

使用mtrace V2排除组播故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[比较mtrace v1和mtrace v2](#)

[mtrace v2详细信息](#)

[ios-XR上的MTRACE v2](#)

[命令的语法](#)

[Examples](#)

[备注](#)

简介

本文档介绍Cisco IOS®XR中的mtrace^{版本2}。

先决条件

要求

本文档没有任何特定的要求。

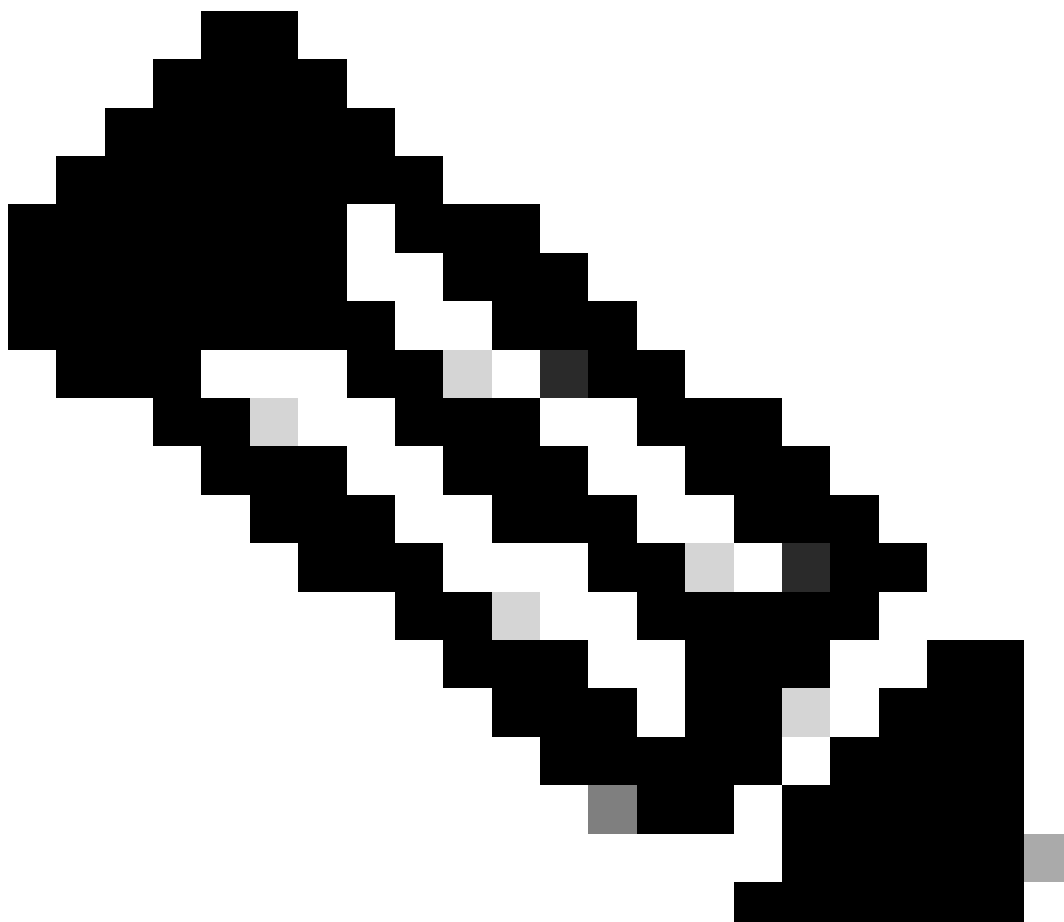
使用的组件

本文档特定于Cisco IOS®XR，但不限于特定的软件版本或硬件。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

比较mtrace v1和mtrace v2

- mtrace v2 Reply消息等同于mTrace v1 Response消息。
- mtrace v1仅支持IPv4组播。mTrace v2支持IPv4和IPv6组播。
- mtrace v1查询和响应消息是IGMP消息。所有mTrace v2数据包都是UDP。
- mtrace v1有一个路由协议字段，该路由协议是用于RPF到上游路由器的组播路由协议。mTrace v2有两个字段：一个用于RPF的单播路由协议，另一个用于运行到上游路由器的组播路由协议。
- mtrace v1和v2的目标相同，数据包语法也非常相似。
- mtrace v1和v2对路由协议和转发代码使用不同的代码集。
- mtrace v2支持地址系列IPv6和特定UDP端口号(33435)。



