**CTC相关**

**AD表**： associated Data表，通过相应的Key表做的hash查找道德，主要存放出口信息、编辑行为、stats及其他一些相关的控制行为；主要包含DsMac、DsIPDa、DsFlow、DsUserId、DsMpls

光模块类型小知识：

从速率方面来看:

SFP是速率一般为3G以下的光模块封装类型；

SFP+是速率为6G-16G光模块的封装类型；

SFP28是速率为25G/32G光模块的封装类型；

QSFP+是速率为40G/56G光模块的封装类型；

QSFP28是速率为100G的光模块封装类型。

# CTC文档结构

Centec\_SDK\_PG\_R3.2\_20200401\_ch.pdf 介绍CTC sdk的目录架构，内容涉及SDK目录、编译、裁剪等。

CLDK\_User\_Guide\_R4.0\_20200612\_ch.pdf 比较偏BSP和boot

CTC7132\_PG\_R1.1\_190916\_ch.pdf 主要介绍功能的实现原理，芯片流程

CTC7132\_programmer\_register\_R1.1\_191203\_en.pdf 寄存器相关

SDK\_API文档的34.1章节描述了扩展端口的相关属性

# 主控和板卡-MuxDemux

主控和线卡的关系可以参见《CTC8180集中式机架方案\_R1.0\_20200827\_ch.pdf》里面有大致讲解。

# FDB相关

## 老化时间

**aging interval 350 aging-tbl mac**

## 查看老化时间

**sho learning-aging config aging-tbl mac**

## 基于端口关闭、开启学习

**port xxx learning disable | enable**

## 查看MAC地址表

### show l2 fdb detail by all 查看所有MAC地址表详细信息

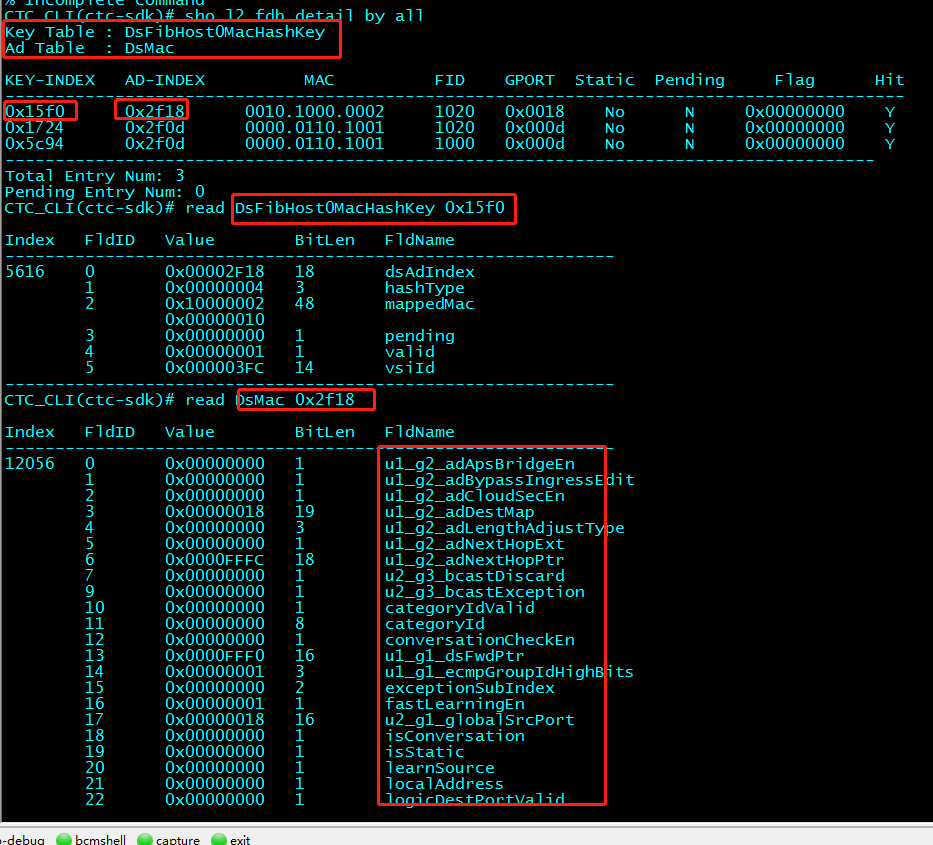
### **show l2 fdb entry by all 查看所有MAC地址表信息**

### **show l2 fdb count by all 查看MAC地址表count**

**Show l2 fdb detail by all 可以查看MAC地址表详细信息**

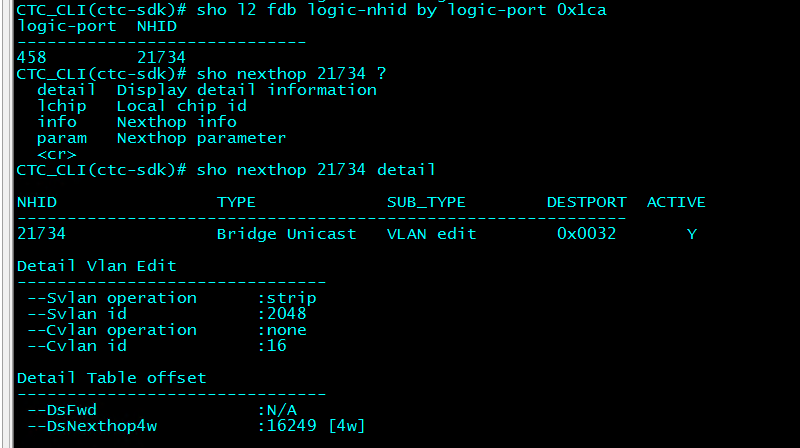
### read DsFibHost0MacHashKey 0x15f0

### read DsMac 0x2f18



### **sho l2 fdb logic-nhid by logic-port 0xxxx**

根据L2 fdb表查找其对应的nexthop表，再根据nexthop表查gport，就知道这个MAC地址表真正对应的物理口了。（N:1的vlan-mapping功能，MAC地址是学在logic port上的，每下发一条vlan-mapping都会增加申请一个logic port，vlan-mapping vlan的MAC都学在logic port上，软件get到MAC地址表后，需要将logic port转换成对应的物理口gport才能对应正确的物理口）



