Multi-Topology Routing

First Published: February 27, 2007
Last Updated: July 23, 2010

Multi-Topology Routing (MTR) allows the configuration of service differentiation through class-based forwarding. MTR supports multiple unicast topologies and a separate multicast topology. A topology is a subset of the underlying network (or base topology) characterized by an independent set of Network Layer Reachability Information (NLRI). A topology can overlap with another or share any subset of the underlying network. MTR provides separate forwarding capabilities on a per topology basis. A separate forwarding table is maintained for each topology, allowing you to broadly apply independent forwarding configurations or add a level of granularity to independent forwarding configurations. MTR can be used, for example, to define separate topologies for voice, video, and data traffic classes.

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 for Multi-Topology Routing" section on page 63.

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Multi-Topology Routing

- You should have a clear understanding of the physical topology and traffic classification in your network before deploying MTR.
- MTR should be deployed consistently throughout the network. Cisco Express Forwarding (CEF) or distributed CEF (dCEF) and IP routing must be enabled on all networking devices.
- We recommend that you deconfigure custom route configurations, such as route summarization and default routes before enabling a topology and that you reapply custom route configuration only after the topology is fully enabled. This recommendation is designed to prevent traffic interruption, as some destinations may be obscured during the transition. It is also a best practice when disabling an existing topology. Custom route configuration is most useful when all of the more specific routes are available in the routing table of the topology.

Restrictions for Multi-Topology Routing

- Only the IPv4 (unicast and multicast) address family is supported.
- Multiple unicast topologies cannot be configured within a Virtual Routing and Forwarding (VRF) instance. However, multiple unicast topologies and a separate multicast topology can be configured under the global address space, and a separate multicast topology can be configured within a VRF.
- All topologies share a common address space. MTR is not intended to enable address reuse. Configuring address reuse in separate topologies is not supported.
- IP Differentiated Services or IP Precedence can be independently configured in a network where MTR is also deployed. However, MTR requires exclusive use of some subset of the DiffServ Code Point (DSCP) bits in the IP packet header for specific topology traffic. For this reason, simultaneous configuration must be carefully coordinated. Remarking DSCP bits in the IP packet header is not recommended or supported on routers that contain class-specific topologies.
- Distance Vector Multicast Routing Protocol (DVMRP) CLI and functionality are not provided in Cisco IOS software images that provide MTR support.

Information About Multi-Topology Routing

MTR Overview

By using MTR, you can configure service differentiation through class-based forwarding. There are two primary components to configuring MTR: independent topology configuration and traffic classification configuration.

A topology is defined as a subset of routers and links in a network for which a separate set of routes is calculated. The entire network itself, for which the usual set of routes is calculated, is known as the base topology. The base topology (or underlying network) is characterized by the NLRI that a router uses to calculate the global routing table to make routing and forwarding decisions. In other words, the base topology is the default routing environment that exists prior to enabling MTR.

Any additional topologies are known as class-specific topologies and are a subset of the base topology. Each class-specific topology carries a class of traffic and is characterized by an independent set of NLRI that is used to maintain a separate Routing Information Base (RIB) and Forwarding Information Base (FIB). This design allows the router to perform independent route calculation and forwarding for each topology.

Within a given router, MTR creates a selection of routes upon which to forward to a given destination. The specific choice of route is based on the class of the packet being forwarded, a class that is an attribute of the packet itself. This design allows packets of different classes to be routed independently from one another. The path that the packet follows is determined by classifiers configured on the routers and interfaces in the network. Figure 1 shows the base topology, which is a superset of the red, blue, and green topologies.

Figure 1 MTR Base Topology

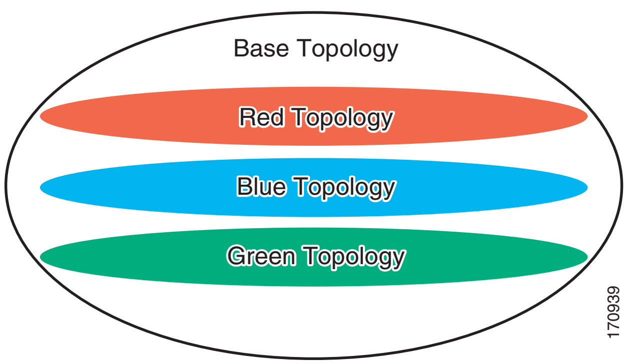


Figure 2 shows an MTR-enabled network that is configured using the service separation model. The base topology (shown in black) uses NLRI from all reachable devices in the network. The blue, red, and purple paths each represent a different class-specific topology. Each class-specific topology calculates a separate set of paths through the network. Routing and forwarding are independently calculated based on individual sets of NLRI that are carried for each topology.

Figure 2 Defining MTR Topologies

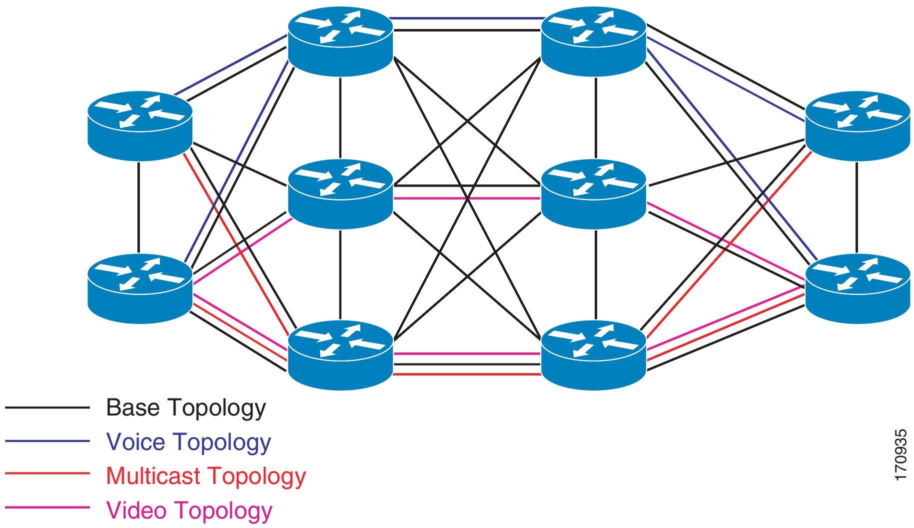
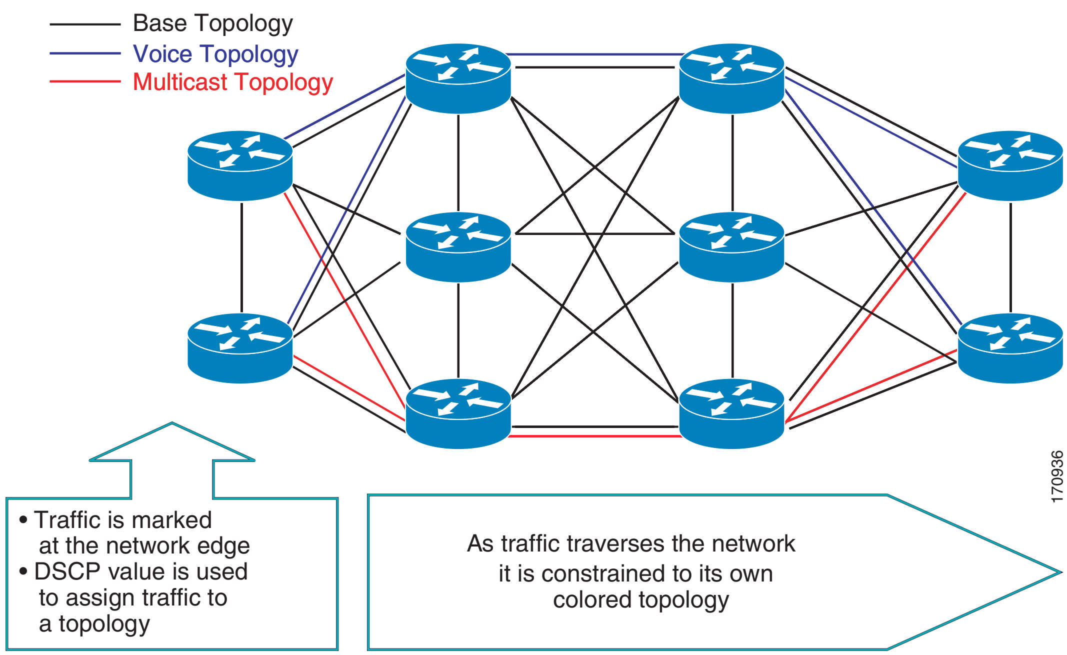


Figure 3 shows that the traffic is marked at the network edge. As the traffic traverses the network, the marking is used during classification and forwarding to constrain the traffic to its own colored topology.

Figure 3 Traffic Follows Class-Specific Forwarding Paths



The same topology can have configured backup paths. In Figure 4, the preferential path for the voice topology is represented by the solid blue line. In case this path becomes unavailable, you can configure MTR to choose the voice backup path represented by the dotted blue line. Both of these paths represent the same topology and none overlap.

Figure 4 *MTR Backup Contingencies Within a Topology*

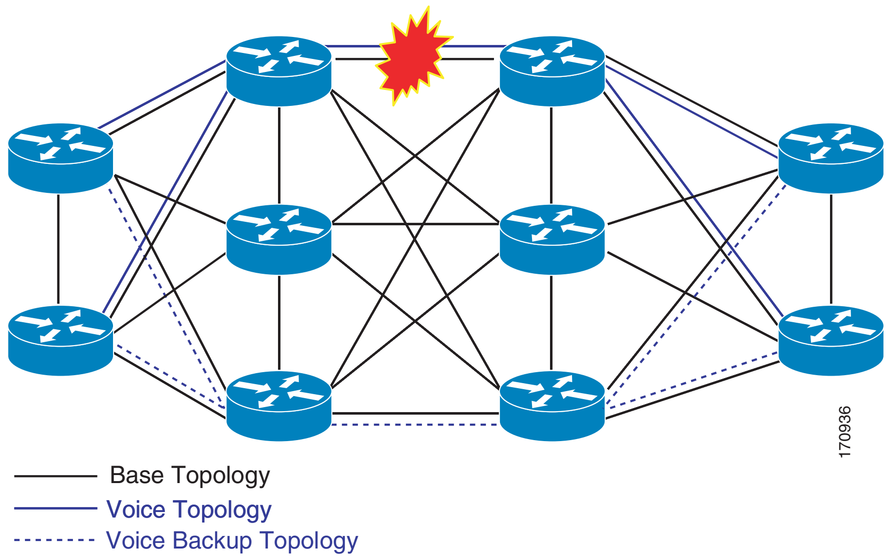
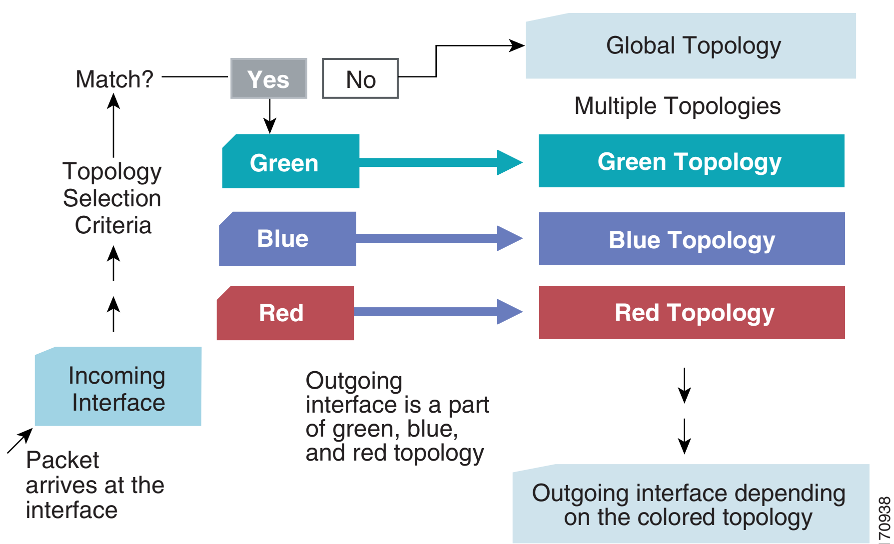


Figure 5 shows the MTR forwarding model at the system level. When a packet arrives at the incoming interface, the marking is examined. If the packet marking matches a topology, the associated topology is consulted, the next hop for that topology is determined, and the packet is forwarded. If there is no forwarding entry within a topology, the packet is dropped. If the packet does not match any classifier, it is forwarded to the base topology. The outgoing interface is a function of the colored route table in which the lookup is done.

Figure 5 *MTR Forwarding at the System Level*



MTR is implemented in Cisco IOS software on a per address family and subaddress family basis. MTR supports up to 32 unicast topologies (including the base topology) and a separate multicast topology. A topology can overlap with another or share any subset of the underlying network. You configure each topology with a unique topology ID. You configure the topology ID under the routing protocol, and the ID is used to identify and group NLRI for each topology in updates for a given protocol.

Unicast Topology Support for MTR

You can configure up to 32 unicast topologies on each router. You first define the topology by entering the **global-address-family** command in global configuration mode. The address family and optionally the subaddress family are specified in this step. You then enter the **topology** subcommand in global address family configuration mode. This command places the router in address family topology configuration mode, and the global topology configuration parameters are applied in this mode.

For each new topology that you configure on a router, you increase the total number of routes from the global routing table by the number of routes that are in each new topology [base+topology(*n*)]. If the router carries a large global routing table, and you plan to add a significant number of routes through MTR topology configuration, you can configure the **maximum routes** (MTR) command in address family topology configuration mode to limit the number of routes that the router accepts for a given topology and installs into the corresponding RIB.

Note Per-interface topology configuration parameters override configurations applied in global address family topology configuration mode and router address family topology configuration mode.

For detailed steps, see the "Configuring a Unicast Topology for MTR" section on page 14.

