排除Catalyst 9000 DHCP中继代理上的慢速或间断DHCP故障

目录

简介

本文档介绍如何排除作为DHCP中继代理的Catalyst 9000系列交换机上的慢速动态主机配置协议(DHCP)地址分配或间歇性DHCP地址分配故障。

先决条件

要求

Cisco 建议您了解以下主题：

- DHCP和DHCP中继代理
- Internet Control Message Protocol (ICMP)
- 控制层面策略 (CoPP)

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Catalyst 9000 系列交换机
- Cisco IOS XE®版本16.x和17.x

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

相关产品

本文档也可用于以下硬件和软件版本：

- 采用Cisco IOS XE® 16.x的Catalyst 3650/3850系列交换机

背景信息

控制平面策略(CoPP)功能可保护CPU免受不必要的流量和拒绝服务(DoS)攻击，从而提高设备的安全性。它还可以保护控制流量和管理流量，使其不会因其他优先级较低的流量大而丢弃流量。

您的设备通常分为三个操作平面，每个平面都有其自己的目标：

- 数据平面，用于转发数据包。
- 控制平面，用于正确路由数据。
- 管理平面，用于管理网络元素。

您可以使用CoPP保护大多数CPU绑定的流量，并确保路由稳定性、可达性和数据包交付。最重要的是，您可以使用CoPP保护CPU免受DoS攻击。

CoPP使用模块化QoS命令行界面(MQC)和CPU队列来实现这些目标。根据特定条件将不同类型的控制平面流量组合在一起，并分配给CPU队列。您可以通过配置硬件中的专用策略器来管理这些CPU队列。例如，您可以修改特定CPU队列（流量类型）的监察器速率，或者可以禁用特定类型流量的监察器。

虽然策略器在硬件中配置，但CoPP不会影响CPU性能或数据平面的性能。但是，由于它限制了通向CPU的数据包数量，因此会控制CPU负载。这意味着等待来自硬件的数据包的服务可以看到更受控的入口数据包速率（该速率可由用户配置）。

问题

在路由接口或SVI上配置**ip helper-address**命令时，Catalyst 9000交换机被配置为DHCP中继代理。配置帮助地址所在的接口通常是下游客户端的默认网关。为使交换机向其客户端提供成功的DHCP中继服务，它必须能够处理入站DHCP发现消息。这要求交换机接收DHCP发现并将此数据包推送到其CPU进行处理。收到并处理DHCP发现后，中继代理会从接收DHCP发现的接口创建一个新的单播数据包，该数据包的目的地为**ip helper-address**配置中定义的IP地址。创建数据包后，硬件会转发数据包并发送到DHCP服务器，然后对其进行处理，最后将其发送回中继代理，以便客户端的DHCP过程可以继续。

出现的一个常见问题是，中继代理的DHCP事务数据包由于受到特定ICMP方案（如ICMP重定向或ICMP目标不可达消息）的影响而无意中受到发送到CPU的流量的影响。此行为可能表现为客户端无法及时从DHCP获取IP地址，甚至无法完成DHCP分配失败。在某些场景中，可能只在一天中的某些时间观察此行为，例如网络负载完全最大化时的峰值工作时间。

如背景部分所述，Catalyst 9000系列交换机具有设备上配置和启用的默认CoPP策略。此CoPP策略用作服务质量(QoS)策略，该策略位于前面板端口上接收的流量路径中，并且发往设备CPU。它根据流量类型和策略中配置的预定义阈值对流量进行速率限制。默认情况下分类和速率有限的流量示例包括路由控制数据包（通常标有DSCP CS6）、拓扑控制数据包(STP BPDU)和低延迟数据包(BFD)。这些数据包应优先处理，因为能够可靠地处理它们会导致稳定的网络环境。

使用**show platform hardware fed switch active qos queue stats internal cpu policer**命令查看CoPP监察器统计信息。

