

IOS-XE Datapath パケット トレース機能

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[参照トポロジ](#)

[使用中のパケットトレース](#)

[クイックスタートガイド](#)

[イネーブル プラットフォーム 条件付きデバッグ](#)

[イネーブル パケットトレース](#)

[パケットトレースとの出力状態制限](#)

[パケットトレース結果を表示する](#)

[FIA トレース](#)

[パケットトレース結果を表示する](#)

[インターフェイスと関連付けられる FIA をチェックして下さい](#)

[トレースされたパケットをダンプして下さい](#)

[トレースを廃棄して下さい](#)

[例ドロップする トレース シナリオ](#)

[トレースをインジェクトし、パントして下さい](#)

[パケットトレース例](#)

[パケットトレース例- NAT](#)

[パケットトレース例- VPN](#)

[パフォーマンスへの影響](#)

[関連情報](#)

概要

この資料にパケットトレース 機能によって Cisco IOS[®]-XE ソフトウェアのための datapath パケットトレースを行う方法を記述されています。

トラブルシューティング中にミスコンフィギュレーションのような問題を、キャパシティ 過負荷 識別するために、また更に通常のソフトウェアバグ、システム内のパケットがどうなるか理解することは必要です。Cisco IOS XE パケットトレース 機能はこの必要に対応します。説明のためにユーザが定義する状態のクラスに基づいてパケットごとの処理詳細をキャプチャ するために使用されるそれはフィールド セーフ方式を提供し。

前提条件

要件

Cisco は Cisco IOS XE バージョン 3.10 および それ 以降で利用可能である、また Cisco IOS XE ソフトウェアを実行する Integrated Services Router (ISR) モジュール 集約 サービス ルータ

(ASR1K)、Cisco 1000V シリーズ Cloud サービス ルータ (CSR1000v)、および Cisco 4451-X シリーズのようなすべてのプラットフォームの知識が Cisco 1000 シリーズあるパケットトレース 機能のことを推奨します (ISR4451-X)。

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くものです。

- Cisco IOS XE ソフトウェア バージョン 3.10S (15.3(3)S) およびそれ以降
- ASR1K

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たものです。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。稼働中のネットワークで作業を行う場合、使用されているコマンドの影響について十分に理解したうえで作業してください。

参照トポロジ

このダイアグラムはこの資料に説明がある例のために使用するトポロジーを説明します:



使用中のパケットトレース

パケットトレース 機能の使用を説明するために、このセクション全体使用する例はリモートホスト 172.16.20.2 (Gig0/0/1 インターフェイスの ASR1K のための入方向) にローカル ワークステーション 172.16.10.2 からのインターネット制御メッセージ プロトコル (ICMP) トラフィックのトレースを (ASR1K の後ろで) 記述したものです。

これら二つのステップの ASR1K のパケットをトレースできます:

1. ASR1K でトレースしたいと思うトラフィックがパケットを選択することをプラットフォーム 条件付きデバッグが可能にしてください。
2. プラットフォーム パケットトレースを有効に してください (パストレースはまたは呼び出しアレイ (FIA) トレースを特色にします)。

クイックスタートガイド

既にこの資料のコンテンツについて詳しく知っていればここにあり、CLI のクイック ルックのためのセクションをほしいと思いますクイックスタートガイドは。これらはツールの使用を説明する少数の例だけ参照し構文を詳しく論議する、約束します要件に適切な使用を設定です以下のセクションを。

1. 設定 プラットフォーム状態:

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)プラットフォーム状態が設定された後、この CLI コマンドでプラットフォーム条件を開始して下さい:

debug platform condition start

2. 設定 パケットトレース:

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.

3. 次に、プラットフォーム パケットトレースを有効に して下さい:

debug platform packet-trace enable

バッファをクリアし、パケットトレースをリセットするコマンドはここにあります:

no debug platform packet-trace enable --> disable the packet trace, this needs to be done before you clear the buffer

clear platform packet-trace statistics --> clear the packet trace buffer

debug platform packet-trace enable --> reenable the packet trace

プラットフォームをクリアするコマンドは調節し、パケットトレース 設定は次のとおりです:

clear platform condition all --> clears both platform conditions and the packet trace configuration

show コマンド

プラットフォーム状態およびパケットトレース 設定に必要なものが確認するために前のコマンドを適用した後ある確認して下さい。

show platform conditions --> shows the platform conditions configured

show platform packet-trace configuration --> shows the packet-trace configurations

show debugging --> this will show both platform conditions and platform packet-trace configured
トレースされた/キャプチャされるパケットをチェックするコマンドはここにあります:

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` -> Tracing the 12th packet, with complete path trace or FIA trace details.

