



パケットトレース

初版：2016年8月3日

パケットトレース機能は、Cisco IOS XE プラットフォームによってデータパケットがどのように処理されているのかを詳細に理解できます。これは、ユーザーが問題を診断し、より効率的にトラブルシューティングするために役立ちます。このモジュールは、パケットトレース機能の使用方法に関する情報を提供します。

- [パケットトレースについて, on page 1](#)
- [パケットトレースの設定に関する使用上のガイドライン, on page 2](#)
- [パケットトレースの設定, on page 3](#)
- [UDF オフセットを使用したパケットトレーサの設定, on page 5](#)
- [パケットトレース情報の表示, on page 8](#)
- [パケットトレースデータの削除, on page 8](#)
- [パケットトレースの設定例, on page 9](#)
- [その他の参考資料, on page 16](#)
- [パケットトレースの機能情報, on page 17](#)

パケットトレースについて

パケットトレース機能は、アカウンティング、サマリー、パスデータという3つのレベルのパケット検査を提供します。各レベルは、一部のパケット処理機能を犠牲にして、パケット処理の詳細なビューを提供します。ただし、パケットトレースは、`debug platform condition` ステートメントに一致するパケットの検査を制限し、大量のトラフィックが発生する環境下でも実行可能なオプションです。

次の表で、パケットトレースによって提供される3つのレベルの検査について説明します。

Table 1: パケットトレースレベル

パケットトレースレベル	説明
アカウンティング	パケットトレースのアカウンティングでは、ネットワークプロセッサに出入りするパケット数が示されます。パケットトレースのアカウンティングは、負荷の軽いパフォーマンス アクティビティであり、無効化されるまで継続的に実行されます。
サマリー	パケットトレースのサマリーレベルでは、限られた数のパケットデータが収集されます。パケットトレースのサマリーは、入力インターフェイスと出力インターフェイス、最終的なパケットの状態、およびパケットのパント、ドロップ、インジェクションを随時追跡します。サマリーデータの収集は、通常のパケット処理と比較してパフォーマンスが高く、問題のあるインターフェイスを分離するのに役立ちます。
パスデータ	パケットトレースのパスデータレベルでは、パケットトレースが最も詳細なレベルで実行されます。限られた数のパケットを対象にデータが収集されます。パケットトレースのパスデータでは、条件付きデバッグ ID を含むデータがキャプチャされます。このデータは、機能デバッグ、タイムスタンプ、および機能固有のパスデータと関連付ける際に役立ちます。 パスデータには、パケットコピーと Feature Invocation Array (FIA) トレースという 2 つのオプション機能もあります。パケットコピーオプションを使用すると、パケットの各種レイヤ（レイヤ 2、レイヤ 3、レイヤ 4）で入力パケットや出力パケットをコピーできます。FIA トレースオプションは、パケット処理中に呼び出されたすべての機能エントリを追跡します。このオプションは、パケット処理中に何が起きているかを把握する際に役立ちます。 Note パスデータの収集では、多くのパケット処理リソースが消費されます。また、オプション機能はパケットパフォーマンスに徐々に影響を及ぼします。そのため、パスデータレベルは限定的なキャパシティで使用するか、パケットパフォーマンスの変化が許容できる状況で请使用してください。

パケットトレースの設定に関する使用上のガイドライン

パケットトレース機能を設定する際は、次のベストプラクティスを考慮してください。

- パケットをより包括的に表示するには、パケットトレース機能を使用する際に入力条件を使用することを推奨します。
- パケットトレースの設定には、データプレーンメモリが必要です。データプレーンメモリが制限されているシステムでは、パケットトレース値をどのように選択するかを慎重に検討してください。パケットトレースによって消費されるメモリ量の概算値は、次の式で求められます。

