

OpenFlow概念_什么是openflow-CSDN博客

OpenFlow协议

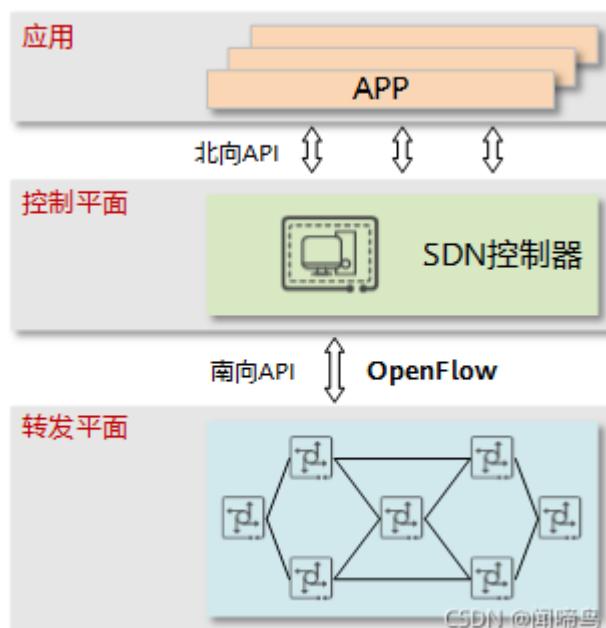
SDN与OpenFlow不是划等号的关系，而是SDN包含OpenFlow的关系。实际上，SDN有多种实现方案，在ONF SDN方案中OpenFlow充当南向接口的作用。南向接口的定义是控制平面与数据平面之间进行交互的协议，南向接口除了可以采用OpenFlow外，还有许多别的协议，如OF-CONFIG、OVSDB、NETCONF、PCEP、XMPP等等

1.概念

OpenFlow是一种网络通信协议，应用于SDN架构中控制器和转发器之间的通信。软件定义网络SDN的一个核心思想就是“转发、控制分离”，要实现转、控分离，就需要在控制器与转发器之间建立一个通信接口标准，允许控制器直接访问和控制转发器的转发平面。OpenFlow引入了“流表”的概念，转发器通过流表来指导数据包的转发。控制器正是通过OpenFlow提供的接口在转发器上部署相应的流表，从而实现对转发平面的控制。

2.OpenFlow的起源与发展

Clean Slate项目的负责人Nick McKeown教授及其团队发现，如果将传统网络设备的数据转发和路由控制两个功能模块相分离，通过集中式的控制器（Controller）以标准化的接口对各种网络设备进行管理和配置，那么这将为网络资源的设计、管理和使用提供更多的可能性，从而更容易推动网络的革新与发展。于是，他们便提出了OpenFlow的概念

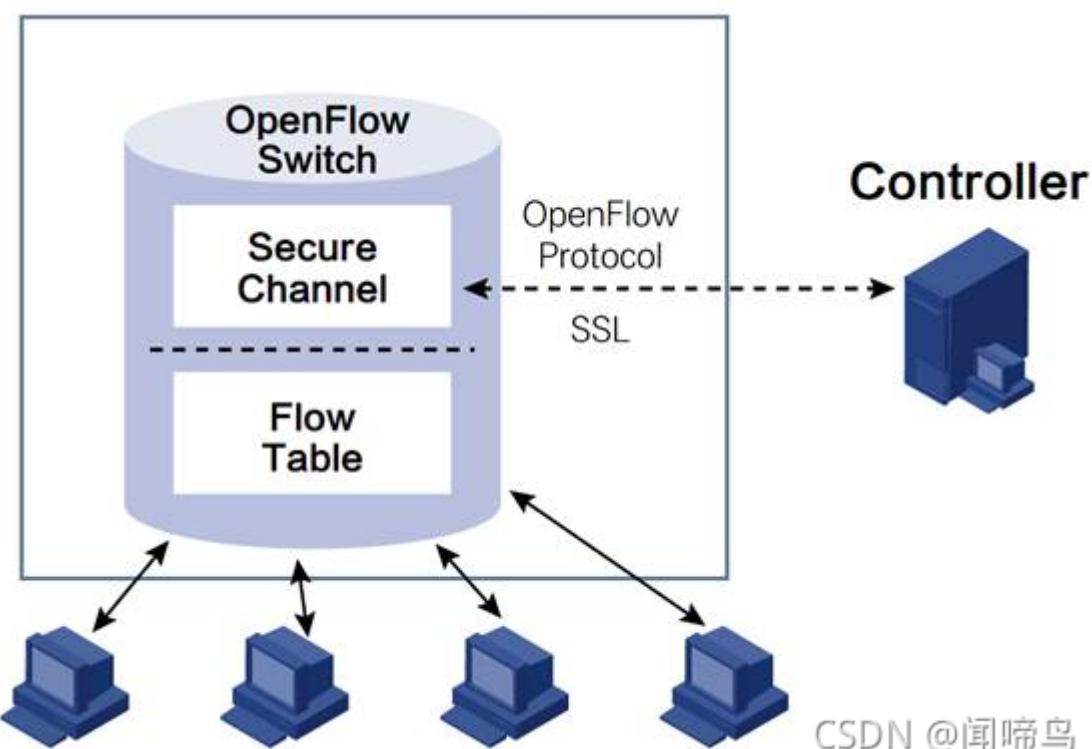


目前使用和支持最多的是OpenFlow1.0和OpenFlow1.3版本。



3. OpenFlow工作原理

整个OpenFlow协议架构由控制器（Controller）、OpenFlow交换机（OpenFlow Switch）以及安全通道（Secure Channel）组成。控制器对网络进行集中控制，实现控制层的功能；OpenFlow交换机负责数据层的转发，与控制器之间通过安全通道进行消息交互，实现表项下发、状态上报等功能。



4. OpenFlow安全通道

安全通道就是连接OpenFlow交换机与控制器的信道，**负责在OpenFlow交换机和控制器之间建立安全链接**。控制器通过这个通道来控制和管理交换机，同时接收来自交换机的反馈。

通过OpenFlow安全通道的信息交互必须按照OpenFlow协议规定的格式来执行，通常采用TLS（Transport Layer Security）加密，在一些OpenFlow版本中（1.1及以上），有时也会通过TCP明文来实现。通道中传输的OpenFlow消息类型包括以下三种：

- **Controller-to-Switch消息**：由控制器发出、OpenFlow交换机接收并处理的消息，主要用来管理或获取OpenFlow交换机状态。

- **Asynchronous消息**: 由OpenFlow交换机发给控制器，用来将网络事件或者交换机状态变化更新到控制器。
- **Symmetric消息**: 可由OpenFlow交换机发出也可由控制器发出，也不必通过请求建立，主要用来建立连接、检测对方是否在线等。

5. OpenFlow交换机

OpenFlow交换机是整个OpenFlow网络的核心部件，主要负责数据层的转发。OpenFlow交换机可以是物理的交换机/路由器，也可以是虚拟化的交换机/路由器。按照对OpenFlow的支持程度，OpenFlow交换机可以分为两类：

- OpenFlow专用交换机：一个标准的OpenFlow设备，仅支持OpenFlow转发。他不支持现有的商用交换机上的正常处理流程，所有经过该交换机的数据都按照OpenFlow的模式进行转发。
- OpenFlow兼容型交换机：既支持OpenFlow转发，也支持正常二三层转发。这是在商业交换机的基础上添加流表、安全通道和OpenFlow协议来获得了OpenFlow特性的交换机。

OpenFlow交换机在实际转发过程中，依赖于流表（Flow Table）。流表是OpenFlow交换机进行数据转发的策略表项集合，指示交换机如何处理流量，所有进入交换机的报文都按照流表进行转发。流表本身的生成、维护、下发完全由控制器来实现。

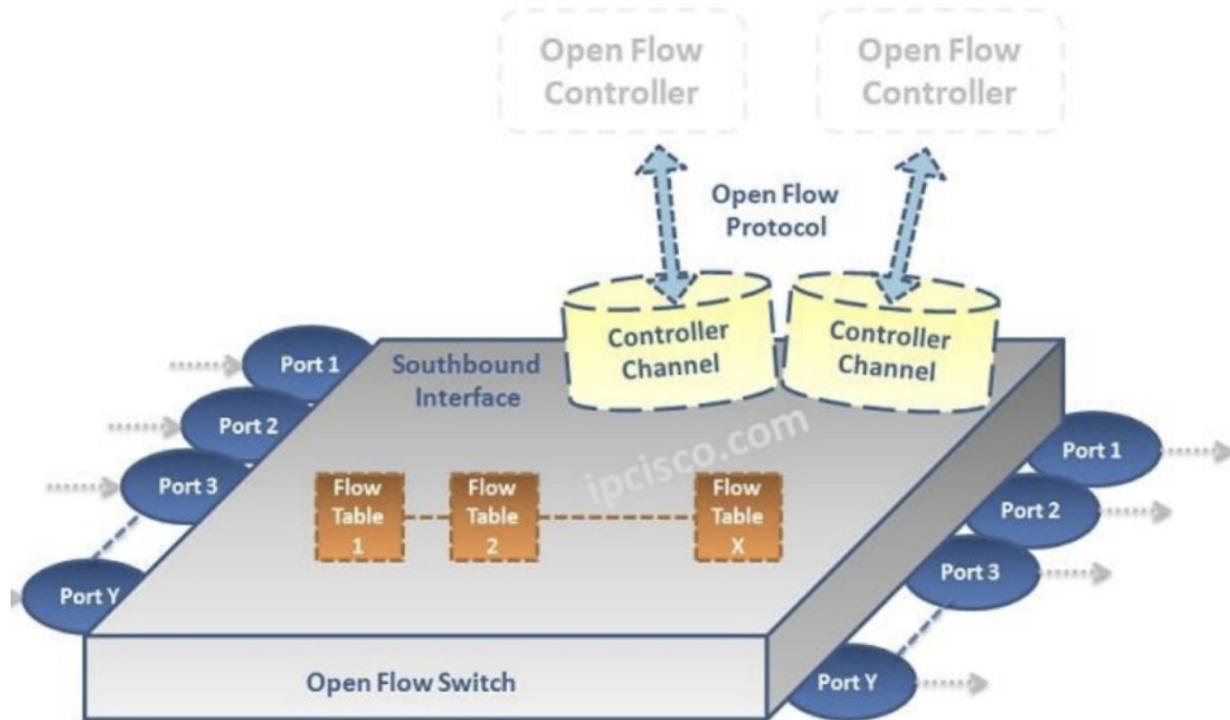
6. 流表项的组成

在传统网络设备中，交换机/路由器的数据转发需要依赖设备中保存的二层MAC地址转发表、三层IP地址路由表以及传输层的端口号等。OpenFlow交换机中使用的“流表”也是如此，不过他的表项并非是指普通的IP五元组，而是整合了网络中各个层次的网络配置信息，由一些关键字和执行动作组成的灵活规则。

