



# UNIS S5600-EI 系列以太网交换机

## OpenFlow 配置指导

北京紫光恒越网络科技有限公司  
<http://www.unis-hy.com>

资料版本: 6W100-20160315  
产品版本: Release 1120

**Copyright © 2016** 北京紫光恒越网络科技有限公司及其许可者版权所有，保留一切权利。

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

**UNIS** 为北京紫光恒越网络科技有限公司的商标。对于本手册中出现的其它公司的商标、产品标识及商品名称，由各自权利人拥有。

由于产品版本升级或其他原因，本手册内容有可能变更。紫光恒越保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，紫光恒越尽全力在本手册中提供准确的信息，但是紫光恒越并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

# 前言

本配置指导主要介绍 UNIS S5600-EI 系列以太网交换机 OpenFlow 相关技术的原理及配置方法。

前言部分包含如下内容：

- [读者对象](#)
- [本书约定](#)
- [技术支持](#)
- [资料意见反馈](#)

## 读者对象

本手册主要适用于如下工程师：

- 网络规划人员
- 现场技术支持与维护人员
- 负责网络配置和维护的网络管理员

## 本书约定

### 1. 命令行格式约定






格 式	意 义
<b>粗体</b>	命令行关键字（命令中保持不变、必须照输的部分）采用 <b>加粗</b> 字体表示。
<i>斜体</i>	命令行参数（命令中必须由实际值进行替代的部分）采用 <i>斜体</i> 表示。
[ ]	表示用“[ ]”括起来的部分在命令配置时是可选的。
{x y ...}	表示从多个选项中仅选取一个。
[x y ...]	表示从多个选项中选取一个或者不选。
{x y ...}*	表示从多个选项中至少选取一个。
[x y ...]*	表示从多个选项中选取一个、多个或者不选。
&<1-n>	表示符号&前面的参数可以重复输入1~n次。
#	由“#”号开始的行表示为注释行。

### 2. 图形界面格式约定

格 式	意 义
<>	带尖括号“<>”表示按钮名，如“单击<确定>按钮”。
[ ]	带方括号“[ ]”表示窗口名、菜单名和数据表，如“弹出[新建用户]窗口”。
/	多级菜单用“/”隔开。如[文件/新建/文件夹]多级菜单表示[文件]菜单下的[新建]子菜单下的[文件夹]菜单项。

### 3. 各类标志

本书还采用各种醒目标志来表示在操作过程中应该特别注意的地方，这些标志的意义如下：

 警告	该标志后的注释需给予格外关注，不当的操作可能会对人身造成伤害。
 注意	提醒操作中应注意的事项，不当的操作可能会导致数据丢失或者设备损坏。
 提示	为确保设备配置成功或者正常工作而需要特别关注的操作或信息。
 说明	对操作内容的描述进行必要的补充和说明。
 窍门	配置、操作、或使用设备的技巧、小窍门。

### 4. 图标约定

本书使用的图标及其含义如下：

	该图标及其相关描述文字代表一般网络设备，如路由器、交换机、防火墙等。
	该图标及其相关描述文字代表一般意义下的路由器，以及其他运行了路由协议的设备。
	该图标及其相关描述文字代表二、三层以太网交换机，以及运行了二层协议的设备。
	该图标及其相关描述文字代表无线控制器、无线控制器业务板和有线无线一体化交换机的无线控制引擎设备。
	该图标及其相关描述文字代表无线接入点设备。
	该图标及其相关描述文字代表无线终结单元。
	该图标及其相关描述文字代表无线终结者。
	该图标及其相关描述文字代表无线Mesh设备。
	该图标代表发散的无线射频信号。
	该图标代表点到点的无线射频信号。
	该图标及其相关描述文字代表防火墙、UTM、多业务安全网关、负载均衡等安全设备。
	该图标及其相关描述文字代表防火墙插卡、负载均衡插卡、NetStream插卡、SSL VPN插卡、IPS插卡、ACG插卡等安全插卡。

## 5. 端口编号示例约定

本手册中出现的端口编号仅作示例，并不代表设备上实际具有此编号的端口，实际使用中请以设备上存在的端口编号为准。

## 技术支持

用户支持邮箱：[zgsm\\_service@thunis.com](mailto:zgsm_service@thunis.com)

技术支持热线电话：400-910-9998（手机、固话均可拨打）

网址：<http://www.unis-hy.com>

## 资料意见反馈

如果您在使用过程中发现产品资料的任何问题，可以通过以下方式反馈：

E-mail：[zgsm\\_info@thunis.com](mailto:zgsm_info@thunis.com)

感谢您的反馈，让我们做得更好！

# 目 录

1 OpenFlow .....	1-1
1.1 OpenFlow概述.....	1-1
1.1.1 OpenFlow背景.....	1-1
1.1.2 OpenFlow介绍.....	1-1
1.1.3 OpenFlow Switch类型 .....	1-1
1.1.4 OpenFlow接口 .....	1-2
1.1.5 OpenFlow实例.....	1-2
1.1.6 OpenFlow流表.....	1-3
1.1.7 Group Table.....	1-5
1.1.8 Meter Table .....	1-5
1.1.9 OpenFlow channel .....	1-6
1.1.10 协议规范 .....	1-7
1.2 OpenFlow配置任务简介 .....	1-7
1.3 配置OpenFlow实例 .....	1-8
1.3.1 创建OpenFlow实例.....	1-8
1.3.2 配置OpenFlow实例的基本能力.....	1-8
1.3.3 激活OpenFlow实例.....	1-12
1.4 配置连接控制器 .....	1-13
1.4.1 配置主连接.....	1-13
1.4.2 配置连接中断模式.....	1-13
1.5 配置OpenFlow定时器.....	1-14
1.6 配置支持动态MAC地址 .....	1-14
1.7 开启OpenFlow的无丢包模式.....	1-15
1.8 配置QINQ模式.....	1-15
1.9 关闭流表项变化成功后打印日志的开关.....	1-15
1.10 刷新OpenFlow实例下所有MAC-IP表的三层表项 .....	1-16
1.11 OpenFlow显示和维护.....	1-16
1.12 OpenFlow典型配置举例 .....	1-17
1.13 附录 A 应用限制 .....	1-18
1.13.1 流表项限制.....	1-18
1.13.2 Action List和Action Set整合的限制.....	1-19
1.13.3 Packet Out的处理限制.....	1-20
1.13.4 Packet in的处理限制.....	1-20

1.13.5 匹配LLDP报文的限制 .....	1-20
1.13.6 Flow Mod的限制 .....	1-21
1.14 附录 B MAC-IP流表.....	1-21
1.14.1 MAC-IP流表支持能力 .....	1-21
1.14.2 MAC-IP流表的限制.....	1-22
1.14.3 MAC-IP流表的Table Miss.....	1-23
1.14.4 Dynamic aware.....	1-23
1.14.5 MAC-IP Table与Extensibility Table的配合 .....	1-23