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 + データサイズ + パケットコピーサイズ)。

パケットトレース機能を有効にすると、統計用に少量の固定メモリが割り当てられます。同様に、パケットごとのデータをキャプチャする場合、サマリーデータ用に各パケットに少量の固定メモリが必要です。ただし、式が示すように、トレース対象に選択したパケット数や、パスデータとパケットのコピーを収集するかどうかによって、消費されるメモリ量が大きく影響される可能性があります。

パケットトレースの設定

パケットトレース機能を設定するには、次の手順を実行します。



Note

パケットトレース機能によって消費されるメモリの量は、パケットトレース設定の影響を受けます。通常のサービスの中断を避けるために、パケットごとのパスデータとコピーバッファのサイズ、およびトレースするパケット数を慎重に選択する必要があります。**show platform hardware qfp active infrastructure exmem statistics** コマンドを使用すると、現在のデータプレーンの DRAM メモリ消費量をチェックできます。

SUMMARY STEPS

1. **enable**
2. **debug platform packet-trace packet** *pkt-num* [**fia-trace** | **summary-only**] [**circular**] [**data-size** *data-size*]
3. **debug platform packet-trace** {**punt** | **inject** | **copy** | **drop** | **packet** | **statistics**}
4. **debug platform condition** [**ipv4** | **ipv6**] [**interface** *interface*] [**access-list** *access-list -name* | *ipv4-address / subnet-mask* | *ipv6-address / subnet-mask*] [**ingress** | **egress** | **both**]
5. **debug platform condition start**
6. **debug platform condition stop**
7. **show platform packet-trace** {**configuration** | **statistics** | **summary** | **packet** {**all** | *pkt-num*}}
8. **clear platform condition all**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
ステップ 1	enable Example: Router> enable	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	debug platform packet-trace packet <i>pkt-num</i> [fia-trace summary-only] [circular] [data-size <i>data-size</i>] Example:	指定した数のパケットのサマリーデータを収集します。デフォルトでは機能パスデータをキャプチャし、必要に応じて FIA トレースを実行します。

	Command or Action	Purpose
	Router# debug platform packet-trace packets 2048 summary-only	pkt-num : 所定の時間に維持されるパケットの最大数を指定します。 fia-trace : サマリーデータ、機能固有のデータなど、詳細なレベルのデータキャプチャを実行します。また、パケット処理中にアクセスされた各機能エントリも表示します。 summary-only : 詳細情報を最小限にしたサマリーデータのキャプチャを有効にします。 circular : 最近トレースされたパケットのデータを保存します。 data-size : 各パケットの機能データと FIA トレースデータを保存するデータバッファのサイズをバイト単位で指定します。パケットで非常に重いパケット処理が実行された場合、ユーザーは必要に応じてデータバッファのサイズを増やすことができます。デフォルト値は 2048 です。
ステップ 3	debug platform packet-trace {punt inject copy drop packet statistics} Example: Router# debug platform packet-trace punt	データからコントロールプレーンへパントされたパケットのトレースを有効にします。
ステップ 4	debug platform condition [ipv4 ipv6] [interface interface][access-list access-list -name ipv4-address / subnet-mask ipv6-address / subnet-mask] [ingress egress both] Example: Router# debug platform condition interface g0/0/0 ingress	パケットをトレースするための一致基準を指定します。プロトコル、IP アドレスおよびサブネットマスク、アクセス制御リスト (ACL)、インターフェイス、方向によるフィルタリング機能を提供します。
ステップ 5	debug platform condition start Example: Router# debug platform condition start	指定した位置基準を有効にしてパケットトレースを開始します。
ステップ 6	debug platform condition stop Example: Router# debug platform condition start	条件を非アクティブにして、パケットのトレースを停止します。

	Command or Action	Purpose
ステップ 7	show platform packet-trace {configuration statistics summary packet {all pkt-num}} Example: Router# show platform packet-trace 14	指定されたオプションに従って、パケットトレースデータを表示します。 show コマンドのオプションの詳細については、{start cross reference} 表 21-1 {end cross reference} を参照してください。
ステップ 8	clear platform condition all Example: Router(config)# clear platform condition all	debug platform condition コマンドおよび debug platform packet-trace コマンドによって提供された設定を削除します。
ステップ 9	exit Example: Router# exit	特権 EXEC モードを終了します。

