



CloudEngine 系列交换机

VXLAN 技术白皮书

文档版本 03

发布日期 2015-05-30

华为技术有限公司



版权所有 © 华为技术有限公司 2015。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <http://enterprise.huawei.com>

目录

1 VXLAN 配置	1
1.1 VXLAN 简介	2
1.2 原理描述	3
1.2.1 基本概念	4
1.2.2 报文格式	5
1.2.3 VXLAN 部署方案	6
1.2.4 数据报文转发	10
1.2.5 VXLAN QoS	18
1.2.6 VXLAN 增强特性	18
1.2.6.1 ARP/MAC 动态学习	19
1.2.6.2 ARP 广播抑制	20
1.2.6.3 VXLAN 集中式多活网关	22
1.2.6.4 VXLAN 分布式网关	28
1.2.6.5 VXLAN 双活接入	37
1.3 应用场景	43
1.3.1 同网段终端用户通信的应用	43
1.3.2 不同网段终端用户通信的应用	44
1.3.3 在虚拟机迁移场景中的应用	45
1.3.4 VXLAN 分布式网关的应用	47
1.3.5 VXLAN 集中式多活网关的应用	48
1.3.6 VXLAN 双活接入的应用	49
1.4 配置注意事项	50
1.5 配置 VXLAN（SNC 控制器方式）	52
1.6 配置 VXLAN（单机方式）	53
1.6.1 配置同网段用户通过 VXLAN 隧道互通	53
1.6.1.1 配置业务接入点实现区分业务流量	55
1.6.1.2 配置 VXLAN 隧道转发业务流量	56
1.6.1.3 （可选）配置提升 VXLAN 网络安全性	57
1.6.1.4 检查配置结果	57
1.6.2 配置不同网段用户通过 VXLAN 三层网关通信	57
1.6.2.1 配置业务接入点实现区分业务流量	59
1.6.2.2 配置 VXLAN 隧道转发业务流量	60

1.6.2.3 配置三层网关实现不同网段业务流量互通.....	61
1.6.2.4 （可选）配置 VXLAN 集中式多活网关.....	62
1.6.2.5 （可选）配置提升 VXLAN 网络安全性.....	63
1.6.2.6 检查配置结果.....	64
1.6.3 配置 VXLAN 分布式网关.....	64
1.6.3.1 配置 VXLAN 二层网关.....	66
1.6.3.2 配置 VXLAN 三层网关.....	68
1.6.3.3 （可选）配置提升 VXLAN 网络安全性.....	70
1.6.3.4 检查配置结果.....	71
1.6.4 配置 VXLAN 双活接入功能.....	71
1.6.4.1 配置双归设备通过 M-LAG 与服务器对接.....	72
1.6.4.1.1 配置 DFS Group.....	73
1.6.4.1.2 配置 peer-link.....	73
1.6.4.1.3 配置绑定 DFS Group.....	74
1.6.4.1.4 检查配置结果.....	76
1.6.4.2 配置双归设备上的虚拟 VTEP.....	76
1.7 维护 VXLAN.....	77
1.7.1 统计并查看 VXLAN 统计信息.....	77
1.7.2 清除 BD 内报文统计信息.....	77
1.7.3 监控 VXLAN 运行状况.....	78
1.7.4 配置 VXLAN 告警上报功能.....	78
1.8 配置举例.....	79
1.8.1 配置同网段用户通过 VXLAN 隧道互通示例（单机方式）.....	79
1.8.2 配置不同网段用户通过 VXLAN 三层网关通信示例（SNC 控制器方式）.....	83
1.8.3 配置集中式多活网关示例（单机方式）.....	99
1.8.4 配置 VXLAN 双活接入示例（单机方式）.....	107
1.8.5 配置 VXLAN 分布式网关示例（单机方式）.....	114
1.8.6 配置 VXLAN 分布式网关+双活接入综合示例（单机方式）.....	122
1.9 参考标准和协议.....	134

1 VXLAN 配置

关于本章

1.1 VXLAN简介

介绍VXLAN的定义、目的和受益。

1.2 原理描述

介绍VXLAN的实现原理。

1.3 应用场景

介绍VXLAN的应用场景。

1.4 配置注意事项

介绍部署VXLAN的注意事项。

1.5 配置VXLAN（SNC控制器方式）

介绍了SNC控制器配合设备实现VXLAN部署的方法。

1.6 配置VXLAN（单机方式）

介绍了不依赖于任何控制器，直接在设备上配置VXLAN的方法。

1.7 维护VXLAN

通过维护VXLAN，可以实现清除VXLAN统计数据、监控VXLAN的运行状况等。

1.8 配置举例

介绍VXLAN配置举例，配置举例中包括组网需求、配置思路、配置过程和配置文件。

1.9 参考标准和协议

介绍VXLAN的参考标准和协议。

1.1 VXLAN 简介

介绍VXLAN的定义、目的和受益。

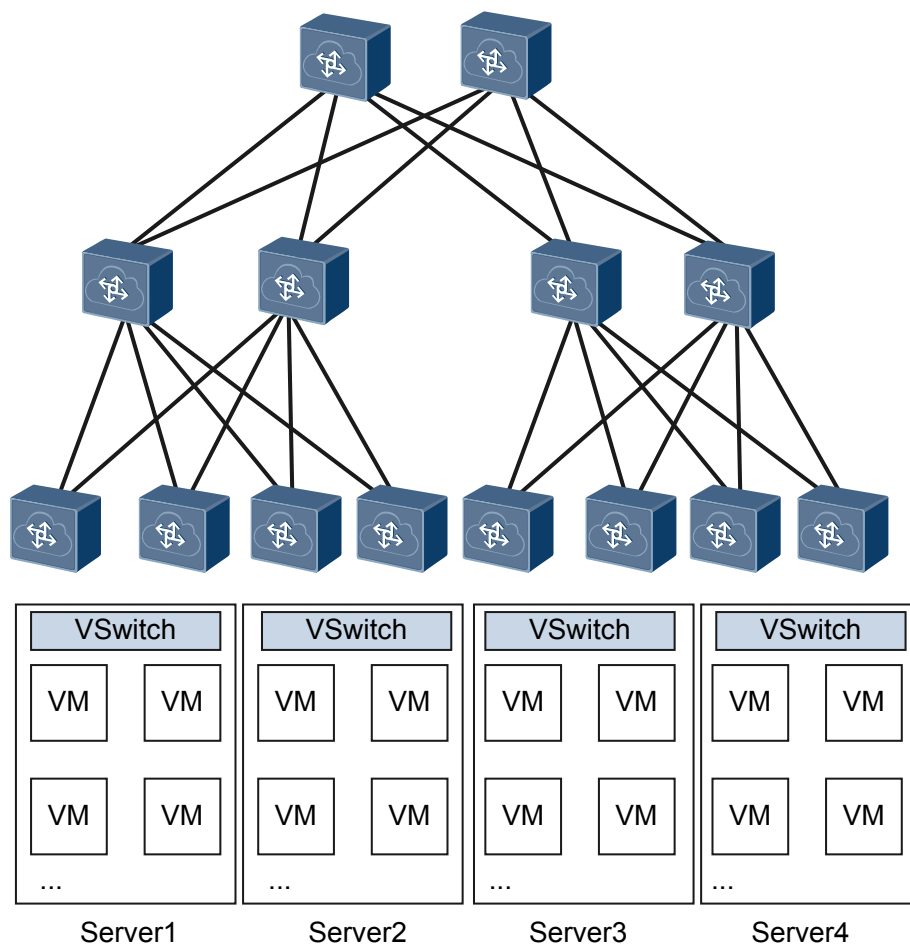
定义

RFC7348定义了VXLAN扩展方案（Virtual eXtensible Local Area Network），采用MAC in UDP（User Datagram Protocol）封装方式，是NVO3（Network Virtualization over Layer 3）中的一种网络虚拟化技术。

目的

作为云计算的核心技术之一，服务器虚拟化凭借其大幅降低IT成本、提高业务部署灵活性、降低运维成本等优势已经得到越来越多的认可和部署。

图 1-1 服务器虚拟化示意图



如图1-1所示，一台服务器可虚拟成多台虚拟机，而一台虚拟机相当于一台主机。主机的数量发生了数量级的变化，这也为虚拟网络带来了如下问题：

- 虚拟机规模受网络规格限制
在大二层网络环境下，数据报文是通过查询MAC地址表进行二层转发，而MAC地址表的容量限制了虚拟机的数量。
- 网络隔离能力限制
当前主流的网络隔离技术是VLAN或VPN（Virtual Private Network），在大规模的虚拟化网络中部署存在如下限制：
 - 由于IEEE 802.1Q中定义的VLAN Tag域只有12比特，仅能表示4096个VLAN，无法满足大二层网络中标识大量用户群的需求。
 - 传统二层网络中的VLAN/VPN无法满足网络动态调整的需求。
- 虚拟机迁移范围受网络架构限制
虚拟机启动后，可能由于服务器资源等问题（如CPU过高，内存不够等），需要将虚拟机迁移到新的服务器上。为了保证虚拟机迁移过程中业务不中断，则需要保证虚拟机的IP地址、MAC地址等参数保持不变，这就要求业务网络是一个二层网络，且要求网络本身具备多路径的冗余备份和可靠性。

针对大二层网络，VXLAN的提出很好地解决了上述问题：

- 针对虚拟机规模受网络规格限制
VXLAN将虚拟机发出的数据包封装在UDP中，并使用物理网络的IP和MAC地址作为外层头进行封装，对网络只表现为封装后的参数。因此，极大降低了大二层网络对MAC地址规格的需求。
- 针对网络隔离能力限制
VXLAN引入了类似VLAN ID的用户标识，称为VXLAN网络标识VNI（VXLAN Network ID），由24比特组成，支持多达16M $((2^{24}-1)/1024^2)$ 的VXLAN段，从而满足了大量的用户标识。
- 针对虚拟机迁移范围受网络架构限制
通过VXLAN构建大二层网络，保证了在虚拟迁移时虚拟机的IP地址、MAC地址等参数保持不变。

受益

随着数据中心在物理网络基础设施上实施服务器虚拟化的快速发展，作为NVO3技术之一的VXLAN：

- 通过24比特的VNI可以支持多达16M的VXLAN段的网络隔离，对用户进行隔离和标识不再受到限制，可满足海量租户。
- 除VXLAN网络边缘设备，网络中的其他设备不需要识别虚拟机的MAC地址，减轻了设备的MAC地址学习压力，提升了设备性能。
- 通过采用MAC in UDP封装来延伸二层网络，实现了物理网络和虚拟网络解耦，租户可以规划自己的虚拟网络，不需要考虑物理网络IP地址和广播域的限制，大大降低了网络管理的难度。

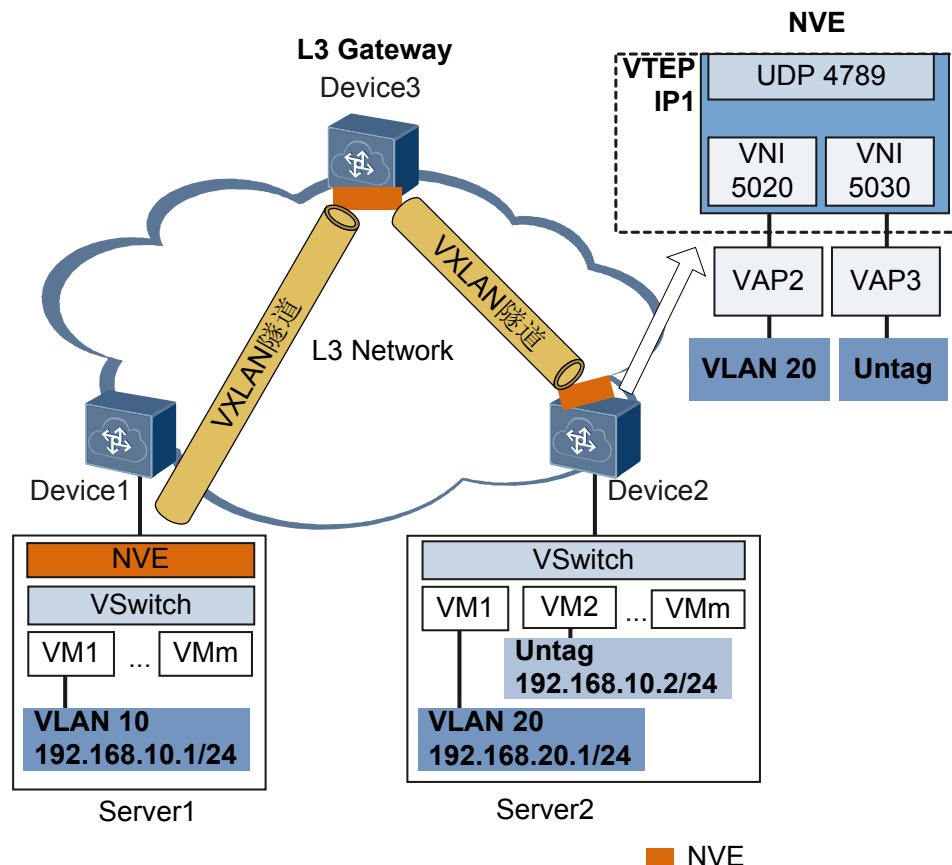
1.2 原理描述

介绍VXLAN的实现原理。

1.2.1 基本概念

VXLAN是NVO3中的一种网络虚拟化技术，通过将VM或物理服务器发出的数据包封装在UDP中，并使用物理网络的IP/MAC作为报文头进行封装，然后在IP网络上传输，到达目的地后由隧道终结点解封装并将数据发送给目标虚拟机或物理服务器。

图 1-2 VXLAN 结构示意图



通过VXLAN，虚拟网络可接入大量租户，且租户可以规划自己的虚拟网络，不需要考虑物理网络IP地址和广播域的限制，降低了网络管理的难度。下面结合图1-2介绍VXLAN相关概念。

- NVE (Network Virtualization Edge)：网络虚拟边缘节点NVE，是实现网络虚拟化功能的网络实体。报文经过NVE封装转换后，NVE间就可基于三层基础网络建立二层虚拟化网络。



设备和服务器上的虚拟交换机VSwitch都可以作为NVE。

- VTEP (VXLAN Tunnel Endpoints)：VTEP是VXLAN隧道端点，封装在NVE中，用于VXLAN报文的封装和解封装。

VTEP与物理网络相连，分配有物理网络的IP地址，该地址与虚拟网络无关。

VXLAN报文中源IP地址为本节点的VTEP地址，VXLAN报文中目的IP地址为对端节点的VTEP地址，一对VTEP地址就对应着一个VXLAN隧道。

- VNI (VXLAN Network Identifier)：VXLAN网络标识VNI类似VLAN ID，用于区分VXLAN段，不同VXLAN段的虚拟机不能直接二层相互通信。

一个VNI表示一个租户，即使多个终端用户属于同一个VNI，也表示一个租户。VNI由24比特组成，支持多达16M $(2^{24}-1)/1024^2$ 的租户。

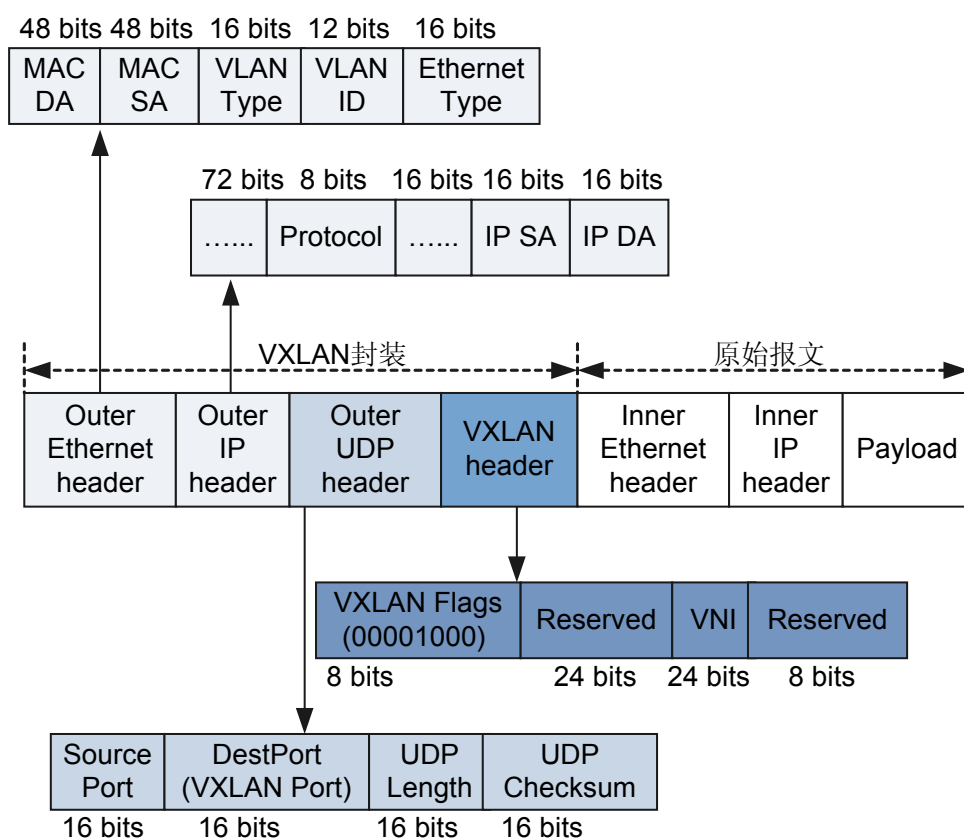
- VAP（Virtual Access Point）：虚拟接入点VAP统一为二层子接口，用于接入数据报文。

为二层子接口配置不同的流封装，可实现不同的数据报文接入不同的二层子接口。

1.2.2 报文格式

VXLAN是MAC in UDP的网络虚拟化技术，所以其报文封装是在原始以太报文之前添加了一个UDP头及VXLAN头封装。具体报文格式如图1-3所示。

图 1-3 VXLAN 报文格式



- VXLAN头封装
 - Flags: 8比特，取值为00001000。
 - VNI: VXLAN网络标识，24比特，用于区分VXLAN段。
 - Reserved: 24比特和8比特，必须设置为0。
- 外层UDP头封装
 - 目的UDP端口号是4789。源端口号是内层以太网头通过哈希算法计算后的值。
- 外层IP头封装

源IP地址为发送报文的虚拟机所属VTEP的IP地址；目的IP地址是目的虚拟机所属VTEP的IP地址。

- 外层Ethernet头封装
 - SA: 发送报文的虚拟机所属VTEP的MAC地址。
 - DA: 目的虚拟机所属VTEP上路由表中直连的下一跳MAC地址。
 - VLAN Type: 可选字段，当报文中携带VLAN Tag时，该字段取值为0x8100。
 - Ethernet Type: 以太报文类型，IP协议报文该字段取值为0x0800。

1.2.3 VXLAN 部署方案

目前，设备支持通过**单机方式**和**控制器方式**来部署VXLAN网络。

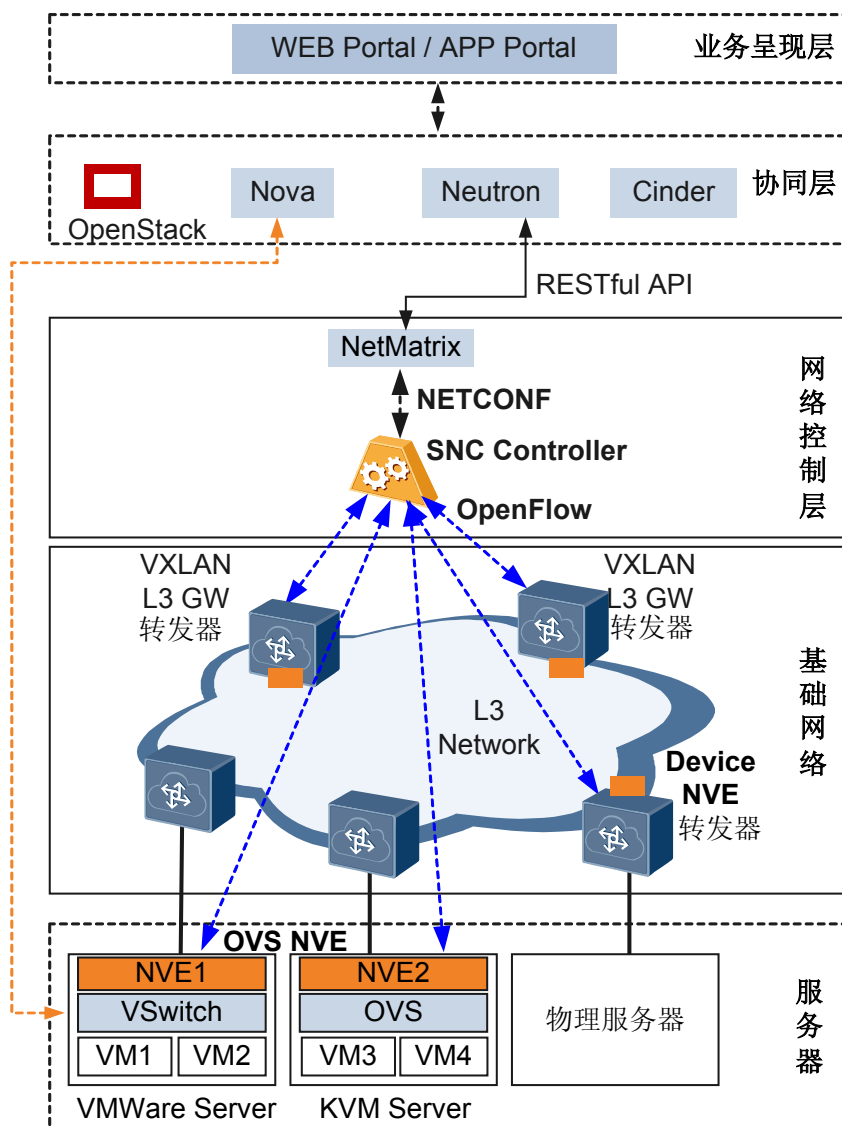
- **单机方式**: 传统网络部署方式，按照网络规划登录到每台设备上配置。云计算数据中心中，此方式无法协同云平台实现网络的自动化部署。
- **控制器方式**: 在大二层网络中，为了方便控制与部署引入了控制器。控制器是统一的网络控制平台，实现网络资源统一协调及管理，协同云平台实现业务和网络的自动化部署。

SNC 控制器方式

- SNC控制器方式概述

SNC控制器方式是指通过SNC控制器动态建立VXLAN隧道，并通过OpenFlow协议向转发器下发相应的流表以指导转发器生成VXLAN隧道和报文在隧道中的转发。这种方式下，设备仅作为转发器，设备只需与控制器建立OpenFlow通道，而无需进行任何VXLAN的配置。

图 1-4 基于 SNC 控制器+VXLAN 解决方案网络框架示意图



如图1-4所示，SNC控制器北向通过NetMatrix与Neutron连接，获得用户虚拟网络的信息。控制器根据用户虚拟网络信息，进行动态计算生成网络相关配置信息及流表信息，并自动映射到物理网络。基于SNC控制器+VXLAN解决方案网络框架介绍如表1-1所示。

表 1-1 基于 SNC 控制器+VXLAN 解决方案网络框架介绍

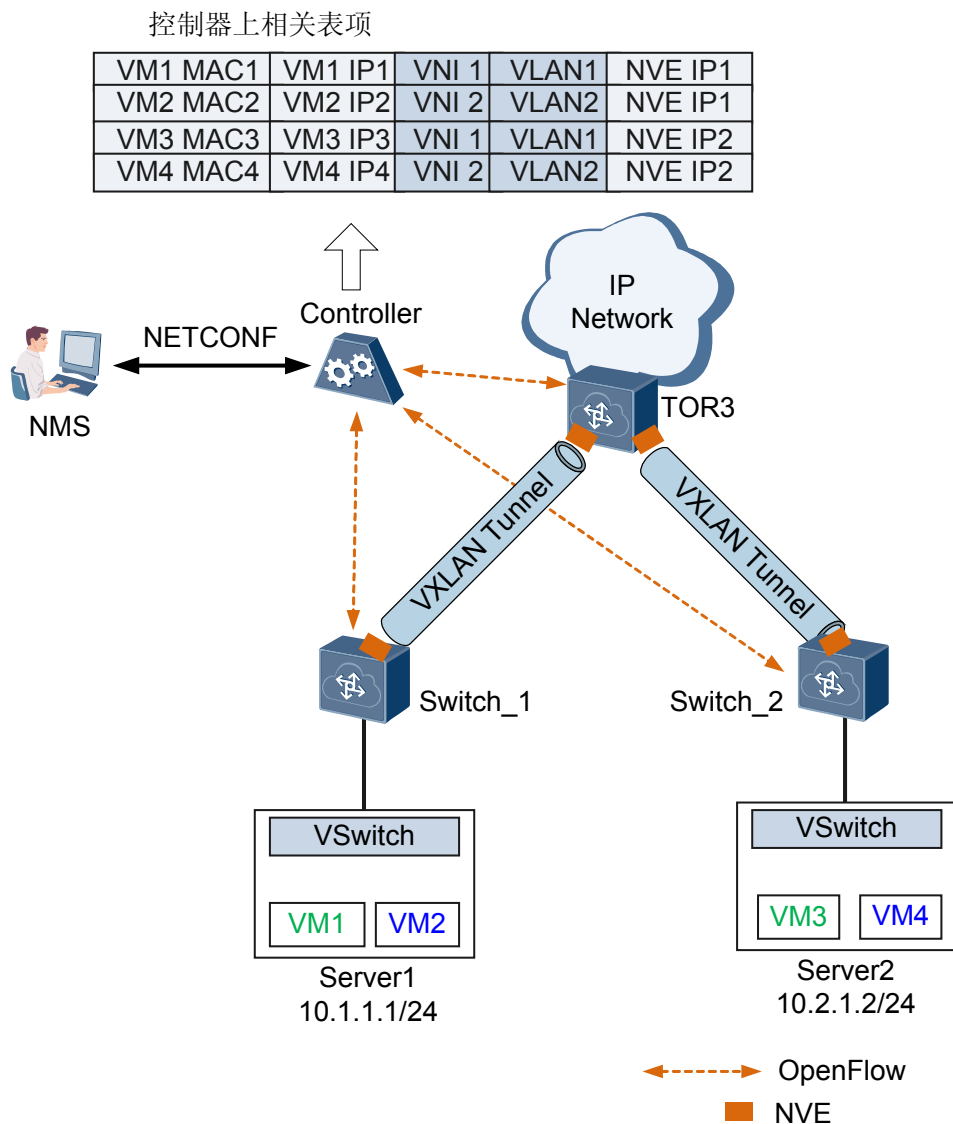
框架名称	说明
业务呈现层	面向运营商、企业、租户，提供业务灵活定制化界面。租户通过Portal定制业务，包括网络业务和主机业务。

框架名称	说明
协同层	<p>Nova、Neutron、Cinder属于云平台，是标准、开放的OpenStack架构，可实现存储、计算和网络资源的协同。</p> <p>租户通过Portal定制不同的业务，云平台的动作也不一样：</p> <ul style="list-style-type: none"> ● 网络业务：Neutron负责分配Network ID、子网ID、VNI等资源，并保存至Neutron DB。 ● 主机业务： <ul style="list-style-type: none"> - Nova：负责虚拟网络的管理，确定VM IP、VM MAC、VM所属服务器的ID，并通告给Neutron。 - Neutron：实现业务与网络ID关联，确定VM IP、VM MAC、VNI、Port、VM所属服务器的ID、NVE IP映射关系。 - Cinder：负责分配存储资源。
网络控制层	<p>由NetMatrix网管系统和控制器组成，完成网络建模和网络实例化。</p> <ul style="list-style-type: none"> ● NetMatrix：通过NETCONF协议与控制器建立通信通道，实现网络配置、信息采集和规划结果下发；通过API（Application Programming Interface）连接Neutron实现业务快速定制和自动发放。 ● 控制器：通过OpenFlow协议与转发器建立通信通道，负责集中管理转发器，实现统一管理物理和虚拟网络，所有的路径计算与管理都由控制器独立完成。 <p>通常，刀片服务器即可作为控制器。</p> <p>说明 控制器和转发器的邻居关系建立是通过OpenFlow协议完成。控制器和转发器的邻居关系建立过程请见SNC控制器与转发器之间通信通道的建立与维护。</p>
基础网络	<p>物理网络和虚拟网络统一规划。</p> <ul style="list-style-type: none"> ● 支持基于硬件的VXLAN网关提高业务性能。 ● 支持对传统VLAN网络的兼容。 ● 转发器只负责数据报文转发。

- SNC控制器与转发器之间通信通道的建立与维护

通过云平台，控制器可及时感知终端租户的状态，获得用户虚拟网络的信息。如图1-5所示，终端租户上线后，通过云平台，控制器根据获得的用户虚拟网络信息进行动态计算，生成网络相关配置信息及流表信息，并自动映射到物理网络。

图 1-5 控制器与转发器数据同步示意图



如图1-5所示，用户可通过NMS或CLI配置维护控制器，NMS和控制器通过NETCONF协议建立连接后，用户通过NMS配置控制器。在控制器上创建NVE，指定VTEP的源、目的IP地址，配置静态MAC地址或ARP表项。由此在控制器上生成静态MAC地址表项或静态ARP表项。

转发器和控制器之间OpenFlow通道建立成功后，转发器会主动将自身节点信息、接口信息上报给控制器。而控制器将静态MAC地址表项或静态ARP表项下发给转发器，控制器通过OpenFlow管理转发器上的静态MAC地址表项或静态ARP表项。

- 当SNC控制器和转发器发生OpenFlow断连，转发器不会删除静态MAC地址表项和静态ARP表项，以保证流量正常转发。
- 当SNC控制器和转发器重新建立OpenFlow连接后，SNC控制器会将本地更新后的所有静态MAC地址表项和静态ARP表项重新下发给转发器，以确保SNC控制器和转发器上的静态MAC地址表项和静态ARP表项数据一致。

如果SNC控制器上更新后的表项数量小于OpenFlow重新连接前转发器上保存的表项数量，OpenFlow重新建立连接后，转发器同步SNC控制器上所有表项，并自动删除多余的表项。

1.2.4 数据报文转发

在VXLAN网络中，业务接入点统一表现为二层子接口，通过在二层子接口上配置流封装实现不同的接口接入不同的数据报文。广播域统一表现为BD（Bridge-Domain），将二层子接口关联BD后，可实现数据报文通过BD转发。

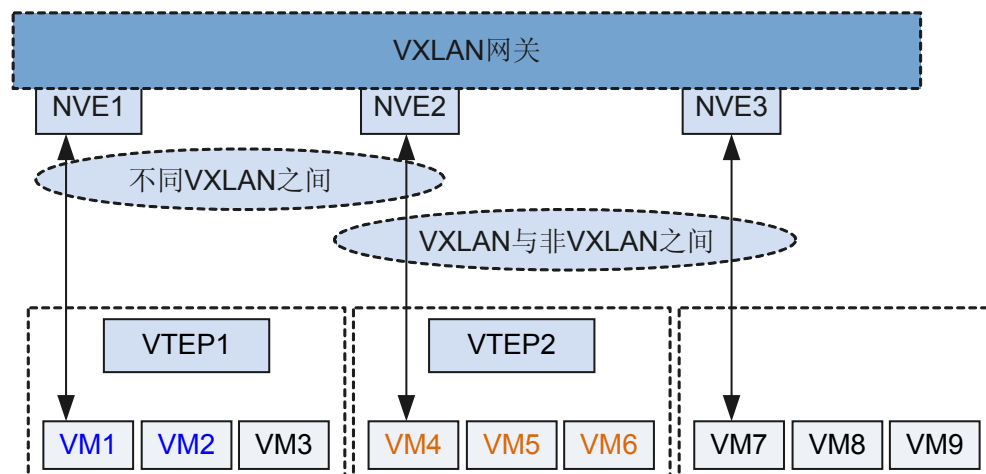
由于VXLAN网络中通过VNI标识不同租户，将VNI以1:1方式映射到BD，可实现不同的租户通过不同的BD转发。

和VLAN类似，不同VNI之间的VXLAN，及VXLAN和非VXLAN之间不能直接相互通信。为了使VXLAN之间，以及VXLAN和非VXLAN之间能够进行通信，VXLAN引入了VXLAN网关，如图1-6所示。

说明

- V100R005C00版本，单机方式VXLAN，只支持部署VXLAN二层网关。指导VXLAN报文转发的MAC地址表项通过动态学习建立。
- NVE可以部署在VSwitch上，也可部署在支持NVE的设备上。本章节主要以NVE部署在支持NVE的设备上为例描述VXLAN网络中数据报文如何转发。

图 1-6 VXLAN 网关示意图



VXLAN网关分为：

- **二层网关**：用于解决租户接入VXLAN虚拟网络的问题，也可用于同一VXLAN虚拟网络的子网通信。

VXLAN二层网关收到用户报文，根据报文中包含的目的MAC地址类型，报文转发流程分为：

- MAC地址是BUM（Broadcast&Unknown-unicast&Multicast）地址，按照**BUM报文转发流程**进行转发。
- MAC地址是已知单播地址，按照**已知单播报文转发流程**进行转发。

- **三层网关**：用于VXLAN虚拟网络的跨子网通信以及外部网络的访问。

BUM 报文转发流程

当BUM报文进入VXLAN隧道，接入端VTEP采用头端复制方式进行报文的VXLAN封装。BUM报文出VXLAN隧道，出口端VTEP对报文解封装。BUM报文具体转发流程如图1-7所示。

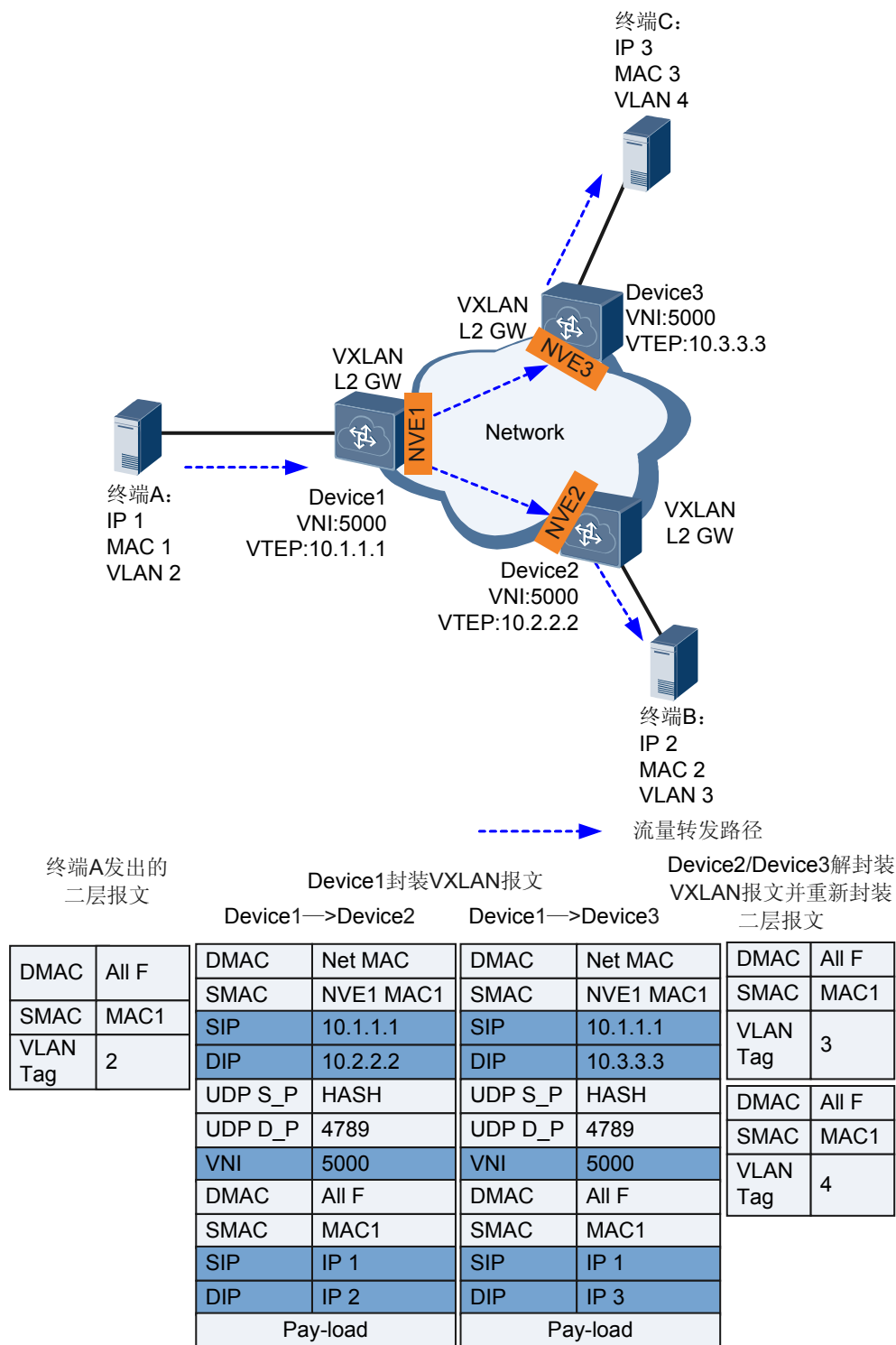
说明

头端复制：接口收到BUM（Broadcast&Unknown-unicast&Multicast）报文，本地VTEP通过控制平面获取属于同一个VNI的VTEP列表，将收到的BUM报文根据VTEP列表进行复制并发送给属于同一个VNI的所有VTEP。

通过头端复制完成BUM报文的广播，不需要依赖组播路由协议。

为了防止环路，VXLAN遵循水平分割原则，即：基于VXLAN隧道收到的BUM报文不会再向其他VXLAN隧道进行头端复制转发，只向对应广播域BD下的用户侧进行广播。

图 1-7 BUM 报文转发过程图



- Device1收到来自终端A的报文，根据报文中接入端口和VLAN信息获取对应的二层广播域，并判断报文的目的地MAC是否为BUM MAC。
 - 是，在对应的二层广播域内广播，并跳转到2。
 - 不是，通过[已知单播报文转发流程](#)。

2. Device1上VTEP根据对应的二层广播域获取对应VNI的头端复制隧道列表，依据获取的隧道列表进行报文复制，并进行VXLAN封装。基于每个出端口和VXLAN封装信息封装VXLAN头和外层IP信息，并从出端口转发。
3. Device2上VTEP收到VXLAN报文后，根据UDP目的端口号、源/目的IP地址、VNI判断VXLAN报文的合法有效性。依据VNI获取对应的二层广播域，然后进行VXLAN解封装，获取内层二层报文，判断报文的目的地MAC是否为BUM MAC。
 - 是，在对应的二层广播域内非VXLAN侧进行广播处理。
 - 不是，再判断是否是本机MAC。
 - 是，上送CPU处理。
 - 不是，在对应的二层广播域内查找出接口和封装信息，并跳转到4。
4. Device2/Device3根据查找到的出接口和封装信息，为报文添加VLAN Tag，转发给对应的B/C。

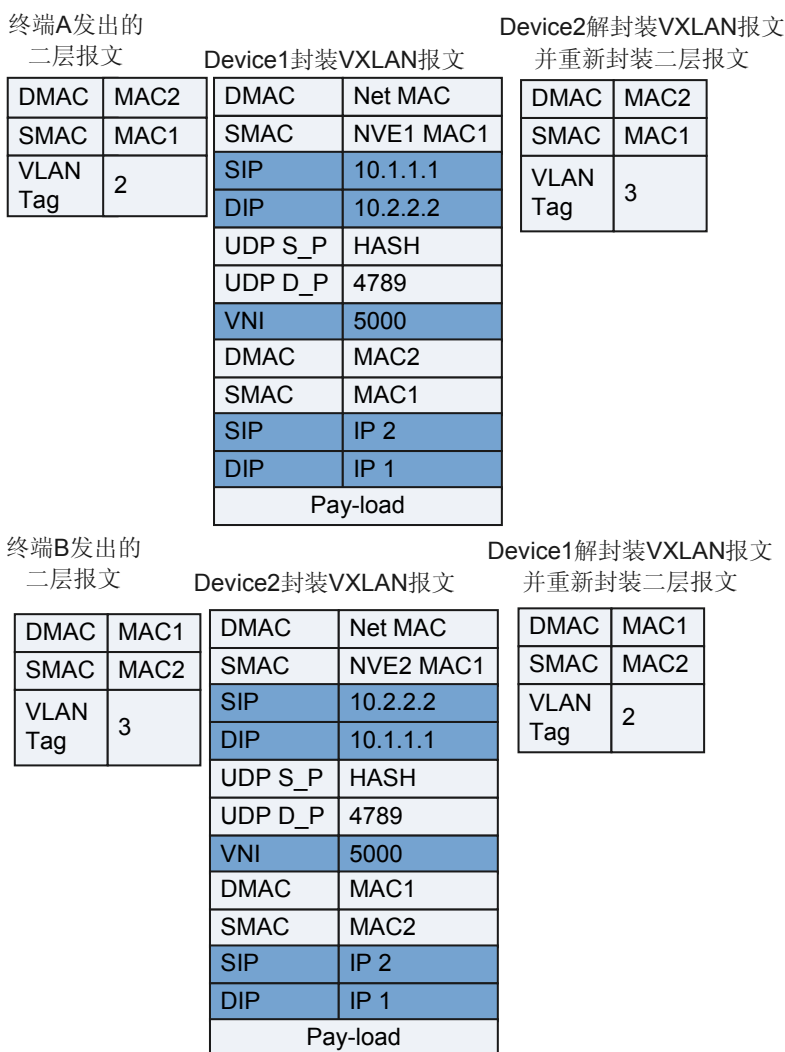
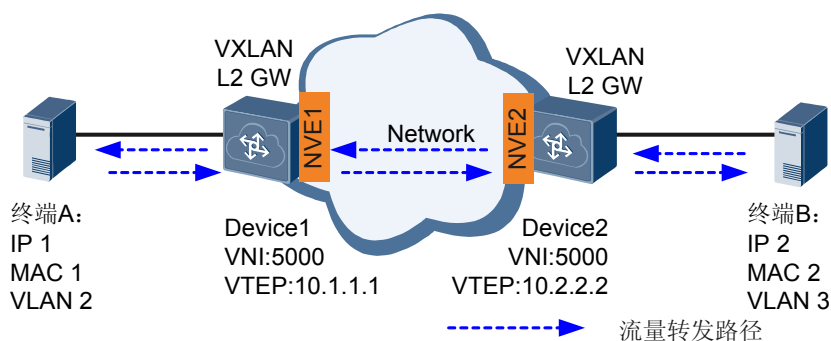
 说明

