

# H3C S5130S-SI[LI]&S5120V2-SI[LI]&S5110V2-SI& S5000V3-EI&S5000E-X&S3100V3-SI 系列以太网交换机

## OpenFlow 配置指导

新华三技术有限公司  
<http://www.h3c.com>

资料版本：6W103-20190822  
产品版本：Release 612x 系列

**Copyright © 2019 新华三技术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。

除新华三技术有限公司的商标外，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。

# 前言

本配置指导主要介绍 OpenFlow 功能的原理及具体配置。

前言部分包含如下内容：

- [读者对象](#)
- [本书约定](#)
- [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. 示例约定

由于设备型号不同、配置不同、版本升级等原因，可能造成本手册中的内容与用户使用的设备显示信息不一致。实际使用中请以设备显示的内容为准。

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。

## 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: [info@h3c.com](mailto:info@h3c.com)**

感谢您的反馈，让我们做得更好！

# 目 录

1 OpenFlow .....	1-1
1.1 OpenFlow简介 .....	1-1
1.1.1 OpenFlow网络构成 .....	1-1
1.1.2 OpenFlow Switch类型 .....	1-1
1.1.3 OpenFlow接口 .....	1-1
1.1.4 OpenFlow实例 .....	1-2
1.1.5 OpenFlow流表 .....	1-2
1.1.6 Group Table .....	1-4
1.1.7 Meter Table .....	1-5
1.1.8 OpenFlow channel .....	1-5
1.1.9 控制器 .....	1-7
1.1.10 协议规范 .....	1-8
1.2 OpenFlow配置任务简介 .....	1-8
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实例的类型 .....	1-9
1.3.3 配置流表和流表项 .....	1-9
1.3.4 配置控制器连接模式 .....	1-10
1.3.5 配置禁止上送控制器的端口类型 .....	1-10
1.3.6 激活OpenFlow实例 .....	1-11
1.3.7 配置OpenFlow实例的属性 .....	1-11
1.4 配置连接控制器 .....	1-12
1.5 配置OpenFlow实例作为SSL服务器监听控制器 .....	1-13
1.6 刷新MAC-IP流表的三层表项 .....	1-13
1.7 使用OpenFlow关闭接口 .....	1-14
1.8 OpenFlow显示和维护 .....	1-14
1.9 OpenFlow典型配置举例 .....	1-15
1.9.1 VLAN类型OpenFlow典型配置举例 .....	1-15
1.10 附录 A 应用限制 .....	1-16
1.10.1 Flow Entry的限制 .....	1-16
1.10.2 Action List和Action Set整合的限制 .....	1-17
1.10.3 Packet Out的处理限制 .....	1-18
1.10.4 Packet in的处理限制 .....	1-18

1.10.5 匹配LLDP报文的限制 .....	1-19
1.10.6 Flow Mod的限制 .....	1-19
1.11 附录 B MAC-IP流表 .....	1-19
1.11.1 MAC-IP流表支持能力 .....	1-19
1.11.2 MAC-IP流表的限制 .....	1-20
1.11.3 MAC-IP流表的Table Miss .....	1-21
1.11.4 Dynamic aware .....	1-21
1.11.5 MAC-IP流表与Extensibility流表的配合 .....	1-21

# 1 OpenFlow

## 1.1 OpenFlow简介

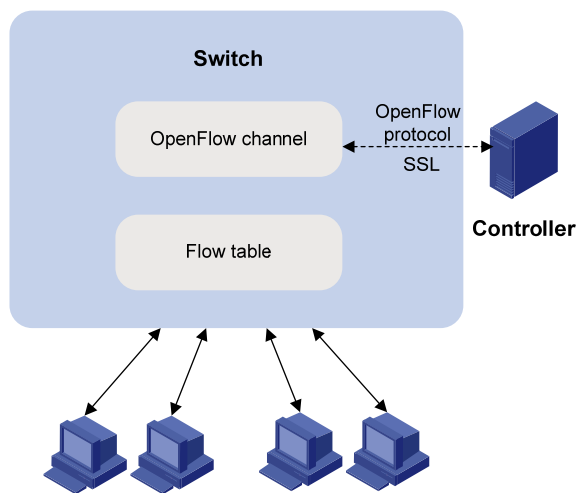
OpenFlow 是 SDN（Software Defined Network，软件定义网络）架构中定义的一个控制器与转发层之间的通信接口标准。OpenFlow 允许控制器直接访问和操作网络设备的转发平面，这些网络设备可能是物理上的，也可能是虚拟的。

