



パケットフローの問題のトラブルシューティング

- パケットフローの問題 (1 ページ)
- インバンドパケット統計の監視 (2 ページ)
- ファブリック接続コマンド (3 ページ)
- パケットトレーサでパケットフローをトラブルシューティング (6 ページ)

パケットフローの問題

パケットは次の理由でドロップされる可能性があります。

- ソフトウェアスイッチのパケットは、コントロールプレーンのポリシー設定 (CoPP) が原因でドロップされる可能性があります。
- ハードウェアスイッチのパケットは、帯域幅の制限により、ハードウェアによってドロップされる可能性があります。

Cisco NX-OS リリース 10.3(1)F 以降、以下の CLI が、Cisco Nexus 9300 および 9500 Cloud Scale スイッチでサポートされます。

- **show hardware internal statistics module-all all** : アクティブなモジュールの統計を表示します。
- **show hardware internal statistics module <module-no> all** : スーパーバイザからの特定のアクティブモジュールの統計情報を表示します。

レート制限によってドロップされたパケット

show hardware rate-limit コマンドを使用し、レート制限のためにパケットがドロップされているかどうかを確認します。

```
switch(config)# show hardware rate-limit module 1
Units for Config: packets per second
```

■ CoPPのためにドロップされたパケット

```
Allowed, Dropped & Total: aggregated since last clear counters

Rate Limiter Class           Parameters
-----
access-list-log               Config    : 100
                                Allowed   : 0
                                Dropped   : 0
                                Total    : 0
```

CoPPのためにドロップされたパケット

show policy-map interface control-plane コマンドを使用し、コマンドを使用して、パケットが CoPP によってドロップされているかどうかを確認します。

```
switch# show policy-map interface control-plane
  class-map copp-system-p-class-exception (match-any)
    match exception ip option
    match exception ip icmp unreachable
    match exception ttl-failure
    match exception ipv6 option
    match exception ipv6 icmp unreachable
    match exception mtu-failure
    set cos 1
    police cir 200 pps , bc 32 packets

    module 27 :
      transmitted 0 packets;
      dropped 0 packets;

    module 28 :
      transmitted 0 packets;
      dropped 0 packets;
```

インバンドパケット統計の監視

show hardware internal cpu-mac inband counters コマンドを使用し、コマンドを使用して、スーパーバイザモジュール、ファブリックモジュール、およびラインカードのインバンドパケット統計情報を表示します。

```
switch# show hardware internal cpu-mac inband counters
eth2 counters:
eth2      Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          BROADCAST MULTICAST  MTU:9400  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth3 counters:
eth3      Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
          inet6 addr: fe80::200:ff:fe01:1b01/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400  Metric:1
          RX packets:425432 errors:0 dropped:0 overruns:0 frame:0
          TX packets:352432 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:253284953 (241.5 MiB)  TX bytes:249647978 (238.0 MiB)
```

```

ps-inb counters:
ps-inb  Link encap:Ethernet  HWaddr 00:00:00:01:1b:01
        inet6 addr: fe80::200:ff:fe01:1b01/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400 Metric:1
          RX packets:128986 errors:0 dropped:0 overruns:0 frame:0
          TX packets:129761 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:221538103 (211.2 MiB)  TX bytes:227158091 (216.6 MiB)

switch# slot 22 show hardware internal cpu-mac inband counters
inband0 counters:
inband0  Link encap:Ethernet  HWaddr 00:00:00:01:16:03
        inet addr:127.2.2.22  Bcast:127.2.255.255  Mask:255.255.0.0
        inet6 addr: fe80::200:ff:fe01:1603/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9676 Metric:1
          RX packets:147425 errors:0 dropped:0 overruns:0 frame:0
          TX packets:147470 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:532
          RX bytes:15479625 (14.7 MiB)  TX bytes:14898335 (14.2 MiB)
          Interrupt:10

knet0_0 counters:
knet0_0  Link encap:Ethernet  HWaddr 02:10:18:e1:6f:50
        inet6 addr: fe80::10:18ff:fe01:6f50/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400 Metric:1
          RX packets:36 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:6 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3816 (3.7 KiB)  TX bytes:0 (0.0 B)

knet0_1 counters:
knet0_1  Link encap:Ethernet  HWaddr 02:10:18:e1:6f:51
        inet6 addr: fe80::10:18ff:fe01:6f51/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9400 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:6 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

```

ファブリック接続コマンド

Cisco NX-OS には、ファブリック接続に関する情報と統計を表示する次のコマンドがあります。

- **show system internal fabric connectivity [module module-number]** : すべてのファブリック モジュールまたは单一モジュールの接続情報を表示します。

```
switch# show system internal fabric connectivity
HiGIG Link-info Linecard slot:4
```

LC-Slot	LC-Unit	LC-HGLink	FM-Slot	FM-Unit	FM-HGLink
4	0	HG02	22	0	HG09
4	0	HG03	22	1	HG09
4	0	HG06	24	0	HG09
4	0	HG07	24	1	HG09
4	1	HG02	22	0	HG10
4	1	HG03	22	1	HG10
4	1	HG06	24	0	HG10

■ ファブリック接続コマンド

```

4      1      HG07      24      1      HG10
4      2      HG02      22      0      HG11
4      2      HG03      22      1      HG11
4      2      HG06      24      0      HG11
4      2      HG07      24      1      HG11

```

HiGIG Link-info Fabriccard slot:22

```

-----  

FM-Slot  FM-Unit   FM-HGLink  LC-Slot  LC-Unit   LC-HGLink  

-----  

22      0      HG09      4      0      HG02  

22      0      HG10      4      1      HG02  

22      0      HG11      4      2      HG02  

22      1      HG09      4      0      HG03  

22      1      HG10      4      1      HG03  

22      1      HG11      4      2      HG03

