Medianet Metadata

This module provides an overview of medianet metadata. It also describes how metadata is used by different components of a network to make policy decisions.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Medianet Metadata

- The metadata transport mechanism (Resource Reservation Protocol [RSVP]) carries metadata only in the downstream direction, that is, toward the destination IP address.
- Metadata does not support high availability (HA). Therefore, after switchover, the RSVP path refresh messages are generated every 30 seconds to update the RSVP and metadata database.
- A path tear can happen in RSVP because of reservation preemption for higher priority reservation, but the flow could still be active. Metadata deletes entries in the database on path tear and reprograms the data forwarding path. The flow continues to be active without any metadata features applied on it.

- RSVP does not support Network Address Translation (NAT). Hence, metadata needs to track flow key and attribute information before and after NAT.

Information About Medianet Metadata

Metadata Overview

The metadata infrastructure provides a framework that allows data from one component to be available to another component on the same network element and across network elements.

Flow metadata is the data that describes a flow in the network. This metadata describes the five-tuple flow along with its attributes. Network elements can take action based on the metadata generated by the endpoints.

The metadata infrastructure consists of two major components—producers and consumers.

- Producers—Metadata producer is any source of metadata. The producer propagates all the attributes of a given flow. Producers can be anywhere in the network–endpoint, proxy agents or intermediate nodes. Metadata generated by the endpoints is supported. Producers use a specific transport protocol such as Resource Reservation Protocol (RSVP) for signaling metadata attributes to store the information in a database, referred to as the control plane database, which can then be used by the consumers.
- Consumers—Metadata consumer is any network element that uses the flow tuple and metadata provided by producers. The flow tuple and metadata can also be propagated along the media path to consumers in different network elements via a transport infrastructure.

Note Only the initiator of metadata is source aware. The initiator stores the source with its list of attributes along with the flow. But the downstream devices get only one list of attributes. The list is a consolidation of attributes from all sources with the attribute from a higher priority source, overriding the attribute from a lower priority source. Media Services Interface (MSI) has the highest priority followed by Media Services Proxy (MSP) and Network Based Application Recognition (NBAR).

Metadata Properties

Metadata is represented as a list of <Attribute, Value> pairs. Actions such as configuring the metadata values and updating and deleting the existing metadata are driven by the producers. Consumers read these metadata values and take appropriate action based on the control plane classification.

Metadata Control Plane Classification

Classifying network traffic allows you to organize traffic (that is, packets) into traffic classes or categories on the basis of whether the traffic matches specific criteria. You can classify network traffic to enable many quality of service (QoS) features on your network.

The metadata control plane classification is activated only when a consumer is registered with the metadata infrastructure. The metadata framework supports Cisco Common Classification Policy Language-based control plane classification.

Cisco Common Classification Policy Language is a replacement for feature-specific configuration commands. Cisco Common Classification Policy Language allows you to create traffic policies based on events, conditions, and actions. If Cisco Common Classification Policy Language classification succeeds, then the <Attribute, Value> pair is distributed to all the registered consumers.

In a scenario where QoS is a metadata consumer, the following steps briefly describe the control plane classification process:

- The required classification **match** commands are provisioned for a class map attached to the relevant target interface.
- Every incoming flow from the producer is matched against the provisioned class.
- If an appropriate match is found, relevant actions specified in the policy are performed.