OpenFlow 的思想是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。

### 1.1.1 OpenFlow网络构成

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



### 1.1.2 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。

### 1.1.3 OpenFlow接口

OpenFlow 接口有如下类型：

- 物理接口：比如以太网接口。可以作为入接口和出接口。



- 逻辑接口：比如聚合接口。可以作为入接口和出接口。
- 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，其他接口都可以作为出接口，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
In Port	报文从入接口转发
Any	接口通配描述，不能作为入接口以及出接口
Local	报文上送本地CPU
Normal	报文正常转发
Flood	报文广播发送

### 1.1.4 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，根据控制器下发的流表项指导流量转发。

下文如果没有特殊说明，交换机指的就是一个 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 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。激活配置后，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 配置的映射 VLAN 后，该接口才是此 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。

### 1.1.5 OpenFlow流表

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。

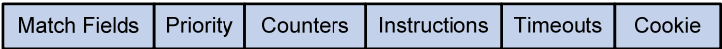
1. 流表类型

流表类型如下：

- **MAC-IP流表：**通过MAC地址表和FIB表实现。只能匹配目的MAC地址、VLAN以及目的IP地址，动作也仅支持修改目的MAC地址、源MAC地址、VLAN以及指定出接口。具体请参见 [1.11 附录 B MAC-IP流表](#)。
- **Extensibility 流表：**扩展流表，使用 ACL 实现。

2. 流表项组成

图1-2 流表项结构



流表项如 [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，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能存在一个，并且按照 [表 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，但是会立即修改报文的内容，其效果是累加的，并且执行顺序是按照下发的顺序执行的。

表1-2 动作指令集定义

Instruction	处理
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
Write-Actions	更改动作集中的所有动作
Goto-Table	进入下一级流表

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，报文被丢弃。通常空指令集、空动作集或执行清空动作集后，报文被丢弃
Group	必选	将报文转交给Group处理，该动作的含义由Group的类型定义

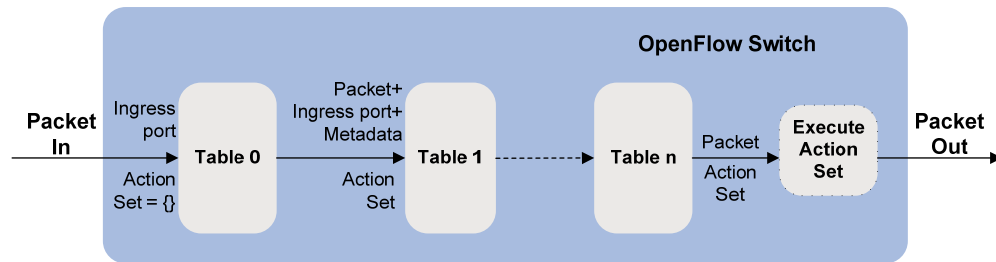
动作名称	可选/必选	描述
Set-Queue	可选	为报文指定队列ID。当报文被转发到特定端口时，队列ID通常被用于基本QoS
Set-Field	可选	识别报文字段的类型，并且可以修改该字段的值。通常只适用于最外层的字段（例如当内外层均有VLAN tag时，该动作只修改最外层的VLAN Tag）

- **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。

### 3. 流表处理流程

如 图 1-3 所示，当报文进入交换机后，必须从流表ID最小的流表开始依次匹配；流表可以按次序从小到大越级跳转，但不能从某一流表向前跳转至流表ID更小的流表。一旦在某个流表匹配到后，会更新此报文的动作集（允许被下一级流表覆盖），到了最后一个流表后，所有的动作会被执行，此时报文的内容会被修改，指定出接口进行转发。如果在某个流表处理时，处理指令包含 **Action List**，则报文的一份拷贝立即执行 **Action List** 而不需要在最后一个流表处理结束后再执行。