```

HiGIG Link-info Fabriccard slot:24

```

-----  

FM-Slot  FM-Unit   FM-HGLink  LC-Slot  LC-Unit   LC-HGLink  

-----  

24      0      HG09      4      0      HG06  

24      0      HG10      4      1      HG06  

24      0      HG11      4      2      HG06  

24      1      HG09      4      0      HG07  

24      1      HG10      4      1      HG07  

24      1      HG11      4      2      HG07

```

- **show system internal interface counters module module-number [nz]** : モジュール上の HG またはファブリックリンクのレートを表示します。nz オプションは 0 以外のカウンタだけを表示します。

```

switch# show system internal interface counters module 22 nz
Internal Port Counters (150 secs rate) for Slot: 22
=====
Interface ASIC ASIC  BCM  TxBitRate(BwUtil)  TxPktRate RxBitRate(BwUtil)  RxPktRate
          Port Inst Port      (bps)        (pps)      (bps)        (pps)
-----
ii22/1/10  HG9    0    10      0( 0.00)      0      33064( 0.00)      17

```

```

switch# show system internal interface counters module 22
Internal Port Counters (150 secs rate) for Slot: 22
=====
Interface ASIC ASIC  BCM  TxBitRate(BwUtil)  TxPktRate RxBitRate(BwUtil)  RxPktRate
          Port Inst Port      (bps)        (pps)      (bps)        (pps)
-----
ii22/1/1   HG0    0    1      0( 0.00)      0      0( 0.00)      0
ii22/1/2   HG1    0    2      0( 0.00)      0      0( 0.00)      0
ii22/1/3   HG2    0    3      0( 0.00)      0      0( 0.00)      0
ii22/1/4   HG3    0    4      0( 0.00)      0      0( 0.00)      0
ii22/1/5   HG4    0    5      0( 0.00)      0      0( 0.00)      0
ii22/1/6   HG5    0    6      0( 0.00)      0      0( 0.00)      0
ii22/1/7   HG6    0    7      0( 0.00)      0      0( 0.00)      0
ii22/1/8   HG7    0    8      0( 0.00)      0      0( 0.00)      0
ii22/1/9   HG8    0    9      0( 0.00)      0      0( 0.00)      0
ii22/1/10  HG9    0   10      0( 0.00)      0      30888( 0.00)     12
ii22/1/11  HG10   0   11      0( 0.00)      0      0( 0.00)      0
ii22/1/12  HG11   0   12      0( 0.00)      0      0( 0.00)      0
ii22/1/13  HG12   0   13      0( 0.00)      0      0( 0.00)      0
ii22/1/14  HG13   0   14      0( 0.00)      0      0( 0.00)      0

```

ii22/1/15 HG14	0	15	0(0.00)	0	0(0.00)	0
ii22/1/16 HG15	0	16	0(0.00)	0	0(0.00)	0
ii22/1/17 HG16	0	17	0(0.00)	0	0(0.00)	0
ii22/1/18 HG17	0	18	0(0.00)	0	0(0.00)	0
ii22/1/19 HG18	0	19	0(0.00)	0	0(0.00)	0
ii22/1/20 HG19	0	20	0(0.00)	0	0(0.00)	0
ii22/1/21 HG20	0	21	0(0.00)	0	0(0.00)	0
ii22/1/22 HG21	0	22	0(0.00)	0	0(0.00)	0
ii22/1/23 HG22	0	23	0(0.00)	0	0(0.00)	0
ii22/1/24 HG23	0	24	0(0.00)	0	0(0.00)	0
ii22/1/33 HG0	1	1	0(0.00)	0	0(0.00)	0
ii22/1/34 HG1	1	2	0(0.00)	0	0(0.00)	0
ii22/1/35 HG2	1	3	0(0.00)	0	0(0.00)	0
ii22/1/36 HG3	1	4	0(0.00)	0	0(0.00)	0
ii22/1/37 HG4	1	5	0(0.00)	0	0(0.00)	0
ii22/1/38 HG5	1	6	0(0.00)	0	0(0.00)	0
ii22/1/39 HG6	1	7	0(0.00)	0	0(0.00)	0
ii22/1/40 HG7	1	8	0(0.00)	0	0(0.00)	0
ii22/1/41 HG8	1	9	0(0.00)	0	0(0.00)	0
ii22/1/42 HG9	1	10	0(0.00)	0	0(0.00)	0
ii22/1/43 HG10	1	11	0(0.00)	0	0(0.00)	0
ii22/1/44 HG11	1	12	0(0.00)	0	0(0.00)	0
ii22/1/45 HG12	1	13	0(0.00)	0	0(0.00)	0
ii22/1/46 HG13	1	14	0(0.00)	0	0(0.00)	0
ii22/1/47 HG14	1	15	0(0.00)	0	0(0.00)	0
ii22/1/48 HG15	1	16	0(0.00)	0	0(0.00)	0
ii22/1/49 HG16	1	17	0(0.00)	0	0(0.00)	0
ii22/1/50 HG17	1	18	0(0.00)	0	0(0.00)	0
ii22/1/51 HG18	1	19	0(0.00)	0	0(0.00)	0
ii22/1/52 HG19	1	20	0(0.00)	0	0(0.00)	0
ii22/1/53 HG20	1	21	0(0.00)	0	0(0.00)	0
ii22/1/54 HG21	1	22	0(0.00)	0	0(0.00)	0
ii22/1/55 HG22	1	23	0(0.00)	0	0(0.00)	0
ii22/1/56 HG23	1	24	0(0.00)	0	0(0.00)	0

