**debug调试命令**

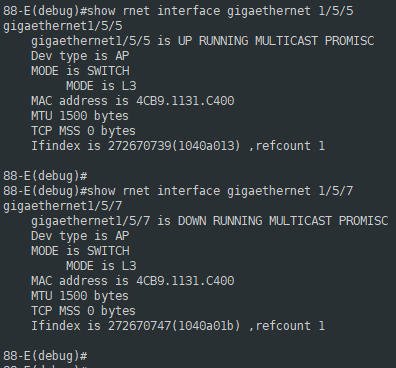
在debug视图下使用的命令，一般是软件用来调试定位使用的。

**接口**

**查看接口的ifindex**

**show rnet interface *interface-type interface-name***

命令在debug视图下使用：



**查看接口的snmp\_port**

**show portmap all**

命令用来查看面板口对应的snmp\_port号，以及对应使用的芯片端口号

**查看接口硬件link状态**

**show link\_status snmpport <1-104>**

**ERPS**

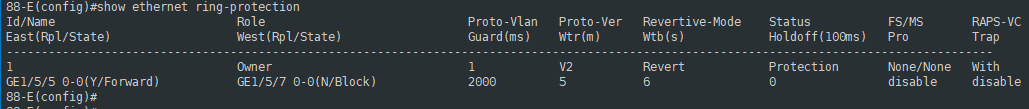
**查看驱动erps配置**

**debug ssp\_erps show**

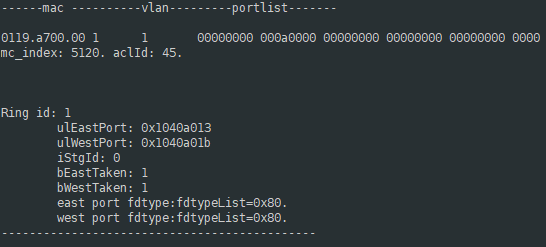
配置一个erps实例：

ethernet ring-protection 1 east gigaethernet 1/5/5 west gigaethernet 1/5/7 node-type rpl-owner rpl east block-vlanlist 1-10

设备show命令查看环配置：



定位问题时可以通过debug ssp\_erps show命令看驱动保存的这个实例配置，与上面的show命令对照，确认驱动的配置是否正确：



从打印信息可以看到协议的组播mac地址最后一个字节对应命令行指定的环id，使用的协议vlan是默认的vlan 1，对应的portlist是按bit位保存相关的snmp\_port值，上面使用到的两个端口snmp\_port分别是45和47，正好对应portlist显示的值。

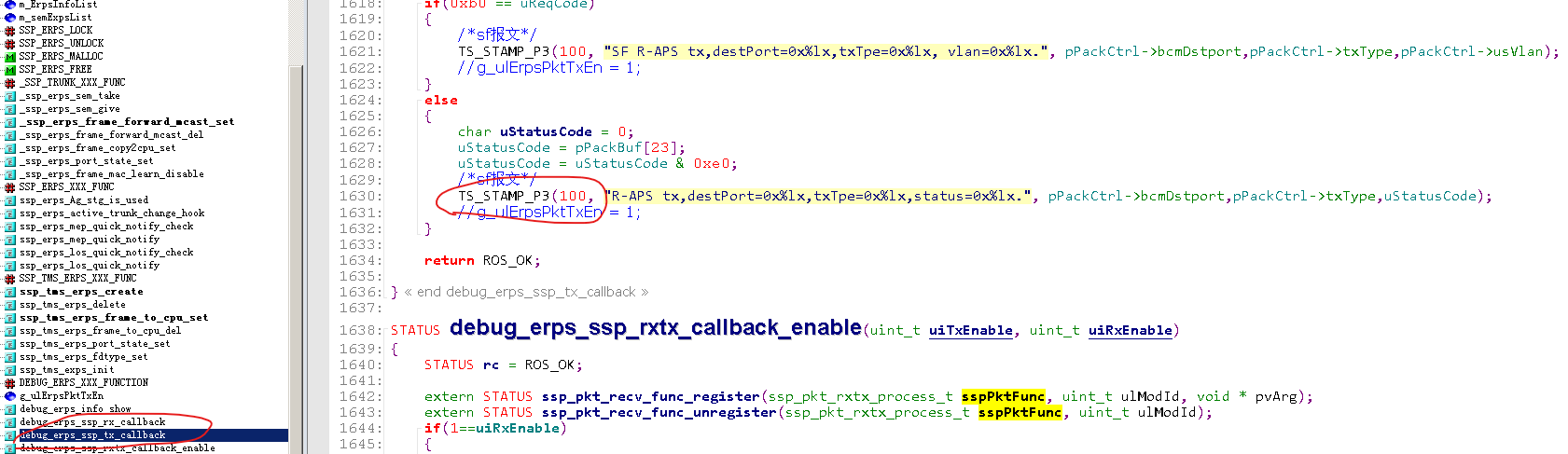
显示的十六进制端口号可以通过命令show rnet interface gigaethernet 1/5/5查看。

**协议报文收发打时间戳**

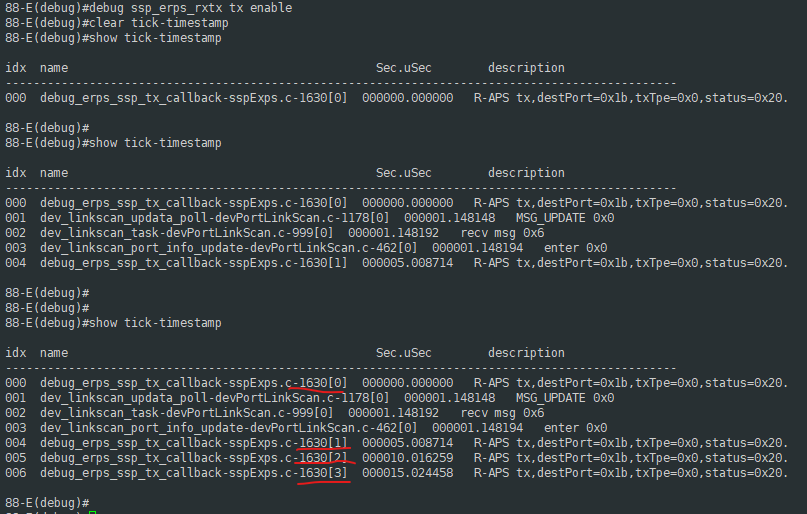
**debug ssp\_erps\_rxtx (rx | tx | both) (enable | disable)**

注意：要先打开时间戳记录功能——start tick-timestamp

设置后在协议包收发处理时会打上处理该报文的时间戳，方便查看协议报文的收发是否正常。比如使能发包的时间戳记录：



查看时间戳记录，会看到在函数里添加的时间戳信息，协议规定的发包间隔5秒也都正确：



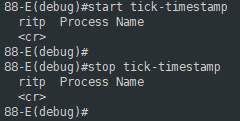
**时间戳**

**打开/关闭时间戳记录功能**

**start tick-timestamp**

**stop tick-timestamp**

如果需要查看代码里面添加的时间戳信息，需要先打开这个开关。



**查看时间戳信息**

**show tick-timestamp [ritp]**

不带参数ritp时是查看平台和驱动所有的时间戳信息，ritp表示只查看驱动添加的时间戳信息。

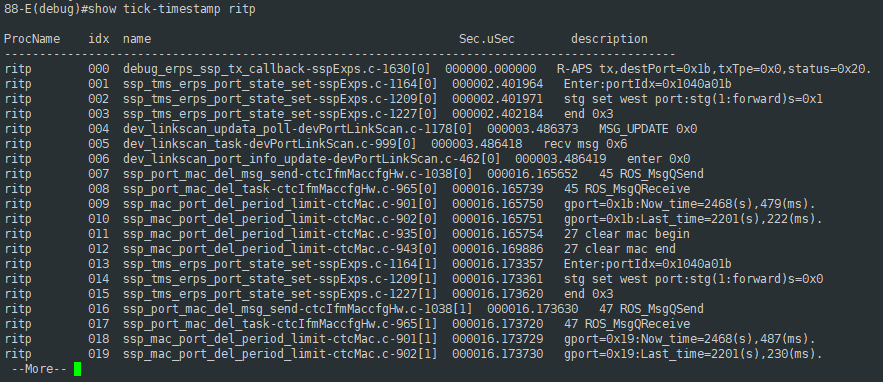
代码里面需要看哪块逻辑耗时情况时，在代码里加上时间戳信息：



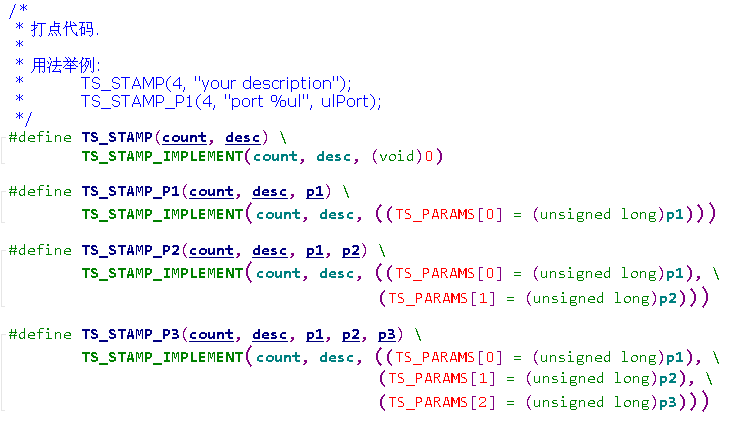
上图是erps设置端口状态加的时间戳信息，配置erps实例或者保护倒换时记录时间信息，先使用命令：

