IOS-XEデータパスパケットトレース機能を使用したトラブルシューティング

内容

はじめに

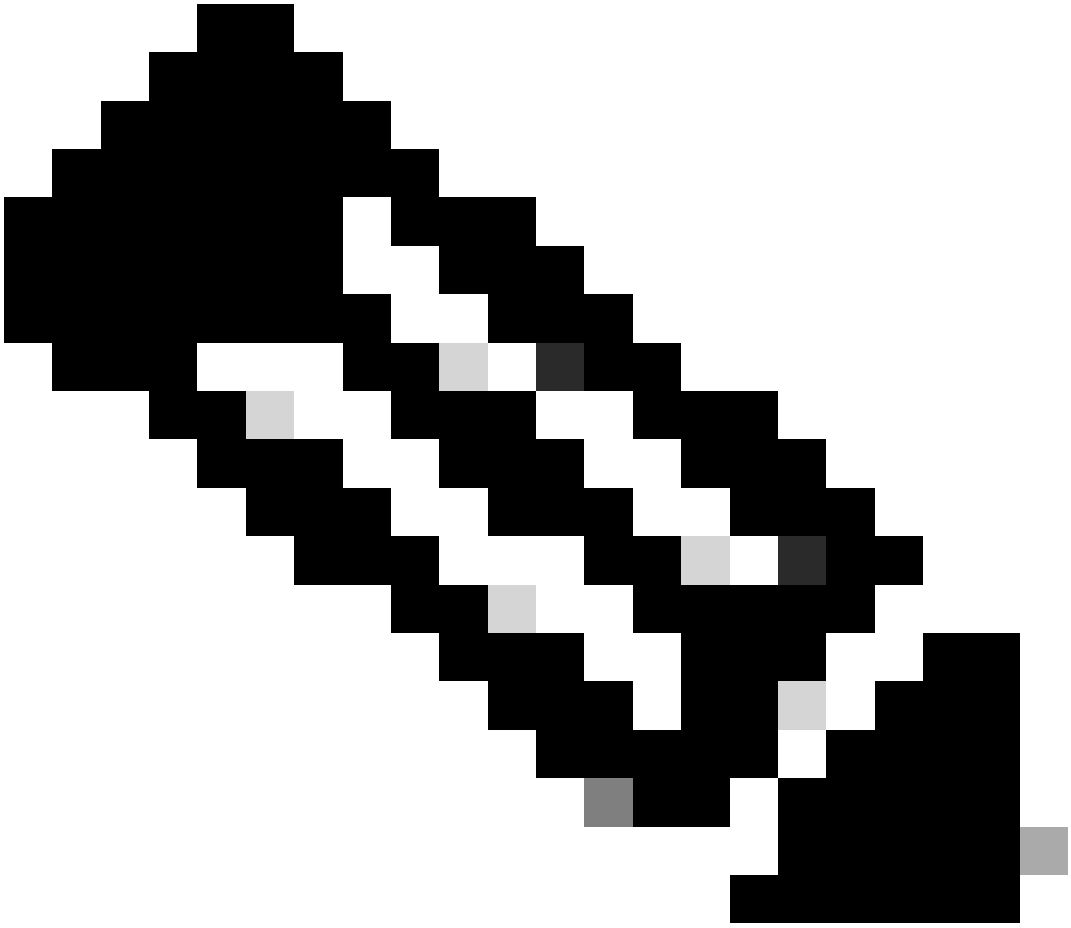
このドキュメントでは、パケットトレース機能を使用してCisco IOS-XE®ソフトウェアのデータパスパケットトレースを実行する方法について説明します。

前提条件

要件

次の情報に関する知識があることが推奨されます。

パケットトレース機能は、QFP(Quantum Flow Processor)ベースのルーティングプラットフォーム上のCisco IOS-XEバージョン3.10以降のリリース（ASR1000、ISR4000、ISR1000、Catalyst 1000、Catalyst 8000、CSR1000v、およびCatalyst 8000vシリーズルータなど）で使用できます。この機能は、Cisco IOS-XEソフトウェアを実行するASR900シリーズアグリゲーションサービスルータまたはCatalystシリーズスイッチではサポートされません。



注：パケットトレース機能は、ASR1000シリーズルータ上の専用管理インターフェイスGigabitEthernet0では動作しません。これは、このインターフェイスに転送されるパケットがQFPで処理されないためです。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco IOS XEソフトウェアリリース3.10S(15.3(3)S)以降

- ASR1000シリーズルータ

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

トラブルシューティング中に設定ミス、容量過負荷、あるいは通常のソフトウェアの不具合などの問題を特定するには、システム内のパケットがどうなったかを理解する必要があります。Cisco IOS XEパケットトレース機能は、このニーズに対応します。これは、アカウンティングに使用され、ユーザ定義の条件のクラスに基づいてパケットごとのプロセスの詳細をキャプチャするために使用されるフィールドセーフ方式を提供します。

参照トポロジ

次の図は、このドキュメントで説明する例で使用するトポロジを示しています。

使用中のパケットトレース

パケットトレース機能の使用を示すために、このセクション全体で使用されている例では、ローカルワークステーション172.16.10.2（ASR1Kの背後）からリモートホスト172.16.20.2への、ASR1K上のインターフェイスGigabitEthernet0/0/1上の入力方向でのInternet Control Message Protocol(ICMP)トラフィックのトレースについて説明します。

次の2つの手順で、ASR1K上のパケットをトレースできます。

1. ASR1Kでトレースするパケットまたはトラフィックを選択するために、プラットフォームの条件付きデバッグを有効にします。

2. path-traceまたはFeature Invocation Array(FIA)トレースオプションを使用して、プラットフォームのパケットトレースを有効にします。

クイックスタートガイド

このドキュメントの内容を十分に理解していて、CLIのクイックスタートガイドのセクションが必要な場合は、次のクイックスタートガイドを参照してください。ここでは、ツールの使用方法を説明する例を数例だけ示します。以降のセクションで構文について詳しく説明し、各自の要件に

適した設定を使用していることを確認してください。

1. プラットフォームの条件を設定します。

```
<#root>

debug platform condition ipv4 10.0.0.1/32 both

--> matches in and out packets with source
 or destination as 10.0.0.1/32


debug platform condition ipv4 access-list 198 egress

--> (Ensure access-list 198 is
 defined prior to configuring this command) - matches egress packets corresponding
 to access-list 198


debug platform condition interface gig 0/0/0 ingress

--> matches all ingress packets
 on interface gig 0/0/0


debug platform condition mpls 10 1 ingress

--> matches MPLS packets with top ingress
 label 10


debug platform condition ingress

--> matches all ingress packets on all interfaces
 (use cautiously)
```

プラットフォーム条件を設定した後、次のCLIコマンドを使用してプラットフォーム条件を開始します。

```
<#root>

debug platform condition start
```

2. パケットトレースを設定します。

```
<#root>

debug platform packet-trace packet 1024
```

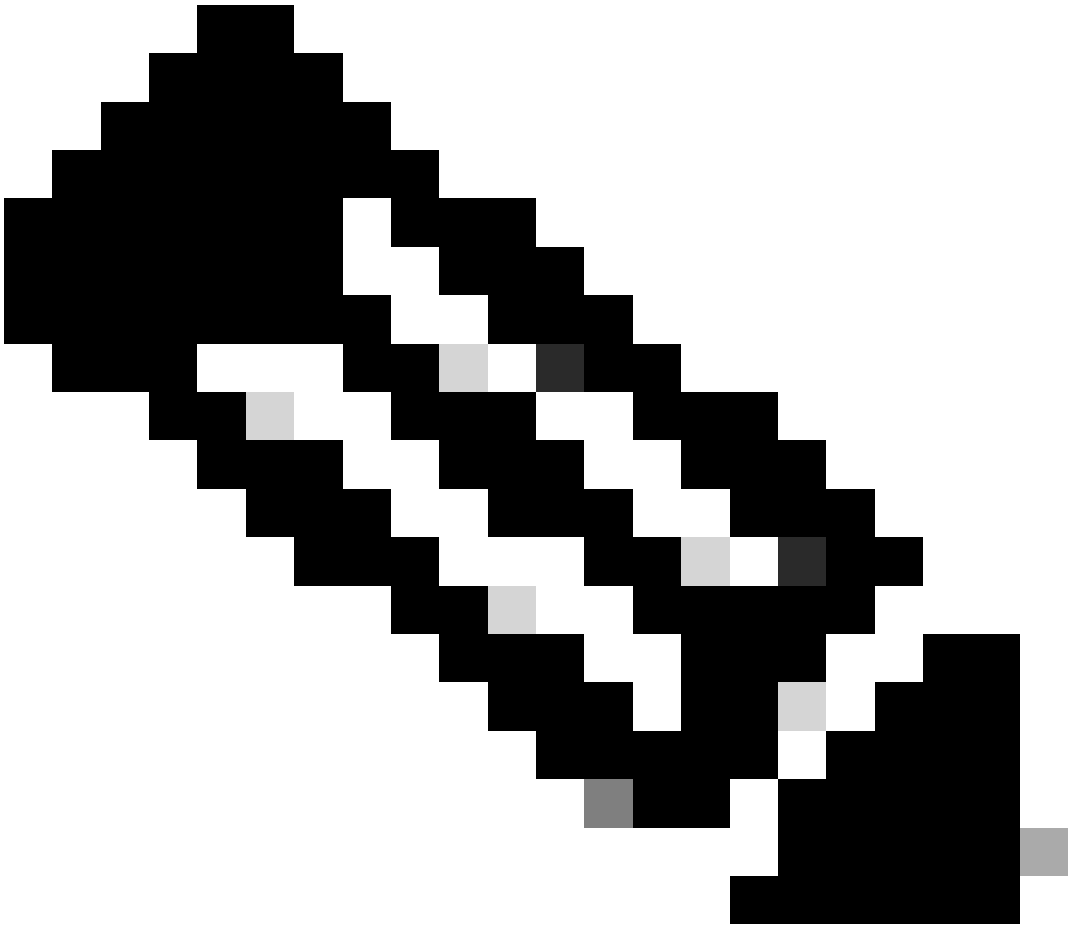
```
 -> basic path-trace, and automatically stops
tracing packets after 1024 packets. You can use "circular" option if needed
```

debug platform packet-trace packet 1024 fia-trace -

```
> enables detailed fia trace, stops
 tracing packets after 1024 packets
```

debug platform packet-trace drop [code <dropcode>]

```
 -> if you want to trace/capture only
packets that are dropped. Refer to Drop Trace section for more details.
```



注：以前のCisco IOS-XE 3.xリリースでは、パケットトレース機能を起動するために、debug platform packet-trace enableコマンドも必要です。Cisco IOS-XE 16.xリリースでは、この機能は不要になりました。

トレースバッファをクリアしてパケットトレースをリセットするには、次のコマンドを入力しま

す。

<#root>

clear platform packet-trace statistics

  --> clear the packet trace buffer

プラットフォームの状態とパケットトレース設定の両方をクリアするコマンドは次のとおりです。

<#root>

clear platform condition all

  --> clears both platform conditions and the packet trace configuration

show コマンド

上記のコマンドを適用した後、プラットフォームの状態とパケットトレースの設定を確認して、必要な情報が揃っていることを確認します。

<#root>

show platform conditions

 --> shows the platform conditions configured

show platform packet-trace configuration

 --> shows the packet-trace configurations

show debugging

 --> this can show both platform conditions and platform packet-trace configured

トレースまたはキャプチャされたパケットを確認するコマンドを次に示します。

<#root>

show platform packet-trace statistics

--> statistics of packets traced

show platform packet-trace summary

```
  --> summary of all the packets traced, with input and
 output interfaces, processing result and reason.
```

show platform packet-trace packet 12

```
 -> Display path trace of FIA trace details for the 12th packet in the trace buffer
```

