# CPU 使用率が高い場合のトラブルシューティング

## Troubleshooting High CPU Utilization

OL-17977-01-J



このマニュアルの内容は、次のとおりです。

# CPU 使用率の概要

スイッチのブート プロセスが完了すると、スイッチの CPU で、次の 2 つの異なる機能が同時に実行されます。

- ネットワークでスイッチが稼動するために必要なさまざまなシステム プロセスが実行される。
- スイッチ ハードウェアに対するパケットの送受信を行う。

CPU 使用率は、システム プロセスが時間を要するほど、またネットワーク パケットの送受信量が多いほど増大します。

通常の動作条件下では、スタック構成にできないスイッチの CPU は、CPU 時間の 5% 以上でビジー状態になります。スイッチがスタック構成の場合、CPU は、7 または 8% 以上の使用率でビジー状態になります。スイッチ スタックでは、CPU 使用率は、マスター スイッチでのみ測定されます。スタック内のメンバーの数が CPU 使用率全体に影響します。

スイッチ タイマーのシスコのバックグラウンド システム プロセスが毎秒数回実行するため、最も単純な配置であっても、スイッチの CPU 使用率が 0% と報告されることはありません。

**（注）** データ トラフィックの通常のパケット スイッチングは、スイッチ ハードウェアで行われ、CPU は関与しないので、過度にビジー状態の CPU の影響を受けることはありません。

CPU が過度にビジー状態になるのは、CPU がスイッチ ハードウェアから受信するパケットが多すぎる場合や、システム プロセスが CPU 時間を消費しすぎる場合です。このどちらかの作用が原因でもう一方に支障が出るほど CPU リソースを使用している場合に、CPU が過度にビジー状態になります。たとえば、ネットワーク上のブロードキャスト ストームのために CPU が多くのパケットを受信している場合は、そのすべてのパケットを処理することで過度にビジー状態になるため、別のシステム プロセスが CPU リソースにアクセスできなくなります。

スイッチの CPU 使用率を確認するには、**show processes cpu sorted** 特権 EXEC コマンドを入力します。出力には、過去 5 秒、過去 1 分、および過去 5 分の CPU のビジー状態が表示されます。また、各システム プロセスがこれらの期間に使用した CPU 使用率も表示されます。

```
Switch# show processes cpu sorted
CPU utilization for five seconds: 5%/0%; one minute: 6%; five minutes: 5%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1        4539     89782         50  0.00%  0.00%  0.00%   0 Chunk Manager
   2        1042   1533829          0  0.00%  0.00%  0.00%   0 Load Meter
   3           0         1          0  0.00%  0.00%  0.00%   0 DiagCard3/-1
   4    14470573   1165502      12415  0.00%  0.13%  0.16%   0 Check heaps
   5        7596    212393         35  0.00%  0.00%  0.00%   0 Pool Manager
   6           0         2          0  0.00%  0.00%  0.00%   0 Timers
   7           0         1          0  0.00%  0.00%  0.00%   0 Image Licensing
   8           0         2          0  0.00%  0.00%  0.00%   0 License Client N
   9     1442263     25601      56336  0.00%  0.08%  0.02%   0 Licensing Auto U
  10           0         1          0  0.00%  0.00%  0.00%   0 Crash writer
  11      979720   2315501        423  0.00%  0.00%  0.00%   0 ARP Input
  12           0         1          0  0.00%  0.00%  0.00%   0 CEF MIB API
  <output truncated>
```

この出力では、過去 5 秒間の CPU 使用率には 2 つの数字（5%/0%）が表示されています。

- 最初の数字 5% で、CPU が過去 5 秒間にどの程度ビジー状態だったかがわかります。この数字は、全アクティブ システム プロセスの総 CPU 使用率であり、割り込みレベルでの時間の割合も含まれています。
- 2 番目の数字 0% は、過去 5 秒間の割り込みレベルでの時間の割合が表示されています。割り込みの割合とは、スイッチ ハードウェアからのパケットの受信に費やされる CPU 時間のことです。割り込みレベルでの時間の割合は、常に総 CPU 使用率以下です。

その他の 2 つの重要な数字として、過去 1 分間の平均使用率（この例では 6%）および過去 5 分間の平均使用率（この例では 5%）が同じ出力行に表示されます。これらの値は、小規模な安定した環境での、スタック構成されていないスイッチにおける標準的な値です。

CPU では、常時多数のシステム プロセスがアクティブになっていると考えられます。この数字は、スイッチ モデル、Cisco IOS リリース、機能セット、およびスイッチ スタック内のスイッチの数（該当する場合）に基づいて変わる可能性があります。たとえば、IP ベース イメージが稼動する Catalyst 3750 スイッチのスタックでは、通常、475 のシステム プロセスがアクティブになっています。LAN ベース イメージが稼動する Catalyst 2960 スイッチのアクティブ プロセス数は、Catalyst 3750 スイッチよりも少なくなっています。一般に、Cisco IOS イメージの機能が多いほど、システム プロセス数も多くなります。

# CPU 使用率が高いことが問題となっている場合

ここでは、CPU 使用率が高い場合にそれを特定し、それが問題であるかどうかを確認する方法について説明します。


- 「CPU 使用率が高くなる正常な状態」（P.5）
- 「CPU 使用率が高いことに起因するネットワークの症状」（P.6）
- 「割り込みの割合の特定」（P.7）


場合によっては、CPU 使用率が高いことは正常であり、ネットワークの問題の原因にはなりません。CPU 使用率が高いことが問題になるのは、スイッチが期待どおりに機能しない場合です。

**show processes cpu history** 特権 EXEC コマンドを入力し、過去 60 秒、60 分、および 72 時間の CPU 使用率を確認します。このコマンドの出力では、CPU のビジー状態の程度がグラフィカルに表示されます。CPU が絶えずビジー状態だったか、あるいは使用率が急増する箇所があったかどうかを確認できます。

この例では、スイッチが再起動されて間もない 46 時間前に CPU 使用率が 100% まで急増しています。また、過去 1 時間以内に 87% まで急増しています。

```
Switch# show processes cpu history

    222222222222222222222222222222333332222222222222222222222
    666666666666666665555577777777771111199999666667777766666
100
 90
 80
 70
 60
 50
 40
 30 ************************************************************
 20 ************************************************************
 10 ************************************************************
   0....5....1....1....2....2....3....3....4....4....5....5....
             0    5    0    5    0    5    0    5    0    5
               CPU% per second (last 60 seconds)

    322222222222222222223222222223233422222332222222228232222
    198998988898888889898089888989070089999710899997989770798
100
 90                                                        *
 80                                                        *
 70                                                        *
 60                                                        *
 50                                    *                   *
 40                                    *                   *
 30 ############################################################
 20 ############################################################
 10 ############################################################
   0....5....1....1....2....2....3....3....4....4....5....5....
             0    5    0    5    0    5    0    5    0    5
               CPU% per minute (last 60 minutes)
              * = maximum CPU%   # = average CPU%
                                                      1
    854454545544455465445544344544544565664574848 08
    966411455779337115175011692330343333488118230 06
100                                                       *
 90 *                                                     **
 80 *                                                  * ***
 70 *                                               *  * * ***
 60 **       **     * **  *                       * **  * * ***
 50 *** * ****** ** ** ***    * *  *  ********** ***
 40 **********************************************
 30 ##############################################*
 20 ##############################################*
 10 ##############################################
   0....5....1....1....2....2....3....3....4....4....5....5....6....6....7.
             0    5    0    5    0    5    0    5    0    5    0    5    0
                  CPU% per hour (last 72 hours)
                 * = maximum CPU%   # = average CPU%
```

CPU 使用率は 52 分前に 87% まで急増しました。

CPU 使用率は 46 時間前に 100% まで急増しました。

205495

既知のネットワーク イベントやアクティビティによって CPU 使用率が急増することは、問題ではありません。87% の急増でも、原因によっては許容できる場合があります。たとえば、ネットワーク管理者が CLI で **write memory** 特権 EXEC コマンドを入力したことによる急増は許容範囲内です。また、大規模なレイヤ 2 ネットワークのトポロジを変更した場合の急増も、正常な反応です。CPU 使用率が急増する原因となるイベントおよびアクティビティのリストについては、「CPU 使用率が高くなる正常な状態」（P.5）を参照してください。

時間の経過とともに、スイッチは一定の持続的な CPU 使用率の範囲で動作するため、それが通常動作の基準と見なされます。**show processes cpu history** 特権 EXEC コマンドの出力を使用すると、過去 72 時間の通常動作の基準使用率を特定できます。不明な理由により CPU 使用率が通常動作の基準を超えている場合は、問題があるとわかります。