start tick-timestamp

打开时间戳记录开关，然后再使用show命令查看这两个打印：



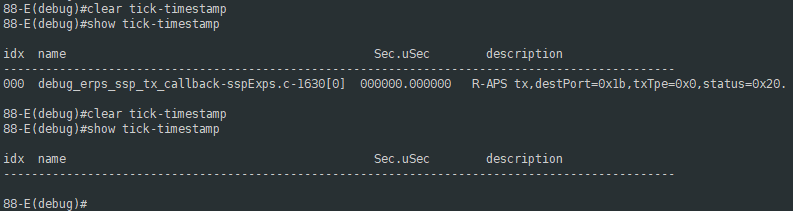
**注：**代码提供多个时间戳打点接口，上图中使用的是其中一种，其他接口如下，需要记录不同内容时选择相应的接口：



**清除时间戳信息**

**clear tick-timestamp [ritp]**

清除所有时间戳记录，或者只清除ritp记录的时间戳信息。一般在show命令显示太多，不方便查看需要的时间戳记录时先清除，等下一个时间戳记录。



**ssp层时间戳命令**

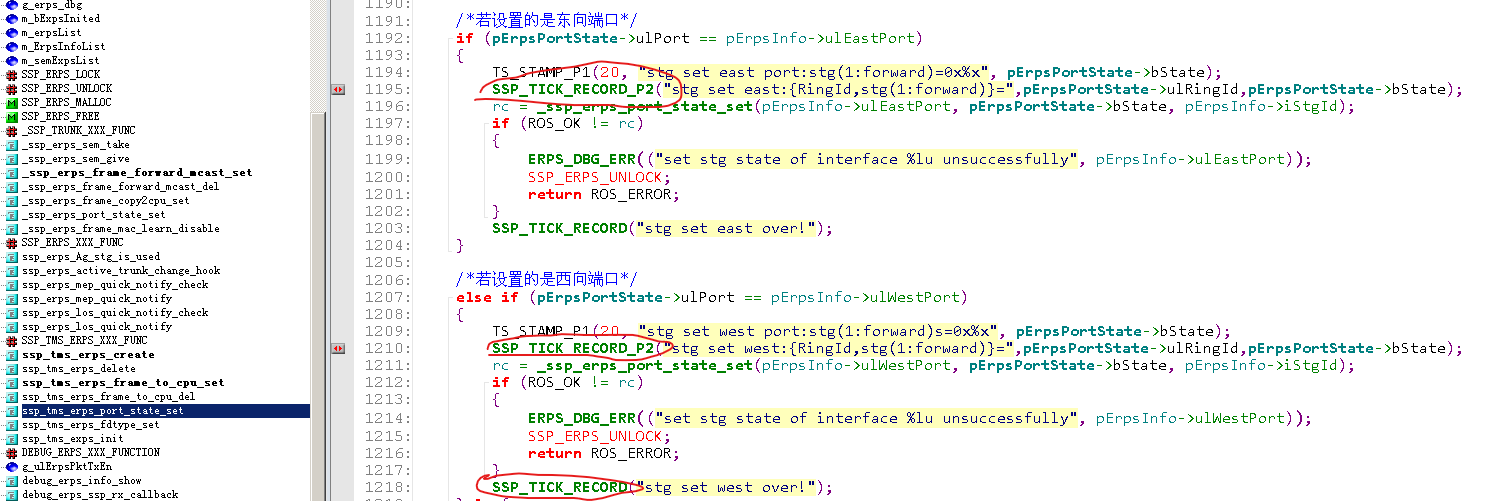
上面介绍的命令是平台提供的时间戳接口，ssp层也有相应的时间戳开关、记录和查看命令。

**debug ssp\_time\_tick (clear | begin | end | show [latest <1-1000>])**

同样的，要想给代码运行过程打上时间戳，需要先打开记录开关：

**debug ssp\_time\_tick begin**

比如，同样在erps设置端口状态加时间戳信息，配置erps实例或者保护倒换时记录时间信息：



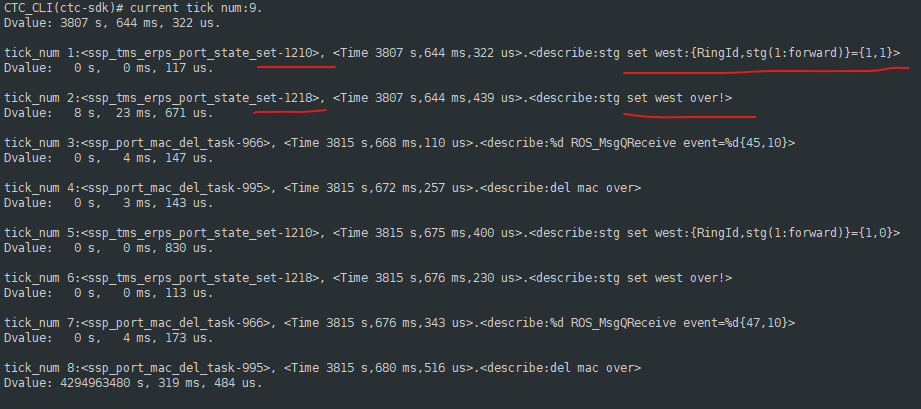
然后配置一个新的环实例

ethernet ring-protection 1 east gigaethernet 1/5/5 west gigaethernet 1/5/7 node-type rpl-owner rpl east block-vlanlist 1-10

再使用命令

**debug ssp\_time\_tick show**

查看添加的时间戳信息：

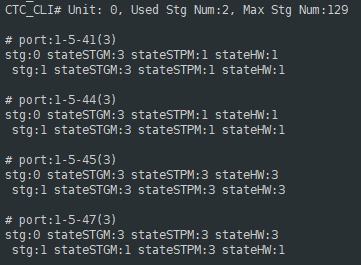


相应的函数名，所在行数，以及记录的参数内容都能看到。

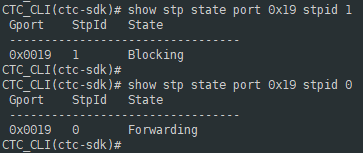
**stg相关**

**查看端口在stg中的状态**

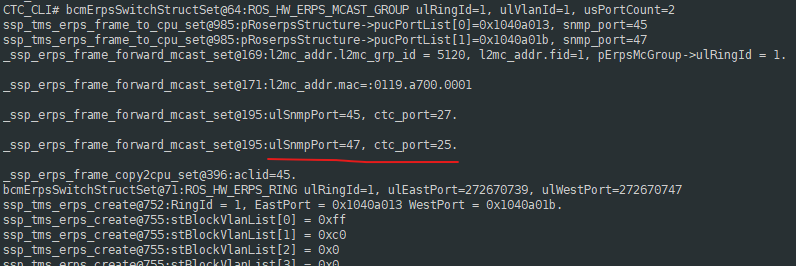
**debug ssp stg port status show**



显示的端口格式是unit-slot-snmp\_port，可以看到snmp\_port47在stg1中的状态是1，也就是block的，用sdk命令确认：



设备接口和sdk使用的接口号怎么对应的暂时没找到命令行，这里是通过配置环实例时的debug打印（debug ssp erps enable）看的：



**逻辑stgid与硬件stgid的使用关系**

**debug ssp stg mod info show**

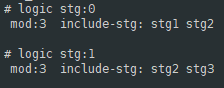


示例中逻辑stg是0，实际分配使用的硬件stg是1，最前面的mod表示这个stg的使用模块，3表示是erps模块在使用这个stg，如果是8，就是mstp模块。

再配置一个环实例：

ethernet ring-protection 2 east gigaethernet 1/5/6 west gigaethernet 1/5/8 node-type rpl-owner rpl east block-vlanlist 5-15