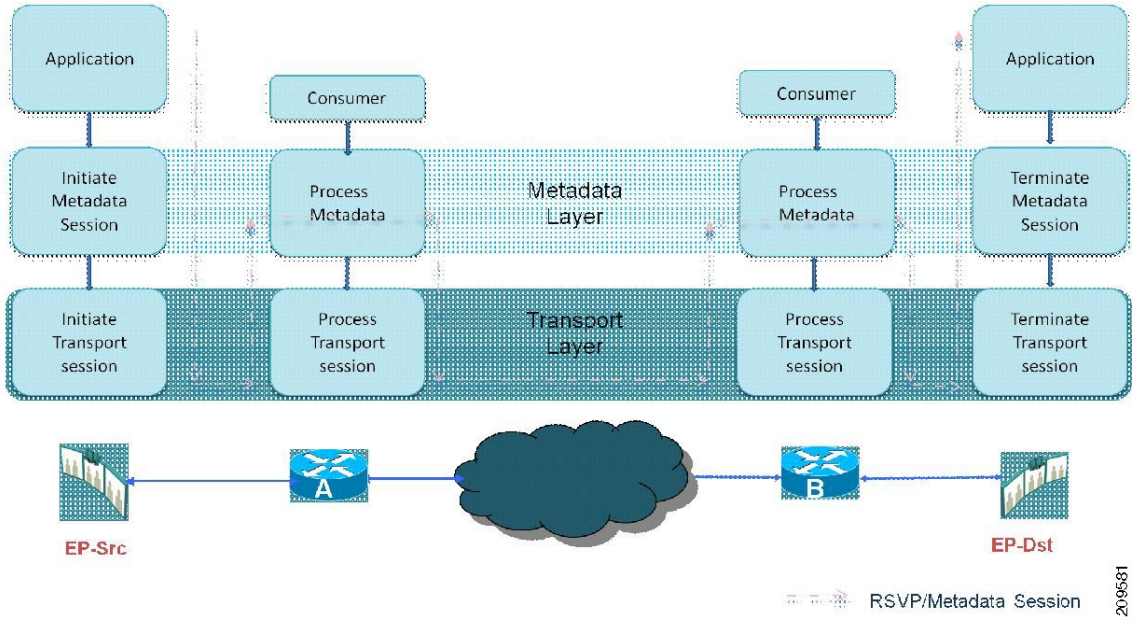
If the control plane classification is successful, then the <Attribute, Value> pair is distributed to all the consumers registered for metadata infrastructure. When packets related to the flow reach the network element, appropriate actions provisioned in the class are applied. For instance, if the action was set dscp 0xef, then this particular QoS action is applied on all packets matching this flow.

Metadata Transport

Metadata generated by the producers must be available at every network element in the media path. The metadata transport mechanism ensures that the metadata is propagated across the network and is delivered to all the network elements in the media path.

The figure below illustrates the metadata transport architecture.

Figure 1: Metadata Transport Architecture



The application at the source endpoint triggers the metadata layer to initiate a metadata session with the appropriate <Attribute, Value> pairs. The information with the <Attribute, Value> pairs is then carried along the media path and terminated at the destination endpoint.

The metadata with the <Attribute, Value> pairs is provided to the consumers at every network element, if the consumers are registered. Additional metadata elements that are generated at every network element can be sent along with the existing metadata. The metadata flows in the down stream of the media path.

Metadata applications have several subapplications. Each subapplication has an identifier. Metadata supports the following sub-applications:

- Traffic-type
- Transport-type
- Signaling-type
- Multiplex-type

Each subapplication is dependent on a specific application. The table below lists subapplications associated with eat application.

Table 1: Application to Subapplications Mapping

Application Name	Traffic Type	Transport Type	Signaling Type	Multiplex Type
cisco-phone	10 (control)	2 (rtp) 3 (rtcp))	1 (sip) 2 (bfcp) 3 (h323) 8 (mgcp) 9 (skinny)	--
citrix	1 (session) 3 (streaming) 4 (tunnel) 5 (realtime) 6 (interactive) 7 (bulk) 8 (background) 9 (desktop)	1 (ica) 5 (rdp)	--	--
vmware-view	1 (session) 2 (usb-redirection) 3 (streaming) 4 (tunnel) 9 (desktop) 11 (desktop-feedback)	4 (pcoip) 5 (rdp)	--	--

Application Name	Traffic Type	Transport Type	Signaling Type	Multiplex Type
wyse-zero-client	3 (streaming)	--	--	--
webex-meeting	3 (streaming) 10 (control) 12 (sharing)	6 (http)	--	--
telepresence-media	10 (control)	2 (rtp) 3 (rtcp)	--	1 (set)
telepresence-control	--	--	1 (sip) 2 (bfcp) 3 (h323) 4 (ccp) 5 (xccp) 6 (mscp) 7 (clue)	--

Metadata Flow Entries

Any producer can add flow metadata into the database and any consumer can access this information.

