

# 基于Cisco IOS软件的Catalyst 4500交换机的高CPU利用率

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[了解 Catalyst 4500 CPU 数据包处理体系结构](#)

[确定 Catalyst 4500 上 CPU 使用率较高的原因](#)

[确定 CPU 使用率的基线](#)

[了解 Catalyst 4500 交换机上的 show processes cpu 命令](#)

[了解 Catalyst 4500 交换机上的 show platform health 命令](#)

[解决常见的高 CPU 使用率问题](#)

[进程交换的数据包导致 CPU 使用率较高](#)

[CPU 使用率较高的其他原因](#)

[分析发往 CPU 的数据流的故障排除工具](#)

[工具 1：使用 SPAN 监控 CPU 数据流 - Cisco IOS 软件版本 12.1\(19\)EW 及更高版本](#)

[工具 2：内置的 CPU 嗅探器 - Cisco IOS 软件版本 12.2\(20\)EW 及更高版本](#)

[工具 3：确定将数据流发送到 CPU 的接口 - Cisco IOS 软件版本 12.2\(20\)EW 及更高版本](#)

[摘要](#)

[相关信息](#)

## 简介

对于计算密集型数据流，Catalyst 4500 系列交换机（包括 Catalyst 4948 交换机）提供了一种成熟的数据包处理方法。这些交换机上的一个常见问题是 CPU 使用率较高。本文档详细介绍了 CPU 数据包处理体系结构，并说明如何确定这些交换机上的 CPU 使用率较高的原因。本文档还列出了导致 Catalyst 4500 系列上 CPU 使用率较高的一些常见网络或配置案例。

**注意：** 如果运行基于 Catalyst OS (CatOS) 的 Catalyst 4500/4000 系列交换机，请参阅文档[运行 CatOS 软件的 Catalyst 4500/4000、2948G、2980G 和 4912G 交换机上的 CPU 使用率](#)。

## 先决条件

### 要求

本文档没有任何特定的要求。

## 使用的组件

本文档中的信息基于以下软件和硬件版本：

- Catalyst 4500 系列交换机
- Catalyst 4948 系列交换机

**注意：** 本文档仅适用于基于 Cisco IOS® 软件的交换机，而不适用于基于 CatOS 的交换机。

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。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

## 规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

## 背景信息

在查看 CPU 数据包处理体系结构和对 CPU 使用率较高进行排除故障之前，您必须了解基于硬件的转发交换机和基于 Cisco IOS 软件的路由器使用 CPU 的不同方式。一种常见的误解是，CPU 使用率较高意味着设备上的资源耗尽且可能会发生崩溃。容量问题是 Cisco IOS 路由器上的 CPU 使用率较高的症状之一。但是，容量问题几乎从未是基于硬件的转发交换机（如 Catalyst 4500）CPU 使用率较高的症状。Catalyst 4500 设计为在硬件专用集成电路 (ASIC) 上转发数据包，并且最高可达每秒 102 兆数据包 (Mpps) 的数据流转发速度。

Catalyst 4500 CPU 执行以下功能：

- 管理配置的软件协议，例如：生成树协议 (STP) 路由协议 Cisco 发现协议 (CDP) 端口聚合协议 (PAgP) VLAN 中继协议 (VTP) 动态中继协议 (DTP)
- 对硬件 ASIC 的配置/动态条目进行编程，例如：访问控制列表 (ACL) CEF 条目
- 在内部管理多种组件，例如：以太网供电 (PoE) 板卡电源风扇盘
- 管理对交换机的访问，例如：Telnet 控制台简单网络管理协议 (SNMP)
- 通过软件路径转发数据包，例如：互联网分组交换 (IPX) - 路由数据包（只在软件路径中受支持）最大传输单元 (MTU) 分段

根据此列表，CPU 使用率较高可能是由于 CPU 接收或处理数据包引起的。一些为了进行处理而发送的数据包对于网络运行可能至关重要。用于生成树拓扑配置的网桥协议数据单元 (BPDU) 就是这样一种重要的数据包。不过，其他数据包可能是软件转发的数据流。以下情况需要交换 ASIC 将数据包发送到 CPU 进行处理：

- 复制到 CPU 的数据包，但原始数据包在硬件中交换例如，主机 MAC 地址识别。
- 发送到 CPU 进行处理的数据包示例包括：路由协议更新 BPDU 有意或无意的数据流泛洪
- 发送到 CPU 进行转发的数据包例如，需要 IPX 或 AppleTalk 路由的数据包。

## 了解 Catalyst 4500 CPU 数据包处理体系结构

Catalyst 4500 具有一个内置的服务质量 (QoS) 机制，用于区分发往 CPU 的各种类型的数据流。该机制根据第 2 层 (L2)/第 3 层 (L3)/第 4 层 (L4) 数据包信息进行区分。Supervisor 数据包引擎有 16 个队列，用于处理各种类型的数据包或事件。[图 1](#) 显示了这些队列。[表 1](#) 列出了这些队列和每个队列中排队的数据包类型。这 16 个队列使 Catalyst 4500 可以根据数据包类型或优先级对数据包进行

排队。

图 1 - Catalyst 4500 使用多个 CPU 队列

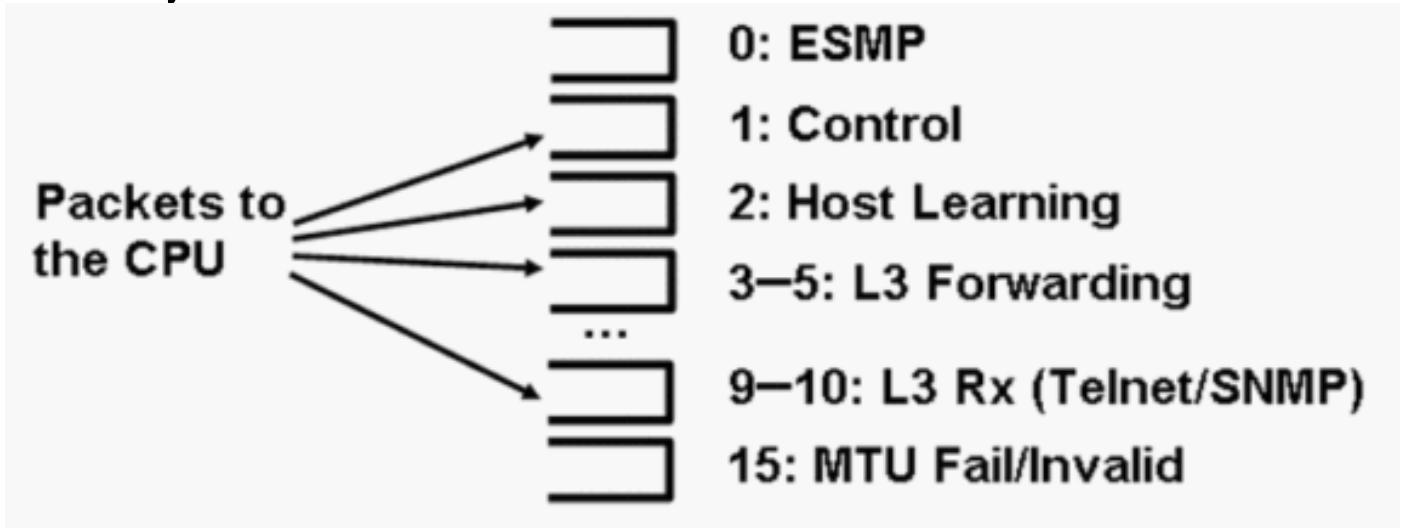


表 1 - Catalyst 4500 队列说明

队列编号	队列名称	排队的数据包
0	Esmp	用于板卡 ASIC 或其他组件管理的 ESMP1 数据包（内部管理数据包）
1		L2控制层面数据包，例如STP、CDP、Pagp、LACP <sup>2</sup> 或者UDLD <sup>3</sup>
2		包含未知源 MAC 地址的帧，这些帧将复制到 CPU 以构建 L2 转发表
3, 4, 5	L3 Fwd Highest, L3 Fwd High/Medium, L3 Fwd Low	在软件方面必须转发的数据包，例如GRE <sup>4</sup> 建立隧道，如果ARP <sup>5</sup> 为目的IP地址是未解决的，数据包发送对此队列。
6, 7, 8	L2 Fwd Highest, L2 Fwd High/Medium, L2 Fwd Low	<p>由于桥接而被转发的数据包</p> <ul style="list-style-type: none"> <li>• 硬件中不支持的协议（例如，IPX 和 AppleTalk）路由的数据包会被桥接到 CPU</li> <li>• ARP 请求和响应</li> <li>• 如果数据包在硬件方面不可能路由由于，数据包和交换机SVI<sup>6/L3</sup>接口的目标 MAC地址一起桥接：IP 报头选项已到期 TTL<sup>7</sup>非 ARPA 封装</li> </ul>

9, 10	L3 Rx High, L3 Rx Low	L3控制层面流量，例如，路由协议，为CPU IP地址示例是注定的包括Telnet、SNMP和SSH <sup>8</sup> 。
11	RPF Failure	失败RPF <sup>9</sup> 检查的组播信息包
12	ACL fwd(snoping)	由监听的DHCP <sup>10</sup> 处理的数据包，动态ARP检查或者IGMP <sup>11</sup> 监听的功能
13	ACL log, unreachable	押ACE <sup>12</sup> 用日志关键字或数据包丢弃的归结于在输出ACL的一拒绝或缺乏路由对目的地这些数据包的生成。
14	ACL sw processing	由于缺少其他 ACL 硬件资源（例如用于安全ACL的TCAM <sup>13</sup> ）而被传送到CPU的数据包
15	MTU Fail/Invalid	因为输出接口 MTU 大小小于数据包的大小而需要分段的数据包

<sup>1</sup> ESMP =简单管理协议。

<sup>2</sup> LACP =链路汇聚控制协议。

<sup>3</sup> UDLD =单向链路检测。

<sup>4</sup> GRE =通用路由封装。

<sup>5</sup> ARP =地址解析协议。

<sup>6</sup> SVI =交换虚拟接口。

<sup>7</sup> TTL =生存时间。

<sup>8</sup> SSH =安全套接协议。

<sup>9</sup> RPF =反向路径转发

<sup>10</sup> DHCP =动态主机配置协议。

<sup>11</sup> IGMP =互联网组管理协议。

<sup>12</sup> ACE =访问控制条目。

<sup>13</sup> TCAM =三个按内容寻址的存储器。

以下队列是独立的队列：

- L2 Fwd Highest 或 L3 Fwd Highest
- L2 Fwd High/Medium 或 L3 Fwd High/Medium

- L2 Fwd Low 或 L3 Fwd Low
- L3 Rx High 或 L3 Rx Low

数据包会根据 QoS 标签 ( 是 IP 服务类型 (ToS) 中的差分服务代码点 (DSCP) 值 ) 在这些队列中排队。例如, DSCP 值为 63 的数据包会排入 L3 Fwd Highest 队列中。可以在 **show platform cpu packet statistics all** 命令的输出中看到这 16 个队列所收到和丢弃的数据包。此命令的输出非常长。发出 **show platform cpu packet statistics** 命令可以只显示非零事件。替代命令是 **show platform cpuport** 命令。只有在运行 Cisco IOS 软件版本 12.1(11)EW 或更早版本时才使用 **show platform cpuport** 命令。已不推荐使用此命令。但是, 这一较旧命令在早于 Cisco IOS 软件版本 12.2(20)EWA 的版本中是 **show tech-support** 命令的一部分。

请将 **show platform cpu packet statistics** 命令用于所有故障排除。

```
Switch#show platform cpu packet statistics all
!--- Output suppressed. Total packet queues 16 Packets Received by Packet Queue Queue Total 5
sec avg 1 min avg 5 min avg 1 hour avg -----
----- Esmpl 0 0 0 0 Control 48 0 0 0 Host Learning 0 0 0 0 L3 Fwd High 0 0
0 0 L3 Fwd Medium 0 0 0 0 L3 Fwd Low 0 0 0 0 L2 Fwd High 0 0 0 0 L2 Fwd Medium 0 0 0 0
L2 Fwd Low 0 0 0 0 L3 Rx High 0 0 0 0 L3 Rx Low 0 0 0 0 RPF Failure 0 0 0 0 ACL
fwd(snooping) 0 0 0 0 ACL log, unreach 0 0 0 0 ACL sw processing 0 0 0 0 MTU Fail/Invalid
0 0 0 0 Packets Dropped by Packet Queue Queue Total 5 sec avg 1 min avg 5 min avg 1 hour avg -
----- Esmpl 0 0 0 0
Control 0 0 0 0 Host Learning 0 0 0 0 L3 Fwd High 0 0 0 0 L3 Fwd Medium 0 0 0 0 L3 Fwd
Low 0 0 0 0 L2 Fwd High 0 0 0 0 L2 Fwd Medium 0 0 0 0 L2 Fwd Low 0 0 0 0 L3 Rx High 0 0
0 0 L3 Rx Low 0 0 0 0 RPF Failure 0 0 0 0 ACL fwd(snooping) 0 0 0 0 ACL log, unreach 0 0
0 0 ACL sw processing 0 0 0 0 MTU Fail/Invalid 0 0 0 0
```

Catalyst 4500 CPU 为表 1 中列出的各个队列分配权重。CPU 根据重要性、类型、数据流优先级或 DSCP 分配权重。CPU 根据队列的相对权重为队列提供服务。例如, 如果控制数据包 ( 例如 BPDUs ) 和 ICMP 回声请求均挂起, 则 CPU 首先为控制数据包提供服务。过量的低优先级数据流或不太重要的数据流不会降低 CPU 处理或管理系统的功能。此机制保证了网络即使在 CPU 使用率较高的情况下也是稳定的。网络能否保持稳定是您必须了解的重要信息。

Catalyst 4500 CPU 数据包处理有另一个非常重要的实施细节。如果 CPU 已经为优先级较高的数据包或进程提供了服务, 但在一个特定时间段有较多空闲的 CPU 周期, 则 CPU 会为优先级较低的队列数据包提供服务或执行优先级较低的后台进程。因处理优先级较低的数据包或执行后台进程而导致 CPU 使用率较高被视为正常现象, 因为 CPU 会连续尝试使用所有可用时间。通过这种方式, CPU 努力在不削弱网络稳定性的情况下使交换机和网络达到最佳性能。除非单个时隙内的 CPU 使用率达到 100%, 否则 Catalyst 4500 会认为 CPU 未被充分利用。

Cisco IOS 软件版本 12.2(25)EWA2 及更高版本已增强了 CPU 数据包和进程处理机制和记帐功能。因此, 对于 Catalyst 4500 部署, 请使用这些版本。