图1-3 OpenFlow 转发示意图



### 4. 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，该表项的匹配域为通配，即匹配任何报文，优先级为 **0**，动作指令与正常表项相同。

#### 1.1.6 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。

### 1.1.7 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。

### 1.1.8 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 **Controller to Switch** 消息、异步消息和同步消息，每种报文类型都有很多子类型。

#### 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，用来查询交换机的消息，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

子类型	描述
Features	用于控制器发送请求来了解交换机的能力，交换机必须回应该报文
Configuration	用于控制器设置，查询交换机的配置，交换机只有在控制器查询时回应

子类型	描述
Modify-State	用于管理交换机的状态，如流表项和端口状态。该命令主要用于增加、删除、修改、交换机内的流表项，组表项以及交换机端口的属性
Multipart	用于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 <b>Packet-In</b> 消息触发，也可以通过控制器直接发送。通常 <b>Packet-Out</b> 消息包含整个之前接收到的 <b>Packet-In</b> 消息所携带的报文或者 <b>buffer ID</b> （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 <b>Packet-Out</b> 消息所携带的报文执行该动作列表。如果动作列表为空， <b>Packet-Out</b> 消息所携带的报文将被交换机丢弃
Barrier	用于确认之前下发动作是否成功。控制器发送 <b>Barrier</b> 请求消息，当交换机确认之前下发的流表等操作都已经成功时会回复 <b>Barrier</b> 应答消息
Role-Request	用于设定或查询 <b>OpenFlow channel</b> 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某些异步事件的消息。例如，当某一条规则因为超时而被删除时，交换机将自动发送一条 **Flow-Removed** 消息通知控制器，以方便控制器作出相应的操作，如重新设置相关规则等。

表1-5 异步消息

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 <b>Table Miss</b> 后转发到保留端口 <b>Controller</b> 端口的报文均要通过 <b>Packet-in</b> 消息送到控制器。也有部分其他流程，如 <b>TTL</b> 检查等，也需要通过该消息和控制器交互。 <b>Packet-In</b> 既可以携带报文，也可以通过在交换机内部设置报文的 <b>Buffer</b> 来仅携带报文头以及其 <b>Buffer ID</b> 传输给控制器。控制器在接收到 <b>Packet-In</b> 消息后会对其接收到的报文或者报文头和 <b>Buffer ID</b> 进行处理，并返回 <b>Packet-out</b> 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，就是双向对称的消息，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

子类型	描述
Hello	当连接启动时交换机和控制器会发送 <b>Hello</b> 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 <b>Echo request/reply</b> 消息，而且对于接收到的 <b>Echo request</b> 消息必须能返回 <b>Echo reply</b> 消息。 该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

子类型	描述
Experimenter	为将来新加入的特性预留的消息

#### 4. OpenFlow定时器

OpenFlow 中包含连接检测定时器和重连定时器，作用如下：

- 连接检测定时器：定义发送 **Echo request** 报文时间间隔，设备发送三次 **Echo request** 报文，仍然没有收到 **Echo reply** 报文，则交换机断开与控制器的连接。
- 重连定时器：定义交换机与控制器断开连接后，下次开始重新连接的时间间隔。

### 1.1.9 控制器

#### 1. 控制器角色

一个OpenFlow交换机可以与多个控制器建立连接，初始连接时，多个控制器的角色相同，权限相同，控制器可以通过OpenFlow消息设置本控制器的角色，各种角色的权限如 [表 1-7](#) 所示。

表1-7 控制器角色

角色	权限
Master	处于该角色的控制器拥有全部权限，可以下发流表项，查询统计信息，接收设备上报的状态信息，在多个控制器中仅能有一个控制器是Master角色
Equal	处于该角色的控制器同样拥有全部权限，相对于Master角色，唯一不同的是可以有多个控制器处于Equal角色
Slave	处于该角色的控制器仅拥有部分权限，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，缺省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控制器的设置异步消息进行修改

#### 2. 控制器连接模式

交换机与多个控制器建立连接时，连接分为 **Single** 模式和 **Multiple** 模式：

- **Multiple** 模式：同一时刻可以与多个控制器建立连接。交换机会同时连接实例内所有的控制器，在与某个控制器连接失败或者断开连接时，在重连定时器超时后重新进行连接，直到连接成功。
- **Single** 模式：同一时刻仅与一个控制器建立连接，其他控制器作为备份。当且仅当当前的连接断开后，交换机会连接下一个控制器，直到连接成功。