终端B/C上回应终端A的消息，按照[已知单播报文转发流程](#)进行转发。

已知单播报文转发流程

已知单播报文具体转发流程如[图1-8](#)所示。

图 1-8 已知单播报文转发过程图



- Device1收到来自终端A的报文，根据报文中接入的端口和VLAN信息获取对应的二层广播域，并判断报文的的目的MAC是否为已知单播MAC。
 - 是，再判断是否为本机MAC。
 - 是，上送主机处理。
 - 不是，在对应的二层广播域内查找出接口和封装信息，并跳转到2。
 - 不是，在对应的二层广播域内进行广播处理，并跳转到2。

2. Device1上VTEP根据查找到的出接口和封装信息进行VXLAN封装和报文转发。
3. Device2上VTEP收到VXLAN报文后，根据UDP目的端口号、源/目的IP地址、VNI判断VXLAN报文的合法有效性。依据VNI获取对应的二层广播域，然后进行VXLAN解封装，获取内层二层报文，判断报文的的目的MAC是否为已知单播报文MAC。
 - 是，在对应的二层广播域内查找出接口和封装信息，并跳转到4。
 - 不是，再判断是否是本机MAC。
 - 是，上送主机处理。
 - 不是，通过**BUM报文转发流程**。
4. Device2根据查找到的出接口和封装信息，为报文添加VLAN Tag，转发给对应的终端B。

三层网关

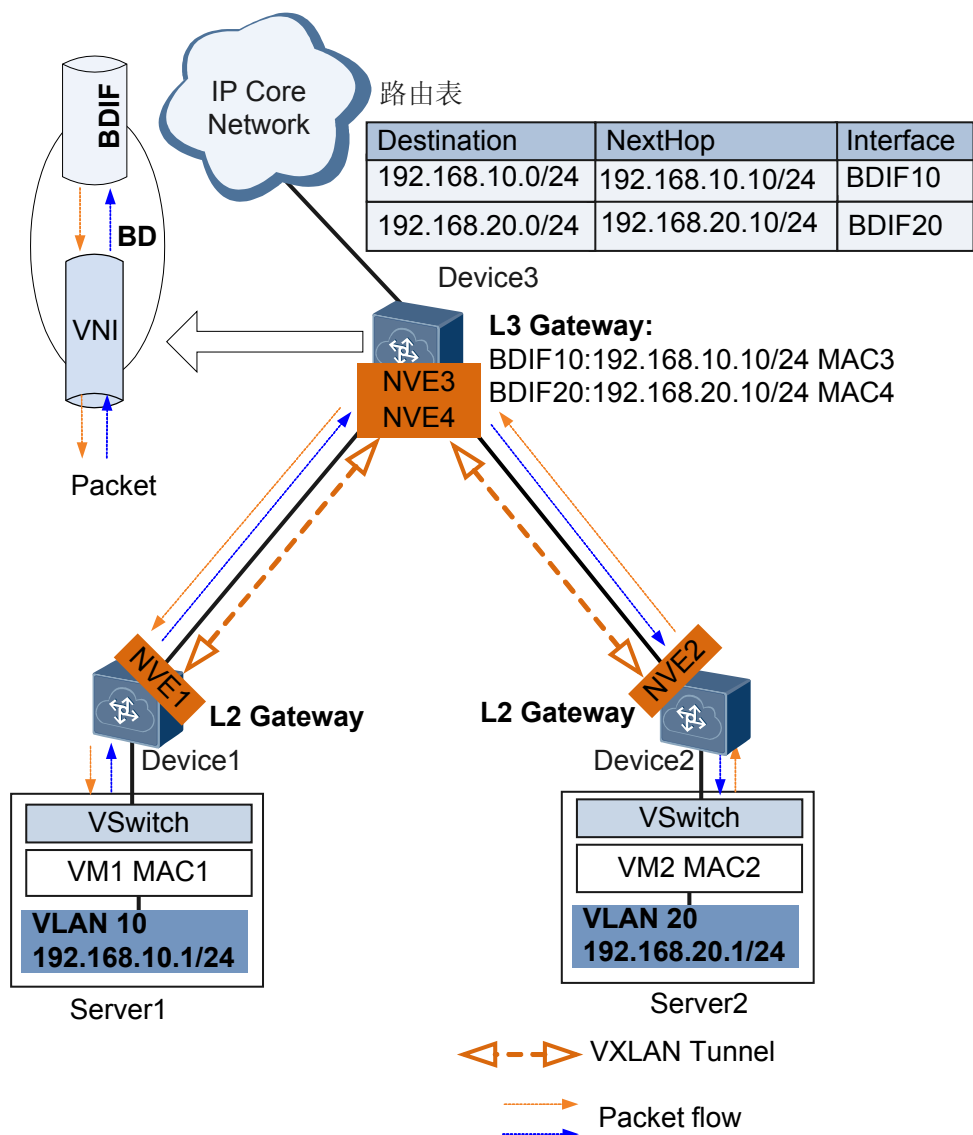
不同网段的VXLAN间通信，及VXLAN和非VXLAN的通信，需要通过IP路由实现。

在VXLAN网络中，将虚拟广播域VN（Virtual Network）对应的VNI（VXLAN Network Identifier）以1:1方式映射到广播域BD，BD成为VXLAN网络的实体，通过BD转发流量。基于BD可创建三层逻辑接口BDIF接口，通过BDIF接口配置IP地址实现不同网段的VXLAN间，及VXLAN和非VXLAN的通信。

说明

BDIF接口类似VLANIF接口。

图 1-9 三层网关通信组网图



如图1-9所示，三层网关通信具体实现过程如下：

- Device3收到VXLAN报文后进行解封装，确认内层报文中的DMAC是否是本网关接口的MAC地址。
 - 是，转给对应目的网段的三层网关处理，并跳转2。
 - 不是，在对应的二层广播域内查找出接口和封装信息。
- 作为VXLAN三层网关的Device3剥除内层报文的以太封装，解析目的IP。根据目的IP查找ARP表项，确认DMAC、VXLAN隧道出接口及VNI等信息。
 - 没有VXLAN隧道出接口及VNI信息，进行三层转发。
 - 有VXLAN隧道出接口及VNI信息，跳转3。
- Device3重新封装VXLAN报文，其中内层报文以太头中的SMAC是网关接口的MAC地址。

说明

Device3与其他Device之间的通信，请参见二层网关实现原理。

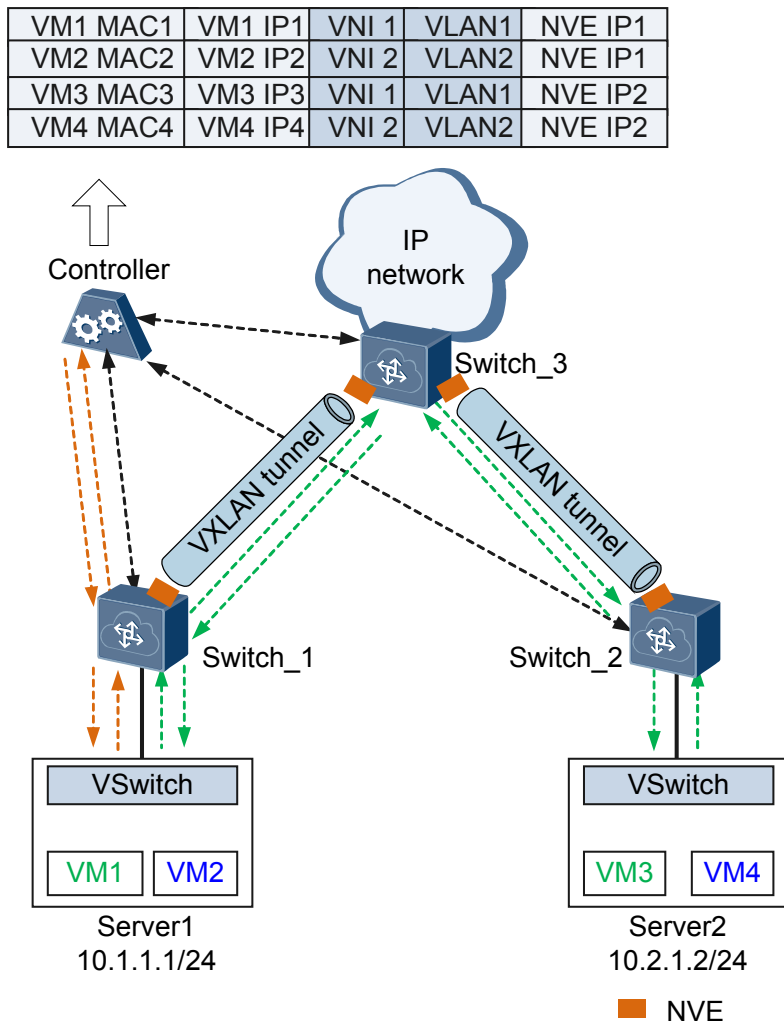
ARP 代答

说明

目前，仅SNC控制器支持ARP代答功能。

如图1-10所示，在传统的终端用户互通过程中，如VM1初次访问VM3时，VM1会向VM3发送ARP广播请求报文。ARP请求报文会在二层网络内广播，VM3收到ARP广播请求报文后进行ARP单播应答。

图 1-10 ARP 代答组网图



为了避免ARP广播请求报文给网络带来广播风暴，可在图1-10所示的控制器上使能ARP代答功能。VM1发送ARP请求报文，请求目的主机VM3的MAC地址具体实现过程如下：

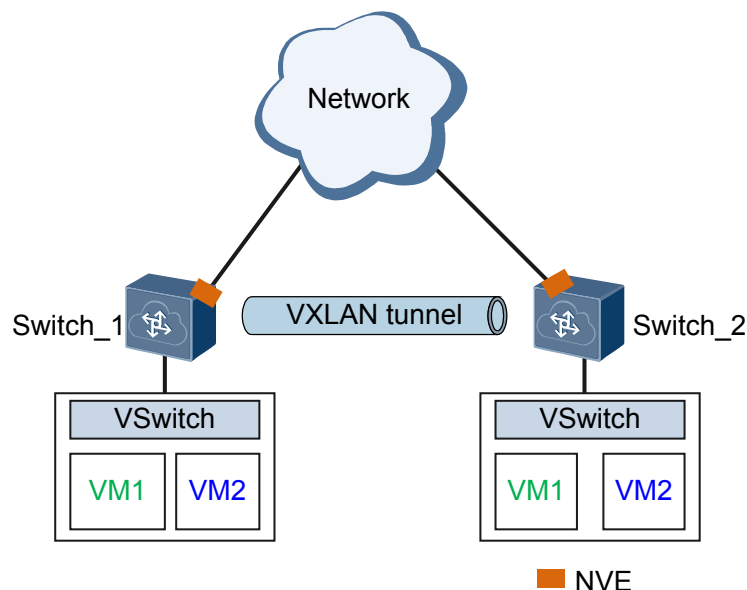
1. VM1发送ARP请求报文（SMAC:MAC1，SIP:IP1，DMAC:FF-FF-FF，DIP: IP3）。
2. NVE1收到ARP请求报文后，通过OpenFlow通道上送控制器处理。
3. 控制器根据IP3查询用户信息库，获得VM3对应的MAC地址MAC3。
4. 控制器封装ARP应答报文，通过OpenFlow通道发送给NVE1。

5. NVE1根据控制器指定的出端口（ARP请求报文的入端口）将ARP应答报文发送给VM1。

1.2.5 VXLAN QoS

VXLAN QoS用来实现原始报文携带的QoS优先级、设备内部优先级（又称为本地优先级，是设备内部区分报文服务等级的优先级）与封装后报文优先级之间的转换，从而设备根据内部优先级提供有差别的QoS服务质量。

图 1-11 VXLAN 组网



如图1-11所示，VXLAN QoS实现的原始报文携带的QoS优先级、设备内部优先级与封装后报文优先级之间的转换过程如下。

1. 原始报文由二层子接口进入Switch_1设备，原始报文按照子接口上指定的VLAN上绑定的DiffServ模板进行映射，将原始报文的802.1p优先级映射为设备内部优先级（PHB行为和报文颜色），以此入队列。
2. 报文进入隧道，对报文进行加封装（外层依次添加VXLAN报文头、UDP报文头、IP报文头和以太报文头），加封装报文外层的802.1p优先级和DSCP优先级由原始报文内部优先级按照DiffServ域的缺省模板进行映射。报文按照映射后的优先级在隧道中进行传输。
3. 报文出隧道时，按照隧道接口上配置的信任类型802.1p或DSCP（以太网接口处于三层模式时只能信任DSCP），按照DiffServ域的缺省模板进行映射，映射为设备内部优先级，进入队列进行传输。
4. 最后，由设备内部优先级按照子接口上指定的VLAN上绑定的DiffServ域模板进行映射，映射到出接口报文的802.1p优先级，报文按照映射后的优先级进行传输。

1.2.6 VXLAN 增强特性



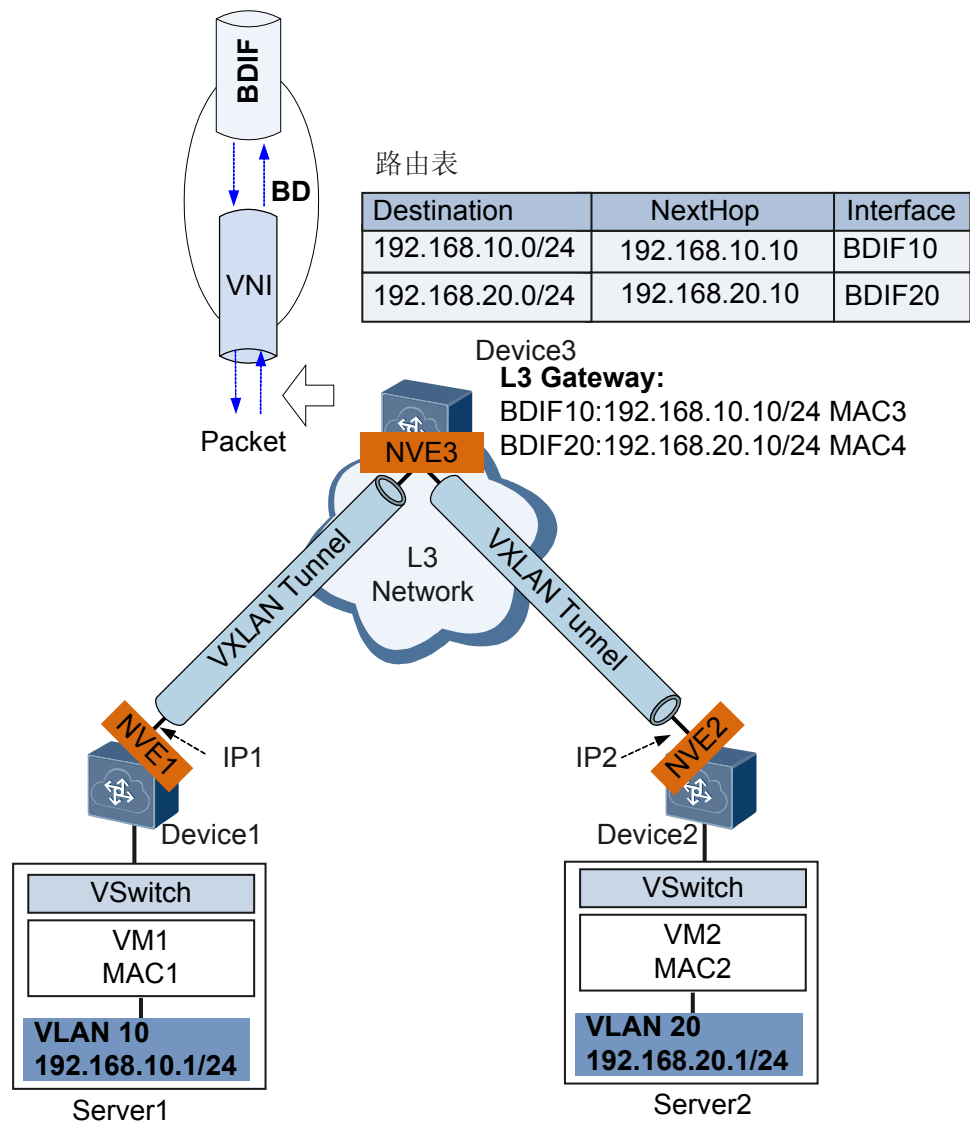
本特性从V100R005C10版本开始支持。

1.2.6.1 ARP/MAC 动态学习

动态ARP/MAC地址表项通过ARP/MAC报文动态学习创建和更新，不需要网络管理员手工维护，大大减少了网络管理员的维护量。

在VXLAN网络中如何通过ARP/MAC动态学习建立动态ARP/MAC地址表项以实现终端租户互通，将基于图1-12详细介绍。

图 1-12 不同网段租户互通实现过程示意图



1. VM1初次和VM2通信，VM1会先发送目的MAC为全F的ARP请求报文。
2. Device1收到ARP请求报文后，更新本地保存的MAC地址表（MAC表中增加VNI ID），并将ARP请求报文在本网段内广播。
3. Device3收到ARP请求报文后，更新本地保存的ARP表项，并向Device1发送源MAC是MAC3的ARP应答报文。
4. Device1收到ARP应答报文后更新本地保存的MAC地址表。

5. VM1收到ARP应答报文后，更新本地保存的ARP表项，并向三层网关发送数据报文。
6. Device1收到报文后，根据本地保存的MAC表，找到出接口是NVE3，进行VXLAN封装，封装后的VXLAN报文如图1-13所示。

图 1-13 VXLAN 报文

DMAC Net MAC	SMAC NVE3 MAC	SIP NVE3	DIP NVE2	UDP S_P HASH	UDP D_P 4789	VNI 600 0	DMAC MAC3	SMAC MAC1	SIP 192.168 .10.1	DIP 192.168 .20.1	Pay- load
--------------------	---------------------	-------------	-------------	--------------------	--------------------	-----------------	--------------	--------------	-------------------------	-------------------------	--------------

7. Device3收到VXLAN报文后先进行解封装，再查找路由表转发报文。
8. 三层网关查找本地保存的ARP表项，发现没有目的租户VM2的IP地址和MAC地址映射关系，三层网关NVE3向NVE2发送ARP请求报文。
9. Device2收到ARP请求报文后，更新本地保存的MAC地址表，并将ARP请求报文在本网段内广播。
10. VM2收到ARP请求报文后，更新本地保存的ARP表项，并向Device2发送ARP应答报文。
11. Device2收到ARP应答报文后更新本地保存的MAC地址表，并向Device3发送ARP应答报文。
12. NVE3收到ARP应答报文，更新本地保存的ARP表项。
13. 三层网关向VM2发送数据报文。三层网关根据本地保存的MAC转发表，找到出接口是NVE2，进行VXLAN封装，封装后的VXLAN报文如图1-14所示。

图 1-14 VXLAN 报文

DMAC Net MAC	SMAC NVE3 MAC	SIP NVE1	DIP NVE3	UDP S_P HASH	UDP D_P 4789	VNI 500 0	DMAC MAC2	SMAC MAC4	SIP 192.168 .10.1	DIP 192.168 .20.1	Pay- load
--------------------	---------------------	-------------	-------------	--------------------	--------------------	-----------------	--------------	--------------	-------------------------	-------------------------	--------------

14. VXLAN报文依据图1-14中的外层路由表转发到Device2。Device2收到VXLAN报文后进行解封装，查找MAC地址表，将数据报文转发至目的租户VM2。

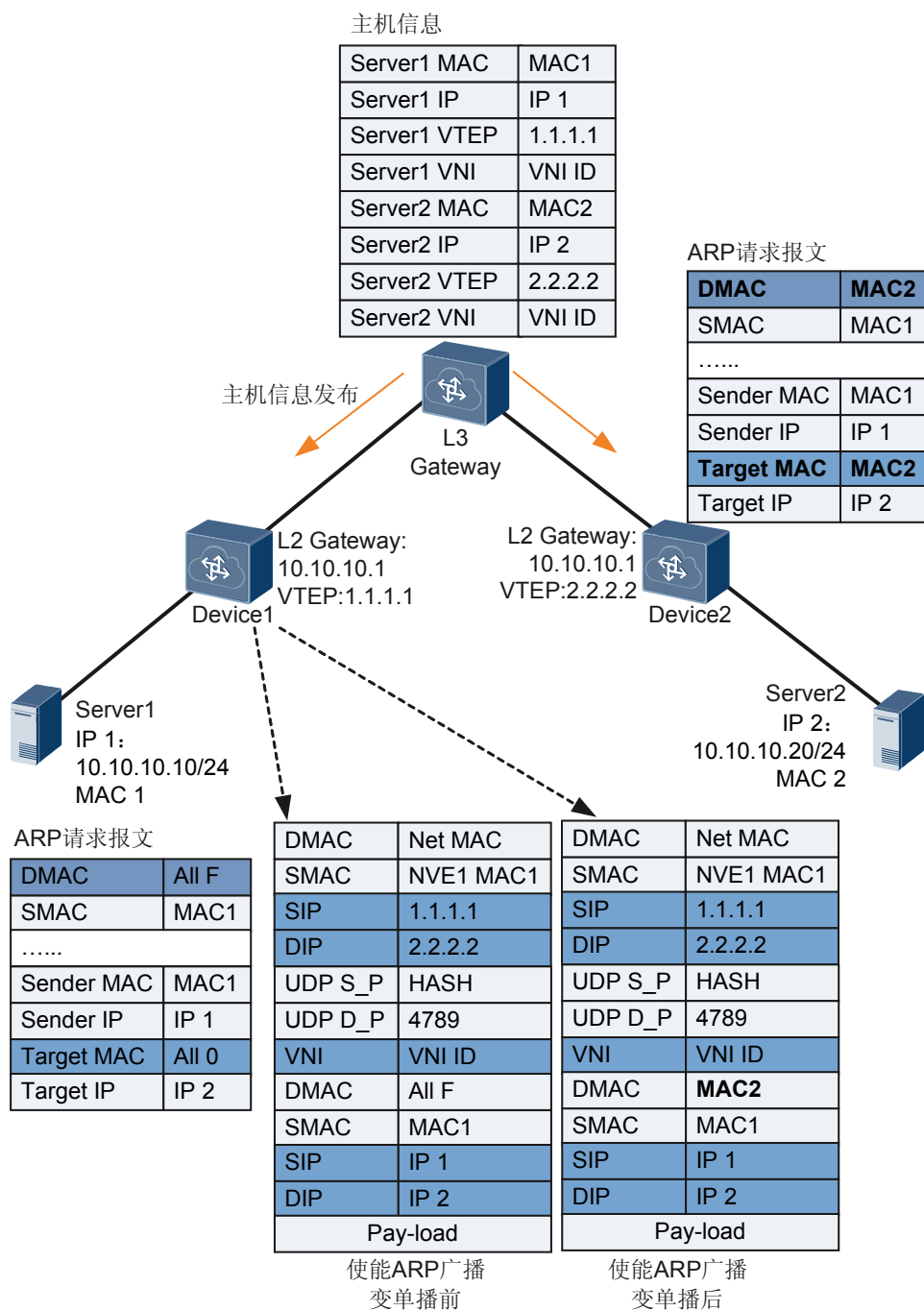
VM2->VM1通信过程与VM1->VM2类似，这里不再赘述。由于VM1->VM2的通信过程中，三层网关、Device1、Device2上都更新了ARP和MAC表项，VM2->VM1通信过程中无需再发送ARP请求，以单播形式通信。

由此实现VM1和VM2之间通过VXLAN三层网关实现跨网段通信。

1.2.6.2 ARP 广播抑制

在终端租户初次互通过程中，终端租户会发送ARP广播请求报文，而ARP请求报文会在二层网络内广播。为了抑制ARP广播请求报文给网络带来的广播风暴，可在VXLAN二层网关设备上使能广播抑制功能。

图 1-15 ARP 广播抑制示意图



如图1-15所示，VXLAN三层网关通过动态学习终端租户的ARP表项，再根据ARP表项生成主机信息（包括主机IP地址、MAC地址、VTEP地址和VNI ID），并将主机信息通过BGP对外发布，使其他的BGP邻居可以学习到主机信息。VXLAN二层网关学习到的主机信息用于广播抑制。

Server1初次访问Server2时，Server1会向Server2发送ARP广播请求报文，请求目的主机Server2的MAC地址，具体实现过程如下：

1. Server1发送ARP请求报文，请求目的主机Server2的MAC地址。

2. 作为VXLAN二层网关的Device1收到ARP请求报文后，查询主机信息。
 - 如果主机信息中有目的主机信息，Device1将ARP请求报文中的广播目的MAC地址替换为目的主机的MAC地址，并进行VXLAN封装后转发。
 - 如果主机信息中没有目的主机信息，ARP请求报文中的广播目的MAC地址不变，Device1进行VXLAN封装后转发。
3. 作为VXLAN二层网关的Device2收到封装后的ARP请求报文进行VXLAN解封装获取内层二层报文，判断报文的目的MAC是否为广播地址。
 - 是，在对应的二层广播域内非VXLAN网络侧进行广播处理。
 - 不是，发送给对应的目的主机。
4. 目的主机Server2收到单播ARP请求报文后，进行ARP应答。
5. Server1收到ARP应答报文建立ARP缓存表，并可以与Server2通信。

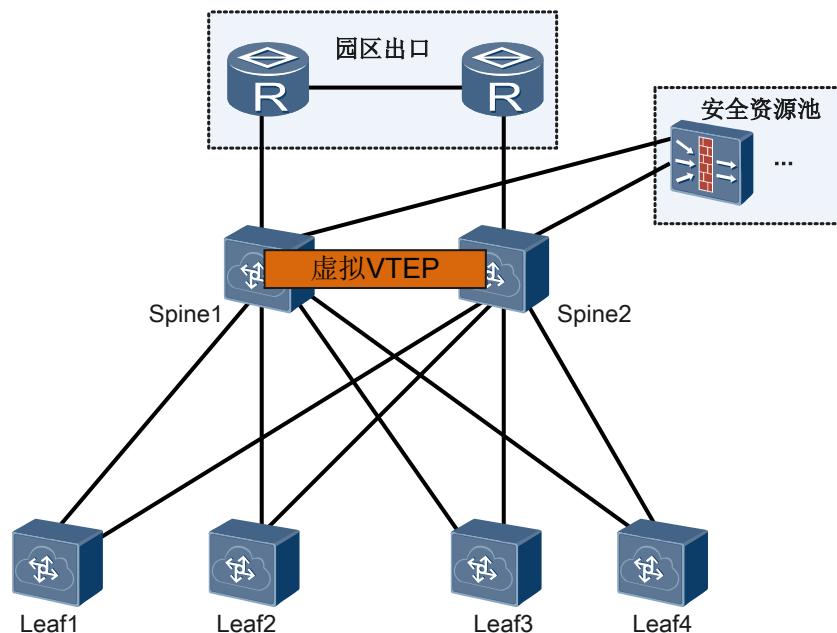
1.2.6.3 VXLAN 集中式多活网关

产生原因

在VXLAN网络中，为了提高可靠性，用户经常会部署多个网关进行主备备份，以保证一台网关设备故障时流量可以及时切换到另外的网关设备上，避免业务中断。

通过部署VRRP，可以解决上述问题。但由于VRRP组网中，仅主网关设备能够进行流量转发，提供网关服务，各网关设备仅在主网关故障后才提供网关服务，导致网关设备利用率低，网关故障收敛性能低。用户希望在保证可靠性的同时，多个网关都可以同时转发流量，充分利用设备资源。

图 1-16 VXLAN 多活网关组网图



通过配置VXLAN集中式多活网关可以解决上述问题。VXLAN集中式多活网关是指在典型的“Spine-Leaf”组网结构下，通过给Spine设备配置相同的VTEP地址，将多个Spine

设备模拟成一个VXLAN隧道端点，然后在所有Spine设备上配置三层网关，使得无论流量发到哪一个Spine设备，该设备都可以提供网关服务，将报文正确转发给下一跳设备；同样，从其他网络访问本网络的流量，无论发到哪一个Spine设备，都能被正确转发给网络内的主机。如[图1-16](#)所示，在Spine1和Spine2配置VXLAN集中式多活网关功能，保证这两台网关设备可以同时转发流量，以提高设备资源利用率和网络故障收敛性能。

采用VXLAN集中式多活网关时，Spine设备作为三层网关设备，所有通过三层转发的终端租户的表项都需要在该设备上生成，而Spine设备的表项规格有限，当VM或服务器数量越来越多时，容易成为瓶颈。因此，VXLAN集中式网关适合部署在中小型网络中。

相关概念

结合[图1-16](#)介绍VXLAN多活网关相关概念：

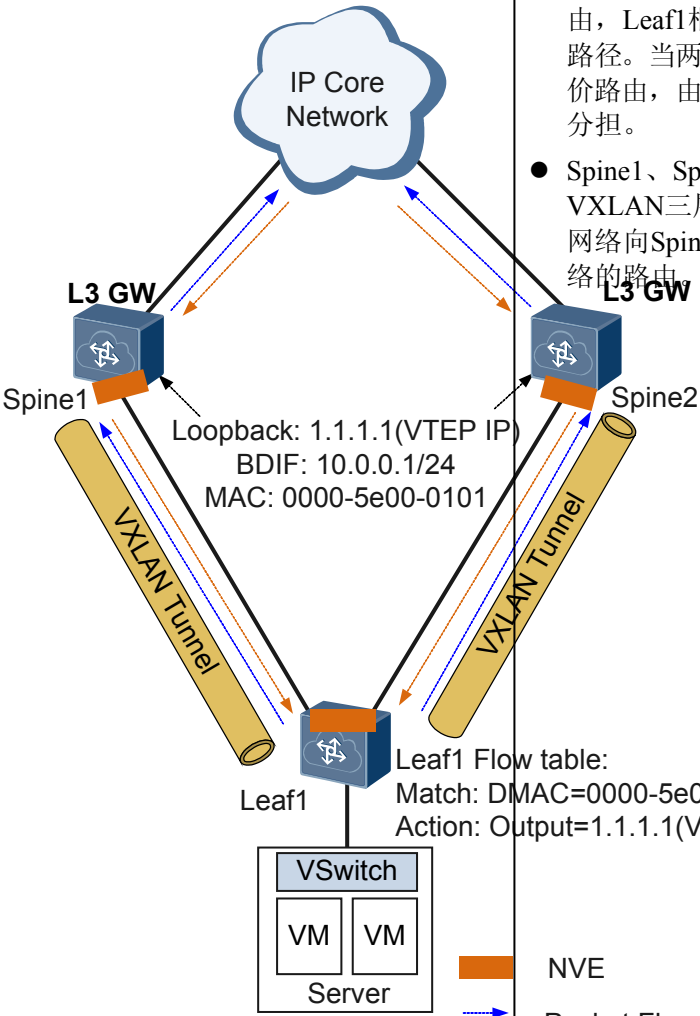
- **Spine**
VXLAN网络中的三层网关设备，通过将VXLAN报文解封后重新转发，实现负责不同子网间的服务器或VM的互相访问以及物理服务器、VM与外部网络的通信。
- **Leaf**
VXLAN网络中的二层接入设备，与物理服务器或VM对接，通过将物理服务器和VM发送过来的报文封装在VXLAN报文中，将对应的流量接入VXLAN网络中。
- **虚拟VTEP**
在VXLAN多活网关方案中，当手动配置多个网关设备为相同的VTEP后，多台网关设备将形成一个多活网关设备组，对外表现为一个虚拟的VTEP。

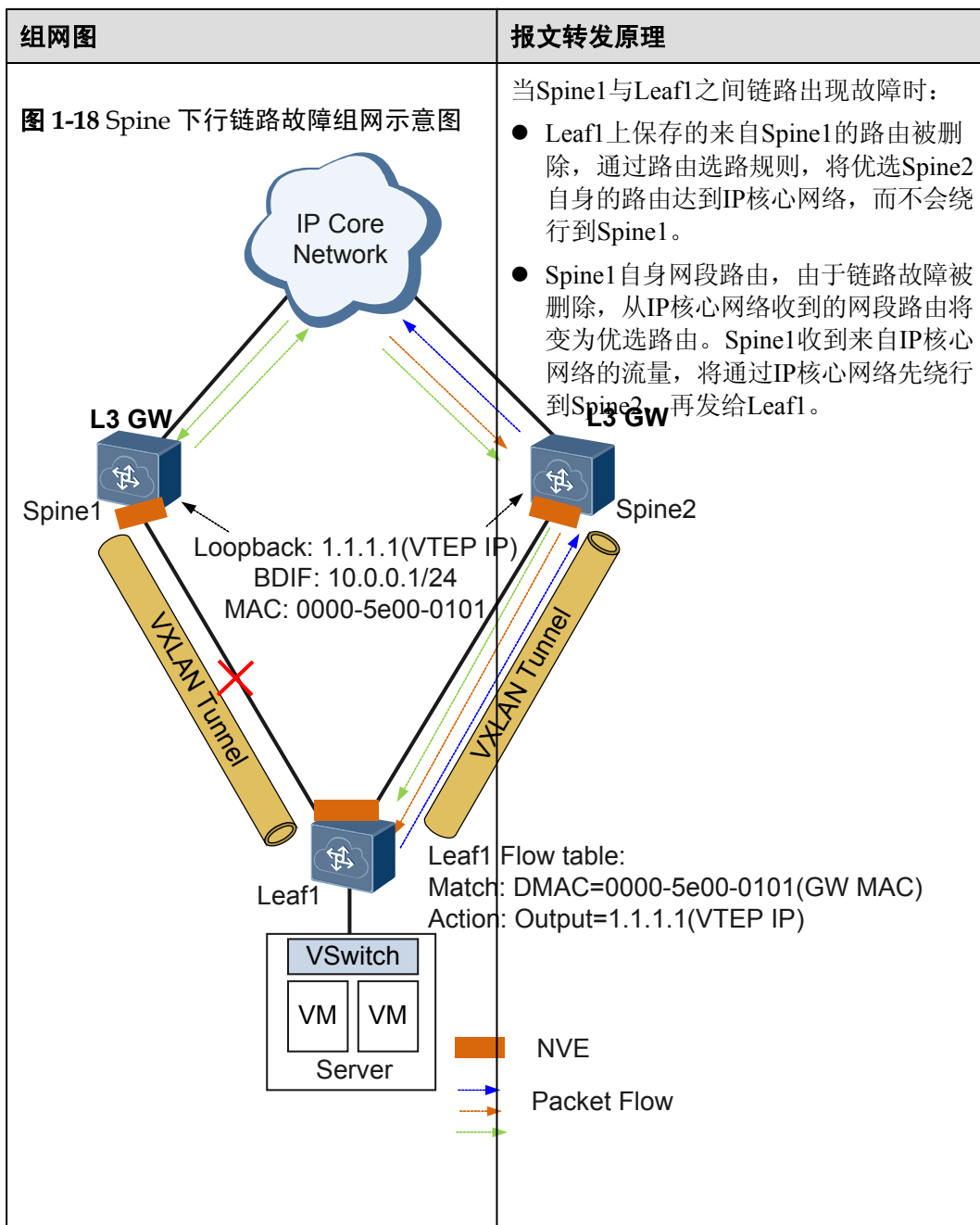
多活网关报文转发原理

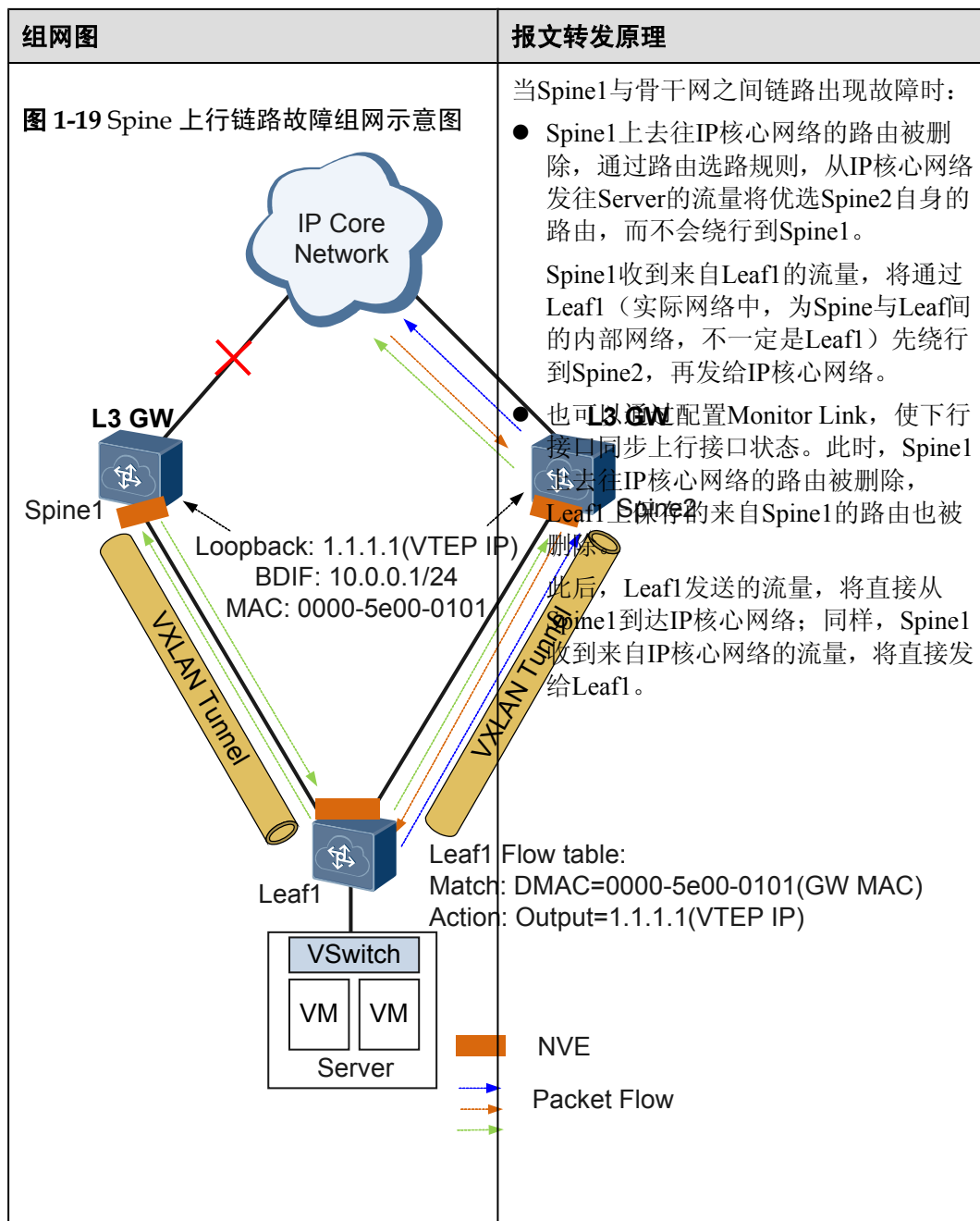
如[表1-2](#)所示，Spine1、Spine2分别与Leaf1之间建立VXLAN隧道，在Spine1、Spine2上配置VXLAN三层网关功能，为Leaf1下的物理服务器或VM提供不同子网间的服务器或VM的互相访问以及物理服务器、VM与外部网络的通信功能。Spine1、Spine2共用BDIF接口地址、MAC地址以及源VTEP地址。

在网络正常和故障情况下，多活网关报文转发原理如[表1-2](#)所示。

表 1-2 多活网关报文转发原理

组网图	报文转发原理
<p>图 1-17 链路正常组网示意图</p> 	<p>当设备之间的网络通信正常时：</p> <ul style="list-style-type: none"> ● Leaf1通过路由协议学习到两条网关路由，Leaf1根据路由选路规则选择最优路径。当两条路径开销值一样时形成等价路由，由此实现链路备份和流量负载分担。 ● Spine1、Spine2向IP核心网络通告 VXLAN三层网关的网段路由，IP核心网络向Spine1、Spine2通告来自其他网络的路由。





ARP 表项同步

在多活网关中，多个网关都会给上层的路由设备发布相同子网的网段路由，最后在上层路由设备上形成到指定网段地址的ECMP。从上层路由设备发往网关的流量会通过ECMP发给某一个网关，若此网关上没有到目的主机的ARP表项，则会导致ARP报文泛洪和流量丢弃。