## [确定 Catalyst 4500 上 CPU 使用率较高的原因](#)

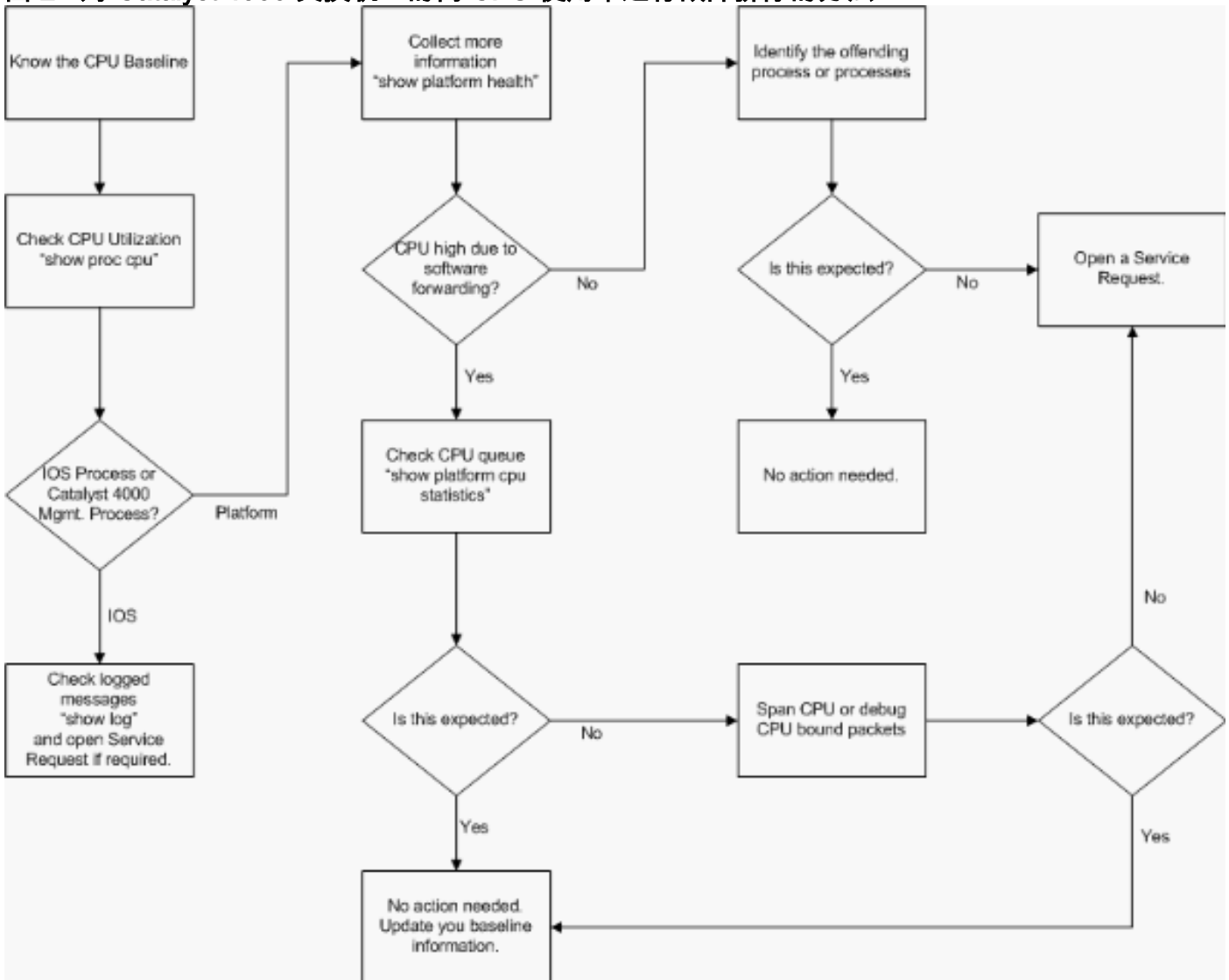
既然已了解 Catalyst 4500 CPU 数据包处理体系结构和设计, 您可能仍希望确定 Catalyst 4500 CPU 使用率较高的原因。Catalyst 4500 具有用于确定 CPU 使用率较高的根本原因的必要命令和工具。在确定了原因之后, 管理员可以执行以下任一操作:

- 更正操作 - 这可能包括更改配置或网络, 也可能包括创建用于进一步分析的 [Cisco 技术支持服务](#) 请求。
- 无操作 - Catalyst 4500 按照预期运行。由于 Supervisor 引擎最大限度地利用了 CPU 周期以执行所有必要的软件数据包转发和后台作业, 因此 CPU 会呈现较高的 CPU 使用率。

即使在某些情况下不必采取更正操作, 也应确保确定 CPU 使用率较高的原因。CPU 使用率较高可能只是网络中出现的问题的一个症状。为了降低 CPU 使用率, 有必要确定该问题的根本原因。

图 2 显示了用来确定 Catalyst 4500 CPU 使用率较高的根本原因的故障排除方法。

图 2 - 对 Catalyst 4500 交换机上的高 CPU 使用率进行故障排除的方法



一般故障排除步骤包括：

1. 发出 **show processes cpu** 命令以确定占用 CPU 周期的 Cisco IOS 进程。
2. 发出 **show platform health** 命令以进一步确定平台特定的进程。
3. 如果高度活跃的进程是 **K2CpuMan Review**，请发出 **show platform cpu packet statistics** 命令以确定发送到 CPU 的数据流类型。如果活动不是由于 **K2CpuMan Review** 进程引起的，请跳过步骤 4，继续执行步骤 5。
4. 通过使用[分析发往 CPU 的数据流的故障排除工具](#)来确定发送到 CPU 的数据包（如果必要）。使用的故障排除工具的一个示例是 CPU 交换端口分析程序 (SPAN)。
5. 请查看本文档和[解决常见的高 CPU 使用率问题](#)部分以了解常见原因。如果仍然无法确定根本原因，请与 [Cisco 技术支持](#)联系。

## 确定 CPU 使用率的基线

重要的第一步是了解在使用您的配置和网络设置时交换机的 CPU 使用率。使用 **show processes cpu** 命令以确定 Catalyst 4500 交换机上的 CPU 使用率。随着您将更多配置添加到网络设置中，以及网络数据流模式的更改，经常更新基线 CPU 使用率可能是必要的。图 2 指明了此要求。

此输出来自一台满载的 Catalyst 4507R。稳态 CPU 使用率大约是 32% 到 38%，这是执行此交换机的管理功能所必需的：

```
Switch#show processes cpu
CPU utilization for five seconds: 38%/1%; one minute: 32%; five minutes: 32%
PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min  TTY Process
  1         0         63         0  0.00%  0.00%  0.00%  0 Chunk Manager
  2        60       50074         1  0.00%  0.00%  0.00%  0 Load Meter
  3         0         1         0  0.00%  0.00%  0.00%  0 Deferred Events
!--- Output suppressed. 27 524 250268 2 0.00% 0.00% 0.00% 0 TTY Background 28 816 254843 3 0.00%
0.00% 0.00% 0 Per-Second Jobs 29 101100 5053 20007 0.00% 0.01% 0.00% 0 Per-minute Jobs 30
26057260 26720902          975 12.07% 11.41% 11.36% 0 Cat4k Mgmt HiPri
 31 19482908 29413060          662 24.07% 19.32% 19.20% 0 Cat4k Mgmt LoPri
 32   4468   162748         27  0.00%  0.00%  0.00%  0 Galios Reschedul
 33    0         1         0  0.00%  0.00%  0.00%  0 IOS ACL Helper
 34    0         2         0  0.00%  0.00%  0.00%  0 NAM Manager
```

五秒 CPU 使用率表示如下：

```
Switch#show processes cpu
CPU utilization for five seconds: 38%/1%; one minute: 32%; five minutes: 32%
PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min  TTY Process
  1         0         63         0  0.00%  0.00%  0.00%  0 Chunk Manager
  2        60       50074         1  0.00%  0.00%  0.00%  0 Load Meter
  3         0         1         0  0.00%  0.00%  0.00%  0 Deferred Events
!--- Output suppressed. 27 524 250268 2 0.00% 0.00% 0.00% 0 TTY Background 28 816 254843 3 0.00%
0.00% 0.00% 0 Per-Second Jobs 29 101100 5053 20007 0.00% 0.01% 0.00% 0 Per-minute Jobs 30
26057260 26720902          975 12.07% 11.41% 11.36% 0 Cat4k Mgmt HiPri
 31 19482908 29413060          662 24.07% 19.32% 19.20% 0 Cat4k Mgmt LoPri
 32   4468   162748         27  0.00%  0.00%  0.00%  0 Galios Reschedul
 33    0         1         0  0.00%  0.00%  0.00%  0 IOS ACL Helper
 34    0         2         0  0.00%  0.00%  0.00%  0 NAM Manager
```

x% 代表总 CPU 使用率，而 y% 代表在中断级别消耗的 CPU。排除 Catalyst 4500 交换机的故障时，请仅重点关注总 CPU 使用率。

## [了解 Catalyst 4500 交换机上的 show processes cpu 命令](#)

此 show processes cpu 输出显示有以下两个进程使用 CPU：Cat4k Mgmt HiPri 和 Cat4k Mgmt LoPri。这两个进程聚合了 Catalyst 4500 上执行重要管理功能的多个平台特定进程。这些进程处理控制层面以及需要软件交换或处理的数据包。

要查看哪些平台特定的进程在 Cat4k Mgmt HiPri 和 Cat4k Mgmt LoPri 的上下文中使用 CPU，请发出 show platform health 命令。

每一个平台特定的进程都有目标/预期的 CPU 使用率。该进程在目标之内时，CPU 会在优先级较高的上下文中执行该进程。show processes cpu 命令输出计算了运行 Cat4k Mgmt HiPri 时的该使用率。如果进程超出了目标/预期的使用率，则该进程会在优先级较低的上下文中运行。show processes cpu 命令输出计算了在执行 Cat4k Mgmt LoPri 时的该附加使用率。此 Cat4k Mgmt LoPri 还用于运行后台进程和优先级较低的其他进程，例如一致性检查和读接口计数器。此机制使 CPU 在必要时可以运行优先级较高的进程，而将剩余的空闲 CPU 周期用于优先级较低的进程。略微超过目标 CPU 使用率或者使用率出现瞬间峰值，并不表示存在需要调查的问题。

```
Switch#show platform health
          %CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
          Target Actual Target Actual     Fg   Bg  5Sec Min Hour   CPU
Lj-poll          1.00  0.02         2         1  100  500    0  0  0  1:09
GalChassisVp-review  3.00  0.29        10         3  100  500    0  0  0  11:15
S2w-JobEventSchedule 10.00  0.32        10         7  100  500    0  0  0  10:14
Stub-JobEventSchedul 10.00 12.09        10         6  100  500   14 13  9  396:35
```

StatValueMan Update	1.00	<b>0.22</b>	1	0	100	500	0	0	0	6:28
Pim-review	0.10	<b>0.00</b>	1	0	100	500	0	0	0	0:22
Ebm-host-review	1.00	<b>0.00</b>	8	0	100	500	0	0	0	0:05
Ebm-port-review	0.10	<b>0.00</b>	1	0	100	500	0	0	0	0:01
Protocol-aging-revie	0.20	<b>0.00</b>	2	0	100	500	0	0	0	0:00
Acl-Flattener e	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
KxAclPathMan create/	1.00	<b>0.00</b>	10	5	100	500	0	0	0	0:39
KxAclPathMan update	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
KxAclPathMan reprogr	1.00	<b>0.00</b>	2	0	100	500	0	0	0	0:00
TagMan-RecreateMtegR	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2CpuMan Review	30.00	<b>10.19</b>	30	28	100	500	14	13	9	397:11
K2AccelPacketMan: Tx	10.00	<b>2.20</b>	20	0	100	500	2	2	1	82:06
K2AccelPacketMan: Au	0.10	<b>0.00</b>	0	0	100	500	0	0	0	0:00
K2AclMan-taggedFlatA	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2AclCamMan stale en	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2AclCamMan hw stats	3.00	<b>1.04</b>	10	5	100	500	1	1	0	39:36
K2AclCamMan kx stats	1.00	<b>0.00</b>	10	5	100	500	0	0	0	13:40
K2AclCamMan Audit re	1.00	<b>0.00</b>	10	5	100	500	0	0	0	13:10
K2AclPolicerTableMan	1.00	<b>0.00</b>	10	1	100	500	0	0	0	0:38
K2L2 Address Table R	2.00	<b>0.00</b>	12	5	100	500	0	0	0	0:00
K2L2 New Static Addr	2.00	<b>0.00</b>	10	1	100	500	0	0	0	0:00
K2L2 New Multicast A	2.00	<b>0.00</b>	10	5	100	500	0	0	0	0:01
K2L2 Dynamic Address	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2L2 Vlan Table Revi	2.00	<b>0.00</b>	12	9	100	500	0	0	0	0:01
K2 L2 Destination Ca	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2PortMan Review	2.00	<b>0.72</b>	15	11	100	500	1	1	0	37:22
Gigaport65535 Review	0.40	<b>0.07</b>	4	2	100	500	0	0	0	3:38
Gigaport65535 Review	0.40	<b>0.08</b>	4	2	100	500	0	0	0	3:39
K2Fib cam usage revi	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib IrmFib Review	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib Vrf Default Ro	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib AdjRepop Revie	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib Vrf Unpunt Rev	2.00	<b>0.01</b>	15	0	100	500	0	0	0	0:23
K2Fib Consistency Ch	1.00	<b>0.00</b>	5	2	100	500	0	0	0	29:25
K2FibAdjMan Stats Re	2.00	<b>0.30</b>	10	4	100	500	0	0	0	6:21
K2FibAdjMan Host Mov	2.00	<b>0.00</b>	10	4	100	500	0	0	0	0:00
K2FibAdjMan Adj Chan	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2FibMulticast Signa	2.00	<b>0.01</b>	10	2	100	500	0	0	0	2:04
K2FibMulticast Entry	2.00	<b>0.00</b>	10	7	100	500	0	0	0	0:00
K2FibMulticast Irm M	2.00	<b>0.00</b>	10	7	100	500	0	0	0	0:00
K2FibFastDropMan Rev	2.00	<b>0.00</b>	7	0	100	500	0	0	0	0:00
K2FibPbr route map r	2.00	<b>0.06</b>	20	5	100	500	0	0	0	16:42
K2FibPbr flat acl pr	2.00	<b>0.07</b>	20	2	100	500	0	0	0	3:24
K2FibPbr consolidati	2.00	<b>0.01</b>	10	0	100	500	0	0	0	0:24
K2FibPerVlanPuntMan	2.00	<b>0.00</b>	15	4	100	500	0	0	0	0:00
K2FibFlowCache flow	2.00	<b>0.01</b>	10	0	100	500	0	0	0	0:23
K2FibFlowCache flow	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2FibFlowCache adj r	2.00	<b>0.01</b>	10	0	100	500	0	0	0	0:20
K2FibFlowCache flow	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:06
K2MetStatsMan Review	2.00	<b>0.14</b>	5	2	100	500	0	0	0	23:40
K2FibMulticast MET S	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2QosDblMan Rate DBL	2.00	<b>0.12</b>	7	0	100	500	0	0	0	4:52
IrmFibThrottler Thro	2.00	<b>0.01</b>	7	0	100	500	0	0	0	0:21
K2 VlanStatsMan Revi	2.00	<b>1.46</b>	15	7	100	500	2	2	1	64:44
K2 Packet Memory Dia	2.00	<b>0.00</b>	15	8	100	500	0	1	1	45:46
K2 L2 Aging Table Re	2.00	<b>0.12</b>	20	3	100	500	0	0	0	7:22
RkiosPortMan Port Re	2.00	<b>0.73</b>	12	7	100	500	1	1	1	52:36
Rkios Module State R	4.00	<b>0.02</b>	40	1	100	500	0	0	0	1:28
Rkios Online Diag Re	4.00	<b>0.02</b>	40	0	100	500	0	0	0	1:15
RkiosIpPbr IrmPort R	2.00	<b>0.02</b>	10	3	100	500	0	0	0	2:44
RkiosAclMan Review	3.00	<b>0.06</b>	30	0	100	500	0	0	0	2:35
MatMan Review	0.50	<b>0.00</b>	4	0	100	500	0	0	0	0:00
Slot 3 ILC Manager R	3.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
Slot 3 ILC S2wMan Re	3.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00



```

Slot 4 ILC Manager R      3.00  0.00   10    0 100  500    0  0  0  0:00
Slot 4 ILC S2wMan Re     3.00  0.00   10    0 100  500    0  0  0  0:00
Slot 5 ILC Manager R      3.00  0.00   10    0 100  500    0  0  0  0:00
Slot 5 ILC S2wMan Re     3.00  0.00   10    0 100  500    0  0  0  0:00
Slot 6 ILC Manager R      3.00  0.00   10    0 100  500    0  0  0  0:00
Slot 6 ILC S2wMan Re     3.00  0.00   10    0 100  500    0  0  0  0:00
Slot 7 ILC Manager R      3.00  0.00   10    0 100  500    0  0  0  0:00
Slot 7 ILC S2wMan Re     3.00  0.00   10    0 100  500    0  0  0  0:00
EthHoleLinecardMan(1    1.66  0.04   10    0 100  500    0  0  0  1:18
EthHoleLinecardMan(2    1.66  0.02   10    0 100  500    0  0  0  1:18
EthHoleLinecardMan(6    1.66  0.17   10    6 100  500    0  0  0  6:38
-----
%CPU Totals              212.80  35.63