Metadata can be updated during the flow. When metadata attributes change during the flow, the network elements are notified through Resource Reservation Protocol (RSVP) PATH TEAR messages.

When the RSVP session terminates via a PATH TEAR message, the metadata framework listens to these messages and deletes the relevant flow entry in the database.

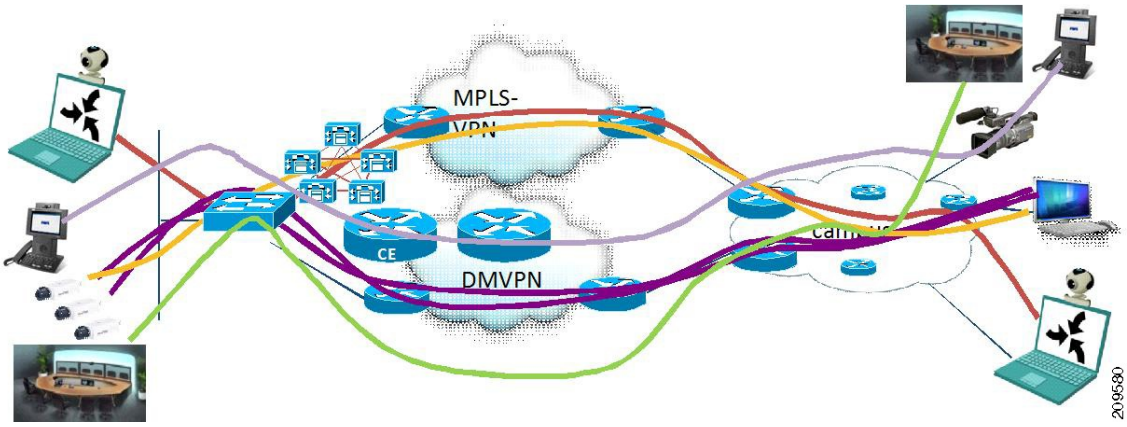
Metadata Reverse Flows

End devices enabled provisioned with metadata producers such as Media Services Interface (MSI), Media Services Proxy (MSP) and Network Based Application Recognition (NBAR) add flows to metadata database. When an end-device cannot signal metadata, a reverse metadata flow session is created to act as a proxy and signal metadata, and support QoS for the reverse session. The reverse sessions are created using the attributes of the forward session. The reverse flow session is enabled only on the device which is connected to the endpoint cannot signal metadata. That is when the device is not provisioned with a metadata producer such as MSP, MSI or NBAR

Medianet Metadata Implementation

The figure below illustrates a sample deployment scenario for the medianet metadata implementation.

Figure 2: Medianet Metadata Implementation



As illustrated in the figure above, two users from different locations can be in a WebEx, Telepresence, or a Cisco IP phone session.

This example assumes the users to be in a WebEx session. WebEx sessions typically require low latency guarantee from the network. QoS configurations can be used to obtain the required behavior. To achieve the required behavior, the required types of policy maps must be configured on the given interface to match the application ID of WebEx. Once this classification provisioning is done, metadata will also have a copy of this information in its classification database. One end of Webex session (endpoint A) signals the application as the metadata, using explicit signaling from the endpoints. The metadata information can be the application name, application ID, application version, and so on. This metadata information flows through the network along the media path.