#### 3. 主连接和辅助连接

交换机与每个控制器建立的连接中，可包含一个主连接和多个辅助连接：

- 主连接：一般用于控制消息的处理（下发流表项、获取数据、信息上报等），使用 **TCP/SSL** 保持可靠的连接。
- 辅助连接：用于提高控制器和 **OpenFlow** 交换机的通信能力。辅助连接的目的地址和端口号可以和主连接不一致。

## 4. 连接中断模式

如果交换机与所有控制器断开连接，则交换机进入连接中断模式，连接中断模式分为两种：

- **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。匹配的流表项中执行 **output controller** 动作转发的流量被丢弃。创建 **OpenFlow** 实例时，缺省为 **Secure** 模式。
- **Smart 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。匹配的流表项中执行 **output controller** 动作转发的流量进行正常转发。
- **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。

### 1.1.10 协议规范

- **OpenFlow Switch Specification Version 1.3.3**

## 1.2 OpenFlow配置任务简介

OpenFlow 配置任务如下：

- (1) [配置OpenFlow实例](#)
  - a. [创建OpenFlow实例](#)
  - b. [配置OpenFlow实例的类型](#)
  - c. （可选）[配置流表和流表项](#)
  - d. （可选）[配置控制器连接模式](#)
  - e. （可选）[配置禁止上送控制器的端口类型](#)
  - f. [激活OpenFlow实例](#)
  - g. （可选）[配置OpenFlow实例的属性](#)
- (2) [配置连接控制器](#)
- (3) （可选）[配置OpenFlow实例作为SSL服务器监听控制器](#)
- (4) （可选）[刷新MAC-IP流表的三层表项](#)
- (5) （可选）[使用OpenFlow关闭接口](#)

## 1.3 配置OpenFlow实例

### 1.3.1 创建OpenFlow实例

- (1) 进入系统视图。  
**system-view**
- (2) 创建 OpenFlow 实例，并进入 OpenFlow 实例视图。  
**openflow instance instance-id**
- (3) （可选）配置 OpenFlow 实例的 Datapath ID。  
**datapath-id id**



缺省情况下，OpenFlow 实例的 Datapath ID 由实例 ID 与设备桥 MAC 组成，其中前 16 个比特为实例 ID，后 48 个比特为设备桥 MAC。

Datapath ID 用来在唯一标识 OpenFlow 实例，不同 OpenFlow 实例的 Datapath ID 不能相同。

- (4) （可选）配置 OpenFlow 报文的 DSCP 值。

```
tcp dscp dscp-value
```

缺省情况下，未配置 OpenFlow 报文的 DSCP 值。

### 1.3.2 配置 OpenFlow 实例的类型

#### 1. 配置限制和指导

对于 VLAN 类型的 OpenFlow 实例，需要注意的是：

- 一个 VLAN 仅能属于一个 OpenFlow 实例，否则会导致流量无法正确处理。
- 如果配置的对应 VLAN 不存在，在激活实例时，设备将会自动创建该 VLAN。
- 同一个接口所属 VLAN 必须属于同一 OpenFlow 实例，否则，控制器下发的不同实例的接口状态消息会相互覆盖。
- VLAN 对应的 VLAN 接口不能配置 BFD MAD 检测功能，该功能的相关内容请参见“虚拟化技术配置指导”中的“IRF”。

对于带内管理 VLAN，需要注意的是：

- 带内管理 VLAN 内的流量进行正常转发，用于 OpenFlow 设备与控制器建立安全通道。
- 带内管理 VLAN 必须是 OpenFlow 实例配置的对应 VLAN 的子集。
- 仅在带内管理 VLAN 的接口不属于 OpenFlow 接口。

#### 2. 配置 OpenFlow 实例为 VLAN 类型

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 OpenFlow 实例视图。

```
openflow instance instance-id
```

- (3) 配置 OpenFlow 实例的类型。

```
classification vlan vlan-id [ mask vlan-mask ] [ loosen ]
```

缺省情况下，未配置 OpenFlow 实例的类型。

- (4) （可选）配置带内管理 VLAN。

```
in-band management vlan { vlan-id [ to vlan-id ] } &<1-10>
```