プラットフォームの条件付きデバッグの有効化

パケットトレース機能は、条件付きデバッグインフラストラクチャを使用して、トレース対象のパケットを決定します。条件付きデバッグインフラストラクチャでは、次の条件に基づいてトラフィックをフィルタリングできます。

- プロトコル
- IPアドレスとマスク
- Access Control List（ACL; アクセス コントロール リスト）
- インターフェイス
- トラフィックの方向（入力または出力）

これらの条件は、パケットにフィルタを適用する場所とタイミングを定義します。

この例で使用されるトラフィックでは、172.16.10.2から172.16.20.2へのICMPパケットの入力方向でプラットフォームによる条件付きデバッグを有効にします。つまり、トレースするトラフィックを選択します。このトラフィックを選択するには、さまざまなオプションを使用できます。

<#root>

ASR1000#

debug platform condition

```
 ?
  egress     Egress only debug
  feature    For a specific feature
  ingress    Ingress only debug
  interface  Set interface for conditional debug
  ipv4       Debug IPv4 conditions
  ipv6       Debug IPv6 conditions
  start      Start conditional debug
  stop       Stop conditional debug
```

この例では、次に示すように、条件を定義するためにアクセスリストが使用されます。

<#root>

ASR1000#

show access-list 150

```
Extended IP access list 150
 10 permit icmp host 172.16.10.2 host 172.16.20.2
ASR1000#
```

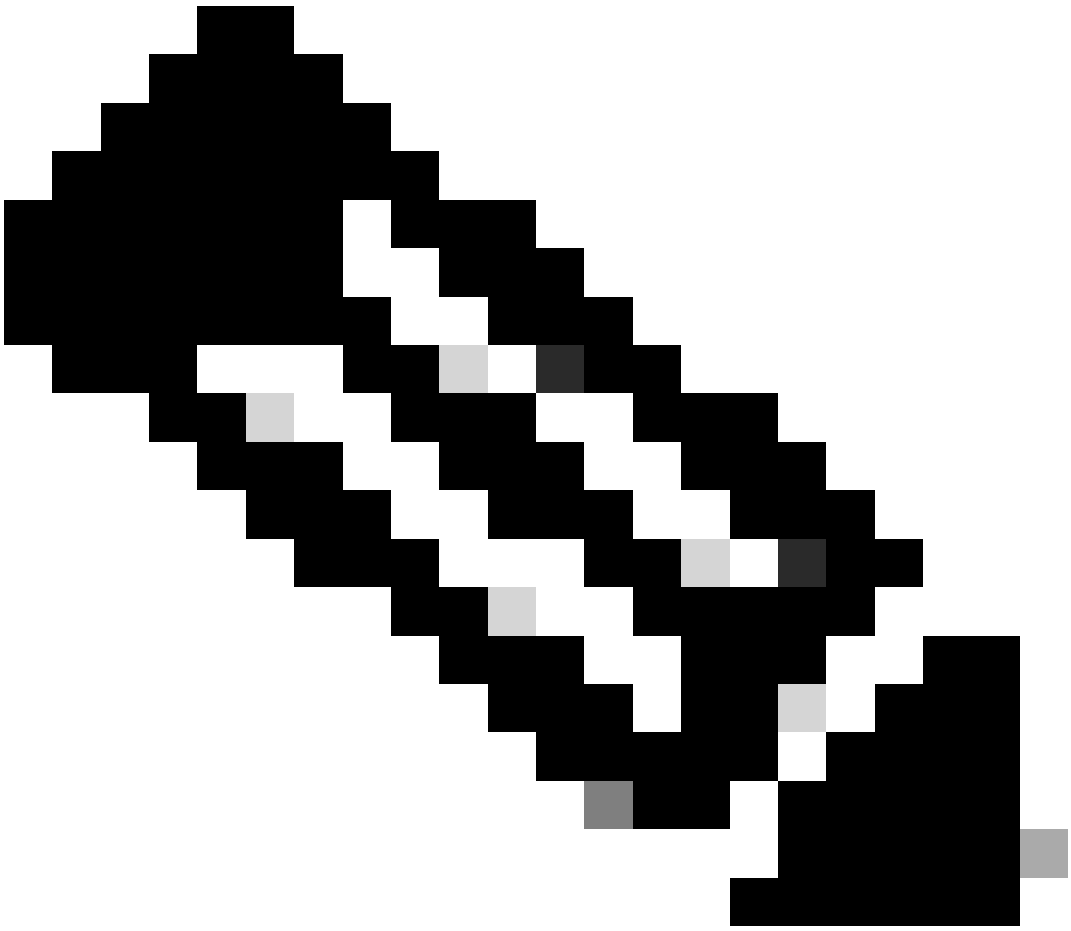
```
debug platform condition interface gig 0/0/1 ipv4
 access-list 150 ingress
```

条件付きデバッグを開始するには、次のコマンドを入力します。

<#root>

```
ASR1000#
```

```
debug platform condition start
```



注：条件付きデバッグインフラストラクチャを停止または無効にするには、debug platform condition stopコマンドを入力します。

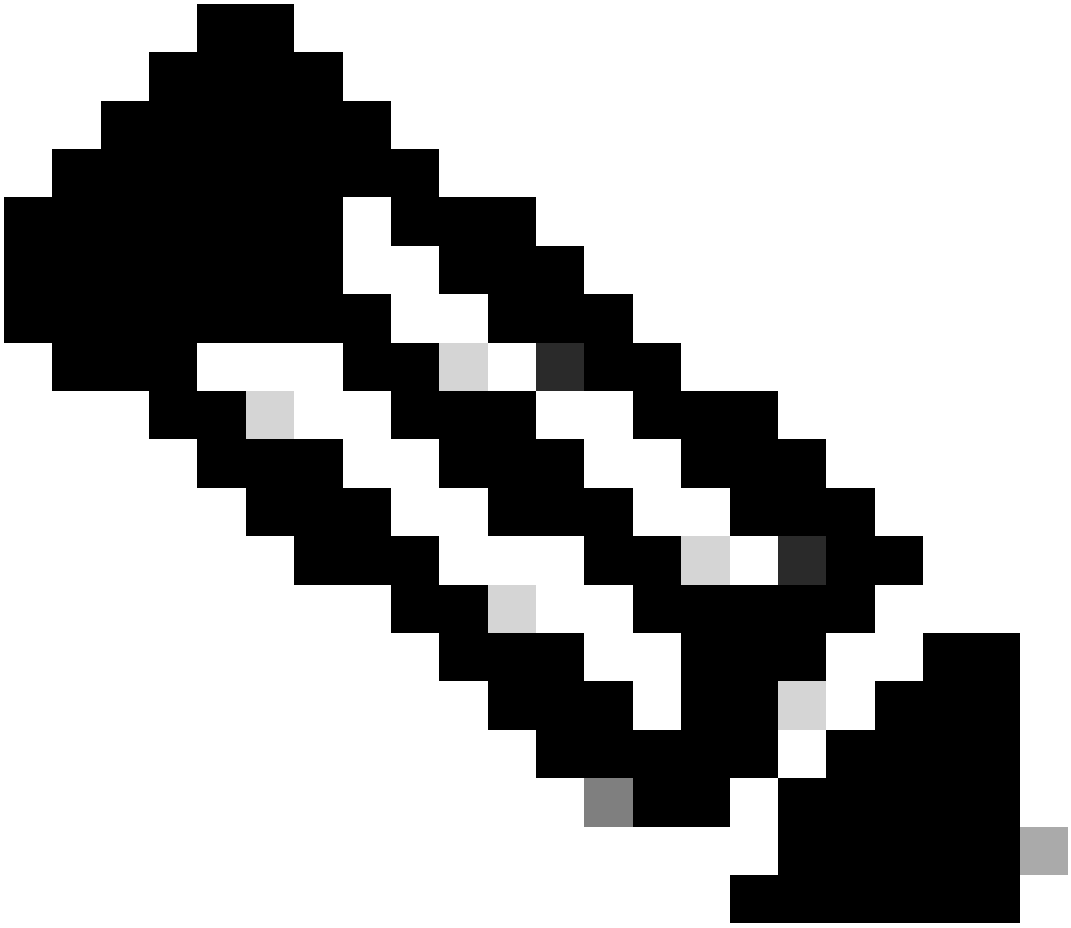
設定されている条件付きデバッグフィルタを表示するには、次のコマンドを入力します。

```
<#root>

ASR1000#

show platform conditions


Conditional Debug Global State:

Start


Conditions                                                                  Direction
----------------------------------------------------------------------------|---------
GigabitEthernet0/0/1                      & IPV4 ACL [150]                   ingress


Feature Condition        Format                  Value
-----------------------|-----------------------|--------------------------------

ASR1000#
```

要約すると、次の設定はこれまでに適用されています。

```
<#root>

access-list 150 permit icmp host 172.16.10.2 host 172.16.20.2

debug platform condition interface gig 0/0/1 ipv4 access-list 150 ingress
debug platform condition start
```

パケットトレースの有効化



注：このセクションでは、パケットオプションとコピーオプションについて詳しく説明します。その他のオプションについては、このドキュメントの後半で説明します。

パケットトレースは、物理インターフェイスと論理インターフェイス（トンネルインターフェイスや仮想アクセスインターフェイスなど）の両方でサポートされています。

パケットトレースのCLI構文を次に示します。

<#root>

```
ASR1000#

debug platform packet-trace

 ?
  copy    Copy packet data
  drop    Trace drops only
  inject  Trace injects only
  packet  Packet count
  punt    Trace punts only
```

<#root>

```
debug platform packet-trace packet <pkt-size/pkt-num> [fia-trace | summary-only]
 [circular] [data-size <data-size>]
```

このコマンドのキーワードの説明を次に示します。

- pkt-num:Packet Numberは、同時に維持されるパケットの最大数を指定します。
- summary-only：サマリーデータのみがキャプチャされるように指定します。デフォルトでは、サマリーデータと機能パスデータの両方がキャプチャされます。
- fia-trace：オプションで、パスデータ情報に加えてFIAトレースを実行します。
- data-size：パスデータバッファのサイズを2,048 ~ 16,384バイトで指定できます。デフォルトは 2,048 バイトです。

<#root>

```
debug platform packet-trace copy packet {in | out | both} [L2 | L3 | L4]
 [size <num-bytes>]
```

このコマンドのキーワードの説明を次に示します。

- in/out：コピーするパケットフローの方向（入力または出力）を指定します。
- L2/L3/L4：パケットのコピーを開始する場所を指定できます。レイヤ2(L2)がデフォルトの場所です。
- size：コピーするオクテットの最大数を指定できます。デフォルトは64オクテットです。

