## 1 以太网基础知识

### 1.1 冲突域

在传统的以同轴电缆为传输介质的以太网中，同一介质上的多个节点共享链路的带宽，争用链路的使用权，这样就会发生冲突，CSMA/CD机制中当冲突发生时，网络就要进行回退，这段回退的时间内链路上不传送任何数据。而且这种情况是不可避免的。同一介质上的节点越多，冲突发生的概率越大。这种连接在同一导线上的所有节点的集合就是一个冲突域。冲突域内所有节点竞争同一带宽，一个节点发出的报文（无论是单播、组播、广播）其余节点都可以收到。

### 1.2 广播域

因为网络中使用了广播，会占用带宽，降低设备的处理效率，必须对广播加以限制。比如全1MAC地址FFFF-FFFF-FFFF为广播地址，所有节点都会处理目的地址为广播地址的数据帧。这种一个节点发送一个广播报文其余节点都能够收到的节点的集合，就是一个广播域。传统的网桥可以根据MAC表对单播报文进行转发，对于广播报文向所有的接口都转发，所以网桥的所有接口连接的节点属于一个广播域，但是每个接口属于一个单独冲突域。

### 1.3 二层交换原理

二层交换设备工作在OSI模型的第二层，即数据链路层，它对数据包的转发是建立在MAC地址基础之上的。二层交换设备不同的接口发送和接收数据独立，各接口属于不同的冲突域，因此有效地隔离了网络中物理层冲突域，使得通过它互连的主机或网络之间不必再担心流量大小对于数据发送冲突的影响。

二层交换设备通过解析和学习以太网帧的源MAC来维护MAC地址与接口的对应关系（保存MAC与接口对应关系的表称为MAC表），通过其目的MAC来查找MAC表决定向哪个接口转发，基本流程如下：

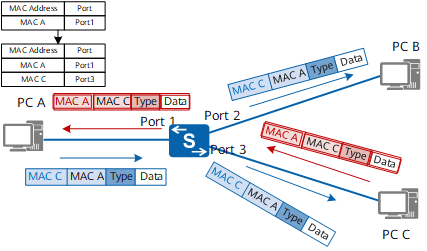
1. 二层交换设备收到以太网帧，将其源MAC与接收接口的对应关系写入MAC表，作为以后的二层转发依据。如果MAC表中已有相同表项，那么就刷新该表项的老化时间。MAC表表项采取一定的老化更新机制，老化时间内未得到刷新的表项将被删除掉。

2. 设备判断目的MAC地址是不是广播地址：

2.1 如果目的MAC地址是广播地址，那么向所有接口转发（报文的入接口除外）。

2.1 如果目的MAC地址不是广播地址，根据以太网帧的目的MAC去查找MAC表，如果能够找到匹配表项，则向表项所示的对应接口转发，如果没有找到匹配表项，那么向所有接口转发（报文的入接口除外）。

从上述流程可以看出，二层交换通过维护MAC表以及根据目的MAC查表转发，有效的利用了网络带宽，改善了网络性能。如图是一个二层交换的示例。



二层交换设备虽然能够隔离冲突域，但是它并不能有效的划分广播域。因为从前面介绍的二层交换设备转发流程可以看出，广播报文以及目的MAC查找失败的报文会向除报文的入接口之外的其它所有接口转发，当网络中的主机数量增多时，这种情况会消耗大量的网络带宽，并且在安全性方面也带来一系列问题。当然，通过路由器来隔离广播域是一个办法，但是由于路由器的高成本以及转发性能低的特点使得这一方法应用有限。基于这些情况，二层交换中出现了VLAN技术。

### 1.4 三层交换原理

早期的网络中一般使用二层交换机来搭建局域网，而不同局域网之间的网络互通由路由器来完成。那时的网络流量，局域网内部的流量占了绝大部分，而网络间的通信访问量比较少，使用少量路由器已经足够应付了。

但是，随着数据通信网络范围的不断扩大，网络业务的不断丰富，网络间互访的需求越来越大，而路由器由于自身成本高、转发性能低、接口数量少等特点无法很好的满足网络发展的需求。因此出现了三层交换机这样一种能实现高速三层转发的设备。

当然，三层交换机并不能完全替代路由器，路由器所具备的丰富的接口类型、良好的流量服务等级控制、强大的路由能力等仍然是三层交换机的薄弱环节。

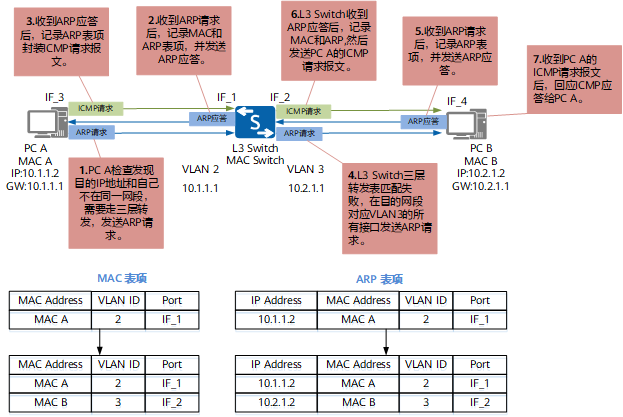
目前的三层交换机一般是通过VLAN来划分二层网络并实现二层交换，同时能够实现不同VLAN间的三层IP互访。不同网络的主机之间互访的流程简要如下：

1. 源主机在发起通信之前，将自己的IP与目的主机的IP进行比较，如果两者位于同一网段（用网络掩码计算后具有相同的网络号），那么源主机直接向目的主机发送ARP请求，在收到目的主机的ARP应答后获得对方的物理层（MAC）地址，然后用对方MAC地址作为报文的目的MAC地址进行报文发送。

