三层以太网交换机基本原理及转发流程

本文简要介绍了三层以太网交换机的二三层转发机制，主要目的是帮助读者进一步了解交换机的基本原理及转发流程，以期有利于更好的从事设备维护工作和建立于进一步学习的索引。

三层以太网交换机的转发机制主要分为两个部分：二层转发和三层交换。

# 二层转发流程

## MAC地址介绍

MAC地址是48 bit二进制的地址，如：00-e0-fc-00-00-06。

可以分为单播地址、多播地址和广播地址。

单播地址：第一字节最低位为0，如：00-e0-fc-00-00-06

多播地址：第一字节最低位为1，如：01-e0-fc-00-00-06

广播地址：48位全1，如：ff-ff-ff-ff-ff-ff

**注意**：

1）普通设备网卡或者路由器设备路由接口的MAC地址一定是单播的MAC地址才能保证其与其它设备的互通。

2） MAC地址是一个以太网络设备在网络上运行的基础，也是链路层功能实现的立足点。

## 二层转发介绍

交换机二层的转发特性，符合802.1D网桥协议标准。

交换机的二层转发涉及到两个关键的线程：地址学习线程和报文转发线程。

**学习线程如下：**

1）交换机接收网段上的所有数据帧，利用接收数据帧中的源MAC地址来建立MAC地址表；

2）端口移动机制：交换机如果发现一个包文的入端口和报文中源MAC地址的所在端口不同，就产生端口移动，将MAC地址重新学习到新的端口；

3）地址老化机制： 如果交换机在很长一段时间之内没有收到某台主机发出的报文，在该主机对应的MAC地址就会被删除，等下次报文来的时候会重新学习。

**注意**： 老化也是根据源MAC地址进行老化。

报文转发线程:

1）交换机在MAC地址表中查找数据帧中的目的MAC地址，如果找到，就将该数据帧发送到相应的端口，如果找不到，就向所有的端口发送；

2）如果交换机收到的报文中源MAC地址和目的MAC地址所在的端口相同，则丢弃该报文；

3）交换机向入端口以外的其它所有端口转发广播报文。

## VLAN二层转发介绍

报文转发线程:

引入了VLAN以后对二层交换机的报文转发线程产生了如下的影响：

1）交换机在MAC地址表中查找数据帧中的目的MAC地址，如果找到（同时还要确保报文的入VLAN和出VLAN是一致的），就将该数据帧发送到相应的端口，如果找不到，就向（VLAN内）所有的端口发送；

2）如果交换机收到的报文中源MAC地址和目的MAC地址所在的端口相同，则丢弃该报文；

3）交换机向（VLAN内）入端口以外的其它所有端口转发广播报文。

以太网交换机上通过引入VLAN，带来了如下的好处：

1）限制了局部的网络流量， 在一定程度上可以提高整个网络的处理能力。

2）虚拟的工作组，通过灵活的VLAN设置，把不同的用户划分到工作组内；

3）安全性，一个VLAN内的用户和其它VLAN内的用户不能互访，提高了安全性。

另外，还有常见的两个概念VLAN的终结和透传， 从字面意思上就可以很好的了解这两个概念。所谓VLAN的透传就是某个VLAN不仅在一台交换机上有效，它还要通过某种方法延伸到别的以太网交换机上，在别的设备上照样有效；终结的意思及相对，某个VLAN的有效域不能再延伸到别的设备，或者不能通过某条链路延伸到别的设备。

VLAN透传可以使用802.1Q技术，VLAN终结可以使用PVLAN技术。

IEEE802.1Q协议是VLAN的技术标准,主要是修改了标准的帧头，添加了一个tag字段，其中包含了VLAN ID等VLAN信息，具体实现这里不谈，如果有兴趣可以看相关的标准和资料。