- **show system internal interface counters detail module module-number** : 単一モジュール上のすべての HG またはファブリック リンクの詳細な統計情報を表示します。

```
show system internal interface counters detail module 4
-----
-----
Interface: ii4/1/3 ASIC Inst# 0/Port# 3/Name HG2
-----
Last Cleared @ Thu Jan 1 00:00:00 2013
(0)
Tx/Rx Rates (per second):
  secs      tx bytes      tx packets    rx bytes      rx packets
  [0] - 10        0            0            0            0
  [1] - 150      9448          60            0            0
  [2] - 300      9448          60            0            0
Mac Pktflow:
  Rx Counters:
    Ingress Packets : 0x0000000000000000/0
    Unicast Packets : 0x0000000000000000/0
    Multicast Packets: 0x0000000000000000/0
    Broadcast Packets: 0x0000000000000000/0
    Jumbo Packets    : 0x0000000000000000/0
    Total Bytes      : 0x0000000000000000/0

  Rx Bytes by Packet Size:
    64:          0x0000000000000000/0
    65 - 127:    0x0000000000000000/0
    128 - 255:   0x0000000000000000/0
```

■ パケットトレーサでパケットフローをトラブルシューティング

```

256 - 511: 0x0000000000000000/0
512 - 1023: 0x0000000000000000/0
1024 - 1518: 0x0000000000000000/0
1519 - 1548: 0x0000000000000000/0

Tx Counters:
Egress Packets : 0x0000000000001351/4945
Unicast qackets: 0x0000000000001351/4945
Multicast qackets: 0x0000000000000000/0
Broadcast Packets: 0x0000000000000000/0
Jumbo Packets : 0x0000000000000000/0
Undersize Packets: 0x0000000000000000/0
Total Bytes : 0x000000000008e756/583510

Tx Bytes by Packet Size
64: 0x0000000000000000/0
65 - 127: 0x0000000000001351/4945
128 - 255: 0x0000000000000000/0
256 - 511: 0x0000000000000000/0
512 - 1023: 0x0000000000000000/0
1024 - 1518: 0x0000000000000000/0
1519 - 1548: 0x0000000000000000/0
trunk: 0x0000000000000000/0

Mac Control:
Rx Pause: 0x0000000000000000/0
Tx Pause: 0x0000000000000000/0
Reset: 0x0000000000000000/0

Mac Errors:
Undersize: 0x0000000000000000/0
Runt: 0x0000000000000000/0
Crc: 0x0000000000000000/0
Input Errors: 0x0000000000000000/0
In Discard: 0x0000000000000000/0
Giants: 0x0000000000000000/0
Output Errors: 0x0000000000000000/0
Output Discard: 0x0000000000000000/0
Bad Proto: 0x0000000000000000/0
Collision: 0x0000000000000000/0
Late Collision: 0x0000000000000000/0
No Carrier: 0x0000000000000000/0

```

パケットトレーサでパケットフローをトラブルシューティング

Packet Tracer

パケットトレーサは、ネットワークプロセッサからパケットをキャプチャできるようにする新しいトラブルシューティングツールです。Cisco Nexus 9000クラウドスケールスイッチで使用可能なELAMツールと同様に、このツールは、ASICがキャプチャされたパケットをどのように転送したかを理解するための情報を提供します。この情報は、パケットフローのトラブルシューティングに役立ちます。

パケット フローの問題をデバッグする SPAN、ERSPAN、Ethalyzer などのさまざまなツールが存在しますが、パケットトレーサは、パフォーマンスのペナルティや環境の中断を伴わずに、ASIC の転送パイプライン内でトラブルシューティングを行えます。

パケットトレーサには次の機能があります：



重要 パケットトレーサを効果的に使用するには、ASIC 転送パイプラインについて包括的に理解している必要があります。この知識は、正確なフィルタを設定し、パケットキャプチャの結果を正確に解釈するために不可欠です。

- IPv4/IPv6 アドレス、TCP/UDP ポートなどのさまざまなプロトコルパラメータに基づいて、パケットをキャプチャするためのフィルタを設定できます。
- 128 バイトフィルタと 128 バイトマスクを使用して着信パケットを照合するため、ASIC レベルで柔軟なフィルタリング機能を提供します。これにより、パケットのペイロード部分でパケットプロトコルパラメータまたは特定のバイトシーケンスを使用してフィルタを設計できます。
- パケットはパイプラインのさまざまな段階を通過するため、転送パイプラインの状態とともに 128 バイトのパケットをキャプチャします。
- パケット サイズ、トラフィック クラスなどの非プロトコル情報を使用してフィルタを作成できます。

パケットトレーサのワークフロー

パケットトレーサを使用してパケットをキャプチャするには、次の手順を実行する必要があります。

- パケットを受信パケットパス (RxPP) または送信パケットパス (TxPP) でキャプチャする必要があるかどうかを決定します。
- プロトコルパラメータまたは特定のバイトシーケンスをフィルタとして使用して、パケットをキャプチャする必要があるかどうかを判断します。
 - プロトコルパラメータをフィルタとして使用してパケットをキャプチャするには、キャプチャするパケットのフレームまたはパケット形式を識別します。これにより、キャプチャするパケットが、IPv4 または IPv6、またはその他の既知のプロトコルが続くイーサネットパケットか、VLAN タグ付きイーサネットパケットの後にウェルノウンプロトコルなどを続けます。詳細については、[パケットのフォーマット \(8 ページ\)](#) を参照してください。



(注)

パケット形式は、フィルタとして選択できるさまざまなプロトコルパラメータに基づいて 128 バイトのフィルタおよびマスクを形成するためのテンプレートとして機能します。

■ パケットのフォーマット

- 特定のバイトシーケンスをフィルタとして使用してパケットをキャプチャするには、特定の連続したバイトシーケンスと開始オフセットを特定します。
- パケットトレーサの開始します。
- パケットトレーサがトリガーされ、結果を表示するまで待ちます。

パケットのフォーマット

ASIC 上のパケットトレーサは、128 バイト パケット フィルタと対応する 128 バイト フィルタ マスクを使用して、特定のパケットをキャプチャします。パケット フィルタは、特定のオフセットに特定の値を使用して、対象のパケットの最初の 128 バイトを表すバイトストリングとして使用します。ASIC は フィルタ マスクを使用して、このバイト パターンと入力/出力 パケットを照合します。この設定により、任意のプロトコルタイプまたはパターンのパケットをキャプチャでき、一致パターンを柔軟に指定できます。