先ほどの例では、過去 37 時間の CPU 使用率は 40 ～ 56% でした。50% 未満の CPU 使用率は許容範囲内と考えられます。この例のような、50% 前後が続く場合も許容範囲内です。50% を超える CPU 使用率が続く場合は、問題が発生しているおそれがあります。このレベルの CPU 使用率では、スイッチは問題なく動作しているように思えますが、スイッチの CPU の使用率で 50% が続く場合は、ダイナミック ネットワーク イベントへの応答機能のセキュリティが侵害されています。

確立された通常動作基準に対して頻繁に発生する不可解な急増や説明のつかない使用率の急増は、不安材料となります。CPU 使用率が高くなる問題の原因を特定するには、「根本的な原因の特定」（P.7）を参照してください。

# CPU 使用率が高くなる正常な状態

一部のネットワーク配置では、CPU は、ビジー状態で正常です。一般に、レイヤ 2 またはレイヤ 3 のネットワークが大規模になるほど、CPU に対するネットワーク関連トラフィックの処理要求が増大します。次の例では、CPU 使用率が高くなる可能性のある動作を示します。

- 「スパニングツリー」（P.5）
- 「IP ルーティング テーブルの更新」（P.5）
- 「Cisco IOS コマンド」（P.5）
- 「CPU 使用率が高くなる原因となるその他のイベント」（P.6）

## スパニングツリー

レイヤ 2 のスパニングツリー インスタンスは、レイヤ 2 スイッチの Per-VLAN Spanning-Tree（PVST; VLAN 単位のスパニングツリー）機能によって設定されたすべての VLAN で実行します。スパニングツリーによって費やされる CPU 時間は、スパニングツリー インスタンスの数およびアクティブなインターフェイスの数によって異なります。インスタンスの数やアクティブなインターフェイスの数が多くなるほど、CPU 使用率は増大します。

## IP ルーティング テーブルの更新

IP ルーティングをイネーブルにしたレイヤ 3 スイッチが大量のルーティング テーブルを受信すると、そのスイッチでルーティング情報の更新処理を行う必要があります。処理中の CPU 使用率の急増は、次の要因に依存します。

- ルーティング更新情報のサイズ
- 更新の頻度
- 更新情報を受信するルーティング プロトコル プロセスの数
- ルート マップまたはフィルタの存在

## Cisco IOS コマンド

Cisco IOS コマンドの中には、CPU 使用率が急増する原因となるものもあります。モジュールは次のとおりです。

- **show tech-support** 特権 EXEC コマンド。
- **write memory** 特権 EXEC コマンド（特にスイッチがスタック内にある場合）。
- スイッチ スタック マスターでの **show-running configuration** 特権 EXEC コマンド。
- 機能のデバッグをイネーブルにする **debug** 特権 EXEC コマンド。デバッグがイネーブルになっている間は、デバッグ メッセージのコンソール出力により CPU 使用率が増大します。

## CPU 使用率が高くなる原因となるその他のイベント

- 高頻度または大量の IGMP 要求：CPU で IGMP メッセージが処理されます。
- 大量の IP SLA モニタリング セッション：CPU で ICMP パケットまたは traceroute パケットが生成されます。
- SNMP ポーリング アクティビティ（特に MIB Walk）: Cisco IOS SNMP エンジンで SNMP 要求が実行されます。
- 多数のクライアントへのリンクが復元されるときなどの大量の同時 DHCP 要求（スイッチが DHCP サーバとして機能している場合）。
- ARP ブロードキャスト ストーム。
- イーサネット ブロードキャスト ストーム。

# CPU 使用率が高いことに起因するネットワークの症状

CPU が過度にビジー状態になっている場合は、システム プロセスが期待どおりに実行できなくなります。システム プロセスが実行しない場合、スイッチ（または直接接続されたネットワーク デバイス）は、ネットワークに問題があるかのように反応します。レイヤ 2 ネットワークの場合は、スパニングツリー再コンバージが行われることがあります。レイヤ 3 ネットワークでは、ルーティング トポロジが変更されることがあります。

スイッチの CPU が過度にビジー状態になっている場合は、次のような既知の症状が発生します。

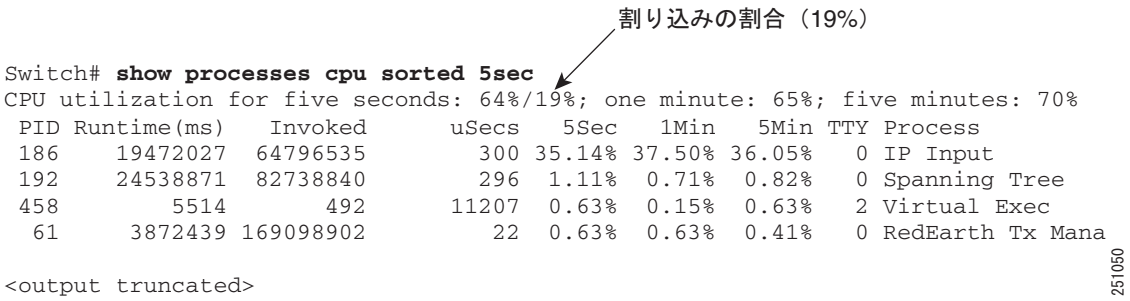
- スパニングツリー トポロジの変更：レイヤ 2 ネットワーク デバイスがルート ポートでスパニングツリー BPDU を適度なタイミングで受信しない場合は、そのデバイスでルート スイッチへのレイヤ 2 パスがダウン状態と見なされ、新しいパスの検索が試行されます。スパニングツリーがレイヤ 2 ネットワークで再コンバージします。
- ルーティング トポロジ変更の変更（BGP ルート フラッピングまたは OSPF ルート フラッピング）。
- EtherChannel リンク バウンス：EtherChannel の相手側のネットワーク デバイスが、EtherChannel リンクを維持するために必要なプロトコル パケットを受信しない場合、これが原因でリンクがダウンすることがあります。
- スイッチが、次のような通常の管理要求に応答できません。
  - ICMP ping 要求
  - SNMP タイムアウト
  - 速度が遅いまたは開始できない Telnet セッションや SSH セッション
- UDLD フラッピング：スイッチが、アグレッシブ モードでピアからのキープアライブを利用します。
- 許容しきい値を超えた SLA 応答による IP SLA 障害。
- スイッチで要求の転送や応答ができない場合の DHCP または IEEE 802.1x の障害。
- パケットがドロップされたり、ドロップ パケットをソフトウェアでルーティングするため遅延が増大します。
- HSRP フラッピング。

## 割り込みの割合の特定

CPU 使用率の履歴では、時間の経過に伴う総 CPU 使用率のみが表示され、割り込みに費やされた CPU 時間は表示されません。CPU 使用率の問題の原因を特定するためには、割り込みに費やされる時間を把握しておくことが重要です。CPU 使用率の履歴には、CPU が絶えずネットワーク パケットを受信している状況は表示されても、その原因は表示されません。

Cisco IOS **show processes cpu sorted 5sec** 特権 EXEC コマンドを入力すると、現在の CPU 使用率および最も CPU 時間を消費している IOS プロセスが表示されます。この例では、持続的な使用率が 50% という基準を超えているため、CPU は過度にビジー状態になっています。

矢印は、割り込みの割合値を示しています。この値は、5 秒間の使用率の割合における 2 番目の数字です。




割り込みの割合値が 0% より大きく 5% 未満の場合は正常です。この値が 5 ～ 10% の場合は許容範囲内です。割り込みの割合値が 10% を超えている場合は、調査する必要があります。調査内容については、「ネットワーク トラフィックの分析」（P.9）を参照してください。

# 根本的な原因の特定

CPU が過度にビジー状態になっていると思われる場合は、その理由が、システム プロセスで CPU 時間を消費しすぎているためか、またはネットワーク パケットを大量に受信しているためかを最初に特定します。この 2 つの根本的な原因のデバッグ技法には、さまざまなものがあります。ここでは、原因を特定してトラブルシューティングを行う方法について説明します。

- 「原因の特定（システム プロセスまたはネットワーク トラフィック）」（P.8）
- 「ネットワーク トラフィックの分析」（P.9）
- 「アクティブなプロセスのデバッグ」（P.22）

**（注）** 必ず、特定のプラットフォーム、および使用しているスイッチのソフトウェア リリースについてのリリース ノートを参照して、Cisco IOS の既知のバグを確認してください。それらの問題はトラブルシューティング手順から除外できます。

