VLAN

一、VLAN基本概念

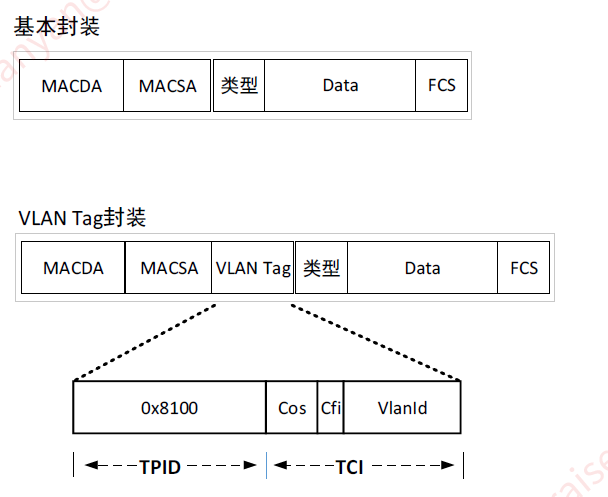
VLAN（Virtual Local Area Network），虚拟局域网，是一种通过将局域网内的设备逻辑地划分成一个个网段从而实现虚拟工作组的技术。

1，VLAN作用

未引入VLAN前，L2网络是一个大的广播域，目的地未知的报文以及广播报文会在整个交换机的所有端口泛洪，这些随意泛洪的报文会浪费端口的带宽，同时也存在安全隐患。为解决此问题，引入VLAN技术。VLAN可以将局域网内的计算机逻辑上分为不同的部分，每个部分相互独立，彼此之间不能互相访问，从而实现二层隔离。划分VLAN 的目的是隔离广播域，减少广播风暴的产生。

2，VLAN格式

引入VLAN概念后，在报文格式上的体现就是增加了VLAN Tag，VLAN Tag包含4个字段，分别是TPID (Tag Protocol ID)，CoS，CFI (canonical format indicator)和VLAN Id四个字段，分别占用16bits，3bits，1bit，12bits。TPID 用于判断本数据帧是否带有VLAN Tag，长度为16bit，802**.**1Q协议规定该字段的取值为0x8100；CoS 用于标识报文的优先级，长度为3bit，一共有8 个优先级，范围是0~7；CFI 用于标识MAC 地址在不同的传输介质中是否以经典格式进行封装，长度为1bit，取值为0表示MAC地址以标准格式进行封装，为1表示以非经典格式封装；VLAN ID 用于标识该报文所属VLAN，长度为12bit，范围为0～4095。由于0和4095 为协议保留值，所以VLAN ID 的有效范围为1～4094。



3，VLAN划分

VLAN是为解决以太网的广播问题和安全性而提出的，一个VLAN内部的广播和单播流量都不会转发到其他VLAN中。划分VLAN有助于控制流量、减少设备投资、简化网络管理、提高网络的安全性。芯片VLAN划分即VLAN Classifier，目的是根据不同规则分类出多种VLAN，并从这些VLAN中选取一个VLAN进行后续处理。一般来说，VLAN的划分可分为以下4种方式：

1）基于端口划分VLAN，是最简单、最有效的VLAN划分方法。根据端口编号划分VLAN，将指定端口加入到指定VLAN。

2）基于数据流划分VLAN，根据流量特点进行VLAN划分，包括VLAN交换机端口、MAC地址、IP地址、网络层协议等，通过hash/tcam方式进行匹配。

3）基于MAC划分VLAN，根据每台主机的MAC地址来划分VLAN。

4）基于子网划分VLAN，根据每台主机的网络地址进行划分。

4，PVLAN

PVLAN（Private VLAN，私有VLAN），提供在同一个VLAN内，接口之间的二层隔离功能，是网络中一种用于解决VLAN资源有效分配的技术。

PVLAN类型

PVLAN将VLAN划分为两种不同属性：Primary VLAN（主VLAN）和Secondary

VLAN（辅助VLAN）。Primary VLAN 和Secondary VLAN 组成一个PVLAN 域，Primary VLAN 可在PVLAN 内部和外部网络进行通信，Secondary VLAN 只能在PVLAN 内部进行通信。Secondary VLAN 根据转发隔离规则不同，分为Isolated VLAN和Community VLAN。配置成isolated VLAN不能和Secondary VLAN里任何端口通信。Community VLAN能和该Secondary VLAN里属于同一个团体内的端口通信。

PVLAN接口模式

Primary VLAN 中可与外部网络进行通信的接口为Promiscuous 接口（混杂接口），

Secondary VLAN 中的接口为Host 接口（主机接口）。根据Secondary VLAN 的两种类

型，主机接口又分为Isolated 接口和Community 接口。

 Promiscuous 接口：混杂接口，属于PVLAN 域所对应的所有PVLAN。处于该模

式下的接口可与所有的接口进行通信。

 Isolated 接口：隔离接口。处于该模式下的接口是被隔离的，互相之间不能访问，

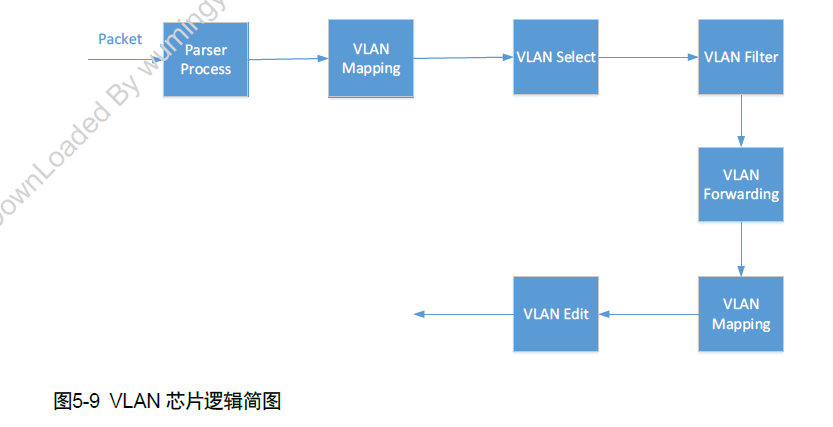
只能与Promiscuous 接口和Trunk 接口通信。

 Community 接口：团体接口。团体接口可以互相通信，但不同团体之间的接口不

能通信，所有团体接口都可以与Promiscuous 接口和Trunk 接口通信。

二、CTC芯片VLAN实现及技术原理

1，VLAN整体流程及芯片架构



芯片涉及VLAN处理流程主要有：Parser Process，Ingress VLAN Mapping，VLAN Select，VLAN Filter，VLAN Forwarding，Egress VLAN Mapping，VLAN Edit等。

Parser Process可用于解析报文携带的VLAN Tag。

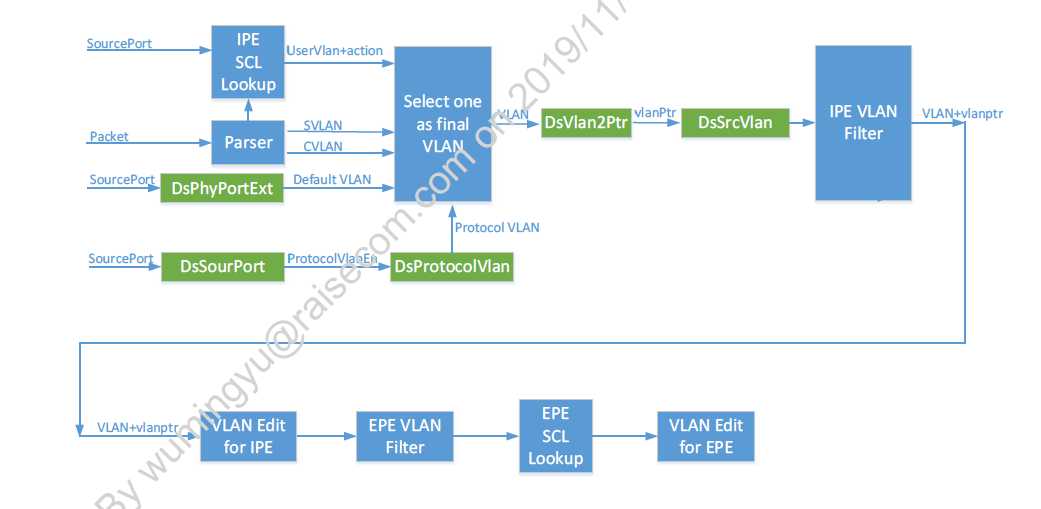
VLAN Mapping包括Ingress VLAN Mapping和Egress VLAN Mapping，是实现QinQ、VLAN Translation及VLAN Classifier的核心，对应芯片的IPE SCL查找和EPE SCL查找。SCL是芯片公共模块，主要功能是根据报文信息和端口配置信息进行组Key，并进行HASH或者TCAM类型的SCL表项查找，根据查找结果获取报文转发和编辑信息，在这里主要是针对VLAN的编辑。

VLAN Select是选择最终用于后续转发的VLAN，芯片支持基于Port，MAC，IP，Eth Type等多种方式灵活的划分VLAN，VLAN Select模块会根据优先级进行VLAN的选择。

VLAN Forwarding支持基于VLAN直接出转发信息和基于VLAN+MAC做Bridging转发。其中，基于VLAN直接出转发信息的实现需要依赖IPE SCL查找结果。