## 端口相关FDB操作

### **show port 0x0000 learning-enable**

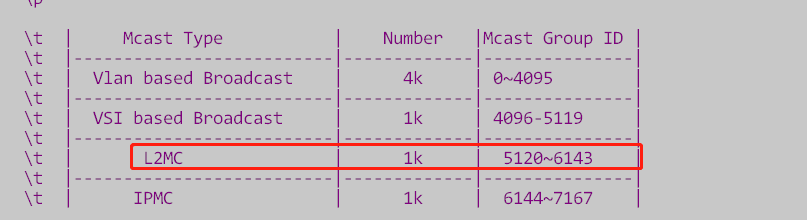
**Port 0x0000 learning enable**

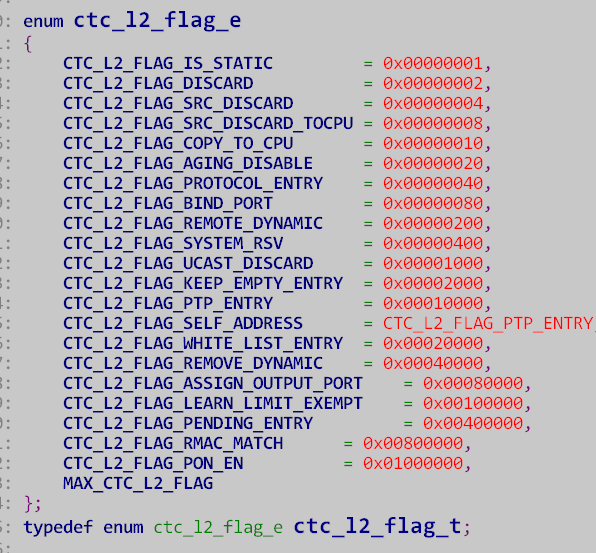
## 白名单

和白名单相关的属性有FID(VLAN、VSI)、MCAST GROUP id，nexthop id，mac，bridge-member等；

白名单功能会关闭VLAN的MAC学习和未知单播、未知组播丢弃，在开启白名单功能的VLAN不要和这几个配置冲突。

白名单能用的**group id，总共有1024个**

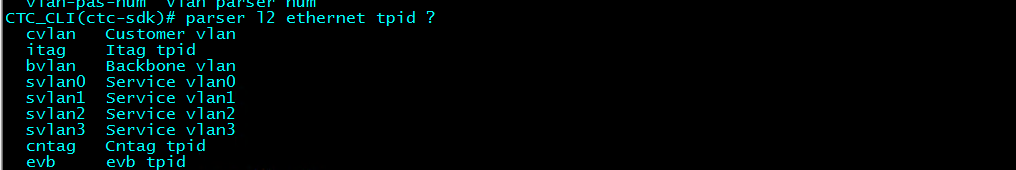




# VLAN相关

## TPID相关

**设置全局tpid**



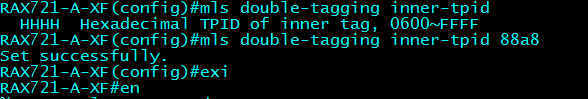
**parser l2 ethernet tpid svlan xxxx**

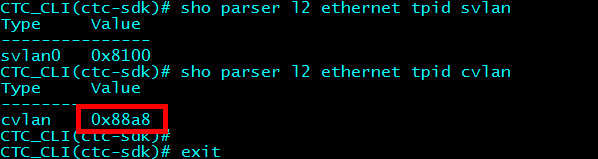
**parser l2 ethernet tpid cvlan xxxx**

**sho parser l2 ethernet tpid cvlan**

**sho parser l2 ethernet tpid svlan**

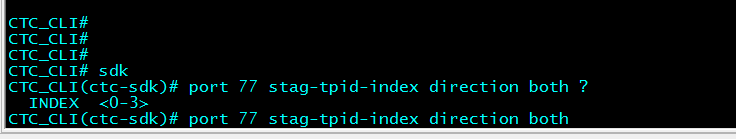
**sho parser l2 ethernet tpid**





### 设置端口的TPID

**port 77 stag-tpid-index direction both xxx**



## 基本VLAN

### 相关命令行

**show port 0x39 vlan-domain**

**show port 0x39 vlan-ctl**

**show port 0x39 dot1q-type**

### 端口加VLAN

**vlan add port 14 vlan 1**

**vlan1 port 14 untagged**

**sho vlan 1 tagged-port**

**sho vlan 1 member-port**

### 端口默认VLAN

**port 14 default vlan 1**

**sho port 14 default-vlan**

### API

**Ctcs\_vlan\_add\_port( )**端口加入VLAN

**Ctcs\_vlan\_set\_tagged\_port()**设置端口加入VLAN的tag模式，untag或者tag

## 基本QINQ

### 相关命令行

**show vlan mapping port 0x39 is-default scl-id 0**

**sho vlan mapping default info port 0x39**

**vlan mapping add port 0x39 mismatch append-stag new-svid 200 scl-id 0**

### API

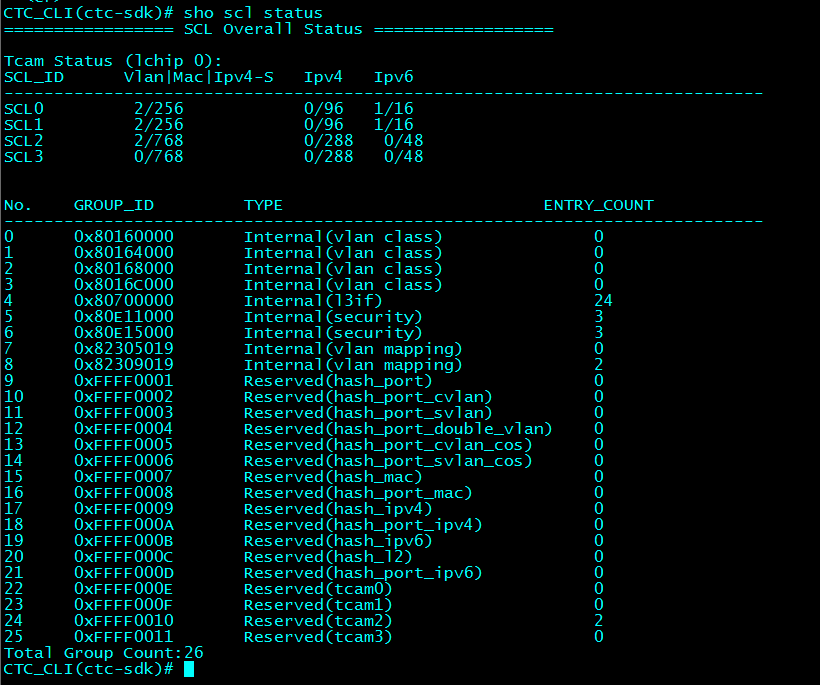
**ctcs\_vlan\_add\_default\_vlan\_mapping(uint8 lchip, uint32 gport, ctc\_vlan\_miss\_t \* p\_action)**