次の例では、条件付きデバッグインフラストラクチャで選択されたトラフィックのパケットトレースを有効にするために使用するコマンドを示します。

<#root>

ASR1000#

```
debug platform packet-trace packet 16
```

パケットトレースの設定を確認するには、次のコマンドを入力します。

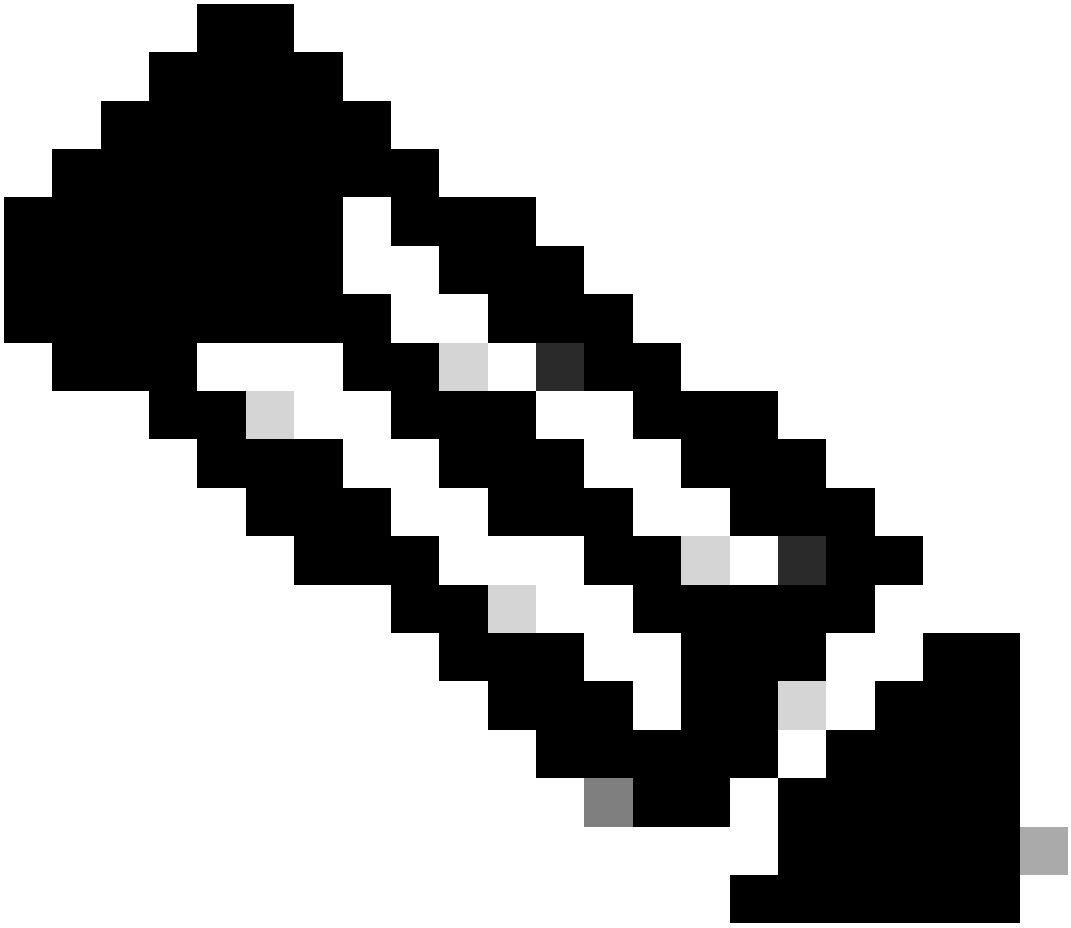
<#root>

```
ASR1000#
show platform packet-trace configuration

debug platform packet-trace packet 16 data-size 2048
```

show debuggingコマンドを入力して、プラットフォーム条件付きデバッグとパケットトレース設定の両方を表示することもできます。

<#root>

```
ASR1000#
show debugging

IOSXE Conditional Debug Configs:


Conditional Debug Global State: Start

Conditions
                                                                Direction
------------------------------------------------------------------------------|---------
GigabitEthernet0/0/1                      & IPV4 ACL [150]                     ingress
...
IOSXE Packet Tracing Configs:

Feature Condition        Format                  Value
----------------------|----------------------|--------------------------------

Feature  Type           Submode                                          Level
-------|--------------|-----------------------------------------------|----------

IOSXE Packet Tracing Configs:

debug platform packet-trace packet 16 data-size 2048
```



注：すべてのプラットフォームデバッグ条件、パケットトレース設定、およびパケットトレース設定をクリアするには、clear platform condition allコマンドを入力します。

要約すると、パケットトレースを有効にするために、これまでに次の設定データが使用されています。

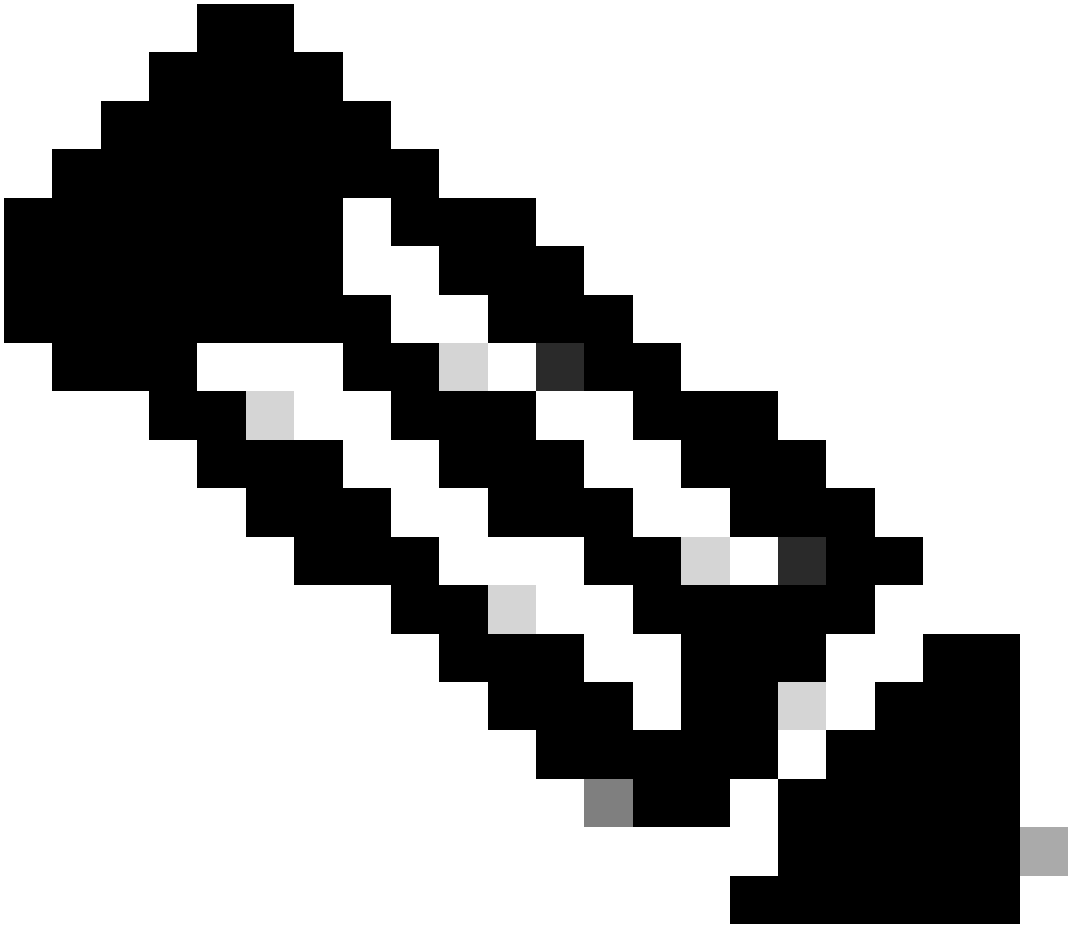
<#root>

```
debug platform packet-trace packet 16
```

パケットトレースに関する出力条件の制限

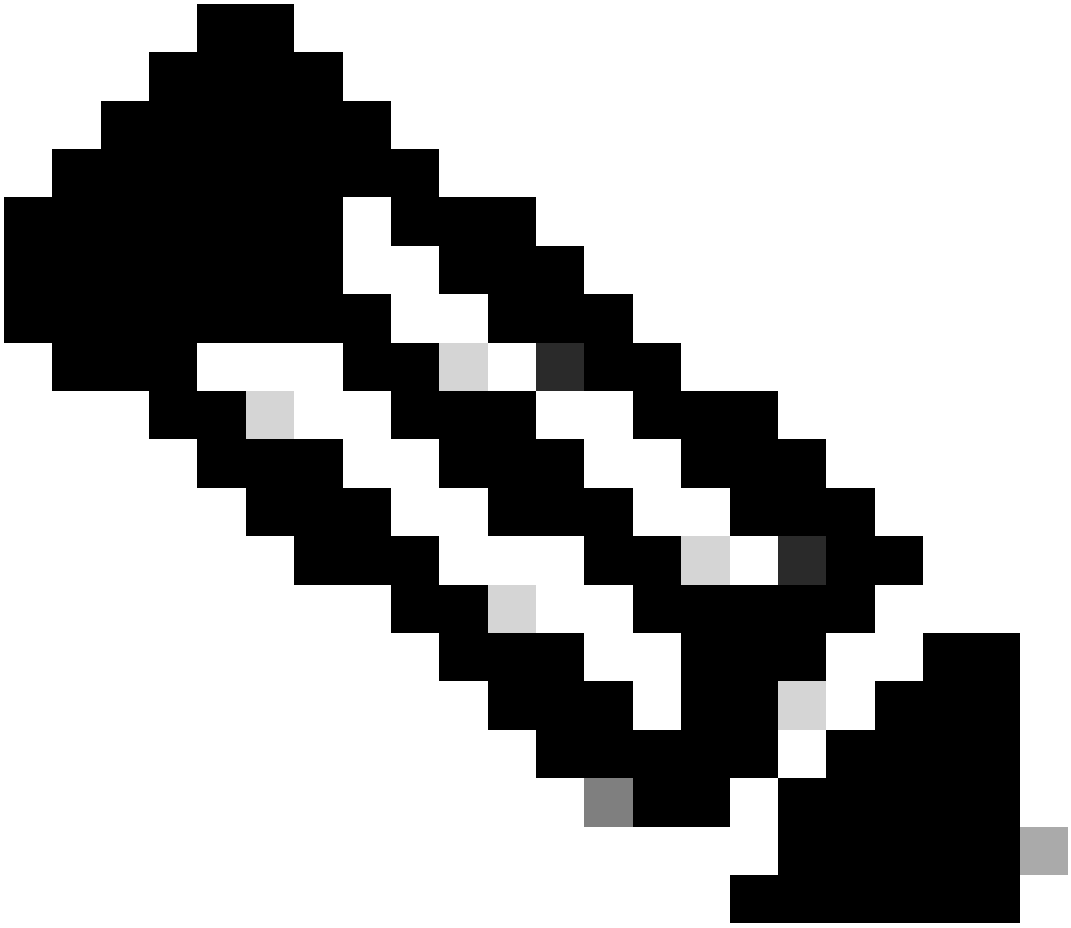
条件は、条件フィルタと、パケットに条件フィルタを適用するタイミングを定義します。たとえば、debug platform condition interface g0/0/0 egressは、パケットがインターフェイスg0/0/0の出力FIAに到達したときに一致したと識別されることを意味します。そのため、入力からそのポイン

トまで行われるすべてのパケット処理が失われます。



注：パケットトレースに入力条件を使用して、できるだけ完全で意味のあるデータを取得することを強く推奨します。出力条件を使用できますが、制限事項に注意してください。

パケットトレースの結果の表示



注：このセクションでは、パストレースが有効であることを前提としています。

パケットトレースでは、次の3つのレベルのインスペクションが提供されます。

- アカウンティング
- パケットごとの要約
- パケットごとのパスデータ

5つのICMP要求パケットが172.16.10.2から172.16.20.2に送信される場合は、パケットトレースの結果を表示するために次のコマンドを使用できます。