2. 当源主机判断目的主机与自己位于不同网段时，它会通过网关（Gateway）来递交报文，即发送ARP请求来获取网关IP地址对应的MAC，在得到网关的ARP应答后，用网关MAC作为报文的目的MAC发送报文。此时发送报文的源IP是源主机的IP，目的IP仍然是目的主机的IP。

下面详细介绍一下三层交换的过程。

如图所示，通信的源、目的主机连接在同一台三层交换机上，但它们位于不同VLAN（网段）。对于三层交换机来说，这两台主机都位于它的直连网段内，它们的IP对应的路由都是直连路由。



图中标明了两台主机的MAC、IP地址、网关，以及三层交换机的MAC、不同VLAN配置的三层接口IP。当 PC A向PC B发起PING时，流程如下：（假设三层交换机上还未建立任何硬件转发表项）

1. 根据前面的描述，PC A首先检查出目的IP地址10.2.1.2（PC B）与自己不在同一网段，因此它发出请求网关地址10.1.1.1对应MAC的ARP请求；

2. L3 Switch收到PC A的ARP请求后，检查请求报文发现被请求IP是自己的三层接口IP，因此发送ARP应答并将自己的三层接口MAC（MAC Switch）包含在其中。同时它还会把PC A的IP地址与MAC地址对应（10.1.1.2与MAC A）关系记录到自己的ARP表项中去（因为ARP请求报文中包含了发送者的IP和MAC）；

3. PC A得到网关（L3 Switch）的ARP应答后，组装ICMP请求报文并发送，报文的目的MAC（即DMAC）＝MAC Switch、源MAC（即SMAC）＝MAC A、源IP（即SIP）＝10.1.1.2、目的IP（即DIP）＝10.2.1.2；

4. L3 Switch收到报文后，首先根据报文的源MAC+VLAN ID更新MAC表。然后，根据报文的目的MAC＋VLAN ID查找MAC地址表，发现匹配了自己三层接口MAC的表项，说明需要作三层转发，于是继续查找交换芯片的三层表项；

5. 交换芯片根据报文的目的IP去查找其三层表项，由于之前未建立任何表项，因此查找失败，于是将报文送到CPU去进行软件处理；

6. CPU根据报文的目的IP去查找其软件路由表，发现匹配了一个直连网段（PC B对应的网段），于是继续查找其软件ARP表，仍然查找失败。然后L3 Switch会在目的网段对应的VLAN 3的所有接口发送请求地址10.2.1.2对应MAC的ARP请求；

7. PC B收到L3 Switch发送的ARP请求后，检查发现被请求IP是自己的IP，因此发送ARP应答并将自己的MAC（MAC B）包含在其中。同时，将L3 Switch的IP与MAC的对应关系（10.2.1.1与MAC Switch）记录到自己的ARP表中去；

8. L3 Switch收到PC B的ARP应答后，将其IP和MAC对应关系（10.2.1.2与MAC B）记录到自己的ARP表中去，并将PC A的ICMP请求报文发送给PC B，报文的目的MAC修改为PC B的MAC（MAC B），源MAC修改为自己的MAC（MAC Switch）。同时，在交换芯片的三层表项中根据刚得到的三层转发信息添加表项（内容包括IP、MAC、出口VLAN、出接口），这样后续的PC A发往PC B的报文就可以通过该硬件三层表项直接转发了；

9. PC B收到L3 Switch转发过来的ICMP请求报文以后，回应ICMP应答给PC A。ICMP应答报文的转发过程与前面类似，只是由于L3 Switch在之前已经得到PC A的IP和MAC对应关系了，也同时在交换芯片中添加了相关三层表项，因此这个报文直接由交换芯片硬件转发给PC A；

这样，后续的往返报文都经过查MAC表到查三层转发表的过程由交换芯片直接进行硬件转发了。

从上述流程可以看出，三层交换机正是充分利用了“一次路由（首包CPU转发并建立三层硬件表项）、多次交换（后续包芯片硬件转发）”的原理实现了转发性能与三层交换的完美统一。

## 2 链路聚合

### 2.1 链路聚合概述

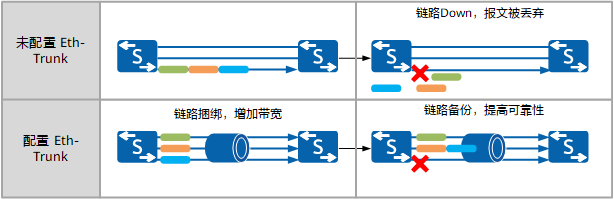
链路聚合通过将多个物理接口捆绑为一个逻辑接口，可以在不进行硬件升级的条件下，达到增加链路带宽的目的。

链路聚合技术主要有以下三个优势：

1. 增加带宽，链路聚合接口的最大带宽可以达到各成员接口带宽之和。

2. 提高可靠性，当某条活动链路出现故障时，流量可以切换到其他可用的成员链路上，从而提高链路聚合接口的可靠性。

3. 负载分担，在一个链路聚合组内，可以实现在各成员活动链路上的负载分担。



**链路聚合组和链路聚合接口**

链路聚合组LAG（Link Aggregation Group）是指将若干条以太链路捆绑在一起所形成的逻辑链路。

每个聚合组唯一对应着一个逻辑接口，这个逻辑接口称之为链路聚合接口。链路聚合接口可以作为普通的以太网接口来使用，与普通以太网接口的差别在于：转发的时候链路聚合组需要从成员接口中选择一个或多个接口来进行数据转发。

**成员接口和成员链路**

组成链路聚合接口的各个物理接口称为成员接口。成员接口对应的链路称为成员链路。

**活动接口和非活动接口、活动链路和非活动链路**

链路聚合组的成员接口存在活动接口和非活动接口两种。转发数据的接口称为活动接口，不转发数据的接口称为非活动接口。

活动接口对应的链路称为活动链路，非活动接口对应的链路称为非活动链路。

**活动接口数上限阈值**

设置活动接口数上限阈值的目的是在保证带宽的情况下提高网络的可靠性。当前活动接口数目达到上限阈值时，再向链路聚合接口中添加成员接口，不会增加链路聚合接口种活动接口的数目，超过上限阈值的链路状态将被置为Down，作为备份链路。