Multicast Topology Support for MTR

Cisco IOS software supports legacy (pre-MTR) IP multicast behavior by default. MTR support for IP multicast must be explicitly enabled. Legacy IP multicast uses reverse path forwarding on routes in the unicast RIB (base unicast topology) to build multicast distribution trees (MDTs).

MTR introduces a multicast topology that is completely independent from the unicast topology. MTR integration with multicast allows the user to control the path of multicast traffic in the network.

The multicast topology maintains separate routing and forwarding tables. The following list summarizes MTR multicast support that is integrated into Cisco IOS software:

- Conventional longest match support for multicast routes.
- RPF support for Protocol Independent Multicast (PIM).
- Border Gateway Protocol (BGP) MDT subaddress family identifier (SAFI) support for Inter-AS Virtual Private Networks (VPNs) (SAFI number 66).
- Support for static multicast routes is integrated into the **ip route topology** command (modifying the **ip mroute** command).

As in pre-MTR software, you enable multicast support by configuring the **ip multicast-routing** command in global configuration mode. You enable MTR support for multicast by configuring the **ip multicast rpf multitopology** command. The **global-address-family** command is entered with the IPv4 address family and multicast subaddress family. You then enter the **topology** command with the **base** keyword, and global topology configuration parameters are applied in this mode.

For detailed steps, see the "Configuring a Multicast Topology for MTR" section on page 16.

MTR Traffic Classification

MTR cannot be enabled on a router until traffic classification is configured, even if only one class-specific topology is configured. Traffic classification is used to configure topology specific forwarding behaviors when multiple topologies are configured on the same router. Traffic classification must be applied consistently throughout the network. Class-specific packets are associated with the corresponding topology table forwarding entries.

Traffic classification is configured by using the Modular QoS CLI (MQC). MTR traffic classification is similar to QoS traffic classification. However, there is an important distinction. MTR traffic classification is defined globally for each topology, rather than at the interface level as in QoS.

A subset of DSCP bits is used to encode classification values in the IP packet header. You configure a class map to define the traffic class by entering the **class-map** command in global configuration mode. Only the **match-any** keyword is supported for MTR. You associate the traffic class with a policy by configuring the **policy-map type class-routing ipv4 unicast** command in global configuration mode. You activate the policy for the topology by configuring the **service-policy type class-routing** command in global address family configuration mode. When configured, the service policy is associated with all interfaces on the router.

Some of the same goals can be achieved through QoS configuration, to which MTR provides a more powerful and flexible alternative.You can configure MTR traffic classification and IP Differentiated Services or IP Precedence-based traffic classification in the same network. However, MTR requires exclusive use of some subset of the DSCP bits in the IP packet header for specific topology traffic. In a network where MTR and QoS traffic classification are configured, simultaneous configuration must be carefully coordinated.

For detailed steps, see the "Configuring MTR Traffic Classification" section on page 19.

Routing Protocol Support for MTR

You must enable IP routing on the router for MTR to operate. MTR supports static and dynamic routing in Cisco IOS software. You can enable dynamic routing per-topology to support inter-domain and intra-domain routing. Route calculation and forwarding are independent for each topology. MTR support is integrated into Cisco IOS software for the following protocols:

- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Integrated Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)

You apply the per-topology configuration in router address family configuration mode of the global routing process (router configuration mode). The address family and subaddress family are specified when entering address-family configuration mode. You specify the topology name and topology ID by entering the **topology** command in address-family configuration mode.

You configure each topology with a unique topology ID under the routing protocol. The topology ID is used to identify and group NLRI for each topology in updates for a given protocol. In OSPF, EIGRP, and IS-IS, you enter the topology ID during the first configuration of the **topology** command for a class-specific topology. In BGP, you configure the topology ID by entering the **bgp tid** command under the topology configuration.

You can configure class-specific topologies with different metrics than the base topology. Interface metrics configured on the base topology can be inherited by the class-specific topology. Inheritance occurs if no explicit inheritance metric is configured in the class-specific topology.

You configure BGP support only in router configuration mode. You configure Interior Gateway Protocol (IGP) support in router configuration mode and in interface configuration mode.

By default, interfaces are not included in non-base topologies. For routing protocol support for EIGRP, IS-IS, and OSPF, you must explicitly configure a non-base topology on an interface. You can override the default behavior by using the **all-interfaces** command in address family topology configuration mode. The **all-interfaces** command causes the non-base topology to be configured on all interfaces of the router that are part of the default address space or the VRF in which the topology is configured.

For detailed steps, see these sections:

- Activating an MTR Topology by Using OSPF, page 22
- Activating an MTR Topology by Using EIGRP, page 24
- Activating an MTR Topology by Using IS-IS, page 26
- Configuring an MTR Topology in Interface Configuration Mode, page 34
- Activating an MTR Topology in Interface Configuration Mode by Using OSPF, page 36
- Activating an MTR Topology in Interface Configuration Mode by Using EIGRP, page 37
- Activating an MTR Topology in Interface Configuration Mode by Using IS-IS, page 39

BGP Routing Protocol Support for MTR

Before using BGP to support MTR, you should be familiar with the following concepts:

- BGP Network Scope, page 8
- MTR CLI Hierarchy Under BGP, page 8
- BGP Sessions for Class-Specific Topologies, page 9
- Topology Translation Using BGP, page 9
- Topology Import Using BGP, page 10

BGP Network Scope

To implement MTR for BGP, the scope hierarchy is required, but the scope hierarchy is not limited to MTR use. The scope hierarchy introduces some new configuration modes such as router scope configuration mode. You enter router scope configuration mode by configuring the **scope** command in router configuration mode. When this command is entered, a collection of routing tables is created.

You configure BGP commands under the scope hierarchy for a single network (globally), or on a per-VRF basis, and are referred to as scoped commands. The scope hierarchy can contain one or more address families.

MTR CLI Hierarchy Under BGP

The BGP CLI provides backward compatibility for pre-MTR BGP configuration and provides a hierarchical implementation of MTR. Router configuration mode is backward compatible with the pre-address family and pre-MTR configuration CLI. Global commands that affect all networks are configured in this configuration mode. For address-family and topology configuration, you configure general session commands and peer templates to be used in the address-family or in the topology configuration mode.

After configuring any global commands, you define the scope either globally or for a specific VRF. You enter address family configuration mode by configuring the **address-family** command in router scope configuration mode or in router configuration mode. Unicast is the default address family if no subaddress family (SAFI) is specified. MTR supports only the IPv4 address family with a SAFI of unicast or multicast.

Entering address family configuration mode from router configuration mode configures BGP to use pre-MTR-based CLI. This configuration mode is backward compatible with pre-existing address family configurations. Entering address family configuration mode from router scope configuration mode configures the router to use the hierarchical CLI that supports MTR. Address family configuration parameters that are not specific to a topology are entered in this address family configuration mode.

You enter BGP topology configuration mode by configuring the **topology** (BGP) command in address family configuration mode. You can configure up to 32 topologies (including the base topology) on a router. You configure the topology ID by entering the **bgp tid** command. All address family and subaddress family configuration parameters for the topology are configured here.

Note Configuring a scope for a BGP routing process removes CLI support for pre-MTR-based configuration.

The following example shows the hierarchy levels that are used when configuring BGP for MTR implementation:

```
router bgp <autonomous-system-number>
 ! Global commands
 scope {global | vrf <vrf-name>}
  ! Scoped commands
  address-family {<afi>} [<safi>]
   ! Address family specific commands
   topology {<topology-name> | base}
    ! topology specific commands
```

For detailed steps, see the "Activating an MTR Topology by Using BGP" section on page 28.

BGP Sessions for Class-Specific Topologies

MTR is configured under BGP on a per-session basis. The base unicast and multicast topologies are carried in the global (default) session. A separate session is created for each class-specific topology that is configured under a BGP routing process. Each session is identified by its topology ID. BGP performs a best-path calculation individually for each class-specific topology. A separate RIB and FIB are maintained for each session.

Topology Translation Using BGP

Depending on the design and policy requirements for your network, you might need to install routes from a class-specific topology on one router in a class-specific topology on a neighboring router. Topology translation functionality using BGP provides support for this operation. Topology translation is BGP neighbor-session based. You configure the **neighbor translate-topology** command by using the IP address and topology ID from the neighbor.

The topology ID identifies the class-specific topology of the neighbor. The routes in the class-specific topology of the neighbor are installed in the local class-specific RIB. BGP performs a best-path calculation on all installed routes and installs these routes into the local class-specific RIB. If a duplicate route is translated, BGP selects and installs only one instance of the route per standard BGP best-path calculation behavior.

Topology Import Using BGP

Topology import functionality using BGP is similar to topology translation. The difference is that routes are moved between class-specific topologies on the same router by using BGP. You configure this function by entering the **import topology** command and specify the name of the class-specific topology or base topology. Best-path calculations are run on the imported routes before they are installed into the topology RIB. This command also includes a **route-map** keyword to allow you to filter routes that are moved between class-specific topologies.

For detailed steps, see the "Importing Routes from an MTR Topology by Using BGP" procedure on page -32.

Interface Configuration Support for MTR

The configuration of an MTR topology in interface configuration mode allows you to enable or disable MTR on a per-interface basis. By default, a class-specific topology does not include any interfaces.

You can include or exclude individual interfaces by configuring the **topology** interface configuration command. You specify the address family and the topology (base or class-specific) when entering this command. The subaddress family can be optionally specified. If no subaddress family is specified, the unicast subaddress family is used by default.

You can include globally all interfaces on a router in a topology by entering the **all-interfaces** command in routing topology configuration mode. Per-interface topology configuration applied with the **topology** (interface) command overrides global interface configuration.

The interface configuration support for MTR has these characteristics:

- Per-interface routing configuration

  IGP routing and metric configurations can be applied in interface topology configuration mode. Per interface metrics and routing behaviors can be configured for each IGP. Interface configuration mode IGP command are documented in the configuration section for each routing protocol.

- OSPF interface topology configuration

  Interface mode OSPF configurations for a class-specific topology are applied in interface topology configuration mode. In this mode, you can configure an interface cost or disable OSPF routing without removing the interface from the global topology configuration.

- EIGRP interface topology configuration

  Interface mode EIGRP configurations for a class-specific topology are applied in interface topology configuration mode. In this mode, you can configure various EIGRP features.

- IS-IS interface topology configuration

  Interface mode IS-IS configurations for a class-specific topology are applied in interface topology configuration mode. By this mode, you can configure an interface cost or disable IS-IS routing without removing the interface from the global topology configuration.

For detailed steps, see the "Configuring an MTR Topology in Interface Configuration Mode" section on page 34.

Network Management Support for MTR

Context-based Simple Network Management Protocol (SNMP) support has been integrated into Cisco IOS software. SNMP support for MTR leverages context-based SNMP to extend support for existing MIBs from representing the management information for just the base topology to representing the same information for multiple topologies.

You can configure the SNMP agent software component on the router to pass a context string to existing MIB access functions. Network management applications can provide these context strings in SNMP transactions to direct those transactions to a specific virtual private network (VPN) routing and forwarding (VRF) instance, a specific topology, and/or routing protocol instance. The SNMP infrastructure on the receiving router verifies that a context string is defined for the router, and that the accompanying internal identifier is defined for that context string, before passing the context string and the internal identifier to the MIB access function.

For detailed steps, see the "Configuring SNMP Support for MTR" section on page 40.

Standard network management utilities, such as ping and traceroute, have been enhanced to support MTR. You can configure a standard or extended ping using the topology name in place of a hostname or IP address. Traceroute has been similarly enhanced. For detailed steps, see the "Testing Network Connectivity for MTR" section on page 47.

ISSU—MTR

All protocols and applications that support MTR and that also support In Service Software Upgrade (ISSU) have extended their ISSU support to include the MTR functionality. See the *Cisco IOS In Service Software Upgrade Process* module for information on ISSU-capable protocols and applications.

ISSU allows a high-availability (HA) system to run in Stateful Switchover (SSO) mode even when different versions of Cisco IOS software are running on the active and standby Route Processors (RPs). This feature allows the system to switch over to a secondary RP that is running upgraded (or downgraded) software and to continue forwarding packets without session loss and with minimal or no packet loss.

This feature is enabled by default.

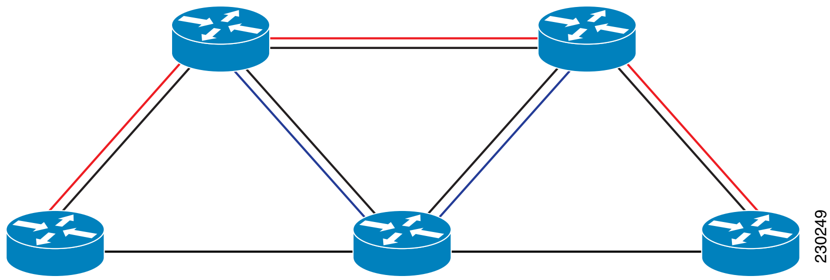
MTR Deployment Models

The base topology is the superset of all topologies in the network. It is defined by NLRI for all reachable routers regardless of the deployment model that is used. MTR can be deployed using the service separation MTR model shown in Figure 6, or it can deployed using the overlapping MTR model shown in Figure 7. Each of these models represent a different approach to deploying MTR. However, these models are not mutually exclusive. Any level of variation of a combined model can be deployed.

Service Separation MTR Model

Figure 6 shows the service separation model where no colored topologies (except for the base) overlap with each other. In the service separation model, each class of traffic is constrained to its own exclusive topology. This model restricts the given class of traffic to a subset of the network. This model is less configuration intensive because no topology-specific metrics need to be configured.

