通性

根据需求，公司B的CE设备R5将使用静态路由方式实现PE-CE连通性；CE设备R7将使用OSPF协议实现PE-CE连通性。

在R5上创建缺省路由。

```
[R5]ip route-static 0.0.0.0 0 10.0.15.1
```

在R1上为VPN实例vpnb创建静态路由。

```
[R1]ip route-static vpn-instance vpng 10.0.5.5 32 10.0.15.5
```

接下来，在R1的BGP视图下使用ipv4-family vpn-instance vpng命令进入VPN实例vpnb的视图，然后将VPN实例vpnb的静态路由引入BGP。

```
[R1]bgp 100
```

```
[R1-bgp]ipv4-family vpn-instance vpng
```

```
[R1-bgp-vpng]import-route static
```

至此，R5与R1之间的PE-CE连通性配置完成。

在R7上进行普通的OSPF配置。

```
[R7]ospf 2 router-id 10.0.7.7
```

```
[R7-ospf-2]area 0
```

```
[R7-ospf-2-area-0.0.0.0]network 10.0.37.0 0.0.0.255
```

```
[R7-ospf-2-area-0.0.0.0]network 10.0.7.7 0.0.0.0
```

在R3上为VPN实例vpnb创建OSPF进程。

```
[R3]ospf 2 vpn-instance vpng
```

```
[R3-ospf-2]area 0
```

```
[R3-ospf-2-area-0.0.0.0]network 10.0.37.0 0.0.0.255
```

配置完成后，在R3上查看OSPF邻居状态。

```
[R3]display ospf peer brief
```

OSPF Process 1 with Router ID 10.0.3.3

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	10.0.2.2	Full

OSPF Process 2 with Router ID 10.0.37.3

Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/2	10.0.7.7	Full

可以看到，R3与R7的邻居状态为Full，表明邻居关系已成功建立。

在R3的OSPF 视图下使用import-route bgp 命令将VPN 实例vpnb的BGP路由引入OSPF。

```
[R3]ospf 2
```

```
[R3-ospf-2]import-route bgp
```

在R3 的BGP 视图下使用ipv4-family vpn-instance vbnb 命令进入VPN 实例vpnb的视图，然后将VPN实例vpnb的OSPF路由引入BGP。

```
[R3]bgp 100
```

```
[R3-bgp]ipv4-family vpn-instance vpnb
```

```
[R3-bgp-vpnb]import-route ospf 2
```

至此，R7与R3之间的PE-CE连通性配置完成。

在R3上使用display bgp vpnv4 vpn-instance vpnb routing-table命令查看VPN实例vpnb的BGP路由表。

```
[R3]display bgp vpnv4 vpn-instance vpnb routing-table
```

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

VPN-Instance vpnb, Router ID 10.0.3.3:

Total Number of Routes: 3

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.5.5/32	10.0.1.1	0	100	0	?
*>	10.0.7.7/32	0.0.0.0	2		0	?
*>	10.0.37.0/24	0.0.0.0	0		0	?

可以看到，R3的BGP路由表中不但拥有10.0.5.5/32和10.0.7.7/32的路由信息，而且还拥有10.0.37.0/24的路由信息。与使用BGP协议或静态路由方式来提供PE-CE连通性不同，使用OSPF协议实现PE-CE的连通时，PE与CE之间的链路在引入路由时也会被引入进BGP。如果需要避免这种情况，需要在将OSPF路由引入BGP时进行过滤，举例如下。


```
[R3]ip ip-prefix 1 deny 10.0.37.0 24
[R3]ip ip-prefix 1 permit 0.0.0.0 32
[R3]route-policy 10 permit node 10
[R3-route-policy]if-match ip-prefix 1
[R3-route-policy]bgp 100
[R3-bgp]ipv4-family vpn-instance vpnb
[R3-bgp-vpn]import-route ospf 2 route-policy 10
```

在R3 上使用display mpls lsp命令查看LSP。

```
[R3]display mpls lsp
```

```
-----
-----
LSP Information: BGP LSP
```

```
-----
-----
FEC                In/Out Label      In/Out IF      Vrf Name
10.0.6.6/32        1026/NULL        -/-            vpna
10.0.7.7/32        1027/NULL        -/-            vpnb
```

```
-----
-----
LSP Information: LDP LSP
```

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.1.1/32	NULL/1024	-/GE0/0/1	
10.0.1.1/32	1024/1024	-/GE0/0/1	
10.0.2.2/32	NULL/3	-/GE0/0/1	
10.0.2.2/32	1025/3	-/GE0/0/1	
10.0.3.3/32	3/NULL	-/-	

可以看到，过滤之后10.0.37.0/24的路由不再有相应的LSP信息。为不影响后面的实验效果，请读者恢复过滤前的配置。

在R3 上使用display mpls lsp命令查看LSP。

[R3]display mpls lsp

LSP Information: BGP LSP

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.6.6/32	1026/NULL	-/-	vpna
10.0.37.0/24	1027/NULL	-/-	vpnb
10.0.7.7/32	1028/NULL	-/-	vpnb

LSP Information: LDP LSP

FEC	In/Out Label	In/Out IF	Vrf Name
10.0.1.1/32	NULL/1024	-/GE0/0/1	
10.0.1.1/32	1024/1024	-/GE0/0/1	
10.0.2.2/32	NULL/3	-/GE0/0/1	
10.0.2.2/32	1025/3	-/GE0/0/1	
10.0.3.3/32	3/NULL	-/-	

可以看到，MP-BGP协议为10.0.7.7/32与10.0.37.0/24均分配了标签。

在R7上以10.0.7.7为源，使用ping命令测试与10.0.4.4/32、10.0.5.5/32、10.0.6.6/32之间的连通性。

```
<R7>ping -c 1 -a 10.0.7.7 10.0.4.4
```

```
PING 10.0.4.4: 56 data bytes, press CTRL_C to break
```

```
Request time out
```

```
--- 10.0.4.4 ping statistics ---
```

```
1 packet(s) transmitted
```

```
0 packet(s) received
```

```
100.00% packet loss
```

```
<R7>ping -c 1 -a 10.0.7.7 10.0.5.5
```

```
PING 10.0.5.5: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.5.5: bytes=56 Sequence=1 ttl=252 time=70 ms
```

--- 10.0.5.5 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 70/70/70 ms

<R7>ping -c 1 -a 10.0.7.7 10.0.6.6

PING 10.0.6.6: 56 data bytes, press CTRL_C to break

Request time out

--- 10.0.6.6 ping statistics ---

1 packet(s) transmitted

0 packet(s) received

100.00% packet loss

可以看到，R7仅能够与同属公司B的10.0.5.5/32进行通信，而不能与属于公司A的10.0.4.4/32和10.0.6.6/32进行通信。

当CE-PE之间运行EBGP时，无需在PE上对客户路由和MP-BGP协议之间进行引入配置，客户的VPNv4路由可以直接通过MPLS/MP-BGP网络传递给对端PE。而当CE-PE之间运行的是静态路由或者是IGP时，则需要进行互相引入的配置，才能使客户的VPN4的路由通过MPLS/MP-BGP网络进行传递。

思考

RD (Route Distinguisher) 和RT (Route Target) 的作用是什么？

第8章 其他

8.1 配置AAA

8.2 配置BFD

8.3 综合实验1

8.4 综合实验2

8.1 配置AAA

原理概述

AAA 是 Authentication（认证）、Authorization（授权）和 Accounting（计费）的简称，它是网络安全的一种管理机制，可以提供认证、授权和计费功能。在 AAA 中，认证是指对网络用户身份的真实性进行检查和验证；授权是指确定通过认证的网络用户可以有权获得哪些网络资源和服务；计费是指记录网络用户对网络资源和服务的使用情况，并根据相关条件进行费用计算。

AAA采用了客户端/服务器模式。AAA客户端就是通常所说的网络接入服务器，它实质上可以是任何一台使能了 AAA 客户端功能的网络设备（例如交换机或路由器）。AAA服务器是专门用来进行认证、授权和计费的服务器，它一般配置在专用的高性能主机上，但在有的情况下也可以在诸如交换机和路由器这样的网络设备上配置 AAA 服务器中的认证和授权功能。

华为设备的 AAA 功能支持的认证方式包括 3 种。第一种：不认证，也就是完全信任任何用户，不对其身份进行检查和验证。第二种：本地认证，也就是将用户信息（用户名、密码等）配置在网络接入服务器上。第三种：远程认证，也就是将用户信息（用户名、密码等）配置在 AAA 服务器上。华为设备的 AAA 功能支持在 AAA 客户端和 AAA 服务器之间运行 RADIUS（Remote Authentication Dial In User Service）协议或 HWTACACS（HuaWei Terminal Access Controller Access Control System）协议。

华为设备的 AAA 功能支持的授权方式有 5 种，支持的计费方式有 3 种。关于这些授权和计费方式的具体描述，读者可以自行查阅华为相关的产品资料。

实验目的

- 理解 AAA 的基本原理和应用场景
- 掌握 AAA 本地认证和本地授权的配置方法

实验内容

实验拓扑如图 8-1 所示，实验编址如表 8-1 所示。本实验模拟了一个简单的企业网络场景，路由器 R1 是公司网络的出口网关，员工的终端电脑通过交换机 S1 与 R1 相连，且希望通过 R1 访问外网地址 10.0.1.1（注意，现实中的外网地址一般都是公网地址，这里使用的 10.0.1.1 是一个私网地址，仅作演示之用）。为了提高网络安全性，需要将 R1 作为本地 AAA 服务器，员工只有在通过认证之后才能访问外网地址 10.0.1.1。公司后来进行网络升级，购置了一台单独的服务器作为 AAA 服务器，AAA 服务器与作为客户端的 R1 之间运行 RADIUS 协议，原来的本地 AAA 认证方式需要修改为远程 AAA 认证方式。

实验拓扑

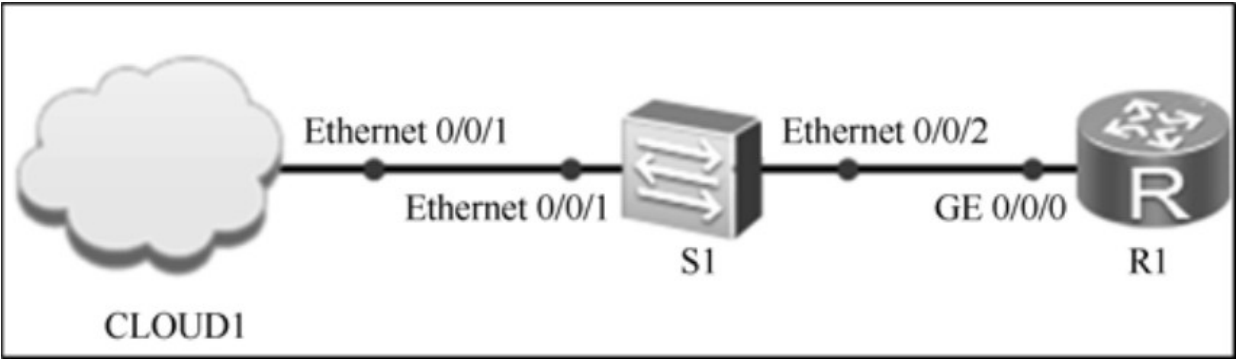


图8-1 配置AAA

实验编址表

表8-1 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
CLOUD1	Ethernet 0/0/1	10.0.12.2	255.255.255.0	N/A

实验步骤

1．基本配置

根据图8-1和表8-1进行相应的基本配置。在本地PC上添加一块Microsoft Loopback Adapter的网卡，并配置静态IP地址为10.0.12.2/24。使用CLOUD1桥接此网卡，桥接时在CLOUD1 新建两个端口，一个绑定到Microsoft Loopback Adapter网卡，另一个绑定到UDP端口，然后在端口映射设置中将两个端口进行映射，并设置为双向通道。设置过程如图8-2所示。

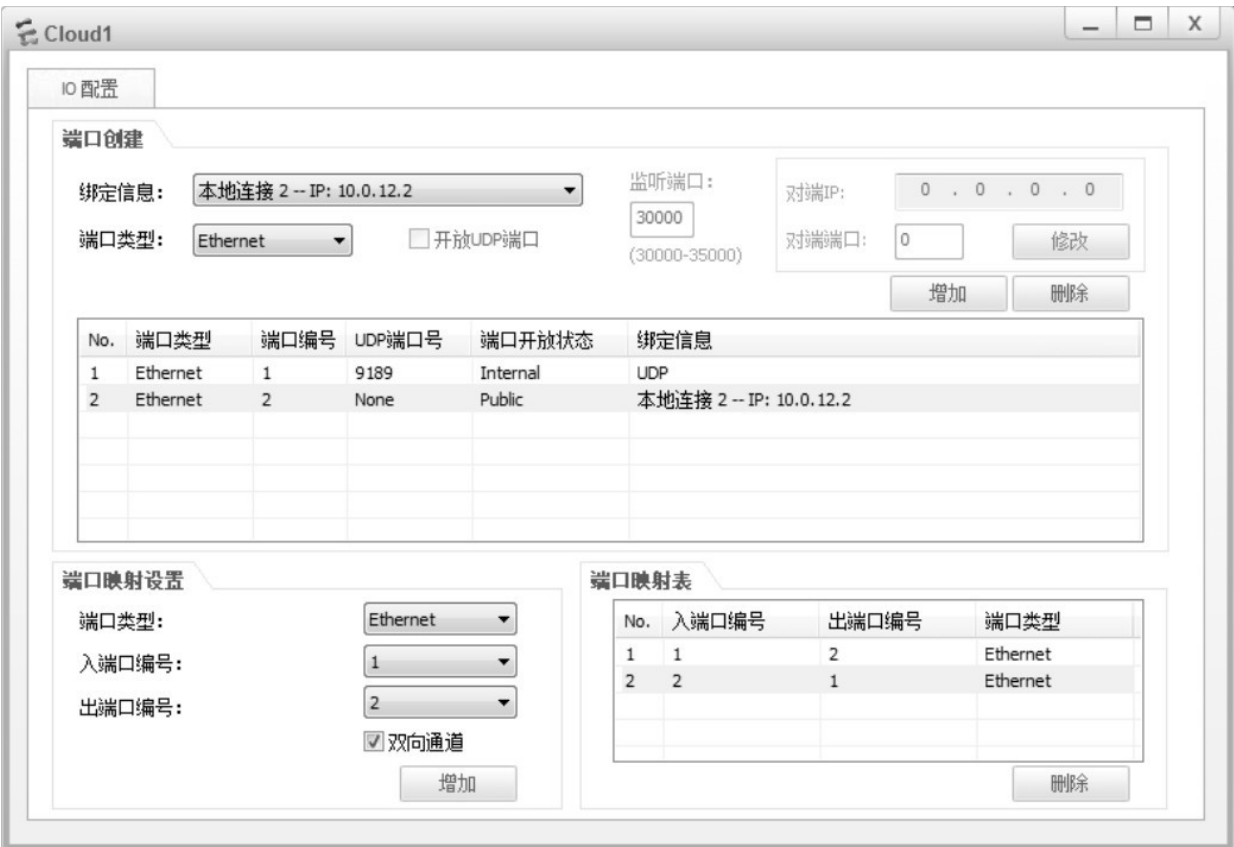


图8-2 配置CLOUD 1

配置完成后，关闭本地PC 上除Microsoft Loopback Adapter之外的所有网卡，然后在本地PC的命令行窗口中进行ping操作。

```
C:\Users\admin>ping 10.0.12.1
```

正在 Ping 10.0.12.1 具有32 字节的数据:

来自 10.0.12.1 的回复:字节=32 时间=43ms TTL=255

来自 10.0.12.1 的回复:字节=32 时间=28ms TTL=255

来自 10.0.12.1 的回复:字节=32 时间=25ms TTL=255

来自 10.0.12.1 的回复:字节=32 时间=23ms TTL=255

10.0.12.1 的 Ping 统计信息:

数据包:已发送= 4，已接收= 4，丢失= 0 (0%丢失)，往返行程的估计时间(以毫秒为单位):

最短= 23ms，最长= 43ms，平均= 29ms


```
C:\Users\admin>ping 10.0.1.1
```

正在 Ping 10.0.1.1 具有 32 字节的数据:

请求超时。

请求超时。

请求超时。

请求超时。

10.0.1.1 的 Ping 统计信息:

数据包:已发送= 4 , 已接收= 0 , 丢失= 4 (100%丢失) ,

可以看到,本地PC与R1 (10.0.12.1) 之间的通信是正常的,但是本地PC与模拟外网地址10.0.1.1之间无法进行通信。

2 . 配置本地AAA

在R1上配置本地AAA,使用命令local-user创建本地用户名为R1,登录密码为123,用户级别为10(用户级别的取值范围为0~15,值越大,级别越高),接入类型为PPP。

```
[R1]aaa
```

```
[R1-aaa]local-user R1 password cipher 123
```

```
[R1-aaa]local-user R1 privilege level 10
```

```
[R1-aaa]local-user R1 service-type ppp
```

使用命令authentication-scheme配置认证方案名为1;使用命令authentication-mode local配置认证方式为本地认证;使用命令authorization-scheme配置授权方案名为1;使用命令authorization-mode local配置授权方式为本地授权;使用命令accounting-scheme配置计费方案名为1;使用命令accounting-mode none配置计费方式为不计费。

```
[R1-aaa]authentication-scheme 1
```

```
[R1-aaa-authen-1]authentication-mode local
```

```
[R1-aaa-authen-1]authorization-scheme 1
```

```
[R1-aaa-author-1]authorization-mode local
```

```
[R1-aaa-author-1]accounting-scheme 1
```

```
[R1-aaa-accounting-1]accounting-mode none
```

使用命令domain配置域名为huawei；使用命令authentication-scheme配置huawei域使用认证方案1；使用命令authorization-scheme配置huawei域使用授权方案1；使用命令accounting-scheme配置huawei域使用计费方案1。

```
[R1-aaa]domain huawei
```

```
[R1-aaa-domain-huawei]authentication-scheme 1
```

```
[R1-aaa-domain-huawei]authorization-scheme 1
```

```
[R1-aaa-domain-huawei]accounting-scheme 1
```

使用命令interface virtual-template 创建名为1 的虚拟接口，配置虚拟接口的IP 地址为10.0.100.1/24。创建名为huawei的全局地址池，配置地址池的地址范围为10.0.100.0 ~ 10.0.100.255，使用命令gateway-list配置网关地址为10.0.100.1。在R1的GE 0/0/0 接口上使用命令pppoe-server bind 绑定虚拟接口。

```
[R1]interface virtual-template 1
```

```
[R1-Virtual-Template1]ip address 10.0.100.1 24
```

```
[R1-Virtual-Template1]remote address pool huawei
```

```
[R1-Virtual-Template1]ppp authentication-mode chap domain huawei
```

```
[R1-Virtual-Template1]ip pool huawei
```

```
[R1-ip-pool-huawei]network 10.0.100.0 mask 24
```

```
[R1-ip-pool-huawei]gateway-list 10.0.100.1
```

```
[R1-ip-pool-huawei]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]pppoe-server bind virtual-template 1
```

在本地 PC 上创建拨号连接，使用所配置的用户名和密码进行拨号，并查看结果，如图8-3所示。



图8-3 在本地PC上创建拨号连接

在R1 上使用命令 `display pppoe-server session all` 查看用户接入情况。

```
<R1>display pppoe-server session all
```

SID	Intf	State	OIntf	RemMAC	LocMAC
1	Virtual-Template1:0	UP	GE0/0/0	001e.9014.66a7	00e0.fc03.f4f4

可以看到，本地PC已经成功建立了PPPoE的连接，其中RemMAC表示拨号主机的MAC地址，而LocMAC则表示R1自身接口的MAC地址。

在本地PC上使用ping命令测试与R1的Loopback 0接口（10.0.1.1）之间的连通性。

```
C:\Users\admin>ping 10.0.1.1
```

正在 Ping 10.0.1.1 具有 32 字节的数据:

来自 10.0.1.1 的回复:字节=32 时间=30ms TTL=255

来自 10.0.1.1 的回复:字节=32 时间=26ms TTL=255

来自 10.0.1.1 的回复:字节=32 时间=25ms TTL=255

来自 10.0.1.1 的回复:字节=32 时间=33ms TTL=255

10.0.1.1 的 Ping 统计信息:

数据包:已发送= 4 , 已接收= 4 , 丢失= 0 (0%丢失), 往返行程的估计时间(以毫秒为单位):

最短= 25ms , 最长= 33ms , 平均= 28ms

可以看到, 认证成功之后, 本地PC可以与模拟外网地址10.0.1.1进行正常通信了。

3 . 配置远程AAA

公司网络升级后, 需要使用远程AAA认证。以下是在AAA客户端R1上的配置过程。

使用命令radius-server template创建名为 share 的 RADIUS 服务器模板, 使用命令radius-server authentication配置RADIUS 主用认证服务器的IP 地址为10.1.1.1, 端口号为 1812, 使用命令 radius-server accounting 配置 RADIUS 计费服务器的 IP 地址为10.1.1.1, 端口号为1813 (注: 另外还需要配置访问账号和密码等内容, 这里略去)。

```
[R1]radius-server template share
```

```
[R1-radius-share]radius-server authentication 10.1.1.1 1812
```

```
[R1-radius-share]radius-server accounting 10.1.1.1 1813
```

配置采用RADIUS协议的AAA认证和计费方案。

```
[R1]aaa
```

```
[R1-aaa]authentication-scheme 2
```

```
[R1-aaa-authen-2]authentication-mode radius
```

```
[R1-aaa-authen-2]accounting-scheme 2
```

```
[R1-aaa-accounting-2]accounting-mode radius
```

创建域名为huawei2的域, 指定相应的认证和计费方案, 并指定相应的RADIUS服务器模板。

```
[R1]aaa
[R1-aaa]domain huawei2
[R1-aaa-domain-huawei2]authentication-scheme 2
[R1-aaa-domain-huawei2]accounting-scheme 2
[R1-aaa-domain-huawei2]radius-server share
```

至此，R1 上的配置工作便告结束，然后还需要在 RADIUS 服务器侧进行配置，具体配置方法请参考相应的 RADIUS 服务器软件的说明。配置完成后，R1 便能够通过RADIUS协议与AAA服务器进行通信了，并对接入用户进行相应的认证和计费。