スイッチの CPU がビジー状態になっている場合は、通常、Telnet や SSH のような管理ツールはあまり役に立ちません。CPU 使用率に関する問題のデバッグには、スイッチ コンソールを使用することをお勧めします。

# 原因の特定（システム プロセスまたはネットワーク トラフィック）

CPU のビジー状態の程度および最も CPU 時間を消費しているオペレーティング システム プロセスを特定するには、**show processes cpu sorted 5sec** 特権 EXEC コマンドを入力します。この出力では、`CPU utilization for five seconds` の 2 番目の数字が割り込みの割合です。割り込みの割合を使用して、問題の原因がシステム プロセスにあるか、または高ネットワーク トラフィックにあるかを特定します。

総 CPU 使用率の割合に比べて割り込みの割合が高すぎる場合、CPU 使用率の問題は、システム ハードウェアから受信するパケット数が多すぎることが原因です。割り込みの割合が高いことを特定する方法については、「割り込みの割合の特定」（P.7）を参照してください。

- 割り込みの割合が高い場合は、ネットワーク トラフィックが多すぎることを表しています。これは、CPU 使用率が高くなる原因として最も一般的です。トラブルシューティングを行うには、「ネットワーク トラフィックの分析」（P.9）を参照してください。
- CPU 使用率の割合が高く割り込みの割合が低い場合は、オペレーティング システム プロセスに問題があることを表しています。トラブルシューティングを行うには、「アクティブなプロセスのデバッグ」（P.22）を参照してください。
- 両方の割合が高い場合や、割り込みの割合が CPU 使用率を大きく左右しているかどうかを特定できない場合は、最初に「ネットワーク トラフィックの分析」（P.9）を参照してください。ここに記載されている情報では CPU 使用率が高くなる問題を解決できない場合は、「アクティブなプロセスのデバッグ」（P.22）を参照してください。

この例では、CPU 使用率は 64% であり、割り込みの割合は 19% という高い状態にあります。使用率の問題は、CPU でネットワークから受信した過剰なパケットを処理していることが原因です。この場合は、「ネットワーク トラフィックの分析」（P.9）を参照してください。

割り込みの割合（19%）

```
Switch# show processes cpu sorted 5sec
CPU utilization for five seconds: 64%/19%; one minute: 65%; five minutes: 70%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
 186    19472027  64796535        300 35.14% 37.50% 36.05%   0 IP Input
 192    24538871  82738840        296  1.11%  0.71%  0.82%   0 Spanning Tree
 458        5514       492      11207  0.63%  0.15%  0.63%   2 Virtual Exec
  61     3872439 169098902         22  0.63%  0.63%  0.41%   0 RedEarth Tx Mana
  99    10237319  12680120        807  0.47%  0.66%  0.59%   0 hpm counter proc
 131     4232087 224923936         18  0.31%  0.50%  1.74%   0 Hulc LED Process
 152     2032186   7964290        255  0.31%  0.21%  0.25%   0 PI MATM Aging Pr
 140    22911628  12784253       1792  0.31%  0.23%  0.26%   0 HRPC qos request
 250    27807274  62859001        442  0.31%  0.34%  0.34%   0 RIP Router
 139     4061081   1603201       2533  0.15%  0.13%  0.15%   0 HQM Stack Proces
 261      197818  12440845         15  0.15%  0.02%  0.00%   0 CEF: IPv4 proces
 266       85849   3778063         22  0.15%  0.04%  0.00%   0 LLDP Protocol
 100     8870886  42013366        211  0.15%  0.13%  0.10%   0 HRPC pm-counters
  37     1025376   7967083        128  0.15%  0.11%  0.08%   0 Per-Second Jobs
  14           0         2          0  0.00%  0.00%  0.00%   0 AAA high-capacit
  13           0         1          0  0.00%  0.00%  0.00%   0 AAA_SERVER_DEADT
  15           0         1          0  0.00%  0.00%  0.00%   0 Policy Manager
  16         260        12      21666  0.00%  0.00%  0.00%   0 Entity MIB API
  17           0         1          0  0.00%  0.00%  0.00%   0 IFS Agent Manage
  20       24444   7964457          3  0.00%  0.00%  0.00%   0 IPC Periodic Tim
  21           0        20          0  0.00%  0.00%  0.00%   0 IPC Managed Time

<output truncated>
```

251049

次の例では、割り込みの割合は CPU 使用率の割合に比べて低くなっています（82% に対して 5%）。CPU 使用率が高く割り込みの割合が比較的低い場合は、1 つ以上のシステム プロセスに時間がかかりすぎていることを表しています。この場合は、「アクティブなプロセスのデバッグ」（P.22）を参照してください。

```
Switch# show processes cpu sorted 5sec
CPU utilization for five seconds: 82%/5%; one minute: 40%; five minutes: 34%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
 217   135928429 493897689        275 45.68% 18.61% 16.78%   0 SNMP ENGINE
  47    61840574 480781517        128 23.80%  8.63%  7.43%   0 hrpc <-response
 158    58014186 265701225        218  1.11%  1.36%  1.35%   0 Spanning Tree
  46     1222030  67734870         18  0.47%  0.14%  0.08%   0 hrpc -> request
  75     1034724   8421764        122  0.15%  0.06%  0.02%   0 hpm counter proc
 223         125       157        796  0.15%  0.13%  0.03%   2 Virtual Exec
 213        2573       263       9783  0.15%  2.43%  0.71%   1 Virtual Exec
 150      578692   3251272        177  0.15%  0.02%  0.00%   0 CDP Protocol
 114     8436933   3227814       2613  0.15%  0.17%  0.16%   0 HRPC qos request
 105     1002819  96357752         10  0.15%  0.10%  0.06%   0 Hulc LED Process
  28      701287     68160      10288  0.15%  0.01%  0.00%   0 Per-minute Jobs
 215     9757808  42169987        231  0.15%  0.58%  0.56%   0 IP SNMP
  12           0         1          0  0.00%  0.00%  0.00%   0 IFS Agent Manage
  13           8     67388          0  0.00%  0.00%  0.00%   0 IPC

!<Output truncated>
```

# ネットワーク トラフィックの分析

割り込みの割合が高い場合、問題の根本的な原因は、CPU で受信しているパケット数が多すぎることです。この問題を解決するには、パケットの送信元を見つけて、フローを停止するか、スイッチのコンフィギュレーションを変更する必要があります。次の項を参照してください。

- 「システム プロセスおよびネットワーク パケット」（P.10）
- 「システム プロセスおよびパント パケット」（P.11）
- 「CPU で受信したネットワーク パケットの特定」（P.12）
- 「CPU へのネットワーク パケットの制限」（P.18）
- 「スイッチ ハードウェアからパントされたパケットの特定」（P.18）
- 「TCAM 使用率に関する問題の特定」（P.19）
- 「TCAM 使用率に関する問題の解決」（P.21）

## システム プロセスおよびネットワーク パケット

スイッチがパケットの管理に使用するシステム プロセスによって、CPU のフラッディングの原因となっているネットワーク パケットのタイプを特定できます。CPU の割り込みの割合が CPU 使用率全体の割合に比べて高い場合、最もアクティブなシステム プロセスを特定するには、**show processes cpu sorted 5sec** 特権 EXEC コマンドを入力します。CPU では、複数のパケット タイプを受信し、アクティブなシステム プロセスを複数実行している可能性があります。この出力には、最もアクティブなプロセスが先頭に示されます。最もアクティブなシステム プロセスが、ネットワーク パケットの受信に対応していると考えられます。

```
Switch# show processes cpu sorted 5sec
CPU utilization for five seconds: 64%/19%; one minute: 65%; five minutes: 70%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
 186    19472027  64796535        300 35.14% 37.50% 36.05%   0 IP Input
 192    24538871  82738840        296  1.11%  0.71%  0.82%   0 Spanning Tree
 458        5514       492      11207  0.63%  0.15%  0.63%   2 Virtual Exec
  61     3872439 169098902         22  0.63%  0.63%  0.41%   0 RedEarth Tx Mana
  99    10237319  12680120        807  0.47%  0.66%  0.59%   0 hpm counter proc
 131     4232087 224923936         18  0.31%  0.50%  1.74%   0 Hulc LED Process
 152     2032186   7964290        255  0.31%  0.21%  0.25%   0 PI MATM Aging Pr
 140    22911628  12784253       1792  0.31%  0.23%  0.26%   0 HRPC qos request
 250    27807274  62859001        442  0.31%  0.34%  0.34%   0 RIP Router

!<Output truncated>
```

