Use mtrace V2 to Troubleshoot Multicast

Contents

Introduction

This document describes mtrace version 2 in Cisco IOS®XR.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is specific to Cisco IOS®XR, but it is not restricted to a specific software release or hardware.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Comparison mtrace v1 and mtrace v2

- mtrace v2 Reply message is equivalent to mTrace v1 Response message.
- mtrace v1 supports only IPv4 multicast. mTrace v2 supports IPv4 and IPv6 multicast.
- mtrace v1 Query and Response messages are IGMP messages. All mTrace v2 packets are UDP.
- mtrace v1 had a field for the Routing Protocol, which is the multicast routing protocol used for RPF towards the upstream router. mTrace v2 has two fields: one for the unicast routing protocol used for RPF and one for the multicast routing protocol running to the upstream router.
- The goal of mtrace v1 and v2 is the same and the packet syntax is very similar.
- mtrace v1 and v2 use different sets of codes for the routing protocols and forwarding codes.
- mtrace v2 supports address family IPv6 and a specific UDP port number (33435).

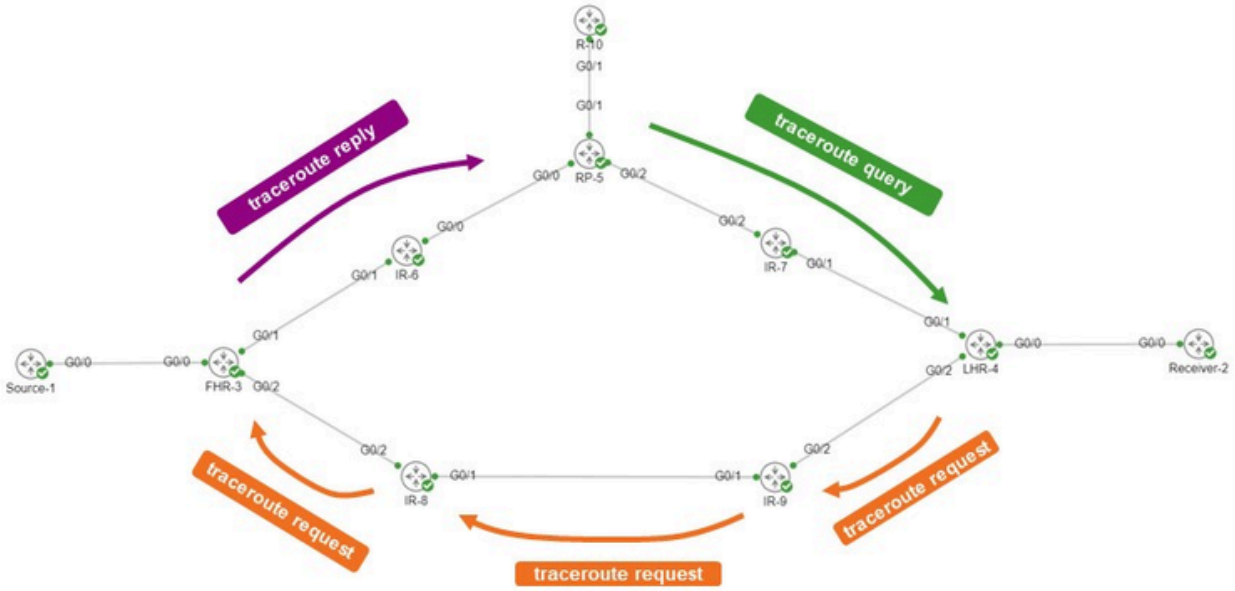
mtrace v2 Details

- The tool allows to trace the path from a source to a destination. It verifies the path taken and can also indicate any issues, for example with Time-To-Live (TTL) or Reverse Path Forwarding (RPF).
- The goal of mtrace v2 and v1 are the same. The way mtrace verifies the path is to send a packet to the destination (Last Hop Router or LHR) and trace the path backwards towards the source (source tree) or the Rendez-Vous Point (RP) router. This means that you have to specify the destination (unicast address), the source (unicast address) and the multicast Group.
- The real power of the mtrace feature is that the mtrace command can be done from any router (originator) in the network. It does not need to be the First Hop Router (FHR) or the RP.