思考

AAA认证与802.1x认证之间存在怎样的关系？

8.2 配置BFD

原理概述

为了减小设备故障对网络业务造成的影响，提高网络的可用性，网络设备需要能够尽快检测到与相邻设备之间的通信故障，以便及时采取措施，保证业务尽快恢复正常。

目前，主要的故障检测机制包括两大类：硬件检测机制和慢Hello检测机制。例如，SDH（Synchronous Digital Hierarchy）链路告警检测机制就是一种硬件检测机制，其优点是发现故障的速度很快，缺点是有些传输介质在有的条件下是无法支持这样的硬件检测机制的。慢Hello检测机制通常是指路由协议的Hello机制，这种检测机制存在的主要问题是发现故障的速度较慢，一般需要秒级的时间，这对于高速链路来说一般是无法接受的，因为秒级的时间将会造成大量的数据丢失。除了这两类故障检测机制外，有的设备厂商还提供了一些专用的

故障检测机制。然而，在进行不同厂商的设备互联时，这样的专用检测机制通常又是难以部署和实施的。

BFD（Bidirectional Forwarding Detection）技术就是为解决现有的故障检测机制的不足而产生的。BFD可以在相邻设备的转发引擎之间的通信通道上提供轻荷快速的故障检测能力，并在发现故障时即时通知上层应用。BFD可以发现的故障包括接口故障、链路故障，甚至可以是转发引擎本身的故障等。

BFD 的检测机制可以概括为：首先在两个设备之间建立起 BFD 会话，然后相互周期性地发送 BFD 控制报文，如果一方在预定的时间范围内没有收到另一方发送的 BFD 控制报文，则认为传输路径上发生了故障。为满足快速检测故障的需求，BFD 规定发送和接收控制报文的时间间隔大致在微秒级别。但是，限于目前的设备处理能力，大部分厂商的设备在实际运用 BFD 时都只能达到毫秒级别。

实验目的

- 理解 BFD 的基本原理和应用场景
- 掌握 BFD 的基本配置方法

实验内容

实验拓扑如图 8-4 所示，实验编址如表 8-2 所示。本实验模拟了一个简单的企业网络场景，公司使用了两台路由器 R1 和 R2 作为双网关出口，R1 与 R2 之间运行 VRRP（Virtual Router Redundancy Protocol）协议，R1 为主网关，R2 为备份网关，员工终端 PC-1 通过交换机 SW1 与出口网关相连。当主网关 R1 发生故障而切换到备份网关 R2 时，如果单纯通过 VRRP 协议进行故障切换则时间较慢，会造成大量的数据包丢失。因此，为了提高网络的可用性，需要在双网关之间配置 BFD 检测机制，以加速 VRRP 故障切换的过程，实现毫秒级的故障切换。

实验拓扑

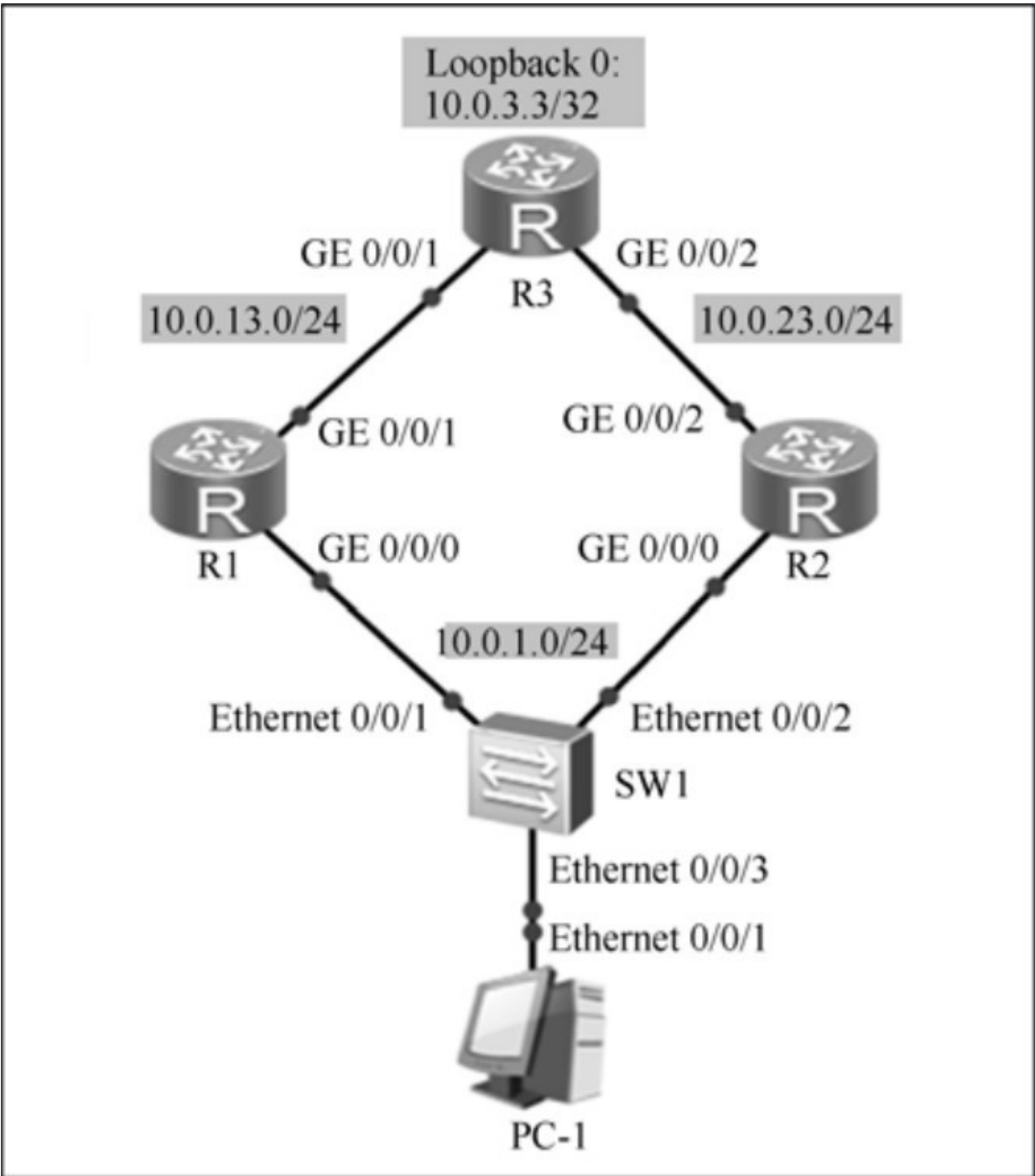


图8-4 配置BFD

实验编址表

表8-2 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0.1	10.0.1.100	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
R2(AR2220)	GE 0/0/0.1	10.0.1.200	255.255.255.0	N/A
	GE 0/0/2	10.0.23.2	255.255.255.0	N/A
R3(AR2220)	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.23.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
PC-1	Ethernet 0/0/1	10.0.1.1	255.255.255.0	10.0.1.254

实验步骤

1．基本配置

根据图8-4和表8-2进行相应的基本配置，并使用ping命令检测R1与R3之间的连通性。

```
<R1>ping -c 1 10.0.13.3
```

```
PING 10.0.13.3: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.13.3: bytes=56 Sequence=1 ttl=255 time=10 ms
```

```
--- 10.0.13.3 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

```
0.00% packet loss
```

```
round-trip min/avg/max = 10/10/10 ms
```

其余直连网段的连通性测试过程在此省略。

2．配置OSPF路由协议

在R1、R2和R3上配置OSPF协议。

```
[R1]ospf 1 router-id 1.1.1.1
```

```
[R1-ospf-1]area 0
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
```

```
[R1-ospf-1-area-0.0.0.0]network 10.0.1.0 0.0.0.255
```

```
[R2]ospf 1 router-id 2.2.2.2
```



```
[R2-ospf-1]area 0
[R2-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
[R2-ospf-1-area-0.0.0.0]network 10.0.1.0 0.0.0.255
[R3]ospf 1 router-id 3.3.3.3
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.13.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.23.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
```

配置完成后，在R3上查看OSPF邻居信息。

```
<R3>display ospf peer
```

OSPF Process 1 with Router ID 3.3.3.3

Neighbors

Area 0.0.0.0 interface 10.0.13.3(GigabitEthernet0/0/1)'s neighbors

```
Router ID: 1.1.1.1      Address: 10.0.13.1
State: Full Mode:Nbr is Slave Priority: 1
DR: 10.0.13.3 BDR: 10.0.13.1 MTU: 0
Dead timer due in 40 sec
Retrans timer interval: 5
Neighbor is up for 00:00:58
Authentication Sequence: [ 0 ]
```

Neighbors

Area 0.0.0.0 interface 10.0.23.3(GigabitEthernet0/0/2)'s neighbors

```
Router ID: 2.2.2.2      Address: 10.0.23.2
State: Full Mode:Nbr is Slave Priority: 1
DR: 10.0.23.3 BDR: 10.0.23.2 MTU: 0
Dead timer due in 29 sec
Retrans timer interval: 5
```

Neighbor is up for 00:00:54

Authentication Sequence: [0]

可以看到，R3与R1和R2已经成功建立起了OSPF邻接关系。

3. 配置VRRP协议

R1 和 R2 使用子接口配置 VRRP，R1 为主网关，R2 为备份网关，虚拟 IP 地址为10.0.1.254。使用子接口的目的是给后续配置VLAN终结做准备。

在R1的GE 0/0/0.1 子接口下创建备份组1，并配置R1 在备份组中的优先级的值为120。为了突出与配置了BFD之后的切换速度差异，现在将VRRP的Hello报文间隔从默认的1s修改为10s。

```
[R1]interface GigabitEthernet 0/0/0.1
```

```
[R1-GigabitEthernet0/0/0.1]vrrp vrid 1 virtual-ip 10.0.1.254
```

```
[R1-GigabitEthernet0/0/0.1]vrrp vrid 1 priority 120
```

```
[R1-GigabitEthernet0/0/0.1]vrrp vrid 1 timer advertise 10
```

在R2 的GE 0/0/0.1 子接口下创建备份组1，R2 在备份组中的优先级的值保持为默认值，VRRP的Hello报文间隔从默认的1s修改为10s。

```
[R2]interface GigabitEthernet 0/0/0.1
```

```
[R2-GigabitEthernet0/0/0.1]vrrp vrid 1 virtual-ip 10.0.1.254
```

```
[R2-GigabitEthernet0/0/0.1]vrrp vrid 1 timer advertise 10
```

在SW1 上创建VLAN 10，配置Ethernet 0/0/1和Ethernet 0/0/2的端口类型为Trunk，允许VLAN 10 的数据通过，端口Ethernet 0/0/3 的类型为Access，加入VLAN 10。

```
[SW1]vlan 10
```

```
[SW1]interface Ethernet0/0/1
```

```
[SW1-Ethernet0/0/1]port link-type trunk
```

```
[SW1-Ethernet0/0/1]port trunk allow-pass vlan 10
```

```
[SW1-Ethernet0/0/1]interface Ethernet0/0/2
```

```
[SW1-Ethernet0/0/2]port link-type trunk
[SW1-Ethernet0/0/2]port trunk allow-pass vlan 10
[SW1-Ethernet0/0/2]interface Ethernet 0/0/3
[SW1-Ethernet0/0/3]port link-type access
[SW1-Ethernet0/0/3]port default vlan 10
```

在R1和R2的子接口上配置Dot1q报文的VLAN终结，并开启ARP广播功能。

```
[R1-GigabitEthernet0/0/0.1]dot1q termination vid 10
[R1-GigabitEthernet0/0/0.1]arp broadcast enable
[R2-GigabitEthernet0/0/0.1]dot1q termination vid 10
[R2-GigabitEthernet0/0/0.1]arp broadcast enable
```

在PC-1 上长ping 路由器R3 的Loopback 0 接口的IP 地址10.0.3.3。如果R1与交换机之间的链路发生故障，VRRP会把工作网关的角色从R1切换到R2。但是，由于现在还没有配置BFD功能，所以此切换过程比较缓慢，切换过程中丢包较多。

请将VRRP备份组1设置的虚拟IP地址10.0.1.254配置为PC-1的网关地址（注：此过程在此省略，请读者自行完成），在PC-1上使用-t参数实现长ping，如图8-5所示，然后在 R1 的 GE 0/0/0.1 接口下执行命令 shutdown，模拟 R1 与 SW1 之间的链路故障。

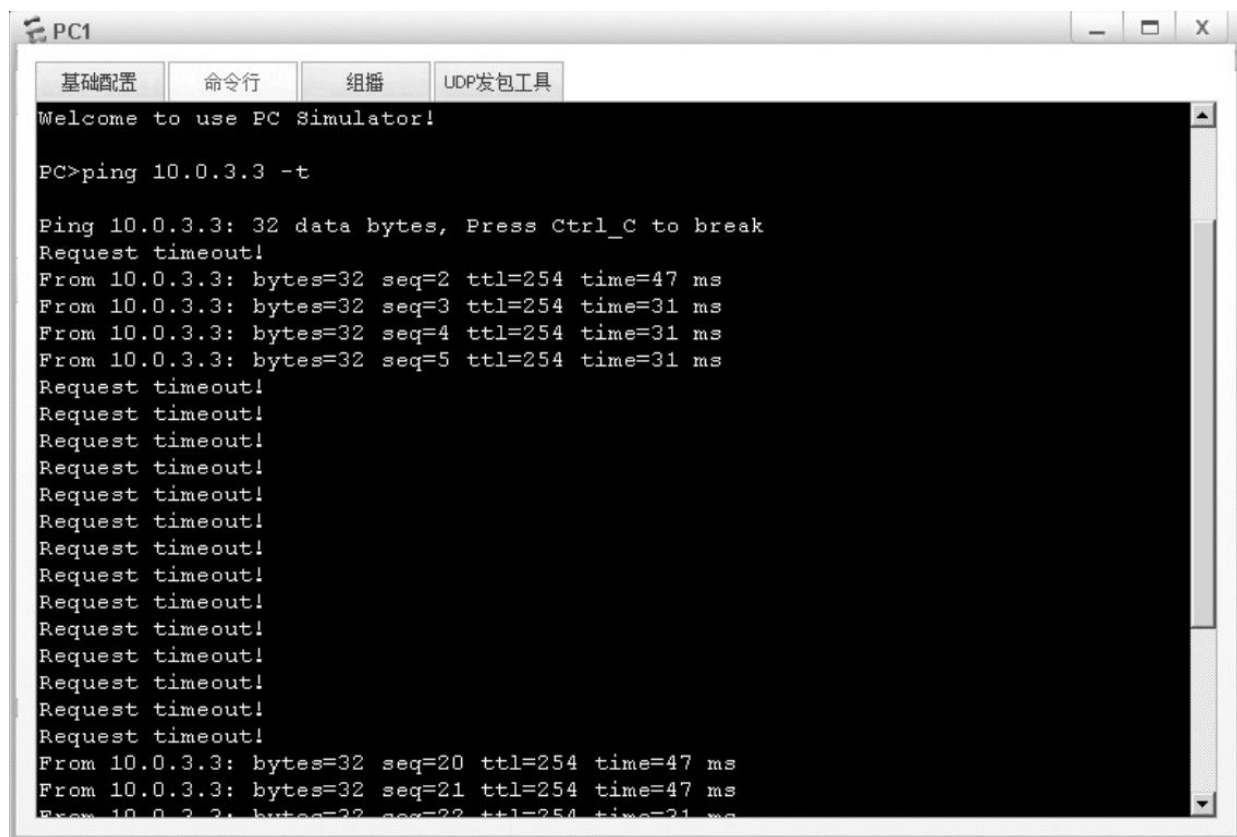


图8-5 无BFD时故障切换过程观察

从图8-5中可以看到，在故障切换的过程中，有14个报文发生了丢失。

4. 配置BFD

以R1为例，全局启用BFD功能，进入BFD视图，配置BFD延迟UP功能。

```
[R1]bfd
```

```
[R1-bfd]delay-up 50
```

建立BFD会话。

```
[R1]bfd 1-2 bind peer-ip 10.0.1.200 interface GigabitEthernet 0/0/0.1
```

配置本地及远端标识符。

```
[R1-bfd-session-1-2]discriminator local 1
```

```
[R1-bfd-session-1-2]discriminator remote 2
```

修改发送BFD Hello 报文的时间间隔为50ms。

```
[R1-bfd-session-1-2]min-tx-interval 50
```

修改接收Hello报文的等待时间为50ms，即每隔50ms期望能收到对方发来的一个BFD Hello 报文。

```
[R1-bfd-session-1-2]min-rx-interval 50
```

修改最多等待接收Hello报文的次数为3，也就是说，如果等待达到了3次仍没收到对方的BFD Hello 报文，则认为链路发生了故障。

```
[R1-bfd-session-1-2]detect-multiplier 3
```

提交BFD配置。

```
[R1-bfd-session-1-2]commit
```

使用相同的方法完成R2的BFD配置。

```
[R2]bfd
```

```
[R2-bfd]delay-up 50
```

```
[R2-bfd]bfd 2-1 bind peer-ip 10.0.1.100 interface GigabitEthernet  
0/0/0.1
```

```
[R2-bfd-session-2-1]discriminator local 2
```

```
[R2-bfd-session-2-1]discriminator remote 1
```

```
[R2-bfd-session-2-1]min-tx-interval 50
```

```
[R2-bfd-session-2-1]min-rx-interval 50
```

```
[R2-bfd-session-2-1]commit
```

在R1、R2 上使用display bfd session all 命令查看BFD 会话状态。

```
<R1>display bfd session all
```

```
-----
```

Local	Remote	PeerIpAddr	State	Type
InterfaceName				

1	2	10.0.1.200	Up	S_IP_IF

GigabitEthernet0/0/0.1

Total UP/DOWN Session Number : 1/0

[R2]display bfd session all

Local	Remote	PeerIpAddr	State	Type

InterfaceName

2	1	10.0.1.100	Up	S_IP_IF

GigabitEthernet0/0/0.1

Total UP/DOWN Session Number : 1/0

可以看到，BFD会话已经是UP状态。

配置VRRP监视BFD。

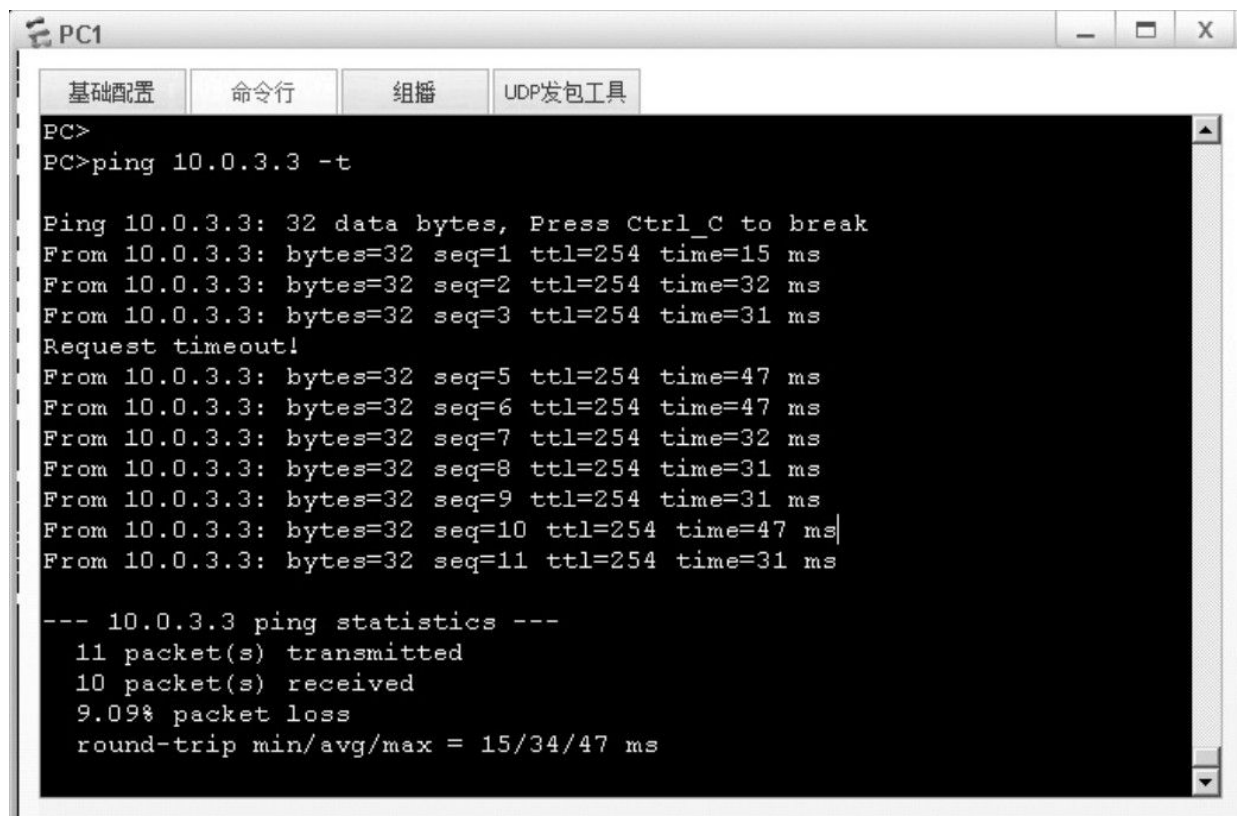
[R1-GigabitEthernet0/0/0.1]dot1q vrrp vid 10

[R1-GigabitEthernet0/0/0.1]vrrp vrid 1 track bfd-session 1 reduced 40

[R2-GigabitEthernet0/0/0.1]dot1q vrrp vid 10

[R2-GigabitEthernet0/0/0.1]vrrp vrid 1 track bfd-session 2 increased 40

在PC-1 上长ping 路由器R3 的Loopback 0 接口的IP 地址10.0.3.3，如图8-6所示。如果R1与交换机之间的链路发生故障，VRRP会把工作网关的角色从R1切换到R2。由于现在配置了BFD，切换过程会明显加快。



```
PC1
基础配置 命令行 组播 UDP发包工具
PC>
PC>ping 10.0.3.3 -t

Ping 10.0.3.3: 32 data bytes, Press Ctrl_C to break
From 10.0.3.3: bytes=32 seq=1 ttl=254 time=15 ms
From 10.0.3.3: bytes=32 seq=2 ttl=254 time=32 ms
From 10.0.3.3: bytes=32 seq=3 ttl=254 time=31 ms
Request timeout!
From 10.0.3.3: bytes=32 seq=5 ttl=254 time=47 ms
From 10.0.3.3: bytes=32 seq=6 ttl=254 time=47 ms
From 10.0.3.3: bytes=32 seq=7 ttl=254 time=32 ms
From 10.0.3.3: bytes=32 seq=8 ttl=254 time=31 ms
From 10.0.3.3: bytes=32 seq=9 ttl=254 time=31 ms
From 10.0.3.3: bytes=32 seq=10 ttl=254 time=47 ms
From 10.0.3.3: bytes=32 seq=11 ttl=254 time=31 ms