注意：在Trunk端口转发报文的时候，如果报文的VLAN Tag等于端口上配置的默认VLAN ID，则该报文的Tag应该去掉，对端收到这个不带Tag信息的报文后， 从端口的PVID获得报文的所属VLAN信息，因此配置的时候必须保证连接两台交换机之间的一条Trunk链路两端的PVID设置相同。

为什么要去Tag呢？

这样做是为了保证一般的用户插到Trunk上以后，仍旧可以正常通信，因为普通用户无法识别带有802.1Q Vlan信息的报文。

使用802.1Q技术可以很好的实现VLAN的透传，可是有的时候需要把VLAN终结掉，也就是说这个VLAN的边界在哪里终止，PVLAN技术可以很好的实现这个功能， 同时达到节省VLAN 的目的。cisco的PVLAN意思是private vlan，而我们的PVLAN意思是primary vlan。

这里的VLAN有两类：Primary vlan和secondary vlan（子VLAN）。

实现了接入用户二层报文的隔离，同时上层交换机下发的报文可以被每一个用户接收到，简化了配置，节省了VLAN资源。具体实现这里不谈，如果有兴趣可以相关资料。

下面谈谈三层交换流程。

用VLAN分段，隔离了VLAN间的通信，用支持VLAN的路由器（三层设备）可以建立VLAN间通信。但使用路由器来互联企业园区网中不同的VLAN显然不合时代的潮流。因为我们可以使用三层交换来实现。

差别1（性能）：传统的路由器基于微处理器转发报文，靠软件处理，而三层交换机通过ASIC硬件来进行报文转发，性能差别很大；

差别2（接口类型）：三层交换机的接口基本都是以太网接口，没有路由器接口类型丰富；

差别3：三层交换机，还可以工作在二层模式，对某些不需路由的包文直接交换，而路由器不具有二层的功能。

首先让我们看一下设备互通的过程：



如图所示：交换机上划分了两个VLAN，在VLAN1，VLAN 2上配置了路由接口用来实现vlan1 和 vlan 2之间的互通。

**A和B之间的互通（以A向B发起ping请求为例）：**

1） A检查报文的目的IP地址，发现和自己在同一个网段；

2） A---->B ARP请求报文，该报文在VLAN1 内广播；

3） B---->A ARP回应报文；

4） A---->B icmp request;

5） B---->A icmp reply;

**A和C之间的互通（以A向C发起ping请求为例）：**

1） A检查报文的目的IP地址，发现和自己不在同一个网段；

2） A---->switch（int vlan 1） ARP请求报文，该报文在VLAN1 内广播；

3） 网关---->A ARP回应报文；

4） A---->switch icmp request（目的MAC是 int vlan 1的MAC，源MAC是A的MAC，目的IP是C，源IP是A）；

5） switch收到报文后判断出是三层的报文。检查报文的目的IP地址，发现是在自己的直连网段；

6） switch（int vlan 2）---->C ARP请求报文，该报文在VLAN2 内广播；

7） C--->switch（int vlan 2） ARP回应报文；

8） switch（int vlan 2）---->C icmp request （目的MAC是 C 的MAC，源MAC是 int vlan 2的MAC，目的IP是C，源IP是A）同步骤4）相比报文的MAC头进行了重新的封装， 而IP层以上的字段基本上不变；

9） C---->A icmp reply，这以后的处理同前面icmp request的过程基本相同。

以上的各步处理中，如果ARP表中已经有了相应的表项，则不会给对方发ARP请求报文。

**怎么样来区分二和三层的数据流？**

3526产品是三层以太网交换机，在其处理流程中既包括了二层的处理功能，又包括了三层的处理功能。

区别二三层转发的基本模型：



如图所示：

三层交换机划分了2个VLAN， A和B之间的通信是在一个VLAN内完成，对与交换机而言是二层数据流，A和C之间的通信需要跨越VLAN，是三层的数据流。

上面提到的是宏观的方法，具体到微观的角度，一个报文从端口进入后，Swtich设备是怎么来区分二层包文，还是三层报文的呢？

从A到B的报文由于在同一个VLAN内部， 报文的目的MAC地址将是主机B的MAC地址，而从A到C的报文，要跨越VLAN，报文的目的MAC地址是设备虚接口VLAN1上的MAC地址。