Resource Reservation Protocol (RSVP) notifies the metadata framework about any incoming flow and provides the metadata information associated with the flow. A match action is performed between the decoded <Attribute, Value> pair and the WebEx metadata properties. If the match is successful, then the same information is propagated to the data plane. The data plane checks the appropriate classification requirements and takes the required QoS actions.

The following example shows how to configure QoS properties to work with the metadata framework. In the following sample configuration, a class map v1 is created.

```
! Creates a class-map with metadata-based filters
class-map match-all v1
match application webex-video
exit
!
```

Next, a policy map p1 is created and the class v1 is added to it. The packets belonging to class v1 are given priority by giving the entire class a guaranteed bandwidth of 1 Mbps. That is, the aggregate of all the flows that match the <Attribute, Value> pair defined in the class v1 are given a guaranteed bandwidth. Any other QoS solutions such as policing, marking, or queueing can also be applied as a classification criterion.

```
! Create policy map and apply the classification properties
policy-map p1
class v1
```

```
priority 1000
exit
```

Then, the policy map is attached to the target interface:

```
! Attach the policy map to the target interface
interface Ethernet 1/0
service-policy output p1
```

For more information about QoS network traffic classification and solutions such as policing, marking, or queuing, see the *Quality of Service Solutions Configuration Guide*.

How to Configure and Verify Medianet Metadata

Enabling Metadata Globally or on a Specific Interface

The first consumer registering for metadata triggers the enabling of metadata. The corresponding egress interface for a given flow enables metadata and Resource Reservation Protocol (RSVP) if they are not enabled already. Although you can disable metadata by using the **no metadata flow** command, we recommend that the metadata be enabled.

Perform this task to enable metadata on a specific interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **metadata flow**
4. **interface** *type number*
5. **metadata flow**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**metadata flow** **Example:** `Device(config)# metadata flow`	Enables metadata globally.
Step 4	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Specifies the interface type and number and enters interface configuration mode.
Step 5	**metadata flow** **Example:** `Device(config-if)# metadata flow`	Enables metadata on the specified interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Provisioning Control Plane Classification

Every flow that enters a network element needs to be classified for appropriate actions. Perform this task to provision control plane classification.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name*
4. **match application** *application-name*
5. **exit**
6. **policy-map** *policy-map-name*
7. **class** *class-map-name*
8. Enter QoS solution commands, as required.
9. **exit**
10. **interface** *type number*
11. **service-policy** *policy-map-name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* **Example:** `Device(config)# class-map class1`	Creates a class map to be used for matching packets to a specified class and enters QoS class-map configuration mode.
Step 4	**match application** *application-name* **Example:** `Device(config-cmap)# match application test-application`	Classifies the class map based on the application name specified.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits QoS class-map configuration mode.
Step 6	**policy-map** *policy-map-name* **Example:** `Device(config)# policy-map pt1`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters QoS policy-map configuration mode.
Step 7	**class** *class-map-name* **Example:** `Device(config-pmap)# class class1`	Specifies the name of the class whose policy you want to create or change.
Step 8	Enter QoS solution commands, as required.	Configures any QoS solution commands such as controlling, policing, classification, or marking. • For example, the **set dscp** command marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte.
Step 9	**exit** **Example:** `Device(config-pmap)# exit`	Exits QoS policy-map configuration mode and enters global configuration mode.
Step 10	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Specifies the interface type and number and enters interface configuration mode.
Step 11	**service-policy** *policy-map-name* **Example:** `Device(config-if)# service-policy pt1`	Attaches a policy map to an input interface.
Step 12	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

Typically, for the metadata information to propagate from the source to the destination, all the network elements along the media path need not support the metadata framework. However, perform the following steps to troubleshoot any flow-metadata-related problems along the path between the source and the destination:

- Perform the **ping** operation to test for the basic connectivity and reachability of the destination network element from the source.
- Enter the **show metadata flow** command and check the output to determine if the egress interface is correctly populated.
- Enable RSVP, if it was disabled intentionally (RSVP is enabled by default).
- Enter the **show metadata flow** command on the network elements along the media path to verify if the content of the metadata flow table is the same as that in the source network element. However, for you to be able to verify the metadata flow table of any network element, you must first enable metadata flow by using the **metadata flow** command.