<#root>

ASR1000#

show platform packet-trace statistics

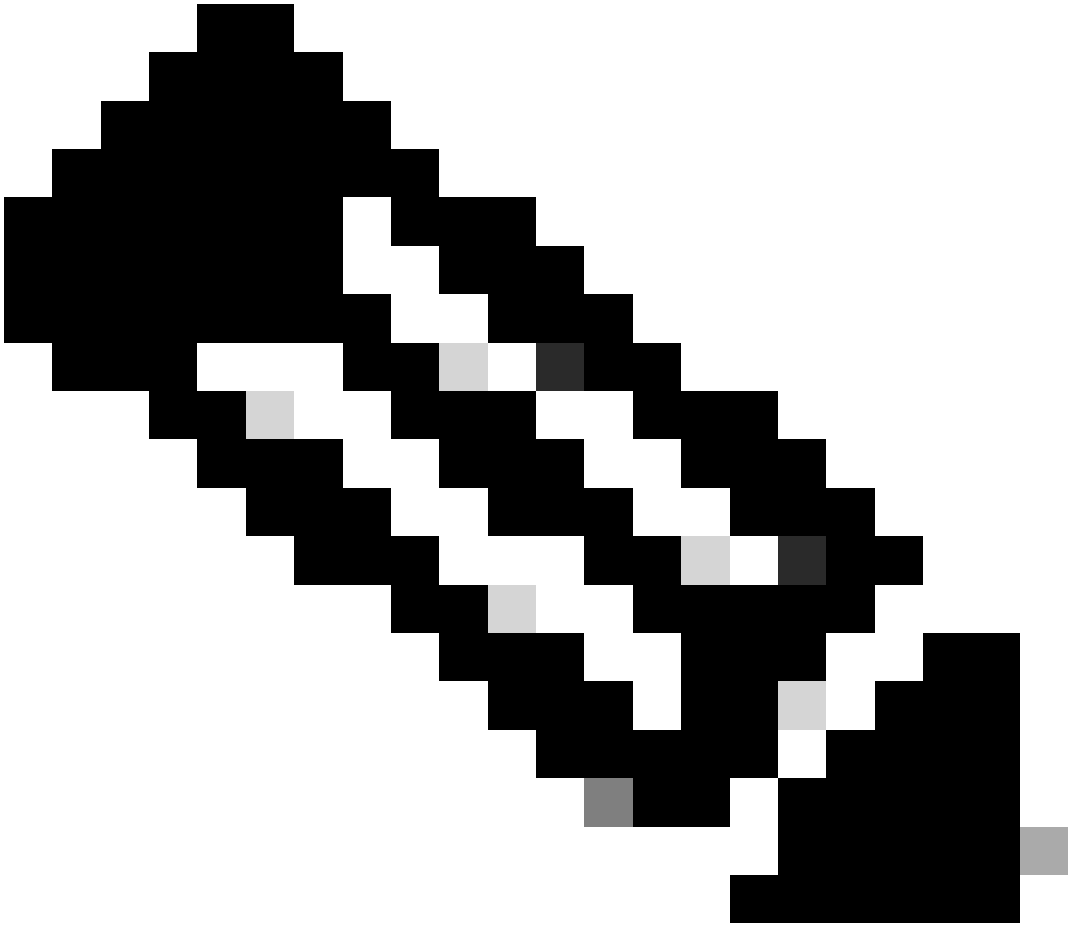
```
Packets Traced: 5


 Ingress  5
 Inject   0
 Forward  5
 Punt     0
 Drop     0
 Consume  0

 ASR1000#
```

```
show platform packet-trace summary
```

```
Pkt

   Input            Output           State  Reason


0

     Gi0/0/1          Gi0/0/0          FWD
 1     Gi0/0/1          Gi0/0/0          FWD
 2     Gi0/0/1          Gi0/0/0          FWD
 3     Gi0/0/1          Gi0/0/0          FWD
 4     Gi0/0/1          Gi0/0/0          FWD

 ASR1000#
```

```
show platform packet-trace packet 0
```

```
 Packet: 0

          CBUG ID: 4
 Summary
 Input     : GigabitEthernet0/0/1
 Output    : GigabitEthernet0/0/0
 State     : FWD
 Timestamp
    Start   : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
    Stop    : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
 Path Trace
 Feature: IPV4
 Source      : 172.16.10.2
 Destination : 172.16.20.2
 Protocol    : 1 (ICMP)

 ASR1000#
```



注:3番目のコマンドは、各パケットのパケットトレースを表示する方法を示す例です。この例では、トレースされた最初のパケットが表示されます。

これらの出力から、5個のパケットがトレースされ、入力インターフェイス、出力インターフェイス、状態、およびパストレースを確認できます。

都道府県	備考
転送	パケットは、出力インターフェイス経由でネクストホップに転送されるように、配信のためにスケジューリング/キューイングされます。
パント	パケットはフォワーディングプロセッサ(FP)からルートプロセッサ(RP)（コントロールプレーン）にパントされます。
DROP	パケットはFPでドロップされます。ドロップの理由の詳細を調べるには、FIAトレースを実行するか、グローバルドロップカウンタを使用するか、またはデータパスデバッグを使用します。
短所	パケットは、ICMP ping要求や暗号化パケットなどのパケットプロセス中に消費されま

	す。

パケットトレース統計情報の出力にあるingressカウンタとinjectカウンタは、外部インターフェイス経由で入力されたパケットとコントロールプレーンから注入されたパケットにそれぞれ対応しています。

FIAトレース

FIAは、パケットが入力または出力で転送されるときに、Quantum Flow Processor(QFP)内のPacket Processor Engine(PPE)によって順次実行される機能のリストを保持します。機能は、マシンに適用される設定データに基づいています。したがって、FIAトレースは、パケットの処理中にシステムを通過するパケットのフローを理解するのに役立ちます。

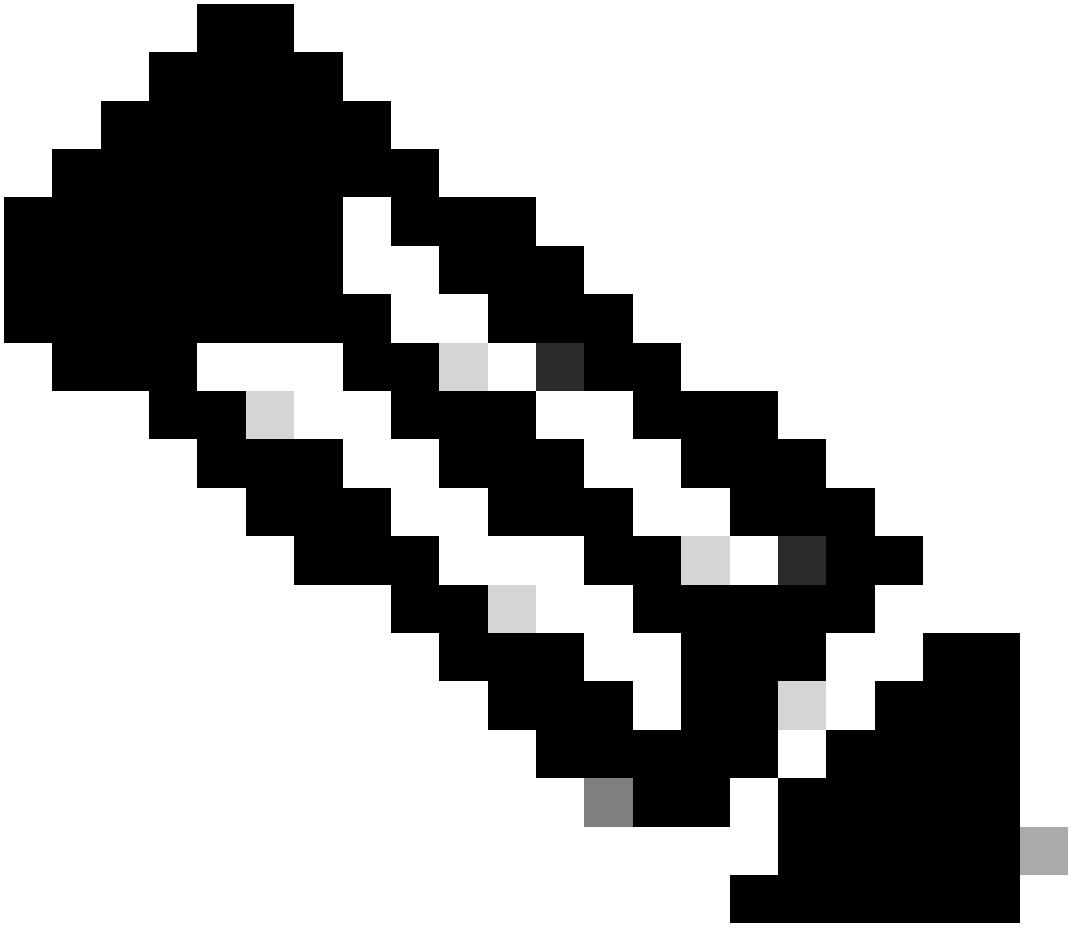
FIAでパケットトレースを有効にするには、次の設定データを適用する必要があります。

<#root>

ASR1000#

debug platform packet-trace packet 16 fia-trace

パケットトレースの結果の表示



注：このセクションでは、FIAトレースが有効であることを前提としています。また、現在のパケットトレースコマンドを追加または変更すると、バッファリングされたパケットトレースの詳細がクリアされるため、一部のトラフィックを再度、送信してトレースする必要があります。

前のセクションで説明したように、FIAトレースを有効にするために使用するコマンドを入力した後、172.16.10.2から172.16.20.2に5つのICMPパケットを送信します。

<#root>

ASR1000#

show platform packet-trace summary

```
Pkt   Input           Output          State  Reason
0     Gi0/0/1         Gi0/0/0         FWD
1     Gi0/0/1         Gi0/0/0         FWD
2     Gi0/0/1         Gi0/0/0         FWD
3     Gi0/0/1         Gi0/0/0         FWD
```

```
4     Gi0/0/1          Gi0/0/0          FWD

ASR1000#
```

show platform packet-trace packet 0

```
Packet: 0           CBUG ID: 9
Summary
  Input     : GigabitEthernet0/0/1
  Output    : GigabitEthernet0/0/0
  State     : FWD
  Timestamp
    Start   : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
    Stop    : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
Path Trace
  Feature: IPV4
    Source      : 172.16.10.2
    Destination : 172.16.20.2
    Protocol    : 1 (ICMP)
  Feature: FIA_TRACE
    Entry     : 0x8059dbe8 - DEBUG_COND_INPUT_PKT
    Timestamp : 3685243309297
  Feature: FIA_TRACE
    Entry     : 0x82011a00 - IPV4_INPUT_DST_LOOKUP_CONSUME
    Timestamp : 3685243311450
  Feature: FIA_TRACE
    Entry     : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
    Timestamp : 3685243312427
  Feature: FIA_TRACE
    Entry     : 0x82004b68 - IPV4_OUTPUT_LOOKUP_PROCESS
    Timestamp : 3685243313230
  Feature: FIA_TRACE
    Entry     : 0x8034f210 - IPV4_INPUT_IPOPTIONS_PROCESS
    Timestamp : 3685243315033
  Feature: FIA_TRACE
    Entry     : 0x82013200 - IPV4_OUTPUT_GOTO_OUTPUT_FEATURE
    Timestamp : 3685243315787
  Feature: FIA_TRACE
    Entry     : 0x80321450 - IPV4_VFR_REFRAG
    Timestamp : 3685243316980
  Feature: FIA_TRACE
    Entry     : 0x82014700 - IPV6_INPUT_L2_REWRITE
    Timestamp : 3685243317713
  Feature: FIA_TRACE
    Entry     : 0x82000080 - IPV4_OUTPUT_FRAG
    Timestamp : 3685243319223
  Feature: FIA_TRACE
    Entry     : 0x8200e500 - IPV4_OUTPUT_DROP_POLICY
    Timestamp : 3685243319950
  Feature: FIA_TRACE
    Entry     : 0x8059aff4 - PACTRAC_OUTPUT_STATS
    Timestamp : 3685243323603
  Feature: FIA_TRACE
    Entry     : 0x82016100 - MARMOT_SPA_D_TRANSMIT_PKT
    Timestamp : 3685243326183

ASR1000#
```