OpenFlow交换机转发面由两部分组成：端口和流表。一个交换机可以有很多种端口，也可以有很多级流表。

它主要由OpenFlow通道和数据平面组成，而数据平面又包括流表、端口、组表和Meter表等：
OpenFlow通道：用于交换机和控制器进行通信（基于OpenFlow交换协议）
流表：即存放流表项的表
端口：是OpenFlow与其他网络协议栈进行数据交换的网络接口，包括物理端口、逻辑端口以及预留端口等
组表：用于定义一组可被多个流表项共同使用的作用
Meter表：用于计量和限速

OpenFlow交换机转发面由两部分组成：端口和流表。一个交换机可以有很多种端口，也可以有很多级流表。下图是思科提供的OpenFlow交换机总体架构。



RULE

ACTIONS

STATS

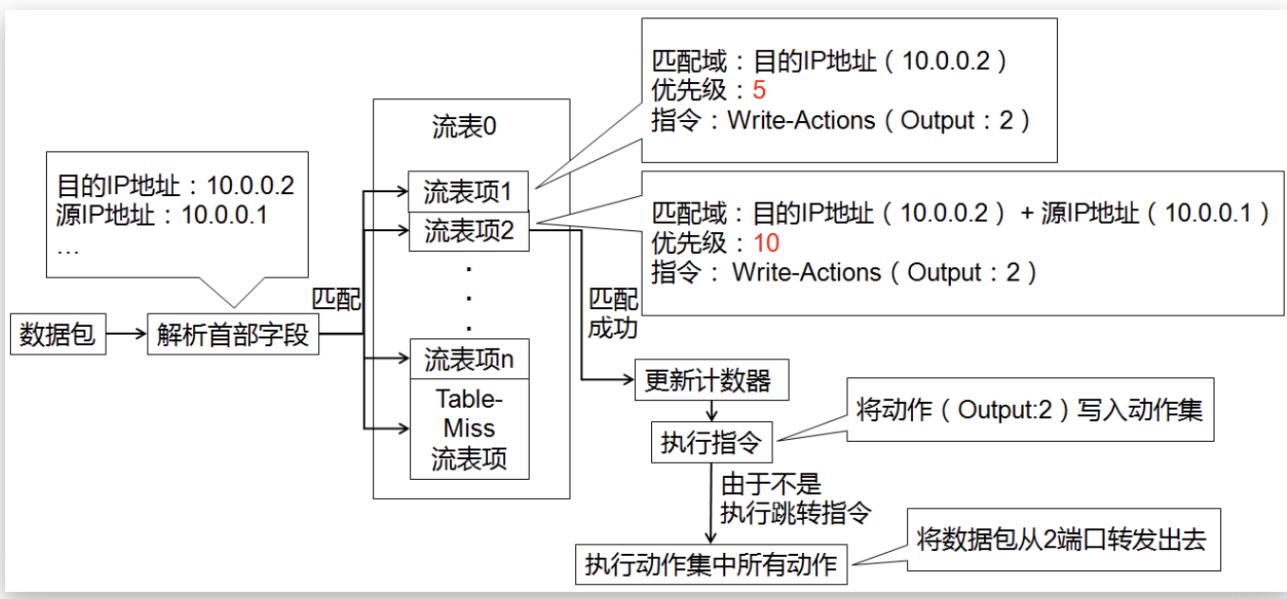
知乎 @冠军长史沈劲

CSDN @快乐学习 ~

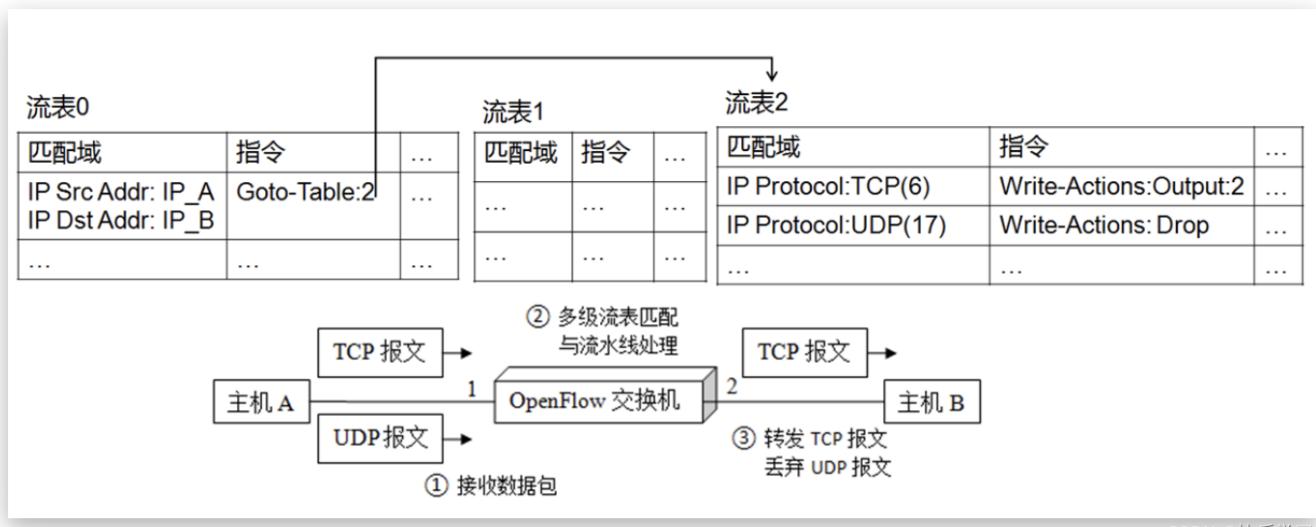
OpenFlow channel (通道) OpenFlow Channel 是指交换机跟Controller之间的连接通道
OpenFlow通道使用OpenFlow交换协议 (OpenFlow switch protocol) , 连接通常基于TLS连接, 但也支持直接TCP连接。

OpenFlow流表的每个流表项都由匹配域 (Match Fields) 、处理指令 (Instructions) 等部分组成。流表项中最为重要的部分就是匹配域和指令，当OpenFlow交换机收到一个数据包，将包头解析后与流表中流表项的匹配域进行匹配，匹配成功则执行指令。因此流表可以简化理解为key-value形式的{匹配域-指令}表。