UDF オフセットを使用したパケットトレースの設定

オフセットを使用してパケットトレース UDF を設定するには、次の手順を実行します。

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **udf udf name header {inner | outer} {13|14} offset offset-in-bytes length length-in-bytes**
4. **udf udf name {header | packet-start} offset-base offset length**
5. **ip access-list extended {acl-name |acl-num}**
6. **ip access-list extended { deny | permit } udf udf-name value mask**
7. **debug platform condition [ipv4 | ipv6] [interface interface] [access-list access-list -name | ipv4-address / subnet-mask | ipv6-address / subnet-mask] [ingress | egress |both]**
8. **debug platform condition start**
9. **debug platform packet-trace packet pkt-num [fia-trace | summary-only] [circular] [data-size data-size]**
10. **debug platform packet-trace {punt | inject|copy | drop |packet | statistics}**
11. **debug platform condition stop**
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
ステップ 1	enable Example: Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	Command or Action	Purpose
ステップ 2	configure terminal Example: <pre>Device# configure terminal</pre>	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 3	udf udf name header {inner outer} {13 14} offset offset-in-bytes length length-in-bytes Example: <pre>Router(config)# udf TEST_UDF_NAME_1 header inner 13 64 1 Router(config)# udf TEST_UDF_NAME_2 header inner 14 77 2 Router(config)# udf TEST_UDF_NAME_3 header outer 13 65 1 Router(config)# udf TEST_UDF_NAME_4 header outer 14 67 1</pre>	個々の UDF 定義を設定します。UDF の名前、オフセット元のネットワーキングヘッダー、抽出するデータの長さを指定できます。 inner キーワードまたは outer キーワードは、カプセル化されていないレイヤ3またはレイヤ4のヘッダーからのオフセットの開始を指定するか、またはカプセル化された packet がある場合は内部 L3/L4 からのオフセットの開始を指定します。 length キーワードはオフセットからの長さをバイト単位で指定します。有効な範囲は 1 ～ 2 です。
ステップ 4	udf udf name {header packet-start} offset-base offset length Example: <pre>Router(config)# udf TEST_UDF_NAME_5 packet-start 120 1</pre>	• header : オフセットの基本設定を指定します。 • packet-start : packet-start からのオフセットベースを指定します。packet-start は、パケットトレースがインバウンド packet 用かアウトバウンド packet 用かによって異なります。パケットトレースがインバウンド packet 用である場合、packet 開始はレイヤ2になります。アウトバウンドの場合は、packet-start はレイヤ3になります。 • offset : オフセット ベースからオフセットさせるバイト数を指定します。オフセット ベース（レイヤ3/レイヤ4 ヘッダー）からの先頭バイトに一致させるには、オフセットを0に設定します。 • length : オフセットからのバイト数を指定します。1 バイトまたは2 バイトだけがサポートされます。追加のバイト数に一致させるには、複数の UDF の定義が必要です。
ステップ 5	ip access-list extended {acl-name [acl-num]} Example: <pre>Router(config)# ip access-list extended acl2</pre>	拡張 ACL コンフィギュレーション モードを有効にします。CLI は拡張 ACL コンフィギュレーション モードを開始します。このモードでは、後続のすべてのコマンドが現在の拡張アクセスリストに適用されます。拡張 ACL は、IP パケットの送信元アドレ