例如，有8条无故障链路在一个链路聚合接口内，每条链路都能提供1G的带宽，现在最多需要5G的带宽，那么上限阈值就可以设为5或者更大的值。其他的链路就自动进入备份状态以提高网络的可靠性。

**活动接口下限阈值**

设置活动接口数下限阈值是为了保证最小带宽，当前活动链路数目小于下限阈值时，链路聚合接口接口的状态转为Down。

例如，每条物理链路能提供1G的带宽，现在最小需要2G的带宽，那么活动接口数下限阈值必须要大于等于2。

**链路聚合模式**

根据是否启用链路聚合控制协议LACP（Link Aggregation Control Protocol），链路聚合分为手工模式和LACP模式。

### 2.1 手工模式链路聚合

手工模式下，链路聚合组建立、成员接口的加入由手工配置，没有链路聚合控制协议LACP的参与。该模式下所有活动链路都参与数据的转发，平均分担流量。如果某条活动链路故障，链路聚合组自动在剩余的活动链路中平均分担流量。

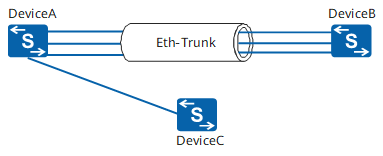
当需要在两个直连设备之间提供一个较大的链路带宽，而其中一端或两端设备都不支持LACP协议时，可以配置手工模式链路聚合。

### 2.2 LACP模式链路聚合

链路聚合控制协议LACP（Link Aggregation Control Protocol），是基于IEEE802.3ad标准的一种实现链路动态聚合与解聚合的协议，以供设备根据自身配置自动形成聚合链路并启动聚合链路收发数据，LACP模式就是采用LACP的一种链路聚合模式。聚合链路形成以后，LACP负责维护链路状态，在聚合条件发生变化时，自动调整链路聚合。

如图所示，DeviceA与DeviceB之间创建链路聚合接口，需要将DeviceA上的四个接口与DeviceB捆绑成一个链路聚合组。由于错将DeviceA上的一个接口与DeviceC相连，这将会导致DeviceA向DeviceB传输数据时可能会将本应该发到DeviceB的数据发送到DeviceC上。而手工模式的链路聚合接口不能及时检测到此故障。

如果在DeviceA和DeviceB上都启用LACP协议，经过协商后，链路聚合接口就会选择正确连接的链路作为活动链路来转发数据，从而DeviceA发送的数据能够正确到达DeviceB。



#### 2.2.1 系统LACP优先级

系统LACP优先级是为了区分两端设备优先级的高低而配置的参数。LACP模式下，两端设备所选择的活动接口必须保持一致，否则链路聚合组就无法建立。此时可以使其中一端具有更高的优先级，另一端根据高优先级的一端来选择活动接口即可。系统LACP优先级值越小优先级越高。

#### 2.2.2 接口LACP优先级

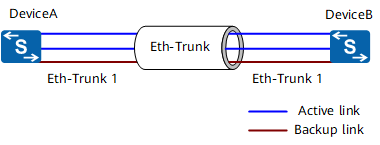
接口LACP优先级是为了区别同一个链路聚合组中的不同接口被选为活动接口的优先程度，优先级高的接口将优先被选为活动接口。接口LACP优先级值越小，优先级越高。

#### 2.2.3 成员接口间M:N备份

LACP模式链路聚合由LACP确定聚合组中的活动和非活动链路，又称为M:N模式，即M条活动链路与N条备份链路的模式。这种模式提供了更高的链路可靠性，并且可以在M条链路中实现不同方式的负载均衡。

如图所示，两台设备间有M+N条链路，在聚合链路上转发流量时在M条链路上分担负载，即活动链路，不在另外的N条链路转发流量，这N条链路提供备份功能，即备份链路。此时链路的实际带宽为M条链路的总和，但是能提供的最大带宽为M+N条链路的总和。

当M条链路中有一条链路故障时，LACP会从N条备份链路中找出一条优先级高的可用链路替换故障链路。此时链路的实际带宽还是M条链路的总和，但是能提供的最大带宽就变为M+N-1条链路的总和。



这种场景主要应用在只向用户提供M条链路的带宽，同时又希望提供一定的故障保护能力时。当有一条链路出现故障，系统能够自动选择一条优先级最高的可用备份链路变为活动链路。

如果在备份链路中无法找到可用链路，并且目前处于活动状态的链路数目低于配置的活动接口数下限阈值，那么系统将会把聚合接口关闭。

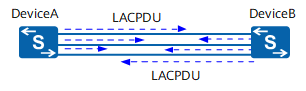
#### 2.2.4 LACP模式实现原理

LACP通过链路聚合控制协议数据单元LACPDU（Link Aggregation Control Protocol Data Unit）与对端交互信息，LACPDU报文中包含设备的系统优先级、MAC地址、接口优先级、接口号和操作Key等信息。

LACP模式链路聚合组建立的过程如下：

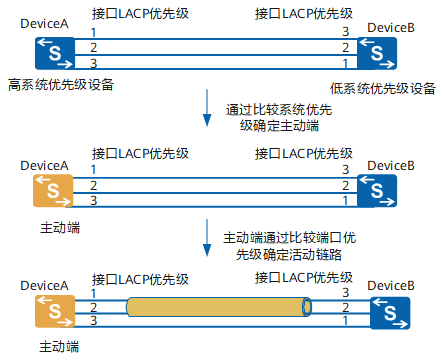
在LACP模式的Eth-Trunk中加入成员接口后，两端互相发送LACPDU报文。