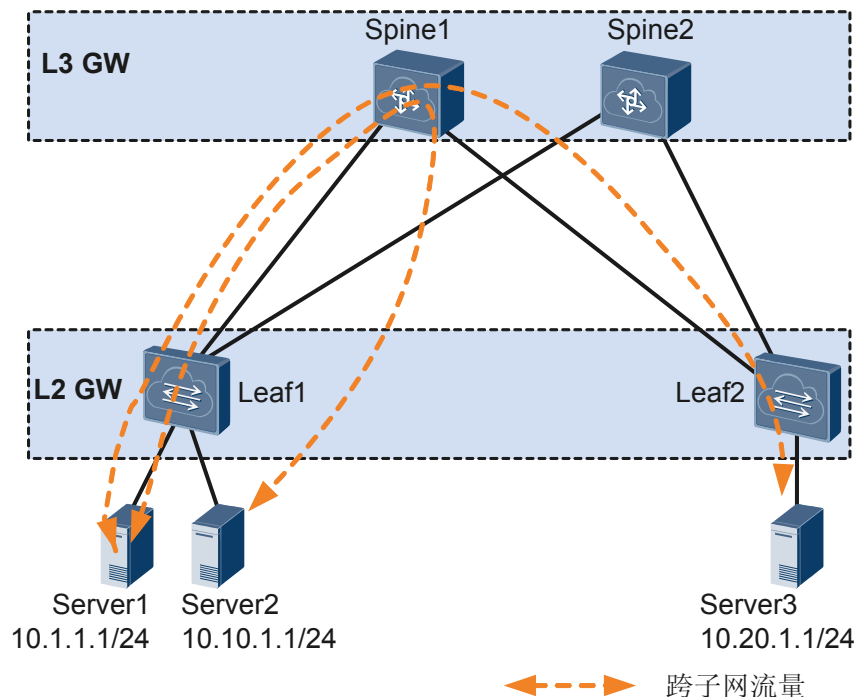
为了保证流量正常转发，所有的多活网关设备需要实现ARP表项同步，即网关所属于子网内任一主机上线，所有网关都学习到此主机的ARP。目前，设备支持以下两种方式来实现ARP表项同步：

- 控制器方式：所有多活网关设备关闭BDIF接口下的ARP动态学习功能，由控制器统一进行管理和控制。当主机上线时，控制器可获取到主机的ARP，并向所有网关设备同步下发ARP表项。
- 单机方式：不依赖控制器，由设备自动进行学习，具体原理如下：
 1. 建立多活网关邻居
 - a. 用户在DFS下指定该网关设备的所有邻居的IP地址后，此设备开始与指定IP地址的设备建立邻居关系。
 - b. 邻居关系建立后，此设备会自动在设备与所有邻居之间创建单播的快速同步通道，用于网关之间ARP表项的同步。
 - c. 设备开始周期性发送心跳报文，更新邻居状态。
 2. ARP表项学习与同步
 - a. 当网关设备接收到ARP请求报文时，该设备首先判断该报文是否为经过VXLAN封装的ARP报文、多活网关功能是否使能，若两个条件均满足，则将报文发送至VXLAN功能的ARP解析模块。
 - b. ARP解析模块解析报文后获取报文中的广播域和VNI信息，根据VNI信息找到对应的网关接口，继续判断网关类型。
 - c. 若是分布式网关，则按照分布式网关方式进行同步，具体请参见[VXLAN 分布式网关](#)；若不是分布式网关，则根据ARP信息生成主机路由，同时构造同步报文（包含ARP报文、VNI、接口类型、接口ID和网关类型信息）。
 - d. 通过快速同步通道，将同步报文发送给其他的所有邻居。
 - e. 邻居设备接收到同步报文后，根据报文中的网关类型判断是否为分布式网关；若是，则按照分布式网关方式进行同步，具体请参见[VXLAN 分布式网关](#)；若不是分布式网关，则直接学习报文中的ARP信息，生成ARP表项，实现同步。

1.2.6.4 VXLAN 分布式网关

产生原因

图 1-20 VXLAN 集中式网关示意图

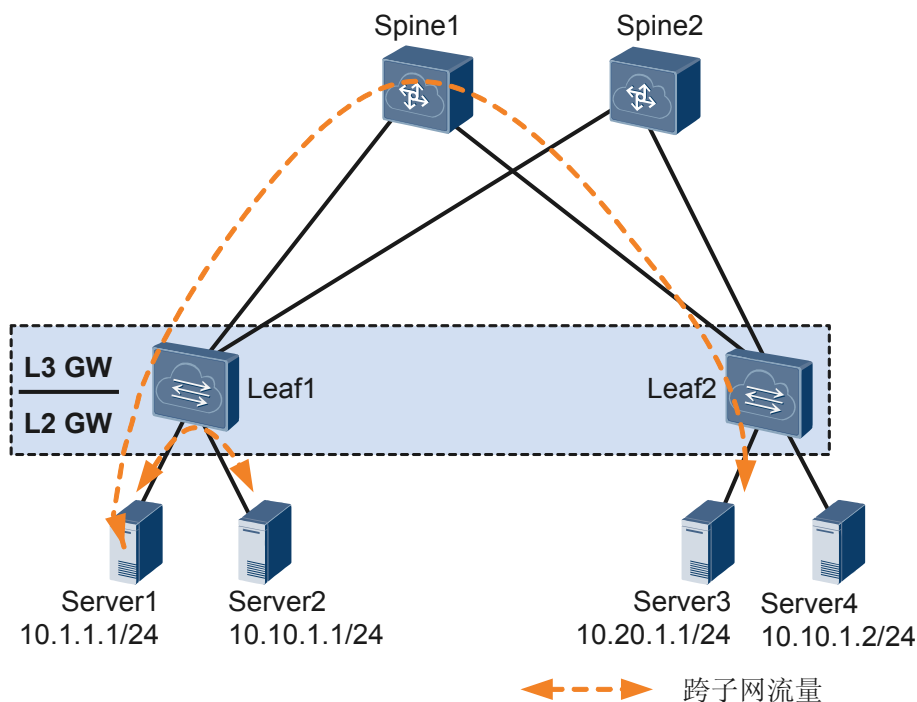


传统的集中式三层网关将服务器的网关设置在汇聚或者Spine节点，如图1-20所示，跨子网的报文都必须经过Spine节点转发，若三层网关集中部署，存在如下问题：

- 转发路径不优化：异地数据中心三层流量都需要经过集中三层网关转发。
- ARP表项规格瓶颈：由于采用集中三层网关，通过三层网关转发的终端租户的ARP表项都需要在三层网关上生成，而三层网关上的ARP表项规格有限，这不利于数据中心网络扩展。

通过配置VXLAN分布式网关可以解决上述问题。VXLAN分布式网关是指在典型的“Spine-Leaf”组网结构下，将Leaf节点作为VXLAN隧道端点VTEP，每个Leaf节点都可作为VXLAN三层网关，Spine节点不感知VXLAN隧道，只作为VXLAN报文的转发节点。如图1-21所示，Server1和Server2不在同一个网段，但是都下挂在Leaf1节点下。Server1和Server2通信时，流量只需要在Leaf1节点进行转发，不再需要经过Spine节点。

图 1-21 VXLAN 分布式网关示意图



VXLAN分布式网关具有如下特点：

- 同一个Leaf节点既可以做VXLAN二层网关，也可以做VXLAN三层网关，部署灵活。
- Leaf节点只需要学习自身下挂服务器的ARP表项，而不必像集中三层网关一样，需要学习所有服务器的ARP表项，解决了集中式三层网关带来的ARP表项瓶颈问题，网络规模扩展能力强。
- 如果有相同子网的服务器在不同Leaf节点下，在Leaf节点上配置三层网关，需要配置相同的网关IP地址、MAC地址，实现终端租户只感知一台三层网关。当终端租户或服务器移动位置，不需要更改服务器的三层网关配置，作为三层网关的Leaf节点也不需要刷新ARP表项，减少了维护工作量。

相关概念

VXLAN分布式网关由Leaf和Spine组成，结合图1-21介绍Leaf节点和Spine节点在VXLAN分布式网关场景中的作用：

- **Spine**
Spine节点关注于高速IP转发，强调的是设备的高速转发能力。
- **Leaf**
 - 作为VXLAN网络中的二层接入设备，与物理服务器或VM对接，用于解决终端租户接入VXLAN虚拟网络的问题。
 - 作为VXLAN网络中的三层网关设备，需要绑定VPN实例，VXLAN隧道的建立依赖于VPN邻居的建立。三层网关进行VXLAN报文封装/解封装，实现跨子网的终端租户通信，以及外部网络的访问。

报文转发原理

- 同子网报文转发

VXLAN分布式网关场景下同子网的BUM（Broadcast&Unknown-unicast&Multicast）报文、已知单播报文转发原理同集中式网关场景下报文原理，详细描述请参考[BUM报文转发流程](#)和[已知单播报文转发流程](#)。

为了抑制ARP广播流量，可在Leaf节点上部署ARP广播报文变单播报文功能，详细描述请参考[1.2.6.2 ARP广播抑制](#)。

- 跨子网报文转发

- 控制平面

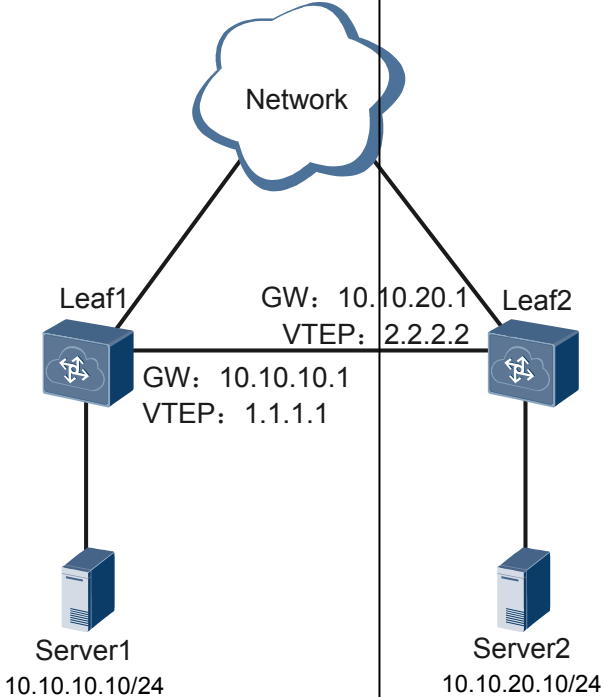
VXLAN分布式网关场景下跨子网互通必须通过三层转发，这就要求Leaf节点间必须互相学习到主机路由。为了实现跨子网互通，控制平面需要满足[表1-3](#)所示要求。

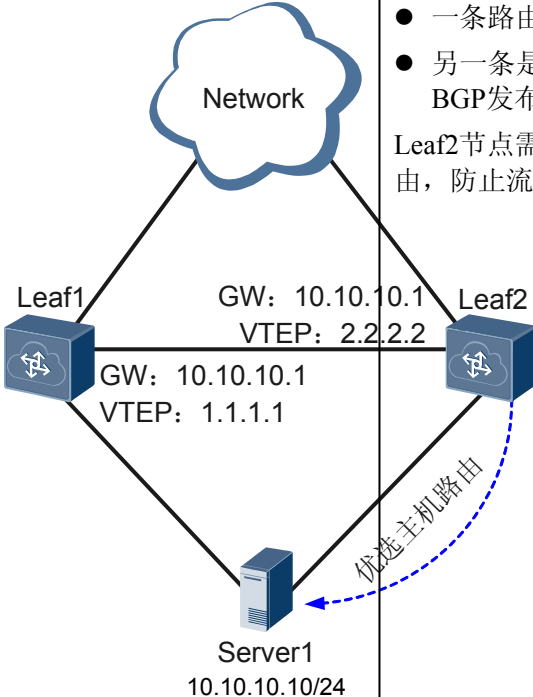
表 1-3 控制平面要求

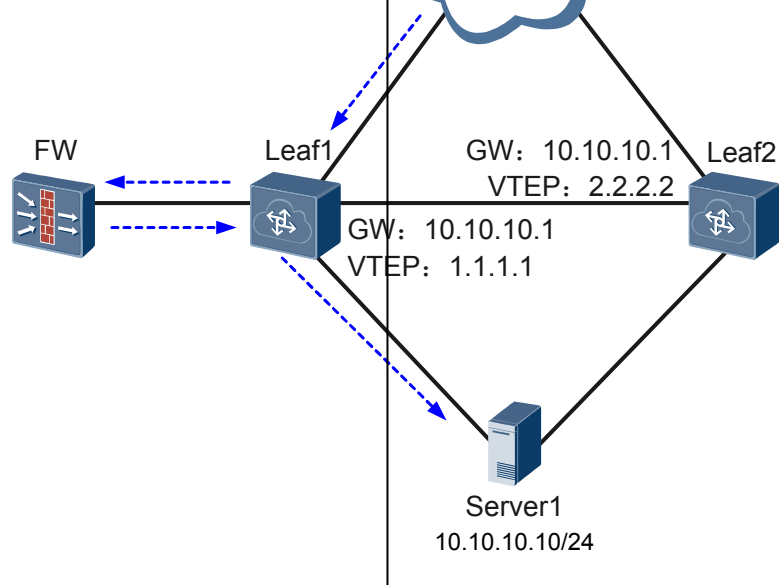
控制平面要求	组网图	报文转发原理
ARP本地主机路由发布	<p>图 1-22 ARP 发布本地路由组网图</p>	<p>作为VXLAN三层网关的Leaf节点学习终端租户的ARP表项，再根据ARP表项生成主机路由，并将主机路由通过BGP对外发布，使其他的BGP邻居可以学习到主机路由。</p>

控制平面要求	组网图	报文转发原理
<p>ARP学习</p>	<p>图 1-23 ARP 学习组网图</p> <p>① Server1发送ARP请求报文</p> <p>② Leaf1收到ARP请求报文，在二层网络内广播，并学习Server1的ARP</p> <p>③ Leaf1回应Server1的ARP请求</p> <p>④ Leaf2从VXLAN隧道侧收到Server1的ARP请求不学习</p>	<p>同一个子网的网关部署在不同的Leaf节点，由主机所在的Leaf节点发布主机路由，其他Leaf节点不能发布该主机的主机路由，防止把访问主机的网络流量引入到非本地的Leaf节点上。该问题可通过Leaf节点学习主机ARP发布主机路由解决。</p> <p>分布式VXLAN网关场景下，Leaf节点只学习用户侧的ARP，并发布主机路由，不学习VXLAN隧道侧的ARP报文。</p>

控制平面要求	组网图	报文转发原理
<p>分布式网关 VXLAN隧道管理</p>	<p>图 1-24 分布式网关 VXLAN 隧道管理组网图</p>	<p>在Leaf节点上，除了给每个子网分配VNI，还会给每个租户（VPN实例）分配一个三层VNI。当跨Leaf节点进行三层转发时，VNI10000通过VXLAN隧道传输到远端Leaf节点，远端Leaf节点通过租户VNI信息来识别VPN，这样即可识别租户是否属于同一个VPN，以及是否需要互通或隔离。</p> <p>以Server1访问其他子网Server为例描述分布式网关VXLAN隧道管理。</p> <ol style="list-style-type: none"> Leaf1学习到Server1的ARP生成主机路由，BGP通过remote-nexthop路径属性发布Server1的主机路由给其他BGP邻居。 Leaf2收到Leaf1发送的主机路由后，根据路由中Leaf1的VNI ID、VTEP的IP地址，结合本地VPN实例的VNI ID、VTEP IP地址动态创建VXLAN隧道，VNI20010。 Leaf1收到Server1发送的报文，Leaf1查找主机路由表，找到去往目的主机的主机路由，根据路由中的VNI ID，结合本地VPN实例的VNI ID、VTEP IP地址动态创建VXLAN隧道，VNI10000。 <p>Leaf1收到Server1发送的报文，Leaf1查找主机路由表，找到去往目的主机的主机路由，根据路由中的VNI ID，结合本地VPN实例的VNI ID、VTEP IP地址动态创建VXLAN隧道，VNI10000。Leaf1封装将报文发送给目的主机所在的Leaf节点，目的主机的Leaf节点VXLAN解封装后，通过查找路由表将报文发送给目的主机。</p> <p>当Leaf1下不再有主机，Leaf1将撤销所有的主机路由，Leaf2发现Leaf1不再有可达的主机路由，将动态删除Leaf之间的VXLAN隧道。</p>

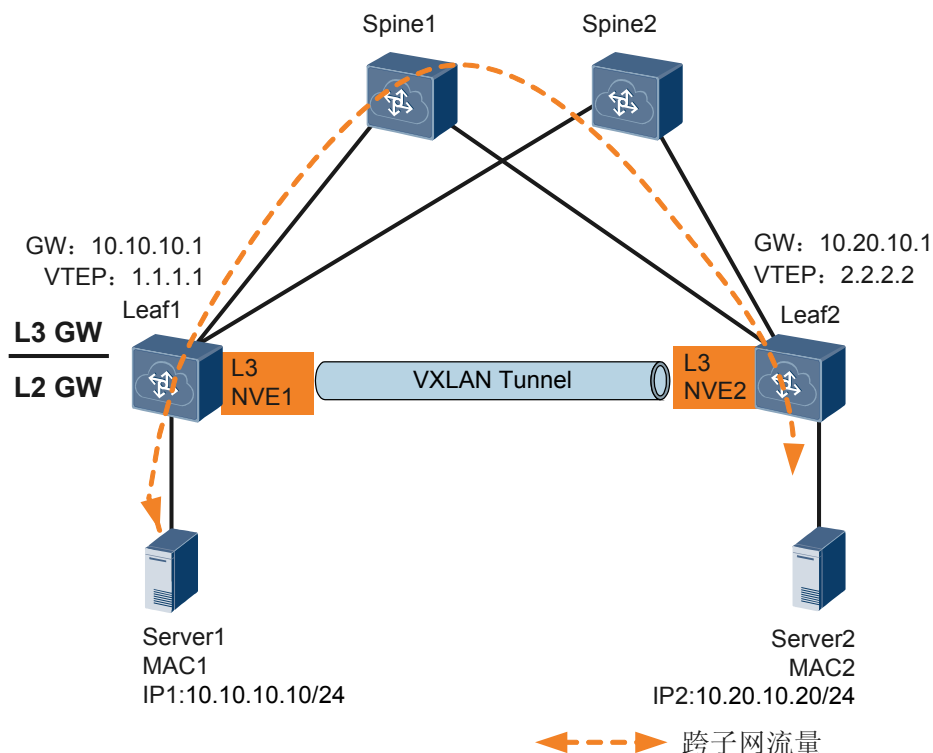
控制平面要求	组网图	报文转发原理
<p>路由优先级控制</p>	<p>图 1-25 BGP 控制路由优先级组网图</p>  <p>The diagram illustrates a VXLAN network topology. At the top is a cloud labeled 'Network'. Below it are two leaf switches, 'Leaf1' and 'Leaf2', connected to the network. Leaf1 is connected to 'Server1' (IP: 10.10.10.10/24) and Leaf2 is connected to 'Server2' (IP: 10.10.20.10/24). Leaf1 has a gateway IP of 10.10.10.1 and a VTEP IP of 1.1.1.1. Leaf2 has a gateway IP of 10.10.20.1 and a VTEP IP of 2.2.2.2. The network cloud is connected to both leaf switches.</p>	<p>Leaf节点通过BGP向对端发布主机路由时，可以根据BGP机制控制主机路由的优先级。</p>

控制平面要求	组网图	报文转发原理
	<p>图 1-26 主机路由优先级控制组网图</p>  <p>The diagram illustrates a network topology for host route priority control. At the top is a cloud labeled 'Network'. Below it are two switches, Leaf1 and Leaf2, connected to the network. Leaf1 is on the left and Leaf2 is on the right. A central horizontal line represents the connection between Leaf1 and Leaf2, with 'GW: 10.10.10.1' and 'VTEP: 2.2.2.2' written above it. Below Leaf1, its configuration is listed as 'GW: 10.10.10.1' and 'VTEP: 1.1.1.1'. Below Leaf2, its configuration is listed as 'GW: 10.10.10.1' and 'VTEP: 2.2.2.2'. At the bottom center is a server icon labeled 'Server1' with the IP address '10.10.10.10/24'. A dashed blue arrow points from Leaf2 to Server1, labeled '优选主机路由' (Preferred Host Route).</p>	<p>主机双活接入场景，Leaf2节点上的路由表中存在两条目的IP是Server1的路由：</p> <ul style="list-style-type: none"> ● 一条路由是直连的主机路由 ● 另一条是Leaf2节点通过BGP发布的路由 <p>Leaf2节点需要优先选择主机路由，防止流量绕行。</p>

控制平面要求	组网图	报文转发原理
	<p>图 1-27 业务节点引流组网图</p> 	<p>在业务节点引流场景下，外部访问Server1的流量需要绕行到业务节点FW，然后再回到Leaf节点。这种情况下，主机路由优先级必须低于引流的策略路由的优先级。</p>

- 转发平面

图 1-28 跨子网报文转发示意图



Server1发出 跨子网IP报文		Leaf1封装VXLAN报文		Leaf2解封封装VXLAN报文	
DMAC	GW MAC	DMAC	Net MAC	DMAC	MAC2
SMAC	MAC1	SMAC	NVE1 MAC1	SMAC	Leaf2 MAC
SIP	IP 1	SIP	1.1.1.1	SIP	IP 1
DIP	IP 2	DIP	2.2.2.2	DIP	IP 2
Pay-load		UDP S_P	HASH	Pay-load	
		UDP D_P	4789		
		VNI	VPN VNI ID		
		DMAC	Leaf2 MAC		
		SMAC	Leaf1 MAC		
		SIP	IP 1		
		DIP	IP 2		
		Pay-load			

如图1-28所示，VXLAN分布式网关下跨子网报文转发实现过程如下：

1. Leaf1收到Server1发出的跨子网IP报文后，进行VXLAN隧道封装。将租户子网所属的VNI映射为VPN实例所属的VNI。根据租户报文信息查找网关出接口，根据接口绑定的VPN实例进行报文三层转发，主机路由的下一跳地址是Leaf2的VTEP地址。
2. Leaf2收到VXLAN报文后进行解封装，根据VNI映射VPN实例，在VPN内查找主机路由，替换以太头发给目标主机。

1.2.6.5 VXLAN 双活接入

产生原因

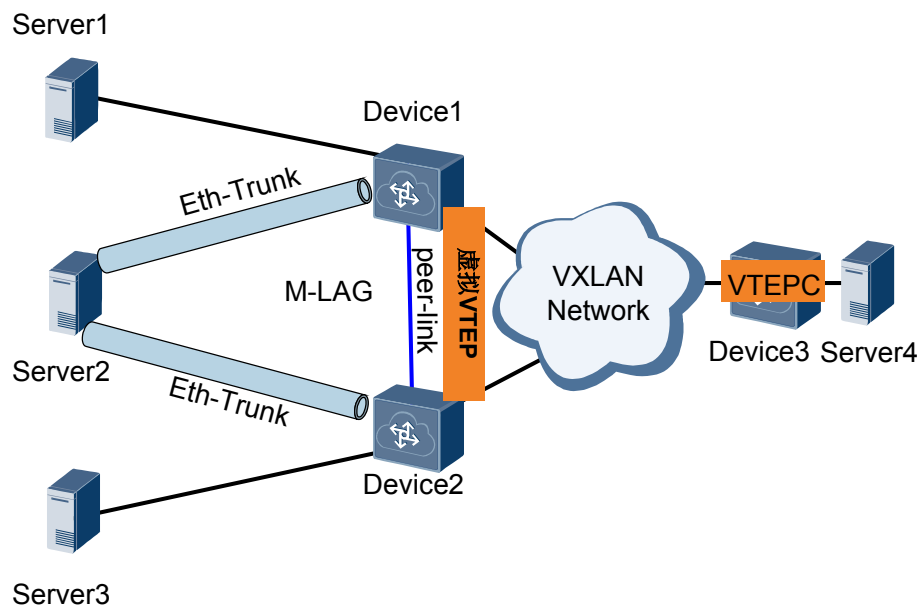
在VXLAN网络中，为了提高可靠性，用户经常采用双归接入的方式将安装有双网卡的服务器接入到VXLAN网络，使得当服务器的一个网卡发生故障时不会导致业务中断。

由于上述方案中，仅主网卡能够进行数据报文收发，备网卡不能进行数据报文收发，导致了网卡和链路带宽资源的浪费。企业管理员希望两个网卡可以同时转发流量，实现双活，从而充分利用网卡和网络带宽资源。

此时，将会存在以下问题：

- 问题1：服务器可能从两个连接服务器的端口收到相同的网络侧发送过来的流量，造成冗余。
- 问题2：与该服务器通信的网络侧设备由于不断收到两台设备发送过去的流量，因此其设备上会不断产生MAC地址漂移现象。

图 1-29 VXLAN 双活接入组网图



通过配置VXLAN双活接入可以解决上述问题。如图1-29所示，Server2采用双归接入的方式接入到VXLAN网络：

- 对于问题1，通过M-LAG技术将服务器双归的两台接入设备虚拟成一台设备，消除了冗余路径。
- 对于问题2，通过虚拟VTEP技术，两台双归的设备使用的是同一个虚拟VTEP，对于远端设备，相当于是通过一台逻辑设备接入到VXLAN网络中，从而消除了MAC地址漂移现象。

相关概念

结合图1-29介绍VXLAN双活接入相关概念：

- **虚拟VTEP**

在VXLAN双活接入方案中，当手动为双归设备配置相同的VTEP IP地址后，设备采用相同VTEP的IP地址封装在VXLAN报文中，对于VXLAN网络的其他设备而言，这两台设备就被虚拟成了一台设备。
- **peer-link**

在VXLAN双活接入方案中，部署Eth-Trunk的两台设备之间必须存在一条直连链路，且该链路必须配置为peer-link。peer-link链路是一条保护链路。

当接口配置为peer-link接口后，设备会自动在该接口上创建QinQ子接口，该接口上不能再配置具体业务。

当流量进入Peer-Link接口时，对于IP报文，按照DSCP优先级进行映射；对于非IP报文，按照命令**port priority**配置的优先级进行映射。
- **DFS Group**

动态交换服务组DFS（Dynamic Fabric Service） Group，主要用于设备之间的配对，确保VXLAN双活接入场景正常运行。
- **M-LAG接口**

指的是部署M-LAG的两台设备上连接接入服务器的Eth-Trunk接口。

接入侧 M-LAG 工作原理

以下介绍M-LAG涉及的协议报文以及它们的作用。

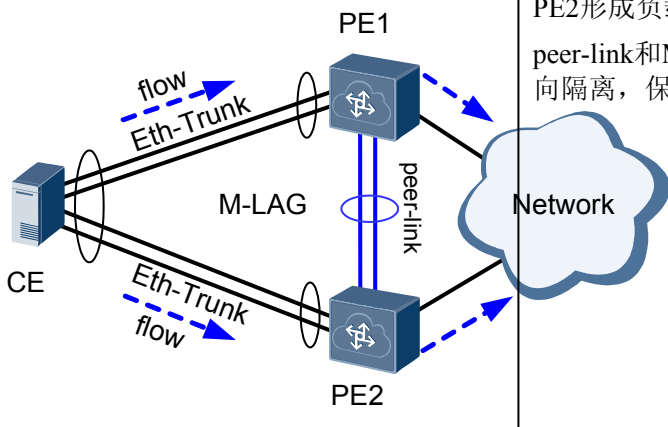
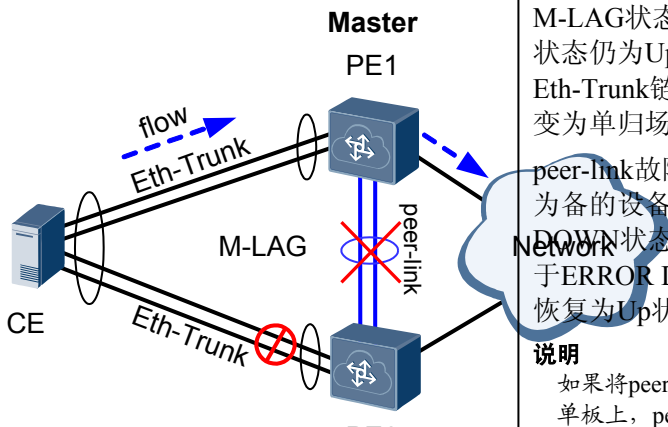
- **M-LAG协商报文**

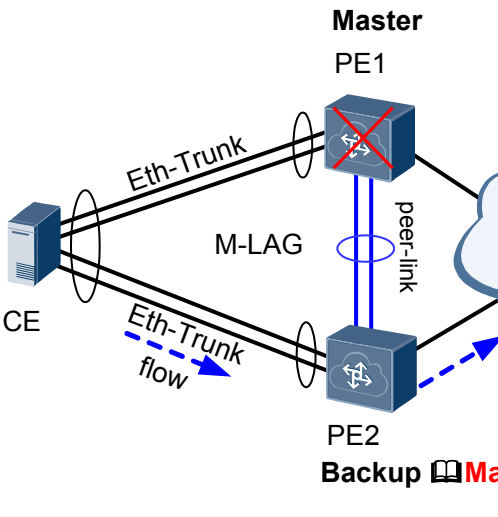
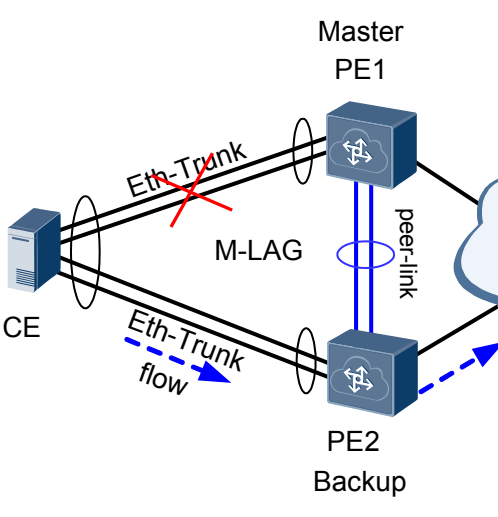
如**图1-29**所示，M-LAG的配置完成后，M-LAG协商报文通过peer-link链路进行交互，先进行DFS Group的配对，配对成功后协商出主备状态。
- **M-LAG心跳报文**

如**图1-29**所示，完成M-LAG状态的协商后，M-LAG心跳报文通过网络侧链路周期性的检测对端状态，保证正常工作。

根据M-LAG协议报文的情况，在网络正常和故障情况下，M-LAG的主备状态及链路状态的确定如**表1-4**所示。

表 1-4 确定 M-LAG 的主备状态及链路状态

组网图	确定成员口Eth-Trunk的主备状态及链路状态
<p>图 1-30 无故障组网示意图</p> 	<p>在VXLAN双活接入场景且为无故障状态下，Eth-Trunk的链路状态均为Up，PE1与PE2形成负载分担，共同进行流量转发。peer-link和M-LAG以及网络侧分别进行单向隔离，保证网络中没有环路。</p>
<p>图 1-31 peer-link 故障组网示意图</p> 	<p>当peer-link故障时，M-LAG主备状态决定了Eth-Trunk的链路状态。M-LAG状态为主的设备侧Eth-Trunk链路状态仍为Up。M-LAG状态为备的设备侧Eth-Trunk链路状态变为Down，双归场景变为单归场景。peer-link故障但心跳状态正常会导致状态为备的设备上M-LAG接口处于ERROR DOWN状态。一旦peer-link故障恢复，处于ERROR DOWN状态的物理接口将自动恢复为Up状态。</p> <p>说明 如果将peer-link接口和M-LAG接口部署在同一单板上，peer-link故障同时M-LAG接口也会故障，这时双归接入的两侧Eth-Trunk链路状态均变为Down，将导致业务流量不通、业务中断。为了提高可靠性，建议将peer-link接口和M-LAG接口的成员接口均部署在不同的单板上。</p>

组网图	确定成员口Eth-Trunk的主备状态及链路状态
<p>图 1-32 M-LAG 状态为主的设备故障组网示意图</p> 	<p>当M-LAG状态为主的设备发生故障时： 通过M-LAG机制，M-LAG状态为备的设备将升级为主，其设备侧Eth-Trunk链路状态仍为Up，流量转发状态不变，继续转发流量。M-LAG状态为主的设备侧Eth-Trunk链路状态变为Down，双归场景变为单归场景。</p> <p>说明 如果是状态为备的设备发生故障，M-LAG的主备状态不会发生变化，M-LAG状态为备的设备侧Eth-Trunk链路状态变为Down。M-LAG状态为主的设备侧Eth-Trunk链路状态仍为Up，流量转发状态不变，继续转发流量，双归场景变为单归场景。</p>
<p>图 1-33 Eth-Trunk 链路故障组网示意图</p> 	<p>当接入VXLAN网络的Eth-Trunk链路发生故障时：</p> <ul style="list-style-type: none"> ● M-LAG主备状态不会变化，流量切换到另一条链路上进行转发。 ● 发生故障的Eth-Trunk链路状态变为Down。 <p>通过M-LAG机制，发生故障的Eth-Trunk链路不再转发流量，VXLAN双归场景变为单归场景。</p>

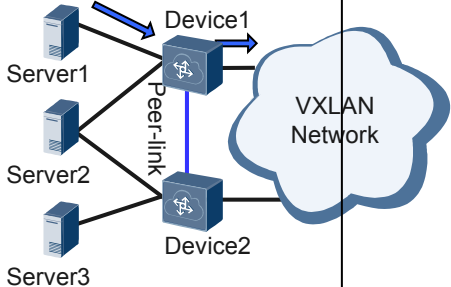
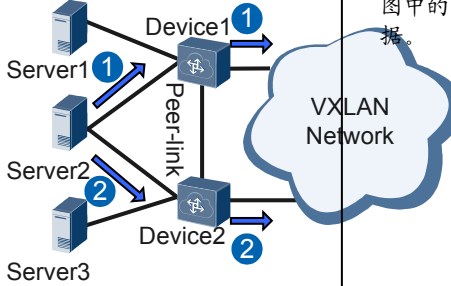
VXLAN 双活接入报文转发流程

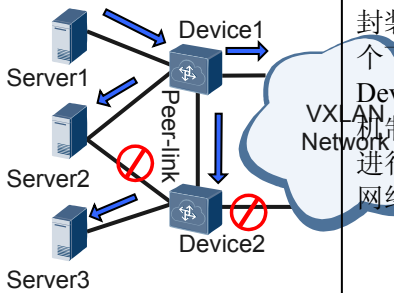
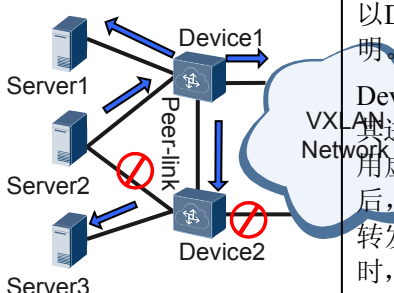
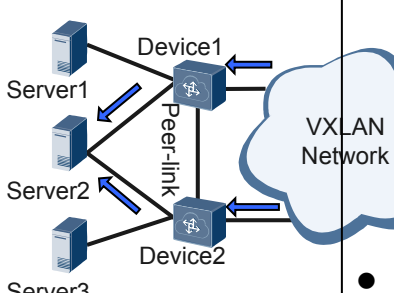
在VXLAN双活接入方案中，在Device1和Device2设备上，通过手动配置方式配置相同的虚拟VTEP，使Device1、Device2进行VXLAN报文封装时采用的是虚拟VTEP。

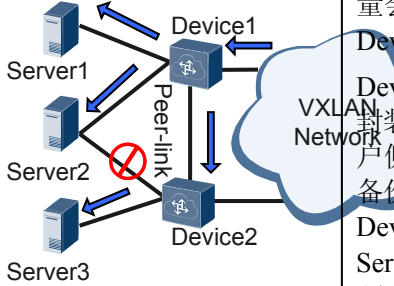
同时，Device1和Device2会进行配对，如果两端的虚拟VTEP一致则匹配成功，否则匹配不成功。配对成功之后，两端互相交换真实的VTEP及MAC地址，当peer-link或某端设备发生故障时，VXLAN协议可以尽快感知并通知另一方，解除双活接入。

如图1-29所示，Device1与Device2之间部署peer-link，Device1与Device2有相同的虚拟VETP，则Server2与Device1、Device2共同构成了VXLAN双活接入方案。对于网络中来自各个方向、不同类型的流量，VXLAN协议会做出不同的处理，具体处理过程如表1-5所示。

表 1-5 VXLAN 双活接入流量处理区分表

流量类型	流量转发示意图	VXLAN双活接入对流量的处理方式
来自非双活端口的单播流量	<p>图 1-34 来自非双活端口的单播流量</p> 	按照正常的单播流量转发流程进行转发。
来自双活端口的单播流量	<p>图 1-35 来自双活端口的单播流量</p>  <p>说明 图中的1和2代表不同的流量数据。</p>	Device1和Device2形成负载均衡，共同进行流量转发。

流量类型	流量转发示意图	VXLAN双活接入对流量的处理方式
来自非双活端口的BUM流量	<p>图 1-36 来自非双活端口的BUM流量</p> 	<p>Device1收到BUM流量后对其进行报文封装，封装时采用虚拟的VTEP。</p> <p>封装完成后，Device1向各个下一跳转发，当流量到达Device2时，通过单向隔离机制，流量只会向Server3进行转发，不会向VXLAN网络和Server2进行转发。</p>
来自双活端口的BUM流量	<p>图 1-37 来自双活端口的BUM流量</p> 	<p>Server2发送的BUM流量会在Device1、Device2之间采用负载分担方式转发，此处以Device1转发为例进行说明。</p> <p>Device1收到BUM流量后对其进行报文封装，封装时采用虚拟的VTEP。封装完成后，Device1向各个下一跳转发，当流量到达Device2时，通过单向隔离机制，流量只会向Server3进行转发，不会向VXLAN网络侧和Server2进行转发。</p>
来自VXLAN网络的单播流量	<p>图 1-38 来自VXLAN网络的单播流量</p> 	<ul style="list-style-type: none"> 对于网络侧发往非双活端口的单播流量，以发往Server1为例，由于流量采用虚拟VTEP进行封装，流量会负载分担到Device1和Device2，发送至Device2的流量会通过peer-link将流量发送至Device1，再通过Device1发往Server1。 对于网络侧发往双活端口的单播流量，由于流量采用虚拟VTEP进行封装，流量会负载分担到Device1和Device2，然后发送至双活接入的设备。

流量类型	流量转发示意图	VXLAN双活接入对流量的处理方式
来自VXLAN网络的BUM流量	<p>图 1-39 来自 VXLAN 网络的 BUM 流量</p>  <p>The diagram illustrates a network topology where a central 'VXLAN Network' (represented by a cloud) sends traffic to two devices, 'Device1' and 'Device2'. These devices are connected to three servers: 'Server1', 'Server2', and 'Server3'. A 'Peer-link' connects Device1 and Device2. Blue arrows show traffic from the VXLAN Network to both Device1 and Device2. From Device1, traffic goes to Server1 and Server2. From Device2, traffic goes to Server2 and Server3. A red 'X' is placed over the link between Device2 and Server2, indicating that traffic is not forwarded to Server2 to avoid a loop.</p>	<p>对于网络侧发往非双活端口的BUM流量，由于流量采用虚拟VTEP进行封装，流量会负载分担到Device1和Device2，以Device1为例。Device1首先对报文进行解封装，之后发送到每一个用户侧端口，由于peer-link与备份接口形成了隔离，到达Device2的流量不会向Server2转发，避免了路由环路。</p>

1.3 应用场景

介绍VXLAN的应用场景。

1.3.1 同网段终端用户通信的应用

业务描述

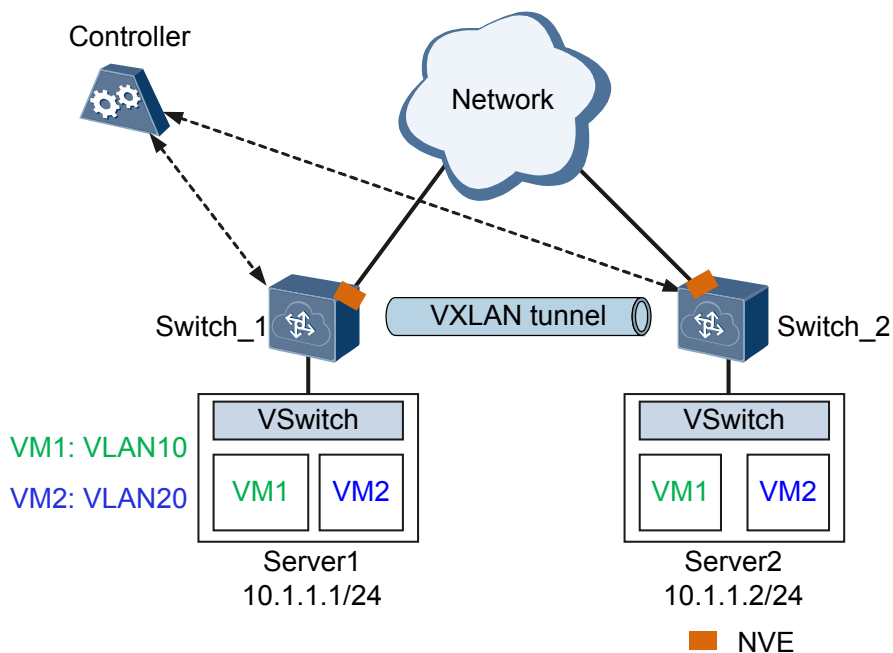
目前，规模化、虚拟化、云计算已成为数据中心的发展方向，同时，数据中心为适应更大的业务量并降低维护成本，逐渐向大二层技术及虚拟化迁移。