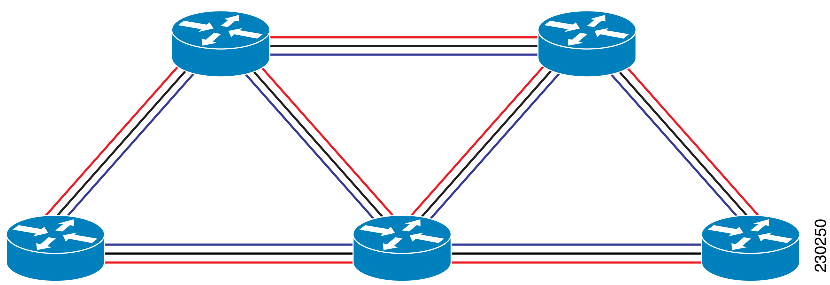
Figure 6 Service-Separation MTR Model



Overlapping MTR Model

In the overlapping MTR model, all topologies are configured to run over all routers in the network. This model provides the highest level of redundancy. All classes of traffic can use all links. Per-topology metrics are then configured to bias different classes of traffic to use different parts of the network. The redundancy that this model provides, however, makes it more configuration intensive. Figure 7 shows the red and gray topologies. All topologies are configured to run over all network routers. In this model, per-topology metrics are configured to bias the preferred routes for each topology.

Figure 7 Overlapping MTR Model



MTR Deployment Configuration

MTR supports both full and incremental deployment configurations. To support these options, MTR provides two different, configurable forwarding rules: strict forwarding mode for full deployment and incremental forwarding mode for an incremental deployment.

Full Deployment

Strict forwarding mode is the default forwarding mode in MTR. In this mode, the router looks for a forwarding route only in the class-specific FIB. If no forwarding route is found, the packet is dropped. In this mode, the router performs a longest match look up for the topology FIB entry. This mode is designed for full deployment, where MTR is enabled on every router in the network or every router in the topology. Strict forwarding mode should be enabled after an incremental deployment transition is been completed or when all routers in the network or topology are MTR enabled. Strict forwarding mode can be enabled after incremental forwarding mode by entering the **no forward-base** command in address family topology configuration mode.

Incremental Deployment

Incremental forwarding mode is designed to support transitional or incremental deployment of MTR, where routers in the network are not MTR enabled. In this mode, the router looks for a forwarding entry first in the class-specific FIB. If an entry is not found, the router looks for the longest match in the base topology FIB. If an entry is found in the base topology FIB, the packet is forwarded on the base topology. If a forwarding entry is not found in the base topology FIB, the packet is dropped.

This mode is designed to preserve connectivity during an incremental deployment of MTR and is recommended for use only during migration (the transition from a non-MTR to MTR enabled network). Class-specific traffic for a given destination is forwarded over contiguous segments of the class-specific topology containing that destination; otherwise, it is forwarded over the base topology.

This forwarding mode can be enabled to support mixed networks where some routers are not configured to run MTR. Incremental forwarding mode is enabled by entering the **forward-base** command in address family topology configuration mode.

Guidelines for Enabling and Disabling MTR

The section provides guidelines and procedures for enabling or disabling MTR in a production network. These guidelines assume that all participating networking devices are running a software image that supports MTR. They are designed to prevent major traffic interruptions due to misconfiguration and to minimize temporary transitional effects that can occur when introducing or removing a topology from a network. The guidelines described below must be implemented in the order that they are described.

First, create a class-specific topology on all networking devices and enable incremental forwarding mode by entering the **forward-base** command in the address family topology configuration.Configure incremental forwarding whenever a topology is introduced or removed from the network. The topology is defined as a global container at this stage. No routing or forwarding can occur within the topology. Routing protocol support should not be configured.

Second, configure classification rules for the class-specific topology. You must consistently apply classification on all routers in the topology; each router has identical classifier configuration. You activate the topology when you attach a valid classification configuration to the global topology configuration. You can use ping and trace route to verify reachability for interfaces and networking devices that are in the same topology and configured with identical classification.

Third, configure routing protocol support and/or static routing. Configure the routers in the topology one at a time. This configuration includes interface, router process, and routing protocol-specific metrics and filters.

Enable routing in the topology by using a physical pattern in a contiguous manner relative to a single starting point. For example, configure all interfaces on a single router, and then all interfaces on each adjacent router. Follow this pattern until the task is complete. The starting point can be on the edge or core of the network. This recommendation is designed to increase the likelihood that class-specific traffic is forwarded on the same paths in the incremental topology as it is on the full topology when MTR is completely deployed.

If your network design requires strict forwarding mode, you should disable incremental forwarding only after you configure routing on all routers in a given topology. At this stage, MTR is fully operational. Class-specific traffic is forwarded only over devices within the topology. Traffic that is not classified or destined for the topology is dropped.

When disabling a topology, reenable incremental forwarding mode. Remove custom route configuration, such as route summarization and default routes before disabling a topology, and reapply custom route configuration only after the topology is reenabled. This recommendation is designed to prevent traffic interruption, as some destinations might be obscured during the transition. Custom route configuration is most useful when all of the more specific routes are available in the routing table of the topology.

Note These recommendations apply only when a given classifier is enabled or disabled for a given topology. All other MTR configuration, including interface and routing protocol specific configuration (other than the topology ID) can be modified dynamically as necessary.

How to Configure Multi-Topology Routing

- Configuring a Unicast Topology for MTR, page 14 (required)
- Configuring a Multicast Topology for MTR, page 16 (required)
- Configuring MTR Traffic Classification, page 19 (required)
- Activating an MTR Topology by Using OSPF, page 22 (optional)
- Activating an MTR Topology by Using EIGRP, page 24 (optional)
- Activating an MTR Topology by Using IS-IS, page 26 (optional)
- Activating an MTR Topology by Using BGP, page 28 (optional)
- Importing Routes from an MTR Topology by Using BGP, page 32 (optional)
- Configuring an MTR Topology in Interface Configuration Mode, page 34 (optional)
- Activating an MTR Topology in Interface Configuration Mode by Using OSPF, page 36 (optional)
- Activating an MTR Topology in Interface Configuration Mode by Using EIGRP, page 37 (optional)
- Activating an MTR Topology in Interface Configuration Mode by Using IS-IS, page 39 (optional)
- Configuring SNMP Support for MTR, page 40 (optional)
- Enabling and Monitoring MTR Topology Statistics Accounting, page 44 (optional)
- Testing Network Connectivity for MTR, page 47 (optional)

Configuring a Unicast Topology for MTR

Perform this task to configure a unicast topology. Only Steps 1 through 4 are required to complete this task. The remaining steps are optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **global-address-family ipv4** [**multicast** | **unicast**]
4. **topology** {**base** | *topology-name*}
5. **all-interfaces**
6. **forward-base**

7. **maximum routes** *number* [*threshold* [**reinstall** *threshold*] | **warning-only**]
8. **shutdown**
9. **end**
10. **show topology** [**cache** [*topology-id*] | **ha** | [[**detail** | **interface** | **lock** | **router**] [**all** | **ipv4** | **ipv6** | **vrf** *vpn-instance*]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**global-address-family ipv4** [**multicast** | **unicast**] **Example:** `Router(config)# global-address-family ipv4`	Enters global address family topology configuration mode to configure the global topology. • The address family for the class-specific topology is specified in this step. The subaddress family can be optionally specified. Unicast is the default if no subaddress family is entered.
Step 4	**topology** {**base** | *topology-name*} **Example:** `Router(config-af)# topology VOICE`	Configures the global topology instance and enters address family topology configuration mode. • The **base** keyword is used to configure the base topology or a multicast topology. • The *topology-name* argument is entered to label a class-specific topology. Topology names are case-sensitive. For example, VOICE and voice identify two different topologies. • MTR supports 32 unicast topologies including the base topology.
Step 5	**all-interfaces** **Example:** `Router(config-af-topology)# all-interfaces`	(Optional) Configures the topology instance to use all interfaces on a router. • By default, no interfaces are used. **Note** The configuration of this command does not override the topology configuration applied in interface configuration mode.

	Command or Action	Purpose
Step 6	**forward-base** **Example:** `Router(config-af-topology)# forward-base`	(Optional) Configures the forwarding mode under a topology instance. • Strict mode (default) configures the router to look for forwarding entries only in the topology-specific FIB. If an entry is not found, the packet is dropped. • Incremental mode (enable form) configures the router to look first in the class-specific topology FIB. If a forwarding route is not found, then the router looks in the base topology FIB.
Step 7	**maximum routes** *number* [threshold [**reinstall** *threshold*] | **warning-only**] **Example:** `Router(config-af-topology)# maximum routes 1000 warning-only`	(Optional) Configures the maximum number of routes that a topology instance accepts and installs into the RIB. • Use the **warning-only** keyword to generate only a warning, to set an upper limit, and to set a lower limit (low water mark) for reinstalling routes after the maximum limit has been exceeded.
Step 8	**shutdown** **Example:** `Router(config-af-topology)# shutdown`	(Optional) Temporarily disables a topology instance without removing the topology configuration (while other topology parameters are configured and other routers are configured with MTR).
Step 9	**end** **Example:** `Router(config-af-topology)# end`	(Optional) Exits routing topology configuration mode and enters privileged EXEC mode.
Step 10	**show topology** [**cache** [*topology-id*] | **ha** | [[**detail** | **interface** | **lock** | **router**] [**all** | **ipv4** | **ipv6** | **vrf** *vpn-instance*]]] **Example:** `Router# show topology`	(Optional) Displays information about class-specific and base topologies.

What to Do Next

Repeat this task for each unicast topology instance that you need to create.

Configuring a Multicast Topology for MTR

Perform this task to configure a multicast topology. Only Steps 1 through 6 are required to complete this task. The remaining steps are optional.

Restrictions

- Only a single multicast topology can be configured, and only the **base** keyword can be entered when the multicast topology is created in Step 6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing** [**vrf** *name*]
4. **ip multicast rpf multitopology**
5. **global-address-family ipv4** [**multicast** | **unicast**]
6. **topology** {**base** | *topology-name*}
7. **route-replicate from** {**multicast** | **unicast**} [**topology** {**base** | *name*}] *protocol* [**route-map** *name* | **vrp** *name*]
8. **use-topology unicast** {**base** | *topology-name*}
9. **shutdown**
10. **end**
11. **show topology** [**cache** [*topology-id]* | **ha** | [[**detail** | **interface** | **lock** | **router**] [**all** | **ipv4** | **ipv6** | **vrf** *vpn-instance*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip multicast-routing** [**vrf** *name*] **Example:** `Router(config)# ip multicast-routing`	Enables IP multicast routing.
Step 4	**ip multicast rpf multitopology** **Example:** `Router(config)# ip multicast rpf multitopology`	Enables MTR support for IP multicast routing.
Step 5	**global-address-family ipv4** [**multicast** | **unicast**] **Example:** `Router(config)# global-address-family ipv4 multicast`	Enters global address family configuration mode to configure the global topology. • The address family for the class-specific topology is specified in this step. The subaddress family can be specified. Unicast is the default if no subaddress family is entered.

	Command or Action	Purpose
Step 6	**topology** {**base** | *topology-name*} **Example:** `Router(config-af)# topology base`	Configures the global topology instance and enters address family topology configuration mode. • Only the **base** keyword can be accepted for a multicast topology.
Step 7	**route-replicate from** {**multicast** | **unicast**} [**topology** {**base** | *name*}] *protocol* [**route-map** *name* | **vrf** *name*] **Example:** `Router(config-af-topology)# route-replicate from unicast topology VOICE ospf 100 route-map map1`	(Optional) Replicates (copies) routes from another multicast topology RIB. • The *protocol* argument is configured to specify the protocol which is the source of the route. Routes can be replicated from the unicast base topology or a class-specific topology. **Note** However, route replication cannot be configured from a class-specific topology that is configured to forward the base topology (incremental forwarding). • Replicated routes can be filtered through a route map before they are installed into the multicast RIB.
Step 8	**use-topology unicast** {**base** | *topology-name*} **Example:** `Router(config-af-topology)# use-topology unicast VIDEO`	(Optional) Configures a multicast topology to perform RPF computations using a unicast topology RIB. • The base or a class-specific unicast topology can be configured. When this command is configured, the multicast topology uses routes in the specified unicast topology table to build multicast distribution trees. **Note** This multicast RIB is not used when this command is enabled, even if the multicast RIB is populated and supported by a routing protocol.
Step 9	**shutdown** **Example:** `Router(config-af-topology)# shutdown`	(Optional) Temporarily disables a topology instance without removing the topology configuration (while other topology parameters are configured and other routers are configured with MTR).
Step 10	**end** **Example:** `Router(config-af-topology)# end`	(Optional) Exits address family topology configuration mode and enters privileged EXEC mode.
Step 11	**show topology** [**cache** [*topology-ID*] | **ha** | [[**detail** | **interface** | **lock** | **router**] [**all** | **ipv4** | **ipv6** | **vrf** *vpn-instance*]] **Example:** `Router# show topology detail`	(Optional) Displays information about class-specific and base topologies.

What to Do Next

The topology is not activated until classification is configured. Proceed to the “Configuring MTR Traffic Classification” section on page 19 to configure classification for a class-specific topology.

Configuring MTR Traffic Classification

Perform this task to define MTR traffic classification. Traffic classification is used to associate different classes of traffic with different topologies when multiple topologies are configured on the same router.

Following the correct order of the commands in this task is very important. Ensure that all configuration that affects traffic classification is complete before entering the **service-policy type class-routing** command.

Prerequisites

- Before configuring MTR traffic classification, you should be familiar with all the concepts documented in the "MTR Traffic Classification" section on page 7.
- A topology must be defined globally (rather than at the interface level as in QoS) before traffic classification can be configured.
- All routers throughout the network have the same definition of classifiers and the same sequencing of classifiers.
- In a network where MTR and QoS traffic classification is configured, simultaneous configuration must be carefully coordinated.