# 1 OpenFlow

## 1.1 OpenFlow概述

### 1.1.1 OpenFlow背景

随着虚拟化技术的大规模应用，传统的网络架构和网络设备的不足越来越多的被暴露出来，如控制平面与转发平面相集成，扩展性低，不易定制，技术更新周期长，过于依赖网络设备商。而虚拟化技术要求设备操作简单灵活、扩展性能高、不需要用专门的芯片来进行转发控制。传统的网络设备已经无法满足虚拟化技术以及数据中心网络的需求。而 SDN（Software Defined Network，软件定义网络）技术可以分离控制平面和网络转发平面，解决数据中心网络中所面临的问题。

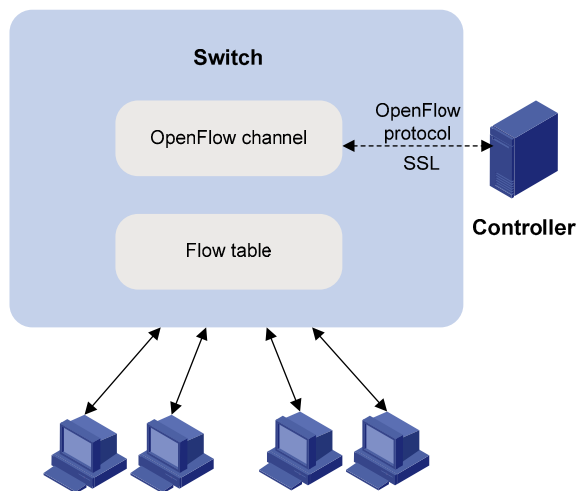
### 1.1.2 OpenFlow介绍

OpenFlow 是 SDN（Software Defined Network，软件定义网络）架构中定义的一个控制器与转发器之间的通信接口标准。OpenFlow 允许控制器直接访问和操作网络设备的转发平面，这些网络设备可能是物理上的，也可能是虚拟的路由器或者交换机。

OpenFlow 的思想：分离控制平面和数据平面，二者之间使用标准的协议通信；数据平面采用基于流的方式进行转发。

OpenFlow网络由OpenFlow设备(Switch)和控制器(Controller)通过安全通道(OpenFlow channel)组成，如 图1-1 所示。Switch与Controller通过TLS或者TCP建立安全通道，进行OpenFlow消息交互，实现表项下发、查询以及状态上报等功能。下文如果没有特殊说明，交换机指的就是OpenFlow设备。

图1-1 OpenFlow 网络组成



### 1.1.3 OpenFlow Switch类型

OpenFlow Switch 有下面两种：



- OpenFlow-Only Switch: 仅支持 OpenFlow 转发。
- OpenFlow-Hybrid Switch: 既支持 OpenFlow 转发, 也支持正常转发。本系列交换机即为 OpenFlow-Hybrid Switch。

### 1.1.4 OpenFlow接口

OpenFlow 接口有下面三类:

- 物理接口: 例如交换机的以太网接口等, 可以作为入接口和出接口。
- 逻辑接口: 例如聚合接口等, 可以作为入接口和出接口。



说明

设备 **不支持**匹配 LoopBack 接口。

- 保留接口: 由转发动作定义的接口, 实现OpenFlow转发功能。除Any接口外, 其他接口仅可以作为出接口, 仅Controller和Local可以作为入接口。具体类型请参见 [表 1-1](#)。

表1-1 保留接口类型

类型	说明
ALL	报文从所有接口发送
Controller	报文上送控制器
Local	报文上送本地CPU
Normal	报文正常转发
Flood	报文广播发送

### 1.1.5 OpenFlow实例

OpenFlow 支持多实例。每个 OpenFlow 实例可以单独连接控制器, 相当于一台独立的交换机, OpenFlow 实例根据控制器下发的流表项指导流量的转发, 下文如果没有特殊说明, 交换机为一个 OpenFlow 实例。

#### 1. 实例划分

OpenFlow 实例通过 VLAN 划分作用范围, 在实例的作用范围内, 流量转发遵循流表中定义的规则。

#### 2. 实例激活

OpenFlow 实例配置完成后, 需要进行激活才能生效。

OpenFlow 需要将设备的支持能力、当前的接口信息等设备信息上报给控制器后, 控制器才能够下发流表项指导转发。

在配置更改的情况下, 需要重新激活实例, 使配置生效。在激活配置时, OpenFlow 实例会与所有控制器断开连接, 然后重新进行连接。

### 3. 实例所属接口

OpenFlow 协议规定 OpenFlow 实例需要将接口信息上报给控制器，这些接口包括物理接口、逻辑接口以及保留接口中的 Local。

对于通过 VLAN 划分作用范围的 OpenFlow 实例，当且仅当接口所属 VLAN 完全包含了 OpenFlow 对应的 VLAN 后，该接口才是该 OpenFlow 实例的接口，可以被上报到控制器。如果配置了 loosen 模式，只要接口所在 VLAN 与实例配置 VLAN 存在交集，接口就属于 OpenFlow 实例。

## 1.1.6 OpenFlow 流表

### 1. 流表项组成

OpenFlow 通过流表（Flow Table）来匹配和处理报文，在同一个流表中按流表项的优先级进行先后匹配。一台交换机上可以包含一个或者多个流表。

流表分为两种类型：

- **MAC-IP 流表：**通过 MAC 地址表和 FIB 表实现。只能匹配报文的目地 MAC 地址、VLAN 以及目的 IP 地址，动作也仅支持修改报文的目地 MAC 地址、源 MAC 地址、VLAN 以及指定出接口。
- **Extensibility 流表：**扩展流表，使用 TCAM（Ternary Content Addressable Memory，三态内容寻址存储器）实现，可以匹配报文的源 MAC 地址、目的 MAC 地址、源 IP 地址、目的 IP 地址、报文优先级、TCP 源端口、TCP 目的端口等。

图1-2 流表项结构

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie
--------------	----------	----------	--------------	----------	--------

流表项结构如 [图 1-2](#) 所示：

- **Match Fields：**匹配规则。可以匹配入接口、报文头等字段。
- **Priority：**优先级。定义流表项之间的匹配顺序，优先级高的先匹配。
- **Counters：**统计计数。统计有多少个报文和字节匹配到该流表项。
- **Instructions：**动作指令集。定义匹配到该流表项的报文需要进行的处理。流表项动作指令集是对动作进行操作，流表项的动作有两种执行类型：
  - **动作集（Action Set）：**一系列动作的组合，设备不会立刻修改报文内容，直到报文不再需要进入下一级流表，动作集里每种动作仅有一个，并且按照 [图 1-2](#) 从上到下的顺序执行。
  - **动作序列（Action List）：**需要立即执行的一系列动作，其动作内容与动作集相同，此时立即修改报文的内容，并且执行顺序按照下发的顺序执行。