```

## 了解 Catalyst 4500 交换机上的 show platform health 命令

**show platform health** 命令提供了许多仅对开发工程师十分有意义的信息。要对高 CPU 使用率进行故障排除，请在输出的 %CPU actual 列中查找一个较高的值。同时，请务必查看该行的右侧，以验证 1 分钟和 1 小时的 average %CPU 列中该进程的 CPU 使用情况。有时，进程会瞬间达到峰值，但不会长时间占用 CPU。一些 CPU 使用率瞬间较高的情况会在硬件编程或编程优化期间发生。例如，在对 TCAM 中的大型 ACL 进行硬件编程期间达到 CPU 使用率峰值是正常的。

在了解 Catalyst 4500 交换机上的 show processes cpu 命令部分的 show platform health 命令输出中，Stub-JobEventSchedul 和 K2CpuMan Review 进程使用的 CPU 周期的数量较高。表 2 提供了有关显示在 show platform health 命令的输出中常见的平台特定进程的一些基本信息。

表 2 - show platform health 命令输出中平台特定的进程的说明

平台特定的进程名	说明
Pim-review	机箱/板卡状态管理
Ebm	以太网网桥模块，例如老化和监控
Acl-Flattener / K2AclMan	ACL 合并进程
KxAclPathMan - TagMan-Review	ACL 状态管理和维护
K2CpuMan Review	执行软件信息包转发的进程，如果看到高 CPU 利用率由于此进程，调查押与使用的 CPU show platform cpu packet statistics 命令的数据包。
K2AccelPacketMan	为了发送来自 CPU 的数据包而与数据包引擎交互的驱动程序
K2AclCamMan	管理用于 QoS 和安全功能的输入和输出 TCAM 硬件
K2AclPolicerTableMan	管理输入和输出监察器
K2L2	代表这些进程对多种 L2 表的维护负责 Catalyst 4500 Cisco IOS 软件的 L2 转发子系统。
K2PortMan Review	管理各种与端口相关的编程功能
K2Fib	FIB1 管理
K2FibFlowC	PBR2 缓存管理

ache	
K2FibAdjMa n	FIB 邻接表管理
K2FibMulti cast	管理多播 FIB 条目
K2MetStats Man Review	管理 MET3 统计信息
K2QosDblMa n Review	管理 QoS DBL <sup>4</sup>
IrmFibThro ttler Thro	IP 路由模块
K2 L2 Aging Table Re	管理 L2 老化功能
GalChassis Vp-review	机箱状态监控
S2w- JobEventSc hedule	管理 S2W5 协议以监控板卡状态
Stub- JobEventSc hedul	基于残域 ASIC 的板卡监控和维护
RkiosPortM an Port Re	端口状态监控和维护
Rkios Module State R	板卡监控和维护
EthHoleLin ecardMan	管理每一个板卡中的 GBIC6

<sup>1</sup> FIB = 转发信息库。

<sup>2</sup> PBR = 基于策略的路由。

<sup>3</sup> 见面 = 组播扩展表。

<sup>4</sup> DBL = 动态缓冲区限制。

<sup>5</sup> S2W = 序列对电线。

<sup>6</sup> GBIC = 千兆位接口转换器。

## [解决常见的高 CPU 使用率问题](#)

本部分介绍了 Catalyst 4500 交换机上的一些常见的高 CPU 使用率问题。

### [进程交换的数据包导致 CPU 使用率较高](#)

CPU 使用率较高的常见原因之一是 Catalyst 4500 CPU 忙于处理软件转发的数据包或控制数据包。软件转发的数据包的示例是 IPX 或控制数据包，例如 BPDU。通常会有少量的这些数据包发送到 CPU。但是，如果总是有大量数据包发送到 CPU，则可能意味着出现了配置错误或网络事件。必须确定导致数据包转发到 CPU 进行处理的事件的原因。通过确定原因，可以针对高 CPU 使用率问题进行调试。

进程交换的数据包导致 CPU 使用率较高的一些常见原因包括：

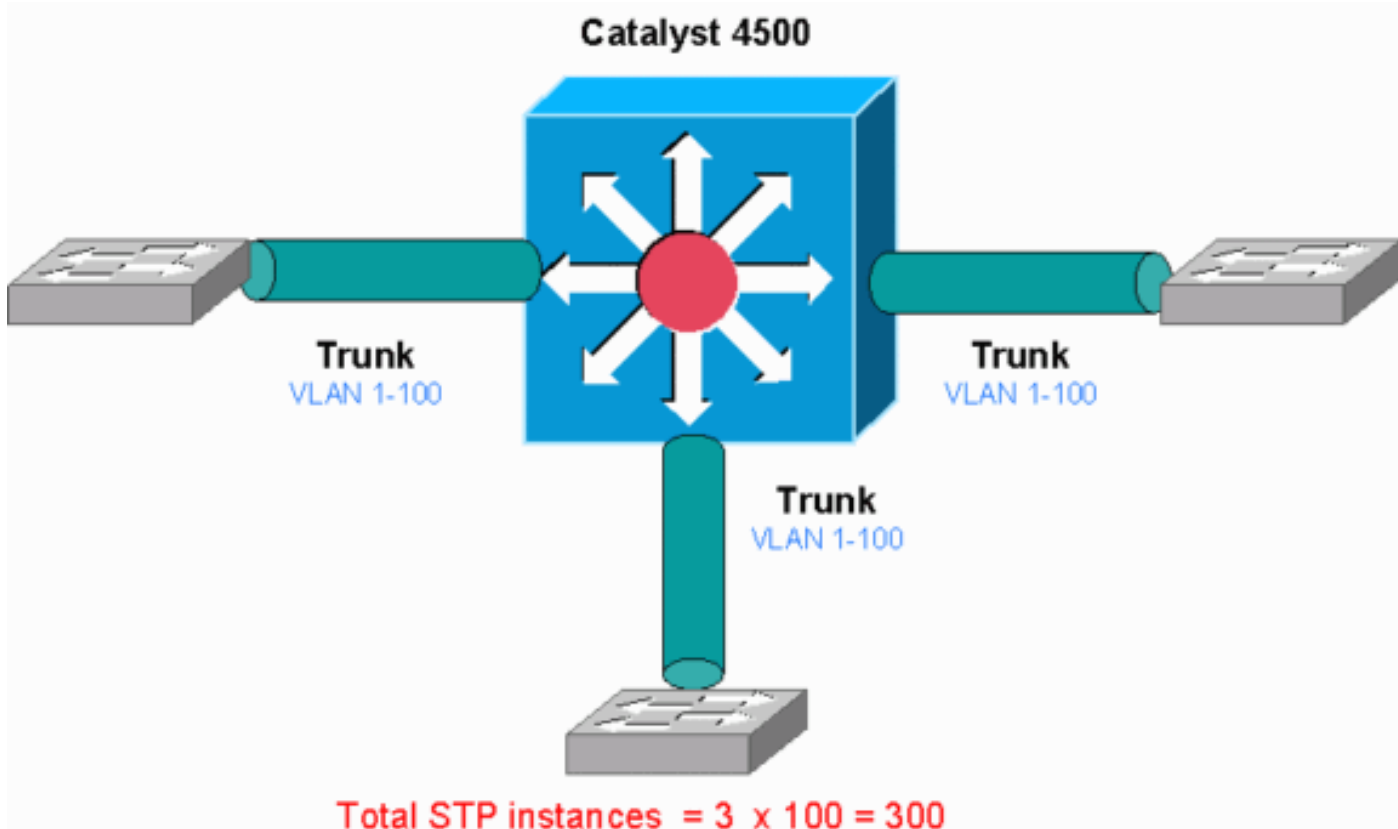
- [大量的生成树端口实例](#)
- [ICMP 重定向；在同一接口上路由数据包](#)
- [IPX 或 AppleTalk 路由](#)
- [主机识别](#)
- [缺少用于安全 ACL 的硬件资源 \(TCAM\)](#)
- [ACL 中的 log 关键字](#)
- [第 2 层转发环路](#)

将数据包交换到 CPU 的其他原因包括：

- MTU 分段 - 确保沿数据包路径的所有接口都具有相同的 MTU。
- ACL 带有 **established** 以外的 TCP 标志
- IP 版本 6 (IPv6) 路由 - 只有在使用软件交换路径时才支持。
- GRE — 这通过软件交换路径仅支持。
- 在输入或输出路由器 ACL (RACL) 中拒绝数据流 **注意**：这在 Cisco IOS 软件版本 12.1(13)EW1 及更高版本中是受到速率限制的。请在 ACL 的接口下发出 **no ip unreachable** 命令。
- 由于直接连接了大量主机而导致过多的 ARP 和 DHCP 数据流发送至 CPU 进行处理如果怀疑存在 DHCP 攻击，可使用 DCHP 监听对来自任何特定主机端口的 DHCP 数据流进行速率限制。
- 合法或行为方式不正确的终端站过多地执行 SNMP 轮询

### [大量的生成树端口实例](#)

Catalyst 4500 在每 VLAN 生成树 + (PVST+) 模式下支持 3000 个生成树端口实例或活动端口。除了 Supervisor 引擎 II+、II+TS 和 Catalyst 4948 之外的所有 Supervisor 引擎上都提供该支持。Supervisor 引擎 II+、II+TS 和 Catalyst 4948 最多可支持 1500 个端口实例。如果超出了这些建议的 STP 实例数，则交换机将显示 CPU 使用率较高。



此图显示了具有三个中继端口的 Catalyst 4500，每个端口承载 1 到 100 个 VLAN。这相当于 300 个生成树端口实例。一般来说，可以使用以下公式计算生成树端口实例：

## Switch#show platform health

	%CPU		RunTimeMax		Priority		Average %CPU			Total CPU
	Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	
Lj-poll	1.00	<b>0.02</b>	2	1	100	500	0	0	0	1:09
GalChassisVp-review	3.00	<b>0.29</b>	10	3	100	500	0	0	0	11:15
S2w-JobEventSchedule	10.00	<b>0.32</b>	10	7	100	500	0	0	0	10:14
Stub-JobEventSchedul	10.00	<b>12.09</b>	10	6	100	500	14	<b>13</b>	<b>9</b>	396:35
StatValueMan Update	1.00	<b>0.22</b>	1	0	100	500	0	0	0	6:28
Pim-review	0.10	<b>0.00</b>	1	0	100	500	0	0	0	0:22
Ebm-host-review	1.00	<b>0.00</b>	8	0	100	500	0	0	0	0:05
Ebm-port-review	0.10	<b>0.00</b>	1	0	100	500	0	0	0	0:01
Protocol-aging-revie	0.20	<b>0.00</b>	2	0	100	500	0	0	0	0:00
Acl-Flattener e	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
KxAclPathMan create/	1.00	<b>0.00</b>	10	5	100	500	0	0	0	0:39
KxAclPathMan update	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
KxAclPathMan reprogr	1.00	<b>0.00</b>	2	0	100	500	0	0	0	0:00
TagMan-RecreateMtegR	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2CpuMan Review	30.00	<b>10.19</b>	30	28	100	500	14	<b>13</b>	<b>9</b>	397:11
K2AccelPacketMan: Tx	10.00	<b>2.20</b>	20	0	100	500	2	<b>2</b>	<b>1</b>	82:06
K2AccelPacketMan: Au	0.10	<b>0.00</b>	0	0	100	500	0	0	0	0:00
K2AclMan-taggedFlatA	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2AclCamMan stale en	1.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2AclCamMan hw stats	3.00	<b>1.04</b>	10	5	100	500	1	<b>1</b>	<b>0</b>	39:36
K2AclCamMan kx stats	1.00	<b>0.00</b>	10	5	100	500	0	0	0	13:40
K2AclCamMan Audit re	1.00	<b>0.00</b>	10	5	100	500	0	0	0	13:10
K2AclPolicerTableMan	1.00	<b>0.00</b>	10	1	100	500	0	0	0	0:38
K2L2 Address Table R	2.00	<b>0.00</b>	12	5	100	500	0	0	0	0:00
K2L2 New Static Addr	2.00	<b>0.00</b>	10	1	100	500	0	0	0	0:00
K2L2 New Multicast A	2.00	<b>0.00</b>	10	5	100	500	0	0	0	0:01
K2L2 Dynamic Address	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2L2 Vlan Table Revi	2.00	<b>0.00</b>	12	9	100	500	0	0	0	0:01
K2 L2 Destination Ca	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2PortMan Review	2.00	<b>0.72</b>	15	11	100	500	1	<b>1</b>	<b>0</b>	37:22
Gigaport65535 Review	0.40	<b>0.07</b>	4	2	100	500	0	0	0	3:38
Gigaport65535 Review	0.40	<b>0.08</b>	4	2	100	500	0	0	0	3:39
K2Fib cam usage revi	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib IrmFib Review	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib Vrf Default Ro	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib AdjRepop Revie	2.00	<b>0.00</b>	15	0	100	500	0	0	0	0:00
K2Fib Vrf Unpunt Rev	2.00	<b>0.01</b>	15	0	100	500	0	0	0	0:23
K2Fib Consistency Ch	1.00	<b>0.00</b>	5	2	100	500	0	0	0	29:25
K2FibAdjMan Stats Re	2.00	<b>0.30</b>	10	4	100	500	0	0	0	6:21
K2FibAdjMan Host Mov	2.00	<b>0.00</b>	10	4	100	500	0	0	0	0:00
K2FibAdjMan Adj Chan	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2FibMulticast Signa	2.00	<b>0.01</b>	10	2	100	500	0	0	0	2:04
K2FibMulticast Entry	2.00	<b>0.00</b>	10	7	100	500	0	0	0	0:00
K2FibMulticast Irm M	2.00	<b>0.00</b>	10	7	100	500	0	0	0	0:00
K2FibFastDropMan Rev	2.00	<b>0.00</b>	7	0	100	500	0	0	0	0:00
K2FibPbr route map r	2.00	<b>0.06</b>	20	5	100	500	0	0	0	16:42
K2FibPbr flat acl pr	2.00	<b>0.07</b>	20	2	100	500	0	0	0	3:24
K2FibPbr consolidati	2.00	<b>0.01</b>	10	0	100	500	0	0	0	0:24
K2FibPerVlanPuntMan	2.00	<b>0.00</b>	15	4	100	500	0	0	0	0:00
K2FibFlowCache flow	2.00	<b>0.01</b>	10	0	100	500	0	0	0	0:23
K2FibFlowCache flow	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2FibFlowCache adj r	2.00	<b>0.01</b>	10	0	100	500	0	0	0	0:20
K2FibFlowCache flow	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:06
K2MetStatsMan Review	2.00	<b>0.14</b>	5	2	100	500	0	0	0	23:40
K2FibMulticast MET S	2.00	<b>0.00</b>	10	0	100	500	0	0	0	0:00
K2QosDb1Man Rate DBL	2.00	<b>0.12</b>	7	0	100	500	0	0	0	4:52
IrmFibThrottler Thro	2.00	<b>0.01</b>	7	0	100	500	0	0	0	0:21
K2 VlanStatsMan Revi	2.00	<b>1.46</b>	15	7	100	500	2	<b>2</b>	<b>1</b>	64:44
K2 Packet Memory Dia	2.00	<b>0.00</b>	15	8	100	500	0	<b>1</b>	<b>1</b>	45:46
K2 L2 Aging Table Re	2.00	<b>0.12</b>	20	3	100	500	0	0	0	7:22
RkiosPortMan Port Re	2.00	<b>0.73</b>	12	7	100	500	1	<b>1</b>	<b>1</b>	52:36