Restrictions

- MTR classification values must be unique for each topology. An error message is generated if you attempt to configure overlapping values.
- A topology cannot be placed in the shutdown state if it is referenced by any active policy map.
- A subset of DSCP bits is used to encode classification values in the IP packet header. Certain DSCP values are reserved. These DSCP values are commonly used by routing software components for purposes unrelated to MTR (for example, OSPF, BFD, and/or SNMP). Using these values for MTR classification is likely to interfere with correct operation of the router and is strongly discouraged. These values include:
  - DSCP 48 (cs6)
  - DSCP 16 (cs2)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map match-any** *class-map-name*
4. **match** [**ip**] **dscp** *dscp-value* [*dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value*]
5. **exit**
6. **policy-map type class-routing ipv4 unicast** *policy-map-name*
7. **class** {*class-name* | **class-default**}
8. **select-topology** *topology-name*
9. **exit**

10. **exit**
11. **global-address-family ipv4** [**multicast** | **unicast**]
12. **service-policy type class-routing** *policy-map-name*
13. **end**
14. **show topology detail**
15. **show policy-map type class-routing ipv4 unicast** [**interface** [*interface-type interface-number*]]
16. **show mtm table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`class-map match-any`** *`class-map-name`* **Example:** `Router(config)# class-map match-any VOICE-CLASS`	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. • The MTR traffic class is defined using this command. **Note** The **match-any** keyword must be entered when configuring classification for MTR.
Step 4	**`match`** `[`**`ip`**`]` **`dscp`** *`dscp-value`* `[`*`dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value`*`]` **Example:** `Router(config-cmap)# match ip dscp 9`	Identifies a DSCP value as a match criteria. • Use the *dcsp-value* argument to define a specific metric value. • Do not use the DSCP values 48 and 16. See "Restrictions" section on page 19 for more information.
Step 5	**`exit`** **Example:** `Router(config-cmap)# exit`	Exits class-map configuration mode.
Step 6	**`policy-map type class-routing ipv4 unicast`** *`policy-map-name`* **Example:** `Router(config)# policy-map type class-routing ipv4 unicast VOICE-CLASS-POLICY`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode. • If you do not specify the **type** keyword option, the command defaults to the QoS policy.

	Command or Action	Purpose
Step 7	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class VOICE-CLASS`	Specifies the name of the class whose policy you want to create or change or specifies the default class and enters policy-map class configuration mode. • The class map is referenced. • For a class map to be referenced in a class-routing policy map, it must first be defined by the **class-map** command as shown in Step 3.
Step 8	**select-topology** *topology-name* **Example:** `Router(config-pmap-c)# select-topology VOICE`	Attaches the policy map to the topology. • The topology name configured by the **topology** command in global address family configuration mode is referenced. See Step 4 of the "Configuring a Unicast Topology for MTR" section on page 14 section.
Step 9	**exit** **Example:** `Router(config-pmap-c)# exit`	Exits policy-map class configuration mode.
Step 10	**exit** **Example:** `Router(config-pmap)# exit`	Exits policy-map configuration mode.
Step 11	**global-address-family ipv4** [**multicast** | **unicast**] **Example:** `Router(config)# global-address-family ipv4`	Enters global address family configuration mode to configure MTR.
Step 12	**service-policy type class-routing** *policy-map-name* **Example:** `Router(config-af)# service-policy type class-routing VOICE-CLASS-POLICY`	Attaches the service policy to the policy map for MTR traffic classification and activates MTR. • The *policy-map-name* argument must match that configured in step 6. **Note** After this command is entered, traffic classification is enabled. Ensure that all configuration that affects traffic classification is complete before entering this important command.
Step 13	**end** **Example:** `Router(config-af)# end`	Exits global address family configuration mode and returns to privileged EXEC mode.
Step 14	**show topology detail** **Example:** `Router# show topology detail`	(Optional) Displays detailed information about class-specific and base topologies.

	Command or Action	Purpose
Step 15	**show policy-map type class-routing ipv4 unicast** [**interface** [*interface-type interface-number*]] **Example:** `Router# show policy-map type class-routing ipv4 unicast`	(Optional) Displays the class-routing policy map configuration. • If you specify the **interface** keyword without the argument, statistics on all interfaces under the global space are displayed.
Step 16	**show mtm table** **Example:** `Router# show mtm table`	(Optional) Displays information about the DSCP values assigned to each topology.

What to Do Next

The next four tasks show how to enable MTR support under a routing protocol. Proceed to "Activating an MTR Topology by Using OSPF" section on page 22 to enable routing protocol support.

Activating an MTR Topology by Using OSPF

Perform this task to configure OSPF for an MTR topology. Only MTR commands are shown in this task.

Prerequisites

- Before using OSPF to support MTR, you should be familiar with the concepts documented in the "Routing Protocol Support for MTR" section on page 7.
- A global topology configuration has been configured and activated.
- Check your OSPF router configuration and enter the topology-aware router configuration commands in router address family configuration mode.
- Several OSPF router configuration commands need to be topology-aware. Before you configure OSPF MTR, you need to enter these commands in router address family configuration mode if they are used in your original OSPF router configuration.
  - **area** *area-id* **default-cost** *cost*
  - **area** *area-id* **filter-list prefix** {*prefix-list-name* **in** | **out**}
  - **area** *area-id* **nssa** [**default-information-originate** [**metric** *metric-number*] [*metric-type*]] | [**no-redistribution**] | [**no-summary**] [**metric**] [*metric-type*]] [**translate type7 suppress-fa**]
  - **area** *area-id* **range** *ip-address mask* [**advertise** | **not-advertise**] [**cost** *cost*]
  - **area** *area-id* **stub** [**no-summary**]
  - **area** *transit-area-id* **virtual-link** *transit-router-id* **topology disable**
  - **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*]
  - **default-metric** *metric-value*
  - **discard-route** [**external** | **internal**]
  - **distance ospf** {**external** *dist1* | **inter-area** *dist2* | **intra-area** *dist3*}

- **distribute-list in (IP)**
- **distribute-list out (IP)**
- **max-metric router-lsa** [**on-startup** {*seconds* | **wait-for-bgp**}]
- **maximum-paths** *maximum maximum-paths* {[*number-of-paths*] [**import** *number-of-paths*] | [**import** *number-of-paths*]}
- **neighbor** *ip-address* [**cost** *number*]
- **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*as-number*] [**metric** {*metric-value* | **transparent**}] [*metric-type type-value*] [**match** {**external** | **internal** | **nssa-external**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
- **summary-address** {**ip-address** *mask* | **prefix** *mask*} [**not-advertise**] [**tag** *tag*]
- **timers throttle spf** *spf-start spf-hold spf-max-wait*
- **traffic-share min across-interfaces**

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vrf-name*]
4. **address-family ipv4** [**multicast** | **unicast**]
5. **topology** {**base** | *topology-name* **tid** *number*}
6. **end**
7. **show ip ospf** [*process-id*] **topology-info** [**multicast**] [**topology** {*topology-name* | **base**}]

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`router ospf process-id [vrf vrf-name]`
	
	Example:
	`Router(config)# router ospf 1`
		Enables an OSPF routing process and enters router configuration mode.

Step 4	`address-family ipv4 [multicast | unicast]`
	
	Example:
	`Router(config-router)# address-family ipv4`
		Enter router address family configuration mode to configure an OSPF address family session.
		• Currently, only the base topology can be configured under the multicast subaddress family.

	Command or Action	Purpose
Step 5	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(config-router-af)# topology VOICE tid 10`	Configures OSPF support for the topology and assigns a TID number for each topology. Enters router address family topology configuration mode. • Use the **tid** keyword and *number* argument to configure a topology ID. The topology ID must be configured in the first configuration of the specified topology. It is optional for subsequent configuration. **Note** The **base** keyword is accepted only for IPv4 multicast. The **tid** keyword is accepted only for IPv4 or IPv6 unicast.
Step 6	**end** **Example:** `Router(config-router-af-topology)# end`	Exits router address family topology configuration mode and returns to privileged EXEC mode.
Step 7	**show ip ospf** [*process-id*] **topology-info** [**multicast**] [**topology** {*topology-name* | **base**}] **Example:** `Router# show ip ospf topology-info topology VOICE`	(Optional) Displays OSPF information about the specified topology.

What to Do Next

If an EIGRP topology configuration is required, proceed to the next task. If an IS-IS topology configuration is required proceed to the "Activating an MTR Topology by Using IS-IS" section on page 26.

Activating an MTR Topology by Using EIGRP

Perform this task to configure EIGRP for an MTR topology. Only MTR commands are shown in this task.

Prerequisites

- Before using EIGRP to support MTR, you should be familiar with the concepts documented in the "Routing Protocol Support for MTR" section on page 7.
- A global topology configuration has been configured and activated.

Restrictions

- Graceful restart in EIGRP works only for base topologies. All other service topologies reset with new adjacencies.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **router eigrp** *name*
4. **address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **autonomous-system** *as-number*
5. **topology** {**base** | *topology-name* **tid** *number*}
6. **end**
7. **show ip protocols topology** *name* [**summary**]
8. **show ip eigrp topology** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *name* **Example:** `Router(config)# router eigrp MTR`	Configures an EIGRP process for MTR, and enters router configuration mode. • You can use the command without configuring MTR, but it defaults to the base topology.
Step 4	**address-family ipv4** [**unicast** | **multicast** | **vrf** *vrf-name*] **autonomous-system** *as-number* **Example:** `Router(config-router)# address-family ipv4 autonomous-system 1`	Enters router address family configuration mode to configure EIGRP for MTR.
Step 5	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(config-router-af)# topology VIDEO tid 100`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters router address family topology configuration mode. • Each topology must be configured with a unique topology ID. The topology ID must be entered each time this command is entered.
Step 6	**end** **Example:** `Router(config-router-af-topology)# end`	Exits router address family configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show ip protocols topology** *name* [**summary**] **Example:** `Router# show ip protocols topology VIDEO`	Displays the status of routing protocols configured in a topology. **Tip** This command can be entered to display the status, under a topology, of any configured routing protocol.
Step 8	**show ip eigrp topology** *name* **Example:** `Router# show ip eigrp topology VIDEO`	Displays the routing table of an EIGRP process configured under a topology.

What to Do Next

If an IS-IS topology configuration is required, proceed to the next task. If a BGP topology configuration is required, proceed to "Activating an MTR Topology by Using BGP" section on page 28.

Activating an MTR Topology by Using IS-IS

To configure MTR for IS-IS, you must perform two tasks. You must activate an MTR topology on an IS-IS router. You must also configure the MTR topology to globally configure all interfaces using the **all-interfaces** address family topology configuration command, or you must configure the IS-IS topology in interface configuration mode to configure only IS-IS interfaces. The order in which you perform the two tasks does not matter.

Perform this task to enable an MTR topology on an IS-IS router and enable support for IPv4 unicast and multicast address families. Only MTR commands are shown in this task.

Prerequisites

- Before using IS-IS to support MTR, you should be familiar with the concepts documented in the "Routing Protocol Support for MTR" section on page 7.
- A global topology configuration has been configured and activated.

Restrictions

- Only the IPv4 address family (multicast and unicast) and IPv6 address family unicast are supported. For information about configuring Multitopology IS-IS for IPv6, see the "Implementing IS-IS for IPv6" module in the *Cisco IOS IPv6 Configuration Guide.*

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*area-tag*]
4. **net** *network-entity-title*
5. **metric-style wide**

6. **address-family ipv4** [**multicast** | **unicast**]
7. **topology** *topology-name* **tid** *number*
8. **end**
9. **show isis neighbors detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*area-tag*] **Example:** `Router(config)# router isis`	Enables the IS-IS routing protocol and optionally specifies an IS-IS process. Enters router configuration mode.
Step 4	**net** *network-entity-title* **Example:** `Router(config-router)# net 31.3131.3131.3131.00`	Configures an IS-IS network entity title (NET) for a Connectionless Network Service (CLNS) routing process.
Step 5	**metric-style wide** [**transition**] [**level-1** | **level-2** | **level-1-2**] **Example:** `Router(config-router)# metric-style wide`	Globally changes the metric value for all IS-IS interfaces. **Note** Wide style metrics are required for prefix tagging.
Step 6	**address-family ipv4** [**multicast** | **unicast**] **Example:** `Router(config-router)# address-family ipv4`	Enters router address family configuration mode under IS-IS router configuration mode.
Step 7	**topology** *topology-name* **tid** *number* **Example:** `Router(config-router-af)# topology DATA tid 100`	Configures IS-IS support for the topology and assigns a TID number for each topology. • IS-IS support for the DATA topology is configured.

	Command or Action	Purpose
Step 8	**`end`** **Example:** `Router(config-router-topology)# end`	Exits router address family configuration mode and returns to privileged EXEC mode.
Step 9	**`show isis neighbors detail`** **Example:** `Router# show isis neighbors detail`	(Optional) Displays information about IS-IS neighbors, including MTR information for the TID values for the router and its IS-IS neighbors.

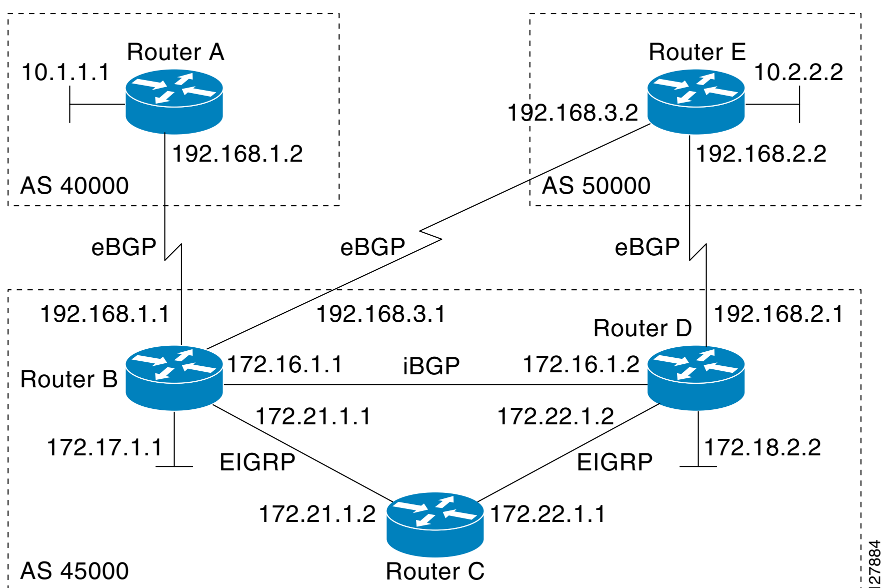
What to Do Next

If a BGP topology configuration is required, proceed to "Activating an MTR Topology by Using BGP" section on page 28.