	Command or Action	Purpose
		スおよび宛先アドレスを ACL に設定されているアドレスと比較して、トラフィックを制御します。
ステップ 6	ip access-list extended { deny permit } udf udf-name value mask Example: <pre>Router(config-acl)# permit ip any any udf TEST_UDF_NAME_5 0xD3 0xFF</pre>	現在のアクセス制御エントリ (ACE) と併せて、UDF で一致するように ACL を設定します。ACL で定義されているバイトは 0xD3 です。マスクは、許可および拒否するトラフィックを指定するように、IP ACL で IP アドレスとともに使用します。
ステップ 7	debug platform condition [ipv4 ipv6] [interface interface] [access-list access-list -name ipv4-address / subnet-mask ipv6-address / subnet-mask] [ingress egress both] Example: <pre>Router# debug platform condition interface gi0/0/0 ipv4 access-list acl2 both</pre>	パケットをトレースするための一致基準を指定します。プロトコル、IP アドレスおよびサブネットマスク、アクセス制御リスト (ACL)、インターフェイス、方向によるフィルタリング機能を提供します。
ステップ 8	debug platform condition start Example: <pre>Router# debug platform condition start</pre>	指定した位置基準を有効にしてパケットトレースを開始します。
ステップ 9	debug platform packet-trace packet pkt-num [fia-trace summary-only] [circular] [data-size data-size] Example: <pre>Router# debug platform packet-trace packet 1024 fia-trace data-size 2048</pre>	指定した数のパケットのサマリーデータを収集します。デフォルトでは機能パスデータをキャプチャし、必要に応じて FIA トレースを実行します。 pkt-num : 所定の時間に維持されるパケットの最大数を指定します。 fia-trace : サマリーデータ、機能固有のデータなど、詳細なレベルのデータキャプチャを実行します。また、パケット処理中にアクセスされた各機能エントリも表示します。 summary-only : 詳細情報を最小限にしたサマリーデータのキャプチャを有効にします。 circular : 最近トレースされたパケットのデータを保存します。 data-size : 各パケットの機能データと FIA トレースデータを保存するデータバッファのサイズをバイト単位で指定します。パケットで非常に重いパケット処理が実行された場合、ユーザーは必要に応じてデータバッファのサイズを増やすことができます。デフォルト値は 2048 です。

	Command or Action	Purpose
ステップ 10	debug platform packet-trace {punt inject copy drop packet statistics} Example: Router# debug platform packet-trace punt	データからコントロールプレーンへパントされたパケットのトレースを有効にします。
ステップ 11	debug platform condition stop Example: Router# debug platform condition start	条件を非アクティブにして、パケットのトレースを停止します。
ステップ 12	exit Example: Router# exit	特権 EXEC モードを終了します。

パケットトレース情報の表示

パケットトレース情報を表示するには、次の **show** コマンドを使用します。

Table 2: show コマンド

コマンド	説明
show platform packet-trace configuration	デフォルトを含むパケットトレース設定が表示されます。
show platform packet-trace statistics	トレースされたすべてのパケットのアカウントिंगデータが表示されます。
show platform packet-trace summary	指定した数のパケットのサマリーデータが表示されます。
show platform packet-trace {all pkt-num} [decode]	すべてのパケットまたは指定したパケットのパスデータが表示されます。 decode オプションを使用すると、バイナリパケットのより人間が判読しやすい形式へのデコードが試みられます。

パケットトレースデータの削除

パケットトレースデータをクリアするには、次のコマンドを使用します。

Table 3: clear コマンド

コマンド	説明
clear platform packet-trace statistics	収集されたパケットトレースデータと統計をクリアします。
clear platform packet-trace configuration	パケットトレース設定と統計をクリアします。

パケットトレースの設定例

ここでは、次の設定例について説明します。

例：パケットトレースの設定

この例では、パケットトレースを設定し、結果を表示する方法について説明します。この例では、ギガビットイーサネットインターフェイス 0/0/1 への着信パケットがトレースされ、最初の 128 パケットの FIA トレースデータがキャプチャされます。また、入力パケットがコピーされます。**show platform packet-trace packet 0** コマンドにより、パケット 0 について、概要データと、パケット処理中にアクセスされた各機能エントリが表示されます。

```
Router>
enable
Router# debug platform packet-trace packet 128 fia-trace
Router# debug platform packet-trace punt
Router# debug platform condition interface g0/0/1 ingress
Router# debug platform condition start
Router#! ping to UUT
Router# debug platform condition stop
Router# 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      : 198.51.100.2
  Destination : 198.51.100.2
  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