表1-2 动作指令集定义

指令	描述
Meter	对匹配到流表项的报文进行限速
Apply-Actions	立即执行动作序列中的动作
Clear-Actions	清除动作集中的所有动作

指令	描述
Write-Actions	更改动作集中的所有动作
Write-Metadata	更改流表间数据，在支持多级流表时使用
Goto-Table	进入下一级流表，Extensibility 流表不支持该指令集

具体动作类型如 [表 1-3](#) 所示。

表1-3 动作类型（1.3.1 版本）

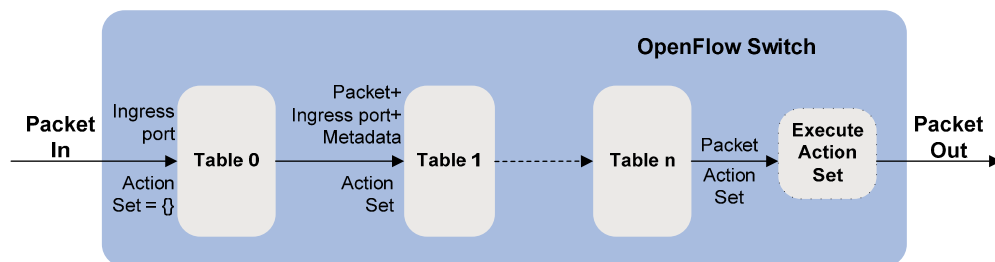
动作名称	可选/必选	描述
Output	必选	转发报文到特定的OpenFlow端口，例如物理端口，逻辑端口以及OpenFlow保留端口
Drop	必选	没有直接的动作来代表Drop，当动作集中不含有Output指令时，报文会被丢弃。通常空指令集，空动作集或者执行清空动作集后，报文会被丢弃
Group	必选	将报文转交给指定的Group表处理，该动作的确切含义由Group的类型定义
Set-Queue	可选	Set-Queue动作为报文指定队列ID。当报文被转发到特定端口时，队列ID通常被用于QoS
Set-Field	可选	识别报文字段的类型，并且可以修改该字段的值。Set-Field动作通常只适用于报文最外层的字段（比如当内外层均有VLAN tag时，该动作只修改最外层的VLAN Tag）

- **Timeouts:** 超时时间。包括了 Idle time 和 Hard time。
  - Idle time: 在 Idle time 时间内，如果没有报文匹配到该流表项，则此流表项被删除。
  - Hard time: 在 hard time 时间超时后，无论是否有报文匹配到该流表项，此流表项都会被删除
- **Cookie:** 控制器下发的流表项的标识。

## (2) 流表处理流程

如 [图 1-3](#) 所示，报文进入设备后，设备先解析报文，然后从第一个流表开始，按照流表编号从小到大依次匹配，当数据包成功匹配一条规则后，将首先更新该规则对应的统计数据（如成功匹配数据包总数），然后根据规则中的指令进行相应操作（比如转下一个流表继续处理、修改或者立即执行该数据包对应的动作等）。当数据包处于最后一个流表时，其对应的动作将被执行（比如转发至某一端口、修改数据包某一字段或丢弃数据包等）。

图1-3 OpenFlow 转发示意图



## 2. Table Miss表项

每个流表都包含一个 **Table Miss** 流表项，该表项用于定义在流表中没有匹配的报文的处理方式，该表项匹配任何报文，优先级为 0，动作指令与正常表项相同。

### 1.1.7 Group Table

Group Table 由 Group 表项组成，Group 表项被流表项所引用，提供额外的报文转发功能。

图1-4 Group 表项结构

Group Identifier	Group Type	Counters	Action Buckets
------------------	------------	----------	----------------

- **Group Identifier:** Group ID，用于识别 Group，32bits。
- **Group Type:** Group 类型。All 表示执行所有动作桶，用于组播或者广播。
- **Counters:** 当报文被 Group 处理时，更新计数器。
- **Action Buckets:** 一个由动作桶组成的有序列表。每个动作桶由许多动作组成。

### 1.1.8 Meter Table

Meter Table 由 Meter 表项组成，Meter 表项被流表项所引用，为所有引用 Meter 表项的流表项提供报文限速的功能。

图1-5 Meter 表项结构

Meter Identifier	Meter Bands	Counters
------------------	-------------	----------

- **Meter Identifier:** Meter ID，用于识别 Meter，32bits。
- **Meter Bands:** 一个 Meter 表项可以包含一个或者多个 Meter Bands，每个 Meter Band 定义了速率以及动作。当报文的速率超过了某些 Meter Band，根据这些 Meter Band 中速率最大的那个定义的动作进行处理。
- **Counters:** 当报文被 Meter 处理时，更新计数器。

图1-6 Meter Bands 结构

Band Type	Rate	Counters	Type Specific arguments
-----------	------	----------	-------------------------

- **Band Type:** Band 类型，定义报文如何处理。为可选，可使用丢弃（drop），即报文高于该速率会被丢弃；以及重新标记 DSCP（dscp remark）。
- **Rate: Meter** 用于选择 Band 的最低速率，即报文速率高于该速率并最接近该速率，该 Band 将被应用。
- **Counters:** 当 Band 处理报文时，更新计数器。
- **Type Specific arguments:** 某些 Band 含有的特定参数。

### 1.1.9 OpenFlow channel

交换机与控制器通过 TLS 或者 TCP 建立 Channel，进行 OpenFlow 消息交互，实现表项下发、查询以及状态上报等功能。

OpenFlow 协议中定义了三种消息类型，即：Controller to Switch 消息、异步消息和同步消息，每种报文类型都有多种子类型。

#### 1. Controller to Switch 消息

Controller to Switch 消息是指由控制器产生并发送到交换机，可以不需要交换机响应。这些消息主要由控制器用来对交换机进行状态查询和修改配置等操作。