- mtrace v2 is specified in RFC 8487: mtrace Version 2: Traceroute Facility for IP Multicast
- mtrace v1on IOS-XR is based on a draft: draft-ietf-idmr-traceroute-ipm
- mtrace v2 does not support mVPN

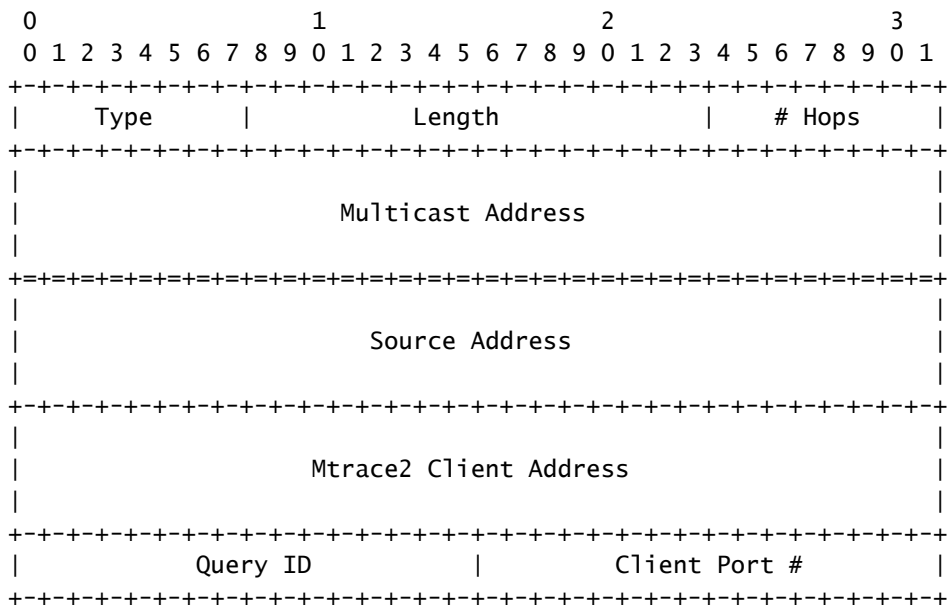
There are three types of packets used for mtrace. These three packets together make mtrace work. The originator sends out a mtrace Query packet towards the Last Hop Router. This LHR turns the Query into a Request packet. This packet is then unicast forwarded, hop-by-hop, towards the upstream router. The LHR and every upstream router add a response data block which contains useful information such as interface address, routing protocol, forwarding code and so on. When the Request arrives at the FHR, it turns the Request into a Reply packet and forwards it to the originator. If the trace is not completed, an intermediate router could return the Reply to the originator as well, if for example a fatal error such as “no route” occurred.

Please take a look at this image for the procedure and the handling of the three mtrace packet types.



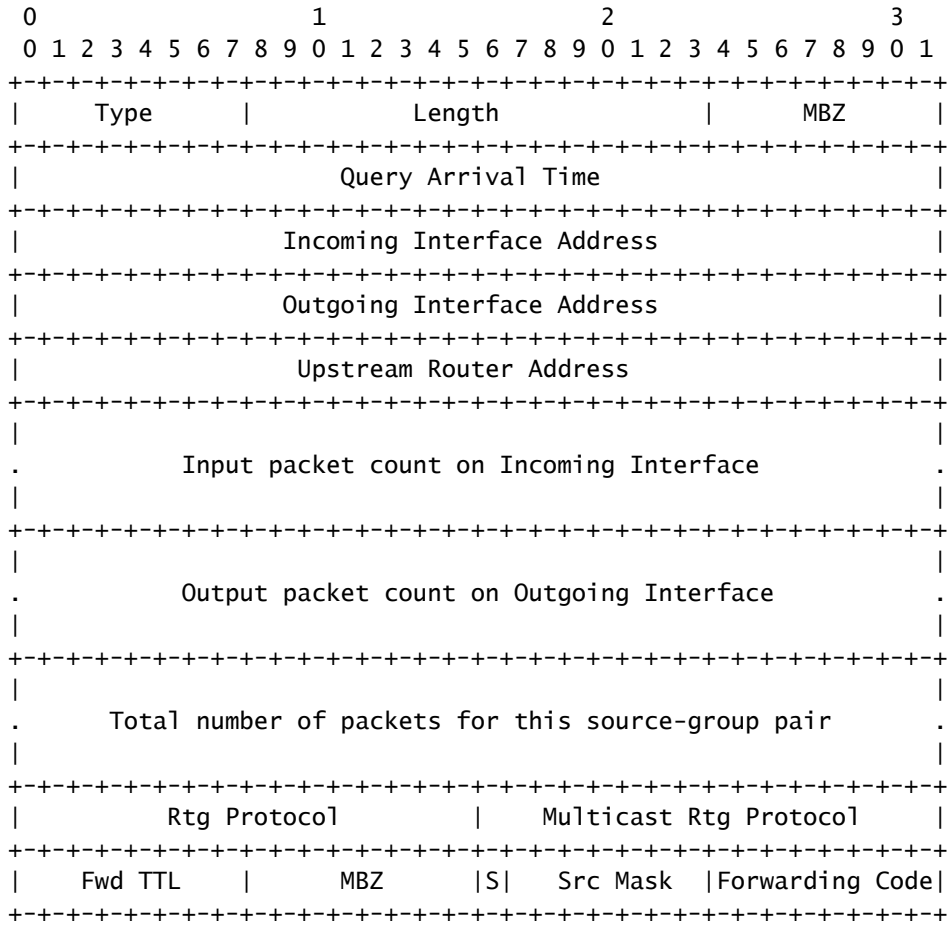
The originator is R-10. The LHR is LHR-4. The FHR is FHR-3. The RP is RP-5. The network is running PIM Sparse mode or Any Source Multicast (ASM).