如图所示，在DeviceA和DeviceB上创建链路聚合组并配置为LACP模式，然后向链路聚合组中手工加入成员接口。此时成员接口上便启用了LACP协议，两端互发LACPDU报文。



如图所示，两端设备均会收到对端发来的LACPDU报文。以DeviceB为例，当DeviceB收到DeviceA发送的报文时，DeviceB会查看并记录对端信息，然后比较系统优先级字段，如果DeviceA的系统优先级高于本端的系统优先级，则确定DeviceA为LACP主动端。如果DeviceA和DeviceB的系统优先级相同，比较两端设备的MAC地址，MAC地址小的一端为LACP主动端。

选出主动端后，两端都会以主动端的接口优先级来选择活动接口，如果主动端的接口优先级都相同则选择接口编号比较小的为活动接口。两端设备选择了一致的活动接口，活动链路组便可以建立起来，从这些活动链路中以负载分担的方式转发数据。



#### 2.2.5 手工模式和LACP模式链路聚合的对比

| **维度** | **手工模式** | **LACP模式** |
| --- | --- | --- |
| Eth-Trunk的建立方式 | Eth-Trunk的建立、成员接口的加入由手工配置，没有链路聚合控制协议的参与。 | Eth-Trunk的建立是基于LACP协议的，LACP为交换数据的设备提供一种标准的协商方式，以供系统根据自身配置自动形成聚合链路并启动聚合链路收发数据。聚合链路形成以后，负责维护链路状态。在聚合条件发生变化时，自动调整或解散链路聚合。 |
| 设备是否需要支持LACP协议 | 不需要 | 需要 |
| 数据转发 | 正常情况下，所有链路都是活动链路。所有活动链路均参与数据转发。如果某条活动链路故障，链路聚合组自动在剩余的活动链路中分担流量。 | 正常情况下，部分链路是活动链路。所有活动链路均参与数据转发。如果某条活动链路故障，链路聚合组自动在非活动链路中选择一条链路作为活动链路，参与数据转发的链路数目不变。 |
| 检测故障 | 只能检测到同一聚合组内的成员链路有断路等有限故障，但是无法检测到链路断连、错连等故障。 | 不仅能够检测到同一聚合组内的成员链路有断路等有限故障，还可以检测到链路故障、链路错连等故障。 |

#### 2.2.5 链路聚合负载分担方式

用户可以根据流量模型设置不同的负载分担方式，流量中某个参数变化越频繁，选择对应负载分担方式的流量就越均衡。例如，在网络中，如果报文的IP地址变化较频繁，那么选择基于目的IP地址、源IP地址或源IP和目的IP地址的负载分担模式更有利于流量在各物理链路间合理的负载分担；如果报文的MAC地址变化较频繁，IP地址比较固定，那么选择基于目的MAC地址、源MAC地址或源MAC和目的MAC地址的负载分担模式更有利于流量在各物理链路间合理的负载分担。

交换机可以基于报文的以下参数进行负载分担：

1. 源MAC地址

2. 目的MAC地址

3. 源MAC地址和目的MAC地址

4. 源IP地址

5. 目的IP地址

6. 源IP地址和目的IP地址

7. VLAN、源物理端口等

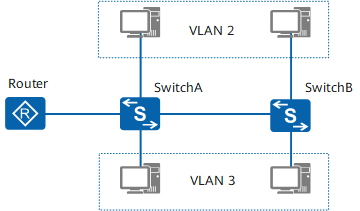
## 3 VLAN

### 3.1 VLAN简介

VLAN（Virtual Local Area Network）即虚拟局域网，是将一个物理的LAN在逻辑上划分成多个广播域的通信技术。

早期以太网是一种基于CSMA/CD（Carrier Sense Multiple Access/Collision Detection）的共享通讯介质的数据网络通讯技术。当主机数目较多时会导致冲突严重、广播泛滥、性能显著下降甚至造成网络不可用等问题。通过交换机实现LAN互连虽然可以解决冲突严重的问题，但仍然不能隔离广播报文和提升网络质量。

在这种情况下出现了VLAN技术，这种技术可以把一个LAN划分成多个逻辑的VLAN，每个VLAN是一个广播域，VLAN内的主机间通信就和在一个LAN内一样，而VLAN间则不能直接互通，这样，广播报文就被限制在一个VLAN内。



如图是一个典型的VLAN应用组网图。两台交换机放置在不同的地点，比如写字楼的不同楼层。每台交换机分别连接两台计算机，他们分别属于两个不同的VLAN，比如不同的企业客户。

使用VLAN能给用户带来以下受益：

1. 限制广播域：广播域被限制在一个VLAN内，节省了带宽，提高了网络处理能力。

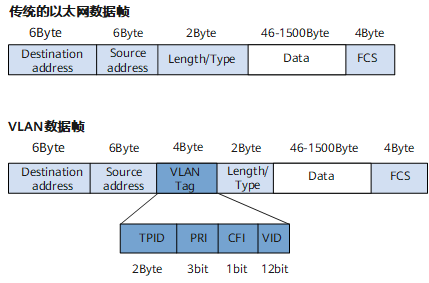
2. 增强局域网的安全性：不同VLAN内的报文在传输时是相互隔离的，即一个VLAN内的用户不能和其它VLAN内的用户直接通信。

3. 提高了网络的健壮性：故障被限制在一个VLAN内，本VLAN内的故障不会影响其他VLAN的正常工作。

4. 灵活构建虚拟工作组：用VLAN可以划分不同的用户到不同的工作组，同一工作组的用户也不必局限于某一固定的物理范围，网络构建和维护更方便灵活。

### 3.2 VLAN标签

要使交换机能够分辨不同VLAN的报文，需要在报文中添加标识VLAN信息的字段。IEEE 802.1Q协议规定，在以太网数据帧的目的MAC地址和源MAC地址字段之后、协议类型字段之前加入4个字节的VLAN标签（又称VLAN Tag，简称Tag），用以标识VLAN信息。如图所示。