--- 10.0.3.3 ping statistics ---
 11 packet(s) transmitted
 10 packet(s) received
  9.09% packet loss
 round-trip min/avg/max = 15/34/47 ms
```

图8-6 有BFD时故障切换过程观察

从图8-6中可以看到，故障切换时间明显减少，丢包数量只有1个，网络的可用性得到了明显的提高。

思考

BFD的delay-up时间是指什么？其默认值是多少？

8.3 综合实验1

实验目的

- 增强分析和配置中小型企业网络的综合能力

实验内容

实验拓扑如图8-7所示，实验编址如表8-3所示。本实验模拟了一个企业网络场景，其中R1和R2为公司总部路由器，交换机S1、S2、S3组

成了总部的园区网，R3、R4、R5为公司分部的路由器。

总部园区网中3台交换机都运行MSTP协议，用来防止二层冗余网络中的环路以及实现不同VLAN间流量的负载分担，同时还配置了MSTP保护功能以提高网络的可靠性和安全性。

R1、R2、S2、S3运行IS-IS路由协议，以实现总部网络的互通。S2和S3使用IS-IS下发的缺省路由访问总部之外的网络。另外，为了提高网络的安全性，还需要配置IS-IS认证功能。

R3、R4、R5 运行 OSPF 路由协议，以实现公司分部网络的互通。总部与分部之间通过BGP路由协议互通，同时需要通过修改BGP路由的属性来实现流量的负载分担。

实验拓扑

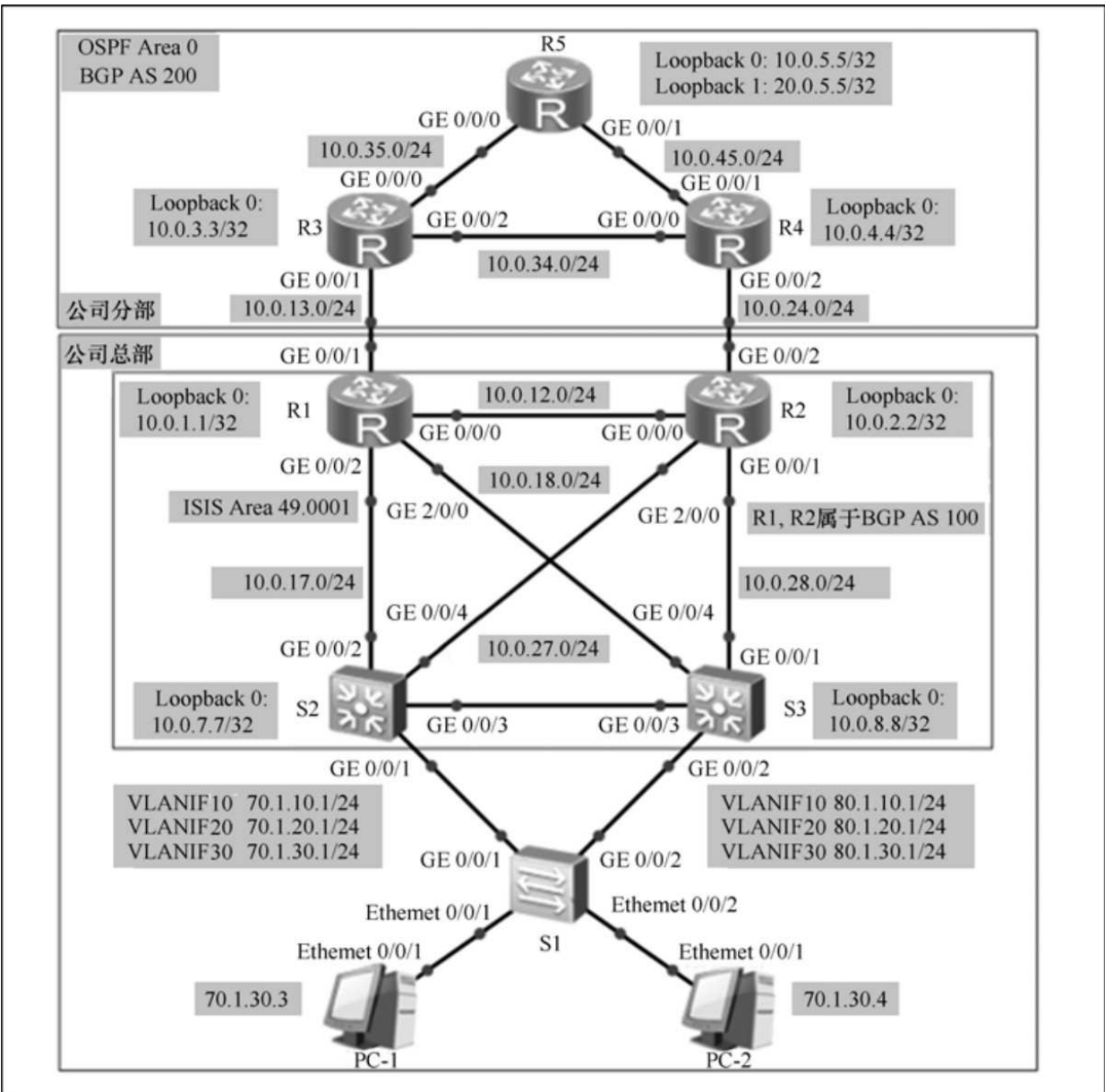


图8-7 综合实验1

实验编址表

表8-3 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.12.1	255.255.255.0	N/A
	GE 0/0/1	10.0.13.1	255.255.255.0	N/A
	GE 0/0/2	10.0.17.1	255.255.255.0	N/A
	GE 2/0/0	10.0.18.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.28.2	255.255.255.0	N/A
	GE 0/0/2	10.0.24.2	255.255.255.0	N/A
	GE 2/0/0	10.0.27.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A

(续表)

设备	接口	IP 地址	子网掩码	默认网关
R3(AR2220)	GE 0/0/0	10.0.35.3	255.255.255.0	N/A
	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.34.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2220)	GE 0/0/0	10.0.34.4	255.255.255.0	N/A
	GE 0/0/1	10.0.45.4	255.255.255.0	N/A
	GE 0/0/2	10.0.24.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
R5(AR2220)	GE 0/0/0	10.0.35.5	255.255.255.0	N/A
	GE 0/0/1	10.0.45.5	255.255.255.0	N/A
	Loopback 0	10.0.5.5	255.255.255.255	N/A
	Loopback 0	20.0.5.5	255.255.255.255	N/A
S2(S5700)	VLANIF 71	10.0.17.7	255.255.255.0	N/A
	VLANIF 72	10.0.27.7	255.255.255.0	N/A
	Loopback 0	10.0.7.7	255.255.255.255	N/A
S3(S5700)	VLANIF 81	10.0.18.8	255.255.255.0	N/A
	VLANIF 82	10.0.28.8	255.255.255.0	N/A
	Loopback 0	10.0.8.8	255.255.255.255	N/A
PC-1	Ethernet 0/0/1	70.1.30.3	255.255.255.0	70.1.30.2
PC-2	Ethernet 0/0/1	70.1.30.4	255.255.255.0	70.1.30.2

实验步骤

1. 基本配置

根据图8-7和表8-3进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

PING 10.0.12.2: 56 data bytes, press CTRL_C to break

Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=10 ms

--- 10.0.12.2 ping statistics ---

1 packet(s) transmitted

1 packet(s) received

0.00% packet loss

round-trip min/avg/max = 10/10/10 ms

其余直连网段的连通性测试过程在此省略。

2. 园区网划分VLAN

根据公司网络规划，在所有交换机上都创建VLAN 2、VLAN 3、VLAN 4、VLAN 10、VLAN 20 和VLAN 30。

```
[S1]vlan batch 2 3 4 10 20 30
```

```
[S2]vlan batch 2 3 4 10 20 30
```

```
[S3]vlan batch 2 3 4 10 20 30
```

为了保证不同交换机上的同一个VLAN的成员之间能够相互通信，需要配置交换机之间相连的端口为Trunk 端口，并允许VLAN 2、VLAN 3、VLAN 10、VLAN 20、VLAN 30通过。

```
[S1]interface GigabitEthernet 0/0/1
```

```
[S1-GigabitEthernet0/0/1]port link-type trunk
```

```
[S1-GigabitEthernet0/0/1]port trunk allow-pass vlan 2 3 10 20 30
```

```
[S1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[S1-GigabitEthernet0/0/2]port link-type trunk
```

```
[S1-GigabitEthernet0/0/2]port trunk allow-pass vlan 2 3 10 20 30
```

```
[S2]interface GigabitEthernet 0/0/1
```

```
[S2-GigabitEthernet0/0/1]port link-type trunk
```

```
[S2-GigabitEthernet0/0/1]port trunk allow-pass vlan 2 3 10 20 30
```

```
[S2-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/3
```

```
[S2-GigabitEthernet0/0/3]port link-type trunk
[S2-GigabitEthernet0/0/3]port trunk allow-pass vlan 2 3 10 20 30
[S3]interface GigabitEthernet 0/0/2
[S3-GigabitEthernet0/0/2]port link-type trunk
[S3-GigabitEthernet0/0/2]port trunk allow-pass vlan 2 3 10 20 30
[S3-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
[S3-GigabitEthernet0/0/3]port link-type trunk
[S3-GigabitEthernet0/0/3]port trunk allow-pass vlan 2 3 10 20 30
```

在 S1 上使用命令 `display vlan` 查看 VLAN 信息。读者可自行在 S2 和 S3 上查看VLAN信息。

```
[S1]display vlan
The total number of vlans is : 6
```

```
-----
U:Up;    D:Down;    TG:Tagged;    UT:Untagged;
MP:Vlan-mapping;    ST:Vlan-stacking;
#:ProtocolTransparent-vlan;    *:Management-vlan;
-----
```

```
-----
VID  Type  Ports
-----
```

```
-----
1    common  UT:    Eth0/0/1(D)    Eth0/0/2(D)
Eth0/0/3(D)    Eth0/0/4(D)
Eth0/0/5(D)    Eth0/0/6(D)    Eth0/0/7(D)    Eth0/0/8(D)
Eth0/0/9(D)    Eth0/0/10(D)   Eth0/0/11(D)   Eth0/0/12(D)
Eth0/0/13(D)   Eth0/0/14(D)   Eth0/0/15(D)   Eth0/0/16(D)
```

```

Eth0/0/17(D) Eth0/0/18(D) Eth0/0/19(D) Eth0/0/20(D)
Eth0/0/21(D) Eth0/0/22(D) GE0/0/1(U) GE0/0/2(U)
2 common TG: GE0/0/1(U) GE0/0/2(U)
3 common TG: GE0/0/1(U) GE0/0/2(U)
4 common
10 common TG: GE0/0/1(U) GE0/0/2(U)
20 common TG: GE0/0/1(U) GE0/0/2(U)
30 common TG: GE0/0/1(U) GE0/0/2(U)
VID Status Property MAC-LRN Statistics Description
-----

```

```

-----
1 enabledefault enable disable VLAN 0001
2 enabledefault enable disable VLAN 0002
3 enabledefault enable disable VLAN 0003
4 enabledefault enable disable VLAN 0004
10 enabledefault enable disable VLAN 0010
20 enabledefault enable disable VLAN 0020
30 enabledefault enable disable VLAN 0030

```

可以看到，S1 上除了缺省VLAN 1 之外，还有VLAN 2、VLAN 3、VLAN 4、VLAN 10、VLAN 20 和VLAN 30，且状态都为Enable，表明VLAN 创建成功。另外，所有Trunk端口都以Tagged模式加入进了相应的VLAN。

根据公司的网络规划，人事部属于VLAN 2，市场部属于VLAN 3；PC-1 是人事部的终端，PC-2 是市场部的终端。由于人事部和市场部有业务往来，需要进行跨 VLAN通信，所以决定在交换机S1 上配置VLAN 聚合，这样做的好处是，既可实现VLAN 2和VLAN 3 之间的通信，又可以节约IP 地址资源。

将PC-1 添加到VLAN 2 , PC-2 添加到VLAN 3。

```
[S1]interface Ethernet 0/0/1
```

```
[S1-Ethernet0/0/1]port link-type access
```

```
[S1-Ethernet0/0/1]port default vlan 2
```

```
[S1-Ethernet0/0/1]interface Ethernet 0/0/2
```

```
[S1-Ethernet0/0/2]port link-type access
```

```
[S1-Ethernet0/0/2]port default vlan 3
```

在PC-1上使用ping命令测试PC-1和PC-2之间的连通性，如图8-8所示。

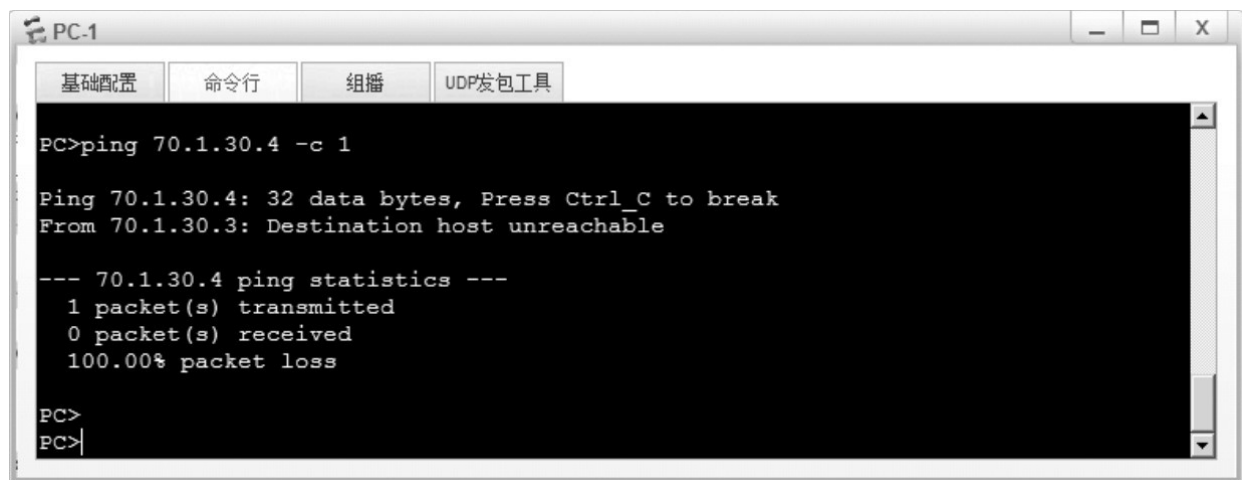


图8-8 PC-1和PC-2之间的连通性测试

从图8-8中可以看到，PC-1与PC-2现在还无法进行通信，尽管它们的IP地址属于同一网段。

在S1 上配置VLAN 4 为Super VLAN，VLANIF 4 的IP 地址为70.1.30.2/24，并配置Proxy ARP。

```
[S1]vlan 4
```

```
[S1-vlan4]aggregate-vlan
```

```
[S1-vlan4]access-vlan 2 to 3
```

```
[S1-vlan4]interface Vlanif 4
```

```
[S1-Vlanif4]ip address 70.1.30.2 24
```

```
[S1-Vlanif4]arp-proxy inter-sub-vlan-proxy enable
```

将 PC-1 和 PC-2 的网关地址都设置为 VLANIF 4 的 IP 地址 70.1.30.2，然后在 PC-1 上使用 ping 命令测试 PC-1 和 PC-2 之间的连通性，如图 8-9 所示。

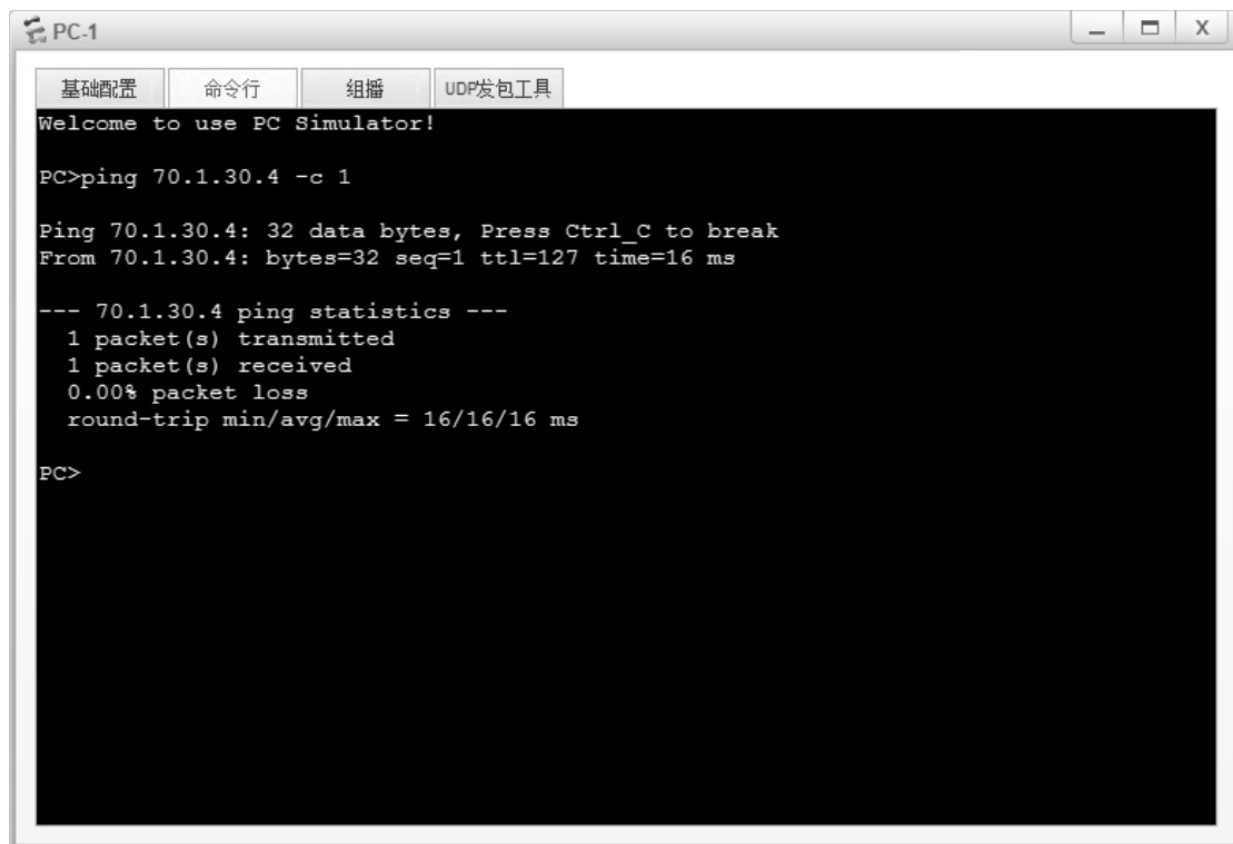


图8-9 PC-1和PC-2之间的连通性测试

从图8-9 中可以看到，PC-1 和 PC-2 现在通过 Super VLAN 4 实现了 VLAN 间的通信。

3. 配置MSTP协议

为了防止网络中的二层环路，同时对不同VLAN间的流量进行负载分担，配置所有交换机都工作在MSTP模式。

```
[S1]stp mode mstp
```

```
[S2]stp mode mstp
```

```
[S3]stp mode mstp
```

创建MSTP 域RG，其中的实例1 包含VLAN 2 和VLAN 3，并以S2 为根交换机；实例2 包含VLAN 10、VLAN 20 和VLAN 30，并以S3 为根交换机，修订版本号都为1。为了保证交换网络中加入了其他不支持MSTP的交换机后，S2仍为整个生成树的根交换机，使用命令stp instance 0 priority 0 配置交换机S2 为CIST 的总根。

```
[S1]stp region-configuration
```

```
[S1-mst-region]region-name RG
```

```
[S1-mst-region]instance 1 vlan 2 3
```

```
[S1-mst-region]instance 2 vlan 10 20 30
```

```
[S1-mst-region]revision-level 1
```

```
[S1-mst-region]active region-configuration
```

```
[S2]stp region-configuration
```

```
[S2-mst-region]region-name RG
```

```
[S2-mst-region]instance 1 vlan 2 3
```

```
[S2-mst-region]instance 2 vlan 10 20 30
```

```
[S2-mst-region]revision-level 1
```

```
[S2-mst-region]active region-configuration
```

```
[S2]stp instance 1 priority 0
```

```
[S2]stp instance 0 priority 0
```

```
[S3]stp region-configuration
```

```
[S3-mst-region]region-name RG
```

```
[S3-mst-region]instance 1 vlan 2 3
```

```
[S3-mst-region]instance 2 vlan 10 20 30
```

```
[S3-mst-region]revision-level 1
```

```
[S3-mst-region]active region-configuration
```

```
[S3]stp instance 2 priority 0
```

配置完成后，在S1上查看生成树信息以及不同实例的根交换机。

[S1]display stp instance1

-----[MSTI 1Global Info]-----

MSTI Bridge ID	:32768.4c1f-cc96-6cee
MSTI RegRoot/IRPC	:0.4c1f-cc35-15d0 / 20000
MSTI RootPortId	:128.23
Master Bridge	:0.4c1f-cc35-15d0
Cost to Master	:20000
TC received	:10
TC count per hello	:0
Time since last TC	:0 days 0h:14m:27s
Number of TC	:7
Last TC occurred	:GigabitEthernet0/0/1

----[Port1(Ethernet0/0/1)][FORWARDING]----

Port Role	:Designated Port
Port Priority	:128
Port Cost(Dot1T)	:Config=auto /Active=1
Designated Bridge/Port	:32768.4c1f-cc96-6cee /128.1
Port Times	:RemHops 19
TC or TCN send	:8
TC or TCN received	:0

----[Port2(Ethernet0/0/2)][FORWARDING]----

Port Role	:Designated Port
Port Priority	:128
Port Cost(Dot1T)	:Config=auto /Active=1
Designated Bridge/Port	:32768.4c1f-cc96-6cee /128.2
Port Times	:RemHops 19
TC or TCN send	:8

TC or TCN received :0
----[Port23(GigabitEthernet0/0/1)][FORWARDING]----
Port Role :Root Port
Port Priority :128
Port Cost(Dot1T) :Config=auto /Active=20000
Designated Bridge/Port :0.4c1f-cc35-15d0 / 128.1
Port Times :RemHops 20
TC or TCN send :5
TC or TCN received :3

----[Port24(GigabitEthernet0/0/2)][DISCARDING]----
Port Role :Alternate Port
Port Priority :128
Port Cost(Dot1T) :Config=auto /Active=20000
Designated Bridge/Port :32768.4c1f-cc77-11c7 / 128.2
Port Times :RemHops 19
TC or TCN send :1
TC or TCN received :7

[S1]display stp instance2

-----[MSTI 2 Global Info]-----
MSTI Bridge ID :32768.4c1f-cc96-6cee
MSTI RegRoot/IRPC :0.4c1f-cc77-11c7 / 20000
MSTI RootPortId :128.24
Master Bridge :0.4c1f-cc35-15d0
Cost to Master :20000
TC received :19
TC count per hello :0
Time since last TC :0 days 0h:13m:47s