表 1 に、一般的なシステム プロセスおよび関連するパケット タイプを示します。記載されているシステム プロセスのいずれかが CPU における最もアクティブなプロセスである場合は、そのプロセスによって、対応するネットワーク パケット タイプが CPU のフラッディングの原因になっていると考えられます。

**表 1　　ネットワーク パケット処理に関連するプロセス**

| システム プロセス名 | パケット タイプ |
|---|---|
| IP Input | IP パケット（ICMP を含む） |
| IGMPSN | IGMP スヌーピング パケット |
| ARP Input | IP ARP パケット |
| SNMP Engine | SNMP パケット |

パケットの送信元を見つけてトラブルシューティングを行う方法については、「CPU で受信したネットワーク パケットの特定」（P.12）を参照してください。

## システム プロセスおよびパント パケット

レイヤ 3 スイッチで、IP ルートが認識されない場合は、IP ルーティング用の IP パケットが CPU にパント（送信）されます。パント パケットは、割り込みレベルで処理されるので、CPU の過度なビジー状態の原因となる場合があります。コマンド出力で表示される割り込みの割合は高くても、表示されている中に最もアクティブなプロセスがない場合（表 1）や、CPU 使用率が高くなる原因となるだけのアクティブなプロセスがない場合は、出力例に示すように、ほとんどパント パケットが原因で CPU 使用率が高くなっていると考えられます。

```
Switch# show processes cpu sorted 5sec
CPU utilization for five seconds: 53%/28%; one minute: 48%; five minutes: 45%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
  78     461805 220334990          2 11.82% 11.53% 10.37%   0 HLFM address lea
 309   99769798   1821129      54784  5.27%  1.53%  1.39%   0 RIP Timers
 192   19448090  72206697        269  1.11%  0.87%  0.81%   0 Spanning Tree
 250   25992246  58973371        440  0.63%  0.27%  0.29%   0 RIP Router
  99    6853074  11856895        577  0.31%  0.46%  0.44%   0 hpm counter proc
 131    3184794 210112491         15  0.31%  0.13%  0.12%   0 Hulc LED Process
 140   20821662  11950171       1742  0.31%  0.28%  0.26%   0 HRPC qos request
 139    3166446   1498429       2113  0.15%  0.11%  0.11%   0 HQM Stack Proces
  67    2809714  11642483        241  0.15%  0.03%  0.00%   0 hrpc <- response
 223     449344  16515401         27  0.15%  0.03%  0.00%   0 Marvell wk-a Pow
  10          0         1          0  0.00%  0.00%  0.00%   0 Crash writer
  11     227226    666257        341  0.00%  0.00%  0.00%   0 ARP Input

!<Output truncated>
```

表 2 に、CPU がパント IP パケットの処理でビジー状態になっている場合の最もアクティブなシステム プロセスを示します。パント パケットの CPU 処理は、示されたプロセスには関連付けられていません。

**表 2　　パント パケット処理を示すプロセス**

| システム プロセス名 | パケット タイプ |
|---|---|
| HLFM address lea | IP 転送マネージャ プロセス |
| Check heaps | メモリ収集プロセス |
| Virtual Exec | Cisco IOS CLI プロセス |
| RedEarth Tx Mana | マイクロプロセッサ通信プロセス |
| hpm counter proc | 統計情報の収集 |

パント パケットのトラブルシューティング手順については、「スイッチ ハードウェアからパントされたパケットの特定」（P.18）を参照してください。

# CPU で受信したネットワーク パケットの特定

CPU に送信されているパケットのタイプを特定する次の技法は、相互補完的に使用したり、単独で使用したりすることができます。

- パケットは、各 CPU 受信キューにおいてカウントされます。このカウントを使用して、受信パケットのタイプを特定できます。「CPU 受信キューのパケット カウントのモニタリング」（P.12）を参照してください。
- **debug** 特権 EXEC コマンドを使用すると、CPU で受信したすべてのパケットをコンソールに出力できます。**debug** コマンドでは、各受信キューを個別にデバッグできます。「スイッチ CPU 受信キューからのパケットのデバッグ」（P.14）を参照してください。
- 送受信される IP パケットはすべてカウントされます。この情報は、特に数の多いパケットや急増しているパケットを特定するのに役立ちます。「IP トラフィック カウントのモニタリング」（P.17）を参照してください。

## CPU 受信キューのパケット カウントのモニタリング

特定のパケット タイプがスイッチのフラッディングの原因となっている場合、そのパケット タイプは適切な CPU キューに格納され、カウントされます。キューごとのパケット カウントを確認するには、**show controllers cpu-interface** 特権 EXEC コマンドを入力します。

```
Switch # show controllers cpu-interface
ASIC    Rxbiterr    Rxunder     Fwdctfix    Txbuflos    Rxbufloc    Rxbufdrain
------------------------------------------------------------------------------
ASIC0      0           0           0           0           0           0
ASIC1      0           0           0           0           0           0

cpu-queue-frames  retrieved  dropped    invalid    hol-block  stray
----------------- ---------- ---------- ---------- ---------- ----------
rpc               726325     0          0          0          0
stp               16108      0          0          0          0
ipc               56771      0          0          0          0
routing protocol  3949       0          0          0          0
L2 protocol       827        0          0          0          0
remote console    58         0          0          0          0
sw forwarding     0          0          0          0          0
host              0          0          0          0          0
broadcast         382        0          0          0          0
cbt-to-spt        0          0          0          0          0
igmp snooping     3567       0          0          0          0
icmp              11256      0          0          0          0
logging           0          0          0          0          0
rpf-fail          0          0          0          0          0
dstats            0          0          0          0          0
cpu heartbeat     322409     0          0          0          0
<output truncated>
```

また、輻輳のために廃棄される CPU バウンド パケットもカウントされます。各 CPU 受信キューには、パケット カウントの最大値が設定されています。受信キューが最大値に達すると、輻輳キュー宛てのパケットは廃棄されます。廃棄されたパケットは、キューごとにカウントされます。特定の CPU キューの廃棄カウントが増大している場合は、そのキューの使用頻度が高いことを意味します。

CPU 受信キュー廃棄カウントを確認し、パケットの廃棄カウントの多いキューを特定するには、**show platform port-asic stats drop** 特権 EXEC コマンドを入力します。このコマンドが **show controllers cpu-interface** コマンドほど有用ではないのは、出力に受信キューの名前ではなく番号が表示され、廃

棄数しか示されないためです。スーパーバイザに送信された CPU 受信キューのドロップ パケットはスイッチ ハードウェアで確認されるため、ドロップ パケットは、コマンド出力では Supervisor TxQueue Drop Statistics と呼ばれています。

```
Switch #show platform port-asic stats drop
Port-asic Port Drop Statistics - Summary
========================================
  RxQueue Drop Statistics Slice0
   RxQueue 0 Drop Stats Slice0: 0
   RxQueue 1 Drop Stats Slice0: 0
   RxQueue 2 Drop Stats Slice0: 0
   RxQueue 3 Drop Stats Slice0: 0
  RxQueue Drop Statistics Slice1
   RxQueue 0 Drop Stats Slice1: 0
   RxQueue 1 Drop Stats Slice1: 0
   RxQueue 2 Drop Stats Slice1: 0
   RxQueue 3 Drop Stats Slice1: 0

!<Output truncated>

  Port 27 TxQueue Drop Stats: 0

  Supervisor TxQueue Drop Statistics
    Queue  0: 0
    Queue  1: 0
    Queue  2: 0
    Queue  3: 0
    Queue  4: 0
    Queue  5: 0
    Queue  6: 0
    Queue  7: 0
    Queue  8: 0
    Queue  9: 0
    Queue 10: 0
    Queue 11: 0
    Queue 12: 0
    Queue 13: 0
    Queue 14: 0
    Queue 15: 0

! <Output truncated>
```

この出力の Supervisor TxQueue Drop Statistics のキュー番号の順序は、**show controllers cpu-interface** コマンド出力のキュー名の順序と同じです。たとえば、この出力の Queue 0 は、前の出力の rpc に対応しています。Queue 15 は cpu heartbeat に対応しています。以下も同様です。

統計情報はリセットされません。アクティブなキュー廃棄を確認するには、このコマンドを何回か入力します。コマンド出力では、他のドロップ統計情報も表示されますが、例では、その一部が省略されています。