Verifying Medianet Metadata Configuration

Use the following commands to verify the metadata configuration.

SUMMARY STEPS

1. **show metadata application table**
2. **show metadata flow classification-table**
3. **show metadata flow statistics**
4. **show metadata flow table**
5. **debug metadata flow**

DETAILED STEPS

Step 1 **show metadata application table**

Example:
```
Device# show metadata application table
```

Displays a list of metadata applications defined on the network element.

Step 2 **show metadata flow classification-table**

Example:
```
Device# show metadata flow classification table
```

Displays metadata control plane classification information.

Step 3 **show metadata flow statistics**

Example:
```
Device# show metadata flow statistics
```

Displays metadata flow statistics. The output includes event and memory details.

Step 4 **show metadata flow table**

Example:
```
Device# show metadata flow table
```
Displays details of every flow.

Step 5 **debug metadata flow**

Example:
```
Device# debug metadata flow all
```
Debugs the metadata flow and checks if the control plane classification was completed successfully.

Troubleshooting Medianet Metadata Flow

In the absence of endpoints, you can simulate the creation of flow entries for troubleshooting metadata flow. Perform this task to troubleshoot metadata flow.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **metadata flow entry** *entry-name*
4. **exit**
5. **metadata flow flow-specifier** *entry-name*
6. **source-ip** *ip-address* **source-port** *port-number*
7. **dest-ip** *ip-address* **dest-port** *port-number*
8. **exit**
9. **metadata flow session-params** *session-name*
10. **application name** *application-name*
11. **exit**
12. **metadata flow entry** *entry-name*
13. **flow-specifier** *flow-specifier-name*
14. **session-params** *session-name*
15. **end**
16. **debug metadata flow all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**metadata flow entry** *entry-name* **Example:** `Device(config)# metadata flow entry entry1`	Creates a flow entry with the specified name with five-tuple information and enters metadata entry configuration mode.
Step 4	**exit** **Example:** `Device(config-md-entry)# exit`	Exits metadata entry configuration mode and enters global configuration mode.
Step 5	**metadata flow flow-specifier** *entry-name* **Example:** `Device(config)# metadata flow flow-specifier flow1`	Enters metadata flow specifier configuration mode.
Step 6	**source-ip** *ip-address* **source-port** *port-number* **Example:** `Device(config-md-flowspec)# source-ip 209.165.201.16 source-port 1000`	Specifies the source IP address and source port number for the endpoint.
Step 7	**dest-ip** *ip-address* **dest-port** *port-number* **Example:** `Device(config-md-flowspec)# dest-ip 209.165.201.25 dest-port 1001`	Specifies the destination IP address and destination port number for the endpoint. • Use the **show metadata flow table** command to check if the metadata flow table is created. Refer to the "Verifying Metadata Attributes" section for sample output from the **show metadata flow table** command. You can check for the ingress and the egress interfaces and the source and destination IP addresses of the flow.
Step 8	**exit** **Example:** `Device(config-md-flowspec)# exit`	Exits metadata flow specifier configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 9	**metadata flow session-params** *session-name* **Example:** `Device(config)# metadata flow session-params session1`	Configures a name for the session that is newly created and adds it to the metadata flow table. • Enters metadata session parameters configuration mode.
Step 10	**application name** *application-name* **Example:** `Device(config-md-session-params)# application name app1`	Associates the specified application name to the session.
Step 11	**exit** **Example:** `Device(config-md-session-params)# exit`	Exits metadata session parameters configuration mode and enters global configuration mode.
Step 12	**metadata flow entry** *entry-name* **Example:** `Device(config)# metadata flow entry entry1`	Enters metadata entry configuration mode.
Step 13	**flow-specifier** *flow-specifier-name* **Example:** `Device(config-md-entry)# flow-specifier flow1`	Associates the flow specifier with the specified flow entry.
Step 14	**session-params** *session-name* **Example:** `Device(config-md-entry)# session-params session1`	Associates the session parameters with the specified flow entry.
Step 15	**end** **Example:** `Device(config-md-entry)# end`	Returns to privileged EXEC mode.
Step 16	**debug metadata flow all** **Example:** `Device# debug metadata flow all`	Debugs all metadata flow information. • Refer to the “Verifying Metadata Attributes” section for sample output from the **debug metadata flow all** command. • To check the control plane classification details, use the **show metadata flow classification-table** command.