```

Number of TC          :8
Last TC occurred      :GigabitEthernet0/0/2
----[Port23(GigabitEthernet0/0/1)][DISCARDING]----
Port Role             :Alternate Port
Port Priority          :128
Port Cost(Dot1T )    :Config=auto /Active=20000
Designated Bridge/Port :32768.4c1f-cc35-15d0 / 128.1
Port Times            :RemHops 19
TC or TCN send        :7
TC or TCN received    :8
----[Port24(GigabitEthernet0/0/2)][FORWARDING]----
Port Role             :Root Port
Port Priority          :128
Port Cost(Dot1T )    :Config=auto /Active=20000
Designated Bridge/Port :0.4c1f-cc77-11c7 / 128.2
Port Times            :RemHops 20
TC or TCN send        :2
TC or TCN received    :11

```

可以看到，在实例1中，S2为根交换机，S1上的GE 0/0/2为阻塞端口；在实例2中，S3为根交换机，S1上的GE 0/0/1为阻塞端口。这说明在不同实例中生成树选择的路径是不同的，并可因此而达到负载分担的目的。

为了保证网络的稳定性，确保当由于链路拥塞或者单向链路故障导致交换机收不到来自上游交换设备的BPDU时，不会产生临时环路，在S1上启用环路保护功能。

```

[S1]interface GigabitEthernet 0/0/1
[S1-GigabitEthernet0/0/1]stp loop-protection

```

```
[S1-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[S1-GigabitEthernet0/0/2]stp loop-protection
```

在S1上查看在生成树中的端口保护模式。

```
<S1>display stp brief
```

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	NONE
0	Ethernet0/0/2	DESI	FORWARDING	NONE
0	GigabitEthernet0/0/1	ROOT	FORWARDING	LOOP
0	GigabitEthernet0/0/2	ALTE	DISCARDING	LOOP
1	Ethernet0/0/1	DESI	FORWARDING	NONE
1	Ethernet0/0/2	DESI	FORWARDING	NONE
1	GigabitEthernet0/0/1	ROOT	FORWARDING	LOOP
1	GigabitEthernet0/0/2	ALTE	DISCARDING	LOOP
2	GigabitEthernet0/0/1	ALTE	DISCARDING	LOOP
2	GigabitEthernet0/0/2	ROOT	FORWARDING	LOOP

可以看到，阻塞端口和根端口都配置了环路保护功能。如果根端口或阻塞端口长时间收不到来自上游交换机的 BPDU，就会向网管发出通知信息，且阻塞端口会一直保持在阻塞状态，不转发报文，从而不会在网络中形成临时环路。

为了加快生成树的收敛速度，配置交换机S1 的Ethernet 0/0/1 和 Ethernet 0/0/2 为边缘端口，并配置保护功能以防止这些端口因收到不合法的BPDU而影响生成树的计算。

```
[S1]stp bpdu-protection
```

```
[S1]interface Ethernet 0/0/1
```

```
[S1-Ethernet0/0/1]stp edged-port enable
```

```
[S1-Ethernet0/0/1]interface Ethernet 0/0/2
```

```
[S1-Ethernet0/0/2]stp edged-port enable
```

配置完成后，在S1上查看边缘端口在生成树中的保护模式。

```
[S1]display stp brief
```

MSTID	Port	Role	STP State	Protection
0	Ethernet0/0/1	DESI	FORWARDING	BPDU
0	Ethernet0/0/2	DESI	FORWARDING	BPDU
0	GigabitEthernet0/0/1	ROOT	FORWARDING	LOOP
0	GigabitEthernet0/0/2	ALTE	DISCARDING	LOOP
1	Ethernet0/0/1	DESI	FORWARDING	BPDU
1	Ethernet0/0/2	DESI	FORWARDING	BPDU
1	GigabitEthernet0/0/1	ROOT	FORWARDING	LOOP
1	GigabitEthernet0/0/2	ALTE	DISCARDING	LOOP
2	GigabitEthernet0/0/1	ALTE	DISCARDING	LOOP
2	GigabitEthernet0/0/2	ROOT	FORWARDING	LOOP

可以看到，S1的两个边缘端口都开启了BPDU保护。如果边缘端口收到了恶意攻击的BPDU，则交换机将关闭边缘端口，并立即通知网管系统。

4．配置IS-IS路由协议

公司总部内R1、R2、S2、S3运行IS-IS路由协议，并且都属于同一个区域，System ID 由Loopback 0 接口地址转换而得到。

```
[S2]isis 1
```

```
[S2-isis-1]network-entity 49.0001.0100.0000.7007.00
```

```
[S2-isis-1]interface Vlanif 71
```

```
[S2-Vlanif71]isis enable
```

```
[S2-Vlanif71]interface Vlanif 72
```

```
[S2-Vlanif72]isis enable
```

```
[S2-Vlanif72]interface LoopBack 0
```

```
[S2-LoopBack0]isis enable
```

```
[S3]isis 1
[S3-isis-1]network-entity 49.0001.0100.0000.8008.00
[S3]interface Vlanif 81
[S3-Vlanif81]isis enable
[S3-Vlanif81]interface Vlanif 82
[S3-Vlanif82]isis enable
[S3-Vlanif82]interface Loopback 0
[S3-LoopBack0]isis enable

[R1]isis 1
[R1-isis-1]network-entity 49.0001.0100.0000.1001.00
[R1-isis-1]interface GigabitEthernet 0/0/0
[R1-GigabitEthernet0/0/0]isis enable
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/2
[R1-GigabitEthernet0/0/2]isis enable
[R1-GigabitEthernet0/0/2]interface GigabitEthernet 2/0/0
[R1-GigabitEthernet2/0/0]isis enable
[R1-GigabitEthernet2/0/0]interface Loopback 0
[R1-LoopBack0]isis enable

[R2]isis 1
[R2-isis-1]network-entity 49.0001.0100.0000.2002.00
[R2-isis-1]interface GigabitEthernet 0/0/0
[R2-GigabitEthernet0/0/0]isis enable
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
[R2-GigabitEthernet0/0/1]isis enable
[R2-GigabitEthernet0/0/1]interface GigabitEthernet 2/0/0
[R2-GigabitEthernet2/0/0]isis enable
[R2-GigabitEthernet2/0/0]interface Loopback 0
```

[R2-LoopBack0]isis enable

在R1上查看IS-IS邻居信息。

<R1>display isis peer

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime
Type	PRI			

0100.0000.2002	GE0/0/0	0100.0000.2002.01	Up	9s
L1(L1L2) 64				
0100.0000.2002	GE0/0/0	0100.0000.2002.01	Up	9s
L2(L1L2) 64				
0100.0000.7007	GE0/0/2	0100.0000.7007.01	Up	9s
L1(L1L2) 64				
0100.0000.7007	GE0/0/2	0100.0000.7007.01	Up	9s
L2(L1L2) 64				
0100.0000.8008	GE2/0/0	0100.0000.8008.01	Up	9s
L1(L1L2) 64				
0100.0000.8008	GE2/0/0	0100.0000.8008.01	Up	9s
L2(L1L2) 64				

Total Peer(s): 6

可以看到，R1已经分别与R2、S2、S3建立起了IS-IS邻居关系。

在R2上查看IS-IS邻居信息。

<R2>display isis peer

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime
Type	PRI			

```

-----
-----
      0100.0000.1001   GE0/0/0   0100.0000.2002.01   Up      25s
L1(L1L2)  64
      0100.0000.1001   GE0/0/0   0100.0000.2002.01   Up      23s
L2(L1L2)  64
      0100.0000.8008   GE0/0/1   0100.0000.8008.02   Up      7s
L1(L1L2)  64
      0100.0000.8008   GE0/0/1   0100.0000.8008.02   Up      8s
L2(L1L2)  64
      0100.0000.7007   GE2/0/0   0100.0000.7007.02   Up      7s
L1(L1L2)  64
      0100.0000.7007   GE2/0/0   0100.0000.7007.02   Up      9s
L2(L1L2)  64

```

Total Peer(s): 6

可以看到，R2已经分别与R1、S2、S3建立起了IS-IS邻居关系。
查看R1和R2上的路由表，读者可自行查看S2和S3上的路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

```

-----
-----
                                Routing Tables: Public
                                Destinations : 22   Routes : 24

Destination/Mask  Proto    Pre  Cost    Flags  NextHop
Interface
10.0.1.1/32      Direct   0    0       D      127.0.0.1  LoopBack0

```


10.0.2.2/32	ISIS-L1	15	10	D	10.0.12.2
GigabitEthernet0/0/0					
10.0.7.7/32	ISIS-L1	15	10	D	10.0.17.7
GigabitEthernet0/0/2					
10.0.8.8/32	ISIS-L1	15	10	D	10.0.18.8
GigabitEthernet2/0/0					
10.0.12.0/24	Direct	0	0	D	10.0.12.1
GigabitEthernet0/0/0					
10.0.12.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.12.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.13.0/24	Direct	0	0	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.13.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.13.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.17.0/24	Direct	0	0	D	10.0.17.1
GigabitEthernet0/0/2					
10.0.17.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
10.0.17.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
10.0.18.0/24	Direct	0	0	D	10.0.18.1
GigabitEthernet2/0/0					

10.0.18.1/32	Direct	0	0	D	127.0.0.1
GigabitEthernet2/0/0					
10.0.18.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet2/0/0					
10.0.27.0/24	ISIS-L1	15	20	D	10.0.17.7
GigabitEthernet0/0/2					
ISIS-L1	15	20	D	10.0.12.2	GigabitEthernet0/0/0
10.0.28.0/24	ISIS-L1	15	20	D	10.0.18.8
GigabitEthernet2/0/0					
ISIS-L1	15	20	D	10.0.12.2	GigabitEthernet0/0/0
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

<R2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 22Routes : 24

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					

10.0.1.1/32		ISIS-L1	15	10	D	10.0.12.1
GigabitEthernet0/0/0						
10.0.2.2/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.7.7/32		ISIS-L1	15	10	D	10.0.27.7
GigabitEthernet2/0/0						
10.0.8.8/32		ISIS-L1	15	10	D	10.0.28.8
GigabitEthernet0/0/1						
10.0.12.0/24	Direct	0	0	D	10.0.12.2	
GigabitEthernet0/0/0						
10.0.12.2/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.12.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/0						
10.0.17.0/24		ISIS-L1	15	20	D	10.0.12.1
GigabitEthernet0/0/0						
ISIS-L1	15	20	D	10.0.27.7	GigabitEthernet2/0/0	
10.0.18.0/24		ISIS-L1	15	20	D	10.0.12.1
GigabitEthernet0/0/0						
ISIS-L1	15	20	D	10.0.28.8	GigabitEthernet0/0/1	
10.0.24.0/24	Direct	0	0	D	10.0.24.2	
GigabitEthernet0/0/2						
10.0.24.2/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/2						
10.0.24.255/32	Direct	0	0	D	127.0.0.1	
GigabitEthernet0/0/2						
10.0.27.0/24	Direct	0	0	D	10.0.27.2	
GigabitEthernet2/0/0						

10.0.27.2/32	Direct	0	0	D	127.0.0.1
GigabitEthernet2/0/0					
10.0.27.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet2/0/0					
10.0.28.0/24	Direct	0	0	D	10.0.28.2
GigabitEthernet0/0/1					
10.0.28.2/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.28.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R2和R3通过IS-IS已经接收到了总部其他网段的路由。

接下来，将S2和S3中VLANIF 10、VLANIF 20、VLANIF 30接口所涉及的用户网段引进IS-IS中。另外，为了减少路由条目，需要将连续网段的路由进行聚合。

首先配置各VLANIF接口的IP地址。

```
[S2]interface Vlanif 10
```

```
[S2-Vlanif10]ip address 70.1.10.1 24
```

```
[S2-Vlanif10]interface Vlanif 20
```

```
[S2-Vlanif20]ip address 70.1.20.1 24
```

```
[S2-Vlanif20]interface Vlanif 30
[S2-Vlanif30]ip address 70.1.30.1 24
[S3]interface Vlanif 10
[S3-Vlanif10]ip address 80.1.10.1 24
[S3-Vlanif10]interface Vlanif 20
[S3-Vlanif20]ip address 80.1.20.1 24
[S3-Vlanif20]interface Vlanif 30
[S3-Vlanif30]ip address 80.1.30.1 24
```

然后将用户网段引入IS-IS中，且对连续网段的路由进行聚合。

```
[S2]isis 1
[S2-isis-1]import-route direct
[S2-isis-1]summary 70.1.0.0 255.255.224.0
[S3]isis 1
[S3-isis-1]import-route direct
[S3-isis-1]summary 80.1.0.0 255.255.224.0
```

查看R1的IP路由表，读者可自行查看R2、S2、S3的IP路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 24 Routes : 26						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
10.0.1.1/32	Direct	0	0	D	127.0.0.1	LoopBack0
.....						

70.1.0.0/19	ISIS-L2	15	74	D	10.0.17.7
GigabitEthernet0/0/2					
80.1.0.0/19	ISIS-L2	15	74	D	10.0.18.8
GigabitEthernet2/0/0					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
.....					

可以看到，R1已经通过IS-IS接收到了由S2和S3发布的、被聚合为19位掩码的用户网段的路由。

<S2>display isis interface

Interface		Id		IPV4.State	IPv6.State	MTU
Type	DIS					
Vlanif71	001	Up	Down	1497	L1/L2	Yes/Yes
Vlanif72	002	Up	Down	1497	L1/L2	Yes/Yes
Loop0	001	Up	Down	1500	L1/L2	--

可以看到，在以太网链路上，缺省情况是进行了DIS选举的。
查看S2的IS-IS链路状态数据库。

Database information for ISIS(1)

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0100.0000.1001.00-00	0x0000000f	0xab80	1193	140
0/0/0				
0100.0000.2002.00-00	0x0000000a	0x539b	389	140
0/0/0				
0100.0000.2002.01-00	0x00000002	0x4bea	389	55
0/0/0				
0100.0000.7007.00-00*	0x0000000e	0xa380	1067	
113 0/0/0				
0100.0000.7007.01-00*	0x00000004	0xa6e2	1067	
55 0/0/0				
0100.0000.7007.02-00*	0x00000003	0x6f09	1067	
55 0/0/0				
0100.0000.8008.00-00	0x0000000d	0xebfb	517	113
0/0/0				
0100.0000.8008.01-00	0x00000003	0x2246	517	55
0/0/0				
0100.0000.8008.02-00	0x00000003	0xe86d	517	55
0/0/0				

Total LSP(s): 9

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self
LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0100.0000.1001.00-00	0x00000012	0x5a39	1193	200

0/0/0

0100.0000.2002.00-00	0x0000000d	0xdb90	389	200

0/0/0

0100.0000.2002.01-00	0x00000002	0x4bea	389	55

0/0/0

0100.0000.7007.00-00*	0x00000015	0x4022	1067	

199 0/0/0

0100.0000.7007.00-01*	0x00000001	0xc1c0	0 (818)	

27 0/0/0

0100.0000.7007.01-00*	0x00000004	0xa6e2	1067	

55 0/0/0

0100.0000.7007.02-00*	0x00000003	0x6f09	1067	

55 0/0/0

0100.0000.8008.00-00	0x00000014	0x24fb	758	

199 0/0/0

0100.0000.8008.01-00	0x00000003	0x2246	517	55

0/0/0

0100.0000.8008.02-00	0x00000003	0xe86d	517	55

0/0/0

Total LSP(s): 10

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，在S2的IS-IS链路状态数据库中，存在着DIS产生的LSP。

修改网络类型为P2P，避免选举DIS。

```
[S2]interface Vlanif 71
```

```
[S2-Vlanif71]isis circuit-type p2p
```

```
[S2-Vlanif71]interface Vlanif 72
```

```
[S2-Vlanif72]isis circuit-type p2p
```

```
[S3]interface Vlanif 81
```

```
[S3-Vlanif81]isis circuit-type p2p
```

```
[S3-Vlanif81]interface Vlanif 82
```

```
[S3-Vlanif82]isis circuit-type p2p
```

```
[R1]interface GigabitEthernet 0/0/0
```

```
[R1-GigabitEthernet0/0/0]isis circuit-type p2p
```

```
[R1-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/2
```

```
[R1-GigabitEthernet0/0/2]isis circuit-type p2p
```

```
[R1-GigabitEthernet0/0/2]interface GigabitEthernet 2/0/0
```

```
[R1-GigabitEthernet2/0/0]isis circuit-type p2p
```

```
[R2]interface GigabitEthernet 0/0/0
```

```
[R2-GigabitEthernet0/0/0]isis circuit-type p2p
```

```
[R2-GigabitEthernet0/0/0]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]isis circuit-type p2p
```

```
[R2-GigabitEthernet0/0/1]interface GigabitEthernet 2/0/0
```

```
[R2-GigabitEthernet2/0/0]isis circuit-type p2p
```

配置完成后，刷新IS-IS链路状态数据库，这里仅以S2为例。

```
<S2>reset isis all
```

```
Warning: The IS-IS process(es) will be reset. Continue?[Y/N]y
```

查看IS-IS接口，这里仅以S2为例。

<S2>display isis interface

Interface information for ISIS(1)						

Interface		Id	IPv4.State		IPv6.State	MTU
Type	DIS					
	Vlanif71	002	Up	Down	1497	L1/L2 --
	Vlanif72	003	Up	Down	1497	L1/L2 --
	Loop0	001	Up	Down	1500	L1/L2 --

可以看到，没有进行DIS的选举。

查看IS-IS链路状态数据库，这里仅以S2为例。

<S2>display isis lsdb

Database information for ISIS(1)					

Level-1 Link State Database					
LSPID		Seq Num	Checksum	Holdtime	Length
ATT/P/OL					

140	0100.0000.1001.00-00 0/0/0	0x00000018	0x27fe		1087
140	0100.0000.2002.00-00 0/0/0	0x00000014	0x1f9		1086
113	0100.0000.7007.00-00* 0/0/0	0x00000015	0x5f7c		1091
113	0100.0000.8008.00-00 0/0/0	0x00000014	0xdb4		1029

Total LSP(s): 4

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0100.0000.1001.00-00	0x00000022	0x8501	1087	200

0/0/0

0100.0000.2002.00-00	0x0000001d	0xfe73		1085

200 0/0/0

0100.0000.7007.00-00*	0x00000022	0x57bc		1094

199 0/0/0

0100.0000.8008.00-00	0x00000021	0xed06	1045	199

0/0/0

Total LSP(s): 4

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，S2的链路状态数据库中已经没有了DIS的LSP，这样就减小了链路状态数据的大小，节约了存储资源和处理器资源。

S2和S3不运行BGP路由协议，所以为了使S2和S3能够访问外网，需要在路由器R1和R2上配置IS-IS下发缺省路由。

[R1]isis 1

[R1-isis-1]default-route-advertise

[R2]isis 1

[R2-isis-1]default-route-advertise

查看S2的IP路由表。

<S2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 21 Routes : 25						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
0.0.0.0/0	ISIS-L2	15	10	D	10.0.17.1	Vlanif71
ISIS-L2	15	10	D	10.0.27.2	Vlanif72	
10.0.1.1/32	ISIS-L1	15	10	D	10.0.17.1	Vlanif71
.....						

可以看到，S2的IP路由表中有了从IS-IS接收到的缺省路由，并采用了负载分担方式。

查看S3的IP路由表。

<S3>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 21 Routes : 25						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
0.0.0.0/0	ISIS-L2	15	10	D	10.0.18.1	Vlanif81

```
ISIS-L2 15 10 D 10.0.28.2 Vlanif82
10.0.1.1/32 ISIS-L1 15 10 D 10.0.18.1 Vlanif81
.....
```

可以看到，S3的IP路由表中也有了从IS-IS接收到的缺省路由，并采用了负载分担方式。

为了提高网络安全性，R1、R2、S2、S3均需要相互通过认证后才能交换IS-IS路由信息。配置认证模式为MD5认证，密钥为huawei。

```
[S2]isis 1
[S2-isis-1]area-authentication-mode md5 huawei
[S3]isis 1
[S3-isis-1]area-authentication-mode md5 huawei
[R1]isis 1
[R1-isis-1]area-authentication-mode md5 huawei
[R2]isis 1
[R2-isis-1]area-authentication-mode md5 huawei
```

查看IS-IS认证模式，这里仅以R1为例。

```
<R1>display isis brief
```

```
ISIS Protocol Information for ISIS(1)
```

```
-----
SystemId: 0100.0000.1001 System Level:L12
Area-Authentication-mode: MD5
Domain-Authentication-mode: NULL
```

```
.....
```

可以看到，IS-IS区域认证模式为MD5认证模式。

5. 配置OSPF路由协议

根据公司分部网络的设计，配置分部的所有路由器运行 OSPF 协议，路由器的Router-ID 采用Loopback 0 接口的IP 地址。

```
[R3]ospf router id 10.0.3.3
[R3-ospf-1]area 0
[R3-ospf-1-area-0.0.0.0]network 10.0.35.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R3-ospf-1-area-0.0.0.0]network 10.0.3.3 0.0.0.0
[R4]ospf router id 10.0.4.4
[R4-ospf-1]area 0
[R4-ospf-1-area-0.0.0.0]network 10.0.45.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.34.0 0.0.0.255
[R4-ospf-1-area-0.0.0.0]network 10.0.4.4 0.0.0.0
[R5]ospf router id 10.0.5.5
[R5-ospf-1]area 0
[R5-ospf-1-area-0.0.0.0]network 10.0.35.0 0.0.0.255
[R5-ospf-1-area-0.0.0.0]network 10.0.45.0 0.0.0.255
[R5-ospf-1-area-0.0.0.0]network 10.0.5.5 0.0.0.0
[R5-ospf-1-area-0.0.0.0]network 20.0.5.5 0.0.0.0
```

在R5上查看OSPF邻居信息。

```
[R5]display ospf peer brief
```

OSPF Process 1 with Router ID 10.0.5.5
Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	10.0.3.3	Full
0.0.0.0	GigabitEthernet0/0/1	10.0.4.4	Full

可以看到，R5与R3、R5与R4都已经成功建立起了OSPF邻接关系。

查看R5的IP路由表，读者可自行查看R3和R4的IP路由表。

```
[R5]display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

Routing Tables: Public						
Destinations : 15 Routes : 16						
Destination/Mask	Proto		Pre	Cost	Flags	NextHop
Interface						
10.0.3.3/32	OSPF		10	1	D	10.0.35.3
GigabitEthernet0/0/0						
10.0.4.4/32	OSPF		10	1	D	10.0.45.4
GigabitEthernet0/0/1						
10.0.5.5/32	Direct	0 0	D	127.0.0.1	LoopBack0	
10.0.34.0/24	OSPF		10	2	D	10.0.45.4
GigabitEthernet0/0/1						
OSPF	10	2 D	10.0.35.3	GigabitEthernet0/0/0		
10.0.35.0/24	Direct		0 0	D	10.0.35.5	
GigabitEthernet0/0/0						
.....						