VLAN Edit可以简单理解为对报文出口转发时VLAN字段的编辑，包含添加VLAN，更新VLAN，删除VLAN。

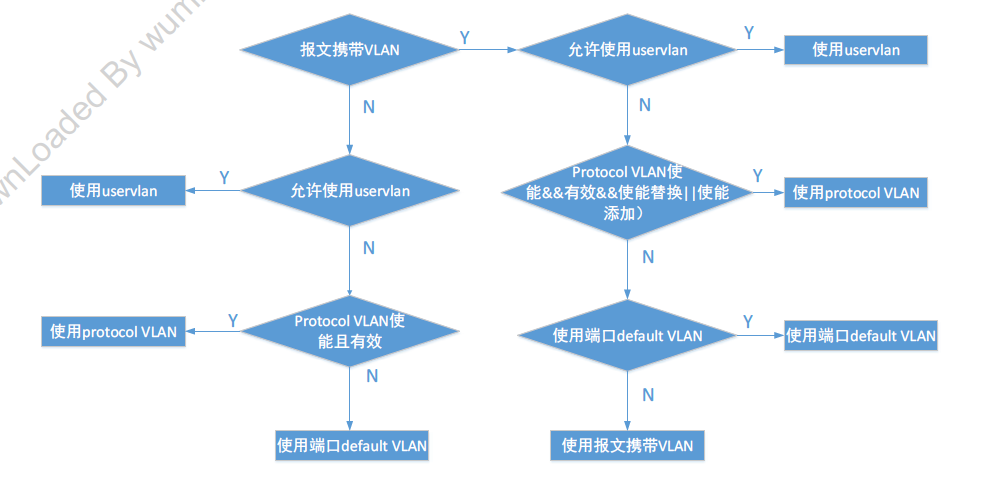
VLAN整体处理流程：



VLAN Mapping：对应IPE SCL Lookup和EPE SCL Lookup模块，主要功能是根据报文信息和端口配置信息进行组Key，并进行HASH或者TCAM类型的SCL表项查找，根据查找结果获取报文转发和VLAN编辑信息。

为了支持不同类型业务，芯片支持多次并行查找。芯片在IPE方向支持六次SCL查找，分为两次HASH和四次TCAM查找（两次HASH Lookup都查的同一块D/SRAM，四次TCAM Lookup查的是不同的TCAM）。其中，SCL HASH查找结果分别对应UserIdAdOuput0和UserIdAdOuput1。SCL TCAM查找结果分别对应SclTcam0AdOuput0，SclTcam1AdOuput1，SclTcam2AdOuput2，SclTcam3AdOuput3。最终，根据UserIdAdOuput索引到DsUserId或DsSclFlow。芯片在EPE方向只支持两次HASH查找，不支持TCAM查找。

VLAN Select：主要负责从多个VLAN中选择一个VLAN做后续处理。

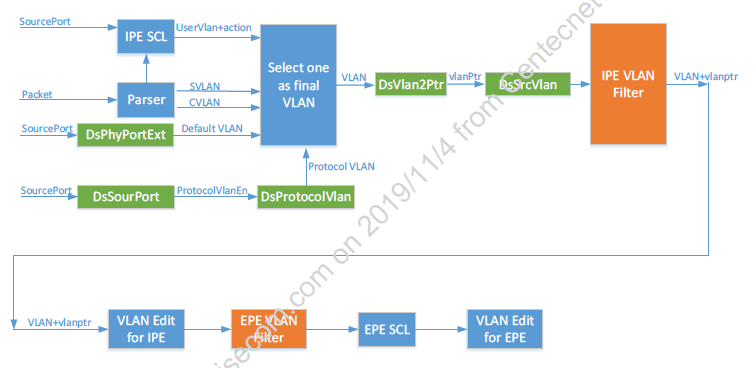


综上，VLAN选取的优先级是：

报文携带vlan：uservlan > protocol vlan > default vlan（当DsSrcPort的defaultReplaceTagEn字段为1时） > 报文携带的VLAN

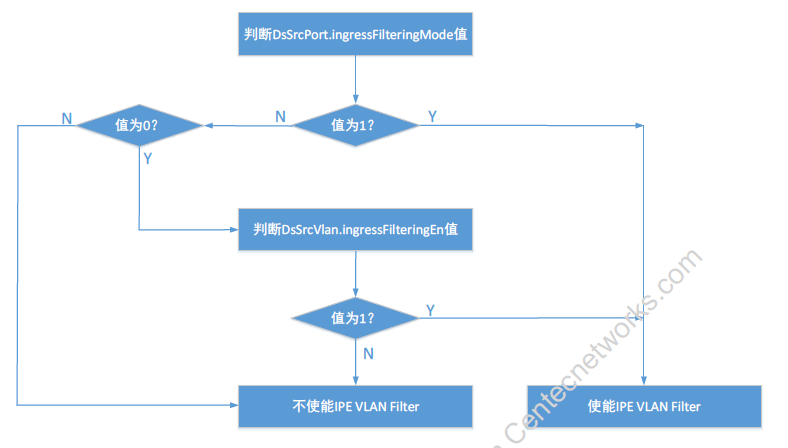
报文未携带vlan：uservlan > protocol vlan > default vlan

VLAN Filter：



在IPE和EPE方向都可以设置。主要目的是检查端口是否加入某个VLAN。具体流程是首先检查端口是否使能VLAN Filter，如果未使能不做检查，如果使能则检查端口是否允许该

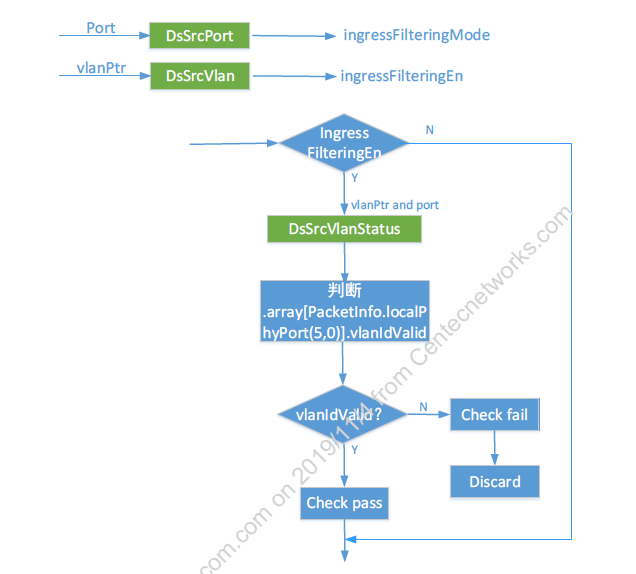
VLAN，检查通过报文正常转发，检查不通过报文作丢弃处理。



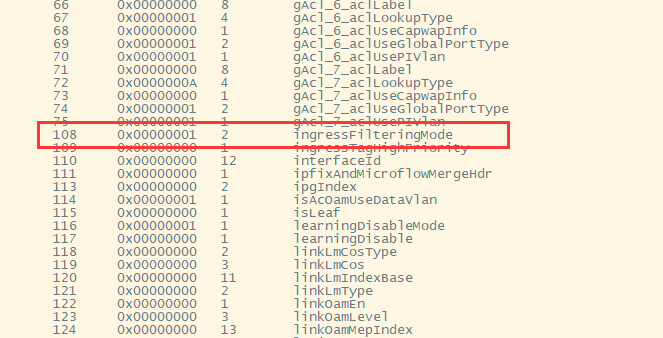
判断是否使能vlan filter流程：1）如果DsSrcPort.ingressFilteringMode值为1，表示基于端口使能IPE VLAN Filter功能；2）如果DsSrcPort.ingressFilteringMode值为0，则进一步通过DsSrcVlan.ingressFilteringEn值判断，为1表示使能，否则表示不使能；3）DsSrcPort.ingressFilteringMode为其它值，表示不使能。

IPE VLAN Filter功能流程：

首先判断是否使能VLAN Filter功能，如果未使能，直接通过；否则通过VLAN Select处理结果得到vlanPtr字段，并将其作为DsSrcVlanStatus的索引。通过DsSrcVlanStatus.array[PacketInfo.localPhyPort(7,0)].vlanIdValid字段判断报文的入端口是否允许该VLAN。如果vlanIdValid为1，表示IPE VLAN Filter检查通过，报文正常转发；如果vlanIdValid为0，表示IPE VLAN Filter检查不通过，报文作丢弃处理。



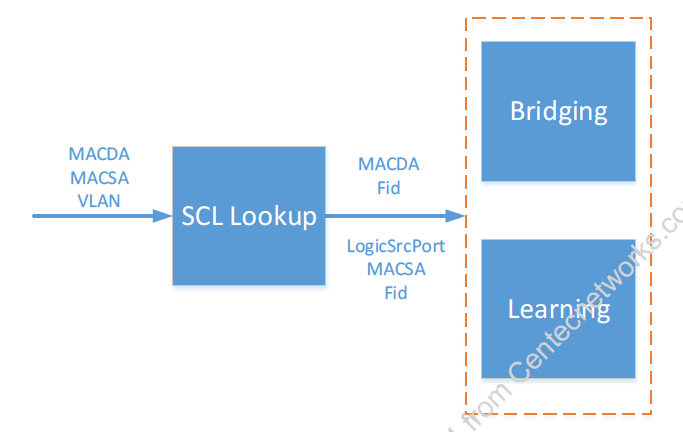
我们的设备默认是在vlan初始化时直接使能了双向vlan filter功能的，在ctc下查看对应寄存器， DsSrcPort.ingressFilteringMode值为1。