ICMP重定向队列（队列6）和BROADCAST队列（队列12）共享相同的PlcIdx（0个监察器索引）。这意味着需要由设备CPU处理的所有广播流量（例如DHCP发现）与也发往ICMP重定向队列中设备CPU的流量共享。这会导致前面提到的问题，因为ICMP重定向队列流量会耗尽需要由BROADCAST队列提供服务的流量，从而导致合法广播数据包被丢弃，从而导致DHCP事务失败。

```
9300-Switch#show platform hardware fed switch active qos queue stats internal cpu policer

CPU Queue Statistics
============================================================================================
(default) (set) Queue Queue
QId PlcIdx Queue Name Enabled Rate Rate Drop(Bytes) Drop(Frames)
--------------------------------------------------------------------------------------------
0 11 DOT1X Auth Yes 1000 1000 0 0
1 1 L2 Control Yes 2000 2000 0 0
2 14 Forus traffic Yes 4000 4000 0 0
3 0 ICMP GEN Yes 600 600 0 0
4 2 Routing Control Yes 5400 5400 0 0
5 14 Forus Address resolution Yes 4000 4000 0 0
6    0      ICMP Redirect               Yes     600       600      0            0  <-- Policer
Index 0
7 16 Inter FED Traffic Yes 2000 2000 0 0
8 4 L2 LVX Cont Pack Yes 1000 1000 0 0
9 19 EWLC Control Yes 13000 13000 0 0
10 16 EWLC Data Yes 2000 2000 0 0
11 13 L2 LVX Data Pack Yes 1000 1000 0 0
12   0      BROADCAST                   Yes     600       600      0            0  <-- Policer
Index 0
13 10 Openflow Yes 200 200 0 0
14 13 Sw forwarding Yes 1000 1000 0 0
15 8 Topology Control Yes 13000 16000 0 0
16 12 Proto Snooping Yes 2000 2000 0 0
17 6 DHCP Snooping Yes 500 500 0 0
18 13 Transit Traffic Yes 1000 1000 0 0
19 10 RPF Failed Yes 250 250 0 0
20 15 MCAST END STATION Yes 2000 2000 0 0
<snip>
```

超过CoPP策略中默认每秒600数据包速率的流量在到达CPU之前被丢弃。

```
9300-Switch#show platform hardware fed switch active qos queue stats internal cpu policer

CPU Queue Statistics
============================================================================================
(default) (set) Queue Queue
QId PlcIdx Queue Name Enabled Rate Rate Drop(Bytes) Drop(Frames)
--------------------------------------------------------------------------------------------
0 11 DOT1X Auth Yes 1000 1000 0 0
1 1 L2 Control Yes 2000 2000 0 0
2 14 Forus traffic Yes 4000 4000 0 0
3 0 ICMP GEN Yes 600 600 0 0
4 2 Routing Control Yes 5400 5400 0 0
5 14 Forus Address resolution Yes 4000 4000 0 0
6    0      ICMP Redirect               Yes     600       600      3063106173577 3925209161
<-- Dropped packets in queue
7 16 Inter FED Traffic Yes 2000 2000 0 0
8 4 L2 LVX Cont Pack Yes 1000 1000 0 0
9 19 EWLC Control Yes 13000 13000 0 0
10 16 EWLC Data Yes 2000 2000 0 0
11 13 L2 LVX Data Pack Yes 1000 1000 0 0
12   0      BROADCAST                   Yes     600       600      1082560387 3133323
<-- Dropped packets in queue
```

```
13 10 Openflow Yes 200 200 0 0
14 13 Sw forwarding Yes 1000 1000 0 0
15 8 Topology Control Yes 13000 16000 0 0
16 12 Proto Snooping Yes 2000 2000 0 0
17 6 DHCP Snooping Yes 500 500 0 0
18 13 Transit Traffic Yes 1000 1000 0 0
19 10 RPF Failed Yes 250 250 0 0
20 15 MCAST END STATION Yes 2000 2000 0 0
<snip>
```

场景1:ICMP重定向

考虑第一种方案的此拓扑：

事件顺序如下：

1. 10.10.10.100上的用户发起到设备10.100.100（远程网络）的telnet连接。

2.目的IP位于不同的子网中，因此数据包将发送到用户的默认网关10.10.10.15。

3.当Catalyst 9300收到此数据包以进行路由时，它会将该数据包传送给其CPU以生成ICMP重定向。

之所以生成ICMP重定向，是因为，从9300交换机的角度来看，笔记本电脑直接将此数据包发送到10.10.10.1处的路由器会更有效，因为无论采用何种方式，该数据包都是Catalyst 9300的下一跳，而且它位于用户所在的VLAN中。