```

```

Router# clear platform condition all
Router# exit

```

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。この例では、インターセプトされた binos アプリケーション宛ての LFTS ベースのパケットが表示されてます。

```

Router# show platform packet-trace packet 10
Packet: 10      CBUG ID: 52
Summary
Input  : GigabitEthernet0/0/0
Output : internal0/0/rp:1
State  : PUNT 55 (For-us control)
Timestamp
Start  : 597718358383 ns (06/06/2016 09:00:13.643341 UTC)
Stop   : 597718409650 ns (06/06/2016 09:00:13.643392 UTC)
Path Trace
Feature: IPV4
Input  : GigabitEthernet0/0/0
Output : <unknown>
Source : 10.64.68.2
Destination : 224.0.0.102
Protocol : 17 (UDP)
SrcPort : 1985
DstPort : 1985
Feature: FIA_TRACE
Input  : GigabitEthernet0/0/0
Output : <unknown>
Entry  : 0x8a0177bc - DEBUG_COND_INPUT_PKT
Lapsed time : 426 ns
Feature: FIA_TRACE
Input  : GigabitEthernet0/0/0
Output : <unknown>
Entry  : 0x8a017788 - IPV4_INPUT_DST_LOOKUP_CONSUME
Lapsed time : 386 ns
Feature: FIA_TRACE
Input  : GigabitEthernet0/0/0

```

```

Output : <unknown>
Entry   : 0x8a01778c - IPV4_INPUT_FOR_US_MARTIAN
Lapsed time : 13653 ns
Feature: FIA_TRACE
Input   : GigabitEthernet0/0/0
Output  : internal0/0/rp:1
Entry   : 0x8a017730 - IPV4_INPUT_LOOKUP_PROCESS_EXT
Lapsed time : 2360 ns
Feature: FIA_TRACE
Input   : GigabitEthernet0/0/0
Output  : internal0/0/rp:1
Entry   : 0x8a017be0 - IPV4_INPUT_IPOPTIONS_PROCESS_EXT
Lapsed time : 66 ns
Feature: FIA_TRACE
Input   : GigabitEthernet0/0/0
Output  : internal0/0/rp:1
Entry   : 0x8a017bfc - IPV4_INPUT_GOTO_OUTPUT_FEATURE_EXT
Lapsed time : 680 ns
Feature: FIA_TRACE
Input   : GigabitEthernet0/0/0
Output  : internal0/0/rp:1
Entry   : 0x8a017d60 - IPV4_INTERNAL_ARL_SANITY_EXT
Lapsed time : 320 ns
Feature: FIA_TRACE
Input   : GigabitEthernet0/0/0
Output  : internal0/0/rp:1
Entry   : 0x8a017a40 - IPV4_VFR_REFRAG_EXT
Lapsed time : 106 ns
Feature: FIA_TRACE
Input   : GigabitEthernet0/0/0
Output  : internal0/0/rp:1
Entry   : 0x8a017d2c - IPV4_OUTPUT_DROP_POLICY_EXT
Lapsed time : 1173 ns
Feature: FIA_TRACE
Input   : GigabitEthernet0/0/0
Output  : internal0/0/rp:1
Entry   : 0x8a017940 - INTERNAL_TRANSMIT_PKT_EXT
Lapsed time : 20173 ns
LFTS Path Flow: Packet: 10      CBUG ID: 52
Feature: LFTS
Pkt Direction: IN
Punt Cause    : 55
subCause      : 0