VLAN Forwarding：可以简化为两种，一是通过SCL查找出相关字段用于后续转发表项查找

（VLAN Bridging），二是通过SCL查找直接出转发信息（VLAN Switching）。

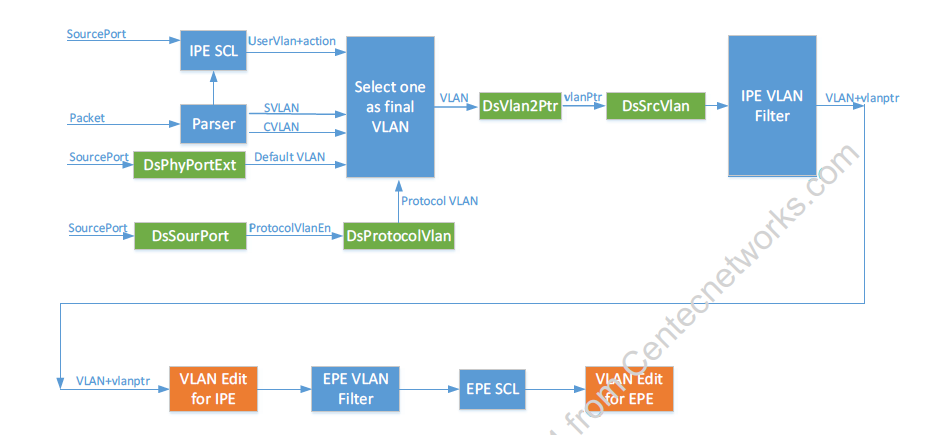
1，VLAN Bridging：根据SCL查找结果索引到DsUserId出Fid，LogicSrcPort。利用MACDA和Fid查找FDB转发表项并做Bridging转发。利用MACSA，Fid和LogicSrcPort做learning处理。



2，VLAN Switching：根据SCL查找结果的dsFwdPtr字段索引到DsFwd，DsFwd保存了转发出口信息和报文出口编辑信息。这种转发行为可以理解为利用SCL查找匹配VLAN等字段直接转发，而不需要利用Bridging或Routing的转发逻辑。



VLAN Edit：VLAN edit在EPE方向执行，对于IPE方向的VLAN编辑在EPE的开始执行，对于EPE方向的VLAN编辑在报文发出去之前执行。



IPE方向的VLAN Edit可以从IPE的SCL查找结果中VLAN action出VLAN编辑行为。EPE方向的VLAN Edit是既可以从DsNexthop出VLAN编辑行为，也可以从EPE SCL查找结果的VLAN action出VLAN编辑行为。

VLAN Edit主要包括VLAN的添加、删除、替换。

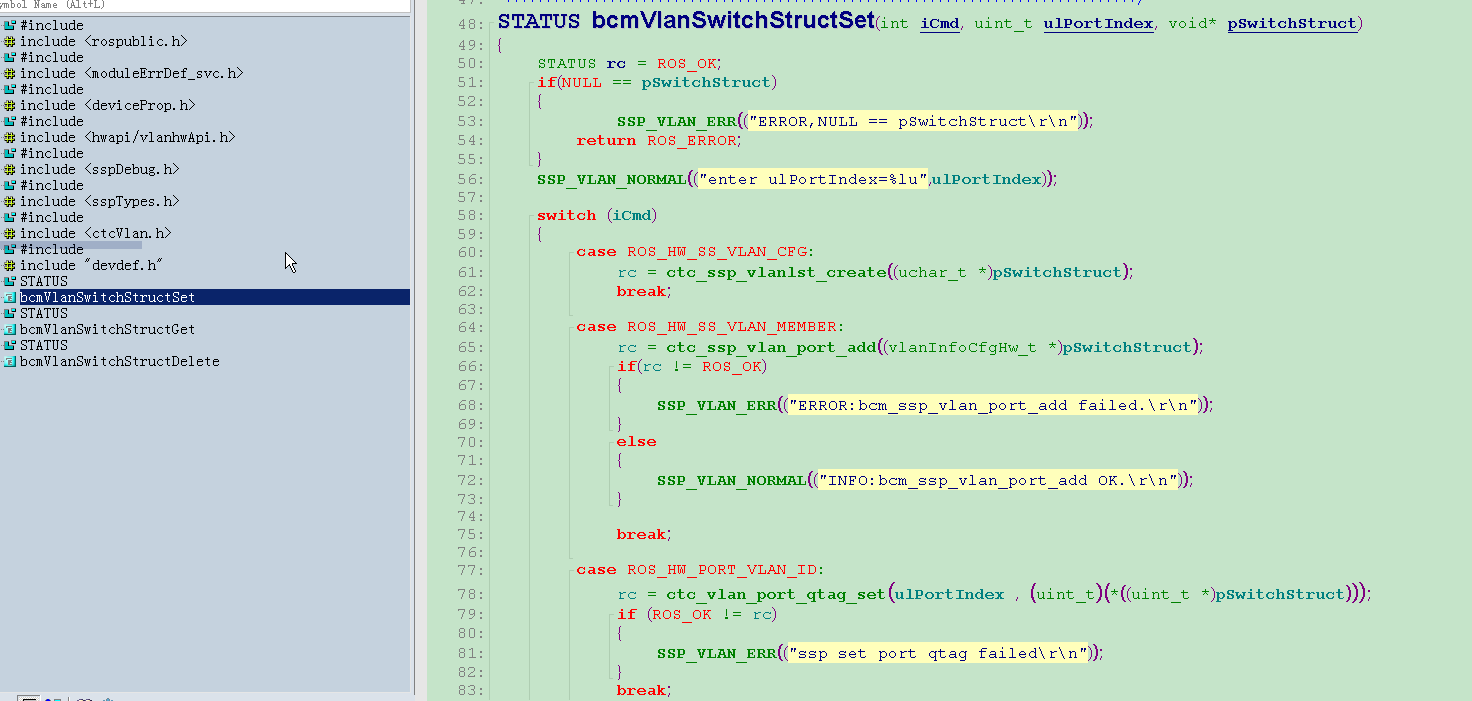
——添加VLAN Tag，只需将SCL映射出的VLAN信息组装好，找到添加的起始位置，将组装好的VLAN信息插入到相应位置，并调整报文长度。

——替换VLAN Tag，只需将SCL映射出的VLAN信息组装好，找到替换的起始位置，将组装好的VLAN信息替换原有的VLAN。

——删除VLAN Tag，只需找到VLAN Tag的起始位置，并将其所占的4个字节从报文中删掉，并调整报文长度。

三、VLAN功能配置命令及代码

VLAN相关命令功能主要包括创建vlan，配置基于access接口的vlan，基于trunk接口的vlan，基于MAC地址，基于IP子网以及基于协议的vlan，PVLAN等。代码方面主要学习研究了下配置层及芯片API部分的代码流程。主要流程通过ctcVlanHw.c里的bcmVlanSwitchStructSet/Del/Get等函数调用到ctcvlan.c下的处理函数实现。



1，创建VLAN

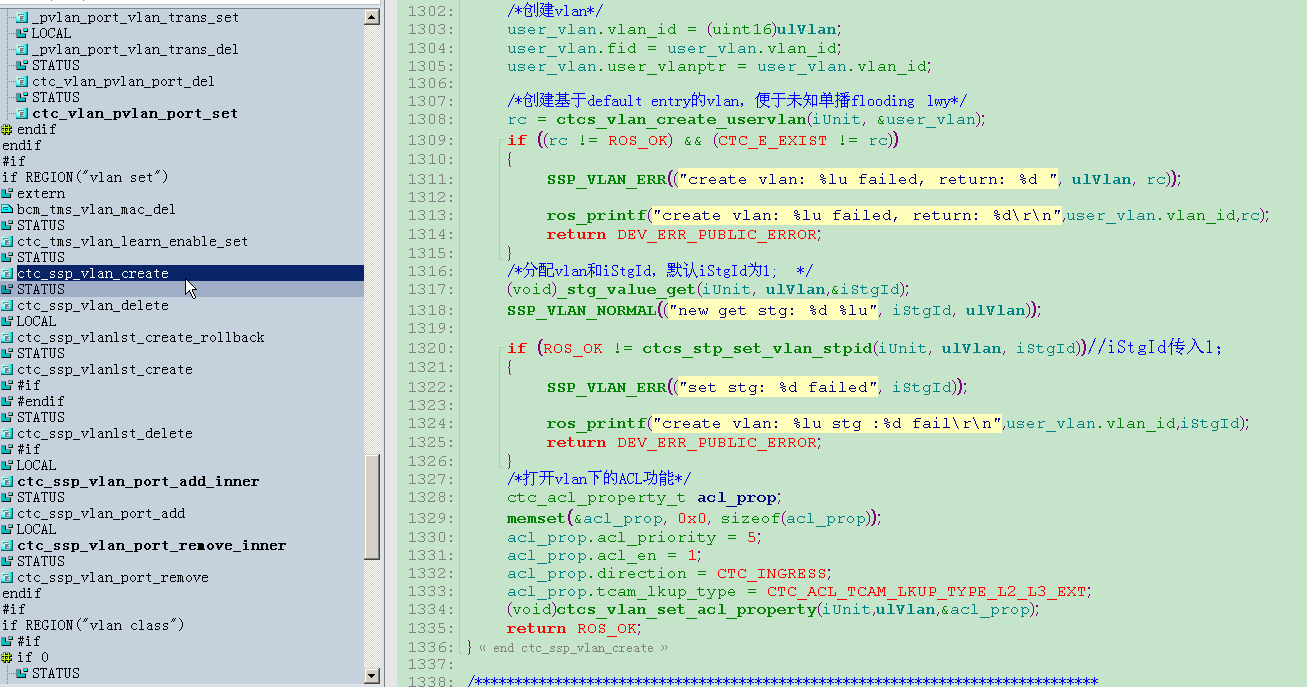
配置命令：create vlan *vlan-list* active

对应ctccli命令为：vlan create vlan VLAN\_ID (uservlanptr VLAN\_PTR fid FID …)