问题在于整个流在CPU上处理，因为它符合ICMP重定向标准。如果其他设备发送符合ICMP重定向方案的流量，则更多流量开始被传送到此队列中的CPU，这可能会影响BROADCAST队列，因为它们共享同一个CoPP监察器。

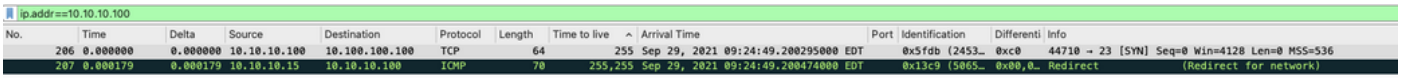
调试ICMP以查看ICMP重定向系统日志。

```
9300-Switch#debug ip icmp       <-- enables ICMP debugs
ICMP packet debugging is on
9300-Switch#show logging | inc ICMP
*Sep 29 12:41:33.217: ICMP: echo reply sent, src 10.10.10.15, dst 10.10.10.100, topology BASE,
dscp 0 topoid 0
*Sep 29 12:41:33.218: ICMP: echo reply sent, src 10.10.10.15, dst 10.10.10.100, topology BASE,
dscp 0 topoid 0
*Sep 29 12:41:33.219: ICMP: echo reply sent, src 10.10.10.15, dst 10.10.10.100, topology BASE,
dscp 0 topoid 0
*Sep 29 12:41:33.219: ICMP: echo reply sent, src 10.10.10.15, dst 10.10.10.100, topology BASE,
dscp 0 topoid 0
*Sep 29 12:43:08.127: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
10.10.10.1
*Sep 29 12:50:09.517: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
10.10.10.1
*Sep 29 12:50:10.017: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
```

```
10.10.10.1     <-- ICMP Redirect to use 10.10.10.1 as Gateway
*Sep 29 12:50:14.293: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
10.10.10.1
*Sep 29 12:50:19.053: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
10.10.10.1
*Sep 29 12:50:23.797: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
10.10.10.1
*Sep 29 12:50:28.537: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
10.10.10.1
*Sep 29 12:50:33.284: ICMP: redirect sent to 10.10.10.100 for dest 10.100.100.100, use gw
10.10.10.1
```

注意：由于规模庞大，建议在启用ICMP调试之前禁用控制台日志记录和终端监控。

Catalyst 9300 CPU上的嵌入式数据包捕获显示CPU上Telnet连接的初始TCP SYN以及生成的ICMP重定向。



ICMP重定向数据包源自发往客户端的Catalyst 9300 VLAN 10接口，并包含ICMP重定向数据包发送到的原始数据包报头。

```
▼ Internet Protocol Version 4, Src: 10.10.10.15, Dst: 10.10.10.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x13c9 (5065)
  ▶ Flags: 0x0000
    Time to live: 255
    Protocol: ICMP (1)
    Header checksum: 0x7f75 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.10.10.15
    Destination: 10.10.10.100
▼ Internet Control Message Protocol
    Type: 5 (Redirect)
    Code: 0 (Redirect for network)
    Checksum: 0x2bec [correct]
    [Checksum Status: Good]
    Gateway address: 10.10.10.1
  ▼ Internet Protocol Version 4, Src: 10.10.10.100, Dst: 10.100.100.100
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
      Total Length: 44
      Identification: 0x5fdb (24539)
    ▶ Flags: 0x0000
      Time to live: 255
      Protocol: TCP (6)
      Header checksum: 0xd7fa [validation disabled]
      [Header checksum status: Unverified]
      Source: 10.10.10.100
      Destination: 10.100.100.100
  ▶ Transmission Control Protocol, Src Port: 44710, Dst Port: 23
```

解决方案

在此场景中，可以阻止向CPU传送的数据包，这也会停止ICMP重定向数据包的生成。

现代操作系统不使用ICMP重定向消息，因此生成、发送和处理这些数据包所需的资源并不是有效使用网络设备上的CPU资源。

或者，将用户指向使用默认网关10.10.10.1，但此类配置可能出于某种原因而存在，并且不在本文档的讨论范围之内。

只需使用**no ip redirects** CLI禁用ICMP重定向。

```
9300-Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
9300-Switch(config)#interface vlan 10
9300-Switch(config-if)#no ip redirects       <-- disable IP redirects
9300-Switch(config-if)#end
```