The mtrace request message looks like this.



The client address is the address of the originator, so the router where you perform the mTrace v2 command.

The response data block holds interesting information. This piece of information is added to the Request message. Every router adds one response data block to the Request message. Here is the response data block.



It is this response block information that is used to display the traceroute output. Each response block is one line in the mtrace output.

The routing protocol and multicast routing protocol numbers are the same value as ipMcastRouteRtProtocol from the IP Multicast MIB (RFC 5132). They are not the same as the values used in mtrace v1.

IANA lists them as follows:

Routing protocols:

```
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]
dhcp       (19), -- DHCP [RFC2132]
```

Multicast routing protocols:

```
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)
```

The forwarding codes for mtrace v2 are shown here. They are not the same as in mtrace v1.

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.

mtrace v2 on IOS-XR

Syntax of the Command

Usage: mtrace <src_addr> [<dest_addr>] [<group_addr>] [<resp_addr>] [<ttl>]

Ensure to specify 2 in order to use 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
  <cr>
```

The source address is the originator's address.

<#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)
  <cr>
```

The destination address is the address of the LHR.

<#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)
  <cr>
```

The Group address is the Group address of the multicast stream being traced.

<#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)
  <cr>
```

The response address is the address the traceroute reply is returned to.

```
<#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)
  <cr>
```

Examples

Notice that the command can be initiated from any router in the network, not necessarily a PIM/multicast enabled router or along the specific shared or source tree under investigation.

```
<#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]
```

You can see the mtrace was performed for the shared tree (*,G). The mtrace started at the Last Hop Router 10.2.4.4 and went backwards on the shared tree to the RP (10.0.0.5). The reason for this is that the LHR-4 router does not have a (S,G) MRIB entry for the source 10.1.3.3 for Group 225.1.1.1.

The [10.1.3.0/24] part is the unicast route used for the RPF info. The RPF info in IOS-XR is always a /32 entry for IPv4. This info is derived from a unicast route. This unicast route is displayed.

There is the multicast protocol displayed. Here this is PIM.

The hop count is displayed in reverse, starting at 0 at the last hop router and going negative until the First Hop Router is reached.

The next is the case of the source tree.

<#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
```

There is an MRIB entry for Source 10.1.3.1. The mtrace command shows a different output when done for that source.

<#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]
```

Notice that the reverse path is now LHR4 – IR-9 – IR-8 – FHR-3. This is the source tree from FHR-3 to LHR-4. This matches the MRIB entries for (S,G).

<#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
```

You could use debug IGMP with mtrace v1 to display the mTrace packets on any router along the trace path. mtrace v2 uses UDP packets, so the IGMP debug cannot be used for mtrace v2.

You can however focus on UDP port 33433 which is used by mtrace v2 packets on IOS-XR.

Example:

Debug UDP mtracev2 packets on the intermediate router.

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:
```

The intermediate router receives and sends a mtrace v2 message.

Notes

Ensure you know which routers are the FHR and LHR. Other routers are not able to complete the mtrace.

If the routers have synchronized clocks, you can measure the time it takes to propagate the mtrace messages, because of the presence of the timestamps. This time is an indication only, as these messages are treated as control messages at every hop.