## 灵活QINQ

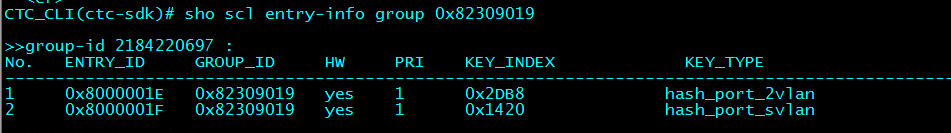
灵活QINQ一般需要入方向下发一条规则修改tag，同时需要在出方向下发一条规则回翻，N:1的vlan-mapping只需要入方向下发规则，回翻是依赖nexthop和mac地址表进行回翻，不需要下发出方向规则。

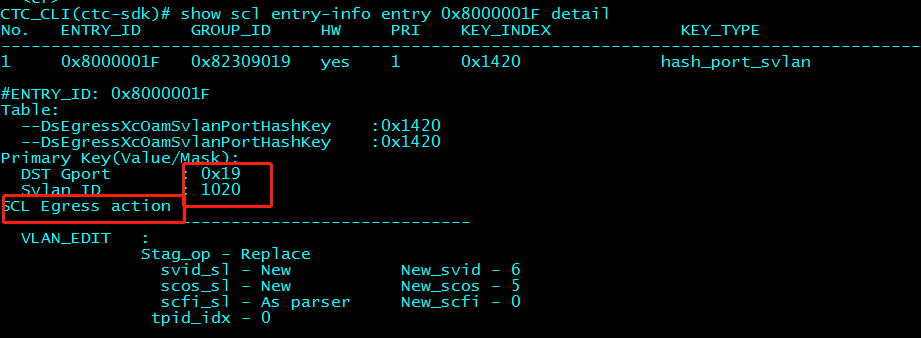
# SCL相关

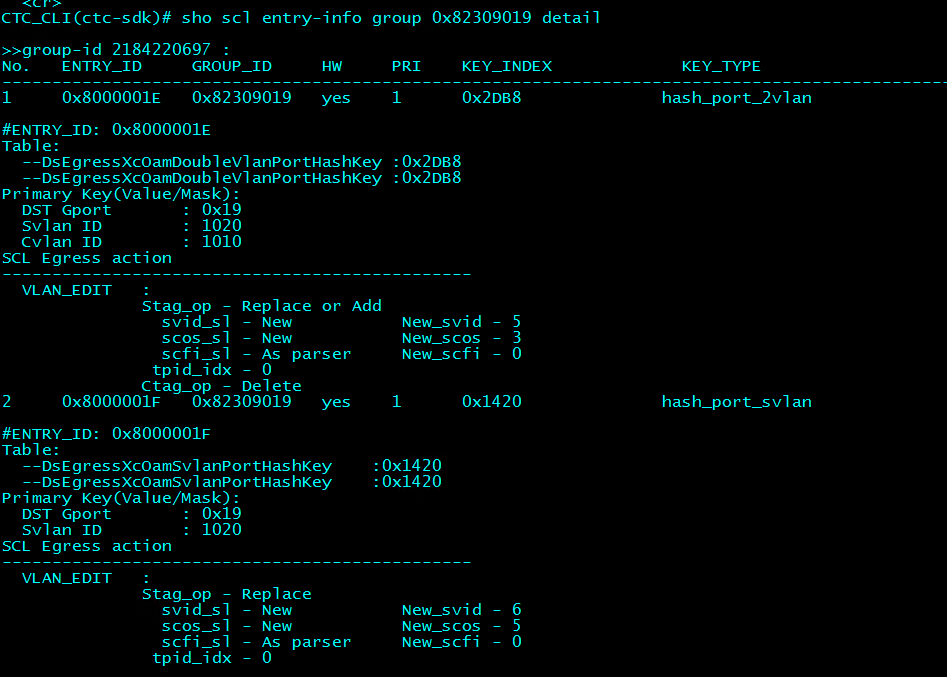
## Show scl status可以查看scl使用情况



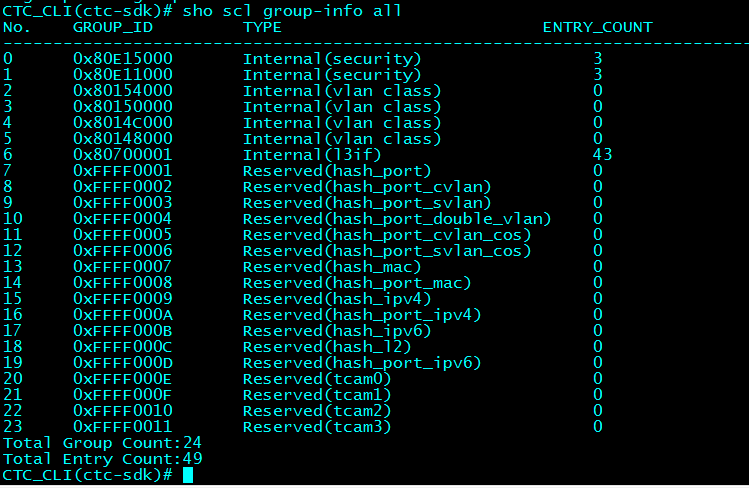
这个信息表里面，Group id是0X82309019是vlan-mapping的出方向回翻的scl表。如果要查看具体scl内容，需要知道entry id具体查看。