缺省情况下，未配置带内管理 VLAN。

### 1.3.3 配置流表和流表项

#### 1. 配置限制和指导

Extensibility 流表 ID 必须大于 MAC-IP 流表 ID。

#### 2. 配置步骤

- (1) 进入系统视图。



**system-view**

- (2) 进入 OpenFlow 实例视图。

**openflow instance** *instance-id*

- (3) 配置流表类型和流表 ID。

**flow-table** { **extensibility** *extensibility-table-id* | **mac-ip** *mac-ip-table-id* } \*

缺省情况下，流表类型为 Extensibility，流表 ID 为 0。

- (4) 配置 Extensibility 的流表项的最大值。

**flow-entry max-limit** *limit-value*

Extensibility 表的流表项的最大个数为 65535。

控制器下发的流表表项数量超过最大值时，向控制器返回失败。

- (5) 配置允许动态 ARP 表项覆盖 OpenFlow ARP 表项。

**precedence dynamic arp**

缺省情况下，禁止动态 ARP 表项覆盖 OpenFlow ARP 表项。

仅 MAC-IP 流表支持本功能。

- (6) 允许控制器下发的流表中包含聚合接口的成员端口。

**permit-port-type member-port**

缺省情况下，禁止控制器下发的流表中包含聚合接口的成员端口。

- (7) 配置 Table Miss 流表项的缺省动作为正常转发。

**default table-miss** **permit**

缺省情况下，Table Miss 流表项的缺省动作为丢弃。

### 1.3.4 配置控制器连接模式

- (1) 进入系统视图。

**system-view**

- (2) 进入 OpenFlow 实例视图。

**openflow instance** *instance-id*

- (3) 配置控制器连接模式。

**controller mode** { **multiple** | **single** }

缺省情况下，控制器连接模式为 Multiple 模式。

### 1.3.5 配置禁止上送控制器的端口类型

#### 1. 功能简介

配置该功能后，交换机不再向控制器上送对应 VLAN 接口的信息。

#### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入 OpenFlow 实例视图。

**openflow instance** *instance-id*

- (3) 配置禁止上送控制器的端口类型。

**forbidden port vlan-interface**

缺省情况下，未配置配置禁止上送控制器的端口类型，即所有接口类型都上送控制器。

### 1.3.6 激活OpenFlow实例

#### 1. 功能简介

新配置或修改 OpenFlow 实例后，需要激活实例使配置生效。重新激活实例时，交换机会断开与所有控制器的连接，清除已下发的流表，更新能力集，重新与控制器建立连接。

#### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入 OpenFlow 实例视图。

**openflow instance** *instance-id*

- (3) 激活 OpenFlow 实例。

**active instance**

缺省情况下，OpenFlow 实例处于未激活状态。

### 1.3.7 配置OpenFlow实例的属性

#### 1. 配置限制和指导

配置 OpenFlow 实例属性时，无需激活 OpenFlow 实例即可生效。

#### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入 OpenFlow 实例视图。

**openflow instance** *instance-id*

- (3) 配置 OpenFlow 实例的描述信息。

**description** *text*

缺省情况下，未配置 OpenFlow 实例的描述信息。

- (4) 配置 OpenFlow 定时器相关参数。

- 配置连接检测定时器。

**controller echo-request interval** *interval*

缺省情况下，连接检测定时器的值为 5。

- 配置重连定时器。

**controller connect interval** *interval*

缺省情况下，重连定时器的值为 60。

- (5) 配置 MAC 地址相关功能。

- 配置 OpenFlow 实例对应的 VLAN 禁止 MAC 地址学习。  
**mac-learning forbidden**  
 缺省情况下，OpenFlow 实例对应的 VLAN 允许 MAC 地址学习。  
 带内管理 VLAN 中配置的 VLAN 不受该功能限制。
- 配置匹配控制器查询或删除流表项指令中的动态 MAC 地址。  
**mac-ip dynamic-mac aware**  
 缺省情况下，忽略控制器查询或删除流表项指令中的动态 MAC 地址。  
 仅 MAC-IP 流表支持本功能。
- (6) 配置禁止 ARP 报文上送的控制器的配置。  
**forbidden packet-in arp controller controller-id-list**  
 缺省情况下，未配置禁止 ARP 报文上送的控制器的配置。  
 配置该功能后，交换机不再向指定控制器上送 ARP 报文，防止设备中的大量 ARP 报文冲击这些控制器。
- (7) 关闭流表项变化成功后打印日志的开关。  
**flow-log disable**  
 缺省情况下，流表项变化成功后打印日志的开关处于开启状态。
- (8) 开启 OpenFlow 环路保护功能。  
**loop-protection enable**  
 缺省情况下，OpenFlow 环路保护功能处于关闭状态。  
 取消激活 OpenFlow 实例后，OpenFlow 下发一条流表项，丢弃该实例对应 VLAN 内的所有流量，避免出现环路。

## 1.4 配置连接控制器

### 1. 配置限制和指导

辅助连接配置和主连接配置不做配置冲突检查，如果辅助连接与主连接配置冲突，则无法建立辅助连接。

### 2. 配置步骤

- (1) 进入系统视图。  
**system-view**
- (2) 进入 OpenFlow 实例视图。  
**openflow instance instance-id**
- (3) 配置主连接。  
**controller controller-id address { ip ipv4-address | ipv6 ipv6-address }  
 [ port port-number ] [ local address { ip local-ipv4-address | ipv6  
 local-ipv6-address } [ port local-port-number ] ] [ ssl ssl-policy-name ]**  
 源 IP 地址必须为 OpenFlow 实例中端口的 IP 地址，否则交换机和控制器之间无法建立连接。
- (4) （可选）配置辅助连接。

```
controller controller-id auxiliary auxiliary-id transport { tcp | udp |  
ssl ssl-policy-name } [ address { ip ipv4-address | ipv6 ipv6-address } ]  
[ port port-number ]
```

辅助连接中未配置 IP 地址或接口号时，则 IP 地址或接口号与主连接一致。

- (5) （可选）配置连接中断模式。

```
fail-open mode { secure | smart | standalone }
```

缺省情况下，连接中断模式为 Secure。

- (6) （可选）开启 OpenFlow 连接备份功能。

```
tcp-connection backup
```

缺省情况下，OpenFlow 连接备份功能处于开启状态。

开启连接备份功能后，设备发生主备倒换时，OpenFlow 实例仍然保持与控制器的连接。

仅基于 TCP 的 OpenFlow 连接支持开启连接备份功能。

## 1.5 配置OpenFlow实例作为SSL服务器监听控制器

### 1. 功能简介

未开启 SSL 服务器时，设备作为 TCP/SSL 客户端主动连接控制器（SSL 服务器）；启动 SSL 服务器之后，设备作为 SSL 服务器端被动等待控制器（SSL 客户端）连接。关于 SSL 的详细介绍，请参见“安全配置指导”中的“SSL”。

### 2. 配置限制和指导

配置该功能后，无需激活 OpenFlow 实例即可生效。

不能通过重复执行本命令修改 OpenFlow 实例启动的 SSL 服务器。如需修改，请先通过 **undo listening port** 命令删除 OpenFlow 实例启动的 SSL 服务器，再执行 **listening port** 命令重新启动 SSL 服务器。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 OpenFlow 实例视图。

```
openflow instance instance-id
```

- (3) 配置 OpenFlow 实例作为 SSL 服务器监听控制器。

```
listening port port-number ssl ssl-policy-name
```