函数实现调用流程：

ctc\_ssp\_vlanlst\_create() --> ctc\_ssp\_vlan\_create() -->ctcs\_vlan\_create\_uservlan()

在ctc\_ssp\_vlan\_create()函数中除了创建vlan，还设置了vlan对应的stpid，以及打开了vlan下的ACL功能。



2，配置接口拒绝接收帧类型（可选）

配置命令：switchport reject-frame {tagged | untagged}

对应ctccli命令：port X vlan-ctl {allow-all | drop-all-tagged | drop-all-untagged | drop-all …}

通过端口vlan-ctl属性设置端口是否丢弃或允许某一类型报文，默认选项为allow-all。

函数实现调用流程：

ctc\_vlan\_port\_frame\_reject\_type\_set() --> ctcs\_port\_set\_vlan\_ctl()

3，配置基于Access口的VLAN（trunk native vlan，allowed vlan等类似）

配置命令：switchport access vlan *vlan-id*

对应ctccli命令： vlan add port X vlan VLANID，

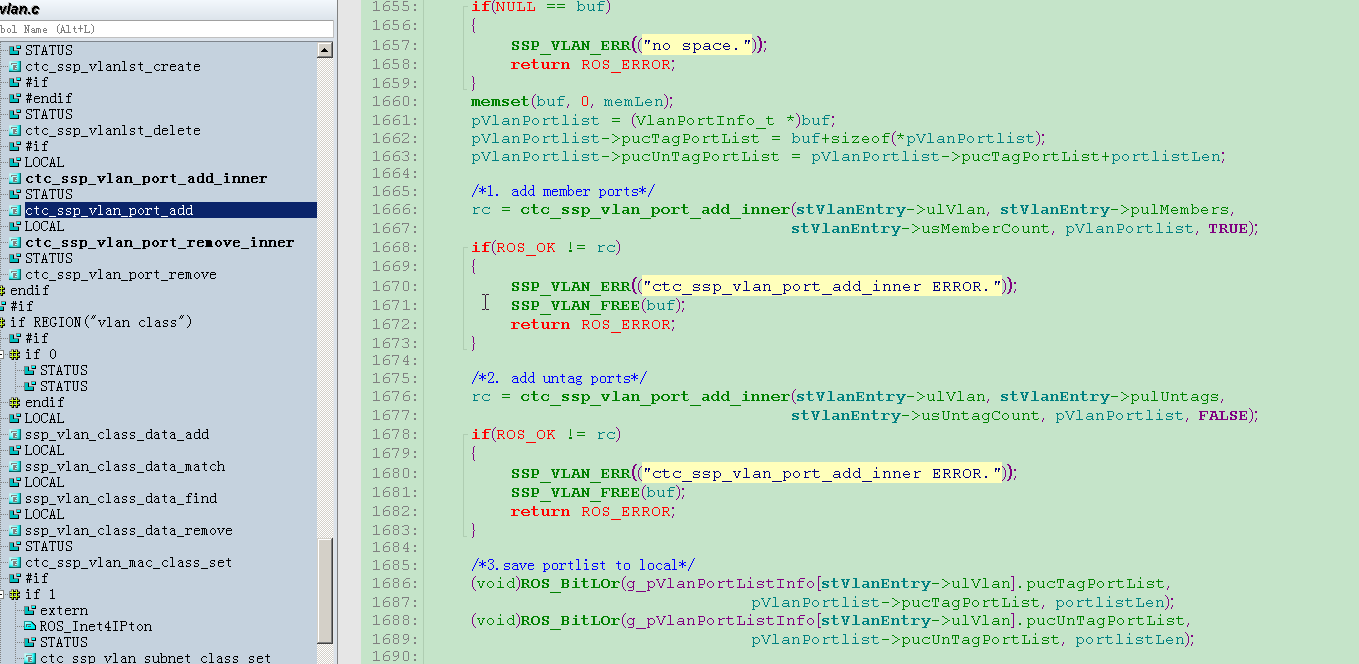
vlan VLANID port X {tagged | untagged}

port X default vlan VLANID

函数实现调用流程：

ctc\_ssp\_vlan\_port\_add() --> ctc\_ssp\_vlan\_port\_add\_inner() --> ctcs\_vlan\_add\_port()，ctcs\_vlan\_set\_tagged\_port()

ctc\_vlan\_port\_qtag\_set() --> \_vlan\_port\_qtag\_set() --> ctc\_tms\_port\_set\_property()，ctcs\_port\_set\_default\_vlan()



4，配置基于MAC地址的VLAN（基于IP子网/协议的VLAN等类似）

配置命令：mac-vlan *mac-address* vlan *vlan-id* [priority *value*]，配置MAC与VLAN的关联

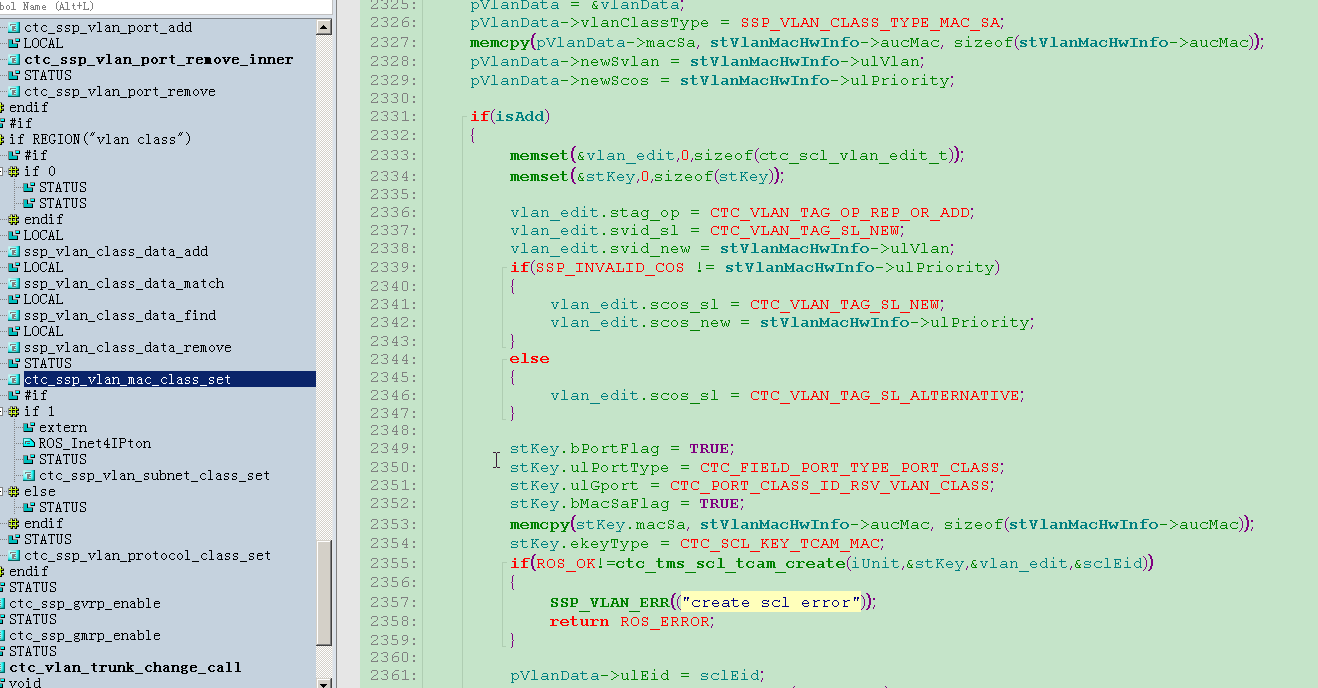
mac-vlan enable（端口模式下），使能MAC-VLAN功能

函数实现调用流程：

ctc\_ssp\_vlan\_mac\_class\_set() --> ctc\_tms\_scl\_tcam\_create() --> ctcs\_scl\_add\_entry()，ctcs\_scl\_add\_key\_field()，ctcs\_scl\_add\_action\_field()，ctcs\_scl\_install\_entry()

ctc\_vlan\_port\_mac\_class\_enable() --> \_vlan\_port\_mac\_class\_enable() –-> ctc\_tms\_scl\_port\_vlan\_class\_enable() --> ctcs\_port\_set\_scl\_property()

主要通过SCL TCAM来实现的



QinQ

一、QinQ基本概念

QinQ（802**.**1Q-in-802**.**1Q），也称Stacked VLAN或Double VLAN。由IEEE 802**.**1ad标准定义，是一项扩展VLAN空间的技术，通过在802**.**1Q标签报文的基础上再增加一层802**.**1Q的Tag来达到扩展VLAN空间的目的。一般应用在骨干网中，通过将用户私网VLAN Tag封装在公网VLAN Tag中，使报文带着两层VLAN Tag穿越运营商的骨干网络（公网），在公网中报文只根据外层VLAN Tag（即公网VLAN Tag）传播，用户的私网VLAN Tag被屏蔽。

1，为什么需要QinQ

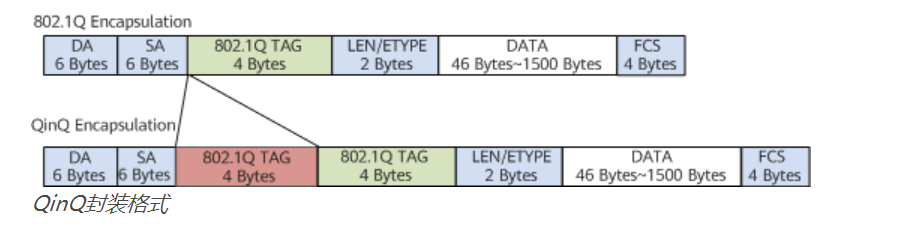
802**.**1Q中定义的Tag域只有12个比特用于表示VLAN ID，所以设备最多可以支持4094个VLAN。而在实际应用中，尤其在城域网中，需要大量的VLAN来隔离用户，4094个VLAN远远不能满足需求，因此诞生了QinQ技术。