VLAN标签包含4个字段，各字段含义如表所示：

| **字段** | **长度** | **含义** | **取值** |
| --- | --- | --- | --- |
| TPID | 2Byte | Tag Protocol Identifier（标签协议标识符），表示数据帧类型。 | 表示帧类型，取值为0x8100时表示IEEE 802.1Q的VLAN数据帧。如果不支持802.1Q的设备收到这样的帧，会将其丢弃。  各设备厂商可以自定义该字段的值。当邻居设备将TPID值配置为非0x8100时， 为了能够识别这样的报文，实现互通，必须在本设备上修改TPID值，确保和邻居设备的TPID值配置一致。 |
| PRI | 3bit | Priority，表示数据帧的802.1p优先级。 | 取值范围为0～7，值越大优先级越高。当网络阻塞时，交换机优先发送优先级高的数据帧。 |
| CFI | 1bit | Canonical Format Indicator（标准格式指示位），表示MAC地址在不同的传输介质中是否以标准格式进行封装，用于兼容以太网和令牌环网。 | CFI取值为0表示MAC地址以标准格式进行封装，为1表示以非标准格式封装。在以太网中，CFI的值为0。 |
| VID | 12bit | VLAN ID，表示该数据帧所属VLAN的编号。 | VLAN ID取值范围是0～4095。由于0和4095为协议保留取值，所以VLAN ID的有效取值范围是1～4094。 |

交换机利用VLAN标签中的VID来识别数据帧所属的VLAN，广播帧只在同一VLAN内转发，这就将广播域限制在一个VLAN内。

在一个VLAN交换网络中，以太网帧主要有以下两种形式：

有标记帧（Tagged帧）：加入了4字节VLAN标签的帧。

无标记帧（Untagged帧）：原始的、未加入4字节VLAN标签的帧。

常用设备中：

用户主机、服务器、Hub、无管理型交换机只能收发Untagged帧。

交换机、路由器和AC既能收发Tagged帧，也能收发Untagged帧。

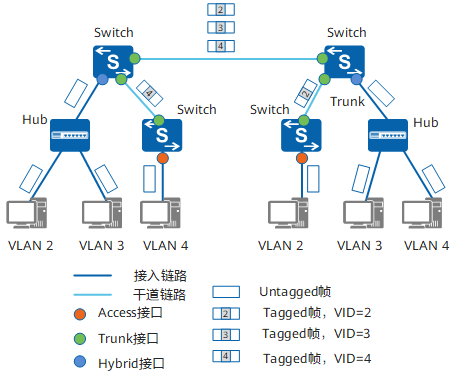
语音终端、AP等设备可以收发一个VLAN的Tagged帧或Untagged帧。

为了提高处理效率，交换机内部处理的数据帧一律都是Tagged帧。

### 3.3 接口的链路类型

交换机内部处理的数据帧一律都带有VLAN标签，而现网中交换机连接的设备有些只会收发Untagged帧，要与这些设备交互，就需要接口能够识别Untagged帧并在收发时给帧添加、剥除VLAN标签。同时，现网中属于同一个VLAN的用户可能会被连接在不同的交换机上，且跨越交换机的VLAN可能不止一个，如果需要用户间的互通，就需要交换机间的接口能够同时识别和发送多个VLAN的数据帧。

根据接口连接对象以及对收发数据帧处理的不同，一般来说分为3种接口链路类型：Access、Trunk和Hybrid，以适应不同的连接和组网。其中Access接口、Trunk接口和Hybrid接口如图所示。



#### 3.3.1 Access接口

Access接口一般用于和不能识别Tag的用户终端（如用户主机、服务器等）相连，或者不需要区分不同VLAN成员时使用。Access接口大部分情况只能收发Untagged帧，且只能为Untagged帧添加唯一VLAN的Tag。交换机内部只处理Tagged帧，所以Access接口需要给收到的数据帧添加VLAN Tag，也就必须配置缺省VLAN。配置缺省VLAN后，该Access接口也就加入了该VLAN。当Access接口收到带有Tag的帧，并且帧中VID与PVID相同时，Access接口也能接收并处理该帧。为了防止用户私自更改接口用途，接入其他交换设备，可以配置接口丢弃入方向带Tag的报文。

#### 3.3.2 Trunk接口

Trunk接口一般用于连接交换机、路由器、AP以及可同时收发Tagged帧和Untagged帧的语音终端。它可以允许多个VLAN的帧带Tag通过，但只允许一个VLAN的帧从该类接口上发出时不带Tag（即剥除Tag）。

#### 3.3.3 Hybrid接口

Hybrid接口既可以用于连接不能识别Tag的用户终端（如用户主机、服务器等）和网络设备（如Hub、无管理型交换机），也可以用于连接交换机、路由器以及可同时收发Tagged帧和Untagged帧的语音终端、AP。它可以允许多个VLAN的帧带Tag通过，且允许从该类接口发出的帧根据需要配置某些VLAN的帧带Tag（即不剥除Tag）、某些VLAN的帧不带Tag（即剥除Tag）。

### 3.4 缺省VLAN

缺省VLAN又称PVID（Port Default VLAN ID）。前面提到，交换机处理的数据帧都带Tag，当交换机收到Untagged帧时，就需要给该帧添加Tag，添加什么Tag，就由接口上的缺省VLAN决定。它的具体作用是：

当接口接收数据帧时，如果接口收到一个Untagged帧，交换机会根据PVID给此数据帧添加等于PVID的Tag，然后再交给交换机内部处理；如果接口收到一个Tagged帧，交换机则不会再给该帧添加接口上PVID对应的Tag。