CPU キューの詳細については、「CPU 受信キュー」（P.23）を参照してください。

## スイッチ CPU 受信キューからのパケットのデバッグ

ネットワークから受信したパケットは、CPU の 16 個の異なるキューに格納されます。CPU に送信されたパケットを特定するには、キューのデバッグをイネーブルにします。**show controllers cpu-interface** 特権 EXEC コマンドの出力を使用して、最初にデバッグを開始するキューを確認します。「CPU 受信キューのパケット カウントのモニタリング」（P.12）を参照してください。

**show controllers cpu-interface** コマンドの出力では、どのキューから開始してよいかわからない場合は、コンソールにデバッグ メッセージが数多く表示されるまで 1 つずつキューのデバッグをイネーブルにすることをお勧めします。CPU 受信キューごとにデバッグをオン/オフにするには、別々に **debug** コマンドを使用します。

受信キューのデバッグを開始する前に、次の手順に従って、他のアプリケーションからコンソールに出力されないようにし、システム ログ バッファを拡大（デフォルトのサイズの場合）して、デバッグ メッセージに詳細なタイムスタンプが出力されるよう設定します。デバッグ セッションが完了すると、デバッグ メッセージがシステム ログに記録されています。

特権 EXEC モードから始め、次の手順に従って CPU キューのデバッグの準備を整えます。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **configure terminal** | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 2 | **no logging console** | コンソール ターミナルへのロギングをディセーブルにします。 |
| ステップ 3 | **logging buffered 128000** | ローカル バッファへのシステム メッセージ ロギングをイネーブルにし、バッファ サイズを 12800 バイトに設定します。 |
| ステップ 4 | **service timestamps debug datetime msecs localtime** | デバッグ メッセージまたはシステム ロギング メッセージにタイムスタンプを適用するようにシステムを設定します。 |
| ステップ 5 | **exit** | 特権 EXEC モードに戻ります。 |

これで、**undebug all** 特権 EXEC コマンドを入力して、コンソールのパケット フラッディングを停止できます。コマンド プロンプトが表示されていなくても、**undebug all** コマンドはいつでも使用できます。コマンドを入力した後、バッファに格納されたデバッグ メッセージが処理され、デバッグ メッセージ バッファが空になるまでに少し時間がかかります。

デバッグ メッセージにより、パケット フラッディングの原因を特定できます。この方法は、パケットがすべて同じタイプになっている場合や同じ送信元から発信されている場合に役に立ちます。

次の例では、コンソールにメッセージが大量に表示されるまで CPU キューを 1 つずつオンにしています。

```
Switch# debug platform cpu-queues ?
  broadcast-q          Debug packets received by Broadcast Q
  cbt-to-spt-q         Debug packets received by cbt-to-spt Q
  cpuhub-q             Debug packets received by CPU heartbeat Q
  host-q               Debug packets received by host Q
  icmp-q               Debug packets received by ICMP Q
  igmp-snooping-q      Debug packets received by IGMP snooping Q
  layer2-protocol-q    Debug packets received by layer2 protocol Q
  logging-q            Debug packets received by logging Q
  remote-console-q     Debug packets received by remote console Q
  routing-protocol-q   Debug packets received by routing protocol Q
  rpffail-q            Debug packets received by RPF fail Q
  software-fwd-q       Debug packets received by software forward Q
  stp-q                Debug packets received by STP Q

Switch# debug platform cpu-queues broadcast-q
debug platform cpu-queue broadcast-q debugging is on
Switch#
Switch# debug platform cpu-queues cbt-to-spt-q
debug platform cpu-queue cbt-sbt-q debugging is on
Switch#
Switch# debug platform cpu-queues cpuhub-q
debug platform cpu-queue cpuhb debugging is on
Switch#
Switch# debug platform cpu-queues host-q
debug platform cpu-queue host-q debugging is on
Switch#
*Mar  2 22:48:06.227: L2B-Q:Dropped Null L3Hwidb: Local Port Fwding L3If:
L2If:GigabitEthernet4/0/23 DI:0x6F5, LT:1, Vlan:139   SrcGPN:185, SrcGID:185,
ACLLogIdx:0x4, MacDA:ffff.ffff.ffff, MacSA: 0009.9b00.edfe   ARP:
00010800_06040001_00099B00_EDFEAC14_8B850000_00000000_AC148B81
   TPFFD:E10000B9_008B008B_00800044-000406F5_1A1A0000_00000000

Switch# debug platform cpu-queues icmp-q
debug platform cpu-queue icmp-q debugging is on
Switch#
*Mar  2 22:48:16.947: ICMP-Q:Dropped Throttle timer not awake: Remote Port Blocked
L3If:Vlan200 L2If:GigabitEthernet1/0/3 DI:0xB4, LT:7, Vlan:200   SrcGPN:3, SrcGID:3,
ACLLogIdx:0x0, MacDA:001d.46be.7541, MacSA: 0000.0300.0101   IP_SA:10.10.200.1
IP_DA:10.10.200.5 IP_Proto:1
   TPFFD:ED000003_008B00C8_00B00222-000000B4_00060000_03090000

*Mar  2 22:48:16.947: ICMP-Q:Dropped Throttle timer not awake: Remote Port Blocked
L3If:Vlan200 L2If:GigabitEthernet1/0/3 DI:0xB4, LT:7, Vlan:200   SrcGPN:3, SrcGID:3,
ACLLogIdx:0x0, MacDA:001d.46be.7541, MacSA: 0000.0300.0101   IP_SA:10.10.200.1
IP_DA:10.10.200.5 IP_Proto:1
   TPFFD:ED000003_008B00C8_00B00222-000000B4_00050000_03090000

*Mar  2 22:48:16.947: ICMP-Q:Dropped Throttle timer not awake: Remote Port Blocked
L3If:Vlan200 L2If:GigabitEthernet1/0/3 DI:0xB4, LT:7, Vlan:200   SrcGPN:3, SrcGID:3,
ACLLogIdx:0x0, MacDA:001d.46be.7541, MacSA: 0000.0300.0101   IP_SA:10.10.200.1
IP_DA:10.10.200.5 IP_Proto:1
   TPFFD:ED000003_008B00C8_00B00222-000000B4_00040000_03090000
```

1 つのパケットをホストキューで受信しました。これは正常です。

205496

この例におけるアクションのシーケンスは、次のとおりです。

- **debug platform cpu-queue host-q** コマンドが入力された後に、1 つのパケットが受信されました。これは正常です。
- 次のコマンド **debug platform cpu-queue icmp-q** が入力されたときに、フラッディングが始まりました。icmp-q で受信したパケットはすべて同じものです。3 つのパケットのみが表示されます。したがって、CPU では、ICMP パケットの受信でフラッディングが発生しています。
- このパケットの VLAN（200）および送信元 MAC アドレス（0000.0300.0101）（太字表示）を含む、送信元について出力を調べます。

```
 *Mar  2 22:48:16.947: ICMP-Q:Dropped Throttle timer not awake: Remote Port Blocked
L3If:Vlan200 L2If:GigabitEthernet1/0/3 DI:0xB4, LT:7, Vlan:200   SrcGPN:3, SrcGID:3,
ACLLogIdx:0x0, MacDA:001d.46be.7541, MacSA: 0000.0300.0101   IP_SA:10.10.200.1
IP_DA:10.10.200.5 IP_Proto:1
   TPFFD:ED000003_008B00C8_00B00222-000000B4_00040000_03090000
```

- VLAN に対して **show mac address-table** 特権 EXEC コマンドを入力して、MAC アドレス テーブルを参照し、この MAC アドレスを学習したインターフェイスを検索します。この出力では、ギガビット イーサネット 1/0/3 インターフェイス（太字表示）でパケットが受信されていることを示しています。

```
Switch# show mac address-table dynamic vlan 200
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
200    0000.0300.0101    DYNAMIC      Gi1/0/3

!<Output truncated>
```

1 つのフローが CPU のフラッディングの原因となっている場合は、さまざまなパケット タイプに対してこうした手順を実行できます。コンソールにメッセージが大量に表示されるまで、さまざまな CPU キューのデバッグをイネーブルにし続けます。CPU キューの詳細については、「CPU 受信キュー」（P.23）を参照してください。

特権 EXEC モードから始め、次の手順に従ってシステム ログのデバッグ メッセージを表示します。

| | コマンド | 目的 |
|---|---|---|
| ステップ 1 | **terminal length 0** | 現在のセッションでの端末画面の行数を 0 に設定します。 |
| ステップ 2 | **show logging** | 標準システム ロギング バッファの内容を表示します。 |
| ステップ 3 | **terminal length 30** | 端末サイズを 30 に設定するか、元の値にリセットします。 |
| ステップ 4 | **exit** | CLI を終了します。 |