随着以太网的进一步发展以及运营商精细化运作的要求，QinQ的双层Tag又有了新的应用场景。它的内外层Tag可以代表不同的信息，如内层Tag代表用户，外层Tag代表业务。另外，QinQ报文带着两层Tag穿越运营商网络，内层Tag透明传送，也是一种简单、实用的VPN技术。

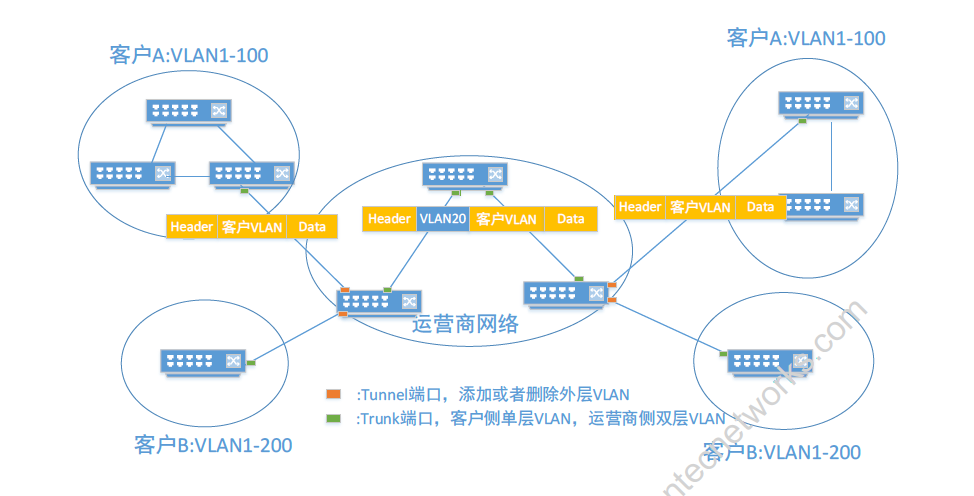
因此，QinQ产生的两大背景是：一是解决日益紧缺的VLAN ID资源问题；二是为小型城域网或企业网提供一种较为简单的二层VPN解决方案。

2，QinQ报文格式及应用场景

QinQ报文有固定的格式，就是在802.1Q的标签之上再打一层802.1Q标签，QinQ报文比正常的802.1Q报文多4个字节。这4个字节用作外层标签，即运营商网络的公网VLAN Tag。原802.1Q的Tag用作内层标签，即私网VLAN Tag。



主要应用场景：QinQ技术使得运行商可以用一个VLAN为含有多个VLAN的用户网络服务。如下图所示，客户A的私网VLAN为1-100，客户B的私网VLAN为1-200。运行商网络中以VLAN 20标识客户A，以VLAN 30标识客户B。当客户A的报文达到运行商网络的边缘交换机时，会被打上一个VLAN 20的外层标签，然后在VLAN 20中转发，离开运行商网络时剥离外层标签。当客户B的报文达到运行商网络的边缘交换机时，会被打上一个VLAN 30的外层标签，然后在VLAN 30中转发，离开运行商网络时剥离外层标签。这样，不同用户网络的报文在公网传输时被完全分开，即使两个用户网络的VLAN范围存在重叠，在公网传输时也不会产生混淆。



3，QinQ类别

QinQ技术分为基本QinQ和灵活QinQ两种。

——基本QinQ

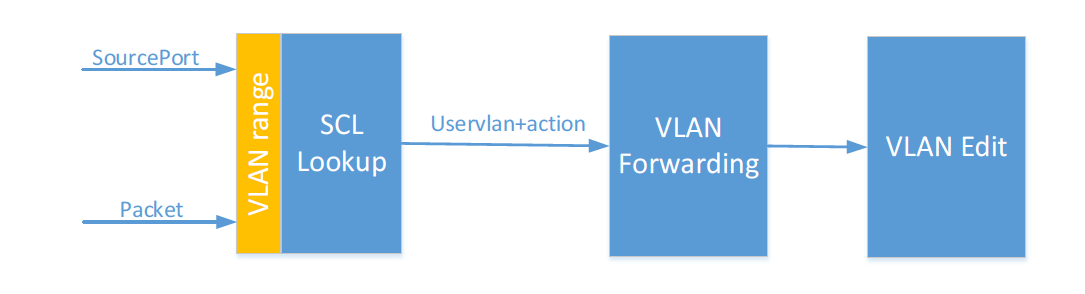
基于端口的QinQ封装。指进入一个端口的所有流量全部封装一个相同的外层VLAN Tag，然后进行转发。这种封装方式不够灵活，用户业务区分不够细致。

——灵活QinQ

基于数据流的QinQ封装。首先对进入端口的数据基于VLAN Tag、MAC地址、IP协议等进行流分类，之后对不同的数据流选择是否封装外层Tag、封装何种外层Tag，以达到区分不同用户或应用的目的。

二、CTC芯片QinQ处理流程

QinQ的芯片处理流程可以简化为：VLAN range处理，SCL查找，VLAN转发，VLAN编辑。

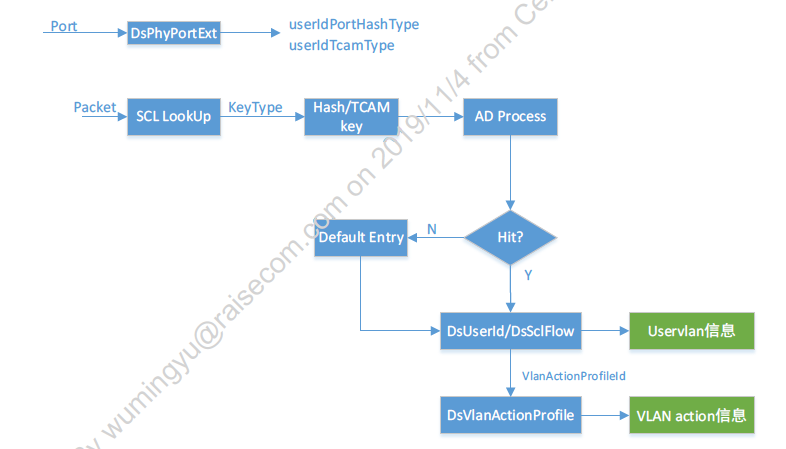


1，VLAN range处理

为了满足上述场景中，将某一段范围内的VLAN映射成运营商网络的VLAN，芯片通过VLAN range来实现该功能。

芯片支持64个VLAN range组，对应DsVlanRangeProfile，每个DsVlanRangeProfile支持配置8个不重叠的VLAN range配置范围，对应字段为vlanMax0~7，vlanMin0~7。

2，IPE SCL查找及处理



IP方向实现QinQ的查找类型可以是hash或者TCAM。若选用hash查找，根据userIdPortHash0Type（userIdPortHash1Type）决定参与查找的key可以是VLAN、port、MAC、IP等；若选用TCAM查找，userIdTcam1Type（userIdTcam2Type）决定参与查找的key可以是L2信息、L3信息或者L2+L3信息。

SCL查找结果既可以索引到DsUserId也可以索引到DsSclFlow，这两种方式都能够实现QinQ的VLAN添加或者删除功能。

如果是添加VLAN操作，添加信息为查表结果中的Uservlan信息，配置如下：

——VLAN Action配置成VTAGACTIONTYPE\_ADD

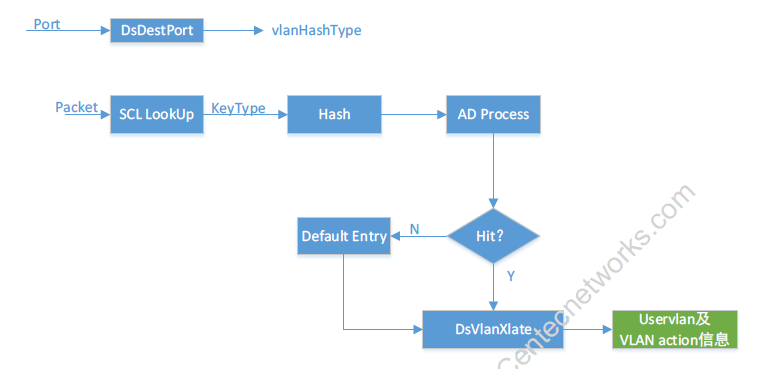
——VlanIdAction配置成VLANIDACTIONTYPE\_USER表示使用userSvlanId/userCvlanId

——CosAction配置成COSACTIONTYPE\_USER表示使用userScos/userCcos

——CfiAction配置成CFIACTIONTYPE\_USER表示使用userScfi/userCcfi

如果是删除VLAN操作，则计算出该VLAN的offset，从该位置删掉VLAN，并调整报文长度。

3，EPE SCL查找及处理



EPE方向通过hash查找方式实现QinQ，两次独立并行。查找的hashType由port属性DsDestPort表中的vlanHash0Type（vlanHash1Type）决定，根据SCL查找结果索引到