随着数据中心在物理网络基础设施上实施服务器虚拟化的快速发展，作为NVO3技术之一的VXLAN技术具有很强的适应性，为数据中心提供了良好的解决方案。

组网描述

如**图1-40**所示，某企业在不同的数据中心的都拥有VM，且位于同一网段。现要实现不同数据中心相同ID的VM的互通。

图 1-40 同网段终端用户通信的应用组网图



特性部署

如图1-40所示，可将交换机设备作为VXLAN二层网关，交换机设备之间建立VXLAN隧道，通过VXLAN二层网关实现同一网段终端用户互通。

1.3.2 不同网段终端用户通信的应用

业务描述

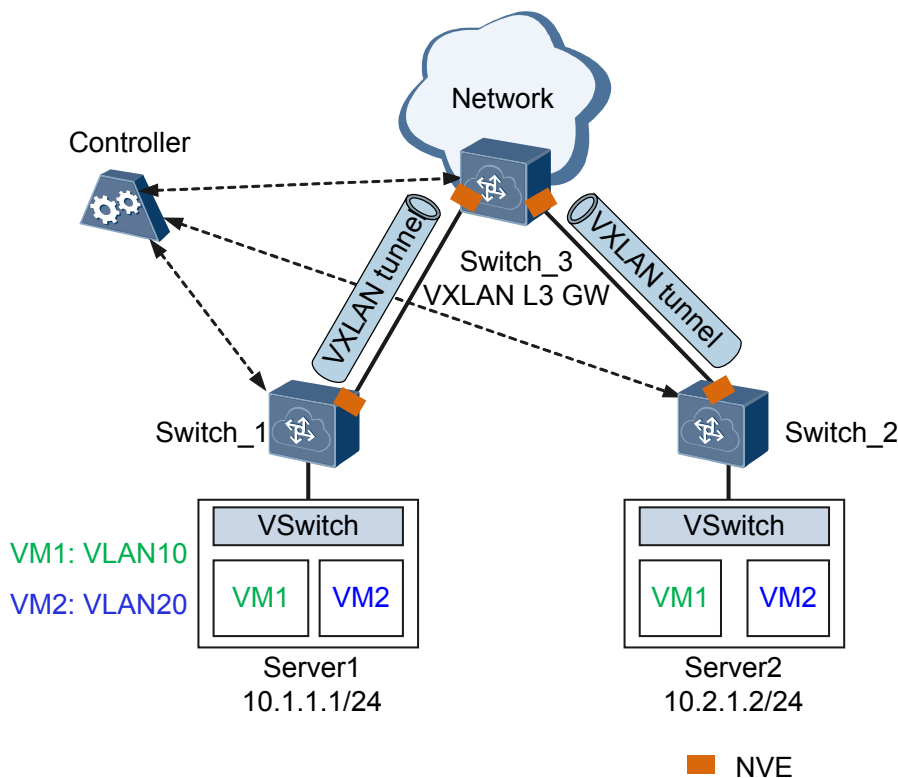
数据中心用来在Internet网络基础设施上加速信息的传递，企业、运营商都在大力建设数据中心。目前，规模化、虚拟化、云计算已成为数据中心的发展方向，同时，数据中心为适应更大的业务量并降低维护成本，逐渐向大二层技术及虚拟化迁移。

随着数据中心在物理网络基础设施上实施服务器虚拟化的快速发展，作为NVO3技术之一的VXLAN技术具有很强的适应性，为数据中心提供了良好的解决方案。

组网描述

如图1-41所示，某企业在不同的数据中都拥有VM，且位于不同网段。现需要实现不同数据中心相同VM的互通。

图 1-41 不同网段终端用户通信的应用组网图



特性部署

如图1-41所示，可将Switch_3设备作为VXLAN三层网关，其他交换机设备作为VXLAN二层网关，交换机设备之间建立VXLAN隧道，通过VXLAN三层网关实现不同网段终端用户互通。

1.3.3 在虚拟机迁移场景中的应用

业务描述

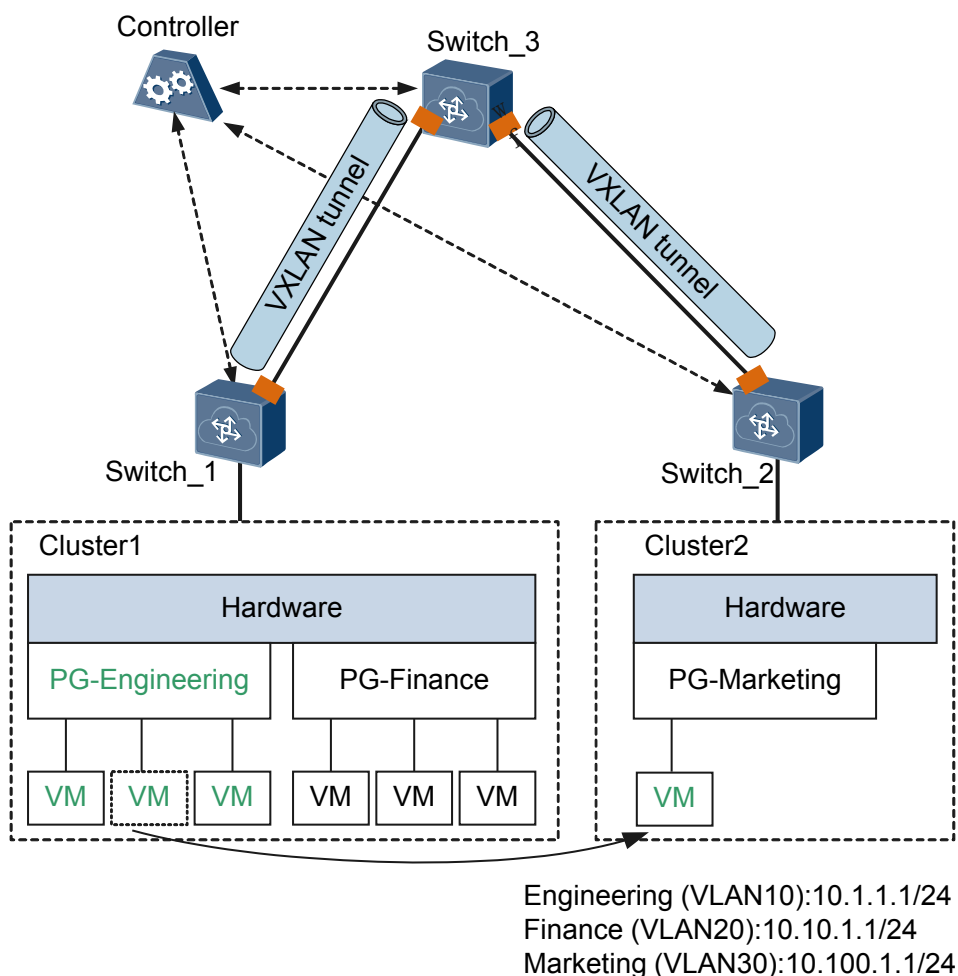
当前数据中心网络中企业通过部署服务器虚拟化来达到整合IT资源、提升资源利用率、降低开支的目的。随着虚拟化水平的不断提高，物理服务器上虚拟机的数量在不断增加，虚拟化环境下运行的应用数量也在不断上升，为虚拟网络带来了很大的挑战。

组网描述

如图1-42所示，某企业在数据中心中有两个群集Cluster，其中工程部门和财务部门都在Cluster1上，营销部门在Cluster2上。

Cluster1上显示计算空间不足，而Cluster2未充分利用。网络管理员需要将工程部门迁移到Cluster2上，而且不影响业务。

图 1-42 企业分布组网图



特性部署

为了保证工程部门迁移过程中业务不中断，则需要保证工程部门的IP地址、MAC地址等参数保持不变，这就要求两个Cluster属于一个二层网络。如果使用传统方法解决此问题，这可能需要网络管理员购买新的物理设备以分离流量，并可能导致诸如VLAN散乱、网络成环以及系统和管理开销等问题。

为了成功将工程部门迁移到Cluster2，可通过VXLAN实现。VXLAN是MAC in UDP的网络虚拟化技术，只要物理网络支持IP转发，所有IP路由可达的终端用户即可建立一个大范围二层网络。

通过VXLAN隧道，工程部门在迁移过程中可保证网络无感知。工程部门从Cluster1迁移到Cluster2后，终端租户会发送免费ARP或RARP报文，所有网关设备上保存的原VM对应的MAC地址表和ARP表都将会被删除，更新为迁移后的VM对应的MAC地址表和ARP表。

1.3.4 VXLAN 分布式网关的应用

业务描述

传统的集中式三层网关将服务器的网关设置在汇聚或者Spine节点，跨网络的报文都必须经过Spine节点转发，若三层网关集中部署，存在如下问题：

- 转发路径不优化：异地数据中心三层流量都需要经过集中三层网关转发。
- ARP表项规格瓶颈：由于采用集中三层网关，通过三层网关转发的终端租户的ARP表项都需要在三层网关上生成，而三层网关上的ARP表项规格有限，这不利于数据中心网络扩展。

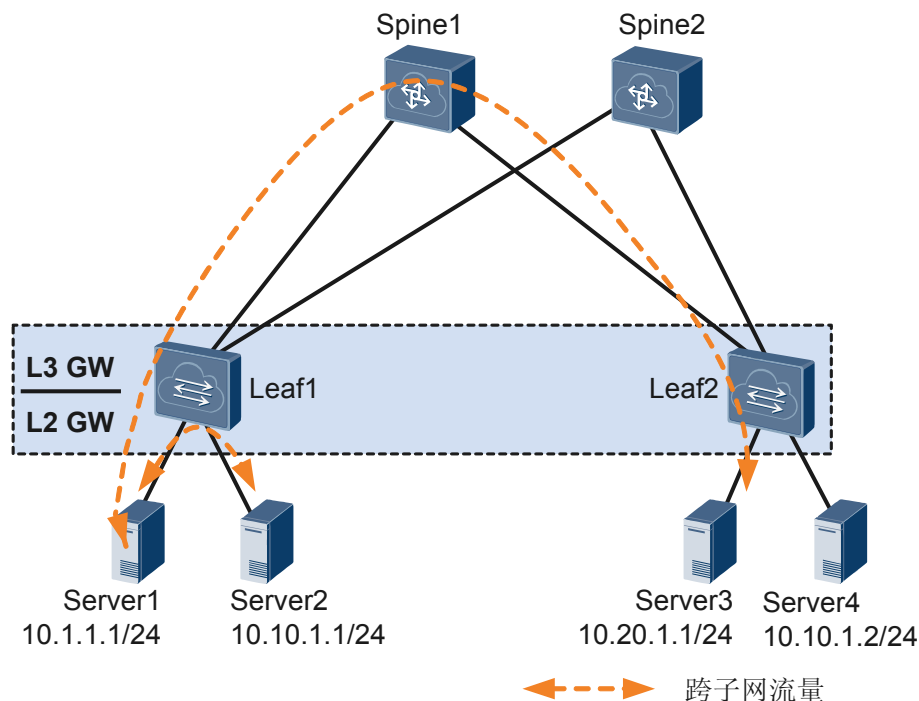
通过配置VXLAN分布式网关可以解决上述问题。VXLAN分布式网关场景下，将Leaf节点作为VXLAN隧道端点VTEP，每个Leaf节点都可作为VXLAN三层网关，Spine节点不感知VXLAN隧道，只作为VXLAN报文的转发节点。

组网描述

如图1-43所示，Server1和Server2不在同一个网段，但是都下挂在Leaf1节点下。在Leaf1上部署VXLAN三层网关，Server1和Server2通信时，流量只需要在Leaf1节点进行转发，不再需要经过Spine节点。

Server1和Server3也不在同一个网段，但是下挂在不同的Leaf节点下。在不同Leaf上部署VXLAN三层网关，Server1和Server3通信时，流量通过VXLAN隧道传输，Spine节点不感知VXLAN隧道，只作为VXLAN报文的转发节点。

图 1-43 VXLAN 分布式网关的应用组网图



特性部署

如图1-43所示，在Leaf节点上同时部署VXLAN二层网关和三层网关：

- 二层网关：用于解决租户接入VXLAN虚拟网络的问题，也可用于同一网段VXLAN虚拟网络的子网通信。
- 三层网关：用于VXLAN虚拟网络的跨子网通信以及外部网络的访问。

部署VXLAN分布式网关时需注意：

- 在VXLAN三层网关上使能ARP本地主机路由发布功能，Leaf节点学习终端租户的ARP表项，再根据ARP表项生成主机路由，并将主机路由通过BGP对外发布，使其他的BGP邻居可以学习到主机路由。
- 如果有相同子网的服务器在不同Leaf节点下，在Leaf节点上配置三层网关，需要配置相同的网关IP地址、MAC地址。当终端租户或服务器移动位置，不需要更改服务器的三层网关配置，减少了维护工作量。
- 当跨Leaf节点进行三层通信时，终端租户必须绑定VPN实例，VXLAN隧道的建立依赖于VPN邻居的建立。

1.3.5 VXLAN 集中式多活网关的应用

业务描述

在VXLAN网络中，为了提高可靠性，用户经常会部署多个网关进行主备备份，以保证一台网关设备故障时流量可以及时切换到另外的网关设备上，避免业务中断。

通过部署VRRP，可以解决上述问题。但由于VRRP组网中，仅主网关设备能够进行流量转发，提供网关服务，备网关设备仅在主网关故障后才提供网关服务，导致网关设备利用率低，网关故障收敛性能低。用户希望在保证可靠性的同时，多个网关都可以同时转发流量，充分利用设备资源。

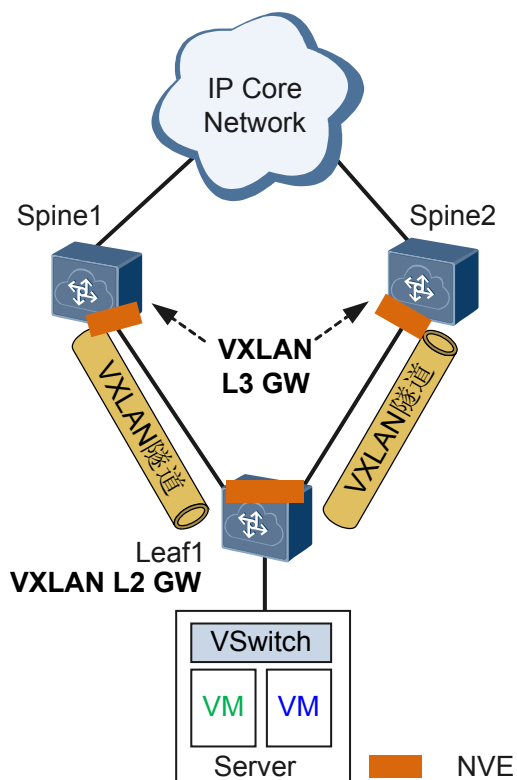
通过VXLAN集中式多活网关可解决上述问题。VXLAN集中式多活网关可保证多台网关设备同时转发流量，实现负载分担，也可实现链路备份，从而提供了更可靠的服务。

组网描述

如图1-44所示，Leaf1双归接入至Spine1和Spine2，现需要通过VXLAN多活网关实现：

- Spine1和Spine2形成负载分担，共同进行流量转发。
- 当一条接入链路或设备发生故障时，流量可以快速切换到另一条链路或设备。

图 1-44 VXLAN 双活网关的应用组网图



特性部署

如图1-44所示，Spine1、Spine2和Leaf1之间建立VXLAN隧道，用于为Leaf1下的VM提供三层网关。

VXLAN多活网关的应用场景必须遵循如下条件：

- Spine1、Spine2上必须部署相同的网关MAC地址、网关IP地址以及源VTEP地址，确保数据中心网络中VM感知的是一台设备，从而实现Spine1和Spine2都能够正常转发数据报文。
- 在Spine1和Spine2之间的链路上部署路由协议，用于解决Leaf1与Spine1或Spine2之间的链路故障情况下，确保数据报文正常转发。

1.3.6 VXLAN 双活接入的应用

业务描述

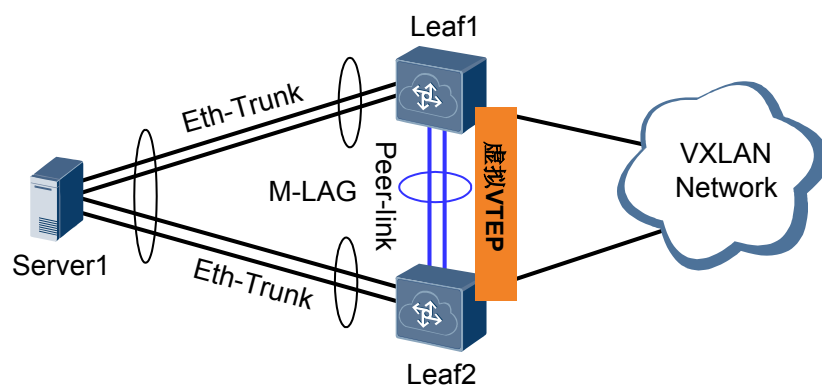
在VXLAN网络中，为了提高可靠性，用户经常采用双归接入的方式将安装有双网卡的服务器接入到VXLAN网络，使得当服务器的一个网卡发生故障时不会导致业务中断。

由于上述方案中，仅主网卡能够进行数据报文收发，备网卡不能进行数据报文收发，导致了网卡和链路带宽资源的浪费。用户希望两个网卡可以同时转发流量，实现双活，从而充分利用网卡和网络带宽资源。

组网描述

如图1-45所示，服务器Server1通过Leaf1和Leaf2双归接入VXLAN网络中。现希望Leaf1和Leaf2能够进行同时进行流量转发，实现流量的负载分担。

图 1-45 VXLAN 双活接入的应用组网图



特性部署

如图1-45所示，在Leaf1和Leaf2上部署VXLAN双活接入可以满足上述需求。

VXLAN双活接入的应用场景必须遵循如下条件：

- Leaf1、Leaf2上必须部署相同的源VTEP地址，确保VXLAN网络中网络侧的其他设备感知到的是一台设备。
- 在Leaf1、Leaf2和服务器Server之间的链路上配置M-LAG，将服务器双归的两台接入设备虚拟成一台设备。

1.4 配置注意事项

介绍部署VXLAN的注意事项。

涉及网元

设备支持通过SNC控制器和单机方式来配置VXLAN，不同的方式涉及的网元存在差异：

表 1-6 支持本特性的最低软件版本

方式	产品	控制器最低支持版本	说明
SNC控制器方式	SNC (Smart Network Controller)	V100R001C30	控制器负责通过OpenFlow协议向转发器下发流表，控制转发器VXLAN隧道的建立以及报文在隧道中的转发。
单机方式	无需其他网元配合。		

License 支持

VXLAN特性是交换机的基本特性，无需获得License许可即可应用此功能。

版本支持

表 1-7 支持本特性的最低软件版本

系列	产品	SNC控制器方式最低支持版本	单机方式最低支持版本
CE12800	CE12804/CE12808/ CE12812/CE12816	V100R003C10	V100R005C00
	CE12804S/ CE12808S	V100R005C00	V100R005C00

特性依赖和限制

在交换机上部署VXLAN功能时，需要注意：

VXLAN的公共约束

- VXLAN的特性限制
 - 目前，VXLAN只支持IPv4网络。
 - SNC控制器和单机方式在使用时互斥，用户只能选择其中一种方式进行部署。
 - 目前，设备在转发BUM（Broadcast&Unknown-unicast&Multicast）报文时，只支持头端复制方式，不支持组播复制方式。
 - 目前，CE设备不支持对VXLAN报文进行分片或重组。当VXLAN网络中同时存在CE和非CE系列设备时，为了避免VXLAN报文在非CE设备上进行了分片，但CE设备无法将其重组造成的转发失败问题，建议在服务器上配置报文的最大帧长不超过1400字节。
 - 在VXLAN隧道出口端设备上，不支持将解封装前的VXLAN报文重定向到端口。
- VXLAN与其他特性间约束
 - 设备不支持VXLAN与以下特性同时配置：SVF、TRILL、FCOE、组播VPN（V100R005C10版本开始支持）。对于V100R005C00版本，取消设备上的VXLAN配置后，可以配置SVF、TRILL或FCOE，但是需要重启设备才能使新配置生效。
 - 设备配置了VXLAN功能后，不能再通过配置单板外扩TCAM来扩大组播表项空间。
 - EA系列单板不支持对VXLAN报文进行镜像和报文捕获。
 - 设备不支持对报文进行VXLAN封装后再进行MPLS封装，也不支持对报文进行MPLS解封装后再进行VXLAN解封装。

采用控制器方式部署VXLAN的特有约束

- 在使用SNC控制器配置VXLAN特性时，转发器不支持配置VPN实例。
- 采用SNC控制器方式部署VXLAN时，引入了双控制面（SNC控制面+转发器本地控制面），双控制面存在转发资源共享。例如：接口的本地VLAN资源、BD ID资

源、VNI资源、IP资源等。为了避免转发资源使用冲突，建议在部署业务前统一规划资源，保证SNC控制面和转发器本地控制面使用不同的编号资源。

- 仅当设备上未划分VS时，才支持采用SNC控制器方式部署VXLAN。
- 对于V100R005C00版本，仅支持通过控制器为转发器下发ARP表项，不支持转发器直接动态学习ARP表项。

采用单机方式部署VXLAN时的特有限制

- 对于V100R003C10和V100R005C00版本，VXLAN特性仅支持在Admin-VS中配置。
- 从V100R005C10版本开始，在所有Port模式VS中，仅Admin-VS支持配置VXLAN功能。
- 从V100R005C10版本开始，所有Group模式的VS都可以配置VXLAN功能。
- 对于V100R005C10版本，Admin-VS与所有Group模式的VS共享BD整机规格。即：假设某VS内已经创建了满规格的BD，则其他VS内将不能再创建BD。
- 如果在设备上同时配置了二层模式和三层模式的NVE接口，则必须为他们配置不同的源端VTEP的IP地址。
- 从V100R005C00版本开始，最多支持同时在4台设备上部署集中式多活网关。
- 配置了VXLAN集中式多活网关后，建议在所有的网关上不要对BDIF接口进行删除或关闭（shutdown）操作。如果必须要删除或关闭某网管的BDIF接口，请保证在操作之前已经通过合理的路由规划将流量引导至其他网关，否则将会导致部分流量丢失。
- 从V100R005C10版本开始，配置分布式网关后，网关收到网络侧的ARP报文将做丢弃处理，只学习用户侧主机的ARP报文。
- 对于V100R005C00版本，二层子接口加入BD后，该BD不支持创建对应的BDIF接口。
- 从V100R005C10版本开始，Default类型的二层子接口加入BD后，该BD不支持创建对应的BDIF接口。

1.5 配置 VXLAN（SNC 控制器方式）

介绍了SNC控制器配合设备实现VXLAN部署的方法。

前提条件

- 在配置通过SNC控制器来部署VXLAN网络时，需要完成以下任务：
 - 控制器和转发器之间路由可达。
 - 控制器和转发器之间的Openflow通信通道已成功建立，具体配置请参见**OpenFlow Agent配置**。
 - 转发器上已通过命令**ip tunnel mode vxlan**配置隧道模式为VXLAN。

操作步骤

步骤1 当SNC控制器和转发器之间建立Openflow通信通道后，设备作为转发器，无需进行VXLAN配置，VXLAN隧道的创建以及指导报文转发的ARP或MAC表项，均在SNC控制器上进行配置，具体请参见《SNC V100R001C30产品文档 解决方案》中的“VXLAN解决方案”。

---结束

1.6 配置 VXLAN（单机方式）

介绍了不依赖于任何控制器，直接在设备上配置VXLAN的方法。

说明

在V100R005C00版本，采用单机方式部署VXLAN网络时，设备仅支持[配置同网段用户通过VXLAN隧道互通](#)。

1.6.1 配置同网段用户通过 VXLAN 隧道互通

同网段用户通过VXLAN隧道互通，也可理解为通过VXLAN二层网关实现同网段用户互通，VXLAN二层网关也可解决租户接入VXLAN虚拟网络的问题。

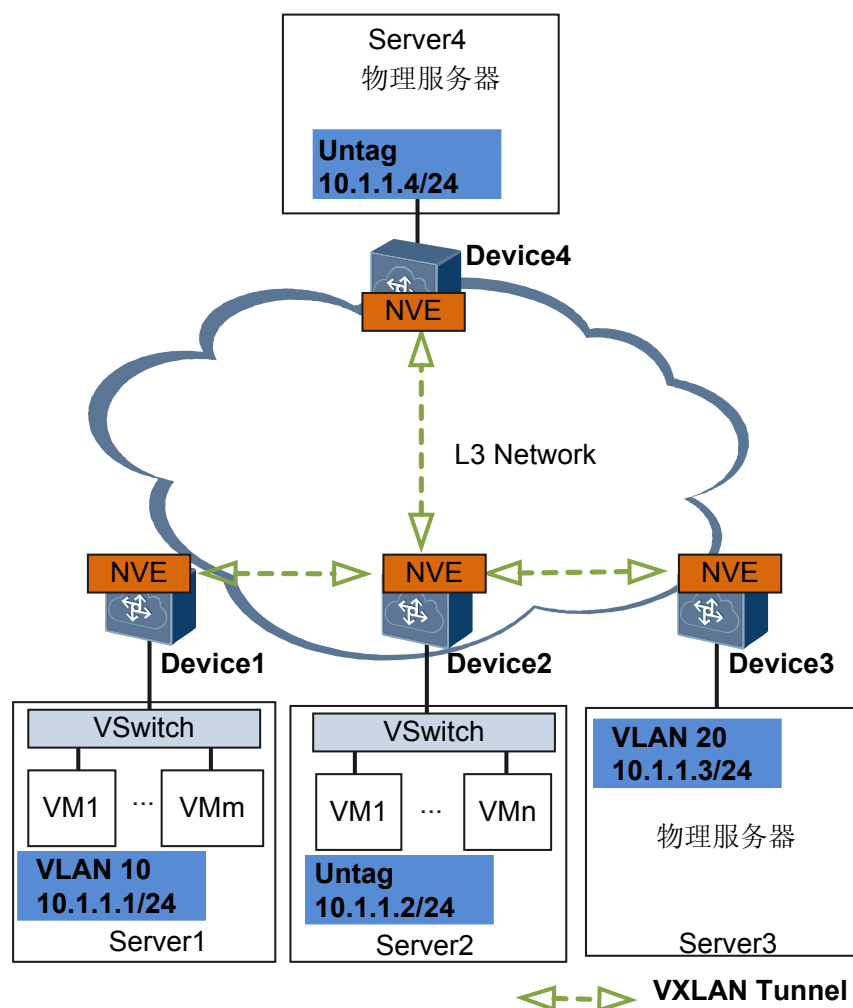
应用环境

如果企业为某租户分配有位于不同地理位置的物理服务器和VM，且这些物理服务器和VM都处于同一网段，当需要实现不同VM之间或VM和物理服务器之间的通信时，需要合理规划VXLAN二层网关，建立VXLAN隧道。

如图1-46所示：

- 当Server2中的VM1需要与Server1中的VM1通信时，需要在Device1、Device2上部署VXLAN二层网关，Device1和Device2之间建立VXLAN隧道实现同网段租户互通。
- 当Server2中的VM1需要与Server3或Server4通信时，需要在Device2、Device3和Device4上部署VXLAN二层网关，使Device2和Device3之间、Device2和Device4之间能够建立VXLAN隧道实现同网段租户互通。

图 1-46 配置同网段用户通过 VXLAN 隧道互通典型场景



由于所有的NVE都部署在支持NVE的设备上，所有的VXLAN报文封装与解封装都在设备上完成。因此，需要在所有部署了NVE的设备（Device1、Device2、Device3和Device4）上执行本任务。

前置任务

在配置同网段用户通过VXLAN隧道互通之前，需完成以下任务：

- 网络三层路由可达。
- 设备上已通过命令 `ip tunnel mode vxlan` 配置隧道模式为 VXLAN。

配置流程

以下任务均为必选配置，请按照顺序配置。

1.6.1.1 配置业务接入点实现区分业务流量

背景信息

在VXLAN网络中，业务接入点统一表现为二层子接口，通过在二层子接口上配置流封装实现不同的接口接入不同的数据报文。广播域统一表现为BD（Bridge-Domain），将二层子接口关联BD后，可实现数据报文通过BD转发。

如表1-8所示，可为二层子接口配置不同的流封装类型以实现不同的接口接入不同的数据报文。

表 1-8 流封装类型

流封装类型	说明
dot1q	<p>允许接口接收Tagged报文。</p> <p>配置二层子接口的流封装类型为dot1q时：</p> <ul style="list-style-type: none"> ● 二层子接口封装的vid，不能与对应二层主接口允许通过的VLAN相同，不能与MUX VLAN中的VLAN相同，也不能与VLAN Mapping和VLAN Stacking的源VLAN相同。 ● 二层子接口和三层子接口封装的VLAN ID不能相同。
untag	<p>允许接口接收Untagged报文。</p> <p>配置流封装类型为untag时，请确保该子接口对应的物理接口上仅有缺省配置。</p> <p>仅支持为二层物理接口（包括Eth-Trunk接口）创建untag类型二层子接口。</p>
default	<p>允许接口接收所有报文，不区分报文中是否带VLAN Tag。</p> <p>配置二层子接口的流封装类型为default时：</p> <ul style="list-style-type: none"> ● 必须确保对应的主接口没有加入任何VLAN。 ● 主接口下创建了default类型二层子接口，不允许再创建其他二层子接口。

操作步骤

步骤1 执行命令**system-view**，进入系统视图。

步骤2 执行命令**bridge-domain bd-id**，创建BD，并进入BD视图。

缺省情况下，没有创建广播域BD。

步骤3（可选）执行命令**description description**，配置BD的描述信息。

缺省情况下，没有配置BD的描述信息。

VXLAN网络中若配置了大量BD，为了方便记忆和管理这些BD，可以对不同的BD执行**description**命令，用来标识BD的某些特征。例如：转发的业务类型等。

步骤4 执行命令**quit**，返回系统视图。

步骤5 执行命令 `interface interface-type interface-number.subnum mode l2`，进入指定二层以太网子接口的视图。

缺省情况下，没有创建二层子接口。

*subnum*是以太网子接口的编号。

执行本命令前，请确保对应的二层主接口上没有 `port link-type dot1q-tunnel`配置。

步骤6 执行命令 `encapsulation { dot1q vid vid | default | untag }`，配置流封装类型实现不同的接口接入不同的数据报文。

缺省情况下，没有配置流封装类型。

步骤7 执行命令 `bridge-domain bd-id`，配置将二层子接口加入BD。

缺省情况下，二层子接口没有加入BD。

说明

对于V100R005C00版本，二层子接口加入BD后，该BD不支持创建对应的BDIF接口。

从V100R005C10版本起，Default类型的二层子接口加入BD后，该BD不支持创建对应的BDIF接口。

步骤8 执行命令 `commit`，提交配置。

---结束

1.6.1.2 配置 VXLAN 隧道转发业务流量

背景信息

VXLAN通过采用MAC in UDP封装来延伸二层网络，是大二层虚拟网络扩展的隧道封装技术。

通过VXLAN，虚拟网络可接入大量租户，且租户可以规划自己的虚拟网络，不需要考虑物理网络IP地址和广播域的限制，降低了网络管理的难度。

操作步骤

步骤1 执行命令 `system-view`，进入系统视图。

步骤2 执行命令 `bridge-domain bd-id`，创建BD，并进入BD视图。

缺省情况下，没有创建广播域BD。

该步骤中的 *bd-id*必需与 [配置业务接入点实现区分业务流量](#) 步骤2中创建的 *bd-id*一致。

步骤3 执行命令 `vlan vni vni-id`，创建VXLAN网络标识VNI并关联广播域BD。

缺省情况下，没有创建VNI。

步骤4 执行命令 `quit`，返回系统视图。

步骤5 执行命令 `interface nve nve-number`，创建NVE接口，并进入NVE接口视图。

缺省情况下，未创建NVE接口。

步骤6 执行命令 `source ip-address`，配置源端VTEP的IP地址。

缺省情况下，源端VTEP没有配置IP地址。推荐使用Loopback接口的IP地址。

步骤7 执行命令 `vni vni-id head-end peer-list ip-address &<1-10>`，配置VNI的头端复制列表。

缺省情况下，没有配置VNI头端复制列表。

步骤8 执行命令 `commit`，提交配置。

----结束

1.6.1.3（可选）配置提升 VXLAN 网络安全性

背景信息

此功能从V100R005C10版本开始支持。

当VXLAN隧道入口收到BUM（Broadcast&Unknown-unicast&Multicast）报文时，本地VTEP会将收到的报文根据VTEP列表进行复制并发送给属于同一个VNI的所有VTEP。为了减少网络中的广播流量，提高网络安全性，可配置静态MAC地址表指定转发路径，也可防止仿冒身份的非法用户骗取数据。

操作步骤

步骤1 执行命令 `system-view`，进入系统视图。

步骤2 执行命令 `mac-address static mac-address bridge-domain bd-id source source-ip-address peer peer-ip vni vni-id`，配置静态MAC地址表项。

缺省情况下，没有配置静态MAC地址表项。

步骤3 执行命令 `commit`，提交配置。

----结束

1.6.1.4 检查配置结果

背景信息

VXLAN配置完成后，请执行下面的命令检查配置结果。

操作步骤

- 执行命令 `display vxlan tunnel [tunnel-id] [verbose]`，查看VXLAN隧道的信息。
- 执行命令 `display vxlan vni [vni-id [verbose]]`，查看VXLAN的配置信息。

----结束

1.6.2 配置不同网段用户通过 VXLAN 三层网关通信

VXLAN三层网关可实现不同网段的VXLAN间通信，也可实现VXLAN和非VXLAN的通信。

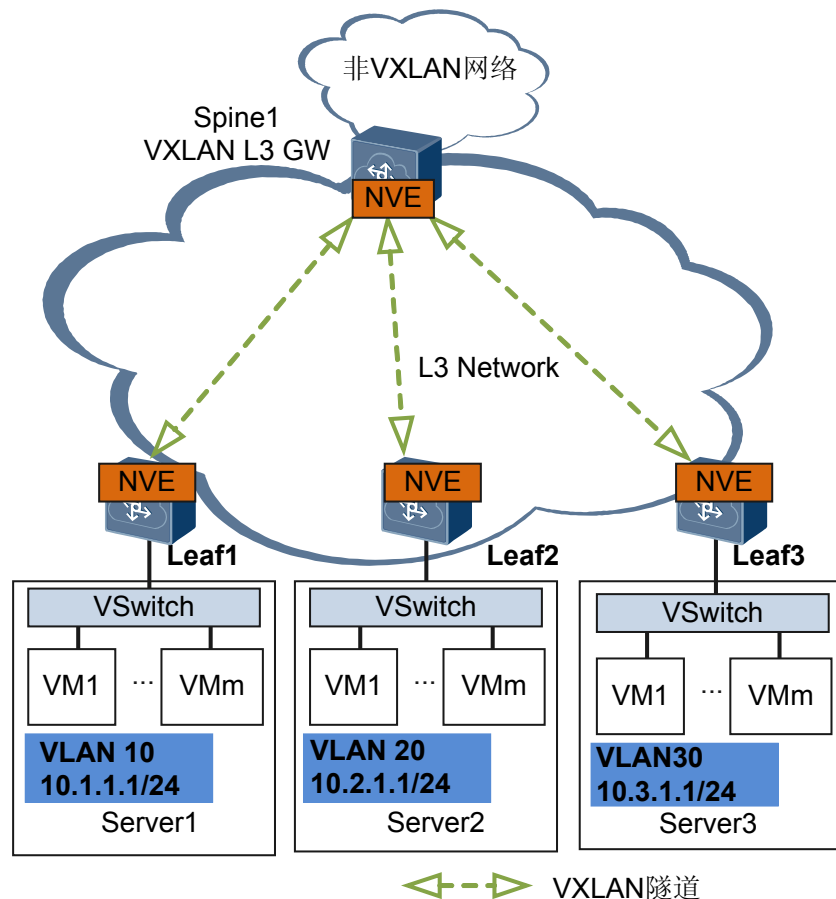
本特性从V100R005C10版本开始支持。

应用环境

如果企业为某租户分配有位于不同地理位置的VM，且这些VM处于不同网段，当不同VM之间，或该VXLAN网络需要与外部非VXLAN网络进行通信时，需要通过本任务配置VXLAN三层网关。

如图1-47所示，当Server1上的VM1需要与非VXLAN网络或是Server2、Server3上的VM1通信时，需要在Device4上部署VXLAN三层网关功能。

图 1-47 配置不同网段用户通过 VXLAN 三层网关通信典型场景



前置任务

在配置不同网段用户通过VXLAN三层网关通信之前，需完成以下任务：

- 网络三层路由可达。
- 设备上已通过命令 `ip tunnel mode vxlan` 配置隧道模式为 VXLAN。

配置流程

请按照以下顺序进行配置不同网段用户通过VXLAN三层网关通信功能。对于可选步骤，请根据情况选择配置。

1.6.2.1 配置业务接入点实现区分业务流量

背景信息

在VXLAN网络中，业务接入点统一表现为二层子接口，通过在二层子接口上配置流封装实现不同的接口接入不同的数据报文。广播域统一表现为BD（Bridge-Domain），将二层子接口关联BD后，可实现数据报文通过BD转发。

如表1-9所示，可为二层子接口配置不同的流封装类型以实现不同的接口接入不同的数据报文。

表 1-9 流封装类型

流封装类型	说明
dot1q	允许接口接收Tagged报文。 配置二层子接口的流封装类型为 dot1q 时： <ul style="list-style-type: none"> ● 二层子接口封装的vid，不能与对应二层主接口允许通过的VLAN相同，也不能与MUX VLAN中的VLAN相同。 ● 二层子接口和三层子接口封装的VLAN ID不能相同。
untag	允许接口接收Untagged报文。 配置流封装类型为 untag 时，请确保该子接口对应的物理接口上仅有缺省配置。 仅支持为二层物理接口（包括Eth-Trunk接口）创建 untag 类型二层子接口。
default	允许接口接收所有报文，不区分报文中是否带VLAN Tag。 配置二层子接口的流封装类型为 default 时： <ul style="list-style-type: none"> ● 必须确保对应的主接口没有加入任何VLAN。 ● 主接口下创建了default类型二层子接口，不允许再创建其他二层子接口。

请在Leaf设备上进行如下配置。

操作步骤

步骤1 执行命令**system-view**，进入系统视图。

步骤2 执行命令**bridge-domain bd-id**，创建BD，并进入BD视图。

缺省情况下，没有创建广播域BD。

步骤3（可选）执行命令**description description**，配置BD的描述信息。

缺省情况下，没有配置BD的描述信息。

VXLAN网络中若配置了大量BD，为了方便记忆和管理这些BD，可以对不同的BD执行**description**命令，用来标识BD的某些特征。例如：转发的业务类型等。

步骤4 (可选) 执行命令 **arp broadcast-suppress enable**, 使能ARP广播报文抑制功能。(此步骤从V100R005C10版本开始支持)

缺省情况下, ARP广播报文抑制功能处于去使能。

 **说明**

在二层网关上使能ARP广播报文抑制功能, 二层网关节点在收到ARP广播报文时, 将其转换为单播报文进行转发, 减少BD域内的广播报文, 可提高网络性能。

VXLAN二层网关学习到的主机信息是通过EVN BGP发布的。因此, 为保证ARP广播报文抑制功能生效, 必须配置EVN BGP。EVN BGP配置如下:

1. 在系统视图下执行命令 **evn bgp**, 进入EVN BGP视图。
2. 执行命令 **source-address ip-address**, 配置建立EVN BGP对等体关系的源地址, 该地址可以用于生成Router-ID、路由下一跳地址以及EVN实例的RD等。
3. 执行命令 **peer ip-address**, 指定EVN BGP对等体的IP地址。
4. 执行命令 **commit**, 提交配置。

步骤5 执行命令 **quit**, 返回系统视图。

步骤6 执行命令 **interface interface-type interface-number.subnum mode l2**, 进入指定二层以太网子接口的视图。

缺省情况下, 没有创建二层子接口。

*subnum*是以太网子接口的编号。

执行本命令前, 请确保对应的二层主接口上没有 **port link-type dot1q-tunnel**配置。

步骤7 执行命令 **encapsulation { dot1q vid vid | default | untag }**, 配置流封装类型实现不同的接口接入不同的数据报文。