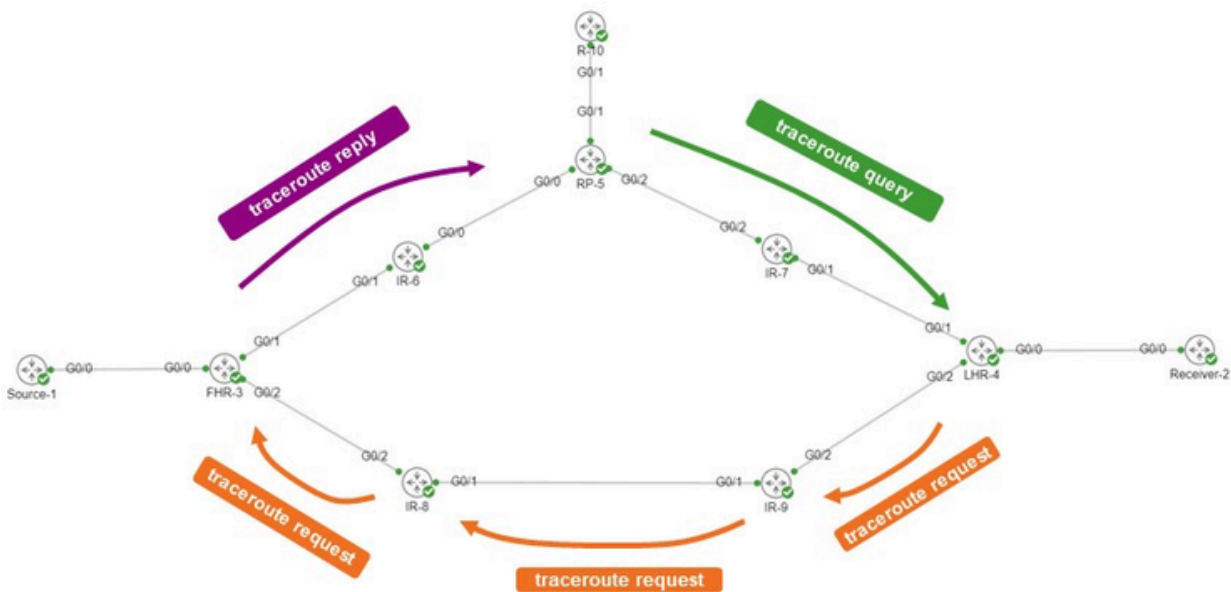
注意：cisco IOS XR软件上的MTRACE v2使用非标准端口33433，而不是标准RFC端口33435。

mtrace v2详细信息

- 该工具可用于跟踪从源到目的地的路径。它可验证采用的路径，也可指出任何问题，例如生存时间(TTL)或反向路径转发(RPF)。
- mtrace v2和v1的目标相同。mtrace验证路径的方法是将数据包发送到目的地（上一跳路由器或LHR），并向后追踪通往源（源树）或Rendez-Vous Point(RP)路由器的路径。这意味着您必须指定目标（单播地址）、源（单播地址）和组播组。
- mtrace功能的真正力量在于，可以从网络中的任何路由器（发起方）执行mtrace命令。它不需要是第一跳路由器(FHR)或RP。
- rfc 8487中规定了MTRACE v2:mtrace版本2:用于IP组播的Traceroute设施
- mtrace v1on IOS-XR基于草案：draft-ietf-idmr-traceroute-ipm
- mtrace v2不支持mVPN

有三种类型的数据包用于mtrace。这三个数据包共同使mtrace正常工作。发起方向最后一跳路由器发送一个mtrace Query数据包。此LHR将查询转换为请求数据包。然后，此数据包被单播逐跳转发到上游路由器。LHR和每个上游路由器都会添加一个响应数据块，其中包含接口地址、路由协议、转发代码等有用信息。当请求到达FHR时，它会将请求转换为回复数据包并将其转发给发起方。如果跟踪未完成，例如发生致命错误，如“no route”，中间路由器可能还会将应答返回到发起者。

请查看此映像，了解三种mtrace数据包类型的过程和处理。



发起方为R-10。LHR为LHR-4。FHR为FHR-3。RP为RP-5。网络运行PIM稀疏模式或任何源组播(ASM)。

mtrace请求消息如下所示。

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Type										Length										# Hops																			
Multicast Address																																							
Source Address																																							
Mtrace2 Client Address																																							
Query ID																				Client Port #																			