可以看到，R5已经接收到了公司分部内其他网段的路由信息。

6. 配置BGP路由协议

在R1、R2、R3、R4、R5上配置BGP协议。R1与R3、R2与R4采用直连物理接口建立EBGP邻居关系；R1与R2使用Loopback 0接口建立

IBGP 邻居关系；R3、R4、R5 之间使用 Loopback 0 接口建立 IBGP 邻居关系。配置 R1 和 R2 的 Router-ID 为其 Loopback 0 接口的地址。

```
[R1]router id 10.0.1.1
[R1]bgp 100
[R1-bgp]peer 10.0.13.3 as-number 200
[R1-bgp]peer 10.0.2.2 as-number 100
[R1-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R2]router id 10.0.2.2
[R2]bgp 100
[R2-bgp]peer 10.0.24.4 as-number 200
[R2-bgp]peer 10.0.1.1 as-number 100
[R2-bgp]peer 10.0.1.1 connect-interface LoopBack 0
[R3]bgp 200
[R3-bgp]peer 10.0.13.1 as-number 100
[R3-bgp]peer 10.0.4.4 as-number 200
[R3-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R3-bgp]peer 10.0.5.5 as-number 200
[R3-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R4]bgp 200
[R4-bgp]peer 10.0.24.2 as-number 100
[R4-bgp]peer 10.0.3.3 as-number 200
[R4-bgp]peer 10.0.3.3 connect-interface LoopBack 0 [R4-bgp]peer
10.0.5.5 as-number 200
[R4-bgp]peer 10.0.5.5 connect-interface LoopBack 0
[R5]bgp 200
[R5-bgp]peer 10.0.3.3 as-number 200
[R5-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```



```
[R5-bgp]peer 10.0.4.4 as-number 200
```

```
[R5-bgp]peer 10.0.4.4 connect-interface LoopBack 0
```

在R1上查看BGP邻居信息。

```
<R1>display bgp peer
```

```
BGP local router ID : 10.0.1.1
```

```
Local AS number : 100
```

```
Total number of peers : 2      Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State		PrefRcv				
10.0.2.2	4	100	12	13	0	00:10:54 Established 0
10.0.13.3	4	200	12	13	0	00:10:03 Established

可以看到，R1的BGP邻居关系一切正常。

在R3上查看BGP邻居信息。

```
<R3>display bgp peer
```

```
BGP local router ID : 10.0.3.3
```

```
Local AS number : 200
```

```
Total number of peers : 3      Peers in established state : 3
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State		PrefRcv				
10.0.4.4	4	200	7	9	0	00:05:02
Established		0				
10.0.5.5	4	200	5	6	0	00:03:49
Established		0				
10.0.13.1	4	100	15	15	0	00:13:09 Established

可以看到，R3的BGP邻居关系一切正常。读者可自行在其他路由器上查看BGP邻居信息。

为了将公司总部的路由信息通告给公司分部，在R1和R2上同时将IS-IS的路由信息引进BGP进程。

```
[R1]bgp 100
```

```
[R1-bgp]import-route isis 1
```

```
[R2]bgp 100
```

```
[R2-bgp]import-route isis 1
```

查看R3的IP路由表中的BGP路由信息。

```
<R3>display ip routing-table protocol bgp
```

Route Flags: R - relay, D - download to fib

Public routing table : BGP

Destinations :11 Routes :11

BGP routing table status :<Active>

Destinations :11 Routes :11

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.1.1/32	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.2.2/32	EBGP	255	10	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.7.7/32	EBGP	255	10	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.8.8/32	EBGP	255	10	D	10.0.13.1
GigabitEthernet0/0/1					

10.0.12.0/24	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.17.0/24	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.18.0/24	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.27.0/24	EBGP	255	20	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.28.0/24	EBGP	255	20	D	10.0.13.1
GigabitEthernet0/0/1					
70.1.0.0/19	EBGP	255	74	D	10.0.13.1
GigabitEthernet0/0/1					
80.1.0.0/19	EBGP	255	74	D	10.0.13.1
GigabitEthernet0/0/1					

BGP routing table status :<Inactive>

Destinations : 0 Routes : 0

可以看到，R3 已经通过BGP 协议接收到了关于总部的路由信息。

查看R5的IP路由表中的BGP路由信息。

<R3>display ip routing-table protocol bgp

结果发现，竟然没有任何显示信息。

直接查看R5的BGP路由表。

<R5>display bgp routing-table

BGP Local router ID is 10.0.5.5

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 22

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	10.0.1.1/32	10.0.13.1	0	100	0	100?
i	10.0.24.2	10	100	0	100?	
i	10.0.2.2/32	10.0.13.1	10	100	0	100?
i	10.0.24.2	0	100	0	100?	
i	10.0.7.7/32	10.0.13.1	10	100	0	100?
i	10.0.24.2	10	100	0	100?	
i	10.0.8.8/32	10.0.13.1	10	100	0	100?
i	10.0.24.2	10	100	0	100?	
i	10.0.12.0/24	10.0.13.1	0	100	0	100?
i	10.0.24.2	0	100	0	100?	
i	10.0.17.0/24	10.0.13.1	0	100	0	100?
i	10.0.24.2	20	100	0	100?	
i	10.0.18.0/24	10.0.13.1	0	100	0	100?
i	10.0.24.2	20	100	0	100?	
i	10.0.27.0/24	10.0.13.1	20	100	0	100?
i	10.0.24.2	0	100	0	100?	
i	10.0.28.0/24	10.0.13.1	20	100	0	100?
i	10.0.24.2	0	100	0	100?	
i	70.1.0.0/19	10.0.13.1	74	100	0	100?
i	10.0.24.2	74	100	0	100?	
i	80.1.0.0/19	10.0.13.1	74	100	0	100?
i	10.0.24.2	74	100	0	100?	

可以看到，R5的BGP路由表中是存在总部路由的，然而这些路由却都是无效的（路由条目的前面没有带星号），原因是这些路由的下一跳R1（10.0.13.1）和R2（10.0.24.2）对于R5来说都是不可达的。解

决这个问题的方法有很多，下面采用在R3和R4上引入直连路由的方法，使R5知道该如何去往R1（10.0.13.1）和R2（10.0.24.2）。

[R3]bgp 200

```
[R3-bgp]import-route direct
```

[R4]bgp 200

```
[R4-bgp]import-route direct
```

查看R5的IP路由表。

```
<R5>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 28 Routes : 29

Interface	Destination/Mask	Proto	Pre	Cost	Flags	NextHop
GigabitEthernet0/0/0	10.0.1.1/32	IBGP		255	0	RD 10.0.13.1
	10.0.2.2/32	IBGP		255	0	RD 10.0.24.2
GigabitEthernet0/0/1	10.0.3.3/32	OSPF	10	1	D	10.0.35.3
	10.0.4.4/32	OSPF	10	1	D	10.0.45.4
GigabitEthernet0/0/1	10.0.5.5/32	Direct	0 0	D	127.0.0.1	LoopBack0
	10.0.7.7/32	IBGP		255	10	RD 10.0.24.2

10.0.8.8/32	IBGP	255	10	RD	10.0.24.2
GigabitEthernet0/0/1					
10.0.12.0/24	IBGP	255	0	RD	10.0.24.2
GigabitEthernet0/0/1					
10.0.13.0/24	IBGP	255	0	RD	10.0.3.3
GigabitEthernet0/0/0					
10.0.17.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.18.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.24.0/24	IBGP	255	0	RD	10.0.4.4
GigabitEthernet0/0/1					
10.0.27.0/24	IBGP	255	0	RD	10.0.24.2
GigabitEthernet0/0/1					
10.0.28.0/24	IBGP	255	0	RD	10.0.24.2
GigabitEthernet0/0/1					
10.0.34.0/24	OSPF	10	2	D	10.0.45.4
GigabitEthernet0/0/1					
OSPF	10	2	D	10.0.35.3	GigabitEthernet0/0/0
.....					
20.0.5.5/32	Direct	0	0	D	127.0.0.1 LoopBack1
70.1.0.0/19	IBGP	255	74	RD	10.0.24.2
GigabitEthernet0/0/1					
80.1.0.0/19	IBGP	255	74	RD	10.0.24.2
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					

```

127.0.0.1/32      Direct    0    0      D      127.0.0.1
InLoopBack0
127.255.255.255/32 Direct    0    0      D      127.0.0.1
InLoopBack0
255.255.255.255/32 Direct    0    0      D      127.0.0.1
InLoopBack0

```

可以看到，R1和R2可达之后，R5便把相应的路由从BGP路由表中添加进了自己的IP路由表。

然而，公司分部的管理员发现，R3和R4上引入直连路由后，R3和R4的IP路由表发生了变化。

查看R3的IP路由表。

```
<R3>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```

-----
Routing Tables: Public
Destinations : 30  Routes : 31

Destination/Mask    Proto   Pre  Cost   Flags  NextHop
-----
Interface
10.0.1.1/32         EGBP    255   0      D      10.0.13.1
GigabitEthernet0/0/1
10.0.2.2/32         IBGP    255   0      RD     10.0.24.2
GigabitEthernet0/0/2
10.0.3.3/32        Direct   0    0      D      127.0.0.1  LoopBack0
.....
10.0.18.0/24        EGBP    255   0      D      10.0.13.1
GigabitEthernet0/0/1

```

```

10.0.24.0/24          IBGP      255    0      RD    10.0.4.4
GigabitEthernet0/0/2
10.0.27.0/24          IBGP      255    0      RD    10.0.24.2
GigabitEthernet0/0/2
10.0.28.0/24          IBGP      255    0      RD    10.0.24.2
GigabitEthernet0/0/2
10.0.34.0/24          Direct    0      0      D      10.0.34.3
GigabitEthernet0/0/2

```

.....

可以看到，R3上有些总部的路由是来自R3的IBGP邻居。

查看R4的IP路由表。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```

-----
Routing Tables: Public
Destinations : 30  Routes : 31

Destination/Mask    Proto    Pre  Cost    Flags  NextHop
-----
Interface
10.0.1.1/32          IBGP      255    0      RD    10.0.13.1
GigabitEthernet0/0/0
10.0.2.2/32          EBGp      255    0      D      10.0.24.2
GigabitEthernet0/0/2
.....
10.0.12.0/24         EBGp      255    0      D      10.0.24.2
GigabitEthernet0/0/2

```


10.0.13.0/24	IBGP	255	0	RD	10.0.3.3
GigabitEthernet0/0/0					
10.0.17.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.18.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.24.0/24	Direct	0	0	D	10.0.24.4
GigabitEthernet0/0/2					

.....

可以看到，R4上有些总部的路由是来自R4的IBGP邻居。

在R3上查看BGP路由表，查找问题的原因。

<R3>display bgp routing-table

BGP Local router ID is 10.0.3.3

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 32

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>	10.0.1.1/32	10.0.13.1	0		0	100?
*>i	10.0.2.2/32	10.0.24.2	0	100	0	100?
*		10.0.13.1	10	0	100?	
*>	10.0.3.3/32	0.0.0.0	0		0	?
.....						
*>i	10.0.24.0/24	10.0.4.4	0	100	0	?
*>i	10.0.27.0/24	10.0.24.2	0	100	0	100?
*		10.0.13.1	20	0	100?	
*>i	10.0.28.0/24	10.0.24.2	0	100	0	100?

可以看到，因为R1和R2上IS-IS路由条目的Cost值不一样，引入BGP时造成了MED值不一样，最后导致R3会偏好一些来自IBGP的总部路由。管理员希望，R3上所获得的总部路由全部来自R1，R4上所获得的总部路由全部来自R2，这样便可以使得从公司分部去往公司总部的报文不会在公司分部内部弯绕。实现这一需求的方法有很多，比如，在R1和R2上将IS-IS路由引入BGP时，可以设定MED的值。

Routing Tables: Public					
Destinations : 30 Routes : 31					
Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.1.1/32	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					
10.0.2.2/32	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					

10.0.3.3/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.4.4/32	OSPF		10	1	D	10.0.34.4
GigabitEthernet0/0/2						
10.0.5.5/32	OSPF		10	1	D	10.0.35.5
GigabitEthernet0/0/0						
10.0.7.7/32	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.8.8/32	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.12.0/24	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.13.0/24	Direct		0	0	D	10.0.13.3
GigabitEthernet0/0/1						
10.0.13.3/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.13.255/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.17.0/24	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.18.0/24	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.24.0/24	IBGP		255	0	R	10.0.4.4
GigabitEthernet0/0/2						
10.0.27.0/24	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.28.0/24	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						

10.0.34.0/24	Direct	0	0	D	10.0.34.3
GigabitEthernet0/0/2					
.....					
20.0.5.5/32	OSPF	10	1	D	10.0.35.5
GigabitEthernet0/0/0					
70.1.0.0/19	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					
80.1.0.0/19	EBGP	255	0	D	10.0.13.1
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R3 上接收到的总部路由现在都来自R3 的EBGP 邻居R1。

查看R4的IP路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

```

-----
Routing Tables: Public
Destinations : 30Routes : 31

```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.1.1/32	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.2.2/32	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.3.3/32	OSPF	10	1	D	10.0.34.3
GigabitEthernet0/0/0					
10.0.4.4/32	Direct	0	0	D	127.0.0.1
10.0.5.5/32	OSPF	10	1	D	10.0.45.5
GigabitEthernet0/0/1					
10.0.7.7/32	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.8.8/32	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.12.0/24	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.13.0/24	IBGP	255	0	RD	10.0.3.3
GigabitEthernet0/0/0					
10.0.17.0/24	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.18.0/24	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.24.0/24	Direct	0	0	D	10.0.24.4
GigabitEthernet0/0/2					
10.0.24.4/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					

10.0.24.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
10.0.27.0/24	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.28.0/24	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
10.0.34.0/24	Direct	0	0	D	10.0.34.4
GigabitEthernet0/0/0					
.....					
20.0.5.5/32	OSPF	10	1	D	10.0.45.5
GigabitEthernet0/0/1					
70.1.0.0/19	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
80.1.0.0/19	EBGP	255	0	D	10.0.24.2
GigabitEthernet0/0/2					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R4 上接收到的总部路由现在都来自R4 的EBGP 邻居R2。

为了让公司总部知道公司分部的路由，在R3和R4上把OSPF路由引进BGP进程。

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 33 Routes : 35						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
0.0.0.0/0	ISIS-L2	15	10	D	10.0.12.2	
GigabitEthernet0/0/0						
10.0.1.1/32	Direct	0 0	D	127.0.0.1	LoopBack0	
10.0.2.2/32	ISIS-L1	15	10	D	10.0.12.2	
GigabitEthernet0/0/0						
10.0.3.3/32	EBGP	255	0	D	10.0.13.3	
GigabitEthernet0/0/1						
10.0.4.4/32	IBGP	255	0	RD	10.0.24.4	
GigabitEthernet0/0/1						
10.0.5.5/32	EBGP	255	1	D	10.0.13.3	
GigabitEthernet0/0/1						
10.0.7.7/32	ISIS-L1	15	10	D	10.0.17.7	
GigabitEthernet0/0/2						
.....						

```

10.0.18.255/32      Direct      0    0      D      127.0.0.1
GigabitEthernet2/0/0
10.0.24.0/24       EBGp        255   0      D      10.0.13.3
GigabitEthernet0/0/1
10.0.27.0/24       ISIS-L1     15    20     D      10.0.17.7
GigabitEthernet0/0/2
ISIS-L1  15  20    D    10.0.12.2  GigabitEthernet0/0/0
10.0.28.0/24       ISIS-L1     15    20     D      10.0.18.8
GigabitEthernet2/0/0
ISIS-L1  15  20    D    10.0.12.2  GigabitEthernet0/0/0
10.0.34.0/24       EBGp        255   0      D      10.0.13.3
GigabitEthernet0/0/1
10.0.35.0/24       EBGp        255   0      D      10.0.13.3
GigabitEthernet0/0/1
10.0.45.0/24       IBGP        255   0      RD     10.0.24.4
GigabitEthernet0/0/1
20.0.5.5/32        EBGp        255   1      D      10.0.13.3
GigabitEthernet0/0/1
70.1.0.0/19        ISIS-L2     15    74     D      10.0.17.7
GigabitEthernet0/0/2

```

.....

可以看到，R1已经接收到了公司分部网络的路由。

考虑到公司分部网络的后续扩展问题，为了避免扩展后有太多的IBGP对等体关系需要建立，决定配置R3为BGP路由反射器。这样，如果公司分部网络中加入了新的BGP路由器，只需将新加入的路由器配置为BGP路由反射器R3的客户端即可。

```
[R3]bgp 200
```



```
[R3-bgp]peer 10.0.4.4 reflect-client
```

```
[R3-bgp]peer 10.0.5.5 reflect-client
```

7. 策略配置

在之前的配置中，R1和R2强制下发了IS-IS缺省路由。如果R1与R3之间的链路发生了故障，则S2去往公司分部的报文会继续发给R1，再由R1转发给R2，这样就产生了次优路径。如果R1与R2之间的链路也断掉了，S2去往公司分部的报文会继续发给R1，而此时R1要去往分部只有经过S2-R2-R4-R5或S3-R2-R4-R5，这样一来，就会有部分报文在R1和S2之间来回转发，形成环路。

为了防止次优路径和环路的产生，在R1和R2上配置 Router-Policy，当R1和R2发布缺省路由时加上此Router-Policy作为限制条件。

```
[R1]acl 2001
```

```
[R1-acl-basic-2001]rule permit source 10.0.13.0 0
```

```
[R1-acl-basic-2001]route-policy isis permit node 10
```

```
[R1-route-policy]if-match acl 2001
```

```
[R1-route-policy]isis 1
```

```
[R1-isis-1]default-route-advertise route-policy isis
```

```
[R2]acl 2001
```

```
[R2-acl-basic-2001]rule permit source 10.0.24.0 0
```

```
[R2-acl-basic-2001]route-policy isis permit node 10
```

```
[R2-route-policy]if-match acl 2001
```

```
[R2-route-policy]isis 1
```

```
[R2-isis-1]default-route-advertise route-policy isis
```

断掉R1与R3之间的链路后，查看S2和S3的IP路由表，然后恢复链路。这里仅以S2为例。

```
<S2>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```

-----
-----
                                Routing Tables: Public
                                Destinations : 21  Routes : 24
Destination/Mask    Proto    Pre  Cost    Flags  NextHop
Interface
    0.0.0.0/0                ISIS-L2  15   10      D      10.0.27.2
Vlanif72
    10.0.1.1/32              ISIS-L1  15   10      D      10.0.17.1
Vlanif71
.....

```

可以看到，当R1与R3之间的链路断掉后，由于添加了策略，R1不再下发缺省路由，只有R2下发了缺省路由。S2的IP路由表中的缺省路由就来自R2（10.0.27.2）。

现在，公司分部管理员发现，路由器R5上去往总部用户网段的路由下一跳都为R4。

```

<R5>display ip routing-table protocol bgp
Route Flags: R - relay, D - download to fib

```

```

-----
-----
                                Public routing table : BGP
                                Destinations : 20Routes : 20
BGP routing table status :<Active>
                                Destinations : 13  Routes : 13
Destination/Mask    Proto    Pre  Cost    Flags  NextHop
Interface

```

10.0.1.1/32	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.2.2/32	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.7.7/32	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.8.8/32	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.12.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.13.0/24	IBGP	255	0	RD	10.0.3.3
GigabitEthernet0/0/0					
10.0.17.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.18.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.24.0/24	IBGP	255	0	RD	10.0.4.4
GigabitEthernet0/0/1					
10.0.27.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
10.0.28.0/24	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
70.1.0.0/19	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
80.1.0.0/19	IBGP	255	0	RD	10.0.13.1
GigabitEthernet0/0/0					
.....					

为了实现从 R5 去往总部的流量能够负载分担，在 R5 上修改 BGP 路由的 Local Preference 属性，从而保证 R5 通过 R3 去访问 S2 所连接的总部用户网段，通过 R4 去访问 S3 所连接的总部用户网段。

```
[R5]acl 2001
[R5-acl-basic-2001]rule permit source 80.1.0.0 0.0.31.255
[R5-acl-basic-2001]route-policy fuzai permit node 10
[R5-route-policy]if-match acl 2001
[R5-route-policy]apply local-preference 200
[R5-route-policy]route-policy fuzai permit node 20
[R5-route-policy]bgp 200
[R5-bgp]peer 10.0.4.4 route-policy fuzai import
```

查看 R5 的 BGP 路由表。

```
<R5>display bgp routing-table
```

BGP Local router ID is 10.0.5.5

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 39

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.13.1	0	100	0	100?
* i		10.0.24.2	0	100	0	100?
.....						
*>i	70.1.0.0/19	10.0.13.1	0	100	0	100?
* i		10.0.24.2	0	100	0	100?
*>i	80.1.0.0/19	10.0.24.2	0	200	0	100?
* i		10.0.13.1	0	100	0	100?

可以看到，在R5上接收来自R4发送的80.1.0.0/19路由时，Local Preference的值已被增大为200。

查看R5的IP路由表中的BGP路由信息。

<R5>display ip routing-table protocol bgp

Route Flags: R - relay, D - download to fib

```
-----
Public routing table : BGP
Destinations : 20  Routes : 20
BGP routing table status : <Active>
Destinations : 13  Routes : 13
Destination/Mask  Proto    Pre   Cost    Flags   NextHop
Interface
10.0.1.1/32       IBGP     255   0        RD      10.0.13.1
GigabitEthernet0/0/0
.....
70.1.0.0/19       IBGP     255   0        RD      10.0.13.1
GigabitEthernet0/0/0
80.1.0.0/19       IBGP     255   0        RD      10.0.24.2
GigabitEthernet0/0/1
.....
```

可以看到，R5的BGP路由表中关于80.1.0.0/19的最优路由（即下一跳为10.0.24.2的那条路由）已经被添加进入了IP路由表。

R5的Loopback 0接口模拟了公司分部的研发部门网段，该研发部门只允许被总部访问，不允许被公司其他分部或其他公司访问。因此，管理员决定在R3和R4上通过修改BGP路由的团体属性，使研发部

门网段20.0.5.5被通告给总部路由器R1和R2的时候带上团体属性No-Export。