表1-4 Controller to Switch 消息（1.3.1 版本）

Controller to Switch 消息类型	描述
Features	控制器发送该请求了解交换机对OpenFlow的支持能力，交换机必须回应该请求
Configuration	控制器发送该消息配置或查询交换机的配置参数，交换机只有在控制器查询时回应
Modify-State	控制器使用该消息管理交换机的状态，如增加、删除、修改、交换机的流表项/Group表项，以及交换机端口的属性
Multipart	控制器收集交换机各方面的信息，例如当前配置，统计信息等
Packet-Out	用于通过交换机特定端口发送报文，这些报文可以通过Packet-In消息触发，也可以通过控制器直接发送。通常Packet-Out消息包含整个之前接收到的Packet-In消息所携带的报文或者buffer ID（用于指示存储在交换机内的特定报文）。这个消息需要包含一个动作列表，当交换机收到该动作列表后会对Packet-Out消息所携带的报文执行该动作列表。如果动作列表为空，Packet-Out消息所携带的报文将被交换机丢弃
Barrier	控制器用该消息确认之前下发动作是否成功。控制器发送Barrier请求消息，当交换机之前下发的流表等操作都已经成功时会回复Barrier应答消息
Role-Request	用于设定或查询OpenFlow channel的角色。通常用于交换机和多个控制器相连的情况
Asynchronous-Configuration	控制器使用该报文设定异步消息过滤器来接收其只希望接收到的异步消息报文，或者向交换机查询该过滤器。通常用于交换机和多个控制器相连的情况

#### 2. 异步（Asynchronous）消息

异步（Asynchronous）消息是由交换机发送给控制器，用来通知交换机上发生的异步事件的消息。例如，当某一条规则因为超时而删除时，交换机将自动发送一条流表项删除消息通知控制器。

表1-5 异步消息（1.3.1 版本）

子类型	描述
Packet-In	转移报文的控制权到控制器。对于所有通过匹配流表项或者Table Miss后转发到保留端口Controller端口的报文均要通过Packet-in消息送到控制器。也有部分其他流程，如TTL检查等，也需要通过该消息和控制器交互。Packet-In既可以携带报文，也可以通过在交换机内部设置报文的Buffer来仅携带报文头以及其Buffer ID传输给控制器。控制器在接收到Packet-In消息后会对其接收到的报文或者报文头和Buffer ID进行处理，并返回Packet-out消息通知交换机如何处理该报文
Flow-Removed	通知控制器将某个流表项从流表的移除。通常该消息在控制器发送删除流表项的消息或者流表项的两个定时器其中之一超时产生
Port-Status	通知控制器端口状态或设置的改变
Error	通知控制器交换机出现的问题或错误

### 3. 对称（Symmetric）消息

对称（Symmetric）消息主要用来建立连接和检测对方是否在线等。

表1-6 对称消息（1.3.1 版本）

子类型	描述
Hello	当交换机和控制器建立连接时，双方发送Hello消息进行交互
Echo	用于验证控制器与交换机之间连接的存活，控制器和交换机都会发送Echo request/reply消息，接收到Echo request消息时，必须能返回Echo reply消息。也可用于测量控制器与交换机之间链路的延迟和带宽
Experimenter	预留消息类型

## 1.1.10 协议规范

OpenFlow Switch Specification Version1.3.1

## 1.2 OpenFlow配置任务简介

表1-7 OpenFlow 配置任务简介

配置任务		说明	详细配置	
配置OpenFlow实例	创建OpenFlow实例	必选	<a href="#">1.3.1</a>	
	配置OpenFlow实例的基本能力	配置OpenFlow实例的类型	必选	<a href="#">1.3.2 1.</a>
		配置流表ID	可选	<a href="#">1.3.2 2.</a>
		配置OpenFlow实例与控制器的连接模式	可选	<a href="#">1.3.2 3.</a>
		配置Extensibility流表项的上限	可选	<a href="#">1.3.2 4.</a>

配置任务		说明	详细配置
	配置带内管理VLAN	可选	<a href="#">1.3.2 5.</a>
	配置OpenFlow实例对应的VLAN禁止MAC地址学习	可选	<a href="#">1.3.2 6.</a>
	配置禁止OpenFlow实例上送Controller的端口类型	可选	<a href="#">1.3.2 7.</a>
	配置Table Miss表项的缺省动作	可选	<a href="#">1.3.2 8.</a>
	配置OpenFlow实例的Datapath ID	可选	<a href="#">1.3.2 9.</a>
	激活OpenFlow实例	必选	<a href="#">1.3.3</a>
配置连接控制器	配置主连接	必选	<a href="#">1.4.1</a>
	配置连接中断模式	可选	<a href="#">1.4.2</a>
配置OpenFlow定时器		可选	<a href="#">1.5</a>
配置支持动态MAC地址		可选	<a href="#">1.6</a>
开启OpenFlow的无丢包模式		可选	<a href="#">1.7</a>
配置QINQ模式		可选	<a href="#">1.8</a>
关闭流表项变化成功后打印日志的开关		可选	<a href="#">1.9</a>
刷新OpenFlow实例下所有MAC-IP表的三层表项		可选	<a href="#">1.10</a>

## 1.3 配置OpenFlow实例

### 1.3.1 创建OpenFlow实例

表1-8 创建 OpenFlow 实例

操作	命令	说明
进入系统视图	<b>system-view</b>	-
创建OpenFlow实例，并进入OpenFlow实例视图	<b>openflow instance <i>instance-id</i></b>	缺省情况下，没有配置OpenFlow实例
(可选)配置OpenFlow实例描述	<b>description <i>text</i></b>	缺省情况下，未配置OpenFlow实例的描述信息

### 1.3.2 配置OpenFlow实例的基本能力

下面配置用于定义 OpenFlow 实例的基本能力，设备与控制器建立连接后，会向控制器上报这些基本能力，控制器根据这些能力下发表项。

## 1. 配置OpenFlow实例的类型

配置对应的 VLAN 后，对应 VLAN 内的流量根据 OpenFlow 流表项转发，其它 VLAN 内的流量进行正常转发。



注意

- 不能将同一个 VLAN 映射到多个实例中，否则，流量无法正确处理。
- 如果 OpenFlow 实例对应的 VLAN 不存在，在激活实例后，不存在的 VLAN 会自动创建。在实例激活后，请不要删除 OpenFlow 实例对应的 VLAN。
- 配置 VLAN 对应的 VLAN 接口下不允许配置 BFD MAD 检测功能，该功能的相关内容请参见“IRF 配置指导”中的“IRF”。

表1-9 配置 OpenFlow 实例的类型

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置OpenFlow实例对应的VLAN	<b>classification vlan</b> <i>vlan-id</i> [ <b>mask</b> <i>vlan-mask</i> ] [ <b>loosen</b> ]	缺省情况下，没有配置OpenFlow实例对应的VLAN

## 2. 配置流表ID

表1-10 配置流表 ID

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置OpenFlow实例下的流表类型和流表ID	<b>flow-table</b> { [ <b>ingress-vlan</b> <i>ingress-table-id</i> ] [ <b>extensibility</b> <i>extensibility-table-id</i> ]   <b>mac-ip</b> <i>mac-ip-table-id</i> } * [ <b>egress-vlan</b> <i>egress-table-id</i> ] }	缺省情况下，Extensibility流表ID为0，没有配置MAC-IP流表ID 一个OpenFlow实例只支持配置一个MAC-IP表和一个Extensibility表，如果多次执行该命令，后面的配置将会覆盖前面的配置 MAC-IP表的ID值必须小于Extensibility表的ID值 只有在配置了QINQ模式，并且设备处于Standalone模式时， <b>ingress-vlan</b> 和 <b>egress-vlan</b> 这两个流表才生效