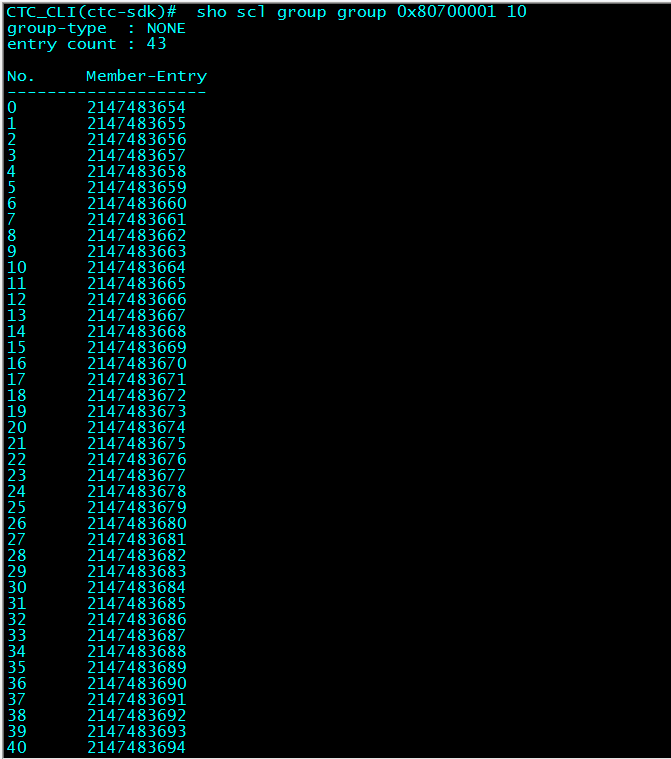




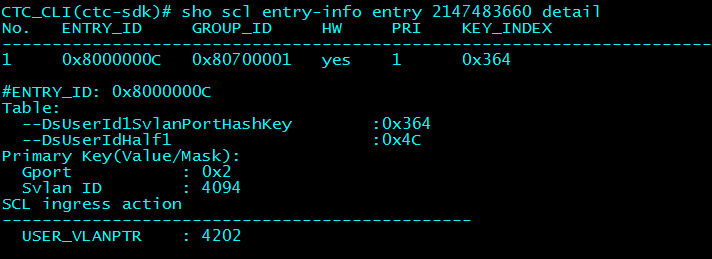


## 查看SCL group里面entry表





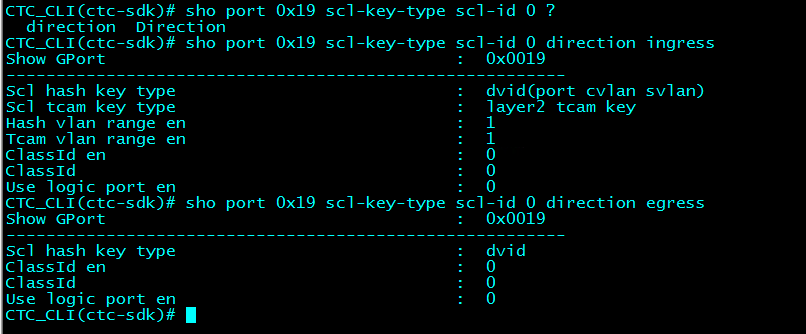
查看具体的entry表



## 查看端口SCL属性

Show port xxx scl-key-type scl-id 0 direction ingress

Show port xxx scl-key-type scl-id 0 direction egress



## MAC/PROTOCOL vlan的SCL配置过程：

**CTC8180**

scl add group 0xffff0010 entry 200 ipv4-entry mac-sa 0000.0000.0006 ffff.ffff.ffff port-class 62 0xff ingress-action vlan-edit stag-op 1 svid-sl 2 new-svid 1860

port 139 scl-key-type scl-id 2 direction ingress type disable tcam-type ip class-id 62

Scl install entry 200

协议VLAN如下配置不生效，eth-type没有下发下去似的；通过代码下发是有的，还需要找下代码和命令行的差别

CTC\_CLI(ctc-sdk)# scl add group 0xffff0010 entry 201 ipv4-entry eth-type 0x0806 0xffff gport 139 0xffff ingress-action vlan-edit stag-op 1 svid-sl 2 new-svid 1540 tpid-index 0

CTC\_CLI(ctc-sdk)# scl install entry 201

**CTC7132**

scl add group 0xffff0010 entry 200 mac-entry mac-sa 0000.0000.0007 ffff.ffff.ffff port-class 62 0xff ingress-action vlan-edit stag-op 1 svid-sl 2 new-svid 1560

port 48 scl-key-type scl-id 3 direction ingress type disable tcam-type mac class-id 62

Scl install entry 200

# 丢包/报文跟踪

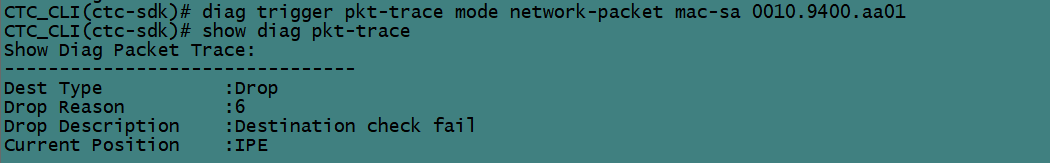
参见SDK用户开发指南SDK\_AN\_DISCAED章节，本章讲解了怎么查丢包原因