缺省情况下，未配置 OpenFlow 实例作为 SSL 服务器监听控制器。

## 1.6 刷新MAC-IP流表的三层表项

### 1. 功能简介

某些情况下，MAC-IP 流表的三层表项会被覆盖，可以通过命令行手工刷新 MAC-IP 流表，使设备重新从控制器获取三层表项。

## 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入 OpenFlow 实例视图。

**openflow instance** *instance-id*

- (3) 刷新 MAC-IP 流表的三层表项。

**refresh ip-flow**

## 1.7 使用OpenFlow关闭接口

### 1. 功能简介

使用 OpenFlow 关闭接口后，通过 **display interface** 命令查看接口显示信息时，Current state 为 OFP DOWN。

使用 OpenFlow 关闭接口后，可以通过如下方式打开接口：

- **undo openflow shutdown** 命令打开接口。
- 通过控制器下发 PORT\_MOD 消息打开接口。

### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入接口视图视图。

**interface** *interface-type interface-number*

- (3) 使用 OpenFlow 关闭接口。

**openflow shutdown**

缺省情况下，未使用 OpenFlow 关闭接口。

## 1.8 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。

表1-8 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实例的控制器信息	<b>display openflow instance</b> <i>instance-id</i> { <b>controller</b> [ <i>controller-id</i> ]   <b>listened</b> }
显示OpenFlow实例的辅助连接信息	<b>display openflow instance</b> <i>instance-id</i> <b>auxiliary</b> [ <i>controller-id</i> [ <b>auxiliary</b> <i>auxiliary-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> ]

操作	命令
显示OpenFlow实例的概要信息	<b>display openflow summary</b>
清除控制器发送和接收报文的统计计数	<b>reset openflow instance <i>instance-id</i> { controller [ <i>controller-id</i> ]   listened } statistics</b>

## 1.9 OpenFlow典型配置举例

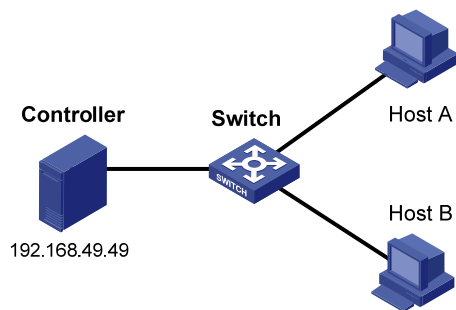
### 1.9.1 VLAN类型OpenFlow典型配置举例

#### 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。
- 配置 OpenFlow 实例 1 连接的控制器，用来控制 Switch 上的流量转发。

#### 2. 组网图

图1-7 VLAN 类型 OpenFlow 配置组网图



#### 3. 配置步骤

# 创建 VLAN 4092 和 4094。

```

<Switch> system-view
[Switch] vlan 4092
[Switch-vlan4092] quit
[Switch] vlan 4094
[Switch-vlan4094] quit
  
```

# 创建实例并映射 VLAN

```

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

# 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
[Switch-of-inst-1] quit
  
```

#### 4. 验证配置

# 显示实例详细信息。

```

[Switch] display openflow instance 1
  
```

```
Instance 1 information:

Configuration information:
  Description      : --
  Active status    : Active
  Inactive configuration:
    None
  Active configuration:
    Classification VLAN, 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: 0x0064001122000101
    Default table-miss: Permit
    Forbidden port: VLAN interface
    Qing Network: Disabled
    TCP connection backup: Enabled
  Port information:
    GigabitEthernet1/0/3
  Active channel information:
    Controller 1 IP address: 192.168.49.49 port: 6633
```

1.10 附录 A 应用限制

1.10.1 Flow Entry的限制

1. 匹配的限制

(1) VLAN 匹配

在使用 VLAN 划分 OpenFlow 实例的情况下，对 VLAN 的匹配存在一定的限制。

表1-9 VLAN 匹配项列表

VLAN	MASK	匹配的报文
-	-	匹配在OpenFlow实例VLAN内的所有报文
0	-	匹配无VLAN tag的报文，但是入接口的PVID必须在OpenFlow实例内
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

VLAN	MASK	匹配的报文
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
其他	其他	不支持

## (2) 协议报文的匹配

对于协议报文，一旦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，但是对于LLDP报文比较特殊，对于LLDP报文的使用限制请参见 [1.10.5](#)。

## (3) MetaData 的匹配

MetaData 用于流表间的匹配信息传递，在非第一级流表支持下发 MetaData 的匹配，如果控制器在第一级流表下发了 MetaData 的匹配项，Switch 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，仅第一级流表支持 Clear actions 与其他 instruction 的动作配合，后续流表仅支持单独下发 Clear Actions。

### (2) Apply actions 的限制

不支持Action List中包含多个Output的情况，仅支持一个Output时，请参见 [1.10.2](#)。

### (3) Write MetaData/MetaMask

在且仅在非最后一级流表的情况下，Switch 支持 Write MetaData/MetaMask 的操作，否则 Switch 返回不支持。

### (4) Go To Table

在且仅在非最后一级流表的情况下，Switch 支持 Go To Table 的操作，否则 Switch 返回不支持。

## 1.10.2 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。
- 其他情况不支持。

### 1.10.3 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 To 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 保留口。