因此交换机区分二三层报文的标准就是看报文的目的MAC地址是否等于交换机虚接口上的MAC地址。

以S3526交换机为例，三层交换机整个处理流程中分成了三个大的部分：

1）平台软件协议栈部分

这部分中关键功能有：

运行路由协议，维护路由信息表；

IP 协议栈功能，在整个系统的处理流程中，这部分担负着重要的功能，当硬件不能完成报文转发的时候，这部分可以代替硬件来完成报文的三层转发。另外对交换机进行telnet, ping, ftp,snmp的数据流都是在这部分来处理。

举例：

show ip route：

Routing Tables:

Destination/Mask Proto Pre Metric Nexthop Interface

0.0.0.0/0 Static 60 0 10.110.255.9 VLAN-Interface2

10.110.48.0/21 Direct 0 0 10.110.48.1 VLAN-Interface1

10.110.48.1/32 Direct 0 0 127.0.0.1 InLoopBack0

10.110.255.8/30 Direct 0 0 10.110.255.10 VLAN-Interface2

10.110.255.10/32 Direct 0 0 127.0.0.1 InLoopBack0

127.0.0.0/8 Direct 0 0 127.0.0.1 InLoopBack0

127.0.0.1/32 Direct 0 0 127.0.0.1 InLoopBack0

维护ARP表

**show arp：**

IpAddress Mac\_Address VLAN ID Port Name Type

10.110.255.9 00e0.fc00.5518 2 GigabitEthernet2/1 Dynamic

10.110.51.75 0010.b555.f039 1 Ethernet0/9 Dynamic

10.110.54.30 0800.20aa.f41d 1 Ethernet0/10 Dynamic

10.110.51.137 0010.a4aa.fce6 1 Ethernet0/12 Dynamic

10.110.50.90 0010.b555.e04f 1 Ethernet0/8 Dynamic

2）硬件处理流程

主要的表项是：二层MAC地址表，和三层的ip fdb表，这两个表中用于保存转发信息，在转发信息比较全的情况下，报文的转发和处理全部由硬件来完成处理，不需要软件的干预。 这两个表的功能是独立的，没有相互的关系，因为一个报文只要一进入交换机，硬件就会区分出这个包是二层还是三层。非此即彼。

例如：show mac all：

MAC ADDR VLAN ID STATE PORT INDEX AGING TIME(s)