验证接口上是否禁用了ICMP重定向。

```
9300-Switch#show ip interface vlan 10
Vlan10 is up, line protocol is up
Internet address is 10.10.10.15/24
Broadcast address is 255.255.255.255
Address determined by setup command
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Multicast reserved groups joined: 224.0.0.102
Outgoing Common access list is not set
Outgoing access list is not set
Inbound Common access list is not set
Inbound access list is BLOCK-TELNET
Proxy ARP is disabled
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
ICMP redirects are never sent       <-- redirects disabled
ICMP unreachables are never sent
ICMP mask replies are never sent
IP fast switching is enabled
IP Flow switching is disabled
IP CEF switching is enabled
IP CEF switching turbo vector
<snip>
```

有关ICMP重定向及其发送时间的详细信息，请访问以下链接：https://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13714-43.html

场景2:ICMP不可达

请考虑同一个拓扑，位于10.10.10.100的用户发起到10.100.100的Telnet连接。这次，访问列表已配置到阻止telnet连接的VLAN 10 SVI的入站方向。


```
9300-Switch#show running-config interface vlan 10
Building Configuration..

Current Configuration : 491 bytes
!
interface Vlan10
ip address 10.10.10.15 255.255.255.0
no ip proxy-arp
ip access-group BLOCK-TELNET in              <-- inbound ACL
end
9300-Switch#
9300-Switch#show ip access-list BLOCK-TELNET
Extended IP access list BLOCK-TELNET
10 deny tcp any any eq telnet            <-- block telnet
20 permit ip any any
9300-Switch#
```

事件顺序如下：

1.位于10.10.10.100的用户发起到设备10.100.100的telnet连接。

2.目的IP位于不同的子网中，因此数据包将发送到用户的默认网关。

3.当Catalyst 9300收到此数据包时，会根据入站ACL对其进行评估并阻止该数据包。

4.由于数据包被阻止，且接口上启用了IP不可达，因此数据包将被传送到CPU，以便设备可以生成ICMP目标不可达数据包。

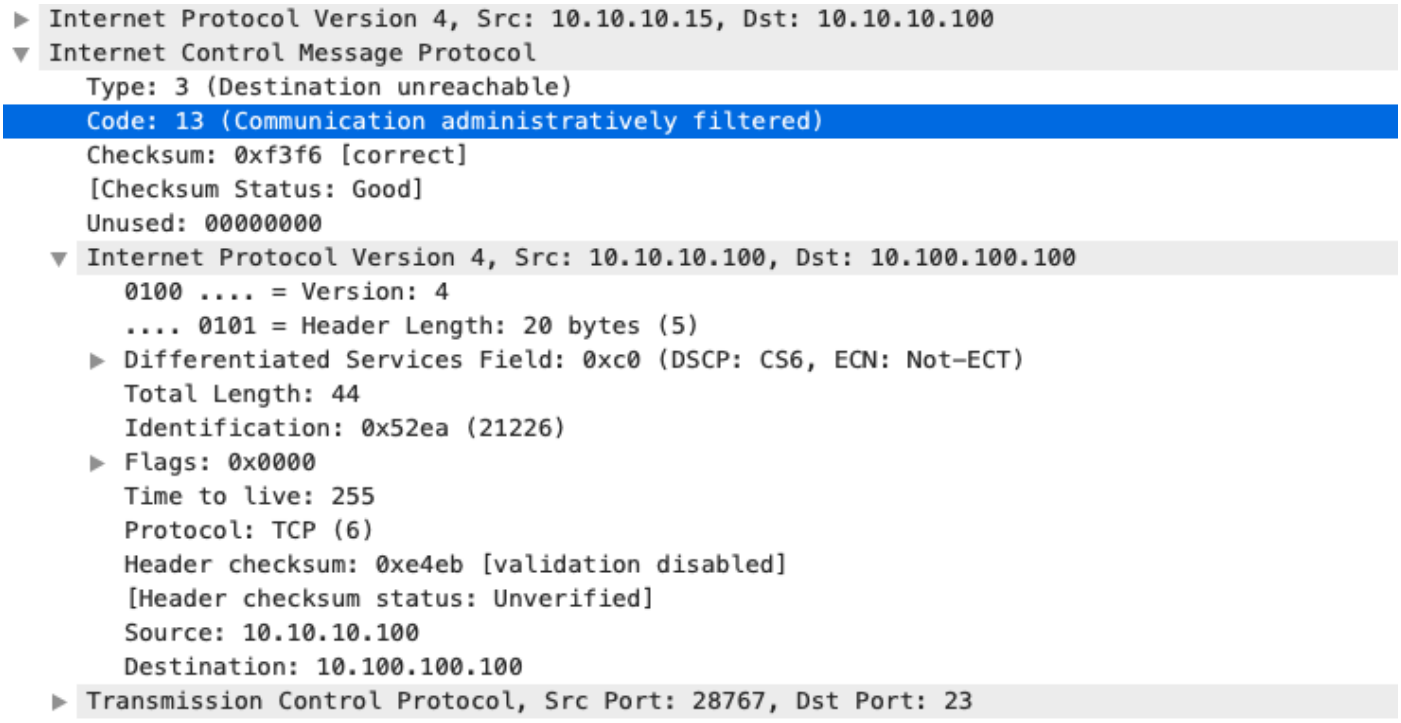
调试ICMP以查看ICMP目标无法到达的系统日志。

```
9300-Switch#debug ip icmp              <-- enables ICMP debugs
ICMP packet debugging is on
9300-Switch#show logging | include ICMP
<snip>
*Sep 29 14:01:29.041: ICMP: dst (10.100.100.100) administratively prohibited unreachable sent to
10.10.10.100    <-- packet blocked and ICMP message sent to client
```