Rkios Module State R	4.00	0.02	40	1	100	500	0	0	0	1:28
Rkios Online Diag Re	4.00	0.02	40	0	100	500	0	0	0	1:15
RkiosIpPbr IrmPort R	2.00	0.02	10	3	100	500	0	0	0	2:44
RkiosAclMan Review	3.00	0.06	30	0	100	500	0	0	0	2:35
MatMan Review	0.50	0.00	4	0	100	500	0	0	0	0:00
Slot 3 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 3 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 4 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 4 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 5 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 5 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 6 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 6 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 7 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 7 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
EthHoleLinecardMan(1	1.66	0.04	10	0	100	500	0	0	0	1:18
EthHoleLinecardMan(2	1.66	0.02	10	0	100	500	0	0	0	1:18
EthHoleLinecardMan(6	1.66	0.17	10	6	100	500	0	0	0	6:38

-----  
 %CPU Totals            212.80   35.63

在上图中，没有接入端口，但三个中继承载 1 到 100 个 VLAN：

Switch#show platform health

	%CPU	%CPU	RunTimeMax	Priority	Average	%CPU	Total			
	Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	CPU
Lj-poll	1.00	0.02	2	1	100	500	0	0	0	1:09
GalChassisVp-review	3.00	0.29	10	3	100	500	0	0	0	11:15
S2w-JobEventSchedule	10.00	0.32	10	7	100	500	0	0	0	10:14
Stub-JobEventSchedul	10.00	12.09	10	6	100	500	14	13	9	396:35
StatValueMan Update	1.00	0.22	1	0	100	500	0	0	0	6:28
Pim-review	0.10	0.00	1	0	100	500	0	0	0	0:22
Ebm-host-review	1.00	0.00	8	0	100	500	0	0	0	0:05
Ebm-port-review	0.10	0.00	1	0	100	500	0	0	0	0:01
Protocol-aging-revie	0.20	0.00	2	0	100	500	0	0	0	0:00
Acl-Flattener e	1.00	0.00	10	0	100	500	0	0	0	0:00
KxAclPathMan create/	1.00	0.00	10	5	100	500	0	0	0	0:39
KxAclPathMan update	2.00	0.00	10	0	100	500	0	0	0	0:00
KxAclPathMan reprogr	1.00	0.00	2	0	100	500	0	0	0	0:00
TagMan-RecreateMtegR	1.00	0.00	10	0	100	500	0	0	0	0:00
K2CpuMan Review	30.00	10.19	30	28	100	500	14	13	9	397:11
K2AccelPacketMan: Tx	10.00	2.20	20	0	100	500	2	2	1	82:06
K2AccelPacketMan: Au	0.10	0.00	0	0	100	500	0	0	0	0:00
K2AclMan-taggedFlatA	1.00	0.00	10	0	100	500	0	0	0	0:00
K2AclCamMan stale en	1.00	0.00	10	0	100	500	0	0	0	0:00
K2AclCamMan hw stats	3.00	1.04	10	5	100	500	1	1	0	39:36
K2AclCamMan kx stats	1.00	0.00	10	5	100	500	0	0	0	13:40
K2AclCamMan Audit re	1.00	0.00	10	5	100	500	0	0	0	13:10
K2AclPolicerTableMan	1.00	0.00	10	1	100	500	0	0	0	0:38
K2L2 Address Table R	2.00	0.00	12	5	100	500	0	0	0	0:00
K2L2 New Static Addr	2.00	0.00	10	1	100	500	0	0	0	0:00
K2L2 New Multicast A	2.00	0.00	10	5	100	500	0	0	0	0:01
K2L2 Dynamic Address	2.00	0.00	10	0	100	500	0	0	0	0:00
K2L2 Vlan Table Revi	2.00	0.00	12	9	100	500	0	0	0	0:01
K2 L2 Destination Ca	2.00	0.00	10	0	100	500	0	0	0	0:00
K2PortMan Review	2.00	0.72	15	11	100	500	1	1	0	37:22
Gigaport65535 Review	0.40	0.07	4	2	100	500	0	0	0	3:38
Gigaport65535 Review	0.40	0.08	4	2	100	500	0	0	0	3:39
K2Fib cam usage revi	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib IrmFib Review	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib Vrf Default Ro	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib AdjRepop Revie	2.00	0.00	15	0	100	500	0	0	0	0:00
K2Fib Vrf Unpunt Rev	2.00	0.01	15	0	100	500	0	0	0	0:23
K2Fib Consistency Ch	1.00	0.00	5	2	100	500	0	0	0	29:25

K2FibAdjMan Stats Re	2.00	0.30	10	4	100	500	0	0	0	6:21
K2FibAdjMan Host Mov	2.00	0.00	10	4	100	500	0	0	0	0:00
K2FibAdjMan Adj Chan	2.00	0.00	10	0	100	500	0	0	0	0:00
K2FibMulticast Signa	2.00	0.01	10	2	100	500	0	0	0	2:04
K2FibMulticast Entry	2.00	0.00	10	7	100	500	0	0	0	0:00
K2FibMulticast Irm M	2.00	0.00	10	7	100	500	0	0	0	0:00
K2FibFastDropMan Rev	2.00	0.00	7	0	100	500	0	0	0	0:00
K2FibPbr route map r	2.00	0.06	20	5	100	500	0	0	0	16:42
K2FibPbr flat acl pr	2.00	0.07	20	2	100	500	0	0	0	3:24
K2FibPbr consolidati	2.00	0.01	10	0	100	500	0	0	0	0:24
K2FibPerVlanPuntMan	2.00	0.00	15	4	100	500	0	0	0	0:00
K2FibFlowCache flow	2.00	0.01	10	0	100	500	0	0	0	0:23
K2FibFlowCache flow	2.00	0.00	10	0	100	500	0	0	0	0:00
K2FibFlowCache adj r	2.00	0.01	10	0	100	500	0	0	0	0:20
K2FibFlowCache flow	2.00	0.00	10	0	100	500	0	0	0	0:06
K2MetStatsMan Review	2.00	0.14	5	2	100	500	0	0	0	23:40
K2FibMulticast MET S	2.00	0.00	10	0	100	500	0	0	0	0:00
K2QosDblMan Rate DBL	2.00	0.12	7	0	100	500	0	0	0	4:52
IrmFibThrottler Thro	2.00	0.01	7	0	100	500	0	0	0	0:21
K2 VlanStatsMan Revi	2.00	1.46	15	7	100	500	2	2	1	64:44
K2 Packet Memory Dia	2.00	0.00	15	8	100	500	0	1	1	45:46
K2 L2 Aging Table Re	2.00	0.12	20	3	100	500	0	0	0	7:22
RkiosPortMan Port Re	2.00	0.73	12	7	100	500	1	1	1	52:36
Rkios Module State R	4.00	0.02	40	1	100	500	0	0	0	1:28
Rkios Online Diag Re	4.00	0.02	40	0	100	500	0	0	0	1:15
RkiosIpPbr IrmPort R	2.00	0.02	10	3	100	500	0	0	0	2:44
RkiosAclMan Review	3.00	0.06	30	0	100	500	0	0	0	2:35
MatMan Review	0.50	0.00	4	0	100	500	0	0	0	0:00
Slot 3 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 3 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 4 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 4 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 5 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 5 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 6 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 6 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 7 ILC Manager R	3.00	0.00	10	0	100	500	0	0	0	0:00
Slot 7 ILC S2wMan Re	3.00	0.00	10	0	100	500	0	0	0	0:00
EthHoleLinecardMan(1	1.66	0.04	10	0	100	500	0	0	0	1:18
EthHoleLinecardMan(2	1.66	0.02	10	0	100	500	0	0	0	1:18
EthHoleLinecardMan(6	1.66	0.17	10	6	100	500	0	0	0	6:38
-----										
%CPU Totals	212.80	35.63								

## 步骤 1：使用 show processes cpu 命令检查 Cisco IOS 进程。

本部分介绍了管理员在缩小高 CPU 使用率问题范围时使用的命令。如果发出 **show processes cpu** 命令，则可以看到两个主进程（Cat4k Mgmt LoPri 和 Spanning Tree）对 CPU 的使用率较高。仅通过此信息，您就可以知道生成树进程占用了很大一部分的 CPU 周期。

```
Switch#show processes cpu
CPU utilization for five seconds: 74%/1%; one minute: 73%; five minutes: 50%
  PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min TTY Process
    1         4         198        20 0.00% 0.00% 0.00% 0 Chunk Manager
    2         4         290        13 0.00% 0.00% 0.00% 0 Load Meter
!--- Output suppressed. 25 488 33 14787 0.00% 0.02% 0.00% 0 Per-minute Jobs 26 90656 223674 405
6.79% 6.90% 7.22% 0 Cat4k Mgmt HiPri 27 158796 59219 2681 32.55% 33.80% 21.43%
0 Cat4k Mgmt LoPri
 28         20        1693        11 0.00% 0.00% 0.00% 0 Galios Reschedul
 29         0         1         0 0.00% 0.00% 0.00% 0 IOS ACL Helper
 30         0         2         0 0.00% 0.00% 0.00% 0 NAM Manager
!--- Output suppressed. 41 0 1 0 0.00% 0.00% 0.00% 0 SFF8472 42 0 2 0 0.00% 0.00% 0.00% 0 AAA
Dictionary R 43 78564 20723 3791 32.63% 30.03% 17.35% 0 Spanning Tree
```

```

44          112          999          112 0.00% 0.00% 0.00% 0 DTP Protocol
45           0          147           0 0.00% 0.00% 0.00% 0 Ethchnl

```

## 步骤 2：使用 show platform health 命令检查 Catalyst 4500 特定的进程。

要了解哪个平台特定的进程占用了 CPU，请发出 **show platform health** 命令。从该输出中，可以看到 **K2CpuMan Review** 进程（处理计算密集型数据包的作业）用尽了 CPU：

```

Switch#show platform health
%CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
      Target Actual Target Actual   Fg   Bg 5Sec Min Hour   CPU
!--- Output suppressed. TagMan-RecreateMtegr 1.00 0.00 10 0 100 500 0 0 0 0:00 K2CpuMan Review
30.00 37.62    30    53 100 500   41 33    1 2:12
K2AccelPacketMan: Tx 10.00 4.95    20    0 100 500    5 4    0 0:36
K2AccelPacketMan: Au  0.10 0.00    0    0 100 500    0 0    0 0:00
K2AclMan-taggedFlatA 1.00 0.00    10    0 100 500    0 0    0 0:00

```

## 步骤 3：检查接收数据流的 CPU 队列以确定计算密集型数据流的类型。

发出 **show platform cpu packet statistics** 命令以检查哪一个 CPU 队列收到了计算密集型数据包。本部分中的输出显示 control 队列收到大量数据包。请使用表 1 中的信息和步骤 1 中做出的结论。可以确定 CPU 处理的数据包，并可以确定 CPU 使用率较高的原因是 BPDU 处理。

```

Switch#show platform cpu packet statistics
!--- Output suppressed. Total packet queues 16 Packets Received by Packet Queue Queue Total 5
sec avg 1 min avg 5 min avg 1 hour avg -----
- - - - - Esmp 202760 196 173 128 28 Control 388623
2121      1740      598      16

```

Packets Dropped by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
Control	17918	0	19	24	3

## 步骤 4：确定根本原因。

发出 **show spanning-tree summary** 命令。可以检查是否是由于存在大量的生成树端口实例而收到 BPDU。该输出清楚地确定了根本原因：

```

Switch#show spanning-tree summary
Switch is in pvst mode
Root bridge for: none
Extended system ID          is enabled
Portfast Default            is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default           is disabled
EtherChannel misconfig guard is enabled
UplinkFast                  is disabled
BackboneFast                 is disabled
Configured Pathcost method used is short
!--- Output suppressed. Name Blocking Listening Learning Forwarding STP Active -----
----- 2994 vlans 0
0      0      5999      5999

```

有大量的 VLAN 使用 PVST+ 模式配置。要解决该问题，请将 STP 模式更改为多生成树 (MST)。在某些情况下，由于在所有中继端口上转发大量的 VLAN，因此 STP 实例数很大。在这种情况下，请手动修剪并非中继所必需的 VLAN，以将 STP 活动端口数降低到远远低于建议值。

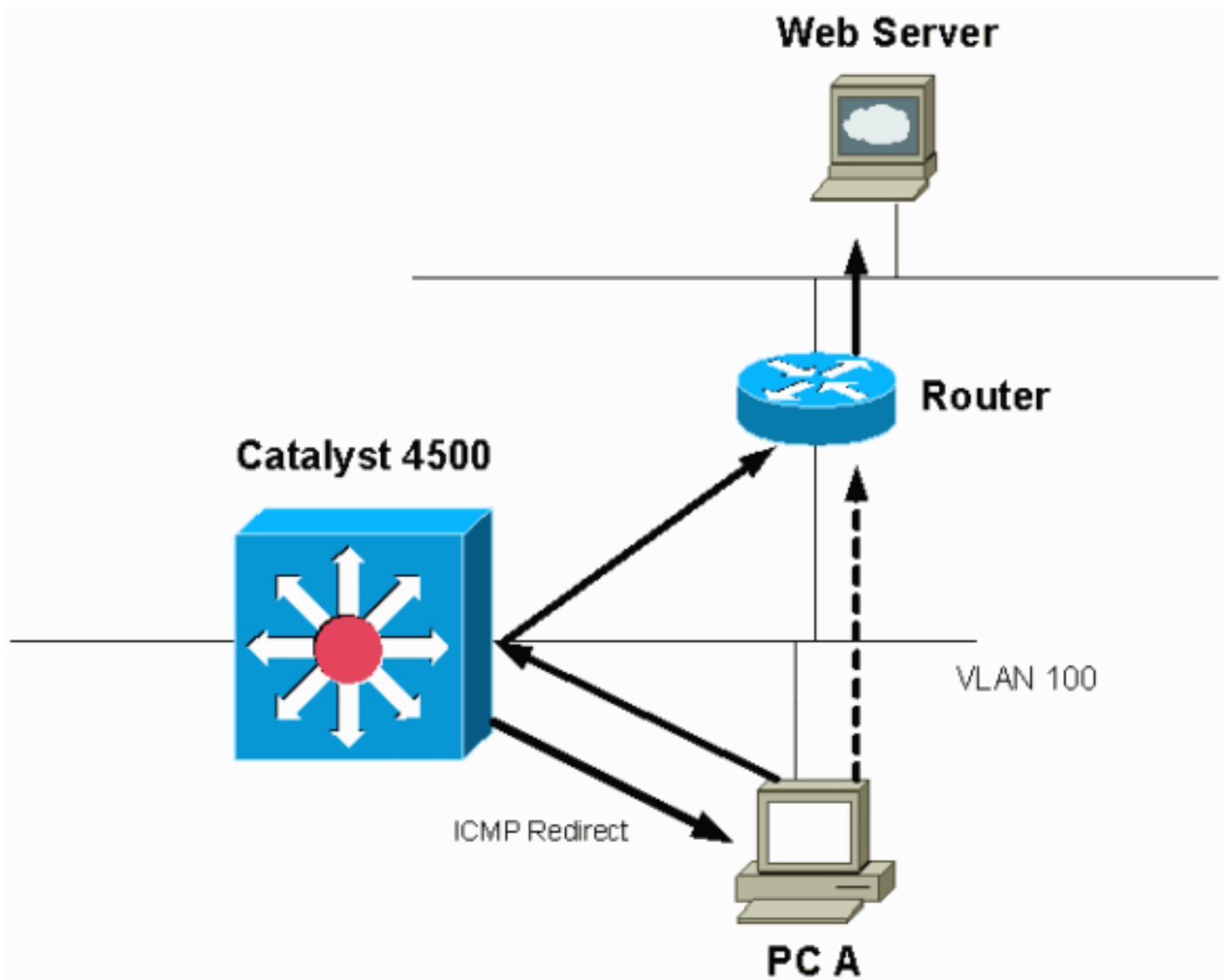
**提示：** 请确保不要将 IP 电话端口配置为中继端口。这是一个常见的错误配置。请使用语音 VLAN