DsVlanXlate，实现QinQ的VLAN添加或者删除功能。

如果是是添加VLAN操作，添加信息为查表结果中的Uservlan信息，配置如下：

——VLAN Action配置成VTAGACTIONTYPE\_ADD，

——VlanIdAction配置成VLANIDACTIONTYPE\_USER表示使用userSvlanId/userCvlanId

——CosAction配置成COSACTIONTYPE\_USER表示使用userScos/userCcos

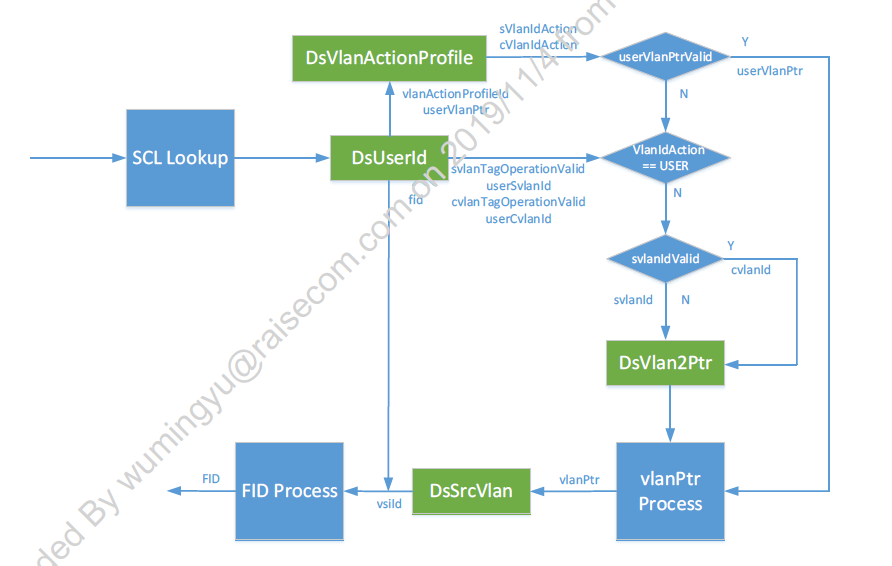
——CfiAction配置成CFIACTIONTYPE\_USER表示使用userScfi/userCcfi

如果是删除VLAN操作，则计算出该VLAN的offset，从该位置删掉VLAN，并调整报文长度。

4，VLAN Forwarding

VLAN Forwarding包含VLAN Switching和VLAN Bridging。

VLAN Bridging转发的Key包含MACDA和FID，其中，MACDA根据报文解析的结果即可得到。FID是芯片转发的概念，可以由VLAN或者metadata来出。FID结合SCL查找的处理流程如下：



首先根据SCL查找结果索引到DsUserId，接下来的处理分为两种：一种是直接通过DsUserId的u2.g2.fid字段出fid；另一种是通过userVlanPtr或者vlanActionProfileId获取FID。

5，VLAN Edit

VLAN编辑操作分为添加VLAN，删除VLAN，替换VLAN。

如果是执行删除VLAN tag操作，只需找到VLAN tag的起始位置，并将其所占的4个字节从报文中删掉，并调整报文长度。完成报文编辑后，从出端口转发出去。

三、QinQ功能配置命令及代码

基于QinQ 的VLAN 转换可以满足以下场景的VLAN 转换

——N:1 VLAN 转换

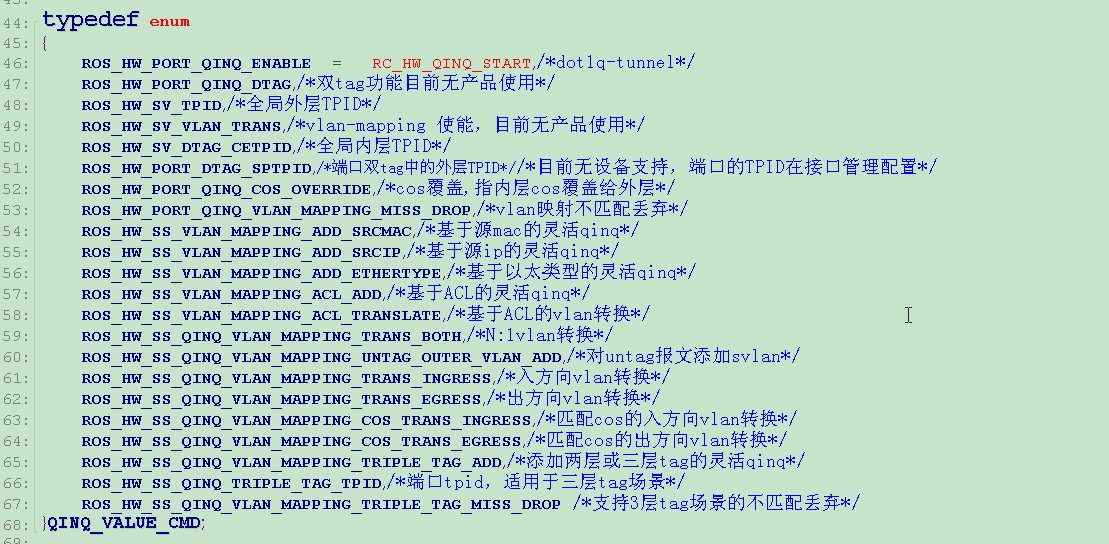
——单层VLAN 转换为双层VLAN

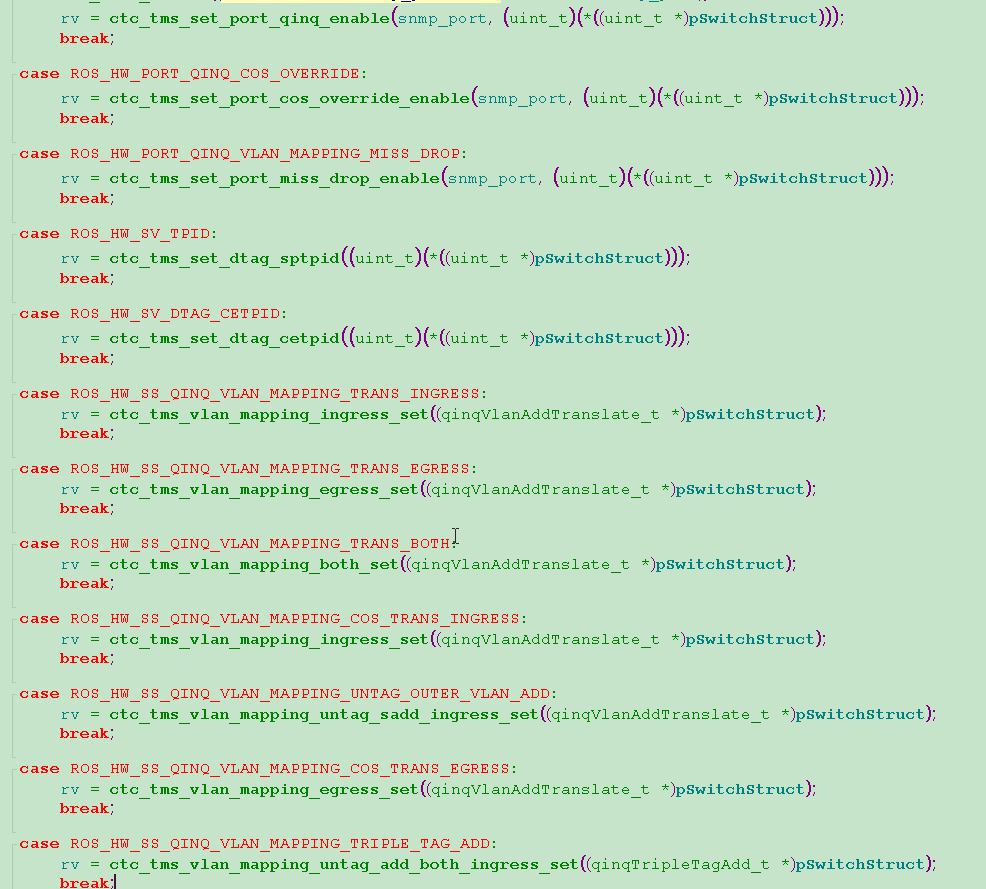
——2:2 VLAN 转换

——双层VLAN 转换为单层VLAN

——Untagged 转换为单层VLAN 或双层VLAN

配置层代码流程：通过ctcQinqHw.c里的ctcQinQSwitchStructSet/Del/Get等函数调用到ctcQinQ.c下处理函数。如下图，以ctcQinQSwitchStructSet为例：