缺省情况下, 没有配置流封装类型。

步骤8 执行命令 **bridge-domain bd-id**, 配置将二层子接口加入BD。

缺省情况下, 二层子接口没有加入BD。

 **说明**

对于V100R005C00版本, 二层子接口加入BD后, 该BD不支持创建对应的BDIF接口。

从V100R005C10版本起, Default类型的二层子接口加入BD后, 该BD不支持创建对应的BDIF接口。

步骤9 执行命令 **commit**, 提交配置。

----结束

1.6.2.2 配置 VXLAN 隧道转发业务流量

背景信息

VXLAN通过采用MAC in UDP封装来延伸二层网络, 是大二层虚拟网络扩展的隧道封装技术。

通过VXLAN, 虚拟网络可接入大量租户, 且租户可以规划自己的虚拟网络, 不需要考虑物理网络IP地址和广播域的限制, 降低了网络管理的难度。

请在Spine和Leaf设备上进行如下配置。

操作步骤

- 步骤1** 执行命令`system-view`，进入系统视图。
- 步骤2** 执行命令`bridge-domain bd-id`，创建BD，并进入BD视图。
缺省情况下，没有创建广播域BD。
该步骤中的`bd-id`必需与[配置业务接入点实现区分业务流量](#)步骤2中创建的`bd-id`一致。
- 步骤3** 执行命令`vlan vni vni-id`，创建VXLAN网络标识VNI并关联广播域BD。
缺省情况下，没有创建VNI。
- 步骤4** 执行命令`quit`，返回系统视图。
- 步骤5** 执行命令`interface nve nve-number`，创建NVE接口，并进入NVE接口视图。
缺省情况下，未创建NVE接口。
- 步骤6** 执行命令`source ip-address`，配置源端VTEP的IP地址。
缺省情况下，源端VTEP没有配置IP地址。推荐使用Loopback接口的IP地址。
- 步骤7** 执行命令`vni vni-id head-end peer-list ip-address &<1-10>`，配置VNI的头端复制列表。
缺省情况下，没有配置VNI头端复制列表。
- 步骤8** 执行命令`commit`，提交配置。
---结束

1.6.2.3 配置三层网关实现不同网段业务流量互通

背景信息

不同网段的VXLAN间，及VXLAN和非VXLAN之间不能直接相互通信。为了能够实现通信，需要通过IP路由实现。

BD是VXLAN网络的实体，通过将VNI（每一个VNI表示一个租户）以1:1方式映射到广播域BD，可以通过BD转发数据报文。基于BD可创建三层逻辑接口BDIF接口，可以实现不同网段的VXLAN间，及VXLAN和非VXLAN之间的三层互通，也可实现二层网络接入三层网络。每个BD对应一个BDIF接口，在为BDIF接口配置IP地址后，该接口即可作为本BD内租户的网关，对需要进行通信的报文进行基于IP地址的三层转发。

请在Spine设备上进行如下配置。

说明

设备支持在BDIF接口下配置DHCP中继功能，使主机从外部的DHCP服务器上申请到IP地址等相关信息。

操作步骤

- 步骤1** 执行命令`system-view`，进入系统视图。

步骤2 执行命令 `interface vbdif bd-id`，创建BDIF接口，并进入BDIF接口视图。

BDIF接口的编号必须对应一个已经创建的BD ID。

步骤3 执行命令 `ip address ip-address { mask | mask-length } [sub]`，配置BDIF接口的IP地址，实现三层互通。

步骤4（可选）执行命令 `mac-address mac-address`，配置BDIF接口的MAC地址。

缺省情况下，BDIF接口的MAC地址是系统的MAC地址。

说明

目前，对于EA系列单板，设备仅支持MAC地址取值为0000-5e00-0101 ~ 0000-5e00-0107；对于非EA系列单板，设备仅支持MAC地址取值为0000-5e00-0101 ~ 0000-5e00-01ff。

步骤5 执行命令 `commit`，提交配置。

----结束

1.6.2.4（可选）配置 VXLAN 集中式多活网关

背景信息

传统网络中，一般采用VRRP来实现网关的保护。一台主网关工作，其他网关处于备用状态，导致网关设备利用率较低；在网关故障后，需要重新计算和选举出新的网关，导致网关故障收敛性能较低。

通过配置VXLAN多活网关，可以将多个VXLAN三层网关虚拟成一个VXLAN三层网关，实现任意网关都能正确转发流量，从而提高设备利用率和故障收敛性能。

请在Spine设备上进行如下配置。

说明

部署VXLAN集中式多活网关后，当管理员由于升级或补丁等原因需要复位设备时，建议：

- 复位前，对于接入侧，为了让上行流量不再转发到该设备，需要拆除接入设备到该设备VTEP地址的ECMP路径，将该设备对外发布的VTEP主机路由优先级调低；对于网络侧，为了避免下行流量转发到该设备，需要降低设备上发布的VXLAN网关的网段路由优先级。
- 复位后，为了避免由于网关设备上缺少ARP表项导致流量丢失，建议在ARP表项同步完成后（以128K ARP表项为例，约20分钟），再重新将VTEP主机路由和VXLAN网关的网段路由由优先级调回原值。

当管理员Shutdown或删除多活网关设备上的BDIF接口时，流量无法切换到其他网关。因为接入侧设备会依据VTEP的ECMP路由将流量继续转发到该设备，但设备无法将接收到的流量继续转发，因此无法将流量平滑迁移到其他多活网关设备上，会导致流量丢失。

设备异常复位重新恢复时，为了避免该设备上ARP表项尚未同步完成即开始转发流量，建议配置路由的延时发布机制。以OSPF为例，请在存在VTEP和BDIF相关路由的OSPF进程中通过命令 `stub-router on-startup [interval]` 配置设备在重启或故障时保持为Stub服务器的时间间隔（以128K ARP表项为例，约20分钟）。

操作步骤

步骤1 执行命令 `system-view`，进入系统视图。

步骤2 请在需要配置多活网关功能的设备上分别配置VXLAN三层网关功能，具体请参见[配置VXLAN隧道转发业务流量](#)和[配置三层网关实现不同网段业务流量互通](#)。需要注意的是：

操作步骤	命令	说明
配置VTEP的源IP地址	source ip-address	请确保多台网关设备的VTEP源IP地址相同。
配置VNI的头端复制列表	vni vni-id head-end peer-list ip-address <1-10>	请确保多台网关设备的头端复制列表相同。
配置BDIF接口的IP地址	ip address ip-address { mask mask-length } [sub]	请确保多台网关设备的BDIF接口的IP地址相同。
配置BDIF接口的MAC地址	mac-address mac-address	请确保多台网关设备的BDIF接口的MAC地址相同。

步骤3 请在需要配置多活网关功能的设备上分别配置多活网关同步组，实现多活网关之间的ARP表项同步。

1. 执行命令**dfs-group dfs-group-id**，创建DFS Group并进入DFS-Group视图。
缺省情况下，系统没有创建DFS Group。
2. 执行命令**source ip ip-address**，配置DFS Group绑定的IPv4地址。
缺省情况下，DFS Group没有绑定IPv4地址。
3. （可选）执行命令**udp port port-number**，配置DFS Group的UDP端口号。
缺省情况下，DFS Group的UDP端口号为61467。
4. 执行命令**active-active-gateway**，创建多活网关并进入多活网关视图。
缺省情况下，未创建多活网关。
5. 执行命令**peer ip-address [vpn-instance vpn-instance-name]**，配置多活网关邻居。
缺省情况下，未配置多活网关邻居。

步骤4 执行命令**commit**，提交配置。

----结束

1.6.2.5 （可选）配置提升 VXLAN 网络安全性

背景信息

终端用户在不同网段场景下，可部署如下特性提高VXLAN网络安全性：

- 当VXLAN隧道入口收到BUM（Broadcast&Unknown-unicast&Multicast）报文时，本地VTEP会将收到的报文根据VTEP列表进行复制并发送给属于同一个VNI的所有VTEP。为了减少网络中的广播流量，提高网络安全性，可配置静态MAC地址表指定转发路径，也可防止仿冒身份的非法用户骗取数据。
- 静态ARP表项通过手工配置和维护，不会被老化，不会被动态ARP表项覆盖。所以配置静态ARP表项可以增加通信的安全性。静态ARP表项可以限制和指定IP地址的设备通信时只使用指定的MAC地址，此时攻击报文无法修改此表项的IP地址和MAC地址的映射关系，从而保护了本设备和指定设备间的正常通信。

在VXLAN三层网关场景下，可在接入用户的设备上配置静态MAC地址、在VXLAN三层网关上配置静态ARP表项实现提升VXLAN网络的安全性。

请在Spine设备和Leaf设备上进行如下配置。

操作步骤

步骤1 执行命令**system-view**，进入系统视图。

步骤2 执行命令**mac-address static mac-address bridge-domain bd-id source source-ip-address peer peer-ip vni vni-id**，配置静态MAC地址表项。

缺省情况下，没有配置静态MAC地址表项。

步骤3 执行命令**arp static ip-address mac-address vni vni-id source-ip source-ip peer-ip peer-ip**，配置ARP静态表项。

缺省情况下，没有配置静态ARP地址表项。



说明
*ip-address*的取值必须和三层网关的地址在同一个网段。

步骤4 执行命令**commit**，提交配置。

---结束

1.6.2.6 检查配置结果

前提条件

已经完成配置不同网段用户通过VXLAN三层网关通信的所有配置。

操作步骤

- 执行命令**display vxlan tunnel [tunnel-id] [verbose]**命令，查看VXLAN隧道的信息。
- 执行命令**display vxlan vni [vni-id [verbose]]**命令，查看VXLAN的配置信息及VNI状态。
- 执行命令**display interface vbdif [bd-id]**，查看BDIF接口的状态信息、配置信息和统计信息。
- 执行命令**display dfs-group dfs-group-id active-active-gateway**，查看DFS Group多活网关的信息。

---结束

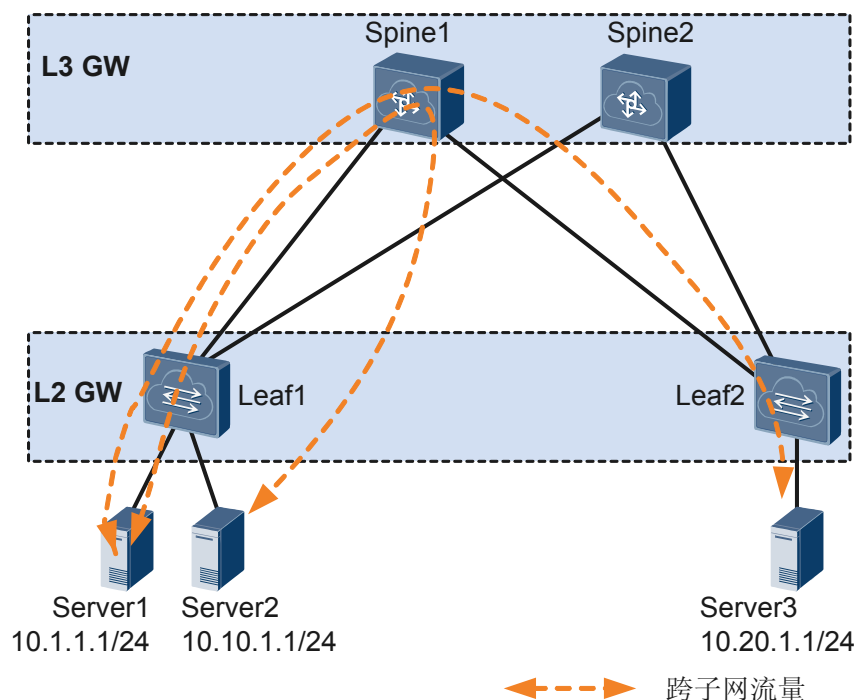
1.6.3 配置 VXLAN 分布式网关

VXLAN分布式网关可解决VXLAN集中式网关的转发路径不优化、三层网关ARP表项规格瓶颈问题。

本特性从V100R005C10版本开始支持。

应用环境

图 1-48 VXLAN 集中式网关示意图

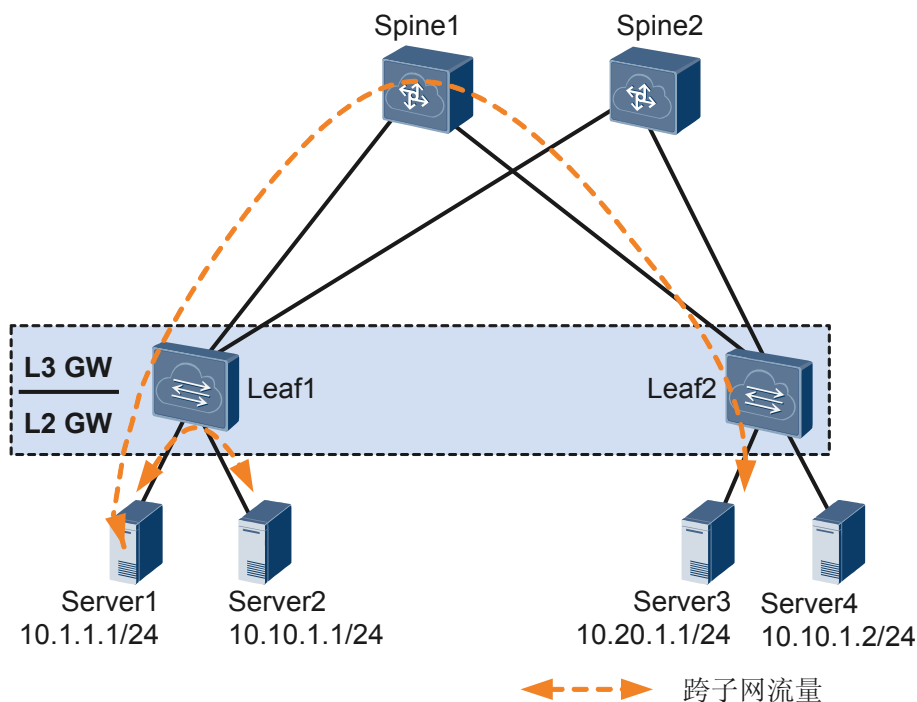


传统的集中式三层网关将服务器的网关设置在汇聚或者Spine节点，如图1-48所示，跨网络的报文都必须经过Spine节点转发，若三层网关集中部署，存在如下问题：

- 转发路径不是最优：异地数据中心三层流量都需要经过集中三层网关转发。
- ARP表项规格瓶颈：由于采用集中三层网关，通过三层网关转发的终端租户的ARP表项都需要在三层网关上生成，而三层网关上的ARP表项规格有限，这不利于数据中心网络扩展。

通过配置VXLAN分布式网关可以解决上述问题。如图1-49所示，Server1和Server2不在同一个网段，但是都连接在Leaf1节点下。Server1和Server2通信时，流量只需要在Leaf1节点进行转发，不再需要经过Spine节点。

图 1-49 VXLAN 分布式网关示意图



前置任务

在配置VXLAN分布式网关之前，需完成以下任务：

- 网络三层路由可达。
- 设备上已通过命令 `ip tunnel mode vxlan` 配置隧道模式为 VXLAN。
- VPN邻居已经成功建立。

配置流程

以下任务均为必选配置，请按照顺序配置。

1.6.3.1 配置 VXLAN 二层网关

背景信息

在VXLAN分布式网关场景下，二层网关主要用于解决租户接入VXLAN虚拟网络的问题，它通过在二层子接口上配置流封装实现不同的接口接入不同的数据报文。广播域统一表现为BD，将二层子接口关联BD后，可实现数据报文通过BD转发。

如表1-10所示，可为二层子接口配置不同的流封装类型以实现不同的接口接入不同的数据报文。

表 1-10 流封装类型

流封装类型	说明
dot1q	<p>允许接口接收Tagged报文。</p> <p>配置二层子接口的流封装类型为dot1q时：</p> <ul style="list-style-type: none"> ● 二层子接口封装的vid，不能与对应二层主接口允许通过的VLAN相同，不能与MUX VLAN中的VLAN相同，也不能与VLAN Mapping和VLAN Stacking的源VLAN相同。 ● 二层子接口和三层子接口封装的VLAN ID不能相同。
untag	<p>允许接口接收Untagged报文。</p> <p>配置流封装类型为untag时，请确保该子接口对应的物理接口上仅有缺省配置。</p> <p>仅支持为二层物理接口（包括Eth-Trunk接口）创建untag类型二层子接口。</p>
default	<p>允许接口接收所有报文，不区分报文中是否带VLAN Tag。</p> <p>配置二层子接口的流封装类型为default时：</p> <ul style="list-style-type: none"> ● 必须确保对应的主接口没有加入任何VLAN。 ● 主接口下创建了default类型二层子接口，不允许再创建其他二层子接口。

操作步骤

步骤1 执行命令**system-view**，进入系统视图。

步骤2 执行命令**bridge-domain bd-id**，创建广播域BD，并进入BD视图。

缺省情况下，没有创建BD。

步骤3（可选）执行命令**description description**，配置BD的描述信息。

缺省情况下，没有配置BD的描述信息。

VXLAN网络中若配置了大量BD，为了方便记忆和管理这些BD，可以对不同的BD执行**description**命令，用来标识BD的某些特征。例如：转发的业务类型等。

步骤4（可选）执行命令**arp broadcast-suppress enable**，使能ARP广播报文抑制功能。（此步骤从V100R005C10版本开始支持）

缺省情况下，ARP广播报文抑制功能处于去使能。



说明

在二层网关上使能ARP广播报文抑制功能，二层网关节点在收到ARP广播报文时，将其转换为单播报文进行转发，减少BD域内的广播报文，可提高网络性能。

VXLAN二层网关学习到的主机信息是通过EVPN BGP发布的。因此，为保证ARP广播报文抑制功能生效，必须配置EVPN BGP。EVPN BGP配置如下：

1. 在系统视图下执行命令`evn bgp`，进入EVPN BGP视图。
2. 执行命令`source-address ip-address`，配置建立EVPN BGP对等体关系的源地址，该地址可以用于生成Router-ID、路由下一跳地址以及EVPN实例的RD等。
3. 执行命令`peer ip-address`，指定EVPN BGP对等体的IP地址。
4. 执行命令`commit`，提交配置。

步骤5 执行命令`quit`，退出BD视图，返回到系统视图。

步骤6 执行命令`interface interface-type interface-number.subnum mode l2`，创建二层子接口，并进入二层子接口视图。

缺省情况下，没有创建二层子接口。

执行本命令前，请确保对应的二层主接口上没有`port link-type dot1q-tunnel`配置。

步骤7 执行命令`encapsulation { dot1q vid vid | default | untag }`，配置流封装类型实现不同的接口接入不同的数据报文。

缺省情况下，没有配置流封装类型。

步骤8 执行命令`bridge-domain bd-id`，将二层子接口加入BD，允许报文通过广播域BD转。

步骤9 执行命令`commit`，提交配置。

---结束

1.6.3.2 配置 VXLAN 三层网关

背景信息

对于租户三层隔离，通常使用VPN技术进行三层路由隔离。在VXLAN分布式网关场景下，当跨三层网关进行三层通信时，三层网关必须绑定VPN实例，VXLAN隧道的建立依赖于VPN隧道的建立。三层网关进行VXLAN报文封装/解封装，实现跨子网的终端租户通信，以及外部网络的访问。

在三层网关上，除了给每个子网分配VNI，还会给每个租户（VPN实例）分配一个三层VNI。当跨三层网关进行三层转发时，VPN实例的VNI ID通过VXLAN隧道传输到远端三层网关，远端三层网关通过租户VNI ID来识别VPN，这样即可识别租户是否属于同一个VPN，以及是否需要互通或隔离。

VXLAN分布式网关场景下跨子网互通必须通过三层转发，这就要求三层网关间必须互相学习到主机路由。为了实现跨子网互通，VXLAN三层网关上必须部署表1-11所示的功能。

表 1-11 VXLAN 三层网关功能

功能	描述
发布主机路由	作为VXLAN三层网关需要学习终端租户的ARP表项，再根据ARP表项生成主机路由，并将主机路由通过BGP对外发布，使其他的BGP邻居可以学习到主机路由。
remote-nexthop属性发布功能	VXLAN三层网关之间的VXLAN隧道通过BGP动态管理，BGP通过remote-nexthop路径属性发布主机路由给其他BGP邻居。

 说明

如果有相同子网的终端租户在不同VXLAN三层网关下，需要为三层网关配置相同的网关IP地址、MAC地址。当终端租户移动位置，不需要更改终端租户的三层网关配置，减少了维护工作量。

操作步骤

步骤1 使能BGP承载VXLAN隧道功能并为终端租户分配VNI。

1. 执行命令**system-view**，进入系统视图。
2. 执行命令**host collect protocol bgp**，使能BGP承载VXLAN隧道功能进行主机信息搜集。
缺省情况下，主机信息搜集功能处于去使能状态。
3. 执行命令**bridge-domain bd-id**，进入BD视图。
4. 执行命令**vxlan vni vni-id**，为终端租户创建VNI并关联广播域BD。
缺省情况下，没有创建VNI。
5. 执行命令**quit**，退出BD视图，返回到系统视图。

步骤2 为VPN实例分配VNI。

1. 执行命令**ip vpn-instance vpn-instance-name**，进入VPN实例视图。
2. 执行命令**vxlan vni vni-id**，为VPN创建VNI并关联VPN实例。
3. 执行命令**quit**，退出VPN实例视图，返回到系统视图。

步骤3 配置源端VTEP地址。

1. 执行命令**interface nve nve-number**，创建NVE接口，并进入NVE接口视图。
缺省情况下，没有创建NVE接口。
2. 执行命令**mode l3**，将NVE接口切换为三层模式。
缺省情况下，NVE接口处于二层模式。
3. 执行命令**source ip-address**，配置源端VTEP的IP地址。
缺省情况下，源端VTEP没有配置IP地址。
4. 执行命令**quit**，退出NVE接口视图，返回到系统视图。

步骤4 配置三层网关绑定VPN实例，使能分布式网关功能并配置发布主机路由功能。

1. 执行命令**interface vbdif bd-id**，创建BDIF接口，并进入BDIF接口视图。

缺省情况下，没有创建BDIF接口。

2. 执行命令 **ip binding vpn-instance *vpn-instance-name***，将BDIF接口绑定VPN实例。
3. 执行命令 **arp distribute-gateway enable**，使能分布式网关功能。

缺省情况下，分布式网关功能处于去使能。

 说明

在三层网关上使能分布式网关功能后，网关收到网络侧的ARP报文将做丢弃处理，只学习用户侧主机的ARP报文。

4. 执行命令 **arp direct-route enable [route-policy *route-policy-name*]**，配置发布主机路由功能。

 说明

主机路由默认优先级最高，当需要降低主机路由的优先级时，可以通过在绑定的路由策略中配置命令 **apply preference**。

5. 执行命令 **quit**，退出BDIF接口视图，返回到系统视图。

步骤5 使能remote-nexthop属性发布功能。

1. 执行命令 **bgp *as-number-plain***，使能BGP协议，并进入BGP视图。
缺省情况下，未使能BGP协议。
2. 执行命令 **ipv4-family *vpn4***，使能BGP的IPv4地址族，并进入BGP的IPv4地址族视图。
缺省情况下，未创建BGP的IPv4地址族视图。
3. 执行命令 **peer { *group-name* | *ipv4-address* } advertise remote-nexthop**，使能remote-nexthop属性发布功能。
缺省情况下，remote-nexthop属性发布功能处于去使能状态。

步骤6 执行命令 **commit**，提交配置。

---结束

1.6.3.3（可选）配置提升 VXLAN 网络安全性

背景信息

终端用户在不同网段场景下，可部署如下特性提高VXLAN网络安全性：

- 当VXLAN隧道入口收到BUM（Broadcast&Unknown-unicast&Multicast）报文时，本地VTEP会将收到的报文根据VTEP列表进行复制并发送给属于同一个VNI的所有VTEP。为了减少网络中的广播流量，提高网络安全性，可配置静态MAC地址表指定转发路径，也可防止仿冒身份的非法用户骗取数据。
- 静态ARP表项通过手工配置和维护，不会被老化，不会被动态ARP表项覆盖。所以配置静态ARP表项可以增加通信的安全性。静态ARP表项可以限制和指定IP地址的设备通信时只使用指定的MAC地址，此时攻击报文无法修改此表项的IP地址和MAC地址的映射关系，从而保护了本设备和指定设备间的正常通信。

在VXLAN三层网关场景下，可在接入用户的设备上配置静态MAC地址、在VXLAN三层网关上配置静态ARP表项实现提升VXLAN网络的安全性。

操作步骤

步骤1 执行命令`system-view`，进入系统视图。

步骤2 执行命令`mac-address static mac-address bridge-domain bd-id source source-ip-address peer peer-ip vni vni-id`，配置静态MAC地址表项。

缺省情况下，没有配置静态MAC地址表项。

步骤3 执行命令`arp static ip-address mac-address vni vni-id source-ip source-ip peer-ip peer-ip`，配置ARP静态表项。

缺省情况下，没有配置静态ARP地址表项。



`ip-address`的取值必须和三层网关的地址在同一个网段。

步骤4 执行命令`commit`，提交配置。

----结束

1.6.3.4 检查配置结果

前提条件

已经完成VXLAN分布式网关的所有配置。

操作步骤

- 使用命令`display arp broadcast-suppress user bridge-domain bridge-domain-id`，查看指定BD域的ARP广播抑制表。
- 使用命令`display vxlan tunnel [tunnel-id] [verbose]`命令，查看VXLAN隧道的信息。

----结束

1.6.4 配置 VXLAN 双活接入功能

通过VXLAN双活接入功能，实现服务器的两个网卡同时转发流量，从而充分利用网卡和网络带宽资源。

本特性从V100R005C10版本开始支持。

应用场景

在VXLAN网络中，为了提高可靠性，用户经常采用双归接入的方式将安装有双网卡的服务器接入到VXLAN网络，使得当服务器的一个网卡发生故障时不会导致业务中断。

由于上述方案中，仅主网卡能够进行数据报文收发，备网卡不能进行数据报文收发，导致了网卡和链路带宽资源的浪费。用户希望两个网卡可以同时转发流量，实现双活，从而充分利用网卡和网络带宽资源。

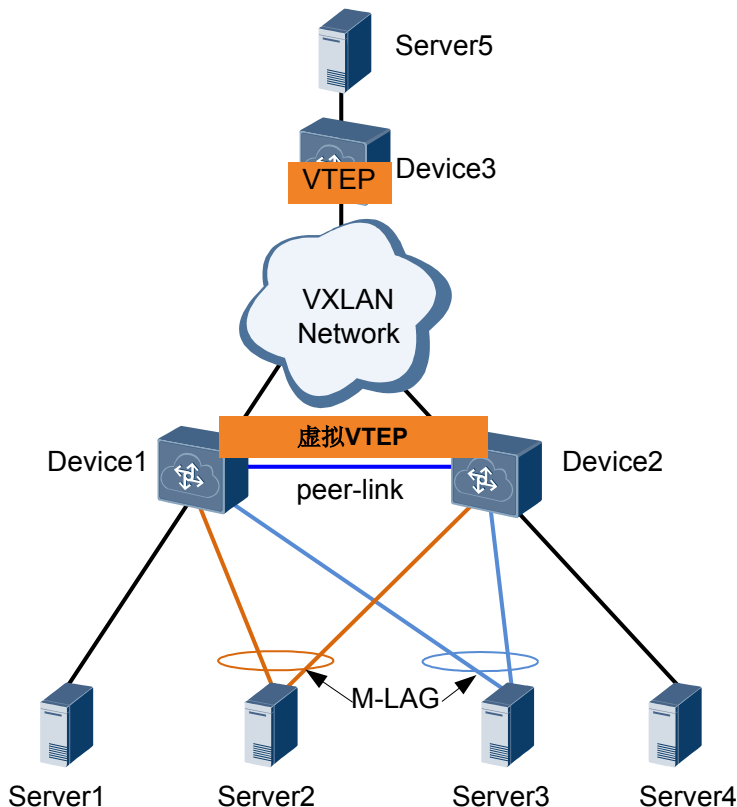
此时，将会存在以下问题：

- 问题1：服务器可能从两个连接服务器的端口收到相同的网络侧发送过来的流量，造成冗余。
- 问题2：与该服务器通信的网络侧设备由于不断收到两台设备发送过去的流量，因此其设备上会不断产生MAC地址漂移现象。

通过配置VXLAN双活接入可以解决上述问题。如图1-50所示，Server2和Server3采用双归接入的方式接入到VXLAN网络：

- 对于问题1，在接入侧，通过应用M-LAG，将Device1和Device2两台接入设备模拟成一台设备，对于服务器来说，相当于是服务器只连接到了一台接入设备上，从而消除了冗余路径。
- 对于问题2，在网络侧，通过使用相同的VTEP IP地址，将两台接入设备模拟成一个VTEP节点，对于远端设备，相当于是通过一台逻辑设备接入到VXLAN网络中，从而消除了MAC地址漂移现象。

图 1-50 VXLAN 双活接入组网图



配置流程

请按照以下顺序进行配置VXLAN双活接入功能。

1.6.4.1 配置双归设备通过 M-LAG 与服务器对接

背景信息

当服务器双归接入两台独立的接入设备时，由于双归接入引入了冗余路径，需要将双归设备在网络中模拟成一台设备，以消除冗余路径。此时，在接入侧，需要应用M-LAG技术，将双归的接入交换机模拟成一台设备。

说明

配置M-LAG时，需要在组成双活系统的两端设备上同时配置。

1.6.4.1.1 配置 DFS Group

背景信息

动态交换服务组DFS Group，主要用于设备之间的配对。为了实现心跳报文的交互，DFS Group需要绑定IP地址并配置对应的UDP端口号，用于和对端进行通信。

当设备双归接入VXLAN网络时，需要配置DFS Group并绑定IP地址。前提是已经配置两台双归设备对应三层接口的IP地址并且保证互通，通常建议采用Loopback接口的IP地址。

操作步骤

步骤1 执行命令`system-view`，进入系统视图。

步骤2 执行命令`dfs-group dfs-group-id`，创建DFS Group并进入DFS-Group视图。
缺省情况下，系统没有创建DFS Group。

步骤3 执行命令`source ip ip-address`，配置DFS Group绑定的IPv4地址。
缺省情况下，DFS Group没有绑定IPv4地址。

步骤4（可选）执行命令`udp port port-number`，配置DFS Group的UDP端口号。
缺省情况下，DFS Group的UDP端口号为61467。

步骤5（可选）执行命令`priority priority`，配置DFS Group的优先级。
优先级用于两台设备间进行主备协商，值越大优先级越高，优先级高的为主用设备。
如果优先级相同，那么比较两台设备的系统MAC地址，MAC地址较小的为主用设备。
缺省情况下，DFS Group的优先级为100。

步骤6 执行命令`quit`，返回系统视图。

步骤7 执行命令`commit`，提交配置。

---结束

1.6.4.1.2 配置 peer-link

背景信息

peer-link链路是位于部署M-LAG的两台设备之间的一条直连聚合链路，用于协议报文的交互和部分流量的传输，保证M-LAG的正常工作。

接口配置为peer-link接口后，该接口上不能再配置其他业务。

前提条件

部署M-LAG的两台设备之间的直连链路已经配置为聚合链路。

说明

为了提高可靠性，建议聚合链路的成员接口分布在不同的单板上，防止某一单板故障导致peer-link故障。

操作步骤

步骤1 执行命令**system-view**，进入系统视图。

步骤2 执行命令**interface eth-trunk trunk-id**，进入Eth-Trunk接口视图。

步骤3 执行命令**mode { lacp-static | lacp-dynamic }**，配置Eth-Trunk的工作模式为LACP模式。

缺省情况下，Eth-Trunk的工作模式为手工负载分担模式。为了提高M-LAG的可靠性，必须配置为LACP模式。

步骤4 执行命令**undo stp enable**，去使能接口的STP功能。

说明

由于两端设备需要模拟成同一个STP根桥，保证设备直连接口不会被阻塞掉，需要将接口STP功能去使能。

步骤5 执行命令**peer-link peer-link-id**，配置接口为peer-link接口。

说明

- 接口配置为peer-link接口后，缺省加入所有VLAN。
- 如果后续需要配置ERPS的控制VLAN、TRILL的Carrier VLAN或FCoE VLAN，需要执行**步骤6**将peer-link接口退出控制VLAN、Carrier VLAN或FCoE VLAN，否则无法配置。
- 如果后续配置了网络侧的VLANIF，建议执行**步骤6**将peer-link接口退出相应的vlan，否则有可能会造成心跳检测失效等问题。

步骤6（可选）执行命令**port vlan exclude { { vlan-id1 [to vlan-id2] } &<1-10> }**，配置peer-link接口不允许通过的VLAN。

步骤7 执行命令**commit**，提交配置。

----结束

1.6.4.1.3 配置绑定 DFS Group

前提条件

部署M-LAG的两台设备与接入设备之间的链路已经分别配置为聚合链路。

说明

如果组成M-LAG的两台设备配置为SVF系统且接入设备通过叶子交换机接入时，需要先在Eth-Trunk视图下配置**extend enable**命令。

操作步骤

- 当聚合链路工作模式采用手工负载分担模式时，执行如下操作：
 1. 执行命令**system-view**，进入系统视图。

2. 执行命令 **interface eth-trunk trunk-id**，进入Eth-Trunk接口视图。
3. 执行命令 **dfs-group dfs-group-id m-lag m-lag-id**，配置绑定DFS Group和用户侧Eth-Trunk接口。

 说明

部署M-LAG的两台设备配置绑定M-LAG的ID必须保持一致。

4. 执行命令 **commit**，提交配置。
- 当聚合链路工作模式采用LACP模式时，执行如下操作：
 1. 执行命令 **system-view**，进入系统视图。
 2. 执行命令 **interface eth-trunk trunk-id**，进入Eth-Trunk接口视图。
 3. 执行命令 **mode { lacp-static | lacp-dynamic }**，配置Eth-Trunk的工作模式为LACP模式。
 4. 执行命令 **dfs-group dfs-group-id m-lag m-lag-id**，配置绑定DFS Group和用户侧Eth-Trunk接口。

 说明

部署M-LAG的两台设备配置绑定M-LAG的ID必须保持一致。

5. 执行命令 **lacp m-lag priority priority**，配置LACP M-LAG的系统优先级。

 说明

- 部署M-LAG的两台设备上成员口Eth-Trunk接口的LACP M-LAG的系统优先级必须保持一致。
- 在系统视图下配置的LACP M-LAG的系统优先级对所有Eth-Trunk接口有效。在Eth-Trunk接口视图下配置的LACP M-LAG的系统优先级仅对该Eth-Trunk接口有效。如果已经在系统视图下执行了本命令，又在指定的Eth-Trunk接口视图下执行本命令，则以Eth-Trunk接口视图下配置的值为准。
- 当设备上配置多个M-LAG时，不同成员口Eth-Trunk接口的LACP M-LAG的系统优先级可以不同，此时需要在Eth-Trunk接口视图下配置LACP M-LAG的系统优先级。
- LACP M-LAG的系统优先级适用于LACP模式的Eth-Trunk组成的M-LAG，而LACP的系统优先级适用于LACP模式的Eth-Trunk接口，可通过 **lacp priority** 命令配置。
如果同时配置了LACP M-LAG的系统优先级和LACP的系统优先级，LACP模式Eth-Trunk加入M-LAG后，使用的是LACP M-LAG的系统优先级，LACP的系统优先级将失效。

6. 执行命令 **lacp m-lag system-id mac-address**，配置LACP M-LAG的系统ID。

 说明

- 部署M-LAG的两台设备上成员口Eth-Trunk接口的LACP M-LAG的系统ID必须保持一致。
- 在系统视图下配置的LACP M-LAG的系统ID对所有Eth-Trunk接口有效。在Eth-Trunk接口视图下配置的LACP M-LAG的系统ID仅对该Eth-Trunk接口有效。如果已经在系统视图下执行了本命令，又在指定的Eth-Trunk接口视图下执行本命令，则以Eth-Trunk接口视图下配置的值为准。
- 当设备上配置多个M-LAG时，不同成员口Eth-Trunk接口的LACP M-LAG的系统ID可以不同，此时需要在Eth-Trunk接口视图下配置LACP M-LAG的系统ID。
- LACP M-LAG的系统ID适用于LACP模式的Eth-Trunk组成的M-LAG，而LACP的系统ID适用于LACP模式的Eth-Trunk接口，LACP的系统ID是固定的（主控板的以太网MAC地址），不能通过配置更改。

7. 执行命令 **commit**，提交配置。

---结束

1.6.4.1.4 检查配置结果

操作步骤

- 执行命令 **display dfs-group dfs-group-id [node node-id m-lag [brief] | peer-link]**，查看M-LAG的信息。

---结束

后续处理

完成M-LAG配置后，如果peer-link故障但心跳状态正常会导致状态为备的设备上部分接口处于ERROR DOWN状态。Error-Down是指设备检测到故障后将接口状态设置为ERROR DOWN状态，此时接口不能收发报文，接口指示灯为常灭。可以通过**display error-down recovery**命令可以查看设备上所有被Error-Down的接口信息。

- 当M-LAG应用于TRILL网络的双归接入时，peer-link故障但心跳状态正常会导致状态为备的设备上M-LAG接口处于ERROR DOWN状态。一旦peer-link故障恢复，处于ERROR DOWN状态的物理接口默认将在2分钟后自动恢复为Up状态。
- 当M-LAG应用于普通以太网、VXLAN网络或IP网络的双归接入时，peer-link故障但心跳状态正常会导致状态为备的设备上除管理网口、peer-link接口和堆叠口以外的物理接口处于ERROR DOWN状态。一旦peer-link故障恢复，处于ERROR DOWN状态的物理接口将自动恢复为Up状态。

接口被Error-Down时，建议先排除引起peer-link故障的原因，不建议直接手动恢复或在系统视图下执行命令**error-down auto-recovery cause m-lag interval interval-value**使能接口状态自动恢复为Up的功能，否则可能会导致业务多包、丢包或不通等故障，请谨慎操作。

1.6.4.2 配置双归设备上的虚拟 VTEP

背景信息

在网络侧，通过为双归设备配置相同的VTEP IP地址，可以将两台接入设备模拟成一个VTEP节点，从而避免环路或MAC地址漂移现象。

说明

请确保两台双归设备上通过命令**source**和**vni**配置的源端VTEP的IP地址和同一VNI的头端复制列表相同。

实际场景中，此任务一般无需单独配置，请在配置[配置VXLAN隧道转发业务流量](#)任务时，完成此配置。

操作步骤

- 步骤1** 执行命令**system-view**，进入系统视图。
- 步骤2** 执行命令**interface nve nve-number**，创建NVE接口，并进入NVE接口视图。
缺省情况下，未创建NVE接口。
- 步骤3** 执行命令**source ip-address**，配置源端VTEP的IP地址。

缺省情况下，源端VTEP没有配置IP地址。推荐使用Loopback接口的IP地址。

步骤4 执行命令 `vni vni-id head-end peer-list ip-address &<1-10>`，配置VNI的头端复制列表。

缺省情况下，没有配置VNI头端复制列表。

步骤5 执行命令 `commit`，提交配置。

---结束

1.7 维护 VXLAN

通过维护VXLAN，可以实现清除VXLAN统计数据、监控VXLAN的运行状况等。

1.7.1 统计并查看 VXLAN 统计信息

背景信息

当需要检查网络状况或处理网络故障时，可以在设备上打开BD的流量统计功能，统计通过VXLAN隧道的流量信息。

操作步骤

步骤1 执行命令 `system-view`，进入系统视图。

步骤2 执行命令 `bridge-domain bd-id`，创建BD，并进入BD视图。

缺省情况下，没有创建广播域BD。

步骤3 执行命令 `statistics enable`，使能BD内报文统计功能。

缺省情况下，BD内报文统计功能处于去使能状态。

步骤4 执行命令 `commit`，提交配置。

---结束

后续处理

- 执行命令 `display bridge-domain bd-id statistics`，查看BD内报文的统计信息。

1.7.2 清除 BD 内报文统计信息

背景信息

当需要查看一定时间内某个BD内报文流量信息，这时必须在统计开始前清除该BD内的报文统计信息，使BD内的报文重新进行统计，保证统计的信息正确性。

说明

清除BD内报文统计信息后，以前的信息将无法恢复，务必仔细确认。

操作步骤

- 在用户视图下，执行命令 **reset bridge-domain *bd-id* statistics**，清除指定BD内报文统计信息。

----结束

1.7.3 监控 VXLAN 运行状况

背景信息

在日常维护工作中，可以在任意视图下选择执行以下命令，了解VXLAN的运行状况。

操作步骤

- 执行命令 **display bridge-domain [*bd-id* [**brief** | **verbose**]]**，查看广播域BD的配置信息。
- 执行命令 **display mac-address [*mac-address*] bridge-domain *bd-id***，查看BD内所有MAC地址表项。

如需要清除BD内学习到的动态MAC地址表项，可在用户视图下执行命令 **reset mac-address bridge-domain *bd-id***。清除动态MAC地址表项后，可能导致业务短暂的中断，且历史表项不可恢复，务必仔细确认。

- 执行命令 **display mac-address static bridge-domain *bd-id***，查看BD内静态MAC地址表项。
- 执行命令 **display mac-address total-number [**static**] bridge-domain *bd-id***，查看BD内的MAC地址表项的数目信息。

----结束

1.7.4 配置 VXLAN 告警上报功能

背景信息

为了方便运维，及时了解VXLAN网络的运行状态，可以配置VXLAN告警上报功能，将VXLAN的状态变化通知给网管系统，提醒用户注意。

操作步骤

步骤1 执行命令 **system-view**，进入系统视图。

步骤2 执行命令 **snmp-agent trap enable feature-name nvo3 [trap-name { *hwnvo3vxlantnl*down | *hwnvo3vxlantnl*up }]**，打开VXLAN告警开关。