インターフェイスに関連付けられたFIAのチェック

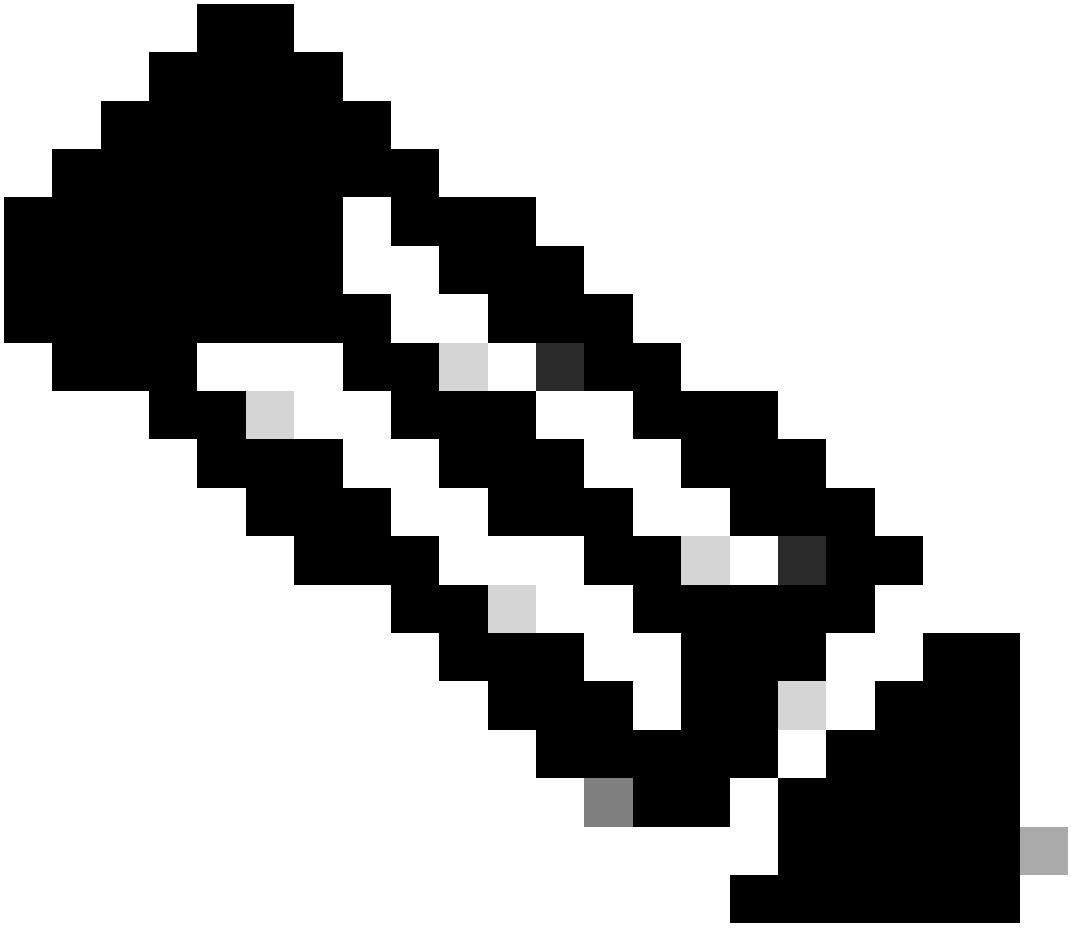
プラットフォーム条件付きデバッグを有効にすると、条件付きデバッグが機能としてFIAに追加されます。インターフェイスでの処理の機能順序に基づいて、条件フィルタを適宜設定する必要があります。たとえば、事前NATアドレスと事後NATアドレスのどちらを条件フィルタで使用する必要があるかなどです。

次の出力は、入力方向で有効になっているプラットフォーム条件付きデバッグに対するFIAの機能の順序を示しています。

<#root>

```
ASR1000#

show platform hardware qfp active interface if-name GigabitEthernet 0/0/1



 General interface information
 Interface Name: GigabitEthernet0/0/1
 Interface state: VALID
 Platform interface handle: 10
 QFP interface handle: 8
 Rx uidb: 1021
 Tx uidb: 131064
 Channel: 16
 Interface Relationships

 BGPPA/QPPB interface configuration information
 Ingress: BGPPA/QPPB not configured. flags: 0000
 Egress : BGPPA not configured. flags: 0000

 ipv4_input enabled.
 ipv4_output enabled.
 layer2_input enabled.
 layer2_output enabled.
 ess_ac_input enabled.

 Features Bound to Interface:
 2 GIC FIA state
 48 PUNT INJECT DB
 39 SPA/Marmot server
 40 ethernet
 1 IFM
 31 icmp_svr
 33 ipfrag_svr
 34 ipreass_svr
 36 ipvfr_svr
 37 ipv6vfr_svr
 12 CPP IPSEC
 Protocol 0 - ipv4_input
 FIA handle - CP:0x108d99cc  DP:0x8070f400
 IPV4_INPUT_DST_LOOKUP_ISSUE (M)
 IPV4_INPUT_ARL_SANITY (M)


  CBUG_INPUT_FIA
```

```
 DEBUG_COND_INPUT_PKT


IPV4_INPUT_DST_LOOKUP_CONSUME (M)
IPV4_INPUT_FOR_US_MARTIAN (M)
IPV4_INPUT_IPSEC_CLASSIFY
IPV4_INPUT_IPSEC_COPROC_PROCESS
IPV4_INPUT_IPSEC_RERUN_JUMP
IPV4_INPUT_LOOKUP_PROCESS (M)
IPV4_INPUT_IPOPTIONS_PROCESS (M)
IPV4_INPUT_GOTO_OUTPUT_FEATURE (M)
Protocol 1 - ipv4_output
FIA handle - CP:0x108d9a34  DP:0x8070eb00
IPV4_OUTPUT_VFR
MC_OUTPUT_GEN_RECYCLE (D)
IPV4_VFR_REFRAG (M)
IPV4_OUTPUT_IPSEC_CLASSIFY
IPV4_OUTPUT_IPSEC_COPROC_PROCESS
IPV4_OUTPUT_IPSEC_RERUN_JUMP
IPV4_OUTPUT_L2_REWRITE (M)
IPV4_OUTPUT_FRAG (M)
IPV4_OUTPUT_DROP_POLICY (M)
PACTRAC_OUTPUT_STATS
MARMOT_SPA_D_TRANSMIT_PKT
DEF_IF_DROP_FIA (M)
Protocol 8 - layer2_input
FIA handle - CP:0x108d9bd4  DP:0x8070c700
LAYER2_INPUT_SIA (M)
CBUG_INPUT_FIA
DEBUG_COND_INPUT_PKT
LAYER2_INPUT_LOOKUP_PROCESS (M)
LAYER2_INPUT_GOTO_OUTPUT_FEATURE (M)
Protocol 9 - layer2_output
FIA handle - CP:0x108d9658  DP:0x80714080
LAYER2_OUTPUT_SERVICEWIRE (M)
LAYER2_OUTPUT_DROP_POLICY (M)
PACTRAC_OUTPUT_STATS
MARMOT_SPA_D_TRANSMIT_PKT
DEF_IF_DROP_FIA (M)
Protocol 14 - ess_ac_input
FIA handle - CP:0x108d9ba0  DP:0x8070cb80
PPPOE_GET_SESSION
ESS_ENTER_SWITCHING
PPPOE_HANDLE_UNCLASSIFIED_SESSION
DEF_IF_DROP_FIA (M)

QfpEth Physical Information
DPS Addr: 0x11215eb8
Submap Table Addr: 0x00000000
VLAN Ethertype: 0x8100
QOS Mode: Per Link

ASR1000#
```



注:CBUG_INPUT_FIAとDEBUG_COND_INPUT_PKTは、ルータに設定されている条件付きデバッグ機能に対応します。

トレースされたパケットのダンプ

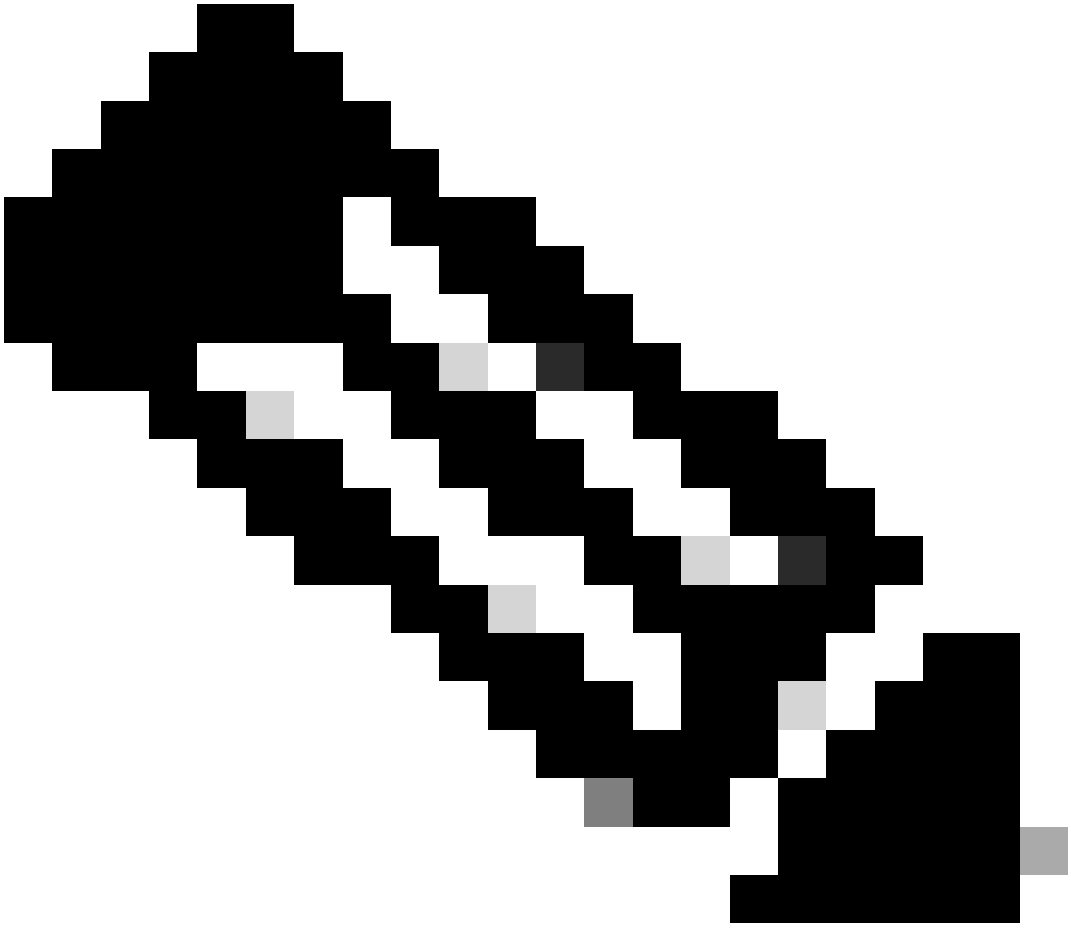
このセクションで説明するように、トレースされたパケットはコピーおよびダンプできます。次の例は、入力方向(172.16.10.2 ~ 172.16.20.2)に最大2,048バイトのパケットをコピーする方法を示しています。

必要な追加コマンドを次に示します。

<#root>

ASR1000#

debug platform packet-trace copy packet input size 2048



注：コピーされるパケットのサイズは、16 ~ 2,048バイトの範囲です。

コピーされたパケットをダンプするには、次のコマンドを入力します。

<#root>

ASR1000#

show platform packet-trace packet 0

```
 Packet: 0           CBUG ID: 14
 Summary
 Input     : GigabitEthernet0/0/1
 Output    : GigabitEthernet0/0/0
 State     : FWD
 Timestamp
     Start   : 1819281992118 ns (05/17/2014 06:40:01.207240 UTC)
     Stop    : 1819282095121 ns (05/17/2014 06:40:01.207343 UTC)
 Path Trace
 Feature: IPV4
```

```
 Source      : 172.16.10.2
 Destination : 172.16.20.2
 Protocol    : 1 (ICMP)
 Feature: FIA_TRACE
 Entry     : 0x8059dbe8 - DEBUG_COND_INPUT_PKT
 Timestamp : 4458180580929

 <some content excluded>

 Feature: FIA_TRACE
 Entry     : 0x82016100 - MARMOT_SPA_D_TRANSMIT_PKT
 Timestamp : 4458180593896
```