```

例：パケットトレースの使用

次に、パケットトレースを使用して Cisco ASR 1006 ルータの NAT 設定でパケットドロップのトラブルシューティングを行うシナリオの例を示します。この例には、パケットトレース機能によって提供される詳細レベルを効果的に利用して問題に関する情報を収集し、問題を切り分けて、解決策を見つける方法が示されています。

このシナリオでは、問題があることはわかりませんが、どこからトラブルシューティングを開始すればよいかはわかりません。したがって、多数の着信パケットのパケットトレースのサマリーにアクセスすることを確認する必要があります。

```

Router# debug platform condition ingress
Router# debug platform packet-trace packet 2048 summary-only
Router# debug platform condition start
Router# debug platform condition stop

```

```
Router# show platform packet-trace summary
Pkt   Input      Output      State Reason
0     Gi0/0/0      Gi0/0/0      DROP  402 (NoStatsUpdate)
1     internal0/0/rp:0 internal0/0/rp:0 PUNT  21 (RP<->QFP keepalive)
2     internal0/0/recycle:0 Gi0/0/0      FWD
```

この出力には、ギガビットイーサネットインターフェイス 0/0/0 の NAT 設定が原因でパケットがドロップされていることが示されています。これによって、問題は特定のインターフェイスで発生していることがわかります。この情報を使用して、トレースするパケットを制限し、データキャプチャのパケット数を減らし、検査レベルを上げることができます。

```
Router# debug platform packet-trace packet 256
Router# debug platform packet-trace punt
Router# debug platform condition interface Gi0/0/0
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Router# show platform packet-trace 15
Packet: 15          CBUG ID: 238
Summary
  Input      : GigabitEthernet0/0/0
  Output     : internal0/0/rp:1
  State      : PUNT 55 (For-us control)
  Timestamp
    Start    : 1166288346725 ns (06/06/2016 09:09:42.202734 UTC)
    Stop     : 1166288383210 ns (06/06/2016 09:09:42.202770 UTC)
Path Trace
  Feature: IPV4
    Input      : GigabitEthernet0/0/0
    Output     : <unknown>
    Source     : 10.64.68.3
    Destination : 224.0.0.102
    Protocol   : 17 (UDP)
    SrcPort    : 1985
    DstPort    : 1985
IOSd Path Flow: Packet: 15    CBUG ID: 238
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From CPP
  Feature: IP
    Pkt Direction: IN
    Source      : 10.64.68.122
    Destination : 10.64.68.255
  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.64.68.122
    Destination : 10.64.68.255
    Interface   : GigabitEthernet0/0/0
  Feature: UDP
    Pkt Direction: IN
    src          : 10.64.68.122(1053)
    dst          : 10.64.68.255(1947)
    length       : 48
```

```
Router#show platform packet-trace packet 10
Packet: 10          CBUG ID: 10
Summary
  Input      : GigabitEthernet0/0/0
  Output     : internal0/0/rp:0
  State      : PUNT 55 (For-us control)
  Timestamp
```

```
Start      : 274777907351 ns (01/10/2020 10:56:47.918494 UTC)
Stop       : 274777922664 ns (01/10/2020 10:56:47.918509 UTC)
Path Trace
Feature: IPV4(Input)
Input      : GigabitEthernet0/0/0
Output     : <unknown>
Source     : 10.78.106.2
Destination : 224.0.0.102
Protocol   : 17 (UDP)
SrcPort    : 1985
DstPort    : 1985

IOSd Path Flow: Packet: 10      CBUG ID: 10
Feature: INFRA
Pkt Direction: IN
Packet Rcvd From DATAPLANE
Feature: IP
Pkt Direction: IN
Packet Enqueued in IP layer
Source      : 10.78.106.2
Destination : 224.0.0.102
Interface   : GigabitEthernet0/0/0

Feature: UDP
Pkt Direction: IN DROP
Pkt : DROPPED
UDP: Discarding silently
src        : 881 10.78.106.2(1985)
dst        : 224.0.0.102(1985)
length     : 60

Router#show platform packet-trace packet 12
Packet: 12      CBUG ID: 767
Summary
Input      : GigabitEthernet3
Output     : internal0/0/rp:0
State      : PUNT 11 (For-us data)
Timestamp
Start      : 16120990774814 ns (01/20/2020 12:38:02.816435 UTC)
Stop       : 16120990801840 ns (01/20/2020 12:38:02.816462 UTC)
Path Trace
Feature: IPV4(Input)
Input      : GigabitEthernet3
Output     : <unknown>
Source     : 12.1.1.1
Destination : 12.1.1.2
Protocol   : 6 (TCP)
SrcPort    : 46593
DstPort    : 23

IOSd Path Flow: Packet: 12      CBUG ID: 767
Feature: INFRA
Pkt Direction: IN
Packet Rcvd From DATAPLANE

Feature: IP
Pkt Direction: IN
Packet Enqueued in IP layer
Source      : 12.1.1.1
Destination : 12.1.1.2
Interface   : GigabitEthernet3

Feature: IP
Pkt Direction: IN
FORWARDEDTo transport layer
```