缺省情况下，VXLAN告警开关处于关闭状态。

步骤3 执行命令 **commit**，提交配置。

----结束

检查配置结果

VXLAN告警上报功能配置成功后，可以按如下操作查看VXLAN告警开关的状态信息。

- 执行命令 **display snmp-agent trap feature-name nvo3 all**，可以查看到 VXLAN 模块的所有告警开关信息。

1.8 配置举例

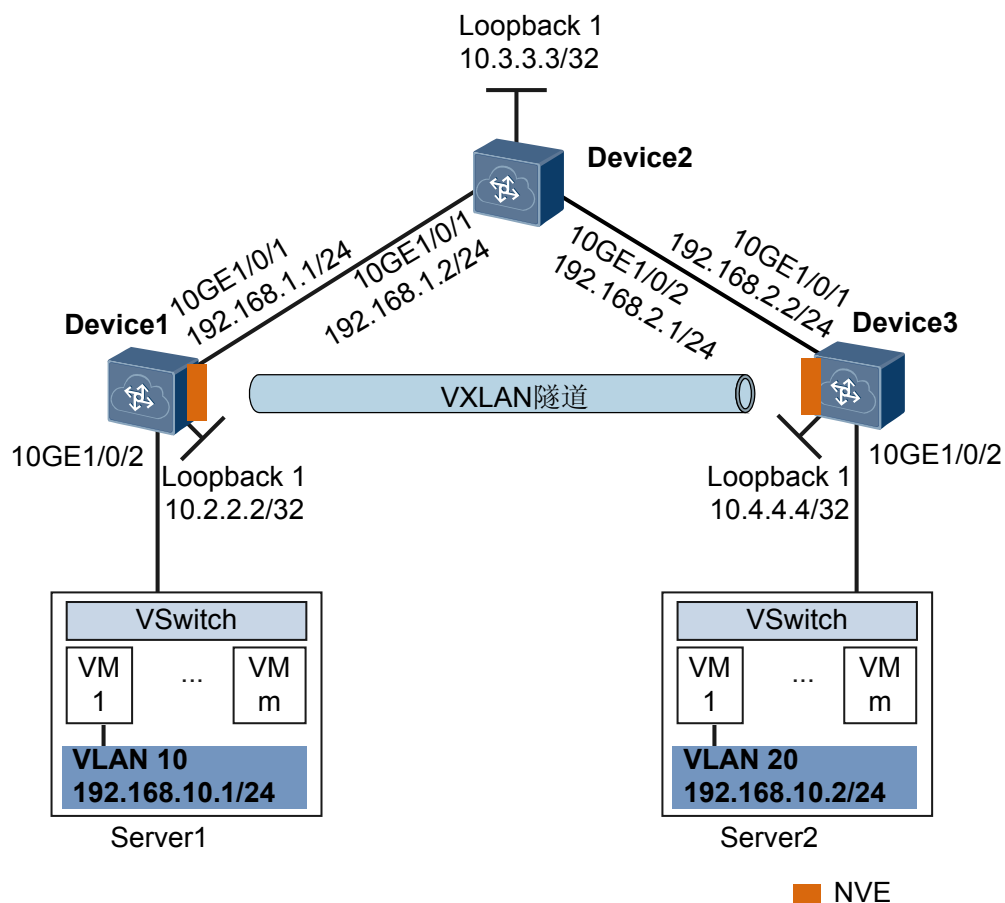
介绍 VXLAN 配置举例，配置举例中包括组网需求、配置思路、配置过程和配置文件。

1.8.1 配置同网段用户通过 VXLAN 隧道互通示例（单机方式）

组网需求

如图 1-51 所示，某企业在不同的数据中心中都拥有自己的 VM，服务器 1 上的 VM1 属于 VLAN10，服务器 2 上的 VM1 属于 VLAN20，且位于同网段。现需要通过 VXLAN 隧道实现不同数据中心相同 VM 的互通。

图 1-51 配置同网段用户通过 VXLAN 隧道互通组网图



配置思路

采用如下思路配置同网段用户通过 VXLAN 隧道互通：

1. 分别在 Device1、Device2 和 Device3 上配置路由协议，保证网络三层互通。

2. 分别在Device1和Device3上配置业务接入点实现区分业务流量。
3. 分别在Device1和Device3上配置VXLAN隧道实现转发业务流量。

数据准备

为完成此配置例，需准备如下的数据：

- VM所属的VLAN ID分别是VLAN10和VLAN20。
- 网络中设备互连的接口IP地址。
- 网络中使用的路由类型是OSPF（Open Shortest Path First）。
- 广播域BD ID是BD10。
- VXLAN网络标识VNI ID是VNI 5010。

操作步骤

步骤1 配置路由协议

按图1-51分别配置转发器Device1、Device2、Device3各接口IP地址。配置OSPF时，需要注意需要发布转发器的32位Loopback接口地址。

配置转发器Device1。Device2和Device3的配置与Device1配置类似，这里不再赘述。

```
<HUAWEI> system-view
[~HUAWEI] sysname Device1
[*HUAWEI] commit
[~Device1] interface loopback 1
[*Device1-LoopBack1] ip address 10.2.2.2 32
[*Device1-LoopBack1] quit
[*Device1] interface 10ge 1/0/1
[*Device1-10GE1/0/1] undo portswitch
[*Device1-10GE1/0/1] ip address 192.168.1.1 24
[*Device1-10GE1/0/1] quit
[*Device1] ospf
[*Device1-ospf-1] area 0
[*Device1-ospf-1-area-0.0.0.0] network 10.2.2.2 0.0.0.0
[*Device1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Device1-ospf-1-area-0.0.0.0] quit
[*Device1-ospf-1] quit
[*Device1] commit
```

OSPF成功配置后，转发器与转发器之间可通过OSPF协议发现对方的Loopback接口的IP地址，并能互相ping通。以Device1 ping Device3的显示为例。

```
[~Device1] ping 10.4.4.4
PING 10.4.4.4: 56 data bytes, press CTRL_C to break
  Reply from 10.4.4.4: bytes=56 Sequence=1 ttl=254 time=5 ms
  Reply from 10.4.4.4: bytes=56 Sequence=2 ttl=254 time=2 ms
  Reply from 10.4.4.4: bytes=56 Sequence=3 ttl=254 time=2 ms
  Reply from 10.4.4.4: bytes=56 Sequence=4 ttl=254 time=3 ms
  Reply from 10.4.4.4: bytes=56 Sequence=5 ttl=254 time=3 ms

--- 10.4.4.4 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/3/5 ms
```

步骤2 配置隧道模式

配置Device1。Device3的配置与Device1配置类似，这里不再赘述。

```
[~Device1] ip tunnel mode vxlan
[*Device1] commit
```

说明

缺省情况下，隧道模式为VXLAN，无需配置此任务。当用户在使用GRE隧道后，需切换至VXLAN时，请在设备上执行此任务。此命令功能需要保存配置并重启设备才能生效，您可以选择立即重启或完成所有配置后再重启。

步骤3 分别在Device1和Device3上配置VXLAN业务接入点

配置Device1。Device3的配置与Device1配置类似，这里不再赘述。

```
[~Device1] bridge-domain 10
[*Device1-bd10] quit
[*Device1] interface 10ge 1/0/2.1 mode l2
[*Device1-10GE1/0/2.1] encapsulation dot1q vid 10
[*Device1-10GE1/0/2.1] bridge-domain 10
[*Device1-10GE1/0/2.1] quit
[*Device1] commit
```

步骤4 分别在Device1和Device3上配置VXLAN隧道

配置Device1。Device3的配置与Device1配置类似，这里不再赘述。

```
[~Device1] bridge-domain 10
[~Device1-bd10] vxlan vni 5010
[*Device1-bd10] quit
[*Device1] interface nve 1
[*Device1-Nve1] source 10.2.2.2
[*Device1-Nve1] vni 5010 head-end peer-list 10.4.4.4
[*Device1-Nve1] quit
[*Device1] commit
```

步骤5 检查配置结果

上述配置成功后，在转发器Device1和Device3上执行**display vxlan vni**命令可查看到VNI的状态是**up**；执行**display vxlan tunnel**命令可查看到VXLAN隧道的信息。以Device1显示为例。

```
[~Device1] display vxlan vni
Number of vxlan vni : 1
VNI          BD-ID          State
-----
5010         10             up
[~Device1] display vxlan tunnel
Number of vxlan tunnel : 1
Tunnel ID   Source          Destination     State  Type
-----
4026531841  10.2.2.2        10.4.4.4        up     static
```

配置完成后，同网段用户通过VXLAN隧道可以互通。

----结束

配置文件

● Device1的配置文件

```
#
sysname Device1
#
bridge-domain 10
vxlan vni 5010
#
interface 10GE1/0/1
```

```
undo portswitch
ip address 192.168.1.1 255.255.255.0
#
interface 10GE1/0/2.1 mode 12
encapsulation dot1q vid 10
bridge-domain 10
#
interface LoopBack1
ip address 10.2.2.2 255.255.255.255
#
interface Nve1
source 10.2.2.2
vni 5010 head-end peer-list 10.4.4.4
#
ospf 1
area 0.0.0.0
network 10.2.2.2 0.0.0.0
network 192.168.1.0 0.0.0.255
#
return
```

● Device2的配置文件

```
#
sysname Device2
#
interface 10GE1/0/1
undo portswitch
ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/2
undo portswitch
ip address 192.168.2.1 255.255.255.0
#
interface LoopBack1
ip address 10.3.3.3 255.255.255.255
#
ospf 1
area 0.0.0.0
network 10.3.3.3 0.0.0.0
network 192.168.1.0 0.0.0.255
network 192.168.2.0 0.0.0.255
#
return
```

● Device3的配置文件

```
#
sysname Device3
#
bridge-domain 10
vxlan vni 5010
#
interface 10GE1/0/1
undo portswitch
ip address 192.168.2.2 255.255.255.0
#
interface 10GE1/0/2.1 mode 12
encapsulation dot1q vid 20
bridge-domain 10
#
interface LoopBack1
ip address 10.4.4.4 255.255.255.255
#
interface Nve1
source 10.4.4.4
vni 5010 head-end peer-list 10.2.2.2
#
ospf 1
```

```
area 0.0.0.0
 network 10.4.4.4 0.0.0.0
 network 192.168.2.0 0.0.0.255
#
return
```

1.8.2 配置不同网段用户通过 VXLAN 三层网关通信示例（SNC 控制器方式）

组网需求

为了方便控制与部署，引入了控制器概念。控制器通过OpenFlow协议将信息下发给转发器，以实现控制器统一维护管理。

作为云计算的核心技术之一，服务器虚拟化凭借其大幅降低IT成本、提高业务部署灵活性、降低运维成本等优势已经得到越来越多的认可和部署。服务器虚拟化后，可支持多租户接入。因为一台服务器可虚拟多台虚拟机，而一台虚拟机相当于一台主机，主机数量发生了数量级的变化，这也为虚拟网络带来问题。例如：虚拟机规模受网络规格限制、网络隔离能力限制及虚拟机迁移范围受网络架构限制。

为了充分发挥服务器虚拟化的优势、解决服务器虚拟化后带来的问题，可以通过部署 VXLAN（Virtual eXtensible Local Area Network）实现服务器虚拟化后可满足16M租户接入，且租户可以规划自己的虚拟网络，不需要考虑物理网络IP地址和广播域的限制。

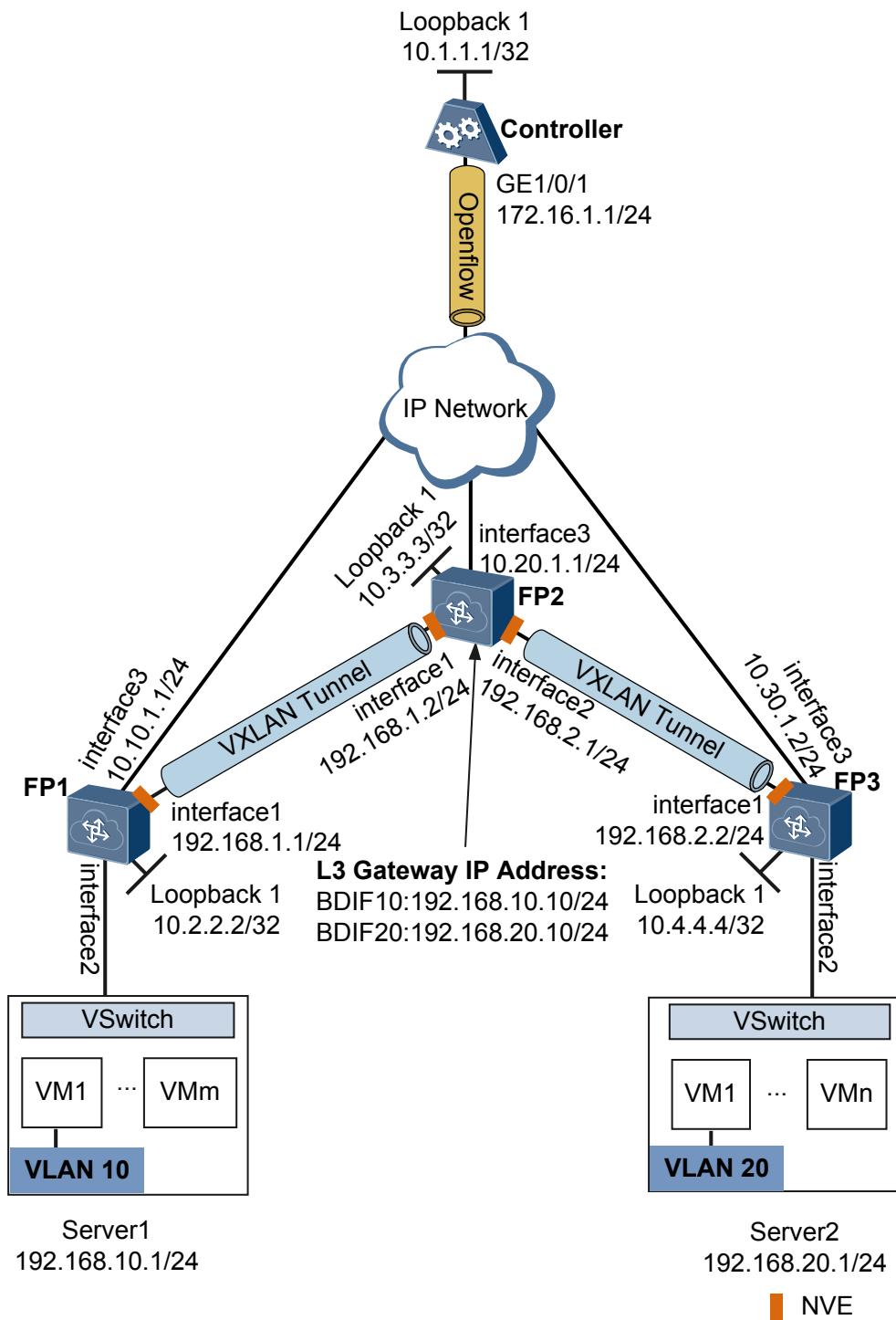
如图1-52所示，某企业在不同的数据中心中都拥有自己的VM，服务器1上的VM1属于VLAN10，服务器2上的VM1属于VLAN20，且两台服务器位于不同网段。现需要通过 VXLAN实现在同一租户网络内，位于不同数据中心的VM的通讯。

图 1-52 基于 SDN 控制器的 VXLAN 典型组网图

说明

本例的配置主要在SNC Controller上进行。SNC Controller的配置仅供参考，详细配置请访问<http://support.huawei.com>，选择“产品支持 > 固定网络 > 运营商IP > 业务网关与控制器 > SIG > SNC >”，获取对应的控制器文档。

本例中FP1、FP2、FP3设备上的interface1、interface2、interface3分别代表10GE1/0/1、10GE1/0/2、10GE1/0/3。



注意事项

- 配置VXLAN前，必须保证控制器和转发器三层路由可达。
- 为了使VXLAN之间，以及VXLAN和非VXLAN之间能够进行通信，VXLAN引入了VXLAN网关。VXLAN网关分为：

- 二层网关：用于同一网段的租户通信。
- 三层网关：用于不同网段的租户通信或VXLAN和非VXLAN用户间的通信。

为了成功实现不同网段的VM互通，VM的缺省网关地址必须是对应三层网关上的BDIF接口的IP地址。

无论二层网关还是三层网关，VXLAN的配置相同，不同点在于三层网关上需要创建BDIF接口并配置IP地址。

- 基于SDN+VXLAN解决方案中，引入了双控制面（SNC控制面+转发器本地控制面），双控制面存在转发资源共享。例如：接口的本地VLAN资源、BD ID资源、VNI资源、IP资源等。为了避免转发资源使用冲突，建议在部署业务前统一规划资源，保证SNC控制面和转发器本地控制面使用不同的编号资源。
- 当在SNC控制器上为某个FP设备配置VXLAN时，若需要配置二层子接口接入BD并为该BD创建对应的BDIF接口，请在vServiceIf二层子接口视图下配置命令**rewrite pop single**。

配置思路

采用如下的思路配置基于SNC控制器的VXLAN：

1. 分别在转发器FP1、FP2、FP3上配置隧道模式。
2. 分别在控制器Controller、转发器FP1、FP2、FP3上配置路由协议，保证网络三层互通。

本示例以OSPF（Open Shortest Path First）为例。

3. 分别在控制器Controller、转发器FP1、FP2、FP3上配置Openflow，建立控制器和转发器之间的通信通道。
4. 在控制器上配置VXLAN三层网关，配置思路如下：
 - a. 分别为转发器FP1、FP3配置业务接入点，实现不同的业务报文通过不同的接口接入网络。
 - b. 分别为转发器FP1、FP2、FP3配置VXLAN，FP1与FP2、FP2与FP3之间建立VXLAN隧道，通过VXLAN隧道实现数据转发。
 - c. 分别为转发器FP1、FP2、FP3配置静态MAC地址表，指导报文正确二层转发。
 - d. 为FP3配置三层逻辑接口BDIF接口，并配置IP地址，确定三层网关地址。
 - e. 为FP3配置静态ARP地址表，指导业务报文正确三层转发。
 - f. 分别为FP1、FP2使能ARP代答，避免ARP广播请求报文给网络带来广播风暴。

操作步骤

步骤1 配置隧道模式

配置转发器FP1。

```
<HUAWEI> system-view
[~HUAWEI] sysname FP1
[*HUAWEI] commit
[~FP1] ip tunnel mode vxlan
[*FP1] commit
```



缺省情况下，隧道模式为VXLAN，无需配置此任务。当用户在使用GRE隧道后，需切换至VXLAN时，请在设备上执行此任务。此命令功能需要保存配置并重启设备才能生效，您可以选择立即重启或完成所有配置后再重启。

FP2和FP3的配置与FP1类似，这里不再赘述。具体配置过程略，请参考本配置举例中的配置文件。

步骤2 配置路由协议

按图1-52分别配置控制器Controller、转发器FP1、FP2、FP3各接口IP地址。配置OSPF时，注意需要发布控制器、转发器的32位Loopback接口地址。

● 配置转发器FP1

表 1-12 FP1 配置

步骤	命令	说明
1	<pre>[~FP1] vlan batch 100 300 [*FP1] interface vlanif 100 [*FP1-Vlanif100] ip address 192.168.1.1 24 [*FP1-Vlanif100] quit [*FP1] interface vlanif 300 [*FP1-Vlanif300] ip address 10.10.1.1 24 [*FP1-Vlanif300] quit [*FP1] interface loopback 1 [*FP1-LoopBack1] ip address 10.2.2.2 32 [*FP1-LoopBack1] quit [*FP1] interface 10ge 1/0/1 [*FP1-10GE1/0/1] port link-type trunk [*FP1-10GE1/0/1] port trunk allow-pass vlan 100 [*FP1-10GE1/0/1] quit [*FP1] interface 10ge 1/0/3 [*FP1-10GE1/0/3] port link-type trunk [*FP1-10GE1/0/3] port trunk allow-pass vlan 300 [*FP1-10GE1/0/3] quit</pre>	配置接口IP地址。
2	<pre>[*FP1] ospf [*FP1-ospf-1] area 0 [*FP1-ospf-1-area-0.0.0.0] network 10.2.2.2 0.0.0.0 [*FP1-ospf-1-area-0.0.0.0] network 10.10.1.0 0.0.0.255 [*FP1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255 [*FP1-ospf-1-area-0.0.0.0] quit [*FP1-ospf-1] quit [*FP1] commit</pre>	配置OSPF。

- 在Controller、FP2、FP3与配置FP1类似，这里不再赘述。具体配置过程略，请参考本配置举例中的配置文件。

OSPF成功配置后，控制器与转发器，转发器与转发器之间可通过OSPF协议发现对方的Loopback接口的IP地址，并能互相ping通。

以Controller的显示为例。

```
[~Controller] ping 10.4.4.4
PING 10.4.4.4: 56 data bytes, press CTRL_C to break
Reply from 10.4.4.4: bytes=56 Sequence=1 ttl=255 time=4 ms
Reply from 10.4.4.4: bytes=56 Sequence=2 ttl=255 time=2 ms
```

```

Reply from 10.4.4.4: bytes=56 Sequence=3 ttl=255 time=2 ms
Reply from 10.4.4.4: bytes=56 Sequence=4 ttl=255 time=2 ms
Reply from 10.4.4.4: bytes=56 Sequence=5 ttl=255 time=1 ms

--- 10.4.4.4 ping statistics ---
 5 packet(s) transmitted
 5 packet(s) received
 0.00% packet loss
 round-trip min/avg/max = 1/2/4 ms
    
```

步骤3 配置Openflow

- 在控制器Controller上配置与转发器FP1的连接

表 1-13 Controller 配置

步骤	命令	说明
1	<code>[~Controller] sdn controller</code>	将设备设置为SDN控制器，并进入SDN控制器视图。
2	<code>[*Controller-sdn-controller] openflow listening-ip 10.1.1.1</code>	配置SDN控制器侦听地址。
3	<code>[*Controller-sdn-controller] fp-id 10</code>	配置转发器FP1的标识是FP10，并进入FP视图。
4	<code>[*Controller-sdn-controller-fp10] type huawei-default</code>	配置转发器的设备类型： <ul style="list-style-type: none"> ● huawei-default: 表示转发器是华为设备。 ● ovs-default: 表示转发器是OVS（openvswitch）设备。
5	<code>[*Controller-sdn-controller-fp10] version default</code>	配置转发器的版本是default。
6	<code>[*Controller-sdn-controller-fp10] role default</code>	配置转发器的角色是default。
7	<code>[*Controller-sdn-controller-fp10] openflow controller</code>	配置控制器与转发器之间的通信通道采用Openflow连接并进入Openflow视图。
8	<code>[*Controller-sdn-controller-fp10-openflow] peer-address 10.2.2.2</code> <code>[*Controller-sdn-controller-fp10-openflow] quit</code> <code>[*Controller-sdn-controller-fp10] quit</code> <code>[*Controller-sdn-controller] quit</code> <code>[*Controller] commit</code> <code>[~Controller] quit</code>	指定转发器FP1的Loopback地址。

在Controller上配置与FP2、FP3的链接与配置FP1类似，这里不再赘述。具体配置过程略，请参考本配置举例中的配置文件。

- 配置转发器FP1

表 1-14 FP1 配置

步骤	命令	说明
1	[~FP1] sdn agent	将FP1设置为转发器，并进入SDN Agent视图。
2	[*FP1-sdn-agent] controller-ip 10.1.1.1	指定控制器的Loopback地址，并进入Controller视图。
3	[*FP1-sdn-agent-ctrl-10.1.1.1] openflow agent	配置转发器与控制器之间的通信通道采用Openflow连接并进入Openflow Agent视图。
4	[*FP1-sdn-agent-ctrl-10.1.1.1-openflow] transport-address 10.2.2.2 [*FP1-sdn-agent-ctrl-10.1.1.1] quit [*FP1-sdn-agent] quit [*FP1] commit	配置Openflow连接的本端地址。

转发器FP2、FP3配置与FP1类似，这里不再赘述。具体配置过程略，请参考本配置举例中的配置文件。

上述配置成功后，分别在转发器和控制器上执行**display sdn openflow session**命令可查看到Openflow连接的状态是“REGISTERED”，说明Openflow通道建立成功。以控制器显示为例：

[~Controller] **display sdn openflow session**

FPID	AgentAddr	ListeningAddr	UpTime	State
10	10.2.2.2	10.1.1.1	0d00h23m28s	REGISTERED
20	10.3.3.3	10.1.1.1	0d00h55m17s	REGISTERED
30	10.4.4.4	10.1.1.1	0d00h00m05s	REGISTERED

步骤4 在控制器上配置VXLAN三层网关

1. 配置业务接入点

- 在控制器上配置FP1

表 1-15 FP1 配置

步骤	命令	说明
1	<pre>[~Controller] quit <Controller> switch fp 10 <Controller-FP10> system-view [-Controller-FP10] bridge-domain 10 [*Controller-FP10-bd10] quit</pre>	创建广播域BD。
2	<pre>[*Controller-FP10] interface vserviceif 10:1</pre>	<p>创建vServiceIf接口，并进入vServiceIf接口视图。</p> <p>在VXLAN网络中，业务接入点统一表示为二层vServiceIf子接口。</p>
3	<pre>[*Controller-FP10-vServiceIf10:1] binding interface 10GE1/0/2 [*Controller-FP10-vServiceIf10:1] vlan assign 10 [*Controller-FP10-vServiceIf10:1] quit</pre>	<p>为vServiceIf接口绑定物理接口。</p> <p>同时基于接口分配VLAN资源，即指定接口允许通过的VLAN资源。</p>
4	<pre>[*Controller-FP10] interface vserviceif 10:1.1 mode 12</pre>	创建二层vServiceIf子接口，并进入子接口视图。
5	<pre>[*Controller-FP10-vServiceIf10:1.1] encapsulation dot1q vid 10</pre>	配置二层vServiceIf子接口的流封装类型，允许接口接收携带一层VLAN Tag是10的报文。
6	<pre>[*Controller-FP10-vServiceIf10:1.1] bridge-domain 10 [*Controller-FP10-vServiceIf10:1.1] quit [*Controller-FP10] commit [-Controller-FP10] quit <Controller-FP10> quit</pre>	将二层vServiceIf子接口加入BD，允许报文通过广播域BD转发。

- 在控制器上配置FP3

表 1-16 FP3 配置

步骤	命令	说明
1	<pre><Controller> switch fp 30 <Controller-FP30> system-view [-Controller-FP30] bridge-domain 20 [*Controller-FP30-bd20] quit</pre>	创建广播域BD。
2	<pre>[*Controller-FP30] interface vserviceif 30:1</pre>	创建vServiceIf接口，并进入vServiceIf接口视图。

步骤	命令	说明
3	<pre>[*Controller-FP30-vServiceIf30:1] binding interface 10GE1/0/2 [*Controller-FP30-vServiceIf30:1] vlan assign 20 [*Controller-FP30-vServiceIf30:1] quit</pre>	<p>为vServiceIf接口绑定物理接口。</p> <p>同时基于接口分配VLAN资源，即指定接口允许通过的VLAN资源。</p>
4	<pre>[*Controller-FP30] interface vserviceif 30:1.1 mode 12</pre>	<p>创建二层vServiceIf子接口，并进入子接口视图。</p>
5	<pre>[*Controller-FP30-vServiceIf30:1.1] encapsulation dot1q vid 20</pre>	<p>配置二层vServiceIf子接口的流封装类型，允许接口接收携带一层VLAN Tag是20的报文。</p>
6	<pre>[*Controller-FP30-vServiceIf30:1.1] bridge-domain 20 [*Controller-FP30-vServiceIf30:1.1] quit [*Controller-FP30] commit [-Controller-FP30] quit <Controller-FP30> quit</pre>	<p>将二层vServiceIf子接口加入BD，允许报文通过广播域BD转发。</p>

2. 配置VXLAN

- 在控制器上配置FP1

表 1-17 FP1 配置

步骤	命令	说明
1	<pre><Controller> switch fp 10 <Controller-FP10> system-view [-Controller-FP10] bridge-domain 10 [-Controller-FP10-bd10] vxlan vni 10 [*Controller-FP10-bd10] quit</pre>	<p>进入FP1转发器视图，创建VXLAN网络标识VNI并关联广播域BD，将VNI以1:1方式映射到BD，通过BD转发流量。</p>
2	<pre>[*Controller-FP10] interface nve 10:1</pre>	<p>创建NVE接口，并进入NVE接口视图。</p>
3	<pre>[*Controller-FP10-Nve10:1] source 10.2.2.2</pre>	<p>配置源端VTEP的IP地址。</p>
4	<pre>[*Controller-FP10-Nve10:1] overlay-encapsulation vxlan</pre>	<p>配置NVE接口的封装类型是VXLAN。</p>

步骤	命令	说明
5	<pre>[*Controller-FP10-Nve10:1] vni 10 head-end peer-list 10.3.3.3 [*Controller-FP10-Nve10:1] quit [*Controller-FP10] commit [-Controller-FP10] quit <Controller-FP10> quit</pre>	<p>配置头端复制列表。通过头端复制列表，源端NVE接口将收到的BUM（Broadcast&Unknown-unicast&Multicast）报文，根据VTEP列表进行复制并发送给属于同一个VNI的所有VTEP。</p>

- 在控制器上配置FP2

表 1-18 FP2 配置

步骤	命令	说明
1	<pre><Controller> switch fp 20 <-Controller-FP20> system-view [-Controller-FP20] bridge-domain 10 [*Controller-FP20-bd10] vxlan vni 10 [*Controller-FP20-bd10] quit [-Controller-FP20] bridge-domain 20 [*Controller-FP20-bd20] vxlan vni 20 [*Controller-FP20-bd20] quit</pre>	<p>进入FP2转发器视图，创建VXLAN网络标识VNI并关联广播域BD，将VNI以1:1方式映射到BD，通过BD转发流量。</p>
2	<pre>[*Controller-FP20] interface nve 20:1</pre>	<p>创建NVE接口，并进入NVE接口视图。</p>
3	<pre>[*Controller-FP20-Nve20:1] source 10.3.3.3</pre>	<p>配置源端VTEP的IP地址。</p>
4	<pre>[*Controller-FP20-Nve20:1] overlay-encapsulation vxlan</pre>	<p>配置NVE接口的封装类型是VXLAN。</p>
5	<pre>[*Controller-FP20-Nve20:1] vni 10 head-end peer-list 10.2.2.2 [*Controller-FP20-Nve20:1] vni 20 head-end peer-list 10.4.4.4 [*Controller-FP20-Nve20:1] quit [*Controller-FP20] commit [*Controller-FP20] quit <Controller-FP20> quit</pre>	<p>配置头端复制列表。</p>

- 在控制器上配置FP3

表 1-19 FP3 配置

步骤	命令	说明
1	<pre><Controller> switch fp 30 <Controller-FP30> system-view [-Controller-FP30] bridge-domain 20 [-Controller-FP30-bd20] vxlan vni 20 [*Controller-FP30-bd20] quit</pre>	进入FP3转发器视图，创建VXLAN网络标识VNI并关联广播域BD，将VNI以1:1方式映射到BD，通过BD转发流量。
2	<pre>[*Controller-FP30] interface nve 30:1</pre>	创建NVE接口，并进入NVE接口视图。
3	<pre>[*Controller-FP30-Nve30:1] source 10.4.4.4</pre>	配置源端VTEP的IP地址。
4	<pre>[*Controller-FP30-Nve30:1] overlay-encapsulation vxlan</pre>	配置NVE接口的封装类型是VXLAN。
5	<pre>[*Controller-FP30-Nve30:1] vni 20 head-end peer-list 10.3.3.3 [*Controller-FP30-Nve30:1] quit [*Controller-FP30] commit [-Controller-FP30] quit <Controller-FP30> quit</pre>	配置头端复制列表。

上述配置成功后，在控制器上指定转发器视图下执行**display vxlan vni**命令可查看到VNI的状态是Up；执行**display vxlan tunnel**命令可查看到VXLAN隧道的信息。以FP2转发器显示为例：

```
<Controller> switch fp 20
<Controller-FP20> display vxlan vni
Number of vxlan vni: 2
VNI          BD-ID          State
-----
10           10             up
20           20             up
<Controller-FP20> display vxlan tunnel
Number of Vxlan tunnel : 2
Tunnel ID   Source          Destination     State  Type
-----
33686018    10.3.3.3        10.2.2.2        up     static
67372036    10.3.3.3        10.4.4.4        up     static
```

3. 在控制器上分别为转发器FP1、FP2、FP3配置静态MAC地址表

- 在控制器上配置FP1

表 1-20 FP1 配置

步骤	命令	说明
1	<pre><Controller-FP20> quit <Controller> switch fp 10 <Controller-FP10> system-view [-Controller-FP10] mac-address static 38eb-d921-0301 bridge-domain 10 nve 10:1 peer 10.3.3.3 vni 10 [*Controller-FP10] mac-address static 1-1-1 vserviceif10:1.1 bridge-domain 10 vid 10 [*Controller-FP10] commit [-Controller-FP10] quit</pre>	<p>配置静态MAC地址表项，包括：</p> <ul style="list-style-type: none"> ● VXLAN隧道的静态MAC地址表项：当VXLAN隧道入口收到BUM报文时，为了减少网络中的广播流量，提高网络安全性，可配置静态MAC地址表指定转发路径，也可防止防冒身份的非法用户骗取数据。 ● VXLAN用户侧的静态MAC地址表项：当部署VXLAN的设备向用户侧转发BUM报文时，为了减少用户网络的广播流量，提高网络安全性，可配置静态MAC地址表指定转发路径，以单播的形式向用户网络转发报文。

- 在控制器上配置FP2

表 1-21 FP2 配置

步骤	命令	说明
1	<pre><Controller-FP10> quit <Controller> switch fp 20 <Controller-FP20> system-view [-Controller-FP20] mac-address static 38eb-d911-0301 bridge-domain 10 nve 20:1 peer 10.2.2.2 vni 10 [*Controller-FP20] mac-address static 38eb-d931-0301 bridge-domain 20 nve 20:1 peer 10.4.4.4 vni 20 [*Controller-FP20] commit [-Controller-FP20] quit</pre>	<p>配置静态MAC地址表项。</p>

- 在控制器上配置FP3

表 1-22 FP3 配置

步骤	命令	说明
1	<pre><Controller-FP20> quit <Controller> switch fp 30 <Controller-FP30> system-view [-Controller-FP30] mac-address static 38eb-d921-0302 bridge-domain 20 nve 30:1 peer 10.3.3.3 vni 20 [*Controller-FP30] mac-address static 2-2-2 vServiceIf30:1.1 bridge-domain 20 vid 20 [*Controller-FP30] commit [-Controller-FP30] quit <Controller-FP30> quit</pre>	配置静态MAC地址表项。

上述配置成功后，在控制器上转发器视图下执行 **display mac-address vxlan** 命令可查看到 VXLAN 的静态 MAC 地址表项信息。

在控制器上以转发器 FP2 显示为例。

```
<Controller> switch fp 20
<Controller-FP20> display mac-address vxlan
```

MAC Address	BD/VLAN	PORT	PEER-IP	VNI	Type
38eb-d911-0301	10	Nve20:1	10.2.2.2	10	static
38eb-d931-0301	20	Nve20:1	10.4.4.4	20	static

Total items: 2

4. 在控制器上为 FP2 转发器创建 BDIF 接口并配置 IP 地址

表 1-23 FP2 配置

步骤	命令	说明
1	<pre><Controller-FP20> system-view [-Controller-FP20] interface vbdif 20:10</pre>	创建 BDIF10 接口，并进入 BDIF10 接口视图。
2	<pre>[*Controller-FP20-Vbdif20:10] ip address 192.168.10.10 24 [*Controller-FP20-Vbdif20:10] quit</pre>	配置 BDIF10 接口 IP 地址。
3	<pre>[*Controller-FP20] interface vbdif 20:20</pre>	创建 BDIF20 接口，并进入 BDIF20 接口视图。
4	<pre>[*Controller-FP20-Vbdif20:20] ip address 192.168.20.10 24 [*Controller-FP20-Vbdif20:20] quit [*Controller-FP20] commit</pre>	配置 BDIF20 接口 IP 地址。

5. 在控制器上为 FP2 转发器配置静态 ARP 表

表 1-24 FP2 配置

步骤	命令	说明
1	<pre>[~Controller-FP20] arp static 192.168.10.1 38eb-d911-0301 vni 10 source-ip 10.3.3.3 peer-ip 10.2.2.2 [*Controller-FP20] arp static 192.168.20.1 38eb-d931-0301 vni 20 source-ip 10.3.3.3 peer-ip 10.4.4.4 [*Controller-FP20] commit [~Controller-FP20] quit</pre>	配置静态ARP表项，使得和指定IP地址的设备通信时只能使用指定的MAC地址，此时攻击报文无法修改此表项的IP地址和MAC地址的映射关系，从而保护了设备间的正常通信。

上述配置成功后，在控制器上FP2转发器视图下执行**display arp all**命令可查看所有ARP表项。

```
<Controller-FP20> display arp all
IP ADDRESS      MAC ADDRESS      EXPIRE (M)  TYPE      INTERFACE      VPN-INSTANCE
                VLAN/CEVLAN     PVC
-----
192.168.10.1    38eb-d911-0301    S--         S--       VxLAN-Tunnel
192.168.20.1    38eb-d931-0301    S--         S--       VxLAN-Tunnel
192.168.10.10   38ba-8fab-6903    I -         I -       Vbdf20:10
192.168.20.10   38ba-8fab-6903    I -         I -       Vbdf20:20
-----
Total:4         Dynamic:0         Static:2     Interface:2
```

6. 在控制器上为转发器FP1、FP3使能ARP代答功能

表 1-25 FP1 和 FP3 配置

步骤	命令	说明
1	<pre><Controller-FP20> quit <Controller> switch fp 10 <Controller-FP10> system-view [~Controller-FP10] bridge-domain 10 [~Controller-FP10-bd10] arp l2-proxy enable [*Controller-FP10-bd10] quit [*Controller-FP10] commit [~Controller-FP10] quit <Controller-FP10> quit</pre>	在控制器上为转发器FP1使能ARP代答功能。
2	<pre><Controller> switch fp 30 <Controller-FP30> system-view [~Controller-FP30] bridge-domain 20 [~Controller-FP30-bd20] arp l2-proxy enable [*Controller-FP30-bd20] quit [*Controller-FP30] commit [~Controller-FP30] quit <Controller-FP30> quit</pre>	在控制器上为转发器FP3使能ARP代答功能。

步骤5 检查配置结果

在VLAN10中的VM1上配置缺省网关为BDIF10接口的IP地址192.168.10.10/24。

在VLAN20中的VM1上配置缺省网关为BDIF20接口的IP地址192.168.20.10/24。

配置完成后，不同网段的VLAN10和VLAN20中的VM能够相互ping通。

----结束

配置文件

- 控制器的配置文件

```
#
sysname Controller
#
sdn controller
openflow listening-ip 10.1.1.1
fp-id 10
  type huawei-default
  version default
  role default
openflow controller
  peer-address 10.2.2.2
fp-id 20
  type huawei-default
  version default
  role default
openflow controller
  peer-address 10.3.3.3
fp-id 30
  type huawei-default
  version default
  role default
openflow controller
  peer-address 10.4.4.4
#
sdn fp service
fp-id 10
fp-id 20
fp-id 30
#
interface GE1/0/1
  undo shutdown
  ip address 172.16.1.1 255.255.255.0
#
interface LoopBack1
  ip address 10.1.1.1 255.255.255.255
#
ospf 1
  area 0.0.0.0
    network 10.1.1.1 0.0.0.0
    network 172.16.1.0 0.0.0.255
#

# Running configuration for fp 10
switch fp 10
#
bridge-domain 10
  vxlan vni 10
  arp l2-proxy enable
#
interface vServiceIf10:1
  binding interface 10GE1/0/2
  vlan assign 10
#
interface vServiceIf10:1.1 mode l2
  encapsulation dot1q vid 10
  bridge-domain 10
#
```

```

interface Nve10:1
 source 10.2.2.2
 vni 10 head-end peer-list 10.3.3.3
#
mac-address static 0001-0001-0001 vServiceIf10:1.1 bridge-domain 10 vid 10
mac-address static 38eb-d921-0301 bridge-domain 10 Nve10:1 peer 10.3.3.3 vni 10
#

#Running configuration for fp 20
switch fp 20
#
bridge-domain 10
 vxlan vni 10
#
bridge-domain 20
 vxlan vni 20
#
interface Vbdif20:10
 ip address 192.168.10.10 255.255.255.0
#
interface Vbdif20:20
 ip address 192.168.20.10 255.255.255.0
#
interface Nve20:1
 source 10.3.3.3
 vni 10 head-end peer-list 10.2.2.2
 vni 20 head-end peer-list 10.4.4.4
#
arp static 192.168.10.1 38eb-d911-0301 vni 10 source-ip 10.3.3.3 peer-ip 10.2.2.2
arp static 192.168.20.1 38eb-d931-0301 vni 20 source-ip 10.3.3.3 peer-ip 10.4.4.4
#
mac-address static 38eb-d911-0301 bridge-domain 10 Nve20:1 peer 10.2.2.2 vni 10
mac-address static 38eb-d931-0301 bridge-domain 20 Nve20:1 peer 10.4.4.4 vni 20
#

# Running configuration for fp 30
switch fp 30
#
bridge-domain 20
 vxlan vni 20
 arp l2-proxy enable
#
interface vServiceIf30:1
 binding interface 10GE1/0/2
 vlan assign 20
#
interface vServiceIf30:1.1 mode l2
 encapsulation dot1q vid 20
 bridge-domain 20
#
interface Nve30:1
 source 10.4.4.4
 vni 20 head-end peer-list 10.3.3.3
#
mac-address static 38eb-d921-0302 bridge-domain 20 Nve30:1 peer 10.3.3.3 vni 20
mac-address static 0002-0002-0002 vServiceIf30:1.1 bridge-domain 20 vid 20
#
return