Sdk下：

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

Show diag pkt-trace

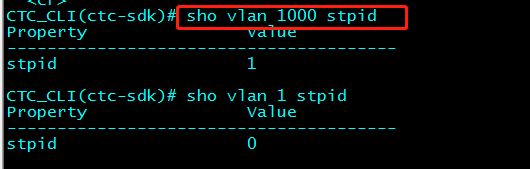
可以跟踪流的丢包环节和原因



# stp相关

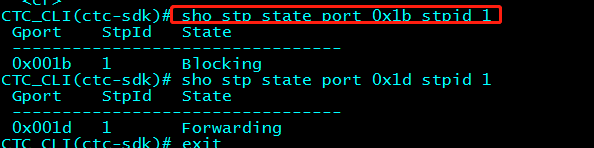
## 查看VLAN所在的STP实例

**sho vlan 1000 stpid**



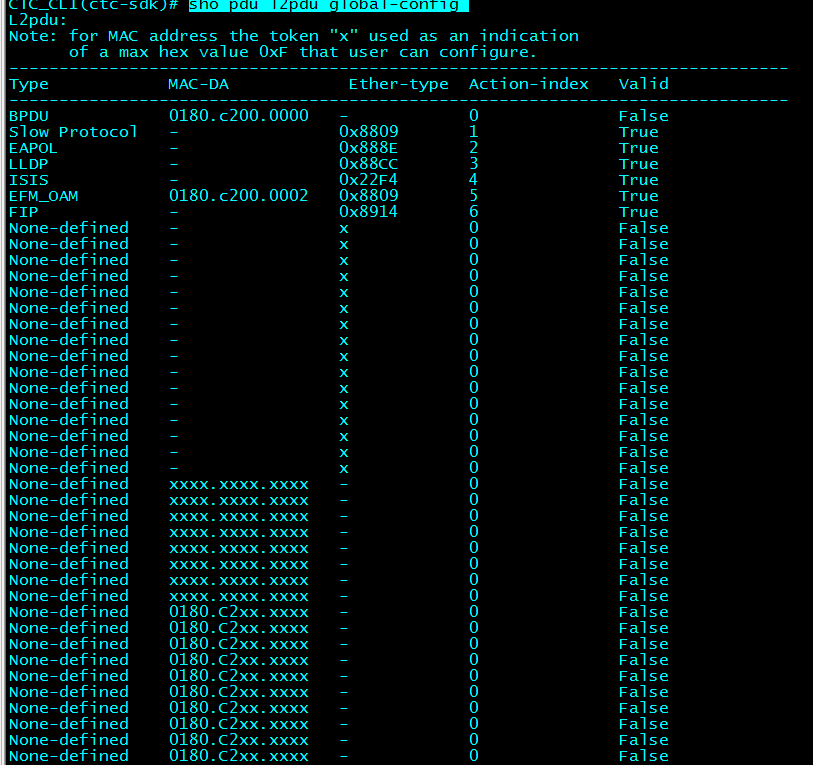
## 查看端口实例状态

**sho stp state port 0x1d stpid 1**

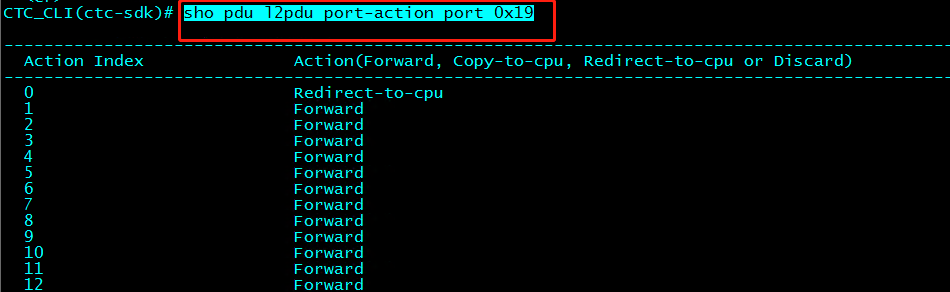


## BPDU报文默认行为

**sho pdu l2pdu global-config**



**sho pdu l2pdu port-action port 0x19**

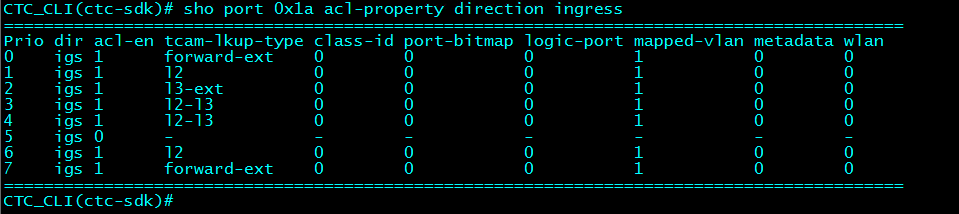


# ACL相关

## 1、查看端口入方向ACL属性：

**sho port 0x1a acl-property direction ingress**

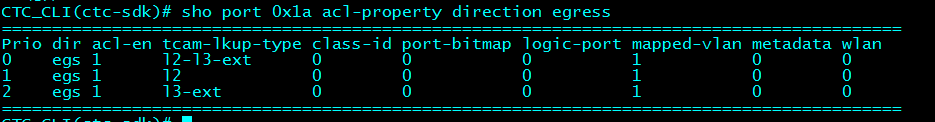
**下发ACL的时候的entry type必须和端口上的保持一致**



## 查看端口出方向ACL属性

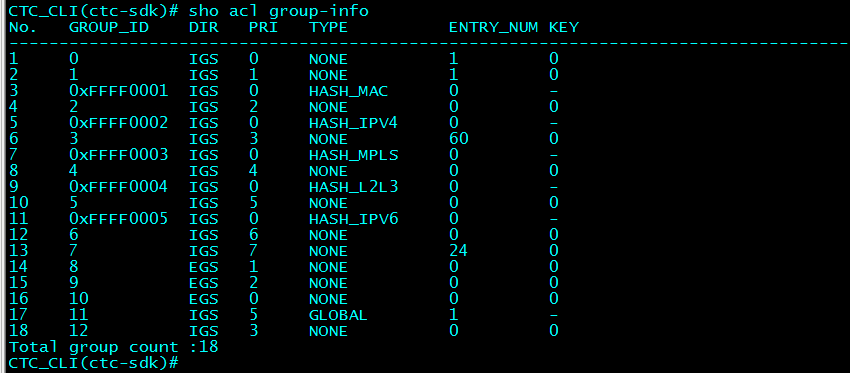
**sho port 0x1a acl-property direction egress**