OpenFlow提供丰富的匹配域字段来定义不同粒度的流，如可以基于目的IP地址定义一条流，也可根据源IP地址 + 目的IP地址来定义一条流



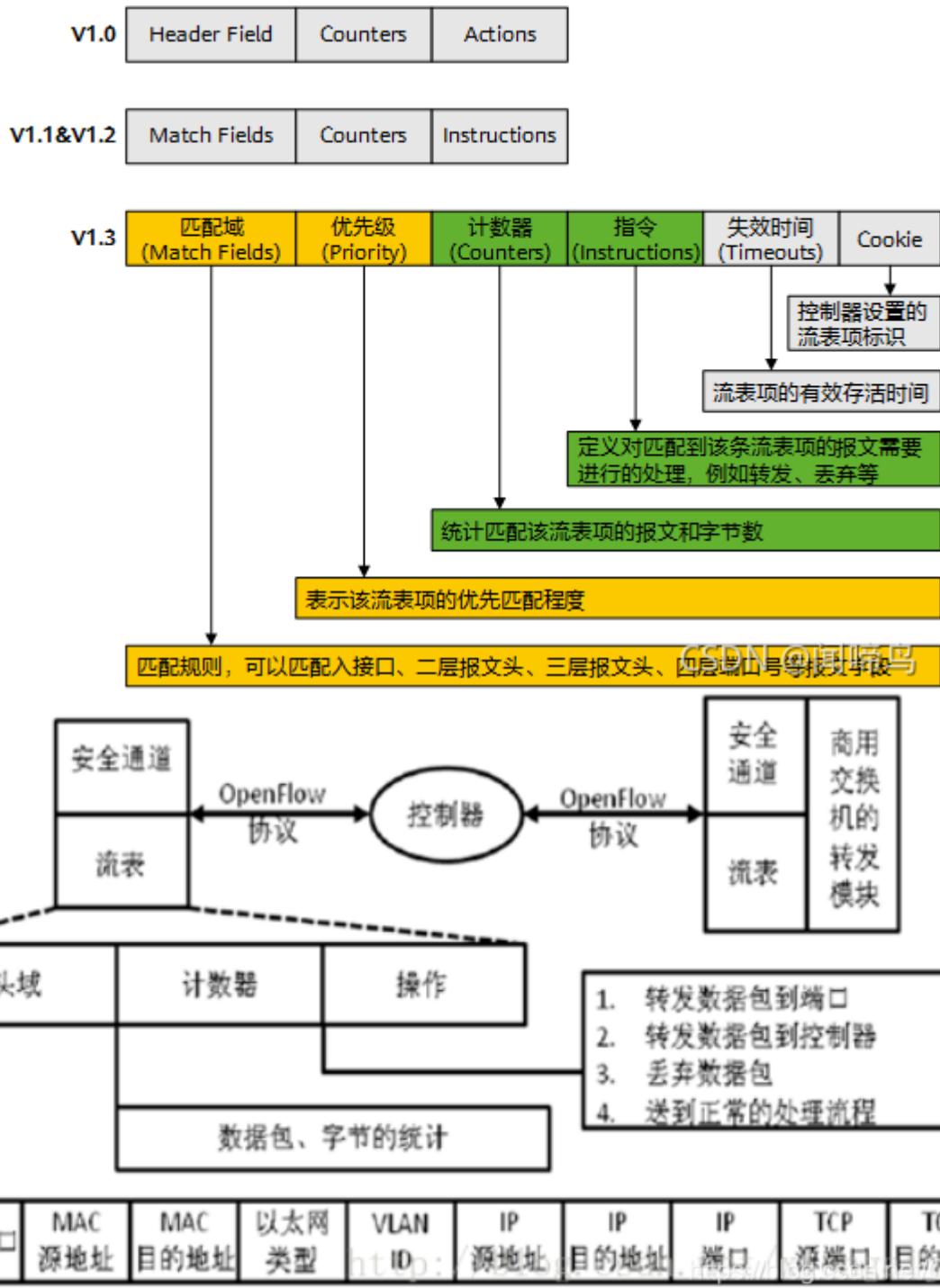
CSDN @快乐学习 ~



CSDN @快乐学习 ~

流表中没有设置Table-Miss流表项，匹配失败时，丢弃数据包。流表中设置有Table-Miss流表项（优先级为0且匹配域为ANY），则最后会匹配该表项，执行相应指令（如丢弃、交由控制器处理、交给下一张流表处理）。

流表项的结构随着OpenFlow版本的演进不断丰富，不同协议版本的流表项结构如下。



匹配域 (Match Fields) : 用于定义某条流, 也是流表匹配的依据

指令 (Instructions) : 表示对该条流应该如何处理

优先级 (Priority) : 表示该流表项的优先匹配程度 **计数器 (Counters) :** 用于统计该条流的信息 **生存时间 (Timeouts) :** 表示流表项的有效存活时间 **Cookie:** 控制器设置用来过滤被流统计、流修改和流删除操作请求影响的流表项 **标志 (Flags) :** 用于流表项管理

7. 指令与动作

7.1 指令 (Instruction)

指令是流表项匹配成功时的处理动作，分为三类

- 1、更新动作集 (Action Set)：添加、修改、清空动作集，前面两个对应Write-Actions指令，清空动作集对应Clear-Actions指令
- 2、修改流水线处理次序：从序号低的表跳转到序号高的表，对应Go-To-Table指令
- 3、其他：更新元数据以及设定触发器，分别对应Write-Metadata指令和Stat-Trigger指令

7.2 动作 (Action)

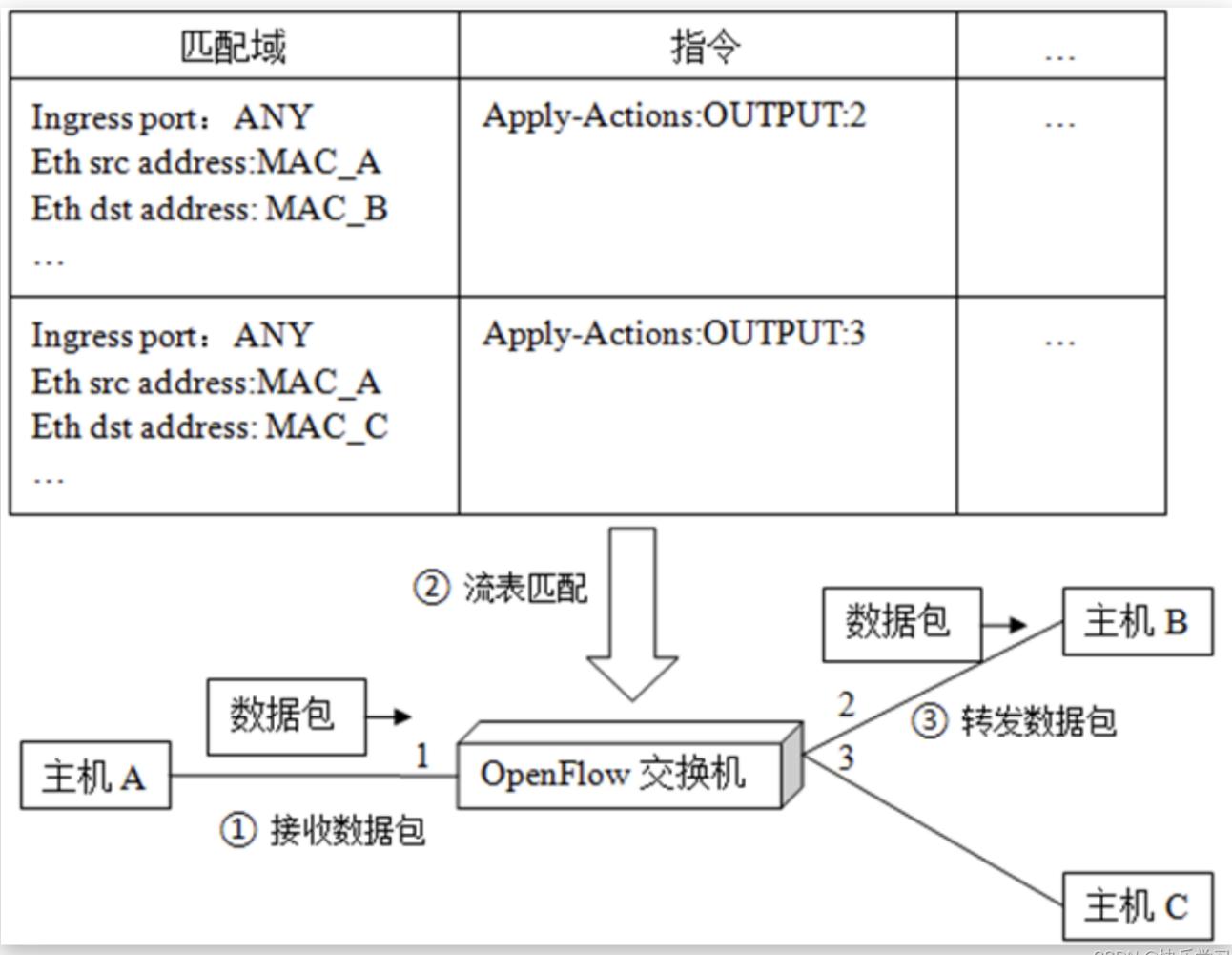
动作 (Action) Action是指对数据包的具体处理动作，可分为两类：一类是定义数据包的转发，另一类是修改数据包包头字段；

指令名称	OpenFlow交换机是否支持	功能
Apply-Actions	可选	立即执行动作列表 (Action List) 中的动作，且不改变动作集 (Action Set)
Clear-Actions	必须	清空动作集
Write-Actions	必须	添加一条动作到动作集或修改动作集中的动作
Goto-Table	必须	跳转到指定流表
Write-Metadata	可选	更新元数据，在多个流表之间传递信息
Stat-Trigger	可选	若流的某个统计信息超过设定阈值，生成一个事件通知控制器

CSDN @快乐学习~

使用单流表转发数据包：

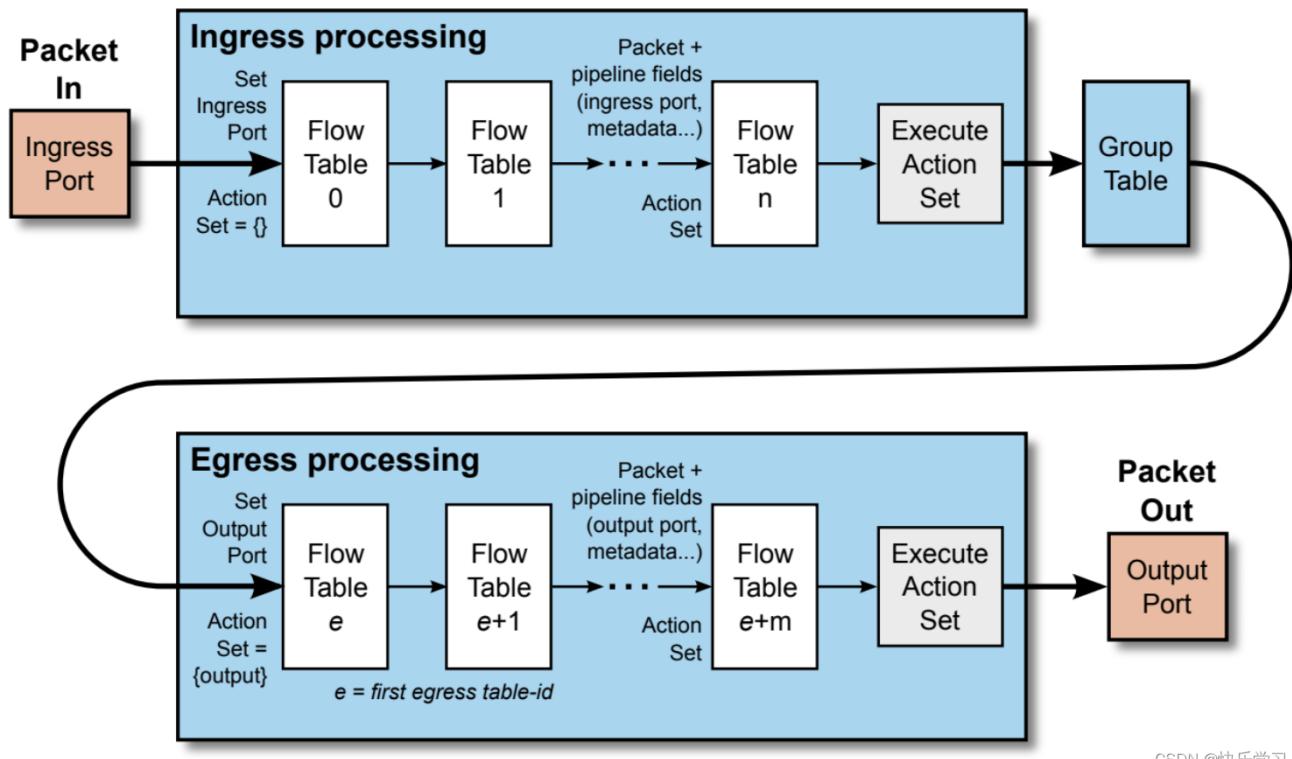
- 假设主机A发送数据包给主机B，使用单张流表的OF交换机处理数据包过程，如图所示 OF交换机从1端口接收数据包 OF交换机解析数据包首部，并查询流表进行流表匹配，匹配第一条流表项，并执行相应指令 将数据包转发到OF交换机的2端口