## 3. 配置OpenFlow实例与控制器的连接模式

OpenFlow 实例与控制器连接时，支持以下两种模式：

- **Single** 模式：同一时刻，OpenFlow 实例仅与一个控制器建立连接，配置的多个控制器之间互为备份，当且仅当当前的连接断开后，OpenFlow 实例连接下一个控制器，直到连接成功。



- **Multiple 模式**：同一时刻，OpenFlow 实例可以与多个控制器建立连接，在与某个控制器连接失败或者断开连接后，在重连时间间隔后重新与之进行连接，直到连接成功。



**说明**  
Single 模式下，设备根据 Controller ID 从小到大顺序依次连接控制器，只有在前一个控制器连接不上时才会连接下一个控制器，如果配置了多个路由不可达的控制器，设备遍历到路由可达的控制器建立连接可能耗时较长。

表1-11 配置 OpenFlow 实例与控制器的连接模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置OpenFlow实例与控制器的连接模式	<b>controller mode { multiple   single }</b>	缺省情况下，OpenFlow实例与控制器的连接模式为 <b>multiple</b>

#### 4. 配置Extensibility流表项的上限

在 OpenFlow 实例中允许定义 Extensibility 流表支持的表项最大值，当控制器下发的流表表项个数超过最大值的时候，向控制器返回失败。

表1-12 配置 Extensibility 流表项的上限

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置Extensibility流表项的上限	<b>flow-entry max-limit</b> <i>limit-value</i>	缺省情况下，Extensibility流表项的上限为65535

#### 5. 配置带内管理VLAN

带内管理 VLAN 是 OpenFlow 实例对应的 VLAN 中的一个或多个 VLAN，专门用于在 OpenFlow 实例内建立该实例与控制器的连接。

配置带内管理 VLAN 后，带内管理 VLAN 中的数据报文不再进行 OpenFlow 转发，并且仅在带内管理 VLAN 中的端口不再属于 OpenFlow 接口。因此，进行该功能的配置前，请用户做好网络规划。



**说明**  
带内管理 VLAN 必须是 OpenFlow 实例对应的 VLAN 的子集。

表1-13 配置带内管理 VLAN

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置带内管理VLAN	<b>in-band management vlan</b> <i>vlan-list</i>	缺省情况下，没有配置带内管理VLAN

## 6. 配置OpenFlow实例对应的VLAN禁止MAC地址学习

配置该功能后，OpenFlow 实例对应的 VLAN 的 MAC 地址学习功能被禁止。



说明

带内管理 VLAN 不受该功能限制。

表1-14 配置 OpenFlow 实例对应的 VLAN 禁止 MAC 地址学习

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置OpenFlow实例对应的VLAN禁止MAC地址学习	<b>mac-learning forbidden</b>	缺省情况下，OpenFlow实例对应的VLAN允许MAC地址学习

## 7. 配置禁止OpenFlow实例上送Controller的端口类型

表1-15 配置禁止 OpenFlow 实例上送 Controller 的端口类型

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置禁止OpenFlow实例上送Controller的端口类型	<b>forbidden port { vlan-interface   vsi-interface } *</b>	缺省情况下，OpenFlow实例中所有接口都上报Controller

## 8. 配置Table Miss表项的缺省动作

使用 **active instance** 命令激活 OpenFlow 实例后，且 Controller 没有下发动作前，Table Miss 表项的缺省动作为丢弃报文，配置该功能后，Table Miss 表项的缺省动作修改为允许报文进行正常转发。

表1-16 配置 Table Miss 表项的缺省动作

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置Table Miss表项的缺省动作	<b>default table-miss permit</b>	缺省情况下, Table Miss表项的缺省动作为丢弃报文

## 9. 配置OpenFlow实例的Datapath ID

OpenFlow 网络中, 每个 OpenFlow 实例都使用唯一的 Datapath ID 来标识本实例。缺省情况下, OpenFlow 实例的 Datapath ID 由实例 ID 和设备桥 MAC 组成, 用户可以配置 Datapath ID。

表1-17 配置 OpenFlow 实例的 Datapath ID

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置OpenFlow实例的 Datapath ID	<b>datapath-id</b> <i>datapath-id</i>	缺省情况下, OpenFlow实例的Datapath ID 由实例ID和设备桥MAC组成, 前16位为实例ID, 后48位为设备桥MAC

### 1.3.3 激活OpenFlow实例

此功能用于激活实例。如果实例已经与控制器建立了连接, 修改 OpenFlow 实例的基本能力配置后, 需要重新激活该 OpenFlow 实例, 此时, OpenFlow 实例会断开与所有控制器的连接, 清除已经下发的流表, 根据当前的配置更新 OpenFlow 实例的能力集, 重新与控制器建立连接。



注意

重新激活 OpenFlow 实例会导致流量中断, 请谨慎使用。

表1-18 激活 OpenFlow 实例

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
激活OpenFlow实例	<b>active instance</b>	缺省情况下, 未激活OpenFlow实例

## 1.4 配置连接控制器

一个OpenFlow交换机可以与多个Controller建立连接，初始连接时，多个Controller的角色相同，权限相同，Controller可以通过OpenFlow消息设置本Controller的角色，各种角色的权限如 [表 1-19](#) 所示。

表1-19 Controller 角色

角色	权限
Master	处于该角色的 Controller 拥有全部权限，可以下发流表项，查询统计信息，接收设备上报的状态信息，在多个 Controller 中仅能有一个 Controller 是 Master 角色
Equal	处于该角色的 Controller 同样拥有全部权限，相对于 Master 角色，唯一不同的是可以有多个 Controller 处于 Equal 角色
Slave	处于该角色的 Controller 仅拥有部分权限，Controller to switch 消息中不能下发流表项，Group 表项以及 Meter 表项，不允许修改接口配置和设备配置，不允许执行 Packet Out 操作。异步消息中，默认情况下设备不会上送 Flow Remove 消息和 Packet In 消息，仅能上送接口状态变化消息，但是异步消息的上送能力可以通过 Controller 的设置异步消息进行修改

### 1.4.1 配置主连接

一个 OpenFlow 实例可以连接多个控制器，但仅允许与一个控制器建立主连接，用于控制消息的处理（下发流表项、获取数据、信息上报等），需要使用 TCP/SSL 保持可靠的连接。