Packet Copy In

```
 a4934c8e 33020023 33231379 08004500 00640160 0000ff01 5f16ac10 0201ac10
 01010800 1fd40024 00000000 000184d0 d980abcd abcdabcd abcdabcd abcdabcd
 abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd abcdabcd
 abcdabcd abcdabcd abcdabcd abcdabcd abcd

 ASR1000#
```

トレースの削除

ドロップトレースは、Cisco IOS XEソフトウェアリリース3.11以降で使用できます。ドロップされたパケットに対してのみパケットトレースを有効にするこの機能の主な特徴は次のとおりです。

- オプションで、特定のドロップコードに対するパケットの保持期間を指定できます。
- ドロップイベントをキャプチャするために、グローバル条件またはインターフェイス条件なしで使用できます。
- ドロップイベントキャプチャとは、パケットの寿命ではなく、ドロップ自体だけをトレースすることを意味します。ただし、条件を調整したり、次のデバッグステップの手がかりを提供したりするために、サマリーデータ、タプルデータ、およびパケットをキャプチャすることはできます。

ドロップタイプのパケットトレースを有効にするために使用するコマンド構文を次に示します。

<#root>

debug platform packet-trace drop [code <code-num>]

廃棄コードは廃棄IDと同じで、show platform hardware qfp active statistics drop detailコマンドの出力に表示されます。

<#root>

```
ASR1000#
```

```
show platform hardware qfp active statistics drop detail

------------------------------------------------------------------------------
   ID
  Global Drop Stats                          Packets                  Octets
------------------------------------------------------------------------------
   60
  IpTtlExceeded                                3                     126
    8
  Ipv4Acl                                     32                    3432
```

ドロップトレースシナリオの例

172.16.10.2から172.16.20.2にトラフィックをドロップするには、ASR1KのGig 0/0/0インターフェイスに次のACLを適用します。

```
access-list 199 deny ip host 172.16.10.2 host 172.16.20.2
access-list 199 permit ip any any
interface Gig 0/0/0
  ip access-group 199 out
```

ローカルホストからリモートホストへのトラフィックをドロップするACLを設定した状態で、次のドロップトレース設定を適用します。

<#root>

```
debug platform condition interface Gig 0/0/1 ingress


debug platform condition start


debug platform packet-trace packet 1024 fia-trace


debug platform packet-trace drop
```

172.16.10.2から172.16.20.2に5つのICMP要求パケットを送信します。次に示すように、廃棄トレースはACLによって廃棄されるこれらのパケットをキャプチャします。

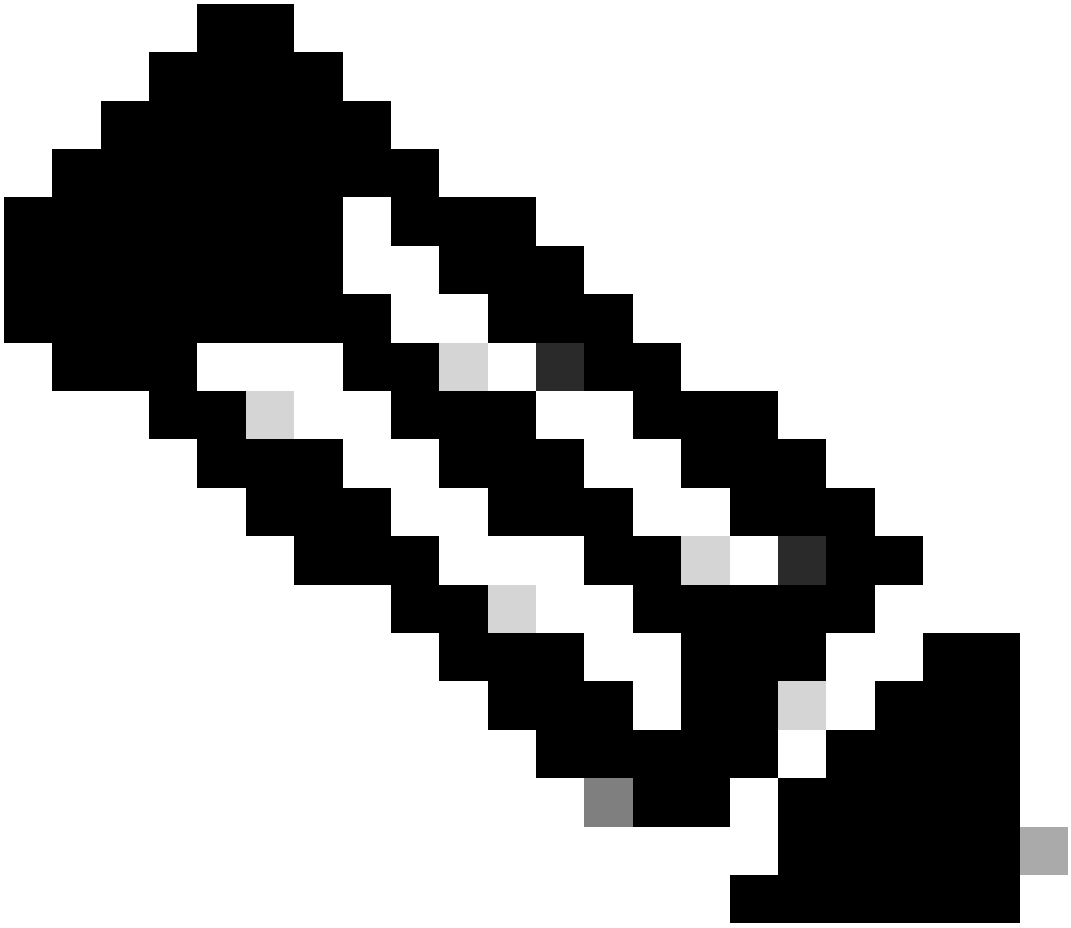
<#root>

```
ASR1000#
show platform packet-trace statistics


 Packets Summary
 Matched  5
 Traced   5
 Packets Received
 Ingress  5
 Inject   0
 Packets Processed
 Forward  0
 Punt     0


Drop     5
 Count       Code  Cause
 5           8     Ipv4Acl


 Consume  0


 ASR1000#
show platform packet-trace summary


 Pkt   Input           Output          State  Reason
 0     Gi0/0/1         Gi0/0/0         DROP   8   (Ipv4Acl)
 1     Gi0/0/1         Gi0/0/0         DROP   8   (Ipv4Acl)
 2     Gi0/0/1         Gi0/0/0         DROP   8   (Ipv4Acl)
 3     Gi0/0/1         Gi0/0/0         DROP   8   (Ipv4Acl)
 4     Gi0/0/1         Gi0/0/0         DROP   8   (Ipv4Acl)


 ASR1K#
debug platform condition stop



 ASR1K#
show platform packet-trace packet 0


 Packet: 0           CBUG ID: 140
 Summary
 Input     : GigabitEthernet0/0/1
 Output    : GigabitEthernet0/0/0


State     : DROP 8   (Ipv4Acl)


 Timestamp
 Start   : 1819281992118 ns (05/17/2014 06:42:01.207240 UTC)
 Stop    : 1819282095121 ns (05/17/2014 06:42:01.207343 UTC)
 Path Trace
 Feature: IPV4
 Source      : 172.16.10.2
 Destination : 172.16.20.2
 Protocol    : 1 (ICMP)
```

```
Feature: FIA_TRACE
Entry      : 0x806c7eac - DEBUG_COND_INPUT_PKT
Lapsed time: 1031 ns
Feature: FIA_TRACE
Entry      : 0x82011c00 - IPV4_INPUT_DST_LOOKUP_CONSUME
Lapsed time: 657 ns
Feature: FIA_TRACE
Entry      : 0x806a2698 - IPV4_INPUT_ACL
Lapsed time: 2773 ns
Feature: FIA_TRACE
Entry      : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
Lapsed time: 1013 ns
Feature: FIA_TRACE
Entry      : 0x82004500 - IPV4_OUTPUT_LOOKUP_PROCESS
Lapsed time: 2951 ns
Feature: FIA_TRACE
Entry      : 0x8041771c - IPV4_INPUT_IPOPTIONS_PROCESS
Lapsed time: 373 ns
Feature: FIA_TRACE
Entry      : 0x82013400 - MPLS_INPUT_GOTO_OUTPUT_FEATURE
Lapsed time: 2097 ns
Feature: FIA_TRACE
Entry      : 0x803c60b8 - IPV4_MC_OUTPUT_VFR_REFRAG
Lapsed time: 373 ns
Feature: FIA_TRACE
Entry      : 0x806db148 - OUTPUT_DROP
Lapsed time: 1297 ns
Feature: FIA_TRACE
Entry      : 0x806a0c98 - IPV4_OUTPUT_ACL
Lapsed time: 78382 ns

ASR1000#
```

トレースの挿入とパント

インジェクトおよびパントパケットトレース機能は、Cisco IOS XEソフトウェアリリース3.12以降で、パント（コントロールプレーンにパントされたFPで受信されるパケット）およびインジェクト（コントロールプレーンからFPにインジェクトされるパケット）パケットをトレースするために追加されました。



注：パントトレースは、ドロップトレースと同様に、グローバル条件またはインターフェイス条件がなくても機能します。ただし、挿入トレースを機能させるには、条件を定義する必要があります。

ASR1Kから隣接ルータにpingを実行したときのpunt nject packet trace およびiの例を次に示します。

<#root>

ASR1000#

debug platform condition ipv4 172.16.10.2/32 both

```
 ASR1000#
```

debug platform condition start

```
 ASR1000#
```

debug platform packet-trace punt

```
 ASR1000#
```

debug platform packet-trace inject

```
 ASR1000#
```

debug platform packet-trace packet 16

```
 ASR1000#
 ASR1000#ping 172.16.10.2
 Type escape sequence to abort.
 Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
 !!!!!
 Success rate is 100 percent (5/5), round-trip min/avg/max = 14/14/15 ms
 ASR1000#
```

ここで、punt およびnject trace rの結果を確認できます。

<#root>

ASR1000#

show platform packet-trace summary

Pkt	Input	Output	State	Reason
0	INJ.2	Gi0/0/1	FWD	
1	Gi0/0/1	internal0/0/rp:0	PUNT	11 (For-us data)
2	INJ.2	Gi0/0/1	FWD	
3	Gi0/0/1	internal0/0/rp:0	PUNT	11 (For-us data)
4	INJ.2	Gi0/0/1	FWD	
5	Gi0/0/1	internal0/0/rp:0	PUNT	11 (For-us data)
6	INJ.2	Gi0/0/1	FWD	
7	Gi0/0/1	internal0/0/rp:0	PUNT	11 (For-us data)
8	INJ.2	Gi0/0/1	FWD	
9	Gi0/0/1	internal0/0/rp:0	PUNT	11 (For-us data)

ASR1000#

show platform packet-trace packet 0

Packet: 0 CBUG ID: 120
Summary

Input : INJ.2

Output : GigabitEthernet0/0/1
State : FWD
Timestamp
Start : 115612780360228 ns (05/29/2014 15:02:55.467987 UTC)
Stop : 115612780380931 ns (05/29/2014 15:02:55.468008 UTC)
Path Trace
Feature: IPV4
Source : 172.16.10.1
Destination : 172.16.10.2
Protocol : 1 (ICMP)

```
ASR1000#
ASR1000#
```

```
show platform packet-trace packet 1
```

```
Packet: 1           CBUG ID: 121
Summary
Input     : GigabitEthernet0/0/1
Output    : internal0/0/rp:0
```

```
State     : PUNT 11  (For-us data)
```

```
Timestamp
Start   : 115612781060418 ns (05/29/2014 15:02:55.468687 UTC)
Stop    : 115612781120041 ns (05/29/2014 15:02:55.468747 UTC)
Path Trace
Feature: IPV4
Source      : 172.16.10.2
Destination : 172.16.10.1
Protocol    : 1 (ICMP)
```