配置来配置 IP 电话端口。此配置创建一个伪中继，但不需要您手工修剪不必要的 VLAN。有关如何配置语音端口的详细信息，请参阅[配置语音接口](#)软件配置指南。非 Cisco IP 电话不支持此语音 VLAN 或辅助 VLAN 配置。必须手动修剪连接了非 Cisco IP 电话的端口。

### [ICMP 重定向；在同一接口上路由数据包](#)

在同一接口上路由数据包或者数据流在同一 L3 接口上进出，可能会导致交换机生成 ICMP 重定向消息。如果交换机知道到最终目标的下一跳设备与发送设备在同一子网中，则交换机将生成发往源的 ICMP 重定向消息。这些重定向消息指示源将数据包直接发送到下一跳设备。这些消息指示下一跳设备具有到目标的更好的路由（比此交换机少一跳的路由）。

在本部分的图中，PC A 与 web 服务器通信。PC A 的默认网关指向 VLAN 100 接口 IP 地址。但是，使 Catalyst 4500 可以到达目标的下一跳路由器与 PC A 在同一子网中。这种情况下的最佳路径是直接发送到 Router。Catalyst 4500 将 ICMP 重定向消息发送到 PC A。该消息指示 PC A 通过路由器（而不是通过 Catalyst 4500）发送目标为 web 服务器的数据包。但是，在许多情况下，终端设备并不对 ICMP 重定向消息做出响应。缺少响应，会导致 Catalyst 4500 耗费大量 CPU 周期为 Catalyst 通过与输入数据包相同的接口转发的所有数据包生成这些 ICMP 重定向消息。



默认情况下已启用 ICMP 重定向。要禁用它，请使用 `no ip icmp redirects` 命令。请在相关的 SVI 或 L3 接口下发出该命令。

**注意：** 因为 `ip icmp redirects` 是默认命令，所以它在 `show running-configuration` 命令输出中是不



可见的。

### 步骤 1：使用 show processes cpu 命令检查 Cisco IOS 进程。

发出 **show processes cpu** 命令。可以看到两个主进程 ( **Cat4k Mgmt LoPri** 和 **IP Input** ) 对 CPU 的使用率较高。仅通过此信息，您就可以知道处理 IP 数据包占用了很大一部分的 CPU。

```
Switch#show processes cpu
CPU utilization for five seconds: 38%/1%; one minute: 32%; five minutes: 32%
PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min  TTY Process
   1         0         63         0  0.00%  0.00%  0.00%  0 Chunk Manager
   2        60       50074         1  0.00%  0.00%  0.00%  0 Load Meter
   3         0         1         0  0.00%  0.00%  0.00%  0 Deferred Events
!--- Output suppressed. 27 524 250268 2 0.00% 0.00% 0.00% 0 TTY Background 28 816 254843 3 0.00%
0.00% 0.00% 0 Per-Second Jobs 29 101100 5053 20007 0.00% 0.01% 0.00% 0 Per-minute Jobs 30
26057260 26720902 975 5.81% 6.78% 5.76% 0 Cat4k Mgmt HiPri 31 19482908 29413060 662
19.64% 18.20% 20.48% 0 Cat4k Mgmt LoPri
!--- Output suppressed. 35 60 902 0 0.00% 0.00% 0.00% 0 DHCP Snooping 36 504625304 645491491
781 72.40% 72.63% 73.82% 0 IP Input
```

### 步骤 2：使用 show platform health 命令检查 Catalyst 4500 特定的进程。

**show platform health** 命令的输出可以证实，使用 CPU 是为了处理计算密集型数据包。

```
Switch#show platform health
%CPU %CPU RunTimeMax Priority Average %CPU Total
      Target Actual Target Actual Fg Bg 5Sec Min Hour CPU
--- Output suppressed. TagMan-RecreateMtegR 1.00 0.00 10 0 100 500 0 0 0 0:00 K2CpuMan Review
330.00 19.18 150 79 25 500 20 19 18 5794:08 K2AccelPacketMan: Tx 10.00 4.95 20 0 100 500 5 4 0
0:36 K2AccelPacketMan: Au 0.10 0.00 0 0 100 500 0 0 0 0:00 K2AclMan-taggedFlatA 1.00 0.00 10 0
100 500 0 0 0 0:00
```

### 步骤 3：检查接收数据流的 CPU 队列以确定计算密集型数据流的类型。

发出 **show platform cpu packet statistics** 命令以检查哪一个 CPU 队列收到了计算密集型数据包。可以看到 L3 Fwd Low 队列收到了很多的数据流。

```
Switch#show platform cpu packet statistics
!--- Output suppressed. Packets Received by Packet Queue Queue Total 5 sec avg 1 min avg 5 min
avg 1 hour avg -----
Esmp 48613268 38 39 38 39 Control 142166648 74 74 73 73 Host Learning 1845568 2 2 2 2 L3 Fwd
High 17 0 0 0 0 L3 Fwd Medium 2626 0 0 0 0 L3 Fwd Low 4717094264 3841
3879 3873 3547
L2 Fwd Medium 1 0 0 0
L3 Rx High 257147 0 0 0
L3 Rx Low 5325772 10 19 13 7
RPF Failure 155 0 0 0
ACL fwd(snooping) 65604591 53 54 54 53
ACL log, unreach 11013420 9 8 8 8
```

### 步骤 4：确定根本原因。

在这种情况下，请使用 CPU SPAN 以确定发送到 CPU 的数据流。有关 CPU SPAN 的信息，请参阅本文档的 [工具 1：使用 SPAN 监控 CPU 数据流 - Cisco IOS 软件版本 12.1\(19\)EW 及更高版本](#) 部分。使用 **show running-configuration** 命令完成数据流分析和配置。在这种情况下，数据包通过同一接口路由，这将导致为每个数据包均发出 ICMP 重定向消息。此根本原因是 Catalyst 4500 上的 CPU 使用率较高的常见原因之一。

您可能希望源设备能够响应 Catalyst 4500 发送的 ICMP 重定向消息并更改目标的下一跳。但是

，不是所有设备都会对 ICMP 重定向消息做出响应。如果设备不做出响应，则 Catalyst 4500 必须为交换机从发送设备收到的每个数据包发送重定向消息。这些重定向消息会占用大量 CPU 资源。解决方案是禁用 ICMP 重定向。在接口下，发出 **no ip redirects** 命令。

如果还配置了辅助 IP 地址，可能会发生此情况。启用了辅助 IP 地址后，会自动禁用 IP 重定向。请务必不要手动启用 IP 重定向。

如此 [ICMP 重定向；在同一接口上路由数据包](#) 部分所示，大多数终端设备不会对 ICMP 重定向消息做出响应。因此，作为通常的做法，请禁用此功能。

## [IPX 或 AppleTalk 路由](#)

Catalyst 4500 只支持通过软件转发路径的 IPX 和 Appletalk 路由。使用此类协议的配置，CPU 使用率较高是正常的。

**注意：**同一 VLAN 中的 IPX 和 AppleTalk 数据流交换不需要进程交换。只有需要路由的数据包才需要软件路径转发。

### [步骤 1：使用 show processes cpu 命令检查 Cisco IOS 进程。](#)

发出 **show processes cpu** 命令以检查哪一个 Cisco IOS 进程占用了 CPU。在此命令输出中，请注意 CPU 使用率最高的进程是 **Cat4k Mgmt LoPri**：

```
witch#show processes cpu
CPU utilization for five seconds: 87%/10%; one minute: 86%; five minutes: 87%
  PID Runtime(ms)   Invoked      uSecs    5Sec    1Min    5Min TTY Process
    1         4         53         75  0.00%  0.00%  0.00%  0 Chunk Manager
!--- Output suppressed. 25 8008 1329154 6 0.00% 0.00% 0.00% 0 Per-Second Jobs 26 413128 38493
10732 0.00% 0.02% 0.00% 0 Per-minute Jobs 27 148288424 354390017 418 2.60% 2.42% 2.77% 0 Cat4k
Mgmt HiPri 28    285796820 720618753          396 50.15% 59.72% 61.31%  0 Cat4k Mgmt LoPri
```

### [步骤 2：使用 show platform health 命令检查 Catalyst 4500 特定的进程。](#)

**show platform health** 命令的输出可以证实，使用 CPU 是为了处理计算密集型数据包。

```
Switch#show platform health
          %CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
          Target Actual Target Actual   Fg   Bg 5Sec Min Hour   CPU
!--- Output suppressed. TagMan-RecreateMtegR 1.00 0.00 10 4 100 500 0 0 0 0:00 K2CpuMan Review
30.00 27.39    30    53 100 500  42 47  42 4841:
K2AccelPacketMan: Tx 10.00  8.03    20    0 100 500  21 29  26 270:4
```

### [步骤 3：检查接收数据流的 CPU 队列以确定计算密集型数据流的类型。](#)

要确定发送到 CPU 的数据流类型，请发出 **show platform cpu packet statistics** 命令。

```
Switch#show platform cpu packet statistics
!--- Output suppressed. Packets Received by Packet Queue Queue Total 5 sec avg 1 min avg 5 min
avg 1 hour avg -----
Esmpl 48613268 38 39 38 39 Control 142166648 74 74 73 73 Host Learning 1845568 2 2 2 2 L3 Fwd
High 17 0 0 0 0 L3 Fwd Medium 2626 0 0 0 0 L3 Fwd Low 1582414 1 1 1 1 L2 Fwd Medium 1 0 0 0 L2
Fwd Low
576905398 1837 1697 1938 1515
L3 Rx High 257147 0 0 0 0
L3 Rx Low 5325772 10 19 13 7
RPF Failure 155 0 0 0 0
ACL fwd(snooping) 65604591 53 54 54 53
```

## 步骤 4：确定根本原因。

因为管理员已配置了 IPX 或 AppleTalk 路由，所以确定根本原因应该十分简单。但为了确认，请对 CPU 数据流执行 SPAN 并确保您所看到的数据流是预期的数据流。有关 CPU SPAN 的信息，请参阅本文档的[工具 1：使用 SPAN 监控 CPU 数据流 - Cisco IOS 软件版本 12.1\(19\)EW 及更高版本部分](#)。

在这种情况下，管理员必须将基线 CPU 更新到当前值。Catalyst 4500 CPU 处理软件交换的数据包时，该 CPU 会按预期运行。

## 主机识别

如果 MAC 地址已经不在 MAC 地址表中，则 Catalyst 4500 会识别各种主机的 MAC 地址。交换引擎将带有新 MAC 地址的数据包副本转发到 CPU。

所有 VLAN 接口（第 3 层）将基于机箱的硬件地址用作它们的 MAC 地址。因此，在 MAC 地址表中没有一个条目，发往这些 VLAN 接口的数据包不会被发送到 CPU 进行处理。

如果有过多的新 MAC 地址需要交换机进行识别，则会导致 CPU 使用率较高。

## 步骤 1：使用 show processes cpu 命令检查 Cisco IOS 进程。

发出 **show processes cpu** 命令以检查哪一个 Cisco IOS 进程占用了 CPU。在此命令输出中，请注意 CPU 使用率最高的进程是 **Cat4k Mgmt LoPri**：

```
Switch#show processes cpu
CPU utilization for five seconds: 89%/1%; one minute: 74%; five minutes: 71%
  PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min  TTY Process
    1         4         53         75  0.00%  0.00%  0.00%  0 Chunk Manager
!--- Output suppressed. 25 8008 1329154 6 0.00% 0.00% 0.00% 0 Per-Second Jobs 26 413128 38493
10732 0.00% 0.02% 0.00% 0 Per-minute Jobs 27 148288424 354390017 418 26.47% 10.28% 10.11% 0
Cat4k Mgmt HiPri 28   285796820 720618753           396 52.71% 56.79% 55.70% 0 Cat4k Mgmt LoPri
```

## 步骤 2：使用 show platform health 命令检查 Catalyst 4500 特定的进程。

**show platform health** 命令的输出可以证实，使用 CPU 是为了处理计算密集型数据包。

```
Switch#show platform health
          %CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
          Target Actual Target Actual    Fg    Bg 5Sec Min Hour CPU
!--- Output suppressed. TagMan-RecreateMtegR 1.00 0.00 10 4 100 500 0 0 0 0:00 K2CpuMan Review
30.00 46.88    30    47 100 500    30 29    21 265:01
K2AccelPacketMan: Tx 10.00 8.03    20    0 100 500    21 29    26 270:4
```

## 步骤 3：检查接收数据流的 CPU 队列以确定计算密集型数据流的类型。

要确定发送到 CPU 的数据流类型，请发出 **show platform cpu packet statistics** 命令。

```
Switch#show platform cpu packet statistics
!--- Output suppressed. Packets Received by Packet Queue Queue Total 5 sec avg 1 min avg 5 min
avg 1 hour avg -----
Esmpp 48613268 38 39 38 39 Control 142166648 74 74 73 73 Host Learning 1845568
1328 1808 1393 1309
L3 Fwd High 17 0 0 0 0
```

L3 Fwd Medium	2626	0	0	0	0
L3 Fwd Low	1582414	1	1	1	1
L2 Fwd Medium	1	0	0	0	0
L2 Fwd Low	576905398	37	7	8	5
L3 Rx High	257147	0	0	0	0
L3 Rx Low	5325772	10	19	13	7
RPF Failure	155	0	0	0	0
ACL fwd(snooping)	65604591	53	54	54	53
ACL log, unreach	11013420	9	8	8	8

#### 步骤 4：确定根本原因。

**show platform health** 命令的输出显示 CPU 发现很多新 MAC 地址。这种情况通常是由于网络拓扑不稳定而导致的。例如，如果生成树拓扑更改，则交换机会生成拓扑更改通知 (TCN)。发出 TCN 将使 PVST+ 模式下的老化时间减少到 15 秒。如果 MAC 地址在此时间段内未被识别，则这些 MAC 地址条目会被清除。对于快速 STP (RSTP) (IEEE 802.1w) 或 MST (IEEE 802.1s) 而言，如果 TCN 来自另一台交换机，则条目会立即老化。此老化导致需要重新识别 MAC 地址。如果拓扑更改很少，则这不是一个大问题。但可能会由于链路抖动、交换机出现故障或无法为 PortFast 启用主机端口而导致过量的拓扑更改。大量的 MAC 表会刷新，并会导致随后进行重新识别。确定根本原因的下一步是排除网络故障。交换机按照预期工作，并将数据包发送到 CPU 以进行主机地址识别。确定并修复导致过多 TCN 的有故障设备。