（注） デバッグする前にシステム ログ バッファの拡大またはタイムスタンプの追加によってコンフィギュレーションを変更した場合は、デバッグが完了したら、こうした設定をデフォルトのコンフィギュレーションに戻すことを考慮してください。

## IP トラフィック カウントのモニタリング

CPU で受信した IP パケット タイプはすべてカウントされます。こうしたパケット カウントには、ハードウェアでスイッチングまたはルーティングされた IP パケットは含まれません。IP パケット タイプのカウントを表示するには、**show ip traffic** 特権 EXEC コマンドを入力します。この出力では、IP パケットがレイヤ 4 タイプ（ICMP、マルチキャスト、ICMP、ARP）に分類されます。特定のカウントが急増している場合は、その IP パケット タイプが CPU のフラッディングの原因となっていると考えられます。

```
Switch# show ip traffic
IP statistics:
  Rcvd:  12420483 total, 840467 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         0 unknown protocol, 222764 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 couldn't fragment
  Bcast: 0 received, 0 sent
  Mcast: 0 received, 0 sent
  Sent:  834640 generated, 928020828 forwarded
  Drop:  189206 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop
         0 options denied, 0 source IP address zero

ICMP statistics:
  Rcvd: 0 format errors, 0 checksum errors, 834640 redirects, 0 unreachable
        0 echo, 0 echo reply, 0 mask requests, 0 mask replies, 0 quench
        0 parameter, 0 timestamp, 0 info request, 0 other
        0 irdp solicitations, 0 irdp advertisements
  Sent: 834640 redirects, 0 unreachable, 0 echo, 0 echo reply
        0 mask requests, 0 mask replies, 0 quench, 0 timestamp
        0 info reply, 0 time exceeded, 0 parameter problem
        0 irdp solicitations, 0 irdp advertisements

TCP statistics:
  Rcvd: 5830 total, 0 checksum errors, 0 no port
  Sent: 0 total

UDP statistics:
  Rcvd: 0 total, 0 checksum errors, 0 no port
  Sent: 0 total, 0 forwarded broadcasts
PIMv2 statistics: Sent/Received
  Total: 0/0, 0 checksum errors, 0 format errors
  Registers: 0/0 (0 non-rp, 0 non-sm-group), Register Stops: 0/0,  Hellos: 0/0
  Join/Prunes: 0/0, Asserts: 0/0, grafts: 0/0
  Bootstraps: 0/0, Candidate_RP_Advertisements: 0/0
  State-Refresh: 0/0

IGMP statistics: Sent/Received
  Total: 0/0, Format errors: 0/0, Checksum errors: 0/0
  Host Queries: 0/0, Host Reports: 0/0, Host Leaves: 0/0
  DVMRP: 0/0, PIM: 0/0

EIGRP-IPv4 statistics:
  Rcvd: 0 total
  Sent: 0 total

ARP statistics:
  Rcvd: 0 requests, 0 replies, 0 reverse, 0 other
  Sent: 92 requests, 87 replies (0 proxy), 0 reverse
  Drop due to input queue full: 444087
```

## CPU へのネットワーク パケットの制限

問題のあるネットワーク パケットによる影響が CPU 使用率に及ばないようにするには、そうしたパケットを入力インターフェイスで阻止します。

- イーサネット ブロードキャストまたはマルチキャスト パケット ストームを制限するには、**storm-control** {**broadcast** | **multicast** | **unicast**} **level** {*level* [*level-low*] | **bps** *bps* [*bps-low*] | **pps** *pps* [*pps-low*]} インターフェイス レベル コンフィギュレーション コマンドを使用します。スイッチのソフトウェア コンフィギュレーション ガイドの「Configuring Port-Based Traffic Control」の章を参照してください。
- CPU 使用率が高くなる根本的な原因がレイヤ 2 のループにある場合は、スパニングツリー コンフィギュレーションが問題となっている可能性があります。スイッチのソフトウェア コンフィギュレーション ガイドの「Configuring STP」の章を参照してください。
- トラフィック ポリシングにより、スイッチに入力されるパケット数を制限できます。ポリシングでは、入力トラフィックの拒否、指定ビット/秒レートへの入力トラフィックの制限、一部のトラフィックの許可と同時に他のトラフィックの制限を実施することができます。MAC アドレス、IPv4 ヘッダー、IPv6 ヘッダー（スイッチで IPv6 がサポートされている場合）、またはレイヤ 4 ポート番号でトラフィックをポリシングできます。スイッチのソフトウェア コンフィギュレーション ガイドの「Configuring Network Security with ACLs」、「Configuring IPv6 ACLs」（スイッチでサポートされている場合）、および「Configuring QoS」の章を参照してください。
- レイヤ 3 スイッチの CPU 使用率に IP ARP パケットによる影響が及ばないようにするには、Dynamic ARP Inspection（DAI; ダイナミック ARP インスペクション）を設定し、**ip arp inspection limit** {**rate** *pps* [**burst interval** *seconds*] | **none**} インターフェイス コンフィギュレーション コマンドを入力して、レート制限機能を使用します。スイッチのソフトウェア コンフィギュレーション ガイドの「Configuring Dynamic ARP Inspection」の章を参照してください。

## スイッチ ハードウェアからパントされたパケットの特定

通常のレイヤ 3 スイッチの動作の一環として、スイッチ ハードウェアに IP ルートがプログラミングされていない場合は、IP ルーティングに備えて IP パケットが CPU にパントされます。IP パケットが不定期に CPU にパントされることは正常であり予想の範囲ですが、パントされる IP パケットが多すぎる場合は、CPU が過度にビジー状態になっています。

IP ルーティングに備えて CPU にパントされる IP パケットはすべてカウントされます。各 CPU 受信キューに格納されるパケット数を確認するには、**show controllers cpu** 特権 EXEC コマンドを使用します。スイッチ ハードウェアでパケットがパントされているときは、`sw forwarding` という名前の行のカウントが増えていきます。`sw forwarding` のカウントが急増しているかどうかを確認するには、このコマンドを何回か入力します。

```
Switch# show controllers cpu-interface
ASIC    Rxbiterr   Rxunder    Fwdctfix   Txbuflos   Rxbufloc   Rxbufdrain
------------------------------------------------------------------------
ASIC0     0          0          0          0          0          0
ASIC1     0          0          0          0          0          0

cpu-queue-frames  retrieved  dropped    invalid    hol-block  stray
----------------- ---------- ---------- ---------- ---------- ----------
rpc               2811788    0          0          0          0
stp               944641     0          0          0          0
ipc               280645     0          0          0          0
routing protocol  813536     0          0          0          0
L2 protocol       8787       0          0          0          0
remote console    2808       0          0          0          0
sw forwarding     65614320   0          0          0          0
host              25         0          0          0          0
broadcast         794570     0          0          0          0
cbt-to-spt        0          0          0          0          0
```

```
igmp snooping     18941      0          0          0          0
icmp              0          0          0          0          0
logging           0          0          0          0          0
rpf-fail          0          0          0          0          0
dstats            0          0          0          0          0
cpu heartbeat     1717274    0          0          0          0
```

また、**show platform ip unicast statistics** 特権 EXEC コマンドを使用して、同様にパント パケットに関する情報を表示することもできます。パント IP パケットは、CPUAdj（この例の太字表示）としてカウントされます。

```
Switch# show platform ip unicast statistics
Global Stats:
        HWFwdLoc:0 HWFwdSec:0 UnRes:0 UnSup:0 NoAdj:0
        EncapFail:0 CPUAdj:1344291253 Null:0 Drop:0

Prev Global Stats:
        HWFwdLoc:0 HWFwdSec:0 UnRes:0 UnSup:0 NoAdj:0
        EncapFail:0 CPUAdj:1344291253 Null:0 Drop:0
```

こうした統計情報は 2 ～ 3 秒ごとに更新されます。CPUAdj カウントの変化を確認するには、このコマンドを何回か入力します。CPUAdj カウントが急増している場合は、IP ルーティングに備えて多くの IP パケットが CPU に転送されています。

## TCAM 使用率に関する問題の特定

レイヤ 3 スイッチでは、TCAM を使用して IP ルーティング データベースが格納されます。レイヤ 3 ルーティング情報用の TCAM スペースは限られています。このスペースが満杯になると、Cisco IOS で学習した新たなルートを TCAM にプログラミングできなくなります。スイッチ ハードウェアで受信した IP パケットの宛先 IP アドレスが TCAM に存在しない場合、その IP パケットは CPU にパントされます。