パケット フィルタとマスクを作成するには、通常、dot1q ヘッダー、IPv4 送信元 IP、宛先 IP、UDP 送信元ポートなどの対象のプロトコル フィールドを特定し、それらの値をパケット フィルタの正しいオフセットに組み込む必要があります。次に、一致するマスクが、フィルタ マスクの同じオフセットに適用されます。パケットは 128 バイトのフィルタおよびマスクと照合されるため、イーサネット ヘッダーを含む代表的なパターンを作成する特定のパケット形式を意識する必要があります。

この図は、特定のパケット形式の作成に役立つパケット形式ツリーを表しています。

図 1: パケットのフォーマットツリー

```
eth
+-- arp
+-- ipv4
|  +-- tcp
|  +-- udp
|  |  +-- vxlan
|  |  +-- eth
|  |  |  +-- arp
|  |  |  +-- llcu
|  |  |  |  +-- snap
|  |  |  |  +-- llci
|  |  |  |  +-- snap
|  |  |  +-- llcs
|  |  |  +-- snap
|  |  +-- ipv4
|  |  |  +-- tcp
|  |  |  +-- udp
|  |  |  +-- icmp
|  |  +-- ipv6
|  |  |  +-- tcp
|  |  |  +-- udp
|  |  |  +-- icmpv6
|  |  +-- qinq
|  |  |  +-- ipv4
|  |  |  |  +-- tcp
|  |  |  |  +-- udp
|  |  |  |  +-- icmp
|  |  |  +-- ipv6
|  |  |  |  +-- tcp
|  |  |  |  +-- udp
|  |  |  |  +-- icmpv6
|  |  +-- vntag
|  |  |  +-- ipv4
|  |  |  |  +-- tcp
|  |  |  |  +-- udp
|  |  |  |  +-- icmp
|  |  |  +-- ipv6
|  |  |  |  +-- tcp
|  |  |  |  +-- udp
|  |  |  |  +-- icmpv6
|  |  +-- mpls
|  |  |  +-- mpls
|  |  |  |  +-- ipv4
|  |  |  |  +-- ipv6
|  |  +-- arp
|  +-- icmp
|  +-- gre
|  |  +-- ipv4
|  |  +-- ipv6
+-- dot1q
```

次の表に、パケットトレーサのサポートされているパケット形式の使用例をいくつか示します。

■ パケットキャプチャの注意事項および制約事項

表 1:サポートされるパケット形式の例

対象トラフィック	使用するパケット形式
VLAN タグ付きの TCP トラフィック	<ul style="list-style-type: none"> IPv4 の場合 : eth-dot1q-ipv4-tcp IPv6 の場合 : eth-dot1q-ipv6-tcp
VLAN タギングによる VXLAN トラフィック	<ul style="list-style-type: none"> IPv4 の場合 : eth-dot1q-ipv4-udp-vxlan-eth-ipv4 IPv6 の場合 : eth-dot1q-ipv6-udp-vxlan-eth-ipv4
内部 VLAN タギングを持つ VXLAN トラフィック	<ul style="list-style-type: none"> IPv4 の場合 : eth-ipv4-udp-vxlan-eth-dot1q-ipv4 IPv6 の場合 : eth-ipv6-udp-vxlan-eth-dot1q-ipv4
内部 VLAN タギングおよび ARP を持つ VXLAN トラフィック	<ul style="list-style-type: none"> IPv4 の場合 : eth-ipv4-udp-vxlan-eth-dot1q-arp IPv6 の場合 : eth-ipv6-udp-vxlan-eth-dot1q-arp
ARP トラフィックに基づいてフィルタ処理します。	eth-arp
ICMP トラフィック	<ul style="list-style-type: none"> IPv4 の場合 : eth-ipv4-icmp IPv6 の場合 : eth-ipv6-icmp
MPLS トラフィック	• eth-mpls-mpls-ipv4 (MPLS は 6 つのラベルに対して最大 6 回追加可能)

パケットキャプチャの注意事項および制約事項

パケットトレーサに関する注意事項と制約事項は次のとおりです :

- packet offset コマンドでは、RxPP または TxPP に最大 10 の条件を設定できます。
- パケットフィルタは、パケットの最初の 128 バイトにのみ一致します。
- パケットトレーサは、RxPP パスまたは TxPP パスのいずれかで実行できます。
- キャプチャモード「連続」はサポートされていません。

- スライスを指定しない場合、出力にはパケットトレースが追跡されているすべてのスライスが表示されます。

パケット トレーサのサポートされているリリースとプラットフォーム

リリース	プラットフォーム (Platform)
10.5(3)F 以降	Cisco N9364E-SG2-Q および N9364E-SG2-O スイッチ
10.6(1)F 以降	Cisco N9336C-SE1 スイッチ

パケット トレーサの展開

次のコマンドを使用して、スイッチでパケット トレーサ機能をトリガーできます。

手順

ステップ 1 **packet-trace** コマンドを実行して、パケット トレース モードを有効にします。

例 :

```
switch# packet-trace
switch(S1HAL-pt)#
```

ステップ 2 次を実行します **trigger init**。 \$rxpp | txppRx パス (Rxpp) または Tx パス (TxPP) でパケットをキャプチャするためのコマンド。

例 :

```
switch(S1HAL-pt)# trigger init rxpp
switch(S1HAL-pt-rxpp)#
```

ステップ 3 パケットをトレースするパケット形式を指定するには、**packet-format** *packet-format* コマンドを実行します。

例 :

```
switch(S1HAL-pt-txpp)# packet-format eth-ipv4-tcp
switch(S1HAL-pt-txpp-pkt-fmt)#
```

この例では、キャプチャされるパケットのフレーム形式は、IPv4 および TCP プロトコルフィルタを使用する Ethernet です。

(注)