例：パケットトレースの使用

```

Source      : 12.1.1.1
Destination : 12.1.1.2
Interface   : GigabitEthernet3

```

```
Feature: TCP
```

```
Pkt Direction: IN
```

```
tcp0: I NoTCB 12.1.1.1:46593 12.1.1.2:23 seq 1925377975 OPTS 4 SYN WIN 4128
```

```
Router# show platform packet-trace summary
```

Pkt	Input	Output	State	Reason
0	INJ.2	Gi1	FWD	
1	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
2	INJ.2	Gi1	FWD	
3	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
4	INJ.2	Gi1	FWD	
5	INJ.2	Gi1	FWD	
6	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
7	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
8	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
9	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
10	INJ.2	Gi1	FWD	
11	INJ.2	Gi1	FWD	
12	INJ.2	Gi1	FWD	
13	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
14	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
15	Gi1	internal0/0/rp:0	PUNT	11 (For-us data)
16	INJ.2	Gi1	FWD	

次に、パケットトレースデータの統計を表示する例を示します。

```
Router#show platform packet-trace statistics
```

```
Packets Summary
```

```
Matched 3
```

```
Traced 3
```

```
Packets Received
```

```
Ingress 0
```

```
Inject 0
```

```
Packets Processed
```

```
Forward 0
```

```
Punt 3
```

Count	Code	Cause
3	56	RP injected for-us control

```
Drop 0
```

```
Consume 0
```

```
PKT_DIR_IN
```

	Dropped	Consumed	Forwarded
INFRA	0	0	0
TCP	0	0	0
UDP	0	0	0
IP	0	0	0
IPV6	0	0	0
ARP	0	0	0

```
PKT_DIR_OUT
```

	Dropped	Consumed	Forwarded
INFRA	0	0	0
TCP	0	0	0
UDP	0	0	0
IP	0	0	0
IPV6	0	0	0
ARP	0	0	0