Configuration Examples for Medianet Metadata

Example: Enabling Metadata Globally or on a Specific Interface

The following example shows how to enable metadata globally:

```
Device> enable
Device# configure terminal
Device(config)# metadata flow
Device(config)# exit
```

The following example shows how to enable metadata on a specific interface:

```
Device> enable
Device# configure terminal
Device(config)# interface fastethernet 0/1
Device(config-if)# metadata flow
Device(config-if)# exit
```

Example: Provisioning Control Plane Classification

```
Device> enable
Device# configure terminal
Device(config)# class-map class1
Device(config-cmap)# match application test-application
Device(config-cmap)# exit
Device(config)# policy-map pt1
Device(config-pmap)# class class1
Device(config-pmap-c)# exit
Device(config-pmap)# exit
Device(config)# interface fastethernet 0/1
Device(config-if)# service-policy pt1
Device(config-if)# end
```

Example: Verifying Metadata

The following is sample output from the **show metadata application table** command:

```
Device# show metadata application table

ID     Name                  Vendor                Vendor id
------------------------------------------------------------------------------
113    telepresence-media    -                     -
114    telepresence-contr$   -                     -
478    telepresence-data     -                     -
414    webex-meeting         -                     -
56     citrix                -                     -
81     cisco-phone           -                     -
472    vmware-view           -                     -
473    wyze-zero-client      -                     -
61     rtp                   -                     -
64     h323                  -                     -
5060   sip                   -                     -
554    rtsp                  -                     -
496    jabber                -                     -
```

The following is sample output from the **show metadata flow classification table**command:

```
Device# show metadata flow classification table

Target          Flow ID    Dir   Policy     Filter(s)
                                 Type
--------------+----------+-----+----------+-----------------------------------

Et0/0           5          OUT   PM         application webex-meeting vendor
                                            Cisco Systems, Inc. version 1.4.5
                                 QOS        application webex-meeting vendor
                                            Cisco Systems, Inc. version 1.4.5
Et0/1.2         3          OUT
Et0/1.2         5          IN
```

The following is sample output from the **show metadata flow statistics** command:

```
Device# show metadata flow statistics

Interface specific report :

Serial2/0: Ingress flows 0, Egress flows 0
Serial2/0: Ingress flows 0, Egress flows 0

Chunk statistics :

Type               Allocated       Returned        Failed

IP Flow             9                0               0
Flow Key            29               20              0
Source List         4                0               0
Flow Info           29               29              0
Attribute Data      29               29              0
Feature Object      2                0               0

Event Statistics:

Add Flow                : 9         Delete Flow             : 0
Received                : 30        Rejected                : 0
Transient               : 0         Posted                  : 29
Ingress Change          : 0         Egress Change           : 11
Unknown                  : 0         Source Limit Exceeded    : 0
```