表1-20 配置主连接

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance <i>instance-id</i></b>	-
配置主连接	<b>controller <i>id</i> address { ip <i>ip-address</i>   ipv6 <i>ipv6-address</i> } [ port <i>port-number</i> ] [ local address { ip <i>ip-address</i>   ipv6 <i>ipv6-address</i> } [ port <i>port-number</i> ] ] [ ssl <i>ssl-policy-name</i> ] [ vrf <i>vrf-name</i> ]</b>	缺省情况下，没有配置主连接 最多支持配置64个控制器

### 1.4.2 配置连接中断模式

一旦 OpenFlow 实例与所有控制器断开连接，则 OpenFlow 实例必须进入连接中断模式，模式分为两种：

- **Secure 模式**：连接断开后，交换机根据流表项转发。不主动删除控制器下发的表项，而是等待表项超时后进行删除，一旦连接建立成功，未超时的表项依然存在。
- **Standalone 模式**：连接断开后，交换机正常转发。

如果交换机与控制器重新连接成功，则继续作为 OpenFlow 设备根据流表项进行转发。

表1-21 配置连接中断模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置连接中断模式	<b>fail-open mode</b> { <b>secure</b>   <b>standalone</b> }	缺省情况下，连接中断模式为Secure模式

## 1.5 配置OpenFlow定时器

控制器和交换机都会发送 Echo request/reply 报文，用于验证连接是否正常。

连接检测定时器用于定义发送 Echo request 报文时间间隔，当超过三次 Echo request 报文发送并且没有收到 Echo reply 报文，则 OpenFlow 实例与控制器的连接断开。

OpenFlow 实例与控制器重连的时间间隔定义 OpenFlow 实例与控制器断开连接后再次开始重新连接的时间间隔。

表1-22 配置 OpenFlow 定时器

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置发送Echo request报文的时间间隔	<b>controller echo-request interval</b> <i>interval-value</i>	缺省情况下，发送Echo request报文的时间间隔为5秒 设备CPU负担较大的情况下，请配置较大的发送Echo request报文的时间间隔
配置OpenFlow实例与控制器重连的时间间隔	<b>controller connect interval</b> <i>interval-value</i>	缺省情况下，OpenFlow实例与控制器重连的时间间隔为60秒

## 1.6 配置支持动态MAC地址

此功能仅在支持 MAC-IP 流表情况下，决定是否支持控制器在查询或者删除流表项时包含动态 MAC 地址。

表1-23 配置支持动态 MAC 地址

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
配置支持动态MAC地址	<b>mac-ip dynamic-mac aware</b>	缺省情况，不支持动态MAC地址

## 1.7 开启OpenFlow的无丢包模式



说明

- 在非 OpenFlow 场景中，请不要开启 OpenFlow 的无丢包模式，否则会影响转发效率和能力级匹配。
- 要使配置生效，必须在配置后重启设备。在重启设备前，请保存当前配置。

在 OpenFlow 场景中，OpenFlow 的流表下发过程中设备可能会出现丢包，从而引发很多问题，比如流量误上送 Controller，导致误下发 OpenFlow 表项等，此时需要开启 OpenFlow 的无丢包模式。在无丢包模式下，设备不会丢包，OpenFlow 在实际网络中可以正常使用，但是匹配能力会受限制，比如不能匹配 IPv6 地址。

表1-24 开启 OpenFlow 的无丢包模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启OpenFlow的无丢包模式	<b>openflow lossless enable</b>	缺省情况下，没有开启OpenFlow的无丢包模式

## 1.8 配置QINQ模式

配置本功能后，带有双层 VLAN Tag 的报文通过 Extensibility 表之后，还是会带双层 VLAN Tag。

表1-25 配置 QINQ 模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance instance-id</b>	-
配置QINQ模式	<b>qinq-network enable</b>	缺省情况下，没有配置QINQ模式，带有双层VLAN Tag的报文通过 Extensibility表之后，只带一个内层的VLAN Tag

## 1.9 关闭流表项变化成功后打印日志的开关

配置本功能后，流表项变化成功后不再打印日志，其它情况仍然继续打印日志。

表1-26 关闭流表项变化成功后打印日志的开关

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
关闭流表项变化成功后打印日志的开关	<b>flow-log disable</b>	缺省情况，流表项变化成功后打印日志的开关处于开启状态

## 1.10 刷新OpenFlow实例下所有MAC-IP表的三层表项

某些情况下，MAC-IP表的三层表项会被覆盖，此时需要手工执行本命令刷新MAC-IP表，重新从控制器获取新的三层表项。

表1-27 刷新OpenFlow实例下所有MAC-IP表的三层表项

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入OpenFlow实例视图	<b>openflow instance</b> <i>instance-id</i>	-
刷新OpenFlow实例下所有MAC-IP表的三层表项	<b>refresh ip-flow</b>	-

## 1.11 OpenFlow显示和维护

在完成上述配置后，在任意视图下执行**display**命令可以显示配置后OpenFlow的运行情况，以及控制器下发的流表项。

表1-28 OpenFlow显示和维护

操作	命令
显示OpenFlow实例的详细信息	<b>display openflow instance</b> [ <i>instance-id</i> ]
显示OpenFlow实例的流表信息	<b>display openflow instance</b> <i>instance-id</i> <b>flow-table</b> [ <i>table-id</i> ]
显示OpenFlow实例的控制器信息	<b>display openflow instance</b> <i>instance-id</i> <b>controller</b>
显示OpenFlow实例的Group表信息	<b>display openflow instance</b> <i>instance-id</i> <b>group</b> [ <i>group-id</i> ]
显示OpenFlow实例的Meter表信息	<b>display openflow instance</b> <i>instance-id</i> <b>meter</b> [ <i>meter-id</i> ]
显示OpenFlow实例的概要信息	<b>display openflow summary</b>

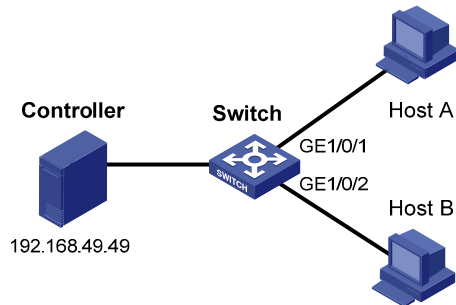
## 1.12 OpenFlow典型配置举例

### 1. 组网需求

- 创建 OpenFlow 实例 1，把 VLAN 4092 和 4094 映射到 OpenFlow 实例中 1，并激活实例。
- 使用 VLAN 1 作为 OpenFlow 实例 1 与控制器通信的 VLAN。
- 配置 OpenFlow 实例 1 连接的控制器，用来控制器能够控制 OpenFlow 实例 1 上的流量转发。

### 2. 组网图

图1-7 OpenFlow 配置组网图



### 3. 配置步骤

#### (1) 配置 VLAN

# 创建 VLAN 4092 和 4094。

```
<Switch> system-view
[Switch] vlan 4092
[Switch-vlan4092] port gigabitethernet 1/0/1
[Switch-vlan4092] quit
[Switch] vlan 4094
[Switch-vlan4094] port gigabitethernet 1/0/2
[Switch-vlan4094] quit
```