```
[R3]acl 2002
[R3-acl-basic-2002]rule permit source 20.0.5.5 0
[R3-acl-basic-2002]quit
[R3]route-policy 1 permit node 10
[R3-route-policy]if-match acl 2002
[R3-route-policy]apply community no-export
[R3-route-policy]route-policy 1 permit node 20
[R3-route-policy]bgp 200
[R3-bgp]peer 10.0.13.1 route-policy 1 export
[R3-bgp]peer 10.0.13.1 advertise-community
[R4]acl 2002
[R4-acl-basic-2002]rule permit source 20.0.5.5 0
[R4-acl-basic-2002]quit
[R4]route-policy 1 permit node 10
[R4-route-policy]if-match acl 2002
[R4-route-policy]apply community no-export
[R4-route-policy]route-policy 1 permit node 20
[R4-route-policy]bgp 200
[R4-bgp]peer 10.0.24.2 route-policy 1 export
[R4-bgp]peer 10.0.24.2 advertise-community
```

在R1和R2上查看带有团体属性的路由条目，这里仅以R1为例。

```
<R1>display bgp routing-table community
```

```
BGP Local router ID is 10.0.1.1
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

Total Number of Routes: 1

Network	NextHop	MED	LocPrf	PrefVal	Community
*> 20.0.5.5/32	10.0.13.3	1		0	no-export

可以看到，R1上关于20.0.5.5/32这条路由携带了团体属性No-Export，因此，它的传递范围将被限制在AS 100 中。至此，整个实验网络的分析和配置工作便告结束。

思考

在本实验的策略配置步骤中，为了实现从 R5 去往总部的流量能够负载分担，保证R5通过R3去访问S2所连接的总部用户网段，通过R4去访问S3所连接的总部用户网段，我们将 80.1.0.0/19 这条路由的 Local Preference 的值修改成了 200。如果要求将80.1.0.0/19这条路由的Local Preference的值修改成50，且保证完全一样的负载分担效果，则哪些配置命令的内容需要进行相应的修改？

8.4 综合实验2

实验目的

- 增强分析和配置中小型企业网络的综合能力

实验内容

实验拓扑如图8-10所示，实验编址如表8-4所示。本实验模拟了一个企业网络场景，其中R1为公司总部的路由器，交换机S1、S2、S3、S4，服务器，终端等设备组成了公司总部的园区网，R2、R3、R4为公司分部的路由器。

公司总部的园区网划分了不同的VLAN。为了防止二层环路及提高交换机的抗攻击性，每台交换机都需要运行RSTP协议，同时配置RSTP

保护功能。

在公司总部网络中，R1、S1、S2运行OSPF路由协议，并需要通过配置OSPF认证功能来提高安全性。由于种种原因，S3和S4不能运行OSPF路由协议，所以网络管理员需要将用户网段的路由引入OSPF进程，在路由引入的同时还需要实现路由聚合。

公司分部网络使用了IS-IS路由协议作为IGP，公司总部网络与公司分部网络之间通过BGP路由协议实现互通，同时，总部与分部之间的通信还需要满足负载分担等许多特别的要求，这些要求将在实验步骤中进行具体的说明。

实验拓扑

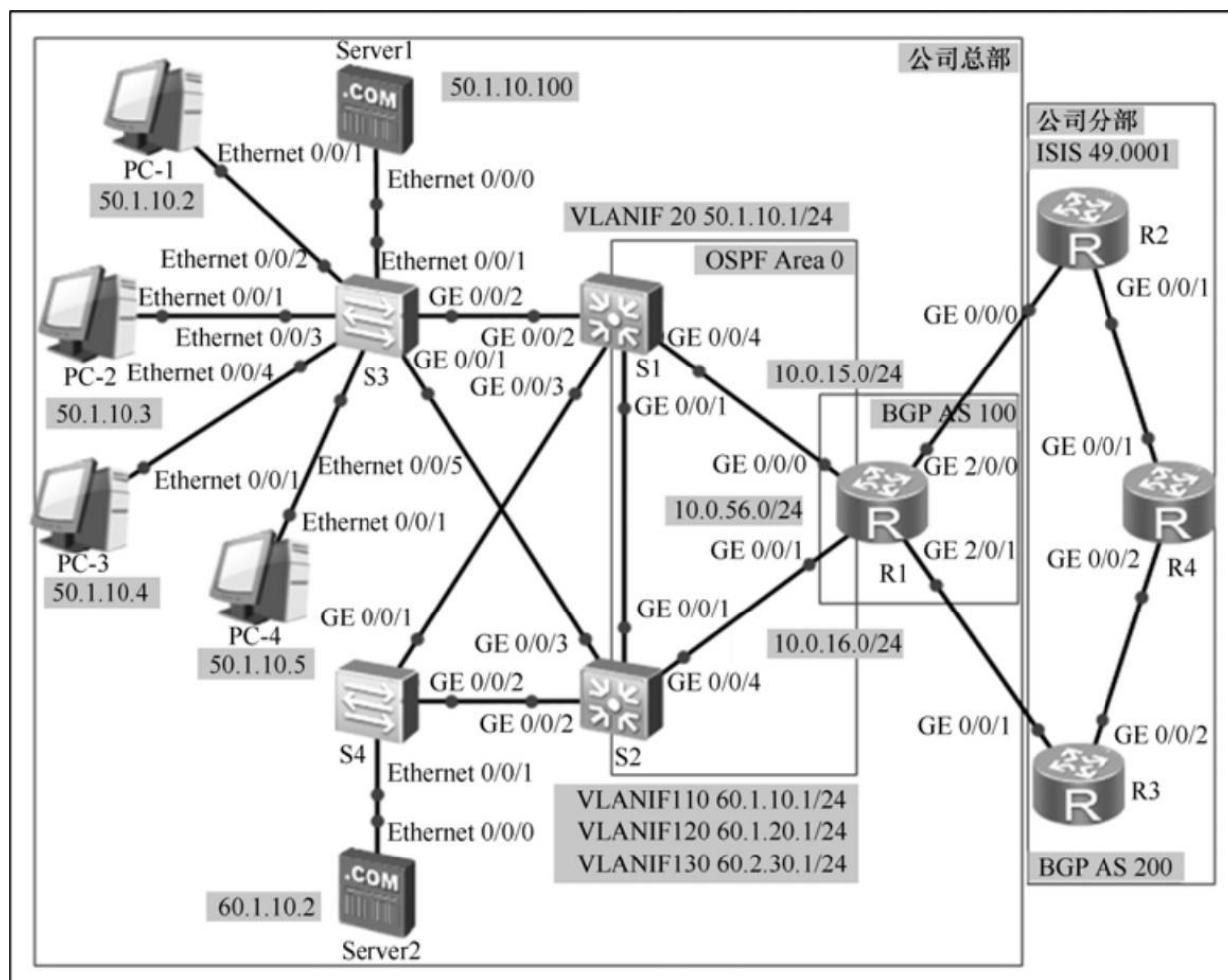


图8-10 综合实验2

实验编址表

表8-4 实验编址

设备	接口	IP 地址	子网掩码	默认网关
R1(AR2220)	GE 0/0/0	10.0.15.1	255.255.255.0	N/A
	GE 0/0/1	10.0.16.1	255.255.255.0	N/A
	GE 2/0/0	10.0.12.1	255.255.255.0	N/A
	GE 2/0/1	10.0.13.1	255.255.255.0	N/A
	Loopback 0	10.0.1.1	255.255.255.255	N/A
R2(AR2220)	GE 0/0/0	10.0.12.2	255.255.255.0	N/A
	GE 0/0/1	10.0.24.2	255.255.255.0	N/A
	Loopback 0	10.0.2.2	255.255.255.255	N/A
R3(AR2220)	GE 0/0/1	10.0.13.3	255.255.255.0	N/A
	GE 0/0/2	10.0.34.3	255.255.255.0	N/A
	Loopback 0	10.0.3.3	255.255.255.255	N/A
R4(AR2220)	GE 0/0/1	10.0.24.4	255.255.255.0	N/A
	GE 0/0/2	10.0.34.4	255.255.255.0	N/A
	Loopback 0	10.0.4.4	255.255.255.255	N/A
	Loopback 1	20.0.4.4	255.255.255.255	N/A
S1(S5700)	VLANIF 51	10.0.15.5	255.255.255.0	N/A
	VLANIF 52	10.0.56.5	255.255.255.0	N/A
	VLANIF 20	50.1.10.1	255.255.255.0	N/A
S2(S5700)	VLANIF 61	10.0.16.6	255.255.255.0	N/A
	VLANIF 62	10.0.56.6	255.255.255.0	N/A
	VLANIF 110	60.1.10.1	255.255.255.0	N/A
	VLANIF 120	60.1.20.1	255.255.255.0	N/A
	VLANIF 130	60.2.30.1	255.255.255.0	N/A

实验步骤

1. 基本配置

根据图8-10和表8-4进行相应的基本配置，并使用ping命令检测R1与R2之间的连通性。

```
<R1>ping -c 1 10.0.12.2
```

```
PING 10.0.12.2: 56 data bytes, press CTRL_C to break
```

```
Reply from 10.0.12.2: bytes=56 Sequence=1 ttl=255 time=10 ms
```

```
--- 10.0.12.2 ping statistics ---
```

```
1 packet(s) transmitted
```

```
1 packet(s) received
```

0.00% packet loss

round-trip min/avg/max = 10/10/10 ms

其余直连网段的连通性测试过程在此省略。

2. 园区网划分VLAN

根据公司网络规划，在所有交换机上都创建VLAN 10、VLAN 20、VLAN 30、VLAN 110、VLAN 120 和VLAN 130。

```
[S1]vlan batch 10 20 30 110 120 130
```

```
[S2]vlan batch 10 20 30 110 120 130
```

```
[S3]vlan batch 10 20 30 110 120 130
```

```
[S4]vlan batch 10 20 30 110 120 130
```

配置连接S1和S3 的端口为Access 端口，允许VLAN 20 通过；配置连接S1 和S4的端口、连接S2 和S3 的端口、连接S2 和S4 的端口为Trunk 端口，允许VLAN 10、VLAN 20、VLAN 30、VLAN 110、VLAN 120 和VLAN 130 通过。

```
[S1]interface GigabitEthernet 0/0/2
```

```
[S1-GigabitEthernet0/0/2]port link-type access
```

```
[S1-GigabitEthernet0/0/2]port default vlan 20
```

```
[S1-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
```

```
[S1-GigabitEthernet0/0/3]port link-type trunk
```

```
[S1-GigabitEthernet0/0/3]port trunk allow-pass vlan 10 20 30 110 120  
130
```

```
[S2]interface GigabitEthernet 0/0/2
```

```
[S2-GigabitEthernet0/0/2]port link-type trunk
```

```
[S2-GigabitEthernet0/0/2]port trunk allow-pass vlan 10 20 30 110 120  
130
```

```
[S2-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
```

```
[S2-GigabitEthernet0/0/3]port link-type trunk
```

```
[S2-GigabitEthernet0/0/3]port trunk allow-pass vlan 10 20 30 110 120
130
```

```
[S3]interface GigabitEthernet 0/0/1
[S3-GigabitEthernet0/0/1]port link-type trunk
[S3-GigabitEthernet0/0/1]port trunk allow-pass vlan 10 20 30 110 120
130
```

```
[S3-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[S3-GigabitEthernet0/0/2]port link-type access
[S3-GigabitEthernet0/0/2]port default vlan 20
[S4]interface GigabitEthernet 0/0/1
[S4-GigabitEthernet0/0/1]port link-type trunk
[S4-GigabitEthernet0/0/1]port trunk allow-pass vlan 10 20 30 110 120
130
```

```
[S4-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
[S4-GigabitEthernet0/0/2]port link-type trunk
[S4-GigabitEthernet0/0/2]port trunk allow-pass vlan 10 20 30 110 120
130
```

另外，在S1上创建VLAN 51 和VLAN 52，在S2上创建VLAN 61 和VLAN 62，并将相应的端口划入这些VLAN。

```
[S1]vlan batch 51 52
[S1]interface GigabitEthernet0/0/1
[S1-GigabitEthernet0/0/1]port link-type access
[S1-GigabitEthernet0/0/1]port default vlan 52
[S1-GigabitEthernet0/0/1]interface GigabitEthernet0/0/4
[S1-GigabitEthernet0/0/4]port link-type access
[S1-GigabitEthernet0/0/4]port default vlan 51
[S2]vlan batch 61 62
```

```
[S2]interface GigabitEthernet0/0/1
[S2-GigabitEthernet0/0/1]port link-type access
[S2-GigabitEthernet0/0/1]port default vlan 62
[S2-GigabitEthernet0/0/1]interface GigabitEthernet0/0/4
[S2-GigabitEthernet0/0/4]port link-type access
[S2-GigabitEthernet0/0/4]port default vlan 61
```

配置完成后，使用命令display vlan 查看VLAN 信息。此处只查看交换机S1 上的VLAN信息，读者可自行查看其他交换机上的VLAN信息。

```
[S1]display vlan
The total number of vlans is : 9
```

```
-----
U:Up;    D:Down;    TG:Tagged;    UT:Untagged;
MP:Vlan-mapping;    ST:Vlan-stacking;
#:ProtocolTransparent-vlan;    *:Management-vlan;
-----
```

```
-----
VID  Type  Ports
-----
```

```
-----
1    common  UT:GE0/0/3(U)    GE0/0/5(D)    GE0/0/6(D)
GE0/0/7(D)
GE0/0/8(D)  GE0/0/9(D)  GE0/0/10(D)  GE0/0/11(D)
GE0/0/12(D)  GE0/0/13(D)  GE0/0/14(D)  GE0/0/15(D)
GE0/0/16(D)  GE0/0/17(D)  GE0/0/18(D)  GE0/0/19(D)
GE0/0/20(D)  GE0/0/21(D)  GE0/0/22(D)  GE0/0/23(D)
```

GE0/0/24(D)

10 common TG:GE0/0/3(U)

20 common UT:GE0/0/2(U)

TG:GE0/0/3(U)

30 common TG:GE0/0/3(U)

51 common UT:GE0/0/4(U)

52 common UT:GE0/0/1(U)

110 common TG:GE0/0/3(U)

120 common TG:GE0/0/3(U)

130 common TG:GE0/0/3(U)

VID Status Property MAC-LRN Statistics Description

1	enable	default	enable	disable	VLAN 0001
10	enable	default	enable	disable	VLAN 0010
20	enable	default	enable	disable	VLAN 0020
30	enable	default	enable	disable	VLAN 0030
51	enable	default	enable	disable	VLAN 0051
52	enable	default	enable	disable	VLAN 0052
110	enable	default	enable	disable	VLAN 0110
120	enable	default	enable	disable	VLAN 0120
130	enable	default	enable	disable	VLAN 0130

可以看到，S1 上除了缺省VLAN 1 和为了与R1、S2 实现三层通信而配置的VLAN 51、VLAN 52 之外，还有VLAN 10、VLAN 20、VLAN 30、VLAN 110、VLAN 120 和VLAN 130，且状态都为Enable，表明VLAN 创建成功。另外，所有Trunk端口都以Tagged模式加入进了相应的VLAN。

在交换机S1上为VLAN 20 创建VLANIF 接口，在交换机S2 上为VLAN 110、120、130创建VLANIF接口。

```
[S1]interface Vlanif20
[S1-Vlanif20]ip address 50.1.10.1 24
[S2]interface Vlanif110
[S2-Vlanif110]ip address 60.1.10.1 24
[S2-Vlanif110]interface Vlanif120
[S2-Vlanif120]ip address 60.1.20.1 24
[S2-Vlanif120]interface Vlanif130
[S2-Vlanif130]ip address 60.2.30.1 24
```

配置完成后，在S1上查看VLANIF接口状态。

```
[S1]display ip interface brief
*down: administratively down
^down: standby
(l): loopback
(s): spoofing
```

The number of interface that is UP in Physical is 5

The number of interface that is DOWN in Physical is 1

The number of interface that is UP in Protocol is 4

The number of interface that is DOWN in Protocol is 2

Interface	IP Address/Mask	Physical	Protocol
MEth0/0/1	unassigned	down	down
NULL0	unassigned	up	up(s)
Vlanif1	unassigned	up	down
Vlanif20	50.1.10.1/24	up	up
Vlanif51	10.0.15.5/24	up	up
Vlanif52	10.0.56.5/24	up	up

可以看到，S1上的VLANIF接口已创建成功，接口的物理状态和协议状态均为UP。

在S2上查看VLANIF接口状态。

[S2]display ip interface brief

*down: administratively down

^down: standby

(l): loopback

(s): spoofing

The number of interface that is UP in Physical is 7

The number of interface that is DOWN in Physical is 1

The number of interface that is UP in Protocol is 6

The number of interface that is DOWN in Protocol is 2

Interface	IP Address/Mask	Physical	Protocol
MEth0/0/1	unassigned	down	down
NULL0	unassigned	up	up(s)
Vlanif1	unassigned	up	down
Vlanif61	10.0.16.6/24	up	up
Vlanif62	10.0.56.6/24	up	up
Vlanif110	60.1.10.1/24	up	up
Vlanif120	60.1.20.1/24	up	up
Vlanif130	60.2.30.1/24	up	up

可以看到，S2上的VLANIF接口也已创建成功，接口的物理状态和协议状态均为UP。

Server2 是属于VLAN 110 的，并且要求Server2 无论从哪个端口接入S4 都必须属于VLAN 110。为此，基于MAC 地址将Server 2 添加进VLAN 110。

[S4]vlan 110

```
[S4-vlan110]mac-vlan mac-address 5489-98cf-220f
```

```
[S4-vlan110]quit
```

```
[S4]interface Ethernet 0/0/1
```

```
[S4-Ethernet0/0/1]port hybrid untagged vlan all
```

```
[S4-Ethernet0/0/1]mac-vlan enable
```

配置完成后，在S4上查看MAC VLAN信息。

```
[S4]display mac-vlan vlan 3
```

```
-----  
MAC Address      MASK      VLAN  Priority  
-----
```

```
-----  
5489-98cf-220f      ffff-ffff-ffff      3      0
```

```
Total MAC VLAN address count: 1
```

可以看到，Server2的MAC地址在VLAN 3当中。

为了防止Server2下电后，S4另外的端口会学习到与Server2同样的MAC地址，或者某些非法用户从其他端口假冒Server2的MAC地址发送数据报文而发生MAC地址漂移，导致正常用户不能与Server2正常通信，可以在S4的Ethernet 0/0/1端口下使用命令mac-learning priority调整端口MAC地址学习的优先级（默认优先级的值为0）。

```
[S4]interface Ethernet 0/0/1
```

```
[S4-Ethernet0/0/1]mac-learning priority 2
```

提高交换机S4的Ethernet 0/0/1端口MAC地址学习的优先级后，可以有效防止非法用户攻击，S4将优先认可此端口所连接设备的MAC地址。

公司总部园区中，公司员工和公司客户都可以访问公司的服务器，公司内部员工之间也可以互相交流，但与公司客户之间是隔离

的，不能够互相访问。公司客户与客户之间不能互访，客户与公司员工也不能互访。这样的需求可以通过Mux VLAN 来实现：在交换机S3上，配置Server1 所在的VLAN 20 为Principal VLAN，配置公司员工PC-1 和PC-2 所在的VLAN 10 为Group VLAN，配置公司客户PC-3 和PC-4 所在的VLAN 30 为Separate VLAN。

```
[S3]vlan 20
[S3-vlan20]mux-vlan
[S3-vlan20]subordinate group 10
[S3-vlan20]subordinate separate 30
[S3-vlan20]interface Ethernet 0/0/1
[S3-Ethernet0/0/1]port link-type access
[S3-Ethernet0/0/1]port default vlan 20
[S3-Ethernet0/0/1]port mux-vlan enable
[S3-Ethernet0/0/1]interface Ethernet 0/0/2
[S3-Ethernet0/0/2]port link-type access
[S3-Ethernet0/0/2]port default vlan 10
[S3-Ethernet0/0/2]port mux-vlan enable
[S3-Ethernet0/0/2]interface Ethernet 0/0/3
[S3-Ethernet0/0/3]port link-type access
[S3-Ethernet0/0/3]port default vlan 10
[S3-Ethernet0/0/3]port mux-vlan enable
[S3-Ethernet0/0/3]interface Ethernet 0/0/4
[S3-Ethernet0/0/4]port link-type access
[S3-Ethernet0/0/4]port default vlan 30
[S3-Ethernet0/0/4]port mux-vlan enable
[S3-Ethernet0/0/4]interface Ethernet 0/0/5
[S3-Ethernet0/0/5]port link-type access
```

```
[S3-Ethernet0/0/5]port default vlan 30
[S3-Ethernet0/0/5]port mux-vlan enable
[S3-Ethernet0/0/5]interface GigabitEthernet 0/0/2
[S3-GigabitEthernet0/0/2]port link-type access
[S3-GigabitEthernet0/0/2]port default vlan 20
[S3-GigabitEthernet0/0/2]port mux-vlan enable
配置完成后，在交换机S3 上查看Mux VLAN 信息。
```

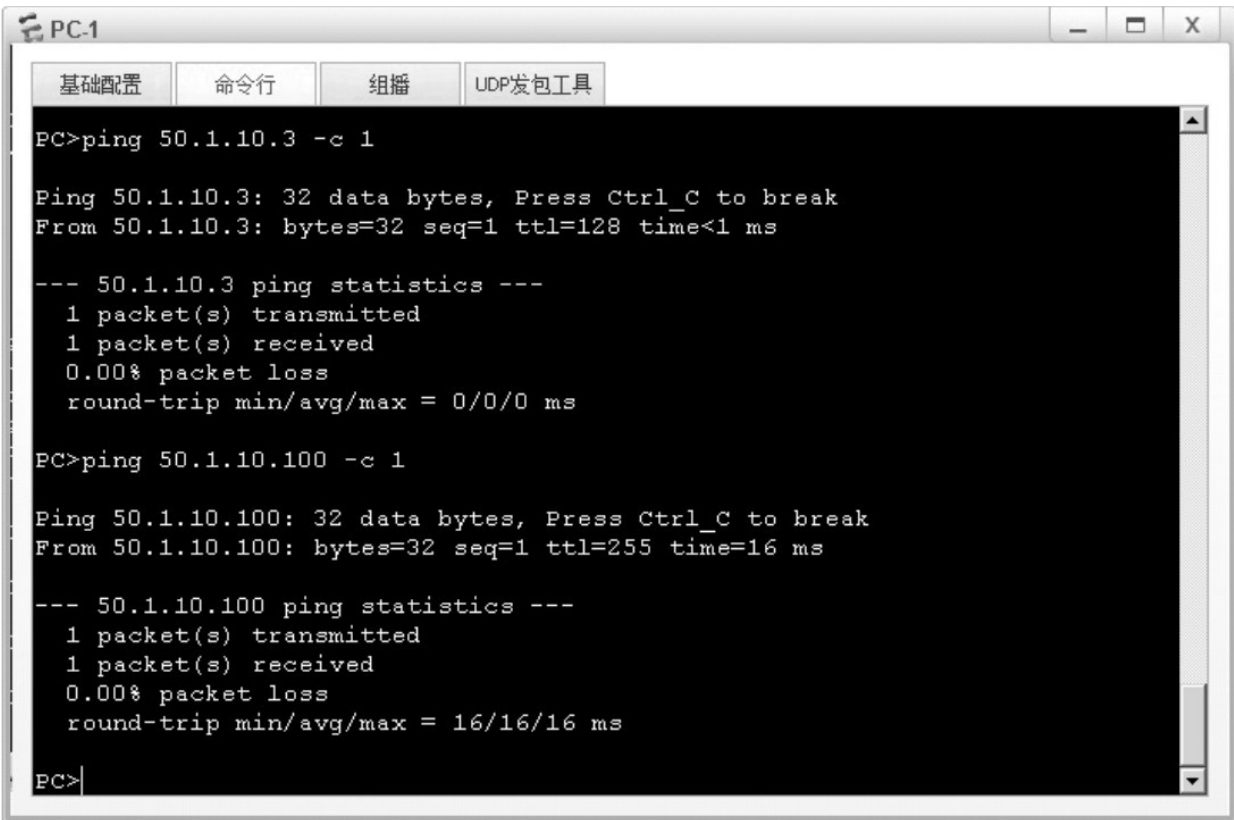
```
<S3>display mux-vlan
```

Principal	Subordinate	Type	Interface

20	-	principal	Ethernet0/0/1GigabitEthernet0/0/2
20	30	separate	Ethernet0/0/4Ethernet0/0/5
20	10	group	Ethernet0/0/2Ethernet0/0/3