**下发ACL的时候的entry type必须和端口上的保持一致**



## 查看ACL group设置

**Show acl group-info**



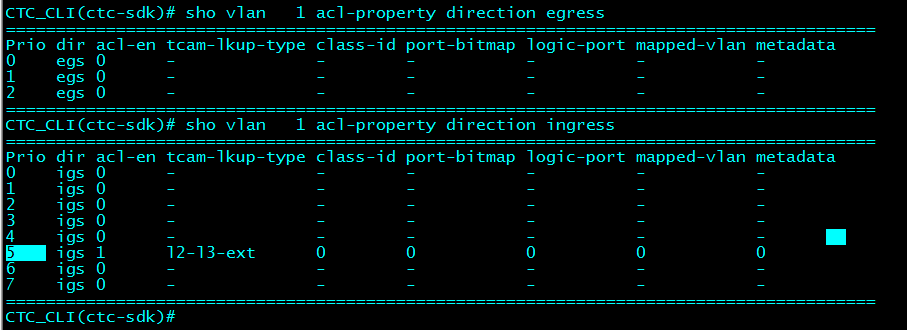
## 查看VLAN的ACL属性设置

**sho vlan 1 acl-property direction egress**

**sho vlan 1 acl-property direction ingress**

**Show port 40 acl-property direction ingress**

**如果ACL不生效的时候，需要看下端口/VLAN的ACL属性是否和配置的ACL一致**



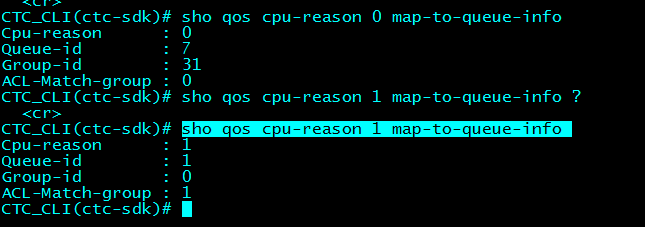
## ACL优先级对应关系

chip global-cfg acl-property direction ingress priority 1 random-log-pri 0

show chip global-cfg acl-property direction ingress priority 1

# Cpu-reason与队列对应关系

**sho qos cpu-reason 1 map-to-queue-info**

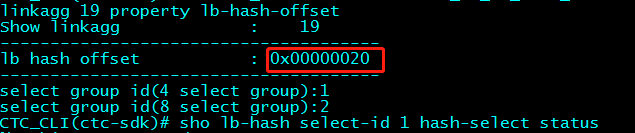


# linkAgg

## 聚合分流算法配置

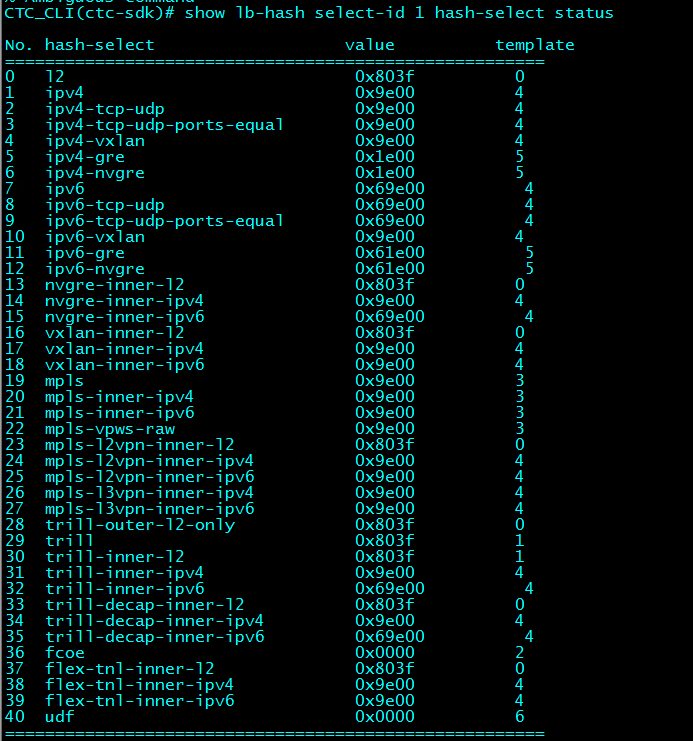
## 聚合分流算法show

**sho linkagg xx property lb-hash-offset** 查看聚合端口的lb-hash-offset

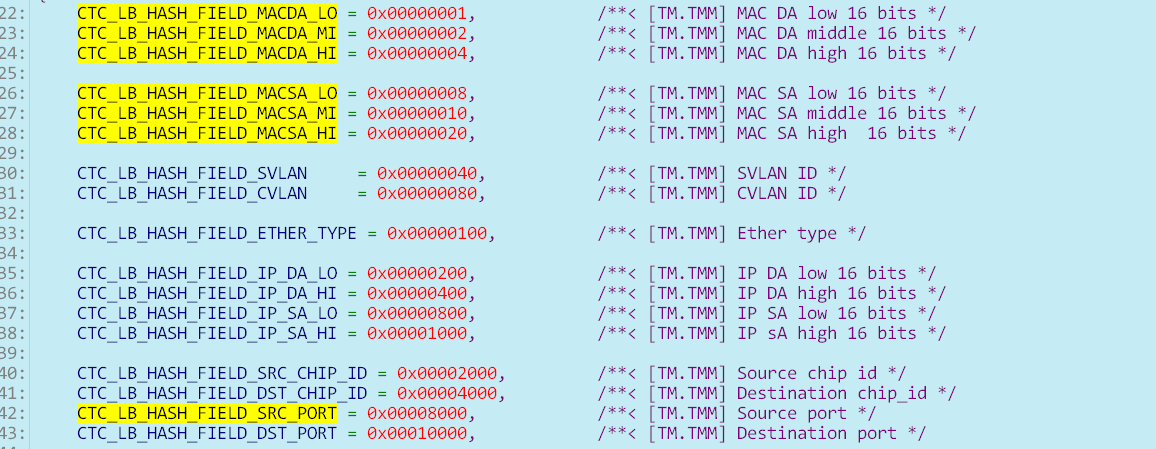


每个select id 生成32bit hash值，select id 0的hash是bit 0-31，select id 1是32-63一次类推，所以lb hash offset 0x20就是在select id。知道聚合口的select id后，查这个select id的查具体的hash算法。

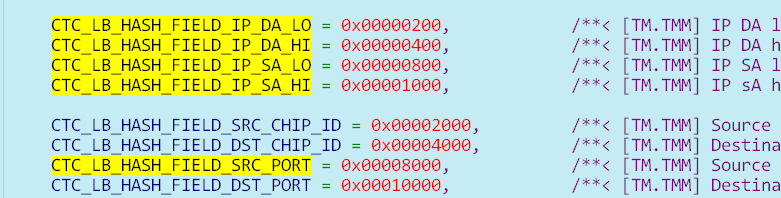
**Show lb-hash select-id xx hash-select status**查看lb-hash select-id的hash元素



上面的表格说明了每种类型的报文有哪些元素参与hash计算分流，可以对照ctc\_lb\_hash\_field\_e枚举。比如上面show的信息中，l2报文的hash值是0x803f，对应**ctc\_lb\_hash\_field\_e**枚举中下面的高亮部分：