Activating an MTR Topology by Using BGP

Perform this task to activate an MTR topology inside an address family by using BGP. This task is configured on Router B in Figure 8 and must also be configured on Router D and Router E. In this task, a scope hierarchy is configured to apply globally, and a neighbor is configured under router scope configuration mode. Under the IPv4 unicast address family, an MTR topology that applies to video traffic is activated for the specified neighbor. There is no interface configuration mode for BGP topologies.

Figure 8 BGP Network Diagram



Prerequisites

- Before using BGP to support MTR, you should be familiar with all the concepts documented in the "Information About BGP Support for MTR" section on page 2.
- A global MTR topology configuration has been configured and activated.

Restrictions

- Redistribution within a topology is permitted. Redistribution from one topology to another is not permitted. This restriction is designed to prevent routing loops. You can use topology translation or topology import functionality to move routes from one topology to another.
- Only a single multicast topology can be configured, and only the base topology can be specified if a multicast topology is created.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **scope** {**global** | **vrf** *vrf-name*}
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **transport** {**connection-mode** {**active** | **passive**} | **path-mtu-discovery** | **multi-session** | **single-session**}
7. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
8. **topology** {**base** | *topology-name*}
9. **bgp tid** *number*
10. **neighbor** {*ip-address*} **activate**
11. **neighbor** {*ip-address* | *peer-group-name*} **translate-topology** *number*
12. **end**
13. **clear ip bgp topology** {***** | *topology-name*} {*as-number* | **dampening** [*network-address* [*network-mask*]] | **flap-statistics** [*network-address* [*network-mask*]] | **peer-group** *peer-group-name* | **table-map** | **update-group** [*number* | *ip-address*]} [**in** [**prefix-filter**] | **out** | **soft** [**in** [**prefix-filter**] | **out**]]
14. **show ip bgp topology** {***** | *topology-name*} **summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 45000`	Enters router configuration mode to create or configure a BGP routing process.
Step 4	**scope** {**global** | **vrf** *vrf-name*} **Example:** `Router(config-router)# scope global`	Defines the scope to the BGP routing process and enters router scope configuration mode. • BGP general session commands that apply to a single network, or a specified VRF, are entered in this configuration mode. • Use the **global** keyword to specify that BGP uses the global routing table. • Use the **vrf** keyword and *vrf-name* argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system-number* **Example:** `Router(config-router-scope)# neighbor 172.16.1.2 remote-as 45000`	Adds the IP address of the neighbor in the specified autonomous system to the multiprotocol BGP neighbor table of the local router.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **transport** {**connection-mode** {**active** | **passive**} | **path-mtu-discovery** | **multi-session** | **single-session**} **Example:** `Router(config-router-scope)# neighbor 172.16.1.2 transport multi-session`	Enables a TCP transport session option for a BGP session. • Use the **connection-mode** keyword to specify the type of connection, either active or passive. • Use the **path-mtu-discovery** keyword to enable TCP transport path maximum transmission unit (MTU) discovery. • Use the **multi-session** keyword to specify a separate TCP transport session for each address family. • Use the **single-session** keyword to specify that all address families use a single TCP transport session.

	Command or Action	Purpose
Step 7	**address-family ipv4** [**mdt** | **multicast** | **unicast**] **Example:** `Router(config-router-scope)# address-family ipv4`	Specifies the IPv4 address family and enters router scope address family configuration mode. • Use the **mdt** keyword to specify IPv4 MDT address prefixes. • Use the **multicast** keyword to specify IPv4 multicast address prefixes. • Use the **unicast** keyword to specify the IPv4 unicast address family. By default, the router is placed in address family configuration mode for the IPv4 unicast address family if the **unicast** keyword is not specified with the **address-family ipv4** command. • Non-topology-specific configuration parameters are configured in this configuration mode.
Step 8	**topology** {**base** | *topology-name*} **Example:** `Router(config-router-scope-af)# topology VIDEO`	Configures the topology instance in which BGP routes class-specific or base topology traffic, and enters router scope address family topology configuration mode.
Step 9	**bgp tid** *number* **Example:** `Router(config-router-scope-af-topo)# bgp tid 100`	Associates a BGP routing process with the specified topology ID. • Each topology must be configured with a unique topology ID.
Step 10	**neighbor** *ip-address* **activate** **Example:** `Router(config-router-scope-af-topo)# neighbor 172.16.1.2 activate`	Enables the BGP neighbor to exchange prefixes for the NSAP address family with the local router. **Note** If you have configured a peer group as a BGP neighbor, do not use this command because peer groups are automatically activated when any peer group parameter is configured.
Step 11	**neighbor** {*ip-address* | *peer-group-name*} **translate-topology** *number* **Example:** `Router(config-router-scope-af-topo)# neighbor 172.16.1.2 translate-topology 200`	(Optional) Configures BGP to install routes from a topology on another router to a topology on the local router. • The topology ID is entered for the *number* argument to identify the topology on the router.
Step 12	**end** **Example:** `Router(config-router-scope-af-topo)# end`	(Optional) Exits router scope address family topology configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 13	**clear ip bgp topology** {***** | *topology-name*} {*as-number* | **dampening** [*network-address* [*network-mask*]] | **flap-statistics** [*network-address* [*network-mask*]] | **peer-group** *peer-group-name* | **table-map** | **update-group** [*number* | *ip-address*]} [**in** [**prefix-filter**] | **out** | **soft** [**in** [**prefix-filter**] | **out**]] **Example:** `Router# clear ip bgp topology VIDEO 45000`	Resets BGP neighbor sessions under a specified topology or all topologies.
Step 14	**show ip bgp topology** {***** | *topology*} **summary** **Example:** `Router# show ip bgp topology VIDEO summary`	(Optional) Displays BGP information about a topology. • Most standard BGP keywords and arguments can be entered following the **topology** keyword. **Note** Only the syntax required for this task is shown. For more details, see the *Cisco IOS IP Routing: BGP Command Reference*.

What to Do Next

Repeat this task for every topology that you want to enable, and repeat this configuration on all neighbor routers that are to use the topologies.

If you want to import routes from one MTR topology to another on the same router, proceed to"Importing Routes from an MTR Topology by Using BGP" section on page 32.

Importing Routes from an MTR Topology by Using BGP

Perform this task to import routes from one MTR topology to another on the same router, when multiple topologies are configured on the same router. In this task, a prefix list is defined to permit prefixes from the 10.2.2.0 network, and this prefix list is used with a route map to filter routes moved from the imported topology. A global scope is configured, address family IPv4 is entered, the VIDEO topology is specified, the VOICE topology is imported, and the routes are filtered using the route map named 10NET.

Prerequisites

- A global topology configuration has been configured and activated.

Restrictions

- Redistribution within a topology is permitted. Redistribution from one topology to another is not permitted. This restriction is designed to prevent routing loops from occurring. You can use topology translation or topology import functionality to move routes from one topology to another.
- Only a single multicast topology can be configured, and only the base topology can be specified if a multicast topology is created.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**ge** *ge-value*] [**le** *le-value*]
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...* | *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]}
6. **exit**
7. **router bgp** *autonomous-system-number*
8. **scope** {**global** | **vrf** *vrf-name*}
9. **address-family ipv4** [**mdt** | **multicast** | **unicast**]
10. **topology** {**base** | *topology-name*}
11. **import topology** {**base** | *topology-name*} [**route-map** *map-name*]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network***/***length* | **permit** *network***/***length*} [**ge** *ge-value*] [**le** *le-value*] **Example:** `Router(config)# ip prefix-list TEN permit 10.2.2.0/24`	Configures an IP prefix list. • In this example, prefix list TEN permits advertising of the 10.2.2.0/24 prefix depending on a match set by the **match ip address** command.
Step 4	**route-map** *map-name* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map 10NET`	Creates a route map and enters route map configuration mode. • In this example, the route map named 10NET is created.
Step 5	**match ip address** {*access-list-number* [*access-list-number...* | *access-list-name...*] | *access-list-name* [*access-list-number...*| *access-list-name*] | **prefix-list** *prefix-list-name* [*prefix-list-name...*]} **Example:** `Router(config-route-map)# match ip address prefix-list TEN`	Configures the route map to match a prefix that is permitted by a standard access list, an extended access list, or a prefix list. • In this example, the route map is configured to match prefixes permitted by prefix list TEN.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-route-map)# exit`	Exits route map configuration mode and returns to global configuration mode.
Step 7	**router bgp** *autonomous-system-number* **Example:** `Router(config)# router bgp 50000`	Enters router configuration mode to create or configure a BGP routing process.
Step 8	**scope** {**global** | **vrf** *vrf-name*} **Example:** `Router(config-router)# scope global`	Defines the scope to the BGP routing process and enters router scope configuration mode. • BGP general session commands that apply to a single network, or a specified VRF, are entered in this configuration mode. • Use the **global** keyword to specify that BGP uses the global routing table. • Use the **vrf** keyword and *vrf-name* argument to specify that BGP uses a specific VRF routing table. The VRF must already exist.
Step 9	**address-family ipv4** [**mdt** | **multicast** | **unicast**] **Example:** `Router(config-router-scope)# address-family ipv4`	Enters router scope address family configuration mode to configure an address family session under BGP. • Non-topology-specific configuration parameters are configured in this configuration mode.
Step 10	**topology** {**base** | *topology-name*} **Example:** `Router(config-router-scope-af)# topology VIDEO`	Configures the topology instance in which BGP routes class-specific or base topology traffic, and enters router scope address family topology configuration mode.
Step 11	**import topology** {**base** | *topology-name*} [**route-ma**p *map-name*] **Example:** `Router(config-router-scope-af-topo)# import topology VOICE route-map 10NET`	(Optional) Configures BGP to move routes from one topology to another on the same router. • The **route-map** keyword can be used to filter routes that moved between topologies.
Step 12	**end** **Example:** `Router(config-router-scope-af-topo)# end`	(Optional) Exits router scope address family topology configuration mode and returns to privileged EXEC mode.

Configuring an MTR Topology in Interface Configuration Mode

Perform this task to configure an MTR topology in interface configuration mode. The configuration of an MTR topology in interface configuration mode allows you to enable or disable MTR on a per-interface basis. By default, a class-specific topology does not include any interfaces.

Prerequisites

A topology must be defined globally before per-interface topology configuration can be configured.

Restrictions

Interfaces cannot be excluded from the base topology by design. However, IGP can be excluded from an interface in a base topology configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable**] | **base**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable**] | **base**} **Example:** `Router(config-if)# topology ipv4 VOICE`	Enters interface topology configuration mode to configure an MTR topology instance on an interface. • Use the **disable** keyword to disable the topology instance on the interface. This form is used to exclude a topology configuration from an interface. • If the **no** form of this command is used, the topology interface configuration is removed. • If the **no** form of this command is used with the **disable** keyword, the topology instance is enabled on the interface.
Step 5	**end** **Example:** `Router(config-if-topology)# end`	Exits interface topology configuration mode and returns to privileged EXEC mode.

Activating an MTR Topology in Interface Configuration Mode by Using OSPF

Perform this task to configure OSPF features used in MTR in interface configuration mode. Configuring a topology in interface configuration mode allows you to enable or disable MTR on per-interface basis. By default, a class-specific topology does not include any interfaces.

Prerequisites

A topology must be defined globally before per-interface topology configuration can be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable**] | **base**}
5. **ip ospf cost** *number*
6. **ip ospf topology disable**
7. **end**
8. **show ip ospf** [*process-id*] **interface** [*interface-type interface-number*] [**brief**] [**multicast**] [**topology** {*topology-name* | **base**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable**] | **base**} **Example:** `Router(config-if)# topology ipv4 VOICE`	Enters interface topology configuration mode to configure MTR. **Note** Entering this command with the **disable** keyword disables the topology instance on the interface. This form is used to exclude a topology configuration from an interface.

	Command or Action	Purpose
Step 5	**ip ospf cost** *number* **Example:** `Router(config-if-topology)# ip ospf cost 100`	Applies a cost to the interface in a topology instance. • The lowest cost number has the highest preference.
Step 6	**ip ospf topology disable** **Example:** `Router(config-if-topology)# ip ospf topology disable`	Prevents OSPF from advertising the interface as part of the topology without disabling the OSPF process or the topology on the interface.
Step 7	**end** **Example:** `Router(config-if-topology)# end`	Exits interface topology configuration mode and returns to privileged EXEC mode.
Step 8	**show ip ospf** [*process-id*] **interface** [*interface-type interface-number*] [**brief**] [**multicast**] [**topology** {*topology-name* | **base**}] **Example:** `Router# show ip ospf 1 interface topology VOICE`	(Optional) Displays OSPF-related interface information. • Displays OSPF and interface information about the specified topology when the **topology** keyword is entered.

