



MPLS 経由の IPv4 および IPv6 トラフィック に一致する統合デバッグ条件

表 1: 機能の履歴

機能名	リリース情報	説明
MPLS 経由の IPv4 および IPv6 に一致する統合デバッグ条件	Cisco IOS XE Catalyst SD-WAN リリース 17.11.1a	この機能により、MPLS ネットワーク経由の IPv4 および IPv6 トラフィックの一致に関連した問題を特定して解決するためのデバッグ条件が導入されます。

- [統合デバッグ条件について \(1 ページ\)](#)
- [統合デバッグ条件の制約事項 \(2 ページ\)](#)
- [統合デバッグ条件のユースケース \(2 ページ\)](#)
- [CLI を使用して MPLS を介した IPv4 および IPv6 トラフィックを照合するためのデバッグ \(3 ページ\)](#)
- [MPLS 経由の IPv4 および IPv6 トラフィックに一致する統合デバッグ条件の確認 \(4 ページ\)](#)

統合デバッグ条件について

Cisco IOS XE Catalyst SD-WAN デバイスは、MPLS パケットを介した IPv4 および IPv6 トラフィックのデバッグ条件を追加する機能をサポートしています。フィルタ条件を指定して、オーバーレイ IP アドレスと、オプションでスタック深度を含むアンダーレイ MPLS ラベルを選択できます。統合デバッグ条件を使用して、特定のフィルタリング基準に一致する特定の packets を識別して MPLS ネットワークのトラブルシューティングを行い、MPLS トラフィックに関連する問題をトラブルシューティングして、ネットワークがスムーズに実行されるようにします。MPLS を介した IPv4 と IPv6 の照合プロセスには、次の 3 つの手順があります。

1. Debugging

2. フィルタリング条件の指定
3. デバイスへのフィルタリング条件の適用

Cisco IOS XE Catalyst SD-WAN デバイスでは、MPLS ラベルは IP VRF を表すために使用され、OMP プロトコルによって配信されます。

統合デバッグ条件の制約事項

- MPLS ネットワークを介した IPv4 および IPv6 のデバッグ条件は、デバイス CLI を使用する場合にのみサポートされます。
- Cisco SD-WAN Manager は、パケットトレースのデバッグ出力の一部として、フィルタリングされたデバッグ結果を表示しません。詳細については、「[パケットトレース](#)」を参照してください。
- MPLS を介した IPv4 トラフィックと MPLS を介した IPv6 トラフィックを同時に照合するデバッグ条件を有効にすることはできません。
- 1 つの MPLS ラベルと、IPv4 または IPv6 トラフィックのいずれかのみを照合することができます。
- アレイごとの構成可能な Feature Invocation Array (FIA) エントリの数 は 31 です。

統合デバッグ条件のユースケース

次に、デバッグ条件を使用して MPLS ネットワーク上の IPv4 および IPv6 トラフィックを照合するユースケースを示します。

- デバッグ条件は、ネットワークの接続またはパフォーマンスの問題のトラブルシューティングに役立ちます。特定の IPv4 および IPv6 トラフィックを MPLS ネットワーク上で照合することで、管理者は問題を引き起こしているトラフィックを分離し、動作をより詳細に分析できます。
- デバッグ条件は、ネットワークに QoS ポリシーを実装する場合にも役立ちます。管理者は、MPLS ネットワーク上の特定の IPv4 および IPv6 トラフィックを照合することで、帯域幅要件、遅延感度、または優先順位などの特性に基づいて、さまざまなタイプのトラフィックに異なる QoS ポリシーを適用できます。詳細については、『[Cisco SD-WAN Forwarding and QoS Configuration Guide](#)』を参照してください。

CLI を使用して MPLS を介した IPv4 および IPv6 トラフィックを照合するためのデバッグ

debug platform condition mpls match-inner コマンドを使用し、*match-inner ipv4*、*match-inner ipv6*、*allow-no-label* などのさまざまなフィルタリング条件を使用して、MPLS を介した IPv4 および IPv6 トラフィックを照合します。デバッグ要件に基づいて、MPLS ラベル情報、内部 IPv4 および IPv6 アドレスを指定します。