0000.21cf.73f4 1 Learned Ethernet0/19 266

0002.557c.5a79 1 Learned Ethernet0/12 225

0004.7673.0b38 1 Learned Ethernet0/9 262

0005.5d04.9648 1 Learned Ethernet0/16 232

0005.5df5.9f64 1 Learned Ethernet0/16 300

MAC地址表是精确匹配的IVL方式， 其中关键的参数是：Vlan ID, Port index。

例如：show ipfdb all：

0: System 1: Learned 2: UsrCfg Age 3: UsrCfg noAge Other: Error

Ip Address RtIf Vtag VTValid Port Mac Status

10.11.83.77 2 2 Invalid GigabitEthernet2/1 00-e0-fc-00-55-18 1

10.11.198.28 2 2 Invalid GigabitEthernet2/1 00-e0-fc-00-55-18 1

10.63.32.2 2 2 Invalid GigabitEthernet2/1 00-e0-fc-00-55-18 1

10.72.255.100 2 2 Invalid GigabitEthernet2/1 00-e0-fc-00-55-18 2

10.75.35.103 2 2 Invalid GigabitEthernet2/1 00-e0-fc-00-55-18 2

10.75.35.106 2 2 Invalid GigabitEthernet2/1 00-e0-fc-00-55-18 2

路由接口索引（RtIf）：该索引用来确定该转发表项位于哪个路由接口下面，对3526产品来讲，支持的路由接口数目是32；

Vlan tag: 该值用来表明所处的VLAN，该VLAN和路由接口是对应的；

Vlan tag有效位（VTValid）：用来标识转发出去的报文中是否需要插入Vlan tag标记。

端口索引（Port）：用来说明该转发表项的出端口；

下一跳MAC：三层设备每完成一跳的转发，会重新封装报文中的MAC头，硬件ASIC芯片一般依据这个域里面的数值来封装报文头。

两个重要的概念：

解析，未解析，每次收到报文，ASIC都会从其中提取出源和目的地址在MAC Table或者 IP Fdb Table中进行查找，如果地址在转发表中可以找到，则认为该地址是解析的，如果找不到，则认为该地址是未解析的。根据这个地址是源，还是目的，还可以有源解析，目的未解析等等的组合。

对于二层未解析，硬件本身可以将该报文在VLAN内广播，但是对于三层报文地址的未解析报文硬件本身则不对该报文进行任何的处理，而产生CPU中断，靠软件来处理。

硬件部分的处理可以用这句话来描述：

收到报文后，判断该报文是二或是三层报文，然后判断其中的源，目的地址是否已经解析，如果已经解析，则硬件完成该报文的转发，如果是未解析的情况，则产生CPU中断，靠软件来学习该未解析的地址。

3）驱动代码部分

其中关键的核心有：

地址解析任务：在该任务中对已经报上来的未解析的地址进行学习，以便硬件完成后续的报文的转发而不需软件干预。

地址管理任务：为了便于软件管理和维护，软件部分保存了一份同硬件中转发表相同的地址表copy。

fib（forwarding information base）表: 这个表的信息来源于ip route table中的路由信息，之所以把它放在了driver部分， 是为了地址解析任务在学IP地址时查找的方便。

举例：

show fib：

Destination/Mask Nexthop Flag Interface

0.0.0.0/0 10.110.255.9 I VLAN-Interface2

10.110.48.0/21 10.110.48.1 D VLAN-Interface1

10.110.48.1/32 127.0.0.1 D InLoopBack0

10.110.255.8/30 10.110.255.10 D VLAN-Interface2

10.110.255.10/32 127.0.0.1 D InLoopBack0

127.0.0.0/8 127.0.0.1 D InLoopBack0

三层转发主要涉及到两个关键的线程：

地址学习线程和报文转发线程，这个和二层的线程是类似的；

1）报文转发线程主要根据地址学习线程生成的转发表（ipfdb table）信息来对报文进行转发，如果里面的信息足够多，这个转发的过程全部由硬件来完成，如果信息不够，则会要求地址学习线程来进行学习，同时该报文硬件不能转发，会交给软件协议栈来进行转发。

2）地址学习线程主要用来生成硬件转发表（ipfdb table）

其实ipfdb table和二层的MAC地址表也是类似的，只不过里面的具体表项所代表的含义和所起的作用不同罢了。

**有一个问题**：在路由器等软件转发引擎中，每收一个报文都会去查路由表查下一跳，然后再查ARP表找下一跳的MAC，可是在三层交换机（如S3526）中，报文转发的时候不需要去查路由表和ARP表，这样的话，这两个表是不是就没有什么作用了？

回答当然是否定的，在S3526的三层转发流程中，过程一般都是这样的，第一个报文硬件无法转发，要进行IP地址的学习，同时为了保证不丢包，该报文也由软件来进行转发，在学习完成以后，第二，第三个报文以后就一直是由硬件来完成转发了，这个过程也可以套用“**一次路由，多次交换**”来形象的进行总结，在一次路由中，要利用路由表和ARP表来学习IP地址，和转发第一个报文，在以后的多次交换过程中，则只要有ipfdb table就可以了。