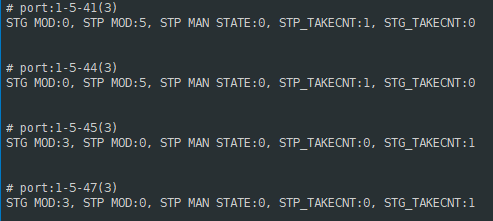
然后查看stgid的对应关系：



因为vlanlist有重叠，所以拆分了stgid。

**端口的stg使用情况**

**debug ssp stg mod port info show**



mod表示端口正在被这个模块使用，3表示erps模块，而且是以stg方式使用的。最后一列表示该端口被使用了一次。

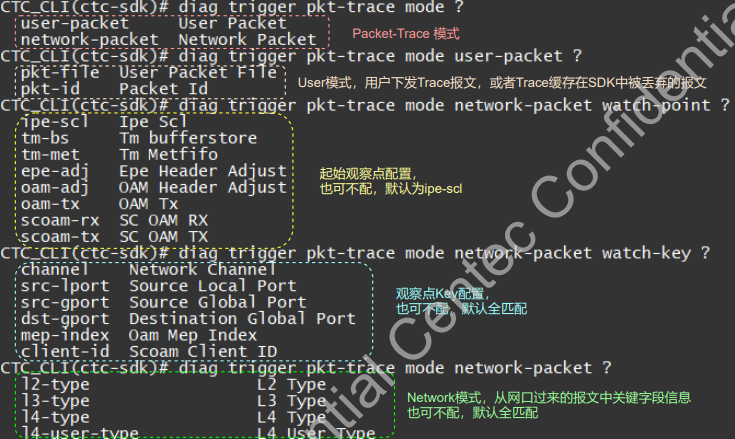
**sdk相关命令**

**调试命令**

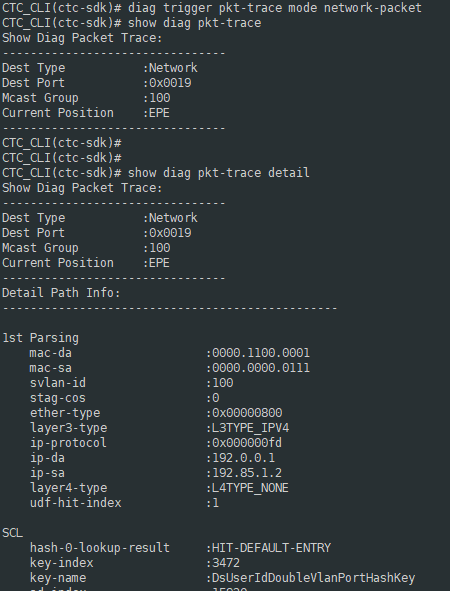
**查看packet trace信息**

**diag trigger pkt-trace mode network-packet**

**show diag pkt-trace {detail}**



network-packet模式后面可以不带参数，默认就是trace从收包到发包的整个流程，然后show命令查看trace信息：

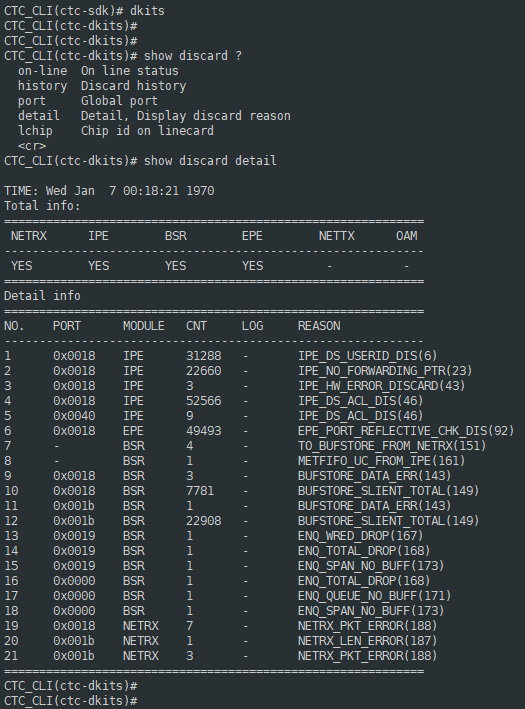


diag命令配置后只能查询一次转发处理的trace信息，不是实时的处理流程，如果不停流，那么要查看最近一次的trace信息，需要重新配置diag trigger命令，然后再show命令查看，如果是多个接口收包，那只能反复配置diag，直到找到对应接口的收包trace信息。

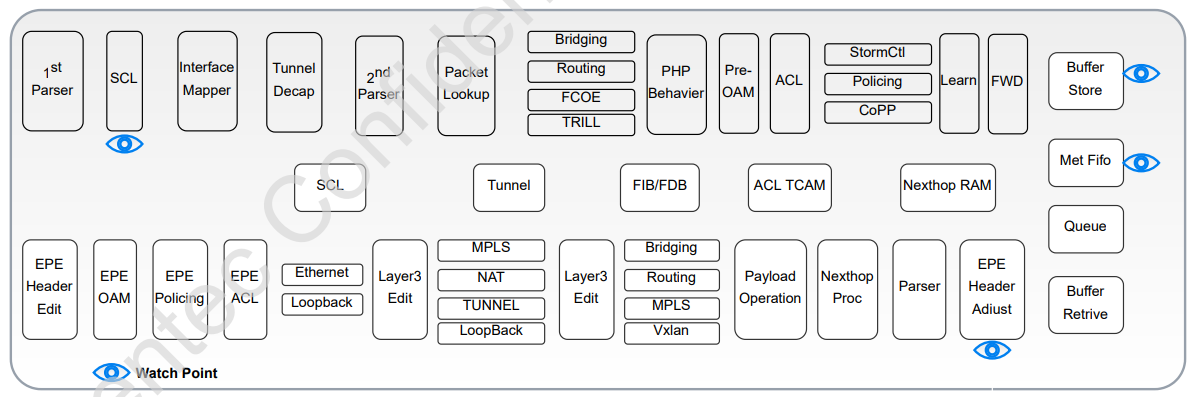
**dkits 查看丢包原因**

**show discard detail**

先敲命令“dkits”进入dkits节点，然后查看丢包的详细信息：



可以看到端口下丢包计数，和丢包原因，是在IPE入方向，还是EPE出方向丢包等。像BSP模块丢包是在IPE和EPE之间的处理出现问题导致的丢包：



一般是在Met Fifo或者Queue队列这两个地方处理出现异常导致的丢包。

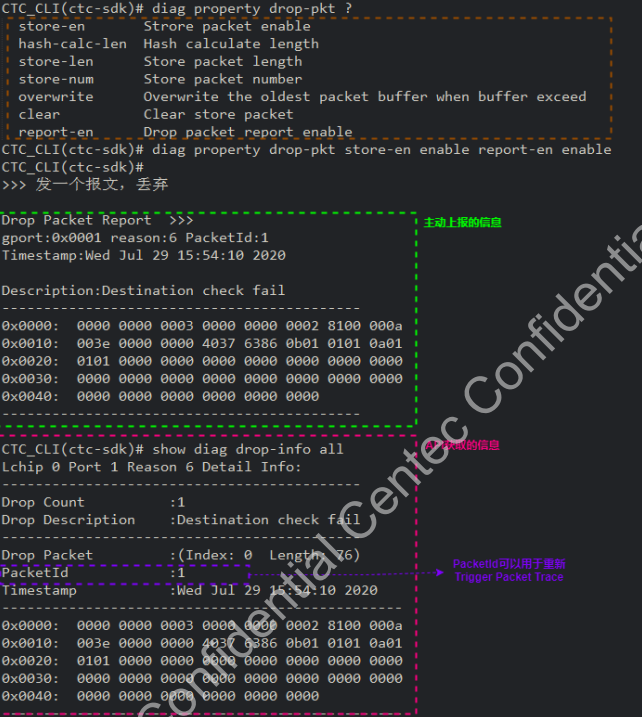
另外，第四列的丢包计数不一定准确，有可能丢包很多，但是查询显示只有一个计数。

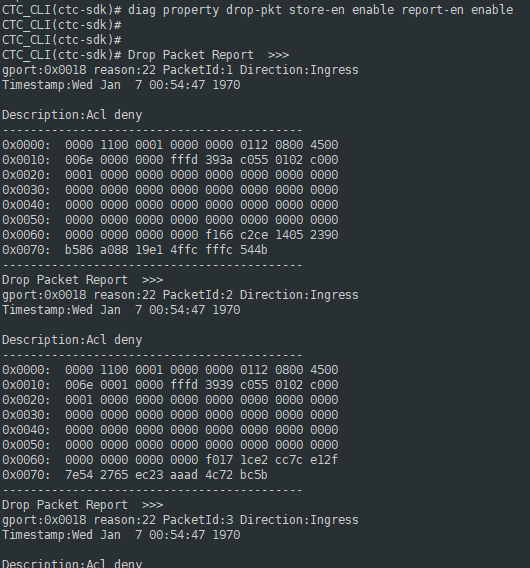
连续敲该show命令，回显的是新出现的丢包原因，比如上一次是端口0x1的丢包原因和计数显示，再敲一次时可能只显示端口0x2的丢包原因和计数，这种情况一般在多个口打流且有丢包时出现，如果只有一个地方丢包，那多次使用show命令，回显的内容应该是差不多的。