1. MPLS ネットワークをデバッグします。

```
debug platform condition mpls
```

2. 要件に応じてデバッグ条件を指定します。

- MPLS ラベルを指定せずに MPLS ネットワーク上で IPv4 トラフィックをデバッグするには、次の条件を使用します。

```
match-inner ipv4
```

- MPLS ラベルを指定せずに MPLS ネットワーク上で IPv6 トラフィックをデバッグするには、次の条件を使用します。

```
match-inner ipv6
```

- **allow-no-label** 条件を使用して、MPLS ラベルがカプセル化されているかどうかに関係なく、IPv4 または IPv6 パケットを照合します。MPLS ネットワークからのカプセル化解除されたルータトラフィックを IPv4 パケットまたは IPv6 パケットとして送信する場合は、**allow-no-label** 条件を使用します。

```
match-inner ipv4{ipv4-source-prefix | any | host | payload-offset |  
protocol}{ipv4-destination-prefix | any | host} {application | both |  
ingress | egress}[bidirection] [allow-no-label]
```

,

```
match-inner ipv6{ipv6-source-prefix | any | host | payload-offset |  
protocol}{ipv4-destination-prefix | any | host} {application | both |  
ingress | egress}[bidirection] [allow-no-label]
```

- 特定の MPLS インターフェイスを通過する IPv4 パケットをフィルタリングするための MPLS ラベルと深さを指定します。

```
[interface interface-name interface-number] mpls depth-of-mpls-label  
match-inner ipv4 {ipv4-source-prefix | any | host | payload-offset |  
protocol}{ipv4-destination-prefix | any | host}{application | both |  
ingress | egress}[bidirection] [allow-no-label]
```

- 特定の MPLS インターフェイスを通過する IPv6 パケットをフィルタリングするための MPLS ラベルと深さを指定します。

```
[interface interface-name interface-number] mpls depth-of-mpls-label  
match-inner ipv6 {ipv6-source-prefix | any | host | payload-offset |
```

```
protocol){ipv6-destination-prefix | any | host}{application | both |
ingress | egress}[bidirection] [allow-no-label]
```

3. 特権 EXEC モードを終了します。

```
exit
```

例

次の例は、debug condition **debug platform condition mpls** コマンドを使用して、MPLS ネットワークを介した IPv4 または IPv6 トラフィックの照合を有効にする方法を示しています。

```
Device# debug platform condition mpls match-inner ipv4
Device# debug platform condition mpls match-inner ipv4 any any
Device# debug platform condition mpls match-inner ipv4 any any both
Device# debug platform condition mpls match-inner ipv4 any any both allow-no-label
```

詳細については、debug platform condition mpls コマンドのページを参照してください。

次の例は、debug condition **debug platform condition interface mpls** コマンドを使用して、特定のインターフェイスについて、MPLS ネットワークを介した IPv4 または IPv6 トラフィックの照合を有効にする方法を示しています。

```
Device# debug platform condition interface
Device# debug platform condition interface Loopback 3 mpls
Device# debug platform condition interface Loopback 3 mpls match-inner ipv6 host
2001:db8:3333:4444:5555:6666:7777:8888
Device# debug platform condition interface Loopback 3 mpls match-inner ipv6 host
2001:db8:3333:4444:5555:6666:7777:8888 any both
Device# debug platform condition interface Loopback 3 mpls match-inner ipv6 host
2001:db8:3333:4444:5555:6666:7777:8888 any both allow-no-label
```

MPLS 経由の IPv4 および IPv6 トラフィックに一致する統合デバッグ条件の確認

デバッグ状態の確認

次に、**show platform conditions** コマンドの出力例を示します。

```
Device# show platform conditions

Conditional Debug Global State: Start

Conditions
  Direction
-----|-----
All Interfaces & MPLS [ALL LABEL] & IPV4 Filter [ALL PROTO]
[host 10.20.24.17] [host 10.20.25.16] both bi

Feature Condition      Type      Value
-----|-----|-----
```