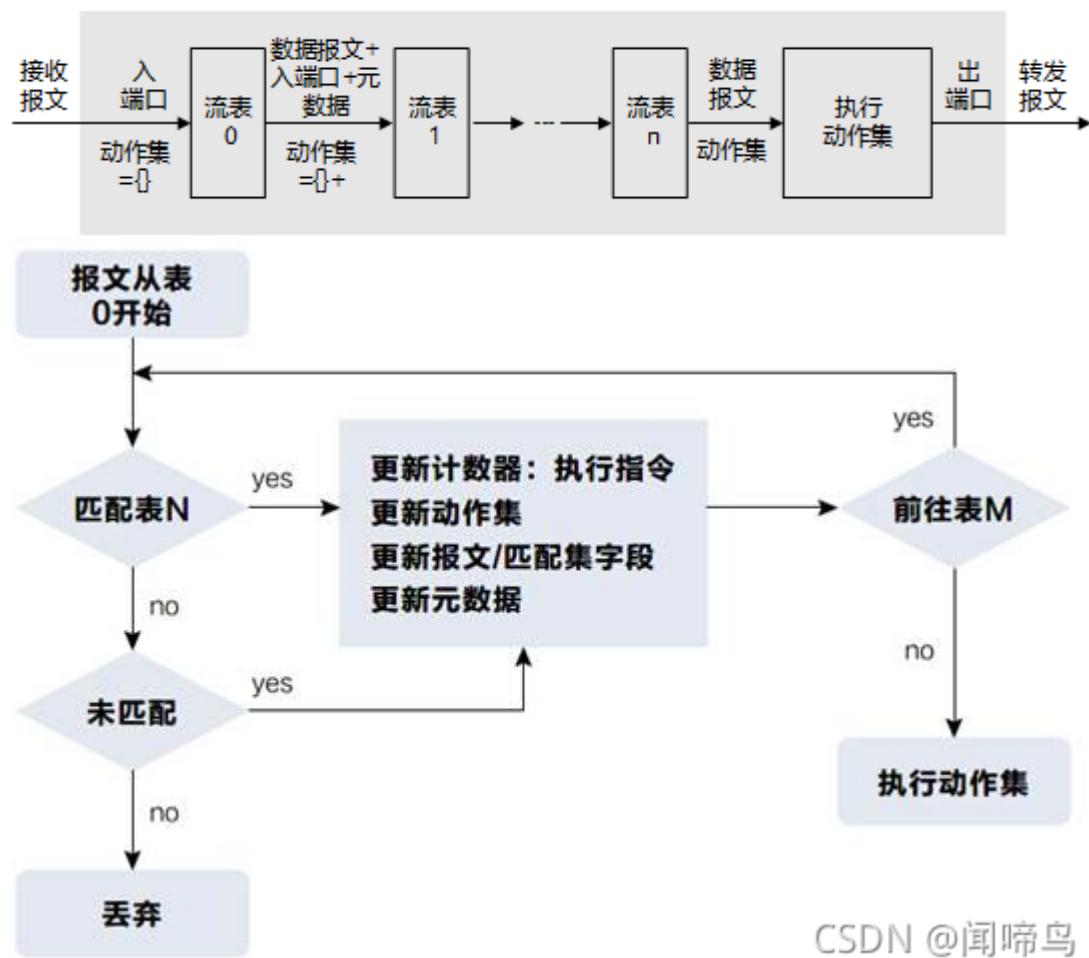
8.多级流表与流水线处理

OpenFlow v1.0采用单流表匹配模式，这种模式虽然简单，但是当网络需求越来越复杂时，各种策略放在同一张表中显得十分臃肿。这使得控制平面的管理变得十分困难，而且随着流表长度与数目的增加，对硬件性能要求也越来越高。

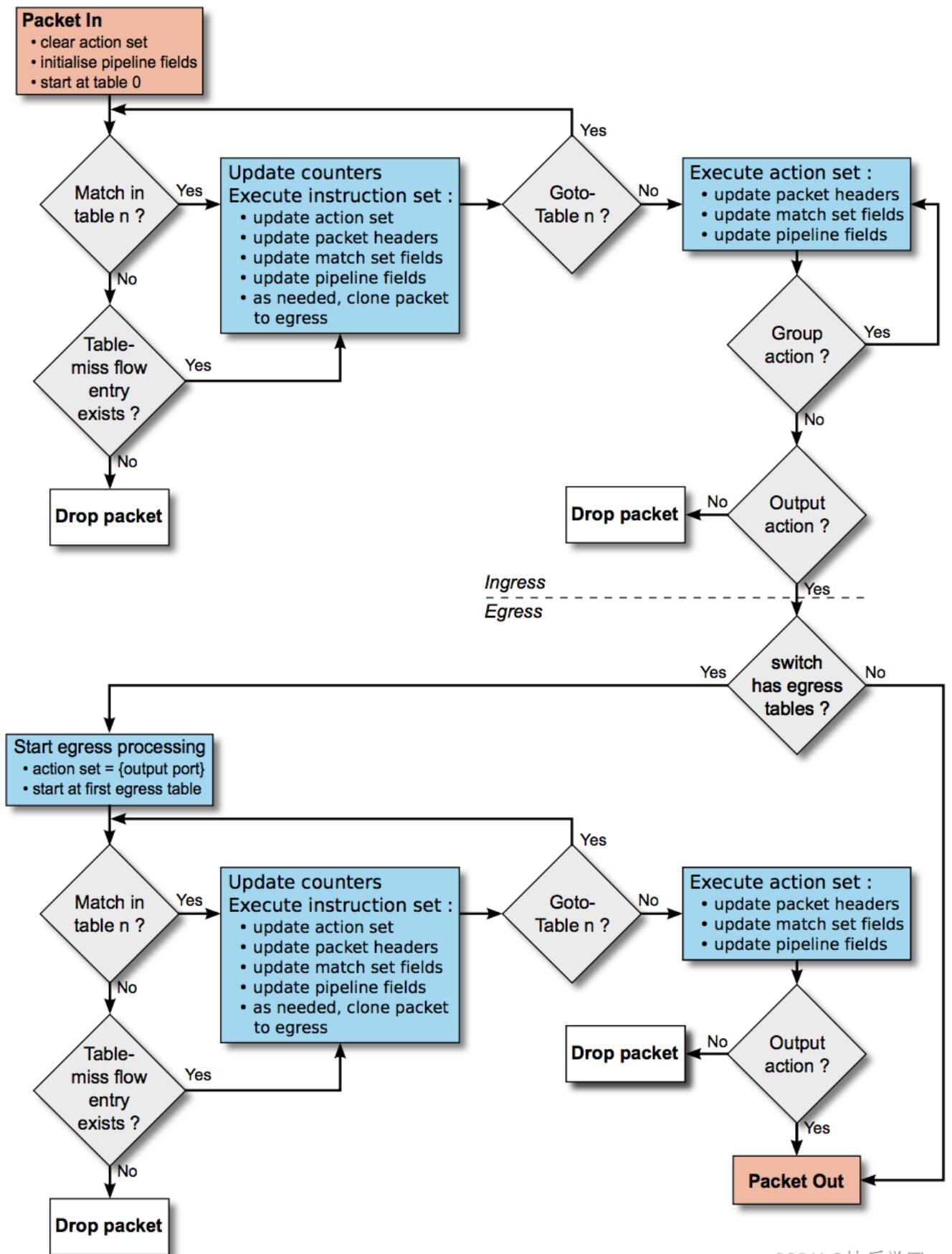
从OpenFlow v1.1开始引入了多级流表和流水线处理机制，当报文进入交换机后，从序号最小的流表开始依次匹配，报文通过跳转指令跳转至后续某一流表继续进行匹配，这样就构成了一条流水线。多级流表的出现一方面能够实现对数据包的复杂处理，另一方面又能有效降低单张流表的长度，提高查表效率。



CSDN @快乐学习 ~



CSDN @闻啼鸟



CSDN @快乐学习 ~

9.流表下发方式

OpenFlow流表的下发分可以是主动 (Proactive) 的，也可以是被动 (Reactive) 的：

主动模式下，控制器将自己收集的流表信息主动下发给OpenFlow交换机，随后交换机可以直接根据流表进行转发。这种方式相当于预置路由。

例如Table-Miss表项，就需要采用Proactive方式在SDN控制器与OpenFlow交换机建立连接后下发，显示的指定数据包查表失败时，OpenFlow交换机的处理方式