可以看到，Server1 所在的VLAN 20 为Principal VLAN，可以与Mux VLAN 内所有的VLAN进行通信，公司员工所在的VLAN 10为Group VLAN，而公司客户所在的VLAN 30 为Separate VLAN。

在PC-1 上使用Ping 命令测试与PC-2、Server 1 之间的连通性，如图8-11 所示。



The screenshot shows a terminal window titled "PC-1" with four tabs: "基础配置", "命令行", "组播", and "UDP发包工具". The "命令行" tab is active. The terminal displays the following text:

```
PC>ping 50.1.10.3 -c 1

Ping 50.1.10.3: 32 data bytes, Press Ctrl_C to break
From 50.1.10.3: bytes=32 seq=1 ttl=128 time<1 ms

--- 50.1.10.3 ping statistics ---
 1 packet(s) transmitted
 1 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 0/0/0 ms

PC>ping 50.1.10.100 -c 1

Ping 50.1.10.100: 32 data bytes, Press Ctrl_C to break
From 50.1.10.100: bytes=32 seq=1 ttl=255 time=16 ms

--- 50.1.10.100 ping statistics ---
 1 packet(s) transmitted
 1 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 16/16/16 ms

PC>
```

图8-11 PC-1与PC-2、Server1之间的连通性测试

从图8-11中可以看到，PC-1能够与PC-2和Server1正常通信。

在PC3上使用Ping命令测试与PC-4、Server1之间的连通性，如图8-12所示。

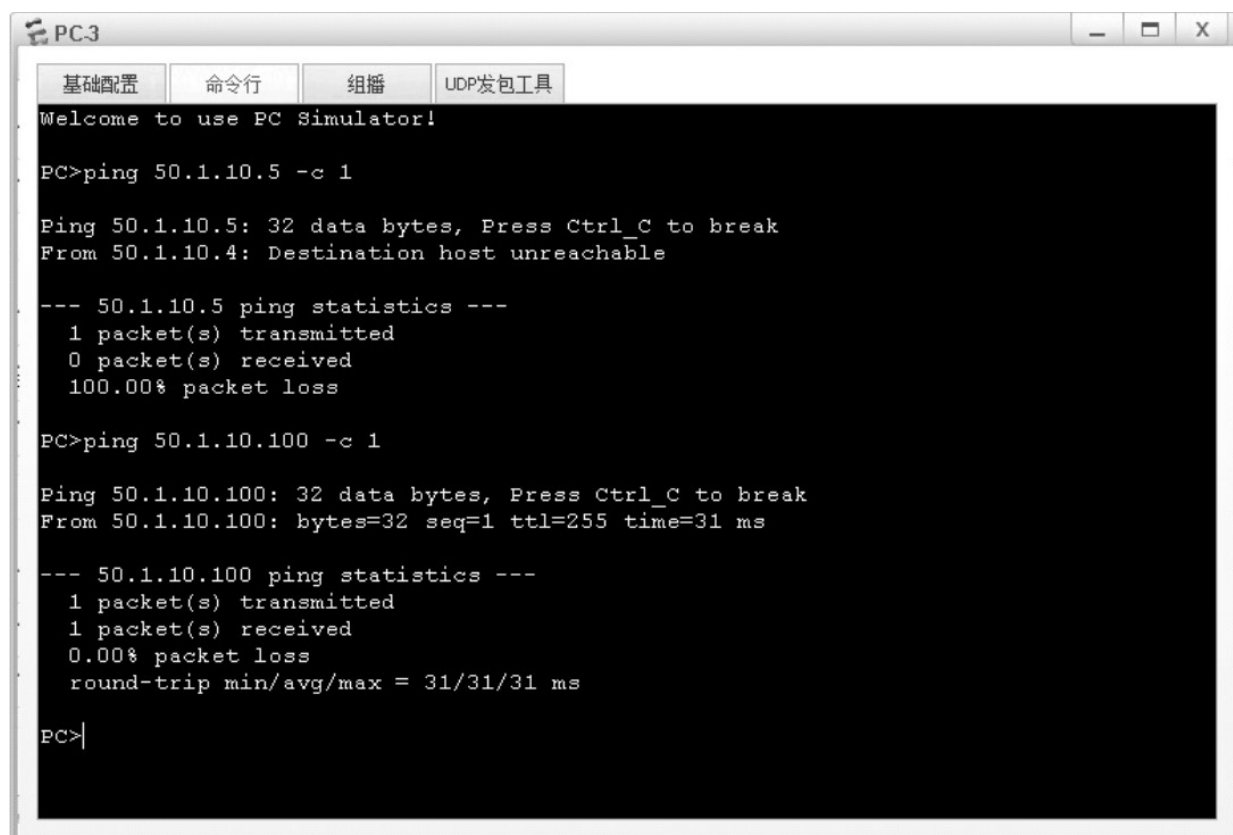


图8-12 PC-3与PC-4、Server1之间的连通性测试

从图8-12中可以看到，PC-3只能与Server1通信，与同是公司客户的PC-4不能进行通信。

3. 配置RSTP协议

为了防止公司总部园区网络的二层环路，配置所有交换机工作在RSTP模式。

```
[S1]stp mode rstp
```

```
[S2]stp mode rstp
```

```
[S3]stp mode rstp
```

```
[S4]stp mode rstp
```

配置S1为RSTP的根交换机，S2为备份根交换机。

```
[S1]stp root primary
```

```
[S2]stp root secondary
```

配置完成后，在S1、S2上执行display stp命令。

[S1]display stp

-----[CIST Global Info][Mode RSTP]-----

CIST Bridge :0 .4c1f-ccda-4975

Config Times :Hello 2s MaxAge 20s FwDly 15s MaxHop

20

Active Times :Hello 2s MaxAge 20s FwDly 15s MaxHop

20

CIST Root/ERPC :0 .4c1f-ccda-4975 / 0

CIST RegRoot/IRPC :0 .4c1f-ccda-4975 / 0

CIST RootPortId :0.0

BPDU-Protection :Disabled

CIST Root Type :Primary root

TC or TCN received :24

TC count per hello :0

STP Converge Mode :Normal

Time since last TC :0 days 0h:0m:27s

Number of TC :22

Last TC occurred :GigabitEthernet0/0/3

[S2]display stp

-----[CIST Global Info][Mode RSTP]-----

CIST Bridge :4096 .4c1f-cc4f-1b95

Config Times :Hello 2s MaxAge 20s FwDly 15s MaxHop

20

Active Times :Hello 2s MaxAge 20s FwDly 15s MaxHop

20

CIST Root/ERPC :0 .4c1f-ccda-4975 / 1

CIST RegRoot/IRPC :4096 .4c1f-cc4f-1b95 / 0

```

CIST RootPortId      :128.1
BPDU-Protection      :Disabled
CIST Root Type       :Secondary root
TC or TCN received   :24
TC count per hello    :0
STP Converge Mode    :Normal
Time since last TC    :0 days 0h:1m:10s
Number of TC         :24
Last TC occurred      :GigabitEthernet0/0/1

```

从显示信息可以看到，S1为根交换机，优先级的值为0，S2为备份根交换机，优先级的值为4096，工作模式为RSTP。

为了提高网络的稳定性，可以配置RSTP的根保护功能，使得无论网络发生什么变化，根交换机的角色都不会改变。根保护是指定端口的特性，当端口角色是指定端口时，根保护功能才能生效。

在S2的指定端口GE 0/0/2和GE 0/0/3上启用根保护特性。

```

[S2]interface GigabitEthernet 0/0/2
[S2-GigabitEthernet0/0/2]stp root-protection
[S2-GigabitEthernet0/0/2]interface GigabitEthernet 0/0/3
[S2-GigabitEthernet0/0/3]stp root-protection

```

配置完成后，在S2上查看生成树端口的保护状态。

```

[S2]display stp brief

```

MSTID	Port	Role	STP State	Protection
0	GigabitEthernet0/0/1	ROOT	FORWARDING	NONE
0	GigabitEthernet0/0/2	DESI	FORWARDING	ROOT
0	GigabitEthernet0/0/3	DESI	FORWARDING	ROOT
0	GigabitEthernet0/0/4	DESI	FORWARDING	NONE

可以观察到，S2 的GE 0/0/2和GE 0/0/3 端口的保护模式为根保护，这样一来，就算从这两个端口接收到桥ID更优的BPDU，也不会导致根交换机发生改变。由于S2的GE 0/0/4 端口连接的是上行路由器，S3 和S4 的指定端口连接的是终端或服务器，因此这些端口无需配置根保护功能。

如果有攻击者伪造拓扑变化BPDU报文来恶意攻击二层网络，则交换机在短时间内会收到大量的拓扑变化BPDU报文，这会给交换机的处理工作造成很大的负担。为此，可以通过配置TC-BPDU保护功能来解决这个问题。

```
[S1]stp tc-protection
[S1]stp tc-protection threshold 2
[S2]stp tc-protection
[S2]stp tc-protection threshold 2
[S3]stp tc-protection
[S3]stp tc-protection threshold 2
[S4]stp tc-protection
[S4]stp tc-protection threshold 2
```

上面所进行的配置的含义是：交换机在单位时间（与RSTP Hello时间间隔一致）内，允许在收到TC-BPDU报文后立即进行地址表项删除操作的最大次数为两次。

为了加快收敛速度，将交换机S3上连接PC和Server1的端口，以及S4的Ethernet 0/0/1端口配置成边缘端口。

```
[S3]interface Ethernet 0/0/1
[S3-Ethernet0/0/1]stp edged-port enable
[S3-Ethernet0/0/1]interface Ethernet 0/0/2
[S3-Ethernet0/0/2]stp edged-port enable
[S3-Ethernet0/0/2]interface Ethernet 0/0/3
```

```
[S3-Ethernet0/0/3]stp edged-port enable
[S3-Ethernet0/0/3]interface Ethernet 0/0/4
[S3-Ethernet0/0/4]stp edged-port enable
[S3-Ethernet0/0/4]interface Ethernet 0/0/5
[S3-Ethernet0/0/5]stp edged-port enable
[S4]interface Ethernet 0/0/1
[S4-Ethernet0/0/1]stp edged-port enable
```

至此，交换机的配置工作已经基本完成。

4．配置OSPF路由协议

在R1、S1、S2上配置OSPF协议。

```
[R1]router id 10.0.1.1
[R1]ospf 1
[R1-ospf-1]area 0
[R1-ospf-1-area-0.0.0.0]network 10.0.15.0 0.0.0.255
[R1-ospf-1-area-0.0.0.0]network 10.0.16.0 0.0.0.255
[R1-ospf-1-area-0.0.0.0]network 10.0.1.1 0.0.0.0
[S1]ospf 1
[S1-ospf-1]area 0
[S1-ospf-1-area-0.0.0.0]network 10.0.15.0 0.0.0.255
[S1-ospf-1-area-0.0.0.0]network 10.0.56.0 0.0.0.255
[S2]ospf 1
[S2-ospf-1]area 0
[S2-ospf-1-area-0.0.0.0]network 10.0.16.0 0.0.0.255
[S2-ospf-1-area-0.0.0.0]network 10.0.56.0 0.0.0.255
```

配置完成后，在R1上查看OSPF邻居信息。

```
<R1>display ospf peer brief
```

```
OSPF Process 1 with Router ID 10.0.1.1
```


Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/0	10.0.56.5	Full
0.0.0.0	GigabitEthernet0/0/1	10.0.56.6	Full

可以看到，R1与S1和S2已经成功建立起了OSPF邻接关系。
查看R1的IP路由表。

<R1>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 18 Routes : 19

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.1.1/32	Direct	0	0	D	127.0.0.1 LoopBack0
.....					
10.0.16.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.56.0/24	OSPF	10	2	D	10.0.15.5
GigabitEthernet0/0/0					
OSPF	10	2	D	10.0.16.6	GigabitEthernet0/0/1
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					

.....

可以看到，R1已经接收到了S1与S2之间网段的路由信息。

将S1 上的VLANIF 20 所对应的网段作为外部路由引入OSPF 进程，并进行路由聚合；将S2 上的VLANIF 110、VLANIF 120、VLANIF 130 所对应的网段作为外部路由引入OSPF 进程，并对VLANIF 110 和VLANIF 120 所对应的网段进行路由聚合。

```
[S1]ospf 1
```

```
[S1-ospf-1]import-route direct
```

```
[S1-ospf-1]asbr-summary 50.1.0.0 255.255.0.0
```

```
[S2]ospf 1
```

```
[S2-ospf-1]import-route direct
```

```
[S2-ospf-1]asbr-summary 60.1.0.0 255.255.224.0
```

在R1上查看IP路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 20 Routes : 21

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.1.1/32	Direct	0 0	D	127.0.0.1	LoopBack0
.....					
10.0.56.0/24	OSPF	10	2	D	10.0.15.5
GigabitEthernet0/0/0					
OSPF	10	2	D	10.0.16.6	GigabitEthernet0/0/1

50.1.0.0/16	O_ASE	150	2	D	10.0.15.5
GigabitEthernet0/0/0					
60.1.0.0/19	O_ASE	150	2	D	10.0.16.6
GigabitEthernet0/0/1					
60.2.30.0/24	O_ASE	150	1	D	10.0.16.6
GigabitEthernet0/0/1					
.....					

可以看到，R1 已经接收到了用户网段的路由，OSPF 外部路由在路由表中表示为O_ASE，聚合后的路由的掩码分别为16位和19位。

为了提高网络安全性，R1、S1、S2 需要相互通过认证之后才能交换路由信息。认证功能的配置如下。

```
[R1]ospf 1
[R1-ospf-1]area 0
[R1-ospf-1-area-0.0.0.0]authentication-mode simple cipher huawei
[S1]ospf 1
[S1-ospf-1]area 0
[S1-ospf-1-area-0.0.0.0]authentication-mode simple cipher huawei
[S2]ospf 1
[S2-ospf-1]area 0
[S2-ospf-1-area-0.0.0.0]authentication-mode simple cipher huawei
```

为了加快OSPF的收敛速度，需要配置OSPF扩展功能，即网络拓扑一旦发生改变，设备需立刻泛洪新的LSA，接收到新LSA的设备立即进行路由计算。

```
[R1]ospf 1
[R1-ospf-1]lsa-originate-interval 0
[R1-ospf-1]lsa-arrival-interval 0
[S1]ospf 1
```

```
[S1-ospf-1]lsa-originate-interval 0
```

```
[S1-ospf-1]lsa-arrival-interval 0
```

```
[S2]ospf 1
```

```
[S2-ospf-1]lsa-originate-interval 0
```

```
[S2-ospf-1]lsa-arrival-interval 0
```

5 . 配置IS-IS路由协议

在公司分部的路由器上配置IS-IS协议。

```
[R2]isis 1
```

```
[R2-isis-1]network-entity 49.0001.0100.0000.2002.00
```

```
[R2-isis-1]interface GigabitEthernet 0/0/1
```

```
[R2-GigabitEthernet0/0/1]isis enable
```

```
[R2-GigabitEthernet0/0/1]interface Loopback 0
```

```
[R2-LoopBack0]isis enable
```

```
[R3]isis 1
```

```
[R3-isis-1]network-entity 49.0001.0100.0000.3003.00
```

```
[R3-isis-1]interface GigabitEthernet 0/0/2
```

```
[R3-GigabitEthernet0/0/2]isis enable
```

```
[R3-GigabitEthernet0/0/2]interface Loopback 0
```

```
[R3-LoopBack0]isis enable
```

```
[R4]isis 1
```

```
[R4-isis-1]network-entity 49.0001.0100.0000.4004.00
```

```
[R4-isis-1]interface GigabitEthernet 0/0/1
```

```
[R4-GigabitEthernet0/0/1]isis enable
```

```
[R4-GigabitEthernet0/0/1]interface GigabitEthernet 0/0/2
```

```
[R4-GigabitEthernet0/0/2]isis enable
```

```
[R4-GigabitEthernet0/0/2]interface Loopback 0
```

```
[R4-LoopBack0]isis enable
```

[R4-LoopBack0]interface Loopback 1

[R4-LoopBack1]isis enable

配置完成后，在R4上查看IS-IS邻居信息。

<R4>display isis peer

Peer information for ISIS(1)

System Id		Interface	Circuit Id	State	HoldTime
Type	PRI				

L1(L1L2)	64	0100.0000.2002	GE0/0/1	0100.0000.2002.01	Up 9s
L2(L1L2)	64	0100.0000.2002	GE0/0/1	0100.0000.2002.01	Up 9s
L1(L1L2)	64	0100.0000.3003	GE0/0/2	0100.0000.3003.01	Up 9s
L2(L1L2)	64	0100.0000.3003	GE0/0/2	0100.0000.3003.01	Up 9s

Total Peer(s): 4

可以看到，R4与R2和R3已经建立了正常的IS-IS邻居关系。

在R4上查看IS-IS链路状态数据库。

<R4>display isis lsdb

Database information for ISIS(1)

Level-1 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0100.0000.2002.00-00	0x0000000c	0xa435	773	86
0/0/0				
0100.0000.2002.01-00	0x00000001	0xb64d	773	55
0/0/0				
0100.0000.3003.00-00	0x0000000c	0xacf1	964	86
0/0/0				
0100.0000.3003.01-00	0x00000005	0x28b5	964	55
0/0/0				
0100.0000.4004.00-00*	0x00000012	0x56c3	821	129
0/0/0				

Total LSP(s): 5

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self
LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length

ATT/P/OL

0100.0000.2002.00-00	0x00000010	0xbed	820	134
0/0/0				
0100.0000.2002.01-00	0x00000001	0xb64d	773	55
0/0/0				
0100.0000.3003.00-00	0x00000016	0xe4de	964	134
0/0/0				

0100.0000.3003.01-00 0x00000005 0x28b5 964 55
0/0/0

0100.0000.4004.00-00* 0x00000017 0x6364 825 153
0/0/0

Total LSP(s): 5

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self
LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R4维护了Level-1和Level-2两个链路状态数据库。

为了减少 IS-IS 邻居关系数量和精简链路状态数据库，配置 R2、
R3、R4 为 IS-IS Level-2路由器。

[R2]isis 1

[R2-isis-1]is-level level-2

[R3]isis 1

[R3-isis-1]is-level level-2

[R4]isis 1

[R4-isis-1]is-level level-2

重新在R4上查看IS-IS邻居信息。

<R4>display isis peer

Peer information for ISIS(1)

System Id	Interface	Circuit Id	State	HoldTime
Type	PRI			

0100.0000.2002	GE0/0/1	0100.0000.2002.01	Up	9s L2

0100.0000.3003 GE0/0/2 0100.0000.3003.01 Up 9s L2
64

Total Peer(s): 2

可以看到，R4现在只需要维护两个Level-2邻居关系，Level-1的邻居关系已经不存在了。

重新查看R4的IS-IS链路状态数据库。

[R4]display isis lsdb

Database information for ISIS(1)

Level-2 Link State Database

LSPID	Seq Num	Checksum	Holdtime	Length
ATT/P/OL				

0100.0000.2002.00-00	0x00000013	0xd4fd	667	86
0/0/0				
0100.0000.2002.01-00	0x00000002	0xb44e	667	55
0/0/0				
0100.0000.3003.00-00	0x0000001a	0x4a46	667	86
0/0/0				
0100.0000.3003.01-00	0x00000007	0x24b7	667	55
0/0/0				
0100.0000.4004.00-00*	0x0000001e	0x5cb1	688	129
0/0/0				

Total LSP(s): 5

*(In TLV)-Leaking Route, *(By LSPID)-Self LSP, +-Self LSP(Extended),

ATT-Attached, P-Partition, OL-Overload

可以看到，R4现在只需维护一个Level-2的链路状态数据库。

查看R4的IP路由表，读者可自行查看其他路由器的IP路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 14 Routes : 14						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
10.0.2.2/32	ISIS-L2	15	10	D	10.0.24.2	
GigabitEthernet0/0/1						
10.0.3.3/32	ISIS-L2	15	10	D	10.0.34.3	
GigabitEthernet0/0/2						
10.0.4.4/32	Direct	0 0	D	127.0.0.1	LoopBack0	
.....						