IP报文的hash元素有如下部分：



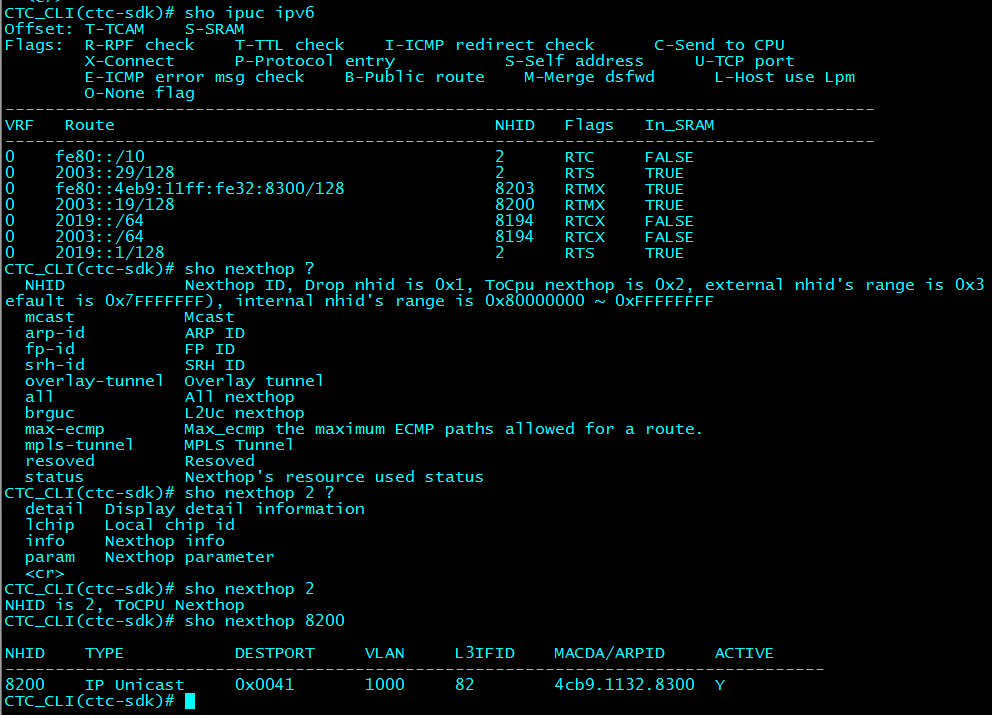
其它以此类推。

## 聚合口配置成员口

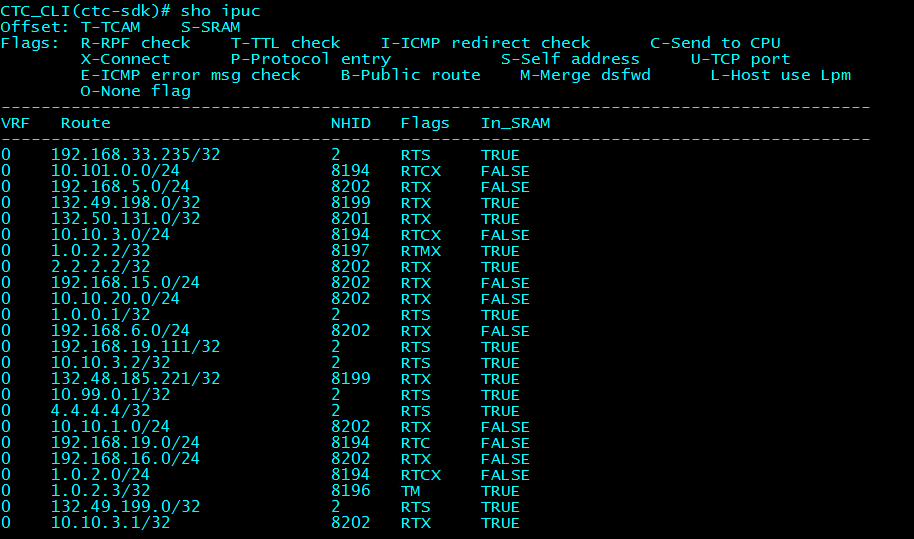
# L3相关

## sho l3if type vlan-if vlan 1000

## Ipv6路由查询



## Ipv4 路由查询



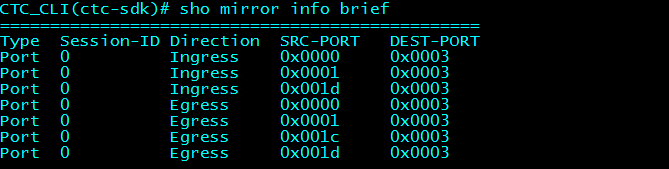
## vlanptr

三层子接口的vlanptr和l3ifid的关系:vlanptr=4096+l3ifid

# Mirror

## 查看mirror信息

**Show mirror info brief**



# 端口相关

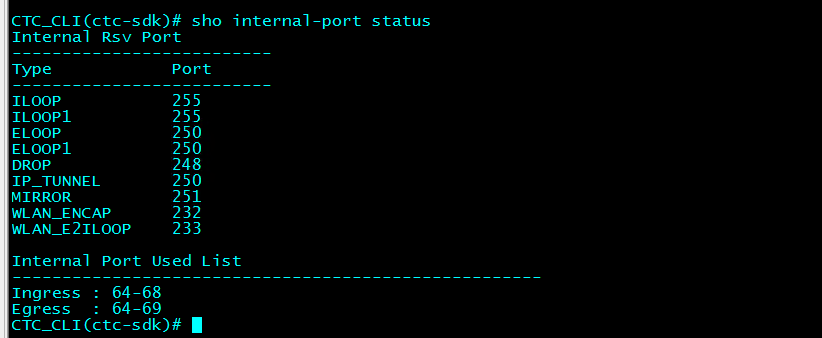
## 端口统计相关

show stats mac-all port 14

Clear stats

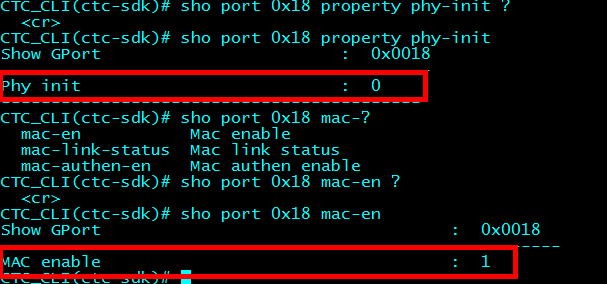
sho qos stats port 14

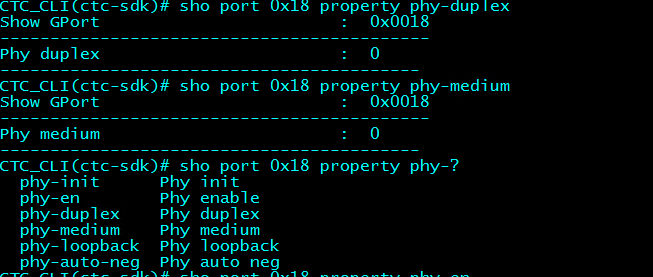
## 看internal-port使用情况，



## 查看端口自协商、duplex、speed等

需要看是phy还是mac，如果是mac，phy的property就没用





相关命令行：

Show port 0x18 mac-en

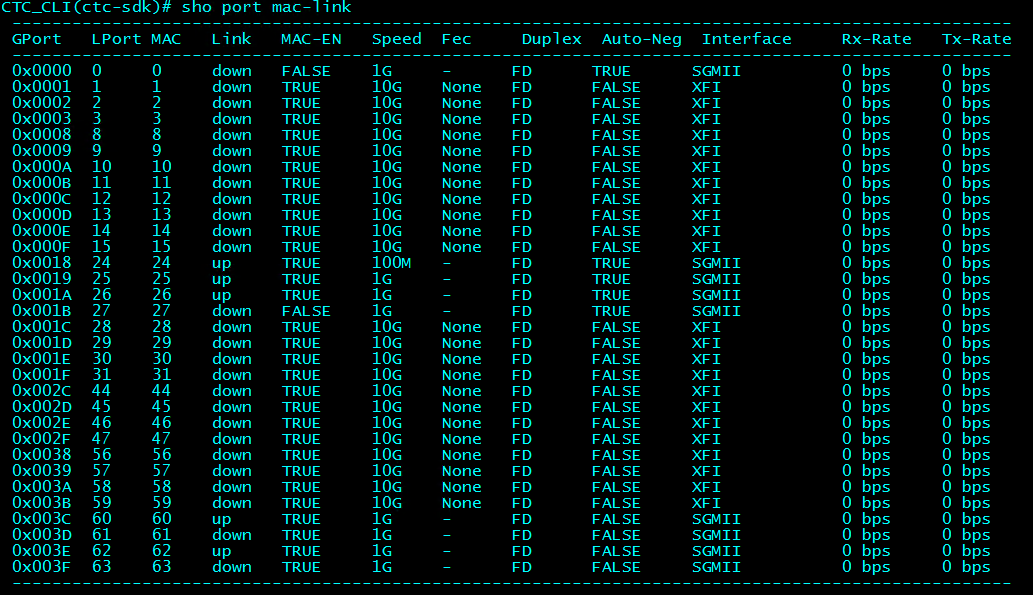
Show port 0x18 property phy-init