このコマンドでは、ヘルプ (?) の下にこのコマンドの使用方法をガイドするユーザー ガイド オプションが表示されます。

ステップ 4 **set outer {l2 | ipv4 | ipv6 | arp | l4 | mpls} | set inner { l2 | ipv4 | ipv6 | arp | l4 | mpls }** コマンドを実行して、カプセル化されていないパケット (set external) またはカプセル化されたパケット (set internal) のさまざまなプロトコルフィルタを設定します。

パケットトレーサの展開

例：

```
switch(S1HAL-pt-txpp-pkt-fmt) # set outer ipv4
```

パケットのフィルタを設定するには、適切なパケット形式を選択する必要があります。パケットのフォーマット (8 ページ) セクションを参照してください。

(注)

カプセル化されたパケットの場合、IP/UDPなどのアンダーレイプロトコルには「外部」という用語が使用され、VXLAN、GREなどのオーバーレイプロトコルには「内部」という用語が使用されます。

ステップ 5 (任意) **show filters** コマンドを実行して、さまざまな非パケットフィルタを設定します。

例：

```
switch(S1HAL-pt-txpp-pkt-fmt) # show filters
```

パケットヘッダー フィルタに加えて、これらのフィルタを設定できます。

(注)

パケットキャプチャは、パケットヘッダー フィルタと非パケット フィルタ (設定されている場合) の両方に基づいて実行されます。

ステップ 6 **start** コマンドを実行して ASIC にフィルタを設定し、キャプチャ操作を開始します。

例：

```
switch(S1HAL-pt-txpp-pkt-fmt) # start
```

ステップ 7 (任意) **status** コマンドを実行し、キャプチャが行われたかどうかに関する情報を提供します。

例：

```
switch(S1HAL-pt-txpp-pkt-fmt) # status
```

このコマンドを複数回実行して、パケットがキャプチャされたかどうかを知ることができます。

ステップ 8 (任意) **stop** コマンドを実行して、開始したキャプチャをキャンセルします。

例：

```
switch(S1HAL-pt-txpp-pkt-fmt) # stop
```

(注)

キャプチャが正常に完了した後に **stop** コマンドを発行しても、キャプチャの結果は変わりません。

ステップ 9 (任意) 設定されたフィルタをクリアするには、**reset** コマンドを実行します。

例：

```
switch(S1HAL-pt-txpp-pkt-fmt) # reset
switch(S1HAL-pt-txpp) #
```

ステップ 10 **report [brief] [slice slice-ids]** コマンドを実行して、キャプチャされたパケットの詳細と、キャプチャの各段階の NPPD デコード情報のダンプを表示します。

例：

```
switch(S1HAL-pt-txpp) # report detail
```

brief オプションの場合、NPPD デコードされた情報はダンプされません。

slice オプションを指定した場合、指定したスライスのパケットキャプチャの詳細が表示されます。

レポートの詳細については、[パケットトレーサの構成例（15 ページ）](#) を参照してください。

ステップ 11 **set npi {err-flag value | initial-TC value | processing-code value | reassembly-ctxt value | single-frag-pkt value | src-pif value | tx-to-rx-rec-data value | unsch-rec-code value}** コマンドを実行して RxPP キャプチャモードでさまざまな非パケットを設定します。

例：

```
switch(S1HAL-pt-rxpp)# set npi src-pif 15
```

パケットヘッダーフィルタに加えて、これらのフィルタを設定できます。

パケットキャプチャは、パケットヘッダーフィルタと非パケットフィルタ（設定されている場合）の両方に基づいて実行されます。

- **[エラー フラグ (Error Flag)] (err-flag)** : このオプションを使用して、ハードウェア エラーが検出されたかどうかを確認します。サイズ：1 ビット。範囲：0～1。
- **[初期 TC (initial-TC) (Initial TC (initial-TC))]** : このオプションを使用して、IFG によって計算されたパケットの初期トラフィック クラスを設定します。サイズ：3 ビット。範囲：0～7。
- **[処理コード (processing-code)]** : 異常なイベントを示すにはこのオプションを使用します。サイズ：7 ビット。範囲：0～127。
- **[リアセンブルコンテキスト (reassembly-ctxt) (Reassembly Context (reassembly-ctxt))]** : このオプションを使用して、リアセンブルするパケットフラグメントを一意に識別します。サイズ：11 ビット。範囲：0～2047。
- **[単一のフラグメントパケット (single-frag-pkt) (Single Fragment Packet (single-frag-pkt))]** : このオプションを使用して、単一のフラグメントパケットのフラグを設定します。サイズ：1 ビット。範囲：0～1。
- **送信元ポートインターフェイス (src-pif)** : このオプションを使用して、送信元の物理インターフェイスを設定します。サイズ：7 ビット。範囲：0～127。
- **Tx から Rx リサイクルデータ (tx-to-rx-rec-data)** : このオプションを使用して、TxNPU から RxNPU に渡されるリサイクルデータを設定します。サイズ：8 ビット。範囲：0～255。
- **[スケジュール解除のリサイクルコード (unsch-rec-code)]** : このオプションを使用して、必要なスケジュールされていないリサイクルのタイプを設定します。サイズ：4 ビット。範囲：0～15。

ステップ 12 **set npi {acc-LM-cache-and-Idx value | colour value | congested value | congestion-level value | cud value | dst-intf value | eop value | err-flag value | is-elephant-flow value | lm-cache-index value | omd value | pkt-size value | sop value | src-slice value | src-slice-sys-port value | start-packing value | traffic-class value}** コマンドを実行して TxPP キャプチャモードでさまざまな非パケットを設定します。