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

この出力では、「Conditional Debug Global State: Start」は、デバッグが有効であることを示しています。デバッグフィルタの設定も確認できます。

パケットトレース統計情報

次に、**show platform packet-trace statistics** コマンドの出力例を示します。

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

```
Packets Summary
```

```
  Matched  2
```

```
  Traced   2
```

```
Packets Received
```

```
  Ingress  0
```

```
  Inject   0
```

```
Packets Processed
```

```
  Forward  0
```

```
  Punt     0
```

```
  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

この出力では、「Matched」と「Traced」は一致したパケットおよびトレースされたパケットの数を示しています。

MPLS 経由で一致する IPv4 および IPv6 をデコード

次に、**show platform packet-trace packet 0 decode** コマンドの出力例を示します。

```
Device# show platform packet-trace packet 0 decode
```

```
Packet: 0 CBUG ID: 39872
```

```
Summary
```

```
Input : GigabitEthernet5
```

```
Output : GigabitEthernet1
```

```
State : FWD
```

```
Timestamp
```

```
Start : 10090556741529 ns (12/02/2022 05:54:03.730220 UTC)
```

```
Stop : 10090556747391 ns (12/02/2022 05:54:03.730226 UTC)
```

```
Path Trace
```

```
Feature: MPLS(Output)
Input : GigabitEthernet5
Output : Tunnel1
Label Stack Entry[1]: 0x003eb17f
StackEnd:YES, TTL:127, EXP:0, Label:1003, is SDWAN:YES
SDWAN Proto: IPV4, SDWAN dst_vpn: 1
Feature: MPLS_OUTPUT_L2_REWRITE
Entry : Output - 0x81323e6c
Input : GigabitEthernet5
Output : Tunnel1
Lapsed time : 239 ns
Feature: DEBUG_COND_MAC_EGRESS
Entry : Output - 0x81499d88
Input : GigabitEthernet5
Output : Tunnel1
Lapsed time : 33 ns
Feature: MPLS_OUTPUT_FRAG
Entry : Output - 0x814cdb3c
Input : GigabitEthernet5
Output : Tunnel1
Lapsed time : 152 ns
Feature: SDWAN_LOSS_PROTECT_TX
Entry : Output - 0x814d962c
Input : GigabitEthernet5
Output : Tunnel1
Lapsed time : 15 ns
Feature: MPLS_SDWAN_TUNNEL_OUTPUT_FINAL
Entry : Output - 0x814d60cc
Input : GigabitEthernet1
Output : Tunnel1
Lapsed time : 157 ns
Feature: SDWAN_TUNNEL_PRE_CHK_LKUP
Entry : Output - 0x814d911c
Input : GigabitEthernet1
Output : Tunnel1
Lapsed time : 19 ns
Feature: SDWAN_TUNNEL_PRE_QOS_OUTPUT
Entry : Output - 0x814d956c
Input : GigabitEthernet1
Output : Tunnel1
Lapsed time : 70 ns
Feature: SDWAN_TUNNEL_OUTPUT_UNIFY_FNF_FINAL
Entry : Output - 0x814aaa68
Input : GigabitEthernet1
Output : Tunnel1
Lapsed time : 95 ns
Feature: IPV4_OUTPUT_IPSEC_SDWAN_FEATURE
Entry : Output - 0x814c7480
Input : GigabitEthernet1
Output : Tunnel1
Lapsed time : 101 ns
Feature: IPV4_OUTPUT_IPSEC_CLASSIFY
Entry : Output - 0x814c7438
Input : GigabitEthernet1
Output : Tunnel1
Lapsed time : 151 ns
Feature: IPV4_IPSEC_FEATURE_RETURN
Entry : Output - 0x814c7478
Input : GigabitEthernet1
Output : Tunnel1
Lapsed time : 35 ns
Feature: IPV4_TUNNEL_PRE_GOTO_OUTPUT
Entry : Output - 0x814d60c4
Input : GigabitEthernet1
```