+++++

客户端地址是发起方的地址，即执行mTrace v2命令的路由器。

响应数据块包含感兴趣的信息。此信息将添加到请求消息中。每台路由器都会向请求消息添加一个响应数据块。以下是响应数据块。

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+++++
|   Type   |   Length   |   MBZ   |
+++++
|           Query Arrival Time           |
+++++
|           Incoming Interface Address     |
+++++
|           Outgoing Interface Address     |
+++++
|           Upstream Router Address       |
+++++
|                                           |
|           Input packet count on Incoming Interface           |
|                                           |
+++++
|                                           |
|           Output packet count on Outgoing Interface         |
|                                           |
+++++
|                                           |
|           Total number of packets for this source-group pair |
|                                           |
+++++
|           Rtg Protocol           |           Multicast Rtg Protocol           |
+++++
|           Fwd TTL           |           MBZ           |S|           Src Mask           |Forwarding Code|
+++++

```

此响应块信息用于显示traceroute输出。每个响应块都是mtrace输出中的一行。

路由协议和组播路由协议编号与IP组播MIB(RFC 5132)中的ipMcastRouteRtProtocol的值相同。它们与mtrace v1中使用的值不同。

IANA列出如下：

路由协议:

- other (1), -- not specified
- local (2), -- local interface
- netmgmt (3), -- static route
- icmp (4), -- result of ICMP Redirect

```

-- the following are all dynamic

-- routing protocols

egp      (5),  -- Exterior Gateway Protocol
ggp      (6),  -- Gateway-Gateway Protocol
hello    (7),  -- FuzzBall HelloSpeak
rip      (8),  -- Berkeley RIP or RIP-II
isis     (9),  -- Dual IS-IS
esIs     (10), -- ISO 9542
ciscoIgrp (11), -- Cisco IGRP
bbnSpfIgp (12), -- BBN SPF IGP
ospf     (13), -- Open Shortest Path First
bgp      (14), -- Border Gateway Protocol
idpr     (15), -- InterDomain Policy Routing
ciscoEigrp (16), -- Cisco EIGRP
dvmrp    (17), -- DVMRP
rpl      (18), -- RPL [RFC-ietf-roll-rpl-19]
dhcpc    (19), -- DHCP [RFC2132]

```

组播路由协议：

```

other(1),          -- none of the following
local(2),          -- e.g., manually configured
netmgmt(3),        -- set via net.mgmt protocol
dvmrp(4),
mospf(5),
pimSparseDense(6), -- PIMv1, both DM and SM
cbt(7),
pimSparseMode(8), -- PIM-SM
pimDenseMode(9),  -- PIM-DM
igmpOnly(10),
bgmp(11),
msdp(12)

```

此处显示了mtrace v2的转发代码。它们与mtrace v1中的不同。

Value	Name	Description
0x00	NO_ERROR	No error.
0x01	WRONG_IF	Mtrace2 Request arrived on an interface for which this router does not perform forwarding for the specified group to the source or RP.
0x02	PRUNE_SENT	This router has sent a prune upstream that applies to the source and group in the Mtrace2 Request.
0x03	PRUNE_RCVD	This router has stopped forwarding for this source and group in response to a Request from the downstream router.
0x04	SCOPED	The group is subject to administrative scoping at this router.
0x05	NO_ROUTE	This router has no route for the source or

		group and no way to determine a potential route.
0x06	WRONG_LAST_HOP	This router is not the proper LHR.
0x07	NOT_FORWARDING	This router is not forwarding this source and group out the Outgoing Interface for an unspecified reason.
0x08	REACHED_RP	Reached the Rendezvous Point.
0x09	RPF_IF	Mtrace2 Request arrived on the expected RPF interface for this source and group.
0x0A	NO_MULTICAST	Mtrace2 Request arrived on an interface that is not enabled for multicast.
0x0B	INFO_HIDDEN	One or more hops have been hidden from this trace.
0x0C	REACHED_GW	Mtrace2 Request arrived on a gateway (e.g., a NAT or firewall) that hides the information between this router and the Mtrace2 client.
0x0D	UNKNOWN_QUERY	A non-transitive Extended Query Type was received by a router that does not support the type.
0x80	FATAL_ERROR	A fatal error is one where the router may know the upstream router but cannot forward the message to it.
0x81	NO_SPACE	There was not enough room to insert another Standard Response Block in the packet.
0x83	ADMIN_PROHIB	Mtrace2 is administratively prohibited.

ios-XR上的MTRACE v2

命令的语法

使用率: mtrace

```
[
  ][
  ][
  ][
  ]
```

确保指定2以使用mtrace v2。

<#root>

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace?
```

```
mtrace mtrace2
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ?
```

```
  ipv4  IPv4 Address family
```

```
  ipv6  ipv6 Address Family
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ipv4 ?
```

```
  Hostname or A.B.C.D  Source to trace route from
```