例：

```
switch(S1HAL-pt-txpp)# set npi dst-intf 25
```

パケットヘッダーフィルタに加えて、これらのフィルタを設定できます。

■ パケットトレーサの展開

パケットキャプチャは、パケットヘッダーフィルタと非パケットフィルタ（設定されている場合）の両方に基づいて実行されます。

- **Access LM キャッシュとインデックス (acc-LM-cache-and-Idx)** : このオプションを使用して、損失測定要求キャッシュを確認します。サイズ：1ビット。範囲：0～1。
- **[色 (Color (colour))]** : このオプションを使用して、パケットのドロップ優先順位を確認します。サイズ：2ビット。範囲：0～3。
- **[輻輳 (Congested)] (輻輳)** : このオプションを使用して、パケットで輻輳イベントが発生したかどうかを確認します。サイズ：1ビット。範囲：0～1。
- **[輻輳レベル (congestion-level)]** : このオプションを使用して、パケットによって発生したキューの輻輳の測定を確認します。サイズ：4ビット。範囲：0～15。
- **[一意のデータ (cud) (Copy Unique Data (cud))]** : このオプションを使用して、パケットコピーが生成された理由を確認します。サイズ：23ビット。範囲：0～8388607
- **[接続先インターフェイス (dst-intf) (Destination Interface (dst-intf))]** : 接続先の物理インターフェイスを設定するには、このオプションを使用します。サイズ：8ビット。範囲：0～255。
- **パケットの終端 (eop) (End Of Packet (eop))** : パケットフラグメントの終端を設定するには、このオプションを使用します。サイズ：1ビット。範囲：0～1。
- **[エラー フラグ (Error Flag)] (err-flag)** : このオプションを使用して、ハードウェアエラーが検出されたかどうかを確認します。サイズ：1ビット。範囲：0～1。
- **エレファントフロー (is-elephant-flow)** : このオプションを使用して、パケットが大規模なフローの一部として識別されているかどうかを確認します。サイズ：1ビット。範囲：0～1。
- **LM キャッシュインデックス (lm-cache-index)** : このオプションを使用して、損失測定要求キャッシュのインデックスをチェックします。サイズ：2ビット。範囲：0～255。
- **[Output queue Mapped Data (omd)]** : このオプションを使用して、出力キューにマップされたデータを設定します。サイズ：9ビット。範囲：0～511。
- **[パケットサイズ (pkt-size) (Packet size (pkt-size))]** : パケットサイズをバイト単位で確認するには、このオプションを使用します。サイズ：14ビット。
- **[パケットの開始 (sop) (Start Of Packet (sop))]** : パケットフラグメントの開始を設定するには、このオプションを使用します。サイズ：1ビット。範囲：0～1。
- **[送信元スライス (src-slice) (Source Slice (src-slice))]** : ソーススライスを設定するには、このオプションを使用します。サイズ：3ビット。
- **[送信元スライスシステムポート (src-slice-sys-port) (Source Slice System Port (src-slice-sys-port))]** : このオプションを使用して、送信元スライスシステムポートを設定します。サイズ：8ビット。範囲：0～255。
- **[パッキングの開始 (start-packing)]** : このオプションを使用して、デュアルパケットの最初のパケットを設定します。サイズ：1ビット。範囲：0～1。

- [トライフィック クラス (Traffic Class) (traffic-class) (Traffic Class (traffic-class))] : このオプションを使用して、パケットのトライフィッククラスを設定します。サイズ : 3 ビット。範囲 : 0 ~ 7.

ステップ 13 **set pkt-offset condition Value offset Value** コマンドを実行して、必要に応じて raw フィルタとマスクを任意のバイトパターンに設定します。

例 :

```
switch(S1HAL-pt-txpp)# set pkt-offset condition 1 offset 0x10 value 0xababab mask 0xffffffff
```

最大 10 のこのような条件を設定できます。この操作は、**set outer** コマンドを使用してプロトコルフィルタを設定することを相互に排他的です。

(注)

- 「0xf」のマスクは、フィルタ内に対応するニブルを考慮する必要があることを示しますが、「0x0」のマスクは、無視する必要があることを意味します。
- このコマンドは、オフセット値がわかっている場合にのみ使用する必要があります。

パッカートレーサの展開の確認

パッカートレーサの展開情報を表示するには、次のコマンドを使用します。

コマンド	目的
show filters	実際の pkt-offset フィルタの設定を表示します。詳細については、「 パケットトレーサの構成例 (15 ページ) 」を参照してください。

パケットトレーサの構成例

次に、パケットトレーサ機能を使用してフィルタとレポートをキャプチャする例を示します。

```
switch# packet-trace

switch(S1HAL-pt)# trigger init rxpp

switch(S1HAL-pt-rxpp)# packet-format eth-dot1q-ipv4

switch(S1HAL-pt-rxpp-pkt-fmt)#
switch(S1HAL-pt-rxpp-pkt-fmt)# set outer ipv4 src-ip 62.0.134.2

switch(S1HAL-pt-rxpp-pkt-fmt)# set outer ipv
ipv4  ipv6
switch(S1HAL-pt-rxpp-pkt-fmt)# set outer ipv4 next-protocol 17

switch(S1HAL-pt-rxpp-pkt-fmt)#
switch(S1HAL-pt-rxpp-pkt-fmt)# show filters

slot  1
=====
```