#### 2. Buffer 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 Buffer ID 和报文，OpenFlow Switch 只会获取 Buffer 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 OpenFlow Switch 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.](#) 的处理。

#### 4. 出接口限制

Packet Out 中指定的 Output 为保留口的 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 Switch 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 接口发送一份。

##### (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，无论入接口是什么接口，报文都在所有 OpenFlow 接口发送一份。

### 1.10.4 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。
- 对于其他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。

### 1.10.5 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。
- LLDP 报文匹配后上送控制器不受实例限制，只要收到报文的接口在 OpenFlow 实例内，并且实例内有匹配 LLDP 报文并上送控制器的表项，则 LLDP 报文就会在该实例上送控制器。

### 1.10.6 Flow Mod的限制

#### 1. Table Miss表项的添加、修改和删除

- Switch 在激活后缺省会生成 Table Miss 表项，其动作是 Drop，此表项不能被控制器通过 Modify 的动作修改，不能被控制器通过 Multipart 消息查询到，仅能由控制器通过 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，会生成缺省的 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。

## 1.11 附录 B MAC-IP流表

OpenFlow Switch 支持两种类型的 Flow Table，MAC-IP 类型和 Extensibility 类型。允许通过命令行指定 Table ID，Flow Table 会根据 Table ID 进行排序。Flow Table 需要重新激活后才能生效。

MAC-IP 流表是使用 MAC 地址表项和路由表项实现 Flow Table；Extensibility 流表使用 ACL 实现 Flow Table。

### 1.11.1 MAC-IP流表支持能力

必选的能力是控制器下发时必须携带的匹配或者动作项，可选能力是下发时可携带可不携带，如果不携带的话由 Switch 添加缺省的匹配或者动作项。

二层表项使用MAC地址表实现，其支持能力如 [表 1-10](#) 所示。

表1-10 MAC-IP 流表二层表项支持能力

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</li></ul>
可选匹配项	无
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（在多级流表存在的情况下，即使控制器不下发，Switch 上缺省下发该动作）</li><li>• Write Meta（在多级流表存在的情况下，即使控制器不下发，Switch 上缺省下发目的 MAC 匹配的 MetaData）</li></ul>

三层表项使用路由表实现，其支持能力如 [表 1-11](#) 所示。

表1-11 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 IP 地址</li> <li>• 单播目的 MAC 地址（必须是匹配 VLAN 对应的 VLAN 接口的 MAC 地址）</li> </ul>
可选匹配项	无
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 对应的 VLAN 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，即使控制器不下发，Switch 上缺省下发该动作）</li> <li>• Write Meta（在多级流表存在的情况下，即使控制器不下发，Switch 上缺省下发目的 IP 地址匹配的 MetaData）</li> </ul>

### 1.11.2 MAC-IP流表的限制

MAC-IP 流表的 Flow Entry 有一定的限制，控制器需要遵循这些限制下发表项，否则可能会造成转发错误。

二层表项的限制如 [表 1-12](#) 所示。

表1-12 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 1-13](#) 所示。

表1-13 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 接口 UP</li> <li>• 目的 MAC 地址是匹配 VLAN 对应的 VLAN 接口的 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 对应的 VLAN 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 接口存在并且处于 UP 状态，且 VLAN 接口会做为 OpenFlow 接口上报（包括了 VLAN 接口的链路状态和 MAC 地址），在 VLAN 接口删除时同时也会上报给控制器，需要由控制器删除对应的三层流表项，因此需要控制器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。

### 1.11.3 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: 报文上送控制器;
- Normal: 报文正常转发。

### 1.11.4 Dynamic aware

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

### 1.11.5 MAC-IP流表与Extensibility流表的配合

#### 1. MetaData/Mask

MAC-IP 流表和 Extensibility 流表通过 MetaData/Mask 可以实现多级流表。

MAC-IP 流表支持 Write MetaData/Mask，Extensibility 流表支持 Match MetaData/Mask。

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 1-14](#)。

表1-14 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 流表和 MAC-IP 流表配合时：

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