Activating an MTR Topology in Interface Configuration Mode by Using EIGRP

Perform this task to configure EIGRP features used in MTR in interface configuration mode. Configuring a topology in interface configuration mode allows you enable or disable MTR on per-interface basis. By default, a class-specific topology does not include any interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable**] | **base**}
5. **eigrp** *as-number* **delay** *value*
6. **eigrp** *as-number* **next-hop-self**
7. **eigrp** *as-number* **shutdown**
8. **eigrp** *as-number* **split-horizon**
9. **eigrp** *as-number* **summary-address** *ip-address wildcard-mask* [**distance**]
10. **end**
11. **show ip eigrp topology** *name* **interfaces**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable**] | **base**} **Example:** `Router(config-if)# topology ipv4 VOICE`	Configures an MTR topology instance on an interface and enters interface topology configuration mode. **Note** Entering this command with the **disable** keyword disables the topology instance on the interface. This form is used to exclude a topology configuration from an interface.
Step 5	**eigrp** *as-number* **delay** *value* **Example:** `Router(config-if-topology)# eigrp 1 delay 100000`	Configures the delay value that EIGRP uses for interface metric calculation. • The *value* argument is entered in microseconds. The example configures an interface delay metric of 100 milliseconds.
Step 6	**eigrp** *as-number* **next-hop-self** **Example:** `Router(config-if-topology)# eigrp 1 next-hop-self`	Configures an EIGRP process to advertise itself as the next hop. • This command is enabled by default.
Step 7	**eigrp** *as-number* **shutdown** **Example:** `Router(config-if-topology)# eigrp 1 shutdown`	Disables an EIGRP process on the interface without disabling the global topology configuration on the interface.
Step 8	**eigrp** *as-number* **split-horizon** **Example:** `Router(config-if-topology)# eigrp 1 split-horizon`	Configures an EIGRP process to use split horizon. • This command is enabled by default.
Step 9	**eigrp** *as-number* **summary-address** *ip-address wildcard-mask* [**distance**] **Example:** `Router(config-if-topology)# eigrp 1 summary-address 10.1.1.0 0.0.0.255`	Configures an EIGRP summary address. • An administrative distance of 5 is applied to EIGRP summary routes if the distance is not specified.

	Command or Action	Purpose
Step 10	**end** **Example:** `Router(config-if-topology)# end`	Exits interface topology configuration mode and returns to privileged EXEC mode.
Step 11	**show ip eigrp topology** *name* **interfaces** **Example:** `Router# show ip eigrp topology VOICE interfaces`	Displays information about interfaces, on which EIGRP is configured, in a topology.

Activating an MTR Topology in Interface Configuration Mode by Using IS-IS

Perform this task to configure IS-IS features used in MTR in interface configuration mode. Configuring a topology in interface configuration mode allows you to enable or disable MTR on per-interface basis. By default, a class-specific topology does not include any interfaces.

Prerequisites

A topology must be defined globally before per-interface topology configuration can be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **ip router isis** *area-tag*
6. **topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]}
7. **isis topology disable**
8. **topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]}
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface Ethernet 2/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.7.17 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 5	**ip router isis** [*area-tag*] **Example:** `Router(config-if)# ip router isis`	Configures an IS-IS routing process for IP on an interface and attaches an area designator to the routing process. **Note** If a tag is not specified, a null tag is assumed and the process is referenced with a null tag.
Step 6	**topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]} **Example:** `Router(config-if)# topology ipv4 DATA`	Configures an MTR topology instance on an interface and enters interface topology configuration mode. **Note** In this example, the topology instance DATA is configured for an MTR network that has a global topology named DATA.
Step 7	**isis topology disable** **Example:** `Router(config-if-topology)# isis topology disable`	(Optional) Prevents an IS-IS process from advertising the interface as part of the topology. **Note** In this example, the topology instance DATA will not advertise the interface as part of the topology.
Step 8	**topology ipv4** [**multicast** | **unicast**] {*topology-name* [**disable** | **base**]} **Example:** `Router(config-if-topology)# topology ipv4 VOICE`	Configures an MTR topology instance on an interface. **Note** In this example, the topology instance VOICE is configured for an MTR network that has a global topology named "VOICE".
Step 9	**end** **Example:** `Router(config-if-topology)# end`	Exits interface topology configuration mode and returns to privileged EXEC mode.

Configuring SNMP Support for MTR

- Associating an SNMP Context with a VRF for MTR, page 40
- Associating an SNMP Context with a Data Topology for MTR, page 41
- Associating an SNMP Context with a Routing Protocol for MTR, page 43

Associating an SNMP Context with a VRF for MTR

Perform this task to associate an SNMP context with a VRF for MTR. The context string is passed on to the MIB access function during SNMP transactions.

Prerequisites

- SNMP must be enabled on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **snmp context** *context-name*
5. **end**
6. **show snmp context mapping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`ip vrf`** *`vrf-name`* **Example:** `Router(config)# ip vrf vrfA`	Defines a VRF instance and enters VRF configuration mode.
Step 4	**`snmp context`** *`context-name`* **Example:** `Router(config-vrf)# snmp context context-vrfA`	Creates an SNMP context for MTR for a specific VRF.
Step 5	**`end`** **Example:** `Router(config-af-topology)# end`	Exits VRF configuration mode and returns to privileged EXEC mode.
Step 6	**`show snmp context mapping`** **Example:** `Router# show snmp context mapping`	(Optional) Displays information about SNMP contexts for MTR.

Associating an SNMP Context with a Data Topology for MTR

Perform this task to associate an SNMP context with a data topology for MTR. The context string is passed on to the MIB access function during SNMP transactions.

Prerequisites

- SNMP must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **global-address-family ipv4** [**multicast** | **unicast**]
4. **topology** {**base** | *topology-name*}
5. **snmp context** *context-name*
6. **end**
7. **show snmp context mapping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**global-address-family ipv4** [**multicast** | **unicast**] **Example:** `Router(config)# global-address-family ipv4`	Enters global address family base topology configuration mode to configure the global topology. • The address family for the class-specific topology is specified in this step. The subaddress family can be optionally specified. Unicast is the default if no subaddress family is entered.
Step 4	**topology** {**base** | *topology-name*} **Example:** `Router(config-af)# topology VOICE`	Configures the global topology instance and enters routing topology configuration mode.
Step 5	**snmp context** *context-name* **Example:** `Router(config-af-topology)# snmp context comp-topol`	Creates an SNMP context for MTR for a specific topology.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-af-topology)# end`	Exits routing topology configuration mode and returns to privileged EXEC mode.
Step 7	**show snmp context mapping** **Example:** `Router# show snmp context mapping`	(Optional) Displays information about SNMP contexts for MTR.

Associating an SNMP Context with a Routing Protocol for MTR

Perform this task to associate an SNMP context with a routing protocol for MTR. The context string is passed on to the MIB access function during SNMP transactions.

Prerequisites

- SNMP must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router ospf** *process-id* [**vrf** *vrf-name*]
4. **snmp context** *context-name*
5. **address-family ipv4** [**multicast** | **unicast**]
6. **topology** {**base** | *topology-name* **tid** *number*}
7. **snmp context** *context-name*
8. **end**
9. **show snmp context mapping**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**router ospf** *process-id* [**vrf** *vrf-name*] **Example:** `Router(config)# router ospf 1`	Enables an OSPF routing process and enters router configuration mode. • You can configure support for multiple routing protocols.
Step 4	**snmp context** *context-name* **Example:** `Router(config-router)# snmp context comp-prot`	Creates an SNMP context for MTR for a specific topology under a routing protocol.
Step 5	**address-family ipv4** [**multicast** | **unicast**] **Example:** `Router(config-router)# address-family ipv4`	Enters global address family configuration mode to configure an OSPF address family session.
Step 6	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(config-router-af)# topology VOICE tid 10`	Configures the global topology instance and enters router address family topology configuration mode.
Step 7	**snmp context** *context-name* **Example:** `Router(config-router-af-topology)# snmp context comp-protocol`	Creates an SNMP context for MTR for a specific topology under a routing protocol.
Step 8	**end** **Example:** `Router(config-router-af-topology)# end`	Exits router address family topology configuration mode and returns to privileged EXEC mode.
Step 9	**show snmp context mapping** **Example:** `Router# show snmp context mapping`	(Optional) Displays information about SNMP contexts for MTR.

Enabling and Monitoring MTR Topology Statistics Accounting

- Enabling Topology Statistics Accounting for MTR, page 44
- Monitoring Interface and Topology IP Traffic Statistics for MTR, page 46

Enabling Topology Statistics Accounting for MTR

Perform this task to enable topology statistics accounting on all interfaces in the global address family for all IPv4 unicast topologies in the default VRF instance and to enable topology accounting for all IPv4 unicast topologies in the VRF instance associated with the specified interface.

Prerequisites

- CEF must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **global-address-family ipv4** [**multicast** | **unicast**]
4. **topology-accounting**
5. **exit**
6. **interface** *type number*
7. **ip topology-accounting**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**global-address-family ipv4** [**multicast** | **unicast**] **Example:** `Router(config)# global-address-family ipv4`	Enters global address family configuration mode.
Step 4	**topology accounting** **Example:** `Router(config-af)# topology accounting`	Enables topology accounting on all interfaces in the global address family for all IPv4 unicast topologies in the default VRF instance.
Step 5	**exit** **Example:** `Router(config-af)# exit`	Exits global address family configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 1/10`	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 7	**ip topology-accounting** **Example:** `Router(config-if)# ip topology-accounting`	Enables topology accounting for all IPv4 unicast topologies in the VPN VRF associated with the specified interface. • This topology accounting is supported only for the default VRF.
Step 8	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring Interface and Topology IP Traffic Statistics for MTR

Perform this task to monitor interface and topology IP traffic statistics for MTR.

SUMMARY STEPS

1. **enable**
2. **show ip interface** [*type number*] [**topology** {*name* | **all** | **base**}] [**stats**]
3. **show ip traffic** [**topology** {*name* | **all** | **base**}]
4. **clear ip interface** *type number* [**topology** {*name* | **all** | **base**}] [**stats**]
5. **clear ip traffic** [**topology** {*name* | **all** | **base**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip interface** [*type number*] [**topology** {*name* | **all** | **base**}] [**stats**] **Example:** `Router# show ip interface FastEthernet 1/10 stats`	(Optional) Displays IP traffic statistics for all interfaces or statistics related to the specified interface. • If you specify an interface type and number, you see information for that specific interface. If you specify no optional arguments, you see information for all the interfaces. • If the **topology** *name* keyword and argument are used, statistics are limited to the IP traffic for that specific topology. • The **base** keyword is reserved for IPv4 unicast base topology.
Step 3	**show ip traffic** [**topology** {*name* | **all** | **base**}] **Example:** `Router# show ip traffic topology VOICE`	(Optional) Displays global IP traffic statistics (an aggregation of all the topologies when MTR is enabled) or statistics related to a particular topology. • The **base** keyword is reserved for the IPv4 unicast base topology.

	Command or Action	Purpose
Step 4	**clear ip interface** *type number* [**topology** {*name* | **all** | **base**}] [**stats**] **Example:** `Router# clear ip interface FastEthernet 1/10 topology all`	(Optional) Resets interface-level IP traffic statistics. • If the **topology** keyword and a related keyword are not used, only the interface-level aggregate statistics are reset. • If all topologies need to be reset, use the **all** keyword as the topology name.
Step 5	**clear ip traffic** [**topology** {*name* | **all** | **base**}] **Example:** `Router# clear ip traffic topology all`	(Optional) Resets IP traffic statistics. • If no topology name is specified, global statistics are cleared.

Testing Network Connectivity for MTR

Perform this task to test network connectivity for MTR. You can configure a standard or extended ping by using the topology name in place of a hostname or an IP address. Traceroute has been similarly enhanced.

SUMMARY STEPS

1. **enable**
2. **ping** [**vrf** *vrf-name* | **topology** *topology-name*] *protocol* [*target-address*] [*source-address*]
3. **traceroute** [**vrf** *vrf-name* | **topology** *topology-name*] [*protocol*] *destination*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [**vrf** *vrf-name* | **topology** *topology-name*] *protocol* [*target-address*] [*source-address*] **Example:** `Router# ping topology VOICE`	Configures the router to transmit ping messages to the target host in a topology. • An extended ping is configured by entering this command with only the topology name.
Step 3	**traceroute** [**vrf** *vrf-name* | **topology** *topology-name*] [*protocol*] *destination* **Example:** `Router# traceroute VOICE`	Configures the router to trace the specified host in a topology. • An extended trace is configured by entering this command with only the topology name. • If the **vrf** *vrf-name* keyword and argument are used, the **topology** option is not displayed because only the default VRF is supported. The **topology** *topology-name* keyword and argument and the DSCP option in the extended traceroute system dialog are displayed only if there is a topology configured on the router.