sho port 0x18 property phy-auto-neg

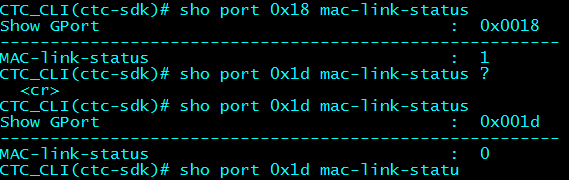
sho port 0x18 property phy-duplex

sho port 0x18 property phy-medium

**show port mac-link**

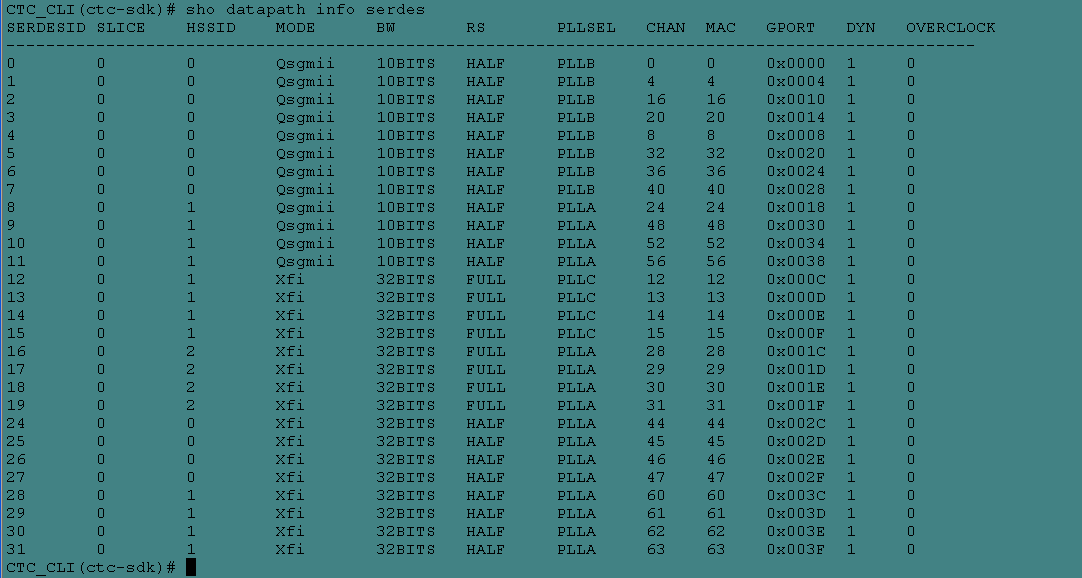


## 查看端口的mac层link状态



## 查看端口和serdes关系

Show datapath info serdes



查看serdes

## 眼图测试命令

Serdes 15 eye width-slow time 5

# 芯片容量

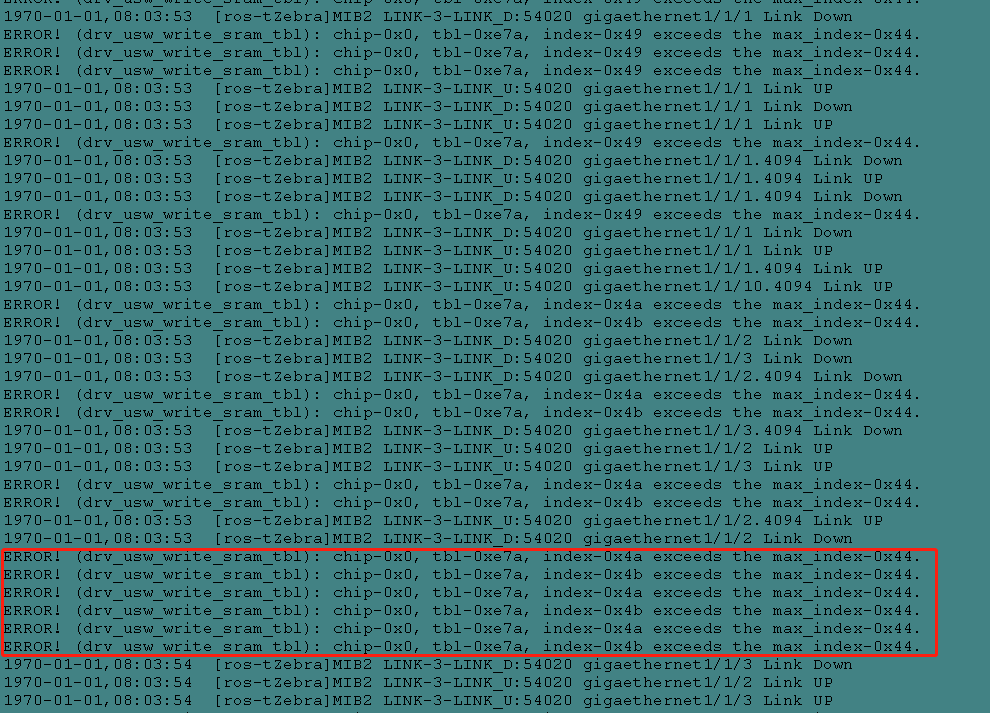
## app模式下Show chip capability

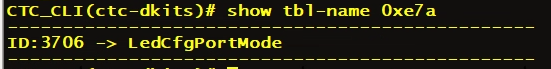
# 扩展端口

## 相关命令行

bpe mux-demux port 1 vlan-base 6 port-base 82 extender-num 5 enable

# 报错排查记录





# SDK发包

1、先把包弄出来，搞成二进制流，wireshark->右键-》复制-》as a HEX stream,然后把结果复制到一个文件，如如aps\_test

2、把文件传到/usr/local/ros/bin目录下

3、进sdk命令行，输入packet tx mode 0 dest-gport 1 pkt-file aps\_test pkt-len 100 count 1