**丢包主动上报和查看**

**diag property drop-pkt store-en enable report-en enable**

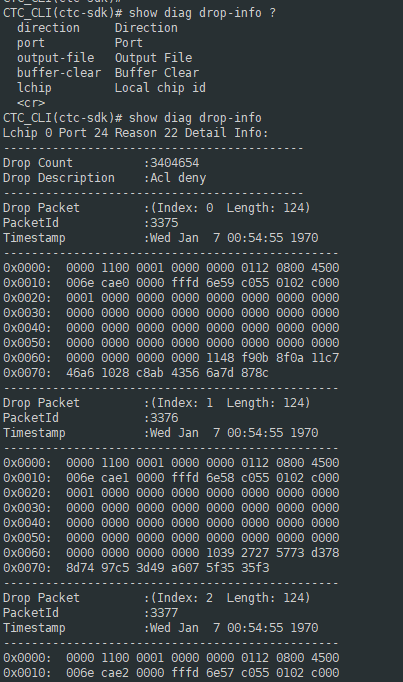
**show diag drop-info**



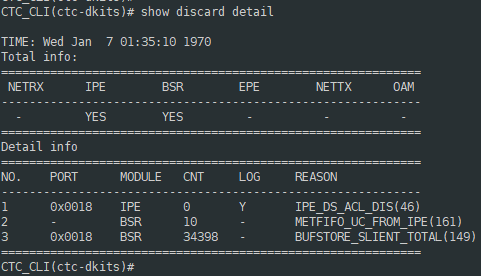


配置diag使能丢包主动上报后，当有丢包产生时会自动打印上图中的信息。比如上图是配置了acl，入口未匹配丢包，使能主动上报后，会自动打印丢包信息，包括丢包原因（Acl deny），报文内容，丢包方向（Ingress）等。

也可以使用show命令查看丢包信息：

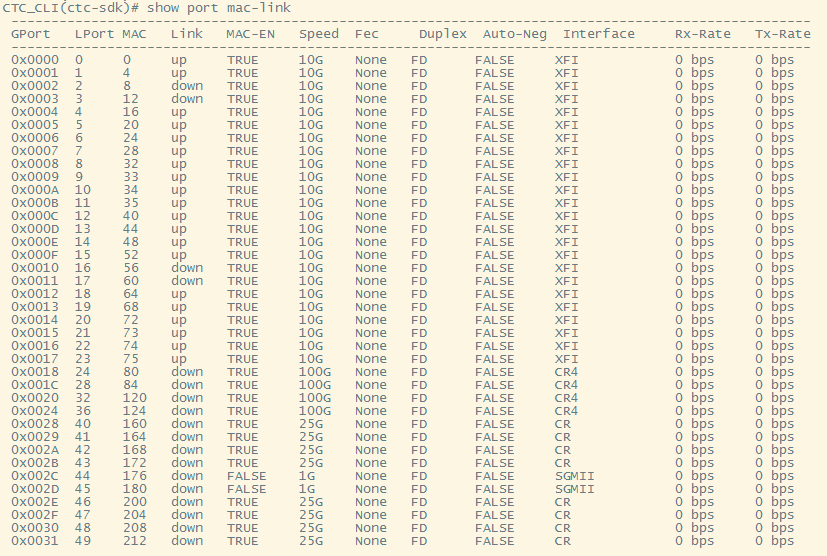


dkits命令查看丢包信息：

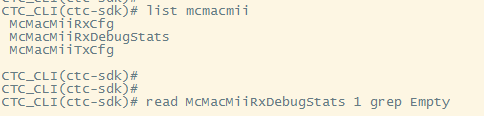


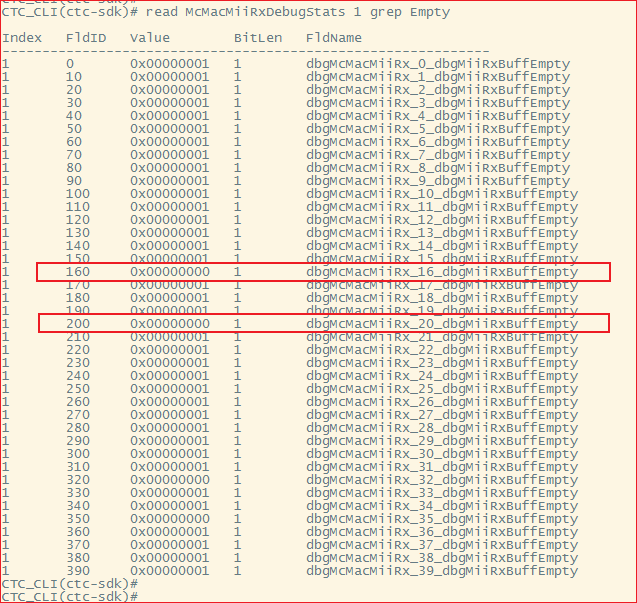
fcs错包问题

show port mac-link



找到接口GPORT对应的MAC号，上图第三列，比如端口0x10或者0x11的MAC分别是56和60，然后读寄存器：





这个寄存器有0-7八个索引，每个索引有40个MAC表项，所以MAC为60时，需要读索引1的第20个表项（MAC是从0开始索引，这个也是从0开始索引）。上图中的两个红框分别对应MAC号56和60的表项。

停止打流之后如果这个值还是为0，那么说明接口收包的fcs错包是这个原因导致的。

**接口**

**查看接口自协商、速率、双工模式和link状态**

**show port mac-link**

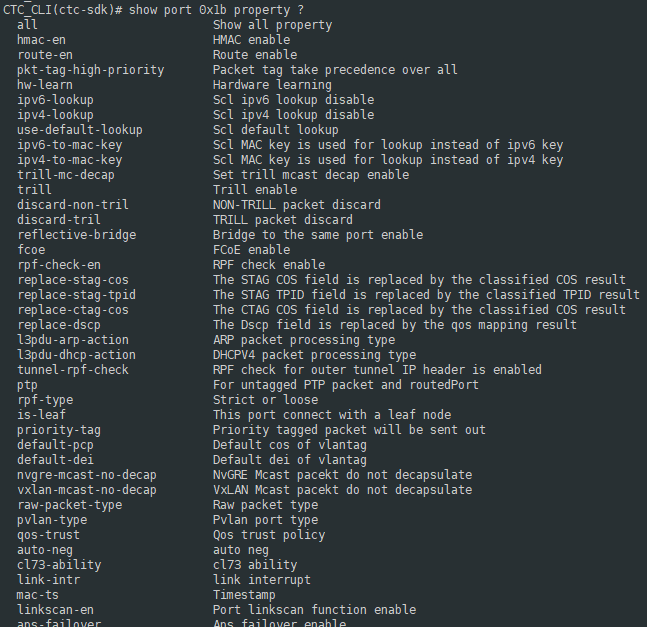
RG8板卡用这个命令看的接口都是link状态，板卡的硬件设计使得接口的硬件状态始终都是link的，所以RG8板卡的接口shutdown后用这个命令看的仍然是linkup，在debug节点下show link\_status snmpport <1-104>。