被动模式下，OpenFlow交换机收到一个报文而查流表失败时，会发送消息询问控制器，由控制器进行决策该如何转发，并计算、下发相应的流表。这种方式相当于按需下发路由，只有在有路由需求且查流表失败时，才会触发新的流表安装。被动模式的好处是交换机无需维护全部的流表，只有当实际的流量产生时才向控制器获取流表记录并存储，当老化定时器超时后可以删除相应的流表，因此可以大大节省交换机芯片空间。

10.组表

独立于流水线之外，每台OpenFlow交换机只有一张组表。组表 (Group Table) 由若干条组表项 (Group Entry) 组成，具有将多个端口定义为一个组的能力，从而实现广播、多播，负载均衡、链路聚合、故障转移等。组表项结构如图所示，定义了一到多个动作桶 (Action Bucket)，用于描述转发到指定端口前，对数据包的处理。

组号	组类型	计数器	动作桶列表
----	-----	-----	-------

组号：用于标识一个组，32位无符号整数

组类型：指定组的动作，根据不同组类型从动作桶列表中选取不同的桶执行

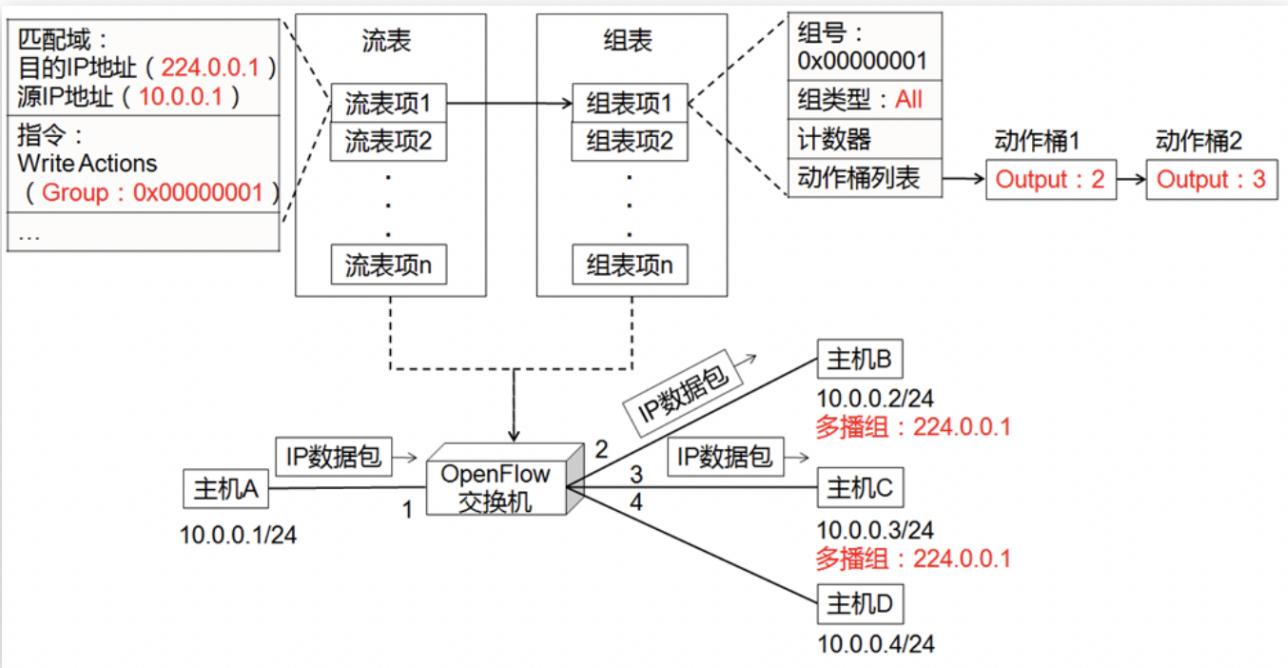
计数器：统计组表项处理的数据包数

动作桶列表 (Action Buckets)：包含一至多个动作桶，每个动作桶包含一个动作集

组类型	实现	作用
All (全选)	必须	执行动作桶列表中的所有桶，可用于实现广播、多播
Indirect (间接)	必须	动作桶列表只有一个动作桶，可用于流量汇聚
Select (单选)	可选	根据某种算法 (Hash、轮询等) 选择一个桶执行，用于实现网络负载均衡、链路聚合等
Fast failover (快速恢复)	可选	选择第一个有效的桶执行，可用于实现故障转移

CSDN @快乐学习 ~

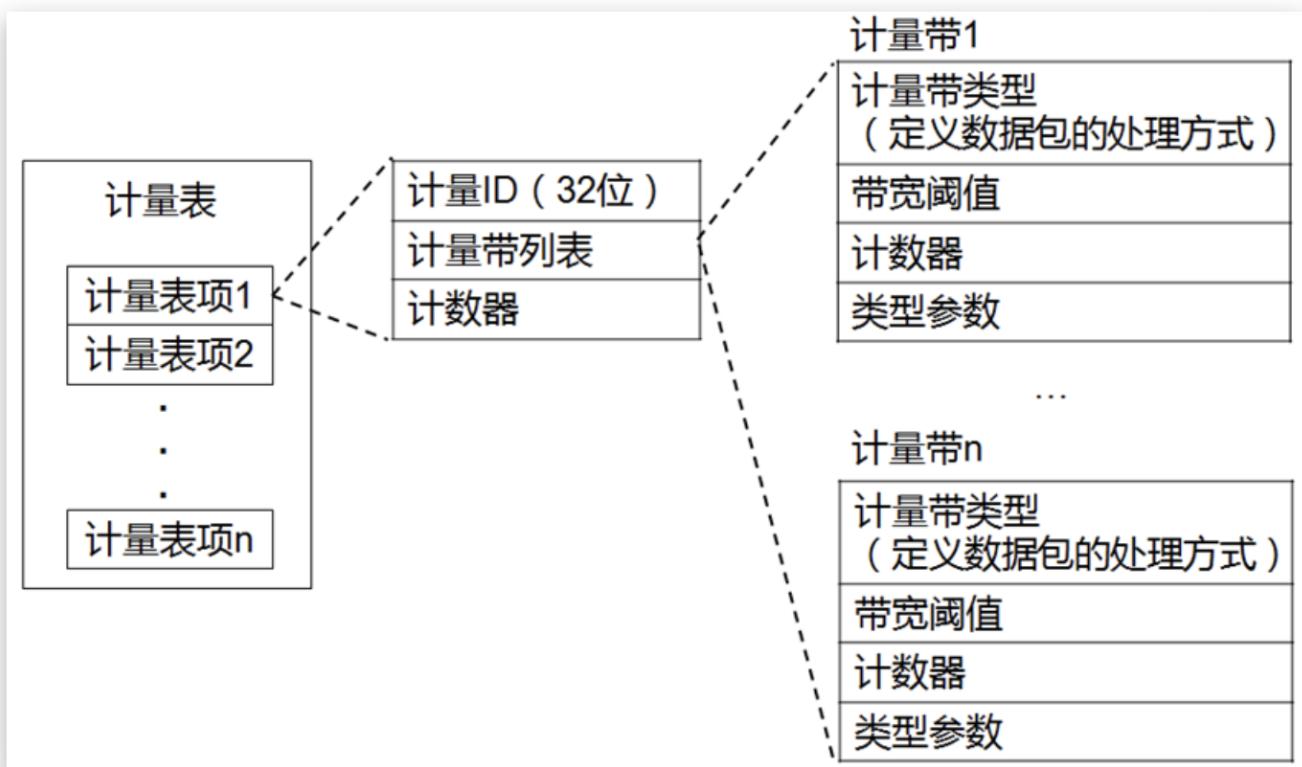
使用组表实现多播：假设主机A使用多播方式向主机B和C发送IP数据包



CSDN @快乐学习 ~

11.计量表

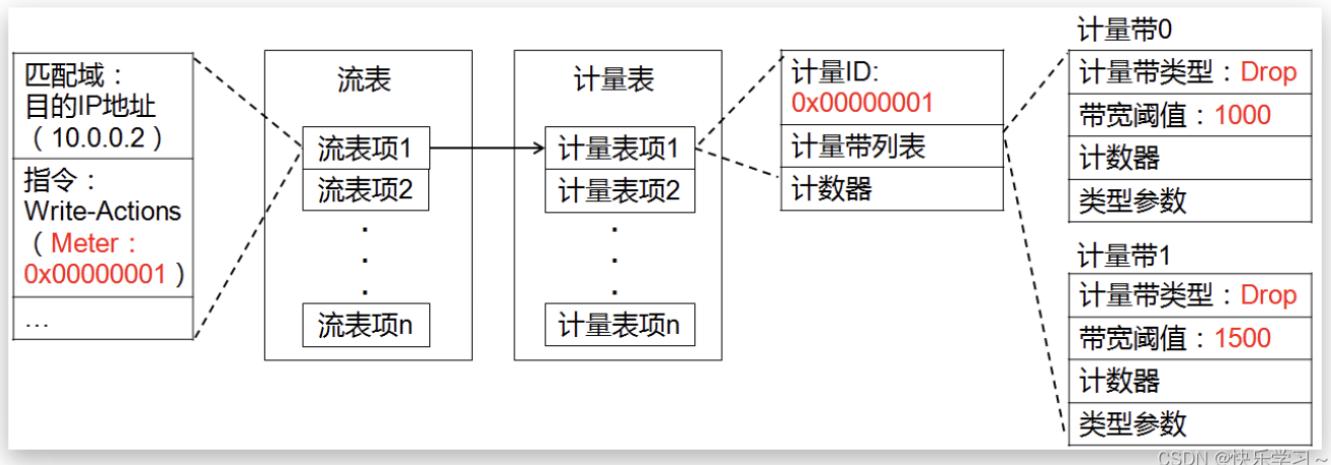
对流进行测量，从而为流提供QoS功能，如限速、DiffServ。每台OpenFlow交换机只有一张计量表 (Meter Table)，由若干计量表项 (Meter Entry) 组成，每个计量表项可以定义一至多个计量带 (Meter Band)，计量带定义了带宽阈值和数据包处理方式 (丢弃、DSCP标记)