您的网络可能具有许多突然发送大量数据流的设备，这将引起 MAC 地址在交换机上老化并随后进行重新识别。在这种情况下，可延长 MAC 地址表老化时间以避免频繁老化 MAC 地址并随后进行重新识别。使用一个较长的老化时间，交换机会在执行老化操作之前将设备 MAC 地址在表中保留较长的一段时间。

**警告：** 请在仔细考虑之后执行此老化更改。如果在您的网络中有移动设备，则此更改可能会导致数据流黑洞。

#### 缺少用于安全 ACL 的硬件资源 (TCAM)

Catalyst 4500 使用 Cisco TCAM 对配置的 ACL 进行编程。TCAM 允许将 ACL 应用到硬件转发路径中。在转发路径中是否使用 ACL 对交换机的性能没有任何影响。不管 ACL 的大小如何，性能都不变，因为 ACL 查找的性能在于线路速率。但是，TCAM 是一种有限的资源。因此，如果配置了过量的 ACL 条目，则会超出 TCAM 容量。[表 3](#) 显示了每一台 Catalyst 4500 Supervisor 引擎和交换机上的可用 TCAM 资源数。

**表 3 - Catalyst 4500 Supervisor 引擎/交换机上的 TCAM 容量**

产品	功能 TCAM (每个方向)	QoS TCAM (每个方向)
Supervisor 引擎 II+/II+TS	具有 1024 个掩码的 8192 个条目	具有 1024 个掩码的 8192 个条目
Supervisor 引擎 III/IV/V 和 Catalyst 4948	具有 2048 个掩码的 16,384 个条目	具有 2048 个掩码的 16,384 个条目
Supervisor 引擎 V-10GE 和 Catalyst 4948-10GE	具有 16,384 个掩码的 16,384 个条目	具有 16,384 个掩码的 16,384 个条目

交换机使用 TCAM 功能对安全 ACL (例如 RACL 和 VLAN ACL (VACL)) 进行编程。交换机还将 TCAM 功能用于安全功能，如用于动态 ACL 的 IP 源防护 (IPSG)。交换机使用 QoS TCAM 对分类

和监察器 ACL 进行编程。

如果 Catalyst 4500 在执行安全 ACL 编程期间耗尽了 TCAM 资源，则会通过软件路径来部分应用 ACL。与这些 ACE 匹配的数据包在软件中进行处理，这会导致 CPU 使用率较高。ACL 是自上而下进行编程的。换句话说，如果 ACL 无法完整地放入 TCAM 中，则 ACL 底部部分的 ACE 很可能未在 TCAM 中进行编程。

发生 TCAM 溢出时，会显示以下警告消息：

```
Switch#show platform cpu packet statistics
!--- Output suppressed. Packets Received by Packet Queue Queue Total 5 sec avg 1 min avg 5 min
avg 1 hour avg -----
Esmpr 48613268 38 39 38 39 Control 142166648 74 74 73 73 Host Learning 1845568
1328 1808 1393 1309
L3 Fwd High 17 0 0 0
L3 Fwd Medium 2626 0 0 0
L3 Fwd Low 1582414 1 1 1 1
L2 Fwd Medium 1 0 0 0
L2 Fwd Low 576905398 37 7 8 5
L3 Rx High 257147 0 0 0
L3 Rx Low 5325772 10 19 13 7
RPF Failure 155 0 0 0
ACL fwd(snooping) 65604591 53 54 54 53
ACL log, unreach 11013420 9 8 8 8
```

可以在 `show logging` 命令输出中看到此错误消息。该消息确定性地指出将进行一些软件处理并因此可能会导致 CPU 使用率较高。

**注意：** 如果更改了一个大型的 ACL，那么，在 TCAM 中对更改的 ACL 再次编程之前，您可能会短暂地看到此消息。

### 步骤 1：使用 show processes cpu 命令检查 Cisco IOS 进程。

发出 `show processes cpu` 命令。可以看到 CPU 使用率较高，因为 `Cat4k Mgmt LoPri` 进程占用了大部分 CPU 周期。

```
Switch#show processes cpu
CPU utilization for five seconds: 99%/0%; one minute: 99%; five minutes: 99%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
1 0 11 0 0.00% 0.00% 0.00% 0 Chunk Manager
2 9716 632814 15 0.00% 0.00% 0.00% 0 Load Meter
3 780 302 2582 0.00% 0.00% 0.00% 0 SpanTree Helper
!--- Output suppressed. 23 18208 3154201 5 0.00% 0.00% 0.00% 0 TTY Background 24 37208 3942818 9
0.00% 0.00% 0.00% 0 Per-Second Jobs 25 1046448 110711 9452 0.00% 0.03% 0.00% 0 Per-minute Jobs
26 175803612 339500656 517 4.12% 4.31% 4.48% 0 Cat4k Mgmt HiPri 27 835809548 339138782
2464 86.81% 89.20% 89.76% 0 Cat4k Mgmt LoPri
28 28668 2058810 13 0.00% 0.00% 0.00% 0 Galios Reschedul
```

### 步骤 2：使用 show platform health 命令检查 Catalyst 4500 特定的进程。

发出 `show platform health` 命令。可以看到 `K2CpuMan Review` ( 处理计算密集型数据包的作业 ) 使用了 CPU。

```
Switch#show platform health
%CPU %CPU RunTimeMax Priority Average %CPU Total
Target Actual Target Actual Fg Bg 5Sec Min Hour CPU
Lj-poll 1.00 0.01 2 0 100 500 0 0 0 13:45
GalChassisVp-review 3.00 0.20 10 16 100 500 0 0 0 88:44
S2w-JobEventSchedule 10.00 0.57 10 7 100 500 1 0 0 404:22
```

```

Stub-JobEventSchedul 10.00 0.00 10 0 100 500 0 0 0 0:00
StatValueMan Update 1.00 0.09 1 0 100 500 0 0 0 91:33
Pim-review 0.10 0.00 1 0 100 500 0 0 0 4:46
Ebm-host-review 1.00 0.00 8 4 100 500 0 0 0 14:01
Ebm-port-review 0.10 0.00 1 0 100 500 0 0 0 0:20
Protocol-aging-revie 0.20 0.00 2 0 100 500 0 0 0 0:01
Acl-Flattener 1.00 0.00 10 5 100 500 0 0 0 0:04
KxAclPathMan create/ 1.00 0.00 10 5 100 500 0 0 0 0:21
KxAclPathMan update 2.00 0.00 10 6 100 500 0 0 0 0:05
KxAclPathMan reprogr 1.00 0.00 2 1 100 500 0 0 0 0:00
TagMan-InformMtegRev 1.00 0.00 5 0 100 500 0 0 0 0:00
TagMan-RecreateMtegR 1.00 0.00 10 14 100 500 0 0 0 0:18
K2CpuMan Review 30.00 91.31 30 92 100 500 128 119 84 13039:02
K2AccelPacketMan: Tx 10.00 2.30 20 0 100 500 2 2 2 1345:30
K2AccelPacketMan: Au 0.10 0.00 0 0 100 500 0 0 0 0:00

```

### 步骤 3：检查接收数据流的 CPU 队列以确定计算密集型数据流的类型。

需要进一步了解是哪一个 CPU 队列以及哪一个类型的数据流发送到该 CPU 队列。发出 **show platform cpu packet statistics** 命令。可以看到 ACL sw processing 队列收到了大量的数据包。因此，TCAM 溢出是此高 CPU 使用率问题的原因。

```

Switch#show platform cpu packet statistics
!--- Output suppressed. Packets Received by Packet Queue Queue Total 5 sec avg 1 min avg 5 min
avg 1 hour avg -----
Control 57902635 22 16 12 3 Host Learning 464678 0 0 0 0 L3 Fwd Low 623229 0 0 0 0 L2 Fwd Low
11267182 7 4 6 1 L3 Rx High 508 0 0 0 0 L3 Rx Low 1275695 10 1 0 0 ACL fwd(snooping) 2645752 0 0
0 0 ACL log, unreach 51443268 9 4 5 5 ACL sw processing 842889240 1453 1532
1267 1179

```

Packets Dropped by Packet Queue

Queue	Total	5 sec avg	1 min avg	5 min avg	1 hour avg
L2 Fwd Low	3270	0	0	0	0
ACL sw processing	12636	0	0	0	0

### 步骤 4：解决这个问题。

在步骤 3 中，已确定了发生这种情况的根本原因。删除导致溢出的 ACL 或最大限度地减小 ACL 以避免溢出。此外，请查看[使用 ACL 配置网络安全](#)配置指南以优化硬件中的 ACL 配置和编程。

### ACL 中的 log 关键字

Catalyst 4500 支持记录与特定 ACL 条目相匹配的数据包详细信息，但是过多的日志记录可能导致 CPU 使用率较高。除了在数据流发现阶段，请避免使用 **log** 关键字。在数据流发现阶段，您将确定流经未明确配置 ACE 的网络的数据流。请不要使用 **log** 关键字来收集统计信息。在 Cisco IOS 软件版本 12.1(13)EW 及更高版本中，日志消息受到速率限制。如果使用日志消息对与 ACL 匹配的数据包数量进行计数，则该计数是不准确的。请改用 **show access-list** 命令以获得准确的统计信息。由于通过查看配置或日志消息可以了解 ACL 日志记录功能的使用情况，因此，可以较容易地确定此根本原因。

### 步骤 1：使用 show processes cpu 命令检查 Cisco IOS 进程。

发出 **show processes cpu** 命令以检查哪一个 Cisco IOS 进程占用了 CPU。在此命令输出中，可发现 CPU 使用率最高的进程是 **Cat4k Mgmt LoPri**：

```
Switch#show processes cpu
```

```

CPU utilization for five seconds: 99%/0%; one minute: 99%; five minutes: 99%
PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min  TTY Process
   1         0         11         0  0.00%  0.00%  0.00%   0 Chunk Manager
   2       9716      632814      15  0.00%  0.00%  0.00%   0 Load Meter
!--- Output suppressed. 26 175803612 339500656 517 4.12% 4.31% 4.48% 0 Cat4k Mgmt HiPri   27
835809548 339138782      2464 86.81% 89.20% 89.76%   0 Cat4k Mgmt LoPri
   28      28668   2058810      13  0.00%  0.00%  0.00%   0 Galios Reschedul

```

## 步骤 2：使用 show platform health 命令检查 Catalyst 4500 特定的进程。

检查使用 CPU 的平台特定进程。发出 **show platform health** 命令。在输出中，请注意 **K2CpuMan Review process** 使用了大部分 CPU 周期。此活动表明 CPU 在处理发往它的数据包时正忙。

```

Switch#show platform health
          %CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
          Target Actual Target Actual   Fg   Bg 5Sec Min Hour CPU
Lj-poll          1.00  0.01         2         0  100  500   0  0  0 13:45
GalChassisVp-review  3.00  0.20        10         16  100  500   0  0  0 88:44
S2w-JobEventSchedule 10.00  0.57        10          7  100  500   1  0  0 404:22
Stub-JobEventSchedul 10.00  0.00        10          0  100  500   0  0  0 0:00
StatValueMan Update  1.00  0.09         1          0  100  500   0  0  0 91:33
Pim-review       0.10  0.00         1          0  100  500   0  0  0  4:46
Ebm-host-review  1.00  0.00         8          4  100  500   0  0  0 14:01
Ebm-port-review  0.10  0.00         1          0  100  500   0  0  0  0:20
Protocol-aging-revie 0.20  0.00         2          0  100  500   0  0  0  0:01
Acl-Flattener    1.00  0.00        10          5  100  500   0  0  0  0:04
KxAclPathMan create/ 1.00  0.00        10          5  100  500   0  0  0  0:21
KxAclPathMan update  2.00  0.00        10          6  100  500   0  0  0  0:05
KxAclPathMan reprogr 1.00  0.00         2          1  100  500   0  0  0  0:00
TagMan-InformMtegRev 1.00  0.00         5          0  100  500   0  0  0  0:00
TagMan-RecreateMtegR 1.00  0.00        10         14  100  500   0  0  0  0:18
K2CpuMan Review    30.00  91.31    30    92 100 500 128 119 84 13039:02
K2AccelPacketMan: Tx 10.00  2.30        20          0  100  500   2  2  2 1345:30
K2AccelPacketMan: Au  0.10  0.00         0          0  100  500   0  0  0  0:00

```

## 步骤 3：检查接收数据流的 CPU 队列以确定计算密集型数据流的类型。

要确定发送到 CPU 的数据流类型，请发出 **show platform cpu packet statistics** 命令。在此命令的输出中，可以看到收到数据包是由于 ACL log 关键字导致的：

```

Switch#show platform cpu packet statistics
!--- Output suppressed. Total packet queues 16 Packets Received by Packet Queue Queue Total 5
sec avg 1 min avg 5 min avg 1 hour avg -----
- ----- Control 1198701435 35 35 34 35 Host Learning 874391 0 0 0 0 L3 Fwd High
428 0 0 0 0 L3 Fwd Medium 12745 0 0 0 0 L3 Fwd Low 2420401 0 0 0 0 L2 Fwd High 26855 0 0 0 0 L2
Fwd Medium 116587 0 0 0 0 L2 Fwd Low 317829151 53 41 31 31 L3 Rx High 2371 0 0 0 0 L3 Rx Low
32333361 7 1 2 0 RPF Failure 4127 0 0 0 0 ACL fwd (snooping) 107743299 4 4 4 4 ACL log, unreach
1209056404    1987    2125    2139    2089

```

Packets Dropped by Packet Queue

```

Queue          Total          5 sec avg 1 min avg 5 min avg 1 hour avg
-----
ACL log, unreach 193094788      509      362      437      394

```

## 步骤 4：解决该问题。

在步骤 3 中，已确定了发生这种情况的根本原因。要避免此问题，请从 ACL 中删除 **log** 关键字。在 Cisco IOS 软件版本 12.1(13)EW1 及更高版本中，数据包速率会受到限制，以使 CPU 使用率不会变得过高。可将访问列表计数器用作记录 ACL 匹配项的方法。可以在 **show access-list acl\_id** 命令输出中看到访问列表计数器。

## 第 2 层转发环路

生成树协议 (STP) 的不良实施和可能影响 STP 的各种问题可能导致第 2 层转发环路。

### 步骤 1：使用 `show processes cpu` 命令检查 Cisco IOS 进程

本部分介绍了管理员在缩小高 CPU 使用率问题范围时使用的命令。如果发出 `show processes cpu` 命令，则可以看到两个主进程 ( Cat4k Mgmt LoPri 和 Spanning Tree ) 对 CPU 的使用率较高。仅通过此信息，您就可以知道生成树进程占用了很大一部分的 CPU 周期。

```
Switch#show processes cpu
CPU utilization for five seconds: 74%/1%; one minute: 73%; five minutes: 50%
 PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min TTY Process
   1         4       198        20  0.00%  0.00%  0.00%  0 Chunk Manager
   2         4       290        13  0.00%  0.00%  0.00%  0 Load Meter
!--- Output suppressed. 25 488 33 14787 0.00% 0.02% 0.00% 0 Per-minute Jobs 26 90656 223674 405
6.79% 6.90% 7.22% 0 Cat4k Mgmt HiPri 27      158796      59219      2681 32.55% 33.80% 21.43%
0 Cat4k Mgmt LoPri
 28         20      1693         11  0.00%  0.00%  0.00%  0 Galios Reschedul
 29         0         1         0  0.00%  0.00%  0.00%  0 IOS ACL Helper
 30         0         2         0  0.00%  0.00%  0.00%  0 NAM Manager
!--- Output suppressed. 41 0 1 0 0.00% 0.00% 0.00% 0 SFF8472 42 0 2 0 0.00% 0.00% 0.00% 0 AAA
Dictionary R 43      78564      20723      3791 32.63% 30.03% 17.35% 0 Spanning Tree
 44        112       999        112  0.00%  0.00%  0.00%  0 DTP Protocol
 45         0       147         0  0.00%  0.00%  0.00%  0 Ethchnl
```