次に、コントロールプレーンからフォワーディングプロセッサに挿入およびパントされるパケットを表示する例を示します。

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

```
Router#show platform packet-tracer packet 0
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 : 198.51.100.38
    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  : 198.51.100.38
  Interface    : GigabitEthernet1
```

```
Feature: IP
Pkt Direction: IN
FORWARDED To transport layer
  Source       : 10.118.74.53
  Destination  : 198.51.100.38
  Interface    : GigabitEthernet1
```

```
Feature: UDP
Pkt Direction: IN
DROPPED
UDP: Checksum error: dropping
Source       : 10.118.74.53(2640)
Destination  : 198.51.100.38(500)
```

```
Router#show platform packet-tracer packet 2
Packet: 2          CBUG ID: 2
```

```
IOSd Path Flow:
  Feature: TCP
  Pkt Direction: OUTtcp0: 0 SYNRCVD 198.51.100.38:22 198.51.100.55:52774 seq 3052140910
  OPTS 4 ACK 2346709419 SYN WIN 4128
```

```
Feature: TCP
Pkt Direction: OUT
```

```

FORWARDED
TCP: Connection is in SYNRCVD state
ACK      : 2346709419
SEQ      : 3052140910
Source   : 198.51.100.38(22)
Destination : 198.51.100.55(52774)

Feature: IP
Pkt Direction: OUTRoute out the generated packet.srcaddr: 198.51.100.38, dstaddr:
198.51.100.55

Feature: IP
Pkt Direction: OUTInject and forward successful srcaddr: 198.51.100.38, dstaddr:
198.51.100.55

Feature: TCP
Pkt Direction: OUTtcp0: O SYNRCVD 198.51.100.38:22 198.51.100.55:52774 seq 3052140910
OPTS 4 ACK 2346709419 SYN WIN 4128
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  : 55.124.18.172
Local Addr : 38.124.18.172

Router#

```

その他の参考資料

標準

標準	タイトル
なし	—

MIB

MIB	MIB のリンク
なし	選択したプラットフォーム、Cisco IOS リリース、およびフィチャセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 {start hypertext}http://www.cisco.com/go/mibs{end hypertext}

RFC

RFC	タイトル
なし	—

シスコのテクニカル サポート

説明	リンク
右の URL にアクセスして、シスコのテクニカルサポートを最大限に活用してください。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。	{start hypertext}http://www.cisco.com/cisco/web/support/index.html{end hypertext}

パケットトレースの機能情報

{start cross reference}表 21-4{end cross reference} に、このモジュールで説明した機能をリストし、特定の設定情報へのリンクを示します。

プラットフォームのサポートおよびソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator を使用すると、ソフトウェアイメージがサポートする特定のソフトウェア リリース、フィチャセット、またはプラットフォームを確認できます。Cisco Feature Navigator にアクセスするには、{start hypertext}http://www.cisco.com/go/cfn{end hypertext} に進みます。Cisco.com のアカウントは必要ありません。



Note

{start cross reference}表 21-4{end cross reference} には、特定のソフトウェア リリース トレインで各機能をサポートするソフトウェアリリースだけが示されています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

Table 4: パケットトレースの機能情報

機能名	リリース	機能情報
パケットトレース	Cisco IOS XE 3.10S	パケットトレース機能は、Cisco IOS XE ソフトウェアによるデータパケットの処理方法に関する情報を提供します。 Cisco IOS XE リリース 3.10S では、この機能は Cisco ASR 1000 シリーズ アグリゲーション サービス ルータに導入されました。 次のコマンドが導入または変更されました。 • debug platform packet-trace packet <i>pkt-num</i> [fia-trace summary-only] [data-size <i>data-size</i>] [circular] • debug platform packet-trace copy packet {input output both} [size <i>num-bytes</i>] [L2 L3 L4] • show platform packet-trace {configuration statistics summary packet {all <i>pkt-num</i>}}
	Cisco IOS XE 3.11S	Cisco IOS XE リリース 3.11S で、この機能が拡張され、次の機能が含まれるようになりました。 • 一致した統計と追跡された統計。 • トレース開始タイムスタンプに加えて、トレース停止タイムスタンプ。 次のコマンドが導入または変更されました。 • debug platform packet-trace drop [code <i>drop-num</i>] • show platform packet-trace packet {all <i>pkt-num</i>} [decode]
	Cisco IOS XE Denali 16.3.1	Cisco IOS XE Denali 16.3.1 で、この機能が拡張され、IOSd とともにレイヤ 3 パケットトレースが含まれるようになりました。 次のコマンドが導入または変更されました。 debug platform packet-trace punt.
	Cisco IOS XE Amsterdam 17.3.1	show platform packet-trace コマンドの出力に、IOSd から発信されたパケットか、IOSd または他の BinOS プロセス宛のパケットに関する追加のトレース情報が含まれるようになりました。

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。