IOSdとLFTSのパント/インジェクトトレースおよびUDFマッチングによるパケットトレースの拡張（17.3.1の新機能）

パケットトレース機能はさらに拡張され、Cisco IOS-XEリリース17.3.1でIOSdまたは他のBinOSプロセスを発信元または宛先とするパケットに対して、追加のトレース情報を提供します。

IOSdドロップトレース

この機能拡張により、パケットトレースがIOSdに拡張され、IOSd内部でのパケットドロップ(通常は***show ip traffic***の出力で報告される)に関する情報を提供できるようになりました。IOSdドロップトレースを有効にするために追加の設定は必要ありません。次に、不正なチェックサムエラーが原因でIOSdによってドロップされたUDPパケットの例を示します。

```
Router#debug platform condition ipv4 10.118.74.53/32 both
Router#debug platform condition start
Router#debug platform packet-trace packet 200
Packet count rounded up from 200 to 256

Router#
Router#show plat pack pa 0
Packet: 0           CBUG ID: 674
Summary
  Input     : GigabitEthernet1
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 17756544435656 ns (06/29/2020 18:19:17.326313 UTC)
    Stop    : 17756544469451 ns (06/29/2020 18:19:17.326346 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet1
    Output      : <unknown>
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Protocol    : 17 (UDP)
      SrcPort   : 2640
      DstPort   : 500

IOSd Path Flow: Packet: 0    CBUG ID: 674
  Feature: INFRA
  Pkt Direction: IN
    Packet Rcvd From DATAPLANE

  Feature: IP
  Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Interface   : GigabitEthernet1

  Feature: IP
  Pkt Direction: IN
  FORWARDED To transport layer
    Source        : 10.118.74.53
    Destination   : 172.18.124.38
    Interface     : GigabitEthernet1

  Feature: UDP
  Pkt Direction: IN
```

DROPPED

UDP: Checksum error: dropping

```
    Source      : 10.118.74.53(2640)
```

```
    Destination : 172.18.124.38(500)
```

IOSd出力パストレース

パケットトレースが拡張され、パストレースとプロトコル処理情報が表示されます。これは、パケットがIOSdから発信され、ネットワークに向けて出力方向で送信されるためです。IOSd出力パストレース情報をキャプチャするために追加の設定は必要ありません。ルータから出力されるSSHパケットの出力パストレースの例を次に示します。

```
     <#root>

Router#show platform packet-trace packet 2
Packet: 2           CBUG ID: 2

IOSd Path Flow:

  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910 OPTS 4 ACK 2346

  Feature: TCP
  Pkt Direction: OUT
  FORWARDED
    TCP: Connection is in SYNRCVD state
    ACK         : 2346709419
    SEQ         : 3052140910
    Source      : 172.18.124.38(22)
    Destination : 172.18.124.55(52774)


  Feature: IP
  Pkt Direction: OUTRoute out the generated packet.srcaddr: 172.18.124.38, dstaddr: 172.18.124.55

  Feature: IP
  Pkt Direction: OUTInject and forward successful srcaddr: 172.18.124.38, dstaddr: 172.18.124.55

  Feature: TCP
  Pkt Direction: OUTtcp0: O SYNRCVD 172.18.124.38:22 172.18.124.55:52774 seq 3052140910 OPTS 4 ACK 2346
Summary
  Input     : INJ.2
  Output    : GigabitEthernet1
  State     : FWD
  Timestamp
```

```
    Start   : 490928006866 ns (06/29/2020 13:31:30.807879 UTC)
    Stop    : 490928038567 ns (06/29/2020 13:31:30.807911 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : internal0/0/rp:0
    Output      : <unknown>
    Source      : 172.18.124.38
    Destination : 172.18.124.55
    Protocol    : 6 (TCP)
      SrcPort   : 22
      DstPort   : 52774
  Feature: IPSec
    Result    : IPSEC_RESULT_DENY
    Action    : SEND_CLEAR
    SA Handle : 0
    Peer Addr : 172.18.124.55
    Local Addr: 172.18.124.38
```

LFTSパケットトレース

LFTS(Linux Forwarding Transport Service)は、CPPからパントされたパケットをIOSd以外のアプリケーションに転送するトランスポートメカニズムです。LFTSパケットトレースの機能拡張により、パストレースの出力に該当するパケットのトレース情報が追加されました。LFTSトレース情報を取得するために追加の設定は必要ありません。NETCONFアプリケーションにパントされたパケットのLFTSトレースの出力例を次に示します。

<#root>

```
Router#show plat packet-trace pac 0
Packet: 0           CBUG ID: 461
Summary
  Input     : GigabitEthernet1
  Output    : internal0/0/rp:0
  State     : PUNT 11  (For-us data)
  Timestamp
    Start   : 647999618975 ns (06/30/2020 02:18:06.752776 UTC)
    Stop    : 647999649168 ns (06/30/2020 02:18:06.752806 UTC)
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet1
    Output      : <unknown>
    Source      : 10.118.74.53
    Destination : 172.18.124.38
    Protocol    : 6 (TCP)
      SrcPort   : 65365
      DstPort   : 830
```

LFTS Path Flow: Packet: 0   CBUG ID: 461

```
  Feature: LFTS
  Pkt Direction: IN
    Punt Cause  : 11
     subCause : 0
```

ユーザ定義フィルタに基づくパケットトレースパターンマッチング（ASR1000プラットフォームのみ）

Cisco IOS XEリリース17.3.1では、ユーザ定義フィルタ(UDF)インフラストラクチャに基づくパケットの任意のフィールドを照合する新しいパケットマッチングメカニズムもASR1000製品ファミリに追加されました。これにより、標準のL2/L3/L4ヘッダー構造に含まれないフィールドに基づく柔軟なパケットマッチングが可能になります。次の例は、L3外部プロトコルヘッダーからの26バイトのオフセットから始まる2バイトのユーザ定義パターン0x4D2と一致するUDF定義を示しています。

```
udf grekey header outer l3 26 2
ip access-list extended match-grekey
 10 permit ip any any udf grekey 0x4D2 0xFFFF

debug plat condition ipv4 access-list match-grekey both
debug plat condition start
debug plat packet-trace pack 100
```

パケットトレースの例

このセクションでは、パケットトレース機能がトラブルシューティングに役立つ例をいくつか紹介します。

パケットトレースの例：NAT

この例では、インターフェイスの発信元ネットワークアドレス変換(NAT)は、ローカルサブネット(172.16.10.0/24)のASR1K(Gig0/0/0)のWANインターフェイスで設定されます。

Gig0/0/0インターフェイスで変換(NAT)される172.16.10.2から172.16.20.2へのトラフィックをトレースするために使用される、プラットフォーム条件とパケットトレースの設定を次に示します。

```
debug platform condition interface Gig 0/0/1 ingress
debug platform condition start
```

```
debug platform packet-trace packet 1024 fia-trace
```

5つのICMPパケットがインターフェイスソースNAT設定を使用して172.16.10.2から172.16.20.2に送信された場合、パケットトレースの結果は次のようになります。

```
      <#root>

ASR1000#
```

show platform packet-trace summary

```
 Pkt   Input           Output          State  Reason
 0     Gi0/0/1         Gi0/0/0         FWD
 1     Gi0/0/1         Gi0/0/0         FWD
 2     Gi0/0/1         Gi0/0/0         FWD
 3     Gi0/0/1         Gi0/0/0         FWD
 4     Gi0/0/1         Gi0/0/0         FWD

ASR1000#
```

show platform packet-trace statistics

```
 Packets Summary
 Matched  5
 Traced   5
 Packets Received
 Ingress  5
 Inject   0
 Packets Processed
 Forward  5
 Punt     0
 Drop     0
 Consume  0

 ASR1000#
```

show platform packet-trace packet 0

```
 Packet: 0           CBUG ID: 146
 Summary
 Input     : GigabitEthernet0/0/1
 Output    : GigabitEthernet0/0/0
 State     : FWD
 Timestamp
 Start   : 3010217805313 ns (05/17/2014 07:01:52.227836 UTC)
 Stop    : 3010217892847 ns (05/17/2014 07:01:52.227923 UTC)
 Path Trace
 Feature: IPV4
 Source      : 172.16.10.2
 Destination : 172.16.20.2
 Protocol    : 1 (ICMP)
 Feature: FIA_TRACE
 Entry      : 0x806c7eac - DEBUG_COND_INPUT_PKT
 Lapsed time: 1031 ns
 Feature: FIA_TRACE
 Entry      : 0x82011c00 - IPV4_INPUT_DST_LOOKUP_CONSUME
 Lapsed time: 462 ns
 Feature: FIA_TRACE
 Entry      : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
 Lapsed time: 355 ns
 Feature: FIA_TRACE
 Entry      : 0x803c6af4 - IPV4_INPUT_VFR
 Lapsed time: 266 ns
 Feature: FIA_TRACE
 Entry      : 0x82004500 - IPV4_OUTPUT_LOOKUP_PROCESS
 Lapsed time: 942 ns
 Feature: FIA_TRACE
 Entry      : 0x8041771c - IPV4_INPUT_IPOPTIONS_PROCESS
 Lapsed time: 88 ns
 Feature: FIA_TRACE
 Entry      : 0x82013400 - MPLS_INPUT_GOTO_OUTPUT_FEATURE
 Lapsed time: 568 ns
 Feature: FIA_TRACE
 Entry      : 0x803c6900 - IPV4_OUTPUT_VFR
 Lapsed time: 266 ns
```

```
Feature: NAT
 Direction   : IN to OUT
 Action      : Translate Source
 Old Address : 172.16.10.2  00028
 New Address : 192.168.10.1 00002
```

```
 Feature: FIA_TRACE
 Entry      : 0x8031c248 - IPV4_NAT_OUTPUT_FIA
 Lapsed time: 55697 ns
 Feature: FIA_TRACE
 Entry      : 0x801424f8 - IPV4_OUTPUT_THREAT_DEFENSE
 Lapsed time: 693 ns
```

```
Feature: FIA_TRACE
Entry      : 0x803c60b8 - IPV4_MC_OUTPUT_VFR_REFRAG
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry      : 0x82014900 - IPV6_INPUT_L2_REWRITE
Lapsed time: 444 ns
Feature: FIA_TRACE
Entry      : 0x82000080 - IPV4_OUTPUT_FRAG
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry      : 0x8200e600 - IPV4_OUTPUT_DROP_POLICY
Lapsed time: 1457 ns
Feature: FIA_TRACE
Entry      : 0x82017980 - MARMOT_SPA_D_TRANSMIT_PKT
Lapsed time: 7431 ns
ASR1000#
```