### 步骤 2：使用 `show platform health` 命令检查 Catalyst 4500 特定的进程。

要了解哪个平台特定的进程占用了 CPU，请发出 `show platform health` 命令。从该输出中，可以看到 `K2CpuMan Review` 进程 ( 处理计算密集型数据包的作业 ) 用尽了 CPU：

```
Switch#show platform health
%CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
      Target Actual Target Actual   Fg   Bg 5Sec Min Hour   CPU
!--- Output suppressed. TagMan-RecreateMtegR 1.00 0.00 10 0 100 500 0 0 0 0:00 K2CpuMan Review
30.00 37.62   30   53 100 500  41 33   1 2:12
K2AccelPacketMan: Tx 10.00 4.95  20   0 100 500  5 4  0 0:36
K2AccelPacketMan: Au  0.10 0.00   0   0 100 500  0 0  0 0:00
K2AclMan-taggedFlatA 1.00 0.00  10   0 100 500  0 0  0 0:00
```

### 步骤 3：检查接收数据流的 CPU 队列以确定计算密集型数据流的类型。

发出 `show platform cpu packet statistics` 命令以检查哪一个 CPU 队列收到了计算密集型数据包。本部分中的输出显示 `control` 队列收到大量数据包。请使用表 1 中的信息和步骤 1 中做出的结论。可以确定 CPU 处理的数据包，并可以确定 CPU 使用率较高的原因是 BPDU 处理。

```
Switch#show platform cpu packet statistics
!--- Output suppressed. Total packet queues 16 Packets Received by Packet Queue Queue Total 5
sec avg 1 min avg 5 min avg 1 hour avg -----
- - - - - Esmp 202760 196 173 128 28 Control 388623
2121      1740      598      16

Packets Dropped by Packet Queue

Queue          Total          5 sec avg 1 min avg 5 min avg 1 hour avg
-----
Control          17918          0          19          24          3
```

### 步骤 4：确定根本原因并解决问题



通常，您可以完成以下步骤来排除故障（一些步骤不是必需的，具体视情况而定）：

1. 确定环路。
2. 查明环路的范围。
3. 断开环路。
4. 修复形成环路的原因。
5. 恢复冗余。

每一个步骤都在[排除转发环路故障 - 排除运行 Cisco IOS 系统软件的 Catalyst 交换机上的 STP 故障](#)中进行了详细说明。

## 步骤 5：实施高级 STP 功能

- **BDPU 防护** - 保护 STP，以防未经授权的网络设备连接到已启用 portfast 的端口。有关详细信息，请参阅[生成树 PortFast BDPDU 防护增强功能](#)。
- **环路防护** - 提高第 2 层网络的稳定性。有关详细信息，请参阅[使用环路防护和 BDPU 迟滞检测功能的生成树协议 \(STP\) 增强功能](#)。
- **根防护** - 强制在网络中放置根网桥。有关详细信息，请参阅[生成树协议根防护增强功能](#)。
- **UDLD** —检测单向链路并且防止转发环路。有关详细信息，请参阅[了解和配置单向链路检测协议 \(UDLD\) 功能](#)。

## CPU 使用率较高的其他原因

以下是 CPU 使用率较高的一些其他已知原因：

- [过度的链路抖动](#)
- [由于 FIB 一致性检查导致 CPU 使用率达到峰值](#)
- [执行 K2FibAdjMan Host Move 进程时 CPU 使用率较高](#)
- [执行 RkiosPortMan Port Review 进程时 CPU 使用率较高](#)
- [使用中继端口连接到 IP 电话时 CPU 使用率较高](#)
- [使用 RSPAN 和第 3 层控制数据包时 CPU 使用率较高](#)
- 在大型 ACL 编程期间达到峰值从接口应用或删除大型 ACL 期间，会出现 CPU 使用率峰值。

### 过度的链路抖动

当一个或多个已连接的链路开始过度抖动时，Catalyst 4500 将显示 CPU 使用率较高。这种情况出现在早于 Cisco IOS 软件版本 12.2(20)EWA 的 Cisco IOS 软件版本。

### 步骤 1：使用 show processes cpu 命令检查 Cisco IOS 进程。

发出 **show processes cpu** 命令以检查哪一个 Cisco IOS 进程占用了 CPU。在此命令输出中，请注意 CPU 使用率最高的进程是 **Cat4k Mgmt LoPri**：

```
Switch#show processes cpu
CPU utilization for five seconds: 96%/0%; one minute: 76%; five minutes: 68%
  PID Runtime(ms)   Invoked    uSecs   5Sec   1Min   5Min TTY Process
    1         0         4         0  0.00%  0.00%  0.00%  0 Chunk Manager
    2      9840     463370         21  0.00%  0.00%  0.00%  0 Load Meter
    3         0         2         0  0.00%  0.00%  0.00%  0 SNMP Timers
!--- Output suppressed. 27 232385144 530644966 437 13.98% 12.65% 12.16% 0 Cat4k Mgmt HiPri 28
564756724 156627753      3605 64.74% 60.71% 54.75% 0 Cat4k Mgmt LoPri
```

## 步骤 2：使用 show platform health 命令检查 Catalyst 4500 特定的进程。

show platform health 命令的输出指明 KxAclPathMan create 进程用尽了 CPU。此进程用于创建内部路径。

```
Switch#show platform health
```

	%CPU		RunTimeMax		Priority		Average %CPU			Total CPU
	Target	Actual	Target	Actual	Fg	Bg	5Sec	Min	Hour	
Lj-poll	1.00	0.03	2	0	100	500	0	0	0	9:49
GalChassisVp-review	3.00	1.11	10	62	100	500	0	0	0	37:39
S2w-JobEventSchedule	10.00	2.85	10	8	100	500	2	2	2	90:00
Stub-JobEventSchedul	10.00	5.27	10	9	100	500	4	4	4	186:2
Pim-review	0.10	0.00	1	0	100	500	0	0	0	2:51
Ebm-host-review	1.00	0.00	8	4	100	500	0	0	0	8:06
Ebm-port-review	0.10	0.00	1	0	100	500	0	0	0	0:14
Protocol-aging-revie	0.20	0.00	2	0	100	500	0	0	0	0:00
Acl-Flattener	1.00	0.00	10	5	100	500	0	0	0	0:00
<b>KxAclPathMan create/</b>	<b>1.00</b>	<b>69.11</b>	<b>10</b>	<b>5</b>	<b>100</b>	<b>500</b>	<b>42</b>	<b>53</b>	<b>22</b>	<b>715:0</b>
KxAclPathMan update	2.00	0.76	10	6	100	500	0	0	0	86:00
KxAclPathMan reprogr	1.00	0.00	2	1	100	500	0	0	0	0:00
TagMan-InformMtegRev	1.00	0.00	5	0	100	500	0	0	0	0:00
TagMan-RecreateMtegR	1.00	0.00	10	227	100	500	0	0	0	0:00
K2CpuMan Review	30.00	8.05	30	57	100	500	6	5	5	215:0
K2AccelPacketMan: Tx	10.00	6.86	20	0	100	500	5	5	4	78:42

## 步骤 3：确定根本原因。

为链路打开/关闭消息启用日志记录。默认情况下没有启用此日志记录。启用日志记录可帮助您非常迅速地缩小引起问题的链路的范围。在所有接口下发出 logging event link-status 命令。可以使用 interface range 命令方便地对一系列接口启用，如以下示例所示：

```
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface range gigabitethernet 5/1 - 48
Switch(config-if-range)#logging event link-status
Switch(config--if-range)#end
Switch#show logging
```

!--- Output suppressed. 3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to down 3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to up 3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to down 3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to up 3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to down 3w5d: %LINK-3-UPDOWN: Interface GigabitEthernet5/24, changed state to up

在确定了有故障的接口或抖动的接口之后，请关闭该接口以解决 CPU 使用率较高的问题。Cisco IOS 软件版本 12.2(20)EWA 及更新版本已改进了 Catalyst 4500 在此抖动链路情况下的行为。因此，对 CPU 的影响不如改进之前那么大。请记住，该进程是后台进程。由于此问题引起的高 CPU 使用率不会对 Catalyst 4500 交换机造成负面影响。

## 由于 FIB 一致性检查导致 CPU 使用率达到峰值

在 FIB 表一致性检查期间，Catalyst 4500 可能会显示 CPU 使用率瞬间达到峰值。FIB 表是 CEF 进程创建的 L3 转发表。一致性检查可维护 Cisco IOS 软件 FIB 表和硬件条目之间的一致性。该一致性确保了数据包不会被错误地路由。该检查每 2 秒进行一次，并且是作为优先级较低的后台进程运行的。此进程是正常行为，不会干扰其他优先级较高的进程或数据包。

show platform health 命令的输出显示 K2Fib Consistency Ch 占用了大部分的 CPU。

**注意：**此进程在一分钟或一小时内的平均 CPU 使用率微乎其微，这一点证实了该检查只是一个短暂的定期检查。此后台进程只使用空闲 CPU 周期。

```
Switch#show platform health
          %CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
          Target Actual Target Actual     Fg    Bg 5Sec Min Hour   CPU
Lj-poll          1.00  0.02      2      1  100  500    0  0  0  1:09
GalChassisVp-review  3.00  0.29     10     3  100  500    0  0  0  11:15
!--- Output suppressed. K2Fib cam usage revi 2.00 0.00 15 0 100 500 0 0 0 0:00 K2Fib IrmFib
Review 2.00 0.00 15 0 100 500 0 0 0 0:00 K2Fib Vrf Default Ro 2.00 0.00 15 0 100 500 0 0 0 0:00
K2Fib AdjRepop Revie 2.00 0.00 15 0 100 500 0 0 0 0:00 K2Fib Vrf Unpunt Rev 2.00 0.01 15 0 100
500 0 0 0 0:23 K2Fib Consistency Ch 1.00 60.40 5 2 100 500 0 0 0 100:23
K2FibAdjMan Stats Re  2.00  0.30     10     4  100  500    0  0  0  6:21
K2FibAdjMan Host Mov  2.00  0.00     10     4  100  500    0  0  0  0:00
K2FibAdjMan Adj Chan  2.00  0.00     10     0  100  500    0  0  0  0:00
K2FibMulticast Signa  2.00  0.01     10     2  100  500    0  0  0  2:04
```

### [执行 K2FibAdjMan Host Move 进程时 CPU 使用率较高](#)

Catalyst 4500 可能会显示执行 **K2FibAdjMan Host Move** 进程时的 CPU 使用率较高。这一高使用率显示在 **show platform health** 命令的输出中。许多 MAC 地址频繁到期或在新端口上被识别，这会导致此 CPU 使用率较高。默认值 `mac-address-table` 过期时间是 5 分钟或 300 秒。此问题的应急方案是增加 MAC 地址过期时间，或者您能设计网络为了避免 MAC 地址移动大量。Cisco IOS 软件版本 12.2(18)EW 及更高版本增强了此处理行为以减少占用的 CPU。请参阅 Cisco bug ID [CSCed15021](#)（[仅限注册用户](#)）。

```
Switch#show platform health
          %CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
          Target Actual Target Actual     Fg    Bg 5Sec Min Hour   CPU
Lj-poll          1.00  0.02      2      1  100  500    0  0  0  1:09
GalChassisVp-review  3.00  0.29     10     3  100  500    0  0  0  11:15
S2w-JobEventSchedule 10.00  0.32     10     7  100  500    0  0  0  10:14
!--- Output suppressed. K2FibAdjMan Stats Re 2.00 0.30 10 4 100 500 0 0 0 6:21 K2FibAdjMan Host
Mov 2.00 18.68 10 4 100 500 25 29 28 2134:39
K2FibAdjMan Adj Chan  2.00  0.00     10     0  100  500    0  0  0  0:00
K2FibMulticast Signa  2.00  0.01     10     2  100  500    0  0  0  2:04
K2FibMulticast Entry  2.00  0.00     10     7  100  500    0  0  0  0:00
```

您能修改 MAC 地址的最大老化时间在全局配置模式的。命令语法是 `mac-address-table` 一个路由器和 `mac-address-table` 过期时间秒钟 [vlan vlan-id] 的过期时间秒钟 Catalyst 交换机的。欲知更多信息，参考 [Cisco IOS 交换服务命令参考指南](#)。

### [执行 RkiosPortMan Port Review 进程时 CPU 使用率较高](#)

Catalyst 4500 可能会在 Cisco IOS 软件版本 12.2(25)EWA 和 12.2(25)EWA1 的 `show platform health` 命令的输出中显示执行 **RkiosPortMan Port Review** 进程时的 CPU 使用率较高。Cisco Bug ID [CSCeh08768](#)（[仅限注册用户](#)）导致了使用率较高，而 Cisco IOS 软件版本 12.2(25)EWA2 已解决了该问题。此进程是后台进程，并且不影响 Catalyst 4500 交换机的稳定性。

```
Switch#show platform health
          %CPU   %CPU   RunTimeMax   Priority   Average %CPU   Total
          Target Actual Target Actual     Fg    Bg 5Sec Min Hour   CPU
Lj-poll          1.00  0.02      2      1  100  500    0  0  0  1:09
GalChassisVp-review  3.00  0.29     10     3  100  500    0  0  0  11:15
S2w-JobEventSchedule 10.00  0.32     10     7  100  500    0  0  0  10:14
!--- Output suppressed. K2 Packet Memory Dia 2.00 0.00 15 8 100 500 0 1 1 45:46 K2 L2 Aging
Table Re 2.00 0.12 20 3 100 500 0 0 0 7:22 RkiosPortMan Port Re 2.00 87.92 12 7 100
500 99 99 89 1052:36
Rkios Module State R  4.00  0.02     40     1  100  500    0  0  0  1:28
Rkios Online Diag Re  4.00  0.02     40     0  100  500    0  0  0  1:15
```

## 使用中继端口连接到 IP 电话时 CPU 使用率较高

如果为语音 VLAN 选项和接入 VLAN 选项都配置了某个端口，则该端口将用作多 VLAN 接入端口。优点是只中继那些为语音和接入 VLAN 选项配置的 VLAN。