イネーブル プラットフォーム 条件付きデバッグ

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にトレースされるべきパケットを判別するために頼ります。条件付きデバッグ インフラストラクチャは基づいてフィルタトラフィックに能力を提供します:

- プロトコル
- IP アドレスおよびマスク
- Access Control List (ACL; アクセス コントロール リスト)
- Interface
- 場周 飛行 方向 (入力か出力)

これらの条件はフィルターがパケットにいつ、どこで加えられるか定義します。

トラフィックに関してはこの例で使用する、172.16.10.2 からの 172.16.20.2 に ICMP パケットのための入方向のプラットフォーム 条件付きデバッグを有効にしてください。すなわち、トレースしたいと思うトラフィックを選択してください。このトラフィックを選択するために使用できるさまざまなオプションがあります。

```
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
```

この例ではここに示されているとして条件を、定義するために、access-list は使用されます:

```
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ために、このコマンドを入力してください:

```
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ディセーブルにするために停止するか、または、**デバッグ プラットフォーム状態停止**コマンドを入力してください。

設定される条件付きデバッグ フィルターを表示するために、このコマンドを入力します:

```
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、この設定はこれまでに適用されました:

```
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
```

イネーブル パケットトレース

注: このセクションはパケットおよびコピー オプションを詳しく記述し、その他のオプションは資料の記述されていた以降です。

パケットトレースはトンネルまたは仮想アクセスインターフェイスのような物理的のおよび論理インターフェイスで、サポートされます。

パケットトレース CLI 構文はここにあります:

```
ASR1000#debug platform packet-trace ?
copy Copy packet data
drop Trace drops only
enable Enable packet trace
inject Trace injects only
packet Packet count
punt Trace punts only
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ための説明はここにあります:

- **pkt 数字**-パケット数はパケットの最大数を規定 します一度に維持される。
- **summary-only** -これは要約データだけキャプチャ されること規定 します。 デフォルトは要約データおよび機能 パス データを両方キャプチャ することです。
- **fia** トレース-これはパスデータ 情報に加えてオプションで FIA トレースを行います。
- **データサイズ**-これは 2,048 から 16,384 バイトからパスデータ バッファのサイズを規定 することを可能にします。 デフォルトは 2,048 バイトです。

```
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ための説明はここにあります:

- **イン/アウト**はこれコピーされるべきパケットフローの方向を-入力や出力 規定 します。
- **L2/L3/L4** -これはパケットのコピーが開始する位置を規定 することを可能にします。 レイヤ 2 (L2) はデフォルト の ロケーションです。
- **サイズ**-これはコピーされるオクテットの最大数を規定 することを可能にします。 デフォルトは 64 オクテットです。

この例に関しては、これらは条件付きデバッグ インフラストラクチャと選択されるトラフィックのためのパケットトレースを有効に するために使用されるコマンドです:

```
ASR1000#debug platform packet-trace packet 16
```

```
ASR1000#debug platform packet-trace enable
```

パケットトレース 設定を検討するために、このコマンドを入力して下さい:

```
ASR1000#show platform packet-trace configuration
debug platform packet-trace enable
debug platform packet-trace packet 16 data-size 2048
```

またプラットフォーム 条件付きデバッグおよびパケットトレース コンフィギュレーションを両方表示するために **show debugging** コマンドを入力できます:

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

注: プラットフォーム デバッグ状態のすべてをおよびパケットトレース コンフィギュレーションおよびデータ クリアするためにクリア プラットフォーム状態をすべてのコマンド入力して下さい。

パケットトレースを有効にするために要約すると、このコンフィギュレーションデータはこれまでに使用されました:

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

パケットトレースとの出力状態制限

条件はパケットに加えられる時条件付きフィルターを定義し。たとえば、**デバッグプラットフォーム状態インターフェイス g0/0/0 出力**はインターフェイス g0/0/0 の出力 FIA に達するときパケットが一致として、そのポイントが抜けているまで入力から起こるそうあらゆるパケット処理識別されることを意味します。

注: Cisco は強く可能性のあるに最も完全のおよび意味のあるデータを入手するためにパケットトレースのために入力状態を使用することを推奨します。出力状態は使用することができますが制限に気づいています。

パケットトレース結果を表示する

注: このセクションはパストレースが有効になると仮定します。

インスペクションの 3 つの特定のレベルはパケットトレースによって提供されます:

- 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 に送信されるときパケットトレース結果を表示するために、これらのコマンドは使用することができます:

```
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: 第 3 コマンドは各パケットのためのパケットトレースを表示する方法を説明する例を提供したものです。この例では、トレースされる最初のパケットは示されています。

これらの出力から、5 つのパケットがトレースされること、そしてインプットインターフェイス、アウトプットインターフェイス、状態およびパスをトレースを表示できることがわかります。

State マークし直して下さい

FWD パケットは配信のために、スケジュールされましたり/出力 インターフェイスによってネクストホップに転送されるためにキューに入ります。

PUNT パケットは Forwarding Processor (FP) から Route Processor (RP) (コントロールプレーン) へのパントされます。

[DROP] パケットは FP で廃棄されます。FIA トレースを実行しますか、グローバルなドロップカウンタを使用するか、またはドロップする原因でより多くの詳細を見つけるために datapath デバッグを有効にしてください。

反対論 パケットは ICMP Ping 要求の間にまたは暗号パケットのようなパケット プロセスの間に、消費

ます。

入力は出力によってがコントロールプレーンからインジェクトされるように見られるパケットおよび外部インターフェイスによって、それぞれ入力するパケットに対応するパケットトレース統計情報のカウンターをインジェクトし。

FIA トレース

FIA はパケットが入力が出力を転送されるとき Quantum フロープロセッサ (QFP) のパケットプロセッサエンジン (PPE) によって次々に実行される機能のリストを保持します。機能はマシンで適用されるコンフィギュレーションデータに基づいています。従って、FIA トレースはパケットが処理されると同時にシステムによってパケットのフローの理解を助けます。

FIA のパケットトレースを有効にするためにこのコンフィギュレーションデータを適用して下さい:

```
ASR1000#debug platform packet-trace packet 16 fia-trace
```

パケットトレース結果を表示する

注: このセクションは FIA トレースが有効になると仮定します。また現在のパケットトレースコマンドを追加するか、または修正するとき、バッファリングされたパケットトレース詳細はクリアされます、従ってそれをトレースできるようにトラフィックを再度送信して下さい。

FIA トレースを有効にするために使用されるコマンドを入力した後送信 172.16.10.2 からの前のセクションに記述されているように 172.16.20.2 への 5 つの ICMP パケット。

```
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#

インターフェイスと関連付けられる FIA をチェックして下さい

プラットフォーム 条件付きデバッグを有効にするとき、それは機能として FIA に追加されます。位置によっては前encap および後encap パケットをトレースするときリストに追加されること、プラットフォーム状態を、のような調節する必要があるかもしれません。

この出力は入方向で有効になる プラットフォーム 条件付きデバッグのための FIA で機能の発注を示したものです:

```
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 および DEBUG_COND_INPUT_PKT はルータで設定され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をダンプして下さい

このセクションが記述すると同時にトレースされると同時にパケットをコピーし、ダンプできます。この例に最大 2,048 バイトをの入方向 (172.16.20.2) のパケットへの 172.16.10.2 コピーする方法を示されています。

必要の追加コマンドはここにあります:

```
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 から 2,048 バイトの範囲にあります。

コピーされたパケットをダンプするためにこのコマンドを入力して下さい:

```
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#

トレースを廃棄して下さい

ドロップするトレースは Cisco IOS XE ソフトウェア バージョン 3.11 および それ 以降で利用できます。それは破棄されたパケットのためのだけパケットトレースを有効にします。機能のいくつかの強調表示はここにあります:

- それはオプションで特定のドロップするコードのためのパケットの保持を規定 することを可能にします。
- それはグローバル なまたはインターフェイス状態なしでドロップ イベントをキャプチャ するために使用することができます。
- ドロップ イベント キャプチャはドロップするだけ自体トレースされることを、パケットのないライフ意味します。 ただし、それはまだ要約データを、タプルデータ キャプチャ することを可能にし、パケットは条件を精製するか、または次のデバッグへの糸口の提供を助けるために歩みます。

ドロップ タイプ 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はここにあります:

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

ドロップするコードは提示プラットフォーム ハードウェア qfp で報告されたようにドロップする ID と同じ、アクティブな統計情報ドロップする detail コマンド出力です:

```
ASR1000#show platform hardware qfp active statistics drop detail
```

