

# Catalyst 6500交换机配置ERSPAN造成CPU高的问题以及解决办法

## 目录

- [软硬件平台](#)
- [问题描述](#)
- [故障诊断步骤](#)
- [场景 1](#)
- [场景 2](#)
- [相关文档](#)

## 软硬件平台

硬件平台：Catalyst 6500 系列交换机

当前，支持ERSPAN的硬件为Supervisor 720 配置 PFC3（其中PFC3A需要硬件版本号为3.2或以上）。

硬件平台：Catalyst 6500 系列交换机

## 问题描述

和传统的SPAN一样，ERSPAN能够将SPAN源接口的流量复制到目的接口上，但是ERSPAN更吸引人的地方在于，被监控接口和目标接口可以在不同的网络设备上，利用IP GRE隧道，使得用户可以跨越3层网络进行远程多台设备的诊断和镜像工作。

图1对比了本地SPAN（Local SPAN），远程SPAN (RSPAN)和ERSPAN的区别：

ERSPAN的配置并不复杂，但是在配置的过程中，如果稍不注意，可能会导致Catalyst 6500 系列交换机CPU高的问题。本文就其中两个最典型的ERSPAN造成高CPU的场景进行分析，并给出解决办法。

## 故障诊断步骤

### 场景 1

在ERSPAN的目的设备上，发现CPU高，而且绝大部分的CPU是中断（interrupt）高（X%/Y%，其中X代表CPU的总数值，Y代表中断，如果Y值很高，则表示中断造成了CPU高的现象，其占据了CPU利用率的绝大部分）。例如我们看到下面的CPU进程输出，中断为90%，中断数值过高一般意

意味着有过的流量流向了CPU，进行了软件转发。

```
6506#show proc cpu
CPU utilization for five seconds: 99%/90%; one minute: 99%; five minutes: 99%
```

通过检查配置，我们发现，在ERSPAN的源设备上，其源会话的配置完全正确：

```
6506(config)#monitor session 1 type erspan-source
6506(config-mon-erspan-src)#source vlan 10
6506(config-mon-erspan-src)#destination
6506(config-mon-erspan-src-dst)#ip address 10.100.100.1
6506(config-mon-erspan-src-dst)#erspan-id 1
6506(config-mon-erspan-src-dst)#origin ip address 10.10.10.1
6506(config-mon-erspan-src-dst)#exit
6506(config-mon-erspan-src)#no shut
6506(config-mon-erspan-src)#exit
```

但是在ERSPAN的目的设备上，只在loopback 0接口下配置了目的IP地址，并没有ERSPAN目的会话的相关配置（或者可能错误的删除掉了相关配置）：

```
6509e(config)#interface loopback0
6509e(config-if)#ip address 10.100.100.1 255.255.255.255
```

这便是造成ERSPAN目的设备CPU高的根本原因。如果在源设备上，ERSPAN的源会话配置完毕并且已经激活，该会话就会把抓到的数据包发向ERSPAN的目的接口。而在目的设备上，目的ip地址已经配置完毕且接口是开启状态，但是ERSPAN的目的会话功能并没有被激活，因此从源设备过来的ERSPAN流量会直接流向目的设备的CPU，造成CPU中断过高。为了避免这样的问题，则需要相应的在目的设备上增加ERSPAN目的会话的相关配置，来终结去向CPU的ERSPAN流量。事实上，推荐的最简洁的目的会话配置可以是只包括目的会话ID以及IP地址的配置，例如：

```
6509e(config)#monitor session 1
6509e(config-mon-erspan-dst)#source
6509e(config-mon-erspan-dst-src)#ip address 10.100.100.1
```

由此可见，如果ERSPAN目的设备的目的会话配置有误或者不完全，则会造成目的设备高CPU现象。为了避免该问题，较合理的配置顺序为先在目的设备上配置ERSPAN会话终结功能，再配置ERSPAN源设备。

## 场景 2

在ERSPAN的源设备上，CPU高，且中断高。

首先，检查相关配置，发现源和目的会话的配置都正确，如下：

ERSPAN源会话配置：

```
6506(config)#monitor session 1 type erspan-source
6506(config-mon-erspan-src)#source vlan 10
6506(config-mon-erspan-src)#destination
6506(config-mon-erspan-src-dst)#ip address 10.100.100.1
6506(config-mon-erspan-src-dst)#erspan-id 1
6506(config-mon-erspan-src-dst)#origin ip address 10.10.10.1
6506(config-mon-erspan-src-dst)#exit
6506(config-mon-erspan-src)#no shut
6506(config-mon-erspan-src)#exit
```

ERSPAN目的会话配置：

```
6509e(config)#monitor session 1 type erspan-destination
6509e(config-mon-erspan-dst)#destination int gig 7/25
6509e(config-mon-erspan-dst)#source
6509e(config-mon-erspan-dst-src)#ip address 10.100.100.1
6509e(config-mon-erspan-dst-src)#erspan-id 1
6509e(config-mon-erspan-dst-src)#exit
6509e(config-mon-erspan-dst)#no shut
6509e(config-mon-erspan-dst)#exit
```

于是我们可以在ERSPAN源设备上通过使用Netdr抓取流向CPU的数据包：

```
6506#debug netdr capture rx
6506#no debug netdr capture rx
6506#show netdr captured-packets
```

通过 **show netdr captured-packets** 的输出，我们发现，绝大多数去CPU的数据包都是长度大于1500 byte的GRE数据包，并且DF位为1。

----- dump of incoming inband packet -----

```
6506#debug netdr capture rx
6506#no debug netdr capture rx
6506#show netdr captured-packets
```

通过检查接口配置发现，其接口的MTU配置为1500 byte。

一般来说，Catalyst 6500交换机会把超过接口MTU值而造成MTU检测失败的数据包发到CPU进行分片或产生ICMP不可达信息。ERSPAN的源设备不会对MTU检测失败的ERSPAN数据包进行分片处理。实际上，源设备把ERSPAN的数据包的DF位置为1，防止该数据包在ERSPAN的全程路径上进行分片处理。而且ERSPAN的目的会话也不会对分片了的数据包进行重组。因此，如果ERSPAN数据包的大小超过了接口的MTU数值，导致MTU检测失败，该数据包会被发往CPU，最终被丢弃。

ERSPAN可以支持的最大3层数据包的大小为9202 byte。因此，为了不造成MTU检测失败，导致CPU高和数据包的丢弃，就需要在ERSPAN的路径上的所有接口上配置足够大的MTU数值。对于3层以太网接口，可以配置的MTU范围为64 byte到9216 byte。因此，对于ERSPAN，推荐的MTU配置为9216 byte，且该MTU数值应该应用于从ERSPAN的源设备到目的设备中间经过的所有接口上。

当然，对于MTU检测失败造成CPU高的问题，我们也可以通过使用命令 **mls rate-limit all mtu-failure rate**（例如，可以将rate数值设为10 pps）来限制由于超过接口MTU值而流向CPU的数据包的数量，但是这样做只能缓解CPU的压力，并不能防止超出MTU的ERSPAN数据包丢失。

## 相关文档

[Catalyst Switched Port Analyzer \(SPAN\)配置示例](#)

[ERSPAN Guidelines and Restrictions](#)