Configuration Examples for Multi-Topology Routing

Examples: Unicast Topology for MTR

Example: Global Interface Configuration

The following example shows how to create a topology instance named VOICE. This topology is configured to use all operational interfaces on the router. Per the default forwarding rule (strict), only packets destined for routes in the VOICE topology RIB are forwarded. Packets that do not have a topology-specific forwarding entry are dropped.

```
global-address-family ipv4
 topology VOICE
 all-interfaces
 end
```

Example: Incremental Forwarding Configuration

The following example shows how to create a topology instance named VIDEO. This topology is configured to accept and install a maximum of 1000 routes in the VIDEO topology RIB. Incremental forwarding mode is configured so that the router forwards packets over the base topology if no forwarding entry is found in the class-specific RIB.

```
global-address-family ipv4
 topology VIDEO
 forward-base
```

```
 maximum routes 1000 90
 end
```

Example: Unicast Topology Verification

The output of the **show topology detail** command displays information about class-specific and base topologies. This information includes the address family, associated interfaces, interface and topology status, topology name, and associated VRF.

```
Router# show topology detail

Topology: base
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Associated interfaces:
    Ethernet0/0, operation state: UP
    Ethernet0/1, operation state: DOWN
    Ethernet0/2, operation state: DOWN
    Ethernet0/3, operation state: DOWN
    Loopback0, operation state: UP

Topology: VIDEO
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Topology fallback is enabled
  Topology maximum route limit 1000, warning limit 90% (900)
  Associated interfaces:

Topology: VOICE
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Topology is enabled on all interfaces
  Associated interfaces:
    Ethernet0/0, operation state: UP
    Ethernet0/1, operation state: DOWN
    Ethernet0/2, operation state: DOWN
    Ethernet0/3, operation state: DOWN
    Loopback0, operation state: UP

Topology: base
  Address-family: ipv4 multicast
  Associated VPN VRF is default
  Topology state is DOWN
  Route Replication Enabled:
    from unicast all
  Associated interfaces:
```

Examples: Multicast Topology for MTR

- Example: Route Replication Configuration, page 50
- Example: Using a Unicast RIB for Multicast RPF Configuration, page 50
- Example: Multicast Verification, page 50

Example: Route Replication Configuration

The following example shows how to enable multicast support for MTR and to configure a separate multicast topology:

```
ip multicast-routing
ip multicast rpf multitopology
!
global-address-family ipv4 multicast
 topology base
 end
```

The following example shows how to configure the multicast topology to replicate OSPF routes from the VOICE topology. The routes are filtered through the BLUE route map before they are installed in the multicast routing table.

```
ip multicast-routing
ip multicast rpf multitopology
!
access-list 1 permit 192.168.1.0 0.0.0.255
!
route-map BLUE
 match ip address 1
 exit
!
global-address-family ipv4 multicast
 topology base
 route-replicate from unicast topology VOICE ospf route-map BLUE
```

Example: Using a Unicast RIB for Multicast RPF Configuration

The following example shows how to configure the multicast topology to perform RPF calculations on routes in the VIDEO topology RIB to build multicast distribution trees:

```
ip multicast-routing
ip multicast rpf multitopology
!
global-address-family ipv4 multicast
 topology base
 use-topology unicast VIDEO
 end
```

Example: Multicast Verification

The following example shows that the multicast topology is configured to replicate routes from the RIB of the VOICE topology:

```
Router# show topology detail

Topology: base
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Associated interfaces:
    Ethernet0/0, operation state: UP
    Ethernet0/1, operation state: DOWN
    Ethernet0/2, operation state: DOWN
    Ethernet0/3, operation state: DOWN
    Loopback0, operation state: UP

Topology: VIDEO
```

```
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Topology fallback is enabled
  Topology maximum route limit 1000, warning limit 90% (900)
  Associated interfaces:

Topology: VOICE
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Topology is enabled on all interfaces
  Associated interfaces:
    Ethernet0/0, operation state: UP
    Ethernet0/1, operation state: DOWN
    Ethernet0/2, operation state: DOWN
    Ethernet0/3, operation state: DOWN
    Loopback0, operation state: UP

Topology: base
  Address-family: ipv4 multicast
  Associated VPN VRF is default
  Topology state is DOWN
  Multicast multi-topology mode is enabled.
  Route Replication Enabled:
    from unicast topology VOICE all route-map BLUE
  Associated interfaces:
```

Examples: MTR Traffic Classification

The following example shows how to configure classification and activate MTR for two topologies:

```
global-address-family ipv4
 topology VOICE
  all-interfaces
  exit
 topology VIDEO
  forward-base
  maximum routes 1000 90
  exit
 exit
class-map match-any VOICE-CLASS
 match ip dscp 9
 exit
class-map match-any VIDEO-CLASS
 match ip dscp af11
 exit
policy-map type class-routing ipv4 unicast MTR
 class VOICE-CLASS
  select-topology VOICE
  exit
 class VIDEO-CLASS
  select-topology VIDEO
  exit
 exit
global-address-family ipv4
 service-policy type class-routing MTR
end
```

The following example shows how to display detailed information about the VOICE and VIDEO topologies:

```
Router# show topology detail

Topology: base
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Associated interfaces:
    Ethernet0/0, operation state: UP
    Ethernet0/1, operation state: DOWN
    Ethernet0/2, operation state: DOWN
    Ethernet0/3, operation state: DOWN
    Loopback0, operation state: UP

Topology: VIDEO
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Topology fallback is enabled
  Topology maximum route limit 1000, warning limit 90% (900)
  Associated interfaces:

Topology: VOICE
  Address-family: ipv4
  Associated VPN VRF is default
  Topology state is UP
  Topology is enabled on all interfaces
  Associated interfaces:
    Ethernet0/0, operation state: UP
    Ethernet0/1, operation state: DOWN
    Ethernet0/2, operation state: DOWN
    Ethernet0/3, operation state: DOWN
    Loopback0, operation state: UP

Topology: base
  Address-family: ipv4 multicast
  Associated VPN VRF is default
  Topology state is DOWN
  Multicast multi-topology mode is enabled.
  Route Replication Enabled:
    from unicast topology VOICE all route-map BLUE
  Associated interfaces:
    Ethernet0/0, operation state: UP
    Ethernet0/1, operation state: DOWN
    Ethernet0/2, operation state: DOWN
    Ethernet0/3, operation state: DOWN
    Loopback0, operation state: UP
```

The following example shows how to display the classification values for the VOICE and VIDEO topologies:

```
Router# show mtm table

MTM Table for VRF: default, ID:0

Topology                Address Family   Associated VRF         Topo-ID

base                    ipv4              default                 0

VOICE                    ipv4             default                 2051
Classifier: ClassID:3
DSCP: cs1
DSCP: 9

VIDEO                    ipv4             default                 2054
```

```
Classifier: ClassID:4
DSCP: af11
```

Examples: Activating an MTR Topology by Using OSPF

The following example shows how to configure the VOICE topology in an OSPF routing process and set the priority of the VOICE topology to the highest priority:

```
router ospf 1
 address-family ipv4
  topology VOICE tid 10
  priority 127
  end
```

In the following example, the **show ip ospf** command is used with the **topology-info** and **topology** keywords to display OSPF information about the topology named VOICE.

```
Router# show ip ospf 1 topology-info topology VOICE

OSPF Router with ID (10.0.0.1) (Process ID 1)

VOICE Topology (MTID 66)

Topology priority is 64
Redistributing External Routes from,
isis
Number of areas transit capable is 0
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFs 10000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Area BACKBONE(0) (Inactive)
SPF algorithm last executed 16:45:18.984 ago
SPF algorithm executed 3 times
Area ranges are
Area 1
SPF algorithm last executed 00:00:21.584 ago
SPF algorithm executed 1 times
Area ranges are
```

Examples: Activating an MTR Topology by Using EIGRP

The following example shows how to activate the VIDEO topology using EIGRP:

```
router eigrp MTR
 address-family ipv4 autonomous-system 1
  network 10.0.0.0 0.0.0.255
  topology VIDEO tid 10
   redistribute connected
   end
```

The following example shows how to display the status of routing protocols configured in the VIDEO topology. EIGRP information is shown in the output.

```
Router# show ip protocols topology VIDEO

*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
```

```
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  EIGRP graceful-restart disabled
  EIGRP NSF-aware route hold timer is 240s
  Topologies : 100(VOICE) 0(base)

  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

The following example shows the EIGRP routing table configured under the VIDEO topology:

```
Router# show ip eigrp topology VIDEO

EIGRP-IPv4 Topology Table for AS(1)/ID(10.1.1.2) Routing Table: VOICE

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.1.0/24, 1 successors, FD is 281600
        via Connected, Ethernet0/0
```

Examples: Activating an MTR Topology by Using IS-IS

The following example shows how to configure both the MTR topologies DATA and VIDEO and IS-IS support for MTR. The DATA and VIDEO topologies are enabled on three IS-IS neighbors in a network.

Router1

```
global-address-family ipv4
 topology DATA
 topology VOICE
 end

interface Ethernet 0/0
 ip address 192.168.128.2 255.255.255.0
 ip router isis
 topology ipv4 DATA
 isis topology disable
 topology ipv4 VOICE
 end

router isis
 net 33.3333.3333.3333.00
 metric-style wide
 address-family ipv4
  topology DATA tid 100
  topology VOICE tid 200
  end
```

Router2

```
global-address-family ipv4
 topology DATA
 topology VOICE
```

```
 all-interfaces
  forward-base
  maximum routes 1000 warning-only
  shutdown
  end

interface Ethernet 0/0
 ip address 192.168.128.1 255.255.255.0
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end

interface Ethernet 1/0
 ip address 192.168.130.1 255.255.255.0
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end

router isis
 net 32.3232.3232.3232.00
 metric-style wide
 address-family ipv4
  topology DATA tid 100
  topology VOICE tid 200
  end
```

Router 3

```
global-address-family ipv4
 topology DATA
  topology VOICE
  all-interfaces
  forward-base
  maximum routes 1000 warning-only
  shutdown
  end

interface Ethernet 1/0
 ip address 192.168.131.1 255.255.255.0
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end

router isis
 net 31.3131.3131.3131.00
 metric-style wide
 address-family ipv4
  topology DATA tid 100
  topology VOICE tid 200
  end
```

Entering the **show isis neighbors detail** command verifies topology translation with the IS-IS neighbor Router1:

```
Router# show isis neighbors detail

System Id      Type Interface IP Address      State Holdtime Circuit Id
R1             L2   Et0/0     192.168.128.2   UP    28       R5.01
```

```
  Area Address(es): 33
  SNPA: aabb.cc00.1f00
  State Changed: 00:07:05
  LAN Priority: 64
  Format: Phase V
  Remote TID:  100, 200
  Local TID:   100, 200
```

Examples: Activating an MTR Topology by Using BGP

Example: BGP Topology Translation Configuration

The following example shows how to configure BGP in the VIDEO topology and how to configure topology translation with the 192.168.2.2 neighbor:

```
router bgp 45000
 scope global
  neighbor 172.16.1.1 remote-as 50000
  neighbor 192.168.2.2 remote-as 55000
  neighbor 172.16.1.1 transport multi-session
  neighbor 192.168.2.2 transport multi-session
   address-family ipv4
    topology VIDEO
     bgp tid 100
     neighbor 172.16.1.1 activate
     neighbor 192.168.2.2 activate
     neighbor 192.168.2.2 translate-topology 200
     end
clear ip bgp topology VIDEO 50000
```

Example: BGP Scope Global and VRF Configuration

The following example shows how to configure a global scope for a unicast topology and also for a multicast topology. After exiting the router scope configuration mode, a scope is configured for the VRF named DATA.

```
router bgp 45000
 scope global
  bgp default ipv4-unicast
  neighbor 172.16.1.2 remote-as 45000
  neighbor 192.168.3.2 remote-as 50000
  address-family ipv4 unicast
   topology VOICE
   bgp tid 100
   neighbor 172.16.1.2 activate
   exit
  address-family ipv4 multicast
   topology base
    neighbor 192.168.3.2 activate
    exit
   exit
  exit
 scope vrf DATA
```

```
  neighbor 192.168.1.2 remote-as 40000
  address-family ipv4
   neighbor 192.168.1.2 activate
   end
```

Example: BGP Topology Verification

The following example shows summary output for the **show ip bgp topology** command. Information is displayed about BGP neighbors configured to use the MTR topology named VIDEO.

```
Router# show ip bgp topology VIDEO summary

BGP router identifier 192.168.3.1, local AS number 45000
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.1.2      4 45000     289     289        1    0    0 04:48:44        0
192.168.3.2     4 50000       3       3        1    0    0 00:00:27        0
```

The following partial output displays BGP neighbor information under the VIDEO topology:

```
Router# show ip bgp topology VIDEO neighbors 172.16.1.2

BGP neighbor is 172.16.1.2,  remote AS 45000, internal link
  BGP version 4, remote router ID 192.168.2.1
  BGP state = Established, up for 04:56:30
  Last read 00:00:23, last write 00:00:21, hold time is 180, keepalive interval is 60
seconds
  Neighbor sessions:
    1 active, is multisession capable
  Neighbor capabilities:
    Route refresh: advertised and received(new)
  Message statistics, state Established:
    InQ depth is 0
    OutQ depth is 0
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                0          0
    Keepalives:           296        296
    Route Refresh:          0          0
    Total:                297        297
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast topology VIDEO
  Session: 172.16.1.2 session 1
  BGP table version 1, neighbor version 1/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
1 update-group member
  Topology identifier: 100
.
.
.
  Address tracking is enabled, the RIB does have a route to 172.16.1.2
  Address tracking requires at least a /24 route to the peer
  Connections established 1; dropped 0
  Last reset never
  Transport(tcp) path-mtu-discovery is enabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Minimum incoming TTL 0, Outgoing TTL 255
Local host: 172.16.1.1, Local port: 11113
Foreign host: 172.16.1.2, Foreign port: 179
```

```
.
.
.
```

Example: Importing Routes from an MTR Topology by Using BGP

The following example shows how to configure an access list to be used by a route map named BLUE to filter routes imported from the MTR topology named VOICE. Only routes with the prefix 192.168.1.0 are imported.

```
access-list 1 permit 192.168.1.0 0.0.0.255
route-map BLUE
 match ip address 1
 exit
router bgp 50000
 scope global
  neighbor 10.1.1.2 remote-as 50000
  neighbor 172.16.1.1 remote-as 60000
   address-family ipv4
    topology VIDEO
     bgp tid 100
     neighbor 10.1.1.2 activate
     neighbor 172.16.1.1 activate
     import topology VOICE route-map BLUE
     end
clear ip bgp topology VIDEO 50000
```

Examples: MTR Topology in Interface Configuration Mode

The following example shows how to disable the VOICE topology on Ethernet interface 0/0.

```
interface Ethernet 0/0
 topology ipv4 VOICE disable
```

Examples: MTR OSPF Topology in Interface Configuration Mode

The following example shows how to disable OSPF routing on interface Ethernet 0/0 without removing the interface from the global topology configuration:

```
interface Ethernet 0/0
 topology ipv4 VOICE
  ip ospf cost 100
  ip ospf topology disable
  end
```

In the following example, the **show ip ospf interface** command is used with the **topology** keyword to display information about the topologies configured for OSPF in interface configuration mode.

```
Router# show ip ospf 1 interface topology VOICE

VOICE Topology (MTID 66)

Serial3/0 is up, line protocol is up
   Internet Address 10.0.0.5/30, Area 1
   Process ID 1, Router ID 44.44.44.44, Network Type POINT_TO_POINT
   Topology-MTID    Cost    Disabled    Shutdown      Topology Name
         4           77        no          no            grc
   Transmit Delay is 1 sec, State POINT_TO_POINT
```

```
   Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
     oob-resync timeout 40
     Hello due in 00:00:05
   Supports Link-local Signaling (LLS)
   Cisco NSF helper support enabled
   IETF NSF helper support enabled
   Index 1/4, flood queue length 0
   Next 0x0(0)/0x0(0)
   Last flood scan length is 1, maximum is 1
   Last flood scan time is 0 msec, maximum is 0 msec
   Neighbor Count is 1, Adjacent neighbor count is 1
     Adjacent with neighbor 10.2.2.2
   Suppress hello for 0 neighbor(s)
```

In the following example, the **show ip ospf interface** command is used with the **brief** and **topology** keywords to display information about the topologies configured for OSPF in interface configuration mode.