The following is sample output from the **show metadata flow table** command:

```
Device# show metadata flow table

Total number of IPV4 metadata flows 6

Flow  To              From            Proto DPort SPort Ingress      Egress

4     10.0.0.1        10.0.0.2        UDP   49008 49007              Se2/0
6     10.0.0.3        10.0.0.4        UDP   49004 49003              Se2/0
5     10.2.0.3        10.2.0.6        UDP   49010 49009              Se2/0
2     10.2.1.6        10.2.2.6        UDP   49004 49003              Se2/0
1     10.2.2.6        10.2.3.6        UDP   49002 49001              Se2/0
3     10.2.3.6        10.2.3.7        UDP   49006 49005              Se2/0


Total number of IPV6 metadata flows 3

To                                      From
Flow  Proto DPort SPort Ingress      Egress

2001:DB8:1::1                           2001:DB8:1::2
9     UDP   49001 49000              Se2/0
2001:DB8:1::3                           2001:DB8:1::4
7     UDP   49001 49000              Se2/0
2001:DB8:1::12                          2001:DB8:1::13
```

```
8     UDP   49003 49002              Se2/0
```

The following is sample output from the **debug metadata flow all** command:

```
Device# debug metadata flow all

*Jul 14 08:07:23.155: FMD SIG: Process RSVP Event RSVP_FMD_EVENT_PAYLOAD_RECEIVED(1)
*Jul 14 08:07:23.155: FMD : fmd_post_events: posting event 0
*Jul 14 08:07:23.167: FMD Process Event - FMD_RSVP_TRANSPORT_ADD
*Jul 14 08:07:23.167: (fmd_add_event_process): For Source IP/Port : 67372036/1000
*Jul 14 08:07:23.167: FMD DB Lookup: Hash 391
*Jul 14 08:07:23.167: FMD Event for Ingress Interface Ethernet0/0 , Egress Interface
Ethernet0/1
*Jul 14 08:07:23.167: FMD Classification Src Type 96, Len 17, Value telepresence-data
*Jul 14 08:07:23.167: FMD Classification Dest Type 95, Len 4, Value
*Jul 14 08:07:23.167: App name telepresence-data id 218104286 in Metadata local app table
*Jul 14 08:07:23.167: FMD Classification Src Type 96, Len 11, Value webex-audio
*Jul 14 08:07:23.167: FMD Classification Dest Type 95, Len 4, Value
*Jul 14 08:07:23.167: App name webex-audio id 12 in Metadata local app table
*Jul 14 08:07:23.167: FMD Classification Src Type 96, Len 11, Value webex-audio
*Jul 14 08:07:23.167: FMD Classification Dest Type 96, Len 17, Value telepresence-data
*Jul 14 08:07:23.167: FMD Classification Src Type 96, Len 11, Value webex-audio
*Jul 14 08:07:23.167: FMD Classification Dest Type 0, Len 0, Value
*Jul 14 08:07:23.167: FMD Classification: Match Passed for type 95 value Router-201
*Jul 14 08:07:23.167: FMD Classification: Found 1 filters matching
*Jul 14 08:07:23.167: FMD Event: Input policy Matched, Add flow to CFT
*Jul 14 08:07:23.167: FMD Event: PPCP Binding Succeeded
*Jul 14 08:07:23.167: FMD fmd_add_update_ingress_cft_fo : fid 4

*Jul 14 08:07:23.167: FMD Event: Local Flow ID 0
*Jul 14 08:07:23.167: (fmd_add_event_process): Update with Template Address 79CD778, Md
Addr 947F810
*Jul 14 08:07:23.167: fmd_add_ipv4_flow_node_to_hash: Hash 391
*Jul 14 08:07:23.167: FMD Event: DB Addition Succeeded
```

Example: Troubleshooting Metadata Flow

The following example shows how to debug metadata globally:

```
Device# debug metadata flow all
```

Additional References for Medianet Metadata

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Metadata commands	Cisco IOS Quality of Service Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5101	*Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Medianet Metadata

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Medianet Metadata

Feature Name	Releases	Feature Information
Medianet Metadata	15.1(1)SY 15.3(1)T	The following commands were introduced or modified: **debug metadata**, **match application (class-map)**, **metadata application-params**, **metadata flow**, **metadata flow (troubleshooting)**, **show metadata application table**, and **show metadata flow**.