源地址是发起方的地址。

```
<#root>
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ipv4 10.1.3.3 ?
```

```
  Hostname or A.B.C.D  Destination of route
```

```
  debug                Mtrace client-side debugging(cisco-support)
```

目的地址是LHR的地址。

```
<#root>
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ipv4 10.1.3.3 10.2.4.4 ?
```

```
Hostname or A.B.C.D  Group to trace route via  
debug                Mtrace client-side debugging(cisco-support)
```

组地址是正在跟踪的组播流的组地址。

```
<#root>
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ipv4 10.1.3.3 10.2.4.4 225.1.1.1 ?
```

```
Hostname or A.B.C.D  response address to receive response  
debug                Mtrace client-side debugging(cisco-support)
```


响应地址是traceroute应答返回的地址。

```
<#root>
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ipv4 10.1.3.3 10.2.4.4 225.1.1.1 10.0.0.10
```

```
?
```

```
<1-255> Time-to-live for multicasted trace request  
debug Mtrace client-side debugging(cisco-support)
```

Examples

请注意，命令可从网络中的任何路由器启动，但不一定是从启用了PIM/组播的路由器启动的，也不一定是从所调查的特定共享树或源树启动的。

```
<#root>
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ipv4 10.1.3.3 10.2.4.4 225.1.1.1 10.0.0.10
```

```
Type escape sequence to abort.
```

```
Mtrace from 10.1.3.3 to 10.2.4.4  
via group 225.1.1.1  
From source (?) to destination (?)  
Querying full reverse path...
```

```
0 10.2.4.4  
-1 10.4.7.4 PIM [10.1.3.0/24]  
-2 10.5.7.7 PIM [10.1.3.0/24]  
-3 0.0.0.0 PIM Reached RP/Core [10.1.3.0/24]
```

您可以看到已针对共享树(*,G)执行mtrace。 mtrace从最后一跳路由器10.2.4.4开始，然后在共享树上返回到RP(10.0.0.5)。 原因在于，LHR-4路由器没有为组225.1.1.1的源10.1.3.3设置(S, G)MRIB条目。

[10.1.3.0/24]部分是用于RPF信息的单播路由。 IOS-XR中的RPF信息始终是IPv4的/32条目。 此信息源自单播路由。 将显示此单播路由。

显示了组播协议。 这是PIM。

跳数以相反方式显示，从最后一跳路由器的0开始，一直变为负数，直到到达第一跳路由器。

接下来是源树的情况。

```
<#root>
```

```
RP/0/RP0/CPU0:LHR-4#
```

```
show mrib route 225.1.1.1
```

```
IP Multicast Routing Information Base
```

```
Entry flags: L - Domain-Local Source, E - External Source to the Domain,
```

```
  C - Directly-Connected Check, S - Signal, IA - Inherit Accept,
```

```
  IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,
```

```
  MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle
```

```
  CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet
```

```
  MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary
```

```
  MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN
```

```
Interface flags: F - Forward, A - Accept, IC - Internal Copy,
```

```
  NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,
```

```
  II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,
```

```
  LD - Local Disinterest, DI - Decapsulation Interface
```

```
  EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,
```

```
  EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,
```

```
  MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface
```

```
  IRMI - IR MDT Interface, TRMI - TREE SID MDT Interface, MH - Multihome Interface
```

```
(* ,225.1.1.1) RPF nbr: 10.4.7.7 Flags: C RPF
```

```
  Up: 1d21h
```

```
  Incoming Interface List
```

```
    GigabitEthernet0/0/0/1 Flags: A NS, Up: 1d21h
```

```
  Outgoing Interface List
```

```
    GigabitEthernet0/0/0/0 Flags: F NS LI, Up: 1d21h
```

```
(10.1.3.1,225.1.1.1)
```

```
RPF nbr: 10.4.9.9 Flags: RPF
```

```
  Up: 1d18h
```

```
  Incoming Interface List
```

```
    GigabitEthernet0/0/0/2 Flags: A, Up: 1d18h
```

```
  Outgoing Interface List
```

```
    GigabitEthernet0/0/0/0 Flags: F NS, Up: 1d18h
```

有源10.1.3.1的MRIB条目。 对该源执行mtrace命令后，该命令会显示不同的输出。

```
<#root>
```

```
RP/0/RP0/CPU0:R-10#
```

```
mtrace2 ipv4 10.1.3.1 10.2.4.4 225.1.1.1 10.0.0.10
```

Type escape sequence to abort.

Mtrace from 10.1.3.1 to 10.2.4.4

via group 225.1.1.1

From source (?) to destination (?)