CSDN @快乐学习 ~

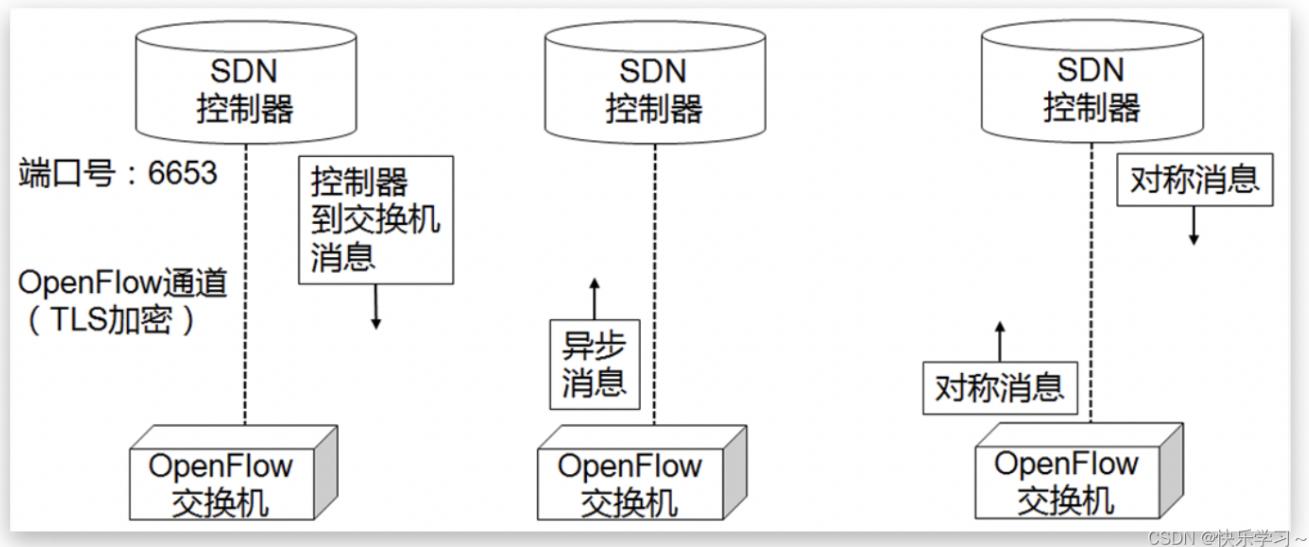
对流进行限速：

假设对某条流X（目的IP地址：10.0.0.2）进行限速，且当前测得流X数据包的速率为1200kBps流表匹配后，交由计量1处理 由于测得数据包速率1200kBps > 带宽阈值1000，根据计量带0定义的处理方式丢弃数据包，从而实现限速。



12. OpenFlow消息

消息按照发送的位置可分为三大类，每一大类中有若干子消息



- Controller-to-Switch消息：SDN控制器主动发送给OpenFlow交换机的消息

- Features：用于获取交换机特性
- Configuration：用来配置和查询交换机参数
- Modify-State：用来修改交换机状态信息（增删改流表项、组表项等）
 - Table-Mod消息
 - Flow-Mod消息（流表操作，添加、删除、修改流表项）
 - Group-Mod消息
 - Port-Mod消息
 - Meter-Mod消息
- Read-State：用来读取交换机状态信息（当前配置、统计信息等）
 - Port-Stats消息
 - Flow-Stats消息
 - ...
- Packet-Out：用来指定交换机将数据包从指定端口转发出去
- Barrier：在不同消息之间使用，确保操作顺序执行
- Role Request：控制器用于询问或设置自身在交换机中的角色，常用于交换机与多控制器连接的场景
- Asynchronous-Configuration：控制器设置异步消息过滤器，只接收感兴趣的异步消息，一般在多控制器场景下使用

CSDN @快乐学习 ~

- Asynchronous（异步）消息：OpenFlow交换机主动发送给SDN控制器的异步消息

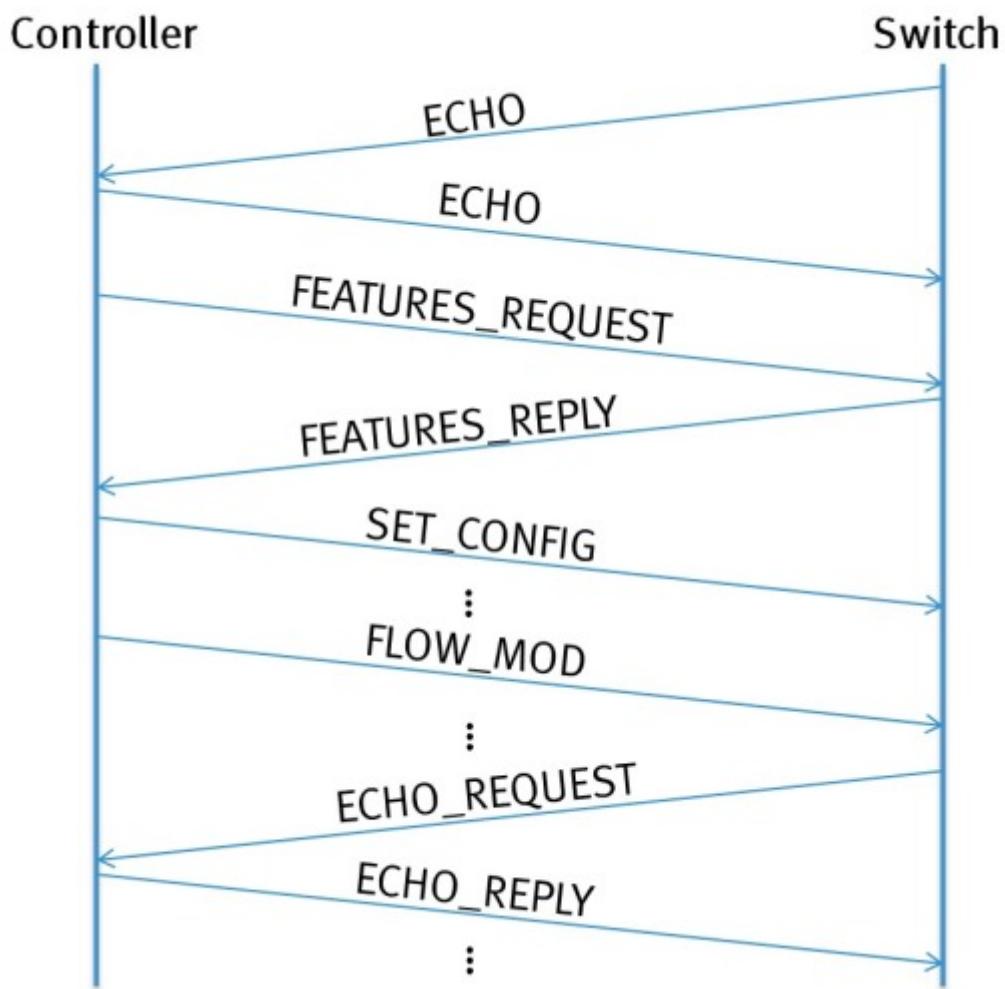
- Packet-In：将数据包交给控制器处理，一般流表匹配中出现Table-Miss时或流表项显示指定将数据包交给控制器时，触发该消息
- Flow-Removed：通知控制器，流表项被删除；流表项超时或控制器删除流表项时触发该消息（需要在交换机配置时使能该消息）
- Port-status：通知控制器，交换机端口状态发生变化
- Role-status：通知控制器，控制器在交换机中的角色发生变化
- Controller-Status：通知控制器，OpenFlow通道状态发生变化
- Flow-monitor：通知控制器，流表发生变化

- Symmetric（对称）消息：可由SDN控制器或OpenFlow交换机主动发送的消息

- Hello：建立控制器与交换机之间的OpenFlow通道
- Echo：检测交换机与控制器之间的连接状态或测量OpenFlow通道的时延和带宽
- Error：用于通告错误
- Experiment：用于实验，测试新特性