パケットトレーサの構成例

```

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  0000 00.. = Differentiated Services Codepoint: Default (0)
  .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 80
Identification: 0x0000 (0)
000. .... = Flags: 0x0
  0.... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Protocol: UDP (17)
Header Checksum: 0x1734 [validation disabled]
[Header checksum status: Unverified]
Source Address: 62.0.134.2
Destination Address: 224.0.0.102
[Stream index: 0]
User Datagram Protocol, Src Port: 1985, Dst Port: 1985
Source Port: 1985
Destination Port: 1985
Length: 60
Checksum: 0x5b42 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]
[Stream Packet Number: 1]
UDP payload (52 bytes)
Cisco Hot Standby Router Protocol
Group State TLV: Type=1 Len=40
  Version: 2
  Op Code: Hello (0)
  State: Standby (5)
  IP Ver.: IPv4 (4)
  Group: 134
  Identifier: Cisco_21:e5:5b (40:14:82:21:e5:5b)
  Priority: 100
  Hellotime: Default (3000)
  Holdtime: Default (10000)
  Virtual IP Address: 62.0.134.3
Text Authentication TLV: Type=3 Len=8
  Authentication Data: Default (cisco)

Packet Summary Decode:

Packet capture Summary : (Captured at RXPP) :

Ingress port details:
Interface : Eth1/22
LTL : 0x58
System Port : 0x98
PIF : 24
Slice : 1
ifg : 1

Packet drop summary:
Packet dropped : NO

Packet punt summary:
Punt Details : Packet is not punted

Packet Details:
=====

decode_termination_input slice_id 1

Packet Bytes(up to 128B) :

```

パケットトレーサの構成例

===== FI Array =====

FIELD_NAME	:	VALUE
array[9]	:	0x0
array[8]	:	0x0
array[7]	:	0x0
array[6]	:	0x0
array[5]	:	0x0
array[4]	:	0x2e00
array[3]	:	0x260f
offset_in_bytes		0x26
protocol_type		PROTOCOL_TYPE_UDP
array[2]	:	0x1214
offset_in_bytes		0x12
protocol_type		PROTOCOL_TYPE_IPV4_L4
array[1]	:	0xe48
offset_in_bytes		0xe
protocol_type		PROTOCOL_TYPE_VLAN
flags		2
array[0]	:	0x11
offset_in_bytes		0x0
protocol_type		PROTOCOL_TYPE_ETHERNET_VLAN

===== RXPP Termination Input =====

FIELD_NAME	:	VALUE
padding_1	:	0x0
pch_label	:	0x0
unsch_rcy_code	:	0x0
tx_to_rx_rcy_data	:	0xf4
mtuViolation	:	0x0
initial_tc	:	0x0
offset_in_fragment	:	0x0
slice_source_system_port	:	0x98
processing_code	:	0x0
destination	:	0x0
use_cache	:	0x0
single_fragment_packet	:	0x1
flow_signature_on_npuh	:	0x0
phb	:	0x0
reassembly_context	:	0x7ff
learn_enable	:	0x0
receive_time_from_nppd	:	0x0
rxnpu_recycle_count	:	0x0
rxnpu_recycle_data	:	0x0

==== RXPP Termination Macro Stack =====

```
NPE-macros-stack[0]: network_rx_mac_af_and_termination_macro  
NPE-macros-stack[1]: network_rx_mac_relay_ipv4_mc_termination_macro
```

==== TRXPP Termination Lookup Keys/Results =====

```
NPE-lookup Keys/Results[0]:  
no lookup hit, bucket #1 context network engine termination  
no lookup hit, bucket #2 context network engine termination
```

```

-----+-----+
| Key Bucket |                               Key Type
|   Key Value          | Result Bucket |          |
|   Result Type         |   | Result Value |          |
+-----+-----+
-----+-----+
|   a   |      npl_service_mapping_ac_port_tag_compound_table_key_t | | |
| 0x5e000066000000138184327c |   d   |          0          |
|   NoneType           |   |          0          |          |
|   d     | npl_ingress_qos_tag_encoding_pack_table_key_option_tag_type_v4_dscp_t |
|   0x0           |   d   | npl_ingress_q |
| os_tag_encoding_pack_table_payloads_t |   0x0   |
+-----+-----+
-----+-----+
NPE-lookup Keys/Results[1]:
no lookup hit, bucket #0 context network engine termination
no lookup hit, bucket #2 context network engine termination
+-----+-----+
| Key Bucket |                               Key Type
| Result Bucket |          Result Type          | Key Value |
|   |          Result Value |          |          |
+-----+-----+
-----+-----+
|   b   | npl_mc_em_termination_attributes_compound_table_key_t | 0x21b  |
|   b   | npl_base_13_lp_attr_union_t |
|   | 0xe00010c32af0086740000000000 |
|   d   | npl_mc_macro_compressed_fileds_pack_table_key_t | 0x388  |
|   d   | npl_mc_macro_compressed_fileds_pack_table_ |
| payloads_t |   0x0   |
+-----+-----+
===== RXPP Termination Output =====

-----+-----+
FIELD_NAME : VALUE
-----+
learn_command : 0x0
lb_command : 0x0
offset_in_fragment : 0x0
slice_source_system_port : 0x98
processing_code : 0x0
destination : 0xe0086
use_cache : 0x0
single_fragment_packet : 0x1
flow_signature_on_npuh : 0x0
phb : 0x0
reassembly_context : 0x7ff
learn_enable : 0x0
receive_time_from_nppd : 0x0
rxnpu_recycle_count : 0x0
rxnpu_recycle_data : 0x0
-----+-----+
===== RXPP Forwarding Macro Stack =====

NPE-macros-stack[0]: network_rx_ipv4_rtf_macro
NPE-macros-stack[1]: network_rx_mac_forwarding_macro
NPE-macros-stack[2]: resolution_macro

===== RXPP Forwarding Lookup Keys/Results =====