注意：由于规模庞大，建议在启用ICMP调试之前禁用控制台日志记录和终端监控。

Catalyst 9300 CPU上的嵌入式数据包捕获显示CPU上Telnet连接的初始TCP SYN以及发送的ICMP目的地不可达。

ICMP目的地不可达数据包源自发往客户端的Catalyst 9300 VLAN 10接口，并包含ICMP数据包发送目的的原始数据包报头。



解决方案

在此场景中，禁用通过ACL阻止的传送数据包以生成ICMP目标不可达消息的行为。

在Catalyst 9000系列交换机上的路由接口上，默认启用IP不可达功能。

```
9300-Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
9300-Switch(config)#interface vlan 10
9300-Switch(config-if)#no ip unreachables      <-- disable IP unreachables
```

检验是否已为该接口禁用这些功能。

```
9300-Switch#show ip interface vlan 10
Vlan10 is up, line protocol is up
Internet address is 10.10.10.15/24
Broadcast address is 255.255.255.255
Address determined by setup command
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Multicast reserved groups joined: 224.0.0.102
Outgoing Common access list is not set
Outgoing access list is not set
Inbound Common access list is not set
Inbound access list is BLOCK-TELNET
Proxy ARP is disabled
Local Proxy ARP is disabled
Security level is default
Split horizon is enabled
ICMP redirects are never sent
ICMP unreachables are never sent     <-- IP unreachables disabled
ICMP mask replies are never sent
IP fast switching is enabled
IP Flow switching is disabled
```

```
IP CEF switching is enabled
IP CEF switching turbo vector
<snip>
```

场景3：超出ICMP TTL

请考虑前两个场景使用的先前拓扑。这次，位于10.10.10.100的用户尝试访问网络中已经停用的资源。因此，Catalyst 9300上不再存在用于托管此网络的SVI和VLAN。但是，路由器仍然具有指向Catalyst 9300 VLAN 10接口的静态路由，作为此网络的下一跳。

由于Catalyst 9300不再配置此网络，因此它不会显示为直接连接，而且9300会将它没有特定路由的任何数据包路由到其静态默认路由，该路由指向位于10.10.10.1的路由器。