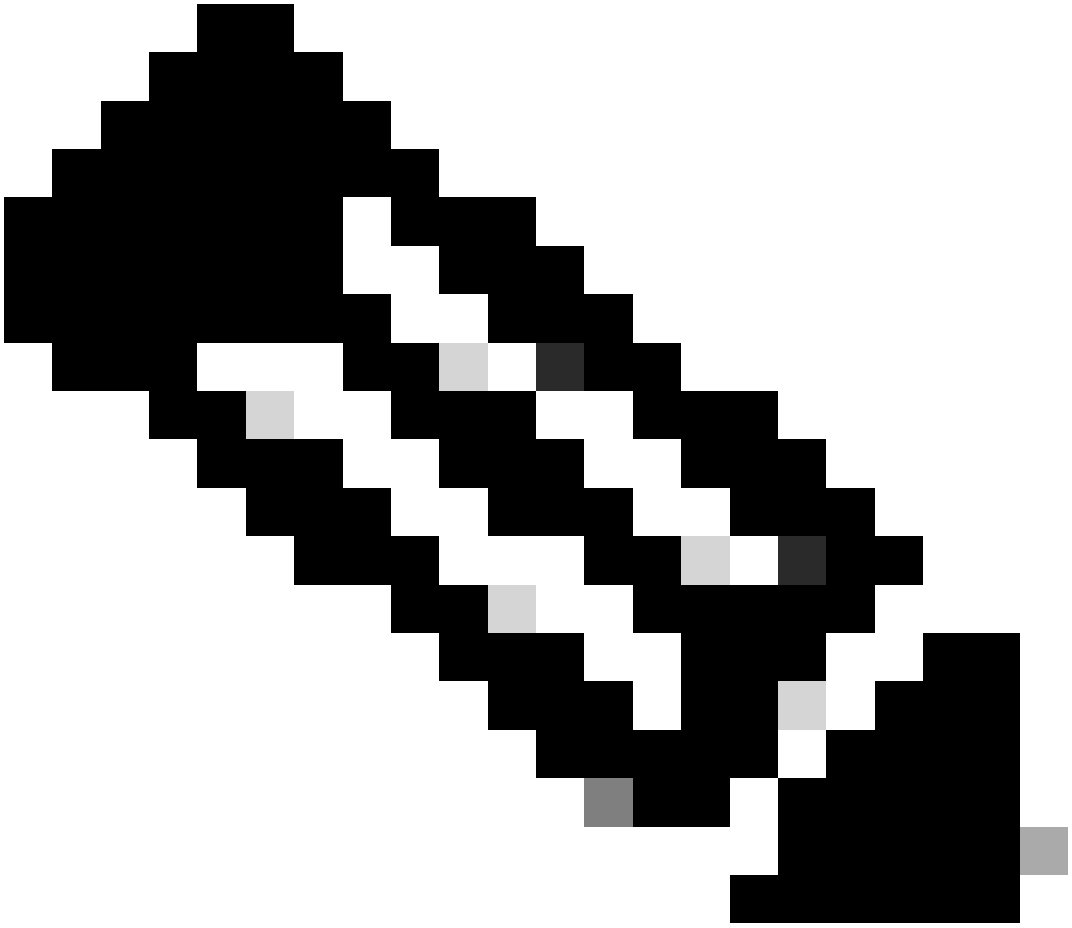
パケットトレースの例：VPN

この例では、ASR1KとCisco IOSルータの間でサイト間VPNトンネルを使用して、172.16.10.0/24（ローカルおよびリモートのサブネット）と172.16.20.0/24（ローカルおよびリモートのサブネット）の間を流れるトラフィックを保護します。

Gig 0/0/1インターフェイスで172.16.10.2から172.16.20.2へ流れるVPNトラフィックをトレースするために使用される、プラットフォーム条件とパケットトレースの設定を次に示します。

```
debug platform condition interface Gig 0/0/1 ingress
debug platform condition start
debug platform packet-trace packet 1024 fia-trace
```

5つのICMPパケットが172.16.10.2から172.16.20.2に送信され、この例ではASR1KとCisco IOSルータ間のVPNトンネルによって暗号化されています。パケットトレースの出力は次のとおりです。



注：パケットトレースでは、パケットの暗号化に使用されるトレース内のQFP Security Association（SA；セキュリティアソシエーション）ハンドルが示されます。これは、IPSec VPNの問題をトラブルシューティングして正しいSAが暗号化に使用されていることを確認する際に役立ちます。

<#root>

ASR1000#

show platform packet-trace summary

```
Pkt   Input            Output           State  Reason
0     Gi0/0/1          Gi0/0/0          FWD
1     Gi0/0/1          Gi0/0/0          FWD
2     Gi0/0/1          Gi0/0/0          FWD
3     Gi0/0/1          Gi0/0/0          FWD
4     Gi0/0/1          Gi0/0/0          FWD

ASR1000#
```

show platform packet-trace packet 0

```
Packet: 0           CBUG ID: 211
Summary
Input     : GigabitEthernet0/0/1
Output    : GigabitEthernet0/0/0
State     : FWD
Timestamp
Start   : 4636921551459 ns (05/17/2014 07:28:59.211375 UTC)
Stop    : 4636921668739 ns (05/17/2014 07:28:59.211493 UTC)
Path Trace
Feature: IPV4
Source      : 172.16.10.2
Destination : 172.16.20.2
Protocol    : 1 (ICMP)
Feature: FIA_TRACE
Entry      : 0x806c7eac - DEBUG_COND_INPUT_PKT
Lapsed time: 622 ns
Feature: FIA_TRACE
Entry      : 0x82011c00 - IPV4_INPUT_DST_LOOKUP_CONSUME
Lapsed time: 462 ns
Feature: FIA_TRACE
Entry      : 0x82000170 - IPV4_INPUT_FOR_US_MARTIAN
Lapsed time: 320 ns
Feature: FIA_TRACE
Entry      : 0x82004500 - IPV4_OUTPUT_LOOKUP_PROCESS
Lapsed time: 1102 ns
Feature: FIA_TRACE
Entry      : 0x8041771c - IPV4_INPUT_IPOPTIONS_PROCESS
Lapsed time: 88 ns
Feature: FIA_TRACE
Entry      : 0x82013400 - MPLS_INPUT_GOTO_OUTPUT_FEATURE
Lapsed time: 586 ns
Feature: FIA_TRACE
Entry      : 0x803c6900 - IPV4_OUTPUT_VFR
Lapsed time: 266 ns
Feature: FIA_TRACE
Entry      : 0x80757914 - MC_OUTPUT_GEN_RECYCLE
Lapsed time: 195 ns
Feature: FIA_TRACE
```

```
 Entry      : 0x803c60b8 - IPV4_MC_OUTPUT_VFR_REFRAG
 Lapsed time: 88 ns
```

```
Feature: IPSec
 Result    : IPSEC_RESULT_SA
 Action    : ENCRYPT
 SA Handle : 6
 Peer Addr : 192.168.20.1
 Local Addr: 192.168.10.1
```

```
 Feature: FIA_TRACE
 Entry      : 0x8043caec - IPV4_OUTPUT_IPSEC_CLASSIFY
 Lapsed time: 9528 ns
 Feature: FIA_TRACE
 Entry      : 0x8043915c - IPV4_OUTPUT_IPSEC_DOUBLE_ACL
 Lapsed time: 355 ns
 Feature: FIA_TRACE
 Entry      : 0x8043b45c - IPV4_IPSEC_FEATURE_RETURN
 Lapsed time: 657 ns
 Feature: FIA_TRACE
 Entry      : 0x8043ae28 - IPV4_OUTPUT_IPSEC_RERUN_JUMP
 Lapsed time: 888 ns
 Feature: FIA_TRACE
 Entry      : 0x80436f10 - IPV4_OUTPUT_IPSEC_POST_PROCESS
 Lapsed time: 2186 ns
 Feature: FIA_TRACE
 Entry      : 0x8043b45c - IPV4_IPSEC_FEATURE_RETURN
 Lapsed time: 675 ns
 Feature: FIA_TRACE
 Entry      : 0x82014900 - IPV6_INPUT_L2_REWRITE
 Lapsed time: 1902 ns
 Feature: FIA_TRACE
 Entry      : 0x82000080 - IPV4_OUTPUT_FRAG
 Lapsed time: 71 ns
 Feature: FIA_TRACE
 Entry      : 0x8200e600 - IPV4_OUTPUT_DROP_POLICY
 Lapsed time: 1582 ns
 Feature: FIA_TRACE
 Entry      : 0x82017980 - MARMOT_SPA_D_TRANSMIT_PKT
 Lapsed time: 3964 ns
 ASR1000#
```

パフォーマンスへの影響

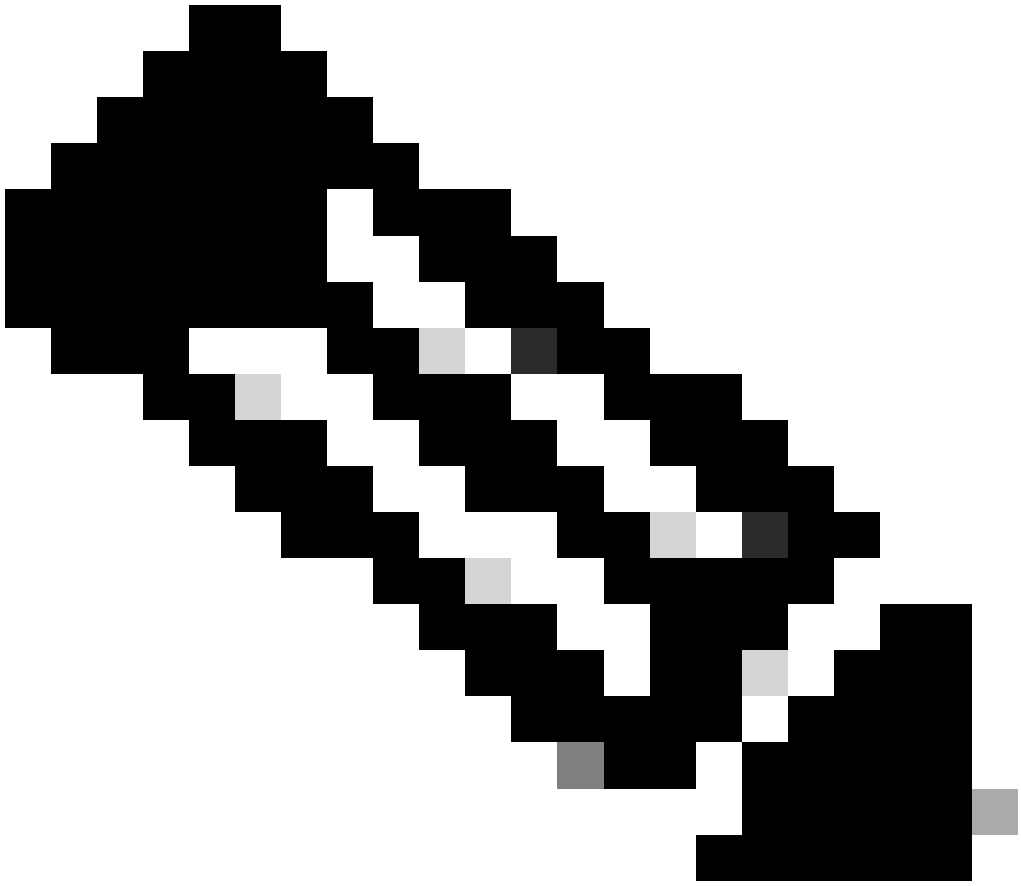
パケットトレースバッファはQFP DRAMを消費するため、設定に必要なメモリ量と使用可能なメモリ量に注意してください。

パフォーマンスへの影響は、有効になっているパケットトレースオプションによって異なります。パケットトレースは、ユーザ設定の条件に一致するパケットなど、トレースされたパケットの転送パフォーマンスにのみ影響します。パケットトレースを設定してキャプチャする情報が詳細に設定されるほど、リソースへの影響が大きくなります。

他のトラブルシューティングと同様に、反復アプローチを採用し、デバッグ状況によって保証される場合にのみ、より詳細なトレースオプションを有効にするのが最善です。

QFP DRAMの使用量は、次の式で推定できます。

必要なメモリ= (統計オーバーヘッド) +パケット数* (サマリーサイズ+パスデータサイズ+コピーサイズ)



注：統計情報のオーバーヘッドは2 KBに、サマリーサイズは128 Bに固定されていますが、パスデータサイズとコピーサイズはユーザが設定できます。

関連情報

- **Cisco ASR1000シリーズアグリゲーションシリーズルータソフトウェアコンフィギュレーションガイド – パケットトレース**
- **Cisco ASR1000シリーズサービスルータでのパケットドロップ**
- シスコのテクニカルサポートとダウンロード

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。