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

例ドロップする トレース シナリオ

172.16.10.2 から 172.16.20.2 にトラフィックを廃棄するために ASR1K のギグ 0/0/0 インターフェイスのこの ACL を適用して下さい:

```
ASR1000#show platform hardware qfp active statistics drop detail
```

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

ローカル ホストからリモートホストにトラフィックを廃棄する ACL を使うとこのドロップする トレース 設定を適用して下さい:

```
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
debug platform packet-trace enable
```

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て下さい。 ドロップする トレースは示されているように 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:

```
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#
```

トレースをインジェクトし、パントして下さい

インジェクトおよびパント パケットトレース 機能は FP で受け取られる Cisco IOS XE ソフトウェア バージョン 3.12 および それ 以降にパント (コントロール プレーンにパントされるパケットを) トレースし、 (コントロール プレーンからの FP にインジェクトされるパケット) パケットをインジェクトするために追加されました。

注: パント トレースはグローバルのなしではたらくことができずまたはドロップする トレースと同様に条件を、インターフェイスさせます。ただし、条件はインジェクト トレースがはたらくことができるように定義する必要があります。

ここに ASR1K から隣接ルータに ping するときパントの例はあり、パケットトレースをインジェクトします:

```
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#debug platform packet-trace enable
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#
```

この場合パントを確認し、トレース結果をインジェクトできます:

```
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)
```

パケットトレース例

このセクションはパケットトレース機能がトラブルシューティングを行うのに役立ついくつかの例を提供します。

パケットトレース例- NAT

この例によって、インターフェイス ソースネットワーク アドレス 変換 (NAT) はローカルサブネット (172.16.10.0/24) のための ASR1K (Gig0/0/0) の WAN インターフェイスで設定されません。

172.16.10.2 から 172.16.20.2 にトラフィックをトレースするために使用するパケットトレース設定およびプラットフォーム状態はここにあります Gig0/0/0 インターフェイスで変換されるように (NAT) なる:

```
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)

5つのICMPパケットが172.16.10.2からインターフェイス出典NAT設定の172.16.20.2に送信されるとき、これらはパケットトレース結果です:

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#
```

パケットトレース例- VPN

この例によってトラフィックを保護するために、サイト間VPNトンネルは 172.16.10.0/24 と 172.16.20.0/24 の間でフローすると ASR1K Cisco IOS ルータの間で使用されます (ローカルおよびリモート サブネット)。

172.16.10.2 からのギグ 0/0/1 インターフェイスの 172.16.20.2 へのフロー VPN トラフィックをトレースするために使用するパケットトレース 設定およびプラットフォーム状態はここにありません:

```
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#
```

この例の ASR1K と Cisco IOS ルータ間の VPN トンネルによって暗号化される 5 つの ICMP パケットが 172.16.10.2 から 172.16.20.2 に送信されるとき、これらはパケットトレース出力です:

注: パケットトレースはトレースでパケットを暗号化するために使用する QFP Security Association (SA) ハンドルを示します正しい SA は暗号化のために使用されることを確認するために IPsec VPN 問題を解決するとき役立つ。

```
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、そう設定が必要とするおおよび利用可能であるメモリ量消費しますメモリ量を意識します。

パフォーマンス影響は有効になるパケットトレース オプションに、依存変わります。パケットトレースはトレースされるユーザ設定状態を一致するそれらのパケットのようなパケットの転送パフォーマンスだけに影響を及ぼします。キャプチャするためにパケットトレースを設定すること粒状おおよび詳細な情報リソースに非常に影響を与えます。

あらゆるトラブルシューティングと同様に、デバッグ状況がそれを保証するときしか反復的なアプローチを行い、より多くの詳しいトレース オプションを有効にしないことが最善です。

QFP DRAM 使用方法はこの数式と推定することができます:

メモリは = (統計オーバーヘッド) + pkts の数字必要としました* (サマリ サイズ + パスデータ サイズ + コピー サイズ)

注: 統計オーバーヘッドおよびサマリ サイズが 2 KB で固定およびである一方、128 B、それぞれ、パスデータ サイズおよびコピー サイズはユーザ側で設定できます。

関連情報

- [Cisco ASR1000 シリーズ 集約 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](#)
- [パケット破棄 on Cisco ASR1000 シリーズ サービス ルータ](#)
- [テクニカルサポートとドキュメント - Cisco Systems](#)