可以看到，R4 已经通过IS-IS 协议获得到R2 和R3 的Loopback 0 网段的路由。

6．配置BGP路由协议

在R1、R2、R3、R4上配置BGP协议，EBGP邻居关系采用直连物理接口来建立，IBGP 邻居关系采用Loopback 0 接口来建立。

[R1]bgp 100

[R1-bgp]peer 10.0.12.2 as-number 200

[R1-bgp]peer 10.0.13.3 as-number 200

[R2]bgp 200

[R2-bgp]peer 10.0.12.1 as-number 100

```
[R2-bgp]peer 10.0.4.4 as-number 200
[R2-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R3]bgp 200
[R3-bgp]peer 10.0.13.1 as-number 100
[R3-bgp]peer 10.0.4.4 as-number 200
[R3-bgp]peer 10.0.4.4 connect-interface LoopBack 0
[R4]bgp 200
[R4-bgp]peer 10.0.2.2 as-number 200
[R4-bgp]peer 10.0.2.2 connect-interface LoopBack 0
[R4-bgp]peer 10.0.3.3 as-number 200
[R4-bgp]peer 10.0.3.3 connect-interface LoopBack 0
```

配置完成后，在R1上查看BGP邻居信息。

```
<R1>display bgp peer
```

```
BGP local router ID : 10.0.1.1
```

```
Local AS number : 100
```

```
Total number of peers : 2      Peers in established state : 2
```

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
10.0.12.2	4	200	7	8	0	00:05:41
Established	0					
10.0.13.3	4	200	5	6	0	00:03:00
Established	0					

可以看到，R1分别与R2和R3建立了跨AS的EBGP邻居关系。

在R4上查看BGP邻居信息。

```
<R4>display bgp peer
```

```
BGP local router ID : 10.0.24.4
```

```
Local AS number : 200
```

Total number of peers : 2 Peers in established state : 2

Peer	V	AS	MsgRcvd	MsgSent	OutQ	Up/Down
State	PrefRcv					
10.0.2.2	4	200	5	5	0	00:03:55
Established	0					
10.0.3.3	4	200	5	5	0	00:03:27
Established	0					

可以看到，R4分别与R2和R3建立了IBGP邻居关系。

为了让公司分部知道公司总部网络的路由，在R1上将OSPF引入BGP进程。

```
[R1]bgp 100
```

```
[R1-bgp]import-route ospf 1
```

在R2上查看IP路由表。

```
<R2>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 22 Routes : 22

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
10.0.1.1/32	EBGP	255	0	D	10.0.12.1
GigabitEthernet0/0/0					
10.0.2.2/32	Direct	0 0	D	127.0.0.1	LoopBack0
10.0.3.3/32	ISIS-L2	15	20	D	10.0.24.4
GigabitEthernet0/0/1					

10.0.4.4/32	ISIS-L2	15	10	D	10.0.24.4
GigabitEthernet0/0/1					
10.0.12.0/24	Direct	0	0	D	10.0.12.2
GigabitEthernet0/0/0					
10.0.12.2/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.12.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/0					
10.0.15.0/24	EBGP	255	0	D	10.0.12.1
GigabitEthernet0/0/0					
10.0.16.0/24	EBGP	255	0	D	10.0.12.1
GigabitEthernet0/0/0					
10.0.24.0/24	Direct	0	0	D	10.0.24.2
GigabitEthernet0/0/1					
10.0.24.2/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.24.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.34.0/24	ISIS-L2	15	20	D	10.0.24.4
GigabitEthernet0/0/1					
10.0.56.0/24	EBGP	255	2	D	10.0.12.1
GigabitEthernet0/0/0					
20.0.4.4/32	ISIS-L2	15	10	D	10.0.24.4
GigabitEthernet0/0/1					
50.1.0.0/16	EBGP	255	2	D	10.0.12.1
GigabitEthernet0/0/0					

60.1.0.0/19	EBGP	255	2	D	10.0.12.1
GigabitEthernet0/0/0					
60.2.30.0/24	EBGP	255	1	D	10.0.12.1
GigabitEthernet0/0/0					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R2的IP路由表中拥有通过EBGP接收到的、下一跳为R1的总部网络路由。

在R3上查看IP路由表。

```
<R3>display ip routing-table
```

Route Flags: R - relay, D - download to fib

```

-----
Routing Tables: Public
Destinations : 22  Routes : 22

Destination/Mask    Proto    Pre  Cost    Flags    NextHop
-----
Interface
10.0.1.1/32         EBGP      255   0       D        10.0.13.1
GigabitEthernet0/0/1
10.0.2.2/32         ISIS-L2   15    20      D        10.0.34.4
GigabitEthernet0/0/2

```

10.0.3.3/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.4.4/32		ISIS-L2	15	10	D	10.0.34.4
GigabitEthernet0/0/2						
10.0.13.0/24	Direct		0	0	D	10.0.13.3
GigabitEthernet0/0/1						
10.0.13.3/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.13.255/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.15.0/24	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.16.0/24	EBGP		255	0	D	10.0.13.1
GigabitEthernet0/0/1						
10.0.24.0/24		ISIS-L2	15	20	D	10.0.34.4
GigabitEthernet0/0/2						
10.0.34.0/24	Direct		0	0	D	10.0.34.3
GigabitEthernet0/0/2						
10.0.34.3/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/2						
10.0.34.255/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/2						
10.0.56.0/24	EBGP		255	2	D	10.0.13.1
GigabitEthernet0/0/1						
20.0.4.4/32		ISIS-L2	15	10	D	10.0.34.4
GigabitEthernet0/0/2						
50.1.0.0/16	EBGP		255	2	D	10.0.13.1
GigabitEthernet0/0/1						

60.1.0.0/19	EBGP	255	2	D	10.0.13.1
GigabitEthernet0/0/1					
60.2.30.0/24	EBGP	255	1	D	10.0.13.1
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R3的IP路由表中拥有通过EBGP接收到的、下一跳为R1的总部网络路由。

在R4上查看IP路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

```

-----
Routing Tables: Public
Destinations : 14  Routes : 14

Destination/Mask    Proto    Pre  Cost    Flags    NextHop
-----
Interface
10.0.2.2/32         ISIS-L2   15   10      D        10.0.24.2
GigabitEthernet0/0/1
10.0.3.3/32         ISIS-L2   15   10      D        10.0.34.3
GigabitEthernet0/0/2

```

10.0.4.4/32	Direct	0	0	D	127.0.0.1	LoopBack0
10.0.24.0/24	Direct		0	0	D	10.0.24.4
GigabitEthernet0/0/1						
10.0.24.4/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.24.255/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.34.0/24	Direct		0	0	D	10.0.34.4
GigabitEthernet0/0/2						
10.0.34.4/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/2						
10.0.34.255/32	Direct		0	0	D	127.0.0.1
GigabitEthernet0/0/2						
20.0.4.4/32	Direct	0	0	D	127.0.0.1	LoopBack1
127.0.0.0/8	Direct		0	0	D	127.0.0.1
InLoopBack0						
127.0.0.1/32	Direct		0	0	D	127.0.0.1
InLoopBack0						
127.255.255.255/32	Direct		0	0	D	127.0.0.1
InLoopBack0						
255.255.255.255/32	Direct		0	0	D	127.0.0.1
InLoopBack0						

可以看到，R4的IP路由表中没有任何关于总部网络的BGP路由条目。

查看R4的BGP路由表。

```
<R4>display bgp routing-table
```

```
BGP Local router ID is 10.0.24.4
```


Status codes:*- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 14

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
i	10.0.1.1/32	10.0.12.1	0	100	0	100?
i		10.0.13.1	0	100	0	100?
i	10.0.15.0/24	10.0.12.1	0	100	0	100?
i		10.0.13.1	0	100	0	100?
i	10.0.16.0/24	10.0.12.1	0	100	0	100?
i		10.0.13.1	0	100	0	100?
i	10.0.56.0/24	10.0.12.1	2	100	0	100?
i		10.0.13.1	2	100	0	100?
i	50.1.0.0/16	10.0.12.1	2	100	0	100?
i		10.0.13.1	2	100	0	100?
i	60.1.0.0/19	10.0.12.1	2	100	0	100?
i		10.0.13.1	2	100	0	100?
i	60.2.30.0/24	10.0.12.1	1	100	0	100?
i		10.0.13.1	1	100	0	100?

可以看到，R4的BGP路由表中存在从R2和R3传递过来的总部路由，但是路由条目前没有“*”，表示是无效路由，其原因是路由的下一跳不可达。根据BGP的路由传递原则，BGP路由在传递给IBGP邻居时是不会改变路由条目的下一跳的，所以R4接收到的总部路由的下一跳都为R1，而R4的IP路由表中没有去往R1的路由条目。下面的配置给出了解决这个问题的方法。

```
[R2]bgp 200
```

```
[R2-bgp]peer 10.0.4.4 next-hop-local
```

[R3]bgp 200

[R3-bgp]peer 10.0.4.4 next-hop-local

配置完成后，查看R4的BGP路由表。

<R4>display bgp routing-table

BGP Local router ID is 10.0.24.4

Status codes:* - valid, > - best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 14

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.2.2	0	100	0	100?
* i		10.0.3.3	0	100	0	100?
*>i	10.0.15.0/24	10.0.2.2	0	100	0	100?
* i		10.0.3.3	0	100	0	100?
*>i	10.0.16.0/24	10.0.2.2	0	100	0	100?
* i		10.0.3.3	0	100	0	100?
*>i	10.0.56.0/24	10.0.2.2	2	100	0	100?
* i		10.0.3.3	2	100	0	100?
*>i	50.1.0.0/16	10.0.2.2	2	100	0	100?
* i		10.0.3.3	2	100	0	100?
*>i	60.1.0.0/19	10.0.2.2	2	100	0	100?
* i		10.0.3.3	2	100	0	100?
*>i	60.2.30.0/24	10.0.2.2	1	100	0	100?
* i		10.0.3.3	1	100	0	100?

可以看到，在R4的BGP路由表中，去往总部的路由的下一跳变为R2或R3了。

查看R4的IP路由表。

<R4>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 21 Routes : 21						
Destination/Mask		Proto	Pre	Cost	Flags	NextHop
Interface						
10.0.1.1/32		IBGP		255 0	RD	10.0.2.2
GigabitEthernet0/0/1						
10.0.2.2/32		ISIS-L2	15	10	D	10.0.24.2
GigabitEthernet0/0/1						
10.0.3.3/32		ISIS-L2	15	10	D	10.0.34.3
GigabitEthernet0/0/2						
10.0.4.4/32	Direct	0 0	D	127.0.0.1	LoopBack0	
10.0.15.0/24		IBGP		255 0	RD	10.0.2.2
GigabitEthernet0/0/1						
10.0.16.0/24		IBGP		255 0	RD	10.0.2.2
GigabitEthernet0/0/1						
10.0.24.0/24		Direct	0	0	D	10.0.24.4
GigabitEthernet0/0/1						
10.0.24.4/32		Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.24.255/32		Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1						
10.0.34.0/24		Direct	0	0	D	10.0.34.4
GigabitEthernet0/0/2						

10.0.34.4/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
10.0.34.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
10.0.56.0/24	IBGP	255	2	RD	10.0.2.2
GigabitEthernet0/0/1					
20.0.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack1
50.1.0.0/16	IBGP	255	2	RD	10.0.2.2
GigabitEthernet0/0/1					
60.1.0.0/19	IBGP	255	2	RD	10.0.2.2
GigabitEthernet0/0/1					
60.2.30.0/24	IBGP	255	1	RD	10.0.2.2
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R4的BGP路由表中的有效且最优的路由被添加进了R4的IP路由表中。

然而，公司总部不希望公司分部访问 60.2.30.0/24 网段，因为这是总部财务部门所属的网段，所以公司总部的网络管理员决定在 R1 上使用路由策略在引入 OSPF 路由时过滤掉这个网段的路由。

```
[R1]acl 2001
```


10.0.24.0/24	Direct	0	0	D	10.0.24.4
GigabitEthernet0/0/1					
10.0.24.4/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.24.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/1					
10.0.34.0/24	Direct	0	0	D	10.0.34.4
GigabitEthernet0/0/2					
10.0.34.4/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
10.0.34.255/32	Direct	0	0	D	127.0.0.1
GigabitEthernet0/0/2					
10.0.56.0/24	IBGP	255	2	RD	10.0.2.2
GigabitEthernet0/0/1					
20.0.4.4/32	Direct	0	0	D	127.0.0.1 LoopBack1
50.1.0.0/16	IBGP	255	2	RD	10.0.2.2
GigabitEthernet0/0/1					
60.1.0.0/19	IBGP	255	2	RD	10.0.2.2
GigabitEthernet0/0/1					
127.0.0.0/8	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.0.0.1/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
127.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					
255.255.255.255/32	Direct	0	0	D	127.0.0.1
InLoopBack0					

可以看到，R4已经学习不到关于60.2.30.0/24这个网段的路由了。读者可自行查看R2和R3的IP路由表，并会发现R2和R3也学习不到关于60.2.30.0/24这个网段的路由。

为了能将公司分部的路由信息通告给公司总部，在R2和R3上将IS-IS路由引入BGP进程。

[R2]bgp 200

```
[R2-bgp]import-route isis 1
```

[R3]bgp 200

```
[R3-bgp]import-route isis 1
```

在R1上查看IP路由表。

```
<R1>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public

Destinations : 27 Routes : 28

Interface	Destination/Mask	Proto	Pre	Cost	Flags	NextHop
	10.0.1.1/32	Direct	0	0	D	LoopBack0
GigabitEthernet2/0/0	10.0.2.2/32	EBGP	255	0	D	10.0.12.2
	10.0.3.3/32	EBGP	255	0	D	10.0.13.3
GigabitEthernet2/0/1	10.0.4.4/32	EBGP	255	10	D	10.0.12.2
GigabitEthernet2/0/0	10.0.12.0/24	Direct	0	0	D	10.0.12.1
GigabitEthernet2/0/0						

```

.....
10.0.16.255/32      Direct      0    0      D      127.0.0.1
GigabitEthernet0/0/1
10.0.24.0/24        EBGp        255   0      D      10.0.12.2
GigabitEthernet2/0/0
10.0.34.0/24        EBGp        255   0      D      10.0.13.3
GigabitEthernet2/0/1
10.0.56.0/24        OSPF        10    2      D      10.0.15.5
GigabitEthernet0/0/0
    OSPF    10    2    D    10.0.16.6  GigabitEthernet0/0/1
20.0.4.4/32         EBGp        255   10     D      10.0.12.2
GigabitEthernet2/0/0
50.1.0.0/16         O_ASE       150    2      D      10.0.15.5
GigabitEthernet0/0/0

```

.....

可以看到，R1通过BGP接收到了公司分部网络的路由。

总部交换机S1和S2由于没有运行BGP路由协议，所以无法获得公司分部网络的路由。为此，可以在R1上通过OSPF非强制方式下发缺省路由，S1和S2通过该缺省路由来访问公司分部网络。OSPF非强制下发缺省路由的条件是，IP路由表中存在非OSPF进程的缺省路由。因此，可以在R2和R3上配置BGP下发缺省路由给R1，使R1的路由表中存在一条来自BGP的缺省路由。

```

[R1]ospf 1
[R1-ospf-1]default-route-advertise
[R2]bgp 200
[R2-bgp]peer 10.0.12.1 default-route-advertise
[R3]bg 200

```


查看R1的IP路由表。

Route Flags: R - relay, D - download to fib

可以看到，R1的IP路由表中有了一条来自BGP的缺省路由。

查看S1的IP路由表。

Route Flags: R - relay, D - download to fib

```

-----
                                Routing Tables: Public
                                Destinations : 17   Routes : 18

Destination/Mask    Proto    Pre  Cost    Flags    NextHop
-----
0.0.0.0/0           O_ASE    150  1       D        10.0.15.1  Vlanif51
10.0.1.1/32         OSPF     10   1       D        10.0.15.1  Vlanif51
.....

```

可以看到，S1 的IP 路由表中有了一条缺省路由。

查看S2的IP路由表。

<S2>display ip routing-table

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 16 Routes : 17						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
0.0.0.0/0	O_ASE	150	1	D	10.0.16.1	Vlanif61
10.0.1.1/32	OSPF	10	1	D	10.0.16.1	Vlanif61
.....						

可以看到，S2的IP路由表中有了一条缺省路由。

至此，公司分部网络与总部网络之间实现了互通。但是，总部网络管理员在 R1 上发现去往分部的10.0.4.4网段和20.0.4.4网段的路由的下一跳都是R2。为了优化网络，实现负载分担，管理员希望通过修改BGP 路由的 MED 属性使得从 R1 去往公司分部10.0.4.4网段的下一跳为R2，从R1去往公司分部20.0.4.4网段的下一跳为R3。

查看R1的BGP路由表中20.0.4.4的MED值。

<R1>display bgp routing-table

BGP Local router ID is 10.0.1.1

Status codes: *- valid, >- best, d - damped,

h - history, i - internal, s - suppressed, S - Stale

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

Total Number of Routes: 20

Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn

```

*> 0.0.0.0      10.0.12.2  0      0      200i
*           10.0.13.3  0      0      200i
.....
*> 10.0.56.0/24  0.0.0.0    2      0      ?
*> 20.0.4.4/32  10.0.12.2  10      0      200?
*           10.0.13.3  10      0      200?
*> 50.1.0.0/16  0.0.0.0    2      0      ?
*> 60.1.0.0/19  0.0.0.0    2      0      ?

```

可以看到，R1 的 BGP 路由表中有两条关于 20.0.4.4 的路由，下一条分别是 R2 和R3，且MED的值都为10。

在R1上配置路由策略，如果路由匹配上20.0.4.4，则将其MED的值改为5，然后在R1上针对来自R3的路由信息应用此策略。

```

[R1]acl 2002
[R1-acl-basic-2002]rule permit source 20.0.4.4 0
[R1-acl-basic-2002]route-policy MED permit node 10
[R1-route-policy]if-match acl 2002
[R1-route-policy]apply cost 5
[R1-route-policy]route-policy MED permit node 20
[R1-route-policy]bgp 100
[R1-bgp]peer 10.0.13.3 route-policy MED import

```

查看R1的IP路由表。

```
<R1>display ip routing-table
```

```
Route Flags: R - relay, D - download to fib
```

```

-----
-----

```

```

Routing Tables: Public
Destinations : 28  Routes : 29

```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
Interface					
0.0.0.0/0	EBGP	255	0	D	10.0.12.2
GigabitEthernet2/0/0					
.....					
20.0.4.4/32	EBGP	255	5	D	10.0.13.3
GigabitEthernet2/0/1					
50.1.0.0/16	O_ASE	150	2	D	10.0.15.5
GigabitEthernet0/0/0					
.....					

可以看到，R1去往20.0.4.4的路由的下一跳变为了R3。

在 PC-1 上使用 tracert 命令验证去往 10.0.4.4 和 20.0.4.4 的报文所经过的路径，如图8-13所示。

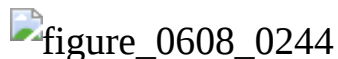


图8-13 路径验证

从图 8-13 中可以看到，去往 10.0.4.4 的报文经过了 R2（10.0.12.2），而去往20.0.4.4的报文经过了R3（10.0.13.3），这样就实现了从公司总部到公司分部的流量的负载分担。

另一方面，公司分部的网络管理员发现，从 R4 去往总部所有网段的路由的下一跳都是R2。为了优化网络，实现负载分担，希望通过修改AS_Path属性使得公司分部经由R2访问用户网段50.1.0.0/16，经由R3访问用户网段60.1.0.0/19。

查看R4的BGP路由表。

```
<R4>display bgp routing-table
```

```
BGP Local router ID is 10.0.24.4
```

```
Status codes: *- valid, >- best, d - damped,
```

```
h - history, i - internal, s - suppressed, S - Stale
```

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

Total Number of Routes: 24

	Network	NextHop	MED	LocPrf	PrefVal	Path/Ogn
*>i	10.0.1.1/32	10.0.2.2	0	100	0	100?
* i	10.0.3.3	0	100	0	100?	
.....						
*>i	20.0.4.4/32	10.0.2.2	10	100	0	?
* i	10.0.3.3	10	100	0	?	
*>i	50.1.0.0/16	10.0.2.2	2	100	0	100?
* i	10.0.3.3	2	100	0	100?	
*>i	60.1.0.0/19	10.0.2.2	2	100	0	100?
* i	10.0.3.3	2	100	0	100?	

可以看到，在R4上去往总部的用户网段50.1.0.0/16和60.1.0.0/19的路由有两个下一跳R2和R3，AS_Path属性都是一样的，只经过了AS 100。接下来，在R4上配置路由策略，如果路由匹配上60.1.0.0/19，则将其AS_Path属性添加上一个重复的AS编号100，然后在R4上针对来自R2的路由信息应用此策略。

```
[R4]acl 2001
```

```
[R4-acl-basic-2001]rule permit source 60.1.0.0 0.0.31.255
```

```
[R4-acl-basic-2001]route-policy AS-PATH permit node 10
```

```
[R4-route-policy]if-match acl 2001
```

```
[R4-route-policy]apply as-path 100 additive
```

```
[R4-route-policy]route-policy AS-PATH permit node 20
```

```
[R4-route-policy]bgp 200
```

```
[R4-bgp]peer 10.0.2.2 route-policy AS-PATH import
```

配置完成后，查看R4的IP路由表。

```
<R4>display ip routing-table
```

Route Flags: R - relay, D - download to fib

Routing Tables: Public						
Destinations : 20 Routes : 20						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	
Interface						
10.0.1.1/32	IBGP		255	0	RD	10.0.2.2
GigabitEthernet0/0/1						
.....						
20.0.4.4/32	Direct	0 0	D	127.0.0.1	LoopBack1	
50.1.0.0/16	IBGP		255	2	RD	10.0.2.2
GigabitEthernet0/0/1						
60.1.0.0/19	IBGP		255	2	RD	10.0.3.3
GigabitEthernet0/0/2						
.....						

可以看到，现在R4去往用户网段50.1.0.0/16的下一跳为R2，去往用户网段60.1.0.0/19的下一跳为R3。在R4上使用tracert命令验证去往这两个网段的报文所经过的路径。

<R4>tracert 50.1.10.1

tracert to 50.1.10.1(50.1.10.1), max hops: 30 ,packet length:
40,press CTRL_C to break

1 10.0.24.2 10 ms 1 ms 20 ms

2 10.0.12.1 20 ms 10 ms 20 ms

3 10.0.15.5 < AS=100 > 30 ms 10 ms 20 ms

<R4>tracert 60.1.10.1

```
tracert to 60.1.10.1(60.1.10.1), max hops: 30 ,packet length:
40,press CTRL_C to break
```

```
1 10.0.34.3 10 ms 10 ms 10 ms
```

```
2 10.0.13.1 20 ms 20 ms 20 ms
```

```
3 10.0.16.6 < AS=100 > 30 ms 10 ms 40 ms
```

可以看到，从 R4 去往用户网段 50.1.0.0/16 的报文是经 R2 转发的，去往用户网段60.1.0.0/19的报文是经R3转发的，这样就实现了从公司分部到公司总部的流量的负载分担。至此，整个实验网络的分析和配置工作便告结束。

思考

现实中，复杂的网络结构和网络需求往往会造成网络设计和部署工作中的疏忽和失误。就以本实验网络为例，看上去我们好像已经完成了正确的网络分析工作和正确的配置工作，但实际上，这个网络还隐藏着一个严重的故障隐患，它可能会导致本该能够进行相互通信的用户之间无法进行通信。请读者朋友们找出故障隐患并提出自己的解决办法。