#### (2) 创建实例并映射 VLAN

```
[Switch] openflow instance 1
[Switch-of-inst-1] classification vlan 4092 mask 4093 loosen
```

#### (3) 配置控制器 1 的 IP 地址为 192.168.49.49，并激活实例

```
[Switch-of-inst-1] controller 1 address ip 192.168.49.49
[Switch-of-inst-1] active instance
```

#### (4) 验证配置

# 显示实例详细信息。

```
[Switch-of-inst-1] display openflow instance 1
Instance 1 information:
```

Configuration information:

Description : --

Active status : Active

Inactive configuration:



```

None
Active configuration:
Classification VLAN, loosen mode, total VLANs(2)
  4092, 4094
In-band management VLAN, total VLANs(0)
  Empty VLAN
Connect mode: Multiple
Mac address learning: Enabled
Flow table:
  Table ID(type): 0(Extensibility), count: 0
Flow-entry max-limit: 65535
Datapath ID: 0x00010cda415e232e
Default table-miss: Drop
Forbidden port: None
Port information:
  GigabitEthernet1/0/1
  GigabitEthernet1/0/2
Active channel information:
  Controller 1 IP address: 192.168.49.49 port: 6633

```

## 1.13 附录 A 应用限制

### 1.13.1 流表项限制

#### 1. 流表项匹配限制

##### (1) VLAN 匹配

配置 OpenFlow 实例对应的 VLAN 后，对 VLAN 的匹配存在一定的限制。

表1-29 VLAN 匹配项列表

VLAN	MASK	匹配的报文
-	-	匹配OpenFlow实例对应VLAN内的所有报文
0	-	匹配无VLAN tag的报文，但是入接口的PVID必须是在OpenFlow实例对应的VLAN范围内
0	有值	不支持
有效VLAN	-/有值	不支持
0x1000	-/非0x1000的值	不支持
0x1000	0x1000	匹配有VLAN tag的报文，但是该VLAN tag必须在OpenFlow实例内
有效VLAN   0x1000	-/有值	根据VLAN+MASK进行匹配，前提是VLAN+MASK必须在OpenFlow实例内
其它	其它	不支持

##### (2) 协议报文的匹配

对于协议报文，相关的协议配置使能后，协议报文不会进入OpenFlow转发处理，仍然由相关协议进行处理，对于LLDP报文的使用限制请参见 [1.13.5 匹配LLDP报文的限制](#)。

### (3) MetaData 的匹配

MetaData 用于在流表间传递匹配信息，非第一级流表支持下发 MetaData 的匹配，如果 Controller 在第一级流表下发了 MetaData 的匹配项，交换机返回不支持。

## 2. Instruction的限制

### (1) Clear actions 的限制

- 单级流表的情况下，支持 Clear actions。
- 多级流表的情况下，设备不支持 Clear actions 与其它 instruction 的动作配合，仅支持单独下发 Clear Actions。

### (2) Apply actions 的限制

不支持Action List中包含多个Output，仅支持一个Output，请参见 [1.13.2 Action List和Action Set 整合的限制](#)。

### (3) Write MetaData

在非最后一级流表的情况下，交换机支持 Write MetaData 的操作，否则交换机返回不支持。

### (4) Go To Table

非最后一级流表的情况下，交换机支持 Go To Table 的操作，否则交换机返回不支持。

## 1.13.2 Action List和Action Set整合的限制

OpenFlow Switch 设备整合 Action Set 和 Action List 为 Action Set，其整合原则如下。

### 1. 非Output Action

Action List 和 Action Set 中的 Action(除 Output 和 Group 外)如果不存在冲突，则全部保留为 Action Set; 如果存在冲突，则以 Action Set 的动作替换 Action List 中的动作 (其原因是 Action List 要执行在 Action Set 之前)。

### 2. Output Action

- 当 Action List 和 Action Set 中都存在一个 Output 的 Action 时，Action List 中的 Output 发送的报文不会对报文进行任何修改，其执行顺序最优，Action Set 中的 Output 会执行 Action List 和 Action Set 中的所有修改。
- 当 Action List 和 Action Set 中仅存在一个 Output 的 Action 时，该 Output 为报文出接口，执行顺序按照 Action Set 的顺序。
- 当 Action List 中存在一个 Output 的 Action，Action Set 中存在一个 Group 的 Action (Output 的 Action 存在与否都可以)时，Action List 中的 Output 发送的报文不会对报文进行任何修改，Group 在 Action Set 中。

### 1.13.3 Packet Out的处理限制

#### 1. 入接口限制

在 Packet out 消息中 Output 为 Normal、Local、In port 或 Controller 时，入接口只能是设备上的物理接口或者逻辑接口，不能是 OpenFlow 保留接口。

#### 2. 报文ID和报文内容同时存在的处理

在 Packet Out 消息中如果同时存在报文 ID 和报文，交换机只会获取报文 ID 对应的缓存报文进行处理，忽略消息中携带的报文。

#### 3. 无VLAN Tag报文的处理

- 如果 Packet Out 消息中的报文没有 VLAN tag，则交换机将入接口的 PVID 做为报文所在的 VLAN 进行转发处理。
- 如果入接口不是设备上的接口，且 Output 是设备上的接口，则使用出接口的 PVID 发送。
- 如果入接口不是设备上的接口，且Output是保留口Flood和All，请参见 [4. 出接口限制](#)。

#### 4. 出接口限制

Packet Out 中指定的 Output 为保留接口的 Flood 和 All 时的处理机制。

##### (1) 出接口为 Flood 的情况

- Packet Out 的报文携带 VLAN tag，则报文在该 VLAN 内广播。
- Packet Out 的报文没有携带 VLAN tag，但是入接口是 OpenFlow Swtich 上的接口，则报文在入接口的 PVID 内广播。
- Packet Out 的报文没有携带 VLAN tag，并且入接口是 Controller，则报文在所有 OpenFlow 接口发送。

##### (2) 出接口为 All 的情况

- Packet Out 的报文携带 VLAN tag，则报文在该 VLAN 内广播。
- Packet Out 的报文没有携带 VLAN tag，报文都在所有 OpenFlow 接口发送。

### 1.13.4 Packet in的处理限制

#### 1. 报文VLAN tag处理

对于 Packet in 消息中的报文：

- 上送时如果其 VLAN tag 与该报文入接口的 PVID 相同，那么该报文的 VLAN tag 会被删除。
- 上送时如果其 VLAN tag 与该报文入接口的 PVID 不同，那么该报文的 VLAN tag 不会被删除。