被中继到电话的 VLAN 会增加 STP 实例的数量。交换机会管理这些 STP 实例。管理的 STP 实例数增加也会导致 STP CPU 使用率升高。

中继所有 VLAN 也会导致不必要的广播、多播和未知单播数据流发送到电话链路。

```
Switch#show processes cpu
```

```
CPU utilization for five seconds: 69%/0%; one minute: 72%; five minutes: 73%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4	165	24	0.00%	0.00%	0.00%	0	Chunk Manager
2	29012	739091	39	0.00%	0.00%	0.00%	0	Load Meter
3	67080	13762	4874	0.00%	0.00%	0.00%	0	SpanTree Helper
4	0	1	0	0.00%	0.00%	0.00%	0	Deferred Events
5	0	2	0	0.00%	0.00%	0.00%	0	IpSecMibTopN
6	4980144	570766	8725	0.00%	0.09%	0.11%	0	Check heaps
26	539173952	530982442	1015	13.09%	13.05%	13.20%	0	Cat4k Mgmt HiPri
27	716335120	180543127	3967	17.61%	18.19%	18.41%	0	Cat4k Mgmt LoPri
33	1073728	61623	17424	0.00%	0.03%	0.00%	0	Per-minute Jobs
34	1366717824	231584970	5901	38.99%	38.90%	38.92%	0	Spanning Tree
35	2218424	18349158	120	0.00%	0.03%	0.02%	0	DTP Protocol
36	5160	369525	13	0.00%	0.00%	0.00%	0	Ethchnl
37	271016	2308022	117	0.00%	0.00%	0.00%	0	VLAN Manager
38	958084	3965585	241	0.00%	0.01%	0.01%	0	UDLD
39	1436	51011	28	0.00%	0.00%	0.00%	0	DHCP Snooping
40	780	61658	12	0.00%	0.00%	0.00%	0	Port-Security
41	1355308	12210934	110	0.00%	0.01%	0.00%	0	IP Input

## 使用 RSPAN 和第 3 层控制数据包时 CPU 使用率较高

用 RSPAN 捕获的第 3 层控制数据包会发往 CPU ( 而不只是发往 RSPAN 目标接口 )，这将导致 CPU 使用率较高。L3 控制数据包会被带有转发到 CPU 操作的静态 CAM 条目捕获。静态 CAM 条目对于所有 VLAN 是全局有效的。为了避免不必要的 CPU 泛洪，请使用 Cisco IOS 软件版本 12.2(37)SG 及更高版本中提供的每 VLAN 控制数据流拦截功能。

```
Switch#show processes cpu
```

```
CPU utilization for five seconds: 69%/0%; one minute: 72%; five minutes: 73%
```

PID	Runtime(ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4	165	24	0.00%	0.00%	0.00%	0	Chunk Manager
2	29012	739091	39	0.00%	0.00%	0.00%	0	Load Meter
3	67080	13762	4874	0.00%	0.00%	0.00%	0	SpanTree Helper
4	0	1	0	0.00%	0.00%	0.00%	0	Deferred Events
5	0	2	0	0.00%	0.00%	0.00%	0	IpSecMibTopN
6	4980144	570766	8725	0.00%	0.09%	0.11%	0	Check heaps
26	539173952	530982442	1015	13.09%	13.05%	13.20%	0	Cat4k Mgmt HiPri
27	716335120	180543127	3967	17.61%	18.19%	18.41%	0	Cat4k Mgmt LoPri
33	1073728	61623	17424	0.00%	0.03%	0.00%	0	Per-minute Jobs
34	1366717824	231584970	5901	38.99%	38.90%	38.92%	0	Spanning Tree
35	2218424	18349158	120	0.00%	0.03%	0.02%	0	DTP Protocol
36	5160	369525	13	0.00%	0.00%	0.00%	0	Ethchnl
37	271016	2308022	117	0.00%	0.00%	0.00%	0	VLAN Manager
38	958084	3965585	241	0.00%	0.01%	0.01%	0	UDLD
39	1436	51011	28	0.00%	0.00%	0.00%	0	DHCP Snooping
40	780	61658	12	0.00%	0.00%	0.00%	0	Port-Security
41	1355308	12210934	110	0.00%	0.01%	0.00%	0	IP Input

静态 ACL 安装在输入功能 TCAM 的顶部，以捕获发往 224.0.0.\* 范围内已知 IP 多播地址的控制数据包。静态 ACL 在启动时安装，并且会在任何用户配置 ACL 之前显示。静态 ACL 总是首先进行匹

配，然后拦截所有 VLAN 上发往 CPU 的控制数据流。

每 VLAN 控制数据流拦截功能提供了捕获控制数据流的可选每 VLAN 路径管理模式。输入功能 TCAM 中的相应静态 CAM 条目在此新模式下将失效。控制数据包由连接到启用了监听或路由功能的 VLAN 的功能特定 ACL 捕获。没有任何功能特定的 ACL 连接到 RSPAN VLAN。因此，并非所有从 RSPAN VLAN 收到的第 3 层控制数据包都会转发到 CPU。

## [分析发往 CPU 的数据流的故障排除工具](#)

如本文档所示，发往 CPU 的数据流是 Catalyst 4500 上的 CPU 使用率较高的主要原因之一。发往 CPU 的数据流可能是由于配置而有意发往 CPU 的数据流，也可能是由于错误配置或拒绝服务攻击而在无意中发往 CPU 的数据流。CPU 具有内置的 QoS 机制，用于防止由于此数据流而导致的负面网络影响。但是，如果它的作用不理想，请确定计算密集型数据流的根本原因并消除该数据流。

### [工具 1：使用 SPAN 监控 CPU 数据流 - Cisco IOS 软件版本 12.1\(19\)EW 及更高版本](#)

Catalyst 4500 允许使用标准的 SPAN 功能监控流进或流出的计算密集型数据流。目标接口连接到一个数据包监控器或一台运行数据包嗅探器软件的管理员便携式计算机。此工具可帮助您迅速、准确地分析 CPU 处理的数据流。使用此工具可以监控绑定到 CPU 数据包引擎的单个队列。

**注意：**交换引擎有 32 个队列用于 CPU 数据流，而 CPU 数据包引擎有 16 个队列。

```
Switch(config)#monitor session 1 source cpu ?
  both    Monitor received and transmitted traffic
  queue   SPAN source CPU queue
  rx      Monitor received traffic only
  tx      Monitor transmitted traffic only
  <cr>

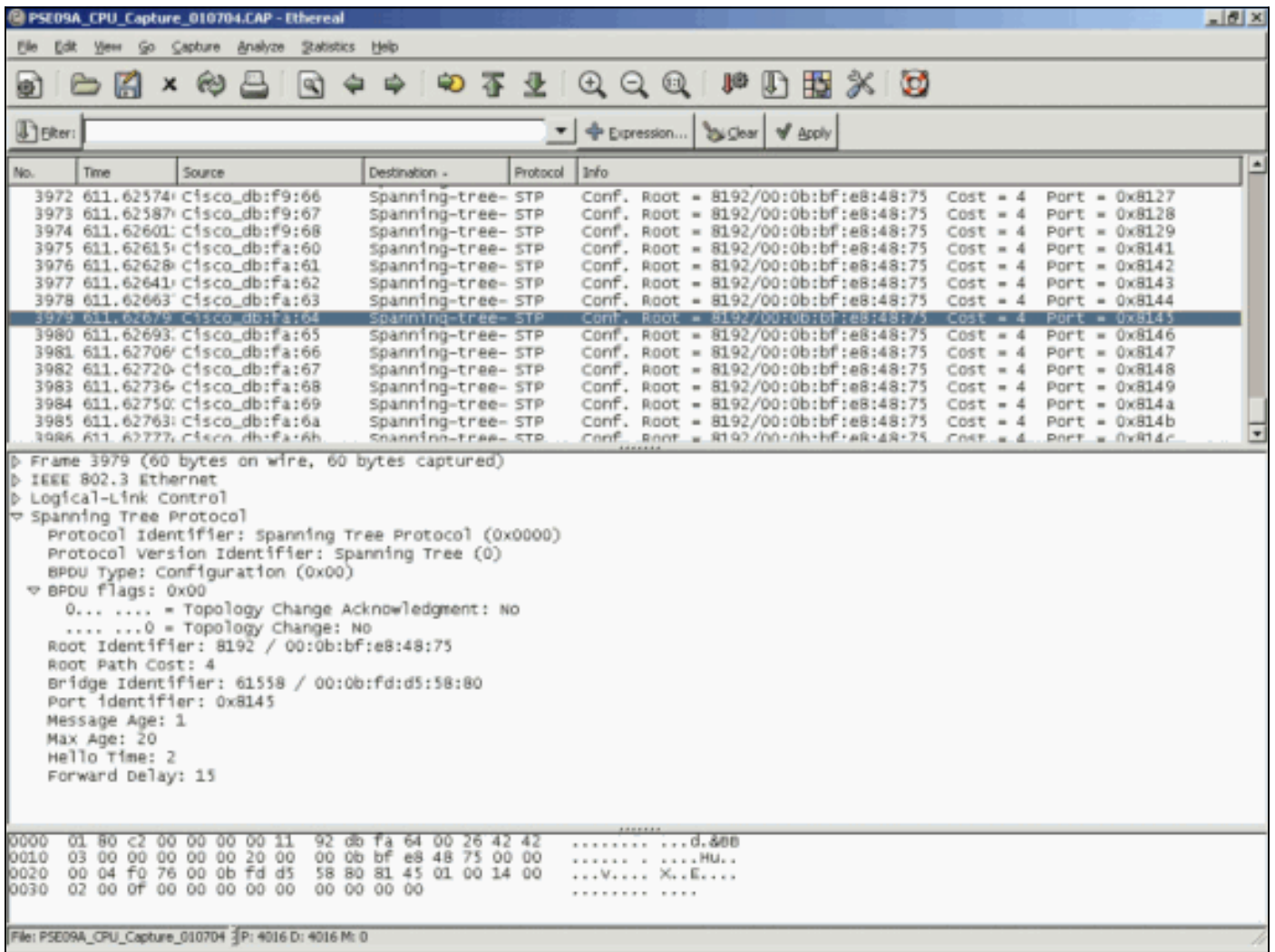
Switch(config)#monitor session 1 source cpu queue ?
  <1-32>   SPAN source CPU queue numbers
  acl      Input and output ACL [13-20]
  adj-same-if  Packets routed to the incoming interface [7]
  all      All queues [1-32]
  bridged  L2/bridged packets [29-32]
  control-packet Layer 2 Control Packets [5]
  mtu-exceeded Output interface MTU exceeded [9]
  nfl      Packets sent to CPU by netflow (unused) [8]
  routed   L3/routed packets [21-28]
  rpf-failure Multicast RPF Failures [6]
  span     SPAN to CPU (unused) [11]
  unknown-sa Packets with missing source address [10]

Switch(config)#monitor session 1 source cpu queue all rx
Switch(config)#monitor session 1 destination interface gigabitethernet 1/3
Switch(config)#end
4w6d: %SYS-5-CONFIG_I: Configured from console by console
```

```
Switch#show monitor session 1
Session 1
-----
Type           : Local Session
Source Ports   :
  RX Only      : CPU
Destination Ports : Gi1/3
  Encapsulation : Native
  Ingress      : Disabled
  Learning     : Disabled
```

如果连接到运行嗅探器程序的 PC，则可以迅速分析数据流。在本部分的窗口中显示的输出中，可以看到 CPU 使用率较高的原因是 STP BPDU 的数量过多。

**注意：** CPU 嗅探器中的 STP BPDU 是正常的。但是，如果您看到的数量多于预期，则表明可能已超过了 Supervisor 引擎建议的限制值。有关详细信息，请参阅本文档的[大量的生成树端口实例部分](#)。



## 工具 2：内置的 CPU 嗅探器 - Cisco IOS 软件版本 12.2(20)EW 及更高版本

Catalyst 4500 提供了内置的 CPU 嗅探器和解码器，用于迅速确定发送到 CPU 的数据流。可以使用 **debug** 命令启用此工具，如本部分中的示例所示。此功能实现了一次可以保留 1024 个数据包的循环缓冲区。新的数据包到达时，它们会覆盖较旧的数据包。解决高 CPU 使用率问题时使用此功能是十分安全的。

```
Switch#debug platform packet all receive buffer
platform packet debugging is on
Switch#show platform cpu packet buffered
Total Received Packets Buffered: 36
-----
Index 0:
7 days 23:6:32:37214 - RxVlan: 99, RxPort: Gi4/48
Priority: Crucial, Tag: Dot1Q Tag, Event: Control Packet, Flags: 0x40, Size: 68
Eth: Src 00-0F-F7-AC-EE-4F Dst 01-00-0C-CC-CC-CD Type/Len 0x0032
Remaining data:
 0: 0xAA 0xAA 0x3 0x0 0x0 0xC 0x1 0xB 0x0 0x0
10: 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16 0x63 0x28
20: 0x62 0x0 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16
```

```

30: 0x63 0x28 0x62 0x80 0xF0 0x0 0x0 0x14 0x0 0x2
40: 0x0 0xF 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x63
Index 1:
7 days 23:6:33:180863 - RxVlan: 1, RxPort: Gi4/48
Priority: Crucial, Tag: Dot1Q Tag, Event: Control Packet, Flags: 0x40, Size: 68
Eth: Src 00-0F-F7-AC-EE-4F Dst 01-00-0C-CC-CC-CD Type/Len 0x0032
Remaining data:

```

```

0: 0xAA 0xAA 0x3 0x0 0x0 0xC 0x1 0xB 0x0 0x0
10: 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16 0x63 0x28
20: 0x62 0x0 0x0 0x0 0x0 0x80 0x0 0x0 0x2 0x16
30: 0x63 0x28 0x62 0x80 0xF0 0x0 0x0 0x14 0x0 0x2
40: 0x0 0xF 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x63

```

**注意：**发出 **debug** 命令时，CPU 使用率几乎总是 100%。发出 **debug** 命令时，CPU 使用率较高是正常的。

## [工具 3：确定将数据流发送到 CPU 的接口 - Cisco IOS 软件版本 12.2\(20\)EW 及更高版本](#)

Catalyst 4500 提供了另一个有用的工具，用于确定发送供 CPU 处理的数据流/数据包最多的接口。此工具可以帮助您快速确定将大量的广播或其他拒绝服务攻击发送到 CPU 的任务设备。解决高 CPU 使用率问题时使用此功能也是十分安全的。

```

Switch#debug platform packet all count
platform packet debugging is on
Switch#show platform cpu packet statistics
!--- Output suppressed. Packets Transmitted from CPU per Output Interface Interface Total 5 sec
avg 1 min avg 5 min avg 1 hour avg -----
----- Gi4/47 1150 1 5 10 0 Gi4/48 50 1 0 0 0 Packets Received at CPU per Input
Interface
Interface                Total                5 sec avg 1 min avg 5 min avg 1 hour avg
-----
Gi4/47                    23130                5          10          50          20
Gi4/48                     50                   1           0           0           0

```

**注意：**发出 **debug** 命令时，CPU 使用率几乎总是 100%。发出 **debug** 命令时，CPU 使用率较高是正常的。

## 摘要

Catalyst 4500 交换机处理 IP 版本 4 (IPv4) 数据包在硬件中的高速转发。某些功能或例外情况可能造成一些数据包通过 CPU 处理路径转发。Catalyst 4500 使用成熟的 QoS 机制处理计算密集型数据包。此机制可确保交换机的可靠性和稳定性，并同时最大限度地减少 CPU 用于执行数据包的软件转发。Cisco IOS 软件版本 12.2(25)EWA2 及更高版本提供了其他用于数据包/进程处理以及记帐的增强功能。Catalyst 4500 有足够的命令和强大的工具帮助确定发生高 CPU 使用率的情况的根本原因。但是，在大多数情况下，Catalyst 4500 上的 CPU 使用率较高并不会导致网络不稳定，也无需担心。

## 相关信息

- [运行 CatOS 软件的 Catalyst 4500/4000、2948G、2980G 和 4912G 交换机上的 CPU 使用率](#)
- [LAN 产品支持页](#)
- [LAN 交换技术支持页](#)
- [技术支持和文档 - Cisco Systems](#)