1）基本QinQ

配置命令：mls double-tagging tpid *tpid-value*

mls double-tagging inner-tpid *tpid-value*

配置外层/内层VLAN Tag的TPID值，缺省情况下，值为0x8100。

对应ctccli命令： parser l2 ethernet tpid svlan0 *value*，

parser l2 ethernet tpid cvlan *value*

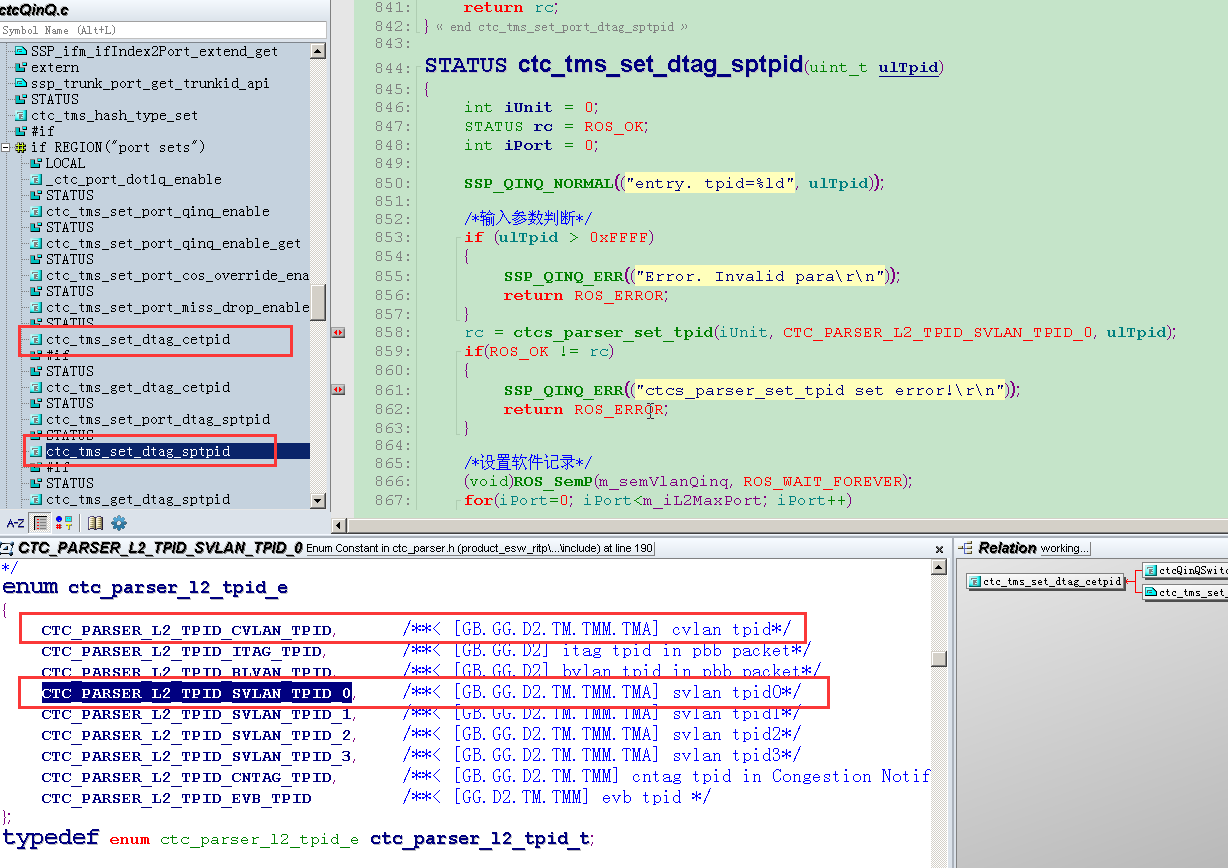
函数实现调用流程：

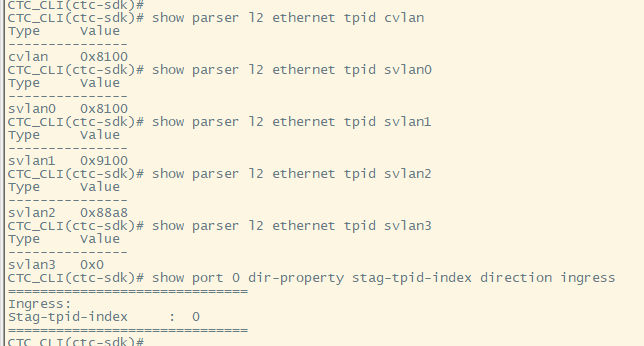
ctc\_tms\_set\_dtag\_sptpid() --> ctcs\_parser\_set\_tpid()

ctc\_tms\_set\_dtag\_cetpid() --> ctcs\_parser\_set\_tpid()

实现方式：

这两个TPID是基于全局生效的，也用于解析VLAN。其中SVLAN有4个值，对应svlan0~svlan3，默认端口选择svlan0对应TPID，所以只设置svlan0的值即可，CVLAN只有1个值。





配置命令：dot1q-tunnel

dot1q-tunnel cos override

switchport vlan-mapping-miss discard

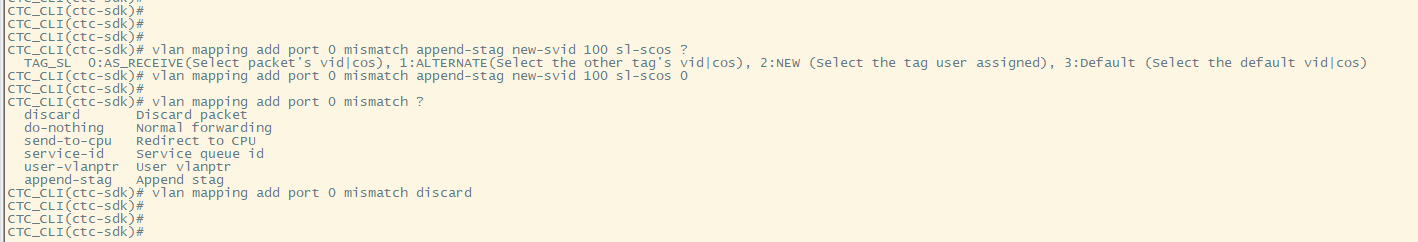
使能接口基本QinQ功能/使能接口CoS优先级覆盖功能/丢弃不匹配灵活QinQ或VLAN转换规则的tagged报文

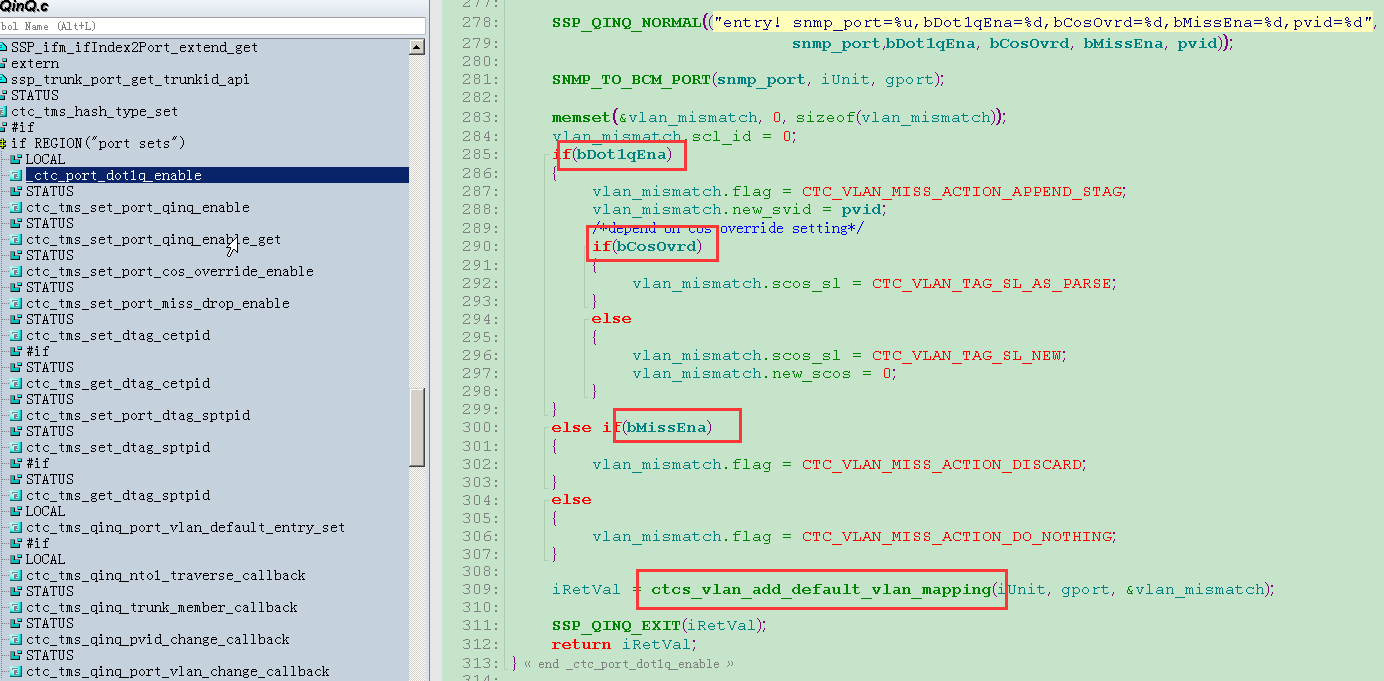
对应ctccli命令：vlan mapping add port X mismatch append-stag/discard

函数实现调用流程：ctc\_tms\_set\_port\_qinq\_enable() / ctc\_tms\_set\_port\_cos\_override\_enable() / ctc\_tms\_set\_port\_miss\_drop\_enable() --> \_ctc\_port\_dot1q\_enable() --> ctcs\_vlan\_add\_default\_vlan\_mapping()

实现方式：

其中dot1q-tunnel和vlan-mapping-miss discard功能主要通过对应动作为append-stag或discard来区分，而cos override通过scos\_sl = CTC\_VLAN\_TAG\_SL\_AS\_PARSE实现，对应含义为select packet’s vid/cos。