Querying full reverse path...

```
0 10.2.4.4
-1 10.4.9.4 PIM [10.1.3.0/24]
-2 10.8.9.9 PIM [10.1.3.0/24]
-3 10.3.8.8 PIM [10.1.3.0/24]
-4 10.1.3.3 PIM [10.1.3.0/24]
```

请注意，反向路径现在为LHR4 - IR-9 - IR-8 - FHR-3。这是从FHR-3到LHR-4的源树。该路径与(S, G)的MRIB条目匹配。

```
<#root>
```

```
RP/0/RP0/CPU0:FHR-3#
```

```
show mrib route 225.1.1.1
```

IP Multicast Routing Information Base

Entry flags: L - Domain-Local Source, E - External Source to the Domain,

C - Directly-Connected Check, S - Signal, IA - Inherit Accept,

IF - Inherit From, D - Drop, ME - MDT Encap, EID - Encap ID,

MD - MDT Decap, MT - MDT Threshold Crossed, MH - MDT interface handle

CD - Conditional Decap, MPLS - MPLS Decap, EX - Extranet

MoFE - MoFRR Enabled, MoFS - MoFRR State, MoFP - MoFRR Primary

MoFB - MoFRR Backup, RPFID - RPF ID Set, X - VXLAN

Interface flags: F - Forward, A - Accept, IC - Internal Copy,

NS - Negate Signal, DP - Don't Preserve, SP - Signal Present,

II - Internal Interest, ID - Internal Disinterest, LI - Local Interest,

LD - Local Disinterest, DI - Decapsulation Interface

EI - Encapsulation Interface, MI - MDT Interface, LVIF - MPLS Encap,

EX - Extranet, A2 - Secondary Accept, MT - MDT Threshold Crossed,

MA - Data MDT Assigned, LMI - mLDP MDT Interface, TMI - P2MP-TE MDT Interface

IRMI - IR MDT Interface, TRMI - TREE SID MDT Interface, MH - Multihome Interface

```
(10.1.3.1,225.1.1.1) RPF nbr: 10.1.3.1 Flags: RPF
```

```
Up: 1d21h
```

```
Incoming Interface List
```

```
GigabitEthernet0/0/0/0 Flags: A, Up: 1d21h
```

```
Outgoing Interface List
```

```
GigabitEthernet0/0/0/2 Flags: F NS, Up: 1d18h
```

可以将debug IGMPmtrace v1用于显示trace路径沿途任何路由器上的mTrace数据包。mtrace v2使用

UDP数据包，因此IGMP调试不能用于mtrace v2。

但是，您可以重点关注IOS-XR上mtrace v2数据包使用的UDP端口33433。

示例：

在中间路由器上调试UDP mtracev2数据包。

IR-9:

```
<#root>
```

```
RP/0/RP0/CPU0:IR-9#
```

```
show access-lists
```

```
ipv4 access-list mtracev2
```

```
10 permit udp any eq 33433 any eq 33433
```

```
RP/0/RP0/CPU0:IR-9#
```

```
debug udp packet v4-access-list mtracev2 location 0/RP0/CPU0
```

```
RP/0/RP0/CPU0:IR-9#
```

```
show debug
```

```
#### debug flags set from tty 'con0_RP0_CPU0' ####
```

```
udp packet flag is ON with value '0x1:0x0:0x4:mtracev2:0x0:::'
```

```
RP/0/RP0/CPU0:IR-9#RP/0/RP0/CPU0:IR-9#
```

```
RP/0/RP0/CPU0:IR-9#
```

```
RP/0/RP0/CPU0:Jun 19 07:20:13.123 UTC: syslog_dev[115]: udp[214] PID-22001:
```

```
R
```

```
42469 ms LEN 60 10.4.9.4:33433 <-> 10.4.9.9:33433
```

```
RP/0/RP0/CPU0:Jun 19 07:20:13.123 UTC: syslog_dev[115]: udp[214] PID-22001:
```

```
RP/0/RP0/CPU0:Jun 19 07:20:13.139 UTC: syslog_dev[115]: udp[214] PID-22062:
```

```
S
```

```
15 ms LEN 100 10.8.9.9:33433 <-> 10.8.9.8:33433
```

```
RP/0/RP0/CPU0:Jun 19 07:20:13.139 UTC: syslog_dev[115]: udp[214] PID-22062:
```

中间路由器接收并发送mtrace v2消息。

备注

确保您知道哪些路由器是FHR和LHR。其它路由器无法完成mtrace。

如果路由器已同步时钟，则由于存在时间戳，您可以测量传播mtrace消息所需的时间。此时间仅作参考指示，因为这些消息在每一跳都被视为控制消息。

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。