当接口发送数据帧时，如果发现此数据帧的Tag的VID值与PVID相同，则交换机会将Tag去掉，然后再从此接口发送出去。

每个接口都有一个缺省VLAN。

对于Access接口，缺省VLAN就是它允许通过的VLAN，修改接口允许通过的VLAN即可更改接口的缺省VLAN。

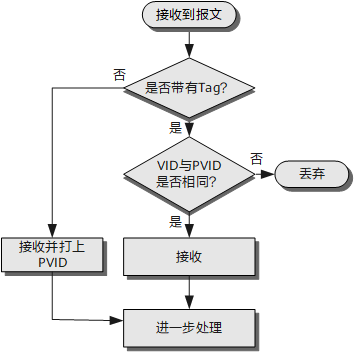
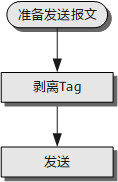
对于Trunk接口和Hybrid接口，一个接口可以允许多个VLAN通过，但是只能有一个缺省VLAN，修改接口允许通过的VLAN不会更改接口的缺省VLAN。

### 3.5 VLAN标签的添加和剥除

接口对收发的以太网数据帧添加或剥除VLAN标签的处理依据接口的接口类型和缺省VLAN。下面分别介绍Access接口、Trunk接口、Hybrid接口对收发数据帧的处理过程。

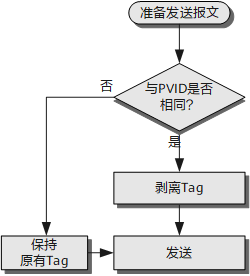
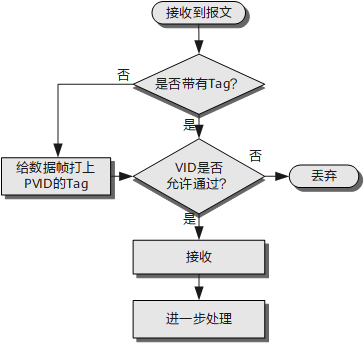
#### 3.5.1 Access接口

Access接口添加和剥除VLAN标签的处理如图所示。

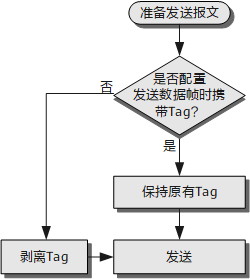
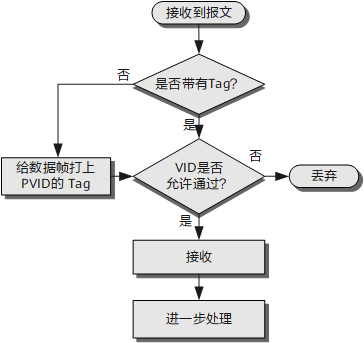
#### 3.5.2 Trunk接口

Trunk接口添加和剥除VLAN标签的处理如图所示。



#### 3.5.3 Hybrid接口

Hybrid接口添加和剥除VLAN标签的处理如[图4-8](https://support.huawei.com/enterprise/zh/doc/EDOC1100365040/f9b8a34a" \l "fig_dc_cfg_vlan_100704)和[图4-9](https://support.huawei.com/enterprise/zh/doc/EDOC1100365040/f9b8a34a" \l "fig_dc_cfg_vlan_100707)所示。



#### 3.5.4 不同类型接口添加或剥除VLAN标签的比较

| **接口类型** | **对接收不带Tag的报文处理** | **对接收带Tag的报文处理** | **发送帧处理过程** |
| --- | --- | --- | --- |
| Access接口 | 接收该报文，并打上缺省的VLAN ID。 | 当VLAN ID与缺省VLAN ID相同时，接收该报文。  当VLAN ID与缺省VLAN ID不同时，丢弃该报文。 | 先剥离帧的PVID Tag，然后再发送。 |
| Trunk接口 | 打上缺省的VLAN ID，当缺省VLAN ID在允许通过的VLAN ID列表里时，接收该报文。  打上缺省的VLAN ID，当缺省VLAN ID不在允许通过的VLAN ID列表里时，丢弃该报文。 | 当VLAN ID在接口允许通过的VLAN ID列表里时，接收该报文。  当VLAN ID不在接口允许通过的VLAN ID列表里时，丢弃该报文。 | 当VLAN ID与缺省VLAN ID相同，且是该接口允许通过的VLAN ID时，去掉Tag，发送该报文。  当VLAN ID与缺省VLAN ID不同，且是该接口允许通过的VLAN ID时，保持原有Tag，发送该报文。 |
| Hybrid接口 | 打上缺省的VLAN ID，当缺省VLAN ID在允许通过的VLAN ID列表里时，接收该报文。  打上缺省的VLAN ID，当缺省VLAN ID不在允许通过的VLAN ID列表里时，丢弃该报文。 | 当VLAN ID在接口允许通过的VLAN ID列表里时，接收该报文。  当VLAN ID不在接口允许通过的VLAN ID列表里时，丢弃该报文。 | 当VLAN ID是该接口允许通过的VLAN ID时，发送该报文。可以通过命令设置发送时是否携带Tag。 |

由上面各类接口链路类型添加或剥除VLAN标签的处理过程可见，Access接口发出的数据帧肯定不带Tag，Trunk接口发出的数据帧只有一个VLAN的数据帧不带Tag，其他都带VLAN标签，Hybrid接口发出的数据帧可根据需要设置某些VLAN的数据帧带Tag，某些VLAN的数据帧不带Tag。

## 4 pause流控

### 4.1 以太网的流量控制种类

以太网的流量控制分为两种：

一种为半双工下的流量控制，一般采用背压技术；

另一种为全双工下的流量控制，即下面要讨论的pause流控。

### 4.2 pause流控的原理和实现

#### 4.2.1 pause流控原理

IEEE802.3x协议为数据链路层的MAC控制子层提供了一个全双工流量控制结构框架，所以pause流控由MAC控制子层来处理。

PAUSE流控是一种简单的停等机制，当本端接收数据过多，导致处理压力大时，可以向对端发送一个pause消息（time不为0，一般为0xFFFF），让对端暂停给本端发送数据。