另外，需要确定接口是接了phy，还是直接用的mac，如果是mac，那么phy相关的属性就没有。

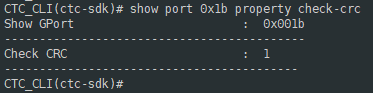
**查看接口属性**

**show port GPORT\_ID property {……}**

这个命令可以查看接口的各种属性，如下：



比如是否进行crc校验：

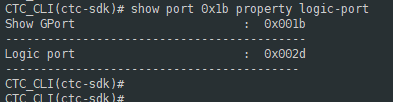


**查看接口逻辑端口号**

**show port GPHYPORT\_ID property logic-port**

一般在设备的debug-hide下可以用show portmap {all}来查看面板口、逻辑口和芯片端口号的映射关系。

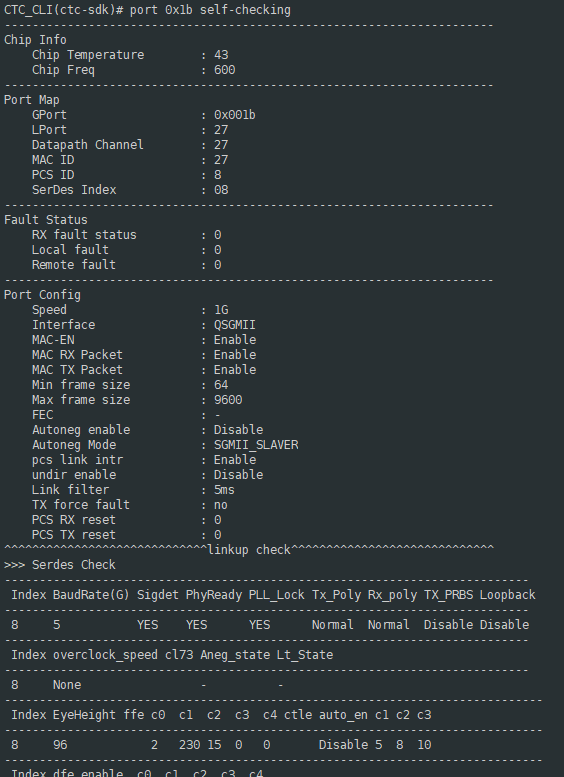
本条命令需要在知道芯片端口号的前提下查看逻辑端口号。



**接口自查属性**

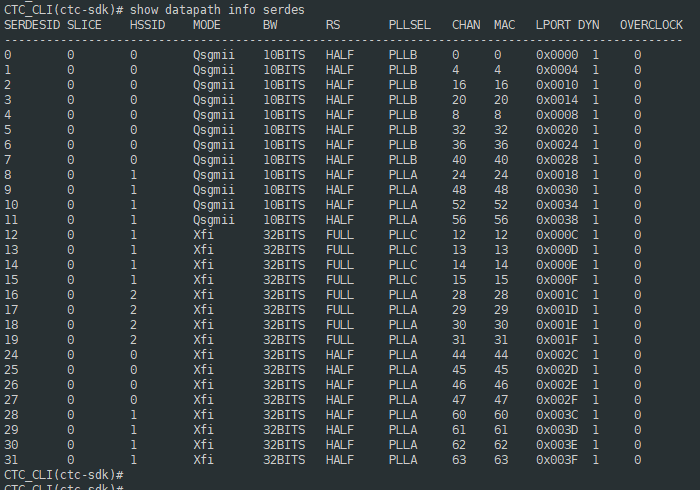
**port GPHYPORT\_ID self-checking**

命令可以查看接口的一些基础属性，包括逻辑端口号、工作模式、协商模式等。



**查看接口对应serdes**

**show datapath info serdes**



根据右数第三列的逻辑端口号查看第一列该端口对应的serdes id。

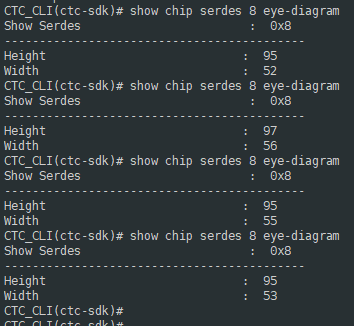
**查看serdes眼图**

**show chip serdes SERDES eye-diagram**

通过命令

**show datapath info serdes**

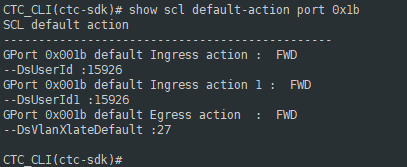
找到接口对应的serdes id后使用本命令查看serdes的眼图：



**scl**

**查看接口默认动作**

**show scl default-action port GPORT\_ID**

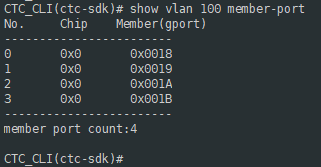


当scl的hash和tcam都没查找到时，会根据这个接口默认的表项（即scl hash default entry）进行处理，上图中显示的ingress和egress方向默认action都是转发。

**vlan**

**查看vlan成员端口**

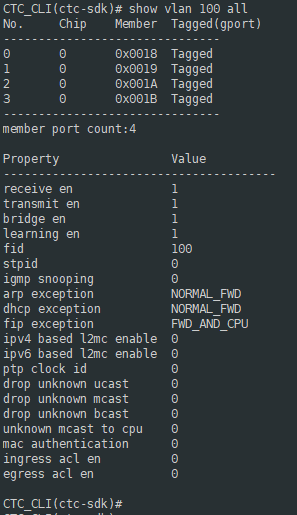
**show vlan VLAN\_ID member-port**



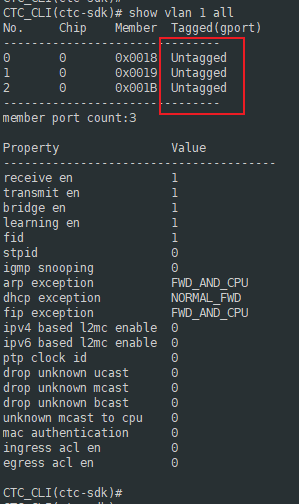
命令只能查看有哪些端口在vlan中，不能显示是以tag还是untag方式加入的vlan。

**查看vlan所有信息（tag或untag类型等）**

**show vlan VLAN\_ID all**

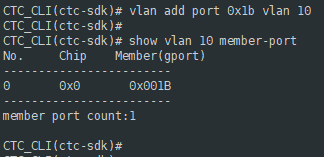


这条命令可以查看是untag还是tag类型，还包括vlan所在的stp实例，mac学习是否使能等信息。



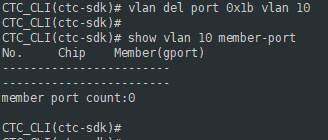
**端口加入vlan**

**vlan add port GPHYPORT\_ID vlan VLAN\_ID**



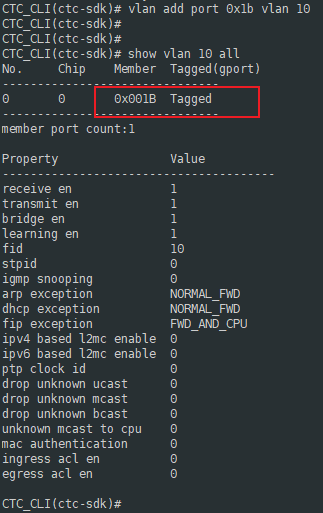
**端口删除vlan**

**vlan del port GPHYPORT\_ID vlan VLAN\_ID**

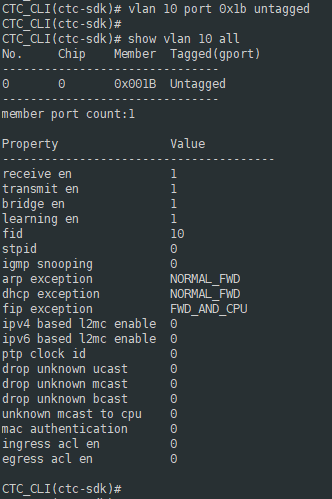


**修改vlan加入端口的类型（tag/untag）**

**vlan VLAN\_ID port GPHYPORT\_ID (tagged | untagged)**

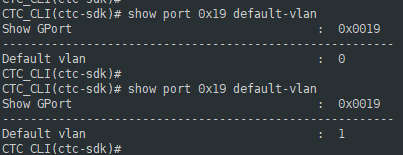


接口加入vlan时默认是tag方式加入，使用本命令可以修改为untagged方式：



**查看端口默认vlan**

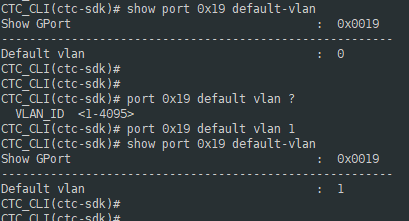
**show port GPHYPORT\_ID default-vlan**



接口配置access vlan或者trunk native vlan时，这个命令显示的vlan就是配置值。

**设置接口默认vlan**

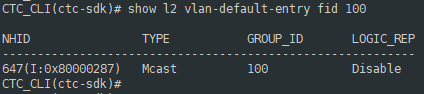
**port GPHYPORT\_ID default vlan VLAN\_ID**