TCAM が満杯かどうかを確認するには、**show platform tcam utilization** 特権 EXEC コマンドを入力します。

```
Switch# show platform tcam utilization

CAM Utilization for ASIC# 0                     Max            Used
                                              Masks/Values    Masks/values
 Unicast mac addresses:                        6364/6364         31/31
 IPv4 IGMP groups + multicast routes:          1120/1120          1/1
 IPv4 unicast directly-connected routes:       6144/6144          4/4
 IPv4 unicast indirectly-connected routes:     2048/2048       2047/2047
 IPv4 policy based routing aces:                452/452          12/12
 IPv4 qos aces:                                 512/512          21/21
 IPv4 security aces:                            964/964          30/30

Note: Allocation of TCAM entries per feature uses
a complex algorithm. The above information is meant
to provide an abstract view of the current TCAM utilization
```

この例では、最大値 2048 に対して 2047 を使用中と表示されていますが、IP indirectly-connected routes リソースは満杯です。スイッチの TCAM が満杯の場合は、TCAM に存在する宛先 IP アドレスのパケットのみがルーティングされます。TCAM に宛先が存在しない他のすべての IP パケットは、CPU にパントされます。**show controllers cpu-interface** コマンドの出力から、TCAM が満杯で sw forwarding カウントが増大している場合は、パント パケットが CPU 使用率が高くなる原因になっているということです。

Cisco IOS では、ルーティング プロトコル（BGP、RIP、OSPF、EIGRP、IS-IS など）から、また静的に設定されたルートから、ルートについて学習します。**show platform ip unicast counts** 特権 EXEC コマンドを入力すると、こうしたルートのうち TCAM に正しくプログラミングされていないルートの数を確認できます。

```
Switch# show platform ip unicast counts
# of HL3U fibs 2426
# of HL3U adjs 4
# of HL3U mpaths 0
# of HL3U covering-fibs 0
# of HL3U fibs with adj failures 0
Fibs of Prefix length 0, with TCAM fails: 0
Fibs of Prefix length 1, with TCAM fails: 0
Fibs of Prefix length 2, with TCAM fails: 0
Fibs of Prefix length 3, with TCAM fails: 0
Fibs of Prefix length 4, with TCAM fails: 0
Fibs of Prefix length 5, with TCAM fails: 0
Fibs of Prefix length 6, with TCAM fails: 0
Fibs of Prefix length 7, with TCAM fails: 0
Fibs of Prefix length 8, with TCAM fails: 0
Fibs of Prefix length 9, with TCAM fails: 0
Fibs of Prefix length 10, with TCAM fails: 0
Fibs of Prefix length 11, with TCAM fails: 0
Fibs of Prefix length 12, with TCAM fails: 0
Fibs of Prefix length 13, with TCAM fails: 0
Fibs of Prefix length 14, with TCAM fails: 0
Fibs of Prefix length 15, with TCAM fails: 0
Fibs of Prefix length 16, with TCAM fails: 0
Fibs of Prefix length 17, with TCAM fails: 0
Fibs of Prefix length 18, with TCAM fails: 0
Fibs of Prefix length 19, with TCAM fails: 0
Fibs of Prefix length 20, with TCAM fails: 0
Fibs of Prefix length 21, with TCAM fails: 0
Fibs of Prefix length 22, with TCAM fails: 0
Fibs of Prefix length 23, with TCAM fails: 0
Fibs of Prefix length 24, with TCAM fails: 0
Fibs of Prefix length 25, with TCAM fails: 0
Fibs of Prefix length 26, with TCAM fails: 0
Fibs of Prefix length 27, with TCAM fails: 0
Fibs of Prefix length 28, with TCAM fails: 0
Fibs of Prefix length 29, with TCAM fails: 0
Fibs of Prefix length 30, with TCAM fails: 0
Fibs of Prefix length 31, with TCAM fails: 0
Fibs of Prefix length 32, with TCAM fails: 693
Fibs of Prefix length 33, with TCAM fails: 0
```

この出力では 693 個のエラーが示されています。この統計情報を使用して、この時点でネットワーク内のアドバタイズされているルートを保持するのにさらに必要な TCAM リソースの数を確認できます。

各ルーティング プロトコルで使用されたルート エントリの数を表示するには、**show ip route summary** 特権 EXEC コマンドを入力します。

```
Switch# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       5           0           320         760
static          0           0           0           0
rip             0           2390        152960      363280
internal        1                                   1172
Total           6           2390        153280      365212
Switch#
```

# TCAM 使用率に関する問題の解決

次の解決方法をお勧めします。

- SDM テンプレートの変更
- IP ルートの最適化

## SDM テンプレートの変更

Switch Database Management（SDM; スイッチ データベース管理）テンプレートにより、限られた TCAM リソースがさまざまな転送タイプに割り当てられます。TCAM 使用率に関する問題を解決するには、スイッチ アプリケーション用の適切な SDM テンプレートを選択する必要があります。

スイッチのアクティブな SDM テンプレートを確認するには、**show sdm prefer** 特権 EXEC コマンドを入力します。

```
Switch# show sdm prefer
 The current template is "desktop default" template.
 The selected template optimizes the resources in
 the switch to support this level of features for
 8 routed interfaces and 1024 VLANs.

  number of unicast mac addresses:                  6K
  number of IPv4 IGMP groups + multicast routes:    1K
  number of IPv4 unicast routes:                    8K
    number of directly-connected IPv4 hosts:        6K
    number of indirect IPv4 routes:                 2K
  number of IPv4 policy based routing aces:         0
  number of IPv4/MAC qos aces:                      0.5K
  number of IPv4/MAC security aces:                 1K
```

このスイッチのデフォルトのテンプレートでは、TCAM 内の 2048 個の間接レイヤ 3 ルートしか許可されていません。さらに TCAM リソースを間接レイヤ 3 ルートに割り当てるには、他の TCAM リソースをいくつか削減する必要があります。IP ルーティングに備えてより多くのリソースを確保するテンプレートに変更するには、**sdm prefer** *template-name* グローバル コンフィギュレーション コマンドを使用します。

スイッチに使用可能な SDM テンプレートのリストを確認するには、**show sdm templates all** 特権 EXEC コマンドを入力します。

```
Switch# show sdm templates all
 Id  Type        Name
  0  desktop     desktop default
  1  desktop     desktop vlan
  2  desktop     desktop routing
  3  aggregator  aggregate default
  4  aggregator  aggregate vlan
  5  aggregator  aggregate routing
  6  desktop     desktop routing pbr
  8  desktop     desktop IPv4 and IPv6 default
  9  desktop     desktop IPv4 and IPv6 vlan
 10  aggregator  aggregate IPv4 and IPv6 default
 11  aggregator  aggregate IPv4 and IPv6 vlan
 12  desktop     desktop access IPv4
 13  aggregator  aggregator access IPv4
 14  desktop     desktop IPv4 and IPv6 routing
 15  aggregator  aggregator IPv4 and IPv6 routing
 16  desktop     desktop IPe
 17  aggregator  aggregator IPe
```

（注） スイッチで使用可能なテンプレートおよび各テンプレートの予約 TCAM リソースを確認するには、スイッチのソフトウェア コンフィギュレーション ガイドの「Configuring SDM Templates」の章を参照してください。

### IP ルートの最適化

レイヤ 3 スイッチの SDM テンプレートの変更が不可能な場合や現実的でない場合は、集約ルートを使用するかルートをフィルタリングすることによって、TCAM 内のルートの数を減らすことができます。

集約ルートを使用すると、ルーティング テーブル サイズが小さくなります。ピア ルータで集約ルートをイネーブルにします。集約ルートは、RIP および EIGRP のデフォルトではイネーブルになっており、OSPF のデフォルトではディセーブルになっています。集約ルートの詳細については、ソフトウェア コンフィギュレーション ガイドの「Configuring IP Unicast Routing」の章を参照してください（レイヤ 3 スイッチのみ）。

ルート フィルタリングを使用すると、不要なルートが TCAM にプログラミングされないようにすることができます。OSPF ルート フィルタリングの詳細については、次の URL にある機能ガイドを参照してください。
http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/routmap.html