```

- FPI 的配置文件

```

#
sysname FP1
#
vlan batch 10 100 300
#
sdn agent
 controller-ip 10.1.1.1

```

```
openflow agent
  transport-address 10.2.2.2
#
interface Vlanif100
  ip address 192.168.1.1 255.255.255.0
#
interface Vlanif300
  ip address 10.10.1.1 255.255.255.0
#
interface 10GE 1/0/1
  port link-type trunk
  port trunk allow-pass vlan 100
#
interface 10GE 1/0/2
  port link-type trunk
  port trunk allow-pass vlan 10
#
interface 10GE 1/0/3
  port link-type trunk
  port trunk allow-pass vlan 300
#
interface LoopBack1
  ip address 10.2.2.2 255.255.255.255
#
ospf 1
  area 0.0.0.0
  network 10.2.2.2 0.0.0.0
  network 10.10.1.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
#
return
```

● FP2的配置文件

```
#
sysname FP2
#
vlan batch 100 200 300
#
sdn agent
  controller-ip 10.1.1.1
  openflow agent
    transport-address 10.3.3.3
#
interface Vlanif100
  ip address 192.168.1.2 255.255.255.0
#
interface Vlanif200
  ip address 192.168.2.1 255.255.255.0
#
interface Vlanif300
  ip address 10.20.1.1 255.255.255.0
#
interface 10GE 1/0/1
  port link-type trunk
  port trunk allow-pass vlan 100
#
interface 10GE 1/0/2
  port link-type trunk
  port trunk allow-pass vlan 200
#
interface 10GE 1/0/3
  port link-type trunk
  port trunk allow-pass vlan 300
#
interface LoopBack1
  ip address 10.3.3.3 255.255.255.255
#
```

```
ospf 1
 area 0.0.0.0
  network 10.3.3.3 0.0.0.0
  network 10.20.1.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
#
return
```

- FP3的配置文件

```
#
sysname FP3
#
vlan batch 20 200 300
#
sdn agent
 controller-ip 10.1.1.1
 openflow agent
  transport-address 10.4.4.4
#
interface Vlanif200
 ip address 192.168.2.2 255.255.255.0
#
interface Vlanif300
 ip address 10.30.1.2 255.255.255.0
#
interface 10GE 1/0/1
 port link-type trunk
 port trunk allow-pass vlan 200
#
interface 10GE 1/0/2
 port link-type trunk
 port trunk allow-pass vlan 20
#
interface 10GE 1/0/3
 port link-type trunk
 port trunk allow-pass vlan 300
#
interface LoopBack1
 ip address 10.4.4.4 255.255.255.255
#
ospf 1
 area 0.0.0.0
  network 10.4.4.4 0.0.0.0
  network 10.30.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
#
return
```

1.8.3 配置集中式多活网关示例（单机方式）

组网需求

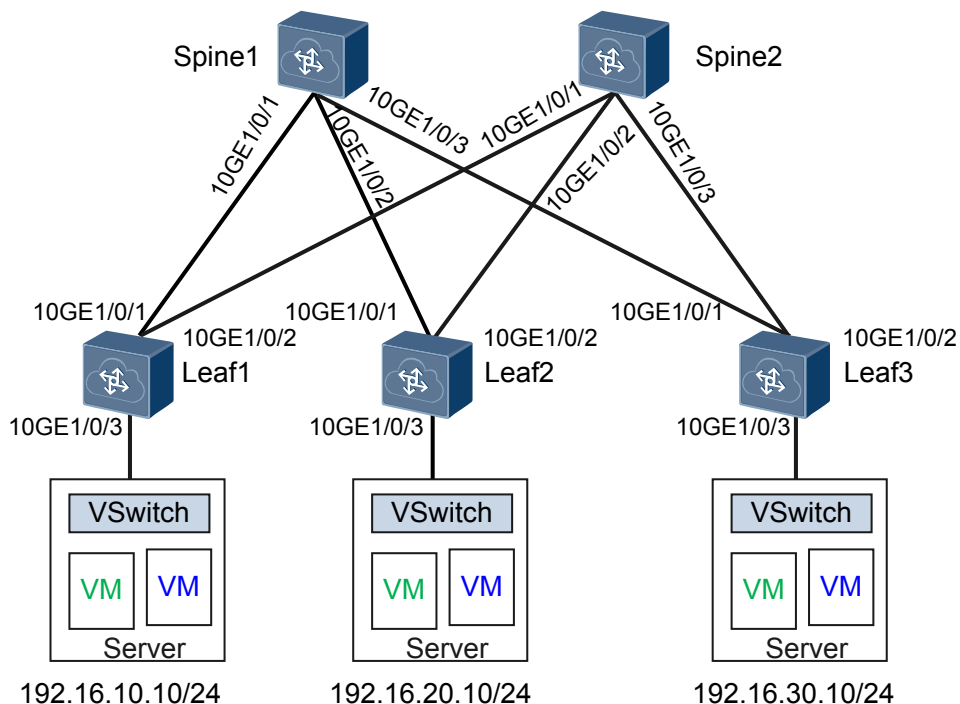
本示例从V100R005C10版本开始支持。

如图1-53所示，某企业数据中心内部网络为“Spine-Leaf”两层结构：

- Spine1和Spine2为基础承载网络中的骨干节点，位于汇聚层；
- Leaf1~Leaf3为基础承载网络中的叶子节点，位于接入层；
- Leaf设备与Spine设备之间全连接，构成ECMP，保证网络的高可用性；Spine节点之间互不连接，Leaf节点之间也不互联。

该数据中心流量主要为虚拟机迁移等形成的南北向流量，因此，客户要求在有基础承载网络上构建一个大二层网络，并且要求网络中的Spine设备同时作为网关设备，可以对Leaf设备发送过来的报文进行负载分担。

图 1-53 配置集中式多活网关示例组网图



注意事项

为了成功实现不同网段的用户互通，用户的缺省网关地址必须是对应三层网关上的BDIF接口的IP地址。

配置思路

采用如下思路配置集中式多活网关：

1. 在Leaf1~Leaf3、Spine1~Spine2上配置OSPF，保证网络三层互通。
2. 在Leaf1~Leaf3、Spine1~Spine2上配置VXLAN隧道，实现在基础三层网络上创建虚拟大二层VXLAN网络。
3. 在Leaf1~Leaf3上配置业务接入点，区分出服务器的流量并转发至VXLAN网络。
4. 在Spine1~Spine2上配置VXLAN三层网关实现VXLAN与非VXLAN网络、不同VXLAN网段之间的通信。

数据准备

为完成此配置例，需准备如下的数据：

- 网络中设备互连的接口IP地址。
- 网络中使用的路由类型是OSPF（Open Shortest Path First）。
- VM所属的VLAN ID分别是VLAN10、VLAN20和VLAN30。

- 广播域BD ID分别是BD10、BD20和BD30。
- VXLAN网络标识VNI ID分别是VNI5000、VNI5001和VNI5002。

操作步骤

步骤1 配置路由协议

分别配置Leaf1~Leaf3、Spine1~Spine2上各接口的IP地址。配置OSPF时，注意需要发布32位Loopback接口地址。

配置Leaf1。Leaf2和Leaf3的配置与Leaf1类似，此处不再赘述。

```
<HUAWEI> system-view
[-HUAWEI] sysname Leaf1
[*HUAWEI] commit
[~Leaf1] interface loopback 1
[*Leaf1-LoopBack1] ip address 10.10.10.3 32
[*Leaf1-LoopBack1] quit
[*Leaf1] interface 10ge 1/0/1
[*Leaf1-10GE1/0/1] undo portswitch
[*Leaf1-10GE1/0/1] ip address 10.1.1.2 24
[*Leaf1-10GE1/0/1] quit
[*Leaf1] interface 10ge 1/0/2
[*Leaf1-10GE1/0/2] undo portswitch
[*Leaf1-10GE1/0/2] ip address 10.2.1.2 24
[*Leaf1-10GE1/0/2] quit
[*Leaf1] ospf
[*Leaf1-ospf-1] area 0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.10.10.3 0.0.0.0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[*Leaf1-ospf-1-area-0.0.0.0] network 10.2.1.0 0.0.0.255
[*Leaf1-ospf-1-area-0.0.0.0] quit
[*Leaf1-ospf-1] quit
[*Leaf1] commit
```

配置Spine1。Spine2的配置与Spine1类似，此处不再赘述。

```
<HUAWEI> system-view
[-HUAWEI] sysname Spine1
[*HUAWEI] commit
[~Spine1] interface loopback 1
[*Spine1-LoopBack1] ip address 10.10.10.1 32
[*Spine1-LoopBack1] quit
[*Spine1] interface loopback 2
[*Spine1-LoopBack2] ip address 10.10.10.10 32
[*Spine1-LoopBack2] quit
[*Spine1] interface 10ge 1/0/1
[*Spine1-10GE1/0/1] undo portswitch
[*Spine1-10GE1/0/1] ip address 10.1.1.1 24
[*Spine1-10GE1/0/1] quit
[*Spine1] interface 10ge 1/0/2
[*Spine1-10GE1/0/2] undo portswitch
[*Spine1-10GE1/0/2] ip address 10.3.1.1 24
[*Spine1-10GE1/0/2] quit
[*Spine1] interface 10ge 1/0/3
[*Spine1-10GE1/0/2] undo portswitch
[*Spine1-10GE1/0/2] ip address 10.5.1.1 24
[*Spine1-10GE1/0/2] quit
[*Spine1] ospf
[*Spine1-ospf-1] area 0
[*Spine1-ospf-1-area-0.0.0.0] network 10.10.10.1 0.0.0.0
[*Spine1-ospf-1-area-0.0.0.0] network 10.10.10.10 0.0.0.0
[*Spine1-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] network 10.3.1.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] network 10.5.1.0 0.0.0.255
```

```
[*Spine1-ospf-1-area-0.0.0.0] quit
[*Spine1-ospf-1] quit
[*Spine1] commit
```

配置完成后，设备之间应能建立OSPF邻居关系，执行**display ospf peer**命令可以看到邻居状态为Full。执行**display ip routing-table**命令可以看到互相之间都可以学习到对方的Loopback路由。

步骤2 配置隧道模式

配置Spine1。Spine2、Leaf1、Leaf2和Leaf3的配置与Spine1类似，这里不再赘述。

```
[~Spine1] ip tunnel mode vxlan
[*Spine1] commit
```

说明

缺省情况下，隧道模式为VXLAN，无需配置此任务。当用户在使用GRE隧道后，需切换至VXLAN时，请在设备上执行此任务。此命令功能需要保存配置并重启设备才能生效，您可以选择立即重启或完成所有配置后再重启。

步骤3 在Leaf1~Leaf3、Spine1~Spine2上配置VXLAN隧道，组建VXLAN网络

配置Leaf1。Leaf2和Leaf3的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] bridge-domain 10
[*Leaf1-bd10] vxlan vni 5000
[*Leaf1-bd10] quit
[*Leaf1] interface nve 1
[*Leaf1-Nve1] source 10.10.10.3
[*Leaf1-Nve1] vni 5000 head-end peer-list 10.10.10.1
[*Leaf1-Nve1] quit
[*Leaf1] commit
```

配置Spine1。Spine2的配置与Spine1类似，此处不再赘述。

```
[~Spine1] bridge-domain 10
[*Spine1-bd10] vxlan vni 5000
[*Spine1-bd10] quit
[*Spine1] bridge-domain 20
[*Spine1-bd20] vxlan vni 5001
[*Spine1-bd20] quit
[*Spine1] bridge-domain 30
[*Spine1-bd30] vxlan vni 5002
[*Spine1-bd30] quit
[*Spine1] interface nve 1
[*Spine1-Nve1] source 10.10.10.1
[*Spine1-Nve1] vni 5000 head-end peer-list 10.10.10.3
[*Spine1-Nve1] vni 5001 head-end peer-list 10.10.10.4
[*Spine1-Nve1] vni 5002 head-end peer-list 10.10.10.5
[*Spine1-Nve1] quit
[*Spine1] commit
```

上述配置成功后，在Leaf1~Leaf3、Spine1~Spine2上执行**display vxlan vni**命令可查看到VNI的状态是**up**；执行**display vxlan tunnel**命令可查看到VXLAN隧道的信息。以Spine1显示为例。

```
[~Spine1] display vxlan vni
Number of vxlan vni: 3
VNI          BD-ID          State
-----
5000         10             up
5001         20             up
5002         30             up
[~Spine1] display vxlan tunnel
Number of vxlan tunnel : 3
```

Tunnel ID	Source	Destination	State	Type
4026531842	10.10.10.1	10.10.10.3	up	static
4026531843	10.10.10.1	10.10.10.4	up	static
4026531844	10.10.10.1	10.10.10.5	up	static

步骤4 在Leaf1~Leaf3上配置业务接入点

配置Leaf1。Leaf2和Leaf3的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] vlan batch 10
[*Leaf1] interface 10ge 1/0/3.1 mode 12
[*Leaf1-10GE1/0/3.1] encapsulation dot1q vid 10
[*Leaf1-10GE1/0/3.1] bridge-domain 10
[*Leaf1-10GE1/0/3.1] quit
[*Leaf1] commit
```

步骤5 在Spine1~Spine2上配置VXLAN三层网关

配置Spine1。Spine2的配置与Spine1类似，此处不再赘述。

```
[~Spine1] interface vbdif 10
[*Spine1-Vbdif10] ip address 192.168.10.1 24
[*Spine1-Vbdif10] mac-address 0000-5e00-0101
[*Spine1-Vbdif10] quit
[*Spine1] interface vbdif 20
[*Spine1-Vbdif20] ip address 192.168.20.1 24
[*Spine1-Vbdif20] mac-address 0000-5e00-0102
[*Spine1-Vbdif20] quit
[*Spine1] interface vbdif 30
[*Spine1-Vbdif30] ip address 192.168.30.1 24
[*Spine1-Vbdif30] mac-address 0000-5e00-0103
[*Spine1-Vbdif30] quit
[*Spine1] commit
```

说明

由于Spine1和Spine2作为多活网关设备，请确保这两台设备上配置的NVE接口的IP地址、BDIF接口的IP地址和MAC地址相同。

步骤6 在Spine1~Spine2上配置多活网关

配置Spine1。Spine2的配置与Spine1类似，此处不再赘述。

```
[~Spine1] dfs-group 1
[*Spine1-dfs-group-1] source ip 10.10.10.10
[*Spine1-dfs-group-1] active-active-gateway
[*Spine1-dfs-group-1-active-active-gateway] peer 10.10.10.20
[*Spine1-dfs-group-1-active-active-gateway] quit
[*Spine1-dfs-group-1] quit
[*Spine1] commit
```

步骤7 检查配置结果

上述配置成功后，在Spine1、Spine2上执行命令**display dfs-group 1 active-active-gateway**，可以查看到DFS Group多活网关的信息。以Spine1显示为例。

```
[~Spine1] display dfs-group 1 active-active-gateway
A:Active      I:Inactive
-----
Peer          System name      State      Duration
10.10.10.20   Spine2           A          0:0:8
```

----**结束**

配置文件

- Leaf1 的配置文件

```
#
sysname Leaf1
#
vlan batch 10
#
bridge-domain 10
  vxlan vni 5000
#
interface 10GE1/0/1
  undo portswitch
  ip address 10.1.1.2 255.255.255.0
#
interface 10GE1/0/2
  undo portswitch
  ip address 10.2.1.2 255.255.255.0
#
interface 10GE1/0/3.1 mode 12
  encapsulation dot1q vid 10
  bridge-domain 10
#
interface LoopBack1
  ip address 10.10.10.3 255.255.255.255
#
interface Nve1
  source 10.10.10.3
  vni 5000 head-end peer-list 10.10.10.1
#
ospf 1
  area 0.0.0.0
    network 10.1.1.0 0.0.0.255
    network 10.2.1.0 0.0.0.255
    network 10.10.10.3 0.0.0.0
#
return
```

- Leaf2 的配置文件

```
#
sysname Leaf2
#
vlan batch 20
#
bridge-domain 20
  vxlan vni 5001
#
interface 10GE1/0/1
  undo portswitch
  ip address 10.3.1.2 255.255.255.0
#
interface 10GE1/0/2
  undo portswitch
  ip address 10.4.1.2 255.255.255.0
#
interface 10GE1/0/3.1 mode 12
  encapsulation dot1q vid 20
  bridge-domain 20
#
interface LoopBack1
  ip address 10.10.10.4 255.255.255.255
#
interface Nve1
  source 10.10.10.4
  vni 5000 head-end peer-list 10.10.10.1
#
ospf 1
```

```
area 0.0.0.0
 network 10.3.1.0 0.0.0.255
 network 10.4.1.0 0.0.0.255
 network 10.10.10.4 0.0.0.0
#
return
```

● Leaf3 的配置文件

```
#
sysname Leaf3
#
vlan batch 30
#
bridge-domain 30
 vxlan vni 5002
#
interface 10GE1/0/1
 undo portswitch
 ip address 10.5.1.2 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 10.6.1.2 255.255.255.0
#
interface 10GE1/0/3.1 mode l2
 encapsulation dot1q vid 30
 bridge-domain 30
#
interface LoopBack1
 ip address 10.10.10.5 255.255.255.255
#
interface Nve1
 source 10.10.10.5
 vni 5000 head-end peer-list 10.10.10.1
#
ospf 1
 area 0.0.0.0
 network 10.5.1.0 0.0.0.255
 network 10.6.1.0 0.0.0.255
 network 10.10.10.5 0.0.0.0
#
return
```

● Spine1 的配置文件

```
#
sysname Spine1
#
dfs-group 1
 source ip 10.10.10.10
#
 active-active-gateway
 peer 10.10.10.20
#
bridge-domain 10
 vxlan vni 5000
#
bridge-domain 20
 vxlan vni 5001
#
bridge-domain 30
 vxlan vni 5002
#
interface Vbdif10
 ip address 192.168.10.1 255.255.255.0
 mac-address 0000-5e00-0101
#
interface Vbdif20
```

```

    ip address 192.168.20.1 255.255.255.0
    mac-address 0000-5e00-0102
#
interface Vbdif30
    ip address 192.168.30.1 255.255.255.0
    mac-address 0000-5e00-0103
#
interface 10GE1/0/1
    undo portswitch
    ip address 10.1.1.1 255.255.255.0
#
interface 10GE1/0/2
    undo portswitch
    ip address 10.3.1.1 255.255.255.0
#
interface 10GE1/0/3
    undo portswitch
    ip address 10.5.1.1 255.255.255.0
#
interface LoopBack1
    ip address 10.10.10.1 255.255.255.255
#
interface LoopBack2
    ip address 10.10.10.10 255.255.255.255
#
interface Nve1
    source 10.10.10.1
    vni 5000 head-end peer-list 10.10.10.3
    vni 5001 head-end peer-list 10.10.10.4
    vni 5002 head-end peer-list 10.10.10.5
#
ospf 1
    area 0.0.0.0
        network 10.1.1.0 0.0.0.255
        network 10.3.1.0 0.0.0.255
        network 10.5.1.0 0.0.0.255
        network 10.10.10.1 0.0.0.0
        network 10.10.10.10 0.0.0.0
#
return

```

- Spine2的配置文件

```

#
sysname Spine2
#
dfs-group 1
    source ip 10.10.10.20
#
active-active-gateway
    peer 10.10.10.10
#
bridge-domain 10
    vxlan vni 5000
#
bridge-domain 20
    vxlan vni 5001
#
bridge-domain 30
    vxlan vni 5002
#
interface Vbdif10
    ip address 192.168.10.1 255.255.255.0
    mac-address 0000-5e00-0101
#
interface Vbdif20
    ip address 192.168.20.1 255.255.255.0
    mac-address 0000-5e00-0102

```

```
#
interface Vbdif30
 ip address 192.168.30.1 255.255.255.0
 mac-address 0000-5e00-0103
#
interface 10GE1/0/1
 undo portswitch
 ip address 10.2.1.1 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 10.4.1.1 255.255.255.0
#
interface 10GE1/0/3
 undo portswitch
 ip address 10.6.1.1 255.255.255.0
#
interface LoopBack1
 ip address 10.10.10.1 255.255.255.255
#
interface LoopBack2
 ip address 10.10.10.20 255.255.255.255
#
interface Nve1
 source 10.10.10.1
 vni 5000 head-end peer-list 10.10.10.3
 vni 5001 head-end peer-list 10.10.10.4
 vni 5002 head-end peer-list 10.10.10.5
#
ospf 1
 area 0.0.0.0
 network 10.2.1.0 0.0.0.255
 network 10.4.1.0 0.0.0.255
 network 10.6.1.0 0.0.0.255
 network 10.10.10.1 0.0.0.0
 network 10.10.10.20 0.0.0.0
#
return
```

1.8.4 配置 VXLAN 双活接入示例（单机方式）

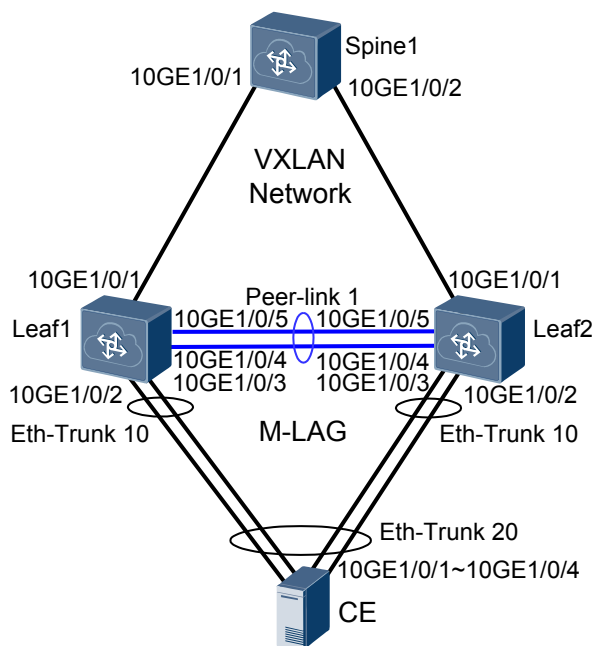
组网需求

本示例从V100R005C10版本开始支持。

如图1-54所示，服务器接入VXLAN网络时，要求：

- 为了保证可靠性，采用双归接入。当一条接入链路发生故障时，流量可以快速切换到另一条链路。
- 为了高效利用带宽，两条链路同时处于active状态，可实现使用负载分担的方式转发流量。

图 1-54 配置 VXLAN 双活接入组网图



配置思路

采用如下思路配置VXLAN双活接入：

1. 分别在Leaf1、Leaf2、Spine1上配置路由协议，实现保证网络三层互通。
2. 分别在Leaf1、Leaf2、Spine1上配置VXLAN基本功能，实现VXLAN网络互通。
3. 创建Eth-Trunk接口：
 - 分别在Leaf1、Leaf2上创建静态LACP模式Eth-Trunk，并加入成员口。
 - 在CE上创建动态LACP模式Eth-Trunk，并加入成员口。
4. 配置M-LAG：
 - 分别在Leaf1和Leaf2上配置DFS Group关联VXLAN。
 - 将Leaf1和Leaf2之间的链路配置为peer-link。
 - 分别在Leaf1和Leaf2上配置绑定DFS Group和用户侧Eth-Trunk接口。

数据准备

为完成此配置例，需准备如下的数据：

- 网络中设备互连的接口IP地址。
- 网络中使用的路由类型是OSPF（Open Shortest Path First）。
- VM所属的VLAN ID分别是VLAN10。
- 广播域BD ID分别是BD10。
- VXLAN网络标识VNI ID分别是VNI5010。

操作步骤

步骤1 配置路由协议

按图1-54分别配置Leaf1、Leaf2、Spine1各接口IP地址。配置OSPF时，注意需要发布转发器的32位Loopback接口地址。

配置Leaf1。Leaf2和Spine1的配置与Leaf1配置类似，这里不再赘述。

```
<HUAWEI> system-view
[-HUAWEI] sysname Leaf1
[*HUAWEI] commit
[-Leaf1] interface loopback 1
[*Leaf1-LoopBack1] ip address 10.2.2.2 32
[*Leaf1-LoopBack1] quit
[*Leaf1] interface loopback 2
[*Leaf1-LoopBack2] ip address 10.3.3.3 32
[*Leaf1-LoopBack2] quit
[*Leaf1] interface 10ge 1/0/1
[*Leaf1-10GE1/0/1] undo portswitch
[*Leaf1-10GE1/0/1] ip address 192.168.1.1 24
[*Leaf1-10GE1/0/1] quit
[*Leaf1] ospf
[*Leaf1-ospf-1] area 0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.2.2.2 0.0.0.0
[*Leaf1-ospf-1-area-0.0.0.0] network 10.3.3.3 0.0.0.0
[*Leaf1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Leaf1-ospf-1-area-0.0.0.0] quit
[*Leaf1-ospf-1] quit
[*Leaf1] commit
```

OSPF成功配置后，Leaf1、Leaf2、Spine1之间可通过OSPF协议发现对方的Loopback接口的IP地址，并能互相ping通。以Leaf1 ping Spine1的显示为例。

```
[-Leaf1] ping 10.1.1.1
PING 10.1.1.1: 56 data bytes, press CTRL_C to break
  Reply from 10.1.1.1: bytes=56 Sequence=1 ttl=254 time=5 ms
  Reply from 10.1.1.1: bytes=56 Sequence=2 ttl=254 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=3 ttl=254 time=2 ms
  Reply from 10.1.1.1: bytes=56 Sequence=4 ttl=254 time=3 ms
  Reply from 10.1.1.1: bytes=56 Sequence=5 ttl=254 time=3 ms

--- 10.1.1.1 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 2/3/5 ms
```

步骤2 配置隧道模式

配置Leaf1。Leaf2和Spine1的配置与Leaf1类似，这里不再赘述。

```
[-Leaf1] ip tunnel mode vxlan
[*Leaf1] commit
```

说明

缺省情况下，隧道模式为VXLAN，无需配置此任务。当用户在使用GRE隧道后，需切换至VXLAN时，请在设备上执行此任务。此命令功能需要保存配置并重启设备才能生效，您可以选择立即重启或完成所有配置后再重启。

步骤3 分别在Leaf1、Leaf2和Spine1上配置VXLAN隧道

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[-Leaf1] bridge-domain 10
[*Leaf1-bd10] vxlan vni 5010
[*Leaf1-bd10] quit
[*Leaf1] interface nve1
[*Leaf1-Nve1] source 10.2.2.2
[*Leaf1-Nve1] vni 5010 head-end peer-list 10.1.1.1
```

```
[*Leaf1-Nve1] quit
[*Leaf1] commit
```

配置Spine1。

```
[~Spine1] bridge-domain 10
[*Spine1-bd10] vxlan vni 5010
[*Spine1-bd10] quit
[*Spine1] interface nve1
[*Spine1-Nve1] source 10.1.1.1
[*Spine1-Nve1] vni 5010 head-end peer-list 10.2.2.2
[*Spine1-Nve1] quit
[*Spine1] commit
```

上述配置成功后，在Spine1上执行**display vxlan vni**命令可查看到VNI的状态是**up**；执行**display vxlan tunnel**命令可查看到VXLAN隧道的信息。

```
[~Spine1] display vxlan vni
Number of vxlan vni: 1
VNI          BD-ID          State
-----
5010         10             up
[~Spine1] display vxlan tunnel
Number of vxlan tunnel : 1
Tunnel ID   Source          Destination     State  Type
-----
4026531841  10.1.1.1       10.2.2.2       up     static
```

步骤4 创建Eth-Trunk接口，并将以太物理接口加入Eth-Trunk接口

服务器上行连接交换机的端口需要绑定在一个聚合链路中且链路聚合模式需要和交换机侧的聚合模式匹配。

在Leaf1上创建Eth-Trunk，配置为LACP模式并加入成员口。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface eth-trunk 1
[*Leaf1-Eth-Trunk1] mode lacp-static
[*Leaf1-Eth-Trunk1] trunkport 10ge 1/0/4 to 1/0/5
[*Leaf1-Eth-Trunk1] quit
[*Leaf1] interface eth-trunk 10
[*Leaf1-Eth-Trunk10] mode lacp-dynamic
[*Leaf1-Eth-Trunk10] trunkport 10ge 1/0/2 to 1/0/3
[*Leaf1-Eth-Trunk10] quit
[*Leaf1] commit
```

步骤5 分别在Leaf1和Leaf2上配置DFS Group

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] dfs-group 1
[*Leaf1-dfs-group-1] source ip 10.3.3.3
[*Leaf1-dfs-group-1] quit
[*Leaf1] commit
```

步骤6 将Leaf1和Leaf2之间的链路配置为peer-link

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface eth-trunk 1
[~Leaf1-Eth-Trunk1] undo stp enable
[*Leaf1-Eth-Trunk1] peer-link 1
[*Leaf1-Eth-Trunk1] quit
[*Leaf1] commit
```

步骤7 分别在Leaf1和Leaf2上配置绑定DFS和用户侧Eth-Trunk接口

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface eth-trunk 10
[~Leaf1-Eth-Trunk10] dfs-group 1 m-lag 1
[*Leaf1-Eth-Trunk10] lACP m-lag system-id 00e0-fc00-0000
[*Leaf1-Eth-Trunk10] quit
[*Leaf1] commit
```

步骤8 分别在Leaf1和Leaf2上配置VXLAN业务接入点

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] vlan batch 10
[*Leaf1] interface eth-trunk 10.1 mode 12
[*Leaf1-Eth-Trunk10.1] encapsulation dot1q vid 10
[*Leaf1-Eth-Trunk10.1] bridge-domain 10
[*Leaf1-Eth-Trunk10.1] quit
[*Leaf1] commit
```

步骤9 检查配置结果

执行命令**display dfs-group 1**，查看M-LAG的相关信息。

```
[~Leaf1] display dfs-group 1 m-lag
*                : Local node
Heart beat state : OK
Node 1 *
  Dfs-Group ID   : 1
  Priority        : 100
  Address        : ip address 10.3.3.3
  State          : Master
  Causation      : -
  System ID      : 0025-9e95-7c11
  SysName        : Leaf1
  Version        : V100R005C10
  Device Type    : CE12800
Node 2
  Dfs-Group ID   : 1
  Priority        : 100
  Address        : ip address 10.4.4.4
  State          : Backup
  Causation      : -
  System ID      : 0025-9e95-7c31
  SysName        : Leaf2
  Version        : V100R005C10
  Device Type    : CE12800
```

查看Leaf1上的M-LAG信息。

```
[~Leaf1] display dfs-group 1 node 1 m-lag brief
* - Local node

M-Lag ID   Interface   Port State   Status
   1       Eth-Trunk 10  Up          active(*)-active
```

查看Leaf2上的M-LAG信息。

```
[~Leaf2] display dfs-group 1 node 2 m-lag brief
* - Local node

M-Lag ID   Interface   Port State   Status
   1       Eth-Trunk 10  Up          active-active(*)
```

----结束

配置文件

- Leaf1 的配置文件

```
#
sysname Leaf1
#
vlan batch 10
#
dfs-group 1
source ip 10.3.3.3
#
bridge-domain 10
vxlan vni 5010
#
interface Eth-Trunk1
stp disable
mode lacp-dynamic
peer-link 1
#
interface Eth-Trunk10
mode lacp-dynamic
dfs-group 1 m-lag 1
lacp m-lag system-id 00e0-fc00-0000
#
interface Eth-Trunk10.1 mode 12
encapsulation dot1q vid 10
bridge-domain 10
#
interface 10GE1/0/1
undo portswitch
ip address 192.168.1.1 255.255.255.0
#
interface 10GE1/0/2
eth-trunk 10
#
interface 10GE1/0/3
eth-trunk 10
#
interface 10GE1/0/4
eth-trunk 1
#
interface 10GE1/0/5
eth-trunk 1
#
interface LoopBack1
ip address 10.2.2.2 255.255.255.255
#
interface LoopBack2
ip address 10.3.3.3 255.255.255.255
#
interface Nve1
source 10.2.2.2
vni 5010 head-end peer-list 10.1.1.1
#
ospf 1
area 0.0.0.0
network 10.2.2.2 0.0.0.0
network 10.3.3.3 0.0.0.0
network 192.168.1.0 0.0.0.255
#
return
```

- Leaf2 的配置文件

```
#
sysname Leaf2
#
vlan batch 10
```

```
#
dfs-group 1
  source ip 10.4.4.4
#
bridge-domain 10
  vxlan vni 5010
#
interface Eth-Trunk1
  stp disable
  mode lacp-dynamic
  peer-link 1
#
interface Eth-Trunk10
  mode lacp-dynamic
  dfs-group 1 m-lag 1
  lacp m-lag system-id 00e0-fc00-0000
#
interface Eth-Trunk10.1 mode 12
  encapsulation dot1q vid 10
  bridge-domain 10
#
interface 10GE1/0/1
  undo portswitch
  ip address 192.168.2.1 255.255.255.0
#
interface 10GE1/0/2
  eth-trunk 10
#
interface 10GE1/0/3
  eth-trunk 10
#
interface 10GE1/0/4
  eth-trunk 1
#
interface 10GE1/0/5
  eth-trunk 1
#
interface LoopBack1
  ip address 10.2.2.2 255.255.255.255
#
interface LoopBack2
  ip address 10.4.4.4 255.255.255.255
#
interface Nve1
  source 10.2.2.2
  vni 5010 head-end peer-list 10.1.1.1
#
ospf 1
  area 0.0.0.0
  network 10.2.2.2 0.0.0.0
  network 10.4.4.4 0.0.0.0
  network 192.168.2.0 0.0.0.255
#
return
```

- Spine1 的配置文件

```
#
sysname Spine1
#
bridge-domain 10
  vxlan vni 5010
#
interface 10GE1/0/1
  undo portswitch
  ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/2
```

```
undo portswitch
ip address 192.168.2.2 255.255.255.0
#
interface LoopBack1
ip address 10.1.1.1 255.255.255.255
#
interface Nve1
source 10.1.1.1
vni 5010 head-end peer-list 10.2.2.2
#
ospf 1
area 0.0.0.0
network 10.1.1.1 0.0.0.0
network 192.168.1.0 0.0.0.255
network 192.168.2.0 0.0.0.255
#
return
```

1.8.5 配置 VXLAN 分布式网关示例（单机方式）

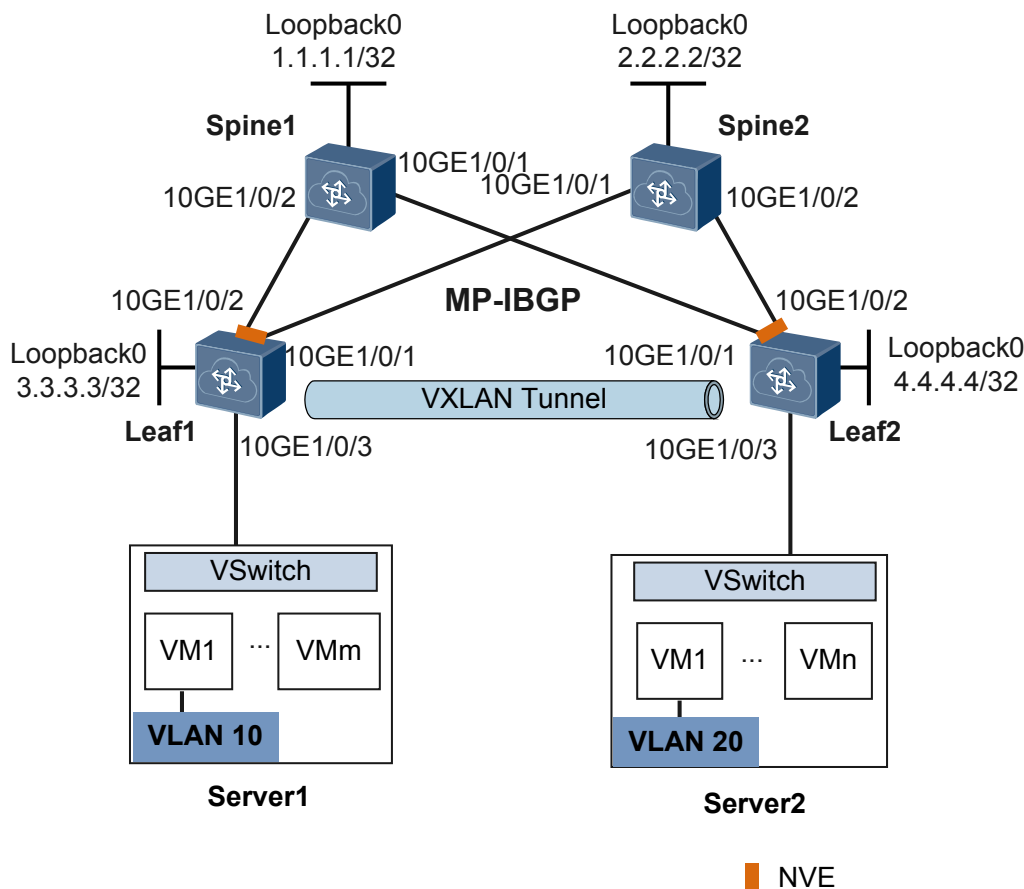
组网需求

本示例从V100R005C10版本开始支持。

VXLAN分布式网关可解决VXLAN集中式网关的转发路径不优化、三层网关ARP表项规格瓶颈问题。

如图1-55所示，某企业在不同的数据中心的都拥有自己的VM，服务器1上的VM1属于VLAN10，服务器2上的VM1属于VLAN20，且位于不同网段。现需要通过VXLAN分布式网关实现不同数据中心相同VM的互通。

图 1-55 配置 VXLAN 分布式网关组网图



配置思路

采用如下思路配置VXLAN分布式网关：

1. 分别在Spine1、Spine2、Leaf1和Leaf2上配置IGP路由协议，在此基础上部署BGP和BGP/MPLS IP VPN，保证网络三层互通。
2. 分别在Leaf1和Leaf2上配置VXLAN二层网关实现区分业务流量。
3. 分别在Leaf1和Leaf2上配置三层VXLAN隧道转发业务流量。
4. 在Leaf1、Leaf2上配置VXLAN三层网关，实现不同网段用户通过VXLAN三层网关互通。
5. 在Spine1、Spine2、Leaf1、Leaf2上配置BGP对IBGP邻居发布remote-nexthop属性，包括隧道地址、L3VPN VNI以及MAC地址等信息，达到设备三层VXLAN隧道互通的目的。

数据准备

为完成此配置例，需准备如下的数据：

- VM所属的VLAN ID分别是VLAN10和VLAN20。
- 网络中设备互连的接口IP地址。
- 网络中使用的IGP路由类型是OSPF。

- 在BGP和BGP/MPLS IP VPN的配置中，Spine1和Spine2为反射器，Leaf1和Leaf2为客户机。
- 广播域BD ID分别是BD10和BD20。
- VXLAN网络标识VNI ID分别是5000、5001、10000、20000。

操作步骤

步骤1 配置三层网络

1. 配置接口的IP地址和OSPF协议

按图1-55分别配置Spine1、Spine2、Leaf1和Leaf2各接口IP地址。配置OSPF时，注意需要发布转发器的32位Loopback接口地址。

配置Spine1。Spine2、Leaf1和Leaf2的配置与Spine1配置类似，这里不再赘述。

```
<HUAWEI> system-view
[~HUAWEI] sysname Spine1
[*HUAWEI] commit
[~Spine1] interface loopback 0
[*Spine1-LoopBack0] ip address 1.1.1.1 32
[*Spine1-LoopBack0] quit
[*Spine1] interface 10ge 1/0/1
[*Spine1-10GE1/0/1] undo portswitch
[*Spine1-10GE1/0/1] ip address 192.168.2.1 24
[*Spine1-10GE1/0/1] quit
[*Spine1] interface 10ge 1/0/2
[*Spine1-10GE1/0/2] undo portswitch
[*Spine1-10GE1/0/2] ip address 192.168.1.1 24
[*Spine1-10GE1/0/2] quit
[*Spine1] ospf
[*Spine1-ospf-1] area 0
[*Spine1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[*Spine1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] quit
[*Spine1-ospf-1] quit
[*Spine1] commit
```

OSPF成功配置后，设备之间可通过OSPF协议发现对方的Loopback接口的IP地址，并能互相ping通。以Spine1 ping Leaf2的显示为例。

```
[~Spine1] ping 4.4.4.4
PING 4.4.4.4: 56 data bytes, press CTRL_C to break
  Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=253 time=55 ms
  Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=253 time=3 ms
  Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=253 time=4 ms
  Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=253 time=3 ms
  Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=253 time=3 ms

--- 4.4.4.4 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 3/13/55 ms
```