**nexthop**

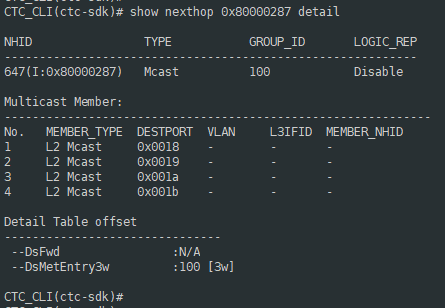
**查看vlan对应的nexthop id**

**show l2 vlan-default-entry fid FID**



**查看nexthop详细信息（成员端口等）**

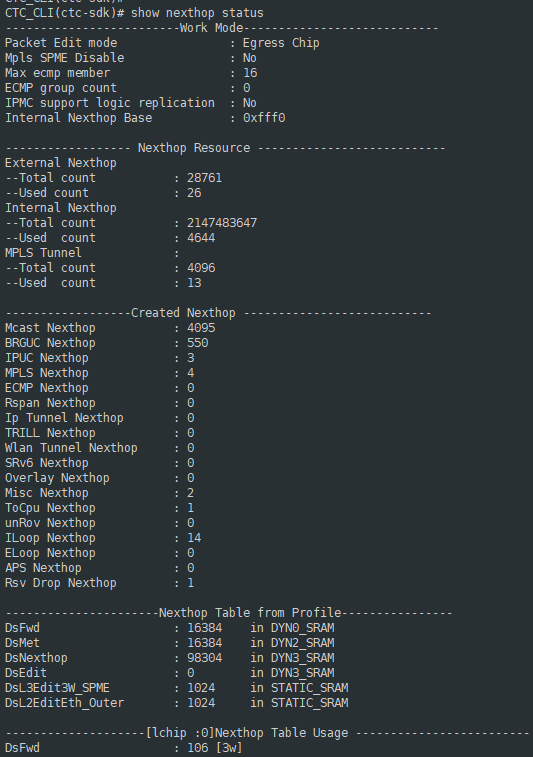
**show nexthop NHID detail**

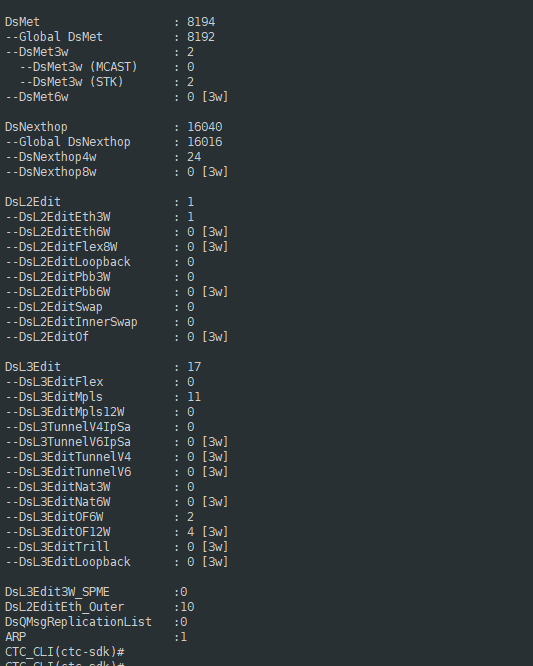


一般用来查看vlan对应的组播成员，端口加入vlan时默认会将端口加到对应的组播组，有时候业务不通，查看端口确实是在vlan里，但是用本命令查看组播成员时，端口不一定在这个组播组。

**查看nexthop各表项使用情况**

**show nexthop status**



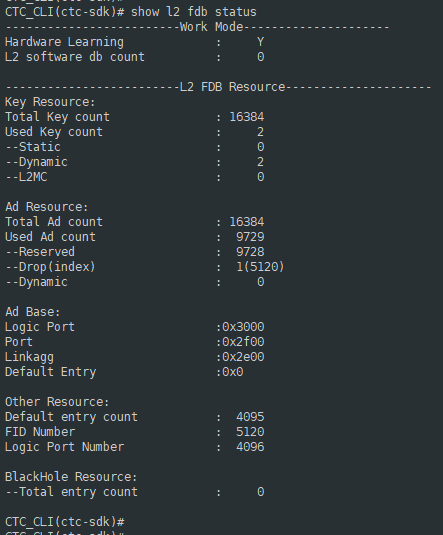


**mac**

**查看mac表工作模式以及资源**

**show l2 fdb status**

查看mac表工作模式以及资源使用情况。

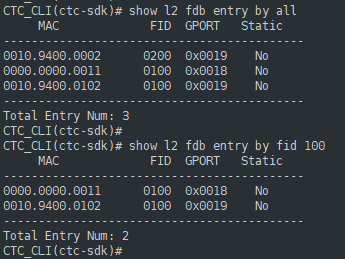


上图显示mac表学习模式是硬件学习，总规格为16k，已使用两个动态mac表项。在其他资源项中，Default entry为4095个，默认是给vlan下的mac学习使用，而fid number总共是有5120个，多出来的1024个资源表项是给vsi使用。