```
Output : GigabitEthernet1
Lapsed time : 461 ns
Feature: CBUG_OUTPUT_FIA
Entry : Output - 0x81499d68
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 35 ns
Feature: IPV4_VFR_REFRAG
Entry : Output - 0x814c894c
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 33 ns
Feature: DEBUG_COND_APPLICATION_OUT_CLR_TXT
Entry : Output - 0x81499d74
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 11 ns
Feature: IPV4_OUTPUT_L2_REWRITE
Entry : Output - 0x81323e50
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 53 ns
Feature: DEBUG_COND_MAC_EGRESS
Entry : Output - 0x81499d88
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 34 ns
Feature: DEBUG_COND_APPLICATION_OUT
Entry : Output - 0x81499d78
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 11 ns
Feature: IPV4_OUTPUT_FRAG
Entry : Output - 0x814c78e8
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 44 ns
Feature: IPV4_OUTPUT_DROP_POLICY
Entry : Output - 0x814d16b8
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 247 ns
Feature: IPV4_OUTPUT_SDWAN_FNF_FINAL
Entry : Output - 0x814aaa4c
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 74 ns
Feature: DEBUG_COND_OUTPUT_PKT
Entry : Output - 0x81499d8c
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 24 ns
Feature: MARMOT_SPA_D_TRANSMIT_PKT
Entry : Output - 0x814df374
Input : GigabitEthernet1
Output : GigabitEthernet1
Lapsed time : 780 ns
Packet Copy In
003eb17f 4500002a 0eb70000 7f11ed0a 10000001 30000001 a2c42710 00160000
801296c3 9baf96f9 9b56c437 0aca
Unable to decode layer 2 trying to skip to layer 3
MPLS
Label Stack Entry[1]
TTL : 127
```

```
Label : 1003
EXP : 0
StackEnd : YES
SDWAN : YES
SDWAN Label : 1003
SDWAN Proto : IPV4
IPv4
Version : 4
Header Length : 5
ToS : 0x00
Total Length : 42
Identifier : 0x0eb7
IP Flags : 0x0
Frag Offset : 0
TTL : 127
Protocol : 17 (UDP)
Header Checksum : 0xed0a
Source Address : 10.0.0.1
Destination Address : 10.0.0.1
UDP
Source Port : 41668
Destination Port : 10000
Length : 22
Checksum : 0x0000
Decode halted - unsupported udp port number
Packet Copy Out
52540095 dbed5254 007ffb83 08004500 0046ab1a 4000ff2f 9d4d0a01 0f0f0a01
10100000 8847003e b17f4500 002a0eb7 00007f11 ed0a1000 00013000 0001a2c4
27100016 00008012 96c39baf 96f99b56 c4370aca
ARPA
Destination MAC : 5254.0095.dbed
Source MAC : 5254.007f.fb83
Type : 0x0800 (IPV4)
IPv4
Version : 4
Header Length : 5
ToS : 0x00
Total Length : 70
Identifier : 0xab1a
IP Flags : 0x2 (Don't fragment)
Frag Offset : 0
TTL : 255
Protocol : 47 (GRE)
Header Checksum : 0x9d4d
Source Address : 10.0.0.1
Destination Address : 10.0.0.1
GRE ver 0
Optional Fields : None
Strict Source Route : NO
Recursion Control : 0
Flags : 0x00
Protocol : 0x8847 (MPLS)
MPLS
Label Stack Entry[1]
TTL : 127
Label : 1003
EXP : 0
StackEnd : YES
SDWAN : YES
SDWAN Label : 1003
SDWAN Proto : IPV4
IPv4
Version : 4
Header Length : 5
```

```
ToS : 0x00
Total Length : 42
Identifier : 0x0eb7
IP Flags : 0x0
Frag Offset : 0
TTL : 127
Protocol : 17 (UDP)
Header Checksum : 0xed0a
Source Address : 10.255.255.255
Destination Address : 10.255.255.254
UDP
Source Port : 41668
Destination Port : 10000
Length : 22
Checksum : 0x0000
Decode halted - unsupported udp port number
```

この例では、「Source Address」と「Destination Address」はデバッグ条件が成功したことを示しています。「MPLS」セクションには、指定された SD-WAN ラベルが表示されます。

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。