#### 2. 报文缓存限制

- 对于上送原因是 No Match 的报文支持缓存，缓存大小是 1K 个报文。
- 对于其它上送原因的报文不支持缓存，整个报文都会被上送，并且 Cookie 是全 F。

### 1.13.5 匹配LLDP报文的限制

OpenFlow 网络可以通过 LLDP 发现拓扑，对于匹配 LLDP 报文存在如下限制：

- 设备上必须全局使能 LLDP。

- OpenFlow 实例内的接口上关闭 LLDP 功能。
- LLDP 报文匹配后上送 Controller 不受实例限制，只要收到 LLDP 报文的接口在 OpenFlow 实例内，并且实例内有匹配 LLDP 报文并上送 Controller 的表项，则 LLDP 报文就会在该实例上送 Controller。

### 1.13.6 Flow Mod的限制

#### 1. Table Miss表项的添加、修改和删除

- OpenFlow 实例在激活后会默认生成 Table Miss 表项，其动作是 Drop，默认表项不能够被 Controller 通过 Modify 的动作修改，不能被 Controller 通过 Multipart 消息查询到，仅能由 Controller 通过 Add 进行添加 Table Miss 的动作进行修改。
- Table Miss 表项仅能通过严格匹配进行修改和删除，在非严格匹配的情况下，即使匹配项是通配也不能够操作 Table Miss 表项。
- Table Miss 表项被删除后，会生成默认的 Table Miss 表项，其动作是 Drop。

#### 2. 普通表项的添加、修改和删除

- 在非严格匹配的情况下，不支持通过 match 域为通配修改所有普通流表项。

## 1.14 附录 B MAC-IP流表

### 1.14.1 MAC-IP流表支持能力

表1-30 MAC-IP 流表二层表项支持能力

支持项	能力
匹配项	<ul style="list-style-type: none"> <li>• VLAN</li> <li>• 单播目的 MAC 地址</li> </ul>
必选动作项	指定出接口
可选动作项	<ul style="list-style-type: none"> <li>• Go to table（在存在多级流表的情况下，即使 Controller 不下发表项，交换机默认下发该动作）</li> <li>• Write Meta（在存在多级流表的情况下，即使 Controller 不下发表项，交换机默认下发目的 MAC 匹配的 MetaData）</li> </ul>

表1-31 MAC-IP 流表三层表项支持能力

支持项	能力
必选匹配项	<ul style="list-style-type: none"> <li>• VLAN</li> <li>• 单播目的 MAC 地址（必须是匹配 VLAN 的虚接口的 MAC 地址）</li> <li>• 单播目的 IP 地址</li> </ul>
可选匹配项	无

支持项	能力
必选动作项	<ul style="list-style-type: none"> <li>指定出接口</li> <li>修改 VLAN</li> <li>修改目的 MAC 地址</li> </ul>
可选动作项	<ul style="list-style-type: none"> <li>修改源 MAC 地址（源 MAC 地址会修改为目的出接口所在 VLAN 虚接口的 MAC 地址）</li> <li>TTL 减 1</li> <li>Go to table（在多级流表存在的情况下，即使 Controller 不下发，Switch 上默认下发该动作）</li> <li>Write Meta（在多级流表存在的情况下，即使 Controller 不下发，Switch 上默认下发目的 IP 地址匹配的 MetaData）</li> </ul>

### 1.14.2 MAC-IP流表的限制

MAC-IP 流表中，Controller 需要遵循如下限制下发表项，否则可能会造成转发错误。

表1-32 MAC-IP 流表二层表项限制

表项类型	限制
匹配项限制	目的 MAC 地址不是本机 MAC 地址
动作项限制	出接口属于匹配的 VLAN

三层表项的限制如 [表 1-33](#) 所示。

表1-33 MAC-IP 流表三层表项限制

表项类型	限制
匹配项限制	<ul style="list-style-type: none"> <li>匹配的 VLAN 所对应的 VLAN 虚接口 UP</li> <li>目的 MAC 地址是匹配 VLAN 的 MAC 地址</li> <li>目的 IP 地址不是本机 IP 地址</li> </ul>
动作项限制	<ul style="list-style-type: none"> <li>指定出接口属于目的 VLAN</li> <li>目的 MAC 地址不是本机 MAC 地址</li> <li>如果修改源 MAC 地址，源 MAC 地址必须是目的出接口所在 VLAN 虚接口的 MAC 地址</li> </ul>



说明

三层表项能够下发的前提是匹配 VLAN 所对应的 VLAN 虚接口存在并且处于 UP 状态，且 VLAN 虚接口做为 OpenFlow 接口上报（包括了 VLAN 虚接口的链路状态状态和 MAC 地址），在 VLAN 虚接口删除时同时也会上报给 Controller，需要由 Controller 删除对应的三层流表项，因此需要 Controller 保证三层表项的正确性，Switch 端不对三层表项的匹配项限制进行检查。

### 1.14.3 MAC-IP流表的Table Miss

MAC-IP 流表的 Table Miss 支持下列 Output Action:

- Go To Table: 进入下一级流表;
- Drop: 丢弃报文;
- Controller: 报文上送 Controller;
- Normal: 报文正常转发。

### 1.14.4 Dynamic aware

在支持 MAC-IP 流表的情况下, 接机支持 Controller 获取和删除动态 MAC 地址表项。Controller 可以通过指定 VLAN、单播 MAC 或者单播 MAC 与 VLAN 来获取和删除动态 MAC 地址表项。

### 1.14.5 MAC-IP Table与Extensibility Table的配合

#### 1. MetaData/Mask

MAC-IP Table 和 Extensibility Table 通过 MetaData 可以实现多级流表。

MAC-IP Table 支持 Write MetaData, Extensibility 支持 Match MetaData。

MetaData Mask每个Bit表示不同的含义, MetaData中对应的Bit位置位表示匹配, 未置位表示通配, 具体参见 [表 1-34](#)。

表1-34 MetaData Mask 含义

MetaData Mask Bit	含义	MetaData
Bit 0	目的 MAC	1, 置位, 表示匹配到目的 MAC
		0, 未置位, 表示未匹配到目的 MAC
Bit 1	源 MAC	1, 置位, 表示匹配到源 MAC
		0, 未置位, 表示未匹配到源 MAC
Bit 2	目的 IP	1, 置位, 表示匹配到目的 IP
		0, 未置位, 表示未匹配到目的 IP
其他	保留	保留

#### 2. 配合限制

Extensibility 表和 MAC-IP 表配合时:

- 当 Extensibility 表中的 Output 动作不是 Normal 时, MAC-IP 表不会生效, 所有动作根据 Extensibility 表进行处理;
- 当 Extensibility 表中的 Output 动作是 Normal 时, Output 动作根据 MAC-IP 表进行处理, 其余动作根据 Extensibility 表进行处理。