```

パケットトレーサの構成例

```

NPE-lookup Keys/Results[0]:
no lookup hit, bucket #1 context network engine forwarding
no lookup hit, bucket #3 context network engine forwarding
Error result bucket # 0 , from table ingress_rtf_ipv4_db1_240_f0_compound_table ,
overlapping previous value
Error result bucket # 1 , from table ingress_rtf_ipv4_db1_240_f0_compound_table ,
overlapping previous value
Error result bucket # 2 , from table ingress_rtf_ipv4_db1_240_f0_compound_table ,
overlapping previous value
Error result bucket # 3 , from table ingress_rtf_ipv4_db1_240_f0_compound_table ,
overlapping previous value
+-----+
-----+
| Key Bucket | Key Type | | Key
Value | Result Bucket | Result Type | |
| Result Value | | | |
+-----+
-----+
| a | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t | | npl_rtf_payload_t
0x4ff45003e008602e000006607c19c0101f056 | b | npl_rtf_payload_t
| 0x0 | | |
| a | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t | | npl_rtf_payload_t
0x4ff45003e008602e000006607c19c0101f056 | c | npl_rtf_payload_t
| 0x0 | | |
| a | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t | | npl_rtf_payload_t
0x4ff45003e008602e000006607c19c0101f056 | d | npl_rtf_payload_t
| 0x0 | | |
| a | npl_ingress_rtf_ipv6_db4_480_f0_compound_table_key_t | | npl_rtf_payload_t
0x4ff45003e008602e000006607c19c0101f056 | a | npl_rtf_payload_t
| 0x0 | | |
| c | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t | | npl_rtf_payload_t
0x3821a | b | npl_rtf_payload_t
| 0x0 | | |
| c | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t | | npl_rtf_payload_t
0x3821a | c | npl_rtf_payload_t
| 0x0 | | |
| c | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t | | npl_rtf_payload_t
0x3821a | d | npl_rtf_payload_t
| 0x0 | | |
| c | npl_ingress_rtf_ipv4_db1_240_f0_compound_table_key_t | | npl_rtf_payload_t
0x3821a | a | npl_rtf_payload_t
| 0x0 | | |
+-----+
-----+
NPE-lookup Keys/Results[1]:
no lookup hit, bucket #0 context network engine forwarding
no lookup hit, bucket #1 context network engine forwarding
no lookup hit, bucket #2 context network engine forwarding
+-----+
-----+
| Key Bucket | Key Type | | Key Value | Result
Bucket | Result Type | | Resu
lt Value | | | |
+-----+
-----+
| d | npl_mac_forwarding_table_compound_table_key_t | 0x8601005e00006612 | a
| npl_mac_forwarding_table_compound_table_payloads_t |
0x0 | | |
+-----+
-----+
NPE-lookup Keys/Results[2]:
no lookup hit, bucket #3 context network engine forwarding

```

```

+-----+-----+-----+
| Key Bucket |           Key Type
|             Key Value           | Result Bucket |
|             Result Type          |             | Result Value |
+-----+-----+-----+
|   a   |           npl_v4_14_resolution_table_compound_key_t
|       0x3300103802183           |   a   | 0xa9100e0086 |
|   b   |           npl_v4_14_resolution_table_compound_key_t
| 0x113e008602e00007c107c180000000000000000 |   a   | 0xa9100e0086 |
|   c   |           npl_select_fwd_q_m_counter_base_pack_table_key_option_false_value_t
|       0x40000000           |   c   | np
|           l_select_fwd_q_m_counter_base_pack_table_payloads_t | 0x0   |
+-----+-----+-----+
===== RXPP Forwarding Output =====

----- FIELD_NAME : VALUE -----
offset_in_fragment      : 0x0
slice_source_system_port : 0x98
processing_code          : 0x0
destination              : 0x0
use_cache                : 0x0
single_fragment_packet   : 0x1
flow_signature_on_npuh   : 0x0
phb                      : 0x0
reassembly_context        : 0x7ff
learn_enable               : 0x0
receive_time_from_nppd   : 0x0
rxnpu_recycle_count      : 0x0
rxnpu_recycle_data        : 0x0
padding_2                 : 0x0
use_ecn                  : 0x0
flb_control_code          : 0x0
padding_1                 : 0x0
ethernet_rate_limiter_type: 0x7
fwd_offset_cmd            : 0x0

===== RXPP TM PD IFG0 =====

----- FIELD_NAME : VALUE -----
color                    : 0x0
counter_meter_command    : 0xfd9fcf5
is_dummy_pd               : 0x1
reorder_data              : 0xb72a
drop                     : 0x1
forwarding_destination   : 0x0
mirror_bitmap             : 0x37ff
source_slice_system_port : 0x0
traffic_class             : 0x0
slice_mode_data           : 0x1fb7ff4fcc9fd1ec2f4000000000000
processing_code            : 0x0
lb_key_msbs_bits          : 0x0
packet_size_bits          : 0x0

===== RXPP TM PD IFG1 =====

```

パケットトレーサの構成例

```

receive_time          :          0x0
meter_color          :          0x0
l2_flood_mc_pruning_or_etm :          0x0
ingress_qos_remark   : 0x20000
fwd_header_type      :          0x0
rx_nw_app_or_lb_key  : 0x443
fwd_offset           : 0x30
slp_qos_id           : 0xf
encap_type           : 0x0
L2_encap              : 0x0
padding               : 0x0
l2_dlp                : 0x0
pif                   : 0x0
ifg                   : 0x0
slp_dm_ptp            : 0x0
is_inject_packet_capture_en : 0x0
is_inject_up           : 0x0
ip_first_fragment     : 0x1
ttl                   : 0x0
collapsed_mc          : 0x0
da_bcast_or_mc_rpf   : 0x0
slp_profile           : 0x0
l2_slp                : 0x9e001
l3_slp                : 0x9e00
is_12                 : 0x1
is_rpf_id              : 0x1
value                 : 0x9e001
sgt                   : 0x0

switch(S1HAL-pt-rxpp-pkt-fmt)#

```

その他の参考資料

関連資料	タイトル/リンク
Cisco Nexus 9364E スイッチハードウェア設置ガイド	Cisco Nexus 9364E-SG2-Q スイッチハードウェアインストールガイド Cisco Nexus 9364E-SG2-O スイッチハードウェアインストールガイド
ELAM の概要	http://www.cisco.com/support/docs/7000/nexus/1688/index.html
Nexus 9000 クラウドスケール ASIC (Tahoe) NX-OS ELAM	Nexus 9000 クラウドスケール ASIC (Tahoe) NX-OS ELAM - Cisco

■ その他の参考資料

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。