当用户尝试连接到192.168.10.0/24地址空间中的资源时，此行为会在网络中引入路由环路。数据包在9300和路由器之间循环，直到TTL超时。

1. 用户尝试连接到192.168.10/24网络中的资源

2.数据包由Catalyst 9300接收，并路由至其下一跳为10.10.10.1的默认路由，然后将TTL递减1。

3.路由器收到此数据包并检查路由表，查找下一跳为10.10.10.15的此网络的路由。它将TTL递减1，并将数据包路由回9300。

4. Catalyst 9300接收该数据包，并再次将其路由回10.10.10.1并将TTL递减1。

此过程会重复，直到IP TTL达到零。

当Catalyst收到IP TTL = 1的数据包时，它会将该数据包传送到CPU并生成ICMP TTL-Exceeded消息。

ICMP数据包类型为11，代码为0（TTL在传输中过期）。无法通过CLI命令禁用此数据包类型

在此方案中，DHCP流量问题开始出现，因为环回的数据包会受到ICMP重定向，因为它们与接收它们的接口不同。

从用户发送的数据包也受到ICMP重定向的影响。在这种情况下，DHCP流量很容易从广播队列中耗尽。在规模上，由于重定向队列中传出的数据包数量较多，此场景会更糟。

在这里，CoPP丢弃通过向192.168.10.0/24网络发出1000次ping来演示，每次ping之间的超时为0秒。9300上的CoPP统计信息将被清除，并且在发送ping之前丢弃零字节。

```
9300-Switch#clear platform hardware fed switch active qos statistics internal cpu policer
<-- clear CoPP stats

9300-Switch#show platform hardware fed switch active qos queue stats internal cpu policer | i
```

```
Redirect|Drop   <-- verify 0 drops
QId PlcIdx Queue Name Enabled Rate Rate Drop(Bytes) Drop(Frames)
6 0 ICMP Redirect Yes 600 600 0               0                        <-- bytes dropped 0
<snip>
```

用户向远程网络发送流量。

```
User#ping 192.168.10.10 timeout 0 rep 1000   <-- User sends 1000 pings
Type escape sequence to abort.
Sending 1000, 100-byte ICMP Echos to 192.168.10.10, timeout is 0 seconds:
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................
Success rate is 0 percent (0/1000)
```

ICMP调试显示由于路由环路而导致的Redirect和TTL-Exceeded系统日志。

```
9300-Switch#debug ip icmp
ICMP packet deubgging is on
*Sep 29 16:33:22.676: ICMP: redirect sent to 10.10.10.100 for dest 192.168.10.10, use gw
10.10.10.1    <-- redirect sent
*Sep 29 16:33:22.678: ICMP: time exceeded (time to live) sent to 10.10.10.100 (dest was
192.168.10.10), topology BASE, dscp 0 topoid 0    <-- TTL exceeded observed
*Sep 29 16:33:22.678: ICMP: time exceeded (time to live) sent to 10.10.10.100 (dest was
192.168.10.10), topology BASE, dscp 0 topoid 0
*Sep 29 16:33:22.678: ICMP: time exceeded (time to live) sent to 10.10.10.100 (dest was
192.168.10.10), topology BASE, dscp 0 topoid 0
<snip>
```

注意：由于规模庞大，建议在启用ICMP调试之前禁用控制台日志记录和终端监控。

由于转发到CPU进行重定向的流量量导致出现CoPP丢弃。请注意，这仅适用于单个客户端。

```
9300-Switch#show platform hardware fed switch active qos queue stats internal cpu policer

CPU Queue Statistics
============================================================================================
(default) (set) Queue Queue
QId PlcIdx Queue Name Enabled Rate Rate Drop(Bytes) Drop(Frames)
--------------------------------------------------------------------------------------------
0 11 DOT1X Auth Yes 1000 1000 0 0
1 1 L2 Control Yes 2000 2000 0 0
2 14 Forus traffic Yes 4000 4000 0 0
3 0 ICMP GEN Yes 600 600 0 0
4 2 Routing Control Yes 5400 5400 0 0
5 14 Forus Address resolution Yes 4000 4000 0 0
6    0      ICMP Redirect               Yes     600       600      15407990     126295    <--
drops in redirect queue
```

```
7 16 Inter FED Traffic Yes 2000 2000 0 0
8 4 L2 LVX Cont Pack Yes 1000 1000 0 0
<snip>
```

解决方案

此场景中的解决方案是禁用ICMP重定向，与场景1中的相同。路由环路也是个问题，但强度会加剧，因为数据包也会被转发以进行重定向。

当TTL为1时，也会传送ICMP TTL-Exceeded数据包，但这些数据包使用不同的CoPP Policer索引，并且不与BROADCAST共享队列，因此不会影响DHCP流量。

只需使用no ip redirects CLI禁用ICMP重定向。

```
9300-Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
9300-Switch(config)#interface vlan 10
9300-Switch(config-if)#no ip redirects        <-- disable IP redirects
9300-Switch(config-if)#end
```

相关信息

- 配置嵌入式数据包捕获
- 了解ICMP重定向
- 技术支持和文档 - Cisco Systems

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。