## アクティブなプロセスのデバッグ

CPU 使用率の割合が高く、割り込みの割合が低い場合、CPU 使用率が高くなる原因は、CPU リソースを消費する 1 つ以上のシステム プロセスにあります。この状況は、CPU 使用率が高いことがネットワーク パケットの受信に起因する場合ほど一般的ではありません。システム プロセスが CPU リソースの大半を消費している場合、通常は、イベントからのトリガーによってそのプロセスがアクティブになっています。異常なイベントが発生していないか syslog を確認します。

表 3 に、CPU リソースの消費の割合が正常なプロセスと高いプロセスの例、それに対して考えられる根本的な原因、および推奨措置を示します。

**表 3　CPU リソースを消費するシステム プロセス**

| プロセス名 | アクティブなリソースの割合 | | 考えられる根本的な原因 | 推奨措置 |
|---|---|---|---|---|
| | 正常 | 高 | | |
| Hulc LED Process | 0 ～ 2% | 24 ポート以下：<br>5% 超<br>48 ポート：8% 超 | 物理リンク フラッピング。 | 物理リンクの喪失や接続の取得が生じていないか syslog を確認します。 |
| Inline Power Twt | 0 | 5% 超 | 電源不良。 | 電源コントローラの障害が報告されていないか syslog を確認します。 |
| HACL | 0 | 50% 超 | 短期間にスイッチで設定した ACL が多すぎます。ACL が自動で（スクリプトから）適用されている場合に、このようになります。 | SDM テンプレートの変更を検討してください。 |
| SNMP Engine | 0 | 40% 超 | 「SNMP Engine プロセス」（P.23）を参照してください。 | |

## SNMP Engine プロセス

SNMP Engine システム プロセスは、スイッチが SNMP クエリーを受信している場合にのみアクティブになります。必要な CPU 時間は、受信した SNMP クエリー パケット数に正比例します。受信した各 SNMP クエリー パケットは、SNMP Engine システム プロセスに転送される前に割り込みレベルで処理されます。SNMP Engine プロセスがビジー状態の場合、**show processes cpu sorted** コマンドの出力に表示される割り込みの割合の中には、ゼロでない割り込みの割合もいくつか含まれます。割り込みの割合は、SNMP Engine システム プロセスによって利用される CPU の割合と比べるとわずかです。

SNMP Engine システム プロセスの CPU 使用率を評価する場合は、スイッチの基準 CPU 使用率を特定することが重要です。通常は、スイッチが一定の時間間隔で SNMP クエリーを受信し、SNMP Engine システム プロセスがクエリーの処理に CPU リソースを消費します。この間隔は、**show processes cpu history** コマンドの出力に表示されます。「CPU 使用率が高いことが問題となっている場合」（P.3）の例を参照してください。

次のような場合には、SNMP Engine システム プロセスが原因で CPU 使用率が非常に高くなることがあります。

- 複数のサーバが同時に SNMP クエリーを実行する。
- スイッチでのフラッシュ ファイル システムの SNMP クエリー。フラッシュ ファイルへのアクセスは、SNMP Gets または SNMP GetNext 操作を目的とした CPU 中心の操作です。
- 全部または一部の SNMP MIB Walk。

# 役立つ情報

## CPU 受信キュー

スイッチ ハードウェアで CPU に送信されるパケットは、パケット タイプに応じて 16 個の CPU キューのいずれかに格納されます。各キューは優先順位が設定されているので、優先順位の低いキューよりも先に優先順位の高いキューに格納されます。各キューには、キューに応じたパケットを保持するためにハードウェアでメモリが確保されているため、1 つのキューまたはパケット タイプで使用可能なすべてのメモリを使用できるわけではありません。

CPU キューおよびその用途は、次のとおりです。

- rpc：リモート プロシージャ コール。シスコのシステム プロセスでスタック間通信に使用されます。
- stp：スパニングツリー プロトコル。独自のキューを備えたレイヤ 2 プロトコル。
- ipc：プロセス間通信。シスコのシステム プロセスでスタック間通信に使用されます。
- routing protocol：他のネットワーク デバイスで受信されるルーティング プロトコル パケットに使用されます。
- L2 protocol：LACP、UDLD などのプロトコル パケットに使用されます。
- remote console：スタック マスター スイッチで **session** *switch-number* 特権 EXEC コマンドを入力して別のスイッチ メンバーのコンソールを開くときのパケットに使用されます。
- sw forwarding：ルーティングするために CPU のハードウェアでパントされるパケットに使用されます。
- host：宛先 IP アドレスが任意のスイッチ IP アドレスと一致するパケットに使用されます。IP ブロードキャスト パケットにも使用されます。
- broadcast：レイヤ 2 ブロードキャスト パケットを受信します。

- cbt-to-spt：PIM_SM のマルチキャスト パケットを受信します。
- igmp snooping：IGMP パケットのキュー。
- icmp：ICMP リダイレクト パケットのキュー。
- logging：ACL ロギング用にハードウェアで生成されたパケットの受信に使用されます。
- rpf fail：リバース パス フォワーディング障害用のキュー。
- dstats：統計情報をドロップします。正常動作時には使用されません。
- cpu heartbeat：自身に送信したパケットを受信する CPU で使用されます。

# CPU トラブルシューティングに使用されるコマンド

**表 4　CPU トラブルシューティング用の show および debug コマンド**

| コマンド | 目的 | 用途 |
|---|---|---|
| **show controllers cpu-interface** | すべての CPU 受信キューのパケット カウントを表示する。 | CPU のフラッディングの原因となっているパケットのタイプを特定します。 |
| **show ip route summary** | 各プロトコルで使用されるルート エントリの数を表示する。 | IP ルーティングにさらに TCAM リソースが必要かどうかを確認する場合に使用します。 |
| **show ip traffic** | スイッチで受信した IP パケット タイプのカウントを表示する。 | あるパケット タイプが急増している場合は、そのパケット タイプが IP スタックのフラッディングの原因になっていることを示します。 |
| **show platform ip unicast counts** | TCAM にプログラミングされていないルートを表示する。 | 現在のネットワーク ルートを維持するためにさらにどのくらいの TCAM リソースが必要であるかを特定する場合に使用します。 |
| **show platform ip unicast statistics** | *CPUAdj* 出力でパント パケット数を示す。 | このコマンドを何回か入力します。*CPUAdj* の値が急増する場合は、スイッチ ハードウェアからパケットがパントされています。 |
| **show platform port-asic stats drop** | 輻輳のために廃棄される CPU パケット数を表示する。 | フラッディングのためにパケットをドロップしている CPU 受信キューを特定します。 |
| **show platform tcam utilization** | TCAM の最大容量および使用状況を表示する。 | TCAM が満杯かどうかを確認します。*IPv4 unicast indirectly-connected routes* 出力が最大になっている場合、IP ルーティング データベースは満杯であり、他の IP アドレスへのパケットはパントされます。 |
| **show processes cpu history** | 過去 60 秒、60 分、および 72 時間の CPU 使用率の履歴を表示する。 | 基準 CPU 使用率を確認し、急増が発生している時期を特定します。 |
| **show processes cpu sorted [5sec]** | CPU 使用率および割り込みに費やされた CPU 時間の割合を表示し、CPU 使用率の順に最もアクティブなシステム プロセスを示す。 | CPU 使用率の問題の原因が過剰なネットワーク パケットにあるか、スイッチで稼動するアクティブなシステム プロセスにあるかを特定します。 |
| **show sdm templates all** | スイッチで使用可能な SDM テンプレートを示す。 | 必要に応じて TCAM リソースを再割り当てすることができるかどうかを確認する場合に入力します。 |
| **debug platform cpu-queue** *queue* | CPU キューをデバッグする。 | 問題と思われる CPU 受信キューごとにこのコマンドを入力します。コンソールには、問題のあるキューが大量に表示されます。 |

## その他のドキュメント

Cisco.com にある別のドキュメントでは、Catalyst 3750 スイッチでの使用率が高いことに関する特定の問題に重点を置いています。ただし、その内容は他のスイッチにも当てはまります。『*Catalyst 3750 Series Switches High CPU Utilization Troubleshooting*』を参照してください。

## 未解決の問題

このマニュアルのトラブルシューティング手順では CPU 使用率が高くなる根本的な原因を特定できない場合は、Technical Assistance Center（TAC）に連絡してください。技術サポート エンジニアは、お客様がデバッグ作業で収集した情報を確認する必要があります。問題解決の時間を短縮するために、シスコ テクニカル サポートに連絡するときにはこの情報を用意しておいてください。

（注） テクニカル サポートのリンクについては、次の項を参照してください。

# マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『*What's New in Cisco Product Documentation*』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』は RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。