CSDN @快乐学习 ~

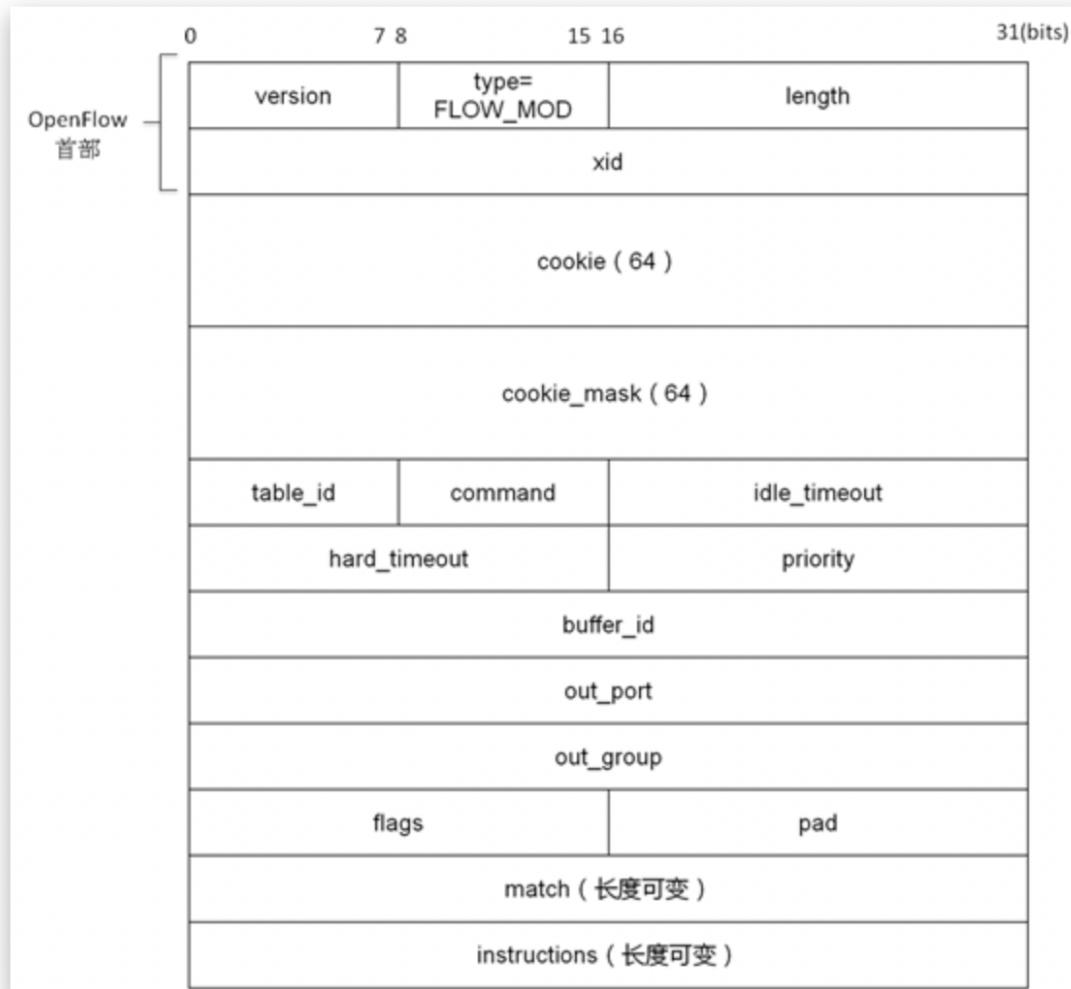
下图展示了OpenFlow和Switch之间一次典型的的消息交换过程，出于安全和高可用性等方面的考虑，OpenFlow的规范还规定了如何为Controller和Switch之间的信道加密、如何建立多连接等（主连接和辅助连接）



12.1 Flow-Mod消息

Flow-Mod消息用于流表操作，包括添加、删除、修改流表项。该消息由控制器下发给交换机，从而指导交换机对数据包的处理。

其在OpenFlow1.3中的消息格式如下图所示。



CSDN @快乐学习 ~

- command:
 - ADD:添加流表项
 - MODIFY: 根据匹配域, 修改所有匹配的流表项, 可能有多条流表项被修改
 - MODIFY_STRICT: 根据匹配域以及优先级, 修改特定的流表项, 只有一条流表项被修改
 - DELETE: 根据匹配域, 删除所有匹配的流表项, 可能有多条流表项被删除
 - DELETE_STRICT: 根据匹配域以及优先级, 删除特定的流表项, 只有一条流表项被删除

12.2 Packet-In消息

Packet-In消息用于将OpenFlow交换机上指定数据包交给控制器处理，一般流表匹配中出现Table-Miss时或流表项显示指定将数据包交给控制器时，触发该消息。此外，它还能用于主动测量时回收探测包，从而结合Packet-Out消息实现对网络拓扑与链路时延的测量，详情参考[基于OpenFlow消息的网络测量方法](#)。

其在OpenFlow1.3中的消息格式如下图所示。



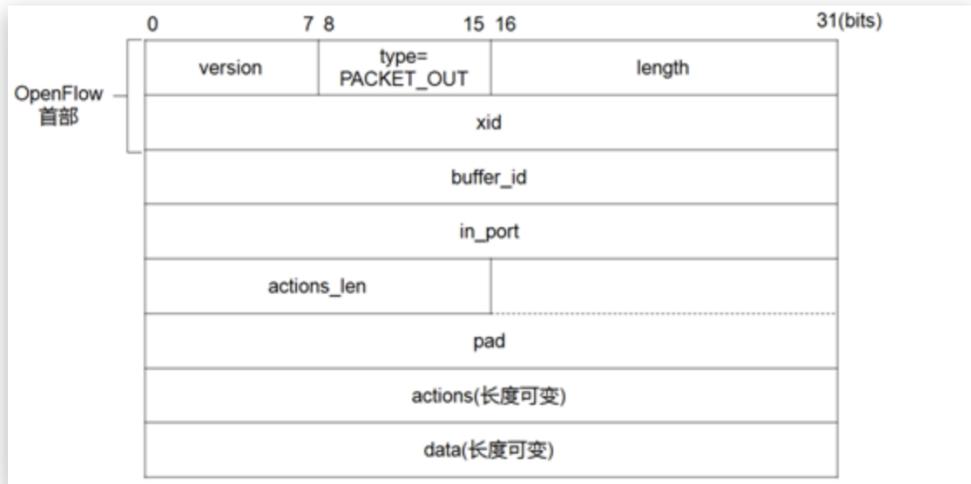
- buffer_id：若该字段为-1，表明交换机未缓存数据包，Packet-In消息需要携带完整数据包发送至控制器；否则，表明数据包已在交换机上缓存，Packet-In消息只携带部分数据包上传至控制器。

CSDN @快乐学习 ~

12.3 Packet-Out消息

Packet-Out消息用于指定交换机将数据包从指定端口转发出去。触发该消息的情况有两种：1.转发Packet-In消息携带的数据包 2.转发控制器主动构造的数据包（如用于链路发现的LLDP报文）。此外，由于该消息能够携带自定义数据包，控制器通过在Packet-Out消息中封装探测包并下发至指定交换机，就能够发起主动测量任务，配合Packet-In消息可实现对网络拓扑与链路时延的测量，详情参考[基于OpenFlow消息的网络测量方法](#)。

其在OpenFlow1.3中的消息格式如下图所示。



- buffer_id: 若该字段为-1，表明交换机未缓存数据包，Packet-Out消息携带控制器创建的数据包发送至交换机；否则，Packet-Out消息表示交换机需要将本地缓存的数据包按照Packet-Out中的actions进行处理。

CSDN @快乐学习 ~

13. OpenFlow的应用场景

随着OpenFlow概念的发展和推广，其研究和应用领域也得到了不断拓展，主要包括网络虚拟化、安全和访问控制、负载均衡等方面。

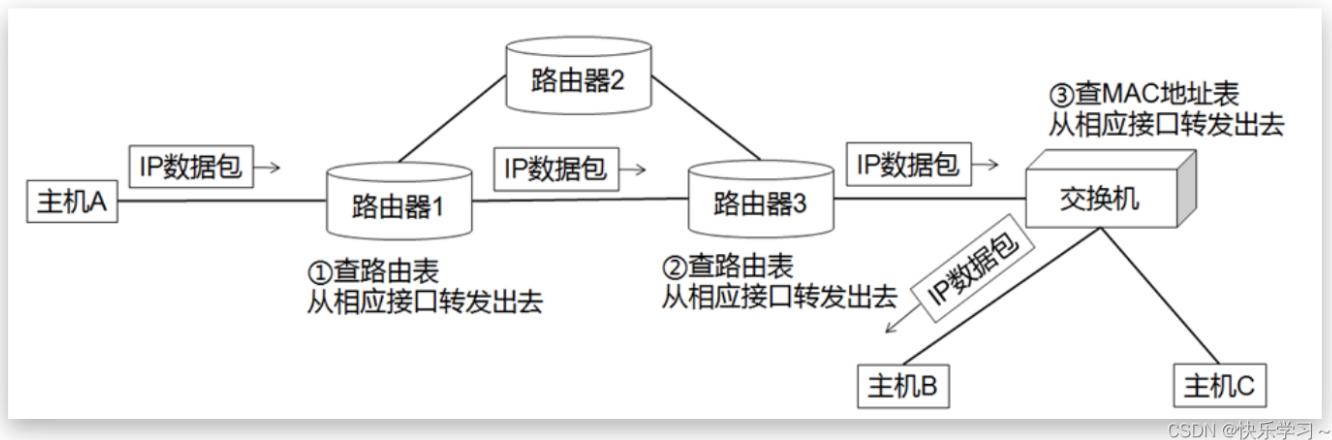
- **OpenFlow在校园网络中的应用** 科研院校网络是OpenFlow的发源地，也是OpenFlow被广泛应用的网络环境。学生或研究人员在进行网络创新性研究时，可能会有全新设计的网络控制协议和数据转发技术需要验证，他们希望有一个平台能帮助他们把网络的控制、转发独立出来，以便能在平台上自由验证他们的研究工作。基于OpenFlow的网络正好可以提供这样一个试验平台，不仅更接近真实网络的复杂度，实验效果好，而且可以节约实验费用。
- **OpenFlow在数据中心网络中的应用** 云数据中心是OpenFlow得以发扬光大的地方。云数据中心部署时存在多租户资源动态创建、流量隔离以及虚拟机动态迁移等虚拟化需求，OpenFlow交换机可以配合云管理平台实现网络资源的动态分配和网络流量的按需传输，实现云服务的网络虚拟化需求并可以改善网络性能。其次，在数据中心的流量很大，如果不能合理分配传输路径很容易造成数据拥塞，从而影响数据中心的高效运行。如果在数据中心部署OpenFlow，可以动态获取各链路的流量传输情况，动态下发OpenFlow流表规则进行均衡调度，实现路径优化以及负载均衡。