**查看指定类型的mac表项**

**show l2 fdb entry by TYPE**

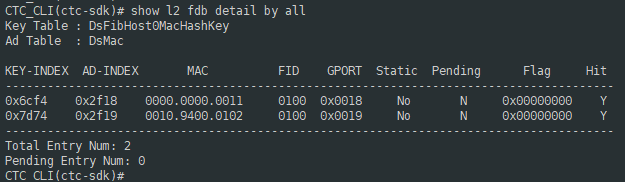
查看指定类型的mac表项，比如mac、mac+vlan、port、vlan等。



**查看所有mac表项详细信息**

**show l2 fdb detail by all**

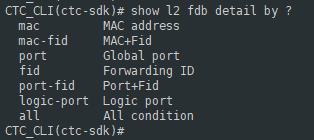
查看所有mac表项的详细信息。



**查看指定类型mac表项详细信息**

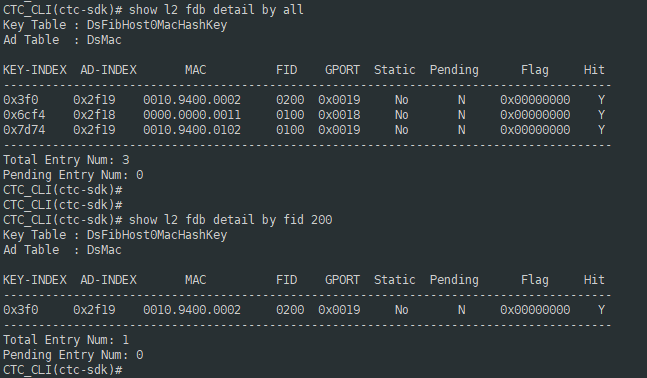
**show l2 fdb detail by TYPE**

查看指定类型的mac表项详细信息。



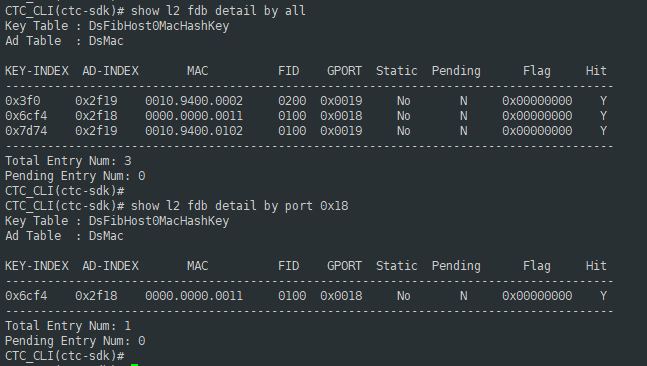
比如查看指定vlan 100下的mac表项：

show l2 fdb detail by fid 100



查看指定接口下的mac表项：

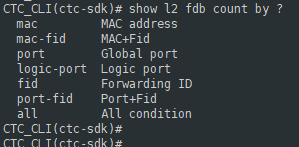
show l2 fdb detail by port 0x18



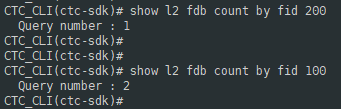
**查看指定类型下的mac计数**

**show l2 fdb count by TYPE**

查看指定类型下的mac表项计数，比如指定vlan或端口下学习的mac个数。



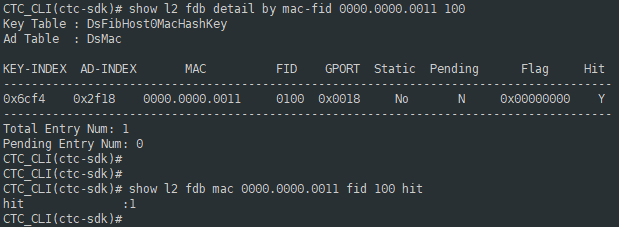
指定vlan 200下学习到的mac个数：



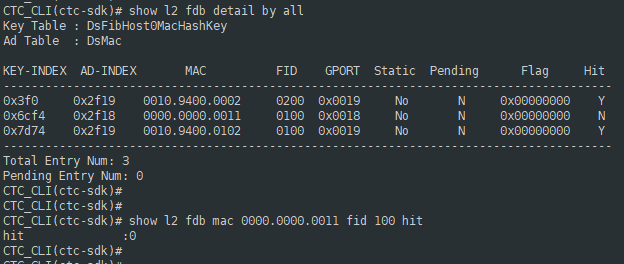
**查看指定的mac表项是否hit**

**show l2 fdb mac MAC fid FID hit**

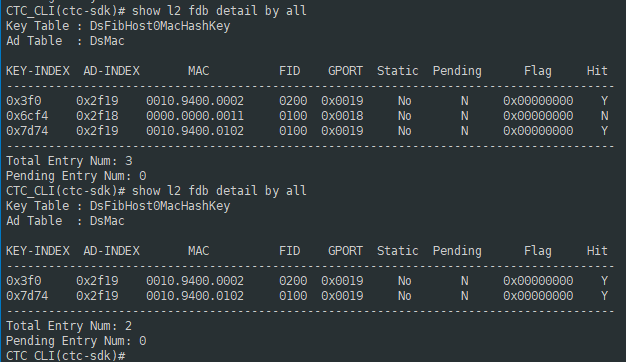
查看指定的mac表项是否hit。一般用来查看该表项对应的报文从接口进入设备后的查表情况，如果有该表项对应的报文接收，hit位置位：



如果老化周期内接口没有收到该表项的报文，hit位置0：



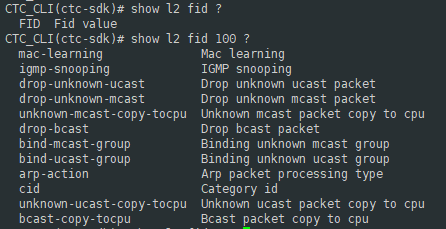
当hit位置0后，如果一个老化周期内仍然没有该表项的报文接收，将自动删除该表项：



**查看vlan的mac学习、未知单播丢弃等**

**show l2 fid FID TYPE**

查看二层mac各个功能的配置情况，比如mac学习是否使能，未知单播或组播如何处理等。

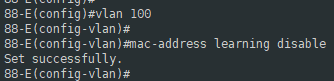


查看vlan 100的mac学习是否使能：

show l2 fid 100 mac-learning



设备命令行关掉vlan 100的mac学习功能：



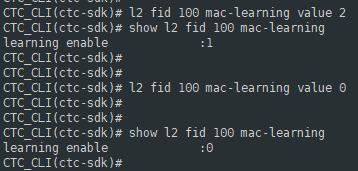
使用sdk命令再次查看：



**vlan的mac学习使能**

**l2 fid FID mac-learning value VALUE**

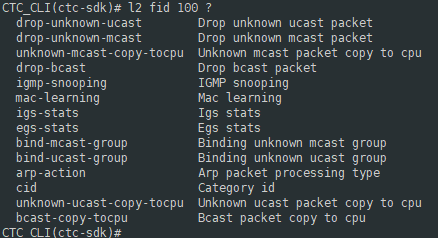
设置指定vlan的mac学习功能使能和关闭。VALUE值为0时mac学习功能关闭，设置为非0值时使能该vlan下的mac学习功能。



**指定vlan的二层配置**

**l2 fid FID TYPE**

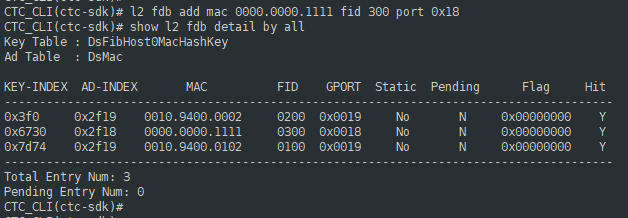
设置指定vlan下的相关二层功能，比如mac学习使能，未知单播或组播的处理等：



**添加动态mac**

**l2 fdb add mac MAC fid FID port GPORT\_ID**

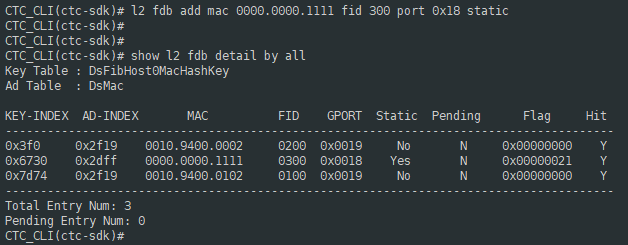
增加一条动态mac表项。



**添加静态mac**

**l2 fdb add mac MAC fid FID port GPORT\_ID static**

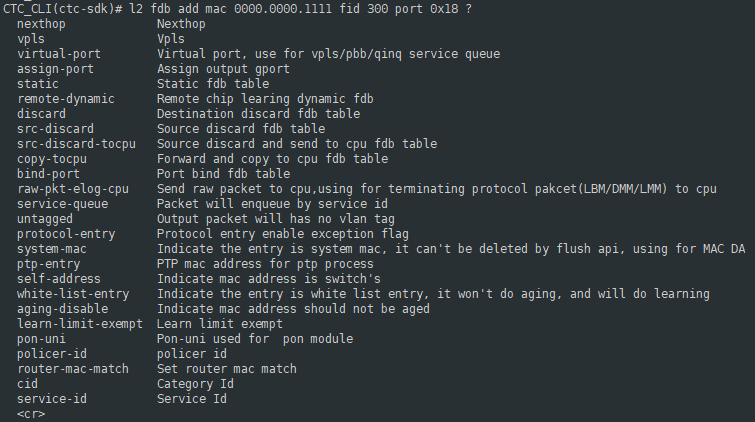
增加一条静态mac表项：



**添加特殊用途的mac表项，黑洞、丢弃、剥掉vlan等**

**l2 fdb add mac MAC fid FID port GPORT\_ID TYPE**

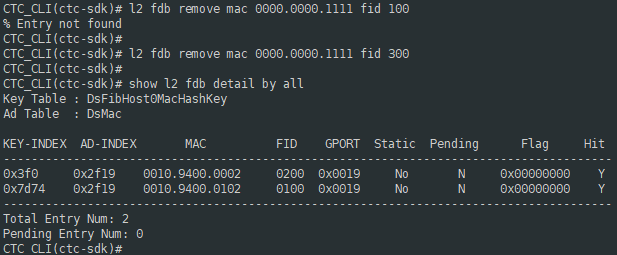
添加的mac表项可以有不同的目的，比如简单的动态和静态表项，也可以指定添加的表项为黑洞mac丢弃，也可以配置该表项对应的报文上送CPU处理等：