2. 部署BGP和BGP/MPLS IP VPN

在BGP和BGP/MPLS IP VPN的配置中，Spine1和Spine2为反射器，Leaf1和Leaf2为客户机。

配置Spine1。Spine2的配置与Spine1类似，这里不再赘述。


```
[~Spine1] bgp 100
[*Spine1-bgp] router-id 1.1.1.1
[*Spine1-bgp] peer 3.3.3.3 as-number 100
[*Spine1-bgp] peer 3.3.3.3 connect-interface loopback0
[*Spine1-bgp] peer 3.3.3.3 reflect-client
[*Spine1-bgp] peer 4.4.4.4 as-number 100
[*Spine1-bgp] peer 4.4.4.4 connect-interface loopback0
[*Spine1-bgp] peer 4.4.4.4 reflect-client
[*Spine1-bgp] ipv4-family vpnv4
[*Spine1-bgp-af-vpnv4] undo policy vpn-target
[*Spine1-bgp-af-vpnv4] peer 3.3.3.3 enable
[*Spine1-bgp-af-vpnv4] peer 3.3.3.3 reflect-client
[*Spine1-bgp-af-vpnv4] peer 4.4.4.4 enable
[*Spine1-bgp-af-vpnv4] peer 4.4.4.4 reflect-client
[*Spine1-bgp-af-vpnv4] quit
[*Spine1-bgp] quit
[*Spine1] commit
```

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] ip vpn-instance vrf1
[*Leaf1-vpn-instance-vrf1] ipv4-family
[*Leaf1-vpn-instance-vrf1-af-ipv4] route-distinguisher 100:1
[*Leaf1-vpn-instance-vrf1-af-ipv4] vpn-target 100:1 export-extcommunity
[*Leaf1-vpn-instance-vrf1-af-ipv4] vpn-target 100:1 import-extcommunity
[*Leaf1-vpn-instance-vrf1-af-ipv4] quit
[*Leaf1-vpn-instance-vrf1] vxlan vni 10000
[*Leaf1-vpn-instance-vrf1] quit
[*Leaf1] bgp 100
[*Leaf1-bgp] router-id 3.3.3.3
[*Leaf1-bgp] peer 1.1.1.1 as-number 100
[*Leaf1-bgp] peer 1.1.1.1 connect-interface loopback0
[*Leaf1-bgp] peer 2.2.2.2 as-number 100
[*Leaf1-bgp] peer 2.2.2.2 connect-interface loopback0
[*Leaf1-bgp] ipv4-family vpnv4
[*Leaf1-bgp-af-vpnv4] peer 1.1.1.1 enable
[*Leaf1-bgp-af-vpnv4] peer 2.2.2.2 enable
[*Leaf1-bgp-af-vpnv4] quit
[*Leaf1-bgp] ipv4-family vpn-instance vrf1
[*Leaf1-bgp-vrf1] import-route direct
[*Leaf1-bgp-vrf1] quit
[*Leaf1-bgp] quit
[*Leaf1] commit
```

步骤2 分别在Leaf1、Leaf2上配置VXLAN二层网关

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] vlan batch 10
[*Leaf1] bridge-domain 10
[*Leaf1-bd10] vxlan vni 5000
[*Leaf1-bd10] quit
[*Leaf1] interface 10ge 1/0/3.1 mode 12
[*Leaf1-10GE1/0/3.1] encapsulation dot1q vid 10
[*Leaf1-10GE1/0/3.1] bridge-domain 10
[*Leaf1-10GE1/0/3.1] quit
[*Leaf1] commit
```

步骤3 分别在Leaf1、Leaf2上配置三层VXLAN隧道

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface Nve1
[*Leaf1-Nve1] mode 13
[*Leaf1-Nve1] source 3.3.3.3
[*Leaf1-Nve1] quit
[*Leaf1] commit
```

步骤4 在Leaf1、Leaf2上配置VXLAN三层网关

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。要注意Leaf2的BDIF接口的IP地址要与Leaf1的属于不同网段。

```
[~Leaf1] interface vbdif 10
[*Leaf1-Vbdif10] arp distribute-gateway enable
[*Leaf1-Vbdif10] ip binding vpn-instance vrf1
[*Leaf1-Vbdif10] ip address 10.1.1.1 255.255.255.0
[*Leaf1-Vbdif10] arp direct-route enable
[*Leaf1-Vbdif10] quit
[*Leaf1] commit
```

步骤5 在Spine1、Spine2、Leaf1、Leaf2上配置BGP对IBGP邻居发布remote-nexthop属性

配置Spine1。Spine2、Leaf1和Leaf2的配置与Spine1类似，此处不再赘述。

```
[~Spine1] bgp 100
[~Spine1-bgp] ipv4-family vpnv4
[~Spine1-bgp-af-vpnv4] peer 3.3.3.3 advertise remote-nexthop
[*Spine1-bgp-af-vpnv4] peer 4.4.4.4 advertise remote-nexthop
[~Spine1-bgp-af-vpnv4] quit
[*Spine1-bgp] quit
[*Spine1] commit
```

步骤6 检查配置结果

上述配置成功后，在Leaf1、Leaf2上执行**display vxlan tunnel**命令可查看到VXLAN隧道的信息。以Leaf1显示为例。

```
[~Leaf1] display vxlan tunnel
Number of vxlan tunnel : 1
Tunnel ID   Source           Destination      State  Type
-----
4026531841  3.3.3.3          4.4.4.4          up     dynamic
```

配置完成后，不同网段中的VM1可以相互通信。

---结束

配置文件

- Spine1的配置文件

```
#
sysname Spine1
#
interface 10GE1/0/1
 undo portswitch
 ip address 192.168.2.1 255.255.255.0
#
interface 10GE1/0/2
 undo portswitch
 ip address 192.168.1.1 255.255.255.0
#
interface LoopBack0
 ip address 1.1.1.1 255.255.255.255
#
bgp 100
 router-id 1.1.1.1
 peer 3.3.3.3 as-number 100
 peer 3.3.3.3 connect-interface LoopBack0
 peer 4.4.4.4 as-number 100
 peer 4.4.4.4 connect-interface LoopBack0
#
ipv4-family unicast
```

```
peer 3.3.3.3 enable
peer 3.3.3.3 reflect-client
peer 4.4.4.4 enable
peer 4.4.4.4 reflect-client
#
ipv4-family vpv4
undo policy vpn-target
peer 3.3.3.3 enable
peer 3.3.3.3 reflect-client
peer 3.3.3.3 advertise remote-nexthop
peer 4.4.4.4 enable
peer 4.4.4.4 reflect-client
peer 4.4.4.4 advertise remote-nexthop
#
ospf 1
area 0.0.0.0
network 1.1.1.1 0.0.0.0
network 192.168.1.0 0.0.0.255
network 192.168.2.0 0.0.0.255
#
return
```

● Spine2的配置文件

```
#
sysname Spine2
#
interface 10GE1/0/1
undo portswitch
ip address 192.168.3.1 255.255.255.0
#
interface 10GE1/0/2
undo portswitch
ip address 192.168.4.1 255.255.255.0
#
interface LoopBack0
ip address 2.2.2.2 255.255.255.255
#
bgp 100
router-id 2.2.2.2
peer 3.3.3.3 as-number 100
peer 3.3.3.3 connect-interface LoopBack0
peer 4.4.4.4 as-number 100
peer 4.4.4.4 connect-interface LoopBack0
#
ipv4-family unicast
peer 3.3.3.3 enable
peer 3.3.3.3 reflect-client
peer 4.4.4.4 enable
peer 4.4.4.4 reflect-client
#
ipv4-family vpv4
undo policy vpn-target
peer 3.3.3.3 enable
peer 3.3.3.3 reflect-client
peer 3.3.3.3 advertise remote-nexthop
peer 4.4.4.4 enable
peer 4.4.4.4 reflect-client
peer 4.4.4.4 advertise remote-nexthop
#
ospf 1
area 0.0.0.0
network 2.2.2.2 0.0.0.0
network 192.168.3.0 0.0.0.255
network 192.168.4.0 0.0.0.255
#
return
```

● Leaf1 的配置文件

```
#
sysname Leaf1
#
vlan batch 10
#
ip vpn-instance vrf1
  ipv4-family
    route-distinguisher 100:1
    vpn-target 100:1 export-extcommunity
    vpn-target 100:1 import-extcommunity
  vxlan vni 10000
#
bridge-domain 10
  vxlan vni 5000
#
interface Vbdif10
  ip binding vpn-instance vrf1
  ip address 10.1.1.1 255.255.255.0
  arp distribute-gateway enable
  arp direct-route enable
#
interface 10GE1/0/1
  undo portswitch
  ip address 192.168.3.2 255.255.255.0
#
interface 10GE1/0/2
  undo portswitch
  ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/3.1 mode l2
  encapsulation dot1q vid 10
  bridge-domain 10
#
interface LoopBack0
  ip address 3.3.3.3 255.255.255.255
#
interface Nve1
  mode l3
  source 3.3.3.3
#
bgp 100
  router-id 3.3.3.3
  peer 1.1.1.1 as-number 100
  peer 1.1.1.1 connect-interface LoopBack0
  peer 2.2.2.2 as-number 100
  peer 2.2.2.2 connect-interface LoopBack0
#
  ipv4-family unicast
    peer 1.1.1.1 enable
    peer 2.2.2.2 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 1.1.1.1 enable
    peer 1.1.1.1 advertise remote-nexthop
    peer 2.2.2.2 enable
    peer 2.2.2.2 advertise remote-nexthop
#
  ipv4-family vpn-instance vrf1
    import-route direct
#
ospf 1
  area 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 192.168.1.0 0.0.0.255
```

```
network 192.168.3.0 0.0.0.255
#
return
```

● Leaf2的配置文件

```
#
sysname Leaf2
#
vlan batch 20
#
ip vpn-instance vrf1
ipv4-family
route-distinguisher 100:1
vpn-target 100:1 export-extcommunity
vpn-target 100:1 import-extcommunity
vxlan vni 20000
#
bridge-domain 20
vxlan vni 5001
#
interface Vbdif20
ip binding vpn-instance vrf1
ip address 20.1.1.1 255.255.255.0
arp distribute-gateway enable
arp direct-route enable
#
interface 10GE1/0/1
undo portswitch
ip address 192.168.2.2 255.255.255.0
#
interface 10GE1/0/2
undo portswitch
ip address 192.168.4.2 255.255.255.0
#
interface 10GE1/0/3.1 mode 12
encapsulation dot1q vid 20
bridge-domain 20
#
interface LoopBack0
ip address 4.4.4.4 255.255.255.255
#
interface Nve2
mode 13
source 4.4.4.4
#
bgp 100
router-id 4.4.4.4
peer 1.1.1.1 as-number 100
peer 1.1.1.1 connect-interface LoopBack0
peer 2.2.2.2 as-number 100
peer 2.2.2.2 connect-interface LoopBack0
#
ipv4-family unicast
peer 1.1.1.1 enable
peer 2.2.2.2 enable
#
ipv4-family vpnv4
policy vpn-target
peer 1.1.1.1 enable
peer 1.1.1.1 advertise remote-nexthop
peer 2.2.2.2 enable
peer 2.2.2.2 advertise remote-nexthop
#
ipv4-family vpn-instance vrf1
import-route direct
#
ospf 1
```

```

area 0.0.0.0
 network 4.4.4.4 0.0.0.0
 network 192.168.2.0 0.0.0.255
 network 192.168.4.0 0.0.0.255
#
return
    
```

1.8.6 配置 VXLAN 分布式网关+双活接入综合示例（单机方式）

组网需求

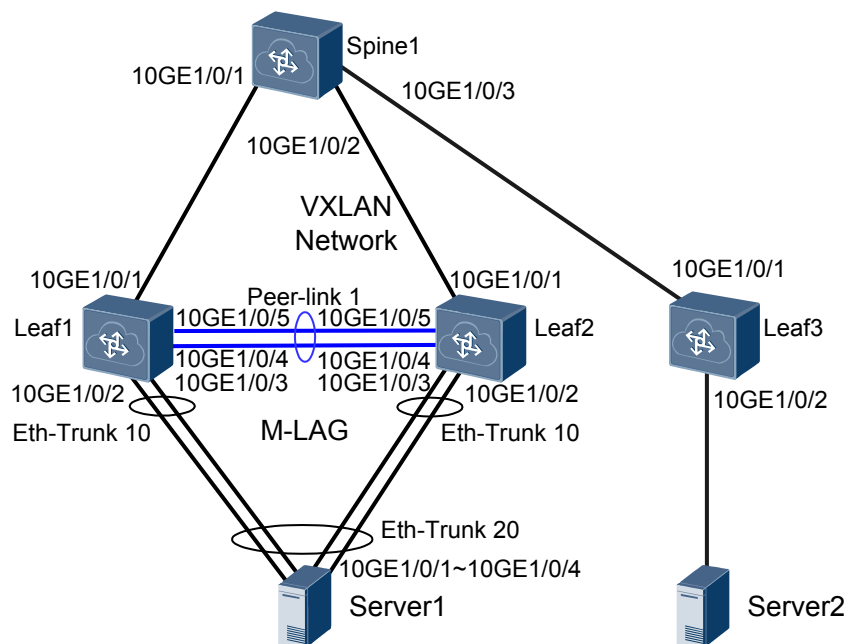
本示例从V100R005C10版本开始支持。

如图1-56所示，某企业数据中心内部网络为“Spine-Leaf”两层结构：

- Spine1为基础承载网络中的骨干节点，位于汇聚层；
- Leaf1~Leaf3为基础承载网络中的叶子节点，位于接入层；

为了保证可靠性，某企业将部分重要的服务器采用双归接入的方式接入网络中，使得当一条接入链路发生故障时，流量可以快速切换到另一条链路。同时，企业管理员希望能够优化数据的转发路径，例如，同一个Leaf下服务器或VM之间的通信流不需要通过Spine设备。通过VXLAN双活接入配合分布式网关功能，可以实现此需求。

图 1-56 配置 VXLAN 分布式网关+双活接入综合示例组网图



配置思路

采用如下思路配置VXLAN分布式网关：

1. 分别在Spine1、Leaf1、Leaf2和Leaf3上配置IGP路由协议，在此基础上部署BGP和BGP/MPLS IP VPN，保证网络三层互通。
2. 分别在Leaf1、Leaf2和Leaf3上配置业务接入点实现区分业务流量。
3. 分别在Leaf1、Leaf2和Leaf3上配置L2和L3 VXLAN隧道转发业务流量。

4. 在Leaf1、Leaf2和Leaf3配置VXLAN三层网关，实现不同网段用户通过VXLAN三层网关互通。
5. 在Spine1、Leaf1、Leaf2和Leaf3上配置BGP对IBGP邻居发布remote-nexthop属性，包括隧道地址、L3VPN VNI以及MAC地址等信息，达到设备三层VXLAN隧道互通的目的。

数据准备

为完成此配置例，需准备如下的数据：

- VM所属的VLAN ID分别是VLAN2、VLAN3和VLAN4。
- 网络中设备互连的接口IP地址。
- 网络中使用的IGP路由类型是OSPF。
- 在BGP和BGP/MPLS IP VPN的配置中，Spine1为反射器，Leaf1、Leaf2和Leaf3为客户机。
- 广播域BD ID分别是BD1、BD2和BD3。
- VXLAN网络标识VNI ID分别是：5000、5001、5002、9998、9999。

操作步骤

步骤1 配置三层网络

1. 配置接口的IP地址和OSPF协议

按图1-56分别配置Spine1、Leaf1、Leaf2和Leaf3各接口IP地址。配置OSPF时，注意需要发布转发器的32位Loopback接口地址。

配置Spine1。Leaf1、Leaf2和Leaf3的配置与Spine1配置类似，这里不再赘述。

```
<HUAWEI> system-view
[~HUAWEI] sysname Spine1
[*HUAWEI] commit
[~Spine1] interface loopback 0
[*Spine1-LoopBack0] ip address 1.1.1.1 32
[*Spine1-LoopBack0] quit
[*Spine1] interface 10ge 1/0/1
[*Spine1-10GE1/0/1] undo portswitch
[*Spine1-10GE1/0/1] ip address 192.168.1.1 24
[*Spine1-10GE1/0/1] quit
[*Spine1] interface 10ge 1/0/2
[*Spine1-10GE1/0/2] undo portswitch
[*Spine1-10GE1/0/2] ip address 192.168.2.1 24
[*Spine1-10GE1/0/2] quit
[*Spine1] interface 10ge 1/0/3
[*Spine1-10GE1/0/3] undo portswitch
[*Spine1-10GE1/0/3] ip address 192.168.3.1 24
[*Spine1-10GE1/0/3] quit
[*Spine1] ospf
[*Spine1-ospf-1] area 0
[*Spine1-ospf-1-area-0.0.0.0] network 1.1.1.1 0.0.0.0
[*Spine1-ospf-1-area-0.0.0.0] network 192.168.1.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] network 192.168.2.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] network 192.168.3.0 0.0.0.255
[*Spine1-ospf-1-area-0.0.0.0] quit
[*Spine1-ospf-1] quit
[*Spine1] commit
```

OSPF成功配置后，设备之间可通过OSPF协议发现对方的Loopback接口的IP地址，并能互相ping通。以Spine1 ping Leaf3的显示为例。

```
[~Spine1] ping 4.4.4.4
PING 4.4.4.4: 56 data bytes, press CTRL_C to break
  Reply from 4.4.4.4: bytes=56 Sequence=1 ttl=253 time=55 ms
  Reply from 4.4.4.4: bytes=56 Sequence=2 ttl=253 time=3 ms
  Reply from 4.4.4.4: bytes=56 Sequence=3 ttl=253 time=4 ms
  Reply from 4.4.4.4: bytes=56 Sequence=4 ttl=253 time=3 ms
  Reply from 4.4.4.4: bytes=56 Sequence=5 ttl=253 time=3 ms

--- 4.4.4.4 ping statistics ---
  5 packet(s) transmitted
  5 packet(s) received
  0.00% packet loss
  round-trip min/avg/max = 3/13/55 ms
```

2. 部署BGP和BGP/MPLS IP VPN

在BGP和BGP/MPLS IP VPN的配置中，Spine1为反射器，Leaf1、Leaf2和Leaf3为客户机。

配置Spine1。

```
[~Spine1] bgp 100
[*Spine1-bgp] router-id 1.1.1.1
[*Spine1-bgp] peer 2.2.2.2 as-number 100
[*Spine1-bgp] peer 2.2.2.2 connect-interface loopback0
[*Spine1-bgp] peer 2.2.2.2 reflect-client
[*Spine1-bgp] peer 3.3.3.3 as-number 100
[*Spine1-bgp] peer 3.3.3.3 connect-interface loopback0
[*Spine1-bgp] peer 3.3.3.3 reflect-client
[*Spine1-bgp] peer 4.4.4.4 as-number 100
[*Spine1-bgp] peer 4.4.4.4 connect-interface loopback0
[*Spine1-bgp] peer 4.4.4.4 reflect-client
[*Spine1-bgp] ipv4-family vpnv4
[*Spine1-bgp-af-vpnv4] undo policy vpn-target
[*Spine1-bgp-af-vpnv4] peer 2.2.2.2 enable
[*Spine1-bgp-af-vpnv4] peer 2.2.2.2 reflect-client
[*Spine1-bgp-af-vpnv4] peer 3.3.3.3 enable
[*Spine1-bgp-af-vpnv4] peer 3.3.3.3 reflect-client
[*Spine1-bgp-af-vpnv4] peer 4.4.4.4 enable
[*Spine1-bgp-af-vpnv4] peer 4.4.4.4 reflect-client
[*Spine1-bgp-af-vpnv4] quit
[*Spine1-bgp] quit
[*Spine1] commit
```

配置Leaf1。Leaf2和Leaf3的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] bgp 100
[*Leaf1-bgp] router-id 2.2.2.2
[*Leaf1-bgp] peer 1.1.1.1 as-number 100
[*Leaf1-bgp] peer 1.1.1.1 connect-interface loopback0
[*Leaf1-bgp] ipv4-family vpnv4
[*Leaf1-bgp-af-vpnv4] peer 1.1.1.1 enable
[*Leaf1-bgp-af-vpnv4] quit
[*Leaf1-bgp] quit
[*Leaf1] commit
```

步骤2 在Leaf上为租户划分Engineering和Business两个网络

配置Leaf1。Leaf2和Leaf3的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] ip vpn-instance Engineering
[*Leaf1-vpn-instance-Engineering] ipv4-family
[*Leaf1-vpn-instance-Engineering-af-ipv4] route-distinguisher 1:1
[*Leaf1-vpn-instance-Engineering-af-ipv4] vpn-target 1:1 export-extcommunity
[*Leaf1-vpn-instance-Engineering-af-ipv4] vpn-target 1:1 import-extcommunity
[*Leaf1-vpn-instance-Engineering-af-ipv4] vpn-target 3:3 import-extcommunity
[*Leaf1-vpn-instance-Engineering-af-ipv4] quit
```



```
[*Leaf1-vpn-instance-Engineering] vxlan vni 9999
[*Leaf1-vpn-instance-Engineering] quit
[*Leaf1] ip vpn-instance Business
[*Leaf1-vpn-instance-Business] ipv4-family
[*Leaf1-vpn-instance-Business-af-ipv4] route-distinguisher 2:2
[*Leaf1-vpn-instance-Business-af-ipv4] vpn-target 2:2 export-extcommunity
[*Leaf1-vpn-instance-Business-af-ipv4] vpn-target 2:2 import-extcommunity
[*Leaf1-vpn-instance-Business-af-ipv4] vpn-target 4:4 import-extcommunity
[*Leaf1-vpn-instance-Business-af-ipv4] quit
[*Leaf1-vpn-instance-Business] vxlan vni 9998
[*Leaf1-vpn-instance-Business] quit
[*Leaf1] commit
```

步骤3 在Leaf上划分子网

配置Leaf1。Leaf2和Leaf3的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] bridge-domain 1
[*Leaf1-bd1] vxlan vni 5000
[*Leaf1-bd1] arp broadcast-suppress enable
[*Leaf1-bd1] quit
[*Leaf1] bridge-domain 2
[*Leaf1-bd2] vxlan vni 5001
[*Leaf1-bd2] arp broadcast-suppress enable
[*Leaf1-bd2] quit
[*Leaf1] bridge-domain 3
[*Leaf1-bd3] vxlan vni 5002
[*Leaf1-bd3] arp broadcast-suppress enable
[*Leaf1-bd3] quit
[*Leaf1] evn bgp
[*Leaf1-evnbgp] source-address 2.2.2.4
[*Leaf1-evnbgp] peer 3.3.3.5
[*Leaf1-evnbgp] peer 4.4.4.5
[*Leaf1-evnbgp] quit
[*Leaf1] interface vbdif 1
[*Leaf1-Vbdif1] ip binding vpn-instance Engineering
[*Leaf1-Vbdif1] ip address 10.1.1.1 255.255.255.0
[*Leaf1-Vbdif1] mac-address 0000-5e00-0101
[*Leaf1-Vbdif1] arp distribute-gateway enable
[*Leaf1-Vbdif1] arp direct-route enable
[*Leaf1-Vbdif1] quit
[*Leaf1] interface vbdif 2
[*Leaf1-Vbdif2] ip binding vpn-instance Engineering
[*Leaf1-Vbdif2] ip address 10.10.1.1 255.255.255.0
[*Leaf1-Vbdif2] mac-address 0000-5e00-0102
[*Leaf1-Vbdif2] arp distribute-gateway enable
[*Leaf1-Vbdif2] arp direct-route enable
[*Leaf1-Vbdif2] quit
[*Leaf1] interface vbdif 3
[*Leaf1-Vbdif3] ip binding vpn-instance Business
[*Leaf1-Vbdif3] ip address 10.100.1.1 255.255.255.0
[*Leaf1-Vbdif3] mac-address 0000-5e00-0103
[*Leaf1-Vbdif3] arp distribute-gateway enable
[*Leaf1-Vbdif3] arp direct-route enable
[*Leaf1-Vbdif3] quit
[*Leaf1] commit
```

步骤4 配置Server1双归接入Leaf1和Leaf2

服务器上行连接交换机的端口需要绑定在一个聚合链路中且链路聚合模式需要和交换机侧的聚合模式匹配。

在Leaf1上创建Eth-Trunk，配置为静态LACP模式并加入成员口。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface eth-trunk 1
[*Leaf1-Eth-Trunk1] mode lacp-static
```

```
[*Leaf1-Eth-Trunk1] trunkport 10ge 1/0/4 to 1/0/5
[*Leaf1-Eth-Trunk1] quit
[*Leaf1] interface eth-trunk 10
[*Leaf1-Eth-Trunk10] mode lacp-dynamic
[*Leaf1-Eth-Trunk10] trunkport 10ge 1/0/2 to 1/0/3
[*Leaf1-Eth-Trunk10] quit
[*Leaf1] commit
```

步骤5 在Leaf1和Leaf2上配置DFS Group

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] dfs-group 1
[*Leaf1-dfs-group-1] source ip 2.2.2.3
[*Leaf1-dfs-group-1] quit
[*Leaf1] commit
```

步骤6 将Leaf1与Leaf2之间的链路配置为peer-link

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface eth-trunk 1
[~Leaf1-Eth-Trunk1] undo stp enable
[*Leaf1-Eth-Trunk1] peer-link 1
[*Leaf1-Eth-Trunk1] quit
[*Leaf1] commit
```

步骤7 分别在Leaf1和Leaf2上配置绑定DFS和用户侧Eth-Trunk接口

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface eth-trunk 10
[~Leaf1-Eth-Trunk10] dfs-group 1 m-lag 1
[*Leaf1-Eth-Trunk10] lacp m-lag system-id 00e0-fc00-0000
[*Leaf1-Eth-Trunk10] quit
[*Leaf1] commit
```

步骤8 在Leaf上配置VXLAN二层网关实现区分业务流量

配置Leaf1。Leaf2的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface interface eth-trunk 10.1 mode l2
[*Leaf1-Eth-Trunk10.1] encapsulation dot1q vid 2
[*Leaf1-Eth-Trunk10.1] bridge-domain 1
[*Leaf1-Eth-Trunk10.1] quit
[*Leaf1] interface interface eth-trunk 10.2 mode l2
[*Leaf1-Eth-Trunk10.2] encapsulation dot1q vid 3
[*Leaf1-Eth-Trunk10.2] bridge-domain 2
[*Leaf1-Eth-Trunk10.2] quit
[*Leaf1] interface interface eth-trunk 10.3 mode l2
[*Leaf1-Eth-Trunk10.3] encapsulation dot1q vid 4
[*Leaf1-Eth-Trunk10.3] bridge-domain 3
[*Leaf1-Eth-Trunk10.3] quit
[*Leaf1] commit
```

步骤9 创建VXLAN隧道

配置Leaf1上的L2 VXLAN隧道。Leaf2和Leaf3的配置与Leaf1类似，此处不再赘述。

```
[~Leaf1] interface Nve 1
[*Leaf1-Nve1] source 2.2.2.5
[*Leaf1-Nve1] vni 5000 head-end peer-list 4.4.4.6
[*Leaf1-Nve1] vni 5001 head-end peer-list 4.4.4.6
[*Leaf1-Nve1] vni 5002 head-end peer-list 4.4.4.6
[*Leaf1-Nve1] quit
[*Leaf1] commit
```

配置Leaf1上的L3 VXLAN隧道。Leaf2和Leaf3的配置与Leaf1相同，此处不再赘述。

```
[~Leaf1] interface Nve 2
[*Leaf1-Nve2] source 2.2.2.2
[*Leaf1-Nve2] mode 13
[*Leaf1-Nve2] quit
[*Leaf1] commit
```

步骤10 通过BGP传递三层网关主机路由信息

配置Spine1。

```
[~Spine1] bgp 100
[~Spine1] ipv4-family vpnv4
[~Spine1-bgp-af-vpnv4] peer 2.2.2.2 advertise remote-nexthop
[*Spine1-bgp-af-vpnv4] peer 3.3.3.3 advertise remote-nexthop
[*Spine1-bgp-af-vpnv4] peer 4.4.4.4 advertise remote-nexthop
[*Spine1-bgp-af-vpnv4] quit
[*Spine1-bgp] quit
[*Spine1] commit
```

配置Leaf1。Leaf2和Leaf3的配置与Leaf1相同，此处不再赘述。

```
[~Leaf1] bgp 100
[~Leaf1-bgp] ipv4-family vpnv4
[*Leaf1-bgp-af-vpnv4] peer 1.1.1.1 advertise remote-nexthop
[*Leaf1-bgp-af-vpnv4] quit
[*Leaf1-bgp] ipv4-family vpn-instance Engineering
[*Leaf1-bgp-Engineering] import-route direct
[*Leaf1-bgp-Engineering] quit
[*Leaf1-bgp] ipv4-family vpn-instance Business
[*Leaf1-bgp-Business] import-route direct
[*Leaf1-bgp-Business] quit
[*Leaf1] commit
```

步骤11 检查配置结果

上述配置成功后，在Leaf1、Leaf2上执行**display vxlan tunnel**命令可查看到VXLAN隧道的信息。

配置完成后，不同网段中的VM1可以相互通信。

----结束

配置文件

- Spine1的配置文件

```
#
sysname Spine1
#
interface 10GE1/0/1
undo portswitch
ip address 192.168.1.1 255.255.255.0
#
interface 10GE1/0/2
undo portswitch
ip address 192.168.2.1 255.255.255.0
#
interface 10GE1/0/3
undo portswitch
ip address 192.168.3.1 255.255.255.0
#
interface LoopBack0
ip address 1.1.1.1 255.255.255.255
#
```

```
bgp 100
router-id 1.1.1.1
peer 2.2.2.2 as-number 100
peer 2.2.2.2 connect-interface LoopBack0
peer 3.3.3.3 as-number 100
peer 3.3.3.3 connect-interface LoopBack0
peer 4.4.4.4 as-number 100
peer 4.4.4.4 connect-interface LoopBack0
#
ipv4-family unicast
peer 2.2.2.2 enable
peer 2.2.2.2 reflect-client
peer 3.3.3.3 enable
peer 3.3.3.3 reflect-client
peer 4.4.4.4 enable
peer 4.4.4.4 reflect-client
#
ipv4-family vpnv4
undo policy vpn-target
peer 2.2.2.2 enable
peer 2.2.2.2 reflect-client
peer 2.2.2.2 advertise remote-nexthop
peer 3.3.3.3 enable
peer 3.3.3.3 reflect-client
peer 3.3.3.3 advertise remote-nexthop
peer 4.4.4.4 enable
peer 4.4.4.4 reflect-client
peer 4.4.4.4 advertise remote-nexthop
#
ospf 1
area 0.0.0.0
network 1.1.1.1 0.0.0.0
network 192.168.1.0 0.0.0.255
network 192.168.2.0 0.0.0.255
network 192.168.3.0 0.0.0.255
#
return
```

● Leaf1 的配置文件

```
#
sysname Leaf1
#
vlan batch 2 to 4
#
dfs-group 1
source ip 2.2.2.3
#
ip vpn-instance Engineering
ipv4-family
route-distinguisher 1:1
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
vpn-target 3:3 import-extcommunity
vxlan vni 9999
#
ip vpn-instance Business
ipv4-family
route-distinguisher 2:2
vpn-target 2:2 export-extcommunity
vpn-target 2:2 import-extcommunity
vpn-target 4:4 import-extcommunity
vxlan vni 9998
#
bridge-domain 1
vxlan vni 5000
arp broadcast-suppress enable
#
```

```
bridge-domain 2
  vxlan vni 5001
  arp broadcast-suppress enable
#
bridge-domain 3
  vxlan vni 5002
  arp broadcast-suppress enable
#
interface Vbdif1
  ip binding vpn-instance Engineering
  ip address 10.1.1.1 255.255.255.0
  arp distribute-gateway enable
  mac-address 0000-5e00-0101
  arp direct-route enable
#
interface Vbdif2
  ip binding vpn-instance Engineering
  ip address 10.10.1.1 255.255.255.0
  arp distribute-gateway enable
  mac-address 0000-5e00-0102
  arp direct-route enable
#
interface Vbdif3
  ip binding vpn-instance Business
  ip address 10.100.1.1 255.255.255.0
  arp distribute-gateway enable
  mac-address 0000-5e00-0103
  arp direct-route enable
#
interface Eth-Trunk1
  stp disable
  mode lacp-dynamic
  peer-link 1
#
interface Eth-Trunk10
  mode lacp-dynamic
  dfs-group 1 m-lag 1
  lacp m-lag system-id 00e0-fc00-0000
#
interface Eth-Trunk10.1 mode 12
  encapsulation dot1q vid 2
  bridge-domain 1
#
interface Eth-Trunk10.2 mode 12
  encapsulation dot1q vid 3
  bridge-domain 2
#
interface Eth-Trunk10.3 mode 12
  encapsulation dot1q vid 4
  bridge-domain 3
#
interface 10GE1/0/1
  undo portswitch
  ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/2
  eth-trunk 10
#
interface 10GE1/0/3
  eth-trunk 10
#
interface 10GE1/0/4
  eth-trunk 1
#
interface 10GE1/0/5
  eth-trunk 1
#
```

```
interface LoopBack0
 ip address 2.2.2.2 255.255.255.255
#
interface LoopBack1
 ip address 2.2.2.3 255.255.255.255
#
interface LoopBack2
 ip address 2.2.2.4 255.255.255.255
#
interface LoopBack3
 ip address 2.2.2.5 255.255.255.255
#
interface Nve1
 source 2.2.2.5
 vni 5000 head-end peer-list 4.4.4.6
 vni 5001 head-end peer-list 4.4.4.6
 vni 5002 head-end peer-list 4.4.4.6
#
interface Nve2
 mode l3
 source 2.2.2.2
#
bgp 100
 router-id 2.2.2.2
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
#
 ipv4-family unicast
  peer 1.1.1.1 enable
#
 ipv4-family vpnv4
  policy vpn-target
  peer 1.1.1.1 enable
  peer 1.1.1.1 advertise remote-nexthop
#
 ipv4-family vpn-instance Engineering
  import-route direct
#
 ipv4-family vpn-instance Business
  import-route direct
#
evn bgp
 source-address 2.2.2.4
 peer 3.3.3.5
 peer 4.4.4.5
#
ospf 1
 area 0.0.0.0
  network 2.2.2.2 0.0.0.0
  network 2.2.2.3 0.0.0.0
  network 2.2.2.4 0.0.0.0
  network 2.2.2.5 0.0.0.0
  network 192.168.1.0 0.0.0.255
#
return
```

● Leaf2的配置文件

```
#
sysname Leaf2
#
vlan batch 2 to 4
#
dfs-group 1
 source ip 3.3.3.4
#
ip vpn-instance Engineering
 ipv4-family
```

```
route-distinguisher 3:3
vpn-target 3:3 export-extcommunity
vpn-target 1:1 import-extcommunity
vxlan vni 9999
#
ip vpn-instance Business
ipv4-family
route-distinguisher 4:4
vpn-target 4:4 export-extcommunity
vpn-target 2:2 import-extcommunity
vxlan vni 9998
#
bridge-domain 1
vxlan vni 5000
arp broadcast-suppress enable
#
bridge-domain 2
vxlan vni 5001
arp broadcast-suppress enable
#
bridge-domain 3
vxlan vni 5002
arp broadcast-suppress enable
#
interface Vbdif1
ip binding vpn-instance Engineering
ip address 10.1.1.1 255.255.255.0
arp distribute-gateway enable
mac-address 0000-5e00-0101
arp direct-route enable
#
interface Vbdif2
ip binding vpn-instance Engineering
ip address 10.10.1.1 255.255.255.0
arp distribute-gateway enable
mac-address 0000-5e00-0102
arp direct-route enable
#
interface Vbdif3
ip binding vpn-instance Business
ip address 10.100.1.1 255.255.255.0
arp distribute-gateway enable
mac-address 0000-5e00-0103
arp direct-route enable
#
interface Eth-Trunk1
stp disable
mode lacp-dynamic
peer-link 1
#
interface Eth-Trunk10
mode lacp-dynamic
dfs-group 1 m-lag 1
lacp m-lag system-id 00e0-fc00-0000
#
interface Eth-Trunk10.1 mode 12
encapsulation dot1q vid 2
bridge-domain 1
#
interface Eth-Trunk10.2 mode 12
encapsulation dot1q vid 3
bridge-domain 2
#
interface Eth-Trunk10.3 mode 12
encapsulation dot1q vid 4
bridge-domain 3
#
```

```
interface 10GE1/0/1
  undo portswitch
  ip address 192.168.2.2 255.255.255.0
#
interface 10GE1/0/2
  eth-trunk 10
#
interface 10GE1/0/3
  eth-trunk 10
#
interface 10GE1/0/4
  eth-trunk 1
#
interface 10GE1/0/5
  eth-trunk 1
#
interface LoopBack0
  ip address 3.3.3.3 255.255.255.255
#
interface LoopBack1
  ip address 3.3.3.4 255.255.255.255
#
interface LoopBack2
  ip address 3.3.3.5 255.255.255.255
#
interface LoopBack3
  ip address 2.2.2.5 255.255.255.255
#
interface Nve1
  source 2.2.2.5
  vni 5000 head-end peer-list 4.4.4.6
  vni 5001 head-end peer-list 4.4.4.6
  vni 5002 head-end peer-list 4.4.4.6
#
interface Nve2
  mode l3
  source 3.3.3.3
#
bgp 100
  router-id 3.3.3.3
  peer 1.1.1.1 as-number 100
  peer 1.1.1.1 connect-interface LoopBack0
#
  ipv4-family unicast
    peer 1.1.1.1 enable
#
  ipv4-family vpnv4
    policy vpn-target
    peer 1.1.1.1 enable
    peer 1.1.1.1 advertise remote-nexthop
#
  ipv4-family vpn-instance Engineering
    import-route direct
#
  ipv4-family vpn-instance Business
    import-route direct
#
evn bgp
  source-address 3.3.3.5
  peer 2.2.2.4
  peer 4.4.4.5
#
ospf 1
  area 0.0.0.0
  network 2.2.2.5 0.0.0.0
  network 3.3.3.3 0.0.0.0
  network 3.3.3.4 0.0.0.0
```



```
network 3.3.3.5 0.0.0.0
network 192.168.2.0 0.0.0.255
#
return
```

- Leaf3的配置文件

```
#
sysname Leaf3
#
ip vpn-instance Engineering
ipv4-family
route-distinguisher 1:1
vpn-target 1:1 export-extcommunity
vpn-target 1:1 import-extcommunity
vpn-target 3:3 import-extcommunity
vxlan vni 9999
#
ip vpn-instance Business
ipv4-family
route-distinguisher 2:2
vpn-target 2:2 export-extcommunity
vpn-target 2:2 import-extcommunity
vpn-target 4:4 import-extcommunity
vxlan vni 9998
#
bridge-domain 1
vxlan vni 5000
arp broadcast-suppress enable
#
bridge-domain 2
vxlan vni 5001
arp broadcast-suppress enable
#
bridge-domain 3
vxlan vni 5002
arp broadcast-suppress enable
#
interface Vbdif1
ip binding vpn-instance Engineering
ip address 10.1.1.1 255.255.255.0
arp distribute-gateway enable
mac-address 0000-5e00-0101
arp direct-route enable
#
interface Vbdif2
ip binding vpn-instance Engineering
ip address 10.10.1.1 255.255.255.0
arp distribute-gateway enable
mac-address 0000-5e00-0102
arp direct-route enable
#
interface Vbdif3
ip binding vpn-instance Business
ip address 10.100.1.1 255.255.255.0
arp distribute-gateway enable
mac-address 0000-5e00-0103
arp direct-route enable
#
interface 10GE1/0/1
undo portswitch
ip address 192.168.3.2 255.255.255.0
#
interface 10GE1/0/2
undo portswitch
ip address 192.168.1.2 255.255.255.0
#
interface 10GE1/0/3.1 mode 12
```

```
encapsulation dot1q vid 10
bridge-domain 10
#
interface LoopBack0
 ip address 4.4.4.4 255.255.255.255
#
interface LoopBack1
 ip address 4.4.4.5 255.255.255.255
#
interface LoopBack2
 ip address 4.4.4.6 255.255.255.255
#
interface Nve1
 source 4.4.4.6
 vni 5000 head-end peer-list 2.2.2.5
 vni 5001 head-end peer-list 2.2.2.5
 vni 5002 head-end peer-list 2.2.2.5
#
interface Nve2
 mode l3
 source 4.4.4.4
#
bgp 100
 router-id 4.4.4.4
 peer 1.1.1.1 as-number 100
 peer 1.1.1.1 connect-interface LoopBack0
#
ipv4-family unicast
 peer 1.1.1.1 enable
#
ipv4-family vpnv4
 policy vpn-target
 peer 1.1.1.1 enable
 peer 1.1.1.1 advertise remote-nexthop
#
ipv4-family vpn-instance Engineering
 import-route direct
#
ipv4-family vpn-instance Business
 import-route direct
#
evn bgp
 source-address 4.4.4.5
 peer 2.2.2.4
 peer 3.3.3.5
#
ospf 1
 area 0.0.0.0
 network 4.4.4.4 0.0.0.0
 network 4.4.4.5 0.0.0.0
 network 4.4.4.6 0.0.0.0
 network 192.168.3.0 0.0.0.255
#
return
```

1.9 参考标准和协议

介绍VXLAN的参考标准和协议。

与VXLAN特性相关的参考标准及协议如下：

文档	描述	备注
draft-ietf-nvo3-framework-04	Framework for DC Network Virtualization	-
draft-ietf-nvo3-dataplane-requirements-02	NVO3 Data Plane Requirements	-
RFC7348	Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks	-