当本端压力消除时，再发送一个pause消息（time为0），让对端继续给本端发送数据。如果本端设备持续拥塞的情况下，本端端口通常会连续发送多个PAUSE帧给对端。只要本端设备的拥塞状态没有解除，相关的端口就会一直发送PAUSE。新的pause报文暂停时间会覆盖上一个暂停时间，当对端接收到新的pause帧后，暂停时间以最新时间为准。

如果pause帧指定的暂停时间超出，对端也会从暂停的位置继续发送数据帧。

#### 4.2.2 pause帧位于网络协议中的哪一层

PAUSE帧位于网络报文协议中的数据链路层（详细点讲应该是数据链路层中的MAC控制子层）。

IEEE802.3 将数据链路层分为三层：LLC，MAC控制子层（可选）和MAC介质访问控制子层。

MAC控制子层规定了通用的全双工流量控制结构。

交换控制电路要防止缓冲区溢出，可以利用MAC控制子层来控制以太网介质访问控制子层的操作。当已用缓冲区容量达到一个预先设定的阈值时，端口向全双工链路对方发出停止发送数据的请求，这个请求通过MAC控制子层产生的控制帧实现。

同样，端口可以接收由其他站点MAC控制子层产生的控制帧，控制帧夹在客户数据帧流中发送，接收方会根据帧的内容将控制帧分离出来，提交到MAC控制子层中的流量控制模块，流量控制模块解析控制帧的内容，提取帧中的控制参数，根据控制参数决定暂停发送的时间。

#### 4.2.3 pause帧消息格式

**PAUSE帧的帧长为64字节，**PAUSE帧各个字段的定义如下：

目的地址 源地址 类型 操作码 操作参数 保留 校验序列

(6字节) (6字节) (2字节) (2字节) (2字节) (42字节) (4字节)

0180-C200-0001 xxxx 0x8808 0x0001 时间参数 CRC

目的地址：协议规定PAUSE的目的地址为保留的组播地址0x01-80-C2-00-00-01。

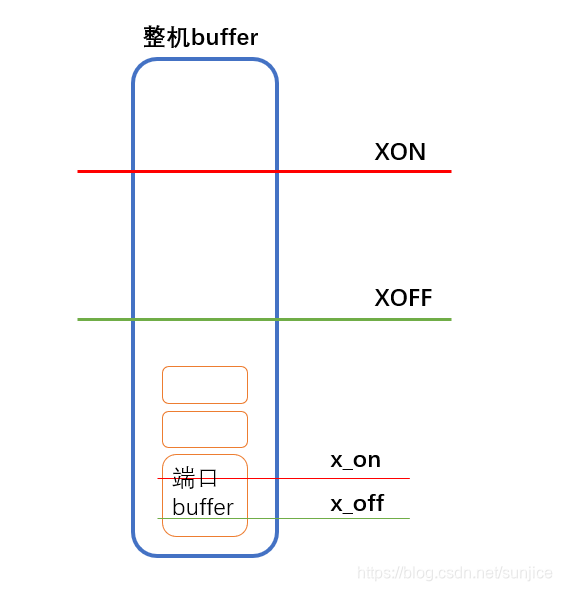
源地址：发送PAUSE帧端口的48位MAC地址。

协议类型：0x8808，MAC控制帧。

操作码：恒为0x0001。

参数-停顿时间：2字节的暂停时间参数。它是PAUSE发送方请求对方停止发送数据帧的时间长度，通常为0xFFFF，也有不为0xFFFF的，时间度量单位是以当前传输速率传输512位数据所用的时间，接收方实际暂停的时间为操作参数字段内容与以当前传输速率传输512位数据所用时间的乘积。