- **OpenFlow在园区网络中的应用** 在园区网络中可以使用OpenFlow对接入层设备进行有效的管控。接入层设备的特点是量大、故障率高，但设备功能和流量策略相对简单。如果使用OpenFlow，可以在控制器上集中统一接入设备进行流表下发、网络监控等维护工作。在要求用户身份认证的场合，可以把认证流量引导到控制器上，在验证用户身份合法后再下发准入规则到用户连接的交换机端口上。在控制器检测到特定网络端口或特定用户流量异常时，可以通过下发规则关停设备端口或限制特定流量，快速恢复网络故障，提高网络可靠性和安全性。

13. 基于OpenFlow的SDN网络中网络设备的工作过程

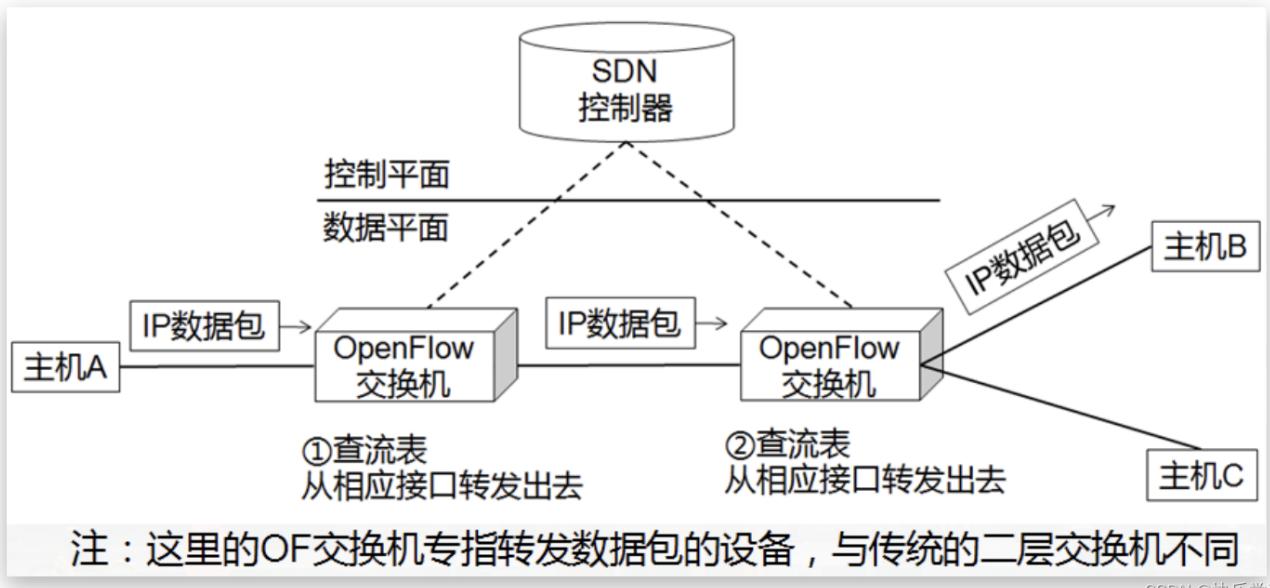
传统网络中网络设备的工作过程

假设主机A向主机B发送IP数据包，且所有路由表和MAC地址表中都有该数据包的相应表项
路由器之间运行分布式路由协议构建路由表。查表成功则基于目的IP地址转发；查表失败时，丢弃数据包
交换机根据自学习算法构建MAC地址表。查表成功则基于目的MAC地址转发；查表失败时，除入端口外其余所有端口转发出去



路由计算、转发规则（流表）下发由控制器完成 OF交换机只需要按照流表进行转发，查表失败时，通过Packets-In消息询问控制器

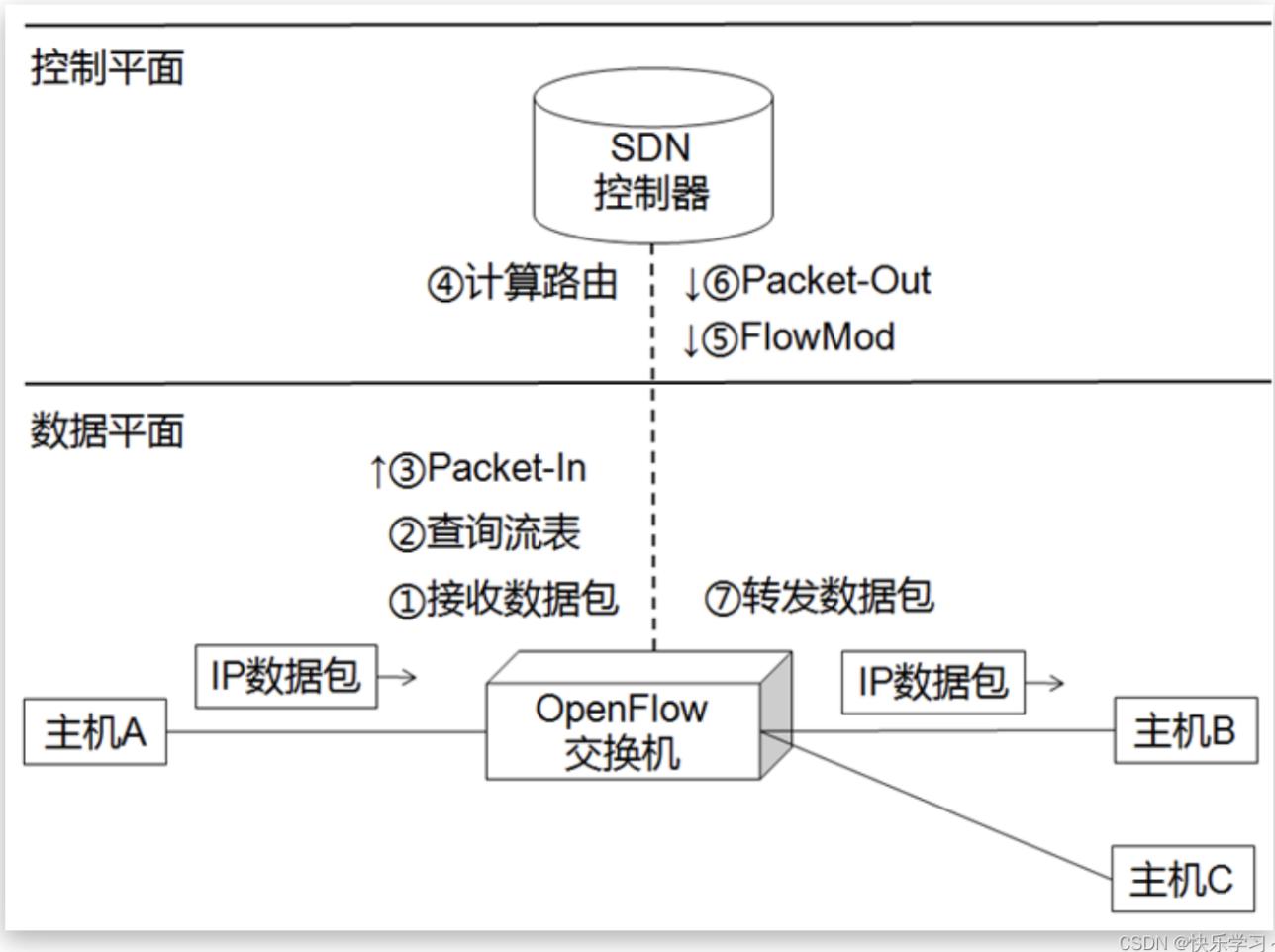
假设主机A向主机B发送IP数据包，且所有流表有该数据包相应表项 OpenFlow交换机查询流表来转发数据包，查表成功则基于匹配域（如目的IP地址+源IP地址）转发；查表失败时，则询问SDN控制器 流表由SDN控制器来构建



CSDN @快乐学习 ~

假设主机A向主机B发送IP数据包，且OpenFlow交换机中流表为空

OF交换机接收IP数据包 OF交换机解析数据包首部并查询流表，由于流表为空，不知道如何转发，因此需要询问控制器 OF交换机向控制器发送Packet-In消息 控制器为主机A发送给主机B的IP数据包计算路由 控制器向OF交换机下发流表，使用FlowMod消息承载流表信息， OF交换机接收该消息后安装流表 控制器向OF交换机发送Packet-Out消息，指示OF交换机按照刚安装好的流表转发IP数据包 OF交换机收到Packet-Out消息后转发数据包



CSDN @快乐学习 ~

路由器，交换机，OpenFlow交换机对包的转发处理

网络设备	转发依据	查表依据	查表失败时，对数据包处理
路由器	路由表	目的IP地址	丢弃
交换机	MAC地址表	目的MAC地址	广播（除入端口外）
OpenFlow交换机	流表	匹配域（如目的IP地址 + 源IP地址）	询问SDN控制器

CSDN @快乐学习 ~

14. FlowVisor

FlowVisor是建立在OpenFlow协议上的网络虚拟化工具。它将物理网络划分为不同的逻辑网络，从而实现虚网划分。它让管理员通过定义流规则来管理网络，而不是修改路由器和交换机的配置。

FlowVisor部署在标准OpenFlow控制器与OpenFlow交换机之间，并对两者是透明的。它将物理网络划分为多个虚网，使每个控制器控制一个虚网，并保证各虚网相互隔离。

FlowVisor的设计原则是：

- FlowVisor对控制器和交换机是透明的，它们都感知不到FlowVisor的存在
- 各虚网之间相互隔离，即使是广播包，各虚网的流量也相互隔离
- 划分虚网的策略是灵活、模块化、可扩展的

OpenFlow消息在进行传输时，FlowVisor会根据配置策略对OpenFlow消息进行拦截、修改、转发等操作。这样，控制器就只能控制其被允许控制的流，但是控制器并不知道它所管理的网络被FlowVisor进行过分片操作。同样，交换机发出的消息经过FlowVisor过滤后，也会被发送到相应的控制器。