```
Router# show ip ospf 1 interface brief topology VOICE

VOICE Topology (MTID 66)

Interface     PID   Area     IP Address/Mask    Cost    State    Nbrs F/C
Se3/0         1     1        10.0.0.5/30        1       UP       0/0
Se2/0         1     1        10.0.0.1/30        1       UP       0/0
```

Examples: MTR EIGRP Topology in Interface Configuration Mode

The following example shows how to set the EIGRP delay calculation on interface Ethernet 0/0 to 100 milliseconds:

```
interface Ethernet 0/0
 topology ipv4 VOICE
  eigrp 1 delay 100000
  eigrp 1 next-hop-self
  eigrp 1 shutdown
  eigrp 1 split-horizon
  eigrp 1 summary-address 10.1.1.0 0.0.0.255
  end
```

The following example shows how to display EIGRP information about interfaces in the VOICE topology:

```
Router# show ip eigrp topology VOICE interfaces

EIGRP-IPv4 interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Et0/0              1        0/0        20       0/2            0           0
```

The following example shows how to display EIGRP information about links in the VOICE topology:

```
Router# show ip eigrp topology VOICE detail-links

EIGRP-IPv4 Topology Table for AS(1)/ID(10.1.1.1) Routing Table: VOICE

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.1.0/24, 1 successors, FD is 25856000, serno 5
        via Connected, Ethernet0/0
```

Examples: MTR IS-IS Topology in Interface Configuration Mode

The following example shows how to prevent the IS-IS process from advertising interface Ethernet 1/0 as part of the DATA topology:

```
interface Ethernet 1/0
 ip address 192.168.130.1 255.255.255.0
 ip router isis
 topology ipv4 DATA
  isis topology disable
  topology ipv4 VOICE
  end
```

Examples: SNMP Support for MTR

In the following example, the context string "context-vrfA" is configured to be associated with vrfA and will be passed on to the MIB access function during SNMP transactions:

```
snmp-server community public
ip vrf vrfA
 snmp context context-vrfA
 exit
```

In the following example, the context string "context-voice" is configured to be associated with the data topology named voice and will be passed on to the MIB access function during SNMP transactions:

```
global-address-family ipv4
 topology voice
  snmp context context-voice
  exit
```

In the following example, the context strings "context-ospf" and "context-voice" are configured to be associated with the OSPF process and topology named voice and will be passed on to the MIB access function during SNMP transactions:

```
router ospf 3
 snmp context context-ospf
 address-family ipv4
 topology voice tid 10
  snmp context ospf-voice
  end
```

The following example shows how the context strings are mapped to the specified VRF, address family, topology, or protocol instance:

```
Router# show snmp context mapping

Context: ospf-voice
  VRF Name:
  Address Family Name: ipv4
  Topology Name: voice
  Protocol Instance: OSPF-3 Router

Context: context-ospf
  VRF Name:
  Address Family Name:
  Topology Name:
  Protocol Instance: OSPF-3 Router

Context: context-vrfA
  VRF Name: vrfA
```

```
  Address Family Name:
  Topology Name:
  Protocol Instance:

Context: context-voice
  VRF Name:
  Address Family Name: ipv4
  Topology Name: voice
  Protocol Instance:
```

Examples: Monitoring Interface and Topology IP Traffic Statistics

In the following example, the **show ip interface** command is used with the *type number* arguments to display IP traffic statistics for the Fast Ethernet interface 1/10:

```
Router# show ip interface FastEthernet 1/10 stats

FastEthernet1/10
    5 minutes input rate 0 bits/sec, 0 packet/sec,
    5 minutes output rate 0 bits/sec, 0 packet/sec,
    201 packets input, 16038 bytes
    588 packets output, 25976 bytes
```

In this example, the **show ip traffic** command is used with the **topology** *instance* keyword and argument to display statistics related to a particular topology:

```
Router# show ip traffic topology VOICE

  Topology: VOICE
  5 minute input rate 0 bits/sec, 0 packet/sec,
  5 minute output rate 0 bits/sec, 0 packet/sec,
  100 packets input, 6038 bytes,
  88 packets output, 5976 bytes.
```

Examples: Testing Network Connectivity for MTR

The following example shows how to send a ping to the 10.1.1.2 neighbor in the VOICE topology:

```
Router# ping topology VOICE 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

The following example shows how to trace the 10.1.1.4 host in the VOICE topology:

```
Router# traceroute VOICE ip 10.1.1.4

Type escape sequence to abort.
Tracing the route to 10.1.1.4

  1 10.1.1.2 4 msec *  0 msec
  2 10.1.1.3 4 msec *  2 msec
  3 10.1.1.4 4 msec *  4 msec
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
MTR commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Multi-Topology Routing Command Reference*
IP routing protocol commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Routing: BGP Command Reference* *Cisco IOS IP Routing: EIGRP Command Reference* *Cisco IOS IP Routing: ISIS Command Reference* *Cisco IOS IP Routing: OSPF Command Reference*
IP multicast commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Multicast Command Reference*
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS IP Quality of Service Solutions Command Reference*
IP routing protocols concepts and tasks	*Cisco IOS IP Routing: BGP Configuration Guide* *Cisco IOS IP Routing: EIGRP Configuration Guide* *Cisco IOS IP Routing: ISIS Configuration Guide* *Cisco IOS IP Routing: OSPF Configuration Guide*
IP multicast concepts and tasks	*Cisco IOS IP Multicast Configuration Guide*
QoS concepts and tasks	*Cisco IOS Quality of Service Solutions Configuration Guide*
Configuring Multitopology IS-IS for IPv6	"Implementing IS-IS for IPv6" module in the *Cisco IOS IPv6 Configuration Guide*
Cisco IOS In Service Software Upgrade Process	*Cisco IOS In Service Software Upgrade Process* module

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multi-Topology Routing

Table 1 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 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.

Table 1 Feature Information for Multi-Topology Routing

Feature Name	Releases	Feature Information
Multi-Topology Routing	12.2(33)SRB 15.0(1)S	MTR introduces the capability to configure service differentiation through class-based forwarding. MTR provides multiple logical topologies over a single physical network. Service differentiation can be achieved by forwarding different traffic types over different logical topologies that could take different paths to the same destination. MTR can be used, for example, to define separate topologies for voice, video, and data traffic classes. The following commands were introduced or modified: **all-interfaces**, **clear ip interface**, **clear ip route topology**, **clear ip traffic**, **debug topology**, **exit-global-af**, **exit-if-topology**, **exit-topo**, **forward-base**, **global-address-family ipv4**, **ip route topology**, **ip topology accounting**, **maximum routes**, **ping**, **route replicate**, **show ip interface**, **show ip protocols topology**, **show ip route topology**, **show ip static route**, **show ip static route summary**, **show ip traffic**, **show topology**, **shutdown**, **topology**, **topology accounting**, **traceroute**.
BGP Support for MTR	12.2(33)SRB 15.0(1)S	This feature provides BGP support for multiple logical topologies over a single physical network. The following sections provide information about this feature: • BGP Routing Protocol Support for MTR, page 8 • Activating an MTR Topology by Using BGP, page 28 • Importing Routes from an MTR Topology by Using BGP, page 32 • Examples: Activating an MTR Topology by Using BGP, page 56 • Example: Importing Routes from an MTR Topology by Using BGP, page 58 The following commands were introduced or modified: **address-family ipv4**, **bgp tid**, **clear ip bgp topology**, **import topology**, **neighbor translate-topology**, **neighbor transport**, **show ip bgp topology**, **scope**, **topology**,

Table 1 Feature Information for Multi-Topology Routing (continued)

Feature Name	Releases	Feature Information
EIGRP Support for MTR	12.2(33)SRB 15.0(1)S	This feature provides EIGRP support for multiple logical topologies over a single physical network. The following sections provide information about this feature: • Routing Protocol Support for MTR, page 7 • Activating an MTR Topology by Using EIGRP, page 24 • Activating an MTR Topology in Interface Configuration Mode by Using EIGRP, page 37 • Examples: Activating an MTR Topology by Using EIGRP, page 53 • Examples: MTR EIGRP Topology in Interface Configuration Mode, page 59 The following commands were introduced or modified: **address-family ipv4**, **eigrp delay**, **clear ip eigrp neighbor**, **eigrp next-hop-self**, **eigrp shutdown**, **eigrp split-horizon**, **eigrp summary-address**, **router eigrp**, **show ip eigrp topology**, **topology**.
IS-IS Support for MTR	12.2(33)SRB	This feature provides IS-IS support for multiple logical topologies over a single physical network. The following sections provide information about this feature: • Routing Protocol Support for MTR, page 7 • Activating an MTR Topology by Using IS-IS, page 26 • Activating an MTR Topology in Interface Configuration Mode by Using IS-IS, page 39 • Examples: Activating an MTR Topology by Using IS-IS, page 54 • Examples: MTR IS-IS Topology in Interface Configuration Mode, page 60 The following commands were introduced or modified: **address-family ipv4**, **isis topology disable**, **show isis neighbors**, **topology**,
ISSU—MTR	12.2(33)SRB1	All protocols and applications that support MTR and also support ISSU have extended their ISSU support to include the MTR functionality. The following section provides information about this feature: • ISSU—MTR, page 11 No commands were introduced or modified in this feature.

Table 1 Feature Information for Multi-Topology Routing (continued)

Feature Name	Releases	Feature Information
MTR Support for Multicast	12.2(33)SRB 15.0(1)M	This feature provides MTR support for multicast and allows the user to control the path of multicast traffic in the network. The following sections provide information about this feature: • Multicast Topology Support for MTR, page 6 • Configuring a Multicast Topology for MTR, page 16 • Examples: Multicast Topology for MTR, page 49 The following commands were introduced or modified: **clear ip route multicast**, **ip multicast rpf multitopology**, **show ip route multicast**, **use-topology**.
OSPF Support for MTR	12.2(33)SRB	This feature provides OSPF support for multiple logical topologies over a single physical network. The following sections provide information about this feature: • Routing Protocol Support for MTR, page 7 • Activating an MTR Topology by Using OSPF, page 22 • Activating an MTR Topology in Interface Configuration Mode by Using OSPF, page 36 • Examples: Activating an MTR Topology by Using OSPF, page 53 • Examples: MTR OSPF Topology in Interface Configuration Mode, page 58 The following commands were introduced or modified: **address-family ipv4**, **area capability default-exclusion**, **ip ospf cost**, **ip ospf topology disable**, **priority**, **router ospf**, **show ip ospf interface**, **show ip ospf topology-info**, **topology**.

Table 1 Feature Information for Multi-Topology Routing (continued)

Feature Name	Releases	Feature Information
QoS/MQC Support for MTR	12.2(33)SRB 15.0(1)S	This feature enables MTR traffic classification. Traffic classification is used to associate different classes of traffic with different topologies when multiple topologies are configured on the same router. A subset of DSCP bits is used to encode classification values in the IP packet header and mark the packet for classification. When MTR traffic classification is enabled, MTR is activated and ready for the routing protocols to start contributing to the topologies. The following sections provide information about this feature: • MTR Traffic Classification, page 7 • Configuring MTR Traffic Classification, page 19 • Examples: MTR Traffic Classification, page 51 The following commands were introduced or modified: **policy-map type class-routing ipv4 unicast**, **select topology**, **service-policy type class-routing**, **show mtm table**, **show policy-map type class-routing ipv4 unicast**.
SNMP Support for MTR	12.2(33)SRB 12.2(33)SB 15.0(1)S	Context-based SNMP functionality has been integrated into Cisco IOS software and can be used to support MTR. SNMP support for MTR leverages context-based SNMP to extend support for existing MIBs from representing the management information for just the base topology to representing the same information for multiple topologies. The following sections provide information about this feature: • Network Management Support for MTR, page 11 • Configuring SNMP Support for MTR, page 40 • Examples: SNMP Support for MTR, page 60 The following commands were introduced or modified: **show snmp context mapping**, **snmp context**.

Glossary

base topology—The entire network for which the usual set of routes are calculated. This topology is the same as the default global routing table that exists today without MTR being used.

class-specific topology—New topologies that are defined over and above the existing base topology; each class-specific topology is represented by its own RIB and FIB.

classification—Selection and matching of traffic that needs to be provided with a different treatment based on its mark. Classification is a read-only operation.

DSCP—DiffServ Code Point. Six bits in the ToS. (Two bits are now used for Explicit Congestion Notification.) These are the bits used to mark the packet.

incremental forwarding mode—Incremental forwarding mode is designed to support transitional or incremental deployment of MTR, where there are routers in the network that are not MTR enabled. In this mode, the router will look for a forwarding entry first in the class-specific FIB. If an entry is not found, the router will then look for the longest match in the base topology FIB. If an entry is found in the base topology FIB, the packet will be forwarded on the base topology. If a forwarding entry is not found in the base topology FIB, the packet is dropped.

marking—Setting a value in the packet or frame. Marking is a read and write operation.

multi-topology—Multi-topology means that each topology will route/forward a subset of the traffic as defined by the classification criteria.

NLRI—Network Layer Reachability Information.

strict forwarding mode—Strict forwarding mode is the default forwarding mode for MTR. Only routes in the topology specific routing table are considered. Among these, the longest match for the destination address is used. If no route containing the destination address can be found in the topology specific table, the packet is dropped.

TID—Topology Identifier. Each topology is configured with a unique topology ID. The topology ID is configured under the routing protocol and is used to identify and group NLRI for each topology in updates for a given protocol.