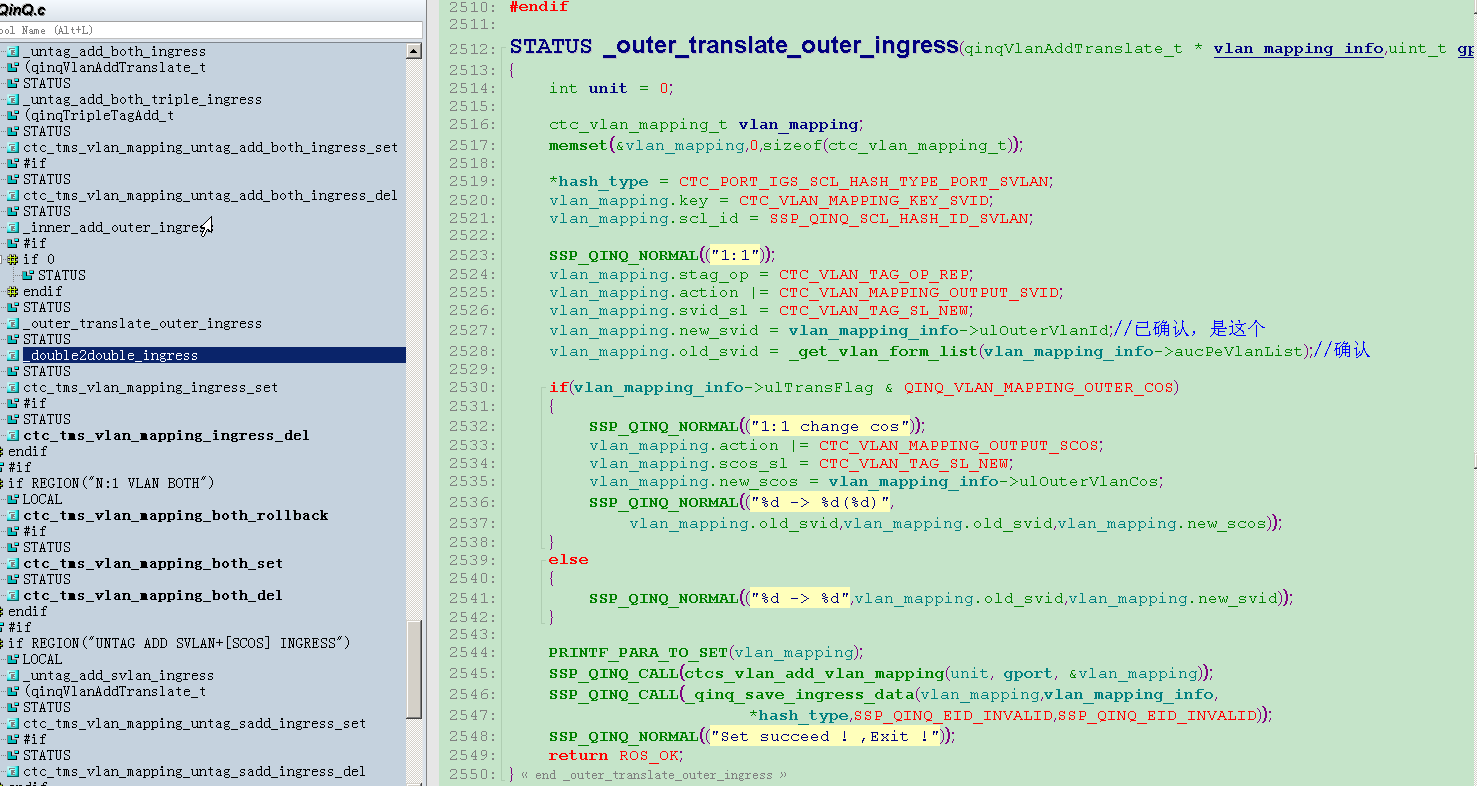


2）灵活QinQ

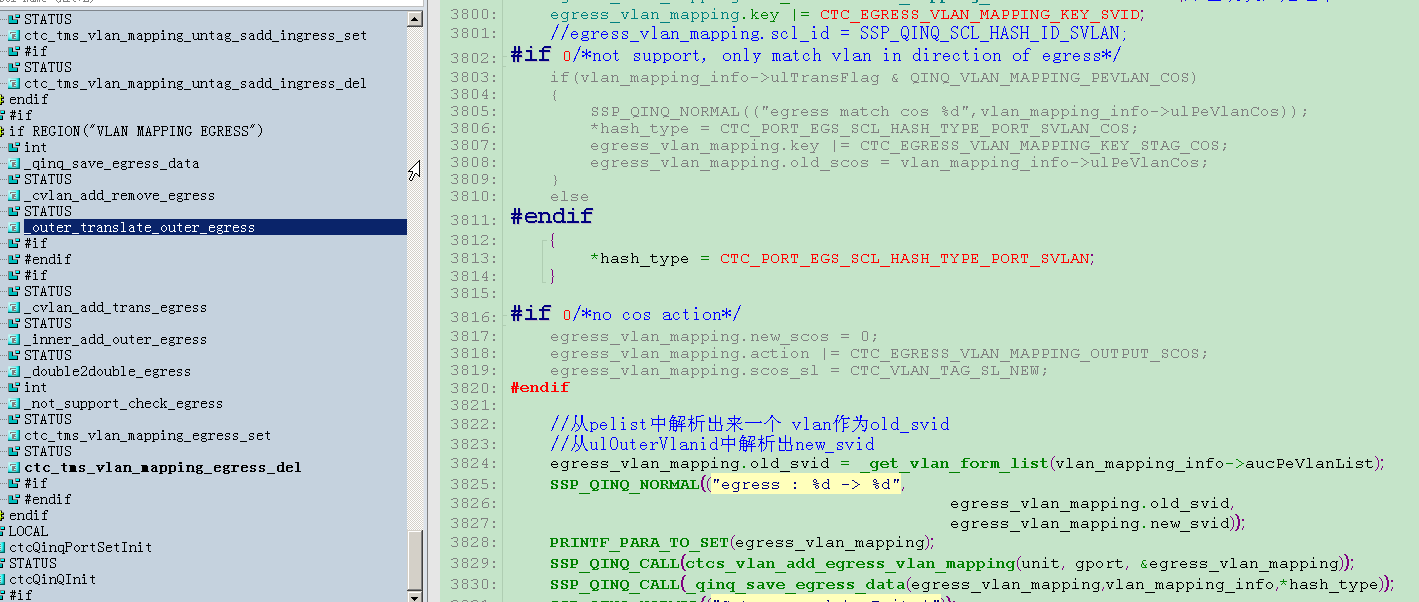
主要包括的功能有N：1 VLAN转换，单层VLAN转换为双层VLAN，2：2 VLAN转换，双层VLAN转换为单层VLAN，untag转换为单层/双层VLAN等。用到的主要模块为使用SCL查找的vlan-mapping。IPE方向支持HASH查找和TCAM查找，对应到底层调用的接口分别为ctcs\_vlan\_add\_vlan\_mapping()和ctcs\_scl\_add\_entry()，ctcs\_scl\_install\_entry()。EPE方向只支持HASH查找，对应底层调用接口为ctcs\_vlan\_add\_egress\_vlan\_mapping()。

举例：

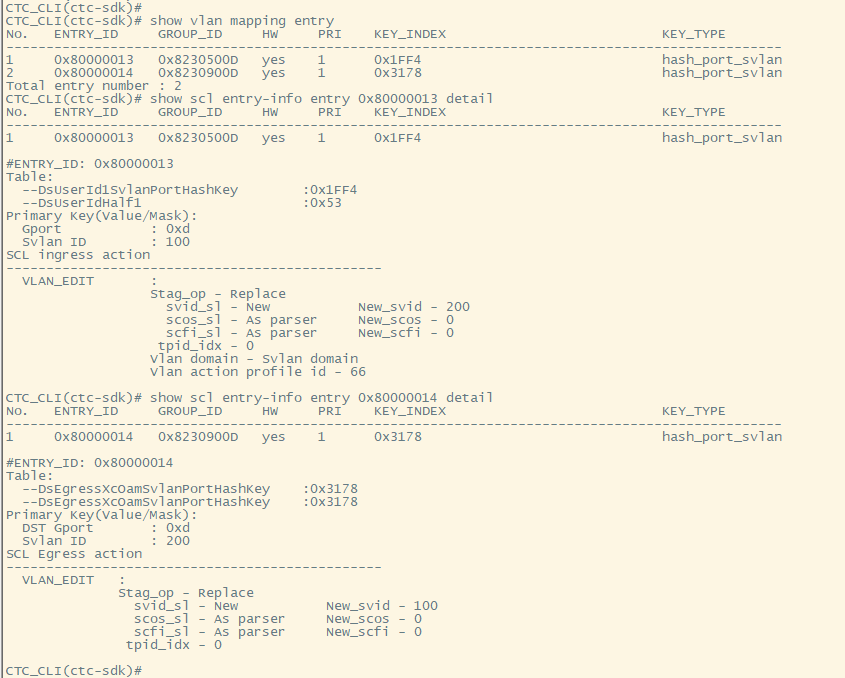
1，1：1VLAN转换，IPE方向函数调用流程为ctc\_tms\_vlan\_mapping\_ingress\_set() --> \_outer\_trans\_outer\_ingress() --> ctcs\_vlan\_add\_vlan\_mapping()，通过传入参数vlan\_mapping的属性stag\_op，action，svid\_sl，old\_svid，new\_svid等来实现vlan编辑。



1：1 VLAN转换，EPE方向函数调用流程为ctc\_tms\_vlan\_mapping\_egress\_set() --> \_outer\_trans\_outer\_egress() --> ctcs\_vlan\_add\_egress\_vlan\_mapping()，通过传入参数vlan\_mapping的属性stag\_op，action，svid\_sl，old\_svid，new\_svid等来实现vlan编辑。



在ctc命令行下可以通过show vlan mapping entry，show scl entry-info entry等查看配置信息。



2，给untag报文加一层vlan，IPE方向函数调用流程为ctc\_tms\_vlan\_mapping\_untag\_sadd\_ingress\_set() --> \_untag\_add\_svlan\_ingress() --> ctc\_tms\_scl\_tcam\_create()，通过传入参数vlan\_edit的属性stag\_op，svid\_sl，svid\_new等来实现vlan编辑。