#### 4.2.4 pause流控处理逻辑



端口级流控：

对于交换机来讲，每个端口有一个rx buffer：

当本端口接收队列rx buffer中的数据超过某一阈值XON时，即触发pause流控，向对端发送一个pause消息，时间设置为0xFFFF。

当本端口接收队列rx buffer中的消息经过处理，低于某一阈值XOFF时，即解除pause流控，向对端发送一个pause消息，时间设置为0x0000。

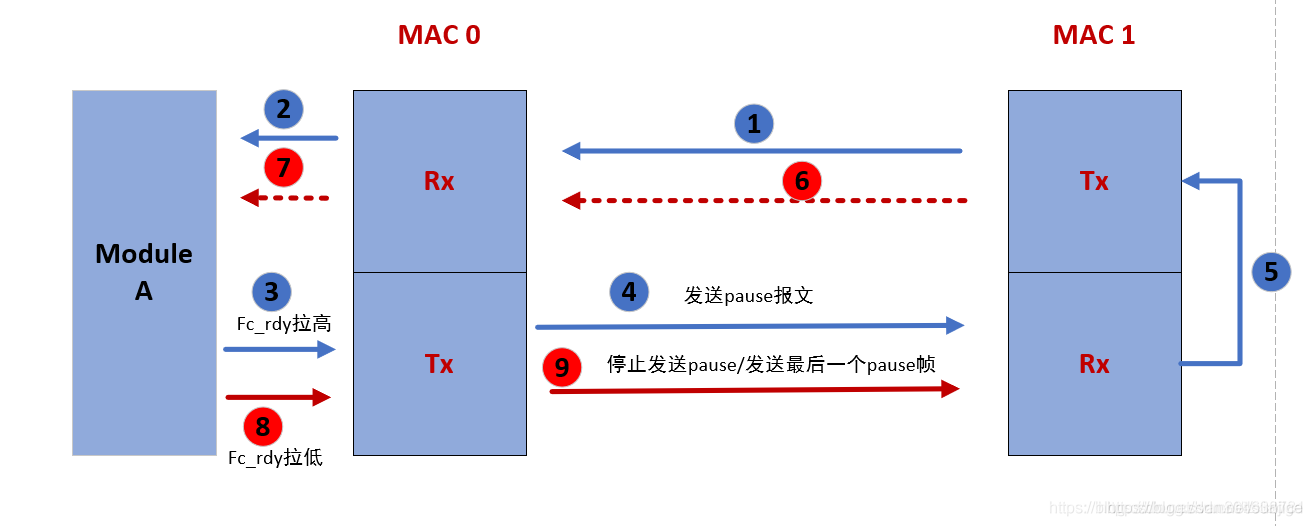
整机流控：

同时，每个端口的rx buffer又是在同一个大buffer中：

当所有端口的rx buffer总大小，超过设置的阈值XON时，会触发整机流控，此时交换机向所有端口发送pause消息；

当所有端口的rx buffer总大小，经过处理低于设置的阈值XOFF时，会解除整机流控。

#### 4.2.5 pause流控芯片上的实现



MAC0和MAC1都包含发送侧tx和接收侧rx。左侧芯片内部mac上游模块A与mac0发送侧有流控信号fc\_rdy。信号高表示模块A无法及时处理输入数据，需要进行流控。

具体流程处理如下：

1~2步：对端mac1发送数据给mac0接收侧，进行发送到模块A

3步：模块A无法即使处理输入的数据，需要减少数据输入，从而将fc\_rdy拉高。

4步：mac0发送侧tx发现流控信号fc\_rdy为高，产生pause帧，发送给mac1接收侧。只要fc\_rdy为高，mac0发送侧tx每隔一段时间发送一个pause帧，间隔时间由配置寄存器控制。间隔时长计算由计数器counting计算。Pause帧内停止发送数据的时间由另外一个配置寄存器控制。只要fc\_rdy为高期间，mac0发送侧不发送正常数据。

5步：mac1接收侧rx接受到pause报文后，提取pause帧内包含的暂停时间，控制发送侧tx停止发送数据

678步：mac1停止发送数据后，模块A处理完之前的数据后将fc\_rdy拉低，表示mac1可以继续发送数据了。

9步：第9步分2种情况。

情况1：fc\_rdy拉低，并且counting在计数没有到一个间隔周期，此时发送pause帧，但是帧内暂停时间为0。Mac1接受到pause帧后，控制tx控制立即开始发送数据。

情况2：fc\_rdy拉低的同时，counting正好计数到一个间隔周期，此时不发送pause帧。等到上一个pause帧的暂停时间到达后，mac1发送侧tx继续发送数据。

pause帧处理协议强制要求：

pause的产生发送过程不能中断一个完整的数据报文。即在第4步中，fc\_rdy拉高后，首先mac0 tx侧需要判断当前是否正常数据报文在传输。如果有，则需要在当前数据报文传输完成后才能发送pause帧。也就是说在发送过程中，只能在完整数据报文的间隙插入pause帧。

新的pause报文暂停时间会覆盖上一个暂停时间。对mac1来说，当mac1接收到新的pause帧后，暂停时间以最新时间为准。

### 4.3 pause流控开关的优缺点对比

协议虽然定义了pause流控机制，但是pause机制并不完善，它有一定的作用，也有很大的副作用，关键看厂商的实现和调校。

#### 4.3.1 pause流控的作用

pause流控，通过简单的停等处理，可以避免流量尖峰时rx buffer溢出造成的数据丢失。 所以它的作用是防止瞬间过载而数据丢失。两个字概括就是削峰：削掉流量尖峰，使数据平稳传输。

我们以100M带宽为例，假设A端真实处理能力为80M/秒，B端真实处理能力为90M/秒，来详细分析其对性能的影响：

当A端流控关闭时，那么最大通过性能即为80M，当流量在80M以下时没有任何影响，但当流量超过80M/秒时，性能会崩溃式跨掉，因为A端处理不过来会造成数据丢失，数据丢失又需要重传，而重传又要消耗一定带宽，造成恶性循环，性能就会急剧下降。

当A端流控打开时，可能真实的最大通过性能为78M/秒，比关闭时还会稍低，因为pause流控并不是在rx buffer满时才触发，而是在某一阈值（比如80%）时就触发了。当流量超过78M/秒达到85M/秒时，虽然其真正通过性能仍为78M/秒，但是经过流控的处理，由于数据并不会丢失（只是短暂的积累在B端2层），最后所有数据包仍会得到正确传输，所以性能的表象是其高峰处理能力可以达到85M甚至是90M（当然收完所有消息的时间会有所延长），这是其削峰能力的表现。

通过上面的分析可以发现，pause流控实际上是通过时间和对端的空间来换取本端的空间，保证数据不丢失。所以流控的打开和关闭，并不影响其真正的最大通过性能，而是通过削峰来保障数据的平稳传输。就好比水库蓄洪削弱洪峰一样。

#### 4.3.2 pause流控的副作用

pause流控的出发点是好的，但是实际应用中，有很大的副作用，主要表现在长时间流量过载的时候。

一是造成浪涌：当本端压力大处理不过来时，会触发pause流控，让对端停止给自己发数据；等本端压力消除后，再通知对端继续发送数据；此时由于数据量大又会立即处理不过来，触发pause流控，如此反复。网络表现是卡顿，时好时坏。

二是带来传染：当某一节点处于长期流控状态时，其相邻节点不能给其发送数据，导致数据阻塞在该相邻节点，时间一长则会造成该相邻节点也进入流控状态，这样一传十，十传百，最后会传染至整个网络，导致全网瘫痪。

#### 4.3.2 pause流控总结

对于产品来讲，需要根据自己产品的性能值、对端产品的性能值、以及产品所在的应用场景，来评估是否需要打开和关闭，特殊场景下甚至可以考虑采用半开的方式（即关闭发送pause，但不关闭响应pause）。

所以对于pause流控最好有开关可以手动控制，来满足不同的应用场景。