**删除指定mac+vlan的mac**

**l2 fdb remove mac MAC fid FID**

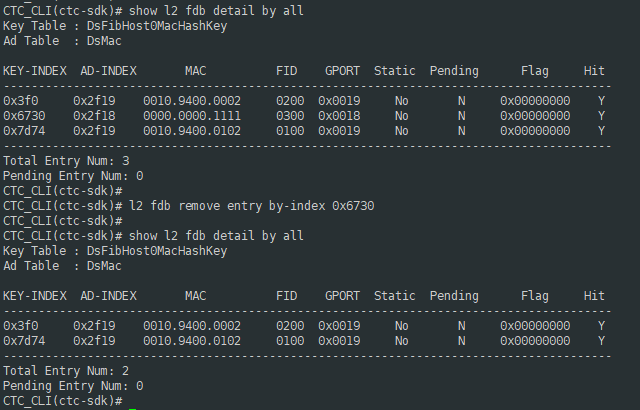
根据mac+vlan条件删除一条动态mac表项。



**删除指定索引的mac**

**l2 fdb remove entry by-index INDEX**

根据mac表项的索引删除对应的表项：

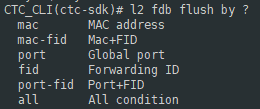


索引值是第一列的KEY-INDEX值，而不是第二列的AD-INDEX。

**删除指定类型的mac**

**l2 fdb flush by TYPE**

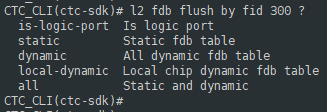
删除指定类型的mac表项：



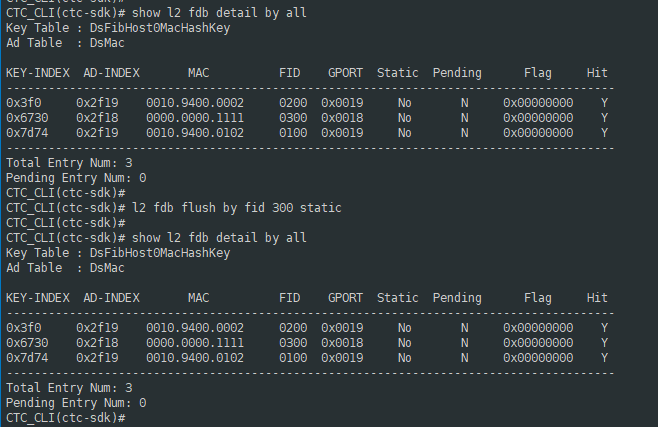
**删除指定vlan的mac**

**l2 fdb flush by fid FID TYPE**

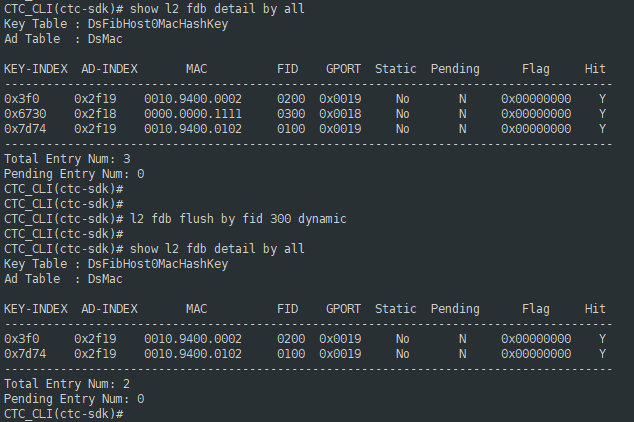
删除指定vlan下的指定类型表项：



可以删除vlan下对应的动态、静态、接口或者所有表项。比如指定删除静态表项时：



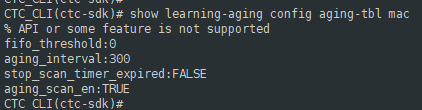
因为vlan 300的表项中没有静态表项，只有一条动态表项，所以指定静态时没有删除，参数指定为动态就可以：



**查看老化时间**

**show learning-aging config aging-tbl mac**

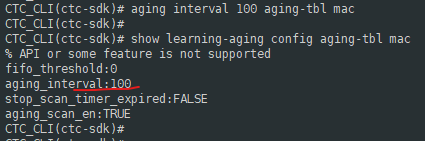
查看mac表项老化时间



**设置老化时间**

**aging interval INTERVAL aging-tbl mac**

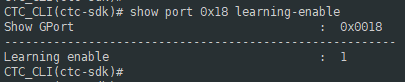
设置mac表项老化时间。



**查看接口mac学习使能**

**show port GPHYPORT\_ID learning-enable**

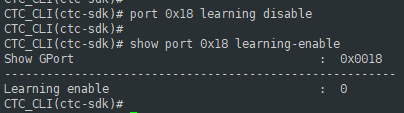
查看接口mac学习使能状态。



**接口的mac学习使能**

**port GPHYPORT\_ID learning (enable|disable)**

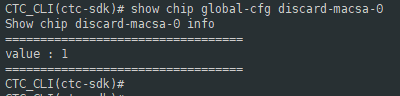
设置接口mac学习使能和关闭。



**查看源mac全0的报文（丢弃或转发）**

**show chip global-cfg discard-macsa-0**

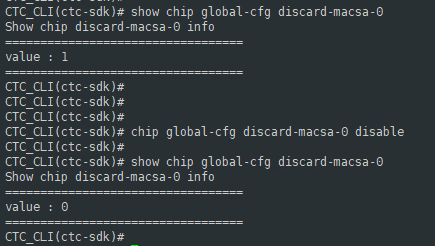
查看源mac全0的报文如何处理，为1表示丢弃。



**设置源mac全0的报文（丢弃或转发）**

**chip global-cfg discard-macsa-0 (enable | disable)**

设置源mac全0的报文如何处理，是丢弃还是转发。

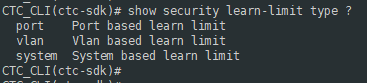


ssp代码初始化enable这项属性，丢弃源mac全0的报文。该条命令设置disable后，源mac全0的报文不再被丢弃，按vlan正常转发。

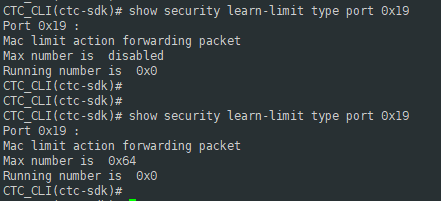
**查看mac学习限制**

**show security learn-limit type TYPE**

设置mac学习限制，包括基于端口、vlan和整机的。



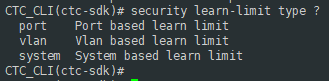
基于端口的学习限制设备默认是没有的，show命令查看显示disable，在接口下使用设备命令（mac-address threshold 100）设置限制100后再次查看：



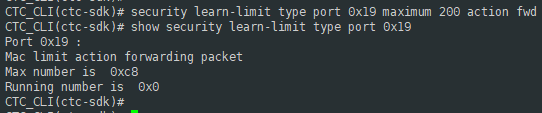
**设置mac学习限制**

**security learn-limit type TYPE**

设置基于端口、vlan和整机的mac学习限制。



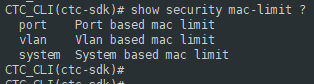
设置接口mac学习限制为200，超过限制后仍然转发：



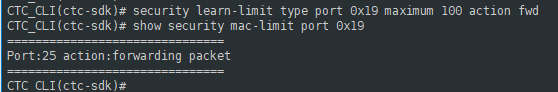
**查看超过mac-limit后的动作（丢弃或转发）**

**show security mac-limit TYPE**

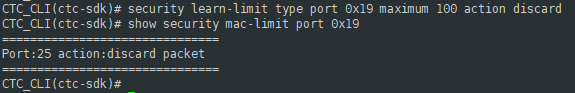
查看配置mac学习限制后，超过限制后报文的处理方式，转发还是丢弃。



配置基于接口的mac学习限制，超过限制后仍然转发，查看：



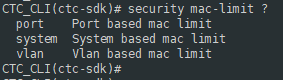
使用learn-limit命令重新配置mac学习限制，超过限制后的动作，改为丢弃：



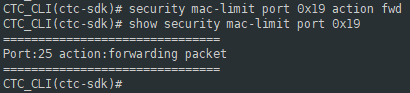
**设置超过mac-limit后的动作（丢弃或转发）**

**security mac-limit TYPE**

设置基于端口、vlan和整机的超过mac学习限制后的处理方式，转发还是丢弃。



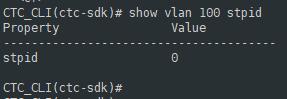
前面learn-limit命令行配置基于端口的mac学习限制动作为丢弃后，使用本命令修改为转发：



**stp**

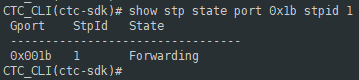
**查看vlan所在stp实例**

show vlan VLAN\_ID stpid



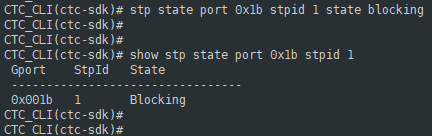
**查看接口stp状态**

show stp state port GPHYPORT\_ID stpid STP\_ID



**设置接口stp状态**

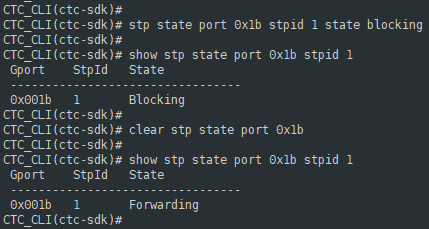
stp state port GPHYPORT\_ID stpid STP\_ID state (forwarding | blocking | learning)



设置接口在指定STP\_ID中的状态是forwarding，blocking或者learning。

**清除接口stp状态**

clear stp state port GPHYPORT\_ID



命令设置接口为默认forwarding状态。